concordion

tutorialspoint

SIMPLYEASYLEARNING

www.tutorialspoint.com


https://www.facebook.com/tutorialspointindia


https://twitter.com/tutorialspoint

About the Tutorial

Concordion is a powerful tool, written in Java, that is used to write and manage automated acceptance tests. It is distributed under the Apache Software License. Its clean and simple concepts make it very easy to learn and use. Concordion can be used along with .NET, Python, Fantom, Scala, and Ruby languages.

Audience

This tutorial has been prepared for beginners to help them understand the basic functionality of Concordion tool.

Prerequisites

Before proceeding with this tutorial, one needs to have a good understanding of Java programming language and basic HTML.

Copyright & Disclaimer

Table of Contents

1. CONCORDION – OVERVIEW

What is Concordion?

Concordion is a powerful tool to write and manage automated acceptance tests in Java based projects. It directly integrates with JUnit framework, making it ready to be used with all popular Java based IDEs like Netbeans, Eclipse, IntelliJ IDEA.

Active Software Specification

Active software specification is a way to specify the behavior of a feature. It also provides a way to implement and verify the software specification by having a connection with the system under development.

An active specification in Concordion is of two parts:

1. A cleanly written requirement document which describes the desired functionality written using XHTML. XHTML based specifications contain descriptions of the functionality provided with acceptance test examples. Example's data is marked using simple HTML tags.

2. Acceptance tests are written in Java language called fixture code. Using a Concordion extension of a standard JUnit test case, tests are implemented. It is the responsibility of the Fixture Code to find the example's data marked by tag and use them to verify the software under development.

Output of Concordion

When Concordion active specification tests are run, the output XHTML files show the original specification and test results. Successful tests are highlighted using "green" color and failed tests are highlighted using "red". Any change in the system will result in failing the test, which ensures that the specifications are always up-to-date. Concordion terms these specifications as active specifications.

Key Features

Following are the key features of Concordion:

- **Specifications as documentation** - Concordion specifications, being highly readable, can be used as active system documentation. As Concordion based specifications are written in HTML, these documents can be hyperlinked.

- **Specifications are live** - Concordion specifications contain working examples of behavior which are executed against the system. Specifications are color-coded so that any one can see whether the examples are working or not. Executing Concordion specifications regularly makes the documentation up-to-date.

- **Separate "what?" from "how?"** - Concordion specifications help separate the implementation and the required behavior of the system. It provides flexibility to change an implementation later on.

- **Simple to learn** - Concordion library is very concise. It has very few commands to learn and examples are automated using JUnit tests so that tests can be run easily and can be integrated with existing projects easily.

- **Powerful Customization** - Concordion provides extensions API which allows to add functionality. For example, Excel spreadsheets can be used as specifications, screenshots can be added to the output, logging information can be displayed, and much more

2. CONCORDION – ENVIRONMENT

Here we will see how to prepare a development environment to make use of Concordion. Before using Concordion, you need to set up JDK, Tomcat, and Eclipse on your system. Let's go step by step.

Step 1 - Setup Java Development Kit (JDK)

You can download the latest version of JDK from Oracle's Java site: Java SE Downloads. You will find instructions for installing JDK in downloaded files. Just follow the given instructions to install and configure the setup. Finally set the PATH and JAVA_HOME environment variables to refer to the directory that contains java and javac, typically java_install_dir/bin and java_install_dir respectively.

If you are running Windows and you have installed the JDK in C:\jdk1.7.0_75, you would have to put the following line in your C:\autoexec.bat file.

```
set PATH=C:\jdk1.7.0_75\bin;%PATH%
set JAVA_HOME=C:\jdk1.7.0_75
```

Alternatively, on Windows NT/2000/XP, you could also right-click on My Computer, select Properties, then Advanced, and Environment Variables. Then, you would update the PATH value and press the OK button.

On Unix (Solaris, Linux, etc.), if the SDK is installed in /usr/local/jdk1.7.0_75 and you use the C shell, you would put the following into your .cshrc file.

```
setenv PATH /usr/local/jdk1.7.0_75/bin:$PATH
setenv JAVA_HOME /usr/local/jdk1.7.0_75
```

Alternatively, if you are using an Integrated Development Environment (IDE) like Borland JBuilder, Eclipse, IntelliJ IDEA, or Sun ONE Studio, then compile and run a simple program to confirm that the IDE knows where you installed Java, otherwise do proper setup as given in the document of the IDE.

Step 2 - Setup Eclipse IDE

All the examples in this tutorial have been written using Eclipse IDE. So we would suggest you should have the latest version of Eclipse installed on your system.

To install Eclipse IDE, download the latest Eclipse binaries fromhttp://www.eclipse.org/downloads/. After downloading the installation, unpack the binary distribution into a convenient location. For example in C:\eclipse on Windows, or /usr/local/eclipse on Linux/Unix and finally set the PATH variable appropriately.

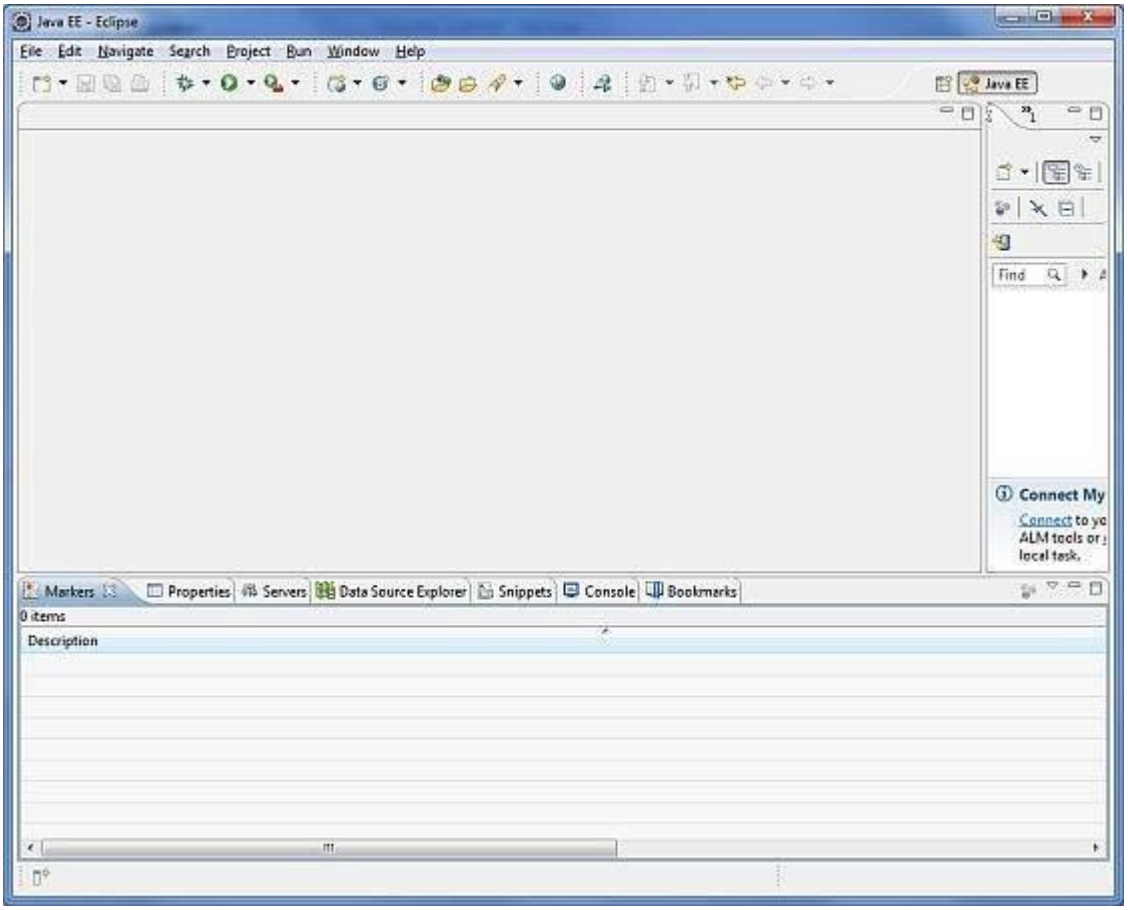
Eclipse can be started by executing the following commands in Windows environment, or you can simply double click on eclipse.exe

```
%C:\eclipse\eclipse.exe
```

Eclipse can be started by executing the following commands on a Unix (Solaris, Linux, etc.) machine:

```
$/usr/local/eclipse/eclipse
```

After a successful startup, if everything is fine, then it should display the following result:



Step 3: Download Junit Archive

Download the latest version of JUnit jar file from http://www.junit.org. At the time of writing this tutorial, we downloaded *Junit-4.10.jar* and copied it into C:\>JUnit folder.

OS	Archive name
Windows	junit4.10.jar
Linux	junit4.10.jar
Mac	junit4.10.jar

Step 4: Set JUnit Environment

Set the **JUNIT_HOME** environment variable to point to the base directory location where JUnit jar is stored on your machine. Let's assume we've stored junit4.10.jar in the JUNIT folder, then you need to take any of the following actions depending on the OS you are working on:

OS	Action
Windows	Set the environment variable JUNIT_HOME to C:\JUNIT
Linux	export JUNIT_HOME=/usr/local/JUNIT
Mac	export JUNIT_HOME=/Library/JUNIT

Step 5: Set CLASSPATH Variable

Set the CLASSPATH environment variable to point to the JUNIT jar location. Let's assume we've stored junit4.10.jar in the JUNIT folder, then you need to take any of the following actions based on the OS you are working on:

OS	Action
Windows	Set the environment variable CLASSPATH to %CLASSPATH%;%JUNIT_HOME%\junit4.10.jar;.;
Linux	export CLASSPATH=$CLASSPATH:$JUNIT_HOME/junit4.10.jar:.
Mac	export CLASSPATH=$CLASSPATH:$JUNIT_HOME/junit4.10.jar:.

Step 6 - Setup Concordion Libraries

Now you can proceed to set up your Concordion libraries. Follow the simple steps given below to download and install the framework on your machine.

Download the latest version of Concordion framework binaries from http://dl.bintray.com/Concordion/downloads/Concordion-1.5.1.zip.

At the time of writing this tutorial, we downloaded Concordion-1.5.1 on our Windows machine. When you unzip the downloaded file, it will give you the following directory structure inside E:\Concordion-1.5.1.

- **lib** - Library folder
    - hamcrest-core-1.3.jar
    - junit-4.12.jar
    - ognl-2.6.9.jar

    - xom-1.2.5.jar
- **src** - Source code folder
    - main
    - test
    - test-dummies
- Concordion-1.5.1.jar

You will find all the Concordion dependency libraries in the directory E:\Concordion\lib. Make sure you set your CLASSPATH variable on this directory properly, otherwise you will face problems while running your application. If you are using Eclipse, then it is not required to set CLASSPATH because all the setting will be done through Eclipse.

Once you are done with this last step, you are ready to proceed for your first Concordion Example which you will see in the next chapter.

3. CONCORDION – FIRST APPLICATION

Let us start programming with Concordion. Before you start writing your first example using Concordion, you have to make sure that you have set up your Concordion environment properly as explained in Concordion - Environment Setup tutorial. We also assume that you have a little bit working knowledge of Eclipse IDE.

So let us proceed to write a simple Concordion application which will print the following acceptance test:

```
Example

When Robert logs in the system, a greeting "Hello Robert!" is displayed.
```

End of ebook preview

If you liked what you saw…

Buy it from our store @ **https://store.tutorialspoint.com**