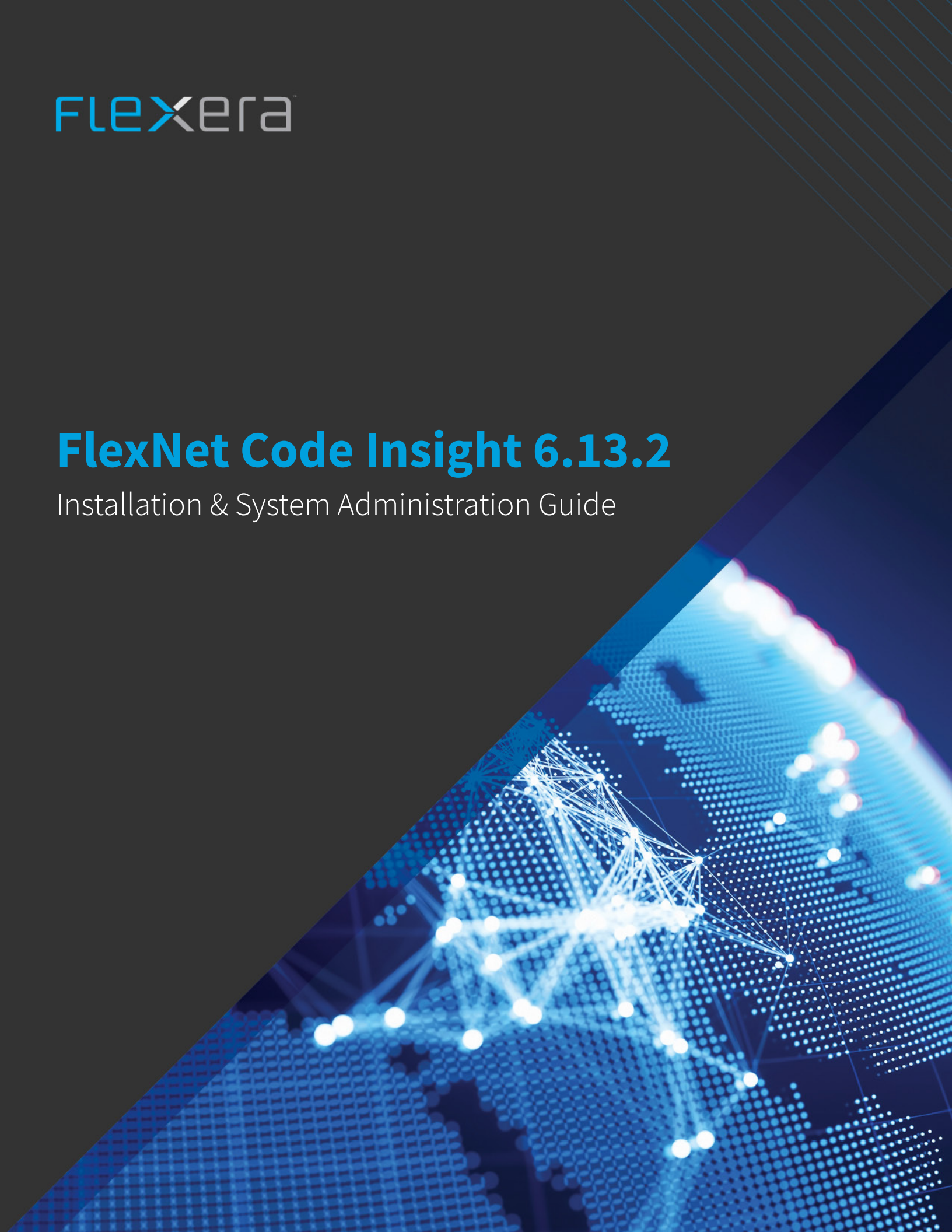




FlexNet Code Insight 6.13.2

Installation & System Administration Guide



Legal Information

Book Name: FlexNet Code Insight 6.13.2 Installation & System Administration Guide
Part Number: FNCI-6132-IG00
Product Release Date: August 2019

Copyright Notice

Copyright © 2019 Flexera.

This publication contains proprietary and confidential information and creative works owned by Flexera 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 is strictly prohibited. Except where expressly provided by Flexera in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera intellectual property rights, whether by estoppel, implication, or otherwise.

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

Intellectual Property

For a list of trademarks and patents that are owned by Flexera, see <https://www.flexerasoftware.com/legal/intellectual-property.html>. All other brand and product names mentioned in Flexera 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 FlexNet Code Insight 6.13.2 Installation and System Administration Guide	9
FlexNet Code Insight Document Set	9
Types of Installation	10
Single Server vs. Multiple Server Deployments	11
Dedicated Disk Drives	11
Installing Manually or Installing Using the Installer	12
Database Server and Application Server	13
Special Consideration for Non-US Customers	15
Special Considerations for LDAP Integration	15
Special Considerations for SCM and Build System Integration	15
Running on a Virtual Server (e.g. VMware)	15
Installable Sub-Components	15
Prerequisites	16
Operating Systems	17
Databases	18
Hardware	18
Supported Hardware Configurations	18
CPU Specifications	19
Software	20
Browsers	22
Source Code Management	22
Network and Firewall Considerations	22
System Settings	23
Environment Variables	23
Linux/Unix Environment Variables	23
Windows Environment Variables	24
Mail Server Requirements and Configuration	24
Setting the Open File Limit (Linux/Unix)	24
Maximum Virtual Memory Limit	26

Database Settings	26
Configuring a Database	26
Character Encoding	27
Setting the UTF-8 for MySQL Installations on Linux	27
Setting the UTF-8 for MySQL Installations on Windows	28
Compliance Library	28
2 Installing FlexNet Code Insight Manually	29
3 Extracting Application Files	31
Choosing an Installation Directory	31
Installing the Compliance Library	32
Installing FlexNet Code Insight	32
Installing the Database Driver	32
Activating FlexNet Code Insight	33
4 Preparing the Database for Installation	35
Preparing the Database	35
Running the Database Setup Scripts	36
5 Configuring the Tomcat Web Server	37
Setting Max Heap	37
Setting the Install Directory	38
Linux Installations: Headless Mode for JVM	38
Enabling HTTP Secure (HTTPS over SSL)	38
Increasing Post Request Size	38
Running as a Windows Service (32-bit and 64-bit)	39
6 Editing Configuration (Properties) Files	41
Entering Required Information	41
Installation Options	43
Configuring Email Services	43
Disabling SMTP Mail Server Authentication	44
Configuring Email Notification Options	44
Configuring Notification Email Templates	44
Configuring a Manual Proxy Server Connection (Optional)	46
Scan Engine Configuration Options	47
Tuning Copyright Detection	47
Controlling False Positives	47
Controlling False Negatives	48
Claiming Detected Copyrights as Yours	49
Tuning Email/URL Detection	50
Controlling False Positives	50

Tuning License Detection	50
Editing the License Detection Patterns File	51
License Text Template Example for Apache License, Version 2.0	51
License Keywords Example for Apache License, Version 2.0	51
Adding Custom Phases to the Scan Process	52
Sample Custom Scan-Phase Adapter Script	52
Rules and Considerations for Creating a Custom Scan-Phase Adapter	52
Class Used in Custom Scan-Phase Adapter	53
Scan Phase Order	53
Multiple Scripts Using Same Method	54
Customizing the Scan Phase Order	54
7 Administration: Starting & Stopping Servers	55
Starting and Stopping the Server Manually	55
Validating a Successful Server Startup	56
Using a Supported Browser	56
Using an .xls File	56
Changing the Number of Log Files Maintained	56
8 Running an Electronic Update	59
Running an Electronic Update for the First Time	59
Running an Electronic Update Manually	59
Using the Electronic Update Server (HTTPS)	59
Using a Local Electronic Update Archive	60
Scheduling Automatic Electronic Updates	60
9 Installing FlexNet Code Insight Using the Installer 61	
Installing FlexNet Code Insight With the Installer	61
Installing FlexNet Code Insight Using the Linux Command Line	65
10 Launching FlexNet Code Insight	73
Starting and Stopping the Server	73
Logging into FlexNet Code Insight	74
Running the Update Service	74
11 Installing the Jenkins Plugin	75
Jenkins Downloads	75
Setting up the Jenkins Plugin	76
Generating a JWT Token	77
12 Using ScriptRunner	79
ScriptRunner Options	79

Generating a JSON Web Token (JWT).....	80
Using the JWT Token with scriptRunner.....	81
13 Exporting & Importing Workspaces.....	83
Usage Overview.....	83
What is Custom Data?.....	84
What Workspace Data is Exported and Imported?.....	84
What Inventory is Exported and Imported?.....	84
About Backward Compatibility.....	85
Installing the Scripts.....	85
Installing Manually.....	85
Using the Scripts with ScriptRunner.....	86
Using the Export Script.....	86
Export Options.....	86
File and Path Options.....	86
Server Options.....	87
What to Export.....	88
Boolean Options.....	88
Other Options.....	89
Combining Export Flags.....	89
Export Usage.....	89
Export Output.....	89
Export Usage Examples.....	90
Using the Import Script.....	90
Import Options.....	91
Import Options that Accept Parameters.....	91
Import Options that Require an MD5 Match.....	92
Import Options that Do Not Require an MD5 match.....	93
Import Options that Accept Boolean Options.....	93
CSV File.....	93
Import Results.....	94
Log file (import.log).....	94
Metadata Tags.....	94
Import Usage Examples.....	95
14 Integrating with LDAP for Authentication (Optional).....	97
Configuring LDAP Integration.....	97
Configurable LDAP Properties.....	99
Data Synchronization.....	99
User Metadata.....	99
Disabled Users.....	99
LDAP Synchronization to Multiple Sources.....	99
LDAP Configuration.....	100
Set Up a User Search Filter.....	100
Default Role Assignment.....	101

Set Up a User List Sync	101
Server Paging	102
LDAP over SSL	103
Troubleshooting and Testing LDAP Configurations	103
15 Configuring Additional Scan Servers (Optional)	105
Overview	105
Configuration Details	105
16 Configuring FlexNet Code Insight using MySQL Commands	107
17 Configuring the FlexNet Code Insight Windows Service	109
18 Using SCM Connectors	111
Using the SCM Command Line Client	111
Recommended Clients	112
Verified Team Explorer Everywhere Client Versions	112
Setting the Environment Variable	113
Workspace Settings	113
IBM Rational ClearCase	114
Using Perforce (P4) to Manage the Codebase	115
Using Subversion (SVN) to Manage the Codebase	117
Git Repositories	119
Git Protocol Options	119
Anonymous HTTP	119
Authenticated HTTP	119
SSH Authentication	120
SSH over HTTPS	123
Git Workspace Configuration	123
Microsoft Team Foundation Server (TFS)	125
Project Copy Settings	126
19 Performing Backup and Recovery	127
Terminology	127
Performing the Backup	127
Performing the Restore	128
20 Frequently Asked Questions	131

FlexNet Code Insight 6.13.2 Installation and System Administration Guide

This guide is intended for system administrators, database administrators, and anyone who wants a deeper understanding of FlexNet Code Insight. This section contains the following topics:

- [FlexNet Code Insight Document Set](#)
- [Prerequisites](#)

FlexNet Code Insight Document Set

The following are the titles of all the documentation in the FlexNet 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) machine. It can also be used by a system administrator and database administrator performing system administration.
User Guide	This guide contains activity-based information for using the features and functionality.
Licensing Guide	This guide contains third-party licensing text.

Types of Installation

The type of installation you choose should be based on the size and business needs of your organization. The table below lists some possible configurations.

Table 1-2 • Installation Types

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

Table 1-2 • Installation Types

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

Single Server vs. Multiple Server Deployments

A single instance of FlexNet Code Insight Core server and multiple FlexNet Code Insight Scan Engine servers are supported for this release. See [Configuring Additional Scan Servers \(Optional\)](#) for details on configuring additional FlexNet Code Insight Scan Engine servers.

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:

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

Installing Manually or Installing Using the Installer

There are two methods for installing FlexNet Code Insight:

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

See the section that applies to your installation method.

Database Server and Application Server

Information about sizing and backing up database and application servers is presented in the following section.

Table 1-3 • Servers

Item	Description
Database Server	We recommend a separate server for the database to eliminate resource contention between the FlexNet Code Insight application and database.
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 my.cnf file in your /etc/mysql directory</p> <p>If you do not have a my.cnf in your /etc/mysql directory, locate one of these files in the /usr/share/mysql directory and copy it to /etc/mysql. Rename the file my.cnf.</p> <ul style="list-style-type: none"> • mysql-large.cnf (MySQL prior to 5.5) • my-large.cnf (MySQL 5.5 to 5.6) • my-default.cnf (MySQL 5.6+) <p>Navigate to the /etc/mysql/my.cnf 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 my.ini or my-default.ini file located in your MySQL Server installation directory (for example, C:\Program Files\MySQL\MySQL Server 5.6), and edit the following lines:</p> <pre>[mysqld] default-character-set=utf8</pre>

Table 1-3 • Servers

Item	Description
Storage Engine	<p>Select InnoDB as the default storage engine if you are using MySQL.</p> <p>Linux:</p> <p>Edit the my.cnf file in the [mysqld] section.</p> <pre>default-storage-engine = INNODB</pre> <p>Windows:</p> <p>Add the following line to the my.ini 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>
Buffer Pools	<p>To support FlexNet 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 1GB.</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 30GB to accommodate the FlexNet 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 30GB.</p> <p>Scale up by 2MB 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 30GB 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

FlexNet 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 FlexNet Code Insight Support for specific details on how to integrate with your existing SCM installation.

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

To run the FlexNet 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 FlexNet Code Insight:

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

Prerequisites

The section lists the prerequisites for installing and using FlexNet 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](#)

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

Operating Systems

FlexNet Code Insight supports the following operating systems.

Table 1-4 • Supported Operating Systems

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

The following operating systems have been used to run FlexNet 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)

Databases

FlexNet Code Insight supports the following database software.

Table 1-5 • Supported and Recommended Databases

Supported	Recommended
<ul style="list-style-type: none">MySQL 5.6, 5.7Oracle 11g, 12cMS SQL Server<ul style="list-style-type: none">2012 r2 Enterprise2014 Enterprise2016 Enterprise	<ul style="list-style-type: none">MySQL 5.7Oracle 11gMS SQL Server<ul style="list-style-type: none">2012 r2 Enterprise2014 Enterprise2016 Enterprise

The MySQL 5.0-5.5 database version has been used to run FlexNet 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 FlexNet Code Insight component configurations:

Table 1-6 • Supported Hardware Configurations

Type	Supported	Recommended
Scan Server	<ul style="list-style-type: none">32GB RAM750GB free hard disk space for the following:<ul style="list-style-type: none">Compliance Library (approximately 500GB on disk)Code Base (materials to be scanned)Workspaces (scanned results)	<ul style="list-style-type: none">32GB or 64GB RAM depending on expected load1TB disk space for the following:<ul style="list-style-type: none">Compliance Library (approximately 500GB on disk)Code Base (materials to be scanned)Workspaces (scanned results)Solid State Drive (SSD)<ul style="list-style-type: none">Significant performance benefits if workspace directory is located on SSD drive

Table 1-6 • Supported Hardware Configurations

Type	Supported	Recommended
Core Server	<ul style="list-style-type: none"> 16GB RAM At least 650MB of space for product and attachments See the Database Server entry below if hosting both Core Server and database on the same machine 	<ul style="list-style-type: none"> 32GB RAM (required if Core Server and database reside on same machine) 30GB of space for product and attachments See the Database Server entry below if hosting both Core Server and database on the same machine
Client	<ul style="list-style-type: none"> 16GB RAM 	<ul style="list-style-type: none"> 32GB RAM
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 30GB (for SQL Server, 50GB) to accommodate the FlexNet Code Insight Data Libraries and other data related to users, teams, projects, and such. If you install the database on the same machine 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.) After starting with the base size, scale up by 2MB for every 5,000 files scanned. Begin by estimating how much you will scan in the first 6 months, and add that to the 30GB base size. As for data volume, FlexNet 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 1G (innodb_buffer_pool_size = 1G).
- For SQL Server, it is strongly recommended that the database and the Core Server reside on the same machine (with a minimum hard-drive requirement of 50GB for the database and 30GB for the Core Server, for a total of 80GB).

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 32GB RAM for the core server (as listed in [Supported Hardware Configurations](#)), the CPU specifications for the machine running the core server include 2-CPU, each at least 2 GHZ+, with 8+ cores (as listed below).

Table 1-7 • CPU Requirements

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

Software

FlexNet Code Insight supports the following software.

Table 1-8 • Supported Software


Software	Description	Download URL
Java JDK	Either of these required on all Core and Scan servers. Use the latest update when possible.	Oracle JDK 8 http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
	<ul style="list-style-type: none"> Oracle JDK 8 (64-bit) 	Zulu Open JDK 8 https://www.azul.com/downloads/zulu/
	You must purchase a license from Oracle to ensure that you receive updates.	
	<ul style="list-style-type: none"> Zulu OpenJDK 8 (64-bit) (from Azul) 	
Java JRE	<p>Oracle JRE 8 (64-bit) (update 172) 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
	 <p>Note • Not required for Workflow-only installations or on client servers that already have the JDK installed.</p>	

Table 1-8 • Supported Software

Software	Description	Download URL	
Database Client	Required to access the FlexNet Code Insight database server and to execute database scripts (not required if database is to be managed directly from the database server). Any basic client application or command line client interface may be used. Several options are listed on the right.	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
Database Driver	JDBC driver required on the Core and Scan servers to enable FlexNet Code Insight access to the database. Download the driver corresponding to your database type and do one of the following: <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 FlexNet Code Insight, copy the downloaded .jar file to the following location: <pre><Code Insight_ROOT_DIR>\<version>\ tomcat\lib\</pre> 	MySQL	mysql-connector-java-5.1.x-bin.jar http://dev.mysql.com/downloads/connector/j/5.1.html
		Oracle	ojdbc6.jar 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 FlexNet Code Insight server.		

Additional Notes about Software Requirements

Note the following about software requirements:

- Support for Java 7 (JDK and JRE) was removed in FlexNet 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 FlexNet 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.

Browsers

FlexNet Code Insight supports the following browsers.

Table 1-9 • 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

FlexNet Code Insight supports the following source code management software.

Table 1-10 • 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

Configure the servers by specifying a fully qualified domain name (for example, hostname.domain.com) or IP address. Enable those port numbers used by FlexNet Code Insight in all of the firewalls. You may use the default port numbers listed below or configure the service to use a custom port.

Table 1-11 • Port Numbers Used in Firewalls

Port #	Details
1433/1521/3306	Database Server Access Port (MS SQL Server, Oracle, MySQL)
8888/443	Tomcat (http/https)
465	External SMTP (mail) Server
389	External Authentication Directory Server (Active Directory/LDAP)

Table 1-11 • Port Numbers Used in Firewalls

Port #	Details
8005 and 8009	Tomcat Connector and Tomcat Shutdown Ports (local access only)



Note • All ports used by the application are also configurable. You may designate a custom port as needed.

System Settings

This section lists the FlexNet 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 FlexNet 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 and scan servers.
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:

```
which java
```
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 and scan servers.
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>\6.10=1.1\config\core**.

Setting the Open File Limit (Linux/Unix)

This procedure specifies the open file limit in Linux/Unix.



Important • You can choose to perform this step before or after the application is installed; however, make sure to perform this step before launching the application.



Important • These steps related to increasing open file size are essential for FlexNet Code Insight to function properly on Unix/Linux platforms.

You need to increase the maximum number of open files setting for your particular operating system from the default number. This number indicates the maximum number of files normal users, for example, non-root users, can have open in a single session.

The following steps are verified for Fedora Core4 and may vary based on the OS/Linux Distribution. We assume you are using the Bourne Shell (sh), Bourne Again Shell (bash), or Korn Shell (ksh).



Note • If you are using C Shell (*csh*) or Tenex C Shell (*tcsh*), you must use the `Limit openfiles NNNN` syntax instead of the `ulimit -n NNNN` syntax.



Note • There are some limitations to using *tcsh*. For example, *tcsh* shell only sets the soft-limit. You can't set the hard-limit using *tcsh*. This may be 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                (seconds, -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 may require a different set up.



Tip • On some systems, it may be necessary to add `ulimit -n 16384` to the source file (such as `.profile`, `.bashrc`, or `.bash_profile`) to make sure 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 **16384**.

```
[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                (seconds, -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 FlexNet 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 FlexNet 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 machines that FlexNet Code Insight is using (both core and Scan Engine servers). For example, in MySQL with a database user named FlexNet 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 5.6`, 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

The FlexNet Code Insight Compliance Library is required before proceeding with the installation. Ensure that you allocate at least 520 GB of space to the drive that will contain the library.

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

```
<CODE_INSIGHT_ROOT_DIR>\PDL\CL\2.34.
```

Installing FlexNet Code Insight Manually

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



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



Task

To install FlexNet Code Insight manually, do the following:

1. Prepare your environment and install the prerequisite software on the Core/Scan Engine server. See [Preparing the Database for Installation](#) for detailed steps.
2. Extract the FlexNet Code Insight application. See [Extracting Application Files](#) for detailed steps.
3. Copy the FlexNet Code Insight Compliance Library. See [Installing the Compliance Library](#) for details.
4. Copy the FlexNet Code Insight application license key to **<Code Insight_ROOT_DIR>**. See [Activating FlexNet Code Insight](#) for details.
5. Prepare the Central Database on the Core/Scan Engine Server or the dedicated server.
6. Download and install the database driver on the Core/Scan Engine server. See [Installing the Database Driver](#) for details.
7. Configure your database. See [Preparing the Database](#) for details.
8. Run the following database scripts, in the following order. See [Running the Database Setup Scripts](#) for details.

palamida_ddl.sql (This creates the Code Insight central database schema and sd data.)
request_form_short.sql or request_form_long.sql (This is the default request form definitions. Most users select the long request form) or a custom request form provided to you by Code Insight.
reports.sql (This is the default report definitions.)
9. Run any custom scripts provided to you by Flexera Software.

10. Configure the properties files. See [Editing Configuration \(Properties\) Files](#) for details.
 - Core Server
 - Scan Servers
 - Detector
 - Runner
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 Flexera.
13. Start the Tomcat web server. See [Administration: Starting & Stopping Servers](#) for details.
14. On initial startup of FlexNet 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 FlexNet Code Insight](#)
- [Installing the Database Driver](#)
- [Activating FlexNet Code Insight](#)

Choosing an Installation Directory

The following instructions assume that the core/scan engine are located on the same server.

- Select a base directory on the Core/Scan Engine server where FlexNet 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 FlexNet 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 Engine 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 FlexNet 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 FlexNet Code Insight



Task

To install FlexNet 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 application deployed)

Installing the Database Driver



Task

To install the database driver, do the following:

1. Download the appropriate JDBC driver for your database type:
 - **MySQL**—Download **mysql-connector-java-5.1.x-bin.jar** from this location:
<http://dev.mysql.com/downloads/connector/j/5.1.html>

- **Oracle**—Download **ojdbc6.jar** from this location:
<http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>
 - **MS SQL Server**—From the following location, download the driver 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 FlexNet 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 FlexNet Code Insight Using the Installer](#) chapter.)
 - If installing FlexNet 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 FlexNet Code Insight Manually](#) chapter.

Activating FlexNet Code Insight

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



Task

To activate FlexNet 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 FlexNet Code Insight database:

- DBA or System Administrator with elevated privileges
- Creates the database
- Performs backups as necessary
- FlexNet 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 FlexNet 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 FlexNet Code Insight is using (both core and Scan Engine servers). This is often done using the `GRANT PRIVELEGES` command. See the reference manual for your database type for more information.
- Create a FlexNet Code Insight user with Read/Write privileges. Assign the user to the FlexNet 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 FlexNet 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. FlexNet Code Insight recommends that you set the Max Heap size to approximately 80% of available memory. For a 16GB system, the value should be set to 12288m. For 32GB 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 16GB system, the value should be set to 12288m. For 32GB 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 FlexNet 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:

```
-DCode InsightInstallPath=$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 for each Code Insight Core and Code Insight Scan Engine server (IP address) that accepts secure connections. Refer to http://en.wikipedia.org/wiki/HTTP_Secure and <https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html> for more details about HTTPS/SSL. To enable an HTTPS connection over SSL, follow the detailed steps in the `<Code Insight_ROOT_DIR>\<version>\tomcat\https\readme.txt` file.

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 the FlexNet Code Insight Windows 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](#)
- [Adding Custom Phases to the Scan Process](#)

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.



Note • 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 FlexNet 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 FlexNet Code Insight Core server, and replace `<HTTP_PORT>` with the port number of the FlexNet 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 FlexNet 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 FlexNet Code Insight Core server. Replace `<HTTP_PORT>` with the port number of the FlexNet 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 FlexNet Code Insight Scan server. Replace `<HTTP_PORT>` with the port of the FlexNet Code Insight Scan server, typically set to 8888.
 - Replace `<ALIAS>` with the alias of your FlexNet 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 FlexNet Code Insight workspaces. The recommended path is `<Code Insight_ROOT_DIR>\workspaces`.
 - (Optional) Set the `serverFileSystemRoot=` to limit access to directories on the FlexNet Code Insight Scan server.
8. In **scan.properties** (`<Code Insight_ROOT_DIR>\<version>\config\scan.properties`) set the `signaturesDirPath` to your FlexNet Code Insight CL. Point to the directory that contains the pattern files ending in **.dat** and the **bdb** directory.

Installation Options

This section provides details about configuring different email-related and scan engine options.

Configuring Email Services



Task

To configure email services, do the following:

1. Configure **core.properties** (`<Code Insight_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.db.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 FlexNet 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 FlexNet Code Insight Support for further instructions.

Configuring a Manual Proxy Server Connection (Optional)

By default, FlexNet 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 Engine 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



Task

To change scan engine configuration options, do the following:

1. Update **scanEngine.properties** (<Code Insight_ROOT_DIR>\<version>\config\scanEngine) with optional scan server settings.
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 FlexNet 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:

1. 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.

2. 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
```

3. 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 whitespace (' ') 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] 2013 would be defined as Copyright \\[C\\] 2013.

4. 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-
\\b\\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 whitespaces (\s*) to allow for leading and trailing whitespaces, so there is no need to pad your patterns with whitespaces.
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 whitespace (' ') 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 whitespace (' ') 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 engine 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_6.13.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 engine 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

```
<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

```
<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>
```

Adding Custom Phases to the Scan Process

The FlexNet 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 FlexNet 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

FlexNet 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.

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

FlexNet Code Insight generates several versions of the scan engine 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 FlexNet Code Insight electronic updates.:

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

FlexNet 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.

Installing FlexNet Code Insight Using the Installer

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

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

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

Installing FlexNet Code Insight With the Installer



Task

To install FlexNet 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 FlexNet 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_6.13.x.jar` file or enter the following at the command prompt:

```
java -jar codeinsight_6.13.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 and scanner on this computer.
 - **Core**—Configure this computer as the core server.
 - **Scanner**—Configure this computer 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 Engine Servers.

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

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

11. Select your database vendor.

FlexNet 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 FlexNet 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 FlexNet 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 FlexNet Code Insight installation directory.

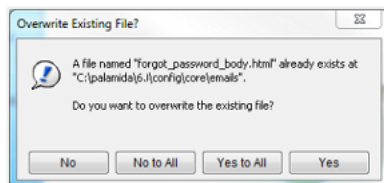
24. In the **Path to Scan Root Directory** field, enter the path to your scan root directory. The web application uses this property to limit the root of the server file system, making it easier to control which files and directories users have access to. Multiple roots are supported, but they must be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

C:\Folder1\Folder2|D:\Folder3



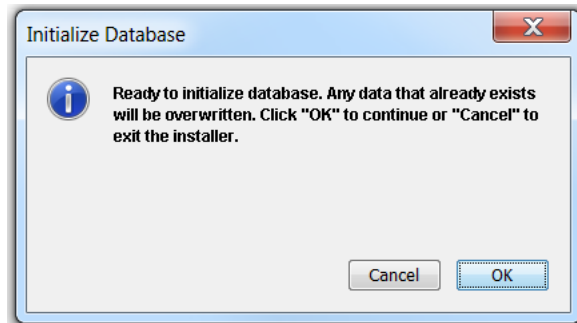
Note • If you do not want to limit access of FlexNet Code Insight users to files and directories, leave this field blank.

25. Click **Next**. The **Pre-Installation Summary** panel appears.
26. Review your FlexNet 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.



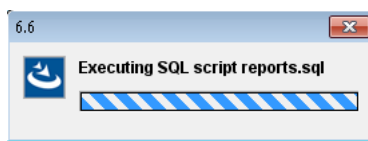
Note • If you are installing into a non-empty directory, there may be files that must be overwritten. Click **Yes to All** to overwrite existing files.

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 FlexNet Code Insight](#).

Installing FlexNet Code Insight Using the Linux Command Line

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



Task To install FlexNet 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	codeInsight.key

Type	File
Database Driver	The JDBC driver that you downloaded based on your database type. See Installing the Database Driver for details.

- Open the command prompt and enter the following to launch the installer:

```
java -jar codeinsight_6.13.x.jar
```

An **Introduction** message appears:

```
=====
Introduction
=====

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

If you have any questions or issues installing this product, please contact:
support@pelamida.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: [ ]
```

- 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
: [ ]
```

- Enter the number of one of the following options and press **Enter**.
 - 1 Standalone**—Both core and scanner on this computer.
 - 2 Core**—Configure this computer as the core server.
 - 3 Scanner**—Configure this computer as 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 Engine 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 FlexNet 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 FlexNet 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.

FlexNet 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 FlexNet 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 9-1: 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 9-2: 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 9-3: 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 FlexNet 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 FlexNet 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 FlexNet 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 FlexNet 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 FlexNet 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 FlexNet Code Insight](#).

Launching FlexNet Code Insight

This section explains how to launch FlexNet Code Insight.

- [Starting and Stopping the Server](#)
- [Logging into FlexNet 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 FlexNet Code Insight server, do the following:

1. To start Tomcat, enter the following:
`<CODE_INSIGHT_ROOT_DIR>\6.13.x\tomcat\bin\startup.bat` (or `startup.sh` if running on Linux)

To stop Tomcat, enter the following:
`<CODE_INSIGHT_ROOT_DIR>\6.13.x\tomcat\bin\shutdown.bat` (or `shutdown.sh` if running on Linux)
2. Open a FlexNet 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 FlexNet Code Insight Core Server.

Logging into FlexNet Code Insight

To log into FlexNet Code Insight, log into the server.



Task

To log into FlexNet 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 FlexNet 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 FlexNet Code Insight plugin for Jenkins lets you start an FlexNet 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 FlexNet 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 FlexNet 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

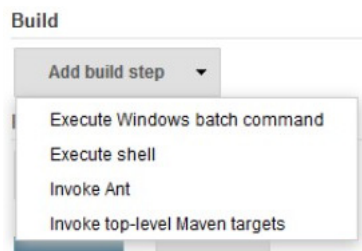
To set up the Jenkins plugin to work with your build, perform the following steps.



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 **FlexNet 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 **FlexNet Code Insight Scan**. The **FlexNet 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 **FlexNet 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 FlexNet 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 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 FlexNet 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:

Save Cancel

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
------	-------	-------------	---------

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

Add Token

* Name: Client Machine #1

* Token Validity: ☐ Never Expires ☒ Expires On 01/27/2016

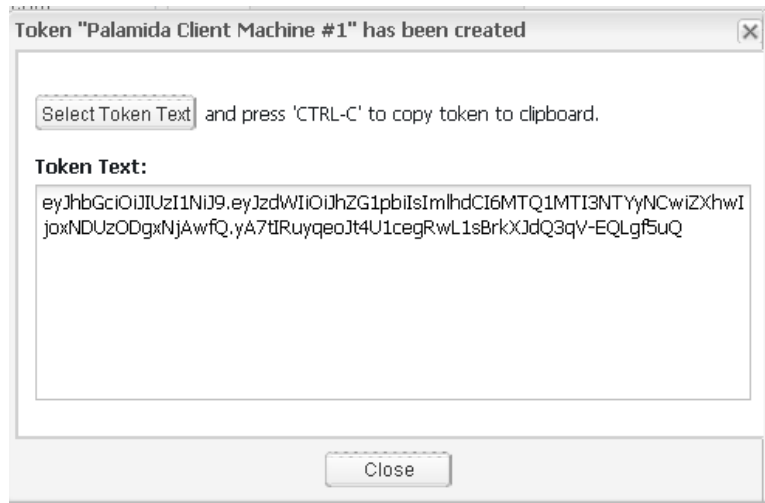
Save Cancel

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 FlexNet 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 FlexNet Code Insight instance to another, including transitioning to newer versions of FlexNet 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_dataflag`.
- 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**: contains indicators, contains exact file matches, contains license matches, copyright holder, etc.
- **File reviewed status**: reviewed status is 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 FlexNet Code Insight. To export from FlexNet Code Insight 6.10.4, for example, you must use the older version of the export script and then import to the latest version (6.13.2) 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 FlexNet Code Insight:

- Export from 6.10.3/6.10.4/6.11 using the scripts designed for those versions, and then import into FlexNet Code Insight 6.13.2 using the 6.13.2 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 6.13.2 by following the standard migration process. Use the FlexNet Code Insight 6.13.2 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 FlexNet 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 machine) 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 engine 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 engine 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 engine machines from config on the indicated server. Export all workspaces from all scan engine machines. This option overrides -server_all, -server, -scan_server, -team, -project, and -workspace options. For instance,

```
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 machine 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. FlexNet 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 FlexNet 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 FlexNet 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 and attribute, set it equal to blank. For example:

```
ldap.user.middleNameAttr=
# LDAP server connection settings, url, userName and password
# this user should have read right to be able to access the schema
# LDAP server connection settings, url, userName and password
# this user should have read right to be able to access the schema
#To enable LDAP Authentication set ldap.enabled to true ldap.enabled=true
#To sync LDAP users to Code Insight user lists "ldap.user.sync.enabled=true"
#URL of the LDAP server, for eg. ldap://<ldap_server>:389 ldap.url=ldap://ad.Code Insight.com:389
#Base node of LDAP server, for all the searches base node will be automatically appended. ldap.base=
dc=ad,dc=Code Insight,dc=com
#User name to login to LDAP server ldap.userName=cn=Manager,dc=ad,dc=Code Insight,dc=com
#Password to login to LDAP server ldap.password=secret
#searchBase and searchFilter are used to import users to Code Insight system.
#DO NOT append ldap.base to ldap.searchBase
# Search base where you can s all the desired users. ldap.searchBase=CN=Users
# Search filter to pull only desired users to the Code Insight System, you can use LDAP Query here.
ldap.searchFilter=(&(objectClass=person)(memberOf=CN=Code InsightAppsecGroup,CN=Users,DC=ad,DC=Code
Insight,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 = false
#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=<LDAP_ATTR>
#LDAP user middleName attribute (Optional) ldap.user.middleNameAttr=<LDAP_ATTR>
#LDAP user lastName attribute (Optional) ldap.user.lastNameAttr=<LDAP_ATTR>
#LDAP user business unit attribute (Optional) ldap.user.businessUnitAttr=<LDAP_ATTR>
#LDAP user job title attribute (Optional) ldap.user.jobTitleAttr=<LDAP_ATTR>
#LDAP user location attribute (Optional) ldap.user.locationAttr=<LDAP_ATTR>
LDAP user telephone attribute (Optional) ldap.user.telephoneAttr=<LDAP_ATTR>
#LDAP user fax attribute (Optional) ldap.user.faxAttr=<LDAP_ATTR>
#LDAP user state attribute (Optional) ldap.user.stateAttr=<LDAP_ATTR>
# 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 Code Insight 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 time an LDAP sync
occurs>
```

Configurable LDAP Properties

You can use the LDAP instead of the FlexNet 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 FlexNet Code Insight. The authentication is a three-step process:

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

Data Synchronization

User Metadata

The metadata for each user (name, email, role, etc.) is pulled from LDAP and refreshed in the FlexNet Code Insight database at a regular frequency via a scheduler module running within FlexNet Code Insight. The data synchronization is a one-way pull from LDAP into the FlexNet Code Insight database. This action overwrites the existing data in the FlexNet 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 FlexNet 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

FlexNet 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:**

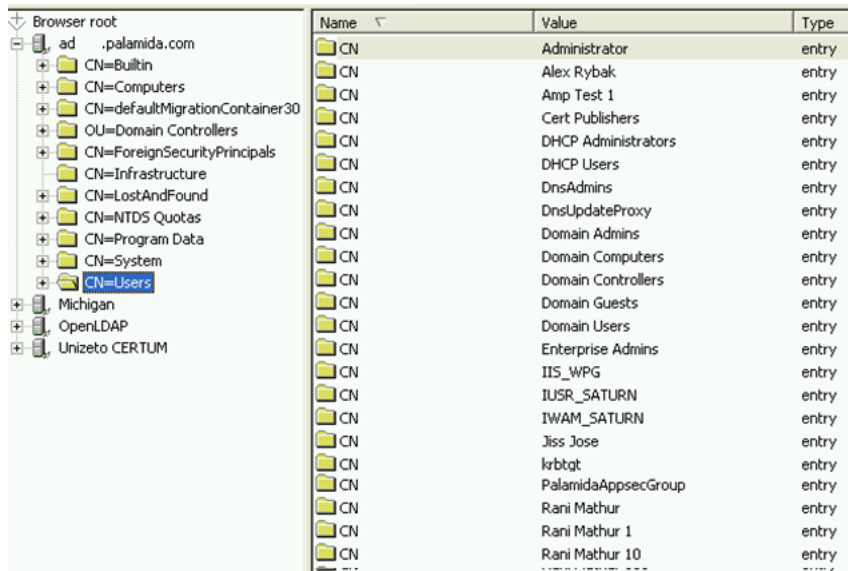
1. Add and define the following properties in **core.ldap.properties** (<FNCL_ROOT_DIRECTORY>/config/core/core.ldap.properties):
 - ldap2.url
 - ldap2.base
 - ldap2.userName
 - ldap2.password
 - ldap2.anonymous
 - ldap2.read.timeout
 - ldap2.searchBase
 - ldap2.searchFilter
2. Restart the server.
3. 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 FlexNet Code Insight, configuring searchBase and searchFilter entries properly is important. searchBase is typically the root node under which you can s all the desired users. searchFilter allows for an LDAP query based on user attributes. We recommend creating a FlexNet 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 14-1: Sample LDAP Tree

Default Role Assignment

The option to set user roles when pulling LDAP users into FlexNet 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 FlexNet Code Insight Administrator must make the change in the Web UI. See the “Administration Menu: User Options” section of the FlexNet Code Insight User Guide for more information.

Set Up a User List Sync

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

The screenshot shows the 'User List Details' page. The 'Name' field is 'Developers' (labeled 'The userlist_name'). The 'Description' field contains 'Developers who will be requesting to use open-source software.' The 'User List Type' is set to 'Requester'. The 'LDAP Query' field contains 'The description of the LDAP query in words' (labeled 'Value of Developers.ldap.description'). Below this, the 'Users' section shows a search bar and a table of results. The table has columns for 'login', 'First Name', 'Last Name', and 'Email'. Two users are listed: 'rmathur1000' and 'rmathur1001', both with email 'rmathur@palamida.com'. The table is labeled '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 FlexNet 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.

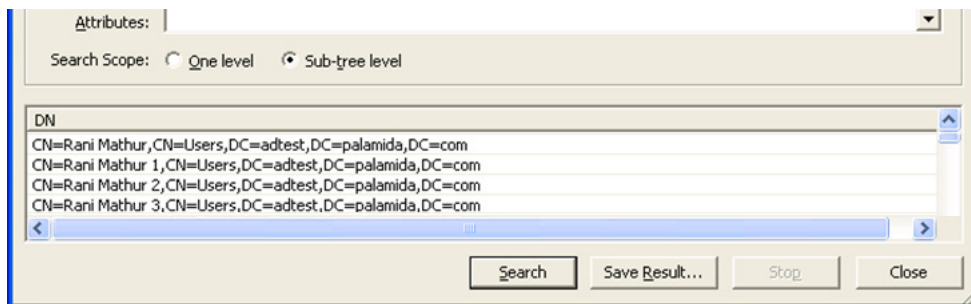
Q: What if I can't connect to the LDAP server using FlexNet Code Insight?

A: Verify your LDAP configuration using LDAP Browser or the FlexNet Code Insight LDAP test groovy script located in the `<Code Insight_ROOT_DIR>\<version>\scriptRunner\scripts` directory. Visit FlexNet Code Insight Customer Community at <https://flexeracomunity.force.com/customer/CCContactSupport> 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 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 FlexNet Code Insight Scan servers:

- The codebases are too large to scan on a single server. You can use additional FlexNet Code Insight Scan servers to distribute the load.
- The system experiences a bottle-neck by a single scan queue, and using additional FlexNet Code Insight Scan servers enables you to perform parallel scans.

Configuration Details

Configuring additional FlexNet Code Insight Scan servers is similar to configuring the first FlexNet Code Insight Scan server with a few exceptions. For each additional Scan Engine server, do as follows:

1. Register each additional FlexNet 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 FlexNet 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 FlexNet Code Insight Scan server. The configuration changes required to configure additional FlexNet 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 FlexNet Code Insight Scan server. Copy this file from the FlexNet Code Insight Core server. This is the only file required in the core directory on each remote FlexNet 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 FlexNet Code Insight Core Server URL.
 - b. Set `serverUrl` to point to the URL of the FlexNet 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 FlexNet Code Insight Scan server
4. Copy the FlexNet Code Insight Key to each FlexNet Code Insight Scan server. Contact Flexera Software if you need to obtain a key.



Note • FlexNet Code Insight Analyzer requires a new key with data services enabled.

5. Copy the FlexNet Code Insight Data libraries to each additional FlexNet Code Insight Scan server. See [Installing the Compliance Library](#) for additional information.
6. Copy Tomcat to each additional FlexNet Code Insight Scan server. See [Extracting Application Files](#) and [Configuring the Tomcat Web Server](#) for additional information.
 - a. Only the FlexNet Code InsightScanEngine directory (exploded war file) is required in the Tomcat web applications directory (<Code Insight_ROOT_DIR>\<version>\tomcat\webapps).
 - b. Remove the FlexNet 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](#).

Visit the FlexNet Code Insight Customer Community at <https://flexeracommunity.force.com/customer/CCContactSupport> for assistance if you need to generate custom source code fingerprints on multiple servers.


Configuring FlexNet Code Insight using MySQL Commands

The following table describes common MySQL commands and how to use them to configure FlexNet Code Insight.

Table 16-1 • Common SQL Commands

Command	Code
Create Database	<pre>CREATE DATABASE <the database name>;</pre> <p>For example:</p> <pre>CREATE DATABASE Code Insight;</pre> <p>Make sure the user has permissions to create a new database.</p>
Create User and Grant All Privileges	<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>
Default User and Password	<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>
Execute the SQL Scripts	<pre>mysql -h <the host> -u <the user> -p <the database> < <path to script file></pre> <p>You can use this command to execute the Code Insight SQL scripts from command line.</p>
List All Database for MySQL Instance	<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 16-1 • Common SQL Commands

Command	Code
Log into MySQL	<pre>mysql -h <the host> -u <the user> -p <the database></pre> <div></div> <div>Note • 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 the FlexNet Code Insight Windows Service

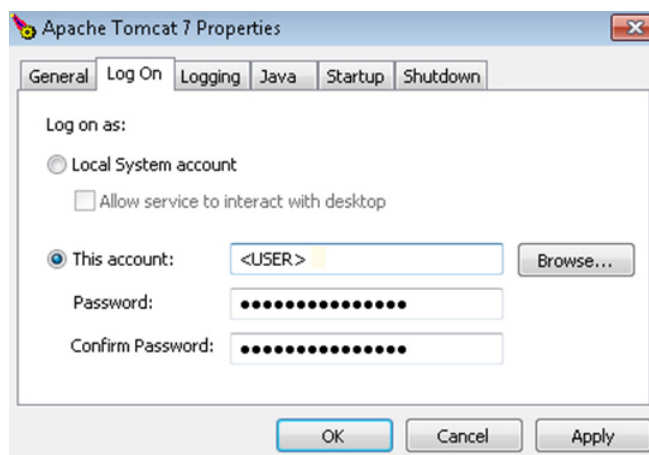
This chapter explains how configure FlexNet Code Insight to run as a Windows service.



Task

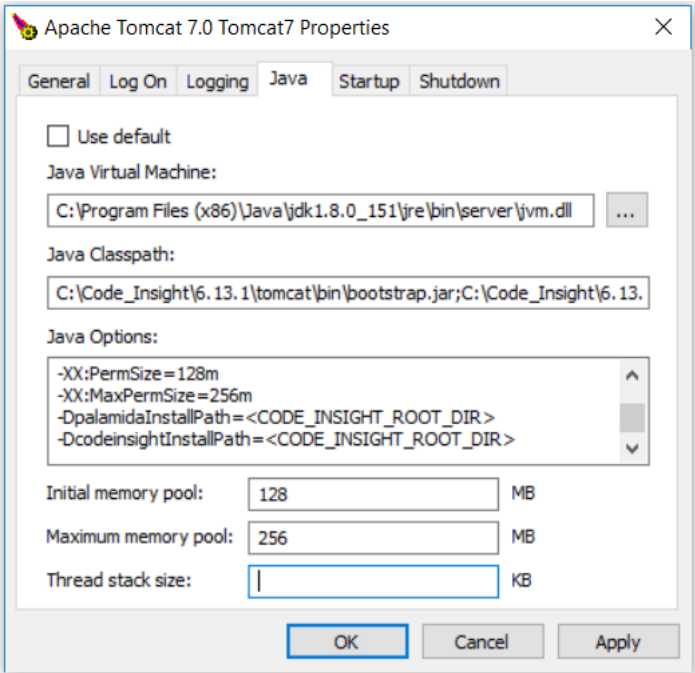
To configure the FlexNet Code Insight Windows service, do the following:

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.
-DpalamidaInstallPath=<CODE_INSIGHT_ROOT_DIR>	Set to your FlexNet Code Insight root directory, such as C:\Code_Insight\6.x .
-DcodeinsightInstallPath=<CODE_INSIGHT_ROOT_DIR>	Set to your FlexNet 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.

Using SCM Connectors

FlexNet 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 FlexNet Code Insight Scan server as this is necessary for FlexNet Code Insight to connect and sync to an SCM repository.



Task

To verify that the SCM client is installed and available to FlexNet Code Insight, do the following:

1. Open a Command Prompt and navigate to the FlexNet 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 FlexNet Code Insight:



Note • Download site links are subject to change.

SCM	Client	Cost	Download Site
Subversion (SVN)	TortoiseSVN	Free	http://tortoisesvn.tigris.org/
Git	Git	Free	http://git-scm.com/downloads
TFS	Team Explorer Everywhere	Free	https://github.com/Microsoft/team-explorer-everywhere/releases
Perforce	Perforce Visual Client (P4V)	Free Trial	https://www.perforce.com/products/helix-core-apps/command-line-client
Clearcase	Rational ClearCase	\$	http://www-03.ibm.com/software/products/en/clearcase



Note • As of FlexNet Code Insight version 6.11.3, legacy client *Team Explorer* is no longer supported for use with FlexNet Code Insight. For compatibility with FlexNet 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 FlexNet 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 FlexNet Code Insight version 6.11.3, legacy client Team Explorer is no longer supported for use with FlexNet Code Insight. For compatibility with FlexNet 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.

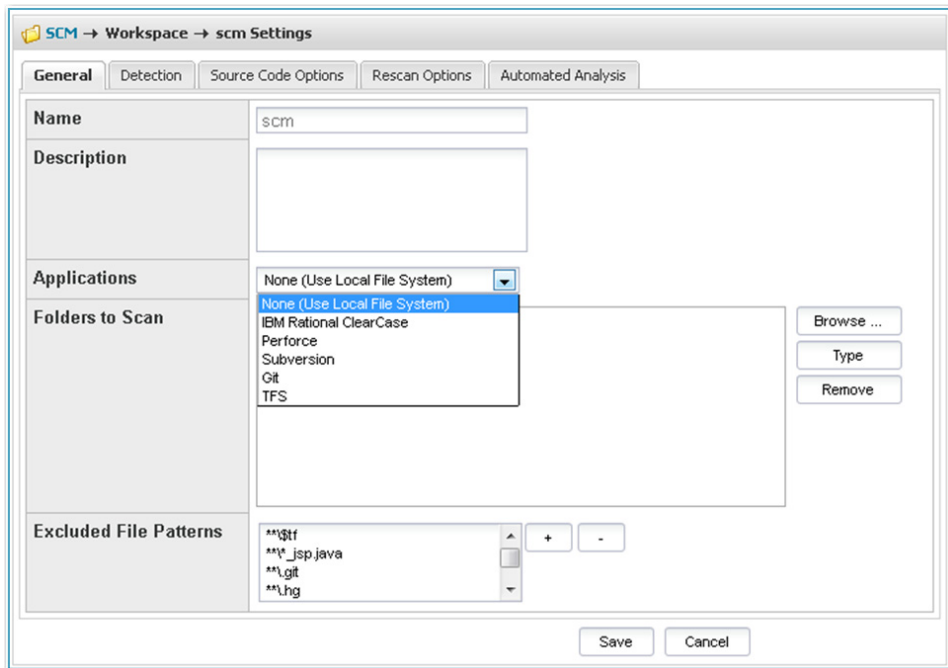


Note • Your SCM may require other environment variables to be set. Consult your SCM documentation for more information.

Workspace Settings

If integration with ClearCase, Perforce, Subversion, Git, or TFS is enabled in the scan engine SCM configuration file (<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

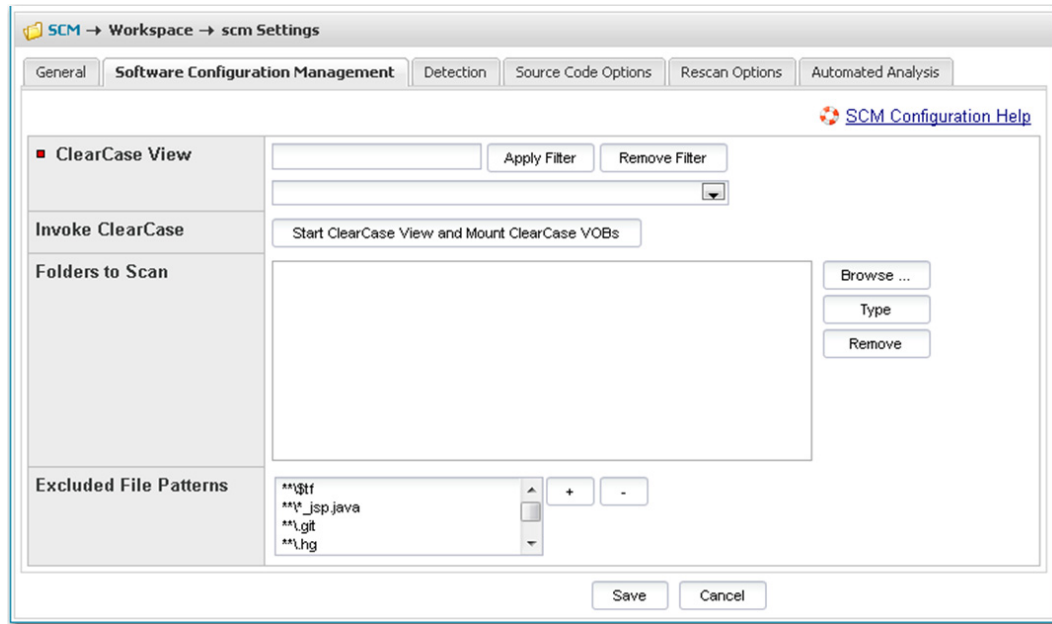
To configure the current workspace to use IBM Rational ClearCase for managing the codebase, perform the following steps.



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 FlexNet Code Insight workspace.
 - **ClearCase VOBs**—Select the ClearCase VOBs to associate with the FlexNet 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: p4: //<server>:<port>/<depot>/<project>
Perforce Username/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"> • Latest Revision • Specific Revision • Changelist • Date • Label

Option	Description
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. 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
Perforce URL	p4://<server>:<port>/<depot>/<project>
SCM Destination Folder	/<sync location>/
Folders to Scan	/<sync location>/<depot>/<project>/<subdirectory>/<folder1>
	
	Note • In this case, only folder 1 will be synced and scanned.

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
Subversion URL	Enter the repository URL in the following format: <code>http://<server>/<svnroot>/<repository></code>
Subversion Username/Password	Enter the username/password to create an authenticated connection to Subversion. In some cases for Subversion, anonymous access may be allowed in which case, you can leave both fields blank.
Subversion Update To	Select the update to directive and provide the criteria if necessary. The following options are available: <ul style="list-style-type: none"> • Latest Revision • Specific Revision
SCM Destination Folder	Select the folder into which the codebase should be copied on the scan server. You can create new subdirectories 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. 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
GitHub Example	<code>http://github.com/myacct/Spoon-Knife.git</code>
Bitbucket Example	<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
GitHub Example	<code>https://myacct:password@github.com/myacct/Hello-World.git</code>
Bitbucket Example	<code>https://myacct:password@bitbucket.org/myacct/bb101repo.git</code>

SSH Authentication

This section describes SSH authentication between a system running FlexNet 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 FlexNet 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	<code>id_rsa_github_test1</code>	<code>id_rsa_github_test1.pub</code>
Bitbucket	<code>id_rsa_bitbucket_test1</code>	<code>id_rsa_bitbucket_test1.pub</code>

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	<code>github.com</code>	<code>bitbucket.org</code>
User	<code>git</code>	<code>git</code>
HostName	<code>github.com</code>	<code>bitbucket.org</code>
PreferredAuthentications	<code>publickey</code>	<code>publickey</code>
IdentityFile	<code>~/.ssh/id_rsa_github_test1</code>	<code>~/.ssh/id_rsa_bitbucket_test1</code>

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
Host	mygithub_01	mygithub_02
User	git	git
HostName	github.com	github.com
PreferredAuthentications	publickey	publickey
IdentityFile	~/.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 FlexNet Code Insight (Palamida) Account

The tasks involved in setting up a FlexNet 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 FlexNet 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 FlexNet 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
Host	github.com	gitssh-https
User	git	git
Port	443	443
HostName	ssh.github.com	ssh.github.com
PreferredAuthentications	publickey	publickey
IdentityFile	~/.ssh/id_rsa_github_test1	~/.ssh/id_rsa_github_test1
URL	git@github.com:account/ repository.git	git@gitssh-https:account/ repository.git

Git Workspace Configuration

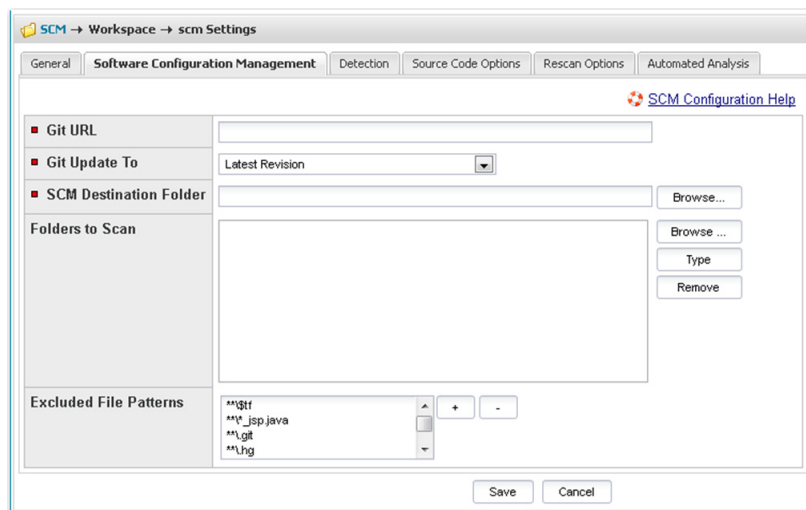
To configure the current workspace to use Git for managing the codebase, perform the following steps.



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:
	Anonymous HTTP http://server/account/repository.git https://server/account/repository.git
	Authenticated HTTP http://account:password@server/account/repository.git https://account:password@server/account/repository.git
	SSH git@server:account/repository.git ssh://server/account/repository.git
	SSH over HTTPS (GitHub only) git@server:account/repository.git ssh://server/account/repository.git
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"> • Latest Revision • Specific Tag
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
TFS URL	Enter the repository URL in the following format: http://<server>:<port>/<tfsroot>/<collection>/<project>
TFS Username/Password	Enter the username/password to create an authenticated connection to the TFS repository.
TFS Update To	Select the update to directive and provide the criteria if necessary. The following options are available: <ul style="list-style-type: none"> • Latest Changelist • Specific Changelist • Label
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.

Option	Description
Excluded File Patterns	Define any file patterns that are to be excluded from the scan.

- 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.

Project Copy

General Options | Custom Copy Options | **Workspace Options**

Pre-Copy Options ☒ Compare Source and Target File Paths (only files that are marked as reviewed, tagged, or associated with groups will be compared)

Workspace #1

Workspace Name

Source: tfs_test

Target:

Workspace Software Configuration Management Settings [SCM Configuration Help](#)

Source:

TFS URL: http://127.0.0.1:8080/tfs/DefaultCollection/ePortal-1.3

TFS Username: palamida

TFS Sync To: Latest Revision

SCM Destination Folder: C:\Cleanroom\tfs_repo\

Target:

TFS URL:

TFS Username:

TFS Password: ☐ Use Existing Password

TFS Sync To: Latest Changeset

SCM Destination Folder:

Workspace Folders to Scan

Source: C:\Cleanroom\tfs_repo\

Target:

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 FlexNet 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 FlexNet 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 FlexNet Code Insight.

Performing the Backup

To perform a backup, perform the following steps.



Task

To perform a backup, do the following:

1. The FlexNet Code Insight Core and all Scan Engine 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 FlexNet 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 FlexNet 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	http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html
MySQL Backup utility	http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html

- **For Oracle**, refer to the following for detailed instructions:
http://www.orafaq.com/wiki/Oracle_database_Backup_and_Recovery_FAQ
6. When the backup is completed, start the FlexNet 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 FlexNet Code Insight Core and all Scan Engine 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
mysqlimport program	http://dev.mysql.com/doc/refman/5.7/en/mysqlimport.html
MySQL Administrator Client Restore utility	http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html

- **For Oracle**, refer to http://www.oraFAQ.com/wiki/Oracle_database_Backup_and_Recovery_FAQ for detailed instructions.
6. Once the restore is completed, you may start the FlexNet 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 FlexNet Code Insight installation:

- Can I use the installer to update an existing installation of FlexNet 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 FlexNet 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.