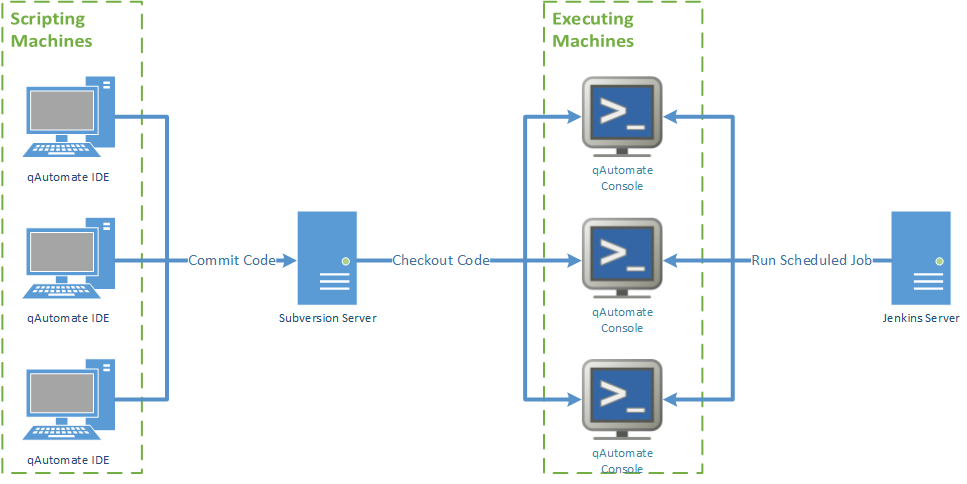
**Guide to Run Katalon in**

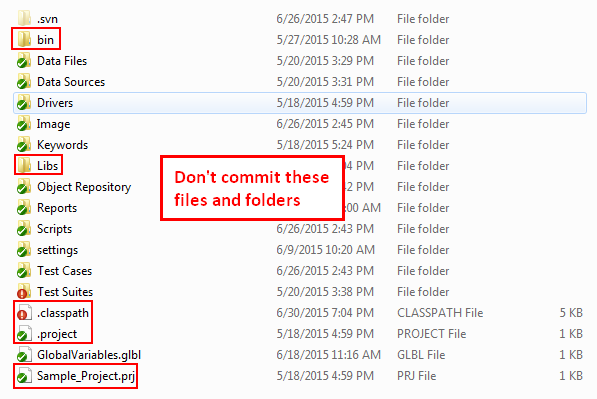
**Console Mode via Jenkins**

Assume that we have two machines: machine A for scripting and machine B for executing. We will use machine B to execute test suite via Jenkins job.



1. **Prepare Katalon and test project**

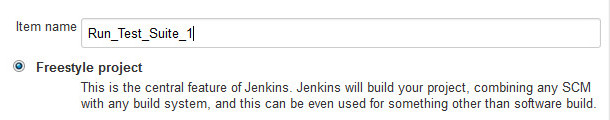
* On machine A, open Katalon and create a Katalon test project, then import it to SVN repository. SVN folder and file .prj should have the same name. Later, when we modify test project, we should NOT commit those folders and files to SVN: **bin, Libs, Reports, .classpath, .project, .properties, {project name}.prj**



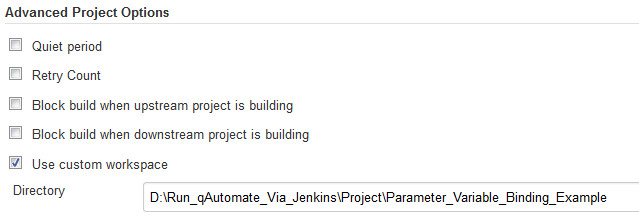
* On machine B, prepare location to store Katalon build and test project, install Jenkins.

1. **Create Jenkins job**

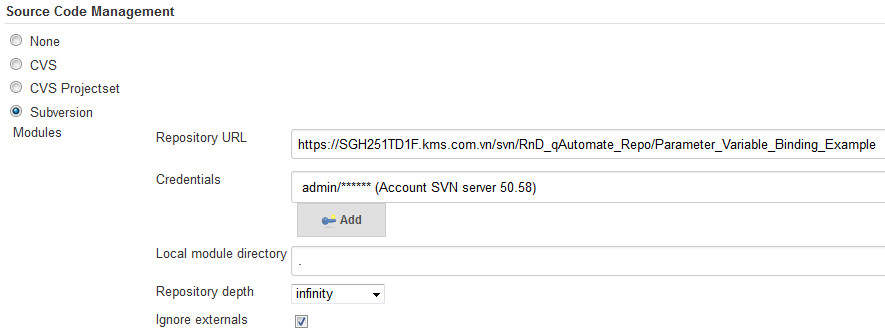
Login Jenkins > **New Item** > select **Freestyle project.**



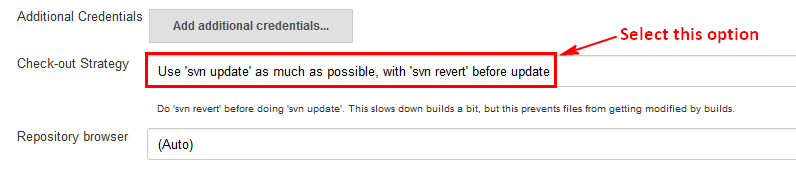
At **Advanced Project Options**, check **Use custom workspace** and specify Directory where we want to check out the project (the location is on machine B – where we’ll execute the test suite).



At **Source Code Management** section, select **Subversion** and fill in required information.



At **Check-out Strategy**, select **Use ‘svn update’ as much as possible, with ‘svn revert’ before update**.



At **Build** section, select **Execute Windows batch command** and enter command to run the test suite in console mode.



For the above example:

First command is to navigate to folder containing Katalon executable file.

Second command is to execute a batch file. This batch file contains command to execute test suite in console mode.

An example for the batch file:

katalon -runMode=console -summaryReport -statusDelay=30 -projectPk="C:\RegressionTest\RegressionTest.prj" -execute -testSuiteID="Test Suites\KeywordRegression\TS\_AlertClassKeywordRegression" -browserType=Firefox

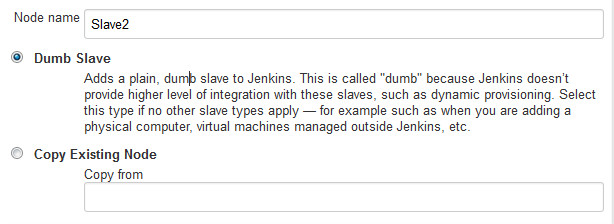
At this time, we can modify test project and commit code to SVN on machine A, then execute test suite on machine B via Jenkins job.

1. **Create Jenkins slave node (optional)**

When we want to execute test suites on multiple machines, we can create slave nodes on those machines and through Jenkins, designate jobs to them.

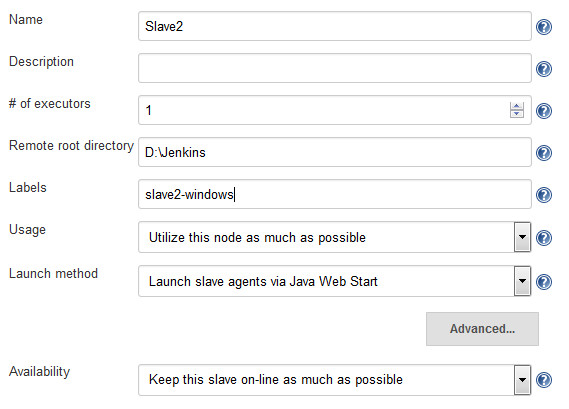
* **Create Jenkins slave node**

Login Jenkins > **Manage Jenkins** > **Manage Nodes** > **New Node**. Enter **Node name** and select **Dumb Slave**.



Enter **Name**, **Remote root directory**, **Labels** for the slave.

At **Launch method** section, select **Launch slave agents via Java Web Start**.



* **Start slave node**

Assume we have machine C to act as slave node. From machine C, login **Jenkins** > **Manage Jenkins** > **Manage Nodes** > **{slave node}**. Click **slave.jar**, save the file to your selected location.

Open command prompt, navigate to **.jar** location and run the command line, for example:

java -jar slave.jar -jnlpUrl http://192.168.50.58:9090/computer/Slave2/slave-agent.jnlp -secret fc4e5fc20e966f64c75c74f3db5c6331dff5882a4d12371a745a9ec7c93980fb

* **Designate Jenkins job to slave node**

Login **Jenkins** > **{job}** > **Configure** > check **Restrict where this build can be run**

At **Label Expression**, enter slave’s label.

