

---

# Squash TF Documentation

**squashtest**

**Mar 25, 2020**



---

## Contents

---

<b>1</b>	<b>Our development tool</b>	<b>1</b>
1.1	Eclipse Tools . . . . .	1
1.2	Other IDE . . . . .	15
<b>2</b>	<b>SKF Download Page</b>	<b>17</b>
2.1	Squash Keyword Framework . . . . .	17
2.2	Squash Eclipse Toolbox . . . . .	17
2.3	Squash Eclipse Plugin . . . . .	17
2.4	Squash IntelliJ IDEA Plugin . . . . .	17
2.5	Notepad++ Squash UDL . . . . .	18
2.6	Archives . . . . .	18
<b>3</b>	<b>SKF Community</b>	<b>19</b>
3.1	Contribution . . . . .	19
3.2	Developer guideline . . . . .	19
<b>4</b>	<b>Keyword test framework</b>	<b>21</b>
<b>5</b>	<b>Engine with plugins</b>	<b>23</b>



## 1.1 Eclipse Tools

### 1.1.1 Squash-TA Toolbox

#### Eclipse Toolbox - Components

The Squash-TA toolbox 1.10.0 contains the following tools:

- Maven 3.5.0,
- Sahi v51,
- Jailer 7.3.1,
- Eclipse Oxygen 64-bits as well as a preconfigured workspace
  - m2e
  - Squash TA eclipse plugin
  - “Run configurations”.
- An uninstall assistant

The version 1.9.0 contains:

- Maven 3.0.4 (for compatibility with framework versions before 1.9.0),
- A patched version of Maven 3.3.3 (you can find here details about the patch),
- Sahi,
- Selenium-server,
- Jailer,
- Eclipse Mars as well as a preconfigured workspace
  - m2e

- Squash TA eclipse plugin
- “Run configurations” and eclipse preferences to import.
- An uninstall assistant

Before the 1.9.0 version the toolbox used to contain:

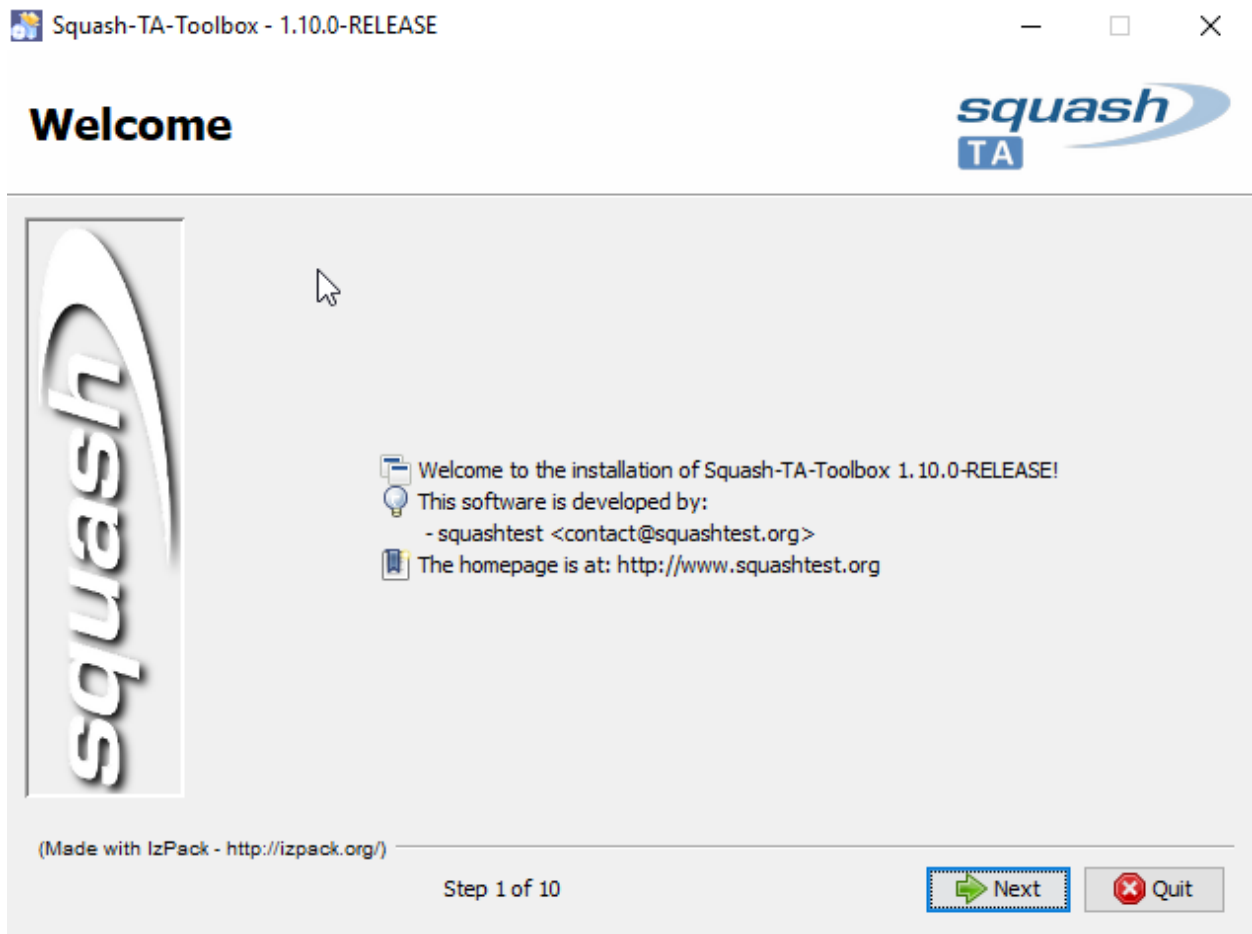
- Maven 3.0.4,
- Sahi,
- Selenium-server,
- Jailer,
- Eclipse Helios as well as a preconfigured workspace
  - m2e
  - Squash TA eclipse plugin
  - “Run configurations” and eclipse preferences to import.
- An uninstall assistant
- open-jdk 6.

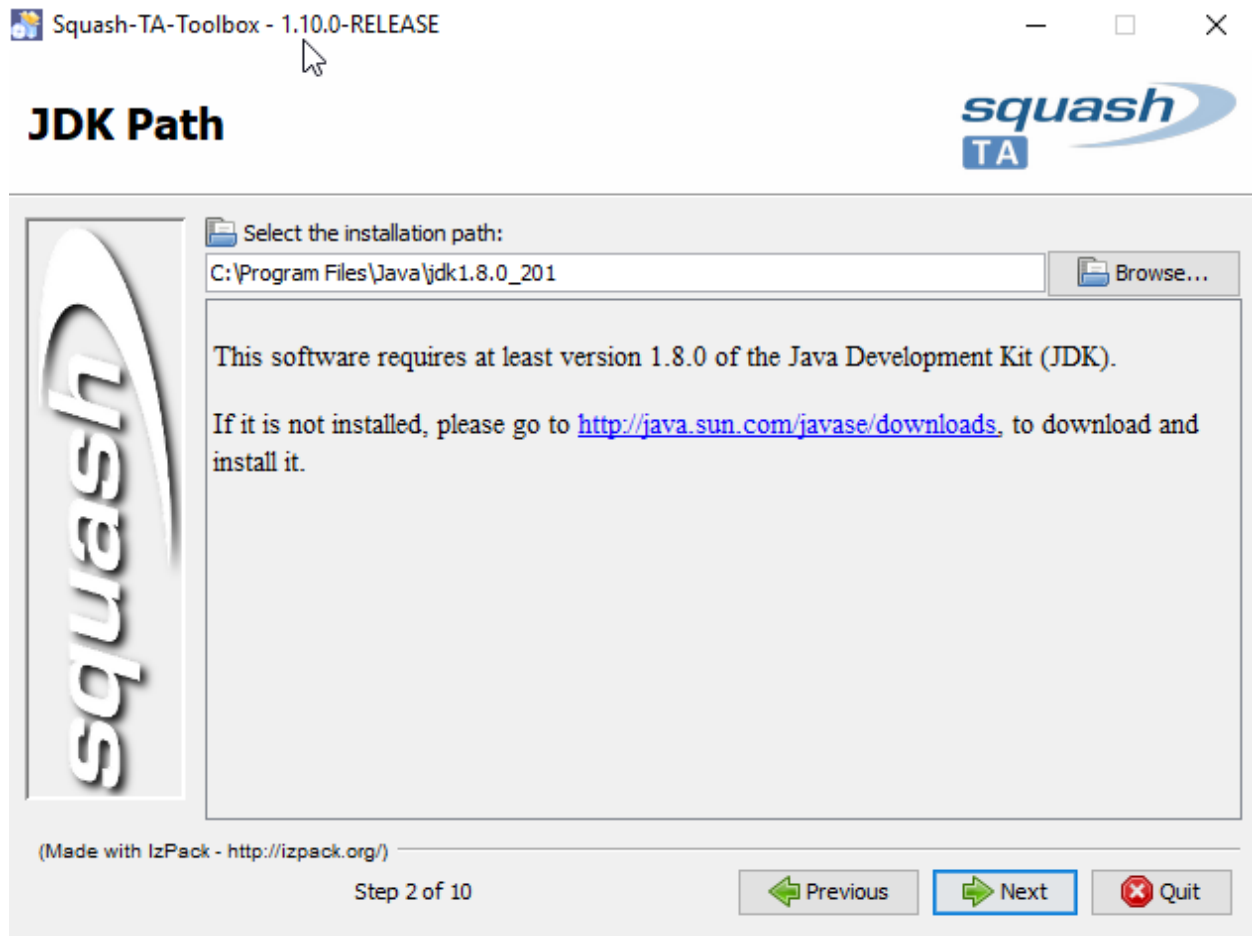
### Install Squash-TA Toolbox

**Prerequisite** : Before starting the installation of SQUASH-TA Toolbox, you must install a JDK between 1.7 and 1.8, both 32 and 64 bits version are compatible. See [here](#) to download it from Oracle website.

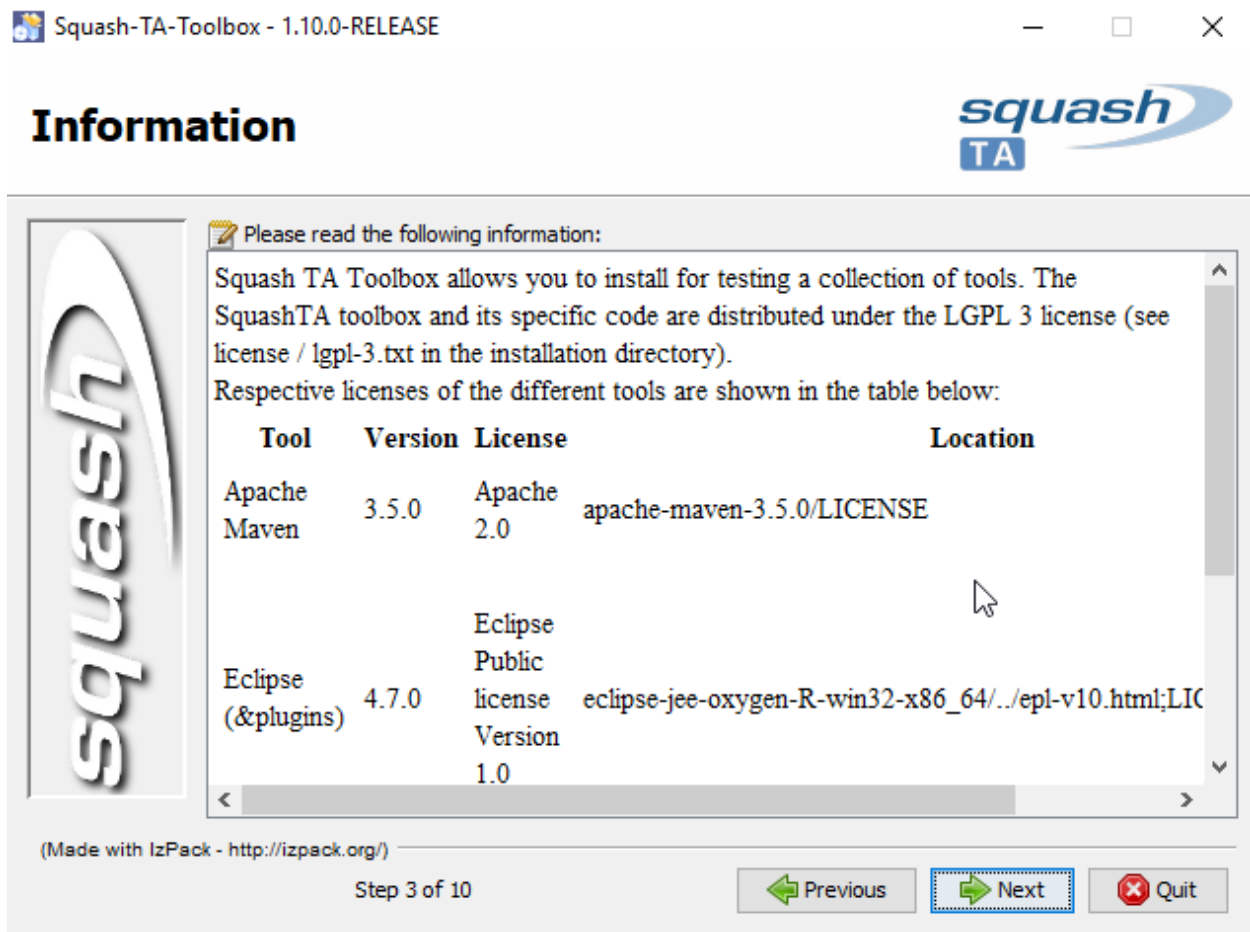
First of all, download the toolbox installer a the the following [link](#).

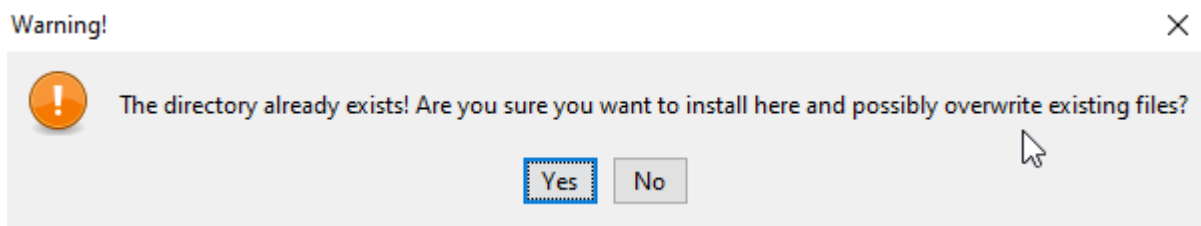
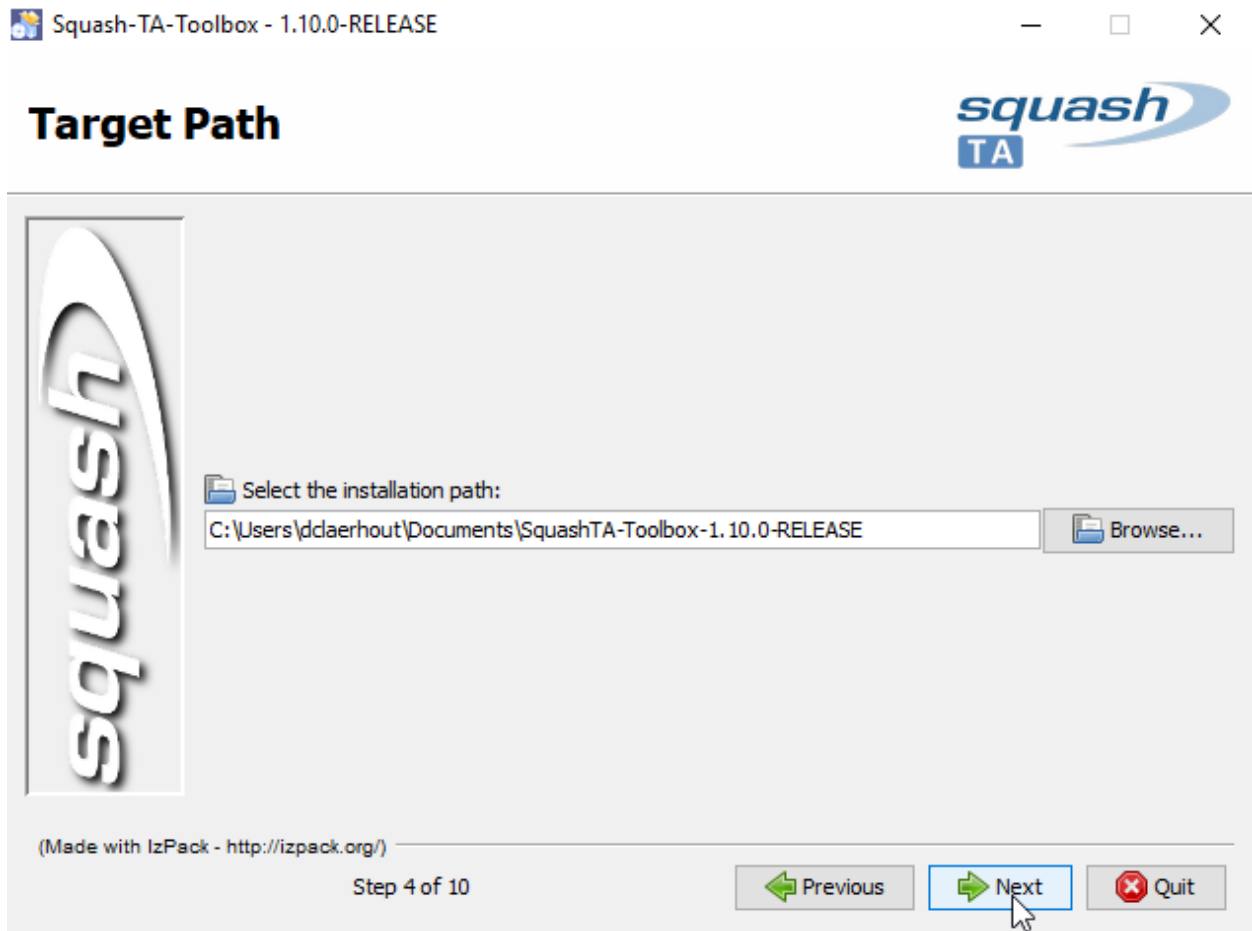
- Launch the installer by double-clicking on it:
- On the first screen, click on **Next**:
- You will then be prompted to indicate the path of your JDK (to do so, click on the **Browse** button. Click on **Next** when it is done:
- On the next screen, you will see all the licenses of the different components of Squash TA Toolbox. Click on **Next**:
- The next step allows you choose your toolbox install location. You can directly type the path in the input field, or click on **Browse** and select your chosen location (you may create your installation folder from there if needed). After choosing the location, click on **Next**:
- If the target folder already exists (even if you’ve just created it), you will get the following alertbox. Click on **Yes** to confirm the location:
- Next, you will have to choose a location to create your new Eclipse workspace for your Squash-TF projects. As for the install location, you may type the path directly or use **Browse** button. If the location you typed does not exist, it will be automatically created. After choosing your location, click on **Next**:

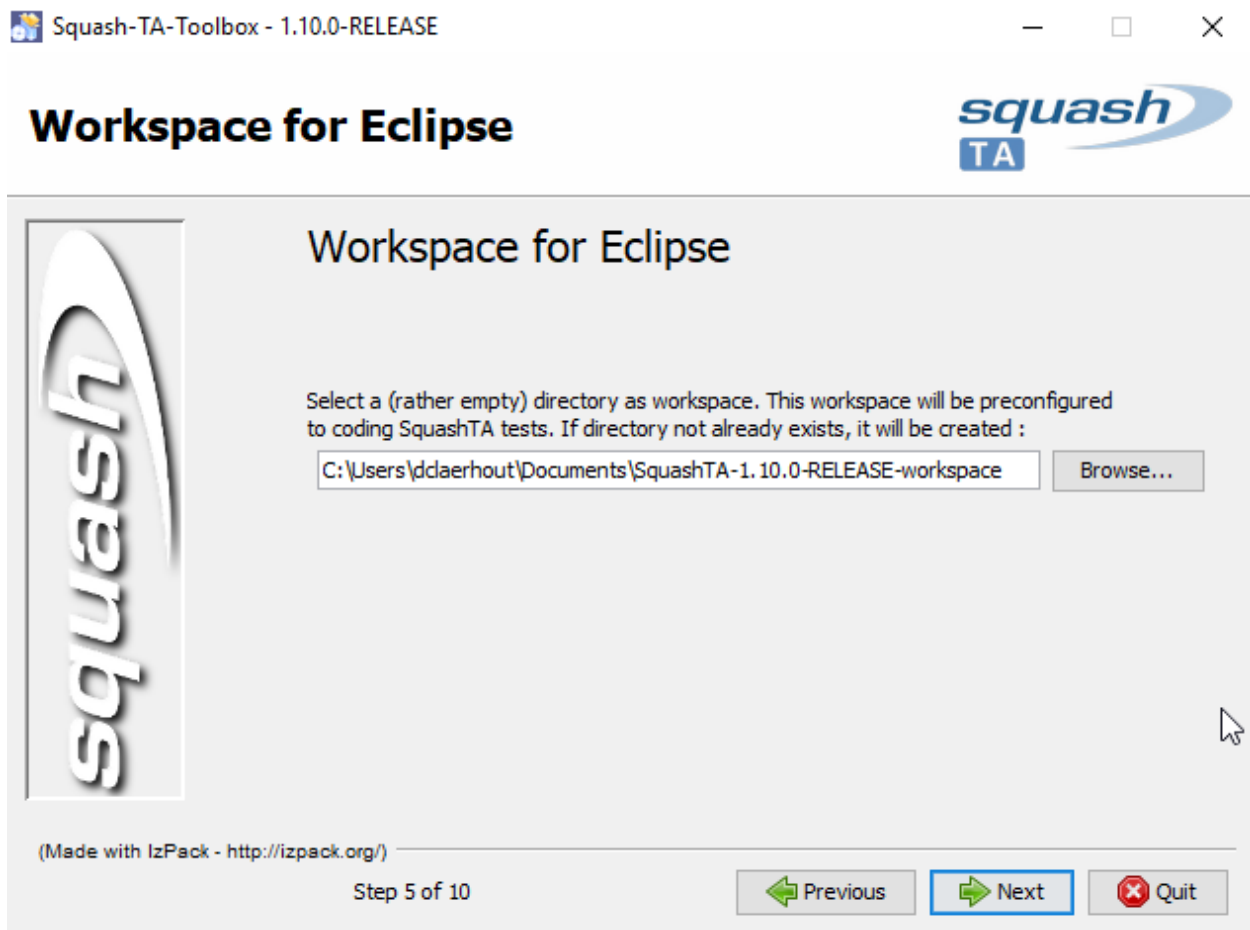




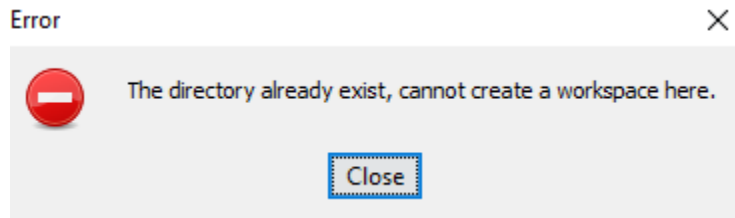




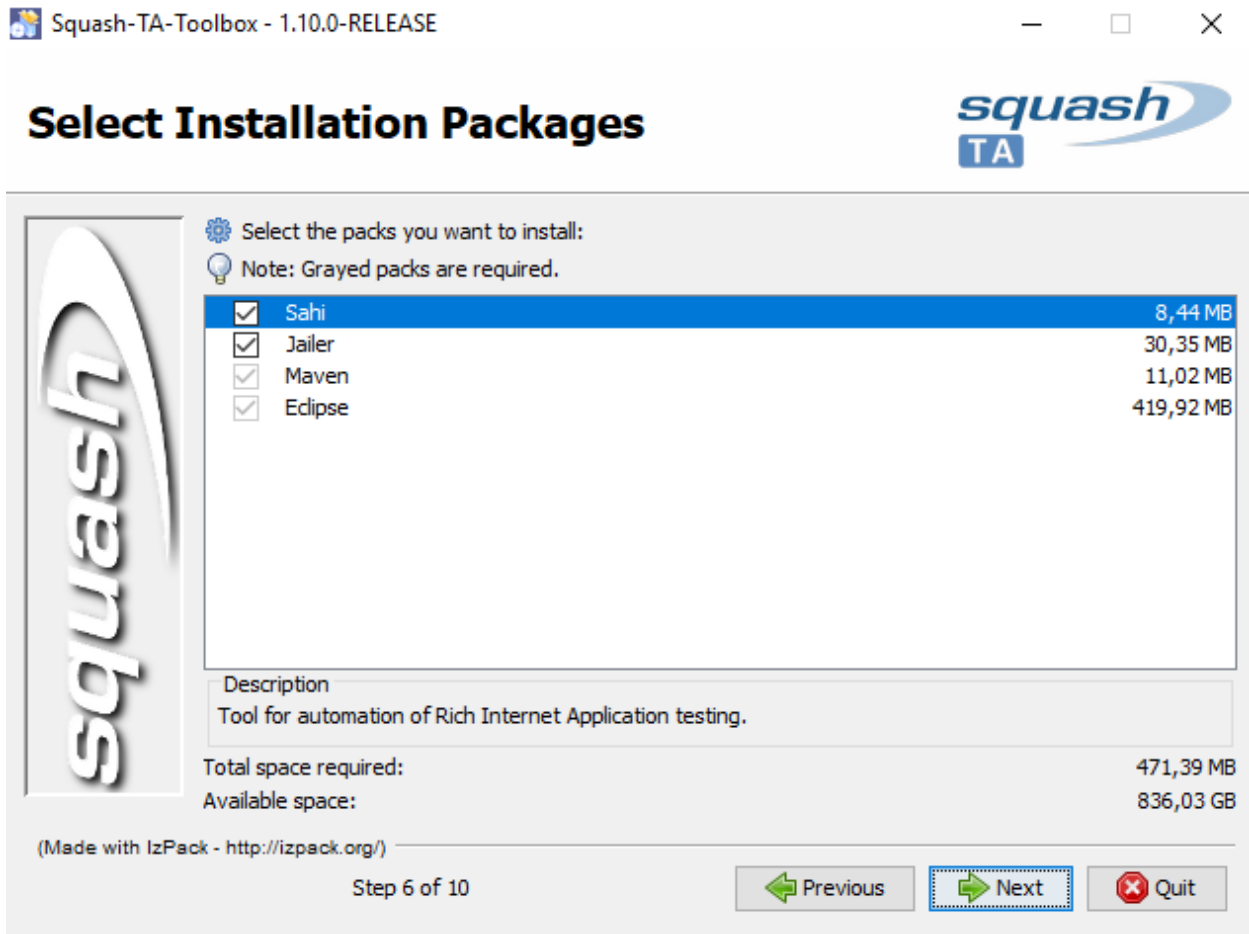




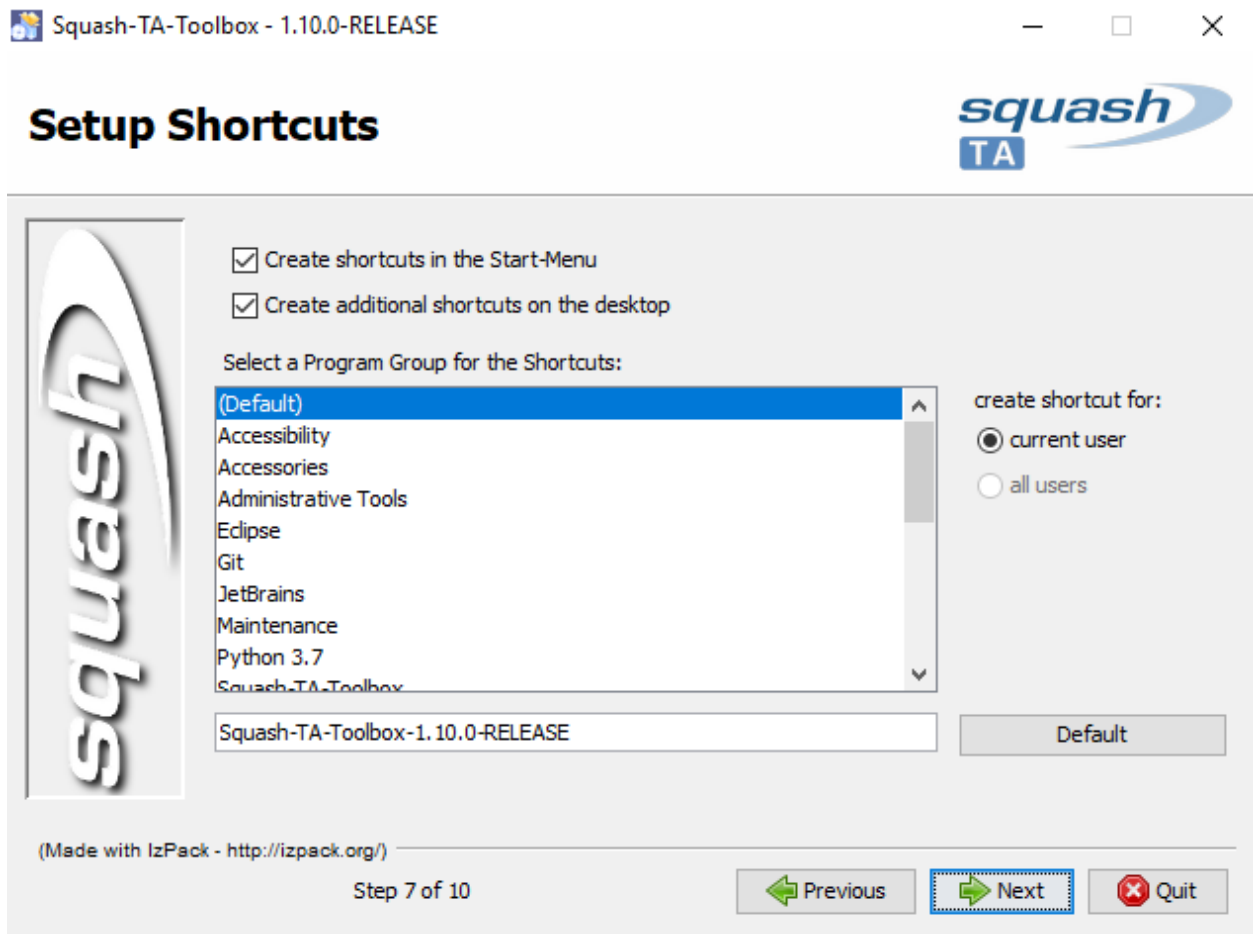
- If the location already exists, you will get the following error message. You can't create the Eclipse workspace in an existing location:

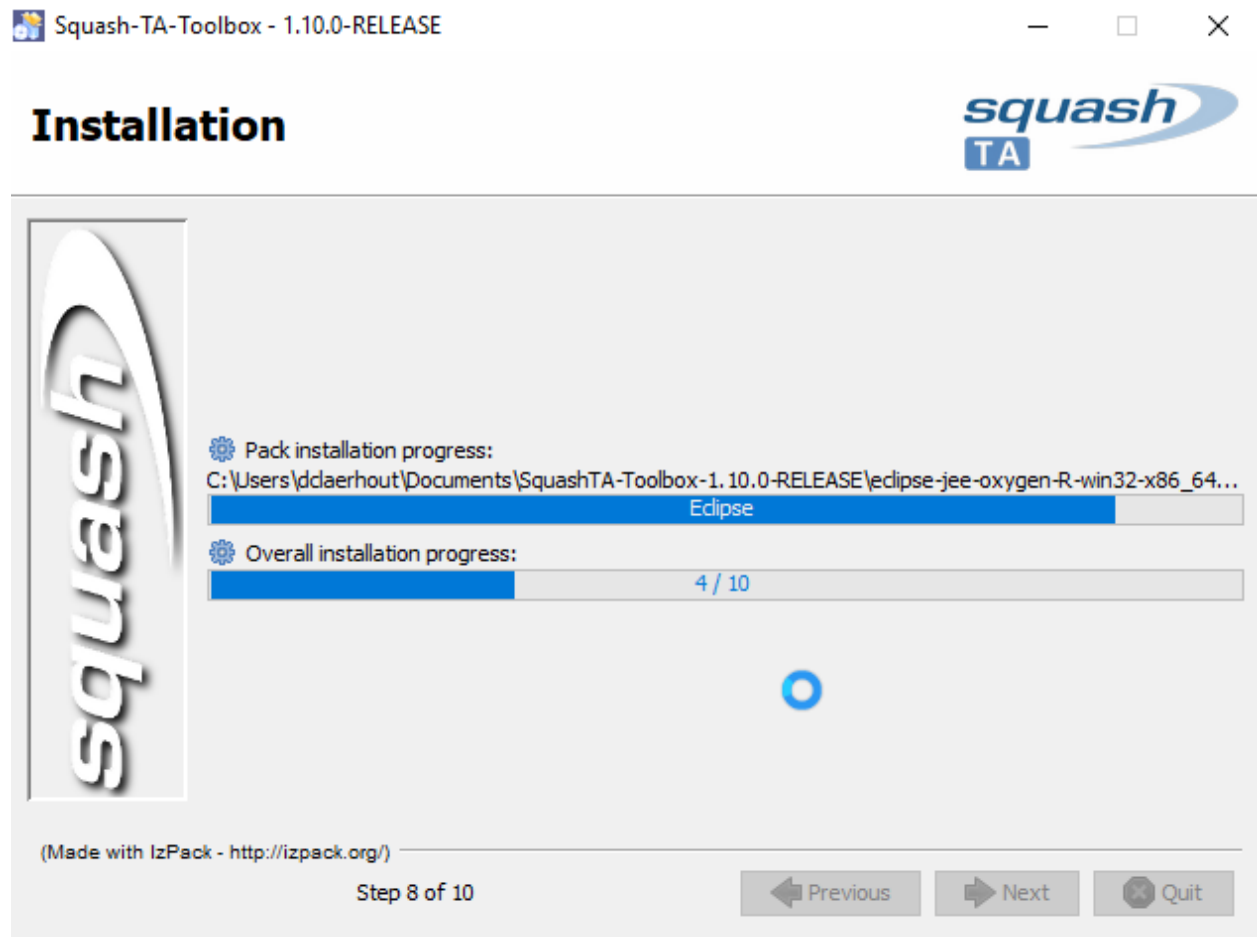


- The next step lets you check the list of installed components. You may reuse some compatible components already installed on your workstation but for now let's assume that you will be getting all tools from the toolbox, and leave all boxes checked. Click on **Next**:

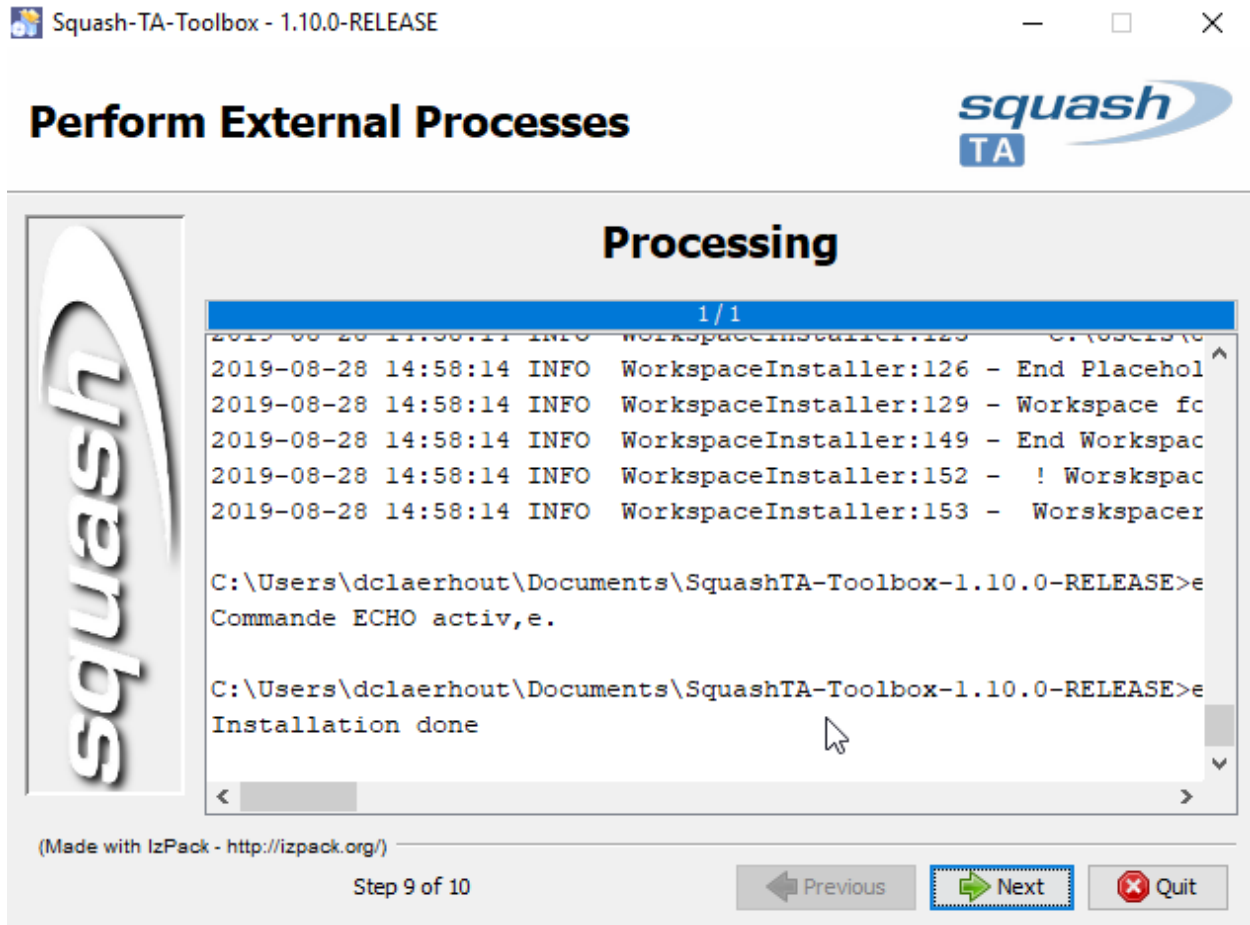


- You can choose the shortcuts you want to create for which users. To install the tools for yourself only, check **current user** on the right radio group. Click on **Next**:
- During this step, installation files are copied to the selected location. When it's done, click on **Next**:





- Some more information about the installation. Click on **Next**:



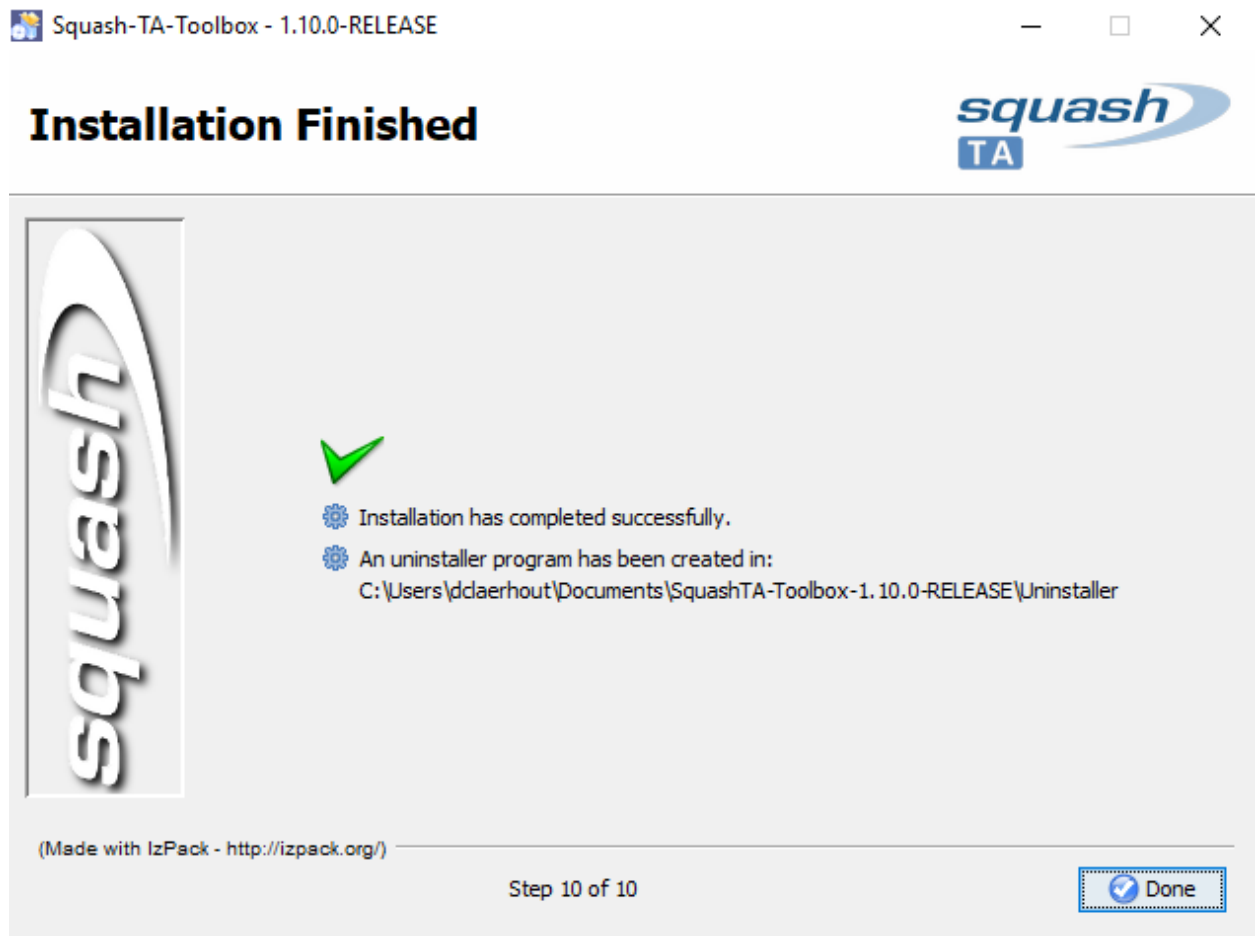
- Done! You have just installed your Squash-TF test automation environment:

## Eclipse Toolbox - Upgrade

- Download the new Squash TA toolbox you want to install [here](#).
- Install the new toolbox version in a separate directory as classical new install (For more information, please consult this [page](#))
- You are ready to work.

### Remark:

- Be careful, if you had special configurations for your project in your previous toolbox, then they will be lost. You had to set it again. You could use the eclipse import existing project feature, but for maven multimodule projects this doesn't work.
- **For 1.9.0** : If you are migrating to version 1.9.0 of Squash-TA-Framework, the logging has been upgraded to log4j 2-5. If you want to use project developed with older version of TA you need to modify your log4j





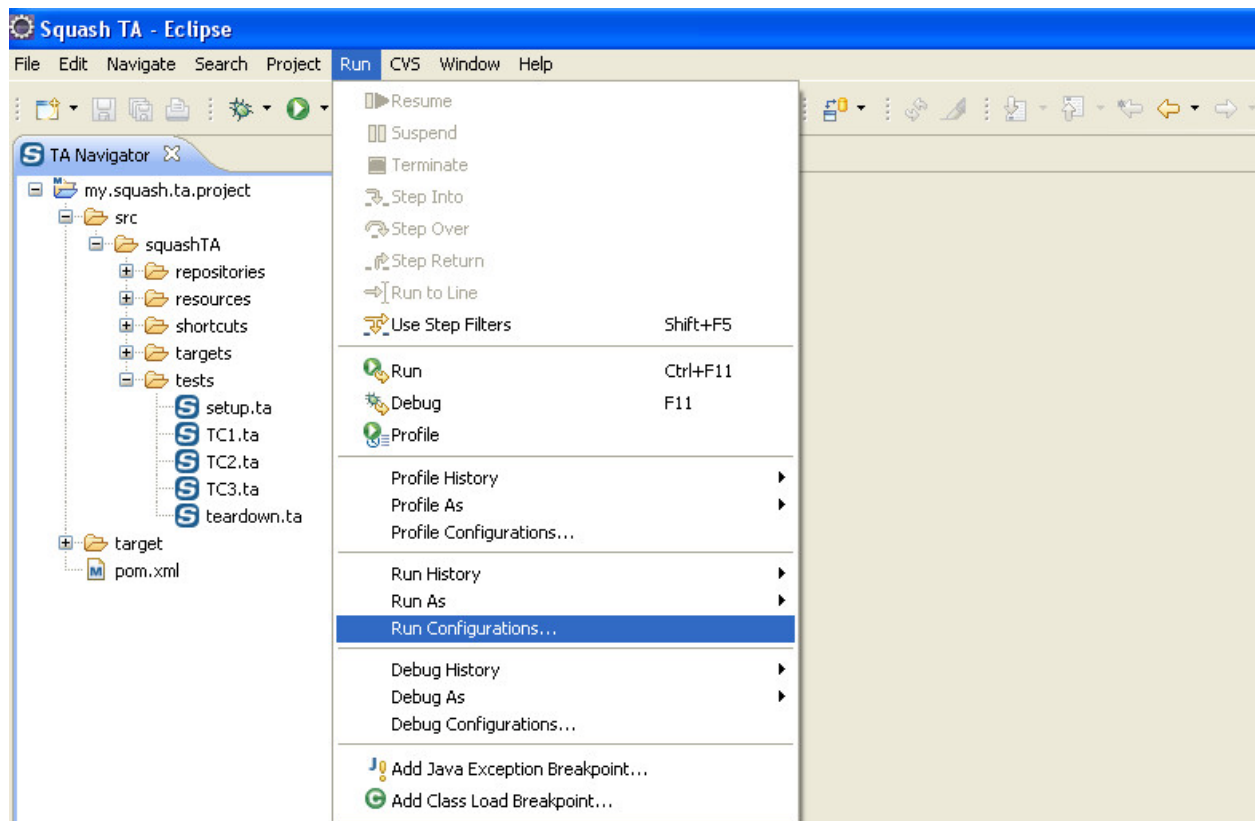
configuration files according to [Apache log4j documentation](#).

## Eclipse “Run configuration” for SKF

Squash TA toolbox default workspace embeds some default eclipse “Run configuration” to run Squash Keyword Framework / Squash TA automation project. We will describe them below. Those eclipse **“Run configuration” are part of the workspace and not of the project**. So we will see in second time how to import them.

## “Run configuration” in Squash-TA toolbox

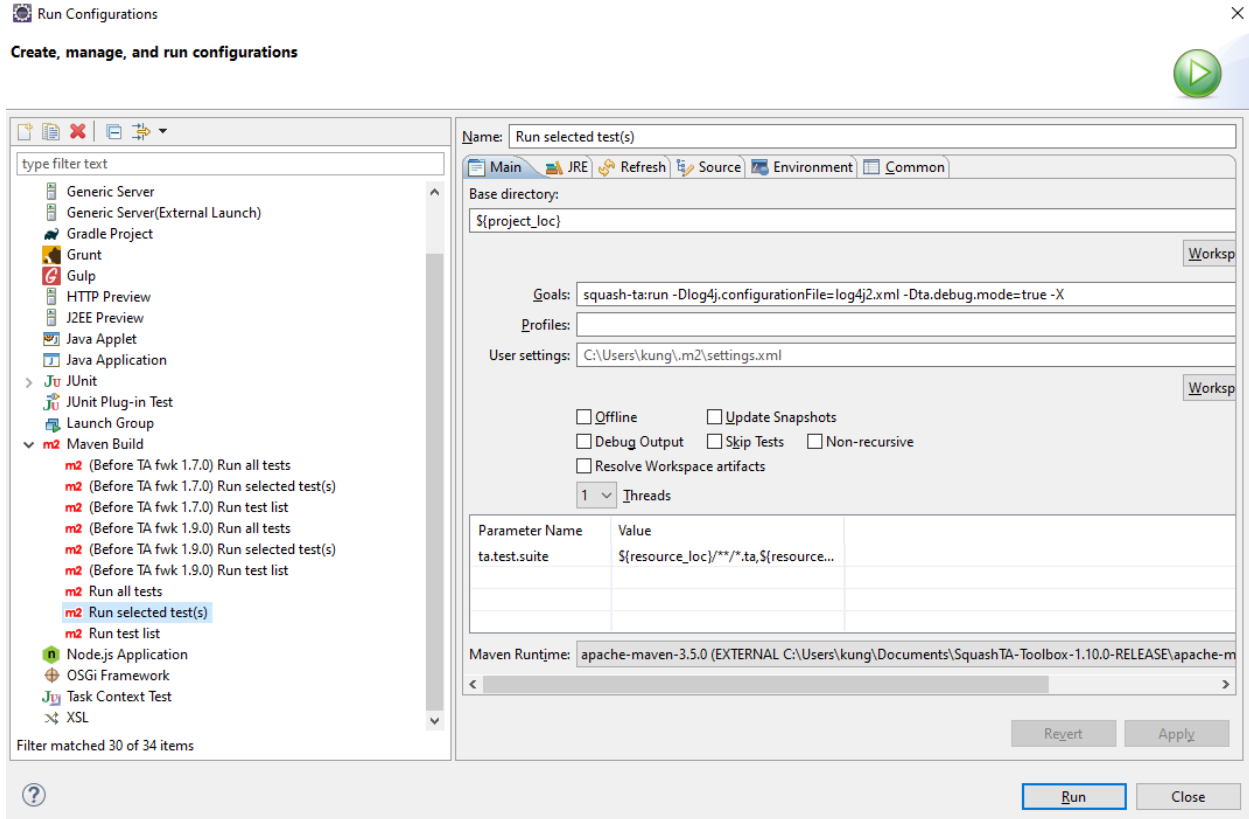
If you have installed the Squash-TA Toolbox and that you are using Squash-TA Eclipse, go to ‘Run/Run configurations’:



There, you have the pre-installed maven run configurations:

Since Squash-TA toolbox 1.9.0, there is three sets of run configuration:

- One set which use the goal “squash-tf:run” to run the test and specify location of lo4j configuration file “-Dlog4j.configurationFile=log4j2.xml”
  - Run all tests
  - Run selected test(s)
  - Run test list
- One set which use the maven lifecycle “squash-tf:run” to run the test
  - (Before TA fwk 1.9.0) Run all tests



- (Before TA fwk 1.9.0) Run selected test(s)
- (Before TA fwk 1.9.0) Run test list
- One set which use the maven lifecycle “integration-test” to run the test
  - (Before TA fwk 1.7.0) Run all tests
  - (Before TA fwk 1.7.0) Run selected test(s)
  - (Before TA fwk 1.7.0) Run test list

**Note:** To run older projects (< 1.9.0), you must use the prefixed run configuration..

### Differences between all Run configuration:

Default Config	<b>squash-ta:run (+log4j)</b> <ul style="list-style-type: none"> <li>- Before TA Framework 1.6.0</li> <li>- squash-ta:run</li> </ul>
Before TA Framework 1.7.0	integration-test

## Import eclipse “Run configuration” for SKF / Squash-TA

To import run configuration in eclipse

# Download the “Run configurations” [here](#)

# In eclipse open : *FileImport*

..figure :: ../\_static/eclipse-toolbox/import.png

# In the dialbox, choose *Run/debug* then *Launch Configurations* and click on *:guilabel: Next*

# Click on *:guilabel: Browse...* and pick the directory where “Run configurations” were download at the first step.

# Select the “Run configurations” you want to import then click on *:guilabel: Finish*

The Squash-TA toolbox is a package which includes all necessary tools to realize automated tests scripts. Apart from the SKF, this package includes Open Source tools as Eclipse, Jailer (For the creation of data sets), Selenium, Sahi. This section describe the components of the Squash TA toolbox and explains how to install and how to upgrade it.

## 1.2 Other IDE

### 1.2.1 Notepad++ Squash TF UDL

A Notepad++ UDL (User defined Language) has been developed. It’s a generous contribution from **edegenetais**.

This is a Notepad++ extension for syntax highlighting for Squash Keyword Framework

You can download it [here](#)

Documentation to install an udl extension : [How to install UDL files](#)

---

**Note:** The old wiki is currently still available here : <https://sites.google.com/a/henix.fr/wiki-squash-ta/user>

---



### 2.1 Squash Keyword Framework

**Latest Version : 1.14.0**

- [Squash Keyword Framework 1.14.0 Release Note](#)

### 2.2 Squash Eclipse Toolbox

**Latest Version : 1.10.0**

- [TA toolbox 1.10.0 Release Note](#)
- [squash-ta-tools-bundle-1.10.0-RELEASE-installer.exe.jar](#)

### 2.3 Squash Eclipse Plugin

**Latest Version : 1.2.1**

- [eclipse plugin Release Note](#)
- [org.squashtest.ta.eclipse.squash-ta-eclipse-repository-1.2.1.20140326-1001.zip](#)

### 2.4 Squash IntelliJ IDEA Plugin

**Latest Version : 1.0.0**

- [IntelliJ IDEA plugin Release Note](#)
- [squash-tf-intellij-plugin-1.0.0-RELEASE.zip](#)

## 2.5 Notepad++ Squash UDL

Contribution from **edegenetais**.

**Latest Version : 1.1**

- Notepad ++ Squash TF UDL

## 2.6 Archives

Archives

### 3.1 Contribution

---

**Note:** Under construction

---

### 3.2 Developer guideline

---

**Note:** Under construction

---

---

**Note:** This part of the site is currently under construction

---

- Our bugtracker : <https://ci.squashtest.org/mantis>
- Our forum : <https://www.squashtest.org/support-99807/forum-squash>

Squash **Keyword Framework (SKF)** is keyword oriented test automation framework using, most of the time, existing Robots. This framework includes a DSL for writing automated tests scripts and an engine to execute the scripts. It allows the use of multiple robots in one single test.

**Warning:** SKF is currently not a part of Squash TF roadmap and therefore no new developments are expected.





---

### Keyword test framework

---

#### **In a keyword tests framework :**

- test scripts are a composition of keywords
- keywords are small blocks of reusable features
- the framework provides a library of default keyword.

#### **In Squash Keyword Framework :**

- keywords are made of macros
- a large set of built-in macros is provided
- macros can be combined to create a higher level macro
- macros could also be created by using low level instructions, if needed



---

### Engine with plugins

---

At the heart of our framework, there is an engine which using plugins to pilot test robots. For each robot, a plugin (connector) has been created which provides :

- the built-in macro
- the low level instructions and their implementation (to pilot the robot).

The Squash Keyword Framework plugin architecture makes it an expandable solution : you can contribute by creating a new plugin for a new robot (or by extending an existing one). See community section for contribution.