



Key Performance Indicator			
Line Counting			
Lines of Code	481219	↗	✓
Source Lines Of Code	401326	↗	✓
Effective Lines Of Code	325196	↗	✓
Cyclomatic Complexity	12265	↗	✓
Comment Rate	16 %	↘	✗
Decision Making			
Business Value	1912	→	I
Technical Debt	3626	↗	I
Maturity Index	65 %	↘	C
Stability Index	84 %	↗	B
Reusability Index	44 %	↘	D

Square 15-A-SP2

Installation Checklist

Reference : SIM_QUICK_Square
Version : 15-A-SP2
Date : 07/10/2015

Installation Checklist

Copyright © 2015 Squoring Technologies

Table of Contents

1. Introduction	1
1.1. The Squore Ecosystem	1
2. Preparing your Server for Squore	2
2.1. Installation Prerequisites	2
2.1.1. Supported Operating Systems	2
2.1.2. Browser Compatibility	2
2.1.3. For All Systems	2
2.1.4. Obtaining a Licence File	3
2.1.5. Packages for Windows	3
2.1.6. Packages for Linux	3
2.2. Third-Party Plugins and Applications	4
2.3. Requirements for LDAP Integration	5
2.4. Requirements for TeamForge Integration	6
3. Preparing your Client for Squore CLI	7
3.1. Installation Prerequisites	7
3.1.1. Supported Operating Systems	7
3.1.2. For All Systems	7
3.1.3. Packages for Windows	7
3.1.4. Packages for Linux	7
3.2. Third-Party Plugins and Applications	9

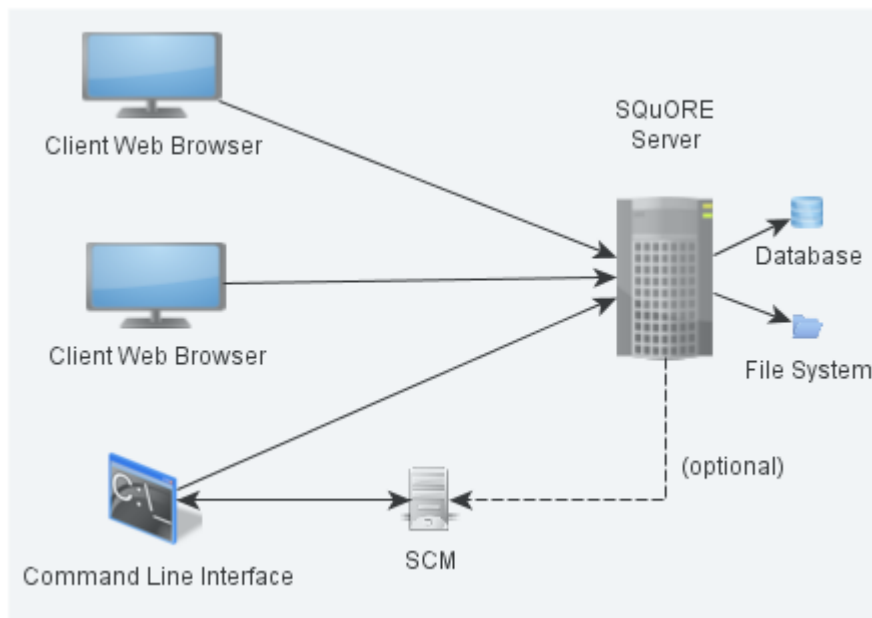
1. Introduction

This document is a checklist and FAQ sheet you can use to make sure that your environment is ready for installing Squore. The first chapter covers common pre-requisites and concepts of the Squore architecture. Each of the following chapters covers more specific topics that you only need to read about if you are interested in using the proposed solutions.

1.1. The Squore Ecosystem

Squore is based on a traditional 3-tier architecture consisting of:

- a database and a data folder for storing project data
- an application server
- a client front-end accessible through a Web Browser
- a Command Line Interface (Squore CLI) to interact with the server from a client machine



The Squore Architecture

As shown in the schema above, Squore Server can provide analysis results to clients without having access to any source code, in scenarios where the analysis is carried out on a client machine with access to the SCM repository, as is the case in most Continuous Integration environments.

If you are planning to access source code hosted in a Subversion, Git, ClearCase, CVS or Synergy repository, a command line client for this repository must be available on the machine where the Squore analysis is carried out. For complete information about all installation pre-requisites, consult Section 2.1, "Installation Prerequisites".

Squore allows analysing source code in the following programming languages: ABAP, Ada, C, COBOL, C++, C#, Fortran 77, FORtran 90, Java, JavaScript, Lustre, PHP, PL/SQL, Python, T-SQL, Visual Basic .NET, XAML.

2. Preparing your Server for Squore

This chapter only covers how to ensure that your environment can be prepared for a Squore installation. For actual instructions on how to install Squore itself, refer to the full Installation and Administration Guide.

2.1. Installation Prerequisites

2.1.1. Supported Operating Systems

The following is a list of the officially supported operating systems for Squore Server:

- CentOS 6
- Fedora 19
- RedHat EL 5
- RedHat EL 6
- SuSe Linux 11.1
- Ubuntu Server 14.04
- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008 R2

2.1.2. Browser Compatibility

Squore is compatible with many browsers. The following is the list of officially supported browsers.

- **Google Chrome 24 and up**
- **Mozilla Firefox 17 and up**
- **Apple Safari 6 and up**
- **Microsoft Internet Explorer 8 and up**

2.1.3. For All Systems

For a successful installation of Squore, you will need:

- The latest version of the Squore installer, which can be downloaded from http://support.squoring.com/download_area.php
- A user account with system administrator privileges
- The Oracle Java Development Kit version 1.6.0_45 (64 bit) or higher

Warning

- Java 1.8 is not supported.
- It is technically possible to run Squore using a 32-bit JDK, however this will limit the memory available to 1GB of RAM to run the application, which will result in poor performance. If you still want to attempt such an installation, consult the troubleshooting page at http://openwiki.squoring.com/index.php/Running_Squore_On_A_32-bit_Java_Installation
- At least 2 GB of space available on the disk for a full installation
- At least 8 GB of RAM on the server
- A valid licence file (optional, since the licence file can be added after installation)

Tip

Keep in mind that the requirements above are the strict minimum. In production, Squore Server generally runs on a dedicated machine with a multi-core processor and 8 to 12GB of RAM. Squore reserves 25% of the available RAM of the machine to the database and another 25% to the server. External processes (like CheckStyle or Findbugs) running on the same machine as Squore may add to the amount of RAM required for analysing source code. Linux is known to offer better performances than Windows when running Squore.

2.1.4. Obtaining a Licence File

Because a licence is linked to the hardware on which Squore Server is installed, Squoring usually delivers a temporary licence to try out Squore first. This give you time to send out your host-id to our team, who will then issue a permanent licence file. In order to find out what your host-id is, follow these steps:

1. Download the host-id checker from http://support.squoring.com/download_area.php
2. Open a terminal on the machine where you installed or plan to install Squore.
3. Run the command `java -jar squore-hostid.jar`.
4. Send the output to support@squoring.com.

2.1.5. Packages for Windows

A JDK is required. The Windows installer contains the tcl and perl runtimes as well as a portable PostgreSQL installation.

2.1.6. Packages for Linux

On Linux platforms, the following must be installed before installing Squore:

- **Perl** version 5.10.1 or greater including the following extra-modules:
 - Mandatory packages:
 - **Archive::Zip** [module details] [<http://search.cpan.org/~phred/Archive-Zip/lib/Archive/Zip.pm>]
 - **Algorithm::Diff** [module details] [<http://search.cpan.org/~nedkonz/Algorithm-Diff/lib/Algorithm/Diff.pm>]
 - **Date::Calc** (since 14-A-SP2) [module details] [<http://search.cpan.org/~stbey/Date-Calc/lib/Date/Calc.pod>]
 - **DBI** [module details] [<http://search.cpan.org/~timb/DBI/DBI.pm>]
 - **DBD::Pg** [module details] [<http://search.cpan.org/~turnstep/DBD-Pg/Pg.pm>]
 - **Digest::SHA** [module details] [<http://search.cpan.org/~mshelor/Digest-SHA/lib/Digest/SHA.pm>]
 - **HTTP::Request** [module details] [<http://search.cpan.org/~gaas/HTTP-Message/lib/HTTP/Request.pm>]
 - **JSON** [module details] [<http://search.cpan.org/~makamaka/JSON/lib/JSON.pm>]
 - **LWP::UserAgent** [module details] [<http://search.cpan.org/~gaas/libwww-perl/lib/LWP/UserAgent.pm>]
 - **LWP::Protocol::https** [module details] [<http://search.cpan.org/~gaas/LWP-Protocol-https/lib/LWP/Protocol/https.pm>]
 - **Time::HiRes** [module details] [<http://search.cpan.org/~zefram/Time-HiRes/HiRes.pm>]
 - **XML::Parser** [module details] [<http://search.cpan.org/~toddr/XML-Parser/Parser.pm>]
 - Optional packages for working with Microsoft Excel:

- **Spreadsheet::XLSX** [module details] [<http://search.cpan.org/~dmow/Spreadsheet-XLSX/lib/Spreadsheet/XLSX.pm>]
- **Spreadsheet::WriteExcel** [module details] [<http://search.cpan.org/~jmcnamara/Spreadsheet-WriteExcel/lib/Spreadsheet/WriteExcel.pm>]
- **Spreadsheet::ParseExcel** [module details] [<http://search.cpan.org/~jmcnamara/Spreadsheet-ParseExcel/lib/Spreadsheet/ParseExcel.pm>]
- **Spreadsheet::BasicRead** [module details] [<http://search.cpan.org/~gng/Spreadsheet-BasicRead/BasicRead.pm>]
- **HTML::Entities** (since 14-A-SP3) [module details] [<http://search.cpan.org/dist/HTML-Parser/lib/HTML/Entities.pm>]
- **OLE::Storage_Lite::PPS** [module details] [http://search.cpan.org/~jmcnamara/OLE-Storage_Lite/lib/OLE/Storage_Lite.pm]
- **Digest::Perl::MD5** [module details] [<http://search.cpan.org/~delta/Digest-Perl-MD5/lib/Digest/Perl/MD5.pm>]
- **Crypt::RC4** [module details] [<http://search.cpan.org/~sifukurt/Crypt-RC4/RC4.pm>]
- Optional packages for working with OSLC systems:
 - **HTML::TreeBuilder** [module details] [<http://search.cpan.org/~cjm/HTML-Tree/lib/HTML/TreeBuilder.pm>]
 - **HTTP::Server::Simple** [module details] [<http://search.cpan.org/~jesse/HTTP-Server-Simple/lib/HTTP/Server/Simple.pm>]
 - **HTTP::Server::Simple::CGI** [module details] [<http://search.cpan.org/~jesse/HTTP-Server-Simple/lib/HTTP/Server/Simple/CGI.pm>]
 - **WWW::Mechanize** [module details] [<http://search.cpan.org/~ether/WWW-Mechanize/lib/WWW/Mechanize.pm>]
 - **Date::Parse** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Date/Parse.pm>]
 - **Date::Language** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Date/Language.pm>]
 - **Date::Format** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Date/Format.pm>]
 - **Time::Zone** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Time/Zone.pm>]

Tip

If some of these modules are not available as packages on your operating system, use your perl installation's `cpan` to install the modules. Using the OS packages is recommended, as it avoids having to reinstall via `cpan` after upgrading your version of perl.

- **Tcl** version 8.5 or greater,
- **PostgreSQL** version 8.4 including at least the **server** component, and optionally, the **pgAdmin** utility. Note that your system must use a UTF-8 locale for the database creation to be carried out successfully. You can force this by running `export LANG=UTF-8` before installing Squore.
- The **rsync** utility

If you are running on a headless server, `java-1.6.0-openjdk` is not sufficient, as it lacks some fonts to render Squore graphics. This is why using Oracle's JDK is recommended.

2.2. Third-Party Plugins and Applications

End users can run third-party static code analysers or rule checkers that are not shipped with the Squore installer for licencing reasons. In this case, it is necessary to download the extra binaries from http://support.squoring.com/download_area.php and deploy them on the server.

The list of third party plugins to be downloaded separately is as follows:

- Checkstyle 5.6
- CPD 4.2.6
- FindBugs 2.0 or 3.0
- Cppcheck 1.61
- PMD 5.0.5
- Polyspace Export
- Stylecop 4.7

Here is a full example of how to deploy Checkstyle into Squore

1. Download the Checkstyle binary from http://support.squoring.com/download_area.php.
2. Extract the contents of the zip file onto Squore Server.
3. Copy the extracted checkstyle-5.6 folder into <INSTALLDIR>/addons/tools/CheckStyle_auto.
4. Instruct all client installations to synchronise with the server so that they get the newly deployed third-party binaries.

Tip

AntiC and Cppcheck on Linux also require special attention: Cppcheck must be installed and available in the path, and antiC must be compiled with the command:

```
# cd <INSTALLDIR>/addons/Antic_auto/bin/ && gcc antic.c -o antic
```

For more information, refer to the Command Line Interface Manual, which contains the full details about special installation procedures for Data Providers and Repository Connectors.

2.3. Requirements for LDAP Integration

In order to configure Squore to integrate with your LDAP Server, you should make sure that you have access to the following information:

- The address of the LDAP server you want to connect to Squore.
- The section(s) of the directory that contain the users that should be allowed to log into Squore.
- The login and password of a user account allowed to browse the section(s) of the directory mentioned above.
- Basic knowledge of your directory structure. Note that Squore was tested with Microsoft Active Directory on Windows Server 2008 and OpenLDAP on Ubuntu 12.04.

If you need to obtain this information to a system administrator, ask for these details:

- *java.naming.provider.url*: The URL of the directory server.
- *baseCtxDN*: The fixed DN of the context to start the user search from.
- *bindDN*: The DN used to bind against the ldap server for the user and roles queries. This is some DN with read/search permissions on the baseCtxDN and rolesCtxDN values.
- *bindCredential*: The password for the bindDN
- *rolesCtxDN*: The fixed DN of the context to search for user roles. This is required to exist, even though it is not used by Squore at the moment.
- *userCompositeName (optional)*: the field in the LDAP account that Squore will import and user as the user's full name.

- *userMail (optional)*: the field in the LDAP account that Squore will import and user as the user's e-mail address.
- *userOrganizationUnit (optional)*: the field in the LDAP account that Squore will import and user as the user's department.

2.4. Requirements for TeamForge Integration

In order to configure Squore to integrate with TeamForge, you will need to provide the following Teamforge details:

- **TeamForge Server URL (without / at the end)**, for example `http://localhost`.
- **TeamForge Server Name (will appear in Squore export format list)**, for example `TeamForge`.
- **TeamForge artifact default status at export**, for example `Open`.
- **TeamForge artifact default priority at export**, for example `4`.
- **SvnEdge Main Repository URL**, for example `http://localhost/svn`.
- **SvnEdge Viewer (viewvc) URL**, for example `http://localhost/viewvc`.
- **External System Id of SvnEdge (for ex: exsy1001)**, for example `exsy1011`.

The rest of the configuration is carried out on the TeamForge side with data obtained from Squore after you create some projects. You can see this information in the full Installation and Administration Guide

3. Preparing your Client for Squore CLI

This chapter only covers how to ensure that your environment can be prepared for a Squore CLI installation. For actual instructions on how to install Squore CLI itself, refer to the full Command Line Interface.

3.1. Installation Prerequisites

The CLI accepts commands and parameters to communicate with Squore Server. Inside the installation folder, some scripts are provided as examples to create projects, save encrypted credentials to disk, and synchronise the client's configuration with the server.

There are two ways to contemplate the CLI deployment:

1. As a way to analyse code and process data on a client machine and send the results to the server.
2. As a way to instruct the server to carry out an analysis of code and other input data.

3.1.1. Supported Operating Systems

The following is a list of the officially supported operating systems for Squore CLI:

- CentOS 6
- Fedora 19
- RedHat EL 5
- RedHat EL 6
- SuSe Linux 11.1
- Ubuntu Server 14.04
- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008 R2

3.1.2. For All Systems

For a successful installation of Squore CLI, you will need:

- The latest version of the Squore CLI installer, which can be downloaded from http://support.squoring.com/download_area.php
- A user account with system administrator privileges
- The Oracle Java Runtime Environment version 1.6.0_45 or higher
- At least 2 GB of space available on the disk for a full installation
- At least 4 GB of RAM on the machine
- The `java` executable should be in the machine's PATH environment variable.

3.1.3. Packages for Windows

A JRE is required. The Windows installer contains the tcl and perl runtimes needed. It will allow you to obtain the configuration needed to create projects from the server.

3.1.4. Packages for Linux

On Linux platforms, the following must be installed before installing Squore CLI:

→ **Perl** version 5.10.1 or greater including the following extra-modules:

→ Mandatory packages:

- **Archive::Zip** [module details] [<http://search.cpan.org/~phred/Archive-Zip/lib/Archive/Zip.pm>]
- **Algorithm::Diff** [module details] [<http://search.cpan.org/~nedkonz/Algorithm-Diff/lib/Algorithm/Diff.pm>]
- **Date::Calc** (since 14-A-SP2) [module details] [<http://search.cpan.org/~stbey/Date-Calc/lib/Date/Calc.pod>]
- **DBI** [module details] [<http://search.cpan.org/~timb/DBI/DBI.pm>]
- **DBD::Pg** [module details] [<http://search.cpan.org/~turnstep/DBD-Pg/Pg.pm>]
- **Digest::SHA** [module details] [<http://search.cpan.org/~mshelor/Digest-SHA/lib/Digest/SHA.pm>]
- **HTTP::Request** [module details] [<http://search.cpan.org/~gaas/HTTP-Message/lib/HTTP/Request.pm>]
- **JSON** [module details] [<http://search.cpan.org/~makamaka/JSON/lib/JSON.pm>]
- **LWP::UserAgent** [module details] [<http://search.cpan.org/~gaas/libwww-perl/lib/LWP/UserAgent.pm>]
- **LWP::Protocol::https** [module details] [<http://search.cpan.org/~gaas/LWP-Protocol-https/lib/LWP/Protocol/https.pm>]
- **Time::HiRes** [module details] [<http://search.cpan.org/~zefram/Time-HiRes/HiRes.pm>]
- **XML::Parser** [module details] [<http://search.cpan.org/~toddr/XML-Parser/Parser.pm>]

→ Optional packages for working with Microsoft Excel:

- **Spreadsheet::XLSX** [module details] [<http://search.cpan.org/~dmow/Spreadsheet-XLSX/lib/Spreadsheet/XLSX.pm>]
- **Spreadsheet::WriteExcel** [module details] [<http://search.cpan.org/~jmcnamara/Spreadsheet-WriteExcel/lib/Spreadsheet/WriteExcel.pm>]
- **Spreadsheet::ParseExcel** [module details] [<http://search.cpan.org/~jmcnamara/Spreadsheet-ParseExcel/lib/Spreadsheet/ParseExcel.pm>]
- **Spreadsheet::BasicRead** [module details] [<http://search.cpan.org/~gng/Spreadsheet-BasicRead/BasicRead.pm>]
- **HTML::Entities** (since 14-A-SP3) [module details] [<http://search.cpan.org/dist/HTML-Parser/lib/HTML/Entities.pm>]
- **OLE::Storage_Lite::PPS** [module details] [http://search.cpan.org/~jmcnamara/OLE-Storage_Lite/lib/OLE/Storage_Lite.pm]
- **Digest::Perl::MD5** [module details] [<http://search.cpan.org/~delta/Digest-Perl-MD5/lib/Digest/Perl/MD5.pm>]
- **Crypt::RC4** [module details] [<http://search.cpan.org/~sifukurt/Crypt-RC4/RC4.pm>]

→ Optional packages for working with OSLC systems:

- **HTML::TreeBuilder** [module details] [<http://search.cpan.org/~cjm/HTML-Tree/lib/HTML/TreeBuilder.pm>]
- **HTTP::Server::Simple** [module details] [<http://search.cpan.org/~jesse/HTTP-Server-Simple/lib/HTTP/Server/Simple.pm>]
- **HTTP::Server::Simple::CGI** [module details] [<http://search.cpan.org/~jesse/HTTP-Server-Simple/lib/HTTP/Server/Simple/CGI.pm>]

- **WWW::Mechanize** [module details] [<http://search.cpan.org/~ether/WWW-Mechanize/lib/WWW/Mechanize.pm>]

- **Date::Parse** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Date/Parse.pm>]
- **Date::Language** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Date/Language.pm>]
- **Date::Format** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Date/Format.pm>]
- **Time::Zone** [module details] [<http://search.cpan.org/~gbarr/TimeDate/lib/Time/Zone.pm>]

Tip

If some of these modules are not available as packages on your operating system, use your perl installation's cpan to install the modules. Using the OS packages is recommended, as it avoids having to reinstall via cpan after upgrading your version of perl.

→ **Tcl** version 8.5 or greater

3.2. Third-Party Plugins and Applications

If you have deployed some third-party tools on Squore Server, they will automatically be downloaded to your client when you launch the client synchronisation script.

On Linux, some plugins may still need your attention: Cppcheck must be installed and available in the path, and antiC must be compiled with the command:

```
# cd <INSTALLDIR>/addons/Antic_auto/bin/ && gcc antic.c -o antic
```

For more information, refer to the Command Line Interface Manual, which contains the full details about special installation procedures for Data Providers and Repository Connectors.