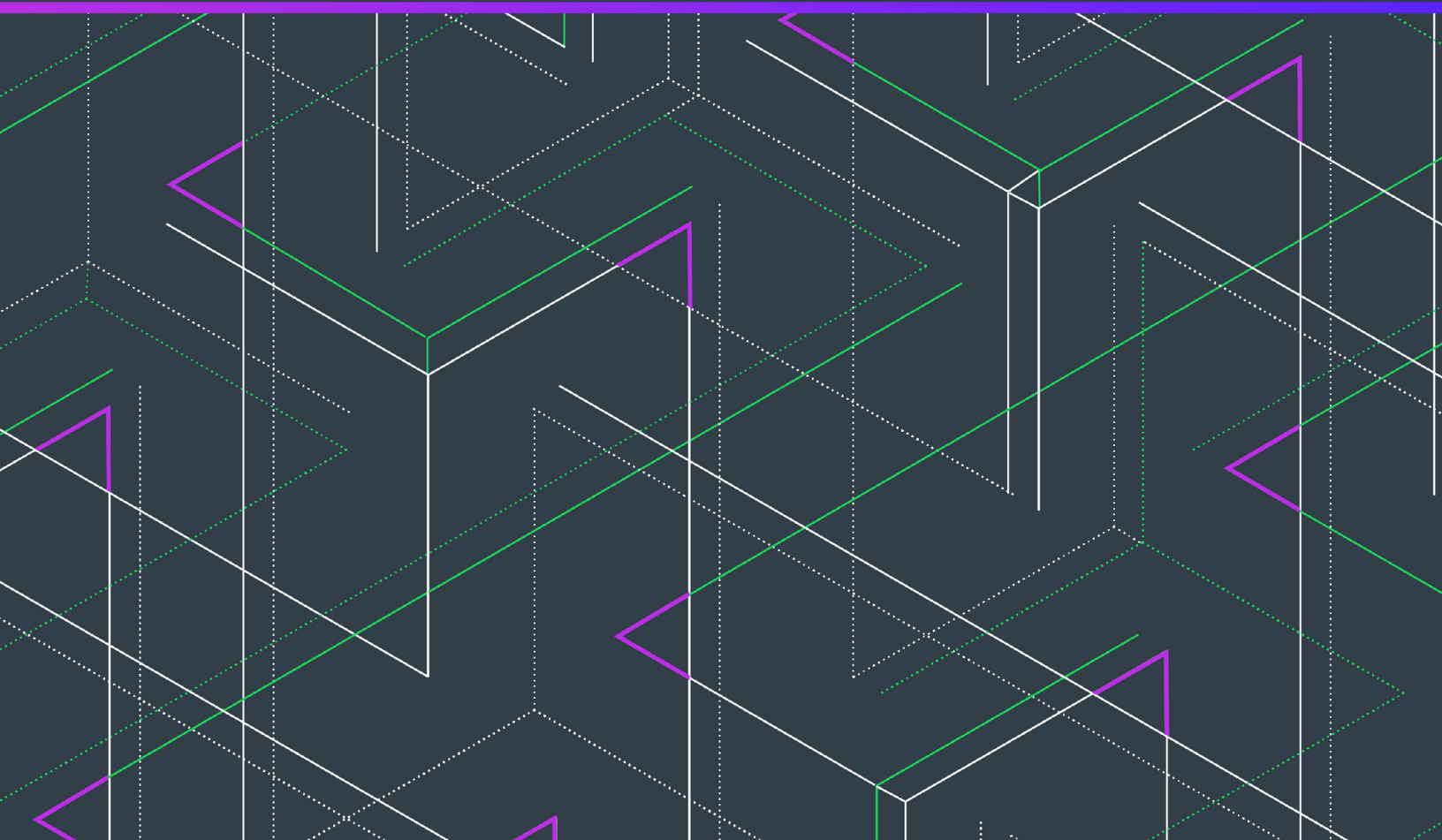


Code Insight 6.14.2 SP1

Installation & Administration Guide



Legal Information

Book Name: Code Insight 6.14.2 SP1 Installation & System Administration Guide

Part Number: RCI-6142_SP1-IG00

Product Release Date: May 2021

Copyright Notice

Copyright © 2021 Flexera Software

This publication contains proprietary and confidential information and creative works owned by Flexera Software and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software is strictly prohibited. Except where expressly provided by Flexera Software in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software, must display this notice of copyright and ownership in full.

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

Intellectual Property

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

Restricted Rights Legend

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

Contents

1	Code Insight 6.14.2 SP1 Installation & Administration Guide	9
	Code Insight Document Set	9
	Product Support Resources	10
	Contact Us	11
2	Code Insight Requirements	13
	Code Insight Server Configurations	13
	Example Server Configurations	14
	Single Instance vs. Separate Instance Deployments	15
	Dedicated Disk Drives	15
	Installation Instructions	16
	Database Server and Application Server Specifications	16
	Database Server Specifications	16
	Special Consideration for Non-US Customers	18
	Special Considerations for LDAP Integration	18
	Special Considerations for SCM and Build System Integration	19
	Running on a Virtual Server (e.g. VMware)	19
	Installable Sub-Components	19
	Prerequisites	19
	Operating Systems	20
	Databases	21
	Hardware	21
	Supported Hardware Configurations	22
	CPU Specifications	23
	Software	24
	Browsers	26
	Source Code Management	26
	Network and Firewall Considerations	26
	Server Identification	27

Code Insight Ports	27
External URLs	28
System Settings	28
Environment Variables	28
Linux/Unix Environment Variables	29
Windows Environment Variables	29
Mail Server Requirements and Configuration	29
Setting the Open File Limit (Linux/Unix)	30
Maximum Virtual Memory Limit	31
Database Settings	31
Configuring a Database	31
Character Encoding	32
Setting the UTF-8 for MySQL Installations on Linux	32
Setting the UTF-8 for MySQL Installations on Windows	33
Compliance Library Space Requirement	33
3 Installing Code Insight Manually	35
4 Extracting Application Files	37
Choosing an Installation Directory	37
Installing the Compliance Library	38
Installing Code Insight	38
Installing the Database Driver	39
Activating Code Insight	40
5 Preparing the Database for Installation	41
Preparing the Database	41
Running the Database Setup Scripts	42
6 Configuring the Tomcat Web Server	43
Setting Max Heap	43
Setting the Install Directory	44
Linux Installations: Headless Mode for JVM	44
Enabling HTTP Secure (HTTPS over SSL)	44
Increasing Post Request Size	44
Running as a Windows Service (32-bit and 64-bit)	45
7 Editing Configuration (Properties) Files	47
Entering Required Information	47
Installation Options	49
Configuring Email Services	49
Disabling SMTP Mail Server Authentication	50
Configuring Email Notification Options	50

Configuring Notification Email Templates	50
Configuring a Manual Proxy Server Connection (Optional)	52
Scan Engine Configuration Options	53
Tuning Copyright Detection	53
Controlling False Positives	54
Controlling False Negatives	54
Claiming Detected Copyrights as Yours	56
Tuning Email/URL Detection	56
Controlling False Positives	56
Tuning License Detection	57
Editing the License Detection Patterns File	57
License Text Template Example for Apache License, Version 2.0	57
License Keywords Example for Apache License, Version 2.0	57
Enabling the Availability of Analyzer for Scans and Reporting	58
Adding Custom Phases to the Scan Process	59
Sample Custom Scan-Phase Adapter Script	59
Rules and Considerations for Creating a Custom Scan-Phase Adapter	60
Class Used in Custom Scan-Phase Adapter	60
Scan Phase Order	60
Multiple Scripts Using Same Method	61
Customizing the Scan Phase Order	61
Adding Custom Inventory Statuses	61
Disabling Security Updates for Archived or Canceled Projects	62
8 Administration: Starting & Stopping Servers	63
Starting and Stopping the Server Manually	63
Validating a Successful Server Startup	64
Using a Supported Browser	64
Using an .xls File	64
Changing the Number of Log Files Maintained	64
9 Running an Electronic Update	67
Running an Electronic Update for the First Time	67
Running an Electronic Update Manually	67
Using the Electronic Update Server (HTTPS)	67
Using a Local Electronic Update Archive	68
Scheduling Automatic Electronic Updates	68
10 Managing NG-Bridge Updates for Code Insight	69
Changing the Scheduled Time for NG-bridge Data Updates	70
Enabling/Disabling NG-bridge Data Updates	71
Downloading NG-bridge Data Updates Releases Manually	71

11 Installing Code Insight Using the Installer	75
Installing Code Insight With the Installer	75
Installing Code Insight Using the Linux Command Line	79
12 Launching Code Insight	87
Starting and Stopping the Server	87
Logging into Code Insight	88
Running the Update Service	88
13 Installing the Jenkins Plugin	89
Jenkins Downloads	89
Setting up the Jenkins Plugin	90
Generating a JWT Token	91
14 Using ScriptRunner	93
ScriptRunner Options	93
Generating a JSON Web Token (JWT)	94
Using the JWT Token with scriptRunner	95
15 Exporting & Importing Workspaces	97
Usage Overview	97
What is Custom Data?	98
What Workspace Data is Exported and Imported?	98
What Inventory is Exported and Imported?	98
About Backward Compatibility	99
Installing the Scripts	99
Installing Manually	99
Using the Scripts with ScriptRunner	100
Using the Export Script	100
Export Options	100
File and Path Options	100
Server Options	101
What to Export	102
Boolean Options	102
Other Options	103
Combining Export Flags	103
Export Usage	103
Export Output	103
Export Usage Examples	104
Using the Import Script	104
Import Options	105
Import Options that Accept Parameters	105
Import Options that Require an MD5 Match	106
Import Options that Do Not Require an MD5 match	107

<i>Import Options that Accept Boolean Options</i>	107
<i>CSV File</i>	108
Import Results	108
<i>Log file (import.log)</i>	108
<i>Metadata Tags</i>	109
Import Usage Examples	109
16 Integrating with LDAP for Authentication (Optional)	111
Configuring LDAP Integration	111
Configurable LDAP Properties	114
Data Synchronization	114
User Metadata	114
Disabled Users	114
LDAP Synchronization to Multiple Sources	114
LDAP Configuration	115
Set Up a User Search Filter	115
Default Role Assignment	116
Set Up a User List Sync	116
Server Paging	117
LDAP over SSL	118
Troubleshooting and Testing LDAP Configurations	118
17 Configuring Code Insight for Single Sign-On (Optional)	119
Overview	119
Prerequisite Tasks for Configuring Code Insight for SSO	120
Configure HTTPS on the Code Insight Server	120
Set Up SSO Users	120
Obtain a Keystore	120
Phase 1: Generate Service Provider Metadata	121
Default Method for Generating Service Provider Metadata	121
Custom Method for Generating Service Provider Metadata	122
Step 1: Download and Configure the Spring Security SAML Extension	122
Step 2: Generate the SP Metadata	124
Step 3: Configure the SSO Common Properties File	125
Phase 2: Obtain the Identity Provider Metadata File	127
Phase 3: Configure the Environment Properties File	128
Phase 4: Login Using SSO Credentials	128
Reference: Troubleshooting and Debugging SSO	128
Reference: Enabling Secure HTTP over SSL	129
Enabling an HTTPS Connection	129
HTTPS Enablement Required for Each Code Insight Server	129
Enabling HTTPS for a Code Insight Server	129

Obtaining and Storing an SSL Certificate for HTTPS Enablement	131
Purchasing and Importing a Secure Site SSL Certificate	131
Creating the Keystore	131
Purchasing the SSL Certificate	132
Importing the Certificate	132
Generating a Self-Signed SSL Certificate	133
18 Configuring Additional Scan Servers (Optional)	135
Overview	135
Configuration Details	135
19 Configuring Code Insight using MySQL Commands	137
20 Configuring Code Insight as a Service	139
Configuring Code Insight as a Windows Service	139
Configure Code Insight as a Linux Service	141
21 Using SCM Connectors	143
Using the SCM Command Line Client	143
Recommended Clients	144
Verified Team Explorer Everywhere Client Versions	144
Setting the Environment Variable	145
Workspace Settings	145
IBM Rational ClearCase	146
Using Perforce (P4) to Manage the Codebase	147
Using Subversion (SVN) to Manage the Codebase	149
Git Repositories	151
Git Protocol Options	151
Anonymous HTTP	151
Authenticated HTTP	151
SSH Authentication	152
SSH over HTTPS	155
Git Workspace Configuration	155
Microsoft Team Foundation Server (TFS)	156
Project Copy Settings	158
22 Performing Backup and Recovery	159
Terminology	159
Performing the Backup	159
Performing the Restore	160
23 Frequently Asked Questions	163

Code Insight 6.14.2 SP1 Installation & Administration Guide

Code Code Insight is an application security solution targeting the widespread use of open source software (OSS). Code Insight's unique software composition analysis technology captures the composition of a code base, and provides your team with an inventory of OSS component usage. The resulting inventory identifies security vulnerabilities and intellectual property issues associated with the inventoried OSS components

The *Code Insight Installation & Configuration Guide* is intended for system administrators, database administrators, and anyone who wants a deeper understanding of Code Insight. It describes how to install, configure, and administer Code Insight.

This introductory chapter provides important information about resources available to Code Insight users and administrators, including the list of guides in the Code Insight documentation set, Product Support resources, and Revenera contact information.

Code Insight Document Set

The following are the titles of all the documentation in the Code Insight library.

Table 1-1 • Document Set

Book Title	Description
Installation and System Administration Guide	This guide can be used by a system administrator and a database administrator installing Enterprise Edition in a large scale multi-server configuration in the corporate IT environment or on a single-server (standalone) instance or machine. It can also be used by a system administrator and database administrator performing system administration.
Quick Start Guide	This guide helps users to quickly start using Code Insight, become familiar with its features and capabilities, and perform the exercises presented in the guide.

Table 1-1 ■ Document Set (cont.)

Book Title	Description
User Guide	This guide contains activity-based information for using the features and functionality.
Licensing Guide	This guide contains third-party licensing text.

Product Support Resources

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

- [Revenera Product Documentation](#)
- [Revenera Community](#)
- [Revenera Learning Center](#)
- [Revenera Support](#)

Revenera Product Documentation

You can find documentation for all Revenera products on the [Revenera Product Documentation](#) site:

<https://docs.revenera.com>

Revenera Community

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

<https://community.revenera.com>

Revenera Learning Center

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

<https://learning.revenera.com>

Revenera Support

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

<https://community.revenera.com>

Contact Us

Reverera 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.reverera.com>

You can also follow us on social media:

- [Twitter](#)
- [Facebook](#)
- [LinkedIn](#)
- [YouTube](#)
- [Instagram](#)

Code Insight Requirements

This chapter describes what you need know and do before you install Code Insight:

- [Code Insight Server Configurations](#)
- [Database Server and Application Server Specifications](#)
- [Prerequisites](#)

Code Insight Server Configurations

Before installing Code Insight, you need to determine your server configuration. Three main servers are involved in the Code Insight installation: Core Server, Scan Server, and Database Server. These servers can be installed on one instance or on separate instances in various configurations. The server configuration you choose to install should be based on the size and business needs of your organization.

Refer to the following sections for more information:

- [Example Server Configurations](#)
- [Single Instance vs. Separate Instance Deployments](#)
- [Dedicated Disk Drives](#)

Example Server Configurations

This table lists some possible Code Insight server configurations.

Table 2-1 ■ Installation Types

Installation	Details
Core/Scan Server <ul style="list-style-type: none">• Code Insight Core Server• Code Insight Scan Server• Code Insight Compliance Library• Code Insight Workspace Directory Database Server <ul style="list-style-type: none">• Database (can be installed on Core Server if necessary)	<ul style="list-style-type: none">• Recommended for product trials and small scale projects• Suitable for medium-size projects• Recommended for Standard Edition (Scan-only) installations
Core Server <ul style="list-style-type: none">• Code Insight Core Server Scan Server <ul style="list-style-type: none">• Code Insight Scan Server• Code Insight Compliance Library• Code Insight Workspace Directory Database Server <ul style="list-style-type: none">• Database (can be installed on Core Server if necessary)	<ul style="list-style-type: none">• The most common type of installation• Suitable for most medium-size projects
Core Server/Library <ul style="list-style-type: none">• Code Insight Core Server• Code Insight Compliance Library• Database Server• Database (can be installed on Core Server if necessary)	Recommended for Governance Edition (Workflow-only) installations

Table 2-1 ■ Installation Types (cont.)

Installation	Details
Core Server <ul style="list-style-type: none"> • Code Insight Core Server • Scan Server 1 • Code Insight Scan Server • Code Insight Compliance Library • Code Insight Workspace Directory Scan Server 2 <ul style="list-style-type: none"> • Code Insight Scan Server • Code Insight Compliance Library • Code Insight Workspace Directory Scan Server n (n represents the total number of scan servers) <ul style="list-style-type: none"> • Code Insight Scan Server • Code Insight Compliance Library • Code Insight Workspace Directory Database Server <ul style="list-style-type: none"> • Database (can be installed on Core Server if necessary) 	Recommended for enterprise installations and large-scale projects

Single Instance vs. Separate Instance Deployments

As suggested in the configurations described in [Code Insight Server Configurations](#), the Code Insight Core Server and the Scan Server can be installed on a single instance or on separate instances. You can also have additional Code Insight Scan Servers installed on separate instances. See [Configuring Additional Scan Servers \(Optional\)](#) for details on configuring additional Code Insight Scan Servers.

Limitation

Code Insight does not support a configuration in which the Core Server runs on a Windows instance and a Scan Server runs on Linux instance.

Dedicated Disk Drives

To avoid a scenario in which the I/O operations are competing for the same disk access, we recommend that you install the following entities on separate drives:

- Code Insight Compliance Library
- Code Insight Scan Server (including the codebase that will be scanned)
- MySQL/Oracle/SQL Server Database (preferable on a separate dedicated server)

Installation Instructions

Once you determine the Code Insight server configuration, you can For more information about installing Code Insight, see either section:

- [Installing Code Insight Manually](#)
- [Installing Code Insight Using the Installer](#)

Database Server and Application Server Specifications

This section contains specifications for sizing and backing up database and application servers:

- [Database Server Specifications](#)
- [Special Consideration for Non-US Customers](#)
- [Special Considerations for LDAP Integration](#)
- [Special Considerations for SCM and Build System Integration](#)
- [Running on a Virtual Server \(e.g. VMware\)](#)
- [Installable Sub-Components](#)

Database Server Specifications

The following table describes specifications and other important information about the database servers

Table 2-2 ▪ Server Specifications

Item	Description
Database Server	We recommend a separate server for the database to eliminate resource contention between the Code Insight application and database.

Table 2-2 ▪ Server Specifications (cont.)

Item	Description
Character Set	<p>Select UTF-8 as the character set when installing your MySQL database server.</p> <p>This is a critical step. To ensure that application data is stored properly in the database, UTF8 must be selected as the default character set when installing MySQL.</p> <p>Linux Installation:</p> <p>Ensure that you have a <code>my.cnf</code> file in your <code>/etc/mysql</code> directory</p> <p>If you do not have a <code>my.cnf</code> in your <code>/etc/mysql</code> directory, locate one of these files in the <code>/usr/share/mysql</code> directory and copy it to <code>/etc/mysql</code>. Rename the file <code>my.cnf</code>.</p> <ul style="list-style-type: none"> • <code>mysql-large.cnf</code> (MySQL prior to 5.5) • <code>my-large.cnf</code> (MySQL 5.5 to 5.6) • <code>my-default.cnf</code> (MySQL 5.6+) <p>Navigate to the <code>/etc/mysql/my.cnf</code> file, and edit the file as follows:</p> <pre>[client] #password= [your_password] port= 3306 socket= /var/lib/mysql/mysql.sock default-character-set = utf8 [mysqld] default-character-set = utf8 default-collation = utf8_general_ci character-set-client = utf8 character-set-server = utf8 skip-character-set-client-handshake</pre> <p>Windows Installation:</p> <p>Navigate to <code>my.ini</code> or <code>my-default.ini</code> file located in your MySQL Server installation directory (for example, <code>C:\Program Files\MySQL\MySQL Server 5.6</code>), and edit the following lines:</p> <pre>[mysqld] default-character-set=utf8</pre>
Storage Engine	<p>Select InnoDB as the default storage engine if you are using MySQL.</p> <p>Linux:</p> <p>Edit the <code>my.cnf</code> file in the <code>[mysqld]</code> section.</p> <pre>default-storage-engine = INNODB</pre> <p>Windows:</p> <p>Add the following line to the <code>my.ini</code> file:</p> <pre>default-storage-engine=innodb</pre> <p>Alternatively, you may use the InnoDB plug-in. For more information, see http://dev.mysql.com/doc/innodb-plugin/1.0/en/innodb-plugin-installation.html</p>

Table 2-2 ▪ Server Specifications (cont.)

Item	Description
Buffer Pools	<p>To support Code Insight electronic updates, you may need to change the memory configuration for your database instance.</p> <p>MySQL</p> <p>Specify a combination of innodb_buffer_pool_instances and innodb_buffer_pool_size so that each buffer pool instance is at least 1 GB.</p> <p>Oracle uses auto memory management. See Memory Configuration and Use.</p> <p>SQL Server uses auto memory management. See Memory Management Architecture.</p>
Sizing	<p>Have a DBA configure your database as you would for any other enterprise web application.</p> <p>Start with a base of 30 GB to accommodate the Code Insight Data libraries and other data such as users, groups, and projects disk space for the database. We recommend that you start with a base of 30 GB.</p> <p>Scale up by 2 MB for every 5,000 files scanned.</p> <p>Begin by estimating much you will scan in the first 6 months and add that to the 30 GB base size.</p> <p>As for data volume, Code Insight does not move large amounts of data and does not have extremely high concurrent transaction rates.</p> <p>Set <code>innodb_buffer_pool_size = 1G</code>.</p>
Backup	<p>This is a DBA activity per the customer's data policies.</p> <p>Perform a database dump or export of the entire central database schema regularly.</p> <p>Back up the workspaces directory at the same time you back up the database so they remain in sync.</p> <p>For additional information on backup and recovery, see Performing Backup and Recovery.</p>

Special Consideration for Non-US Customers

Due to the way we store vulnerability scores in the central database, non-US Oracle customers should use the `en_US.UTF8` locale to avoid update service errors.

Special Considerations for LDAP Integration

Code Insight can be configured to use an external LDAP or AD Directory Server to authenticate users for login. See [Integrating with LDAP for Authentication \(Optional\)](#) for more information.

Special Considerations for SCM and Build System Integration

The scm.properties file located in <CODE_INSIGHT_ROOT_DIR>\<version>\config\scanEngine contains parameters to enable the SCM connectors. SCM clients can be enabled for the following clients:

- ClearCase
- Perforce
- Subversion
- Git
- TFS (Team Foundation Server)

Consult [Revenera Support](#) for specific details on how to integrate with your existing SCM installation.

Running on a Virtual Server (e.g. VMware)

To run the Code Insight Scan Server on a virtual server using VMware (or other vendor), it is recommended that you configure the full domain name by setting the Primary DNS Suffix to the full DNS name you use to connect.

Installable Sub-Components

The following sub-components are included with Code Insight:

- Code Insight application: codeinsight_<version>.zip
 - Code Insight Core Server (Web UI and Detector Client)
 - Code Insight Scan Server
 - Code Insight ScriptRunner Bundle
- Code Insight Compliance Library (separate deliverable)
- Code Insight License Key (separate deliverable)

Prerequisites

The section lists the prerequisites for installing and using Code Insight.

- [Operating Systems](#)
- [Databases](#)
- [Hardware](#)
- [Software](#)
- [Browsers](#)
- [Source Code Management](#)

- [Network and Firewall Considerations](#)
- [System Settings](#)
- [Mail Server Requirements and Configuration](#)
- [Setting the Open File Limit \(Linux/Unix\)](#)
- [Maximum Virtual Memory Limit](#)
- [Database Settings](#)
- [Compliance Library Space Requirement](#)

For the up-to-date prerequisites, always refer to the latest *Code Insight Release Notes*.

Operating Systems

Code Insight supports the following operating systems.

Table 2-3 ■ Supported Operating Systems

Supported	Recommended
<ul style="list-style-type: none">• Ubuntu 18.04• Ubuntu 16.04• Ubuntu 14.0.4• RHEL 7.2, 7.8, or 8.2 Enterprise (64-bit)• RHEL 7.0 (64-bit)• RHEL 6.5 (64-bit)• CentOS 7 (64-bit)• CentOS 6.5 (64-bit)• Win 10 Enterprise or Professional (64-bit)• Win 8.1 Enterprise or Professional (64-bit)• Win 7 Enterprise or Professional (64-bit)• Windows Server 2016 Standard• Windows Server 2012 Enterprise or Professional (64-bit)	<ul style="list-style-type: none">• Ubuntu 18.0.4• RHEL 7.2, 7.8, or 8.2 Enterprise (64-bit)• CentOS 7 (64-bit)• Win 10 Enterprise or Professional (64-bit)• Windows Server 2016 Datacenter

Additional Notes About Operating System Support

Note the following about Code Insight support for operating systems:

- [Other Operating Systems Not Fully Verified for this Release](#)
- [Operating System Limitation](#)

Other Operating Systems Not Fully Verified for this Release

The following operating systems have been used to run Code Insight in the past but have not been fully verified as part of the current release:

- Mac OS (all versions)
- Windows Server 2008 R2 Enterprise Edition (64-bit)
- Windows XP Professional (64-bit)
- Windows 7 Ultimate (64-bit)
- CentOS 5, 6.5 (64-bit)

Operating System Limitation

Code Insight does not support a configuration in which the Core Server runs on a Windows instance and a Scan Server runs on Linux instance. See [Single Instance vs. Separate Instance Deployments](#).

Databases

Code Insight supports the following database software.

Table 2-4 ■ Supported and Recommended Databases

Supported	Recommended
<ul style="list-style-type: none">• MySQL 5.6, 5.7, 8• Oracle 11g, 12c, 18c, 19c• MS SQL Server<ul style="list-style-type: none">• 2012 r2 Enterprise• 2014 Enterprise• 2016 Enterprise	<ul style="list-style-type: none">• MySQL 5.7, 8• Oracle 12c, 19c• MS SQL Server<ul style="list-style-type: none">• 2012 r2 Enterprise• 2014 Enterprise• 2016 Enterprise

Additional Notes About Database Support

Note the following about database support:

- Oracle 11g has an end-of-life status. Thus, there is no guarantee that this Oracle version continues to work properly with Code Insight.
- The MySQL 5.0-5.5 database version has been used to run Code Insight in the past but has not been fully verified as part of the current release.

Hardware

The following hardware is supported:

- [Supported Hardware Configurations](#)
- [CPU Specifications](#)

Supported Hardware Configurations

Use the following table to determine hardware requirements for Code Insight component configurations:

Table 2-5 ▪ Supported Hardware Configurations

Type	Supported	Recommended
Scan Server	<ul style="list-style-type: none"> 32 GB RAM At least 1.25 TB hard disk space for the following (assuming that the Scan Server and Compliance Library are hosted on the same instance): <ul style="list-style-type: none"> Codebases (materials to be scanned) Workspaces (scanned results) Compliance Library (approximately 1 TB) 	<ul style="list-style-type: none"> 32 GB or 64 GB RAM depending on expected load 1.5 TB hard disk space for the following (assuming that the Scan Server and Compliance Library are hosted on the same instance): <ul style="list-style-type: none"> Codebases (materials to be scanned) Workspaces (scanned results)* Compliance Library (approximately 1 TB) <p>* Performance can benefit significantly if the workspace directory is located on a Solid State Drive (SSD) drive.</p>
Core Server	<ul style="list-style-type: none"> 16 GB RAM At least 650 MB of space for product and attachments <p>See the Database Server entry below if both Core Server and database are hosted on the same instance.</p>	<ul style="list-style-type: none"> 32 GB RAM (required if Core Server and database reside on same instance) 30 GB of space for product and attachments <p>See the Database Server entry below if hosting both Core Server and database are hosted on the same instance.</p>
Client	<ul style="list-style-type: none"> 16 GB RAM 	<ul style="list-style-type: none"> 32 GB RAM

Table 2-5 ▪ Supported Hardware Configurations (cont.)

Type	Supported	Recommended
Database Server	Database Sizing: <ul style="list-style-type: none"> The recommendation is that you have a DBA configure your database as you would for any other Enterprise Web Application. For disk space, the recommendation is to start with a base of 30 GB (for SQL Server, 50 GB) to accommodate the Code Insight Data Libraries and other data related to users, teams, projects, and such. <p>If you install the database on the same instance as the Core Server, calculate the hard-drive requirement by adding the database base size to the recommended Core Server disk space. (Also see Additional Notes about Hardware Requirements.)</p> <ul style="list-style-type: none"> After starting with the base size, scale up by 2 MB for every 5,000 files scanned. Begin by estimating how much you will scan in the first 6 months, and add that to the 30 GB base size. As for data volume, Code Insight does not move enormous amounts of data, nor does it have extremely high concurrent transaction rates. 	

Additional Notes about Hardware Requirements

Note the following about hardware requirements:

- Ensure that you allocate sufficient buffer pool size to the database. Otherwise, the Electronic Update might not complete. For MySQL, set the innodb buffer pool size to a minimum of 1 G (innodb_buffer_pool_size = 1G).
- For SQL Server, it is strongly recommended that the database and the Core Server reside on the same instance (with a minimum hard-drive requirement of 50 GB for the database and 30 GB for the Core Server, for a total of 80 GB).

CPU Specifications

Use the following table to determine CPU requirements based on the memory needed for your server configuration, as described in [Supported Hardware Configurations](#).

For example, if you intend to use the recommended 32 GB RAM for the Core Server (as listed in [Supported Hardware Configurations](#)), the CPU specifications for the instance running the Core Server include 2-CPU, each at least 2 GHZ+, with 8+ cores (as listed below).

Table 2-6 ▪ CPU Requirements

Memory	CPU (Cores)
64 GB	2-CPU (each at least 2 GHZ+) with 8+ cores on the server
32 GB	2-CPU (each at least 2 GHZ+) with 8+ cores on the server
16 GB	2-CPU (each at least 2 GHZ+) with 4+ cores on the server

Software

Code Insight supports the following software.

Table 2-7 ■ Supported Software

Software	Description	Download URL	
Java JDK	<p>Either of these required on all Core and Scan Servers. Use the latest update when possible.</p> <ul style="list-style-type: none"> Oracle JDK 8u281 (64-bit) <p>You must purchase a license from Oracle to ensure that you receive updates.</p>	Oracle JDK 8	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
		Zulu Open JDK 8	https://www.azul.com/downloads/zulu/
Java JRE	<p>Oracle JRE 8u281 (64-bit) required on client server to launch Detector.</p> <p>In general, use the latest Java update when possible. You must purchase a license from Oracle to ensure that you receive updates.</p>	Oracle JRE 8	http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
Database Client	<p>Required to access the Code Insight database server and to execute database scripts (not required if database is to be managed directly from the database server).</p> <p>Any basic client application or command line client interface may be used. Several options are listed on the right.</p>	MySQL	http://www.heidisql.com/download.php
		Oracle	http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html
		MS SQL Server	https://msdn.microsoft.com/en-us/library/mt238290.aspx



Note ■ Not required for Workflow-only installations or on client servers that already have the JDK installed.

Table 2-7 ■ Supported Software (cont.)

Software	Description	Download URL
Database Driver	JDBC driver required on the Core and Scan Servers to enable Code Insight access to the database.	MySQL mysql-connector-java-5.1.45.jar (MySQL 8) https://downloads.mysql.com/archives/c-j/ (select Product Version 5.1.45 and download)
	Download the driver corresponding to your database type and do one of the following:	mysql-connector-java-5.1.x-bin.jar (MySQL 5.6, 5.7) http://dev.mysql.com/downloads/connector/j/5.1.html
	<ul style="list-style-type: none"> If using the supplied installer (codeinsight_6.x.jar) to install Code Insight, copy the driver .jar file to the directory containing the installer. The installation process automatically copies the driver to the tomcat\lib location. If manually installing Code Insight, copy the downloaded .jar file to the following location: <pre><Code Insight_ROOT_DIR>\<version>\ tomcat\lib\</pre> 	Oracle ojdbc8.jar (Oracle 18c, 19c) https://www.oracle.com/database/technologies/appdev/jdbc-ucp-183-downloads.html ojdbc6.jar (Oracle 11g, 12c R1) or ojdbc7.jar (Oracle 12c R2) http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html
		MS SQL Server Use this site to download the driver appropriate for the type of Java JDK (JDK or OpenJDK) that you are using: https://docs.microsoft.com/en-us/sql/connect/jdbc/system-requirements-for-the-jdbc-driver?view=sql-server-2017
Other	An email account is required to send email notifications from the Code Insight server.	

Additional Notes about Software Requirements

Note the following about software requirements:

- Support for Java 7 (JDK and JRE) was removed in Code Insight 6.12.0. Ensure that you use Java 8 (JDK and JRE) with a compatible update version.
- Code Insight provides support for Zulu OpenJDK 8 only. Other OpenJDK applications might work with Code Insight but are not recommended.
- Support for Java 11 is not available.
- Java Updates released after the Code Insight 6.12.3 release date are not guaranteed to be compatible. If you encounter an issue running a newer update, notify support, which will resolve these issues on a best effort basis and issue a hotfix as needed.

- For the Oracle 19c database, the recommendation is to use the ojdbc8.jar database driver (as listed in this table), not the ojdbc10.jar driver.

Browsers

Code Insight supports the following browsers.

Table 2-8 ▪ Supported Browsers

Supported	Recommended
<ul style="list-style-type: none">• Firefox (latest stable version)• Google Chrome (latest stable version)• Internet Explorer 10, 11	<ul style="list-style-type: none">• Firefox (latest stable version)• Google Chrome (latest stable version)• Internet Explorer 11

Source Code Management

Code Insight supports the following source code management software.

Table 2-9 ▪ Supported Source Code Management Software

SCM	Sample Client Download
GIT	http://git-scm.com/downloads
Subversion (SVN)	http://tortoisesvn.tigris.org/
Team Foundation Server (TFS)	http://www.microsoft.com/en-us/download/details.aspx?id=30656
Perforce	http://www.perforce.com/product/components/perforce-visual-client
ClearCase	http://www-03.ibm.com/software/products/en/clearcase

Network and Firewall Considerations

If the Code Insight Core Server or Scan Server is behind a firewall, you need to configure the firewall to ensure that each server has access to Code Insight locations and to known repository sites for third-party component, license, and vulnerability information and other data:

- [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 or Scan Server:

- 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 or service to use custom ports.

Table 2-10 ▪ Default Port Numbers Used by Code Insight

Port #	Details
1433/1521/3306	Database Server Access Port (MS SQL Server, Oracle, MySQL, respectively)
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 used by Code Insight:

Table 2-11 ■ External Host URLs Used by Code Insight

Code Insight Component/ Functionality	Hosts
CodeAware Analyzers	https://api.bintray.com
	https://api.nuget.org/v3-flatcontainer/
	https://oss.sonatype.org
	https://packagist.org
	https://pypi.org/pypi/
	https://registry.bower.io/packages/
	https://registry.npmjs.org/
	https://rubygems.org/api/v1/gems/
	https://search.maven.org/
Vulnerability database access	https://spdx.org/licenses/
	https://nvd.nist.gov
	https://web.nvd.nist.gov/
Electronic Update	https://updates.palamida.com/
Remote file access	https://palamida-dp-nbhood.s3.amazonaws.com/

System Settings

This section lists the Code Insight environment variables, open file limit (Linux/Unix), and the maximum virtual memory limit.

Ensure that the following settings are in place before proceeding with the installation.

- [Environment Variables](#)
- [Setting the Open File Limit \(Linux/Unix\)](#)
- [Maximum Virtual Memory Limit](#)

Environment Variables

This section lists the Code Insight environment variables.

- [Linux/Unix Environment Variables](#)
- [Windows Environment Variables](#)

Linux/Unix Environment Variables

Perform the following steps to verify Linux/Unix environment variables.



Task

To verify Linux/Unix environment variables:

1. After installing the JDK, verify that the JAVA_HOME and PATH variables are set correctly on the Core Server and each Scan Server.
2. To determine your environment variables settings, enter the following on the command line:

```
echo $JAVA_HOME
echo $PATH
```

3. To find the exact path to your Java executable, enter the following:
4. To set JAVA_HOME and PATH variables, open the file .bash_profile under /home/<user> and add the following text:

```
PATH=$PATH:$HOME/bin
export JAVA_HOME=/usr/java/<insert jdk name>
export PATH=/usr/java/jdk1.7.0_##/bin:/usr/java/<insert jdk name>/lib:$PATH export PATH
```

Windows Environment Variables

Perform the following steps to verify Windows environment variables.



Task

To verify Windows environment variables:

1. After installing the JDK, verify that the JAVA_HOME variable is set correctly on the Core Server and each Scan Server.
2. To find out your environment variables settings, navigate to **Control Panel >System > Advanced System Settings**.
3. Click the **Environment Variables** button.
4. Look for the JAVA_HOME system variable and make sure that it is set appropriately to your JDK version
5. To set the JAVA_HOME variable, click on the **New** or **Edit** button depending on whether JAVA_HOME is present in the list of variables, and point the variable to your JDK install directory, such as the following:

```
C:\Program Files\Java\jdk1.7.0_##
```

Mail Server Requirements and Configuration

Code Insight requires access to an SMTP mail server for sending workflow and user-management related email messages to users. Authenticated SMTP is supported. SMTP over SSL is supported using starttls. Mail server configuration properties are set in the core.properties file located in <Code Insight_ROOT_DIR>\<version>\config\core.

Setting the Open File Limit (Linux/Unix)

This procedure specifies the open-file limit on Linux or Unix platforms.

The open-file limit is a setting that controls the maximum number of open files for individual users (such as non-root users). The default open-file limit is typically 1024. However, in order 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 size is essential for ensuring that Code Insight functions properly on Unix or Linux platforms. You can choose to perform this task before or after the application is installed; however, make sure to perform this step before launching the application.

The following procedure is verified for Fedora Core4 and can vary based on the OS/Linux Distribution. The procedure assumes that you are using the Bourne Shell (sh), Bourne Again Shell (bash), or Korn Shell (ksh). Note the following:

- If you are using C Shell (csh) or Tenex C Shell (tcsh), use the `limit openfiles NNNN` syntax instead of the `ulimit -n NNNN` syntax.
- Some limitations exist when using tcsh. For example, the tcsh shell only sets the soft-limit; you cannot set a hard-limit. This limitation is undesirable in some situations.



Task

To specify the open-file limit in Linux/Unix:

1. Check the current value with the `ulimit -a` command. The system lists all settings. The Open Files setting (-n) will probably be set to the default of 1024 as shown below.

```
[fire-4]$ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
max nice                (-e) 0
file size               (blocks, -f) unlimited
pending signals         (-i) 69120
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
open files              (-n) 1024
pipe size               (512 bytes, -p) 8
POSIX message queues    (bytes, -q) 819200
max rt priority         (-r) 0
stack size              (kbytes, -s) 10240
cpu time                (-t) unlimited
max user processes      (-u) 69120
virtual memory          (kbytes, -v) unlimited
file locks              (-x) unlimited
```

2. Open `/etc/security/limits.conf` (if running Fedora Core 4 or higher) or `/etc/sysctl.conf` (if running Red Hat 6.5 or higher) and add the following entries:

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



Note ▪ Other distributions might require a different set up.



Tip ▪ On some systems, it might be necessary to add `ulimit -n 65536` to the source file (such as `.profile`, `.bashrc`, or `.bash_profile`) to ensure the change is applied.

3. Log off, and then login again.
4. Check to make sure the new value is reflected by entering `ulimit -a`. The system will re-list all of the settings. The **open files** setting (**-n**) is set to **65536**.

```
[fire-4]$ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
max nice                (-e) 0
file size               (blocks, -f) unlimited
pending signals         (-i) 69120
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
open files              (-n) 16384
pipe size               (512 bytes, -p) 8
POSIX message queues    (bytes, -q) 819200
max rt priority         (-r) 0
stack size              (kbytes, -s) 10240
cpu time                (-t) unlimited
max user processes      (-u) 69120
virtual memory          (kbytes, -v) unlimited
file locks              (-x) unlimited
```

Maximum Virtual Memory Limit

In some cases, the maximum allowed virtual memory limit is lower than the desired size.



Task

To check the current value, do the following

1. Execute the `ulimit -v` command.
2. If not already set to “unlimited”, we recommend that you follow the procedures in [Setting the Open File Limit \(Linux/Unix\)](#). For more information, see http://www.network-theory.co.uk/docs/gccintro/gccintro_77.html.

Database Settings

This section explains how to configure a database and set the UTF-8 for MySQL.

- [Configuring a Database](#)
- [Character Encoding](#)

Configuring a Database

To configure a database for Code Insight, perform the following steps.



Task

To configure a database:

1. Use an existing database or create a new database and schema for use by Code Insight. Make sure your user has the right privileges to access the schema.
2. Ensure that the schema name conforms to the following guidelines:
 - The first character of the database name is alphabetic.
 - The database name only contains alphanumeric characters or the following special characters: `_` `$` `#`.
 - Do not use dashes (`-`) in the database schema name.

3. Ensure that the database server is configured to accept connections from the client instances that Code Insight is using (both Core Server and Scan Servers). For example, in MySQL with a database user named Code Insight, this can be accomplished with the following command.

```
GRANT ALL PRIVILEGES on *.* to 'CodeInsight'@'%' with GRANT OPTION;
```

4. Ensure you select the correct character set (UTF-8) and storage engine (InnoDB).
5. To check if you have the correct character set, use the following database command.

```
SHOW VARIABLES LIKE 'character%';
```

6. Make sure the `character_set_database` has **utf8** in the **Value** column.
7. To check if you have **InnoDB** set up, use the following database command.

```
SHOW ENGINES;
```

8. The **InnoDB** row should show **YES** or **DEFAULT** in the **Support** column.

Character Encoding

This section explains how to set UTF-8 character encoding for MySQL Installations on both Linux and Windows.

- [Setting the UTF-8 for MySQL Installations on Linux](#)
- [Setting the UTF-8 for MySQL Installations on Windows](#)

Setting the UTF-8 for MySQL Installations on Linux

This section explains how to set UTF-8 character encoding for MySQL Installations on Linux.



Task **To set the UTF-8 character encoding for MySQL installations on Linux:**

1. Ensure that you have a `my.cnf` file in your `/etc/mysql` directory

If you do not have a `my.cnf` file in your `/etc/mysql` directory, locate one of the following files in the `/usr/share/mysql` directory and copy it to `/etc/mysql`. Rename the file to `my.cnf`.
 - `mysql-large.cnf` (MySQL prior to 5.5)
 - `my-large.cnf` (MySQL 5.5 to 5.6)
 - `my-default.cnf` (MySQL 5.6+)
2. To change the default encoding to UTF-8, navigate to the `/etc/mysql/my.cnf` file, and edit the file as follows:

```
[client]
#password = [your_password]
port = 3306
socket = /var/lib/mysql/mysql.sock
default-character-set = utf8

[mysqld]
default-character-set = utf8
default-collation = utf8_general_ci
```

```
character-set-client = utf8
character-set-server = utf8
skip-character-set-client-handshake
```

3. To set the storage engine to **InnoDB**, edit the `my.cnf` file in the `[mysqld]` section.

```
default-storage-engine = INNODB
```

4. As an alternative, you may use the **InnoDB** plug-in.



Note ■ For more information, see: <http://dev.mysql.com/doc/innodb-plugin/1.0/en/innodb-plugin-installation.html>.

Setting the UTF-8 for MySQL Installations on Windows

This section explains how to set UTF-8 character encoding for MySQL installations on Windows.



Task

To set the UTF-8 character encoding for MySQL installations on Windows:

1. Navigate to `my.ini` or `my-default.ini` file located in your MySQL Server installation directory, such as `C:\Program Files\MySQL\MySQL Server 5.7`, and edit the following lines:

```
[mysqld]
default-character-set=utf8
```

2. To set the storage engine to **InnoDB**, add the following line to the `my.ini` file:

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

Compliance Library Space Requirement

The Code Insight Compliance Library is required before proceeding with the installation. Ensure that you allocate approximately 1 TB of space to the drive that will contain the library.

Obtain and extract the Code Insight Compliance Library into the location of your choice, such as:

`<CODE_INSIGHT_ROOT_DIR>/PDL/CL/2.43.`

Installing Code Insight Manually

Perform the following sequential steps to install Code Insight manually. For the procedure to install Code Insight using the Installer, see [Installing Code Insight Using the Installer](#).



Note ▪ The examples in this section assume that the Code Insight Core and Scan Server are located on the same server.



Task

To install Code Insight manually, do the following:

1. Prepare your environment and install the prerequisite software on the Core/Scan Server. See [Preparing the Database for Installation](#) for detailed steps.
2. Extract the Code Insight application. See [Extracting Application Files](#) for detailed steps.
3. Copy the Code Insight Compliance Library. See [Installing the Compliance Library](#) for details.
4. Copy the Code Insight application license key to <Code Insight_ROOT_DIR>. See [Activating Code Insight](#) for details.
5. Prepare the Central Database on the Core/Scan Server or the dedicated server.
6. Download and install the database driver on the Core/Scan Server. See [Installing the Database Driver](#) for details.
7. Configure your database. See [Preparing the Database](#) for details.
8. Run these database scripts in the order shown. See [Running the Database Setup Scripts](#) for details.
 - palamida_ddl.sql (used to create the Code Insight central database schema and sd data)
 - One of the following:
 - request_form_long.sql (containing the default request form definitions)
 - request_form_short.sql
 - The custom request form that Revenera might have provided you

- `reports.sql` (containing the default report definitions)
9. Run any custom scripts provided to you by Revenera.
 10. Configure the properties files located in these directories under `<CodeInsight_ROOT_DIR>\config`. See [Editing Configuration \(Properties\) Files](#) for details.
 - `core`
 - `scanEngine`
 - `detector`
 - `scriptRunner`
 11. Configure the Tomcat web server. See [Configuring the Tomcat Web Server](#) for details.
 12. Perform any optional configuration steps:
 - Configure LDAP. See [Integrating with LDAP for Authentication \(Optional\)](#) for details.
 - Configure additional Scan Servers. See [Configuring Additional Scan Servers \(Optional\)](#) for details.
 - Configure any other custom files and pages (that is, `.html` and `.json` files for custom dashboards and grids) recommended by Revenera.
 13. Start the Tomcat web server. See [Administration: Starting & Stopping Servers](#) for details.
 14. On initial startup of Code Insight, the system automatically schedules an electronic update to run 2 minutes after the Tomcat server is started. If it fails, verify your electronic update settings. See [Running an Electronic Update](#) for details.

Extracting Application Files

This section contains the following topics:

- [Choosing an Installation Directory](#)
- [Installing the Compliance Library](#)
- [Installing Code Insight](#)
- [Installing the Database Driver](#)
- [Activating Code Insight](#)

Choosing an Installation Directory

The following instructions assume that the Core/Scan Server are located on the same instance.

- Select a base directory on the Core/Scan Server where Code Insight and all related content will be installed. This directory is referred to as <Code Insight_ROOT_DIR> in this document.
- For Windows, the recommended location is C:\CodeInsight.
- For Linux, the recommended location is \$HOME/apps/CodeInsight.

Linux users, we strongly recommended that you create a new Code Insight user for the system. Do not run the application as root.

- You may place the codebase (files to scan) in any location accessible by the Scan Server. Although not required, we recommend placing it on a separate drive to avoid the IO operations competing for the same disk space.
- When configuring the scan settings for a workspace, you will be prompted to select the scan locations. Files must be visible to the server in order to be scanned.

Installing the Compliance Library



Task *To install the Code Insight Compliance library (CL), do the following:*

Extract the library into <Code Insight_ROOT_DIR>\PDL\CL\2.x (where x is based on your library version). We recommend a 1TB drive to hold the CL and future updates to the CL.

Installing Code Insight



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

1. Extract CodeInsight_<version>.zip into <Code Insight_ROOT_DIR>.
2. Confirm the successful extraction of the following content directories:
 - <Code Insight_ROOT_DIR>\<version> \Code InsightInstallSysAdminGuide.pdf
 - <Code Insight_ROOT_DIR>\<version>\Code InsightInstallerGuide.pdf
 - <Code Insight_ROOT_DIR>\<version>\config (property files and other required directories)
 - <Code Insight_ROOT_DIR>\<version>\dbScripts (database sql scripts to run as part of the installation)
 - palamida_ddl.sql (Code Insight schema and sd data)
 - request_form_long.sql or request_form_short.sql (default request form definitions)
 - reports.sql (default report definitions)
 - <Code Insight_ROOT_DIR>\<version>\docs (documentation including tutorial for this release)
 - <Code Insight_ROOT_DIR>\<version>\logs (all logs generated by Code Insight are written to this directory)
 - <Code Insight_ROOT_DIR>\<version>\scriptRunner
 - <Code Insight_ROOT_DIR>\<version>\third?party (third?party licenses governing use of third?party materials in Code Insight 6.x)
 - <Code Insight_ROOT_DIR>\<version>\tomcat (pre?configured Tomcat with applicationdeployed)

Installing the Database Driver



Task

To install the database driver, do the following:

1. Download the appropriate JDBC driver for your database type:

Database	Version	JDBC Driver	Download Site
MySQL	8	mysql-connector-java-5.1.45.jar	https://downloads.mysql.com/archives/c-j/ (select Product Version 5.1.45 and download)
	5.6, 5.7	mysql-connector-java-5.1.x-bin.jar	http://dev.mysql.com/downloads/connector/j/5.1.html
Oracle	18c, 19c	ojdbc8.jar	https://www.oracle.com/database/technologies/appdev/jdbc-ucp-183-downloads.html
	12c R2	ojdbc7.jar	http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html
	11g, 12c R1	ojdbc6.jar	http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html
MS SQL Server	Driver version appropriate for the Java JDK or OpenJDK you are using		https://docs.microsoft.com/en-us/sql/connect/jdbc/system-requirements-for-the-jdbc-driver?view=sql-server-2017

2. Do one of the following:

- If using the supplied installer (codeinsight_6.x.jar) to install Code Insight, copy the JDBC driver to the directory where the installer file is located. (Instructions for using the installer are found in the [Installing Code Insight Using the Installer](#) chapter.)
- If installing Code Insight manually, copy the downloaded JDBC driver to the following directory:
`<Code Insight_ROOT_DIR>\<version>\tomcat\lib\`
 Instructions for installing Code Insight manually are found in the [Installing Code Insight Manually](#) chapter.

Activating Code Insight

You will need a Code Insight license key to allow access to the application. Contact Revenera to obtain your license key. A new palamida.key or codeInsight.key with data services enabled is required for use of the Code Insight Analyzer feature.



Task

To activate Code Insight, do the following;

1. Copy the codeInsight.key (or palamida.key) file to <Code Insight_ROOT_DIR>\<version>\
2. If replacing an existing key, change any passwords contained in your property files (i.e. core.db.properties) to their unencrypted form (encryption on each Code Insight key is unique).

Preparing the Database for Installation

This section contains details for preparing the database and running database set up scripts:

- [Preparing the Database](#)
- [Running the Database Setup Scripts](#)

Preparing the Database

The following user accounts are recommended for management of the Code Insight database:

- DBA or System Administrator with elevated privileges
- Creates the database
- Performs backups as necessary
- Code Insight User with Read/Write privileges
- Connects to the database
- Executes database scripts
- Updates the database as necessary

The following should be performed by a DBA or user with elevated privileges:

- Use an existing database or create a new database and schema for use by Code Insight. Below are reference links for proper syntax depending on your database vendor:
 - [MySQL Reference Manual](#)
 - [Oracle® 11g Release 2 Database Administrator's Guide](#)
 - [MS SQL Server Reference Manual](#)
- Ensure that the database name conforms to the following guidelines:
 - The first character of the database name is alphabetic.

- The database name only contains alphanumeric characters or the _, \$, #.
- Do not use dashes (-) in the database name.
- Ensure that the is configured to accept connections from the client machines that Code Insight is using (both core and Scan Servers). This is often done using the GRANT PRIVELEGES command. See the reference manual for your database type for more information.
- Create a Code Insight user with Read/Write privileges. Assign the user to the Code Insight database with a read/write role.
- Start the database. Typically, you can configure the database to run automatically as a service (Windows).

Running the Database Setup Scripts



Task

To run the database setup scripts, do the following:

1. Navigate to the location of the database scripts:

`<Code Insight_ROOT_DIR>\<version>\dbScripts\<DB_VENDOR>`

Ensure that you access the appropriate scripts for your database vendor.

2. Run the following scripts (for your database vendor) in the following order:

- palamida_ddl.sql (Code Insight schema and sd data).
- request_form_short.sql or request_form_long.sql (default request form definitions, most users select the long request form).
- reports.sql (default report definitions).

Configuring the Tomcat Web Server

This section provides details related to configuring files related to running Code Insight with Tomcat:

- [Setting Max Heap](#)
- [Setting the Install Directory](#)
- [Linux Installations: Headless Mode for JVM](#)
- [Enabling HTTP Secure \(HTTPS over SSL\)](#)
- [Increasing Post Request Size](#)
- [Running as a Windows Service \(32-bit and 64-bit\)](#)

Setting Max Heap



Task

To set Max Heap, do the following:

1. Go to the `<Code Insight_ROOT_DIR>\<version>\tomcat\bin\` directory and edit the `catalina.bat` file if running Windows or `catalina.sh` file if running Linux.
2. Update the Max Heap size (`-Xmx`) for Tomcat. Code Insight recommends that you set the Max Heap size to approximately 80% of available memory. For a 16 GB system, the value should be set to 12288m. For 32 GB systems, the value should be set to 25600. See the example below.
3. Update the Min Heap size (`-Xms`) to be equal to or less than the max heap size. For a 16 GB system, the value should be set to 12288m. For 32 GB system, the value should be set to 25600. See the example below.

Windows

```
set CATALINA_OPTS=-Xms12288m -Xmx12288m -XX:PermSize=1024m -XX:MaxPermSize=1024m
```

Linux

```
CATALINA_OPTS="-Xms12288m -Xmx12288m -XX:PermSize=1024m -XX:MaxPermSize=1024m"
```

Setting the Install Directory



Task *To set the install directory, do the following:*

1. Edit the `<Code Insight_ROOT_DIR>\<version>\tomcat\bin\catalina.bat` (or `catalina.sh`) file to point to your Code Insight Install directory.
2. Update the value of `palamidaInstallPath` as shown in these examples.

On Windows:

```
-DpalamidaInstallPath="<Code Insight_ROOT_DIR>/<version>/"
```

On Linux:

```
-DCodeInsightInstallPath="$HOME/apps/Code Insight/<version>"
```

Linux Installations: Headless Mode for JVM

For Linux installations, the Headless Mode for JVM options should be enabled. Verify that the value of `-Djava.awt.headless` is set to `true` (this is the default). See <http://www.oracle.com/technetwork/articles/javase/headless-136834.html> for more details about Headless Mode.

Enabling HTTP Secure (HTTPS over SSL)

To implement SSL, a web server must have an associated certificate configured for each Code Insight Core and Code Insight Scan Server (IP address) that accepts secure connections. For complete instructions, see [Reference: Enabling Secure HTTP over SSL](#), or follow the detailed steps in the `<Code Insight_ROOT_DIR>\<version>\tomcat\https\readme.txt` file. For more information about HTTPS/SSL, refer to http://en.wikipedia.org/wiki/HTTP_Secure and <https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>.

Increasing Post Request Size

To allow Tomcat to accept large data amounts in a submitted web form, the `maxPostSize` attribute must be increased from the default value of 10MB to your preferred size. Follow the steps below to change the `maxPostSize`:



Task *To increase post request size, do the following:*

1. Edit the `<Code Insight_ROOT_DIR>\<version>\tomcat\conf\server.xml`.

2. Search for the Look for Connector tag:

```
<Connector executor="tomcatThreadPool"  
port="8080" protocol="HTTP/1.1"  
connectionTimeout="30000"  
maxPostSize="10485760"  
redirectPort="8443" />
```

3. Add or edit the maxPostSize attribute to the required value in bytes. For example, 5MB would be 5242880.

Running as a Windows Service (32-bit and 64-bit)

For detailed steps on configuring Apache Tomcat as a Windows service, see [Configuring Code Insight as a Service](#).

Editing Configuration (Properties) Files

The following topics in this section provide specific details for editing the Configuration Properties file:

- [Entering Required Information](#)
- [Installation Options](#)
- [Tuning Copyright Detection](#)
- [Tuning Email/URL Detection](#)
- [Tuning License Detection](#)
- [Enabling the Availability of Analyzer for Scans and Reporting](#)
- [Adding Custom Phases to the Scan Process](#)
- [Adding Custom Inventory Statuses](#)
- [Disabling Security Updates for Archived or Canceled Projects](#)

Entering Required Information



Task

To enter required information in the Configuration Properties file, do the following:

1. Update the core.db.properties (located in <Code Insight_ROOT_DIR>\<version>\config\core\ with your database settings.
2. Uncomment the six lines that start with #db in the corresponding section according to your database vendor (MySQL, Oracle, or SQL Server).
3. Update the settings to match your database environment.
4. Replace <DB_HOST>, <DB_PORT>, <DB_NAME>, <DB_USERNAME>, and <DB_PASSWORD> with the appropriate values.

If no database port number is entered in lieu of <DB_PORT>, the port is defaulted to the following values:

- 3306 for MySQL Server Database
 - 1521 for Oracle Database
 - 1433 for MS SQL Server Database
5. The database password is only visible before initial startup. At startup, the application replaces the plain-text database password with an encrypted value and uses the encrypted value for all subsequent server (re)starts. To change the password, the Code Insight Administrator must shut down the server, enter a new plain-text password in the `core.db.properties` file, and restart the server. During startup, the application rereads the plain-text password and writes the encrypted password into `core.db.properties`.
6. Update `core.properties` located in `<Code Insight_ROOT_DIR>\<version>\config\core\` by doing the following:
- Enter the `core.server.url` by replacing `<CORE_SERVER_NAME>` with the name or IP address of the Code Insight Core Server, and replace `<HTTP_PORT>` with the port number of the Code Insight Core Server, which is typically set to 8888.
 - Uncomment the lines that start with `# scan.server`.
 - Replace each occurrence of `<ALIAS>` with an alias value of your choosing.
 - Replace `<SCAN_SERVER_NAME>` with the name or IP address of the Code Insight Scan Server. You can use 127.0.0.1 to refer to a local host.
 - Replace `<HTTP_PORT>` with your HTTP port, which, in general, would be 8888.
 - (Optional) Configure the “Emails Configuration” section of the property file. Email is required to receive system-generated passwords for newly created application users.
 - (Optional) Update the Electronic Updates Settings Frequency Settings section (if necessary):

```
# Electronic Updates Settings Frequency Settings
auto.update.enabled = true
auto.update.jobFrequency = 0 0 12 ? * SUN
```



Note - By default, electronic updates are configured to run weekly on Sunday at noon (12:00pm)

- (Optional) Set the web session timeout by updating the `session.timeout` parameter with the desired value in number of minutes.
 - (Optional) Update `web.home.page` to change your preferred home page. Possible values are `my-tasks`, `my-shortcuts`, `dashboard`, or `my-projects`. The default value is the dashboard tab on the Home page. If you have entered `my-tasks`, `my-shortcuts`, or `dashboard`, the corresponding tab will be the default active one for every visit to the Home page.
7. Update `scanEngine.properties` located in `<Code Insight_ROOT_DIR>\<version>\config\scanEngine` by doing the following:
- Enter the `core.server.url` by replacing `<CORE_SERVER_NAME>` with the name or IP address of the Code Insight Core Server. Replace `<HTTP_PORT>` with the port number of the Code Insight Core Server, typically set to 8888.
 - Enter the `serverURL` by replacing the `<SCAN_SERVER_NAME>` with the name or IP address of the Code Insight Scan Server. Replace `<HTTP_PORT>` with the port of the Code Insight Scan Server, typically set to 8888.

- Replace <ALIAS> with the alias of your Code Insight Scan Server. Ensure that the alias matches the one specified in core.properties.
 - Set the workspaceBaseDirPath to the location where you will store your Code Insight workspaces. The recommended path is <Code Insight_ROOT_DIR>\workspaces.
 - (Optional) Set the serverFileSystemRoot= to limit access to directories on the Code Insight Scan Server.
8. In scan.properties (located in <CodeInsight_ROOT_DIR>\<version>\config\scanEngine\scan.properties), set the signaturesDirPath to the path of the directory containing both the files ending in .dat and the bdb directory in your Code Insight CL installation. (Include the CL installation path in the value.)

Installation Options

This section provides details about configuring different email-related and Scan Server options.

Configuring Email Services



Task

To configure email services, do the following:

1. Configure core.properties (located in <CodeInsight_ROOT_DIR>\<version>\config\core\) as required to establish connection with the SMTP mail server:


```
send.email.enabled = true (Set to false to disable email notifications)
smtp.sender.email = <SENDER_EMAIL_ADDRESS>
smtp.host.name = <EMAIL_HOST_NAME>
smtp.userName = <EMAIL_USERNAME>
smtp.password = <EMAIL_PASSWORD>
smtp.host.port = 465
mail.smtp.auth = true
mail.smtp.starttls.enable = true
```
2. If email notifications are enabled, during the initial startup, the application replaces the plain-text SMTP password with an encrypted value. The application reads the plain text password and writes the encrypted password back into the core.properties file in the following format: <ENCRYPTED_PASSWORD>ENCRYPTED. For subsequent server restarts, the application uses the encrypted password. To change the password, the administrator must bring down the server.
3. Enter the new plain-text password into the core.properties file.
4. Restart the server. During startup, the application, again, reads the plain text password and writes the encrypted password back into the core.properties.
5. Ensure that the following email address (Code Insight-notifier@Code Insight.com) is on your junk email sender's safe list so that Code Insight notification emails do not end up in your junk email folder.

Disabling SMTP Mail Server Authentication



Task

To disable SMTP mail server authentication, do the following:

1. If your email server does not require authentication, use the following email settings:

```
emailsender.authentication = false  
smtp.password = mail.smtp.auth = false mail.smtp.starttls.enable = false
```

2. Also, modify an Apache Tomcat configuration file located as follows:

```
<Code Insight_ROOT_DIR>\<version>\tomcat\webapps\palamida\WEB-INF\classes\application-context.xml
```

3. Comment out the user name and password properties (shown in bold) from the mailSender section.

```
<bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">  
<property name="host" value="${smtp.host.name}"/>  
<property name="port" value="${smtp.host.port}"/>  
<!--  
<property name="username" value="${smtp.userName}"/>  
<property name="password" value="${smtp.password}"/>  
-->  
<property name="javaMailProperties">  
<props>  
<prop key="mail.smtp.host">${smtp.host.name}</prop>  
<prop key="mail.smtp.user">${smtp.userName}</prop>  
<prop key="mail.smtp.port">${smtp.host.port}</prop>  
<prop key="mail.smtp.auth">${mail.smtp.auth}</prop>  
<prop key="mail.smtp.starttls.enable">${mail.smtp.starttls.enable}</prop>  
</props>  
</property>  
</bean>
```

Configuring Email Notification Options

The following email notification options are set by default in the product. To change the type of users that are notified when comments are posted, policies are changed or requests are modified, add the following properties to the core.properties file and modify as necessary:

```
comment.notification.role = requester,reviewer policy.change.notification.role = policy_admin  
request.change.notification.role = requester,reviewer
```

Configuring Notification Email Templates



Task

To configure notification email templates, do the following:

1. Notifications for workflow and user management actions are sent to the user as html-formatted email messages, the location of which is <Code Insight_ROOT_DIR>\<version>\config\core\emails.
2. These notification messages are configurable using a template-based formatting engine and can be modified as necessary.

3. To customize these messages, we recommended that you make a new template file for each customized template and reference the customized template in core.properties by adding one or more of the following properties and modifying to point to the new template.:

```
# Email Templates
email.account.registered.subject = emails/new_account_subject.html email.account.registered.body =
emails/new_account_body.html
email.forgot.password.subject = emails/forgot_password_subject.html email.forgot.password.body =
emails/forgot_password_body.html
email.new.request.subject = emails/notify_new_request_subject.html email.new.request.body = emails/
notify_new_request_body.html email.renewed.request.subject = emails/
notify_renewed_request_subject.html email.renewed.request.body = emails/
notify_renewed_request_body.html
email.new.requester.subject = emails/notify_new_requester_subject.html email.new.requester.body =
emails/notify_new_requester_body.html email.old.requester.subject = emails/
notify_old_requester_subject.html email.old.requester.body = emails/notify_old_requester_body.html
email.completed.request.subject = emails/notify_completed_request_subject.html
email.completed.request.body = emails/notify_completed_request_body.html
email.comment.to.request.subject = emails/new_comment_subject.html email.comment.to.request.body =
emails/new_comment_body.html
emailReviewer.request.subject = emails/reviewer_request_subject.html emailReviewer.request.body =
emails/reviewer_request_body.html emailReviewer.reject.subject = emails/
reviewer_reject_subject.html emailReviewer.reject.body = emails/reviewer_reject_body.html
emailReviewer.approve.subject = emails/reviewer_approve_subject.html emailReviewer.approve.body =
emails/reviewer_approve_body.html emailReviewer.renewed.request.subject = emails/
reviewer_renewed_request_subject.html emailReviewer.renewed.request.body = emails/
reviewer_renewed_request_body.html
email.policy.changes.body = emails/policy_changed_body.html email.policy.changes.subject = emails/
policy_changed_subject.html email.policy.create.body = emails/policy_created_body.html
email.policy.create.subject = emails/policy_created_subject.html email.policy.delete.body = emails/
policy_deleted_body.html email.policy.delete.subject = emails/policy_deleted_subject.html
email.request.form.data.changes.body = emails/request_form_data_change_body.html
email.request.form.data.changes.subject = emails/request_form_data_change_subject.html
email.request.recall.body = emails/request_recall_body.html email.request.recall.subject = emails/
request_recall_subject.html
email.task.define.project.policies.body = emails/project_define_policies_body.html
email.task.define.project.policies.subject = emails/project_define_policies_subject.html
email.task.define.project.details.body = emails/project_define_details_body.html
email.task.define.project.details.subject = emails/project_define_details_subject.html
email.task.manage.audit.process.body = emails/project_manage_audit_process_body.html
email.task.manage.audit.process.subject = emails/project_manage_audit_process_subject.html
#Inventory Workflow Description
email.task.select.compliance.action.body = emails/inventory_select_compliance_action_body.html
email.task.select.compliance.action.subject = emails/
inventory_select_compliance_action_subject.html
email.task.create.remediation.Notes.body = emails/inventory_create_remediation_Notes_body.html
email.task.create.remediation.Notes.subject = emails/
inventory_create_remediation_Notes_subject.html
email.task.review.inventory.body = emails/inventory_review_inventory_body.html
email.task.review.inventory.subject = emails/inventory_review_inventory_subject.html
email.task.review.cve.body = emails/inventory_review_cve_body.html email.task.review.cve.subject =
emails/inventory_review_cve_subject.html
email.task.review.request.body = emails/task_review_request_body.html
email.task.review.request.subject = emails/task_review_request_subject.html
email.task.unassignment.body = emails/task_unassignment_body.html email.task.unassignment.subject =
emails/task_unassignment_subject.html
```

```
email.task.reminder.body = emails/task_reminder_body.html email.task.reminder.subject = emails/
task_reminder_subject.html
email.pull.inventory.completion.body = emails/pull_inventory_completion_body.html
email.pull.inventory.completion.subject = emails/pull_inventory_completion_subject.html
email.project.inventory.update.body = emails/project_inventory_update_body.html
email.project.inventory.update.subject = emails/project_inventory_update_subject.html
email.inventory.question.body = emails/inventory_question_body.html
email.inventory.question.subject = emails/inventory_question_subject.html
email.inventory.question.answered.body = emails/inventory_question_answered_body.html
email.inventory.question.answered.subject = emails/inventory_question_answered_subject.html
email.inventory.checklistitem.body = emails/inventory_checklistitem_body.html
email.inventory.checklistitem.subject = emails/inventory_checklistitem_subject.html
email.inventory.checklistitem.completed.body = emails/inventory_checklistitem_completed_body.html
email.inventory.checklistitem.completed.subject = emails/
inventory_checklistitem_completed_subject.html
email.project.summary.body = emails/project_summary_body.html email.project.summary.subject =
emails/project_summary_subject.html
email.project.copy.body = emails/project_copy_body.html email.project.copy.subject = emails/
project_copy_subject.html
email.scan.task.body = emails/scan_task_body.html email.scan.task.subject = emails/
scan_task_subject.html
email.license.expiration.body = emails/license_expiration_body.html
email.license.expiration.subject = emails/license_expiration_subject.html
email.scan.server.down.body = emails/scan_server_down_body.html email.scan.server.down.subject =
emails/scan_server_down_subject.html
email.update.service.failure.body = emails/update_service_failure_body.html
email.update.service.failure.subject = emails/update_service_failure_subject.html
email.clupgrade.available.body = emails/clupgrade_available_body.html
email.clupgrade.available.subject = emails/clupgrade_available_subject.html
email.clupgrade.error.body = emails/clupgrade_error_body.html email.clupgrade.error.subject =
emails/clupgrade_error_subject.html email.clupgrade.success.body = emails/
clupgrade_success_body.html email.clupgrade.success.subject = emails/clupgrade_success_subject.html
email.clupgrade.stopped.body = emails/clupgrade_stopped_body.html email.clupgrade.stopped.subject =
emails/clupgrade_stopped_subject.html
```

4. Contact Code Insight Support for further instructions.

Configuring a Manual Proxy Server Connection (Optional)

By default, Code Insight uses automatic proxy server settings. This is used to load the remote files for the side-by-side comparison of source matches in the Detector client. To provide credentials to authenticate to a proxy server, do the following on the Core\Scan Server.



Task

To authenticate to a proxy server, do the following:

1. Configure `core.proxy.server.properties` in `<Code Insight_ROOT_DIR>\<version>\config\core\` as required to establish a connection with the proxy server:

```
#Indicates whether JetS3t should auto-detect the HTTP proxy settings appropriate for the local
machine, default: true
#httpClient.proxy-autodetect=false
#Sets the host name of a HTTP proxy server
#httpClient.proxy-host=<PROXY_HOST>
```

```
#Sets the port number of a HTTP proxy server
#httpClient.proxy-port=<PROXY_PORT>
#Sets the user name credential for proxy authentication
#httpClient.proxy-user=<PROXY_USER>
#Sets password credential for proxy authentication
#httpClient.proxy-password=<PROXY_PASSWORD>
#Sets the domain credential for proxy authentication
#httpClient.proxy-domain=<PROXY_DOMAIN>
```

Scan Engine Configuration Options

The following properties are used to configure a given Scan Server on an instance.



Task

To update scan engine configuration for a given Scan Server, do the following:

1. Open scanEngine.properties (<Code Insight_ROOT_DIR>\<version>\config\scanEngine) on the instance where the Scan Server is installed, and edit the settings as needed.
2. To control which directories can be selected for scanning, set the serverFileSystemRoot property:
 - Uncomment the serverFileSystemRoot property and specify a list of base directories accessible via the Code Insight Scan Server.



Note ▪ This property may be required to run on some platforms such as Windows Server 2008.

- This property controls which locations can be selected by the Application Administrator as allowable for scanning for all projects in a given group, as well as which locations can be selected for scanning as part of the workspace settings.

Tuning Copyright Detection

This section contains three different ways you can tune copyright detection results. The files mentioned (ignoredCopyrights.txt, customCopyrights.txt, and claimedCopyrights.txt) are located in the following directory:

```
<Code Insight_ROOT_DIR>\<version>\tomcat\webapps\palamidaScanEngine\WEB-INF\classes\config
```

To customize any of the default settings, copy one or more of these files into the following directory:

```
<Code Insight_ROOT_DIR>\<version>\config\scanEngine
```

Modify the copied files to override the default settings.

Controlling False Positives



Task

To control false positives or to ignore copyrights that would normally be detected, do the following:

Modify the `ignoredCopyrights.txt` file. This file is used to define the copyright patterns, including partial patterns that result in the `-unparseable-` bucket, that are ignored by the copyright scanner.

- A set of case-sensitive strings is provided by default and cause the copyright scanner to skip the current line in the file being scanned if these strings are encountered. The set is as follows:

```
the above copyright
retain the copyright
COPYRIGHT HOLDER
COPYRIGHT OWNER
copyright notice
copyright condition
```

- Make sure to follow these rules when defining copyright patterns to ignore:
 - Do not use commas (,) while defining copyright, date or owner patterns. Since commas are used as delimiters, they will end up tokenizing your patterns into multiple values. For example: “Code Insight, Inc.” should be defined as “Code Insight Inc.” (Note that there is no comma in the owner pattern.)
 - Do not use any of the following reserved characters: `\n\r|#!%:;,/*" space`. If you use these reserved characters, they will be replaced by a space (' ') character.
 - If Java special characters are part of the defined copyright pattern, ensure that double slashes are used to escape both the properties file parsing as well as the Java interpretation. For example, “Copyright [C] 2013” would be defined as “Copyright \\[C\\] 2013”.
- The following regular expression patterns are part of the standard patterns used by the copyright scanner to force partial copyright matches into the `-unparseable-` bucket when the application cannot determine the copyright owner. To discard these copyright matches rather than have them end up in the `-unparseable-` bucket, you must uncomment the desired regular expression patterns:

```
(?i)\\bcopyright(?:s|ed)?\\s+(?:and\\s+)?(?:notice|holder|license|trademark|law|patent|this|disclaimer|label| under|GPL|message|history|abandoned|s|information)s?\\b
(?i)\\b(?:BSD-
style|the|is|by|additional|pieces|and|earlier|other|proper|following|original)\\s+copyright(?:s|ed)?
\\b
\\[yyyy\\] (?i)\\bUniversity\\s+copyright-
\\bi\\s+copyright\\b
\\bCDDL HEADER\\b
=(?:head\\d+|item)\\b (?i)\\bdisclaims\\bcopyright\\b (?i)(?<!<copyright>.{0,100}</copyright>\\s*$
```

Controlling False Negatives

To control false negatives or to force the copyright scanner to detect copyright patterns that would ordinarily be ignored, you must modify the `customCopyrights.txt` file. Read and follow the instructions when defining custom copyright patterns.

Every copyright pattern consists of the following parts:

- **Copyright:** Enter the copyright indicator pattern or case-sensitive string to be matched, as in the example:
- `\\(c\\), \\(C\\), Copyright`
- **Date:** Enter the date pattern. (This can be left empty.)
- **Owner:** Enter the copyright owner pattern or case-sensitive string.
- **Sequence:** Enter a comma-separated list of elements in the order the elements are to be matched, as in the example:
`copyright,date,owner` or `copyright,owner,date`



Task

To control false negatives, do the following:

1. Make sure to follow these rules when defining copyright patterns. Follow the syntax shown in the following pattern:


```
Code Insight:(c) Code Insight, Inc.
C2:Copyright Jack Smith 2008, 2009
C3:Copyright (C) Author S Name
pattern.names = Code Insight,C2,C3
Code Insight.copyright = \\(c\\) Code Insight.date = Code Insight.owner = Code Insight Inc.
Code Insight.sequence = copyright,date,owner
C2.copyright = [C|c]opyright C2.date = 2008 2009
C2.owner = Jack Smith
C2.sequence = copyright,owner,date
C3.copyright = Copyright \\(C\\) C3.date =
C3.owner = Copyright \\(C\\) C3.sequence = copyright,date,owner
```
2. Ensure that your pattern names are defined in the pattern.names list. If a pattern name is not defined in the list, it will not be processed.
3. Each copyright pattern element is padded with one or more spaces (`\s*`) to allow for leading and trailing spaces, so there is no need to pad your patterns with spaces.
4. Do not use commas (,) while defining copyright, date or owner patterns. Commas are used as delimiters, so they will tokenize your patterns into multiple values. For example, `Code Insight, Inc.` should be defined as `Code Insight Inc.` There is no comma in the owner pattern.
5. Do not use any of the following reserved characters: `\n \r | # ! % : ; , / * " space`. If you use these reserved characters, they will be replaced by a space (' ') character.
6. If Java special characters are part of the defined copyright pattern, ensure that double slashes are used to escape both the properties file parsing as well as the Java interpretation. For example, `Copyright [C] 2008` would be defined as `Copyright \\[C\\] 2008`.

Claiming Detected Copyrights as Yours



Task

To claim detected copyrights, do the following:

1. The claimed Copyrights.txt file may be used to force detected copyrights (based on default copyright patterns, or those supplemented via the customCopyrights.txt file) into a single bucket called -claimed-.
2. The purpose of modifying this file is to suppress third-party indicators due to detection of copyrights that belong to the user (own, purchased, acquired, contracted work, etc.). If a scanned file only contains a claimed copyright, it won't be tagged as containing third-party indicators.
3. Make sure to follow these rules when defining copyright patterns for which to take ownership and move to the - claimed- bucket:
 - Do not use commas (,) while defining copyright, date or owner patterns. Since commas are used as delimiters, they will end up tokenizing your patterns into multiple values. For example, Code Insight, Inc. should be defined as Code Insight Inc. (Note that there is no comma in the owner pattern.)
 - Do not use any of the following reserved characters: \n\r | # ! % : ; , / * " space. If you use these reserved characters, they will be replaced by a space (' ') character.
 - If Java special characters are part of the defined copyright pattern, ensure that double slashes are used to escape both the properties file parsing as well as the Java interpretation. An example of this is Copyright [C] 2008 would be defined as Copyright \\[C\\] 2008.

Tuning Email/URL Detection

This section explains how you can tune email/URL detection results. The ignoredEmailURL.txt file is located in the following directory:

```
<Code Insight_ROOT_DIR>\<version>\tomcat\webapps\palamidaScanEngine\WEB-INF\classes\config
```

If you want to customize this file, you can either edit the file in this directory directly, or copy it to <Code Insight_ROOT_DIR>\<version>\config\scanEngine, and modify it there.

The Scan Server parses these files before each workspace scan. The ones in <Code Insight_ROOT_DIR>\<version>\config\scanEngine take precedence.

Controlling False Positives

The file mentioned above may be used to force the scanner to skip a detected Email and URL. The purpose of this is to not treat a file as having third-party content if it only has an internal email or URL within a string literal or comment.

- Each line in the file is treated as a string to match.
- Regular expressions are *not* supported.
- An email or URL is ignored if it contains any of the strings defined in the file.
- The comparison is case-insensitive.



Note - Files that contain only ignored Emails and/or URLs will not be tagged as containing them. If you look at the file in the Detector client, though, the ignored Emails and URLs are highlighted because the highlighting in the Detector client uses its own search rather than being based on the scan results.

Tuning License Detection

This section explains how to tune license detection results. The `license_detection.xml` file is located in the following directory:

```
./palamida_6.14.x/tomcat/webapps/palamida/WEB-INF/classes/
```

To customize this file add additional license patterns, or disable existing ones, you can edit the file in the directory or copy it to `<Code Insight_ROOT_DIR>\<version>\config\core\licenses` and modify it there.

The Scan Server parses this file before each scan. The one in `<Code Insight_ROOT_DIR>\<version>\config\core\licenses` take precedence.

Editing the License Detection Patterns File

The file mentioned above may be used to force the scanner to automatically detect a license and create a system group for all files containing the defined patterns for this license. The following are patterns used for license detection:

- **License Text Templates:** Allow matches against the entire license. The * are wildcards replaced with the copyright holder name or product name.
- **License Keywords:** Allow matching license references based on defined key word sets. All keywords defined in a keyword set must be matched for the license to be detected by the scanner.

License Text Template Example for Apache License, Version 2.0

The following is a license text example for the Apache License, Version 2.0.

```
<licenseTextTemplate>
Licensed to the * under one or more contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership. The ASF licenses this file to you
under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0.
Unless required by applicable law or agreed to in writing, software distributed under the License is
distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
</licenseTextTemplate>
```

License Keywords Example for Apache License, Version 2.0

The following is an example list of license keywords for the Apache License, Version 2.0.

```
<keywords>
keywordSet>
```

```
<keyword text="Licensed under the Apache License"/>
</keywordSet>
<keywordSet>
<keyword text="Apache Software Foundation"/>
</keywordSet>
<keywordSet>
<keyword text="Apache License, Version 2.0"/>
</keywordSet>
<keywordSet>
<keyword text="Apache License Version 2.0"/>
</keywordSet>
<keywordSet>
<keyword text="http://www.apache.org/licenses/LICENSE-2.0"/>
</keywordSet>
</keywords>
```

Enabling the Availability of Analyzer for Scans and Reporting

As of Code Insight 6.13.3, the Analyzer analysis technique is no longer available by default for workspace scanning and reporting. Once you migrate to 6.13.3 or later, you must manually enable the Analyzer in Code Insight if you want to make it available.



Task

To enable the availability of the Analyzer workspace scanning and reporting, do the following:

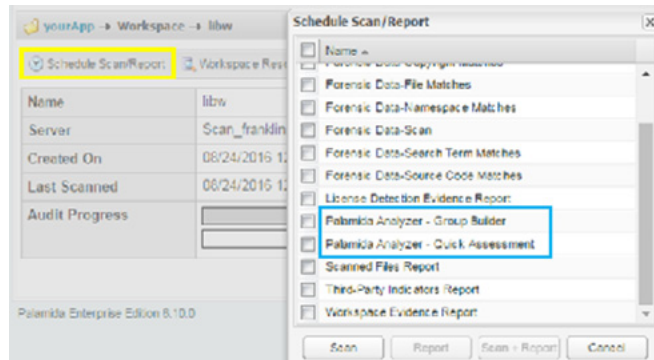
- Anytime after you have migrated to Code Insight 6.13.3 or later, open the <Code Insight_ROOT_DIR>\config\scanEngine\scanEngine.properties file in a text editor.
- Update the disableAnalyzer property to false:
`disableAnalyzer=false`
- Save the file and restart the server to apply the change.

The **Enable Analyzer** option is now available on **Automated Analysis** tab for any workspace. Users might need to adjust the settings on this tab for a given workspace. (Depending on the scripts run during the migration process, **Enable Analyzer** might or might not be automatically selected. See the Release Notes for migration details.)

General	Detection	Source Code Options	Rescan Options	Automated Analysis
Enable POM File Analyzer <input type="checkbox"/>				
Enable Auto-WriteUp™ <input checked="" type="checkbox"/>				
Enable Multi-Indicator Detector <input checked="" type="checkbox"/>				
Add Non-Indicator Files to Groups? <input type="checkbox"/>				
Enable Analyzer <input checked="" type="checkbox"/>				
Enable CodeAware <input type="checkbox"/>				
CodeAware dependency level <input type="text" value="No Dependencies"/>				

Note: If a project workspace was previously scanned with Analyzer, you are strongly recommended not to enable

The **Schedule Scan/Report** dialog now lists Analyzer-specific reports.



Adding Custom Phases to the Scan Process

The Code Insight scan process loads and executes a set of groovy scripts, called *scan-phase adapters*, to perform the various phases of the scan process. Code Insight ships with a set of standard scan-phase adapters located in the `config/scanEngine/autorun` directory of your Code Insight installation. During the scan process, the scan automatically loads and executes only those adapters specific to your scan requirements and configuration. (For example, if Auto-Writeup is enabled on the **Automated Analysis** tab, the scan will load and execute `autorun.Auto-WriteUp.groovy` to perform Auto-Writeup as a phase during the scan process. If Auto-Writeup is not enabled, the adapter is not loaded.)

Code Insight also lets you provide your own scan-phase adapters (in addition to the standard adapters) to add phases that perform custom tasks at various points in the scan process. For example, you might need a phase that collects and returns scan results to your custom Web-based UI before analysis techniques such as Auto-Writeup, CodeAware, or Analysis are run.

Refer to the following sections for more information about providing a custom scan-phase adapter:

- [Sample Custom Scan-Phase Adapter Script](#)
- [Rules and Considerations for Creating a Custom Scan-Phase Adapter](#)

Sample Custom Scan-Phase Adapter Script

To assist you in creating a scan-phase adapter script, Code Insight provides a sample groovy script called `autorun.CustomScanPhaseAdapter.groovy.example`, located with the other adapters in the `config/scanEngine/autorun/` directory. The script includes example code implementations of the various scan-phase methods available for creating the adapter, along with comments describing each method. You can actually test this sample adapter to see how the various scan-adapter methods execute. (To run the test, ensure that the adapter is in the `autorun` directory and that you have renamed it `autorun.CustomScanPhaseAdapter.groovy`. Then execute a scan.)

The sample script can serve as a template for developing a groovy script for your own scan-phase adapter. After you create the script, you must save it to the `config/scanEngine/autorun` directory. For more information about creating and naming a custom scan-phase adapter, see the next section, [Rules and Considerations for Creating a Custom Scan-Phase Adapter](#).

Rules and Considerations for Creating a Custom Scan-Phase Adapter

The following highlights information you need to know when creating a custom scan-phase adapter:

- [Class Used in Custom Scan-Phase Adapter](#)
- [Scan Phase Order](#)
- [Multiple Scripts Using Same Method](#)
- [Customizing the Scan Phase Order](#)

Class Used in Custom Scan-Phase Adapter

The groovy script for your custom scan-phase adapter must define a class that extends the Code Insight class `ScanPhaseAdapter`. Use the following syntax to define the class for the custom scan-phase adapter, where `<CustomAdaptorClass>` represents this class name:

```
workspace.getInternalWorkspace().addScanPhaseListener(new <CustomAdapterClass>(workspace))
static class <CustomAdapterClass> extends ScanPhaseAdapter{
    Workspace localWorkspace;
    <CustomAdapterClass>(WorkspaceCover workspaceCover) {
        localWorkspace = workspaceCover.getInternalWorkspace();
    }
    <code for custom scan-phase>
}
```

Scan Phase Order

Your custom scan-phase adapter code uses one (or possibly more) of the following methods corresponding to scan phases. The scan calls these methods in the order shown:

- `void beforeScan()`
- `void afterScan()`
- `void afterCommit()`
- `void afterAnalysis()`
- `void list<string> afterAnalysis2(list<string> inFiles)`

Note the following:

- Auto-Writeup executes in the `afterAnalysis` method, while CodeAware and Analyzer execute in the `afterAnalysis2` method. Most likely your custom adapter code will run in the `afterAnalysis2` method.
- Use the `afterScan` and `afterCommit` methods judiciously in a custom adapter script as these can break the scan.

Thus, your custom scan-phase adapter script is loaded and run based on the order in which the scan calls the method used in your custom adapter code. For example, if your script uses the `afterAnalysis` method in which to run your custom adapter code, the custom adapter will execute in the same `afterAnalysis` phase as Auto-Writeup but before the `afterAnalysis2` phase in which CodeAware and Analyzer run.

Refer to the comments in `autorun.CustomScanPhaseAdapter.groovy.example` for detailed descriptions of each of the scan-phase methods. For more information about this sample script, see [Sample Custom Scan-Phase Adapter Script](#).

Multiple Scripts Using Same Method

Two or more scan-phase adapter scripts can use the same scan-phase method, such as `afterAnalysis2`. When the scan calls the method, it is executed across all scripts containing it. The order in which the method is executed across the scripts depends on the alphabetic order of the script names. For example, the `autorun.CodeAwareAnalyzer.groovy` script uses the `afterAnalysis2` method in which to execute `CodeAware`. If your custom adapter script, named `autorun.CustomAdapter.groovy`, also uses the `afterAnalysis2` method in which to run your custom adapter code, the `autorun.CodeAwareAnalyzer.groovy` script runs first alphabetically, thus executing `CodeAware` first, followed by the execution of your custom adapter script. If you want your script to load before `CodeAware`, you need to provide a name for your script that alphabetically positions it before the `CodeAware` script, such as `autorun.bbb.groovy`.

The exception to the alphabetic order in which a given method is run across scripts is the Analyzer scan-phase adapter script (`autorun.Analyzer.groovy`), which is always run last in the Analyzer2 scan phase, and thus at the end of the scan process. Therefore, if you want your custom scan-phase adapter to run after the Analyzer, use file name conventions similar to the following (which uses `customAdapter` as the example name of the custom scan adapter):

- Rename the Analyzer adapter script file to `zzz.autorun.Analyzer.groovy`.
- Save your custom script with a file name that is positioned alphabetically *after* the Analyzer adapter name. For example, you might name the custom adapter script `zzz.customAdapter.groovy`.

If you prefer not to use alphabetical order in which to run your custom scan-phase adapter within a given scan phase, you can customize the order in which it runs. See [Customizing the Scan Phase Order](#).

Customizing the Scan Phase Order

Code Insight provides the file `autorun.file.load.sequencer.json.example` in the `config/scanEngine/autorun` directory to enable you to position your custom scan-phase adapter script so that it runs at a desired time (instead of running alphabetically by script name).

To use this file, rename it to `autorun.file.load.sequencer.json`. Refer to the comments in the file for instructions on how to properly position your custom scan-phase adapter to execute at the desired time.

You are strongly recommended *not* to change the execution order of the standard scan-phase adapter scripts.

Adding Custom Inventory Statuses

Code Insight provides standard status options to communicate the status of an inventory item during its review process—"Ready for Review", "Approved", "Rejected", "Pending Review", and "Needs More Information". Code Insight also lets you define custom inventory statuses to add to the standard statuses. In this way, the available status options can best reflect the inventory review workflow at your site.

Use the following procedure to create one or more custom statuses.



Task

To create custom statuses, do the following:

1. Open `<Code Insight_ROOT_DIR>_<version>\config\core\inventory.custom.status.web.json`.
The file contents show a template to which to add one or more custom statuses.
2. Add the status labels as shown, enclosing each label in quotations; and, for multiple statuses, separate each entry with a comma:

```
{
    "statuses": [
        "Approve Internal", "Approve Legal", "Reject internal"
    ]
}
```

3. Save and close the file.
4. Restart the Tomcat server to enable the statuses.

Once a custom status is enabled, it is available for selection from the **Review Status** dropdown on the **Inventory Details** page for a given inventory item in the project workspace.

Disabling Security Updates for Archived or Canceled Projects

By default, an Electronic Update checks the inventory across all Code Insight projects, including archived or canceled projects, to determine where to apply the latest security-vulnerability updates. If you have a large number of archived or canceled projects (or these projects have large inventories), Electronic Update performance can be negatively impacted. Code Insight provides a global option to disable the security-vulnerability updates for all such projects during an Electronic Update.



Task

To disable security-vulnerability updates for archived or canceled projects, do the following:

1. Open the `<Code Insight_ROOT_DIR>\config\core\core.properties` file in a text editor.
2. Update the `disable.security.updates.for.archived.project` property to true:
`disable.security.updates.for.archived.projects = true`
3. Save the file and restart the server to apply the change.

Administration: Starting & Stopping Servers

The following topics provide information about starting, stopping, and validating the server:

- [Starting and Stopping the Server Manually](#)
- [Validating a Successful Server Startup](#)
- [Changing the Number of Log Files Maintained](#)

Starting and Stopping the Server Manually



Task

To start and stop Tomcat, do the following:

1. Execute one of the following:

<Code Insight_ROOT_DIR>\<version>\tomcat\bin\startup.bat (or startup.sh on Linux installations)

or

catalina.bat run > ../logs/catalina.out 2>&1 (in Windows)

or

./catalina.sh run > ../logs/catalina.out 2>&1 (in Linux)

2. To stop Tomcat, execute:

<Code Insight_ROOT_DIR>\<version>\tomcat\bin\shutdown.bat (or shutdown.sh on Linux installations)

Validating a Successful Server Startup

You can validate a successful server startup in the following ways.

- [Using a Supported Browser](#)
- [Using an .xls File](#)

Using a Supported Browser



Task **To use a supported browser to validate a successful server startup, do the following:**

1. Log into Code Insight using a supported browser.
2. Go to the URL: `http://<server>:8888/palamida`.
3. Enter username: `admin`.
4. Enter password: `Password123`.

Using an .xls File

In addition, you can use a downloadable .xls file (Excel spreadsheet file) to create a demo configuration.



Task **To use an .xls file to validate a successful server startup, do the following:**

1. Go to Administration.
2. Select **Import** from the pull-down menu.
3. Select **Download a sample workbook for bulk import**.



Note ▪ You can also manually create user accounts via the web UI.

Changing the Number of Log Files Maintained

Code Insight generates several versions of the Scan Server log files to assist in troubleshooting. All log files are written to the following:

```
<Code Insight_ROOT_DIR>\<version>\logs\
```

By default, 30 versions of the `scanEngineDetails.log` and `scanEngineSupport.log` files are saved before the log file contents are overwritten.



Task

To change the number of log file versions, do the following:

1. Open the log4j scan engine properties file:
`<Code Insight_ROOT_DIR>\<version>\tomcat\webapps\palamidaScanEngine\WEB-INF\classes\log4j-scanEngine.properties`
2. Change the following entry to the desired number of log file versions:
`log4j.appender.FILE.maxBackupIndex=30.`

Running an Electronic Update

This section provides the following information about running Code Insight electronic updates.:

- [Running an Electronic Update for the First Time](#)
- [Running an Electronic Update Manually](#)
- [Scheduling Automatic Electronic Updates](#)

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

Running an Electronic Update for the First Time

On initial startup of Code Insight, the system will automatically schedule an electronic update to run 2 minutes after the Tomcat server is started.



Note ▪ *There is currently no way to disable the update from running the first time.*

Running an Electronic Update Manually

To run an updated manually, log in as an Application Administrator, navigate to the Administration Updates menu option.

Using the Electronic Update Server (HTTPS)

On the “Last Update” tab, click on the “Check for Electronic Update” button to connect to the Code Insight Update HTTPS server and check whether an update is available. If one is available, follow the prompts to process the update.

Using a Local Electronic Update Archive



Task

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

1. Switch to the “Manual Update” tab and browse for the manifest (update_manifest.txt) and data file (update.zip).
2. Follow the prompts to process the update.

Scheduling Automatic Electronic Updates

To configure the system to automatically process electronic updates, the following settings need to be set in the <Code> Insight_ROOT_DIR>\<version>\config\core\core.properties file:

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



Note ▪ If an update has never successfully completed on this server, the property above does not apply (even when set to “true”).

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

Refer to <http://www.quartz-scheduler.org/documentation/quartz-2.x/tutorials/crontrigger.html> for further information about the cron expression format.

Managing NG-Bridge Updates for Code Insight

Starting with the 6.14.2 SP1 release, Code Insight began support for a secondary data source for digest matches as an overlay to the data in the CL. This second source, 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 matching functionality.

Exact-file Matching with the CL and NG-bridge

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

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 on the **Exact Matches** tab for a scanned file in the Code Insight Detector.

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 enabled once you have installed or upgraded Code Insight so that you can begin obtaining NG-bridge data updates. When you start up Tomcat for the first time, an NG-bridge data update is automatically triggered if Code Insight detects that no previous updates have ever been downloaded. If previous updates are detected, the NG-bridge updates will start occurring at the regularly scheduled time (by default, 2 AM).

If you want to disable the NG-bridge data update facility (for example, if you are not performing exact-file matching), follow the procedure described in [Enabling/Disabling NG-bridge Data Updates](#). Once disabled, the facility can always be re-enabled 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.



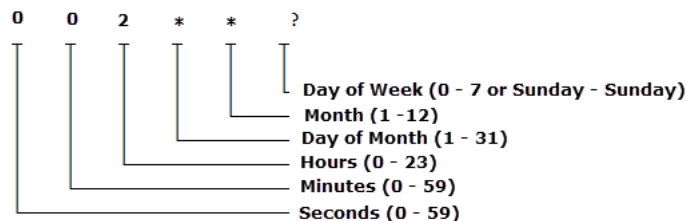
Task

To change the scheduled time NG-bridge data updates, do the following:

1. Open the `<codeInsightInstallPath>\config\scanEngine\scanEngine.properties` file.
2. Change the current `checkForUpdatesCron` property to reflect the new time. The following shows the default value, which triggers a daily CL update process at 2 am:

```
checkForUpdatesCron= 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 disable or enable the internal NG-bridge data update facility that downloads and processes the updates. By default, this update facility is initially enabled. However, if your site does not need exact-file matching during scans, you can disable the facility.

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:

1. Navigate to the `<codeInsightInstallPath>\config\scanEngine\` directory, and open the `scanEngine.properties` file.
2. Change the `enable.ngbridge` property to **false**.

Enabling NG-Bridge Data Updates

The Code Insight System Administrator or database administrator can use this procedure to re-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:

1. Navigate to the `<codeInsightInstallPath>\config\scanEngine\` directory, and open the `scanEngine.properties` file.
2. Change the `enable.ngbridge` property to **true**.

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 10-1 ■ Arguments for Script Used to Download NG-Bridge Data Update Files

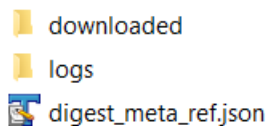
Argument	Description
Argument with defaults	Explicitly providing the following arguments is optional. If the argument is omitted, its default is used during execution.
delete.previous.downloads	<p>The true or false boolean for deleting any previously downloaded NG-bridge data update files found in the location specified by file.download.path. The default for this argument is false so that these files are retained.</p> <p>If you choose to delete previously downloaded update files, note that the diget.meta.ref.json is always retained.</p>
file.download.path	<p>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.</p> <p>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).</p>
update.https.url	<p>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.</p>

Table 10-1 ■ Arguments for Script Used to Download NG-Bridge Data Update Files (cont.)

Argument	Description
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.



- **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.

Installing Code Insight Using the Installer

This section describes the steps to install Code Insight with the Installer:

- [Installing Code Insight With the Installer](#)
- [Installing Code Insight Using the Linux Command Line](#)

For the manual installation process, see [Installing Code Insight Manually](#).

Installing Code Insight With the Installer



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

1. Ensure that you have met all the prerequisites in [Prerequisites](#).
2. Place the following items in the same directory, such as: C:\CodeInsight\Installer\:

Type	File
Installer Jar File	codeinsight_6.x.jar x is the latest release number of Code Insight.
Key	codeInsight.key
Database Driver	The JDBC driver that you downloaded based on your database type. See Installing the Database Driver for details.

3. To launch the installer, double-click the codeinsight_6.14.x.jar file or enter the following at the command prompt:

```
java -jar codeinsight_6.14.x.jar
```

The **Introduction** panel opens.

4. Click **Next**. The **Choose Install Set** panel appears.
5. Choose one of the following installation options:
 - **Standalone**—Both Core Server and Scan Server on this instance.
 - **Core**—Configure this instance as the Core Server.
 - **Scanner**—Configure this instance as the Scan Server.

The Core Server controls your Web UI and Detector Client. The Scan Server is where actual scanning is performed.



Note ▪ This guide addresses the Standalone install but it is also common to have a single instance of the Core Server and multiple Scan Servers.

6. Click **Next**. The **Choose Install Folder** panel appears.
7. Specify the Code Insight Installation directory:
 - To choose an existing directory, click **Choose** and navigate to the directory where you want to install Code Insight. When the directory appears, click **Select** and then click **Next**.



Note ▪ For Windows, the recommended location is `C:\CodeInsight\<version>`, which is referred to as `<CODE_INSIGHT_ROOT_DIR>` throughout this document.

- To accept the default installation directory, click **Next**.



Note ▪ To install Code Insight on another drive, click the **Choose** button and enter the drive letter (such as `D:\`). Then press `Enter`, and browse to the directory on the other drive.

8. Click **Next**. The **Total System Memory** panel appears.
9. Select one of the options to specify the total system memory (RAM) on this machine.



Note ▪ If the amount of RAM on your machine is not listed, round up or down to the closest number.



Tip ▪ To find the RAM of your machine, select **Control Panel** from the Windows **Start** menu and select **System and Security**. Select **View amount of RAM and processor speed** from under the System heading. Note the amount of installed memory (RAM). As an alternative, open the command prompt and type `systeminfo` | find "Total Physical Memory" and divide the resulting number by 1024.



Note ▪ The heap space for the Java VM will be automatically set to 80% of the total RAM amount. This is the recommended amount for running Code Insight.

10. Click **Next**. The **Choose Database** panel appears.

11. Select your database vendor.

Code Insight offers support for **Oracle**, **MySQL** and **SQL Server** database types. The input fields will be different depending on which database type you choose.



Note ▪ You must have an empty schema and user account configured with the correct privileges before selecting the Database vendor. See the [Databases](#) section for details.

12. Click **Next**. A **Configure Database** panel specific to the selected database type appears.



Note ▪ If you see an error indicating your database driver is missing, ensure that the database driver is in the same directory as the Code Insight installer jar file before proceeding with the installation.

13. On the **Configure Database** panel, enter information pertaining to your database vendor.



Note ▪ The IP address defaults to the IP address of the server on which you are running the Installer. You may use `localhost` instead of IP if the database is on this machine.



Note ▪ The Database password will be encrypted after the first launch of the application.

14. Click **Next**. The **core.properties** panel appears.
15. Enter an alias for your Scan Server in the **Scanner Server Alias** field. This is the alias you will see in the Code Insight Web UI when you configure and launch your scans.
16. The **Core/Scan Server IP Address** field automatically defaults to the IP address of the machine you using to run the installer.
17. The **Scan Server HTTP Port** field automatically defaults to the recommended value of 8888, but may be changed as necessary.
18. Click **Next**. The **Email Configuration** panel appears
19. If you would like to receive email notifications for events such as project creation and scan completion, select the appropriate **True** option and specify your email settings.



Note ▪ Contact your network administrator to determine your SMTP settings.

20. Click **Next**. The **scan.properties** panel appears.
21. In the **Path to Compliance Library (CL) Directory** field, enter the location of your Compliance Library.

If your Compliance Library resides on another drive, you can manually type in the path or alternatively, press the **Choose** button, then enter the drive letter (such as D: \) and then browse to the directory.



Note ▪ Your Compliance Library directory is the directory that contains a `bdb` folder and pattern files that end in `.dat` (such as `C:\CodeInsight\PDL\CL\2.32`).

22. Click **Next**. The **scanEngine.properties** panel appears.
23. In the **Path to Workspace Directory** field, specify the location of your workspace directory, the directory in which your workspace data will be stored. The default and recommended location is one level above the <CODE_INSIGHT_ROOT_DIR>.



Tip ▪ To simplify future workspace migration, keep the workspace folder outside of your Code Insight installation directory.

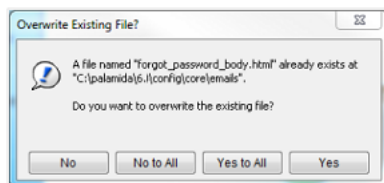
24. In the **Path to Scan Root Directory** field, enter the path to your scan root directory. The web application uses this property to limit the root of the server file system, making it easier to control which files and directories users have access to. Multiple roots are supported, but they must be separated by a pipe character (|):

C:\Folder1\Folder2|D:\Folder3



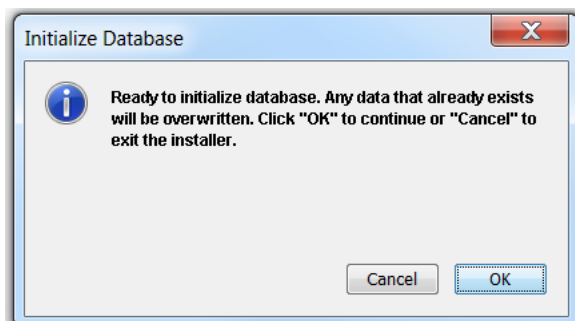
Note ▪ If you do not want to limit access of Code Insight users to files and directories, leave this field blank.

25. Click **Next**. The **Pre-Installation Summary** panel appears.
26. Review your Code Insight installation name and location.
27. Click the **Install** button when you are ready to proceed with the installation. The **Installing** panel appears.
28. If an **Overwrite Existing File?** dialog box appears, click the **Yes to All** button.



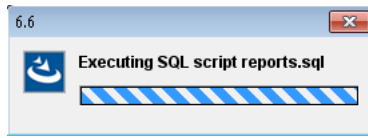
Note ▪ If you are installing into a non-empty directory, there may be files that must be overwritten. Click **Yes to All** to overwrite existing files.

The installer must execute a series of scripts and write to your database schema. The **Initialize Database** dialog appears, prompting you to specify whether you are ready to initialize the database.



29. Ensure that the database is on and that the server you are on has access to the database, and then click **OK**.

Script execution messages like the following will appear:



30. If you encounter a database error during the installation process, you will be notified. Correct the error and run the installer again or manually run the database scripts after the installation completes.



Note ▪ You may check the <CODE_INSIGHT_ROOT_DIR>/Logs directory for more information about each script that was executed and for any other errors that occur during the installation process.

When the installation is complete, the **Install Complete** panel appears.

31. Click the **Done** button. The installation is complete and you are now ready to launch the application.
32. Proceed with the steps in [Launching Code Insight](#).

Installing Code Insight Using the Linux Command Line

This section provides the procedure for installing Code Insight via a Linux command line.



Task

To install Code Insight via a Linux command line, do the following:

1. Ensure that you've met all the prerequisites in [Prerequisites](#).
2. Place the following items in the same directory, such as /home/codeinsight/installer:

Type	File
Installer Jar File	codeinsight_6.x.jar
Key	palamida.key
Database Driver	The JDBC driver that you downloaded based on your database type. See Installing the Database Driver for details.

3. Open the command prompt and enter the following to launch the installer:

```
java -jar codeinsight_6.14.x.jar
```

An **Introduction** message appears.

```
-----
Introduction
-----

InstallAnywhere will guide you through the installation of Palamida Enterprise
Edition 6.7.

If you have any questions or issues installing this product, please contact
support@palamida.com or call (415) 777 9400.

If installing this product on Linux do not do so as root. Doing so will
prevent the application from running correctly.

PRESS <ENTER> TO CONTINUE: █
```

4. Press **Enter** when you are ready to continue.
 - To accept the default value at any time, press **Enter**.
 - To exit the installation at any time, press **CTRL+C**. The **Choose Install Set** screen appears:

```
=====
Choose Install Set
=====

Please choose the Install Set to be installed by this installer.

->1- Standalone (both core and scanner on this computer)
   2 Core (configure this computer as the core server)
   3- Scanner (configure this computer as a scan server)

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
: █
```

5. Enter the number of one of the following options and press **Enter**.
 - **1 Standalone**—Configure this instance as both the Core Server and the Scan Server.
 - **2 Core**—Configure this instance as the Core Server.
 - **3 Scanner**—Configure this instance as the Scan Server.

The Core Server controls your Web UI and Detector Client. The Scan Server is where actual scanning is performed.



Note ▪ This guide addresses the Standalone install but it is also common to have a single instance of the Core Server and multiple Scan Servers.

The **Choose Install Folder** screen appears.

```
-----
Choose Install Folder
-----

Where would you like to install?

Default Install Folder: /home/palamida/6.7

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
: /home/palamida/6.7

INSTALL FOLDER IS: /home/palamida/6.7
IS THIS CORRECT? (Y/N): Y █
```

6. Enter the Code Insight installation directory.

- You can specify an existing directory or a new directory.
- If you choose to create a new directory, it will be physically created after you have gone through all the prompts for setting up the product.

For Linux, the recommended location is `/home/codeinsight/<version>/`. This directory is referred to as `<CODE_INSIGHT_ROOT_DIR>` throughout this document.

7. Press **Enter**. The **Total System Memory** screen appears.

```
=====
Total System Memory

1- 8 GB
2- 16 GB
3- 32 GB
4- 64 GB

ENTER THE NUMBER OF THE DESIRED CHOICE: 2
```

8. Select the total system memory (RAM) on this machine by entering a number between 1 and 4.

To find the RAM of your machine, type the following command in the command prompt: `free -g`. Then note the number in the **Total** column and round up or down to the nearest number.



Note - The heap space for the Java VM will automatically be set to 80% of the total RAM amount. This is the recommended amount for running Code Insight.

9. Click Enter. The **Choose Database** screen appears.

```
=====
Choose Database
-----

Note: ensure the database server is configured to accept connections from this
computer and an appropriate database schema and user account have been created.

1- Oracle
2- MySQL
3- SQLServer

ENTER THE NUMBER OF THE DESIRED CHOICE: 
```

10. Select your database vendor.

Code Insight offers support for **Oracle**, **MySQL** and **SQL Server**. The input fields will be different depending on which database type you choose.



Note - You must have an empty schema and user account configured with the correct privileges before selecting the Database vendor. See [Configuring a Database](#) for details.

If you get an error, make sure that your database driver is in the same directory as the Code Insight installer jar file before proceeding.

```
=====
Missing JDBC Driver
-----

A JDBC driver matching the string mysql-connector-java*-bin.jar was not found
```



Note ▪ The system will exit after this message. After you have placed the database driver in the directory, restart the installation.

11. Press **Enter**. A **Configure Database** panel specific to the selected database type appears.

```
=====
Configure Database
-----

Database IP Address (DEFAULT: 10.100.6.11):
Database Port (DEFAULT: 3306):
Database User Name (DEFAULT: User1): User1
Database User Password (DEFAULT: Password): Password
Database Schema Name (DEFAULT: ): Schema1
```

Figure 11-1: MySQL Example

```
=====
Configure Database
-----

Database IP Address (DEFAULT: 10.100.6.11):
Database Port (DEFAULT: 1521):
Database User Name (DEFAULT: ): User1
Database User Password (DEFAULT: ): Password1
Database SID (DEFAULT: orcl):
```

Figure 11-2: Oracle Example

```
=====
Configure Database
-----

Database Server IP Address (Default: 127.0.1.1): 10.100.6.48
Database User Name (Default: ): User1
Database User Password (Default: ): Password1
Database Schema Name (Default: ): Palamida_Schema
```

Figure 11-3: SQL Server Example

12. On the **Configure Database** screens, enter your database IP address. By default, this field points to the IP address of the machine which you are currently using.



Note ▪ The IP address defaults to the IP address of the server on which you are running the Installer. You may use *localhost* instead of IP if the database is on this machine.



Note ▪ The Database password is encrypted after the first launch of the application.

13. Press **Enter**. The **core.properties** screen appears.

```
core.properties

Scan Server Alias (Default: Scanner1): Scanner1

Core/Scan Server IP address (Default: 127.0.1.1): 10.100.6.185

Scan Server HTTP port (Default: 8888): 8888
```

14. Enter an alias for your Scan Server in the **Scan Server Alias** field. This is the alias you will see in the Code Insight Web UI when you configure and launch your scans.



Note ▪ The **Core/Scan Server IP address** automatically defaults to the IP Address of the machine you are using. The HTTP and RMI port default to the recommended ports used by the Code Insight application. If you have configured different ports, be sure to change the port numbers.

15. Press **Enter**. The **Email Configuration** screen appears.

```
Email Configuration
-----

Do you want the application to send email notifications?
    1- Yes
    ->2- No

ENTER THE NUMBER FOR YOUR CHOICE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT:: 1

=====

Email settings
-----

Sender's Email Address (DEFAULT: ): notifier@company.com

SMTP Host Name (DEFAULT: ): smtpx.hostname.net

SMTP User Name (DEFAULT: ): notifier@company.com

SMTP User Password (DEFAULT: ): password

SMTP Host Port (DEFAULT: 465): 465
```

16. To receive email notifications for events such as project creation and scan completion, enter **1** for Yes.



Note ▪ Contact your network administrator to determine your SMTP settings.

17. Press **Enter**. The **scan.properties** screen appears.

```
=====
scan.properties
-----

Path to Compliance Library (CL) Directory (Default: ): /home/palamida/PDL/CL/2.30/
```

18. In the **Path to Compliance Library (CL) Directory** field, enter the location of your Compliance Library.



Note ▪ Your Compliance Library directory is the directory that contains a *bdb* folder and pattern files that end in *.dat* (such as *C:\CodeInsight\PDL\CL\2.32*).

19. Press **Enter**. The **scanEngine.properties** screen appears.

```
=====
scanEngine.properties
-----

Path to Workspace Directory (Default: /home/palamida/6.6/../../workspaces):
Path to Scan Root Directory (Default: ):
```

20. In the **Path to Workspace Directory** field, specify the location of your workspace directory, the directory in which your workspace data will be stored. The default and recommended location is one level above the `<CODE_INSIGHT_ROOT_DIR>`.



Tip ▪ Keeping the workspace folder outside of your Code Insight installation directory is recommended in order to simplify future workspace migration.

21. In the **Path to Scan Root Directory** field, enter the path to your scan root directory. The web application uses this property to limit the root of the server file system, making it easier to control which files and directories users have access to.

Multiple roots are supported, but they must be separated by a pipe character (`|`):

`C:\Folder1\Folder2|D:\Folder3`



Note ▪ If you do not want to limit access of Code Insight users to files and directories, leave this field blank.

22. Press **Enter**. The **Pre-Installation Summary** screen appears.

```
-----
Pre-Installation Summary
-----

Please Review the Following Before Continuing:

Product Name:
    6.7

Install Folder:
    /home/palamida/6.7

PRESS <ENTER> TO CONTINUE: [ ]
```

23. Review your Code Insight installation name and location, and press Enter when you are ready to proceed with the Installation. The **Installing** screen appears.

```
-----
Installing...
-----

[=====|=====|=====|=====]
[-----|-----|-----|-----]
-----
```



Note ▪ If you are installing into a non-empty directory, there may be files that need to be overwritten and a message box will be displayed. Select option **1 - Yes to All** to overwrite the existing files.

24. The Installer executes a series of scripts and writes to your database schema. Wait while it completes the installation.
25. If a database error occurs during the installation process, a notification message is displayed:

```
-----
Error executing palamida_ddl.sql
-----

Please review the following errors. They may warrant investigation.

Unable to establish database connection.
```

26. Correct the error and run the installer again or manually run the database scripts after the installation completes.



Note ▪ You may check the <CODE_INSIGHT_ROOT_DIR>/Logs directory for more information about each script that was executed.

When the installation is complete, the **Install Complete** screen appears.

```
=====
Install Complete
=====

Congratulations. 6.6 has been successfully installed to:
/home/palamida/6.6

On initial startup of Palamida EE, the system will automatically schedule an
Electronic Update to run 2 minutes after the Tomcat server is started. If
it fails, verify your Electronic Update settings.

Refer to the Installation and System Administration Guide: Administration:
Running an Electronic Update section for more information.

PRESS <ENTER> TO CONTINUE: █
```

27. Press **Enter** to continue. You have completed the installation and are now ready to launch the application.

Proceed with the steps in [Launching Code Insight](#).

Launching Code Insight

This section explains how to launch Code Insight.

- [Starting and Stopping the Server](#)
- [Logging into Code Insight](#)
- [Running the Update Service](#)

Starting and Stopping the Server

To start and stop the server, perform the following steps.



Task

To start and stop the Code Insight server, do the following:

1. To start Tomcat, enter the following:
`<CODE_INSIGHT_ROOT_DIR>\6.14.x\tomcat\bin\startup.bat` (or `startup.sh` if running on Linux)

To stop Tomcat, enter the following:
`<CODE_INSIGHT_ROOT_DIR>\6.14.x\tomcat\bin\shutdown.bat` (or `shutdown.sh` if running on Linux)
2. Open a Code Insight-supported browser.
3. Navigate to the following URL:
`http://<core_server>:8888/palamida`

Replace `<core_server>` with the name or IP address of the Code Insight Core Server.

Logging into Code Insight

To log into Code Insight, log into the server.



Task **To log into Code Insight, do the following:**

1. Log in as an administrator, using the following credentials:
 - **User:** admin
 - **Password:** Password123

Running the Update Service

On initial startup of Code Insight, the system automatically schedules an electronic update to run 2 minutes after the Tomcat server is started. If it fails, verify your electronic update settings.

Installing the Jenkins Plugin

The Code Insight plugin for Jenkins lets you start an Code Insight scan directly from Jenkins after a successful Jenkins build. This section contains the following topics:

- [Jenkins Downloads](#)
- [Setting up the Jenkins Plugin](#)
- [Generating a JWT Token](#)



Note ▪ The Jenkins plugin requires that a project and workspace exist in Code Insight before you attempt to scan using the plugin.

Jenkins Downloads

Before you proceed, ensure that you have set up a team, project and workspace in the Code Insight web UI.

Activity	URL
Download Jenkins	http://jenkins-ci.org/
Installing Jenkins procedure	https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins
Starting and Accessing Jenkins procedure	https://wiki.jenkins-ci.org/display/JENKINS/Starting+and+Accessing+Jenkins

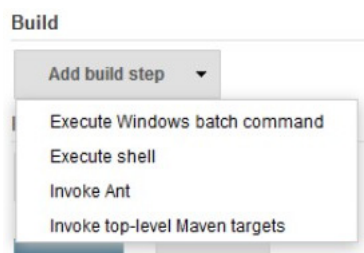
Setting up the Jenkins Plugin

To set up the Jenkins plugin to work with your build, perform the following steps.



Task To set up the Jenkins plugin, do the following:

1. Access your Jenkins server instance.
2. Navigate to **Manage Jenkins > Manage Plugins > Advanced tab > Upload Plugin**.
3. Browse to the palamida.hpi file and click **Upload**.
4. Restart the Jenkins server after installing **Code Insight Scan** plugin.
5. Create a new Jenkins project:
 - a. Click **New Item**.
 - b. Enter a name.
 - c. Select **Freestyle Project**.
 - d. Click **OK**.
6. Configure a project:
 - a. Select **Add build step** from the **Build** dropdown menu.



- b. Select **Code Insight Scan**. The **Code Insight Scan** area is displayed:

Server URL	<input type="text" value="http://localhost:8888/"/>
Project ID	<input type="text" value="6"/>
Token	<input type="text" value="eyJhbGciOiJIUzI1NiU9IjE5LjZkdWliOiJhbGV4IiwiaWF0IjoxNDUyNTUyNTg4fQ.m60UTXkt3yHBz-EoudqTB5kl0Y4wsmXCHBi2ZxePvoo"/>
Pass Jenkins git configuration	<input type="checkbox"/>

- c. Enter the following information into the **Code Insight Scan** plugin config section:
 - **Server URL**—Enter the URL of the Scan Server. Do not include /palamida in the URL.
 - **Project ID**—Enter a project ID. This needs to be created ahead of time, including workspaces.
 - **Token**—Enter the JWT token. It can be generated in the Code Insight Web UI and pasted into Jenkins. For details, see [Generating a JSON Web Token \(JWT\)](#).
 - **Pass Jenkins git configuration**—Optionally, select this option and provide Git configuration values. This can be used to perform a Git clone/sync via autorun script if desired.

- d. Click **Save**.
7. To create a build task and schedule a scan of all workspaces in the referenced project, click **Build Now** in the left navigation bar.

Generating a JWT Token

For the procedure to generate a JWT token, see [Generating a JSON Web Token \(JWT\)](#).

Using ScriptRunner

This chapter describes ScriptRunner:

- [ScriptRunner Options](#)
- [Generating a JSON Web Token \(JWT\)](#)
- [Using the JWT Token with scriptRunner](#)

ScriptRunner is required to run all Code Insight Java scripts—that is, those groovy scripts standard with Code Insight or custom groovy scripts. Its use is restricted to users with the Scripting Administrator role who have a valid JWT token. Users are prompted for a valid token when running ScriptRunner for the first time (token is not required for subsequent runs). The token is then stored on the client machine for future use. See [Generating a JSON Web Token \(JWT\)](#) for instructions on generating a token.

ScriptRunner Options

The following options are for scriptrunner.

- The `-p` option is no longer supported for specifying user password. You will be prompted for a password when you run scriptRunner for the first time. The `-u` flag is still applicable.
- The `-c` option should be used to pass the Core Server URL, replacing `<core_server_host>` with the name or IP address of the Core Server and replacing `<port>` with the port of the Core Server in the following example:
`-c http://<core_server_host>.com:<port>/Code Insight/`
- Add the port number to Workspace URI when using the `-w` option, replacing `<scan_server_host>` with the name or IP address of the Scan Server, `<port>` with the port number of the Scan Server and `<workspace_name>` with the name of the workspace in the following example:
`-w http://<scan_server_host>.com:<port>/<workspace_name>`

Generating a JSON Web Token (JWT)

Access to the Code Insight application from scriptRunner or any external application (REST API calls, Jenkins, etc.) requires a valid JWT Token for authentication and information exchange. JWT tokens are generated in the user interface and may be passed to scriptRunner or in the header of a request during a REST API call.



Task

To generate a JWT token, do the following:

1. Log into Code Insight as a *Scripting Administrator*.
2. Click **My Settings** in the upper right-hand corner. The **My Setting** page appears:

My Settings

*** First Name** System

Middle Name

*** Last Name** Administrator

*** Email** devnull@palamida.com

Job Title

Business Unit

Location

Telephone

Facsimile

*** Login Question** company

*** Login Answer** palamida

Password

Confirm Password

Your Roles
Participant, Policy Administrator, Requester, Reviewer, Scripting Administrator, System Administrator

Detector Heap Size 1024 MB

AUTH Tokens + Add Token

Name	Token	Create Date	Actions
------	-------	-------------	---------

Save Cancel

3. Click **Add Token**. The **Add Token** dialog appears:

Add Token

*** Name** Client Machine #1

*** Token Validity**

☐ Never Expires ☒ Expires On 01/27/2016

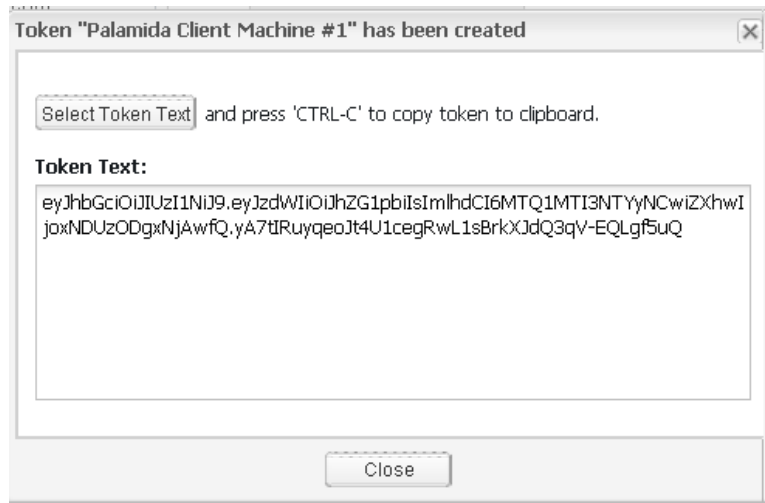
Save Cancel

4. Enter a token name and expiration date; and then click **Save**.



Note - If no date is specified, the default expiration of 1 month from today will apply.

5. Click **Select Token Text** or press **CTRL-C** to copy the token text to the clipboard.



6. Click **Close**. The token is now available for use with scriptRunner or REST APIs.

Using the JWT Token with scriptRunner

After you create the JWT token in Code Insight, you can use it with scriptRunner.



Task To use the token with scriptRunner, do the following:

1. Navigate to the scriptRunner/bin directory.

```
cd /<Code Insight_ROOT_DIR>/scriptRunner/bin/
```
2. Launch scriptRunner, using the `-c` flag to specify the Core Server URL and the `-u` flag to specify the authorized user.

```
./scriptRunner.sh -u admin -c http://localhost:8888/Code Insight/
```
3. When scriptRunner prompts you for the token, paste the token onto the command line and press **Enter**. The Groovy console will be launched if the token is valid. You may exit the console by typing **Exit**.

The token is stored in the scriptRunner.properties file for the user account on the client server. You will not be required to enter it again if you run scriptRunner from this server.

Exporting & Importing Workspaces

The Groovy scripts described in this section can be used to export and import audit data from one workspace to another. The scripts process all workspace data, including groups/inventory, associated files and file status, associated custom data, inventory checklist items, questions/answers, comments and inventory status. For more information about custom data, see [What is Custom Data?](#)

This section contains the following topics:

- [Usage Overview](#)
- [Installing the Scripts](#)
- [Using the Scripts with ScriptRunner](#)

Usage Overview

These scripts have been designed for use in the following scenarios:

- In-place backup and restoration of workspaces.
- Backup and restoration of workspaces across projects and across servers.
- Migrating workspaces from one Code Insight instance to another, including transitioning to newer versions of Code Insight.
- Copying inventory across projects, along with the inventory checklist items, questions/answers, comments and status.



Note ▪ The scripts are designed to work in conjunction with scanning. After export, a scan of the target workspace is required before import.

What is Custom Data?

Custom data refers to user-created licenses, components, and component versions. When custom data is exported, it is written to a custom data XML file (customData.xml) that is separate from an exported workspace XML file (workspaceData.xml). The following rules apply to the export and import of custom data:

- Custom data is exported any time you export a workspace which references custom data.
- If multiple workspaces are exported all at once, only one custom data XML file containing all the pieces of custom data referenced by all of the exported workspaces is generated.
- To export all of the custom data from the Core Server, use the `-export_all_custom_data` flag.
- Custom data can be imported along with a workspace which uses that custom data, or the custom data can be imported by itself.
- If you attempt to import a workspace into a database that does not contain custom data used by that workspace, the workspace import will fail.

What Workspace Data is Exported and Imported?

The following workspace data is exported and imported by the scripts:

- **Group data:** Name, component, version, license and group field values; custom groups.
- **File data:** File paths and file status.
- **Custom data:** Custom components, versions, licenses.
- **File indicators and tags:** Indicators, exact file matches, license matches, copyright holders, and so forth.
- **File reviewed status:** Reviewed status represented as a tag.



Note - To retain file indicators and tags (including the reviewed tag), ensure that you use the `-export_tags` option during export. You can verify that reviewed status is retained by searching the generated XML file for the term "reviewed".

What Inventory is Exported and Imported?

In addition to the workspace, the following content is exported and imported by the scripts:

- **Comments:** public and private comments and comment replies.
- **Checklist items:** checked and unchecked items.
- **Questions and Answers:** questions and replies.
- **Inventory Status:** the status (approved, rejected, pending review, ready for review).

About Backward Compatibility

The new Groovy scripts cannot be used for the export or import of workspaces from previous versions of Code Insight. To export from Code Insight 6.10.4, for example, you must use the older version of the export script and then import to the latest version (6.14.2 SP1) using the new import script.



Note ▪ To obtain copies of older product-version scripts, contact SCA support.

The following options are available for exporting to and importing from workspaces created in previous releases of Code Insight:

- Export from 6.10.3/6.10.4/6.11 using the scripts designed for those versions, and then import into Code Insight 6.14.2 SP1 using the 6.14.2 SP1 import script.



Note ▪ This process will not export questions and answers, comments, or inventory status.

- Update the 6.10.3/6.10.3/6.11 workspace to 6.14.2 SP1 by following the standard migration process. Use the Code Insight 6.14.2 SP1 export script to export from the workspace, including the questions and answers, comments, and inventory status.

Installing the Scripts

The export and import scripts can be found in the scriptRunner/scripts directory of your Code Insight installation. For example:

```
scriptRunner/scripts/exportWorkspaceData.groovy  
scriptRunner/scripts/importWorkspaceData.groovy
```

Once you locate the scripts, skip to the [Using the Scripts with ScriptRunner](#) for further instructions.

Installing Manually

If you received a zipfile (workspace_import_export.zip) containing the export and import scripts, you can unzip the file and place the scripts and classes in their proper locations.



Task

To install the scripts manually, do the following:

1. Unzip the workspace_import_export.zip file.
2. Place the driving scripts (exportWorkspaceData.groovy and importWorkspaceData.groovy) in the scriptRunner/scripts directory. These Groovy scripts (not classes) cannot function alone; they must be executed within the ScriptRunner framework.
3. Place the groovy_classes folder in the scriptRunner/lib folder. The files in the groovy_classes folder are Groovy classes (not scripts) that are called from the driving scripts.



Note ▪ These files should never be called directly from ScriptRunner as they cannot function alone.

Using the Scripts with ScriptRunner

These scripts will only function properly when executed through the ScriptRunner framework. For instance, in a stand-alone environment (both the Core Server and Scan Server running on the same instance) with IP address 111.122.133.144, invoke ScriptRunner in the following manner. Use the -u flag to specify the authorized user and the -c flag to specify the Core Server URL:

```
./scriptRunner.sh -u username_foo -c http://111.122.133.144:8888/palamida/
```

If this is your first time running scriptRunner, you will be prompted to enter a JWT token. Paste the token in for authorization. For information about generating a JWT token, see [Generating a JWT Token](#).



Note ▪ The flags used with ScriptRunner are not passed through to the export and import scripts. The -c flag is only used by ScriptRunner. You must still pass into the export script the server name from which to export a workspace.

To pass the server name from which to export a workspace to the export script use the following:

```
exportWorkspaceData.groovy -server http://111.122.133.144:8888/palamida/
```

Otherwise the export script will use its hard-coded server name default value (i.e., localhost). In the above case, the entire command would be:

```
./scriptRunner.sh -u username_foo -c http://111.122.133.144:8888/palamida/./scripts/  
exportWorkspaceData.groovy -server http://111.122.133.144:8888/palamida/
```



Note ▪ Ensure that the URL passed to the -c flag always contains a trailing “/”.

Using the Export Script

The exportWorkspaceData.groovy script uses the ScriptRunner framework to export workspaces from one or more servers.

Export Options

For the current list of export script options, refer to the output from the script's -h flag.

File and Path Options

-h or -help

Prints a help page of all the available options.

-output <file>

The name of the xml file to export the workspaces to, without extension. The default is workspaceData. If a single workspace is exported, the result will be an XML file (with this name) containing that one workspace.

If multiple workspaces are being exported, the result will be a ZIP file (with this name) containing all the exported workspace XML files, the custom data XML file (if necessary), and a log file indicating the significant events of the export.

-output_path <directory>

The directory into which the export files will be written. This flag defaults to the directory from which the script is executed. For instance, if the script is executed from scriptRunner/bin, you will find a workspaceData.xml or workspaceData.zip file inside the scriptRunner/bin directory.

-custom_data_file <file>

The name of the file that will contain the exported custom data (if any). No suffix is required. The default is customData. For instance, by default you will find a customData.xml file inside the scriptRunner/bin directory.

Server Options

-server <urlOfCoreServer>

Export from the Core Server with the name 111.122.133.144 (default: localhost). For instance,

```
exportWorkspaceData.groovy -server http://111.122.133.144:8888/palamida/
```

-scan_server <urlOfScanServer>

Export from the Scan Server with the name 111.122.133.144 (default: value of -server flag). For instance,

```
exportWorkspaceData.groovy -scan_server http://111.122.133.144:8888/palamida/
```

-server_all <urlOfScanServer>

Export all workspaces from the Scan Server with the name 111.122.133.144. This option overrides -scan_server, -team, -project, and -workspace options. For instance,

```
exportWorkspaceData.groovy -scan_all http://111.122.133.144:8888/palamida/
```

-server_all_from_config <urlOfScanServer>

Read Scan Server instances from config on the specified Scan Server. Export all workspaces from all Scan Server instances. This option overrides -server_all, -server, -scan_server, -team, -project, and -workspace options, as in the example:

```
exportWorkspaceData.groovy -server_all_from_config http://111.122.133.144:8888/palamida/
```

What to Export

-team <teamName>

Export all workspaces in all projects belonging to this team. This option overrides -workspace. If -team and -project are used together, this will export all workspaces for that project and should be used when different projects have the same name, but were created by different teams.

-project <projectName>

Export all workspaces in this project. This option overrides -workspace. If -team and -project are used together, this will export all workspaces for that project and should be used when different projects have the same name, but were created by different teams. This option will also cause the -pa (process active only) flag to be ignored.

-workspace <workspaceName>

Export only the workspace with this name. This option causes the -pa (process active only) flag to be ignored.

-exclude_file_extensions <extensions>

When exporting files, skip all files that have the suffixes listed here. You must specify a comma-separated list without spaces.

Boolean Options

-pa

If present, process only active ('In Progress') projects.

-pe

If present, do not process empty groups.

-pm

If present, export metadata for the groups being exported.

-ps

If present, do not process system groups.

-export_all_custom_data

If present, export all custom data, not just custom data referenced by exported groups. If this is not set, only export custom data that is referenced by the workspaces being exported.

-export_tags

If present, the tags for the group files in the exported workspaces are exported. This includes the “reviewed” file status tag.

Other Options

-retries <number>

The number of times to retry exporting of a failed workspace (integer). Most often, an export failure is due to the workspace being scanned.

-wait <seconds>

The number of seconds to wait before retrying the exporting of a failed workspace (integer).

Combining Export Flags

The following scenarios occur when export flags are combined in the export script:

- Option `-server_all_from_config` will cause the script to ignore the following flags: `-server_all`, `-server`, `-scan_server`, `-team`, `-project`, and `-workspace`.
- Option `-server_all` causes the script to ignore the following flags: `-scan_server`, `-team`, `-project`, and `-workspace`.
- Option `-team` causes the script to ignore the `-workspace` flag.
- Option `-project` causes the script to ignore the `-workspace` flag.
- Options `-team` and `-project` together will export all workspaces for that project and should be used when different projects have the same name, but were created by different teams.
- Options `-workspace` or `-project` causes the `-pa` (process active only) flag to be ignored.
- If `-export_all_custom_data` is not set, export only custom data referenced by the workspaces being exported.

Export Usage

In a stand-alone environment, the Core Server is the Scan Server, so there is no need to use the `-scan_server` flag.

In a clustered environment, export can be run from any instance that can execute ScriptRunner (Core Server, Scan Server, Detector, Client) by using one or more of the Server Options flags indicate the servers from which workspaces should be exported. For more information, see [Server Options](#).

Workspaces can be designated for export using a variety of flags (for example, `-workspace` to export one workspace, `-server_all` to export all workspaces on that server). However, if the export script does not understand which workspaces to export, it will prompt you to choose one workspace from the indicated Scan Server.

Export Output

The export script displays onscreen status messages as it runs, starting with a list of the flags (and their values) it will use for this execution.

All output (XML files, ZIP file, log file, etc) will be written to the location specified by the `-output_path` flag (if present) or to the directory from which the script was executed (if the `-output_path` flag was not specified).

If one workspace is exported, the output is: one XML file containing the workspace's information, one log file (export.log), and (if there is custom data for that workspace) one XML file containing that custom data.

If multiple workspaces are exported, the output is one ZIP file containing the following:

- One XML file for each exported workspace,
- One log file (export.log), and (if there is custom data for any exported workspace)
- One XML file containing all custom data for the exported workspaces.

In general, only data not generated by a scan are exported. Because an exported XML file must be imported into a workspace of scanned files, exporting scan-generated information would be redundant. For example, only files that are in groups are exported to the XML file. The other files are skipped because importing them would not add any information that was not acquired from the scan itself.

Export Usage Examples

The export script has many options. The following are some of the common usages:

1. Export the workspace named foo from stand-alone server (some_server_name) to the file named /home/FNCI/exportedWorkspace.xml:

```
exportWorkspaceData.groovy -server some_server_name -workspace foo -output exportedWorkspace -  
output_path /home/FNCI/
```

2. Export the workspace named foo from Core Server (some_core_server_name) and Scan Server (some_scan_server_name) to the file named /home/FNCI/exportedWorkspace.xml:

```
exportWorkspaceData.groovy -server some_server_name -scan_server some_scan_server_name -workspace  
foo -output exportedWorkspace -output_path /home/FNCI/
```

3. Export all workspaces from the Scan Server some_scan_server_name to a set of XML files which will be compressed into a file named /home/FNCI/workspaceDataFromLocalhost.zip. The Core Server (localhost) still has to be defined:

```
exportWorkspaceData.groovy -server localhost -server_all some_scan_server_name -output  
workspaceDataFromLocalhost -output_path /home/FNCI/
```

4. Export all of the custom data on the stand-alone server localhost to an XML file named foo.xml, but do not export any workspaces:

```
exportWorkspaceData.groovy -custom_data_file foo -export_all_custom_data
```

Using the Import Script

The Import script can be run from any machine that can execute ScriptRunner (Core Server, Scan Server, Detector, client). Use the -server flag to set the value of the Core Server. In a clustered environment, you must also set -scan_server to the Scan Server containing the workspace you just scanned.

Custom data (user-created licenses, components, and component versions) can be imported along with a workspace that uses that custom data, or the custom data can be imported by itself. For more information, see [What is Custom Data?](#)



Task

To use the import script, do the following:

1. Create a new workspace on the Core Server onto which you want to perform the import. This will be the 'target workspace' that will accept the audit data. This workspace must contain all the files that were present in the exported workspace.



Note ▪ In a clustered environment, both the Core Server and the target Scan Server must be designated (-server and -scan_server flags).

2. Execute a scan of the workspace files. Data that is not created by this scan will be copied over during import.
3. Open the XML file containing the workspace data to be imported. If the file paths in the XML file do not match the file paths in the workspace you just scanned, the file information from the XML file will not be imported. You can do either one of the following:
 - Note that values needed for the -adv_file_comparison and -adv_file_comparison_depth import flags.
 - Edit the export XML file to change the file paths to exactly match the file locations in the target workspace.
4. Run the import script. The following are descriptions of available flags and examples:
 - Required options for importing a workspace:
 - input (-f)
Indicates the XML file that contains the workspace information to be imported.
 - workspace (-w)
Import data into the workspace with this name.
 - If the XML file containing the workspace information references custom data (that is, user-created licenses, components, or component versions), also indicate which XML file contains that custom data using the -custom_data_file (-x) flag.
5. Open Detector for the target workspace and verify that the audit information was imported correctly.

Import Options

For a current list of these options, see the output from the script's -h flag.

Import Options that Accept Parameters

-input <file> (-f)

The file containing the workspaces to be imported.

-custom_data_file <file> (-x)

Import custom data from this file.

-server <hostnameOrIP> (-c)

Import the workspace onto this Core Server (default: localhost).

-scan_server <hostnameOrIP> (-s)

Import the workspace onto this Scan Server (default: value of -server flag).

-workspace <workspaceName> (-w)

Import data into the workspace with this name.

-reformat_xml_file <file> (-z)

Reformat this XML file and do nothing else. If this option is set, that XML file will be reformatted and the script will exit without importing anything.

-adv_file_comparison <option> (-a)

Determines whether to match files using only a portion of each file's path. The default is *never*. This statement accepts the following values: *never*, *if_no_absolute_match*, *always*. If this flag is not set, a file element from the imported XML file must match the full file path and name in the workspace.

-adv_file_comparison_depth (-d)

This option is ignored if -adv_file_comparison is not set. This option specifies how much of a file's path/name/MD5 to compare (default: md5_file_name).

Import Options that Require an MD5 Match

md5_only:

To find matches between file elements from the imported XML file and files in the workspace, only compare the MD5 hashes.

md5_file_name

For each file element and each file in the workspace, compare the MD5 hash and the file's name.

md5_file_name_dir_depth_1

Compare the MD5 hash, the file's name, and each file's parent directory. For example: /a/b/c/foo.gif will match /d/e/c/foo.gif, but not /d/e/f/foo.gif (assuming the MD5 hashes also match).

md5_file_name_dir_depth_2

Compare the MD5 hash, the file's name, each file's parent directory, and each file's parent directory's parent directory.

md5_file_name_dirdepth#

And so on, to an arbitrary depth.

Import Options that Do Not Require an MD5 match

no_md5_file_name

For each file element and each file in the workspace, compare only the file's name.

no_md5_file_name_dirdepth#

Compare the file's name and each file's parent directory to the indicated depth, as shown above.

-include_tags (-i)

Import only these tags. Ignore all other tags. This option is not compatible with `-exclude_tags (-e)`. The option accepts a comma-delimited list of tags and converts the % character into a space, allowing its use with tags that have spaces in their names.

-exclude_tags (-e)

Do not import these tags. Import all other tags. This option is not compatible with `-include_tags (-i)`. The option accepts a comma-delimited list of tags and converts the % character into a space, allowing its use with tags that have spaces in their names.

-path_search_replace_csv (-p)

Location of the CSV file with the path fragments to find and replace. For more information, see [CSV File](#).

Import Options that Accept Boolean Options

All of these options default to *false*, including any of these flag sets that flag to *true*.

-dryrun (-y)

If present, simulate an import without actually saving anything. Generates a log file containing what would be the significant events encountered during an actual import.

-check_md5_hash (-m)

If present, in addition to checking the full file path, also check each file's MD5 hash. If there is a match, *do not* import the file. This option is ignored if advanced file matching (see above flags) is used.

-update_existing_group_data (-u)

If present, update existing group data with the content from the XML workspace file.

-annotate_adv_search_results (-r)

If present, create a new metadata tag for each file matched with advanced file matching (see above flags).

CSV File

The `-path_search_replace_csv (-p)` flag takes one value: the location of a CSV file that maps the file paths from the workspace XML, indicated by `-input (-f)`, to the paths on the Scan Server, indicated by `-scan_server (-s)`.

The CSV file has the following format:

```
/Path/To/Find/In/XML/File/One,Path/To/Replace/It/With/On/Scan/Server/1  
/Path/To/Find/In/XML/File/Two,Path/To/Replace/It/With/On/Scan/Server/2  
/Path/To/Find/In/XML/File/Three,Path/To/Replace/It/With/On/Scan/Server/3
```

Paths will be found and replaced exactly as shown in the CSV file. No regular expressions are allowed.

The path to be found will be matched only to the beginning of each path in the XML file being imported, which limits the possibility of accidentally changing parts of other paths. The portion of a path that matches is the only portion that is replaced.

Every path to be found will be tried on every path in the XML file being imported.

Matches are attempted in the order listed in the CSV file. If a match is found, the matching portion of the path is replaced before the next match is attempted. For each path, this allows for repeated changes as the search/replace map is iterated over.

Paths may contain any combination of forward slashes and backslashes. Backslashes must be escaped with other backslashes.

Any path that contains at least one backslash must be wrapped in double-quotes. For example, if you export from a Windows machine and import onto a Linux machine, your CSV file might look like the following:

```
"C:\\Path\\To\\Find\\In\\XML\\File\\One",Path/To/Replace/It/With/On/Scan/Server/1  
"C:\\Path\\To\\Find\\In\\XML\\File\\Two",Path/To/Replace/It/With/On/Scan/Server/2  
"C:\\Path\\To\\Find\\In\\XML\\File\\Three",Path/To/Replace/It/With/On/Scan/Server/3
```

In addition, two operators are available: `CONVERT_TO_FORWARD_SLASH` and `CONVERT_TO_BACKSLASH`. Place either in the first field of a line. The second value in that line is ignored. The conversion is performed on each forward slash and backslash in the entire file path.

Import Results

The import script display onscreen status messages as it runs, starting with a list of the flags and their values to be used for this execution.

Log file (import.log)

Located in the directory from which the workspace or custom data files were imported, it contains the following:

- All significant actions taken during the import.
- All issues encountered during the import.
- All files that were not associated with any group.

Metadata Tags

When a problem is encountered during the import of a group, a metadata tag (display name: Import Notes, field name: import-notes) is created for that group detailing the issue. If such a metadata tag already exists for that group, the new issue is added to the existing tag.

If the `-annotate_adv_search_results` flag has been included on the command line and a file is matched using advanced file matching, a metadata tag (display name: File Path Matched by Advanced Logic, field name: file-path-matched-by-advanced-logic, tag value: Yes) is created for that file.

Import Usage Examples

The following are some examples of the use of the Import script:

1. Import the workspace data contained in `workspaceData.xml` into the workspace `foo`. This will not overwrite existing audit group data:

```
importWorkspaceData.groovy --input /home/FNCI/workspaceData.xml --workspace foo
```

2. Import the workspace data contained in `workspaceData.xml` into the workspace `foo` in a clustered environment with Core Server (`some_core_server`) and Scan Server (`some_scan_server`), where the scan of workspace `foo` was run on `some_scan_server`. This will not overwrite any existing audit group data:

```
importWorkspaceData.groovy --server some_core_server --scan_server some_scan_server --input /home/FNCI/workspaceData.xml --workspace foo
```

3. Import the workspace data contained in `workspaceData.xml` into the workspace `foo` and overwrite any existing audit group data with audit group data from that XML file:

```
importWorkspaceData.groovy --input /home/FNCI/workspaceData.xml --workspace foo --update_existing_group_data
```

4. Import the workspace data contained in `workspaceData.xml` into the workspace `foo` along with the custom data from `custom.xml`. If `workspaceData.xml` references any custom data from `custom.xml`, you must import `custom.xml` before (or at the same time) as `workspaceData.xml`:

```
importWorkspaceData.groovy --custom_data_file /home/FNCI/custom.xml --input /home/FNCI/workspaceData.xml --workspace foo
```

5. Import only the custom data from file `custom.xml`:

```
importWorkspaceData.groovy --custom_data_file /home/FNCI/custom.xml
```

6. Perform a practice import (dryrun) of custom data from the file `custom.xml`. The import script will walk through all the import steps, but will not write any data to the database:

```
importWorkspaceData.groovy --custom_data_file /home/FNCI/custom.xml --dryrun
```


Integrating with LDAP for Authentication (Optional)

This section provides details about modifying the LDAP configuration file and data synchronization:

- [Configuring LDAP Integration](#)
- [Data Synchronization](#)
- [LDAP Configuration](#)

Configuring LDAP Integration

The Lightweight Directory Access Protocol (LDAP) configuration file (`core.ldap.properties`) is located in the `<Code Insight_ROOT_DIR>\<version>\config\core` directory. Code Insight imports LDAP user metadata into its database. User passwords are never stored into our system. LDAP user authentication is handled on a real-time basis against the LDAP server for every login. If the user does not exist in LDAP, authentication is performed against the Code Insight database.

Imported LDAP users are given a default requester role in the system. The users that are added during the LDAP sync process are not allowed to use the Forgot Password or Modify Preferences features of the application. The Application administrator can set a role for a user if it is different than that of a participant.

The system can maintain user information for users that do not exist in LDAP. An example of this scenario is an external contractor who needs access to LDAP but does not exist in LDAP. In such a case, the user information is managed via Code Insight directly.

- You define LDAP user import frequency by modifying the `ldap` configuration file. The default setting in the properties file is `ldap.jobFrequency=0 0 6 * * ?`. This means that every morning at 6 AM the system syncs up with the LDAP server.
- Refer to <http://www.quartz-scheduler.org/documentation/quartz-2.x/tutorials/crontrigger> for further information about the cron expression format. LDAP frequency example syntax is as follows:

`ldap.jobFrequency=0 0 6 * * ?` (every morning 6 AM) `ldap.jobFrequency=0 0/1 * * * ?` (every minute)
- During the initial startup, the application replaces the plain-text LDAP password with an encrypted value.

- The application reads the plain text password and writes the encrypted password back into the core.ldap.properties file in the following format: <ENCRYPTED_PASSWORD>ENCRYPTED. For subsequent server re-starts, the application uses the encrypted password.
- To change the password, the administrator needs to bring down the server, then, you can enter the new plain-text password into the core.ldap.properties file, and restart the server. During the server startup, the application will again read the plain-text password and write the encrypted password back into core.db.properties.
- If your LDAP server supports anonymous access, and you prefer to connect to the LDAP server anonymously rather than via a password-protected account, you can enable this by specifying ldap.anonymous=true in the core.ldap.properties file.

The configurable properties from the LDAP configuration files are shown below. Ensure that you DO NOT comment out any attribute mappings. If you do not want to map an attribute, set it equal to blank.

```
# LDAP server connection settings, url, userName and password
# this user should have read right to be able to access the schema

# Do you want to sync users from LDAP ?
ldap.user.sync.enabled = false

#To enable LDAP Authentication set ldap.enabled to true
ldap.enabled = false

# To enable anonymous access to read LDAP directory turn this property on,
# this will allow you to not specify ldap.userName and ldap.password properties.
ldap.anonymous=false

#URL of the LDAP server, for eg. ldap://<ldap_server>:389
ldap.url = ldap://10.100.1.27:389

#Base node of LDAP server, for all the searches base node will be automatically appended.
ldap.base = dc=adtest,dc=palamida,dc=com

#User name to login to LDAP server
ldap.userName = cn=administrator,CN=users,DC=adtest,DC=palamida,DC=com

#Password to login to LDAP server
ldap.password = palamida123

#searchBase and searchFilter are used to import users to Palamida system.
#DO NOT append ldap.base to ldap.searchBase

# Search base where you can see all the desired users.
ldap.searchBase = CN=Users

# Search filter to pull only desired users to the Palamida System, you can use LDAP Query here.
ldap.searchFilter =
(&(objectClass=person)(memberOf=CN=PalamidaAppsecGroup,CN=Users,DC=adtest,DC=palamida,DC=com))

# LDAP user login filter, sAMAccountName={0} (for Active directory)
ldap.loginFilter = sAMAccountName={0}

# Turn it on if LDAP server has paging enabled, mostly for Active Directory
ldap.serverPaging = true
```

```
#Turn it on to sync LDAP users inside subtree directory
ldap.search.subtree = true

# page size if using paging
ldap.page.size = 1000

#LDAP user login attribute, loginAttr and loginFilter must always be in sync
ldap.user.loginAttr = sAMAccountName

#Uniquely identifiable attribute for each user, if none found loginAttr will be used.
ldap.user.externalIdAttr = sAMAccountName

#LDAP user email attribute, it should always have a valid email address value
ldap.user.emailAttr = mail

# Do NOT comment out unwanted attribute mappings, set them equal to blank to not assign a value
#LDAP user firstName attribute (Optional)
ldap.user.firstNameAttr = title
#LDAP user middleName attribute (Optional)
ldap.user.middleNameAttr = title
#LDAP user lastName attribute (Optional)
ldap.user.lastNameAttr = title
#LDAP user business unit attribute (Optional)
ldap.user.businessUnitAttr = title
#LDAP user job title attribute (Optional)
ldap.user.jobTitleAttr = title
#LDAP user location attribute (Optional)
ldap.user.locationAttr = title
#LDAP user telephone attribute (Optional)
ldap.user.telephoneAttr = title
#LDAP user fax attribute (Optional)
ldap.user.faxAttr = title
#LDAP user state attribute (Optional)
ldap.user.stateAttr = title

# Use Cron frequency syntax
# Refer to http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html for examples
# runs every 4 hours
ldap.jobFrequency=0 0 */4 * * ?
# runs every morning at 6 AM
ldap.jobFrequency=0 0 6 * * ?
# runs every Monday at 6 AM
# ldap.jobFrequency=0 0 6 ? * MON

# Default roles to be assigned to all the ldap users in Palamida system.
# Comma separated list should be provided below with possible values of requester, reviewer,
participant.
ldap.user.role =

# Associations of LDAP queries to User Lists
# LDAP Query for user list consists of 2 parameters (<userlist_name>.ldap.description &
<userlist_name>.ldap.query).
# These 2 parameters need to be configured for each user list that is associated with an LDAP query.
# The LDAP query description will be shown in the Web UI in place of the actual query.
# The LDAP query will be executed each time an LDAP sync occurs.
```

```
#<userlist_name>.ldap.description = <enter description of LDAP query to be shown on user list details  
page in Web UI>  
#<userlist_name>.ldap.query = <enter LDAP query to be executed for this user list each tyme an LDAP sync  
occurs>
```

Configurable LDAP Properties

You can use the LDAP instead of the Code Insight user name and password information to authenticate users. The groups, projects, and IP policies, however, are still managed via the administration functions in Code Insight. The authentication is a three-step process:

- The users enter their user names and passwords on the Code Insight login page.
- Code Insight calls into LDAP to authenticate the users and obtain user information that is passed back to Code Insight.
- If a user is not found in LDAP, the user is authenticated against the Code Insight database. A scenario in which this might occur is when contractors who have access to Code Insight do not exist in the customer's LDAP/AD system.

Data Synchronization

The following sections describe LDAP data synchronization:

- [User Metadata](#)
- [Disabled Users](#)
- [LDAP Synchronization to Multiple Sources](#)

User Metadata

The metadata for each user (name, email, role, etc.) 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 Code Insight database. User data for those users that do not exist in LDAP is not affected by this process.

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.

LDAP Synchronization to Multiple Sources

Code Insight supports LDAP synchronization for authentication and for authorization to two or more sources. Users may be authenticated against one LDAP directory and authorized with roles and permissions from a secondary source.

**Task**

To configure a secondary LDAP server for authorization, do as follows:

1. Add and define the following properties in `core.ldap.properties` (<FNCI_ROOT_DIRECTORY>/config/core/core.ldap.properties):

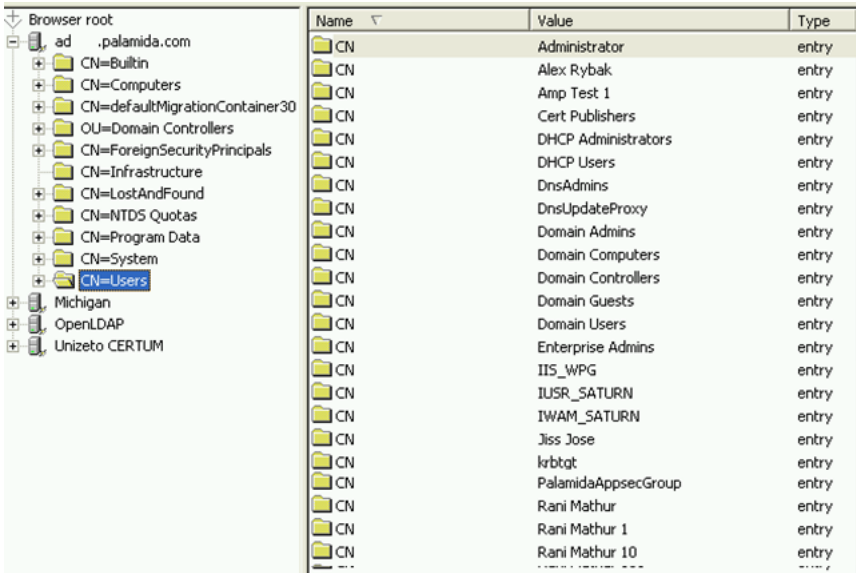
`ldap2.url`
`ldap2.base`
`ldap2.userName`
`ldap2.password`
`ldap2.anonymous`
`ldap2.read.timeout`
`ldap2.searchBase`
`ldap2.searchFilter`
2. Restart the server.
3. Note the following property behavior:
 - If an `ldap2.name` property is not specified or specified with no value, its `ldap.name` counterpart applies. For example, if “`ldap2.read.timeout =` ”, the `ldap.read.timeout` value is applied during the sync.
 - If either `ldap2.searchBase` or `ldap2.searchFilter` is specified with a value, both apply.
 - If one of these properties (`ldap2.searchBase` or `ldap2.searchFilter`) is specified, the value for the other will be an empty string. For example: “`ldap2.searchFilter = (objectClass=*)`” implies “`ldap2.searchBase =` ”
 - A sync also requires appropriate `ldap.user.name` properties

LDAP Configuration

This section contains information pertaining to configuration of LDAP search filters, user lists, server paging and SSL.

Set Up a User Search Filter

Pulling only required users into Code Insight, configuring `searchBase` and `searchFilter` entries properly is important. `searchBase` is typically the root node under which you can find all the desired users. `searchFilter` allows for an LDAP query based on user attributes. We recommend creating a Code Insight system-specific group and making all of the desired users part of this group. Then, use this group in search Filter, as show below.



Name	Value	Type
CN	Administrator	entry
CN	Alex Rybak	entry
CN	Amp Test 1	entry
CN	Cert Publishers	entry
CN	DHCP Administrators	entry
CN	DHCP Users	entry
CN	DnsAdmins	entry
CN	DnsUpdateProxy	entry
CN	Domain Admins	entry
CN	Domain Computers	entry
CN	Domain Controllers	entry
CN	Domain Guests	entry
CN	Domain Users	entry
CN	Enterprise Admins	entry
CN	IIS_WPG	entry
CN	IUSR_SATURN	entry
CN	IWAM_SATURN	entry
CN	Jiss Jose	entry
CN	krbtgt	entry
CN	PalamidaAppsecGroup	entry
CN	Rani Mathur	entry
CN	Rani Mathur 1	entry
CN	Rani Mathur 10	entry

Figure 16-1: Sample LDAP Tree

Default Role Assignment

The option to set user roles when pulling LDAP users into Code Insight is available via the `ldap.user.role` property in `core.ldap.properties`. The available roles are requester, reviewer and participant. All other roles must be set manually in the Web UI.



Task

To set the user roles, do the following:

1. Modify the `core.ldap.properties` file to include the roles in a comma separated list. For example, the following entry will assign all LDAP users the role of requester, reviewer and participant.

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

If no default role assignment is set, all LDAP users will be assigned the role of requester only.

2. Restart the server for this property to take effect.



Note - User roles can only be added via LDAP sync; they cannot be removed using this property. To remove a user role the Code Insight Administrator must make the change in the Web UI. See the “Administration Menu: User Options” section of the Code Insight User Guide for more information.

Set Up a User List Sync

Code Insight provides the option to manage user lists dynamically via LDAP so that user list data is updated each time an LDAP sync takes place. For example, if a user is part of the Developer user list and is removed from LDAP, that user will automatically be removed from the Developer user list next time an LDAP sync takes place.



Task

To set up a user list sync, do the following:

1. Select **Users** from the **Administration** menu.
2. Select the **User Lists** tab and click **Add New User List**.
3. Enter a name for the user list. LDAP user list names must not contain any spaces. If spaces are necessary, they must to be escaped in the property file according to instructions below.
4. Select the user list type. User list type must match the type of users that will be added to the list. For example, a list that contains users with the role of Requester should be of type Requester.
5. Click **Save**.



Note ▪ Do not add users via the web UI because they will be added during the LDAP sync.

6. Modify the `core.ldap.properties` file to include the user list name, a description of the user list that will be visible in the Web UI and an LDAP Query to associate to the user list. For example:

```
Developers_List.ldap.description = The description of the LDAP query in words
Developers_List.ldap.query = (&(objectclass=user)(objectcategory=person))
```

7. If the user list name contains spaces, they must be escaped as such:

```
Developers\ List.ldap.description
Developers\ List.ldap.query
```

8. Restart the Code Insight server. The query populates the **Developers** user list with all users that match the specified query:

User List Details

Name	Developers	The userlist_name												
Description	Developers who will be requesting to use open-source software.													
User List Type	Requester													
LDAP Query	The description of the LDAP query in words	Value of Developers.ldap.description												
Users	<div>Search: <input type="text"/></div> <table border="1"> <thead> <tr> <th>login</th> <th>First Name ▲</th> <th>Last Name</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>rmathur1000</td> <td></td> <td></td> <td>rmathur@palamida.com</td> </tr> <tr> <td>rmathur1001</td> <td></td> <td></td> <td>rmathur@palamida.com</td> </tr> </tbody> </table>		login	First Name ▲	Last Name	Email	rmathur1000			rmathur@palamida.com	rmathur1001			rmathur@palamida.com
login	First Name ▲	Last Name	Email											
rmathur1000			rmathur@palamida.com											
rmathur1001			rmathur@palamida.com											

Results of query in Developers.ldap.query

Server Paging

LDAP and Active Directory support server paging controls the number of records the system is pulling at any given time. Set `ldap.serverPaging` to true to allow Code Insight to enable paging.



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: `ldap.serverPaging=false`.

LDAP over SSL

SSL provides data encryption security for user information passed over the network. You must use ldaps://URL with 636 port, which is the default dedicated port for SSL.

Troubleshooting and Testing LDAP Configurations

You can use an LDAP Browser such as Apache Directory Studio to verify that settings are configured correctly.

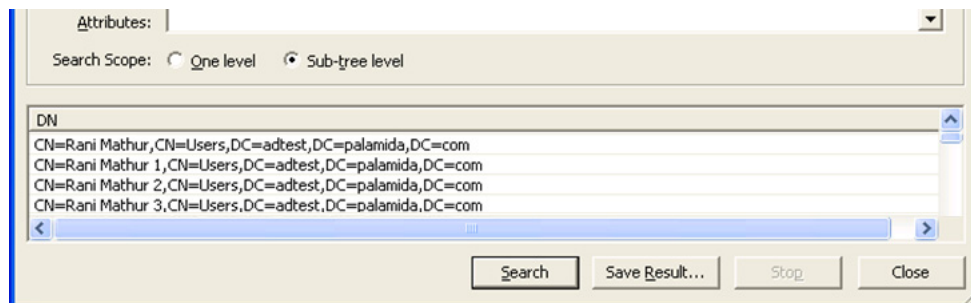
Q: What if I can't connect to the LDAP server using Code Insight?

A: Verify your LDAP configuration using LDAP Browser or the Code Insight LDAP test groovy script located in the `<Code Insight_ROOT_DIR>\<version>\scriptRunner\scripts` directory. Contact [Reverera Support](#) for instructions on executing the script or for help on connecting.

Q: How do I set up searchBase and searchFilter?

A: Using the LDAP Browser, do as follows:

1. Click **Search**.
2. Enter [ldap.searchBase, ldap.base] in the **Search DN** field.
3. Enter ldap.searchFilter in the **Filter** field.
4. Set the **Search** scope to Sub-tree level.
5. Click **Search**.
6. Build or modify your searchBase using this tool (searchBase may contain LDAP queries).



Configuring Code Insight for Single Sign-On (Optional)

This chapter provides the following instructions for configuring Code Insight for using Single Sign-On (SSO):

- [Overview](#)
- [Prerequisite Tasks for Configuring Code Insight for SSO](#)
- [Phase 1: Generate Service Provider Metadata](#)
- [Phase 2: Obtain the Identity Provider Metadata File](#)
- [Phase 3: Configure the Environment Properties File](#)
- [Phase 4: Login Using SSO Credentials](#)
- [Reference: Troubleshooting and Debugging SSO](#)
- [Reference: Enabling Secure HTTP over SSL](#)

Overview

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 IdP) is any SSO service—such as Okta, Ping Federate, and others—offering SAML authentication services. The Service Provider (also called SP) is an application, such as Code Insight, that is configured to participate in the SSO service. When an SP user logs in using credentials for an SSO session, a SAML message is sent to the IdP, requesting user authentication. If the user password is valid, the IdP returns a SAML message, stating that the user is logged in at the IdP. The user, in turn, is logged into the SP.

The Code Insight administrator can use the instructions in the remaining sections in this chapter to configure Code Insight as a SP in an SSO session.

This configuration is performed on the Code Insight Core Server.

Prerequisite Tasks for Configuring Code Insight for SSO

Before configuring Code Insight for SSO, ensure that the following tasks have been performed:

- [Configure HTTPS on the Code Insight Server](#)
- [Set Up SSO Users](#)
- [Obtain a Keystore](#)

Configure HTTPS on the Code Insight Server

The HTTPS communication protocol must be used to exchange SAML messages between the Service Provider (SP) and Identity Provider (IdP). For instructions on configuring HTTPS on the instance where the Code Insight Core Server resides (and on each instance where a remote Scan Server resides), see the [Reference: Enabling Secure HTTP over SSL](#) section.

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 the [Integrating with LDAP for Authentication \(Optional\)](#).

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 IdP first, using the IdP's instructions. Then synchronize the users from the LDAP server to Code Insight. See the [Integrating with LDAP for Authentication \(Optional\)](#).

Without LDAP

If you do not use LDAP, you must manually create the SSO users both in Code Insight (see "Creating New Users" in the *Code Insight User Guide*) and at the IdP site, ensuring that the user information is the same in both locations.

Ensure that the user's login is the user's email address.

Obtain a Keystore

A keystore is required to configure Code Insight as an SP. Any of these keystores are supported:

- The default keystore shipped with Code Insight
- The same keystore used to configure HTTPS for the Code Insight Core Server

- Another keystore created with a signed certificate

The method you choose to configure SSO determines which of these keystores you can use. See the next section [Phase 1: Generate Service Provider Metadata](#) for details.

Phase 1: Generate Service Provider Metadata

The first phase in configuring Code Insight for SSO involves generating your site's Service Provider (SP) metadata. This metadata is required to register your site with the Identity Provider (IdP). Code Insight offers two methods for creating SP metadata:

- **Default method**—Provides secure SSO communications. This method, which uses the default keystore shipped with Code Insight, is a notably easy process, requiring minimal input from the user. See [Default Method for Generating Service Provider Metadata](#) for details.
- **Custom method**—Ensures that additional encryption and signing are used to authenticate communication between the SP and the IdP. This method requires either the keystore used to configure HTTPS for the Code Insight Core Server or another keystore. Substantial user input is required to configure the SP metadata and related components for the keystore. See [Custom Method for Generating Service Provider Metadata](#) for details.

Default Method for Generating Service Provider Metadata

The following process describes how to use the default method to generate the SP metadata for SSO configuration. This method, which uses the default keystore shipped with Code Insight, is an easy way to generate SP metadata that ensures secure SSO communications.

If you require additional encryption or signing for SSO, generate your SP metadata using the custom method, as described in [Custom Method for Generating Service Provider Metadata](#).



Task

To use the default method for generating SP metadata, do the following:

1. From the appropriate archive in your Code Insight Core Server installation, extract the these files to your own empty directory:
 - On Windows, extract the following files from `fnciInstallPath\docs\SSO.zip`:
 - `palamida_metadata.bat`
 - `palamida_replace.vbs`
 - `sp_metadata_template.xml`
 - On Linux, extract the following files from `fnciInstallPath/docs/SSO.zip`:
 - `palamida_metadata.sh`
 - `sp_metadata_template.xml`

2. Run the appropriate command to generate the `SPMetadata.xml` file containing the SP metadata.

- On Windows:

```
palamida_metadata.bat "entity_id" "server_url" SPMetadata.xml
```

- On Linux:

```
bash ./palamida_metadata.sh entity_id server_url SPMetadata.xml
```

The following describes the variables used in the command and provides command examples.

- **entity_id**—The unique identifier for your Code Insight Core Server as an SP in the format `<w>:<x>:<y>:<z>`. This is usually specified by the Identity Provider but is not mandated by SSO.
- **server_url**—The HTTPS URL handling the SP's user sign-in requests. This is usually the URL for your Code Insight Core Server in the following format: `HTTPS://myhost.mycompany.com:port`. The *port* value should match the port used in the HTTPS configuration for the Core Server. (The default and recommended port value for HTTPS is 8443.)

Example commands:

```
palamida_metadata.bat "ww:xx:yy:zz" "https://myhost.companyA.com:8443" SPMetadata.xml
```

```
bash ./palamida_metadata.sh ww:xx:yy:zz https://myhost.companyA.com:8443 SPMetadata.xml
```

3. Copy resulting `SPMetadata.xml` file to `fnciInstallPath/config/core/security` in the Core Server installation.

Custom Method for Generating Service Provider Metadata

This process describes how to generate SP metadata using a secure keystore, ensuring that a signed certificate is used to authenticate all communication between the SP and IdP. (The keystore can be the same one used to configure HTTPS for the Code Insight Core Server or another keystore created with a signed certificate.) This custom method requires the Spring Security SAML Extension web application to generate the SP metadata according to your input. The following topics describe each step in this process:

- [Step 1: Download and Configure the Spring Security SAML Extension](#)
- [Step 2: Generate the SP Metadata](#)
- [Step 3: Configure the SSO Common Properties File](#)

If you do not require additional encryption or signing for SSO, you can generate your SP metadata using the default keystore for SSO, as described in [Default Method for Generating Service Provider Metadata](#).

Step 1: Download and Configure the Spring Security SAML Extension

This procedure prepares the Spring Security SAML application to generate the SP metadata.



Task

To download and configure the Spring Security SAML Extension:

1. Use these instructions to download and install the SAML extension:
 - a. Download the spring-security-saml-1.0.4.RELEASE-dist.zip file from the following location:
<https://repo.spring.io/list/release/org/springframework/security/extensions/spring-security-saml/1.0.4.RELEASE/>
 - b. Extract the contents of the .zip file to c:/samlapp.
 - c. In a command line, change to the directory c:/samlapp/spring-security-saml-1.0.4.RELEASE/sample.
 - d. In the sample directory, build the web application with maven, using the command `mvn package`.
 - e. Copy the spring-security-saml2-sample.war file from the C:/samlapp/spring-security-saml-1.0.4.RELEASE/sample/target directory to the tomcat/webapps directory in your Code Insight Core Server installation (`fnciInstallPath`).

If necessary, refer to the following link for more a more detailed description of the SAML extension installation and configuration:

<https://docs.spring.io/autorepo/docs/spring-security-saml/1.0.x-SNAPSHOT/reference/pdf/spring-security-saml-reference.pdf>

2. Copy your secure keystore to the following location in your Core Server installation:

`fnciInstallPath/tomcat/webapps/spring-security-saml2-sample/WEB-INF/classes/security`

3. Open the following file in your Core Server installation:

`fnciInstallPath/tomcat/webapps/spring-security-saml2-sample/WEB-INF/SecurityContext.xml`

4. Locate the keyManager bean definition within the file. It looks similar to this:

```
<bean id="keyManager" class="org.springframework.security.saml.key.JKSKeyManager">
  <constructor-arg value="classpath:security/myKeystore.jks"/>
  <constructor-arg type="java.lang.String" value="myKeystorePassword"/>
  <constructor-arg>
    <map>
      <entry key="myAlias" value="myAliasPassword"/>
    </map>
  </constructor-arg>
  <constructor-arg type="java.lang.String" value="myAlias"/>
</bean>
```

5. In the keyManager bean definition, replace the following values as needed with the properties defined for your secure keystore:

- **myKeystore**—The name of the keystore that you are using for SSO.



Important • Ensure that the keystore is copied to WEB-INF/classes/security folder of the sample web application.

- **myKeystorePassword**—The password for the keystore.
- **myAlias**—The alias defined for the private key contained in the keystore.

- **myAliasPassword**—The password for the private key alias.

Step 2: Generate the SP Metadata

The following procedure generates the SP metadata using the Spring Security SAML application.



Task To generate the SP metadata using Spring Security SAML:

1. Start the Spring Security SAML Extension web application by running the following command:

```
fnciInstallPath/tomcat/webapps/spring-security-saml2-sample
```
2. Once the application is started, navigate to **Metadata Administration | Login | Generate new service provider metadata**.
3. In the **Metadata Generate Filter** section, provide the following values (or values appropriate to your site):

Field	Value
Store for the current session	Select No .
Entity ID	<p>Provide the identifier for the Code Insight Core Server as an SP in the format <code><w>:<x>:<y>:<z></code>, as in the example:</p> <pre>palamida:cust:test:server1</pre> <p>This ID must be unique among the other entity IDs. It is usually specified by the Identity Provider but is not mandated by SSO.</p>
Entity base URL	<p>Provide the HTTPS URL handling the SP user's sign-in requests. This is usually the URL for the Core Server in <code>HTTPS://myhost.mycompany.com:port</code> format, where <i>port</i> is the default port for the Core Server. (For default Code Insight ports, see Network and Firewall Considerations.)</p>
Entity alias	Enter defaultAlias .
Signing key	Enter the password for the private key alias. (This value should be the same as the myAliasPassword value entered in the keyManager bean definition described in Step 1: Download and Configure the Spring Security SAML Extension .)
Encryption key	Enter the alias defined for the private key contained in the keystore. (This value should be the same as the myAlias value entered in the keyManager bean definition described in Step 1: Download and Configure the Spring Security SAML Extension .)
Signature security profile	Select MetalOP .
SSL/TLS security profile	Select PKIX .

Field	Value
SSL/TLS hostname verification	Select Standard hostname verifier .
SSL/TLS client authentication	Select None .
Sign metadata	Select Yes .
Signing algorithm	Enter http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 .
Sign sent AuthNRequests	Select Yes .
Require signed authentication Assertion	Select No .
Require signed LogoutRequest	Select No .
Require signed LogoutResponse	Select No .
Require signed ArtifactResolve	Select No .
Single sign-on bindings	Select SSO HTTP-POST as the default. (Uncheck SSO Artifact .)
Supported NameIDs	Select Transient , E-Mail , and X509 Subject .
Enable IPD Discovery profile	Select No .

4. Generate the metadata.
5. Save the contents of the **Metadata** text box to `SPMetadata.xml`, and copy this file to `fnciInstallPath/config/core/security` in the Code Insight Core Server installation.
6. Save the contents of the **Configuration** text box to `Extended.xml` in a temporary location of your choice for later reference. (You will need this file when updating the `core.sso.properties` in [Step 3: Configure the SSO Common Properties File](#).)
7. (Optional) If you want signing but not encryption in SSO communications, open the file `SPMetadata.xml` file, locate the encryption tag (usually the second tag), and remove everything between the tag and `</md:KeyDescriptor>`, including the encryption tag and `</md:KeyDescriptor>`.

Step 3: Configure the SSO Common Properties File

This step configures the `core.sso.properties` file to enable SSO for Code Insight.



Task

To configure the SSO common properties file:

1. In a text editor, open the `fnciInstallPath/config/core/core.sso.properties` file in the Code Insight Core Server installation. The following shows the file contents:

```
## this file contains all sso placeholder values.
## this file contains all the sso configuration placeholder values.
#
#saml.keystore=file:///c:/<path>/<keystore.jks>
#saml.keystore.password=<keystore_password>
#saml.keystore.alias=<keystore_alias>
#saml.keystore.alias.password=<keystore_alias_password>

# only change the values that are different for your extended metadata
#saml.metadata.local=true
#saml.metadata.alias=defaultAlias
#saml.metadata.idpDiscoveryEnabled=false
#saml.metadata.idpDiscoveryURL=null
#saml.metadata.idpDiscoveryResponseURL=null
#saml.metadata.ecpEnabled=false
#saml.metadata.securityProfile=metaiop
#saml.metadata.sslSecurityProfile=pkix
#saml.metadata.sslHostnameVerification=default
#saml.metadata.signingKey=null
#saml.metadata.signingAlgorithm=null
#saml.metadata.signMetadata=false
#saml.metadata.encryptionKey=null
#saml.metadata.tlsKey=apollo
#saml.metadata.requireLogoutRequestSigned=false
#saml.metadata.requireLogoutResponseSigned=false
#saml.metadata.requireArtifactResolveSigned=false
#saml.metadata.supportUnsolicitedResponse=true
#saml.verifySignature=false
```

2. Uncomment and update the following properties required for Service Provider security and identification. Ensure that these values for these properties match the corresponding values that you provided for the `keyManager` bean definition in `SecurityContext.xml`, as described in [Step 1: Download and Configure the Spring Security SAML Extension](#):

SSO Property	Description
saml.keystore	Enter the path and name of the keystore that you created for SSO. (Ensure that the keystore resides in this location.)
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.verifySignature	Set to <code>false</code> .

3. Open the `Extended.xml` file you saved when you also generated the `SPMetadata.xml` file in [Step 2: Generate the SP Metadata](#).

4. For each `saml.metadata.property` in the `extendedMetadata` configuration section of the `core.sso.properties` file that differs from its corresponding value in the `Extended.xml` file, do the following:
 - a. Uncomment the `saml.metadata.property`.
 - b. Update its value to match the one in `Extended.xml`.

For example, the `Extended.xml` file might contain the following property:

```
<property value="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" name="signingAlgorithm"/
```

If the corresponding value in the `core.sso.properties` file is different, replace its value with the one in `Extended.xml`:

```
saml.metadata.signingAlgorithm=http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
```

5. Save the `core.sso.properties` file.

Phase 2: Obtain the Identity Provider Metadata File

The next phase in SSO setup is to obtain the Identity Provider (IdP) metadata file, using the process required by the IdP.



Task

To obtain the Identity Provider metadata:

1. Follow the IdP's instructions for obtaining the IdP metadata.

For example, the IdP might require Code Insight to send them the `SPMetadata.xml` file (generated in [Phase 1: Generate Service Provider Metadata](#)) to generate the metadata.

Alternatively, you might be required to generate the IdP metadata file using the IdP user interface. In this situation, you will need to provide access to the `SPMetadata.xml` file and specify the following single-sign-on URL for Code Insight (also specified in the `SPMetadata.xml`):

```
https://myhost.mycompany.com:8443/palamida/saml/SSO
```

2. Once you obtain the IdP metadata, save it as `IDPMetadata.xml` in `fnciInstallPath/config/core/security` in the Core Server installation.

Note that the IdP should use `X509SubjectName` as the `NameID` format.

If you are configuring SSO with AD FS (Active Directory Federation Services) as the Identity Provider (IdP), refer to the Spring Security SAML Extension documentation (see the following link) for instructions to create a relying-party trust on AD FS. Refer specifically to the content that deals with integration with IdPs.

<https://docs.spring.io/autorepo/docs/spring-security-saml/1.0.x-SNAPSHOT/reference/pdf/spring-security-saml-reference.pdf>

Phase 3: Configure the Environment Properties File

This creates and configures the `env.properties` file to enable SSO for Code Insight.



Task

To create and configure the SSO common properties file, do the following:

1. Create the text file `fnciInstallPath/config/core/env.properties` in the Code Insight Core Server installation.
2. Add the following line and save the file:

```
spring.profiles.active=sso
```

Phase 4: Login Using SSO Credentials

Once you complete the steps described previously, Code Insight users defined as SSO users should be able to log into an SSO session managed by the Identity Provider (IdP) and obtain access to Code Insight.

Ensure that the `SPMetadata.xml` and `IDPMetadata.xml` files are located in `fnciInstallPath/config/core/security` in the Code Insight Core Server installation.

About User Authentication

If SSO login for a user fails, an attempt is made to authenticate the user from the Code Insight database and, if that fails, through LDAP.

To disable database authentication, do not define users in the database. (This implies that no row where `EXTERNAL_ID_` is null exists in the `PAS_USER` table.) To disable LDAP authentication, set `ldap.enabled` to **false** in `core.ldap.properties`.

Disabling SSO

Use this step to disable SSO if necessary.



Task

To disable SSO, do the following:

Rename or remove the `fnciInstallPath/config/core/env.properties` file in the Code Insight Core Server installation.

Reference: Troubleshooting and Debugging SSO

If you have configured Code Insight to use SSO but are having issues with SSO, consider these basic troubleshooting suggestions:

- Make sure the Identity Provider (IdP) metadata is current. If you need to re-obtain the IdP metadata, follow the instructions in [Phase 2: Obtain the Identity Provider Metadata File](#). Ensure that the metadata is stored as `IDPMetadata.xml` in `fnciInstallPath/config/core/security` in the Code Insight Core Server installation.

- Ensure that times on both the IdP system and the Code Insight Core Server are the same. The system clock for the IdP system and for the Core Server should both use a time server.
- Test your SSO process using SAML SSOCircle as the IdP.

Reference: Enabling Secure HTTP over SSL

Use the instructions in these sections to enable an HTTPS connection for each instance where a Code Insight server—the Core Server or a remote Scan Server—resides:

- [Enabling an HTTPS Connection](#)
- [Obtaining and Storing an SSL Certificate for HTTPS Enablement](#)

For more information about HTTPS, refer to the following sites:

- http://en.wikipedia.org/wiki/HTTP_Secure
- <http://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html>

Enabling an HTTPS Connection

The following topics describe how to enable an HTTPS connection for a Code Insight server:

- [HTTPS Enablement Required for Each Code Insight Server](#)
- [Enabling HTTPS for a Code Insight Server](#)

HTTPS Enablement Required for Each Code Insight Server

Follow these instructions on the instance where the Code Insight Core Server is installed, and then repeat them on each instance where a remote Scan Server resides. This process requires a purchased SSL certificate or a self-signed certificate, as well as a keystore, for each server.

However, if the Scan Server and Core Server are installed on the same instance, use these instructions once to set up single HTTPS connection for both servers on the instance. They will share the same connection (hence, the same SSL certificate and the same keystore).

Enabling HTTPS for a Code Insight Server

Use this procedure to enable an HTTPS connection for Code Insight on each instance where a Code Insight server resides.



Task

To enable an HTTPS connection on the instance where a Code Insight server resides, do the following:

1. Purchase a Secure Site SSL certificate, or generate your own self-signed SSL certificate. For instructions, see [Obtaining and Storing an SSL Certificate for HTTPS Enablement](#).
2. Ensure that the keystore in which the certificate has been stored is copied to the `fnciInstallPath/tomcat` directory.

3. Open the `fnciInstallPath/tomcat/bin/catalina.bat` file (or the `catalina.sh` file, depending on your operating system).

4. Uncomment the following property if necessary, set it to `true`, and save the file:

```
set -Dpalamida.ssl=true
```

5. Back up the `fnciInstallPath/tomcat/conf/server.xml` file to another directory (outside of the `conf` directory), and then copy `server.xml` from `fnciInstallPath/tomcat/https` to `fnciInstallPath/tomcat/conf`.

The `server.xml` file contains a default configuration that references the keystore at `fnciInstallPath/tomcat/palamida.jks`. You will need to update information as it pertains to your keystore in step 5.

6. 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" />
```

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

Palamida SSL: Edit this section to match your certificate information.

This section shows the default values for the certificate.

```
<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="palamida.jks"
  keystorePass="palamida"
  keyAlias="palamida"
  keyPass="palamida" []
  clientAuth="false"
  sslProtocol="TLS" sslEnabledProtocols="TLSv1.2"
  ciphers="TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WIT
H_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WIT
H_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA"
/>
```

8. Update the following parameters in this section to reflect your keystore and certificate information (as described in [Obtaining and Storing an SSL Certificate for HTTPS Enablement](#)).

If the **keystorePass** and **keyAliasPass** values are the same, you can specify just the **keystorePass** value.

```
.....
keystoreFile="name of the keystore that you created"
keystorePass: "password for the keystore"
keyAlias: "alias for the private key (certificate entry) in the keystore"
keyPass: "password for the private key in the keystore" (needed only if different from keystore
password)
```



Note ▪ For security purposes, do not change the default value "TLSV1.2" 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.

9. Restart the Tomcat server after making changes to the server.xml file to enable HTTPS. For more information on starting and stopping the Tomcat server, see [Administration: Starting & Stopping Servers](#).

Obtaining and Storing an SSL Certificate for HTTPS Enablement

Enabling an HTTPS connections requires an SSL certificate stored in a keystore. The SSL certificate can be either one that is purchased from a Certificate Authority or a self-signed SSL certificate. The following procedures describe how you create a keystore, obtain a certificate, and store it in the keystore.

- [Purchasing and Importing a Secure Site SSL Certificate](#)
- [Generating a Self-Signed SSL Certificate](#)

The Code Insight Core Server and each remote Scan Server requires its own certificate and keystore. See [HTTPS Enablement Required for Each Code Insight Server](#) for more information.

Purchasing and Importing a Secure Site SSL Certificate

Use the following steps for purchasing an SSL certificate and importing it into a keystore that you create:

- [Creating the Keystore](#)
- [Purchasing the SSL Certificate](#)
- [Importing the Certificate](#)

Creating the Keystore

Use this procedure to create the keystore into which you will import the certificate.



Task

To create the keystore into which you will import the SSL certificate, do the following:

1. From DOS or Linux, enter the following command:

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

Provide the following information in the command:

- **myKey**—The alias that you want to create for the private key (the certificate entry) that will be included in the keystore.
- **myKeystore**—The file name for the keystore you are creating. Use the .jks extension.

2. When prompted, enter a password for the keystore.

When prompted with “What is your first and last name?”, enter the fully qualified hostname for the server, such as `myserver.mycompany.com`.

3. When prompted for the key password, simply press Enter to use the same password as the keystore password.

Purchasing the 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>

The next two sections provide basic instructions for generating a certificate-signing request (CSR) to purchase a certificate from a Certificate Authority and for importing the certificate into the keystore. However, you should follow any special instructions from the Certificate Authority for performing these two processes.

Generating the Certification Request (CSR)

The following provides a procedure for generating a certification request (CSR) into the file `myKEY.csr` to send to the Certificate Authority. Consult the Certificate Authority for any specific instructions.



Task To generate the CSR, do the following:

1. From DOS or Linux, enter the following command to create the CSR file:

```
keytool -certreq -keyalg RSA -alias myKey -file myKey.csr -keystore myKeystore.jks
```

where you provide the following (as defined when you created the keystore in [Creating the Keystore](#)):

- **myKey**—The alias that you created for the private key (the certificate entry) that will be imported in the keystore.
 - **myKeystore**—The file name for the keystore that you created. Provide the path if you are not running the command from the directory in which the keystore resides.
2. Request a 2048-bit certificate from the Certificate Authority, sending the CSR file.

The Certificate Authority returns the file in a p7b file.

Importing the Certificate

The following is a procedure for importing the SSL certificate into the keystore that you created in [Creating the Keystore](#).



Task To import the SSL certificate into the keystore, do the following:

1. From DOS or Linux, enter the following command to import the SSL certificate into the keystore:

```
keytool -import -trustcacerts -alias myKey -file myCertificate.p7b -keystore myKeystore.jks
```

Provide the following parameters in the command:

- **myKey**—The alias that you created for the private key (the certificate entry) that you are importing into the keystore (created in [Purchasing and Importing a Secure Site SSL Certificate](#)).
 - **myCertificate**—The file name (including the .p7b extension) for the certificate you received from the Certificate Authority. Provide the path if the you are not running the command from the directory in which the certificate resides.
 - **myKeystore**—The file name for the keystore that you created in [Purchasing and Importing a Secure Site SSL Certificate](#). Provide the path if the you are not running the command from the directory in which the keystore resides.
2. If additional certificates are required, import them using the following command:
- ```
keytool -import -trustcacerts -file certificateFile -keystore myKeystore.jks
```
- where you provide the following for each certificate:
- **certificateFile**—The file name (including the .p7b extension) for the certificate you received from the Certificate Authority. Provide the path if the you are not running the command from the directory in which the certificate resides.
  - **myKeystore**—The file name for the keystore that you created in [Purchasing and Importing a Secure Site SSL Certificate](#). Provide the path if the you are not running the command from the directory in which the keystore resides.
3. Copy the keystore to the *fnciInstallPath*/tomcat directory. (The keystore is configured in the *server.xml* file, as described in [Enabling an HTTPS Connection](#).)

## Generating a Self-Signed SSL Certificate

The following procedure creates both a self-signed SSL certificate (and its private key) and the keystore in which to store the certificate. Consult the Certificate Authority for any specific instructions.



### Task

**To create a self-signed SSL certificate and store it in a keystore, do the following:**

1. From a command line, enter the following command:

```
keytool -genkey -keyalg RSA -sigalg SHA256withRSA -alias myKey -keypass password -keystore
myKeystore.jks -storepass password -validity 3600 -keysize 2048 -ext
san=ip:ipAddress,dns:domainName...
```

where you provide the following details specified to create the keystore that will store the certificate:

- **myKey**—An alias for the private key you are creating.
- **myKeystore**—A file name for the keystore you are creating.
- **password**—A password used for both the private key and keystore. Enter this value for both -keypass and -storepass.

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

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

2. When prompted with “What is your first and last name?”, enter the fully qualified hostname for the server, such as **myserver.mycompany.com**.
3. Copy the keystore to the *fnciInstallPath*/tomcat directory. (It is configured in the *server.xml* file, as described in [Enabling an HTTPS Connection](#).)

# Configuring Additional Scan Servers (Optional)

This section provides details about the optional configuring of additional Scan Servers:

- [Overview](#)
- [Configuration Details](#)

## Overview

The follow scenarios require you to configure additional Code Insight Scan Servers:

- The codebases are too large to scan on a single server. You can use additional Code Insight Scan Servers to distribute the load.
- The system experiences a bottle-neck by a single scan queue, and using additional Code Insight Scan Servers enables you to perform parallel scans.

## Configuration Details

Configuring additional Code Insight Scan Servers is similar to configuring the first Code Insight Scan Server with a few exceptions. For each additional Scan Server, do as follows:

1. Register each additional Code Insight Scan Server on the Core Server.
2. In `core.properties` (`<Code Insight_ROOT_DIR>\<version>\config\core\`) on the Core Server, add an alias for the new Code Insight Scan Server and define the associated properties. Separate multiple aliases with commas.

```
scan.server.aliases = <ALIAS1>, <ALIAS2>, <ALIAS3>
```

```
scan.server.<ALIAS>.web = http://<SCAN_SERVER_IP_ADDRESS1>:<HTTP_PORT>/Code InsightScanEngine
```

```
scan.server.<ALIAS2>.web = http://<SCAN_SERVER_IP_ADDRESS2>:<HTTP_PORT>/Code InsightScanEngine
```

```
scan.server.<ALIAS3>.web = http://<SCAN_SERVER_IP_ADDRESS3>:<HTTP_PORT>/Code InsightScanEngine
```

3. Modify the configuration files on each additional Code Insight Scan Server. The configuration changes required to configure additional Code Insight Scan Servers are as follows:
  - a. The `core.db.properties` (`<Code_Insight_ROOT_DIR>\<version>\config\core\`) file defines the database connection details for use by the Code Insight Scan Server. Copy this file from the Code Insight Core Server. This is the only file required in the core directory on each remote Code Insight Scan Server.
  - b. Update `scanEngine.properties` (`<Code_Insight_ROOT_DIR>\<version>\config\scanEngine\`) with your Scan Server settings:
    - a. Set the `coreServerUrl` to point to your Code Insight Core Server URL.
    - b. Set `serverUrl` to point to the URL of the Code Insight Scan Server you are configuring.
    - c. Set the `scanServerName` to the alias name of the server you are configuring from `core.properties` on the Core Server, denoted as `<ALIAS>` in Step 2, that registers this instance of the Code Insight Scan Server.
4. Copy the Code Insight Key to each Code Insight Scan Server. Contact Revenera if you need to obtain a key.



---

**Note** ▪ *Code Insight Analyzer requires a new key with data services enabled.*

5. Copy the Code Insight Data libraries to each additional Code Insight Scan Server. See **Installing the Compliance Library** for additional information.
6. Copy Tomcat to each additional Code Insight Scan Server. See **Extracting Application Files** and **Configuring the Tomcat Web Server** for additional information.
  - a. Only the `CodeInsightScanEngine` directory (exploded war file) is required in the Tomcat web applications directory (`<Code_Insight_ROOT_DIR>\<version>\tomcat\webapps`).
  - b. Remove the Code Insight directory before starting Tomcat.
  - c. Ensure that the Tomcat cache is flushed. You can do this by removing all contents from the `<Code_Insight_ROOT_DIR>\<version>\tomcat\work\Catalina\localhost\` directory.
  - d. Copy the database driver to Tomcat. See **Installing the Database Driver**.

For assistance if you need to generate custom source code fingerprints on multiple servers, contact **Revenera Support**.


# Configuring Code Insight using MySQL Commands

The following table describes common MySQL commands and how to use them to configure Code Insight.

**Table 19-1** • Common SQL Commands

| Command                                     | Code                                                                                                                                                                                                                                                                                             |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Create Database</b>                      | <pre>CREATE DATABASE &lt;the database name&gt;;</pre> <p>For example:</p> <pre>CREATE DATABASE Code Insight;</pre> <p>Make sure the user has permissions to create a new database.</p>                                                                                                           |
| <b>Create User and Grant All Privileges</b> | <p>Access from a specific server:</p> <pre>CREATE USER 'user'@'server' IDENTIFIED BY 'password';</pre> <p>Access from any server:</p> <pre>CREATE USER 'user'@'%' IDENTIFIED BY 'password';</pre> <pre>GRANT ALL PRIVILEGES ON *.* TO 'user'@'server' WITH GRANT OPTION; FLUSH PRIVILEGES;</pre> |
| <b>Default User and Password</b>            | <p>On Linux, the default user for most Linux MySQL installs is root the password is blank.</p> <p>On Windows, Windows prompts for user name and password during the setup process.</p>                                                                                                           |
| <b>Execute the SQL Scripts</b>              | <pre>mysql -h &lt;the host&gt; -u &lt;the user&gt; -p &lt;the database&gt; &lt; &lt;path to script file&gt;</pre> <p>You can use this command to execute the Code Insight SQL scripts from command line.</p>                                                                                     |
| <b>List All Database for MySQL Instance</b> | <pre>SHOW DATABASES;</pre> <p>This will list all databases. After running the create database command, you can use this to verify it was successfully created.</p>                                                                                                                               |

**Table 19-1** ▪ Common SQL Commands

| Command        | Code                                                                                                                                                                                                                                                                                                                                                      |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Log into MySQL | <pre>mysql -h &lt;the host&gt; -u &lt;the user&gt; -p &lt;the database&gt;</pre> <div></div> <div><b>Note</b> ▪ The <i>-p</i> option prompts for a password when you hit Enter; the statement that follows <i>-p</i> is the name of the database not the password.</div> |

# Configuring 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 environment. Repeat the appropriate procedure on each instance on which you have installed a Code Insight server (Core or Scan Server):

- [Configuring Code Insight as a Windows Service](#)
- [Configure Code Insight as a Linux Service](#)

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

## Configuring Code Insight as a Windows Service

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

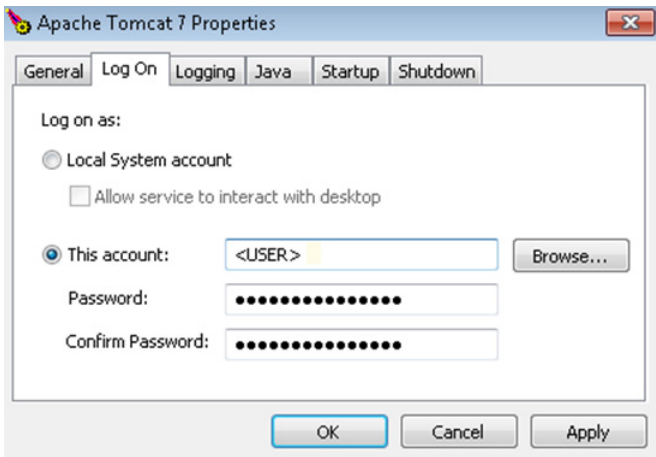


---

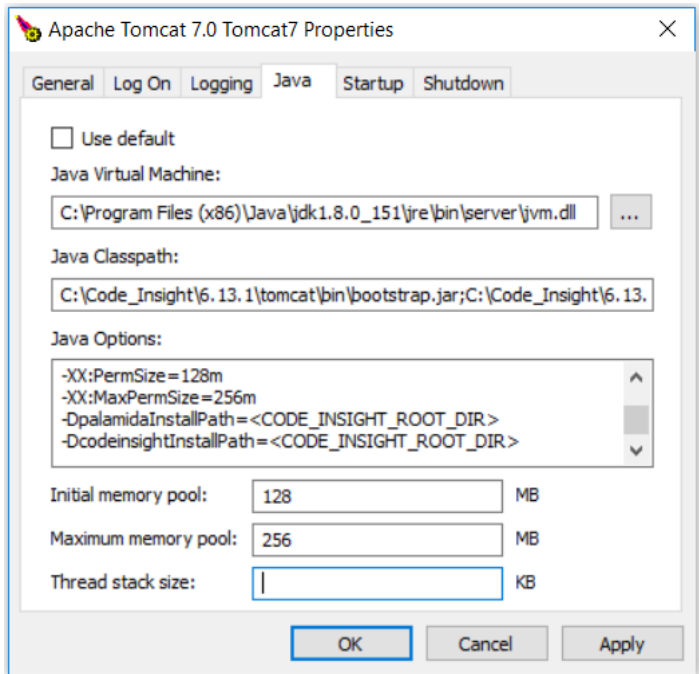
**Task**

**To configure the Code Insight Windows service, do the following:**

1. Stop the Tomcat server.
2. Using the command prompt, navigate to <CODE\_INSIGHT\_ROOT\_DIR>\tomcat\bin.
3. Execute the `service.bat install` command to install the Apache Tomcat Windows service.
4. Run <CODE\_INSIGHT\_ROOT\_DIR>\tomcat\bin\tomcat7w.exe.
5. Select the **Log On** tab:



- 6. Enter the user for which the Apache Tomcat Windows service will run.
- 7. Select the **Java** tab:



- 8. In the **Java Options** text area, append the following lines and make sure that there are no trailing spaces:

| Option               | Description                                                  |
|----------------------|--------------------------------------------------------------|
| -XX:PermSize=128m    | Set to appropriate value for your system.                    |
| -XX:MaxPermSize=256m | Set to appropriate value for your system.                    |
| -Dpalamida.ssl=true  | Only required to enable HTTPS (SSL); not required otherwise. |

| Option                                           | Description                                                           |
|--------------------------------------------------|-----------------------------------------------------------------------|
| -DpalamidaInstallPath=<CODE_INSIGHT_ROOT_DIR>    | Set to your Code Insight root directory, such as C:\Code_Insight\6.x. |
| -DcodeinsightInstallPath=<CODE_INSIGHT_ROOT_DIR> | Set to your Code Insight root directory, such as C:\Code_Insight\6.x. |

9. Modify the **Initial memory pool** value to **256 MB**.
10. Modify the **Maximum memory pool** value to **12288 MB**.



**Note** ▪ Set the maximum memory pool no higher than 80% of your available memory.

11. Click **Apply**.
12. Start the Tomcat server.

## Configure Code Insight as a Linux Service

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



**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
```

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 [Configuring Code Insight for Single Sign-On \(Optional\)](#).

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 \(Linux/Unix\)](#) for details.

# Using SCM Connectors

Code Insight supports multiple source code management (SCM) connectors to allow workspaces to obtain the appropriate codebase before a scan. This section explains the use of SCM connectors in the following topics:

- [Using the SCM Command Line Client](#)
- [Recommended Clients](#)
- [Setting the Environment Variable](#)
- [Workspace Settings](#)

## Using the SCM Command Line Client

Before you proceed, ensure that an SCM command line client is installed and configured on the Code Insight Scan Server as this is necessary for Code Insight to connect and sync to an SCM repository.



### Task

**To verify that the SCM client is installed and available to Code Insight, do the following:**

1. Open a Command Prompt and navigate to the Code Insight root directory.
2. Execute a command specific to your SCM. For example:
  - `ct help`
  - `git help`
  - `tf help`
  - `p4 help`
  - `svn 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.

# Recommended Clients

The following is a list of clients known to work well with Code Insight:



**Note** - Download site links are subject to change.

| SCM              | Client                       | Cost       | Download Site                                                                                                                                             |
|------------------|------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Subversion (SVN) | TortoiseSVN                  | Free       | <a href="http://tortoisesvn.tigris.org/">http://tortoisesvn.tigris.org/</a>                                                                               |
| Git              | Git                          | Free       | <a href="http://git-scm.com/downloads">http://git-scm.com/downloads</a>                                                                                   |
| TFS              | Team Explorer Everywhere     | Free       | <a href="https://github.com/Microsoft/team-explorer-everywhere/releases">https://github.com/Microsoft/team-explorer-everywhere/releases</a>               |
| Perforce         | Perforce Visual Client (P4V) | Free Trial | <a href="https://www.perforce.com/products/helix-core-apps/command-line-client">https://www.perforce.com/products/helix-core-apps/command-line-client</a> |
| Clearcase        | Rational ClearCase           | \$         | <a href="http://www-03.ibm.com/software/products/en/clearcase">http://www-03.ibm.com/software/products/en/clearcase</a>                                   |



**Note** - As of Code Insight version 6.11.3, legacy client Team Explorer is no longer supported for use with Code Insight. For compatibility with Code Insight, upgrade to Team Explorer Everywhere client and update the PATH variable to point to the location containing the new client.

## Verified Team Explorer Everywhere Client Versions

This release of Code Insight introduces support for Team Explorer Everywhere Client. The following versions of Team Explorer Everywhere Client have been verified for this release:

- TEE-CLC-11.0.0
- TEE-CLC-12.0.2
- TEE-CLC-14.0.2
- TEE-CLC-14.0.3
- TEE-CLC-14.0.4
- TEE-CLC-14.114.0
- TEE-CLC-14.118.0
- TEE-CLC-14.123.1

Downloads for these versions are available at <https://github.com/Microsoft/team-explorer-everywhere/releases>.



**Note** - As of Code Insight version 6.11.3, legacy client Team Explorer is no longer supported for use with Code Insight. For compatibility with Code Insight, upgrade to Team Explorer Everywhere client and update the PATH variable to point to the location containing the new client.

## Setting the Environment Variable

If you are running the SCM command line client from a Windows machine, verify that your SCM client location is added to the PATH environment variable.



### 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 the **Environment Variables** button
3. Look for the PATH system variable and make sure that it is set appropriately to the location of your SCM bin directory.

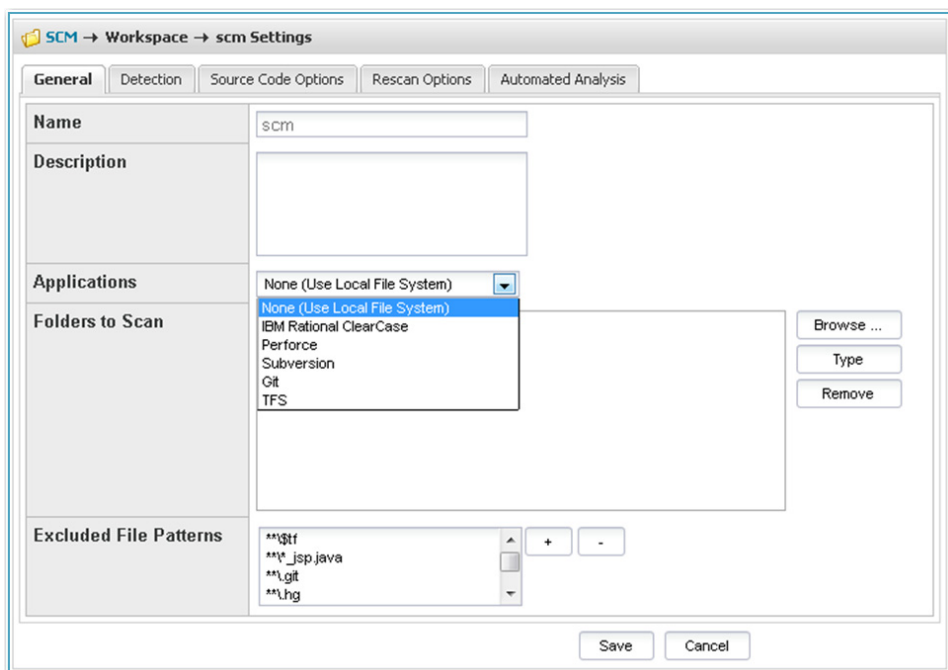


**Note** - Your SCM may require other environment variables to be set. Consult your SCM documentation for more information.

## Workspace Settings

If integration with ClearCase, Perforce, Subversion, Git, or TFS is enabled in the SCM configuration file for the Scan Server (<CODE\_INSIGHT\_ROOT>/config/scanEngine/scm.properties), the **Applications** dropdown in the **Workspace Settings – General** tab is populated with the available options.

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



## IBM Rational ClearCase

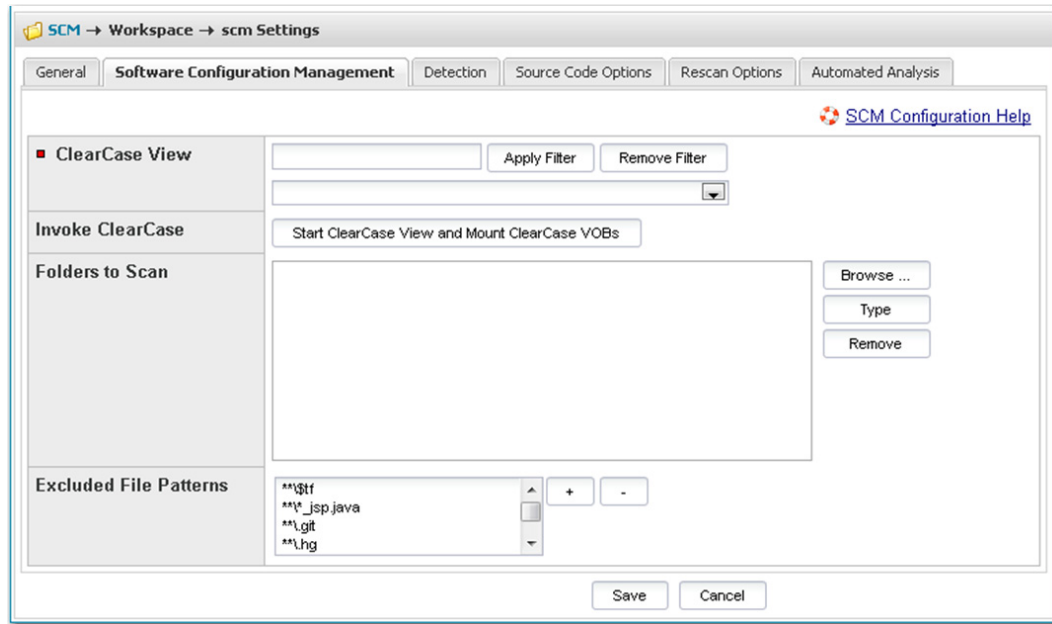
To configure the current workspace to use IBM Rational ClearCase for managing the codebase, perform the following steps.



### Task

**To configure the current workspace to use IBM Rational ClearCase, do the following:**

1. On the **Workspace Settings – General** tab, select **IBM Rational ClearCase** in the **Application** dropdown.
2. Click on the **Software Configuration Management** tab, and the **Software Configuration Management** tab screen appears. On this screen, you can enter ClearCase configuration options.



3. The following options are available for ClearCase:
  - **ClearCase View**—Select the ClearCase view to associate with the Code Insight workspace.
  - **ClearCase VOBs**—Select the ClearCase VOBs to associate with the Code Insight workspace.
4. Click the **Start ClearCase View and Mount ClearCase VOBs** button to ensure that ClearCase has been initialized for your selected view and VOBs.
5. In the **Folders to Scan** area, select the folders to scan. You can create new subdirectories if necessary via the browse dialog – **Create Child Directory**.
6. In the **Excluded File Patterns** area, define any file patterns that are to be excluded from the scan.
7. Both snapshot and dynamic ClearCase views are supported:
  - To scan a snapshot view, simply point to the folder.
  - To scan a dynamic view, use the following path structure in the **Folders to Scan** area:
 

```
/<view_location>/<vob_location>/<project>/<subdirectory>/<folder_1>
```

 For example:
 

```
/views/myview/vobs/myvob/ePortal-1.3/src
```

## Using Perforce (P4) to Manage the Codebase

To configure the current workspace to use Perforce for managing the codebase, perform the following steps.



**Task**      **To configure the current workspace to use Perforce, do the following:**


1. On the **Workspace Settings – General** tab, select **Perforce** in the **Application** dropdown.
2. Click on the **Software Configuration Management** tab, and the **Software Configuration Management** tab screen opens as shown below. On this screen, you can enter your Perforce configuration options.

3. The following options are available for Perforce:

| Option                         | Description                                                                                                                                                                                                                                                |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Perforce URL                   | Enter the repository URL in the following format:<br><br>p4: //<server>:<port>/<depot>/<project>                                                                                                                                                           |
| Perforce Username/<br>Password | Enter the username/password to create an authenticated connection to Perforce. In some cases for Perforce, anonymous access may be allowed in which case, you can leave the password field blank.                                                          |
| Perforce Sync To               | Select the sync to directive and provide the criteria if necessary. The following options are available: <ul style="list-style-type: none"><li>• Latest Revision</li><li>• Specific Revision</li><li>• Changelist</li><li>• Date</li><li>• Label</li></ul> |

| Option                        | Description                                                                                                                                          |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SCM Destination Folder</b> | Select the folder into which the codebase should be copied on the Scan Server. You can create new subdirectories if necessary via the browse dialog. |
| <b>Folders to Scan</b>        | Select the folders to scan within the SCM Destination Folder. You can create new subdirectories if necessary via the browse dialog.                  |
| <b>Excluded File Patterns</b> | Define any file patterns that are to be excluded from the scan.                                                                                      |

4. If you wish to sync to a specific folder in a project rather than to the whole project, you may do so by creating the same folder structure as in the repository and selecting the folder as your scan path. For example:

| Location                      | URL                                                                                                                                                            |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Perforce URL</b>           | p4://<server>:<port>/<depot>/<project>                                                                                                                         |
| <b>SCM Destination Folder</b> | /<sync location>/                                                                                                                                              |
| <b>Folders to Scan</b>        | /<sync location>/<depot>/<project>/<subdirectory>/<folder1>                                                                                                    |
|                               |  <p><b>Note</b> ▪ In this case, only folder 1 will be synced and scanned.</p> |

## Using Subversion (SVN) to Manage the Codebase

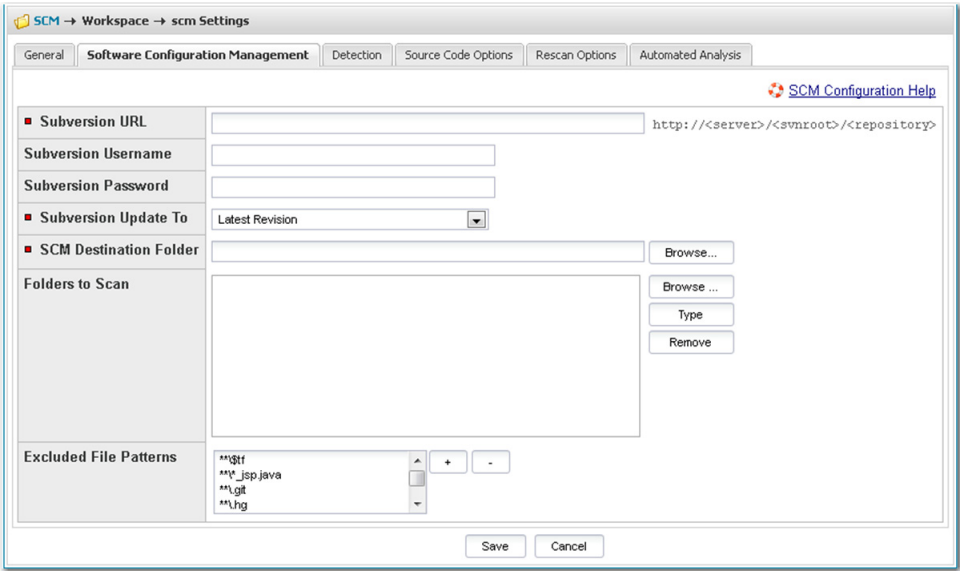
To configure the current workspace to use Subversion for managing the codebase, perform the following steps.



### Task

**To configure the current workspace to use Subversion, do the following:**

1. On the **Workspace Settings – General** tab, select **Subversion** in the **Application** dropdown.
2. Select the **Software Configuration Management** tab. The **Software Configuration Management** page appears.



3. Enter Subversion configuration options. The following options are available for Subversion:

| Option                               | Description                                                                                                                                                                                              |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Subversion URL</b>                | Enter the repository URL in the following format:<br><br>http://<server>/<svnroot>/<repository>                                                                                                          |
| <b>Subversion Username/ Password</b> | Enter the username/password to create an authenticated connection to Subversion.<br><br>In some cases for Subversion, anonymous access may be allowed in which case, you can leave both fields blank.    |
| <b>Subversion Update To</b>          | Select the update to directive and provide the criteria if necessary. The following options are available: <ul style="list-style-type: none"><li>● Latest Revision</li><li>● Specific Revision</li></ul> |
| <b>SCM Destination Folder</b>        | Select the folder into which the codebase should be copied on the Scan Server. You can create new subdirectories via the browse dialog.                                                                  |
| <b>Folders to Scan</b>               | Select the folders to scan within the SCM Destination Folder. You can create new subdirectories if necessary via the browse dialog.                                                                      |
| <b>Excluded File Patterns</b>        | Define any file patterns that are to be excluded from the scan.                                                                                                                                          |

4. To sync to a specific folder in a repository rather than to the whole repository, you may do so by adding the folder to the **Subversion URL**, such as:

http://<server>/<svnroot>/<repository>/<subdirectory>/<folder1>

In this example, only folder 1 will be synced and scanned.

# Git Repositories

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. Each protocol is discussed below.

- [Git Protocol Options](#)
- [Git Workspace Configuration](#)

## Git Protocol Options

With Git, you have the following protocol options:

- [Anonymous HTTP](#)
- [Authenticated HTTP](#)
- [SSH Authentication](#)
- [SSH over HTTPS](#)

### Anonymous HTTP

This protocol can be used for a public repository. Public repositories can be cloned without providing an account and password.

| Type                     | Example                                                  |
|--------------------------|----------------------------------------------------------|
| <b>GitHub Example</b>    | <code>http://github.com/myacct/Spoon-Knife.git</code>    |
| <b>Bitbucket Example</b> | <code>http://bitbucket.org/myacct/myquotefork.git</code> |

### 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                                                                 |
|--------------------------|-------------------------------------------------------------------------|
| <b>GitHub Example</b>    | <code>https://myacct:password@github.com/myacct/Hello-World.git</code>  |
| <b>Bitbucket Example</b> | <code>https://myacct:password@bitbucket.org/myacct/bb101repo.git</code> |

## SSH Authentication

This section describes SSH authentication between a system running Code Insight and Git servers such as GitHub and Bitbucket. The following options are possible:

- Use one SSH keypair for all Git servers.
- Use a separate keypair for each Git server.
- Use multiple keypairs for some or all Git servers.

SSH does not rely on account passwords but rather on a pair of keys, one a private key and the other a public key. Though a private key file may be protected by a password, no password should be specified for private keys used by Code Insight.

### Creating Keypairs

Use `ssh-keygen` to create a keypair for each Git server. Make the passphrase empty by hitting return twice. For example:

```
ssh-keygen -f ~/.ssh/id_rsa_github_test1 -C "github test 1"
ssh-keygen -f ~/.ssh/id_rsa_bitbucket_test1 -C "bitbucket test 1"
```

The created files are:

| Type      | Private Key            | Public Key                 |
|-----------|------------------------|----------------------------|
| GitHub    | id_rsa_github_test1    | id_rsa_github_test1.pub    |
| Bitbucket | id_rsa_bitbucket_test1 | id_rsa_bitbucket_test1.pub |

The private keys remain in the `.ssh` folder on Linux or the `<user_home>\.ssh` folder on Windows. Each public key will be stored on a Git server under a `palamida_account` as described below.

### Adding to the Config File

Update `.ssh/config` (on Linux) or `<user_home>\.ssh\config` (on Windows).

| Property                 | Github                     | Bitbucket                     |
|--------------------------|----------------------------|-------------------------------|
| Host                     | github.com                 | bitbucket.org                 |
| User                     | git                        | git                           |
| HostName                 | github.com                 | bitbucket.org                 |
| PreferredAuthentications | publickey                  | publickey                     |
| IdentityFile             | ~/.ssh/id_rsa_github_test1 | ~/.ssh/id_rsa_bitbucket_test1 |

There is a correspondence between the name on the Host line and the name used in the URL. When there is only one keypair per host, it is convenient to specify Host as above. This means the URL for git clone is:

```
git clone git@github.com:account/repository.git
```

The following definitions allow multiple keys to be used with GitHub or Bitbucket:

| Property                        | Github 1                   | Github 2                   |
|---------------------------------|----------------------------|----------------------------|
| <b>Host</b>                     | mygithub_01                | mygithub_02                |
| <b>User</b>                     | git                        | git                        |
| <b>HostName</b>                 | github.com                 | github.com                 |
| <b>PreferredAuthentications</b> | publickey                  | publickey                  |
| <b>IdentityFile</b>             | ~/.ssh/id_rsa_github_test1 | ~/.ssh/id_rsa_github_test2 |

The URLs are changed to use the values of Host from the config file. The appropriate git clone commands are:

```
git clone git@mygithub_01:account/repository.git
git clone git@mygithub_02:account/repository.git
```

Both clone commands will connect to github.com which is the value of **HostName**. The first command will use the private key id\_rsa\_github\_test1. The second command will use the private key id\_rsa\_github\_test2.

## Setting Up a Code Insight (Palamida) Account

The tasks involved in setting up a Code Insight (Palamida) account are:

- Define a palamida\_account.
- Add a public key to the palamida\_account.
- Grant the palamida\_account access to repositories to be scanned.

## Setting Up a palamida\_account

Below are instructions for setting up a palamida\_account on GitHub and Bitbucket:

- [Setting Up a palamida\\_account on GitHub](#)
- [Setting Up a palamida\\_account on Bitbucket](#)

### Setting Up a palamida\_account on GitHub

To set up a palamida\_account on GitHub, perform the following steps.



#### Task

**To set up a palamida account on GitHub, do the following:**

1. Create the account on GitHub.
2. Click **Account settings**, **SSH Keys**, **Add SSH** key.

3. Enter a **Title** and paste the contents of the public key file.

### Setting Up a palamida\_account on Bitbucket

To set up a palamida\_account on Bitbucket, perform the following steps:



---

**Task**      *To setup a palamida\_account on Bitbucket, do the following:*

1. Create the account on Bitbucket.
2. Click **Manage Account, SSH keys, Add key**.
3. Enter a **Label** and paste the contents of the public key file.

### Granting Access to a Repository

Below are instructions for granting access to the repository on GitHub and Bitbucket:

- [Granting Access to a Repository on GitHub](#)
- [Granting Access to a Repository on Bitbucket](#)

#### Granting Access to a Repository on GitHub

For a repository that will be scanned, give access to the palamida\_account.



---

**Task**      *To grant access to a repository on GitHub, do the following:*

1. From GitHub's **Repository** page, click **Settings, Collaborators**.
2. Add the palamida\_account.
3. If you want to verify that the account has been created, from the Code Insight system enter:  

```
git ls-remote git@github.com:account_name/repository_name.git
```

#### Granting Access to a Repository on Bitbucket

For repository that will be scanned, give access to the palamida\_account.



---

**Task**      *To grant access to a repository on Bitbucket, do the following:*

1. From Bitbucket's **Repository** page, click **Administration, Access management**.
2. Add the palamida\_account as a **User** with **Read** access.
3. If you want to verify, from the Code Insight system enter:  

```
git ls-remote git@bitbucket.org:account_name/repository_name.git
```

## SSH over HTTPS

The standard SSH port is 22. To run SSH over port 443, perform the steps discussed above in the [SSH Authentication](#) section. The only difference is in `.ssh/config` on Linux or `<user_home>\.ssh\config` on Windows. Some examples are:

| Property                        | Example 1                                 | Example 2                                   |
|---------------------------------|-------------------------------------------|---------------------------------------------|
| <b>Host</b>                     | github.com                                | gitssh-https                                |
| <b>User</b>                     | git                                       | git                                         |
| <b>Port</b>                     | 443                                       | 443                                         |
| <b>HostName</b>                 | ssh.github.com                            | ssh.github.com                              |
| <b>PreferredAuthentications</b> | publickey                                 | publickey                                   |
| <b>IdentityFile</b>             | ~/.ssh/id_rsa_github_test1                | ~/.ssh/id_rsa_github_test1                  |
| <b>URL</b>                      | git@github.com:account/<br>repository.git | git@gitssh-https:account/<br>repository.git |

## Git Workspace Configuration

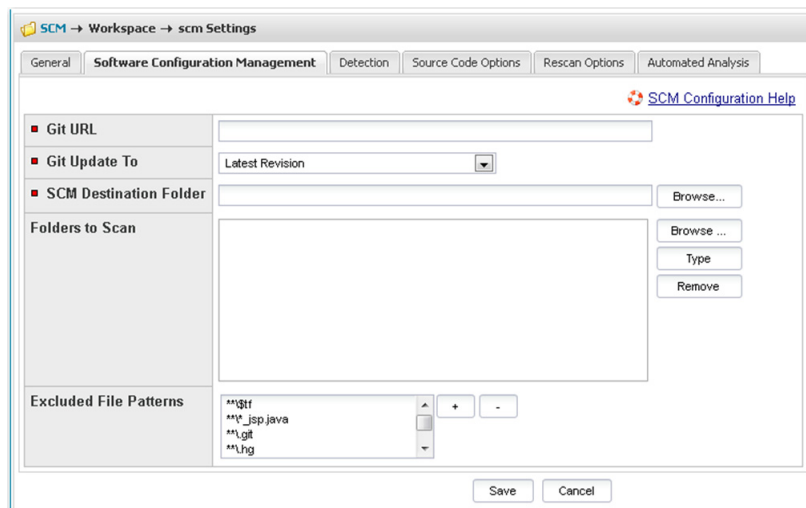
To configure the current workspace to use Git for managing the codebase, perform the following steps.



### Task

**To configure the current workspace to use Git, do the following:**

1. On the **Workspace Settings – General** tab, select **Git** in the **Application** dropdown.
2. Click on the **Software Configuration Management** tab, and the **Software Configuration Management** tab screen appears. On this screen, you can enter Git configuration options.



3. The following options are available for Git:

| Option                 | Description                                                                                                                                                                                         |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Git URL                | Enter the repository URL in the appropriate format for your desired protocol:                                                                                                                       |
|                        | <b>Anonymous HTTP</b><br><code>http://server/account/repository.git</code><br><code>https://server/account/repository.git</code>                                                                    |
|                        | <b>Authenticated HTTP</b><br><code>http://account:password@server/account/repository.git</code><br><code>https://account:password@server/account/repository.git</code>                              |
|                        | <b>SSH</b><br><code>git@server:account/repository.git</code><br><code>ssh://server/account/repository.git</code>                                                                                    |
|                        | <b>SSH over HTTPS (GitHub only)</b><br><code>git@server:account/repository.git</code><br><code>ssh://server/account/repository.git</code>                                                           |
| Fit Update To          | Select the update to directive and provide the criteria if necessary. The following options are available: <ul style="list-style-type: none"><li>• Latest Revision</li><li>• Specific Tag</li></ul> |
| SCM Destination Folder | Select the folder into which the codebase should be copied on the Scan Server. You can create new subdirectories if necessary via the browse dialog.                                                |
| Folders to Scan        | Select the folders to scan within the SCM Destination Folder. You can create new subdirectories if necessary via the browse dialog.                                                                 |
| Excluded File Patterns | Define any file patterns that are to be excluded from the scan.                                                                                                                                     |

4. The system does not currently support syncing to a specific folder within a GIT repository without pulling in all the data in that repository.

## Microsoft Team Foundation Server (TFS)

To configure the current workspace to use TFS for managing the codebase, perform the following steps.



### Task

**To configure the current workspace to use TFS, do the following:**

1. On the **Workspace Settings – General** tab, select **TFS** in the **Application** dropdown.
2. Click on the **Software Configuration Management** tab, and the **Software Configuration Management** screen appears. On this screen, you can enter your TFS configuration options.

3. The following options are available for TFS:

| Option                        | Description                                                                                                                                                                                                                      |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TFS URL</b>                | Enter the repository URL in the following format:<br><code>http://&lt;server&gt;:&lt;port&gt;/&lt;tfsroot&gt;/&lt;collection&gt;/&lt;project&gt;</code>                                                                          |
| <b>TFS Username/Password</b>  | Enter the username/password to create an authenticated connection to the TFS repository.                                                                                                                                         |
| <b>TFS Update To</b>          | Select the update to directive and provide the criteria if necessary. The following options are available: <ul style="list-style-type: none"> <li>• Latest Changelist</li> <li>• Specific Changelist</li> <li>• Label</li> </ul> |
| <b>SCM Destination Folder</b> | Select the folder into which the codebase should be copied on the Scan Server. You can create new subdirectories if necessary via the browse dialog.                                                                             |
| <b>Folders to Scan</b>        | Select the folders to scan within the SCM Destination Folder. You can create new subdirectories if necessary via the browse dialog.                                                                                              |
| <b>Excluded File Patterns</b> | Define any file patterns that are to be excluded from the scan.                                                                                                                                                                  |

4. To sync to a specific folder in a project rather than to the whole project, add the folder to the URL:

`http://<server>:<port>/<tfsroot>/<collection>/<project>/<subdirectory>/<folder_1>`

In this example, only folder 1 will be synced and scanned.

## Project Copy Settings

As part of the project copy configuration, if an SCM application has been configured for any of the source workspaces, the target workspace in the new project also must be configured for the same SCM application. The user is presented with input fields for each target workspace where the source workspace has an SCM application configured.

The screenshot shows the 'Project Copy' dialog box with the 'Workspace Options' tab selected. The dialog is divided into several sections:

- Pre-Copy Options:** Includes a checkbox 'Compare Source and Target File Paths' (checked) with a note: '(only files that are marked as reviewed, tagged, or associated with groups will be compared)'.
- Workspace #1:** A list on the left side of the dialog.
- Workspace Name:** A section with 'Source: tfs\_test' and a 'Target:' input field.
- Workspace Software Configuration Management Settings:** A section containing:
  - Source:** Fields for 'TFS URL:' (http://127.0.0.1:8080/tfs/DefaultCollection/ePortal-1.3), 'TFS Username:' (palamida), 'TFS Sync To:' (Latest Revision), and 'SCM Destination Folder:' (C:\Cleanroom\tfs\_repo\).
  - Target:** Fields for 'TFS URL:', 'TFS Username:', 'TFS Password:' (with a 'Use Existing Password' checkbox), 'TFS Sync To:' (Latest Changeset), and 'SCM Destination Folder:' (with 'Browse...' and 'Type' buttons).
- Workspace Folders to Scan:** A section with 'Source: C:\Cleanroom\tfs\_repo\' and a 'Target:' input field (with 'Browse...' and 'Type' buttons).

At the bottom of the dialog are 'Copy Project' and 'Cancel' buttons.

Depending on which SCM application was configured in the source workspace, the set of input fields will vary. Refer to the [Workspace Settings](#) section for detailed explanation of each input field.

# Performing Backup and Recovery

To prevent permanent data loss, it is recommended that you perform regular backups of the configuration files, workspaces, and central database associated with Code Insight. This section explains how to perform a complete backup of the system:

- [Terminology](#)
- [Performing the Backup](#)
- [Performing the Restore](#)

## Terminology

The `<CODE_INSIGHT_ROOT_DIR>` refers to the directory in which Code Insight has been installed.

The `<CODE_INSIGHT_USER_HOME>` refers to the `$user.home/.codeinsight/config/` directory, which contains client logs and Detector client properties used by Code Insight.

## Performing the Backup

To perform a backup, perform the following steps.



---

### Task

**To perform a backup, do the following:**

1. The Code Insight Core Server and all Scan Servers must be stopped to perform the backup.
  - See [Administration: Starting & Stopping Servers](#) for details on stopping the servers.
  - If you have a multi-Scan Server environment, perform these steps for each Scan Server.
2. Copy or tar/zip the `<CODE_INSIGHT_USER_HOME>` directory.
  - This directory contains the client logs and Detector client properties used by Code Insight.

- If you have a multi-Scan Server environment, perform these steps for each Scan Server.
- 3. Copy or tar/zip the <CODE\_INSIGHT\_ROOT\_DIR> directory.
  - This directory contains all the components of the Code Insight system.
  - If you have a multi-Scan Server environment, perform these steps for each Scan Server.
- 4. Copy or tar/zip the workspaces directory.
  - The workspaces directory is defined in the <CODE\_INSIGHT\_USER\_HOME>/scanEngine/scanEngine.properties file. Search for the workspaceBaseDirPath parameter. Typically, this would be located in <CODE\_INSIGHT\_ROOT\_DIR>/workspaces.
  - This is only necessary if the workspaces directory has been defined outside of the <CODE\_INSIGHT\_ROOT\_DIR>.
  - This directory contains all of the workspace data and metadata.
  - If you have a multi-Scan Server environment, perform these steps for each Scan Server.
- 5. Take a complete database dump of the central database configured in the <CODE\_INSIGHT\_USER\_HOME>/core/core.db.properties file.
  - **For MySQL**, you can use one of the following:

| Utility              | URL                                                                                                                       |
|----------------------|---------------------------------------------------------------------------------------------------------------------------|
| mysqldump program    | <a href="http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html">http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html</a> |
| MySQL Backup utility | <a href="http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html">http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html</a> |

- **For Oracle**, refer to the following for detailed instructions:  
[http://www.oracle.com/wiki/Oracle\\_database\\_Backup\\_and\\_Recovery\\_FAQ](http://www.oracle.com/wiki/Oracle_database_Backup_and_Recovery_FAQ)
- 6. When the backup is completed, start the Code Insight servers.
  - See the **Administration: Starting & Stopping Servers** section of this guide for details on starting the servers.
  - If you have a multi-Scan Server environment, perform these steps for each Scan Server.

## Performing the Restore

To perform a restore, use the following steps.



### **Task** To perform a restore, do the following:

1. The Code Insight Core and all Scan Servers must be stopped to perform the restore.
  - See the **Administration: Starting & Stopping Servers** section of this guide for details on stopping the servers.
  - If you have a multi-Scan Server environment, perform these steps for each Scan Server.

2. Delete the current content of the <CODE\_INSIGHT\_USER\_HOME> directory, and extract the <CODE\_INSIGHT\_USER\_HOME> backup to this location.
  - If you have a multi-Scan Server environment, do this for each Scan Server.
3. Delete the current content of the <CODE\_INSIGHT\_ROOT\_DIR> directory, and extract the <CODE\_INSIGHT\_ROOT\_DIR> backup to this location.
  - If you have a multi-Scan Server environment, perform these steps for each Scan Server.
4. Delete the current content of the workspaces directory, and extract the backup to this location.
  - This is only necessary if the workspaces directory has been defined outside of the <CODE\_INSIGHT\_ROOT\_DIR>.
  - If you have a multi-Scan Server environment, perform these steps for each Scan Server.
5. Restore the complete database dump to the central database.
  - **For MySQL**, you can use one of the following:

| Utility                                           | URL                                                                                                                           |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>mysqlimport program</b>                        | <a href="http://dev.mysql.com/doc/refman/5.7/en/mysqlimport.html">http://dev.mysql.com/doc/refman/5.7/en/mysqlimport.html</a> |
| <b>MySQL Administrator Client Restore utility</b> | <a href="http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html">http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html</a>     |

- **For Oracle**, refer to [http://www.orafaq.com/wiki/Oracle\\_database\\_Backup\\_and\\_Recovery\\_FAQ](http://www.orafaq.com/wiki/Oracle_database_Backup_and_Recovery_FAQ) for detailed instructions.
6. Once the restore is completed, you may start the Code Insight servers.
    - See [Administration: Starting & Stopping Servers](#) for details on starting the servers.
    - If you have a multi-Scan Server environment, perform these steps for each Scan Server.



# Frequently Asked Questions

This section lists the following frequently asked questions about Code Insight installation:

- Can I use the installer to update an existing installation of Code Insight?
- If an installation was canceled or did not complete successfully, can I re-run the installer with the same root directory and database schema?
- Can I install into a directory that does not exist?
- Can I input a domain name or “localhost” instead of an IP address for the Core, Scan and s?
- What can I do if the GUI installer doesn’t allow me to install to a drive other than C:\?
- How can I see which scripts were run by the installer and which tables were created?
- What is the Scan Root Directory?
- What should I do if my maximum memory is not listed?
- What if I make a mistake?
- Can I configure multiple Scan Servers using the Installer?

## Can I use the installer to update an existing installation of Code Insight?

No, running the Installer on an existing installation will overwrite existing project and workspace data.

## If an installation was canceled or did not complete successfully, can I re-run the installer with the same root directory and database schema?

Yes, you may run the Installer again using the same information. If prompted to overwrite existing files, click **Yes to All**.

## Can I install into a directory that does not exist?

Yes, you may install into a directory that does not exist. The directory will be created after you have gone through all the prompts for setting up the product

### Can I input a domain name or “localhost” instead of an IP address for the Core, Scan and s?

Yes, you may use a domain name or `localhost` as long as the DNS point settings have been configured for the IP address and your machine can resolve the name.

### What can I do if the GUI installer doesn't allow me to install to a drive other than C:\?

Click **Choose**. In the folder edit box, type the drive letter (such as `D:\`) and click **Enter**. Use the mouse to select the directory to install into and click **OK**.

### How can I see which scripts were run by the installer and which tables were created?

You may check the `<CODE_INSIGHT_ROOT_DIR>/logs` directory for more information about each script that was executed during the installation.

### What is the Scan Root Directory?

The Scan Root Directory is the root of the server file system. You may use this property to control which files and directories users have access to. It can be specified during the Installer session or at a later time by editing the `scanEngine.properties` file.

### What should I do if my maximum memory is not listed?

Choose the option that is closest to the actual memory on your machine.

### What if I make a mistake?

In the Windows GUI, you may press the **back** button to fix any user-input errors until you get the product is being installed (Installation page).

In the Console, exit the Installer by pressing **CTRL-C** and run it again.

### Can I configure multiple Scan Servers using the Installer?

Yes, you may use the Installer to configure multiple Scan Servers by running “Install Set 2 – Core” on the Core Server and running “Install Set 3 – Scanner” on each Scan Server. When you are done, manually edit `core.properties` to list every Scan Server.