revenera™

Code Insight 2023 R2

Installation & Configuration Guide

Legal Information

Book Name: Code Insight 2023 R2 Installation & Configuration Guide

Part Number: RCI-2023R2-IG00

Product Release Date: May 2023

Copyright Notice

Intellectual Property

Restricted Rights Legend

Contents

1

Code Insight 2023 R2 Installation & Configuration Guide

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

The *Code Insight Installation & Configuration Guide* describes how to install and configure Code Insight for use at your site. The guide includes the following sections.

Table 1-1 ▪ Code Insight Installation & Configuration Guide Navigation Table

Topic	Content
Installing Code Insight	Instructions for preparing to install, installing, and starting Code Insight. The chapter also includes optional system administration tasks, such as how to run Code Insight as a Windows service, enable HTTPS, or enable a networking proxy server connection.
Configuring Code Insight	Instructions for Code Insight System Administrator tasks, such as scheduling Code Insight Electronic Updates, managing Code Insight users, defining global project defaults and the scan profiles associated with projects, specifying the CVSS version, and configuring an email server for Code Insight notifications. The chapter also provides optional administrative procedures such as configuring Code Insight for LDAP and single sign-on and more.
Integrating with Source Code Management	(Required if you intend to synchronize with a remote source-code management component, or SCM, to obtain data to scan) Steps to ensure that the SCM command-line client is properly installed on the Code Insight Scan Server and that connectivity between the SCM client and the SCM server is properly configured.

Table 1-1 ▪ Code Insight Installation & Configuration Guide Navigation Table (cont.)

Topic	Content
Integrating with Application Lifecycle Management	Description of how to create one or more Jira connector instances, enabling Code Insight users to create, manage, and track external Jira work items associated with OSS or third-party inventory directly from Code Insight.
Upgrading Code Insight	Steps on how to upgrade from a previous Code Insight version to the current version.
Code Insight User Roles and Permissions	A reference to the various user roles and permissions that Code Insight offers as a means to control user access to its functionality at your site.

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. You can also access the Product and License Center from the Community's **Other Resources** menu to download Revenera products.

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

Installing Code Insight

This section contains the following topics covering the installation and startup of Code Insight:

- System Requirements
- Preparing to Install Code Insight
- Installing Code Insight
- Opening the Code Insight Web UI
- Starting and Stopping Tomcat
- Running Code Insight as a Service
- Enabling Secure HTTP Over SSL
- Configuring a Networking Proxy Server Connection
- Installing the Compliance Library
- Keeping Exact-Match Data Up to Date
- Upgrading the JRE
- Uninstalling Code Insight

System Requirements

Before installing Code Insight, ensure that the following requirements are addressed for your system:

- A supported database instance and its associated connector. See Database Support for a description of supported databases and connectors.
- A Code Insight license key file (`codeinsight.key`).
- On Linux instances, ensure that the number of open file handles is greater than 50k, a value typically set with the `ulimit` command. For more information about the open file limit, see Setting the Open-File Limit for Linux/Unix.

> **Important ▪** *This requirement for the open file limit is absolutely essential for Code Insight to function properly on Unix and Linux platforms.*

- Any requirements specific to your Code Insight plugin and remote data source. Refer to the *Code Insight Plugins Guide* for details.

The following sections describe additional requirements:

- Platform Support
- Java Runtime Edition Requirement
- Database Support
- Browser Support
- Recommended Hardware for Deployment Configurations
- Recommended Software

Platform Support

Code Insight supports the following platforms:

- CentOS 6.*x*, 7.*x,* 8.*x*
- RHEL 6.*x*, 7.*x*, 8.*x*
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012
- Ubuntu 18.04.*x*, 20.04 LTS

Java Runtime Edition Requirement

Code Insight requires the Java Runtime Edition (JRE) and officially supports Oracle JRE 8u192, 8u301, and 8u311. (All other versions under 8u311 are supported, but unofficially.) Currently, Oracle JRE 8u192 is installed as part of the Code Insight installation; a separate download is not necessary. If you want to upgrade the Oracle JRE version currently installed with Code Insight, you must perform additional steps once the Code Insight installation is complete. For more information, see Upgrading the JRE.

Alternatively, you can have Code Insight use to your own system installation of the Oracle JRE as long as it is a version supported by Code Insight. To configure Code Insight to use your own JRE, simply update the JAVA_HOME and JRE_HOME variables in `<codeInsightInstallation>/tomcat/bin/catalina.sh` (or `catalina.bat`) with the JRE path once the Code Insight installation has completed.

Database Support

Code Insight requires that either a MySQL or SQL Server database be installed. The following lists components required to install and configure a database for use by Code Insight:

- MySQL Required Components

- SQL Server Required Components

MySQL Required Components

The following describes the components needed to install and run MySQL as the Code Insight database:

- The community edition of MySQL 8.0 (also known as MySQL 5.8) or MySQL 5.7, downloaded from https://dev.mysql.com/downloads/mysql.

  > **Note •** *Code Insight does not support the Docker version of My SQL. (It supports the native version only.)*

- The appropriate JDBC driver file **that** enables Code Insight to connect to the your MySQL database version. Download this driver from the specified site and store it in a location accessible to the Code Insight installer.

  - For MySQL 8.0, use the latest version of `mysql-connector-java-8.0.x.jar`. Download this file from https://dev.mysql.com/downloads/connector/j/.

  - For MySQL 5.7, use `mysql-connector-java-5.1.x-bin.jar`. Download this file from http://dev.mysql.com/downloads/connector/j/5.1.html.

  During installation, the installer will copy the downloaded driver to the `tomcat/lib` folder, where it must reside for use by Code Insight.

- An environment that can support the required size settings listed in Required MySQL Database Settings.

- A database instance configured with the settings described in Required MySQL Database Settings and Binary Logging Option for MySQL.

SQL Server Required Components

The following lists the required components needed to install and run SQL Server as the Code Insight database:

- SQL Server 2019 (recommended for best performance) or 2016 Sp2.

- The JDBC driver file, `mssql-jdbc-6.4.0.jre8.jar`, which enables Code Insight to connect to your SQL Server database. Download this driver from https://www.microsoft.com/en-us/download/details.aspx?id=56615 and store in a location accessible to the Code Insight installer.

  During installation, the installer will copy the downloaded driver to the `tomcat/lib` folder, where it must reside for use by Code Insight.

- The package `sql_server_pre_install_scripts.zip` containing the scripts needed to set up the SQL Server database for Code Insight. See Downloading the Scripts Needed to Set Up the SQL Server Database for instructions on the download process.

- At least one disk (OS or non-OS) with 100 GB free space.

Downloading the Scripts Needed to Set Up the SQL Server Database

Use the following steps to download the package containing the script files needed to set up the SQL Server database for Code Insight.

Task   ***To download the package containing the scripts, do the following:***

1. Log into the Customer Community page of the Revenera website:

   https://community.flexera.com/t5/Revenera-Community/ct-p/Revenera_Community

2. From the **Other Resources** menu, select the **Product and License Center** option.

   Note ▪ This option is available only if you are a Product and License Center user or administrator. See Revenera Support for obtaining appropriate Product and License Center permissions.

3. From the **Product and License Center** page, select **Downloads**; and, from the list of available downloads, select **Code Insight**. (Alternatively, you might be able to select **Code Insight** directly from the **My Downloads** list on the **Product and License Center** page.)

4. Select the version of Code Insight from the list. The **Downloads** page appears.

5. Click the **sql_server_pre_install_scripts.zip** link to download the SQL Server scripts.

6. When the download finishes, extract the following files to a location accessible for later execution using the SQL Server console, as described in Setting Up the SQL Server Database:

   - codeinsight_serversettings.sql

   - codeinsight_db_creation_with_maintainenceplan.sql

   A third script, codeinsight_db_drop_with_maintainenceplan.sql, is used to drop the database and is *not* used as part of the database setup. Instructions for dropping the database are found in Dropping the SQL Server Database.

Browser Support

Code Insight supports the following browsers:

- Chrome (latest stable version)

- Internet Explorer (latest stable version)

- Firefox (latest stable version)

*Note ▪ Code Insight no longer allows uppercase or mixed case when entering the application's URL. To start Code Insight in a browser, you must enter **codeinsight** in lowercase.*

Recommended Hardware for Deployment Configurations

The recommended deployments and configurations are explained in this section:

- Deployment Entities

- Rules and Guidelines for Deployment Configurations

- Supported Deployment Configurations

Deployment Entities

Code Insight deployment can be configured on a single instance or on multiple instances. Each deployment consists of the following elements:

Table 2-1 ▪ Deployment Elements

Entity	Description
Core Server	Main interface to Code Insight.
Scan Server	(Required for local scans only, not for remote scans) Contains the Scan Server and the codebases to be scanned. Multiple Scan Servers are supported.
Database	Central database containing all library metadata supplied by the Electronic Update and all stored scan results.
Compliance Library (CL) (Optional)	Library containing all the data required to perform source-code fingerprint (snippet) matching and exact-file matching. The Scan Server must have access to the CL through a mapped or mounted drive.

Rules and Guidelines for Deployment Configurations

Your Code Insight configuration deployment should adhere to the following rules and guidelines. Keep these in mind as you determine the appropriate configuration for your site, as described in the next section, Supported Deployment Configurations:

- (Recommended) Use the Single Instance configuration (see the table), in which the Core Server, Scan Server, database, and Compliance Library (CL) are installed on the same instance.

- (Strongly recommended in a multiple-instance configuration) Use instances that are geographically close to each other. Otherwise, you might experience degradation in performance.

- If installing multiple Scan Servers, install only one Scan Server on a given instance. For exceptions, contact support for Code Insight through the Revenera Community (see Revenera Support).

- If installing multiple Scan Servers, consider installing the Core Server and the first Scan Server on the same instance and then each additional Scan Server on separate instances. This is a common configuration but not a required one.

- Ensure that the Core Server and each Scan Server belong to the same Code Insight version.

- Ensure that the instances hosting the Core Server and Scan Servers all use the same operating-system platform.

- If using the CL, install it on the same instance as the Scan Server, but on a drive or volume different from the one on which the Scan Server resides. When installing multiple Scan Servers, install the CL on each instance hosting a Scan Server.

- Installing on NFS/Shared drives is not recommended. Performance significantly degrades when Code Insight or the database is installed on an NFS/Shared drive. The recommendation is to install on a fast-spinning disk (minimum 7200 RPM) or a Solid State Drive (SSD) drive to optimize Code Insight scan performance.

Supported Deployment Configurations

The following table shows the various deployment configurations for various Code Insight entities.

Table 2-2 ▪ Supported Deployment Configurations

Configuration	CPU (Cores)	Memory	Disk Space
Single Instance (highly recommended): Core Server Scan Server Database Compliance Library (CL)	2-CPU (each at least 2 GHZ+) with 8+ cores on the instance	64 GB	**Server:** 500 GB High-speed Disk for the Database (SSD Recommended) 500 GB High-speed Disk for the Core/Scan Server to store the codebase 950 GB High-speed Disk for the CL
Instance 1: Core/Scan Server/Compliance Library (CL) **Instance 2:** Database	2-CPU (each at least 2 GHZ+) with 8+ cores on each server	**Instance 1:** 32 GB **Instance 2:** 32 GB	**Instance 1:** 500 GB High-speed Disk for Core/Scan Server to store the codebase 950 GB High-speed Disk for the CL **Instance 2:** 500 GB High-speed Disk for the Database (SSD Recommended)
Instance 1: Core Server **Instance 2:** Database **Instance 3:** Scan Server/ Compliance Library (CL)	2-CPU (each at least 2 GHZ+) with 8+ cores on each server	**Instance 1:** 32 GB **Instance 2:** 32 GB **Instance 3:** 32GB	**Instance 1:** 250GB High-speed Disk for Core Server **Instance 2:** 500 GB High-speed Disk for the Database (SSD Recommended) **Instance 3:** 500 GB High-speed Disk for Scan Server to store the codebase 950 GB High-speed Disk for the CL

Table 2-2 ▪ Supported Deployment Configurations (cont.)

Configuration	CPU (Cores)	Memory	Disk Space
Instance 1: Single Instance configuration (see the first table entry) **Instances 2 through x:** Scan Server/Compliance Library (CL)	2-CPU (each at least 2 GHZ+) with 8+ cores on each server	**Instance 1:** 64 GB **Instances 2 through x:** 32 GB	**Instance 1:** 500 GB High-speed Disk for the Database (SSD Recommended) 500 GB High-speed Disk for the Core/Scan Server to store the codebase 950 GB High-speed Disk for the CL **Each Instance 2 through x:** 500 GB High-speed Disk for Scan Server to store the codebase 950 GB High-speed Disk for the CL
Instance 1: Core/Scan Server/Compliance Library (CL) **Instance 2:** Database **Instances 3 through x:** Scan Server/Compliance Library (CL)	2-CPU (each at least 2 GHZ+) with 8+ cores on each server	**Instance 1:** 32 GB **Instance 2:** 32 GB **Instances 3 through x:** 32 GB	**Instance 1:** 500 GB High-speed Disk for Core/Scan Server to store the codebase 950 GB High-speed Disk for the CL **Instance 2:** 500 GB High-speed Disk for the Database (SSD Recommended) **Each Instance 3 through x:** 500 GB High-speed Disk for Scan Server to store the codebase 950 GB High-speed Disk for the CL
Instance 1: Core Server **Instance 2:** Database **Instances 3 through x:** Scan Server/Compliance Library (CL)	2-CPU (each at least 2 GHZ+) with 8+ cores on each server	**Instance 1:** 32 GB **Instance 2:** 32 GB **Instances 3 through x:** 32GB	**Instance 1:** 250GB High-speed Disk for Core Server **Instance 2:** 500 GB High-speed Disk for the Database (SSD Recommended) **Each Instance 3 through x:** 500 GB High-speed Disk for Scan Server to store the codebase 950 GB High-speed Disk for the CL

Recommended Software

The following software is recommended for Code Insight.

Database Client

A SQL client or command-line interface is necessary to run database scripts. The following free SQL clients are available:

- HeidiSQL: http://www.heidisql.com/download.php

- MySQL Workbench: http://www.mysql.com/products/workbench/

Preparing to Install Code Insight

Installing Code Insight is a simple, prompt-driven process, but before beginning the installation, you will need to do the following:

- Ensure that you have met the prerequisites in System Requirements.

- Follow the procedure in Setting Up the Database.

- Perform any additional environmental and communication configuration for Code Insight, such as the following:

  - Network and Firewall Considerations

  - Setting the Open-File Limit for Linux/Unix

Setting Up the Database

Before you install Code Insight, a database administrator must set up the MySQL or SQL Server database for use by Code Insight:

- Setting Up the MySQL Database

- Setting Up the SQL Server Database

Setting Up the MySQL Database

The following topics describe how configure the MySQL database for Code Insight:

- Setting Up a MySQL Instance for Code Insight

- Required MySQL Database Settings

- Updating the MySQL Database Settings

- Binary Logging Option for MySQL

- Sample Procedure for Creating an Appropriate Database Schema and User

Setting Up a MySQL Instance for Code Insight

The database administrator needs to perform the following steps to set up the MySQL database for Code Insight.

Task   ***To set up the MySQL database for Code Insight, do the following:***

1. Install the MySQL instance.

    You might need to configure certain settings once the installation is complete.

    Note • *Installing the instance on a machine other than the one on which the Code Insight Core Server is installed might cause performance degradation.*

2. Create a database schema (with a recommended name of *codeinsight*) and a database user who has appropriate privileges to install Code Insight and whom Code Insight will use internally to manage the database. At a minimum, this user requires the following permissions: ALTER, DROP, CREATE, DELETE, INDEX, INSERT, and UPDATE. The procedure described in Sample Procedure for Creating an Appropriate Database Schema and User can be used to perform these tasks.

3. Configure the instance as described in Required MySQL Database Settings and Binary Logging Option for MySQL

Required MySQL Database Settings

Code Insight requires the MySQL database configuration described in this table to ensure best performance.

Table 2-3 • Required MySQL Database Settings

Property	System Variable	Recommended Value
Storage Engine	default-storage-engine	**innodb**
Character Set	character-set-server	**utf8mb4** (required value for all supported MySQL versions)
Collation	collation-server	**utf8mb4_unicode_ci** (required value for MySQL 5.7) **utf8mb4_0900_ai_ci** (required value for MySQL 8.0)
InnoDB Buffer Pool Size	innodb_buffer_pool_size	**12G** (12 GB)
InnoDB Log File Size	innodb_log_file_size	**8G** (8 GB)
Maximum Allowed Packets	max_allowed_packet	**100M** (100 MB)

If you need to verify the current settings in your MySQL installation, click the appropriate **Property** link in the table for a description of the verification command. If you need to change settings in your installation, use the next procedure.

Updating the MySQL Database Settings

Use the following procedure to update settings for the MySQL database. Refer to Required MySQL Database Settings when making the changes. Also see the subtopics that follow the procedure.

Task   ***To configure the MySQL database, do the following:***

1. Within your MySQL installation, do one of the following:

    ● As root user in Linux, open the `my.cnf` file (typically located in `/etc/`).

    ● As a Windows administrator, open `my.ini` file (typically located in `C:\ProgramData\MySQL\MySQL Server` *version*).

2. Edit the settings as shown in the previous table. (If necessary, click the appropriate **Property** link in the table for a description of how to configure a given setting.)

3. After you have updated the settings, restart the database server.

Storage Engine

Specify **InnoDB** for the **default-storage-engine** property. By default in MySQL, **InnodDB** is already specified for this property, so you most likely will not need to change it.

To verify the current default storage engine, use the following command:

```
SELECT * FROM INFORMATION_SCHEMA.ENGINES;
```

If you need to add the **default-storage-engine** property (or update the current value of this property), use the appropriate procedure:

● In Linux, add the following line to the `[mysqld]` section of the `my.cnf` file (or update the existing property value):

```
default-storage-engine=innodb
```

● In Windows, add the following line to the `[mysqld]` section of the `my.ini` file (or update the existing property value):

```
default-storage-engine=innodb
```

Character Set

Specify **utf8mb4** for the **character-set-server** property when installing the MySQL database server for Code Insight. (This value is applied at the database/schema level.)

*Important ▪ Ensure that the **character-set-server** value is set to **utf8mb4**. Any other value has been know to produce undesirable results during a scan, forcing users to have to set up the database again since no rollback options are available. As protection, the Code Insight installer will not proceed with the installation once it detects a value other than **utf8mb4** for the **character-set-server** property in the database.*

To verify the current character set, use the following command:

```
SELECT @@character_set_database;
```

If you need to add the **character-set-server** property (or update the current value to the required value), use the appropriate procedure:

- In Linux, add the following line to the `[mysqld]` section of the `my.cnf` file (or update the existing property to the required value):

```
character-set-server=utf8mb4
```

- In Windows, add the following line to the `[mysqld]` section of the `my.ini` file (or update the existing property to the required value):

```
character-set-server=utf8mb4
```

Collation

Select **utf8mb4_unicode_ci** for the **collation-server** property when installing the MySQL database server for Code Insight. (This value is applied at the database/schema level.)

*Important ▪ Ensure that the **collation-server** value is set to **utf8mb4_0900_ai_ci** (for MySQL 8.0) or **utf8mb4_unicode_ci**.(for MySQL 5.7). Any other value has been know to produce undesirable results during a scan, forcing users to have to set up the database again since no rollback options are available. As protection, the Code Insight installer will not proceed with the installation once it detects a value other than **utf8mb4_unicode_ci** for the **collation-server** property in the database.*

To verify the current collation, use the following command:

```
SELECT @@collation_database;
```

If you need to add the **collation-server** property (or update the current value to the required value), use the appropriate procedure:

- In Linux, add the following line to the `[mysqld]` section of the `my.cnf` file (or update the existing property to the required value):

```
collation-server=utf8mb4_unicode_ci
```

- In Windows, add the following line to the`[mysqld]` section of the `my.ini` file (or update the existing property to the required value):

For MySQL 8.0: `collation-server=utf8mb4_0900_ai_ci`

For MySQL 5.7: `collation-server=utf8mb4_unicode_ci`

InnoDB Buffer Pool Size

Set the **innodb_buffer_pool_size** property to *at least* 12G (gigabytes).

To verify the current InnoDB buffer pool size, use the following command. The returned value is in gigabyte (G) units.

```
SELECT @@innodb_buffer_pool_size/1024/1024/1024;
```

If you need to add the **innodb_buffer_pool_size** property (or update the current value of this property), use the appropriate procedure:

- In Linux, add the following line to the [mysqld] section of the my.cnf file (or update the existing property value):

```
innodb_buffer_pool_size=12G
```

- In Windows, add the following line to the [mysqld] section of the my.ini file (or update the existing property value):

```
innodb_buffer_pool_size=12G
```

InnoDB Log File Size

Set the **innodb_log_file_size** property to *at least* 8G (gigabytes).

To verify the current InnoDB log file size, use the following command. The returned value is in gigabyte (G) units.

```
show variables like 'innodb_log_file_size%';
```

If you need to add the **innodb_log_file_size** property (or update the current value of this property), use the appropriate procedure:

- In Linux, add the following line to the [mysqld] section of the my.cnf file (or update the existing property value):

```
innodb_log_file_size=8G
```

- In Windows, add the following line to the [mysqld] section of the my.ini file (or update the existing property value):

```
innodb_log_file_size=8G
```

Maximum Allowed Packets

Set the **max_allowed_packet** property to **100M** (megabytes).

To verify the current maximum packet size, use the following command. The returned value is in megabyte (M) units.

```
SHOW VARIABLES LIKE 'max_allowed_packet';
```

If you need to add the **max_allowed_packet** property (or update the current value of this property), use the appropriate procedure:

- In Linux, add the following line to the [mysqld] section of the my.cnf file (or update the existing property value):

```
max_allowed_packet=100M
```

● In Windows, add the following line to the [mysqld] section of the my.ini file (or update the existing property value):

```
max_allowed_packet=100M
```

Binary Logging Option for MySQL

MySQL offers Binary Logging, an advanced feature that captures changes between backups and stores this information in binary log files. The log files—containing information about each statement that modified (or might have modified) the database and the amount of time it took to make the change—are mainly used for data recovery and replication efforts. The files reside in either location:

● The /var/lib/mysql directory on Linux

● The program data directory on Windows (for example, C\ProgramData\MySQL\MySQL Server 8.0\Data)

Possible Issues with Binary Logging

Binary Logging can cause issues when a Code Insight Electronic Update or scan is run. During either of these processes, the database can be updated with a significant number of insert, update, and delete events. With Binary Logging enabled, details for each event are also written to rolling binary log files, with each file being about 1 GB in size. If these log files are not purged regularly, out-of-memory issues will occur.

The user can decide whether Binary Logging should be enabled.

Disabling or Enabling Binary Logging

By default, Binary Logging is enabled in MySQL8, but disabled in MySQL5.

Task   ***To disable or enable Binary Logging, do the following:***

1. Within your MySQL installation, do one of the following:

   ● As root user in Linux, open the my.cnf file (typically located in /etc/).

   ● As a Windows administrator, open my.ini file (typically located in C:\ProgramData\MySQL\MySQL Server *version*).

2. To enable Binary Logging, add or uncomment the log-bin line:

```
#Binary Logging
log-bin="<binaryLogBaseName>"
```

   or

   To disable Binary Logging, comment-out the log-bin line:

```
#Binary Logging
#log-bin="<binaryLogBaseName>"
```

3. Restart the database server.

Sample Procedure for Creating an Appropriate Database Schema and User

The following is a sample procedure that the database administrator can use to create a Code Insight database schema and the database user who will install Code Insight. Additionally, at installation, this same user is automatically identified to Code Insight (in `core.db.properties`) as the user Code Insight will use internally to manage the database. At a minimum, this user requires the following permissions: ALTER, DROP, CREATE, DELETE, INDEX, INSERT, and UPDATE.

Task   ***To create a database schema and user, do the following:***

1. At the command line, log into MySQL as the root user:

    ```
    mysql -u root -p
    ```

2. Type the MySQL root password, and press **Enter**.

3. To create a database and user, type the following command, replacing the username (*codeinsight_user)* with the user you want to create, and replace *codeInsight%1234* with the user's password:

    ```
    CREATE DATABASE codeinsight;
    CREATE USER codeinsight_user IDENTIFIED BY 'codeInsight%1234';
    GRANT ALL ON codeinsight.* TO 'codeinsight_user'@'%';
    ```

4. Provide the database schema and user credentials to the person who will install Code Insight.

Setting Up the SQL Server Database

Setting up the SQL Server database for Code Insight involves two phases:

● Phase 1: Install the SQL Server Instance

● Phase 2: Set Up the SQL Server Database for Code Insight

● Note about Running the Code Insight Maintenance Jobs on SQL Server Databases

Phase 1: Install the SQL Server Instance

A database administrator can perform these steps.

Task   ***To install the SQL Server instance, do the following:***

1. Install the SQL Server instance, following the instructions included with the SQL Server installer. During the installation, select the appropriate options that do the following:

    ● Set the character set (or collation) is to `SQL_Latin1_General_CP1_CI_AS`.

    ● Enable the SQL Server Agent.

2. When the installation is complete, start up the SQL Server Agent using the instructions provided in the SQL Server documentation. This a required step for setting up the SQL Server database, described in the next section, Phase 2: Set Up the SQL Server Database for Code Insight.

Phase 2: Set Up the SQL Server Database for Code Insight

Once the SQL Server instance has been installed and the SQL Server Agent started, the DBO performs the following steps to set up the database for Code Insight.

Task **To set up the SQL Server database for Code Insight, do the following:**

1. Ensure that you have downloaded and extracted the required the Code Insight scripts, as described in Downloading the Scripts Needed to Set Up the SQL Server Database.

   The following is a brief description of the scripts:

   - **codeinsight_serversettings.sql**—This script configures the database server to enable the maximum performance for Code Insight. The script sets the following server parameters:

     - **Cost of parallelism**—15 (the threshold at which the optimizer chooses parallel processing)

     - **Max degree of parallelism**—Number of threads created specifically for this configuration.

     - **Max memory configuration**—The server's maximum utilization (60 percent) of total memory.

     - **TF**—Trace flags 111, 1118, 2371.

     You are strongly recommended to review existing configurations in this script and note their values in case a rollback is needed. However, do not edit this script.

   - **codeinsight_db_creation_with_maintainenceplan.sql**—This script creates the database and schedules maintenance jobs. Specifically, it performs the following operations:

     - Creates a database with 4 data files and 1 log file.

     - Creates a new folder called MSSQLDATA on a non-OS disk. If only one drive exists, the database is created on the OS drive itself.

     - Creates a subfolder with the database name under the MSSQLDATA folder.

     - Creates a daily maintenance job to perform an Update Statistics every 6 hours (no downtime needed).

     - Creates maintenance job to perform an Update Statistics and Index Reorg every two weeks (no downtime needed). The default is to run at 10 pm per server time zone every two weeks.

     You can edit some settings in this script as described in Step 5.

2. Ensure that the SQL Server Agent is running.

3. Open the codeinsight_serversettings.sql script, and execute it.

   Do not edit this script.

4. Open the codeinsight_db_creation_with_maintainenceplan.sql script, edit the @dbname setting if necessary, and then execute the script.

   The default value for @dbname is **fnciv7**. To edit this setting, simply overwrite the current value with the preferred database name. If you provide a database name that already exists, the script execution will fail.

5. Create the user who will install Code Insight. At installation, this same user is automatically identified to Code Insight (in core.db.properties) as the user Code Insight will use internally to manage the database.

6. Assign this user at least the minimally required permissions: ALTER, DROP, CREATE, DELETE, INDEX, INSERT, and UPDATE.

Note ▪ Code Insight uses this same user to migrate the database during a Code Insight upgrade. If you intend to upgrade Code Insight in the future, this user must have the db_ddladmin role to perform the migration. However, you can add this role at the time of the upgrade and then revoke it once the upgrade is complete.

Note about Running the Code Insight Maintenance Jobs on SQL Server Databases

You are strongly recommended *not* to execute any service related to Code Insight (for example, an Electronic Update or a scan) or any other job against the SQL Server database while a Code Insight maintenance job is running on the database. If you do run another process at the same time as a Code Insight Maintenance job, expect some delay in that process. Additionally, Code Insight might experience performance-related issues or unexpected behavior.

Network and Firewall Considerations

If the Code Insight Core Server, Scan Server, or plugin is behind a firewall, you need to configure the firewall to ensure that each server or plugin has access to Code Insight:

● Server Identification

● Code Insight Ports

● External URLs

Server Identification

In all firewalls, specify either of the following to identify the instance on which you are installing the Code Insight Core Server, Scan Server, or plugin:

● A fully qualified domain name (for example, *hostname.domain.com*)

● An IP address (static IP address recommended)

Code Insight Ports

In all firewalls, enable port numbers used by Code Insight. You can use the default port numbers listed below or configure the application to use custom ports.

Table 2-4 ▪ Default Port Numbers Used by Code Insight

Port #	Details
3306	MySQL database server access port
1433	SQL Server database server access port

Table 2-4 ▪ Default Port Numbers Used by Code Insight (cont.)

Port #	Details
8888/443	Tomcat (http/https, respectively)
465	External SMTP (mail) server
389	External authentication directory server (Active Directory/LDAP)
8005 and 8009	Tomcat Connector and Tomcat shutdown ports, respectively (local access only)

External URLs

In all firewalls, provide access to the following external host URLs needed by Code Insight:

Table 2-5 ▪ External Host URLs Used by Code Insight

Code Insight Component/ Functionality	Hosts
CodeAware Analyzers	https://api.nuget.org/v3-flatcontainer/
	https://packagist.org
	https://proxy.golang.org/
	https://pypi.org/pypi/
	https://registry.bower.io/packages/
	https://registry.npmjs.org/
	https://rubygems.org/api/v1/gems/
	https://search.maven.org/
	https://spdx.org/licenses/
Vulnerability database access	https://nvd.nist.gov
	https://web.nvd.nist.gov/
	https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator
	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
Electronic Update	https://updates.palamida.com/
Remote file access	https://palamida-dp-nbhood.s3.amazonaws.com/
Notices text	https://login.flexera.com/
	https://sca-api.revenera.com/

Special Note About Custom Repositories

When Code Insight processes a `pom.xml` file, it looks for declarations of direct dependencies (that is, OSS or third-party components on which a top-level component detected in your product code is dependent). Code Insight then reports these direct dependencies as inventory items linked to the top-level (parent) inventory item.

If transitive-dependency support is enabled in the scan profile, Code Insight calls into the Maven Central repository for each direct dependency to determine whether the dependency is resolvable. If it is resolvable, Code Insight then searches the Maven Central repository to obtain the list of transitive dependencies (that is, dependencies of the dependency), provided that these dependencies are not available in local maven repository.

In some cases, you might have provided repositories that are in addition to the Maven Central repository. If a direct dependency is not resolvable through the Maven Central repository (or if its transitive-dependency data is found neither in the local maven repository nor in the Maven Central repository), Code Insight might call into these other repositories to determine the dependency's resolvability or to obtain its transitive-dependency data or both. For this reason, if you have configured custom repositories as part of your package manager manifest files, the URLs for those repositories need to be a part of your list of allowed external URLs.

Setting the Open-File Limit for Linux/Unix

The open-file limit is a setting that controls the maximum number of open files for individual users. The default open-file limit is typically 1024, but can be set with the `ulimit` command by the root user. For Code Insight to function properly in a Linux or Unix environment, the open-file limit must be set to handle more than 50K files on each instance hosting the Core Server or a Scan Server.

Important • Increasing the open-file limit is absolutely essential for Code Insight to function properly on Unix/Linux platforms.

When *not* running Code Insight as a service, you must use the procedure described here to set the open-file limit for individual Code Insight users or groups. If you *do* intend to run Code Insight as a service, you must set the open-file limit at the service level, using the procedure described in Opening the Code Insight Web UI. Best practice is to also set the open-file limit at the user or group level, as described here, should situations arise where Code Insight is not run as a service.

The following are types of open-file limits:

- **soft limit**—Set in `/etc/security/limits.conf` by a normal user.

- **hard limit**—Set in `/etc/security/limits.conf` by root user.

- **system wide limit**—Set in `/etc/sysct1.conf` by root user.

Soft limits are the currently enforced limits; hard limits are the maximum limits on the system. The following procedure sets a soft and a hard open-file limit for user or group you specify. To run the procedure, you should log in as the root user so that you can set both limit types.

Task   ***To set open file limits on a Linux RedHat system, do the following:***

1. In a command-line window on your instance, type `ulimit -a` to see a list of current file limits.

2. Locate the *open files (-n)* setting:

    - If the setting is less than 50K, continue to the next step.

    - If the setting is more than 50K, you do not need to perform this procedure.

3. Open the file `/etc/security/limits.conf`, and add the following entries for each specific user or group as needed:

    ```
    <userName> soft nofile 65536
    <userName> hard nofile 65536
    ```

     or

    ```
    @<groupName> soft nofile 65536
    @<groupName> hard nofile 65536
    ```

    Alternatively, you can substitute `<userName>` or `@<group name>` with the wildcard * for a default entry:

    ```
    * soft nofile 65536
    * hard nofile 65536
    ```

4. Save the file and log in again for the changes to take effect.

5. On the command line, type `ulimit -a`, and verify that the *open files (-n)* setting reads 65536.

Note ▪ *Other distributions, such as a Ubuntu and CentOS, might require a different setting. See instructions for your specific Linux distribution and shell type.*

Installing Code Insight

Use the following instructions to install Code Insight. You have the option to launch the Code Insight installer with a UI that walks you through the installation process. Alternatively, you can run a silent installation, which involves no user interaction during the installation process but instead uses a response file to provide your installation input.

- Information to Collect Before Running the Installer

- Launching the Code Insight Installer UI

- Performing a Silent Installation

- Special Post-Installation Step: Configuring the MySQL SSL Option in Code Insight

Information to Collect Before Running the Installer

When you have met the requirements listed in System Requirements and are ready to install Code Insight, best practice is to collect information required by the installer before starting the installation. The following is basic information you will need to provide the installer:

- Types of Code Insight Deployment

- License Key and JDBC Information

Types of Code Insight Deployment

The following are the types of Code Insight deployment that the installer can perform:

- **Standalone**—Code Insight is deployed as both the Core Server and a Scan Server. This is the recommended baseline configuration. If you are installing additional Scan Servers, the recommendation is to install the first Scan server using the **Standalone** configuration. Then install the additional Scan Servers on separate instances using the **Scanner** configuration. All Scan Servers must point to the same database.

- **Core**—Code Insight is deployed as the Core Server only.

- **Scanner**—Code Insight is deployed as a Scan Server only. To install multiple Scan Servers, run this installation on each instance that you want to designate as a Scan Server. (The recommendation is that only one Scan Server be installed on a given a instance.) For more information about managing Scan Servers once they are installed, see Adding or Editing Scan Servers or Checking Server Status. All Scan Servers should point to the same database.

The Core Server controls the Code Insight Web UI Client. The Scan Server is where actual scanning is performed. (Note that a Scan Server has no Web UI capabilities.)

License Key and JDBC Information

The following is a list of information that the installer will require during a given installation.

- The location of the license key file, codeinsight.key. If you do not have a license key file, see Revenera Support for instructions on obtaining support for Code Insight through the Revenera Community.

- The name and path of the JDBC driver used to connect Code Insight with the SQL Server or MySQL database. Prior to installation, the driver must be downloaded to a location accessible to the Code Insight installer before installation starts. For more information, refer to either MySQL Required Components or SQL Server Required Components, depending on your database type.

  The Code Insight installer will copy the downloaded driver to your tomcat/lib folder during installation.

Launching the Code Insight Installer UI

After you have met the System Requirements, including creating a database with remote access privileges, and have collected required installer information, you are ready to run the Code Insight Installer to install the Code Insight Core Server and one or more Scan Servers. You will need to run a separate installation on each instance on which you want to install the servers, depending on your server configuration, as described in Recommended Hardware for Deployment Configurations.

You can cancel the installation by clicking **Cancel** on any installation panel.

Note ▪ The user who installs Code Insight should not have elevated privileges because this same user also installs and starts Tomcat, which should not run under elevated privileges.

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

1. Download the Code Insight installer from the Product and License Center:

    -   For Windows, `CodeInsight-<BUILD>.exe`

    -   For Linux, `CodeInsight-<BUILD>.bin`

2. Navigate to the directory where you downloaded the installer, and launch the installer.

3.   Follow the prompts to install Code Insight.

4. When the installation is complete, do the following:

    a. Start the Tomcat server if it is not already running. See Starting and Stopping Tomcat.

        The recommended best practice is *not* to run Tomcat under elevated privileges.

    b. If Code Insight is not already running, use the procedures in Opening the Code Insight Web UI to launch the application.

Important ▪ If the installation does not complete properly, contact support for Code Insight (see Revenera Support).

Performing a Silent Installation

Code Insight can be installed in silent mode, which involves no user interaction with Code Insight installer. Instead, once you launch the installer, it reads a response file to obtain all the details necessary for executing the installation. You are responsible for providing this file, which you can automatically generate from user input captured during an installation using the installer UI.

Note ▪ The user performing the silent Code Insight installation should not have elevated privileges as this same user also installs and starts Tomcat as part of the installation—and Tomcat should not run under elevated privileges.

The following sections provide details about running the silent installation:

-   Downloading the Code Insight Installer

-   Creating a Response File for the Silent Installation

-   Running a Silent Installation

-   Installation Properties in the Response File

Downloading the Code Insight Installer

To perform a silent installation, you must download the Code Insight installer.

Task   ***To download the installer, do the following:***

Access the Product and License Center through the Revenera Community, and download the Code Insight installer appropriate for your operating system:

● For Windows, `CodeInsight-<BUILD>.exe`

● For Linux, `CodeInsight-<BUILD>.bin`

Creating a Response File for the Silent Installation

A Code Insight silent installation requires a response file that the installer reads to determine the installation parameters. Best practice is to create this file through a command-line option that automatically generates the response file from the user input captured by the installer UI during an actual installation. The file is saved as `installer.properties` and resides in the same directory as the installer. You can then use this response file for future silent installations.

Task   ***To generate the response file for a silent installation, do the following:***

1. Launch the Code Insight installer using the following command.

   Important ▪ *The -r command-line option requires at least one argument. To meet this requirement, specify the destination path to use the default name or specify a full path and file name to rename the file.*

   `<CodeInsightInstallerName>.[bin/exe] -r <InstallerFilePath>|<InstallerFilePath/FileName>`

   For example:

   `CodeInsight-7.17.0-98.exe -r C:\CodeInsight_Installer`

2. Complete the required fields as you walk through the installer UI.

   Once the installation is complete, the response file is generated with the default name `installer.properties` and is saved to the installer directory. If you stopped the installer at any point, a response file is still generated with whatever inputs were entered up to the stopping point.

   For a description of the properties in the file, see Installation Properties in the Response File. The file can be tweaked and renamed as needed for use by the silent installation.

   Note ▪ *The other default name recognized by the installer is `<CodeInsightInstallerName>.properties`, but you can give the response any name as long as you explicitly designate its name when you run the silent installation.*

Running a Silent Installation

The following sections provide details about the executing the Code Insight installation in silent mode:

- Basic Procedure for Executing a Silent Installation

- Example Commands for Running a Silent Installation

Basic Procedure for Executing a Silent Installation

Use the following basic procedure to run a Code Insight installation in silent mode.

Task   ***To execute a silent installation, do the following:***

1. Ensure that these steps are complete:

   - You have gathered the information listed in Information to Collect Before Running the Installer.

   - The Code Insight installer has been downloaded (see Downloading the Code Insight Installer).

   - The response file is created (see Creating a Response File for the Silent Installation).

2. At a command line, navigate to the directory containing the Code Insight installer.

3. Execute a command in the following format to run the silent installation:

   `<CodeInsightInstallerName>.[bin|exe] -i silent [-f <FileName>|<FilePath>|<FilePath/FileName>]`

   where

   - `<CodeInsightInstallerName>.[bin|exe]` is the name of the Code Insight installer (for example, `CodeInsight-0.00.0-00.bin`).

   - Either or both of these elements can be used with the `-f` command-line option to identify a response file that uses a *non-default* path or name.

     - `<FilePath>` identifies the path in which the response file resides.

     - `<FilePath/FileName` identifies the path and name of the response file.

   For command examples and more information about the path and name defaults for a response file, see Example Commands for Running a Silent Installation.

4. Once the installation is complete, do the following:

   a. If the installer did not automatically start Tomcat, use the procedure in Starting and Stopping Tomcat to start it.

      Best practice is *not* to run Tomcat under elevated privileges.

   b. If the installer did not automatically launch the Code Insight Web UI in a browser, use the procedure in Opening the Code Insight Web UI to do so.

Important • *If the installation does not complete properly, contact Revenera Support for assistance.*

Example Commands for Running a Silent Installation

The following provides some example variations of the command used to run an installation in silent mode.

Note that the `-f` command-line option uses an argument to identify a non-default file name or path (or both) for the response file. When the argument for either the file name or path is omitted (or the `-f` option is omitted altogether), the installer assumes that the default value for each omitted element is the following:

● For the file name, either `installer.properties` or `<CodeInsightInstallerName>`.`properties`.

● For the path, the directory in which the installer resides.

Neither the Response File Name Nor Path Explicitly Included

The following example command implies that the response file resides in the same directory as the Code Insight installer and has the default name `installer.properties` or `<CodeInsightInstallerName>.properties`.

```
CodeInsight-0.00.0-00.exe -i silent
```

File Name Explicitly Included

The following command uses a response file called `MyResponse.txt`, which resides in the same directory as the installer. The `-f` command-line option is used to identify the non-default file name.

```
CodeInsight-0.00.0-00.exe -i silent -f MyResponse.txt
```

Response File Path Explicitly Included

The following command uses a response file that resides in `C:\responseFiles` and has the default name `installer.properties` or `<CodeInsightInstallerName>.properties`. The `-f` command-line option is used to identify the non-default path, which can be either absolute or relative to the directory in which the Code Insight Installer resides.

```
CodeInsight-0.00.0-00.exe -i silent -f C:\responseFiles
```

Response File Path and Name Explicitly Included

The following command uses the response file `C:\propertiesFiles\MyResponse.txt`. The `-f` command-line option is used to identify this non-default file path and name. The file path can be designated as either absolute or relative to the directory in which the Code Insight Installer resides.

```
CodeInsight-0.00.0-00.exe -i silent -f C:\responseFiles\MyResponse.txt
```

Installation Properties in the Response File

This section provides a description of the properties in the response file used by the Code Insight silent installation. It also includes the contents of a sample response file for reference.

● Property Descriptions

● Sample Response File Content

Property Descriptions

Refer to the following table for a description of the properties that should be included in the response file.

Table 2-6 ▪ Installation Properties

Category	Property	Description
Installation directories		These properties identify the directory in the Code Insight installer resides and the directory in which to install Code Insight.

	INSTALLER_LAUNCH_ DIR	The path in which the downloaded Code Insight installer resides on your instance. (For more information, see Downloading the Code Insight Installer.)
		
		Examples:
		
		`INSTALLER_LAUNCH_DIR=/tmp/`
		`    codeinsight_install_files`
		
		`INSTALLER_LAUNCH_DIR=D:/CodeInsight/`
		`    InstallationFiles/2023R1`

	USER_INSTALL_DIR	The directory in which to install the Code Insight on your instance. Best practice is to provide an absolute path for the installation location.
		
		Examples:
		
		`USER_INSTALL_DIR=/opt/CodeInsight/2023R1`
		
		`USER_INSTALL_DIR=D:/CodeInsight/2023R1`

Database connector file		The following properties identify the path and name of downloaded JDBC driver that enables Code Insight to connect with the MySQL or SQL Server database. (For more information, see SQL Server Required Components or SQL Server Required Components.) During installation, the Code Insight installer copies the driver to `tomcat/lib` in your Code Insight installation directory.

	USER_JDBC_DRIVER	The name of the downloaded JDBC driver file for your database.
		
		Examples:
		
		`USER_JDBC_DRIVER=mysql-connector-java-8.0.21.jar`
		
		`USER_JDBC_DRIVER=mssql-jdbc-6.4.0.jre8.jar`

	PATH_OF_USER_JDBC_ DRIVER_FILE	The path of the downloaded JDBC driver file.
		
		Examples:
		
		`PATH_OF_USER_JDBC_DRIVER_FILE=/tmp/`
		`    codeinsight_install_files`
		
		`PATH_OF_USER_JDBC_DRIVER_FILE=D:/CodeInsight/`
		`    InstallationFiles/2023R1`

Table 2-6 ▪ Installation Properties (cont.)

Category	Property	Description

Database connector file (continued)	DRIVER_PATH	The path and name of the downloaded JDBC driver.
		
		Examples:
		
		`DRIVER_PATH=/tmp/codeinsight_install_files/mysql-`
		`    connector-java-8.0.21.jar`
		
		`DRIVER_PATH=D:/CodeInsight/InstallationFiles/`
		`    2021R4/mysql-connector-java-8.0.21.jar`

Name of the Code Insight license key		The following properties Identify the file name and path of the Code Insight license key that Revenera has provided you. If you do not have a license key file, contact Revenera Support.
	USER_PALAMIDA_KEY_ FILE	The name of the file containing the Code Insight license key. **Example:** `USER_PALAMIDA_KEY_FILE=codeinsight.key`

	PATH_OF_PALMIDAKEY_ FILE	The path of the file containing the Code Insight key.
		
		Example
		
		`PATH_OF_PALMIDAKEY_FILE=D:/CodeInsight/`
		`    InstallationFiles/2023R1`

Database configuration		Details about the database that Code Insight will use.

	SELECTED_DB	The database type that Code Insight will use: MYSQL or SQLSERVER.
		
		Examples:
		
		`SELECTED_DB=MYSQL`
		
		`SELECTED_DB=SQLSERVER`

	DB_HOST	The host ID for the database.
		
		Examples:
		
		`DB_HOST=localhost`
		
		`DB_HOST=http://ABC-r7mysql.com`
		
		`DB_HOST=http://11.11.1.111`

Table 2-6 ▪ Installation Properties (cont.)

Category	Property	Description
Database configuration (continued)	**DB_PORT**	The port used by Code Insight to communicate with the database. See Code Insight Ports. **Examples:** `DB_PORT=3306` (default for MySQL) `DB_PORT=1433` (default for SQL Server)
	EXISTING_SCHEMA	The name of the schema for the Code Insight database. **Example:** `EXISTING_SCHEMA=codeinsight`
	DB_USER	The name of the database user that Code Insight uses to access the database. The installation process automatically identifies this user (in `core.db.properties`) as the one that Code Insight will use internally to access and manage the database during Code Insight operations. **Example:** `DB_USER=codeinsight_user`
	DB_PASS	The password of the database user identified for DB_USER. **Example:** `DB_PASS=codeInsight%1234`
	SCHEMA_POPULATED_ WARNING	The option that determines whether the Code Insight database (if it already exists) is overwritten during the installation. If the installer determines that the value for this property is missing, it rolls back the installation. ● **0**—The installation overwrites the current database ● **1**—The installation does not overwrite the current database. ● *no value*—The installer rolls back the installation (that is, no installation occurs). **Example:** `SCHEMA_POPULATED_WARNING=0`

Table 2-6 ▪ Installation Properties (cont.)

Category	Property	Description
Type of Code Insight deployment		The following three fields specify how the installer should deploy Code Insight on the given instance. Use the same value for all three fields.

	CHOSEN_FEATURE_LIST
	
	CHOSEN_INSTALL_FEATURE_LIST
	
	CHOSEN_INSTALL_SET
		The type of Code Insight deployment that the installer should perform. For more information, see Types of Code Insight Deployment.
		
		• **Standalone**—Deploy Code Insight as both the Core Server and a Scan Server.
		
		• **Core**—Deploy the Code Insight as the Core Server only.
		
		• **Scanner**—Deploy the Code Insight as a Scanner only.
		
		Example:
		
		`CHOSEN_FEATURE_LIST=Standalone`
		`CHOSEN_INSTALL_FEATURE_LIST=Standalone`
		`CHOSEN_INSTALL_SET=Standalone`

Automatic startup after installation completes		These properties are used to automatically start Tomcat and launch Code Insight once the installation completes. Ignore the `ia.startTomcatFlagcmd` property that is included in the generated response. *Important* ▪ *Tomcat should not run under elevated privileges. Therefore, the user who installs Code Insight should not have elevated privileges because this same user also installs and starts Tomcat.*
	ia.startTomcatFlag	Specify **true** if you want the installer to automatically start Tomcat (which in turn starts Code Insight) once the installation completes. Otherwise, specify **false**. (Tomcat can be started manually as described in Starting and Stopping Tomcat.) If you want to installer to also launch the Code Insight Web UI in a browser, configure the next property. **Example:** `ia.startTomcatFlag=true`

Table 2-6 ▪ Installation Properties (cont.)

Category	Property	Description
Automatic startup after installation completes (continued)	**ia.startBrowserFlag**	If ia.`startTomcatFlag` is `true`, do either: ● Specify **true** for this option if you want the installer to automatically open the Code Insight Web UI in a browser once the installation completes. ● Specify `false` if you want to manually open the Code Insight Web UI in a browser after the installation completes. (To launch the Web UI in browser manually, see in Opening the Code Insight Web UI.) If ia.`startTomcatFlag` is `false`, you must both start Tomcat and open the Code Insight UI in a browser manually after the installation completes. **Example:** `ia.startBrowserFlag=true`
Variables		You can provide explicit values for the following properties. However, best practice is to set up these properties as variables since they are based on other values in the response file. Once these variables are defined, you are strongly advised not to modify them.
	CATALINA_HOME	The path of the Catalina home directory for Code Insight. **Variable:** `CATALINA_HOME=$USER_INSTALL_DIR$/tomcat`

	TOMCAT_STARTUP_FILE	The path and name of the Tomcat startup file.
		
		Variable:
		
		`TOMCAT_STARTUP_FILE=$CATALINA_HOME$/bin/`
		`    startup.bat`

	JAVA_HOME	The path of the Java home directory for Code Insight. **Variable:** `JAVA_HOME=$USER_INSTALL_DIR$/jre`
	JRE_HOME	The path of the JRE home directory for Code Insight. **Variable:** `JRE_HOME=$USER_INSTALL_DIR$/jre`
	USER_DB_IP_ADDR	The host name of the Code Insight database. **Variable:** `USER_DB_IP_ADDR=$DB_HOST$`

Table 2-6 ▪ Installation Properties (cont.)

Category	Property	Description
Variables (continued)	USER_DB_PORT	The port used by Code Insight to communicate with the database. **Variable:** USER_DB_PORT=$DB_PORT
	USER_DB_SCHEMA_ NAME	The schema name for the Code Insight database. **Variable:** USER_DB_SCHEMA_NAME=$EXISTING_SCHEMA$
	USER_DB_USER_NAME	The name of the database user that will install the Code Insight database and that Code Insight will use internally to access and manage the database. **Variable:** USER_DB_USER_NAME=DB_USER
	USER_DB_USER_ PASSWORD	The password of the database user. **Variable:** USER_DB_USER_PASSWORD=DB_PASS

Sample Response File Content

The following is an example of the response file content.

```
#Choose Install Set
#------------------
CHOSEN_FEATURE_LIST=Standalone
CHOSEN_INSTALL_FEATURE_LIST=Standalone
CHOSEN_INSTALL_SET=Standalone

#Choose Install Folder
#---------------------
USER_INSTALL_DIR=/home/qaadmin/codeinsight
USER_PALAMIDA_KEY_FILE=codeinsight.key
PATH_OF_PALMIDAKEY_FILE=/home/qaadmin/Installer
USER_JDBC_DRIVER=mysql-connector-java-5.1.41-bin.jar
PATH_OF_USER_JDBC_DRIVER_FILE=/home/qaadmin/Installer
SELECTED_DB=MYSQL
DB_HOST=10.75.116.108
DB_PORT=3306
EXISTING_SCHEMA=vnext_dummy
DB_USER=root
DB_PASS=Root%123
USER_DB_IP_ADDR=10.75.116.108
USER_DB_PORT=3306
USER_DB_SCHEMA_NAME=vnext_dummy
USER_DB_USER_NAME=root
```

```
USER_DB_USER_PASSWORD=Root%123
INSTALLER_LAUNCH_DIR=/home/qaadmin/codeinsight
DRIVER_PATH=/home/qaadmin/Installer/mysql-connector-java-5.1.41-bin.jar
SCHEMA_POPULATED_WARNING=1
TOMCAT_STARTUP_FILE=/home/qaadmin/codeinsight/tomcat/bin/startup.bat
ia.startTomcatFlag=false
ia.startBrowserFlag=false
ia.startTomcatFlagcmd=false
CATALINA_HOME=/home/qaadmin/codeinsight/tomcat
JAVA_HOME=/home/qaadmin/codeinsight/jre
JRE_HOME=/home/qaadmin/codeinsight/jre
#Install
```

Special Post-Installation Step: Configuring the MySQL SSL Option in Code Insight

Additional configuration is required when Code Insight is configured to use a MySQL database *and* any of the following:

- MySQL 8.0 connector (any version)

- MySQL 5.7 connector version greater than 5.1.37

- Oracle JRE 8u291 or greater

If *any* of these conditions are met, the `useSSL` property in Code Insight's `core.db.properties` file must be explicitly configured to indicate whether or not the MySQL instance is enabled for SSL communications. If this property is not properly set, issues with Code Insight startup or connectivity can occur. (For example, the Tomcat startup can hang while loading the `ngbridge.properties` file.)

Use the following instructions to set this property correctly in Code Insight:

- Verify SSL Enablement in MySQL

- Next Steps When SSL Is Enabled in MySQL

- Next Steps When SSL Is Not Enabled in MySQL

This configuration should be performed immediately after Code Insight is installed to avoid possible issues. Apply the configuration to Code Insight Core Server and each Scan Server installation.

Note ▪ This section describes how to configure Code Insight if the MySQL instance is enabled for SSL. It does not describe how to enable SSL in the MySQL instance. For details on enabling SSL in MySQL, see the MySQL documentation.

Verify SSL Enablement in MySQL

First, determine whether the SSL connectivity is properly enabled in the MySQL instance (that is, certificates have been created and a truststore set up according to the MySQL documentation).

Task   ***To determine whether SSL is enabled and properly configured in MySQL, do the following:***

1. Connect to the MySQL instance, and run the `status` command.

   In the results, locate the **SSL** property.

   - If this property is set to `Not is use`, SSL has *not* been enabled for the MySQL instance (that is, the `require_secure_transport = ON` value is not specified in the `my.cnf` file for the MySQL instance). Continue with Next Steps When SSL Is Not Enabled in MySQL to configure the `useSSL` property in Code Insight.

   - SSL *is* enabled for the MySQL instance if the `SSL` property is set to a value similar to this:

     ```
     cipher in use DHE-RSA-AES128-GCM-SHA256
     ```

     Continue with the next step.

2. To determine whether the enabled SSL is properly configured in MySQL, run the following command:

   ```
   show global variables like '%ssl%';
   ```

   The enabled SSL *is* properly configured in MySQL if the results show the following:

   - The values of the `have_openSSL` and `have_SSL` properties show `YES`.

   - The `ssl_ca`, `ssl_cert`, and `ssl_key` properties each have a value showing the path of the corresponding certificate.

   If SSL is properly configured, continue with Next Steps When SSL Is Enabled in MySQL.

Next Steps When SSL Is Enabled in MySQL

If SSL is enabled in the MySQL instance, use these instructions to configure Code Insight.

Task   ***To configure Code Insight when the MySQL instance is enabled for SSL, do the following:***

1. Shut down Tomcat if it is running (see Starting and Stopping Tomcat).

2. Set up Code Insight for SSL communications with the MySQL instance (if this setup has not already been performed). See Setting Up Code Insight for SSL Communications with MySQL for instructions.

3. In the Code Insight installation directory, navigate to the `tomcat/bin/config/core.db.properties` file.

4. Locate the following line (and uncomment it if necessary):

   db.url=jdbc:mysql://<DB_HOST>:<DB_PORT>/<DB_NAME>?autoReconnect=true

5. Append &useSSL=true&verifyServerCertificate=true to the line:

   ```
   db.url=jdbc:mysql://<DB_HOST>:<DB_PORT>/
       <DB_NAME>?autoReconnect=true&useSSL=true&verifyServerCertificate=true
   ```

6. Start up Tomcat to establish a connection between Code Insight and the MySQL instance.

Setting Up Code Insight for SSL Communications with MySQL

If MySQL is enabled for SSL *and* Code Insight is currently *not* configured for SSL communications with MySQL, use these instructions to configure Code Insight.

Task **To set up Code Insight for SSL communications with the MySQL instance, do the following:**

1. Shut down Tomcat if it is running (see Starting and Stopping Tomcat).

2. Obtain the `ca.pem` certificate from the MySQL instance, and move it to the `bin` directory of the JRE installation used by Code Insight (for example, `<codeInsightInstallation>/jre/bin`).

3. From the same `bin` directory in the JRE installation, import the certificate to the Java truststore using the Java `keytool` utility. (This utility is located in the `bin` directory of JRE.) The following is a sample command used to import the certificate:

   ```
   -keytool -importcert -alias <alias_name> -file <path_to_certificate_file_ca.pem> -keystore
       <jreInstallation>/lib/security/cacerts> -storepass <password>
   ```

   In the command, replace the following:

   - `<alias_name>` with the alias given to certificate when it was created

   - `<password>` with the password used to access the truststore

   - `<path_to_certificate_file_ca.pem>` with the path where the `ca.pem` certificate file is stored

4. Start up Tomcat to establish a connection between Code Insight and the MySQL instance.

Next Steps When SSL Is Not Enabled in MySQL

If SSL is not enabled in the MySQL instance, use these instructions to configure Code Insight.

Task **To configure Code Insight when the MySQL instance is not enabled for SSL, do the following:**

1. Shut down Tomcat if it is running (see Starting and Stopping Tomcat).

2. In the Code Insight installation directory, navigate to the `tomcat/bin/config/core.db.properties` file.

3. Locate the following line (and uncomment it if necessary):

   `db.url=jdbc:mysql://<DB_HOST>:<DB_PORT>/<DB_NAME>?autoReconnect=true`

4. Append `&useSSL=false` to the line:

   `db.url=jdbc:mysql://<DB_HOST>:<DB_PORT>/<DB_NAME>?autoReconnect=true&useSSL=false`

5. Start up Tomcat to establish a connection between Code Insight and the MySQL instance.

Opening the Code Insight Web UI

This section explains how to launch Code Insight Web UI.

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

1. Launch a web browser and navigate to the following URL, entering the server hostname. If necessary, check with your site's system administrator to obtain the correct hostname.

    http://<your_server_host_name>:<portNumber>/codeinsight/

    An example URL might be **http://localhost:8888/codeinsight/**.

    The Code Insight login page opens.

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

    Note ▪ The Code Insight default login name is `admin`; *the default password is* `Password123`.

3. Click **Login**. The Code Insight dashboard is displayed.

    Important ▪ For increased security, it is highly recommended that you change the default password for `admin` *after the first login. For details, Creating or Editing Users in the "Configuring Code Insight" chapter.*

Roles and Permissions in Code Insight

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

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

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

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

●   The Managing Users section in this guide describes the management of user accounts and the assignment users to system roles.

●   The "Assigning and Removing Project Users" section in the *Code Insight User Guide* describes the assignment of users to project roles.

●   The Code Insight User Roles and Permissions chapter in this guide serves as a reference to the various Code Insight system and project roles available and the permissions granted to each role. As you prepare use the Code Insight, refer to this section to determine the roles required to perform certain Code Insight functionality and the permissions the roles enable.

Starting and Stopping Tomcat

A Tomcat server is automatically installed when you install Code Insight on an instance. When the Tomcat server is started on a given instance, the Core or Scan Server (or both) installed on that instance is automatically started as well. Additionally, on the Core Server, the connection to the Code Insight UI in a browser is enabled. (The Scan Server does not support a Web UI.)

From time to time, it is necessary to start and stop the Tomcat server. For example, you might need to start the Tomcat server after the initial Code Insight installation or stop and restart Tomcat during a Code Insight upgrade or for other reasons. This section provides the procedures for starting and shutting down the Tomcat server.

Note that, in a "standalone" installation where the Core Server and a Scan Server are installed on the same instance, a single Tomcat server is installed on the instance. When you install additional Scan Servers on separate instances, Tomcat is installed on each instance. You will need to start or stop the Tomcat server on each instance separately.

The recommended best practice is *not* to run Tomcat under elevated privileges.

Starting the Tomcat Server

Use this procedure to start the Tomcat server.

Task **To start the Tomcat server, do the following:**

1. Ensure the appropriate JDBC driver file resides in `tomcat\lib`. See Information to Collect Before Running the Installer.

2. Navigate to the directory where Code Insight is installed and open the `tomcat\bin` directory (for example, `C:\codeInsight\tomcat\bin`).

3. Execute the `startup.bat` file for Windows or the `startup.sh` file for Linux. As the Tomcat startup runs, messages are displayed on the Tomcat console. The Tomcat startup may take several minutes to complete.

   When a startup message similar to the following appears in the Tomcat console, you can open Code Insight in your browser:

   `10-Aug-2017 10:06:34.796 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 58823 ms`

Stopping the Tomcat Server

Use this procedure to shut down the Tomcat server.

Task **To shut down the Tomcat server, do the following:**

1. Navigate to the directory where Code Insight is installed and open the tomcat\bin directory (for example, `C:\codeInsight\tomcat\bin`).

2. Execute the `shutdown.bat` file for Windows or the `shutdown.sh` file for Linux.

Running Code Insight as a Service

Running Code Insight as a service whenever your system starts up can save time. This section provides the appropriate procedure to configure Code Insight as a service in either a Windows environment or a Linux (RedHat 7 or CentOS 7) environment. Repeat this procedure on each instance on which you have installed a Code Insight server (Core or Scan):

- In a Windows Environment

- In a Linux Environment

Recommended best practice is *not* to run Tomcat (automatically installed on each instance running a Code Insight server) under elevated privileges.

In a Windows Environment

Perform the following procedure to run Code Insight as a Windows service.

Task   **To run Code Insight as a Windows service, do the following:**

1.   Using the command prompt, navigate to this location:

     `<CODE_INSIGHT_ROOT_DIR>\tomcat\bin`

2.   Stop the Tomcat server. See Enabling Secure HTTP Over SSL.

3.   Open the `service.bat` file with a text editor.

4.   Set the `JRE_HOME` environment variable by adding this line at the beginning of the file. (You can copy this line from the `catalina.bat` file.)

     set JRE_HOME=C:\<CODE_INSIGHT_ROOT_DIR>\jre

5.   Under the `Set default Service name` comment, set the following parameters:

     - SERVICE_NAME=**CodeInsight**

     - DISPLAYNAME=**Code Insight**

6.   Change `Description` to reflect the name of the service, which is *Code Insight*.

7.   In the `JvmOptions` line, add the following options to the existing list of options.

     Note ▪ *Be sure to separate the options from each other with a semi-colon (;).*

     - **-Dcodeinsight.ssl=false**

     - **-DcodeinsightInstallPath=<CODE_INSIGHT_ROOT_DIR>**

       `<CODE_INSIGHT_ROOT_DIR>` is the directory path where Code Insight is installed.

If Code Insight uses a proxy server, also add these options to the `JvmOptions` line to run the proxy server as a service:

Proxy Setting	Description
-Dhttps.proxyHost=<HOST>	`<HOST>` is the IP address or hostname for the proxy server.
-Dhttps.proxyPort=<PORT>	`<PORT>` is the port used by the proxy server.
-Dhttps.proxyUser=<UNAME>	`<UNAME>` is user ID used to log into the proxy server.
-Dhttps.proxyPassword=<PWD>	`<PWD>` is the password used to log into the proxy server: • If you have configured Tomcat Vault to encrypt the proxy password, enter the password alias defined in the vault configuration. • If you have not configured Tomcat Vault to encrypt the password, enter the plain-text password. For information about configuring Tomcat Vault to encrypt the password, refer to Configuring a Proxy Connection Using an Encrypted Password.
-Djdk.http.auth.tunneling. disabledSchemes	Required as is.

8. Set the `JvmMs` parameter (minimum size of the Tomcat JVM heap) to `12288 ^`.

9. Set the `JvmMx` parameter (maximum size of Tomcat JVM heap) to `26079`. This is the minimum amount of RAM required for Code Insight on a 32 GB instance.

    In general, the maximum heap size should be no greater than 80 percent of your available memory.

10. Save the `service.bat` file and exit the text editor.

11. At a command prompt, enter the following command to add a system environment variable with name `CODEINSIGHT_ROOT`. In the command, replace `C:\<CODE_INSIGHT_ROOT_DIR>` with the path of your Code Insight installation directory.

    `setx CODEINSIGHT_ROOT "C:\<CODE_INSIGHT_ROOT_DIR>"`

12. Execute the `service.bat install` command to install the Apache Tomcat Windows service.

13. When the service is installed, open **Windows Services** and search for the Service name you specified in step 4 (in this case, *CodeInsight*).

14. Right-click the CodeInsight service and select **Start**.

In a Linux Environment

Perform the following procedure to run Code Insight as a service on Linux (RedHat 7, CentOS 7, or Ubuntu 18.04).

Task   ***To run Code Insight as a service in Linux, do the following:***

1. Create a file named `CodeInsight.service` with the following content.

    ```
    [Unit]
    Description=Tomcat Service CodeInsight.service
    After=syslog.target network.target

    [Service]
    User=<userId>
    WorkingDirectory=<codeInsight_install_path>
    Type=forking
    ExecStart=<codeInsight_install_path>tomcat/bin/startup.sh
    ExecStop=/bin/kill -15 $MAINPID
    LimitNOFILE=65536

    [Install]
    WantedBy=multi-user.target
    ```

    Note the following:

    *   The `CodeInsight.service` file name is case-sensitive when referenced in the file content.

    *   The `<userId>` value for the `User` property is the user ID that will run the Code Insight service. This user ID should not run under elevated privileges.

        *   For Ubuntu, this should be the user ID that installed Code Insight (not the root user).

        *   For RedHat and CentOS, this should be a user ID with non-elevated privileges. You can ensure that such a user ID is used by explicitly including the `User` property in this file and specifying the appropriate ID. As an alternative, especially for cases where the user ID starts with a number, you can omit this property from the `.service` file and instead specify the ID using the `login` argument in the `ExecStart` command, as in the example:

            ```
            ExecStart=/usr/bin/su --login <loginUserId> -c <codeInsight_install_path>tomcat/bin/
            startup.sh
            ```

            Note • *If the* `startup.sh` *file does not have EXECUTE permission, ensure that the Code Insight user that you specify to run the service has EXECUTE permission on this file.*

2. Copy the `CodeInsight.service` file to the `/etc/systemd/system` directory:

    ```
    $ sudo cp CodeInsight.service /etc/systemd/system
    ```

3. Stop the Tomcat server. See Enabling Secure HTTP Over SSL.

4. Execute the following command to notify systemd that the Code Insight service has been added:

    ```
    $ sudo systemctl daemon-reload
    ```

5. Use the following commands to start, stop, or restart the Code Insight service. (The `CodeInsight.service` file name is case-sensitive in the commands.)

    ```
    $ sudo systemctl start CodeInsight.service
    $ sudo systemctl stop CodeInsight.service
    $ sudo systemctl restart CodeInsight.service
    ```

6. Execute the following command to enable the starting of Code Insight upon booting. (The `CodeInsight.service` file name is case-sensitive in this command.)

    ```
    systemctl enable CodeInsight.service
    ```

    From this point on, when you start your system, Code Insight will start up automatically.

Note ▪ The LimitNOFILE value `65536`, *defined in the* `CodeInsight.service` *file in step 1 above, is the open-file limit required by Code Insight. Best practice is to also set this value for individual Code Insight users or groups as a backup should situations arise when Code Insight is not run as a service. See Setting the Open-File Limit for Linux/Unix for details.*

Enabling Secure HTTP Over SSL

To implement SSL, a Secure Site SSL Certificate must exist on each instance that hosts the Code Insight Core Server or a Scan Server and that accepts secure connections. (When the Core Server and Scan Server are installed on the same instance, they share the same certificate.) Refer to http://en.wikipedia.org/wiki/HTTP_Secure and http://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html for more details about HTTPS.

Use these instructions for enabling an HTTPS connection, including how to procure a certificate:

● Enabling an HTTPS Connection

● Obtaining and Implementing a Purchased Secure Site SSL Certificate

● Generating and Implementing a Self-signed Certificate

Note ▪ For security, we recommend that Code Insight always be installed over SSH.

Enabling an HTTPS Connection

Use these instructions to enable the HTTPS connection on each server.

Task   ***To enable an HTTPS connection, do the following:***

1. Obtain and implement a Secure Site SSL certificate. You can purchase an SSL certificate or generate a self-signed certificate. Consult one of the following sections:

    ● Obtaining and Implementing a Purchased Secure Site SSL Certificate

    ● Generating and Implementing a Self-signed Certificate

2. Edit the `<CODEINSIGHT_ROOT_DIR>\tomcat\bin\catalina.bat` file (or the `catalina.sh` file depending on your operating system):

    `set -Dcodeinsight.ssl=true (default value is false)`

3. Back up the `<CODEINSIGHT_ROOT_DIR>\tomcat\conf\server.xml` file to another directory (outside of the `conf` directory).

4. Copy `server.xml` from `<CODEINSIGHT_ROOT_DIR>\tomcat\https` to `<CODEINSIGHT_ROOT_DIR>\tomcat\conf`.

    The new `server.xml` file contains a default configuration that references a keystore at `<CODEINSIGHT_ROOT_DIR>\tomcat\codeinsight.jks`. You will need to update this information as needed for your certificate, as described in step 7.

5. In the `server.xml` file, locate the following text, and ensure that the `SSLEngine` value is **on**:

    ```
    <Listener
    className="org.apache.catalina.core.AprLifecycleListener"
    SSLEngine="on" />
    ```

6. In the `server.xml` file, locate for the following text that introduces the section describing the SSL certificate:

    `FNCI SSL: Edit this section to match your certificate information.`

    This section shows the default values for the certificate:

    ```
    <!-- FNCI SSL: Edit this section to match your certificate information -->
    <Connector protocol="org.apache.coyote.http11.Http11Protocol"
        port="8888"
        minSpareThreads="25"
        enableLookups="false"
        disableUploadTimeout="true"
        acceptCount="100"
        maxThreads="150"
        maxHttpHeaderSize="8192"
        scheme="https"
        secure="true"
        SSLEnabled="true"
        keystoreFile="codeinsight.jks"
        keystorePass="codeinsight"
        keyAlias="codeinsight"
        keyPass="codeinsight"
        clientAuth="false"
        sslProtocol="TLS"
        sslEnabledProtocols="TLSv1.2"
        ciphers="ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
        ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:
        ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384"
        compressableMimeType="text/html,text/xml,text/css,text/javascript,
        application/x-javascript,application/javascript,application/json"
        compression="on"
        compressionMinSize="128"
        noCompressionUserAgents="gozilla, traviata"
    />
    ```

📄

*Note ▪ For security purposes, do not change the default value "TSLV1.2" for the **sslEnabledProtocols** parameter in this SSL section. Additionally, the "ciphers" value in this section can change over time. Revenera will notify you of any changes to this value so that you can manually update the value here.*

7. Update the following parameters in this section to reflect your installed SSL certificate information:

    ● **keystoreFile**—The file name of the keystore containing the certificate

    ● **keystorePass**—The password of the keystore

    ● **keyAlias**—The alias for the certificate entry in the keystore

    ● **keyPass**—The password for the certificate entry

    📄

    Note ▪ If the keystore and alias passwords are the same, you can specify keyPass, keystorePass or both.

8. Ensure that the value for the `cipher` parameter is up to date. If a new set of ciphers is introduced in TLS v1.2, Revenera will notify you and provide you with the new set so that you can replace the current `cipher` value. (Update the `server.xml` file found only in `<CODEINSIGHT_ROOT_DIR>\tomcat\`**https**.)

9. Restart the Tomcat server after making changes to the `server.xml` file or to a keystore. For more information, see Enabling Secure HTTP Over SSL.

Obtaining and Implementing a Purchased Secure Site SSL Certificate

The following are two sources for purchasing a Secure Site SSL Certificate:

● http://www.verisign.com/ssl/buy-ssl-certificates/secure-site-ssl-certificates/index.html

● https://www.thawte.com/ssl-digital-certificates/ssl/index.html

Follow your vendor's instructions for generating a certificate signing request (CSR).

Importing the Purchased SSL Certificate into a Keystore

After you have obtained the purchased SSL certificate, you must import it into a keystore. The following is an example command that both creates a keystore on the Tomcat server (where it needs to reside) and imports the SSL certificate into this keystore. However, you should use the instructions provided by the certificate vendor to import your certificate into a keystore.

```
keytool -import -alias "<keyAlias>" -file <yourPurchasedCertificateFile> -keystore
    <CODEINSIGHT_ROOT_DIR>\tomcat\<keystoreFile> -storepass "<keypass>"
```

Importing the SSL Certificate into cacerts

Once the SSL certificate has been imported into a keystore, use the following steps to then import the certificate into cacerts.

Task ***To import a purchased SSL certificate to cacerts, do the following:***

1. Export the certificate from the keystore and import it into cacerts, located in
    `<CODEINSIGHT_ROOT_DIR>\jre\lib\security`. To do so, run the following commands in the order shown.

    ```
    keytool -export -alias "<keyAlias>" -file <file>.crt -keystore <file>.jks

    keytool -delete -alias "<keyAlias>" -keystore cacerts

    keytool -import -alias "<keyAlias>" -keystore cacerts -file <file>.crt
    ```

    Note • *The default password for cacerts is* `changeit`.

2. (Optional) To verify that the certificate has been imported into cacerts, run the following command to view
    the contents of cacerts:

    ```
    keytool -list -v -keystore cacerts
    ```

Enabling HTTPS

With the SSL certificate installed, you need to perform these final steps to enable HTTPS on the instance.

Task ***To enable HTTPS, do the following:***

1. If the keystore created for the SSL certificate does not already reside on the Tomcat server (see Importing the
    Purchased SSL Certificate into a Keystore), copy it to `<CODEINSIGHT_ROOT_DIR>\tomcat\`.

2. Follow the procedure in Enabling an HTTPS Connection to complete the configuration steps that enable
    HTTPS on the instance running Code Insight.

Generating and Implementing a Self-signed Certificate

Use this procedure to generate a self-signed certificate.

Task ***To generate your own self-signed certificate with a keystore in place of a purchased one, do the following:***

1. Execute the following command found in the JDK:

    ```
    keytool -genkey -keyalg RSA -sigalg SHA256withRSA -alias "<keyAlias>" -keystore <keystoreFile> -
        storepass "<keypass>" -validity <numDays> -keysize 2048 -ext
        san=<ip:ipAddress,dns:domainName...>
    ```

    Provide the following values in the command:

    - **keyAlias**—The alias for the certificate entry in the keystore

    - **keystoreFile**—The file name of the keystore containing the certificate

    - **keyPass**—The password for the certificate entry

- **ip:***ipAddress***,dns:***domainName***...**—One or more values specified for the san (subject alternative name) parameter, each value indicating an IP address or domain name (hostname) secured by the certificate.

  Enter as many values as needed, separating each with a comma, to ensure that a given domain can be accessed during SSL communication. (For example, you might want to enter both the IP address and domain name for the instance containing a Scan Server to ensure that the instance can be accessed by whichever identifier is used during communication.) Enter each IP address in the format **ip:***ipAddress* and each domain name in the format **dns:***domainName*. The following shows an example san parameter:

  ```
  -ext san=ip:93.184.222.33,dns:localhost
  ```

2. Enter the server's hostname or IP address when prompted, *What is your first and last name?*

3. Leave the remainder of the prompts blank, except for the last one:

   ```
   Is CN=<yourServerNameOrIPAddress>, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
   ```

   For this prompt, type **yes**.

4. Export the certificate from the keystore and import it into cacerts, located in <CODEINSIGHT_ROOT_DIR>\jre\lib\security. To do so, run the following commands in the order shown.

   ```
   keytool -export -alias "<keyAlias>" -file <file>.crt -keystore <file>.jks

   keytool -delete -alias "<keyAlias>" -keystore cacerts

   keytool -import -alias "<keyAlias>" -keystore cacerts -file <file>.crt
   ```

5. Copy the generated keystore to <CODEINSIGHT_ROOT_DIR>\tomcat\.

6. Follow the procedure in Enabling an HTTPS Connection to complete the configuration steps that enable HTTPS on the instance running Code Insight.

   If a self-signed certificate is used on the Code Insight server, each client instance that is used to access Code Insight should add a certificate exception to the browser.

Example: Generating and Implementing a Self-signed Certificate

The following example demonstrates how to generate and store a self-signed certificate for use by Code Insight. The example assumes that Code Insight is installed on the C drive; and, for simplicity, it uses the name "codeinsight" to identify the keystore, alias, and password.

1. Create a working folder in which to generate a keystore and a self-signed certificate. This example uses the folder mywork on the C drive.

2. From a command line, navigate to the working folder:

   ```
   cd C:\mywork
   ```

3. Run the following command, which generates a keystore (codeinsight.jks) and a self-signed certificate and then imports the certificate into the keystore. The certificate is generated with the name of the keystore (codeinsight.crt).

   ```
   keytool -genkey -keyalg RSA -sigalg SHA256withRSA -alias codeinsight -keystore codeinsight.jks -
       storepass codeinsight -validity 3600 -keysize 2048
   ```

4. Import the new certificate into cacerts by running the following commands in the order shown. These commands will export the newly generated certificate from the keystore to the mywork folder, delete any existing "codeinsight" certificate in cacerts, and then import the certificate into cacerts.

```
keytool -export -alias codeinsight -file codeinsight.crt -keystore codeinsight.jks

keytool -delete -alias codeinsight -keystore C:\CodeInsight\jre\lib\security\cacerts

keytool -import -alias codeinsight -keystore C:\CodeInsight\jre\lib\security\cacerts -file
    C:\mywork\codeinsight.crt
```

To ensure that the certificate has been imported into cacerts, run the following command, which outputs a list of certificates stored in cacerts. The list should include `codeinsight.crt`.

```
keytool -v -list -keystore C:\CodeInsight\jre\lib\security\cacerts -alias codeinsight
```

5. Copy the keystore created in Step 3 to Tomcat:

```
copy c:\mywork\codeinsight.jks C:\CodeInsight\tomcat\
```

6. In `catalina.bat`, make the following changes, and then save the file:

```
-Dcodeinsight.ssl=true
```

7. Replace `tomcat\conf\server.xml` with the `server.xml` in `tomcat\https`, and then make the changes to the replacement `server.xml` as described in Enabling an HTTPS Connection. Save the file.

8. Restart Tomcat. For more information, see Starting and Stopping Tomcat.

9. In a browser, open Code Insight using the HTTPS protocol:

```
https://<hostname>:8888/codeinsight
```

10. To enable HTTPS communications between the Core Server and a Scan Server, perform these steps:

    a. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.

    b. From the **Administration** page, select the **Scan Servers** tab.

    c. Add a new Scan Server, or select a Scan Server to edit.

    d. In the **Host** field, enter the hostname for the Scan Server.

    e. In the **Port** field, enter the HTTPS port for the Scan Server.

Note ▪ You might need to accept browser warnings the first time that the application comes up; these messages should go away after the initial session.

Configuring a Networking Proxy Server Connection

By default, Code Insight uses automatic proxy server settings for any communications over the internet. However, Code Insight can be manually configured to use an enterprise network proxy that is compliant with your company's IT policies. The following procedures help you to configure Code Insight to use your enterprise proxy.

- Configuring the Proxy Server Connection Using an Unencrypted Password

- Configuring a Proxy Connection Using an Encrypted Password

- Identifying Code Insight Instances Not to Be Accessed Through the Proxy

Configuring the Proxy Server Connection Using an Unencrypted Password

Use the following procedure to configure a proxy server connection using an unencrypted password. (The password is stored in plain text in the `catalina.sh/.bat` script used to start up the proxy server.) This procedure must be performed on the instance hosting the Core Server and on each separate instance hosting a Scan Server.

If you prefer to configure the proxy server connection using an encrypted password, refer to the procedure described in Configuring a Proxy Connection Using an Encrypted Password.

Task   ***To manually configure a proxy server connection, do the following:***

1. Navigate to the `tomcat/bin` folder on the instance. This folder resides within the directory where CodeInsight is installed.

2. Open `catalina.sh` or `catalina.bat` for editing.

3. Locate the following command and uncomment it:

   `rem set CATALINA_OPTS=%CATALINA_OPTS% -Dhttps.proxyHost=<HOST> -Dhttps.proxyPort=<PORT> -Dhttps.proxyUser=<USER> -Dhttps.proxyPassword=<PASSWORD> -DproxyProtocol=<PROTOCOL> -Djdk.http.auth.tunneling.disabledSchemes=`

   Set the following values for the proxy server in the command:

   - **proxyHost**—IP address or Hostname of the proxy server.

   - **proxyPort**—Port used for the proxy connection.

   - **proxyUser**—User name used to authenticate the proxy. Omit this value for a transparent proxy connection.

   - **proxyPassword**—Password (in plain text) used to authenticate the proxy. Omit this parameter for a transparent proxy connection.

   - **proxyProtocol**—Either `http` or `https`.

4. Save the `catalina.sh/.bat` file.

5. Navigate to the `<CODE_INSIGHT_ROOT_DIR>config/core` folder, and open the `jets3t.properties` file.

6. Edit the file as follows and then save it. (This configuration ensures that the Analysis Workbench dual-pane feature, enabling users to download and compare remote files, directs its calls properly through the proxy.)

   - Set `httpclient.proxy-autodetect` parameter to `false` to ensure that the correct proxy is used (that is, the one defined for Code Insight here and in the `catalina` file).

   - Set the same proxy host, port, user ID, and password as described in step 3 above.

   - Provide the proxy domain name for `httpclient.proxy-domain`, if one is used.

7. Restart the Tomcat server so the proxy server configuration takes effect. For information about stopping and restarting Tomcat, see Starting and Stopping Tomcat.

Configuring a Proxy Connection Using an Encrypted Password

When you configure a proxy server for Code Insight using an encrypted password (as described in Configuring the Proxy Server Connection Using an Unencrypted Password), the password is stored in plain text in the `catalina.sh` or `.bat` file used to start up the proxy. If the storage of the plain-text password is not acceptable at your site, you have the option to store the proxy password as an encrypted string in a secure vault (which is configured using a Tomcat Vault utility shipped with Code Insight). Then, when the proxy server is launched, Tomcat can access the vault to retrieve the encrypted password.

Use the following steps to first configure the vault, store the encrypted password, and then configure Code Insight to use the encrypted password in its connection with the proxy server. You will need to perform this configuration process on the instance hosting the Core Server and repeat it on each separate instance hosting a Scan Server.

- Step 1: Enable Tomcat Vault for Use by Tomcat

- Step 2: Create the Java Keystore for the Vault

- Step 3: Initialize the Password Vault

- Step 4: Store the Proxy Password in the Vault

- Step 5: Use the Stored Proxy Password in Your Tomcat Configuration

Step 1: Enable Tomcat Vault for Use by Tomcat

The Tomcat Vault jar file and scripts are already installed with Code Insight. The following procedure enables the Tomcat Vault utility for use by Tomcat to manage the password vault on the current Linux or Windows instance.

Task   ***To enable Tomcat Vault for use by Tomcat, follow this procedure:***

1. Navigate to the `/tomcat/conf/catalina.properties` file in your Code Insight installation folder, and add the following lines:

   `org.apache.tomcat.util.digester.PROPERTY_SOURCE=org.apache.tomcat.vault.util.PropertySourceVault`

   `org.apache.tomcat.util.digester.REPLACE_SYSTEM_PROPERTIES=true`

   This code loads the password vault at the startup of the Code Insight Core Server or Scan Server, using the configuration information you will define in Step 3: Initialize the Password Vault.

2. (On a Linux instance only) Navigate to the `/tomcat/lib` directory in your Code Insight installation folder, and execute the following command to obtain the permissions needed to access `tomcat-vault.jar`:

   `chmod 775  tomcat-vault.jar`

3. Navigate to the `/tomcat/bin` directory in your Code Insight installation folder.

4. (On a Linux instance only) Execute the following command to obtain the permissions needed to access the `vault` script:

   `chmod 775  vault.sh`

5. Open `vault.sh` (on Linux) or `vault.bat` (on Windows) for editing.

6. Update the `JAVA_HOME` parameter with the absolute path of the JRE used by Code Insight, and then save the file.

    ● If you are using the JRE shipped with Code Insight, update the parameter with the value shown. (This value is the default when Code Insight is shipped.)

        In vault.sh on a Linux machine:

        export JAVA_HOME=**../../jre**

        In vault.bat on a Windows machine:

        set JAVA_HOME=**..\..\jre**

    ● If you are not using the JRE shipped with Code Insight, replace the parameter's default value with the absolute path to the system JRE.

Step 2: Create the Java Keystore for the Vault

The password vault managed by Tomcat Vault requires a Java keystore in which to store encrypted passwords. Use the following procedure to create this keystore.

Creating the Java keystore is a one-time process when configuring the password vault.

Task   ***To create Java keystore for the password vault, follow this procedure:***

1. Create a folder called `tomcat-vault` directly under the `/tomcat` folder in your Code Insight installation folder. This folder will be used to store the keystore and other vault files.

2. Run the following command to create the keystore:

    keytool -genseckey -keystore <CODE_INSIGHT_INSTALL_DIR>**/tomcat/tomcat-vault/**<KEYSTORE_NAME> **-alias** <alias> **-storetype jceks -keyalg AES -keysize 256 -storepass** <password> **-keypass** <password> **-validity** <days>

    The following is a command using example values:

    keytool -genseckey -keystore D:\CodeInsight\tomcat\tomcat-vault\vault.keystore -alias my_vault -storetype jceks -keyalg AES -keysize 256 -storepass password123 -keypass password123 -validity 730

Refer to this table for a description of the parameters used in the `keytool` command used to create a keystore for the password vault. Unless specified otherwise in the table, you can use any value for a given parameter.

Parameter	Description
keystore	The absolute path and name of the keystore: `<CODE_INSIGHT_INSTALL_DIR>/tomcat/tomcat-vault/<KEYSTORE_NAME>` where • `<CODE_INSIGHT_INSTALL_DIR>` is the directory in which Code Insight is installed. • `<KEYSTORE_NAME>` is the name you give to the keystore. The example command below uses `vault.keystore` as the name, but you can provide any name.
alias <alias>	The alias used to identify the keystore. You can specify any alias name.
storetype	The keystore type. The value must be `jceks`.
keyalg	The name of the algorithm used for key encryption. The value must be `AES`.
keysize	The bit size used for key encryption. The value must be `256`.
storepass <password>	The password used to access the keystore. While you can specify any password, you must use the same password for the `keypass` parameter.
keypass <password>	The password used to access the generated key pair. Use the same password that you specified for the `storepass` parameter.
validity	The number of days before the keystore will expire. The default is 90 days (as per Oracle JRE), but you change this value as needed. The command example below uses `730`.

Step 3: Initialize the Password Vault

The next step is to initialize the password vault so that it can be used to store the password information securely. This step is performed by running the Tomcat Vault script (`tomcat-vault.sh` on Linux and `tomcat-vault.bat` on Windows). The script can be run in silent mode or interactively.

• Initializing the Vault in Silent Mode

• Initializing the Vault in Interactive Mode

• Vault Parameters

Initialing the password vault is a one-time process.

Initializing the Vault in Silent Mode

The password vault can be initialized in silent mode (non-interactively) by providing the required input to the `vault.sh/.bat` script as set of arguments. The required `vault.properties` file is also created as output of the script.

Task   ***To initialize the password vault in silent mode, use this procedure:***

1. Navigate to `<CODE_INSIGHT_INSTALL_DIR>/tomcat/bin` folder, where `<CODE_INSIGHT_INSTALL_DIR>` is the directory in which Code Insight is installed.

2. Execute the following command, using the `vault.sh` (on Linux) or `vault.bat` (on Windows). For a description of each parameter used in the command, see Vault Parameters.

   ```
   vault.bat --keystore <CODE_INSIGHT_INSTALL_DIR>/tomcat/tomcat-vault/vault.keystore --keystore-
       password <password> --alias <alias-name> --enc-dir <CODE_INSIGHT_ROOT_DIR>\tomcat\tomcat-vault
       --iteration 120 --salt <random_salt> --generate-config
       <CODE_INSIGHT_ROOT_DIR>\tomcat\conf\vault.properties
   ```

   The following shows the command with example values:

   ```
   vault.bat --keystore D:\CodeInsight\tomcat\tomcat-vault\vault.keystore --keystore-password
       password123 --alias my_vault --enc-dir D:\CodeInsight\tomcat\tomcat-vault --iteration 120 --salt
       1234abcd --generate-config D:\CodeInsight\tomcat\conf\vault.properties
   ```

   If you need to make updates to this configuration later (including the addition of passwords to the vault), re-issue this same command with all its parameters. (The only non-required parameters are `--iteration` and `--salt`, which, if omitted, revert to the default values, **23** and **12345678**, respectively, in the configuration.)

Initializing the Vault in Interactive Mode

The password vault can be initialized in interactive mode, providing you with an interface that prompts you for the necessary parameter values. The following procedure describes how to run the initialization in this mode.

Unlike the silent mode, performing the vault initialization in interactive mode does not automatically generate the required `vault.properties` file once the initialization finishes. You must create this file manually. (Instructions for creating the file are included as the last step in this procedure.)

Task   ***To initialize the password vault in interactive mode, use this procedure:***

1. Navigate to `<CODE_INSIGHT_INSTALL_DIR>/tomcat/bin` folder, where `<CODE_INSIGHT_INSTALL_DIR>` is directory in which Code Insight is installed.

2. Execute the following command with no arguments:

   ```
   vault.bat
   ```

3. When asked to enter a digit, enter **0** to start an interactive session.

   An interface is displayed, enabling you to enter the required information to initialize the vault. Once you have entered all required fields, the initialization process runs.

   For a description of each field, see Vault Parameters.

```
Tomcat Vault Tool

VAULT_HOME: "D:\CodeInsight\tomcat"

JAVA: "C:\Program Files\Java\jdk1.8.0_291\bin\java"

JAVA_OPTS: ""

===================================================================
Please enter a Digit::  0: Start Interactive Session 1: Remove Interactive Session  Other: Exit
0
Starting an interactive session
Enter directory to store encrypted files:D:\CodeInsight\tomcat-vault
Enter Keystore URL:D:\CodeInsight\tomcat-vault\vault.keystore
Enter Keystore password:
Enter Keystore password again:
Values match
Enter 8 character salt:1234abcd
Enter iteration count as a number (Eg: 44):120
Enter Keystore Alias:my_vault
Initializing Vault
Oct 18, 2021 2:50:06 PM org.apache.tomcat.vault.security.vault.PicketBoxSecurityVault init
INFO: Default Security Vault Implementation Initialized and Ready
Vault Configuration in tomcat properties file:
********************************************
KEYSTORE_URL=D:/CodeInsight/tomcat-vault/vault.keystore
KEYSTORE_PASSWORD=MASK-3bgDaYn3zIhccSwJOTCOAq
KEYSTORE_ALIAS=my_vault
SALT=1234abcd
ITERATION_COUNT=120
ENC_FILE_DIR=D:/CodeInsight/tomcat-vault/
********************************************
Vault is initialized and ready for use
Handshake with Vault complete
Please enter a Digit::  0: Store a secured attribute  1: Check whether a secured attribute exists  2: Remove a secured attribute  Other: Exit
```

4. Once the initialization completes, create a file containing the vault parameters for use in the encryption process:

   a. Navigate to the `<CODE_INSIGHT_INSTALL_DIR>/tomcat/conf` directory, and create a file called `vault.properties`.

   b. Add the following parameters from the initialization output to the file. Be sure that the parameters and their values are duplicated exactly and that the `KEYSTORE_PASSWORD` value is the *masked password* shown in the output.

   c. Save the file.

Once the `vault.properties` file is created, you can proceed to add the proxy password to the vault in the same interactive session; or you can exit the session and store the password at later time. For either option, refer to Using Interactive Mode to Store the Password for instructions.

Vault Parameters

The following describes the parameters used to initialize the password vault in silent or interactive mode. The name of a given parameter in silent mode in listed in the first column, "Parameter in Silent Mode". The field name of the same parameter in interactive mode is listed in the second column ("Field in Interactive Mode").

Table 2-7 ▪

Parameter in Silent Mode	Field in Interactive Mode	Value
--enc-dir	Enter directory to store encrypted files	The absolute path in which the encrypted files for the vault are to be stored. This is typically the directory that contains the keystore created for the vault in Step 2: Create the Java Keystore for the Vault: `<CODE_INSIGHT_INSTALL_DIR>`**\tomcat\tomcat-vault** where `<CODE_INSIGHT_INSTALL_DIR>` is the directory in which Code Insight is installed. However, you can specify any path accessible to Tomcat.
--keystore	Enter Keystore URL	The absolute path and name of the keystore created in Step 2: Create the Java Keystore for the Vault.
--keystore-password	Enter Keystore password	The password used to access the keystore. This must be the same value defined for both `storepass` and `keypass` in Step 2: Create the Java Keystore for the Vault.
N/A	Enter Keystore password again	(Interactive mode only) The same keystore password entered for the previous **Keystore password** field. If the two passwords match, the message "Values match" is displayed, and you can proceed with the vault configuration.
--salt	Enter 8 character salt	A random string of exactly 8 characters that will be used in the encryption process. Special characters such as *, +, ", and \ are not supported. By default, this value is `12345678` but can be changed as long as the value contains 8 characters and does not use the special characters listed.
--iteration	Enter iteration count as a number	The number of times that the encryption algorithm is run. By default, this number is 23, but can be changed.
--alias	Enter Keystore Alias	The alias used for the keystore (as defined in Step 2: Create the Java Keystore for the Vault).

Table 2-7 ▪ (cont.)

Parameter in Silent Mode	Field in Interactive Mode	Value
--generate-config	N/A	The path and name of the file that is automatically generated at the end of the initialization phase. The file will contain the vault configuration properties required for the encryption process. Enter the following path and name for this file: `<CODE_INSIGHT_INSTALL_DIR>`**/tomcat/conf/** **vault.properties** where `<CODE_INSIGHT_INSTALL_DIR>` is the directory in which Code Insight is installed. In interactive mode, this file must be created manually, as described in Initializing the Vault in Interactive Mode.

Step 4: Store the Proxy Password in the Vault

The next phase involves storing the proxy password in the password vault. This process runs the same vault script used to initialize the vault. You can perform this step in silent mode or interactively after the vault initialization process (described in Step 3: Initialize the Password Vault) has completed.

Using Silent Mode to Store the Password

The following procedure explains how to add the proxy password to the vault in silent mode (non-interactively). Even though the vault is already initialized, you must re-enter the same parameter values used to initialized the vault, as well as provide the additional parameters needed to store the password.

Task   ***To use silent mode to store the proxy password, do the following:***

1. Navigate to `<CODE_INSIGHT_ROOT_DIR>/tomcat/bin` folder, where `<CODE_INSIGHT_INSTALL_DIR>` is the directory in which Code Insight is installed.

2. Execute the following command, using the `vault.sh` (on Linux) or `vault.bat` (on Windows). Use the same parameter values used to initialize the vault (see Vault Parameters), as well as include the additional parameters needed to store the password (see Parameters Used for Password Storage).

   vault.bat --keystore `<CODE_INSIGHT_ROOT_DIR>`**\tomcat\tomcat-vault\vault.keystore --keystore-password**
       `<password>` **--alias** `<alias_name>` **--enc-dir** `<CODE_INSIGHT_ROOT_DIR>`**\tomcat\tomcat-vault --**
       iteration 120 --salt `<random_salt>` **--vault-block** `<block_name>` **--attribute** `<attribute_name>` **--**
       sec-attr `<password>`

   The following shows the command with example parameter values. The command uses the vault initialized in Initializing the Vault in Silent Mode. The additional parameter values used to store the proxy password are in blue italics.

```
vault.bat --keystore D:\CodeInsight\tomcat\tomcat-vault\vault.keystore --keystore-password
    password123 --alias my_vault --enc-dir D:\CodeInsight\tomcat\tomcat-vault --iteration 120 --salt
    1234abcd --generate-config D:\CodeInsight\tomcat\conf\vault.properties --vault-block my_block
    --attribute proxy_pwd --sec-attr sca
```

Using Interactive Mode to Store the Password

The following procedure describes how use the interactive mode to store the proxy password in the vault. You can perform this step as a continuation of the interactive session already opened once the vault initialization process has completed; or, if you closed that session, you can open another session on the initialized vault at any later time to store the password.

Task   ***To use interactive mode to store the proxy password, do the following:***

1. If you are still in the same session that initialized the password vault (described in Initializing the Vault in Interactive Mode), proceed to the next step.

    Or

    If you have closed the session in which you initialized the password vault, open another session on the vault, using the steps described in Initializing the Vault in Interactive Mode. Even though the vault is already initialized, you must re-enter same parameter values that were used to initialize the vault. Once the session is established, proceed to the next step.

2. Below the messages "Vault is initialized and ready for use…Handshake with Vault complete" at the bottom of the interface, locate the "Please enter a digit" field.

3. Enter the value **0** to store a secure attribute—in this case, the proxy password.

4. Complete the "attribute" parameters required to store the proxy password. For a description of these parameters, see Parameters Used for Password Storage.

Once you have entered all required parameters, the password is stored. The interface displays the message "Secured attribute value has been stored in the vault" and lists the storage parameters for the password.

Parameters Used for Password Storage

The following describes the parameters used to securely store a proxy password in the password vault in silent or interactive mode. The name of a given parameter in silent mode in listed in the first column, "Parameter in Silent Mode". The field name of the same parameter in interactive mode is listed in the second column ("Field in Interactive Mode").

Table 2-8 ▪

Parameter in Silent Mode	Field in Interactive Mode	Value
--sec-attr	Please enter secured attribute value (such as a password)	The proxy password to be stored. The password is entered in plain text.
N/A	Please enter secured attribute value (such as password) again	(Interactive mode only) The same plain-text password entered for the previous field **Please enter secured attribute (such as password)**. If the two passwords match, the message "Values match" is displayed, and you can proceed with the password-storage process.
--vault-block	Enter Vault Block	A user-defined name for a location in the vault in which to store the proxy password. If you do not specify a block, one is created for you.
--attribute	Enter Attribute Name	An alias for the password. This will be name by which the password is referred.

Table 2-8 ▪ (cont.)

Parameter in Silent Mode	Field in Interactive Mode	Value
generate-config	**N/A**	The path and name of the file that is automatically generated at the end of the initialization phase. The file will contain the vault configuration properties required for the encryption process.
		Enter the following path and name for this file:
		`<CODE_INSIGHT_INSTALL_DIR>`**/tomcat/conf/** **vault.properties**
		where `<CODE_INSIGHT_INSTALL_DIR>` is the directory in which Code Insight is installed.
		In interactive mode, this file must be created manually, as described in Initializing the Vault in Interactive Mode.

Step 5: Use the Stored Proxy Password in Your Tomcat Configuration

After storing the proxy password in the password vault, you must configure the proxy connection for Code Insight in the `catalina. properties` file. This configuration will point to the secured password by its alias name so that proxy server can be accessed.

Note ▪ When you have configured Tomcat Vault to store the encrypted password, you must set up the connection with the proxy server in `catalina.properties`. This differs from the setup of a proxy connection with an unencrypted password, which is configured in the `catalina.sh/.bat` file, as described in Configuring the Proxy Server Connection Using an Unencrypted Password.

Task   ***To set up the proxy connection using the encrypted password, do the following:***

1. Navigate to the `<CODE_INSIGHT_ROOT_DIR>/tomcat/conf` folder, where `<CODE_INSIGHT_INSTALL_DIR>` is the directory in which Code Insight is installed.

2. Open the file `catalina.properties` file for editing.

3. Add the following properties to the file content:

    ```
    https.proxyHost=<proxyHost>
    https.proxyPort=<proxyPort>
    https.proxyUser=<proxyUser>
    https.proxyPassword=${VAULT::<block_name>::<attribute_name>::}
    proxyProtocol=<proxyTransferProtocol>

    jdk.http.auth.tunneling.disabledSchemes=
    ```

    where these values are provided:

    ● **proxyHost**—The IP address or Hostname of the proxy server.

- **proxyPort**—The port used for the proxy connection.

- **proxyUser**—The user name used to authenticate the proxy connection.

- **proxyPassword**—The identifier for the encrypted password used to authenticate the proxy connection, where:

  - block_name is a user-defined location for the password in the password vault

  - attribute_name is the alias for the password

  Both of these parameters were defined when the proxy password was stored in the password vault. Using the example in Step 4: Store the Proxy Password in the Vault, you would enter the following property:

  https.proxyPassword=${VAULT::my_block::proxy_pwd::}

  Note that, on an Ubuntu machine, you must use parentheses instead of brackets:

  https.proxyPassword=$(VAULT::my_block::proxy_pwd::)

- **proxyTransferProtocol**—Either **http** or **https**.

4. Save catalina.properties.

5. Navigate to the <CODE_INSIGHT_ROOT_DIR>config/core folder, and open the jets3t.properties file.

6. Edit the file as follows and then save it. (This configuration ensures that the Analysis Workbench dual-pane feature, enabling users to download and compare remote files, directs its calls properly through the proxy.)

   - Set httpclient.proxy-autodetect parameter to **false** to ensure that the correct proxy is used (that is, the one defined for Code Insight here and in the catalina file).

   - Set the same proxy host, port, and user ID used in step 3.

   - For the password, enter the *plain-text* version of the password that you stored in the vault in Step 4: Store the Proxy Password in the Vault. This is the value you entered for sec_attr (in silent mode) or **Please enter secured attribute value** (in interactive mode).

   - Provide the proxy domain name for httpclient.proxy-domain, if one is used.

7. Restart the Tomcat server so the proxy server configuration takes effect. For information about stopping and restarting Tomcat, see Starting and Stopping Tomcat.

8. Check the logs. The first item listed should be the following:

   org.apache.tomcat.vault.security.vault.PicketBoxSecurityVault.init Default Security Vault Implementation Initialized and Ready

   This message indicates that the password vault storing the encrypted password is ready for use by the proxy connection.

Identifying Code Insight Instances Not to Be Accessed Through the Proxy

If you have configured proxy support for an instance on which a Code Insight Scan Server resides, you can define an additional property on that instance to specify any instances (hosts) that the Scan Server should access directly—that is, *not* through the proxy. Use the following steps to configure this property on each instance hosting a Scan Server.

Task   ***To identify non-proxy instances to the Scan Server, do the following:***

1. Navigate to the `tomcat/conf` folder on the instance. This folder resides within the directory where CodeInsight is installed.

2. Open `catalina.properties` in a text editor.

3. Add the following property to the file content:

   http.nonProxyHosts=<hostName>

   For <hostName>, specify the hostname for each instance that the Scan Server should access directly without going through the proxy. To pass multiple hostnames, separate each hostname by the pipe (|) character. Additionally, you can use the wildcard character (*) for pattern-matching as needed.

   The following shows a possible format for the `nonProxyHosts` property value:

   http.nonProxyHosts=<hostName1|hostName2|*.hostName>

   The following shows an example property that applies the above format:

   http.nonProxyHosts=MyMachine1|MyMachine2|*.ExtraScanServerMachine

Installing the Compliance Library

The Code Insight Compliance Library (CL) is a library used by the codebase scan to perform exact-file and source-code fingerprint (snippet) matching. Code Insight compares elements of scanned codebase files with information contained in the CL to generate file-level evidence on which you can take action.

Using the CL is optional. The exact-file and source-code fingerprint (snippet) matching capabilities available with the CL are in addition to the Automated Analysis techniques basic to all scans to identify components, versions, licenses, and security vulnerabilities and to generate inventory.

Use the following instructions to install the CL on a drive accessible to the Code Insight Scan Server. For optimal performance, install the CL on the same instance as the Scan Server but on a different drive or volume from the one on which the Scan Server is installed.

Repeat this procedure on each instance hosting a Scan Server.

For more information about keeping the MD5 data used for exact-matching current, see Keeping Exact-Match Data Up to Date.

Task   ***To install the Compliance Library, do the following:***

1. Download the Compliance Library (CL) installer from the Product and License Center:

    ●   For Windows, `CodeInsightComplianceLibrary-`*`version`*`.exe`

    ●   For Linux, `CodeInsightComplianceLibrary-`*`version`*`.bin`

2. Navigate to the directory where you downloaded the installer, and launch the installer.

3. Follow the prompts to install the CL.

4. When the installation is complete, navigate to the **Scan Servers** tab on the **Administration** page to configure the CL for use by future scans. Refer to Adding or Editing Scan Servers or Checking Server Status for instructions.

Keeping Exact-Match Data Up to Date

Starting with the 2020 R4 release, Code Insight began support for a secondary data source for digest matches as an overlay to the data in the Compliance Library. Updates to the second data source (NG-bridge) are planned on a regular basis and will keep the MD5 data for exact-file matching up to date. Each NG-bridge data update release is incremental, providing only changes since the last update release. For more information on how to manage these updates for your site, see Managing NG-bridge (Digest Data) Updates for Code Insight.

Upgrading the JRE

The JRE required by Code Insight is automatically installed and configured as part of the Code Insight installation. The version currently installed is Oracle JRE 8u192. However, Code Insight also officially supports Oracle JRE 8u301 and 8u311 (and unofficially all other versions under 8u311). If your site wants to use a version later than 8u192, use the instructions in this section to perform the upgrade once the Code Insight has been installed.

Perform the upgrade on the Code Insight Core Server and each Scan Server.

Note ▪ An Oracle JRE version greater than 8u202 requires a subscription before you can download its installation package.

Task   ***To install and configure a supported later version of the required Oracle JRE, do the following:***

1. Shut down Code Insight if it is currently running.

2. In the Code Insight installation directory, delete or rename the current `jre` folder.

3. From the Oracle website, download the JRE associated with Java 8u301 or 8u311 (or another supported version) for the Windows or Linux environment.

4. Install and configure the JRE on the instance where Code Insight is installed. Use the following set of instructions appropriate for your instance platform.

On a Linux instance:

a.   Extract the downloaded `.tar.gz` file for the JRE (for example, `jre-8u311-linux-x64.tar.gz`). The extracted folder name is `jre1.8.0_301` or `jre1.8.0_311` (or the appropriate name associated with the JRE version you downloaded).

b.   Copy the extracted folder to the Code Insight installation directory (for example, `/home/qaadmin/ codeinsight`).

c.   Rename the copied folder to **jre**. (which is the name already configured for the JRE directory in the `<codeInsightInstallation>/tomcat/bin/catalina.sh` file).

If you do not want to rename the folder to **jre**, you must update the path for the JAVA_HOME and JRE_HOME variables in the `catalina.sh` file with the folder name you designated. (By default, these variables point to the `jre` folder after the Code Insight installation.)

On a Windows instance:

a.   Extract the downloaded `.tar.gz` file for the JRE (for example, `jre-8u311-windows-x64.tar.gz`). The extracted folder name is `jre1.8.0_301` or `jre1.8.0_311` (or the appropriate name associated with the JRE version you downloaded).

b.   Copy the extracted folder to the Code Insight installation directory (for example, `C:\codeinsight`).

c.   Rename the copied folder to **jre**. (which is the name already configured for the JRE directory in the `<codeInsightInstallation>/tomcat/bin/catalina.bat` file).

If you do not want to rename the folder to **jre**, you must update the path for the JAVA_HOME and JRE_HOME variables in the `cataline.bat` file with the folder name you designated. (By default, these variables point to the `jre` folder after the Code Insight installation.)

5.   Restart Tomcat (see Starting and Stopping Tomcat).

Uninstalling Code Insight

The following procedures describe how to uninstall Code Insight on a given instance. An uninstaller for Code Insight is available in the directory where the product is installed. Repeat this procedure for each instance on which you are want to uninstall the Code Insight server configuration currently deployed on the instance.

This section also includes instructions to drop the SQL Server database used as the Code Insight database, should this action be necessary.

●   Uninstalling on Windows

●   Uninstalling on Linux

●   Dropping the SQL Server Database

Uninstalling on Windows

Use the following procedure to uninstall Code Insight on a Windows instance.

Task   ***To uninstall Code Insight in Windows, do the following:***

1. Navigate to the directory where Code Insight is installed.

2. Open the `Uninstall_CodeInsight` folder.

3. Double-click `Uninstall CodeInsight.exe`.

4. Follow the on-screen prompts to uninstall Code Insight. The uninstall process will leave behind some files. Review them and delete as needed.

Uninstalling on Linux

Use the following procedure to uninstall Code Insight on a Linux instance.

Task   ***To uninstall Code Insight in* Linux*, do the following:***

1. Navigate to the directory where Code Insight is installed.

2. Open the `Uninstall_CodeInsight` folder.

3. Execute the `Uninstall CodeInsight` command and follow the on-screen prompts to uninstall Code Insight. The uninstall process will leave behind some files. Review them and delete as needed.

Dropping the SQL Server Database

If you need to drop the SQL Server database used as the Code Insight database, follow this procedure. Dropping the database also drops its maintenance plans.

Task   ***To drop the SQL Server database and its maintenance plans, do the following:***

1. If you have not already done so, download the `codeinsight_db_drop_with_maintainenceplan.sql` script. See Downloading the Scripts Needed to Set Up the SQL Server Database.

2. Open the script, and set the `@dbname` value to the name of the database to be dropped (if the value is not set to the correct name).

3. Execute the script.

3

Configuring Code Insight

After Code Insight had been installed, the Code Insight System Administrator must perform a number of configuration tasks before the user can begin using Code Insight. This chapter describes these configuration tasks:

- Adding or Editing Scan Servers or Checking Server Status

- Managing Users

- Setting Up Electronic Updates

- Managing NG-bridge (Digest Data) Updates for Code Insight

- Configuring an Email Server

- Configuring Code Insight for LDAP

- Configuring Code Insight to Use Single Sign-On

- Configuring Extended Logging

- Managing Scan Profiles

- Enabling Calculation of SHA-1 Digests for Scanned Files

- Setting Project Defaults

- Setting the Common Vulnerability Scoring System (CVSS) Version

- Creating and Managing Custom Fields for Inventory

- Creating and Managing Custom Fields for Projects

- Configuring Code Insight for Exports to SBOM Insights

- Accessing Code Insight Server REST API Documentation

- Enabling Cross-Origin Resource Sharing

- Managing Authorization Tokens

- Configuring the Session Timeout

Note • *The first time you open Code Insight, an Electronic Update will begin. The update can take 4 or more hours to complete. You cannot use the application to scan files until the update finishes. However, you can configure Code Insight while the update is in progress.*

For information about the permissions granted to the Code Insight System Administrator role, see the Code Insight User Roles and Permissions chapter.

Optionally, see Accessing Code Insight Server REST API Documentation in this chapter for information about Code Insight REST APIs that enable you to create your own administrative tool for managing scan operations and retrieving data from scan results.

Adding or Editing Scan Servers or Checking Server Status

A Code Insight Scan Server scans the source code and binary files that make up your codebases to help you identify open source code that can expose your applications to compliance issues and security vulnerabilities. The following sections provide instructions on managing these servers:

- Adding or Editing Scan Servers

- Checking the Current Status of a Scan Server

- About Scanning without the Compliance Library

Adding or Editing Scan Servers

Before users can assign project codebases to a Scan Server in order to scan them, the Scan Server must first be installed either on the same instance as the Code Insight Core Server or on a separate instance, as described in Installing Code Insight. (The Scan Server must have the same version as the Core Server.) As Code Insight System Administrator, you must then "add" the Scan Server to the Code Insight system—that is, identify the server to the Code Insight Core Server to make it available for scanning purposes, as described in this section.

If multiple Scan Servers have been installed, you can add more than one of these servers, thus providing the means for users to distribute codebase scans across servers. Keep in mind each of these Scan Servers should be installed on a separate instance with a unique host ID and port identification. The codebase for a given project can be assigned to only one Scan Server (but multiple project codebases can be assigned to a single Scan Server). All codebases assigned to a given Scan Server are stored on that server in a location that you specify.

The following procedure describes how to add an installed Scan Server to the Code Insight system and, once added, how to edit its properties as needed.

For information about Code Insight scans and their assignment to project codebases, see "About Code Insight Scans" in the "Using Code Insight" chapter in the *Code Insight User Guide*.

Task   ***To add or edit your Scan Server, do the following:***

1. Ensure that the Scan Server that you want to add or edit is running. (The Scan Server starts when the Tomcat server is started, as described in Starting and Stopping Tomcat.)

    ● For a Scan Server whose properties you are editing, ensure its status is green in the list of Scan Servers on the **Scan Servers** tab, which you access using steps 2 and 3 below.

    ● For a Scan Server whose status you want to change from disabled to enabled, manually determine whether Tomcat is running on the instance. (The gray status on the list of Scan Servers does not indicate whether Scan Server is running.)

    ● For a Scan Server that you adding, manually determine whether Tomcat is running on the instance.

2. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

3. Select the **Scan Servers** tab. The tab displays a grid listing the Scan Servers that have been added.

4. Do either of the following:

    ● To add a new Scan Server, click **Add**.

    ● To edit an already-defined Scan Server, click the ✎ (**Edit**) button in its entry. The **Scan Server** dialog appears.

5. Complete or update the fields the following fields:

Field	Description
Alias	The user-defined name for the Scan Server. This value must be unique among all Scan Servers identified to the Code Insight system, including disabled ones. (See **Status** is this table for a description of enabled and disabled Scan Servers.)
Host	The hostname (such as `kr1.eng.companyA.com`) or IP address of the instance hosting the Scan Server. If the Scan Server is on the same instance as the Core Server, enter `localhost`. The same host-and-port combination must be unique among the *enabled* Scan Servers. (See **Status** is this table for a description of enabled Scan Servers.)
Port	The port used by the Scan Server on the host instance. By default, the port is `8888`. The same host-and-port combination must be unique among the *enabled* Scan Servers. (See **Status** is this table for a description of enabled Scan Servers.)

Field	Description
CL Path	(Optional) The path for the Code Insight Compliance Library (CL), downloaded from the Product and License Center (see Installing the Compliance Library). If the path is specified, the CL is accessed as part of the scan to perform exact-file and source-code fingerprint (snippet) matching. Elements of scanned codebase files are compared with information contained in the CL to generate file-level evidence on which you can take action. The validity of the entered path is checked when you click **Save**.
	Alternatively, leave this field blank to scan your codebase *without using the CL*. (Code Insight provides the scan profile "Basic Scan Profile (without CL)" to perform the scan.) This type of scan generates inventory from Code Insight's Automated Analysis feature but has limitations, as described in About Scanning without the Compliance Library.
	Keep in mind that, when you run a scan using the CL, you obtain a deeper, more comprehensive scan on your codebase.
Codebase Path	The path on the Scan Server where Code Insight will store and manage all uploaded code for projects that use this Scan Server. Ensure you have adequate disk space to store the codebases. The recommended starting size for this directory is 500GB.
	The directory must already exist. The validity of the entered path is checked when you click **Save**.
	Once the Scan Server is added to the Code Insight system, you cannot edit this field.
Status	By default, the Scan Server is enabled for scanning.
	However, if necessary for an existing Scan Server, select **Disabled** to make the Scan Server unavailable for further scans. Once disabled, the server is no longer displayed in the **Scan Server** dropdown list during project creation or when setting global project defaults. Additionally, this field becomes read-only on the **Edit Project** window.
	Note the following when attempting to disable a Scan Server:
	● If this Scan Server is the system default Scan Server (as defined on the **Project Defaults** tab), you must change this default to another server before you can disable the current server. See Setting Project Defaults for instructions on updating the default Scan Server.
	● If this Scan Server is associated with one or more projects, a warning is displayed before you can disable the server. Once you click **Yes**, the **Start Scan** and **Upload Project Codebase** options are disabled on the **Summary** page for each project associated with the server.
	If you attempt to re-enable a disabled Scan Server when another currently *enabled* Scan Server has the same host-and-port combination or alias, you receive an error when you click **Save**.

6. Click **Save** to add the Scan Server to the Code Insight system. Errors are generated when the following conditions exist:

    *   The Scan Server you are adding or editing is not running.

    *   The version of the Scan Server you are adding is different from the Core Server version.

    *   The codebase path or CL path is invalid.

Checking the Current Status of a Scan Server

Use the following procedure to check on the status of the Scan Servers defined on your Code Insight system.

Task   ***To check the current status of a Scan Server, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Scan Servers** tab. The tab displays a grid listing the Scan Servers that have been added. (For a description of the grid columns identifying properties for each Scan Server entry, refer to the table in the previous section, Adding or Editing Scan Servers.)

    The color-code status for each Scan Server is displayed next to its alias name in the **Alias** column.

Alias	Host
■ scan	localhost

The following describes color code for Scan Server status:

Table 3-1 ▪ Scan Server Statuses

Status Code	Description
■	The green icon indicates that the Scan Server is "enabled" for scanning and is currently running (turned on). Scans are run in queue order.
■	The red icon indicates that the Scan Server is "enabled" for scanning but is currently not running (that is, it is turned off). Any attempts to associate a project with the Scan Server or upload a codebase to the server generates an error. Additionally, any attempt to initiate a scan will result in the scan's being queued. However, once the server is active, the scan will start based on queue order. (Users can click the **Past Scans** link on the project **Summary** page to view details about the scheduled scan.)

Table 3-1 • Scan Server Statuses (cont.)

Status Code	Description
■	Scan Server is "disabled" (that is, cannot be used for scanning). Whether the server is running or not has no effect on this status. If an enabled server is needed for scans on a project assigned to a disabled Scan Server, a new project must be created.

About Scanning without the Compliance Library

By default, when Code Insight scans a codebase, it uses the data in the Compliance Library (CL) to provide evidence of third-party code—exact-file matches and source-code fingerprint (snippet) matches—in your codebase.

However, if you do not have access to the CL—for example, you are running Code Insight on a virtual instance or have not yet installed the CL—or do not want to enable your installed CL, leave the **CL Path** field blank on the **Scan Servers** tab on the **Administration** page (see Adding or Editing Scan Servers or Checking Server Status). You must then use the "Basic Scan Profile (without CL)" scan profile to perform a basic scan on your codebase.

The basic scan uses Code Insight's Automated Analysis feature to perform the following:

- Generates inventory and detect vulnerabilities

- Finds evidence based on emails, URLs, and pre-defined search terms

- Employs all automated detection techniques

In the absence of a CL, Code Insight will not detect exact-file matches and source-code fingerprint matches.

You can also create a custom basic scan profile with your own pre-defined search terms, as well as specify scan exclusions for folders or files to exclude from the codebase scan, such as **/.git or **/.hg.

For more information about the "Basic Scan Profile (without CL)" scan profile and about creating and managing scan profiles in general, see Managing Scan Profiles. For instructions on associating a scan profile with a project, see "Applying a Scan Profile to the Project" in the "Using Code Insight" chapter in *Code User Guid*e.

Managing Users

The following topics describe how to manage users in your Code Insight system:

- Creating or Editing Users

- Managing User Permissions for System Activities

- Finding Users

- Disabling User Accounts

At the system level, you can also assign users to project roles so that these assignments then default each time a project is created. This task is described in Setting Project Defaults.

Creating or Editing Users

The following procedure describes how to create or edit users for your-code Insight installation.

Note ▪ *If you are using an LDAP server to synchronize the user data, you can skip this procedure. To configure an LDAP server, see Configuring Code Insight for LDAP.*

Task   ***To create or edit a user, do the following:***

1. As Code Insight System Administrator, click **administration** on the **Code Insight Dashboard**. The **Administration** page appears with a list of side tabs.

2. Select the **Users/Permissions** tab, which lists all current users in your Code Insight system.

3. To create a new user, click **Add User**; or to edit an existing user, click the Edit icon ✏.

    The **Add User** or **Edit User** dialog appears.

4. Enter information in the fields to create or edit the user:

    ● **Login**—The user's login name.

    ● **First Name**—The user's first name.

    ● **Last Name**—The user's last name.

    ● **Email**—The user's email address.

    ● **Password**—The user's password, which should be a minimum of 8 characters with no spaces and have at least one number and one capital letter.

    ● **Password Confirm**—Reenter the password from the field above.

    ● **Question**—A security question that can be answered by the user to retrieve a lost password. The question must be a minimum of 3 characters.

    ● **Answer**—The answer to the security question.

5. When you finish entering information for the user, click **Submit** to add the user to the list.

6. To assign permissions to administrate Code Insight, manage policies, and create projects, see the next section, Managing User Permissions for System Activities.

Managing User Permissions for System Activities

Use the procedures described in this section to grant or revoke the following types to user permissions used to manage system-wide activities:

- **System Administrators**—Grants permissions to configure Code Insight at the system level—scheduling Code Insight Electronic Updates, managing Code Insight user accounts and permissions, defining global project defaults and the scan profiles associated with projects, specifying the CVSS version for vulnerability reporting, configuring an email server for Code Insight notifications, setting up Code Insight for LDAP and single sign-on, and integrating Code Insight with application management system (ALMs), such a Jira.

- **Manage Policy**—Grants the user permission to manage policies that automate the inventory review process—that is, automatically mark published inventory items as approved, rejected, or requiring a manual review—without the need for manual reviews.

  Policy details are described in "Managing Policies" in the "Using Code Insight" chapter in *Code User Guide*.

- **Create Project**—(Displayed only if you selected **No** for **Allow all users to create projects?**) Grants the user permission to create projects and project folders. Users automatically become the Project Contact for each project they create and are assigned to the Project Administrator role as well as other project roles.

  A **Create New** button, enabling users to create projects and project folders, is visible on the **Projects** page for only those users granted this permission. Project and folder creation is described in "Creating a Project" and "Managing Items in the Project List" in the "Using Code Insight" chapter in *Code Insight User Guide*.

> *Note ▪ In addition to these permissions, roles can be assigned to users at the individual project level, as described in the Code Insight User Guide.*

See the topics in this section for more information:

- Grant System Permissions to Users

- Revoke User Permissions

Grant System Permissions to Users

Follow these steps to grant one or more permissions to individual users.

Task   ***To grant permissions to users, do the following:***

1. As Code Insight System Administrator, navigate to the **Users/Permissions** tab on the **Administration** page. (For instructions on getting to this tab, see the initial steps in Creating or Editing Users.)

2. Click **Manage Permissions** to open the **Manage Permissions** dialog.

   Note that all users assigned to a given role are shown on this tab, whether the user was manually assigned the role or had inherited the role (for example, through project migration).

3. Select the **Yes** or **No** option for **Allow all users to create projects?** to determine whether all or selected users will have permission to create projects. If you select **No**, a **Create Project** pane is added to the dialog to enable you to select the users to which to grant this permission. (The default is **Yes**, allowing any user to create projects.)

4. To assign users to a given role, drag and drop one or more user names from the **Select Users** list to the desired "role" pane (**System Administrators**, **Manage Policy**, or **Create Project**). Repeat this step a necessary. (A user can be assigned to multiple roles.)

5. Click **Close** to return to the **Users/Permissions** tab.

Revoke User Permissions

Follow these steps to revoke a user's permissions.

Task   ***To revoke user permissions, do the following:***

1. Navigate to the **Users/Permissions** tab on the **Administration** page. (For instructions on getting to this tab, see the initial steps in Creating or Editing Users.)

2. Click **Manage Permissions** to open the **Manage Permissions** dialog.

3. To remove a user from a role, click ✖ next to the user's name in the appropriate "role" pane. You can remove any user from a role, even those who inherited the role.

4. Once you have revoked the necessary permissions, click **Close** to return to the **Users/Permissions** tab.

Finding Users

As a Code Insight System Administrator, you might need to search for Code Insight users to manage their permissions. You can search for users on the **Users** tab or on the **Summary** tab for the project.

Task   ***To find users, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ☰ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Users** tab.

3. In the **Enter Search Criteria** field, enter a character string by which to search user information in any of the fields.

4. Click **Search**.

Disabling User Accounts

Code Insight supports disabling user accounts in the browser.

Note ▪ The Admin user account is created automatically; it cannot be disabled.

Task   ***To disable user accounts, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon 🔲 in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Users** tab.

3. Click the **Edit** ✐ icon in the **Actions** column for the user account you want to disable. The **Edit User** dialog appears.

4. Select the **Disable Account** checkbox, and click **Submit**. The **Success** dialog appears.

5. Click **OK**. The user account is now disabled. The user will receive the message, "`Invalid Username and/or Password. If you believe you entered a valid user, please contact your System Administrator`" when attempting to log into Code Insight.

Setting Up Electronic Updates

An initial full Electronic Update is run automatically after your initial startup of Code Insight. It provides the basis of a local data library used by Code Insight to identify OSS and third-party code in your codebase. After this initial update, Code Insight provides a means for you to schedule additional Electronic Updates to keep the library up to date, helping to ensure that the latest component, version, license, and vulnerability information is available for your product. You can schedule these updates to execute automatically at a regular frequency or manually through the Administration interface.

At a basic level, an Electronic Update is executed as either a *server* or a *local* update, depending on the method used to retrieve the Electronic Update files from Revenera. You configure the type of update to run based on your site requirements.

By default, scheduled Electronic Updates are *incremental*—that is, each update applies only changes that have occurred since the previous update. However, when necessary, you can force a *full* update, which overwrites all data from the previous update. Full updates should be run manually only and with the understanding that they require considerable overhead.

Refer to the following for more information:

● Server vs Local Electronic Updates

● Running Server Electronic Updates

● Running Local Electronic Updates

● Configuring the Use of SFTP for Obtaining Update Files

- Notification of an In-Progress Electronic Update

Server vs Local Electronic Updates

Code Insight enables you to configure the Electronic Update to run as either a server or local update. The difference between the two methods is the means by which the Code Insight server obtains the files required to run the update:

- During a **server** Electronic Update, the most recent Electronic Update files are automatically downloaded from Revenera to the Code Insight server as part of the update process. The default protocol for a server Electronic Update is HTTPS but can be configured to be SFTP if needed (see Configuring the Use of SFTP for Obtaining Update Files for instructions). For additional information on configuring and triggering updates, see Running Server Electronic Updates.

- For a **local** (offline) Electronic Update, you must manually download the Electronic Update files from Revenera to your machine or to a local SFTP server. In both cases, the location should be locally accessible to the Code Insight server so that Code Insight can access the downloaded files, unpack them, and apply the data to the Code Insight database schema. This type of Electronic Update is useful when the Code Insight server has no external Internet access or when a specific Electronic Update version is needed for testing or demonstration purposes. For additional information, see Running Local Electronic Updates.

You can switch between running server and local updates as needed. You can also configure either of the two types of updates to download over SFTP, such that Electronic Update files are downloaded either from Revenera's SFTP server or from your own local SFTP server. For more information, see Configuring the Use of SFTP for Obtaining Update Files.

Running Server Electronic Updates

Code Insight lets you run an Electronic Update as either a server or local update (see Server vs Local Electronic Updates for descriptions of the two update types). The following topics describe the various ways in which to manage *server* Electronic Updates.

- Scheduling Server Electronic Updates That Run Automatically

- Disabling Automatic Server Electronic Updates

- Running a Server Electronic Update Manually

By default, the Electronic Update process downloads the update files from Revenera using HTTPS. If you want the Electronic Update to use SFTP instead, see Configuring the Use of SFTP for Obtaining Update Files. This configuration must be performed prior to running updates.

Scheduling Server Electronic Updates That Run Automatically

For server Electronic Updates, Code Insight enables you to configure updates to run automatically at a specified frequency as described in this section. (Note that you can always force a server or local update between the scheduled updates. See Running a Server Electronic Update Manually or Running Local Electronic Updates for details.)

Alternatively, you can disable regularly-scheduled automatic updates altogether and manually run the updates as needed. (See Disabling Automatic Server Electronic Updates for details.)

Note ▪ Codebase scans cannot be performed during the Electronic Update process, but a scan that is already underway will not be interrupted when an automatic update process is scheduled to begin. The update will be queued and automatically run based on queue order.

Task   ***To schedule an automatic server Electronic Update, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ☰ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Electronic Updates** tab.

3. Click **Server** for **Electronic Update Type**.

4. From the **Update Frequency** dropdown list, select the frequency at which to run the Electronic Update:

    ● **Never**—If you select **Never**, Electronic Updates will not run automatically. (Selection of this option hides any additional dropdown lists.)

       If you need to run an update, you can do so manually as needed. See Running a Server Electronic Update Manually for details.

    ● **Daily**—If you select **Daily**, a second dropdown list is displayed, prompting you to choose the time of day when you want the Electronic Update to occur.

    ● **Weekly**—If you select **Weekly**, both the "time of day" and the "day of week" dropdown lists are displayed. Select the time of day and the day of the week when you want the Electronic Update to occur.

5. When you have finished setting the execution frequency for the update, select **Save Schedule**. A prompt appears to notify you that your edits have been saved.

Electronic Updates will run automatically based on the schedule you have set.

Disabling Automatic Server Electronic Updates

Use this procedure to disable the automatic triggering of Electronic Updates. Once you disable the execution of the scheduled updates, an Electronic Update will no longer run automatically. However, you can run updates manually as needed (see Running a Server Electronic Update Manually for details).

Task   ***Tao disable automatic server Electronic Updates, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ☰ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Electronic Updates** tab.

3. Click **Server** for **Electronic Update Type**.

4. From the first dropdown list in the **Update Frequency** section, select **Never** to disable automatic updates.

5.   Click **Save Schedule**.

Running a Server Electronic Update Manually

You can trigger a server Electronic Update any time. For example, you might need to run an update between automatic updates if you cannot wait for the next update to determine critical information, such as the impact of new vulnerabilities on your product. Or you might need to force a full update if the most recent update did not complete properly.

If automatic server Electronic Updates are disabled (see Disabling Automatic Server Electronic Updates), you can use this procedure to request a server update whenever needed or to run a local update if necessary.

When you manually run an Electronic Update, it is initiated immediately or once any pending scans complete.

Note ▪ Running an Electronic Update might cause an unexpected delay in starting a codebase scan. However, if a scan is already underway when the update process is triggered, the update is queued and automatically run based on queue order.

Task   ***To run a server Electronic Update manually, do the following:***

1.   On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2.   Select the **Electronic Updates** tab.

3.   Click **Server** for **Electronic Update Type**.

4.   For the **Run Update Now** option, select the update scope:

   ●   **Incremental Update**—Schedule an incremental Electronic Update. However, if no changes have occurred in the Electronic Update data since it was last run, the update is *not* initiated.

   ●   **Full Update**—Force a full Electronic Update whether or not changes have occurred in the Electronic Update data since it was last run. Use this option judiciously as a full Electronic Update takes several hours to complete, similar to the time required to run the initial update when Code Insight was installed.

5.   Click **Update** to initiate the Electronic Update immediately or once pending scans have completed.

Running Local Electronic Updates

Code Insight lets you run an Electronic Update as either a server or local update (see Server vs Local Electronic Updates for descriptions of the two update types). The following topics describe how to run a *local* Electronic Update.

   ●   Files Required for a Local Electronic Update

   ●   Running the Local Electronic Update from Your Machine

   ●   Running the Local Electronic Update from Your Own SFTP Server

For a description of the local Electronic Update and its comparison to the server Electronic Update, see Server vs Local Electronic Updates.

Files Required for a Local Electronic Update

Before running a local Electronic Update, you must manually download the Electronic Update files from Revenera to your machine or to a local SFTP server. In both cases, the location should be accessible to the Code Insight server, so that Code Insight can access the downloaded files, unpack them and apply the data to the Code Insight database schema.

Work with your Revenera representative to determine the best way for your site to receive notifications from Revenera when a new Electronic Update is available and to obtain download details.

The following files must be manually downloaded from Revenera to run the local Electronic Update:

- **Update Manifest file**—The manifest file, `update_manifest.txt`, which contains the following:

  - Information that Code Insight uses to determine whether to perform the update.

  - The expected hash value for each data file in the `update.zip` file (see the next bullet). This information is compared with the hash values of the actual files in the archive to ensure that the files have not changed.

- **Update Data file**—The `update.zip` file, an archive of the data files containing the CVSS information used by Code Insight to perform the Electronic Update.

  Code Insight uses the hash information in the manifest file (see the previous bullet) to ensure that the data files are the expected ones and that they have not changed or been tampered with.

Running the Local Electronic Update from Your Machine

Use the following procedure to run a local Electronic Update from your machine after you have obtained the `update_manifest.txt` file and `update.zip` files.

Task   ***To run a local Electronic Update, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Electronic Updates** tab.

3. Click **Local** for **Electronic Update Type**.

4. Click **Select File** next to the **Update Manifest File** field to select the `update_manifest.txt` that Code Insight will upload to perform the update. (This file was manually downloaded from Revenera prior to this request for an update. You can locate this file at the locally accessible location to which it was saved.)

5. Click **Select File** next to the **Update Data File** field to select the `update.zip` file that Code Insight will upload to perform the update. (This file was manually downloaded from Revenera prior to this request for an update. You can locate this file at the locally accessible location to which it was saved.)

6. For **Run Update Now**, select the update scope:

    ●   **Incremental Update**—Schedule an incremental Electronic Update. However, if no changes have occurred in the Electronic Update data since it was last run, the update is *not* initiated.

    ●   **Full Update**—Force a full Electronic Update whether or not changes have occurred in the Electronic Update data since it was last run. Use this option judiciously as a full Electronic Update takes several hours to complete, similar to the time required to run the initial update when Code Insight was installed. (You might need to force a full update, for example, if the most recent update did not complete properly.)

7. Click **Update** to initiate the Electronic Update immediately or once pending scans have completed.

Running the Local Electronic Update from Your Own SFTP Server

The instructions for running a local Electronic Update from your own SFTP Server are similar to those for running from a Revenera server (see Running Server Electronic Updates for more information), with the following two exceptions:

●   Instead of downloading the files from the Revenera server, Electronic Update downloads the files from your own local SFTP server, configured using instructions in the Configuring SFTP to Obtain Update Files Using Your Own SFTP Server section.

●   As a prerequisite step for the use of a local SFTP server, ensure that the update files (`update_manifest.txt` and `update.zip`) are downloaded from Revenera to the SFTP server before each Electronic Update. This step can either be done manually or, for example, by using a script to automate the download of the update files based on your update schedule.

Configuring the Use of SFTP for Obtaining Update Files

By default, the Electronic Update files are downloaded over HTTP. However, you can configure the process to use SFTP instead, so that the update files are downloaded either from Revenera's SFTP server or from your own local SFTP server.

For additional security, you can also configure a proxy server to handle communications between the SFTP server and Code Insight.

Refer to the following sections for more information:

●   Configuring SFTP to Obtain Update Files Using the Revenera SFTP Server

●   Configuring SFTP to Obtain Update Files Using Your Own SFTP Server

●   Configuring the Use of an SFTP Proxy Server

Configuring SFTP to Obtain Update Files Using the Revenera SFTP Server

To enable the Electronic Update to download update files from the Revenera SFTP server, the Code Insight database administrator must add the `update.sftp.enable` property to the `pas_global_properties` table, setting the property to **true**.

After SFTP is enabled, Code Insight will automatically retrieve the default information about the Revenera SFTP server to establish the necessary connection needed to download the update files, unpack them, and apply the data to the Code Insight database schema whenever a server Electronic Update occurs.

Task   ***To configure the use of SFTP for Electronic Updates using the Revenera SFTP server, do the following:***

Execute this command against the Code Insight database:

```
INSERT INTO PAS_GLOBAL_PROPERTIES (SERVER_ID_, KEY_, VALUE_, ENCRYPTED_) VALUES ('0',
    'update.sftp.enable', 'true', 0);
```

Configuring SFTP to Obtain Update Files Using Your Own SFTP Server

To configure the use of your own SFTP server for Electronic Updates communications, the Code Insight database administrator needs to insert new rows in the `pas_global_properties` table that both enable SFTP and identify the SFTP server and its credentials to Code Insight.

After SFTP is configured, Code Insight automatically retrieves information about your SFTP server from the database to establish the connections needed to download the update files, unpack them, and apply the data to the Code Insight database schema whenever Electronic Update occurs.

As a prerequisite step for the use of a local SFTP server, ensure that the update files (`update_manifest.txt` and `update.zip`) are downloaded from Revenera to the SFTP server before each Electronic Update. This step can either be done manually or, for example, using a script to automate the download of the update files based on your update schedule.

Task   ***To configure the use of SFTP for Electronic Updates using your own SFTP server, do the following:***

Execute this command against the Code Insight database, replacing variables with information about your SFTP server:

```
INSERT INTO PAS_GLOBAL_PROPERTIES
(SERVER_ID_, KEY_, VALUE_, ENCRYPTED_) VALUES
('0', 'update.sftp.enable', 'true', 0),
('0', 'sftp.server.url', '<hostname>', 0),
('0', 'sftp.server.port', '<port>', 0),
('0', 'sftp.server.username', '<username>', 0),
('0', 'sftp.server.password', '<password>', 0),
('0', 'update.file.name', '<path_to_update.zip>', 0),
('0', 'update.manifest.file.name', '<path_to_manifest.txt>', 0);
```

The following describes the variables that the database administrator needs to define in the command:

- **hostname**—The IP address or URL of the SFTP server.

- **port**—The port used for the SFTP server

- **username**—The username used to authenticate the SFTP server.

- **password**—The password used to authenticate the SFTP server.

- **path_to_update.zip**—The path of the `update.zip` file on your SFTP server.

- **path_to_manifest.txt**—The path of the `manifest.txt` file on your SFTP server.

Configuring the Use of an SFTP Proxy Server

For additional security, you can configure a proxy server to handle communications between the SFTP server (Revenera's or your own local server) and Code Insight.

To use a proxy server, the Code Insight database administrator needs to insert new rows in the `pas_global_properties` table in the Code Insight database to provide the information needed to communicate with the proxy.

Task   ***To configure the use of a proxy server between the SFTP server and Code Insight, do the following:***

Execute this command against the Code Insight database, replacing variables with information about your SFTP server:

```
INSERT INTO PAS_GLOBAL_PROPERTIES
(SERVER_ID_, KEY_, VALUE_, ENCRYPTED_) VALUES
('0', 'update.socks.proxy-host', '{host}', 0),
('0', 'update.socks.proxy-port', '{port}', 0),
('0', 'update.socks.proxy-user', '{username}', 0),
('0', 'update.socks.proxy-password', '{password}', 0),
('0', 'update.socks.proxy-type', '{SOCKS4/SOCKS5}', 0);
```

The following describes the variables that the database administrator needs to define in the command:

- **host**—IP address or Hostname of the proxy.

- **port**—Port used for proxy.

- **username**—User name used to authenticate the proxy. Omit this value for a transparent proxy connection.

- **password**—Password used to authenticate the proxy. Omit this value for a transparent proxy connection.

- **SOCKS4/SOCKS5**—The type of SOCKS Internet protocol used by the proxy. Specify one. (These are protocols currently supported.)

Notification of an In-Progress Electronic Update

Whenever an Electronic Update is in progress, a banner is displayed at the top of the Code Insight UI, indicating that an update is running and that scheduled scans will resume once the update completes. The banner is shown for any Electronic Update—whether server or local, forced or automatically run by schedule—and persists across all Code Insight pages.

The banner is automatically closed once the Electronic Update completes. (Users cannot manually close the banner.)

Managing NG-bridge (Digest Data) Updates for Code Insight

The NG-bridge module delivers digest-match data to Code Insight that is used to perform exact matching by the scanner. NG-bridge is Code Insight's next generation bridge solution that complements the Compliance Library (CL) with digest-match data beyond that provided in CL 2.43. The NG-bridge component is included with Code Insight and is a separate module that runs along-side the product to support "exact match" functionality.

Background

During exact-file matching, the scanner compares each scanned codebase file with the data set across the CL and NG-bridge and reports on any exact matches to open-source software (OSS) files or third-party files in the collection. This matching process compares the MD5s of codebase files against the MD5s stored in the CL and the NG-bridge index for OSS or third-party files.

Automatic Updates Managed by an Internal Update Facility

Starting with the 2020 R4 release, Code Insight began support for a secondary data source for digest matches as an overlay to the data in the CL. Updates to the second data source (NG-bridge) are planned on a regular basis and will keep the MD5 data for exact-file matching up to date. Each NG-bridge data update release is incremental, providing only changes since the last update release. During a Comprehensive scan, the MD5 of a codebase file is checked against both the CL and the NG-bridge index to search for a match. If a match is found in any location, it is recorded in the Analysis Workbench as evidence.

Code Insight provides an internal NG-bridge data update facility that automatically checks for, downloads, and processes update releases on your machine at a regularly scheduled time.

If your site does not have Internet access, Code Insight offers a manual download option for the NG-bridge data update releases. After you have downloaded the update files, the internal update facility will process these files at the next scheduled update time. (Without Internet access, automatic NG-bridge downloads will continue to trigger at the scheduled time but fail with an error message. These attempts do not impact your system nor the processing of the files you have manually downloaded.)

By default, the NG-bridge data update facility is initially disabled. To enable it so that you can obtain NG-bridge data updates, follow the procedure described in Enabling/Disabling NG-bridge Data Updates. (If you do not have the CL installed or are not performing exact-file matching, you can keep the facility disabled.) Once enabled, the facility can always be disabled as necessary for your site.

Configuring the Update Process for Your Site

The following procedures can be used to configure NG-bridge data updates for your site:

- Changing the Scheduled Time for NG-bridge Data Updates

- Enabling/Disabling NG-bridge Data Updates

- Downloading NG-bridge Data Updates Releases Manually

Changing the Scheduled Time for NG-bridge Data Updates

By default, the NG-bridge is configured to check for updates daily at 2 am. However, the Code Insight System Administrator or the database administrator can use the following procedure to change this scheduled time to suit your site's requirements.

Task   ***To change the scheduled time NG-bridge data updates, do the following:***

1. Open the `<codeInsightInstallPath>\config\scanEngine\ngbridge.properties` file.

2. Change the current `bridge.cronexpression` property to reflect the new time. The following shows the default value, which triggers a daily CL update process at 2 am:

   `bridge.cronexpression= 0 0 2 * * ?`

   This diagram defines the parts of the cron expression representing **2 am** to help you understand how to set the value you want:
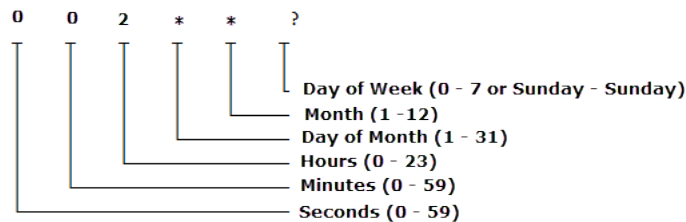
Enabling/Disabling NG-bridge Data Updates

The Code Insight System Administrator or the database administrator can use these procedures to enable or disable the internal NG-bridge data update facility that downloads and processes the updates. By default, this update facility is initially disabled. Your site does not need to enable the facility if the scans at your site do not perform exact-file matching or if the CL is not installed.

Enabling NG-Bridge Data Updates

The Code Insight System Administrator or database administrator can use this procedure to enable the NG-bridge data updates to establish the automatic download and processing of the digest-match data used by Code Insight.

Task   ***To re-enable internal NG-bridge data update facility, do the following:***

In the PAS_GLOBAL_PROPERTIES table in the Code Insight database, change the `enable.ngbridge` property to `true`.

Disabling the NG-Data Bridge Updates

This procedure disables the internal NG-bridge data update facility. Disablement might be necessary to control the cadence at which digest match data is updated in Code insight or to accommodate changes in your site's scan requirements (for example, exact-file matching is no longer used during scans).

Important • *If you intend to manually download the NG-bridge data updates, do not disable the internal NG-bridge data update facility. You will still need to have this facility enabled to process the manually downloaded data file for digest-match data.*

Task **To disable internal NG-bridge data update facility, do the following:**

In the PAS_GLOBAL_PROPERTIES table in the Code Insight database, change the `enable.ngbridge` property to `false`.

Downloading NG-bridge Data Updates Releases Manually

If the Code Insight Scan Server does not have Internet access, you can use the following procedure to manually download the files for the NG-bridge digest-match data updates to the proper location on the server. Once the files are downloaded and properly placed on your instance, the NG-bridge data update can be processed.

Task **To download the NG-bridge data update files manually and initiate processing, do the following:**

1. From a machine that has Internet connectivity, open the Flexera Product and License Center.

2. Locate the `ngdownloader-1.0.0.zip` file under the current Code Insight version, and download the appropriate archive format (Linux or Windows).

3. Extract the archive to any location. However, if want to download the actual NG-bridge data update files to this same location, make sure the location is locally accessible to the Code Insight server—for example, on a shared drive or a local USB drive. (You also have the option to change the download location in step 4.)

    The following files are included in the archive:

    *   `ngdownloader-1.0.0.jar`

    *   `run.sh` (or `run.bat`)

4. Set up the `run` script to download the files for the NG-bridge data update. Use any of the following arguments to configure the download. When including arguments, use the format shown in this example:

    ```
    run.bat file.download.path=custompath delete.previous.downloads=true
    ```

Table 3-2 ▪ Arguments for Script Used to Download NG-Bridge Data Update Files

	Argument	Description
Argument with defaults	Explicitly providing the following arguments is optional. If the argument is omitted, its default is used during execution.	
	delete.previous. downloads	The `true` or `false` boolean for deleting any previously downloaded NG-bridge data update files found in the location specified by `file.download.path`. The default for this argument is `false` so that these files are retained.
		If you choose to delete previously downloaded update files, note that the `diget.meta.ref.json` is always retained.
Argument with defaults (continued)	file.download.path	The path identifying the location to which to download the NG-bridge data update files. The path should be relative to the current location and must be locally accessible to the Code Insight server, such as on a shared drive or a local USB drive.
		The default for this argument is a "downloads" folder in the current location (that is, the location which you extracted the `ngdownloader.jar` and `run` script files).
	update.https.url	The URL for the Revenera site (`https://updates.palamida.com/`) from which the NG-bridge data update files will be downloaded. Best practice is to omit this argument so that the correct Revenera location defaults for command.
Required credentials	You must enter a value for each of the following credentials to access the Revenera URL.	
	update.https.username	The user ID authorized to connect to the Revenera site.
	update.https.password	The password authorized to connect to the Revenera site.
Arguments when using a proxy	You must enter value for each of the following arguments if you are using a proxy server to access the Revenera site from which the NG-bridge data update files will be downloaded.	
	Dhttps.proxyHost	The hostname or IP address of the proxy server.
	Dhttps.proxyPort	The port used by the proxy server.
	Dhttps.proxyUser	The user ID authorized to access the proxy server.
	Dhttps.proxyPassword	The password authorized to access the proxy server.

5. Execute the `run` command.

The following folders and files are created in the download path you specified.

📁 downloaded

📁 logs

📄 digest_meta_ref.json
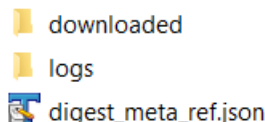
- **digest_meta_ref.json**—Used to determine which NG-bridge data update files need to be downloaded. Only files for new or missing updates are downloaded. Previous updates are not re-downloaded. Once all appropriate update files have been downloaded, the contents of the `digest_meta_ref.json` file are updated with the current list of all update releases that have been downloaded to this location up to this point. This same file is then used again to determine what new updates need to be downloaded in the next update process.

- **downloaded**—The folder containing all the NG-bridge data update files downloaded during the `run` script execution. You can use the `file.download.path` argument in the `run` command to change this folder name (or the entire URL or path) for the download.

- **logs**—The folder containing the log files generated during the download process.

6. Once the download is complete, copy the contents of the "downloaded" folder to the following location:

   `<codeInsightInstallPath>\tomcat\temp\ngbridge_updates\meta`

   At the next scheduled time for an NG-bridge data update, the internal update facility determines that update files have been downloaded and processes the files so that their data is ready to be used with the CL during Comprehensive scans.

Configuring an Email Server

Code Insight can send email alerts that are triggered by certain events. For example, when a scan completes or when a new vulnerability is detected in the project inventory. It is highly recommended that the email server configuration be set up for the application. Email server configuration is available in Code Insight on the **Administration** page. This section provides the procedure for configuring email.

📋

Task   ***To configure your email server, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Email Server** tab.

3. Enter information and make selections in the fields:

   - **Enable Email Server**—Select **Yes** to enable Code Insight to use the email server or **No** to leave it disabled. The default is **No**. The rest of the fields on this page are not available until you select **Yes**.

   - **Sender's Email Address**— Enter the email address of the sender.

   - **SMTP Host Name**—Enter the SMTP hostname.

   - **SMTP Host Port**—Enter the port number of the SMTP host.

- **SMTP User Name**—Enter the SMTP user name. This field can be left blank for anonymous SMTP configuration.

- **SMTP User Password**—Enter the SMTP user password. This field can be left blank for anonymous SMTP configuration.

- **Enable SMTP over TLS**—Select **Yes** to use Transport Layer Security (TLS) to secure email over SMTP or select **No** to leave this option disabled.

4. Click **Save** to save your settings.

Configuring Code Insight for LDAP

Code Insight supports user authentication and authorization through LDAP (Lightweight Directory Access Protocol). The following topics describe how to configure the synchronization of user identification data from LDAP to Code Insight and thus enable LDAP user authentication for Code Insight:

- Synchronizing User Identification Data

- About the LDAP Directory Structure

- Setting Up a User Search

- Implementing LDAP in Code Insight

- LDAP Tab Field Descriptions

Synchronizing User Identification Data

Code Insight provides the ability to import user identification data from LDAP. This section explains the type of user identification data that is imported.

- User Metadata

- User Email Requirement

- Disabled Users

User Metadata

The metadata for each user (name, email, and so forth) is pulled from LDAP and refreshed in the Code Insight database at a regular frequency via a scheduler module running within Code Insight. The data synchronization is a one-way pull from LDAP into the Code Insight database. This action overwrites the existing data in the database. User data for those users that do not exist in LDAP is not affected by this process.

The LDAP passwords for users are not stored the Code Insight database. All user authentication occurs on the LDAP server once a user enters a user name and password to access Code Insight.

User Email Requirement

Code Insight requires that all users have an email address. Therefore, only those users with a valid email address specified as a user attribute on the LDAP server will be synchronized. For more information, see LDAP Search Query.

Disabled Users

Users who are disabled in Code Insight will still have their data synchronized with LDAP, but will have the disabled flag set to "true" and will not be granted access to the application.

About the LDAP Directory Structure

An LDAP server stores attribute-based data in a hierarchical branching structure called a Directory Information Tree (DIT). A DIT can contains a broad range of information about different type of data objects, including users, account groups, and resources such as printers or applications. The following topics provide insight into a DIT:

- DIT Hierarchy

- Sample Directory Information Tree

- Distinguished Name for an Object

- LDAP Base

DIT Hierarchy

DIT data is arranged in directory levels, some of which include:

- Domain component (DC) or organization (O)

- Organizational unit (OU)

- Common name (CN)

Before configuring Code Insight for LDAP, you are strongly recommended to understand the LDAP directory structure at your site. A complete description of possible LDAP directory hierarchies is beyond the scope of this document. Consult with your site's LDAP administrator for more information about your account's specific LDAP configuration and directory structure.

Sample Directory Information Tree

A typical LDAP directory structure can contain thousands of entries arranged in a complex structure. An example of a DIT is illustrated here. In the example, the DIT contains 4 levels of entries, including 2 domain components, 2 organizational units, and 8 common names (3 of which are groups and 5 of which are users).
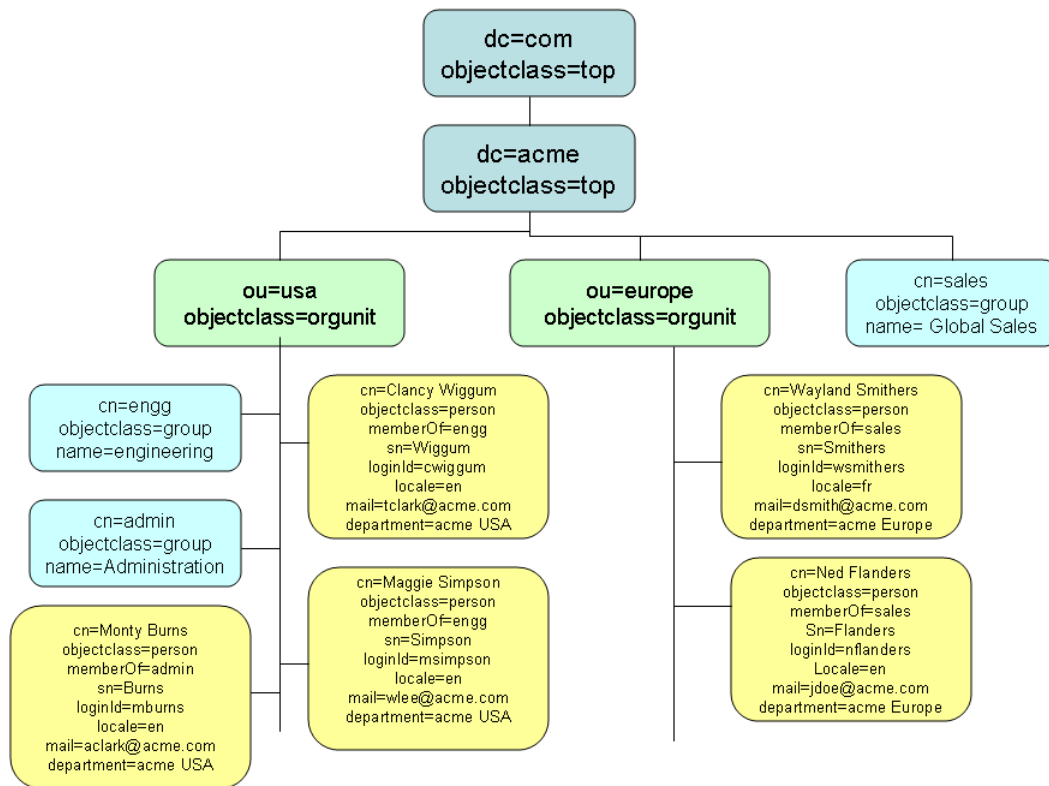
Figure 3-1: Example DIT in LDAP

Distinguished Name for an Object

Every object in the LDAP directory structure has a unique path to its place in the directory. This path is the object's Distinguished Name, or DN. For example, based on the example DIT in Figure 2-1, the DN for the organizational unit "usa" is the following:

```
OU=usa,DC=acme,DC=com
```

The DN for the user "Monty Burns" is the following:

```
CN=Monty Burns,OU=usa,DC=acme,DC=com
```

The DN for the group "engg" is the following:

```
CN=engG,OU=usa,DC=acme,DC=com
```

The DN can contain spaces within an attribute value and between attributes (for example, after the comma separating two attributes).

LDAP Base

The **LDAP** tab on the **Administration** page provides the **LDAP Base** field to identify the Distinguished Name (DN) of the base domain for LDAP synchronization at your site.

The LDAP base domain is the top-level directory to which all other objects in your LDAP directory belong. In essence, the base domain represents your organization. This directory is identified by domain components (DCs), which make up its Distinguished Name (DN). For example, based on the example DIT in Figure 2-1, you would enter the following DN in the **LDAP Base** field as the base domain for the Acme organization:

```
DC=acme,DC=com
```

In some cases, sub-domains are set up in the LDAP directory structure. For example, the Acme organization might have two major divisions, Hardware and Software (not shown in the example), identified as sub-domain components in the directory structure. If Code Insight synchronizes with the Software sub-domain only, the DN for the LDAP base would be the following:

```
DC=software,DC=acme,DC=com
```

Setting Up a User Search

To synchronize only Code Insight users in LDAP to Code Insight, you must set up a user search query that retrieves the users to synchronize from the LDAP server. This process involves properly configuring the **LDAP Base** (described previously in LDAP Base), the **LDAP Search Base**, and **LDAP Search Query** fields on the **LDAP** tab on the **Administration** page. The following topics provide more information about these components used to set up the query:

- LDAP Search Base

- LDAP Search Query

LDAP Search Base

The **LDAP Search Base** value is typically the directory, relative to the LDAP base directory, under which you store all Code Insight objects on the LDAP server. For example, based on the example DIT in Figure 2-1, the **LDAP Search Base** value might be the following, which searches for Code Insight objects belonging only to the "usa" organizational unit under the LDAP base:

```
OU=usa
```

LDAP internally identifies the Distinguished Name, or DN, for the LDAP search base as the **LDAP Base + LDAP Search Base** value. In this example, LDAP recognizes the DN for the search base as `OU=usa,DC=acme,DC=com`.

As another example, to search for all Code Insight objects in the "usa", "europe", and "sales" organizational units, you would leave the **LDAP Search Base** field blank so that your search base defaults to the **LDAP Base** directory. In this case, LDAP recognizes the search base as `DC=acme,DC=com`.

If you leave this field blank, the search is performed at the LDAP base level.

LDAP Search Query

The **LDAP Search Query** uses one or more user attributes to define a subset of the LDAP search base directory; and only the users in this subset are synchronized with Code Insight. Best practice is to create a DIT object in the search base directory, such as a group, that is specific to Code Insight and then make all Code Insight users a part of that object.

---

*Important ▪ Code Insight requires that all users have a valid email address. Even if users meet all the criteria of the LDAP search query, only those users with a valid email address specified as a user attribute on the LDAP server will be synchronized. Consequently, ensure that all Code Insight users have their email address assigned to this attribute on the server and that, on the **LDAP** tab, you have designated the correct label for the attribute as defined on the server (see the "Email" field description in LDAP Tab Field Descriptions).*

The following topics describe more about defining the user search query:

- Sample Search Query

- Sub-tree Search

- Server Paging

Sample Search Query

LDAP search query is entered in the **LDAP Search Query** field on the **LDAP** tab. This query is used to search the **LDAP Search Base** directory on the LDAP server to retrieve only those users that you want to synchronize to Code Insight. Each attribute in a query is listed in parenthesis in the format (`attribute=value`).

For example, based on the sample DIT described in the previous section, suppose all (and only) Code Insight users belong to the "usa" organizational unit. The **LDAP Search Base** node should then be set to usa; and the following query can be used for **LDAP Search Query** to retrieve and synchronize users (entities with the object class of "person") to Code Insight:

    (objectClass=person)

However, suppose that Code Insight users are only those users belonging to the "engineering" group under the "usa" node. The following query can then be used to retrieve and synchronize the appropriate users to Code Insight:

    (&(objectClass=person)(memberOf=CN=engg,OU=usa,DC=acme,DC=com))

Although objectClass and memberOf are the most commonly used filters, a query can filter objects by other attributes, such as "department" in the following example:

    (&(objectClass=person)(department=acme USA))

Sub-tree Search

The **Search Sub-tree** option on the **LDAP** tab controls whether to enable deep searches through subtrees of the path defined by **LDAP Base** + **LDAP Search Base**. While helpful in locating users in certain cases, a deep search can negatively affect performance (and therefore, by default, is not enabled).

Subtree examples in the DIT in Figure 2-1 are the organizational units "usa" and "europe" belonging to DC=acme,DC=com. Suppose that the "usa" subtree also has a subtree called "California" (not shown in the example), which contains users. If the **LDAP Base** is DC=acme,DC=com and the **LDAP Search Base** is usa, the following would occur when a query is executed, depending on the status of the **Search Sub-tree** option:

- If the option is enabled, the query searches for users in both the sub-tree "usa" (the search base) and its subtree "California".

- If the option is disabled, the query searches for users in "usa" but not in its subtree "California".

If a synchronization was previously run with the **Search Sub-tree** option enabled and is then run again with the option disabled, any users previously synchronized from subtrees under the base are assigned a "disabled" status. For example, suppose user "Monty Burns" belongs to "usa" (the search base) and "Karen Smith" belongs to "California" (a sub-tree of the base). When a synchronization is run with **Search Sub-tree** enabled, both "Monty Burns" and "Karen Smith" are synchronized and are active. However, if the option is then disabled and another synchronization is run, both users are synchronized but only "Marty Burns" remains active; "Karen Smith" is flagged as "disabled".

Users who are not LDAP users are not affected by this option. See also Disabled Users.

Server Paging

LDAP and Active Directory support server paging controls the number of records the system is pulling at any given time. Configure the **LDAP Page Size** entries as desired. The default page size is 1000.

> *Note ▪ SunOne Directory Server does not support server paging in certain releases* http://kb.globalscape.com/KnowledgebaseArticle10218.aspx. *If you are using SunOne Directory Server, ensure that server paging is disabled.*

Implementing LDAP in Code Insight

This section explains the basic procedure for implementing LDAP in Code Insight. For detailed descriptions of the fields on the LDAP tab, see LDAP Tab Field Descriptions.

Task   ***To configure Code Insight for LDAP, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **LDAP** tab.

3. Select **Yes** in the **Enable LDAP** field and then complete the remaining fields on the **LDAP** tab. See LDAP Tab Field Descriptions for descriptions of all the fields.

4. (Optional) Select **Test LDAP Server Connection** to ensure that Code Insight is properly connected to the LDAP server. The connection will be tested with the values displayed in the fields on the **LDAP** tab.

5. Do either:

    ● Select **Save** to save the LDAP configuration.

    ● Select **Sync Now** to save your settings and synchronize Code Insight with user data from the LDAP server. If you do not select **Sync Now**, the user synchronization is performed at the time specified in the **LDAP User Sync Frequency** field.

LDAP Tab Field Descriptions

The **LDAP** tab on the **Administration** page enables LDAP user authentication for Code Insight. The tab contains the following columns and fields:

Table 3-3 ▪ LDAP tab Field Descriptions

Section	Column/Field	Description
[LDAP enablement]		This option enables the use of LDAP for your Code Insight system. When LDAP is enabled, the settings used to configure Code Insight for LDAP are made available for editing on this tab. You can use this option to turn off LDAP whenever necessary.
	Enable LDAP	Select **Yes** or **No** to determine if LDAP will be used for user authentication. The default is **No**.
LDAP Connection Details		These settings configure the Code Insight connection to the LDAP server. This connection is required for each synchronization process of LDAP user information to Code Insight and for authentication each time a user logs into Code Insight.
	LDAP URL	Specify the URL of the LDAP server in the following format: `ldap://<ldap_server_host>:<ldap_port>` where `<ldap_server_host>` is either the hostname or IP address of the LDAP server; and `<ldap_port>` is the port on which the server listens for requests. The following is an example URL, which uses the default LDAP server port 389: `ldap://acme.com:389` If using SSL to provide data encryption security for user information passed over the network, specify the `ldaps://` protocol with the port 636, which is the default dedicated port for SSL: `ldaps://acme.com:636` *Note ▪ When the LDAP directory service is Active Directory, requests from users located outside the global catalog's base domain will fail to authenticate if you use the port specified above. This occurs because requests sent to the default LDAP port 389 (or 636 if SSL is used) search for objects only within the global catalog's base domain. To authenticate users from outside the base domain, change the LDAP port to 3268 (or 3269 if SSL is used). Requests sent to this port search for objects in the entire forest.*

Table 3-3 ▪ LDAP tab Field Descriptions (cont.)

Section	Column/Field	Description
LDAP Connection Details (continued)	**Authentication Type**	Select the type of LDAP authentication used to establish a connection with the LDAP server: • **Anonymous**—Code Insight will establish a connection with the LDAP server without the use of user credentials. (When this option is selected, the **LDAP Username** and **LDAP Password** fields in this section are disabled.) This authentication type is generally used for testing purposes. • **Authenticated**—Code Insight requires the user credentials provided in the **LDAP Username** and **LDAP Password** fields to authenticate and establish a connection with the LDAP server.
	LDAP Username	Depending on your LDAP setup, enter either of the following to identify the user used connect to the LDAP server: • The user's login ID, such as `mburns` • The user's Distinguished Name (DN), such as: `CN=Monty Burns,OU=usa,DC=acme,DC=com` For more information about providing the DN, see Distinguished Name for an Object. This identification, along with the associated password (see the next field), is used to authenticate the connection to the LDAP server. Note that the user must have READ permissions to query the LDAP server (and therefore does not need to be an LDAP administrator). This field is disabled if **Anonymous** is selected for **Authentication Type**.
	LDAP Password	Enter the password associated with the user specified for **LDAP Username**. This field is disabled if **Anonymous** is selected for **Authentication Typ**e.
LDAP Query Details		The following fields define the query that identifies the subset of users on the LDAP server to be synchronized to Code Insight. This query is used for the initial synchronization process and for each subsequent synchronization performed per the **LDAP User Sync Frequency** value.

Table 3-3 ▪ LDAP tab Field Descriptions (cont.)

Section	Column/Field	Description
LDAP Query Details (continued)	**LDAP Base**	Specify the Distinguished Name (DN) of the LDAP base domain in the Directory Information Tree (DIT) on your LDAP server. This domain is the top-level directory to which all other objects in the directory structure belong; it typically represents your organization. The base domain is identified by domain components (DCs), which make up its DN. For example, the base domain in the example DIT in Figure 2-1 is the following:
		`DC=acme,DC=com`
		In some cases, a sub-domain can be a part of the base domain:
		`DC=software,DC=acme,DC=com`
		For more information, see LDAP Base.
	LDAP Search Base	Specify the DIT directory, relative to the LDAP base directory, under which you store all Code Insight objects on the LDAP server and from which you search for Code Insight users.
		In reference to the example DIT in Figure 2-1, if you enter `OU=usa` for the search base, all searches for user information will be performed below the directory "usa". (LDAP internally identifies the DN for this directory as the **LDAP Base** + **LDAP Search Base** value.) For more information, see LDAP Search Base.
		If you leave this field blank, the search is performed at the LDAP base level.
	LDAP Search Query	Specify the search query used to retrieve the users from **LDAP Search Base** directory to synchronize to Code Insight. Each attribute in a query is listed in parenthesis in the format (*attribute=value*), such as in the following, which searches for only those users belonging to the "engineering" group under the "usa" node:
		`(&(objectClass=person)(memberOf=CN=engg,OU=usa,DC=acme,DC=com))`
		For other search query examples, see LDAP Search Query.
	Use Paging	Select **Yes** if the LDAP server has paging enabled for synchronization results. If you select **Yes**, the **LDAP Page Size** field is enabled, enabling you to customize the page size.
		Select **No** if the server does not have paging enabled. If you select **No**, the server sends 1000 elements per page by default unless this behavior is changed at the organization level on the LDAP server.
	LDAP Page Size	Indicate the page size you want for the synchronization results. The default page size is 1000 elements.

Table 3-3 ▪ LDAP tab Field Descriptions (cont.)

Section	Column/Field	Description
LDAP Query Details (continued)	**LDAP User Sync Frequency**	Specify the frequency at which Code Insight will synchronize user data with the LDAP server:
		● **Never**—Select this option to disable the automatic user synchronization. A synchronization occurs only if the user clicks the **Sync Now** button. For all other values, automatic user synchronization is enabled per the configured frequency. (This is the default value.)
		● **Hourly**—Enter an integer value representing the number of hours between user synchronizations.
		● **Daily**— Select a time at which the user synchronization will run every day.
		● **Weekly**—Select a day of the week and a time of the day when the user synchronization will run each week.
	Search Sub-tree	Select this checkbox to enable deep searches through the subtrees of the path defined by **LDAP Base** + **LDAP Search Base**. Note that, while helpful in locating users in certain cases, a deep search can negatively affect performance (and therefore, by default, is not enabled). For more information, see Sub-tree Search.
LDAP User Property Mappings		The following information maps LDAP attribute labels to their corresponding labels in Code Insight (the field names shown below). These mappings are used for LDAP synchronization to Code Insight and for user authentication each time a user logs into Code Insight.
	Login	Enter the user attribute label on your LDAP server corresponding to the user **Login** field in Code Insight. This is the same attribute that the user will use to log into Code Insight.
	First Name	Enter the user attribute label on your LDAP server corresponding to the user **First Name** field in Code Insight.
	Last Name	Enter the user attribute label on your LDAP server corresponding to the user **Last Name** field in Code Insight.
	Email	Enter the user attribute label on your LDAP server corresponding to the user **Email** field in Code Insight.
		Note ▪ Only those users with a valid email address specified as a user attribute on the LDAP server will be synchronized. Therefore, ensure that you have entered the correct label here for the email attribute on your LDAP server and that each user has valid email for this attribute on the server. See LDAP Search Query for more information.

Table 3-3 ▪ LDAP tab Field Descriptions (cont.)

Section	Column/Field	Description
LDAP User Property Mappings (continued)	Login Filter	Specify a filter for the user-login search performed in the LDAP search base location. For example, the value `(sAMAccountName={0})`, when used against the **LDAP Search Query** results, searches for each entry where the `sAMAccountName` is equal to the user login name.

Configuring Code Insight to Use Single Sign-On

Single sign-on (SSO) is an authentication service that enables a user to use one set of credentials (usually a name and password) to access multiple applications. This service involves an exchange of SAML (Security Assertion Markup Language) protocol messages between the user, the identity provider, and the service provider.

The Identity Provider (also called an IdP) is any SSO service, such as Okta, Ping Federate, and others, offering SAML authentication services. The Service Provider (also called an SP) is an application, such as Code Insight, that is configured to participate in the SSO service. When a Service Provider user logs in using credentials for an SSO session, a SAML message is sent to the Identity Provider, requesting user authentication. If the user password is valid, the Identity Provider returns a SAML message, stating that the user is logged in at the Identity Provider. The user, in turn, is logged into the Service Provider.

A Code Insight System Administrator can configure Code Insight as a Service Provider in an SSO session. The following sections provide instructions for doing so:

- Prerequisite Tasks for Configuring Code Insight for SSO

- Configuring Code Insight for SSO

- Log In Using SSO Credentials

Prerequisite Tasks for Configuring Code Insight for SSO

Perform the following tasks before configuring Code Insight for SSO:

- Configure HTTPS on the Code Insight Server

- Set Up SSO Users

Configure HTTPS on the Code Insight Server

The HTTPS communication protocol must be used to exchange SAML messages between the SP and IdP. For instructions on configuring HTTPS on the Code Insight server, see Enabling Secure HTTP Over SSL in the "Installing Code Insight" chapter.

The keystore that you use to configure HTTPS can be used for SSO configuration. Alternatively, you can create a separate keystore for SSO, using the same instructions found in Enabling Secure HTTP Over SSL.

Set Up SSO Users

You can define SSO users for Code Insight with or without LDAP.

With LDAP

If you intend for SSO to integrate with your LDAP server for user access to Code Insight, follow these rules:

- Make sure that Code Insight and the Service Provider are configured for the LDAP server. For instructions to configure Code Insight, see Configuring Code Insight for LDAP.

  To configure the Service Provider, follow the Service Provider instructions.

- When setting up users on the LDAP server, ensure that the user's login is the user's email address.

- Synchronize users from the LDAP server to the Identity Provider first, using the Identity Provider's instructions. Then synchronize the users from the LDAP server to Code Insight. See Configuring Code Insight for LDAP.

Without LDAP

If you do not use LDAP, you must manually create the SSO users both in Code Insight (see Managing Users) and at the Identity Provider site, ensuring that the user information is the same in both locations.

Ensure that the user's login is the user's email address.

Configuring Code Insight for SSO

Follow these steps for configuring Code Insight for SSO:

- Step 1: Copy the Directory That Will Contain Provider Metadata

- Step 2: Prepare the Environment Properties File

- Step 3: Configure the SSO Common Properties File

- Step 4: Customize the Sample Service Provider Metadata File

- Step 5: Obtain the Identity Provider Metadata File

Note that, in these instructions, *SCA_install_home* refers to the Code Insight installation location.

Step 1: Copy the Directory That Will Contain Provider Metadata

Copy the `security` directory from *SCA_install_home*/samples/sso/config/core to *SCA_install_home*/config/core.

This directory will serve as the storage location for the Service Provider and Identity Provider metadata files, as described in Step 4: Customize the Sample Service Provider Metadata File and Step 5: Obtain the Identity Provider Metadata File.

Step 2: Prepare the Environment Properties File

This step prepares the `env.properties` file to enable SSO on the Code Insight server.

Task   ***To prepare the "env.properties" file, do the following:***

1. Copy the env.properties file from *SCA_install_home*/samples/sso/config to *SCA_install_home*/config/core.

2. In a text editor, open the *SCA_install_home*/config/core/env.properties file, and ensure that the value of the following property to **sso**.

    spring.profiles.active=**sso**

3. Save the file.

Step 3: Configure the SSO Common Properties File

This step configures the core.sso.common.properties file to enable SSO on the Code Insight server.

Task   ***To prepare the "core.sso.common.properties" file, do the following:***

1. Copy the core.sso.common.properties file from *SCA_install_home*/samples/sso/config to *SCA_install_home*/config/core.

2. In a text editor, open the *SCA_install_home*/config/core/core.sso.common.properties file. The following shows the file contents.

    ```
    ## this file contains all sso placeholder values.
    saml.keystore=file:///c:/<path>/keystore.jks
    saml.keystore.password=keysore_password
    saml.keystore.alias=keystore_alias
    saml.keystore.alias.password=keystore_alias_password

    # for extendedMetadata configuration
    saml.metadata.local=true
    saml.metadata.alias=metadata_alias
    saml.metadata.idpDiscoveryEnabled=false
    saml.metadata.idpDiscoveryURL=
    saml.metadata.idpDiscoveryResponseURL=
    saml.metadata.ecpEnabled=false
    saml.metadata.securityProfile=metaiop
    saml.metadata.sslSecurityProfile=pkix
    saml.metadata.sslHostnameVerification=default
    saml.metadata.signingKey=keystore_alias_password
    saml.metadata.signingAlgorithm=null
    saml.metadata.signMetadata=false
    saml.metadata.encryptionKey=keystore_alias
    saml.metadata.tlsKey=private_key_alias_for_SSL/TLS_client_authentication
    #private Set<String> trustedKeys=
    saml.metadata.requireLogoutRequestSigned=false
    saml.metadata.requireLogoutResponseSigned=false
    saml.metadata.requireArtifactResolveSigned=false
    saml.metadata.supportUnsolicitedResponse=true
    #for SP
    saml.entity.id=ww:xx:yy:zz
    saml.base.url=https://myhost.mycompany.com:8443
    ```

3. Update the properties (highlighted above) required for Service Provider security and identification, and then save the file. The properties that you need to edit or that require explicit configuration are described in this table:

SSO Property	Description
saml.keystore	Enter the path and name of the keystore that you created for SSO. This can be the same keystore that you are using for HTTPS or a different one. See Configure HTTPS on the Code Insight Server in the "Installing Code Insight" chapter for more information.
saml.keystore.password	Enter the password for the keystore.
saml.keystore.alias	Enter the alias defined for the private key contained in the keystore.
saml.keystore.alias.password	Enter the password for the private key alias.
saml.metadata.alias	Enter the metadata alias.
saml.metadata.idpDiscovery URL	Leave this field blank. Do not enter `null`.
saml.metadata.idpDiscovery ResponseURL	Leave this field blank. Do not enter `null`.
saml.metadata.signingKey	Enter the password for the private key alias.
saml.metadata.encryptionKey	Enter the alias defined for the private key contained in the keystore.
saml.metadata.tlsKey	Enter the alias of private key generated for SSL/TLS client authentication.
saml.entity.id	Enter a unique identifier for your Code Insight server as a Service Provider. The recommended value is the hostname for the Code Insight server. Note that, even though the server's hostname is the recommended value, the entity ID is an immutable value identifying the Service Provider in an SSO session; it is not used to identify a location.
saml.base.url	The HTTPS URL handling the Service Provider's user sign-in requests. This is usually the URL for your Code Insight server in `HTTPS://myhost.mycompany.com:port` format. Note that the default port for the Code Insight server is 8443.

Step 4: Customize the Sample Service Provider Metadata File

This step customizes the sample Service Provider metadata file for your Code Insight server.

Task　　**To customize the sample Service Provider metadata file, do the following:**

1. In a text editor, open the *SCA_install_home*/config/core/security/SPMetadata.xml file.

2. Update the following properties, and save the file:

SSO Property	Description
entityID="*ENTITY_VALUE*"	Replace ENTITY_VALUE with the same entity ID as the one you provided the env.properties file in Step 2: Prepare the Environment Properties File.
SingleLogoutService... *FULLY_QUALIFIEDHOSTNAME...*	Replace *FULLY_QUALIFIEDHOSTNAME* with the fully qualified hostname of the Code Insight server.
AssertionConsumerService... *FULLY_QUALIFIEDHOSTNAME...*	Replace *FULLY_QUALIFIEDHOSTNAME* with the fully qualified hostname of the Code Insight server.
requestSigned	Set to **true** to indicate that the Service Provider must sign authentication requests.
wantAssertionSigned	Set to **true** to indicate that the Service Provider requires signed assertions received from Identity Provider.

Step 5: Obtain the Identity Provider Metadata File

This final step in setting up SSO for Code Insight is to obtain the Identity Provider metadata file. The Identity Provider might require that you send the Code Insight SPMetadata.xml file (set up in Step 4: Customize the Sample Service Provider Metadata File) in order to provide the Identity Provider metadata file.

Alternatively, you might be required to generate the Identity Provider metadata file using the Identity Provider UI. You will need to provide the single-sign-on URL for Code Insight (also specified in the SPMetadata.xml):

    https://*myhost.mycompany.com*:8443/codeinsight/saml/SSO

Task　　**To obtain the Identity Provider metadata, do the following:**

1. Follow the Identity Provider's instructions for obtaining the Identity Provider metadata.

2. Once you obtain the Identity Provider metadata, save it as IDPMetadata.xml in the *SCA_install_home*/config/ core/security directory.

Log In Using SSO Credentials

Once you complete the steps described in this section, Code Insight users defined as SSO users should be able to log in to an SSO session managed by the Identity Provider and obtain access to Code Insight.

Configuring Extended Logging

Code Insight integrates with the Splunk Enterprise to provide extended logging capabilities. Splunk has the ability to capture, index, and correlate real-time Code Insight log data in a searchable repository and, using this repository, generate graphs, reports, alerts, dashboards, and visualizations. These interpretations of log data enable you to identify operational and security issues quickly and efficiently.

For details on how to integrate Code Insight with Splunk, refer to the following KB article in the Revenera Community:

https://community.flexera.com/t5/FlexNet-Code-Insight-Customer/FlexNet-Code-Insight-v7-Integration-with-Splunk/ta-p/133655

Managing Scan Profiles

The following topics describe how to manage scan profiles:

- About Scan Profiles

- Creating or Editing Scan Profiles

- Viewing the Settings for a Specific Scan Profile

- Description of the Scan Profile Settings

- Creating Exclusion Patterns for Scan Profiles

About Scan Profiles

A scan profile is a set of scan settings that are grouped together and then applied at scan time. A given project is associated with a scan profile by default, but the Project Administrator can assign the project to a different scan profile at any time, as described in "Applying a Scan Profile to the Project" in the "Using Code Insight" chapter in the *Code User Guide*.

Code Insight provides the following standard (pre-defined) scan profiles. You can modify these profiles (with the exception of the Standard Scan Profile) and assign them to projects. (See Description of the Scan Profile Settings for the default settings used by each of these profiles.)

- **Basic Scan Profile (without CL)**—Defines a scan that uses Automated Analysis to detect evidence of open-source software (OSS) and third-party code in your codebase and generate an inventory of the findings. This scan does not perform exact-file or source-code matching and therefore does not use the Compliance Library (CL).

- **Standard Scan Profile**—Defines a scan that includes the basic scan features but also performs exact-file matching (that is, identifies codebase files that have an exact MD5 match in the CL). This scan requires the CL.

📄

Note • This scan profile cannot be modified. Its settings are used as the template when creating a scan profile.

- **Comprehensive Scan Profile**—Defines a scan that includes the basic scan features but also performs exact-file and source-code matching. (Source-code matches are strings in the codebase files that have an exact match to content in files in the CL). This scan requires the CL.

In most cases, the standard scan profiles are enough to get started. However, if they do not meet your needs, you can create your own custom scan profiles.

Creating or Editing Scan Profiles

When a scan profile is created, the data from the Standard Scan Profile is copied, including any search terms and exclusions. However, you can update any of this information for the scan profile you are creating.

You can also edit information in any custom and standard scan profiles (except the Standard Scan Profile). Note the following:

- Scan profiles changes can result in costly rescans, especially when settings involved with source-code matching change. For details, refer to the "Rescanning Your Codebase" in the "Using Code Insight" chapter in the *Code User Guid*e.

- Scan profiles changes do not affect the current scan. Changes are applied to the next scheduled scan.

The following procedure describes how to create or edit a scan profile.

Task   ***To create or edit a scan profile, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Scan Profiles** tab to open the list of existing scan profiles.

3. Perform either action:

    - To create a new scan profile, click **Add Scan Profile** above the list.

    - To edit an existing scan profile, locate the profile and click the **Edit** ✐ icon in the **Actions** column of the profile row.

    The **Create** (or **Edit**) **Scan Profile** dialog is displayed.

4. Complete the fields on the dialog. See the later section, Description of the Scan Profile Settings, for a description of each setting.

5. Click **Save** to save the scan profile.

Viewing the Settings for a Specific Scan Profile

Use the following procedure to view the settings (in read-only format) for an existing scan profile.

Task   **To view scan-profile settings in read-only format:**

1. Refer to steps 1 and 2 in the previous section, About Scan Profiles, for instructions on accessing the **Scan Profiles** tab.

2. From the list of scan profiles on the tab, locate the scan profile whose settings you want to view.

3. Click the **View** 🖹 icon in the **Actions** column of the profile row.

The **View Scan Profile** dialog is opened, listing the settings defined for the profile. See the next section, Description of the Scan Profile Settings, for a description of each setting.

Description of the Scan Profile Settings

The following table describes the settings that define a scan profile, standard or custom. It also shows the default value for a given setting in each of the standard scan profiles provided by Code Insight. For example, to view the default settings enabled for:

● The **Basic Scan Profile (without CL)**, see the "Basic Default" column in the table.

● The **Standard Scan Profile**, see the "Standard Default" column.

● The **Comprehensive Scan Profile**, see the "Comprehensive Default" column.

Note ▪ The Comprehensive Scan Profile and Standard Scan Profile rely on data stored in the Compliance Library (CL) to detect evidence for Exact Matches and Source Code Matches.

Table 3-4 ▪ Scan Profile Dialog

Field	Description	Basic Default	Standard Default	Comprehensive Default
Name	Enter or edit the profile name.	**Basic Scan Profile**	**Standard Scan Profile**	**Comprehensive Scan Profile**
Perform Package/ License Discovery in Archives	Select this option to have the Scan Server recursively perform package discovery and license detection within all archive files encountered in the project codebase. By default, this option is selected.	Selected	Selected	Selected

Table 3-4 ▪ Scan Profile Dialog (cont.)

Field	Description	Basic Default	Standard Default	Compre-hensive Default
Dependency Support	Determine the level of dependency scanning to be performed by the Scan Server. The available options include: ● **No Dependencies**: Only top-level inventory items are reported without any dependencies. (Default) ● **Only First Level Dependencies**: Only first-level (or direct) dependencies are reported along with top-level inventory items. ● **All Transitive Dependencies**: All first-level and transitive dependencies are reported along with top-level inventory items. The Scan Server calls out to the relevant package management repository to obtain transitive dependency information. For a description of Code Insight dependency support for supported ecosystems, see "Automated Analysis" in the *Code Insight User Guide*.	**No Depend-encies**	**No Depend-encies**	**No Depend-encies**
Automatically Add Related Files to Inventory	Select this option to have the system associate additional files to existing inventory items based on the data available in automatic detection rules.	Selected	Selected	Selected

Table 3-4 ▪ Scan Profile Dialog (cont.)

Field	Description	Basic Default	Standard Default	Comprehensive Default
Rescan Options	By default, when a user initiates a regular rescan (that is, not a forced full rescan), only those files that have changed since the last scan are scanned. However, certain Code Insight events that have occurred since the previous scan can result in a rescan of all files (a full rescan). For a description of these events, see "Default Scan Behavior" in the *Code Insight User Guide*. These options are used to override this default rescan behavior so that, even if any of the events that would normally call for a full rescan have occurred, all rescans will skip unchanged files and scan changed files only.			
	Do not rescan files that have not changed since previous scan	Select this option so that rescans always skip unchanged files and scan only those files that have changed since the last scan (even if events have occurred since the last scan that call for a full rescan).	Not selected	Not selected	Not selected
	Apply this option to:	If the **Do not rescan files...** option is selected, further clarify *which* unchanged files to skip during the rescan: ● All unchanged files ● Only unchanged files marked as reviewed ● Only unchanged files associated with inventory ● Only unchanged files that are both marked as reviewed and associated with inventory	N/A	N/A	N/A
Exact Matches	Select this option to enable the detection and recording of scanned files that exactly match entire-file data in the Compliance Library (CL).	**Disabled**	**Enabled**	**Enabled**

Table 3-4 ▪ Scan Profile Dialog (cont.)

Field		Description	Basic Default	Standard Default	Comprehensive Default
Source Code Matches		Select this option to enable the detection and recording of any source-code snippets in the scanned files that match data in the Compliance Library (CL).	**Disabled**	**Disabled**	**Enabled**
		If you enable this source-code matching, specify any of the following additional parameters for the matching process.			
	Include System-Identified Files	Select this option if you want the Scan Server to perform source-code matching for files that have already been associated with one or more inventory items during automated analysis.	N/A	N/A	Selected
	Include Files with Exact Matches	Select this option if you want the Scan Server to perform source-code matching for files that have already been identified as having exact-file matches in the CL.	N/A	N/A	Selected
	Minimum Source Code Matches	Enter the minimum number of source-code matches that the scan needs to detect in a given codebase file before reporting the file as having such matches. (A *source-code match* is a snippet of code in a codebase file that matches an open-source code snippet found in the CL data.)	N/A	N/A	**1**
		Enter a new minimum value from 1 to 32767. (The default is 3.)			
		For example, if this value is increased to 10, ten code snippets in a given codebase file must match data in the CL before the scan reports the file as having source-code matches.			
		In general, the higher this value, the fewer source-code matches an analyzer has to review.			
Search Terms		Provide a list of search terms to be used in the scan. Use the **+** button to add a term and the **-** button to remove a term.	Standard terms listed	Standard terms listed	Standard terms listed
Scan Exclusions		Provide a list of file extensions to be excluded from the scan. Use the **+** button to add an exclusion term and the **-** button to remove an exclusion. See "Creating Exclusion Patterns for Scan Profiles" in the *Code Insight Installation & Configuration Guide* for further instructions.	Standard exclusions listed	Standard exclusions listed	Standard exclusions listed

Creating Exclusion Patterns for Scan Profiles

Code Insight provides the ability to create exclusion patterns for use in your scans and to add them to your scan profile in **Create** (or **Edit**) **Scan Profile** page. This section provides information about the syntax required when creating exclusion patters and examples of valid exclusion patterns.

Code Insight uses Apache Ant path-style syntax to exclude files during scanning. Patterns are paths that are relative to a base directory. Only files found in or below the base directory are considered for exclusion. For in-depth information about *ant* exclusion patterns, see https://ant.apache.org/manual/dirtasks.html.

Note ▪ *Exclusion patterns are not validated.*

Using the Single Asterisk (*) and Question Mark (?)

Using a single asterisk (*) matches zero or more characters in a single file name or directory name. Using the question mark (?) matches one character.

If you create an exclusion pattern of ***.xml**, and add it to the list of **Scan Exclusions**, your scan will exclude files such as x.xml, FooBar.xml, and codeinsight.xml, but not codeinsight.jar because it does not end with .xml. In other words, codeinsight.jar will display in scan results (if it is in your codebase) because it does not match ***.xml**.

If you add the exclusion pattern **aa/***, your scan will exclude files such as aa/x.xml or aa/bb but will include aa/bb/x.xml because it does not match **aa/***. That is, the * can match only a single name—that of a directory (one directory deep) *or* a file— not both names, as in bb/x.xml, which includes a directory name (bb) and a file name (x.xml).

If you create an exclusion pattern of **?.codeinsight**, your scan will exclude files such as x.codeinsight and A.codeinsight, but will include xx.codeinsight or aaa.codeinsight because neither has just one character before .codeinsight.

Note ▪ *You can combine asterisks (*) and question marks (?) in your exclusion patterns.*

Using Double Asterisks

Double asterisks (**) span multiple directory paths. If you create an exclusion pattern of ****/codeinsight**, the files in the aa/bb/cc/codeinsight directory structure will be excluded from the scan.

Example Exclusion Patterns

The following shows some example patterns used to exclude files from scans.

Table 3-5 ▪ Example Exclusion Patterns

Example Pattern	Description
**/SVN/*	Excludes all the files in the `SVN` directories that are located anywhere in the directory tree (for example, `SVN/Repository` and `apache/SVN/Entries`). However, `org/apache/SVN/foo/bar/Entries` will be included in the scan because the `/foo/bar/Entries` component is not matched by `/*`, which represents only a single directory name (one directory deep) or a single file name.
/ePortal-2.0/src/	Excludes all the files in the `/ePortal-2.0/src/**` directory tree (for example, `/ePortal-2.0/src/index.html` and `/ePortal-2.0/src/test.xml`). However, `/ePortal-2.0/xyz.java` will be included in the scan because the `/src` component is missing.
**/images	Excludes all files in `images` directories located anywhere in the directory tree. The exception to this pattern is **/.git. See Note About Excluding the .git Directory from Scans for more information.

Keep the following in mind as you specify exclusion patterns:

● If a pattern ends with / or \, double asterisks (**) are appended. For example, `codeinsight/data/` is interpreted as `codeinsight/data/**`.

● Exclusion patterns are not validated by Code Insight. You must test your patterns externally.

Note About Excluding the .git Directory from Scans

Basically, the configuration file (`config` or `gitconfig`) contained in a `.git` directory for a Git repository is always scanned—and is the only file scanned—whether or not you exclude the `.git` directory from scans. The following explains this behavior:

● If you add the `.git` directory to the **Scan Exclusion** list to prevent it from being scanned, the configuration file contained in the folder is still scanned because it is required by Automated Analysis to detect components in the Git repository.

● If the `.git` directory is not included in the **Scan Exclusion** list (that is, you intend for the files in the `.git` directory to be scanned), the configuration file is scanned, but no other files in the directory are scanned. Scans ignore the remaining files in the directory because they contain data that is not required in the detection of components and evidence in the Git repository—data such as the repository's commit history as well as its log information and metadata.

Enabling Calculation of SHA-1 Digests for Scanned Files

Code Insight automatically calculates the MD5 digest for project files during a scan as a means to determine which files have changed between scans and to "exact match" scanned files with OSS or third-party files (whose digests are stored in the Code Insight Compliance Library).

However, Code Insight can be configured to also calculate file SHA-1 digests during scans should your site want these digests.

Note ▪ MD5 digests are always calculated whether or not SHA-1 support is enabled.

The enablement or disablement of SHA-1 support requires that a database administrator set a global property in the Code Insight database. (By default, this support is enabled when a customer site first installs Code Insight 2021 R3 or later or migrates from a pre-2021 R3 version.)

The following topics describe how the database administrator enables or disables SHA-1 support for Code Insight and, if it is enabled, how users can view the latest digest values for scanned files.

- Enabling the SHA-1 Support

- Disabling SHA-1 Support

- Viewing the SHA-1 Value for Project Files

- SHA-1 Support When Installing or Migrating to the Latest Code Insight Version

Enabling the SHA-1 Support

Code Insight's ability to calculate SHA-1 digests for files during project scans is controlled by the `scan.digest.sha1.enabled` property located in the `pas_global_properties` table in the Code Insight database. This section describes how the database administrator enables SHA-1 support so that SHA-1 digests are automatically calculated for all files during scans (standard or remote).

Note ▪ If you have installed a new instance of the current Code Insight version or have migrated from a pre-2021 R3 instance to the current version, SHA-1 support is enabled by default in the new instance. See SHA-1 Support When Installing or Migrating to the Latest Code Insight Version for additional details.

Task   **To enable SHA-1 support for your Code Insight instance, do the following:**

Execute this command against the Code Insight database:

```
UPDATE PAS_GLOBAL_PROPERTIES SET VALUE_ = 'true' WHERE KEY_ = 'scan.digest.sha1.enabled';
```

Refer to the next sections for describe how SHA-1 digests for files are handled during scans after SHA-1 support is enabled.

Standard scans with SHA-1 support enabled

The following occurs when a Code Insight standard scan is run after SHA-1 support is enabled. (A *standard scan* is performed by a Scan Server on a project codebase that is uploaded to the Scan Server itself.)

- **When SHA-1 support is enabled before the initial scan of a codebase**—During the initial scan, SHA-1 values are calculated for all files and updated to the `PSE_SCANNED_FILES` table in Code Insight database. Additionally, all files are scanned.

- **When SHA-1 support is enabled anytime after the initial scan of an existing codebase**—During the first scan after SHA-1 enablement, SHA-1 values are calculated for all files (new and existing) and updated to the `PSE_SCANNED_FILES` table. (While SHA-1 values are calculated and updated to the database for all files, only new files and those files that were modified since the last scan are actually scanned.)

- **For each rescan thereafter**—SHA-1 digests are calculated for only new files and those existing files that were modified since the last scan. Additionally, only new files and modified existing files are (re)scanned.

Remote scans with SHA-1 support enabled

The following occurs when a Code Insight remote scan is run after SHA-1 support is enabled. (A *remote scan* is performed by a Code Insight scan-agent plugin on a remote file system. Scan results, including file information, are sent to an associated project on the Core Server.)

- **When SHA-1 support is enabled before the initial scan of new file system**—During the initial scan, SHA-1 values are calculated for all files and updated to the `PSE_REMOTE_SCANNED_FILES` table in the Code Insight database. Additionally, all files are scanned.

- **When SHA-1 support is enabled at a time after the initial scan of an existing project**—During the first scan after SHA-1 enablement, SHA-1 values are calculated for all files—every existing file (modified or not) and every new file—and updated to the `PSE_REMOTE_SCANNED_FILES` table. Additionally, all files are (re)scanned.

- **For each rescan thereafter**—SHA-1 digests are re-calculated for all existing files (modified or not), calculated for any new files, and then updated to the `PSE_REMOTE_SCANNED_FILES` table. Additionally, all files are (re)scanned.

Disabling SHA-1 Support

This section describes how the Code Insight database administrator configures the `scan.digest.sha1.enabled` property (located in the `PAS_GLOABL_PROPERTIES` table in the Code Insight database) to disable SHA-1 support. When support is disabled, SHA-1 digests are no longer calculated for files during the scans.

Task   ***To disable SHA-1 support for your Code Insight instance, do the following:***

Execute this command against the Code Insight database:

```
UPDATE PAS_GLOBAL_PROPERTIES SET VALUE_ = 'false' WHERE KEY_ = 'scan.digest.sha1.enabled';
```

The following describes how SHA-1 digests for files are handled during scans after SHA-1 support is switched from enabled to disabled. (For a description of standard and remote scans, see Enabling the SHA-1 Support.)

- **Each time a *standard* rescan is run on a codebase**—Any existing file that has been modified since the previous scan (or is a new file) has its SHA-1 value set to `NULL` in the `PSE_SCANNED_FILES` table and is (re)scanned. Existing files that have *not* been modified since the previous scan retain their current SHA-1 value (either a digest or `NULL`) in the table and are not rescanned.

- **Each time a *remote* rescan is run on a file system**—The SHA-1 values for all files—all existing files, modified or not, and any new files— are set to `NULL` in the `PSE_REMOTE_SCANNED_FILES` table. Additionally, all files are scanned.

Viewing the SHA-1 Value for Project Files

Users can use the following Code Insight APIs to view the SHA-1 digest for a file:

- **Get details of a file by ID** (`files/{fileId}`)—Retrieves attributes for the specified file.

- **Fetch all scanned files for a project** (`/projects/{projectId}/allscannedfiles`)—Retrieves attributes for every file scanned in the specified project.

For more information about Code Insight REST interface, see the *Rest API Guide* Swagger documentation available from the **Help** menu.

Note ▪ Currently, SHA-1 digests for files are not shown in the Code Insight Web UI.

SHA-1 Support When Installing or Migrating to the Latest Code Insight Version

The following sections describe how SHA-1 support is handled when a new instance of the latest Code Insight version is installed or when a previous Code Insight version is migrated to the latest version.

Installing a New Instance of the Latest Insight Version

When you install a new instance of the latest Code Insight version, SHA-1 support will be enabled by default. The SHA-1 digests will be calculated during the initial scan (standard or remote) of files in the projects that you create in the new instance; and these digests will be updated to the Code Insight database. However, if SHA-1 support is disabled at some point before the initial scan, the SHA-1 values for files will be set to NULL.

Migrating to the Latest Version

By default the SHA1 property will be enabled in the upgraded version. SHA1 values for the files will be NULL if the property was disabled in the earlier version (or was unavailable, as in a pre-2021 R3 version). If the property was enabled in the earlier version, the existing SHA-1 values are retained during the migration.

Setting Project Defaults

The settings on the **Project Defaults** tab on the **Administration** page work provide a convenient way to default fields used to configure new projects to ensure consistency and enable an easier project creation experience for users.

Task   ***To set project defaults, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ▤ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. Select the **Project Defaults** tab.

3. Update global default values as needed, using the information in the Project Default Descriptions table.

Project Default Descriptions

The following table lists the project default descriptions.

Table 3-6 ▪ Project Defaults

Category	Field	
General Options	These options set defaults for project creation and assign default users to project roles. Users can change these defaults when creating a project or when editing a project or its users using **Manage Project | Edit Project | General** or **Manage Project | Edit Project | Edit Project Users** on the project **Summary** tab.	
	Project Visibility	Select the default for visibility status—**Public** or **Private**—for projects. (The initial system default is **Public**.)
		Any user in the system read-only access to a public project. To what degree a user can interact with the project depends on whether the user has a project role and what the role is—Project Administrator, Analyst, or Reviewer.
		However, private projects are hidden from all users except the Project Contact and those users assigned as Project Administrators, Analysts, Reviewers, or Observers of the project. Additionally, project and vulnerability ID searches will not return private projects unless the user performing the search has the permissions to see a given private project.
	Project Risk	Select the default risk value (**Low**, **Medium**, or **High**) for projects. To edit, select another value from the dropdown list. The initial system default is **Medium**.
	Project Users	Click the **Edit Project Users** link to open the **Edit Default Project Users** page. From here you assign project roles—Analysts, Reviewers, and Observers—that will default for any new project created (but which can then be edited at the project level). See the "Edit (Default) Project Users Page" in the in the online help or the *Code Insight User Guide* for details.

Table 3-6 ▪ Project Defaults (cont.)

Category	Field	
General Options (continued)	**On the data import or rescan, delete inventory with no associated files**	This option determines whether "empty" system-generated inventory items are deleted in the target project during project imports and rescans. Empty inventory items have no associated files. ● **Selected**—Deletes empty inventory items from the target project during project imports and rescans. Only inventory items with associated files are retained/created. ● **Unselected**—Retains/creates all inventory items—with or without matching associated files in the target codebase—in the target project during imports and rescans. For example, you might want to retain inventory items to save their analysis details. (Users will need to manually delete inventory that is not applicable in the current project.) This configuration (unselected) is required when importing a scanned codebase into an inventory-only project, which has no codebase, to ensure inventory is generated in the target project.
Scan Settings		These options identify the default Scan Server and scan profile for projects. Users can change these settings when creating a project or when editing a project using **Manage Project	Edit Project	Scan Settings** from the project **Summary** tab.
	Scan Profile	Select the scan profile to default for projects. Click ⓘ to view the details of the scan profile.
	Scan Server	Select the Scan Server to default for projects. Note that only those Scan Servers in an "enabled" state are available for selection (see Adding or Editing Scan Servers or Checking Server Status). If only one Scan Server has been identified to the system, this server is automatically selected as the default.
Automated Inventory Publish Options		These options configure defaults for automatically publishing project inventory as part of the project scan process. Users can change these settings at the project level by navigating to the project **Summary** tab and selecting **Manage Project	Edit Project	Scan Settings**. If the **Auto-publish system-created inventory items meeting this minimum Confidence Level** is selected to enable auto-publication, the other auto-publish options are made available.

Table 3-6 ▪ Project Defaults (cont.)

Category	Field	
Automated Inventory Publish Options (continued)	**Auto-publish system-created inventory items meeting this minimum Confidence Level**	Select this option to enable the auto-publication of system-generated inventory items. (By default, the option is selected.) Then select the minimum Inventory Confidence level required to determine which items to auto-publish: ● **Low**—Automatically publish all system-generated inventory. ● **Medium**—Automatically publish only those system-generated inventory items with Medium and High confidence levels. ● **High**—Automatically publish only those system-generated inventory items with a High confidence level. For a description of the Confidence levels and how they are used, see Inventory Confidence.
	Do not auto-publish inventory items with an undetermined license	Select this option to *not* auto-publish any system-generated inventory item with an undetermined license (that is, an inventory item whose **License** value is **I don't know**). An undetermined license can occur under the following conditions: ● The scan was not able to identify a license for the given component during the scan and therefore set the **I don't know** license value. ● The inventory item has multiple possible disjunctive licenses (for example, "GPLV2 or MIT"). However, the scan could find no evidence of the desired selected license and therefore set the **I don't know** license value. ● The inventory item has multiple possible conjunctive licenses (for example, "GPLv2 and MIT"). However, since Code Insight currently supports only a single selected license, the scan automatically set the **I don't know** value for the inventory item. This option is available only if **Auto-publish system-created inventory items meeting this minimum Confidence level** is selected. By default, when you first open Code Insight instance after it has been installed or migrated, this option is unselected, allowing the auto-publication of inventory with undetermined licenses.
	Mark associated file as reviewed	Select this option if you want Code Insight to automatically mark the files associated with each automatically published inventory item as "reviewed". This option is available only if **Auto-publish system-created inventory items meeting this minimum Confidence level** is selected.

Table 3-6 ▪ Project Defaults (cont.)

Category	Field	
Automated Review Options	These options configure defaults for enabling policies that automatically accept or reject inventory when it is published. Users can change these settings when creating a project or when editing a project using **Manage Project | Edit Project | Review and Remediation Settings** from the project **Summary** tab.	
	Policy Profile	Select the default policy profile to associate with all new projects. (The system default is **Default License Policy Profile**.)
		The policy profile contains a set of policies that use components, versions, licenses, and vulnerability scores and severities as criteria to automatically reject or approve inventory items during a codebase scan (or post-scan).
		For more information about policy profiles in general, see "Managing Policy Profiles" in the online help or the *Code Insight User Guide*.
	automatically reject inventory items impacted by a new vulnerability that violates your policy	Indicate the default action to take for published inventory affected by a new security vulnerability downloaded as part of an Electronic Update. The selected action applies to both non-reviewed and previously approved inventory items on the **Project Inventory** tab.
		● Select this checkbox to automatically reject those project inventory items impacted by a new security vulnerability only if this vulnerability has a CVSS score or severity *greater than* the thresholds configured as policy for the Code Insight project. For each inventory item rejected due to a new security vulnerability, the ⚑ icon and a tip are added to indicate the status change and its reason.
		If a new vulnerability does not exceed policy thresholds, the current status of the inventory item is not affected.
		● Leave the checkbox unselected to retain the current status of inventory items impacted by the new vulnerability.
		For information about setting policies that define CVSS-score and severity thresholds used to reject or approve inventory items automatically, see "Policies Page" and "Policy Details Page" in the online help or the *Code Insight User Guide*. For information about associating these policies with a project, see "Managing Policy Profiles" in either of these same resources.
Manual Review Options	These options configure defaults for project inventory not automatically reviewed by policy. Users can change these settings at the project level by navigating to **Manage Project | Edit Project | Review and Remediation Settings** from the project **Summary** tab.	

Table 3-6 ▪ Project Defaults (cont.)

Category	Field	
Manual Review Options (continued)	**What should happen if inventory items are not reviewed by policy?**	Indicate the default action to trigger for those inventory items that are *not* affected by policy (and therefore have a **Not Reviewed** status) during the publication of inventory either as part of a scan or manually by a user: ● **do nothing**—Simply show the status of the inventory item as **Not Reviewed** on the **Project Inventory** tab. ● **send an email notification to the project contact**—Automatically send an email to the Project Contact, stating the need for a manual review of the item. The value for **Select the minimum priority...** (described in the next table entry) affects this option. ● **automatically create a manual review task**—Automatically create a manual review task assigned to the default legal or security reviewer (or both reviewers), and send an email, notifying the reviewer(s). The Project Contact is automatically designated as the creator of manual review task. Information about managing such a task to track the progress of a manual review is found in "Creating and Managing Tasks for Project Inventory" in the online help or the *Code Insight User Guide*. The value for **Select the minimum priority...** (described in the next table entry) affects this option.
	Select the minimum priority to perform the action selected above	(Enabled when an option other than **do nothing** is selected for the previous field.) Select the default minimum inventory priority (**P1**, **P2**, **P3**, or **P4**) to which the value for the previous field applies. For example, if the previous field is set to **send an email notification to the project contact** and minimum priority is set to **P3**, then the email notification will be sent for only those non-reviewed inventory items with a P1, P2, or P3 priority. No email notification will be sent for P4 inventory items. 📄 *Note ▪ This option has no effect when the **do nothing** value is selected.*

Table 3-6 ▪ Project Defaults (cont.)

Category	Field	
Manual Review Options (continued)	**What type of manual reviews will be performed on this project?**	Set the default type of manual review tasks to be generated: ● **Legal Only**—Review tasks are generated for those non-reviewed inventory items that so not meet legal policy criteria. The tasks are automatically assigned to the default Legal reviewer. ● **Security Only**—Review tasks are generated for only those non-reviewed inventory items that have security vulnerabilities. The tasks are automatically assigned to the default Security reviewer. ● **both Legal and Security**—Review tasks are generated for all non-reviewed inventory items that do *not* meet legal policy criteria; these are assigned to the default Legal reviewer. Additionally, review tasks are generated for those non-reviewed inventory tasks associated with security vulnerabilities and are assigned to the default Security reviewer. With this value, a single inventory item might have both a legal review task and security review task generated. However, if the default reviewers are the same user, a single task is created, describing the requirement for both a legal and security manual review.
	Select reviewers for this project	If desired, designate a new default Legal reviewer or Security reviewer (or both) to which to assign manual review tasks. (The Project Contact is the designated as the initial system default for both reviewers.) Then, depending on the type of manual review selected for the project (see the **What type of manual reviews will be performed...** option described previously), Code Insight determines which reviewer (Legal or Security or both) is assigned the task and then notified of the task by email. The reviewer(s) can then manage the task accordingly, possibly reassigning it to another user. For details about managing and reassigning tasks, see "Creating and Managing Tasks for Project Inventory" in the online help or the *Code Insight User Guide*. To select a new default reviewer, click **Change User** next to the name of the current **Legal reviewer** or **Security reviewer** assignee, then select a user from the **Select new...contact** dialog, and click **Apply**. (To reset the reviewer to the Project Contact, click **Reset**.) When a new default reviewer is selected, that user is automatically given the role of project "reviewer" should the user not currently have this role. However, should the current reviewer reassign a specific task to another user, the "reviewer" role is not automatically assigned to that user. If "Project Contact" is specified as a default reviewer, the Project Contact's actual user name is displayed for the reviewer in the project.

Table 3-6 ▪ Project Defaults (cont.)

Category	Field	
Remediation Options	These options configure defaults for rejected project inventory. Users can change these settings at the project level by navigating to **Manage Project	Edit Project	Review and Remediation Settings** from the project **Summary** tab.	
	What should happen if inventory items are rejected?	Indicate the default action to trigger for those inventory items that are automatically rejected by policy during the publication of inventory either as part of a scan or manually by a user:

● **do nothing**—Simply show the status of the inventory item as Reject on the Project Inventory tab.

● **send an email notification to the project contact**—Automatically send an email to the Project Contact, stating the need for remediation work on the inventory item.

● **automatically create a remediation task**—Automatically create a remediation task assigned to the default development contact (see the **Assignee for remediation work** option) and send an email, notifying the this contact about the assigned task. The Project Contact is automatically designated as the task creator.

● **automatically create a remediation task and an external work item**—Automatically do the following:

  ● Create a remediation task assigned to the default development contact (see the **Assignee for remediation work** option) and send an email, notifying the contact about the assigned task. (See the previous bulleted item for more information.) The Project Contact is automatically designated as the creator of manual review task.

  ● Create a work item and associate it with the task. The work item is created in your Application Lifecycle Management (ALM) system by using the settings defined for the ALM instance with which the Code Insight project is associated. For more information about configuring an ALM instance for the project, see "ALM Settings" in the online help or the *Code Insight User Guide*.

Note ▪ *Currently Code Insight supports only Jira as an ALM system and Jira issues as work items.*

Table 3-6 ▪ Project Defaults (cont.)

Category	Field	
Remediation Options (continued)	**Assignee for remediation work**	If desired, designate a new default development contact—for example, an engineering manager—to which to assign remediation tasks. (The Project Contact is the initial system default.) This contact can then manage the task accordingly—for example, reassigning it to another user or manually creating an external work item and assigning it to someone on the development team. For details about managing and reassigning tasks, see "Creating and Managing Tasks for Project Inventory" in the online help or the *Code Insight User Guide*.
		To select a new contact, click **Change User** next to the name of the current assignee, select a user from the **Select new...contact** dialog, and click **Apply**. (To reset the reviewer to the Project Contact, click **Reset**.) If "Project Contact" is specified as the default, the Project Contact's actual user name is displayed as the remediation assignee in the project.

Setting the Common Vulnerability Scoring System (CVSS) Version

Code Insight can be configured to use either CVSS v2.0 or CVSS v3.x (3.1 and 3.0) for security vulnerability CVSS scores and severities. Initially, the system defaults to CVSS v2.0.

Switching between CVSS versions will affect CVSS scores and severity values for vulnerabilities, as displayed in the Web UI (see "Security Vulnerabilities Associated with Inventory" in the *Code Insight User Guide*). The change also has an impact on policies based on CVSS scores and vulnerability severities (see "Managing Policies" in the *Code Insight User Guide*).

Refer to the following topics for more information about setting the CVSS version:

● Differences in Vulnerability Severities Between Scoring Systems

● Setting the CVSS Version

Differences in Vulnerability Severities Between Scoring Systems

For insight into differences between the two scoring systems, the following table shows the severity levels available in the two CVSS versions and the range of scores that define each severity:

Table 3-7 ▪ Vulnerability Severity Differences Between CVSS Versions

Severity	CVSS v3.x Score Range	CVSS v2.0 Score Range
Critical	9.0 - 10.0	--
High	7.0 - 8.9	7.0 - 10.0

Table 3-7 ▪ Vulnerability Severity Differences Between CVSS Versions (cont.)

Severity	CVSS v3.x Score Range	CVSS v2.0 Score Range
Medium	4.0 - 6.9	4.0 - 6.9
Low	0.1 - 3.9	0.1 - 3.9
Unknown (v3.x) None (v2.0)	N/A	N/A

For information on how switching CVSS versions can impact policies, see "Policy Details Page" in the *Code Insight User Guide*.

Setting the CVSS Version

Use the following procedure to configure the CVSS version for Code Insight.

Task ***To set the CVSS version, follow these steps:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking click the icon ▤ in the upper right corner of the Code Insight Web UI. From the Code Insight main menu that opens, click **Administration**.)

2. Select the **System Settings** tab.

3. In the **Security Vulnerability Options** section, select either **CVSS v2.0** or **CVSS v3.x**.

4. Click **Save**.

Creating and Managing Custom Fields for Inventory

The standard fields used to describe OSS and third-party inventory or to define projects in Code Insight might not provide all the detail that your site requires to manage these entities. For example, to process and finalize a Bill of Materials for your product, you might want to know whether a thorough inbound review of a given inventory item has been completed or what encryption algorithms are used for a given inventory item.

To address this need for additional detail, Code Insight enables the System Administrator to create and manage custom fields that are available for all projects or for all inventory across the projects in your Code Insight system. Project Analysts and Reviewers can then update the values for these fields for any inventory item. They can also filter inventory on custom field values at the project and global levels.

Depending on its configuration, a custom field for inventory can be made available in both the Code Insight Web UI (that is, on the **Project Inventory** tab and in the **Analysis Workbench**) and the Code Insight REST API. Alternatively, the field can be configured to be visible in the REST API only.

Currently, a maximum of five custom inventory fields can be created.

📋

Note ▪ Once a custom field is created, it cannot be deleted. However, it can be disabled and re-enabled as needed.

The following describes how to create and manage custom fields for inventory:

- Success Messages When Creating or Updating Custom Fields

- Creating a Custom Field for Inventory

- Attributes Used to Define a Custom Field for Inventory

- Editing a Custom Field for Inventory

- Disabling or Re-enabling a Custom Field for Inventory

- Availability of a Custom Field for Inventory

Success Messages When Creating or Updating Custom Fields

When you successfully create or update a custom field, a message box is displayed in the upper corner of the Code Insight UI to inform you that the operation has successfully completed. The message persists for a couple of seconds, but you can click the ⊗ button in the box to close the message sooner.
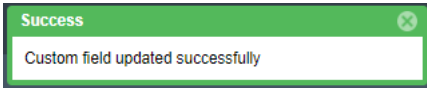
Creating a Custom Field for Inventory

Use the following procedure to create a custom field for inventory.

Once custom fields are created, project Analysts and Reviewers can update the values for these fields for any inventory item. They can also filter inventory on custom field values at the project and global levels.

📋

Task   ***To create a custom field for inventory, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ☰ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. On the **Administration** page, select the **System Settings** tab on the left.

3. In the **Custom Fields for Inventory** section, click **Add Field** to open the **Add Custom Field** dialog.

4. Enter the attributes of the custom field you are creating. For a description of each attribute, see Attributes Used to Define a Custom Field for Inventory.

5. Click **Save**. The new custom field is added to the list of custom fields in the **Custom Fields for Inventory** section on **System Settings** tab. Additionally, the field's configuration determines its availability in Code Insight. For more information about the availability of the field once you save its definition, see Disabling or Re-enabling a Custom Field for Inventory.

Attributes Used to Define a Custom Field for Inventory

The following attributes are used to define a custom field for inventory.

Table 3-8 ▪ Attributes Used to Define a Custom Field for Inventory

Attribute	Description
Enabled	The attribute controlling whether the custom field is activated in Code Insight.
	• **Yes**—The field will be activated and made available in Code Insight. (Default)
	Use the **Visible in UI** attribute (see next) to determine whether the field will be displayed in both the Code Insight REST interface and the Web UI or only in the REST interface.
	• **No**—The field will not be available in Code Insight. All other attributes defined here are ignored (until the field is enabled).
Visible in UI	The attribute that controls whether the custom field is visible in both the Code Insight REST interface and the Web UI (that is, on the **Inventory Details** tab in both the **Project Inventory** tab and the **Analysis Workbench**) *or* in the Code Insight REST interface only.
	• **Yes**—The field is visible in both the Code Insight REST interface and the Web UI, enabling project Analysts and Reviewers to use either the REST interface or the UI to view and update field's value for individual projects. (Default)
	• **No**—The field is not visible in the Code Insight Web UI. Users must use the REST interface to view and update the field's value for individual inventory items.
Field Label	(Required) The name of the custom field. The maximum length is 30 characters.

Table 3-8 ▪ Attributes Used to Define a Custom Field for Inventory (cont.)

Attribute	Description
Help Text	Information that is displayed when a user selects the ⓘ icon for the field in the Web UI. When updating this attribute, be sure to provide content that helps users enter an appropriate value for the custom field. For example, you might describe the purpose the field and the type of value it requires. If you include a string that starts with **http://** or **https://**, the string will be hyperlinked. The maximum length is 150 characters. If this attribute is left blank, the ⓘ icon will not be available for the custom field in the Web UI.

Editing a Custom Field for Inventory

Use the following procedure to edit the attributes that define a custom field for inventory. This procedure also allows you to disable or re-enable a custom field. If you intend only to disable or re-enable the field, refer to Disabling or Re-enabling a Custom Field for Inventory.

Task   ***To edit a custom inventory field, do the following:***

1. Follow the instructions in Step 1 in Creating a Custom Field for Inventory to access the **Administration** page.

2. On the **Administration** page, select the **System Settings** tab.

3. From the list of custom fields In the **Custom Fields for Inventory** section, locate the field whose attributes you want to edit, and click the ✏ icon in the **Action** column at the end of the row.

    The **Edit Custom Field** dialog opens.

4. Update the field attributes as needed. For a description of each attribute, see Attributes Used to Define a Custom Field for Inventory.

5. Click **Save** to update the custom field definition with your changes. Your changes are reflected in the list of custom fields in the **Custom Fields for Inventory** section on **System Settings** tab.

    These changes might also affect the availability of the custom field in Code Insight. For more information about the availability of the field once you save its definition, see Disabling or Re-enabling a Custom Field for Inventory.

Disabling or Re-enabling a Custom Field for Inventory

Use the following procedure to disable or re-enable a custom field for inventory. If you want to edit other attributes in the field definition, use the procedure in Editing a Custom Field for Inventory instead.

When a field is disabled, it is no longer available (that is, it is visible neither in the Code Insight Web UI nor in the REST interface) across all inventory. When a field is re-enabled, it is made available to all inventory. (The **Visible in UI** value determines whether the field is visible in both the Code Insight REST interface and the Web UI or in the REST interface only.) For more information, see Disabling or Re-enabling a Custom Field for Inventory.

Task   ***To disable or re-enable a custom field for inventory, do the following:***

1. Follow the instructions in Step 1 in Creating a Custom Field for Inventory to access the **Administration** page.

2. On the **Administration** page, select the **System Settings** tab.

3. From the list of custom fields In the **Custom Fields for Inventory** section, locate the field that you want to disable or re-enable, and click the 🖉 icon in the **Action** column at the end of the row.

   The **Edit Custom Field** dialog opens.

4. Set the **Enabled** attribute to **No** to disable the custom field or to **Yes** to re-enable the field.

5. Click **Save** to update the custom field definition with your changes. Your changes are reflected in the list of custom fields in the **Custom Fields for Inventory** section on **System Settings** tab.

Availability of a Custom Field for Inventory

The availability of a custom field for inventory in Code Insight is determined by the configuration of the **Enabled** and **Visible in UI** fields. The following sections describe the various availability states for a custom field once you save its definition:

- Enabled and Web-UI Visible

- Enabled and Not Web-UI Visible

- Not Enabled

*Note ▪ If no custom fields for inventory have been defined in Code Insight, the **Custom Fields** tab shows the message "There are no custom fields configured".*

Enabled and Web-UI Visible

If the **Enabled** and **Visible in UI** attributes for a custom field are both set to **Yes**, the field is made available in the Code Insight Web UI and the REST interface for inventory across all projects. Note the following about the custom fields in the Web UI:

- All custom fields are of the "textarea" type.

- The maximum size of a custom-field value is 64k (64000 characters).

Example Custom Fields for an Inventory Item on the Project Inventory Tab

The following example shows two custom fields (**Exclude from Notices Report** and **Encryption Algorithms**) on the **Inventory Details** tab within the **Project Inventory Details** pane for the selected inventory item on the **Project Inventory** tab:
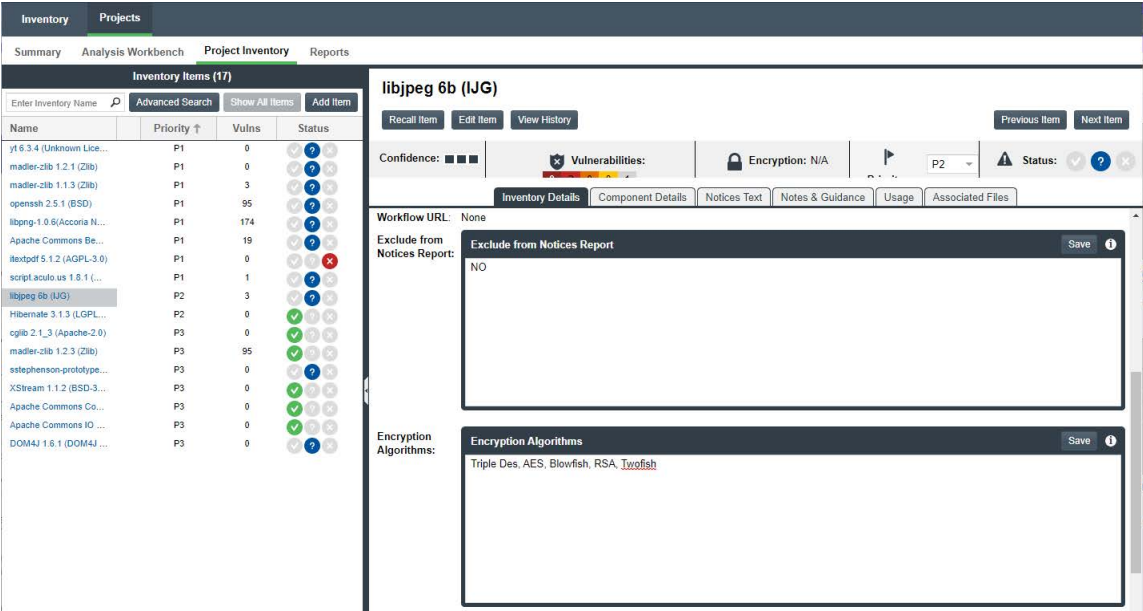
Note the following about the custom fields for inventory at this location:

- The custom fields are displayed below all the inventory metadata, so users need to scroll down to view fields.

- When a user clicks the ⓘ icon in the upper-right corner of a field, a pop-up opens, showing the content

  entered for **Help Text** attribute in the custom field's definition. If no help content has been defined, the ⓘ
  icon is not displayed.

- The **Save** button is enabled only when users enter value text in the field.

Example Custom Fields for an Inventory Item in the Analysis Workbench

The following shows these same fields on the **Inventory Details** tab for a specific inventory item in the **Analysis Workbench**.
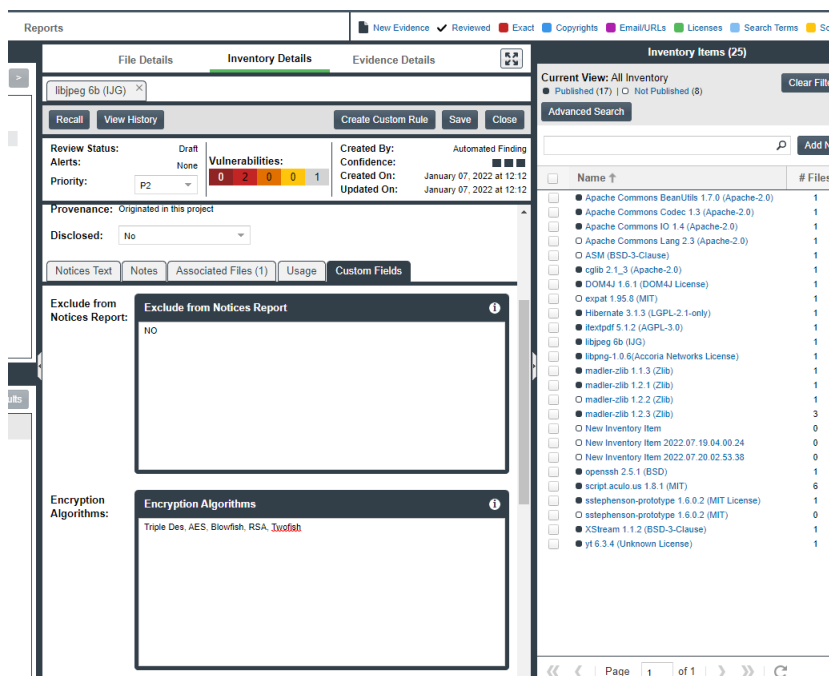
Note the following about the custom inventory fields at this location:

● The fields are displayed on the **Custom Fields** tab. The tab is listed below the inventory metadata so users need to scroll down to view the fields.

● When a user clicks the ⓘ icon in the upper-right corner of a field, a pop-up opens, showing the content entered for **Help Text** attribute in the custom field's definition. If no help content has been defined, the ⓘ icon is not displayed.

● If users enter values for custom fields, they save the values by clicking the **Save** button next to **Create Custom Rule** button.

REST Interface for Custom Fields for Inventory

Users can use the **Get Project Inventory** and **Get Inventory by ID** REST APIs to view values for custom fields for inventory items. They can also use the **Update Inventory** and **Create Inventory** REST APIs to update the custom-field values for an inventory item.

For information about these REST APIs, refer to the *Rest API Guide* Swagger documentation. To access this guide, click the ☰ in the Code Insight Web UI to open the main menu. Then select **Help** > **Documentation** > **Rest API Guide**.

Enabled and Not Web-UI Visible

If the **Enabled** attribute is set to **Yes** and the **Visible in UI** attribute is **No**, the custom field for inventory is not visible in the Code Insight Web UI. However, users can access the field's values using the REST interface, as described in the previous section, REST Interface for Custom Fields for Inventory.

Not Enabled

If the **Enabled** attribute is set to **No** despite whether the **Visible in UI** attribute is **Yes** or **No**, the custom inventory field is available neither in the Web UI nor through the REST interface. However, the field can always be made available again by re-enabling the field.

If all fields are disabled, the **Custom Fields** tab or section is blank. The REST interface shows an empty array for the "customFields" section in the **Get Project Inventory** and **Get Inventory by ID** API responses and, for **Update Inventory** and **Create Inventory** APIs, produces the message "Invalid custom field Id or field is disabled" if you attempt to update custom field values.

Creating and Managing Custom Fields for Projects

The standard fields used to describe projects in Code Insight might not provide all the detail that your site requires to manage projects. To address this need for additional detail, Code Insight enables the System Administrator to create and manage custom fields that are made available for all projects in your Code Insight system. Project administrators then update the values for these fields as needed within a given project.

Depending on its configuration, a custom project field can be made available in both the Code Insight Web UI (that is, on a project's **Summary** tab and its **Summary** > **Manage Project** > **Edit Project** > **Custom Fields** tab) and the Code Insight REST interface. Alternatively, the field can be configured for availability in the REST interface only.

Currently, a maximum of 30 custom project fields can be created.

Note ▪ Once a custom field is created, it cannot be deleted. However, it can be disabled and re-enabled as needed.

The following describes how to create and manage custom fields for projects:

- Creating a Custom Field for Projects

- Attributes Used to Define a Custom Field for Projects

- Editing a Custom Field for Projects

- Disabling or Re-enabling a Custom Field for Projects

- Availability of a Custom Field for Projects

See also Success Messages When Creating or Updating Custom Fields.

Creating a Custom Field for Projects

Use the following procedure to create a custom field for use by all projects.

Task   ***To create a custom field for projects, do the following:***

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ☰ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. On the **Administration** page, select the **System Settings** tab on the left.

3. In the **Custom Fields for Projects** section, click **Add Field** to open the **Add Custom Field** dialog.

Custom Fields for Project					
Add Field					
Enabled	**Visible in UI**	**Field Label**	**Help Text**	**Type**	**Action**
Yes	Yes	Additional Review Contacts	List any alternate contacts to perform manu…	Text Area	✎
Yes	Yes	DBA Email	Provide the email address for the DBA who…	Text Field	✎

4. Enter the attributes of the custom field you are creating. For a description of each attribute, see Attributes Used to Define a Custom Field for Projects.

   *Note ▪ When setting up the **SBOM Bucket Name** field for exporting inventory data to SBOM Insights, you must define specific attributes. See Configuring Code Insight for Exports to SBOM Insights for instructions.*

5. Click **Save**. The new custom field is added to the list of custom fields in the **Custom Fields for Project** section on **System Settings** tab. Additionally, the field's configuration determines its availability in Code Insight. For more information about the availability of the field once you save its definition, see Disabling or Re-enabling a Custom Field for Projects.

Attributes Used to Define a Custom Field for Projects

The following attributes are used to define a custom field for projects.

Table 3-9 ▪ Attributes Used to Define a Custom Field for Projects

Attribute	Description
Enabled	The attribute controlling whether the custom field is activated in Code Insight. ● **Yes**—The field will be activated and made available in Code Insight for projects. (Default) Use the **Visible in UI** attribute (see next) to determine whether the field will be visible in both the Code Insight REST interface and the Web UI or only in the REST interface. ● **No**—The field will not be available in Code Insight. All other attributes defined here are ignored (until the field is enabled).

Table 3-9 ▪ Attributes Used to Define a Custom Field for Projects (cont.)

Attribute	Description
Visible in UI	The attribute that controls whether the custom field is visible in both the Code Insight REST interface and the Web UI (that is, on a project's **Summary** tab and on its **Summary** > **Manage Project** > **Edit Project** > **Custom Fields** tab) *or* in the REST interface only. ● **Yes**—The field is visible in both the Code Insight REST interface and the Web UI, enabling project users to use either the REST interface or the UI to view and update field's value for individual projects. (Default) ● **No**—The field is not visible in the Code Insight Web UI. Users must use the REST interface to view and update the field's value for individual projects.
Field Label	(Required) The name of the custom field. The maximum length is 30 characters.
Field Type	The type of custom field: ● **Text Field**—A text field that has a maximum of 128 characters. ● **Text Area**—A large text field that has a maximum of 512 characters. (Default) ● **Drop-Down List**—A dropdown list of multiple options from which a user selects one option. When you select this field type, the **Drop-Down List Options** field is enabled to define the options (see the next description).
Drop-Down List Options	(Available when **Field Type** is **Drop-Down List**) The field that defines and manages the options in the dropdown list. ● To add an option to the dropdown list, click **Add Item** next to the field. The **Add Item** pop-up is displayed, allowing you to create an option label up to 30 characters. ● To remove an existing option, click the **x** to the right of the option label. ● To edit an option label, remove the option and re-add the option with the updated label. ● No limit exists on the number of options the field can have.

Table 3-9 ▪ Attributes Used to Define a Custom Field for Projects (cont.)

Attribute	Description
Help Text	Information that is displayed when a user selects the ⓘ icon for the custom field in the Web UI. When updating this attribute, be sure to provide content that helps users enter an appropriate value for the custom field. For example, you might describe the purpose the field and the type of value it requires. If you include a string that starts with **http://** or **https://**, the string will be hyperlinked. The maximum length of the help text is 150 characters. If this attribute is left blank, the ⓘ icon will not be available for the custom field in the Web UI.

Editing a Custom Field for Projects

Use the following procedure to edit the attributes that define a custom field for projects. (This procedure also allows you to disable or re-enable a custom field. If you intend only to disable or re-enable the field, refer to Attributes Used to Define a Custom Field for Projects.)

Task   ***To edit a custom field for projects, do the following:***

1. Follow the instructions in Step 1 in Creating a Custom Field for Projects to access the **Administration** page.

2. On the **Administration** page, select the **System Settings** tab.

3. From the list of custom fields In the **Custom Fields for Project** section, locate the field whose attributes you want to edit, and click the ✏ icon in the **Action** column at the end of the row.

    The **Edit Custom Field** dialog opens.

4. Update the field attributes as needed. For a description of each attribute, see Attributes Used to Define a Custom Field for Projects.

5. Click **Save** to update the custom field definition with your changes. Your changes are reflected in the list of custom fields in the **Custom Fields for Project** section on **System Settings** tab.

    These changes might also affect the availability of the custom field in Code Insight. For more information about the availability of the field once you save its definition, see Disabling or Re-enabling a Custom Field for Projects.

Disabling or Re-enabling a Custom Field for Projects

Use the following procedure to disable or re-enable a custom field for projects. If you want to edit other attributes in the field definition, use the procedure in Editing a Custom Field for Projects instead.

When a field is disabled, it is no longer available (that is, it is visible neither in the Code Insight Web UI nor in the REST interface) for any project. When a field is re-enabled, it is made available to all projects. (The **Visible in UI** value determines whether the field is visible in both the Code Insight REST interface and the Web UI or in the REST interface only.) For more information, see Disabling or Re-enabling a Custom Field for Inventory.

Task   ***To disable or re-enable a custom field for projects, do the following:***

1. Follow the instructions in Step 1 in Creating a Custom Field for Projects to access the **Administration** page.

2. On the **Administration** page, select the **System Settings** tab.

3. From the list of custom fields In the **Custom Fields for Project** section, locate the field that you want to disable or re-enable, and click the 🖉 icon in the **Action** column at the end of the row.

   The **Edit Custom Field** dialog opens.

4. Set the **Enabled** attribute to **No** to disable the custom field or to **Yes** to re-enable the field.

5. Click **Save** to update the custom field definition with your changes. Your changes are reflected in the list of custom fields in the **Custom Fields for Project** section on **System Settings** tab.

Availability of a Custom Field for Projects

The availability of a custom field for projects in Code Insight is determined by the configuration of the **Enabled** and **Visible in UI** fields. The following sections describe the various availability states for a custom field once you save its definition (and describe what happens when no custom fields are configured):

- Enabled and Web-UI Visible

- Enabled and Not Web-UI Visible

- Not Enabled

- No Custom Fields Configured

Enabled and Web-UI Visible

If the **Enabled** and **Visible in UI** attributes for a custom field are both set to **Yes**, the field is made available in the Code Insight Web UI and the REST interface for all projects. See the following sections for more information:

- Example Custom Fields on the Custom Fields Tab When Editing a Project

- Example Custom Fields on the Project Summary Tab

- REST Interface for Custom Fields for Projects

Example Custom Fields on the Custom Fields Tab When Editing a Project

The following example shows three custom fields (**Additional Review Contacts**, **DBA Email**, and **New Scan Required**) on the **Manage Project** > **Edit Project** > **Custom Fields** tab for a project:

When a user clicks the ⓘ icon in the upper-right corner of a field, a pop-up opens, showing the content entered for **Help Text** attribute in the custom field's definition. If no help content has been defined, the ⓘ icon is not displayed.

Example Custom Fields on the Project Summary Tab

All custom fields for projects are visible in the **Custom Fields** pane on the **Summary** tab for a given project, along with the current values that have been defined for these fields in the project. The following example shows custom fields displayed on the **Summary** tab for a project.

Note the following about the custom fields shown on the **Summary** tab:

● The field values are view-only.

● If the value for a specific field does not fit within the **Custom Fields** pane, users can click the **Show more** link to view the entire value in a pop-up.

● If a field has no value defined for the project, a hyphen is displayed.

- When a user clicks the ⓘ icon in the upper-right corner of a field, a pop-up opens, showing the content entered for **Help Text** attribute in the custom field's definition. If no help content has been defined, the ⓘ icon is not displayed.

REST Interface for Custom Fields for Projects

Users can use the **Get Project Information** REST API to view values for custom fields for a given project. Project administrators can also use the **Update Project** and **Create Project** REST APIs to update values for the custom fields in a specific project.

For information about these REST APIs, refer to the *Rest API Guide* Swagger documentation. To access this guide, click the ▤ in the Code Insight Web UI to open the main menu. Then select **Help** > **Documentation** > **Rest API Guide**.

Enabled and Not Web-UI Visible

If the **Enabled** attribute is set to **Yes** and the **Visible in UI** attribute is **No**, the custom field is not visible in the Code Insight Web UI. However, users can view and update the field's values using the REST interface (see REST Interface for Custom Fields for Inventory).

If all fields are currently configured to be not visible in the UI, the **Custom Fields** tab (on a project's **Edit Project** tab) and the **Custom Fields** pane (on the project's **Summary** tab) show no fields.

Not Enabled

If the **Enabled** attribute is set to **No** despite whether the **Visible in UI** attribute is **Yes** or **No**, the custom project field is available neither in the REST interface nor in the Web UI. However, the field can always be made available again by re-enabling the field.

If all custom fields for projects are disabled, the following occurs:

- The **Custom Fields** tab (on a project's **Edit Project** tab) and the **Custom Fields** pane (on the project's **Summary** tab) show no fields.

- The REST interface shows an empty array for the "customFields" section in the **Get Project Information** API responses and, for **Update Project** and **Create Project** APIs, produces the message "Invalid custom field Id or field is disabled" if you attempt to update custom field values.

No Custom Fields Configured

If no custom fields for projects have been configured in Code Insight, the following occurs:

- The **Custom Fields** tab on the **Edit Project** tab shows the message: "There are no custom fields configured."

- The **Custom Fields** pane on the **Summary** tab shows no fields.

- The REST interface shows an empty array for the "customFields" section in the **Get Project Information** API responses and, for **Update Project** and **Create Project** APIs, produces the message "Invalid custom field Id or field is disabled" if you attempt to update custom field values.

Configuring Code Insight for Exports to SBOM Insights

SBOM Insights (a Revenera SCA product) gives organizations the ability to manage security and legal risk by maintaining a complete, accurate SBOM (Software Bill of Materials) in the cloud. SBOM Insights aggregates this SBOM over multiple sources and provides full visibility of its contents to security and legal teams, as well as to supply chain partners.

If Code Insight has been configured to perform SBOM Insights exports, Project Analysts can export inventory data from a given Code Insight project to SBOM Insights. When the export process is finished, SBOM Insights automatically imports the exported data to a bucket, where the data is managed and aggregated with SBOMs from other sources. (For complete information about SBOM Insights, click here to access the SBOM Insights user documentation.)

To enable the export of inventory data from Code Insight to SBOM Insights, the Code Insight System Administrator must perform two tasks:

- Configure the Code Insight connection to SBOM Insights.

- Define a custom project field that lets Project Administrators identify an SBOM Insights bucket for their project's exports to SBOM Insights.

Refer to the following sections for information about this configuration:

- Overview of the Export Configuration and Process

- Configuring Code Insight to Enable Exports to SBOM Insights

Overview of the Export Configuration and Process

To provide context for the System Administrator's role in the process of exporting Code Insight inventory data to SBOM Insights, refer to the following table. It provides an overview of the configuration tasks and the process involved in the export.

Table 3-10 ▪ Configuration and Process Involved in Exporting Project Inventory to SBOM Insights

Phase	Performed By	Description	For More Information
1	Code Insight System Administrator	Configures Code Insight to enable SBOM exports.	See this current section.
2	Code Insight Project Administrator	Assigns the Code Insight project to a specific SBOM Insights bucket.	Refer to "Assigning the Project to an SBOM Insights Bucket" in the *Code Insight User Guide*.
3	Code Insight Project Analyst	Initiates the process that exports the project's inventory to SBOM Insights and imports it to the specified bucket.	Refer to "Exporting Project Inventory to SBOM Insights" in the *Code Insight User Guide*.

Table 3-10 ▪ Configuration and Process Involved in Exporting Project Inventory to SBOM Insights (cont.)

Phase	Performed By	Description	For More Information
4	SBOM Insights	Automatically imports the exported inventory to the assigned bucket as a set of "SBOM parts".	Click here to open to the section in the SBOM Insights help describing SBOM parts and their import into SBOM Insights.
5	Any Code Insight user	Accesses the Code Insight **Jobs** queue to track the progress of the export.	Refer to "Monitoring the Code Insight Jobs Queue" in the *Code Insight User Guide*.

Configuring Code Insight to Enable Exports to SBOM Insights

To enable exports to SBOM Insights, the Code Insight System Administrator must perform the following configuration procedures:

- Configuring the Connection to SBOM Insights
- Creating a Custom Field for Specifying a Bucket in Projects

Configuring the Connection to SBOM Insights

As part of enabling Code Insight exports to SBOM Insights, the System Administrator must define necessary properties to connect Code Insight to SBOM Insights.
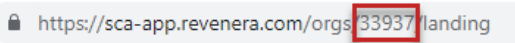
Task   *To configure the connection to SBOM Insights, do the following:*

1. On the **Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs. (You can also access this page by clicking the icon ☰ in the upper right corner of the Code Insight web page to open the Code Insight main menu. From this menu, select **ADMINISTRATION**.)

2. On the **Administration** page, select the **System Settings** tab on the left.

3. In the **Configure SBOM Insights** section, complete the following fields to configure a connection to SBOM Insights. All these fields are required to export project inventory to SBOM Insights.

Field	Description
SBOM Insights URL	Provide the URL for your SBOM Insights instance, as in this example: **https://sca-app.revenera.com** This URL is shown in the address on your browser once you log into SBOM Insights, as shown in this example: 🔒 https://sca-app.revenera.com/orgs/33937/landing
Organization Id	Provide the Organization ID in SBOM Insights to which the Code Insight data will be exported. This ID is shown in the address on your browser once you log into SBOM Insights, as shown in this example: 🔒 https://sca-app.revenera.com/orgs/33937/landing
API Refresh Token	Enter the API refresh token generated in SBOM Insights. This token is required to give Code Insight access to SBOM Insights. Instructions for generating this token are found here in the SBOM Insights user documentation.

4. Click **Test Connection** to determine whether a connection can be successfully established between Code Insight and SBOM Insights.

   • If a connection is established, a "Test Connection Successful" message is displayed in the upper right of the screen.

   • If the connection fails, an error message is displayed, explaining the error. Edit the connection information as needed and test the connection again.

5. Click **Save** to store the connection properties in the Code Insight database.

   This step also tests the connection. If the connection fails, an error message is displayed and the data is not saved. Edit the information as needed and try to save again.

Creating a Custom Field for Specifying a Bucket in Projects

Use the following procedure to create the custom field used by a Project Analyst to specify the SBOM Insights bucket for a given project's export process. Once the inventory for a given project is exported, SBOM Insights automatically imports the inventory (now called a set of "SBOM parts") to this bucket, where the "parts" are managed and viewed.

Task   ***To create custom field to specify an SBOM Insights bucket in a project, do the following:***

1. Follow steps 1 through 3 in Creating a Custom Field for Inventory to begin the process of creating a custom field.

2. Configure the following attributes for the custom field.

   Except for the **Help Text** value, the value listed for each field in the table is the *required* value for exporting project inventory to SBOM Insights. (For general information about the fields, see Attributes Used to Define a Custom Field for Projects.)

Field	To this...
Enabled	Select **Yes**.
Visible in UI	Select **Yes**.
Field Label	Enter **SBOM Bucket Name**.
Field Type	Select **Text Field**.
Help Text	(Optional) Enter a description or brief instructions for the user who must identify the SBOM Insights bucket. The following is an example: **Enter the exact name of the bucket in SBOM Insights to which SBOM Insights automatically imports the project's inventory once the export is complete.** You can leave this field blank although best practice is to provide help for the user.

3. Click **Save**. If the field is successfully saved, it is now available to all projects, enabling Project Administrators to specify the SBOM Insights bucket required for exporting project inventory to SBOM Insights.

Accessing Code Insight Server REST API Documentation

You can create an administration client (tool) that communicates with the Code Insight server using Code Insight public REST APIs to manage and retrieve project information. These APIs use a REST-style interface and JSON. For more information about this REST interface, see the *Rest API Guide* Swagger documentation available from the **Help** menu.

Use the following procedure to access the REST API Swagger documentation for Code Insight.

For information about obtaining the JSON Web Token (JWT) required to access the REST interface, see Managing Authorization Tokens.

Task   ***To view REST API Swagger documentation for Code Insight, do the following:***

1. From any page in Code Insight, click ☰ and select **Help** from the menu. The **Documentation** menu appears.

2. Click **Rest API Guide**. Swagger documentation for the Code Insight REST API is displayed on a tab in your browser.

3. To navigate the Swagger documentation, click the down arrow next to an API group to expand the group to its list of APIs.

4. Click the method button (**GET**, **POST**, **PUT**, **DELETE**) next to a specific API to expand the API details.

5. (Optional) In the details for the API, click **Try it out** to issue the API against the current Code Insight server. (You must provide the security token and required parameters and then click **Execute**.) The Swagger application generates a cURL command, makes the Rest API call, and displays the response in the details for the API.

Enabling Cross-Origin Resource Sharing

CORS (Cross-Origin Resource Sharing) is an HTTP-header-based tool that allows secure cross-origin requests and data transfers between browsers and servers. Cross-origin situations occur when the Web location from which a request originates is different from the Web location of the server to which the request is sent. Cross-origin locations can differ in scheme (HTTP or HTTPS), domain (such as `mylocation.com` versus `yourlocation.com`), or port.

For example, if you have created a script or application that makes Code Insight REST API calls, these calls will most likely result in errors if the script resides at a Web location different from Code Insight Core Server web location.

To enable a secure, successful exchange of requests and responses in a cross-origin scenario, you need to set up a CORS filter on the Code Insight Core Server. This filter identifies the origins (clients) from which the Core Server (server) will accept requests and specifies the types of headers and HTTPS methods that the server will support in the requests.

The following sections describe how to configure the CORS filter:

*   Configuring the CORS Filter

*   CORS Initialization Parameters

*   Identifying Origins for the cors.allowed.origins Initialization Parameter

*   About HTTP Headers

Configuring the CORS Filter

By default, the CORS filter is not configured on the Code Insight Core Server. To configure the filter, you must add the following code snippet to the `Built-in Filter Mappings` section in the `<codeInsightInstallation>/tomcat/conf/web.xml` file. See CORS Initialization Parameters for more information about the filter parameters.

Once you have configured the CORS filter, you must restart the Core Server.

```
<filter>
        <filter-name>CorsFilter</filter-name>
        <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
        <init-param>
            <param-name>cors.allowed.origins</param-name>
            <param-value>*</param-value>
        </init-param>
        <init-param>
            <param-name>cors.allowed.methods</param-name>
            <param-value>GET,POST,HEAD,PUT,DELETE,OPTIONS,PATCH</param-value>
        </init-param>
        <init-param>
            <param-name>cors.allowed.headers</param-name>
            <param-value>Access-Control-Allow-Origin,Access-Control-Request-
Method,Authorization,Content-Type</param-value>
        </init-param>
        <init-param>
            <param-name>cors.preflight.maxage</param-name>
            <param-value>86400</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>CorsFilter</filter-name>
        <url-pattern>/api/*</url-pattern>
    </filter-mapping>
```

CORS Initialization Parameters

The following provides more information about the initialization parameters used to define in the CORS filter set up for use by Code Insight. These parameters can be adjusted for Code Insight installed at your site.

Table 3-11 ▪ CORS Initialization Parameters

Initialization Parameter	Definition

```<filter>```
```<filter-name>CorsFilter</filter-name>```
```<filter-class>```
   ```org.apache.catalina.filters.CorsFilter```
 ```</filter-class>```
 ```</filter>```
 ```...```
 ```<filter-mapping>```
 ```<filter-name>CorsFilter</filter-name>```
 ```<url-pattern>/*</url-pattern>```
 ```</filter-mapping>```
	The basic code that enables the CORS filter. The filter adds the appropriate `Access-Control-*` headers to responses and issues the 403 return code when a request is invalid or not permitted.

`cors.allowed.origins`	The origins (clients) whose requests the server will accept. For security purposes, you should replace the asterisk * value (shown for this parameter in the code snippet provided in Configuring the CORS Filter) with the URL for each specific origin accepted by the server. For details, see Identifying Origins for the cors.allowed.origins Initialization Parameter.

Table 3-11 ▪ CORS Initialization Parameters (cont.)

Initialization Parameter	Definition
`cors.allowed.methods`	The HTTP methods that are allowed in cross-origin requests to access Code Insight data. The value in the provided code snippet (see Configuring the CORS Filter) permits all methods, but you can adjust this list according to your site's requirements. (The default methods include GET, POST, and HEAD.) 📋 ***Note ▪*** *The HEAD method is used to retrieve only headers from the server, similar to a GET but with no message body returned.* The listed methods are included as part of the `Access-Control-Allow-Methods` header in the pre-flight response so that the client knows which methods are allowed.
`cors.allowed.headers`	The HTTP request headers allowed in actual requests. Be sure to include the `Authorization` header, which is required for Code Insight REST API calls. Additionally, for POST or PUT requests, the `Content-Type` header needs to be passed along with `Authorization` header. The headers specified here are returned as part of the `Access-Control-Allowed-Headers` header in the server's response to a pre-flight request, informing the client which headers are allowed in requests.
`cors.exposed.headers`	(Not shown in the code snippet) The specific headers that can be exposed to the client as part of the response, enabling the client to then use these headers. These headers are returned as part of the `Access-Control-Expose-Headers` header in the Core Server's response to a pre-flight request.
`cors.preflight.maxage`	The maximum number of seconds that the results of the pre-flight request can be cached. (The results include the information contained in the `Access-Control-Allow-Methods` and `Access-Control-Allow-Headers` headers.) The provided code snippet (see Configuring the CORS Filter) uses the value 86400, representing 24 hours, but you can adjust this value as needed. The CORS default value is 1800.

Identifying Origins for the cors.allowed.origins Initialization Parameter

The `cors.allowed.origin` parameter, used to configure the CORS filter, identifies the clients (origins) that are allowed to issue requests to the server. The code snippet provided in Configuring the CORS Filter shows an asterisk * as the value for this parameter, indicating that a request can come from any origin. For security purposes, this value should be set to the one or more specific request origins that the server supports.

Defining an Origin

Use the following format for each origin added:

```
Origin: <scheme> "://" <hostname> [ ":" <port> ]
```

Note the following:

- Use all lower case in the URL. The value is case-sensitive.

- Be sure to include the port number, as it is required. However, you do not have to include the port number if it is known to be configured as part of host name.

- The host name can be the fully qualified domain name (FQDN).

- When specifying multiple origins, separate them with commas, as in this example:

```
Origin: http://www.machine123.org:8080, http://www.machine1000.smc.com:8080
```

Example Origin Formats

The following are examples of origin formats:

- `http://www.w3.org` (port known to be configured with hostname)

- `https://www.apache.org` (port known to be configured with hostname)

- `http://www.abc.domain.com` (port known to be configured with hostname)

- `http://<origin_Machine_hostName>:8080`

- `https://<FQDN(fully_qualified_domain_name)>:8080`

- `http://<origin_host_IP_address>:8080`

About HTTP Headers

The following table provides information about some of the HTTP headers used in requests and responses.

Table 3-12 ▪ HTTP Headers

Header	Type	Definition
`Access-Control-Request-Method`	Request	Used by browsers when issuing a pre-flight request inform the server which HTTP method will be used when the actual request is made.

Table 3-12 ▪ HTTP Headers (cont.)

Header	Type	Definition
Origin	Request	Identifies the URL from which the request originated.
Access-Control-Request-Headers	Request	Used by browsers when issuing a pre-flight request to inform the server which HTTP headers the client intends to send in the actual request. The server's response to this header is the Access-Control-Allowed-Headers header, informing the client which headers are allowed in the request.
Access-Control-Allow-Origin	Response	Indicates that the origin of the request can access the response. If the origin is not permitted to send the request, a 403 code is returned for a pre-flight request, while an actual request fails with a CORS error.

Managing Authorization Tokens

Code Insight uses a JSON Web Token (JWT) to authorize user access to the Code Insight public REST API interface (see the previous section, Accessing Code Insight Server REST API Documentation). Code Insight enables you to generate and manage one or more of these authorization tokens.

An authorization token is for use by the Code Insight user account that creates it. Thus, an authorization token that your user account generates will give you REST access to only the Code Insight functionality for which your account has permissions. Additionally, you can view and manage only those authorization tokens for the user account under which you are logged in.

Authorization tokens are created and managed from **Preferences** page, as described in the following procedures:

- Accessing the Preferences Page

- Generating an Authorization Token

- Copying the Authorization Token to the Clipboard

- Editing the Token Name

- Deleting an Authorization Token

Accessing the Preferences Page

Use these steps to open the **Preferences** page.

Task   ***To open the Preference page, use these steps:***

1. Click the **Open Menu** icon in the upper right of any Code Insight page:

2. Select **Preferences** to open the **Preferences** page.

Generating an Authorization Token

Use the following procedure to generate an authorization token.

Task   ***To generate an authorization token, do the following:***

1. Access the **Preferences** page (see Accessing the Preferences Page).

2. From the **AUTH Tokens** pane, click **Add Token**.

3. Enter a name for the new token and specify an expiration date (or choose **Never Expires**).

4. Click **Save**.

Copying the Authorization Token to the Clipboard

Use the following procedure to copy an authorization token to the clipboard so that you can paste it in your REST API interface.

Task   ***To copy an authorization token to the Clipboard, do the following:***

1. Access the **Preferences** page (see Accessing the Preferences Page).

2. From the **AUTH Tokens** pane, locate the token you want to copy, and click the **Copy to clipboard** icon in the **Actions** column.

3. Click **Select Token Text** to select the entire token value, and then press Ctrl + C to copy it.

4. Paste token in the appropriate location for use by the REST interface.

Editing the Token Name

You can edit only the name of an authorization token, not its expiration date or value.

Task   ***To edit the token name, do the following:***

1.   Access the **Preferences** page (see Accessing the Preferences Page).

2.   From the **AUTH Tokens** pane, locate the token you want to edit, and click the **Edit** ✎ icon.

3.   Update the token name as needed.

4.   (Optional) To copy the token value to the Clipboard for pasting into the REST interface, click **Select Token Text** to select the entire token value, and then press Ctrl + C to copy it.

Deleting an Authorization Token

Use the following procedure to delete an authorization token.

Task   ***To delete an authorization token, do the following:***

1.   Access the **Preferences** page (see Accessing the Preferences Page).

2.   From the **AUTH Tokens** pane, locate the token you want to edit, and click the **Delete** ✕ icon.

Configuring the Session Timeout

Use this procedure to configure the timeout session for Code Insight. The default is 240 minutes.

Task   ***To set the timeout session for Code Insight, do the following:***

1.   Open the `\tomcat\webapps\codeinsight\WEB-INF\web.xml` file in your Code Insight installation.

2.   Locate the **Session configuration** section.

    ```
    <!-- Session configuration -->
    <session-config>
    <session-timeout>240</session-timeout>
    </session-config>
    ```

3.   Adjust the session value (in minutes) and save the file.

4

Integrating with Source Code Management

The following topics are covered in this section:

- Obtaining Codebase Files for Scanning

- SCM Support

- SCM Command-Line Client

- Git Configuration

- Perforce Authentication

- Subversion Configuration

- TFS Protocol and Credentials Configuration

Obtaining Codebase Files for Scanning

To support deep scanning, it is necessary to bring the project codebase files to the Scan Server. Code Insight provides the following ways to bring codebase files into the system:

- **Upload a codebase into Code Insight**—Uploading a codebase is useful to project Analysts who typically perform ad-hoc scans on an arbitrary snapshot of code provided by the product team.

- **Use a version control SCM connector**—Code Insight SCM (Source Code Management) connectors provide an automated way to fetch the code based on criteria, such as build, release, calendar, checkin, and other information. SCM connectors support various authentication mechanisms, including anonymous, username and password, and token, key, or ticket on a Scan Server.

  The connectors for all SCM systems supported by Code Insight are provided as part of your Code Insight installation.

  See the next section, SCM Support, for information about the supported SCM systems.

SCM Support

Code Insight provides connector support for the following SCM systems, enabling remote codebases in these systems to be obtained before a scan:

● Git

● Perforce

● SVN (Subversion)

● TFS (Team Foundation Server)

The next sections describe the prerequisites that need to be in place before the Code Insight Scan Server can integrate with any of the supported SCM systems:

● SCM Command-Line Client

● Git Configuration

● Perforce Authentication

● Subversion Configuration

● TFS Protocol and Credentials Configuration

For information about configuring a synchronization instance to a specific codebase in the SCM system, see the "Configuring Source Code Management" chapter in the *Code Insight User Guide*.

SCM Command-Line Client

Before you proceed, ensure that the appropriate SCM command-line clients are installed and configured on the Code Insight Scan Server. The SCM client for a given repository type is required on the Scan Server so that Code Insight can connect to and synchronize with external repositories of that type. See the following topics for information about SCM clients:

● Installing an SCM Client

● Verifying SCM Client Installation

● Setting the Environment Variable on Windows

● Prerequisite If Running Code Insight as a Service

Installing an SCM Client

The following is a list of clients known to work effectively with Code Insight. Download the client from the specified third-party site and install it on the same instance as the Code Insight Scan Server. Use the installation instructions provided by the site.

Note ▪ *Download site links are subject to change.*

SCM	Client	Download Site
Git	Git	http://git-scm.com/downloads
		Also see More About Git Client Versions Supported by Code Insight.
Perforce	Perforce	https://www.perforce.com/downloads
Subversion	Two clients to choose from:	
	TortoiseSVN	https://tortoisesvn.net/downloads.html
	Apache Subversion	https://subversion.apache.org/download.cgi
		or
		https://subversion.apache.org/packages.html
Team Foundation Server (TFS)	Team Explore Everywhere Command Line Client (TEE-CLC)	https://github.com/Microsoft/team-explorer-everywhere/releases

More About Git Client Versions Supported by Code Insight

Code Insight supports Git client versions 2.30 (minimum) through the latest version.

Note that Red Hat Enterprise Linux and its derivatives (such as Oracle Linux, CentOS, Scientific Linux, and others) typically ship with older versions of Git. To install a more recent version of Git, you can download a tarball and build from source or use a third-party repository such as IUS Community Project. The steps for installing from source are found at this location:

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

TEE-CLC Requirement for a TFS Connection

TEE-CLC is the TFS client required by Code Insight to connect to and synchronize with an TFS collection. Once this client is installed on the same instance where the Code Insight Scan Server resides, run the following command to accept the end-user license agreement:

```
tfs -eula
```

If Code Insight attempts to connect to TFS before this command is run, the connection fails.

Verifying SCM Client Installation

To verify that the SCM client is installed and available to Code Insight, open a command prompt and navigate to the Code Insight root directory. Execute a command specific to your SCM, such as:

- `git help`

- `p4 help`

- `svn help` or `svn --version`

- `tf help`

If the system cannot find the command specified, verify that the SCM client directory is part of the PATH variable on this server. Consult your SCM documentation for more information on how to install and configure the client.

Additionally, best practice is to actually check out a sample repository on the SCM server to ensure that connectivity between the SCM client and SCM server is configured appropriately.

Setting the Environment Variable on Windows

If you run the SCM command line client from a Windows instance, add your SCM client location to the PATH environment variable.

Note ▪ Your SCM may require other environment variables to be set. Consult your SCM documentation.

Task   **To set the environment variable, do the following:**

1. To find your PATH environment variable settings, navigate to **Control Panel > System > Advanced System Settings**.

2. Click **Environment Variables**.

3. Look for the PATH system variable and make sure that it is set to the location of your SCM bin directory.

4. If you edit the system variable, ensure that you save your changes.

Prerequisite If Running Code Insight as a Service

If Code Insight is configured to run as a service, the user context under which the service runs must have the appropriate permissions to run the given SCM client.

Git Configuration

Once you have installed the Git SCM client, use the following instructions to ensure that Code Insight is properly configured to synchronize with Git repositories.

- Protocol Configuration

- Configuration to Ensure Proper Storage of User Credentials

Protocol Configuration

Git repositories reside on public servers, such as GitHub and Bitbucket, or on Git servers within a corporate network. The Git URL used to clone the repository into your SCM destination folder will vary depending on your desired protocol. The following are the available protocol options. You will enter then enter the URL using the desired protocol when you configure the SCM instance in Code Insight to connect to repository you want to clone in Code Insight.

- Anonymous HTTP

- Authenticated HTTP

- HTTPS

- SSH

Note ▪ Ensure that you are able to run the Git client on the machine where the Code Insight Scan Server resides.

Anonymous HTTP

This protocol can be used for a public repository. Public repositories can be cloned without providing an account and password.

Type	Example
GitHub Example	`http://github.com/myacct/Spoon-Knife.git`
Bitbucket Example	`http://bitbucket.org/myacct/myquotefork.git`

Authenticated HTTP

This protocol can be used for a private repository. Provide an account and password as shown in the URL format below. Use a colon between the account and password.

Type	Example
GitHub Example	`http://myacct:password@github.com/myacct/Hello-World.git`
Bitbucket Example	`http://myacct:password@bitbucket.org/myacct/bb101repo.git`

HTTPS

Code Insight supports an anonymous or authenticated HTTPS protocol between the Git client (installed on the same machine as the Code Insight Scan Server) and the Git server, such as GitHub and Bitbucket, containing the repository to be synchronized to Code Insight.

HTTPS Configuration

Ensure that the SSL certificate verification between the Git client, installed on the same machine as the Code Insight Scan Server, and the Git server is successful. This verification might include importing the Git server certificate into the local cacert authority. Because the details of this process is outside the scope of Code Insight documentation, refer to the appropriate Git server or client documentation for further details.

In a trusted environment, one option to ease the integration process is to skip the SSL certificate validation step by running the following step from the machine running the Scan Server:

```
git config --global http.sslVerify false
```

URL Format

For an anonymous HTTPS protocol, use a URL similar to one of these:

Type	Example
GitHub Example	https://github.com/myacct/Spoon-Knife.git
Bitbucket Example	https://bitbucket.org/myacct/myquotefork.git

For authenticated HTTPS protocol, provide an account and password, separating them with a colon as shown in these examples:

Type	Example
GitHub Example	https://myacct:password@github.com/myacct/Hello-World.git
Bitbucket Example	https://myacct:password@bitbucket.org/myacct/bb101repo.git

SSH

Code Insight supports SSH authentication between the Git client (running on the same machine as the Code Insight Scan Server) and the Git server, such as GitHub and Bitbucket, containing the repository to be synchronized to Code Insight.

Setting Up SSH Authentication

For instructions on how to set up an SSH authentication for communication between a Git client and the GitHub server, refer to the GitHub documentation (such as https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh).

If you want to use HTTPS along with SSH for communication between the Git client and the Git server, you must perform the additional steps described in the next section.

Configuring the Use of HTTPS Along with SSH

Use the appropriate procedure to configure use of HTTPS along with SSH for communications between the Git client and the Git server:

- Configuration in a Linux Environment

- Configuration in a Windows Environment

These procedures assume the following:

- SSH authentication between Git client and the Git server has already been set up.

- The Git server containing the repository to be synchronized to Code Insight is configured for HTTPS.

Depending on your Git version, the paths referenced in these procedures might be different from the paths used in your version.

Configuration in a Linux Environment

Use this procedure to configure the use of HTTPS along with SSH when the Git client and Git server run in a Linux environment.

Task ***To configure the use of HTTPS along with SSH in a Linux environment, do the following:***

1. As the user who will need to establish a connection between the Git client and the Git server for synchronization purposes, log on to the machine running both the Git client and the Scan Server.

2. Execute the following command:

   `git config --global http.sslCAPath /etc/pki/tls/certs`

   This command establishes a secure connection between the Git client and the Git server.

3. On the machine running the Git server, locate the file containing the Git HTTPS root certificate signed by CA.

4. Copy the certificate file to the `/etc/pki/tls/certs` folder on the machine running both the Git client and the Scan Server.

Configuration in a Windows Environment

Use this procedure to configure the use of HTTPS along SSH when the Git client and Git server run in a Windows environment.

Task ***To configure the use of HTTPS along with SSH in a Windows environment, do the following:***

1. On the machine running the Git server, export the Git HTTPS root certificate to a file. You can do this from within your browser.

2. On the machine running both the Git client and the Scan Server, locate the `ca-bundle.crt` file in the appropriate Git folder (for example, `C:\Program Files\Git\usr\ssl\certs`).

3. Open `ca-bundle.crt` in a text editor.

4. Copy the content of the certificate file that you exported, and paste it at the end of the content in the `ca-bundle.crt` file.

5. From the machine running both the Git client and the Scan Server, execute the following command to establish a secure connection between the Git server and the Git client:

```
git config --global http.sslCAInfo C:\Program Files\Git\usr\ssl\certs\ca-bundle.crt
```

Configuration to Ensure Proper Storage of User Credentials

Whether you have performed a fresh installation of the current Code Insight version or have migrated from a pre-2021 R4 version to the current version, perform the following steps to make sure that the Git SCM connector can properly store user credentials:

- Step 1: Ensure a Supported Git Client Version Is Installed

- Step 2: Ensure Git Cache Is Enabled

- Step 3: Enable the Client to Store Credentials Against the Repository Path

Step 1: Ensure a Supported Git Client Version Is Installed

The Git client 2.30 or later must be installed on the Scan Server.

Task   ***To ensure that the latest Git client is installed, perform these steps:***

1. Determine whether the Git client is installed on the Scan Server instance.

2. Perform the appropriate action:

   - If the Git client is installed, use the following command to determine its version:

     ```
     git --version
     ```

     If the version is earlier than 2.30.0, install version 2.30.0 or later of the client. If you are running a Linux machine, see More About Git Client Versions Supported by Code Insight. Otherwise, see the download site listed in Installing an SCM Client.

   - If the Git client is not installed, install a client version 2.30.0 or later. See the download site listed in Installing an SCM Client.

Step 2: Ensure Git Cache Is Enabled

The Git cache must be enabled on the Scan Server instance to ensure that Git user credentials are securely stored and that connections to the Git repository over HTTPS are successful.

Task   ***To ensure that Git cache is enabled, do the following:***

1. Perform the following:

   - On Linux, execute the following command to enable Git cache:

     ```
     git config --global credential.helper "cache --timeout=14400"
     ```

     The "credential helper" (a mechanism that fetches the user credentials and manages their caching) accepts the `--timeout` option, which updates the number of seconds that the helper's daemon is kept running, thus determining how long the credentials are stored in cache. In this case, the cache will time out after `14400` seconds (4 hours). The default is 900 seconds (15 minutes).

   - In Windows, no action is required. The Git Credential Manager, where the credentials are securely stored, is auto-enabled.

2. To verify that the cache is enabled, execute the following command:

   ```
   git config --list
   ```

   - On Linux, if the cache is properly set, the output of the command should contain the following element:

     ```
     credential.helper=cache --timeout=<value>
     ```

   - In Windows, if the output includes the `credential.helper=<value>` property, the credential helper is set.

Step 3: Enable the Client to Store Credentials Against the Repository Path

The following configuration instructs the Git client to provide the path portion of the remote URL to credential helpers. When the path is supplied, the Git Credential Manager uses the host name plus the path as the key when reading or writing credentials.

Task   ***To enable the client to store credentials against the repository path, do the following:***

1. Execute the following command:

   ```
   git config --global credential.httpusepath true
   ```

2. Restart Tomcat.

Perforce Authentication

Perforce repositories reside on an enterprise Perforce server. The Code Insight Perforce connector supports the following types of authentication to access a repository on the Perforce server and synchronize it with Code Insight:

- **Perforce authentication**—An authenticated TCP or SSL protocol is supported for communication between the Code Insight connector and the Perforce server.

Note that the Code Insight Perforce connector supports access to repositories that reside only on a Perforce server configured with Security Level 1, 2, or 3. The connector does not support access to repositories on a Perforce server configured with Security Level 0, in which users are created without passwords.

- **LDAP authentication**—The Code Insight Perforce connector supports LDAP authentication on the Perforce server. If Perforce is configured with LDAP, the Perforce SCM instance set up in Code Insight must include the LDAP credentials to access the repository.

Subversion Configuration

The following describes configuration you might need for Code Insight synchronization with Subversion:

- Anonymous HTTP
- Subversion Authentication

Anonymous HTTP and HTTPS

Either of these protocols can be used for a public repository. Public repositories can be cloned without providing a user name and password. The following is an example of a repository URL using an anonymous HTTP protocol:

    http://svn.eionet.europa.eu/repositories/Python/

The following is an example of a repository using an anonymous HTTPS protocol:

    https://svn.apache.org/repos/asf/abdera/

Subversion Authentication

Subversion repositories reside on an enterprise VisualSVN server or a comparable server on Linux. The Code Insight Subversion connector supports the following types of authentication to access a repository on the given server and synchronize it with Code Insight:

- **Subversion authentication**—An authenticated TCP or SSL protocol is supported for communication between the Code Insight connector and the server where the repository resides.

- **LDAP authentication**—The Code Insight Subversion connector supports LDAP authentication on the server where the repository resides. If the server is configured with LDAP, the Subversion SCM instance set up in Code Insight must include the LDAP credentials to access the repository.

TFS Protocol and Credentials Configuration

The following describes configuration you might need for Code Insight synchronization with TFS:

- HTTPS Protocol Support
- Special Requirement for VSTS Projects in TFS

HTTPS Protocol Support

HTTPS is supported for communication between Code Insight and TFS. Perform the following steps to enable the SSL configuration for HTTPS.

Task   ***To enable SSL configuration, do the following:***

1. Export the Secure Site SSL certificate from the browser location (shown here) for the given TFS instance:

   https://<TFS-Host>/tfs/DefaultCollection/<Project>

2. Import the certificate in the Java (JRE) keystore, using the following command (replacing `tfs.cer` with the actual certificate file name). The certificate should be imported to the same machine where the TEE-CLC and Code Insight Scan Server reside (see TEE-CLC Requirement for a TFS Connection).

   `keytool -import -trustcacerts -keystore cacerts -storepass changeit -noprompt -alias tfs -file`
   `tfs.cer`

Requirements for Synchronization with TFS

Note the following requirements for synchronization with TFS:

- Minimum Team Explorer Everywhere (TEE) Version
- Special Requirement for VSTS Projects in TFS

Minimum Team Explorer Everywhere (TEE) Version

The command-line client installed on the Code Insight server must be TEE-CLC-14 or greater.

Special Requirement for VSTS Projects in TFS

If Code Insight is synchronizing with a VSTS (Visual Studio Team Services) project in TFS, alternate VSTS authentication credentials are required for the synchronization.

Task   ***To enable alternate authentication credentials needed for Code Insight synchronization with a VSTS project in TFS, do the following:***

1. In Visual Studio, enable a set of alternate authentication credentials. (See the Visual Studio documentation for instructions.)

2. Specify these alternate credentials for the **Username** and **Password** in the TFS SCM instance configuration in Code Insight. See "Adding a TFS SCM Instance to the Code Insight Project" in the "Configuring Source Code Management" chapter in the *Code Insight User Guide*.

5

Integrating with Application Lifecycle Management

This chapter covers the following topics:

- About Integration with Application Lifecycle Management (ALM) Systems

- About the Jira Connector

- Prerequisites for Configuring the Jira Connector

- Configuring the Jira Connector

- Migration of Jira ALM instances from a pre-2023 R2 Release

About Integration with Application Lifecycle Management (ALM) Systems

Code Insight integrates with application lifecycle management (ALM) systems, enabling Code Insight users to create and manage external work items associated with inventory directly from Code Insight. With this integration, inventory requiring further review and remediation can be tracked externally as part of the user's existing issue-tracking system.

For example, a Code Insight scan might uncover security vulnerabilities or copyleft licenses requiring further review by the Security and Legal teams. With an ALM integration, these reviews and any resulting remedial work can be quickly converted into work items in the ALM system.

Currently, Code Insight supports integration only with the ALM system Jira. Therefore, this chapter focuses the configuring this integration to enable users to create and manage issues on their Jira Server from Code Insight.

About the Jira Connector

The following sections provide background on the Code Insight Jira connector and how it is used to integrate with the Jira system.

- The Jira Connector and Instances

- Jira Issues Created from Code Insight

- The Jira Integration Process

The Jira Connector and Instances

Integration with the Jira system is enabled through a corresponding Code Insight connector that supports pre-populated data (in the form of one or more Jira ALM *instances*) used to connect to the Jira system and to set up Jira issues. To configure the Jira connector, the Code Insight System Administrator defines one or more of these instances, as described in this chapter.

At the connector level, the System Administrator also controls the frequency of synchronization between the Jira system and Code Insight. The synchronization process keeps Code Insight up to date with state of Jira issues.

Jira Issues Created from Code Insight

Once a Jira ALM instance is created in Code Insight, it is available for association with a Code Insight project whose inventory might need remedial measures. When a project is associated with a specific Jira ALM instance, project users create Jira issues on the Jira Server from Code Insight and track the remediation progress.

The Jira Integration Process

The following tasks are involved in the integration of Code Insight with the Jira system:

Table 5-1 ▪ Jira Integration Process

Task	Performed by	Task Description	Documentation Resource
1	Code Insight System Administrator	Configures the Jira connector by setting up one or more Jira ALM instances, each providing the details to connect to specific Jira Server and to a Jira project on that server.	The current chapter
2	Code Insight Project Administrator	Associates a Code Insight project with a single Jira ALM instance that will be used set up Jira issues on the Jira Server from within Code Insight. (While a Code Insight project can be associated with only one instance, a single instance can be associated with multiple Code Insight projects.)	"Associating the Project with an Application Life Cycle System to Create Work Items" in the *Code Insight User Guide*
3	Code Insight project user	Creates a work item for project inventory in Code Insight. The Jira ALM instance associated with the project connects to the Jira Server and uses the work-item data to create a Jira issue on the Jira Server.	"Creating and Viewing External Work Items for a Project Inventory Task" in the *Code Insight User Guide*
4	Code Insight project user	Tracks the remediation progress through links in Code Insight to the external Jira issues. Code Insight also provides users with the current state of Jira issues through synchronization with the Jira Server.	"Viewing a Work Item" in the *Code Insight User Guide*.

Prerequisites for Configuring the Jira Connector

The following requirements must be met to ensure a successful creation of an issue on the Jira Server.

- **Latest Jira connector**—Ensure that your Code Insight installation contains the latest Jira connector, particularly after migration to the latest Code Insight version. (The Jira connector is located on the core server in the `config/core/plugins` directory.)

- **Proper user credentials to access Jira**—To access the Jira Server, each instance configured for the Jira connector requires the credentials of a valid user on the server. Additionally, the specified user must have full access to the Jira ALM instance on the Jira Server, particularly if Captcha or Single Sign-On is enabled on the Jira Server.

- **Compatible field configuration for Jira issues and Code Insight work items**—The field configuration for Jira issues on the Jira Server must include **Issue Type**, **Description**, **Summary**, **Assignee**, and **Priority** to correspond with the work-item fields available on the Jira ALM instance in Code Insight. Any other field used to define issues on the Jira Server must be configured as optional (that is, it does not require a value). Deviation from this field configuration on the Jira Server can cause the creation of Jira issues from Code Insight to fail. For information about the configuration of fields used to define Jira issues, see https://support.atlassian.com/jira-cloud-administration/docs/specify-field-behavior/.

- **Same priority scheme for Jira issues and Code Insight work items**—The priority scheme for issues on the Jira Server must match the priority scheme used in the Jira ALM instance on Code Insight. If a different priority scheme is used on the Jira Server, the creation of an issue from Code Insight can fail. Code Insight uses the default Jira priority scheme 1 through 5, where 1 indicates "Highest" and 5 indicates "Lowest" (as described in https://confluence.atlassian.com/adminjiraserver/defining-priority-field-values-938847101.html.)

You can use the **Test Connection** button for a specific Jira ALM instance on the **ALM** tab to validate a successful connection to the Jira Server (see Adding a Jira ALM Instance). However, this test does not guarantee a successful creation of a Jira issue from Code Insight. All the requirements listed in this section must be in place to help bring about a successful integration with Jira.

Configuring the Jira Connector

To enable integration a Jira system, the System Administrator must configure the Jira connector with one or more instances on the Jira system. The following topics describe how to configure and maintain these instances:

- Adding a Jira ALM Instance
- Jira ALM instance Fields
- Synchronizing with the Jira System
- Updating a Jira ALM instance
- Deleting a Jira ALM instance

Adding a Jira ALM Instance

To enable the Jira connector that is available with Code Insight, the Code Insight System Administrator must add one or more Jira ALM instances to the connector. Once the instances are added, they are available to all Code Insight projects on their **Manage Project** > **Edit Project** window, enabling the Code Insight Project Administrator to configure a project with one of the instances.

Task ***To add a Jira ALM instance, do the following:***

1. As System Administrator, use either of these steps to open the **Administration** page:

    ● From the Code Insight dashboard that is displayed when you open Code Insight, click **administration**.

    ● From the Code Insight web page, click ▤ in the upper right corner, and select **ADMINISTRATION** from the menu.

2. Select the **ALM** tile on the left to open the **ALM** tab.

3. Select **Jira** from the **Application** dropdown list.

4. Click **Add Instance**. A new **Instance #*x* - Jira** tab is displayed (where *x* is an automatically assigned sequential tab number).

5. Enter values for the fields that define the new Jira ALM instance. For important details about these fields, refer to Jira ALM instance Fields. Note the following:

    ● The only fields required to set up the instance are those fields that identify the instance and define the connection to the Jira Server:

        ● **ALM Instance Name**

        ● **JIRA Server URL**

        ● **Authentication Type**

        ● **JIRA Username** (required only for **On Premise Jira: Basic Auth** and **Cloud Jira: Basic HTTP** authentication types)

        ● **JIRA Password/API Token**

      All remaining fields are optional when setting up an instance.

    ● After you complete the fields required for the connection, you can click the **Test Connection** button to see whether a successful connection is established with the Jira Server. However, clicking **Save** after you completed the required and any optional fields will also test the connection.

6. Click **Save** to save the instance and test the connection to the Jira ALM instance.

    If the connection with the Jira Server is successful, a "Connection successful" message is displayed.

    If an error occurs, check that the required fields contain valid information.

Jira ALM instance Fields

The **ALM** tab allows you to configure Jira ALM instances that enable integration between Code Insight and your site's Jira system for the purpose of creating Jira issues in that system from Code Insight. The following table describes the fields on this tab.

Table 5-2 ▪ ALM Tab

Group	Field	Description
Connector-level fields		The following fields configure an ALM connector (in this case, the Jira connector).
	Application	To add instances to the Jira connector, select **Jira** from the **Application** dropdown. Your selection is displayed in the non-editable **Application** field in the body of the instance definition.
		The field is required.
	Add Instance	Click this button to open a new **Instance #*n* - Jira** tab (within the **ALM** tab) to create a new Jira ALM instance.
	Existing Issues Sync Frequency	Click the ✏ to the right of this field to select the synchronization frequency that will apply to *all* the instances configured on the **ALM** tab. The synchronization process keeps Code Insight up to date with the status of the Jira issues created from Code Insight through the Jira ALM instances. Synchronization options include:
		● **Never**—Never run a synchronization. (Default)
		● **Hourly**—Run a synchronization every *x* number of hours. For example, enter **2** to run a synchronization every 2 hours.
		● **Daily**—Run a synchronization daily at *x* clock time. For example, select **1:15 AM** from the associated list to run a synchronization at 1:15 am every day. (You can also enter a custom time.)
		● **Weekly**—Run a synchronization on *x* weekday at *x* clock time every week. For example, select **Monday** and **11:00 PM** from the associated lists to run a synchronization every Monday at 11 pm. (You can also enter a custom clock time.)
		Click ✔ to accept the updated synchronization frequency or ✖ to restore the previous frequency.

Table 5-2 ▪ ALM Tab (cont.)

Group	Field	Description
Functions available for the currently open instance tab		The following buttons apply to the currently open Jira ALM instance tab.
	Test Connection	Click this button to validate that the supplied **ALM Instance Name**, **JIRA Server URL**, **Authentication Type**, **JIRA Username**, and **JIRA Password/API Token** for the current Jira ALM instance enables Code Insight to successfully connect to the Jira Server. If the connection fails, check that these fields contain valid information. See a description of these fields later in this table.
	Delete Instance	Click this button to delete the current Jira ALM instance. If one or more projects are currently associated this instance, an error message is displayed, stating that you cannot delete the project due to project "references". You must disassociate all projects from the instance before you can delete it.
Fields required to connect to the Jira system		The fields described in this section provide the information necessary for this Jira ALM instance to connect to the Jira Server. These are the only fields required to set up the Jira ALM instance. (All remaining fields on this tab are optional when creating the instance.) Once you have supplied values for these fields, you can use the **Test Connection** to determine whether your configuration successfully connects to the Jira Server. If not, ensure that you have entered a valid value each field. The connection is also tested when you attempt to save the instance. *Important* ▪ *For a description of the prerequisites needed to help ensure a successful connection, see Prerequisites for Configuring the Jira Connector.*
	ALM Instance Name	(Required) Enter a name for the Jira ALM instance. The name must be unique among all other Jira ALM instances defined for your Code Insight installation.
	JIRA Server URL	(Required) Enter the URL of the Jira Server to which Code Insight will connect. Provide the URL in the format `http(s):<serverName_or_ipAddress>`.

Table 5-2 ▪ ALM Tab (cont.)

Group	Field	Description
	Authentication Type	(Required) Select the type of Jira deployment at your site: • **Jira (On Cloud): Basic HTTP**—Jira is deployed on the Cloud Jira Server. You must provide a Jira user name and API token in the **Jira Username** and **Jira Password/API Token** fields, respectively. • **On Premise: Basic Auth**—Your site uses an on-premise Jira Server and Data Center that requires a Jira user name and password in the **Jira Username** and **Jira Password/API Token** fields, respectively, as credentials. • **On Premise Jira: Bearer Token**—Your site uses an on-premise Jira Server and Data Center that requires a personal access token (PAT) in the **Jira Password/API Token** field as credentials. (No **Jira Username** value is required.)
		Use the following fields to provide the credentials needed for the selected **Authentication Type**. For a successful connection, ensure that the specified user is a valid user on the Jira Server. Additionally, make sure that the user has full access to the URL instance specified for the Jira Server. This is particularly important if Captcha or Single Sign-On is enabled on the server. 📄 *Note ▪ After a successful connection, the user is automatically designated as the reporter of the each Jira issue created from this instance.*
	JIRA Username	(Required for **Jira (On Cloud): Basic HTTP** and **On Premise: Basic Auth**) Enter the user name for the Jira user.
	JIRA Password/API Token	(Required) Enter the appropriate value based on the **Authentication Type** selection: • **For "Jira (On Cloud): Basic HTTP"**—Enter the API token associated with the user name. • **For "On Premise: Basic Auth"**—Enter the password associated with the user name. • **For "On Premise: Bearer Token"**—Enter the user's personal access token (PAT).

Table 5-2 ▪ ALM Tab (cont.)

Group	Field	Description
Jira project for the instance		The following field identifies the Jira project (on the Jira Server) with which any Jira issue created from this instance will be associated.
	Default Project Key	Provide a default value for the key that identifies the Jira project with which the Jira issues (created from this instance) will be associated. This value can be edited when a project is associated with the instance. This field is optional when configuring the instance (but is required when a Code Insight project is associated with this instance, as described in "Associating the Project with an Application Life Cycle System to Create Work Items" in the *Code Insight User Guide*).
Fields used to define a Jira issue		The fields described in this section are used to define the Jira issue. Note the following: ● These fields are optional when setting up an instance. ● A value entered for any of these fields serves as the field's default. However, it can be overwritten by a Project Manager when associating a project with this instance or by a project user when creating a Jira issue. (For more information, see "Associating the Project with an Application Life Cycle System to Create Work Items" and "Creating and Viewing External Work Items for a Project Inventory Task" in the *Code Insight User Guide*.) ● If you enter a default value, ensure that it is valid. Validation of these field values takes place during the creation of a Jira issue. At that time, if information entered for these fields is invalid (for example, the **Assignee** value does not exist in the Jira system), the information will still be saved, but the user will not be able to create the issue on the Jira Server. *Important ▪ A Jira issue on the Jira Server must include the **Issue Type**, **Priority**, **Assignee**, **Summary Text**, and **Description Text** fields because these are the fields used to define the Jira issue in Code Insight. Any other field used to define a Jira issue on the Jira Server must be configured as optional (that is, as not requiring a value). For complete information, see the* Prerequisites for Configuring the Jira Connector.
	Default Issue Type	Enter the type of issue created on the Jira Server— **Bug** or **Task**.

Table 5-2 ▪ ALM Tab (cont.)

Group	Field	Description
	Default Priority	Select the priority level for the Jira issue: **1**—Highest **2**—High **3**—Medium **4**—Low **5**—Lowest ⚠ ***Important ▪*** *The priority scheme for issues on the Jira Server must match this priority scheme. If a different priority scheme is used on the Jira Server, the creation of the Jira issue from Code Insight can fail. For complete information, see the "Prerequisites for Configuring the Jira Connector" section in the "Code Insight Installation & Configuration Guide".*
	Default Assignee	Enter the email for the user on the Jira Server to whom you want to assign any Jira issues created from this instance.
	Default Summary Text	Enter the text that will display as the summary for the issue on the Jira Server. This field supports the use of Code Insight variables, as described in Using Code Insight Variables.
	Default Description Text	Enter the text that will display as the description for the issue on the Jira Server. This field supports the use of Code Insight variables, as described in Using Code Insight Variables.

Using Code Insight Variables

The **Default Summary Text** and **Default Description Text** fields support Code Insight variables that automatically pass information about the current Code Insight project and inventory item to the content in these fields.

Supported Variables

The following table lists the available variables for use in the text entered in the **Default Summary Text** and **Default Description Text** fields:

Table 5-3 ▪ Supported Code Insight Variables For Use in Work-Item Summary and Description Text

$PROJECT_NAME	Name of the Code Insight project containing the issue
$INVENTORY_ITEM_NAME	Name of the inventory item containing the issue
$COMPONENT_NAME	Name of the component associated with the inventory item
$VERSION_NAME	Version of the component associated with the inventory item
$LICENSE_NAME	Name of the selected license for the inventory item
$NUMBER_VULNERABILITIES	Total number of security vulnerabilities associated with the inventory item
$NUMBER_FILES	Total number of files associated with the inventory item
$INVENTORY_URL	Link to the inventory item

When the work item is created, the included variables are replaced by their respective values.

Example Use of Variables

The following is example text you might enter in the **Default Summary Text** field. The text includes some of the available variables:

```
The $INVENTORY_ITEM_NAME inventory item in the project $PROJECT_NAME contains
$NUMBER_VULNERABILITIES vulnerabilities that require review. Go to $INVENTORY_URL to see the
vulnerable inventory item.
```

If your Code Insight project name is *MySampleProject* and the name of the inventory item name for which you create a work item is *Apache Commons BeanUtils*, the work item and Jira issue will display the following summary:

```
The Apache Commons BeanUtils 1.7.0 (Apache 2.0) inventory item in the project MySampleProject
contains 18 vulnerabilities that require review. Go to https://my.sample.server:8888/codeinsight to
see the vulnerable inventory item
```

Synchronizing with the Jira System

Code Insight provides an option to run a synchronization process between the Jira system and Code Insight to keep the state of work items across all instances in Code Insight up to date with state of their corresponding issues in Jira. This one-way synchronization updates the following fields for a work item in Code Insight: **Status**, **Type**, **Priority**, **Assignee**, **Summary Text**.

The following procedure describes how to set the frequency of this synchronization process.

*Note ▪ The **Existing Issues Sync Frequency** configuration applies to all the instances on the **ALM** tab. If not explicitly set, the frequency defaults to **Never**.*

Task   ***To configure the synchronization frequency, do the following:***

1. As System Administrator, use either of these steps to open the **Administration** page:

   - From the Code Insight dashboard that is displayed when you open Code Insight, click **administration**.

   - From the Code Insight web page, click ▤ in the upper right corner, and select **ADMINISTRATION** from the menu.

2. Select the **ALM** tile on the left to open the **ALM** tab.

3. Click the **Edit Sync Frequency** icon ✎ in the upper right of the **ALM** tab (to the right of the **Existing Issues Sync Frequency** field).

4. Select one of the frequency options—**Never**, **Hourly**, **Daily**, or **Weekly**—and complete their respective sub-options. For a detailed description of these options, see Jira ALM instance Fields.

5. Click the **Save Changes** icon ✔ to save the update across all the instances or the **Cancel** ✖ icon to restore the default value **Never**.

Jira Issue Statuses

If the status of a Jira issue on the Jira Server changes, the change is reflected in **Status** column on for the issue in the **Work Items for...** window once a synchronization with the Jira Server is run. The change can also result in an update to the **# Open Work Items** and **# Closed Work Items** for tasks assigned to a given inventory item.

The following lists the default status values. (Custom statuses are not currently supported.)

- The default Open status values include **Open**, **Reopen**, **New**, **To Do**, **In Progress**, and **Backlog**.

- The default Closed status values include **Done**, **Resolved**, **Verified**, and **Closed**.

For more information, refer to "Viewing Jira Issues Assigned to an Inventory Task" in the *Code Insight User Guide*.

Updating a Jira ALM instance

Use the following procedure to update the fields on an existing Jira ALM instance.

These updates impact the ALM settings for only those projects that associate with this Jira ALM instance once it has been updated. The updates do not affect the ALM settings for those projects associated with the previous version of the instance.

Task   ***To update a Jira ALM instance, do the following:***

1. As System Administrator, use either of these steps to open the **Administration** page:

    - From the Code Insight dashboard that is displayed when you open Code Insight, click **administration**.

    - From the Code Insight web page, click ☰ in the upper right corner, and select **ADMINISTRATION** from the menu.

2. Select the **ALM** tile on the left to open the **ALM** tab.

3. Open the Jira ALM instance tab you want to edit.

4. Update the values for fields that as needed. For important details about these fields, refer to Jira ALM instance Fields.

    If you have updated any of the fields used to connect used to connect to the Jira system, you can click the **Test Connection** button or wait until you save the changes to see whether a successful connection is established with the Jira Server.

5. Click **Save** to save the instance and test the connection to the Jira ALM instance.

    If the connection with the Jira Server is successful, a "Connection successful" message is displayed.

Deleting a Jira ALM instance

The Code Insight System Administrator can delete an ALM instance as long as no projects currently reference the instance.

If the instance that you want to delete is referenced by a project, it cannot be deleted until the instance is disassociated from the project. See the *Code Insight User Guide* for instructions on how unassociate an instance from a project.

Task   ***To delete an ALM instance, do the following:***

1. As System Administrator, use either of these steps to open the **Administration** page:

    - From the Code Insight dashboard that is displayed when you open Code Insight, click **administration**.

    - From the Code Insight web page, click ☰ in the upper right corner, and select **ADMINISTRATION** from the menu.

2. Select the **ALM** tile on the left to open the **ALM** tab.

3. Select the **Instance** tab for the instance you want to delete.

4. Click the **Delete Instance** button.

Migration of Jira ALM instances from a pre-2023 R2 Release

Starting in Code Insight 2023 R2, **Authentication Type** was included as a property defining a Jira ALM instance. When a site migrates from a pre-2023 R2 Code Insight release to the current release, the **Authentication Type** field for the migrated Jira ALM instances is configured as follows:

- If the **JIRA Server URL** value (or the DNS—domain name system—for a Jira Server URL using a DNS) includes *Atlassian*, the value for the **Authentication Type** field is **Cloud Jira: Basic HTTP (Username & API Token)**. (*Atlassian* is the common DNS for cloud-based JIRA servers.)

- If the **JIRA Server URL** value (or the DNS for a Jira Server URL using a DNS) does *not* include *Atlassian*, the value for the **Authentication Type** field is **On Premise Jira: Basic Auth (Username & Password)**.

Jira ALM instances defined for your Code Insight instance are found on the **ALM** tab on the **Administration** page.

Upgrading Code Insight

The information in this chapter applies to Code Insight version 7 (Code Insight v7) upgrades only. Use the procedures described to upgrade from **Code Insight 2017 R2 or later** to **Code Insight 2023 R2**. Do not use these procedures to upgrade from **Code Insight v6** to **Code Insight v7**.

This chapter contains the following information about the upgrade process:

- Upgrade Considerations
- Upgrade Steps
- Other Upgrade Tasks

Upgrade Considerations

Definitions Used in the Upgrade Description

The following definitions are used in the upgrade procedure (in Upgrade Steps):

- **vCurrent** refers to the currently installed version of Code Insight (for example, Code Insight 2022 R1).
- **vNew** refers to the Code Insight version to which you are upgrading (for example, Code Insight 2023 R2).
- `catalina.*` refers to the `catalina.bat` file on Windows systems or `catalina.sh` file on Linux systems

Instance Upgrade

The Code Insight 2023 R2 instance (**vNew**) is installed in parallel to your current instance (**vCurrent**) such that no files are overwritten.

Database Upgrade

Important ▪ Ensure that you perform a full backup of the database schema prior to upgrade.

Note the following about the database upgrade:

● The Code Insight 2023 R2 instance will use your **vCurrent** database schema.

● Database schema changes have occurred beginning with Code Insight 2020 R2 up to and including the current release, Code Insight 2023 R2.

● As of Code Insight 2019 R2, all database migration is performed automatically when you start Tomcat. Manual execution of database migration scripts is required only if you are migrating from Code Insight 2018 R2 (or earlier) to the current release.

● Code Insight uses the database user assigned to Code Insight in `core.db.properties` to migrate the database during an upgrade. This user already has the minimum permissions required to manage (and initially install) Code Insight. While these permissions are sufficient for migrating a MySQL database, the migration of a SQL Server database requires that the user have the db_ddladmin role. If the user in `core.db.properties` is not currently assigned this role, add the role before the performing the upgrade. If necessary, you can revoke this role once the upgrade completes.

● With the introduction of the **View History** feature for inventory in 2021 R3, an upgrade to the current version of Code Insight might take longer than previous upgrades, especially if the number of inventory items in your Code Insight system has increased since the last upgrade. For example, the upgrade for a system with about 1 million inventory items can now take around 15-20 minutes, which might be longer than your previous upgrades.

The extra time required for the upgrade is due to the processing of inventory items in all projects to ensure that the inventory is included in the history. However, once an inventory item is added to the history, it does not need to go through this initialization process in subsequent upgrades.

> 📄
>
> *Note ▪ Even though the processing time for the database upgrade might be significant, Tomcat should never be restarted during this process. If you have any concerns about the processing time, contact Revenera Support for assistance.*

Upgrade Limitations

For any limitations or known issues related to the upgrade process, refer to the latest version of the *Code Insight Release Notes*.

Upgrade Steps

Use the following instructions to upgrade from a previous Code Insight release to Code Insight 2023 R2. Each step includes the Code Insight server on which step is performed. The servers are identified as such:

● **Core**—Code Insight Core Server

● **Scan**—Code Insight Scan Server

● **Database**—Database server used by Code Insight

If a step applies to both the Core Server and Scan Server and both servers are installed on the same instance, you need to perform the step once on the instance.

Table 6-1 ▪ Steps to Upgrade from a Previous Code Insight Release

Step	Server	Summary	Description	Required / Optional
1	Core/ Scan	Download & install Code Insight 2023 R2	Download the `.zip` archive of the Code Insight 2023 R2 release from the Product and License Center in the Customer Community portal. Unzip the archive into the **vNew** directory, parallel to the **vCurrent** directory. For example, if your **vCurrent** directory is `D:/codeInsight/2022R1`, your **vNew** directory will be `D:/codeInsight/2023R1`.	Required
2	Core/ Scan	Shut down Tomcat	Shut down Tomcat to stop your **vCurrent** instance by executing `shutdown.bat` or `shutdown.sh` in **vCurrent**`/tomcat/bin/`.	Required
3	Database	Back up database	Back up your **vCurrent** database schema.	Required
4	Database	Migrate database	If migrating from 2018 R3 or later, skip this step since the database schema will be migrated automatically when you first start up the application after the upgrade. If migrating from 2018 R2 or earlier, you must manually migrate the **vCurrent** database schema. To do so, apply the database migration script(s) located in **vNew**`/dbScripts/install/<database_type>/` in consecutive order from earliest to latest. For example, if upgrading from Code Insight 2017 R1 to Code Insight 2023 R2, you should run all the migration scripts, starting with `migrateTo2017R1SP1.sql` and ending with `migrateTo2018R3.sql`. After 2018 R3, the database schema will be migrated automatically when you first start the application after the upgrade. For the privileges required to migrate the database, see Database Upgrade.	Required (for Code Insight 2018 R2 or earlier versions only)
5			Copy specific files from the **vCurrent** home directory to the **vNew** directory as directed in Steps 5a-5k.	
5a	Core/ Scan	License key	Copy **vCurrent**`/codeinsight.key` to **vNew**`/`. 📄 **Note** ▪ *If you have a new license key provided by Revenera, copy the new license key file to **vNew/** instead.*	Required

Table 6-1 ▪ Steps to Upgrade from a Previous Code Insight Release (cont.)

Step	Server	Summary	Description	Required / Optional
5b	Core/ Scan	Database connector	Copy the appropriate database connector to **vNew**/`tomcat/` `lib/`: ● **MySQL 5.7**—**vCurrent**/`tomcat/lib/mysql-connector-java-5.*-` `bin.jar` ● **MySQL 8.0**—**vCurrent**/`tomcat/lib/mysql-connector-java-` `8.0.*.jar` ● **SQL Server**—**vCurrent**/`tomcat/lib/sql*.jar`	Required
5c	Core/ Scan	Server and database properties	Copy the following property files to **vNew**/`config/core/`: **vCurrent**/`config/core/internal.properties` **vCurrent**/`config/code/core.db.properties` 📄 *Note ▪ If you are using a new license key file, change the encrypted password in the `core.db.properties` file to its "plaintext" form and save the file.*	Required
5d	Core/ Scan	Project data indexes	Copy the **vCurrent**/`proj_indexes/` folder to **vNew**/. If you are migrating from a version prior to Code Insight 2018 R1, skip this step as the index format has changed. Instead, rescan the projects to re-create the indexes.	Required
5e	Core	Data library indexes	Copy the **vCurrent**/`pdl_indexes/` folder to **vNew**/.	Required
5f	Core/ Scan	JRE	To use the Oracle JRE embedded with Code Insight, copy the **vCurrent**/`jre/` folder to **vNew**/. For more information about the embedded JRE, see Java Runtime Edition Requirement. To use your system Oracle JRE, skip this step. 📄 *Note ▪ If your system Oracle JRE version is earlier than 8u192, consider upgrading to a version 8u192 or later supported by Code Insight. For more information about supported Oracle JRE versions, see Java Runtime Edition Requirement.* 📄 *Note ▪ Whether Code Insight is using the embedded JRE or your system JRE, note the JRE location on your system. You will need to supply this information in Step 6 when configuring the JAVA_HOME and JRE_HOME variables in the `catalina.*` file.*	Required

Table 6-1 ▪ Steps to Upgrade from a Previous Code Insight Release (cont.)

Step	Server	Summary	Description	Required / Optional
5g	Core	Reports	If you want to retain the Code Insight reports you have generated, copy the `reports` folder from **vCurrent** to **vNew**.	Optional
5h	Core	Custom reports	If you have created custom reports and want to retain them, copy the `custom_report_scripts` folder from **vCurrent** to **vNew**.	Optional
5i	Core/ Scan	Custom properties	If you manually configured advanced auto discovery settings in the `codeaware.properties` file and want to retain the settings, copy **vCurrent**`/config/codeaware.properties` to **vNew**`/config/.` 📝 *Note* ▪ *If you do not have the file in your **vCurrent** instance or do not need to retain the settings, skip this step.* 📝 *Note* ▪ *The majority of the settings previously configurable using the* `codeaware.properties` *file are now available in the Web UI.*	Optional
5j	Core/ Scan	jet3st properties	Copy **vCurrent**`/config/core/jets3t.properties` to **vNew**`/config/.` if you have made any changes to this file as part of configuring a proxy server.	Optional
5k	Scan	Rule files	Obtain the following rule files (used in scans) from Revenera and place them in the **vNew**`/config/.codeaware/updates` directory. Contact Revenera Support for help in obtaining the files. ● `fnca-data-update.tar.gz` ● `manifest.txt` 📝 *Note* ▪ *Perform this step only if Code Insight is running in an offline ("air-gapped") environment.*	Required (for offline environments only)

Table 6-1 ▪ Steps to Upgrade from a Previous Code Insight Release (cont.)

Step	Server	Summary	Description	Required / Optional
6	Core/ Scan	Configure Tomcat	Edit the **vNew**/tomcat/bin/catalina.* file as follows: • Set JAVA_HOME to the absolute path of your JRE. This is either the system JRE path or the embedded JRE path if you followed step 5f (**vCurrent**/jre/). • Set JRE_HOME to the absolute path of your JRE. This is either the system JRE path or the embedded JRE path if you followed step 5f (**vCurrent**/jre/). • Set CATALINA_OPTS to the same value as set in **vCurrent**/tomcat/ bin/catalina.* (for example, CATALINA_OPTS= **-Xms12288m -Xmx12288m**). • Set -DcodeinsightInstallPath=<**vNew**> (for example, -DcodeinsightInstallPath=**"D:/codeInsight/2021R4"**). *Note ▪ For proxy and SSL configuration, additional edits to the* catalina.* *file are necessary. See the next steps.*	Required
7	Core/ Scan	Configure your proxy	If your current Code Insight instance is running over a proxy, copy the following values from the **vCurrent**/tomcat/bin/catalina.* to **vNew**/tomcat/bin/catalina.*: -Dhttps.proxyHost=<PROXY_HOST> -Dhttps.proxyPort=<PROXY_PORT> -Dhttps.proxyUser=<USERID> -Dhttps.proxyPassword=<PASSWORD> -Djdk.http.auth.tunneling.disabledSchemes= To configure a proxy for the first time, see Enabling Secure HTTP Over SSL in the "Installing Code Insight" chapter.	Optional

Table 6-1 ▪ Steps to Upgrade from a Previous Code Insight Release (cont.)

Step	Server	Summary	Description	Required / Optional
8	Core/ Scan	Configure SSL	If your current Code Insight instance is running over SSL, perform these steps: ● Copy the following file from **vCurrent** to **vNew**: **vCurrent**/tomcat/conf/server.xml ● If the codeinsight.jks file is present in **vCurrent**, copy it to **vNew**. If its location is *other than* under **vCurrent**/tomcat (and subsequently **vNew**/tomcat), ensure that server.xml reflects the correct keystore path. ● In the **vNew**/tomcat/bin/catalina.* file, set the following property: -Dcodeinsight.ssl=true To configure SSL for the first time, see Enabling Secure HTTP Over SSL in the "Installing Code Insight" chapter.	Optional
9	Core	Configure SSO	If your current Code Insight instance is running over SSO, copy the following files and folders from **vCurrent** to **vNew**: **vCurrent**/config/core/core.sso.common.properties **vCurrent**/core/security/ **vCurrent**/config/core/env.properties **vCurrent**/config/core/security/SPMetadata.xml **vCurrent**/config/core/security/IDPMetadata.xml To configure SSO for the first time, see Configuring Code Insight to Use Single Sign-On in the "Configuring Code Insight" chapter.	Optional

Table 6-1 ▪ Steps to Upgrade from a Previous Code Insight Release (cont.)

Step	Server	Summary	Description	Required / Optional
10	Core/ Scan	Configure service	If your current Code Insight instance is running as a service, use the following procedures to migrate the service according to your platform requirements: **On Windows:** 1. Copy **vCurrent**\tomcat\bin\service.bat to **vNew**\tomcat\bin\. 2. In **vNew**\tomcat\bin\service.bat, update these properties: set JRE_HOME=**vNew**\jre -DcodeinsightInstallPath=**vNew** setx CODEINSIGHT_ROOT **"vNew"** 3. In the **vCurrent**\tomcat\bin folder, run the following at a command prompt: service.bat remove 4. In **vNew**\tomcat\bin folder, run the following at a command prompt: service.bat install **On Linux (RedHat and CentOS):** Update the following properties in the CodeInsight.service file under the /etc/systemd/system directory: WorkingDirectory=**vNew** ExecStart=/usr/bin/su --login <loginUserId> -c **vNew**/ tomcat/bin/startup.sh (where <loginUserId> is the user ID running the Code Insight service) *Note ▪ If the startup.sh file does not have EXECUTE permission, ensure that the Code Insight user that you specify to run the service has EXECUTE permission on this file.*	Optional

Table 6-1 ▪ Steps to Upgrade from a Previous Code Insight Release (cont.)

Step	Server	Summary	Description	Required / Optional
10 (cont'd)			**On Ubuntu:** Update the following properties in the `CodeInsight.service` file under the `/etc/systemd/system` directory: WorkingDirectory=**vNew** ExecStart=***vNew***tomcat/bin/startup.sh *Note ▪ If the `startup.sh` file does not have EXECUTE permission, ensure that the Code Insight user that you specify to run the service has EXECUTE permission on this file.*	
11	Core/ Scan	Remove specific webapps files from Tomcat	Delete the following files. **For an instance running the Core Server only:** **vNew**/tomcat/webapps/codeinsightScanner (folder) **vNew**/tomcat/webapps/codeaware.war **For an instance running the Scan Server only:** **vNew**/tomcat/webapps/codeinsight (folder) **For an instance running both the Core Server and a Scan Server:** Do not delete any files.	Required
12	Core/ Scan	Start Tomcat	To start Tomcat in the **vNew** instance, execute **startup.bat** or **startup.sh** in **vNew**/tomcat/bin/. *Note ▪ The database schema is automatically migrated when you start Tomcat.*	Required
13	Core	Launch Code Insight	Open a web browser and navigate to http://<SERVER_HOST_NAME>:<PORT>/codeinsight/ (for example, **http://localhost:8888/codeinsight/**.	Required
14	Core	Run Electronic Update	Go to **Administration > Electronic Updates** to run the update to obtain the latest compliance library and automated discovery rules data.	Required
16	Scan	Rescan	To benefit from the latest automated detection rules, updates to the data library, and other Code Insight enhancements, rescan the existing projects. Alternatively, you can create and scan the same codebase in a new project.	Optional

Other Upgrade Tasks

Once you have completed the Code Insight upgrade steps, you might need to perform these tasks to complete the upgrade for your Code Insight system.

Table 6-2 ▪ Other Possible Upgrade Tasks

Server	Summary	Description	Required / Optional
Scan Server	Configure the Git SCM connector	When migrating from a pre-2021 R4 version of Code Insight, ensure the Git SCM connector provided with the new Code Insight is configured to properly store user credentials. Follow the instructions in Configuration to Ensure Proper Storage of User Credentials.	Required (for Git SCM repository syncs only)
Core	Migrate of inventory-only projects to the new single-project type	In 2020 R3, all Scan Server and remote scans began using the same project type, enabling the results of both types of scans to co-exist in a single project when needed. No new inventory-only projects are supported. ● Existing standard projects from 2020 R2 or earlier are automatically migrated to support the results from both server and remote scans. ● Existing inventory-only projects will be migrated as they are; they will continue to be supported for use with 2020 R2 or earlier projects and plugins (but will be deprecated in the future). For information about project migration in general and how to manually migrate an inventory-only project to the new project type, see the following KB article in the Revenera Community for instructions: https://community.flexera.com/t5/FlexNet-Code-Insight-Customer/Code-Insight-2020-R3-Changes-to-Projects/ta-p/160059	Optional

Table 6-2 ▪ Other Possible Upgrade Tasks (cont.)

Server	Summary	Description	Required / Optional
Remote	Update scan-agent plugins	If you are using one or more Code Insight scan agent plugins to perform remote scans, you might need to upgrade the plugins to be compatible with Code Insight 2023 R2. (The new, single project type introduced in 2020 R3 requires upgraded plugins.) To perform the upgrade, follow these steps: 1. Obtain the latest plugins `.zip` file from the Product and License Center accessed from the Customer Community portal. 2. Install and configure the plugin on the remote server according to the instructions in the *Code Insight Plugins Guide*. 3. Delete the scan agent index from the remote server (typically located in `<user_home>/.codeaware` on the remote server) so that the index will be rebuilt using the new plugin. 4. Ensure that the environment variable `CODEINSIGHT_ROOT` is set on the remote server. 5. Verify that the 7-zip tool is installed on the remote server. If it is missing, copy it from **vNew**/`7-zip`. 6. Rescan the codebase.	Optional
Remote	Rescan remote codebases to see license evidence in Code Insight	In the 2021 R1 release, the Code Insight began reporting license evidence found in files scanned by a scan agent on a remote system. (Previously, no evidence of any type was reported for these files.) If you have migrated from a pre-2021 R1 release and want to see the license evidence for these files in the Code Insight Web UI, you must rescan the remote codebase.	Optional

7

Code Insight User Roles and Permissions

This chapter serves as a reference to the various user roles and permissions that Code Insight offers as a means to control user access to its functionality at your site:

- System Roles and Permissions

- Project Roles and Permissions

- Roles and Permissions to Manage Project Task Flow

System Roles and Permissions

The following table lists the roles and associated permissions used to manage Code Insight at the system level. The initial Code Insight System Administrator (and any subsequent System Administrators) manages user accounts and assigns system-level roles to any of these users as needed. For more information, see Managing Users in the "Configuring Code Insight" chapter.

One user can be assigned multiple roles.

Table 7-1 ▪ System Roles and Permissions

Responsibility	Permissions	Notes	Roles		
			System Admin	**Policy Manager**	**Project Creator**
Administer Code Insight	Manage user accounts and permissions, create other system administrators, create policy managers, and allow all/or specified users to create projects		✔	—	—
	Schedule or force Electronic Updates		✔	—	—
	Configure an email server workflow notifications		✔	—	—
	Configure LDAP users		✔	—	—
	Configure Application Lifecycle (ALM) instances to manage inventory review tasks		✔	—	—
	Configure Scan Servers and scan profiles		✔	—	—
	Define global project defaults		✔	—	—
	Determine the CVSS version used for security vulnerability reporting		✔	—	—
	Create and manage custom fields for inventory and projects		✔	—	—
	View Code Insight logs		✔	—	—
	Suppress security vulnerabilities		✔	—	—

Table 7-1 ▪ System Roles and Permissions (cont.)

			Roles		
			System Admin	Policy Manager	Project Creator
Manage polices for automating inventory review processes			—	✔	—
Create projects	Create public and private projects	The user who creates a project automatically becomes the Project Contact for that project. (See Project Roles and Permissions for additional Project Contacts permissions.)	—	—	✔
	Manage project folders (in **Projects** pane)		—	—	✔

Project Roles and Permissions

The following table lists the various roles and associated permissions used to manage a given project in Code Insight. The project creator automatically becomes the initial Project Contact and Project Administrator. In turn, a Project Administrator can assign Analyst, Reviewer, and Observer roles to Code Insight users, as well as create other Project Administrators. The Project Administrator can also remove users from any of these roles.

For details about these roles and the procedure for assigning them, see "Assigning Project Roles to Users" in the "Using Code Insight" chapter in the *Code Insight User Guide*.

Users can be assigned multiple project roles.

Table 7-2 ▪ Project Roles and Permissions

Responsibility	Permissions	Notes	Analyst	Reviewer	Observer*	Proj. Contact	Proj. Admin	Sys. Admin
						Roles		
Manage project	Reassign the project contact		—	—	—	✔	✔	✔
	Manage project users		—	—	—	—	✔	—
	Rename the project		—	—	—	—	✔	—
	Create/edit custom field values for a project (including **SBOM Bucket Name**)		—	—	—	—	✔	—
	Move projects in **Projects** pane		—	—	—	—	✔	—
	Manage scan settings		—	—	—	—	✔	—
	Manage review/ remediation settings		—	—	—	—	✔	—
	Manage Source Control Management (SCM) and Application Lifecycle (ALM) instances		—	—	—	—	✔	—
	Delete the project		—	—	—	—	✔	—
	Branch or copy the project		—	—	—	—	✔	—
Invoke/stop scans			✔	—	—	—	✔	—
Upload codebases			✔	—	—	—	✔	—
Import/export project data			✔	—	—	—	✔	—

Table 7-2 ▪ Project Roles and Permissions (cont.)

	Roles					
	Analyst	**Reviewer**	**Observer***	**Proj. Contact**	**Proj. Admin**	**Sys. Admin**
Assign project to an SBOM bucket					✔	
Export to SBOM Insights	✔					
View project inventory	✔	✔	✔	✔	✔	✔**

Table 7-2 ▪ Project Roles and Permissions (cont.)

		Roles					
		Analyst	**Reviewer**	**Observer***	**Proj. Contact**	**Proj. Admin**	**Sys. Admin**
Review project inventory	Recall inventory	✔	✔	—	—	—	—
	Approve/reject inventory	—	✔	—	—	—	—
	Set inventory priority	—	✔	—	—	—	—
	Edit/create inventory	Only Analysts have access to the **Add Item** and **Edit Item** buttons to create/ edit project inventory properties.	✔	—	—	—	—	—
	Create and manage work items in the project's associated ALM (application life cycle management) system		—	✔	—	—	—	—
	Update Notices text and notes	This permission refers to inventory's **Notices Text** field (on the **Notices Text** tab) and the information on the **Notes & Guidance** tab (except **Detection Notes**).	✔	✔	—	—	—	—
	Edit custom field values on the **Inventory Details** tab		✔	✔	—	—	—	—

Table 7-2 ▪ Project Roles and Permissions (cont.)

			Roles					
			Analyst	**Reviewer**	**Observer***	**Proj. Contact**	**Proj. Admin**	**Sys. Admin**
	View evidence found in files listed on the **Associated Files** tab and manage the inventory's file associations	For Analysts only, the file path for an associated file is hyperlinked, enabling them to open to the file's **File Details** tab in **Analysis Workbench** to view evidence. In **Analysis Workbench**, Analysts can also add/remove files associated with inventory.	✔	—	—	—	—	—
Use Analysis Workbench	View/analyze codebase files		✔	—	—	—	—	—
	Edit alerts		✔	—	—	—	—	—
	Create, edit, and recall inventory and manage custom detection rules		✔	—	—	—	—	—
	Edit **Notices Text** field on **Notices Text** tab		✔	—	—	—	—	—
	Edit **Audit Notes** field on the **Notes** tab		✔	—	—	—	—	—
	Edit custom field values on the **Custom Fields** tab		✔	—	—	—	—	—

Table 7-2 ▪ Project Roles and Permissions (cont.)

		Roles					
		Analyst	**Reviewer**	**Observer***	**Proj. Contact**	**Proj. Admin**	**Sys. Admin**
Generate reports	Any user (not just one with a project role) can generate reports. For a "private" project, the Observer is considered an "any user", restricted to viewing project inventory and generating reports.	✔	✔	✔	✔	✔	✔

* The Observer role is available for only projects defined as "Private". Private projects are hidden from all users except the Project Contact, the System Administrator (restricted to **Summary** tab only), and those users assigned as Project Administrators, Analysts, Reviewers, and Observers of the project. An Observer is limited to viewing project inventory and generating reports for the "Private Project".

** In general, a System Administrator has permission to access both public and private projects. However, the **Project Inventory** tab for a private project is visible to a System Administrator only if the user assigned to the System Administrator role is also assigned to a role in the project (Project Administrator, Project Contact, Observer, Analyst, or Reviewer).

Roles and Permissions to Manage Project Task Flow

The following table lists the project roles and permissions used to manage tasks to review or remediate inventory items in a project.

Table 7-3 ▪ Project Task-Flow Roles and Permissions

		Roles					
		Analyst	**Reviewer**	**Observer**	**Project Contact**	**Task Assignee**	**Project Admin**
Permissions	**Notes**						
Create/edit tasks	Any user assigned to a project role can create and edit tasks.	✔	✔	✔	✔	✔	✔
Reassign tasks		—	—	—	—	✔	✔

Table 7-3 ▪ Project Task-Flow Roles and Permissions (cont.)

		Roles					
		Analyst	Reviewer	Observer	Project Contact	Task Assignee	Project Admin
Close manual review tasks		—	✔	—	—	—	—
Close remediation tasks		—	—	—	—	✔	✔
Close miscellaneous tasks	Any user assigned to a project role can close a miscellaneous task.	✔	✔	✔	✔	✔	✔