Using Jenkins Build Parameters

This article details a useful way to embed parameters in your Jenkins build job (URLs, credentials, etc.)

Overview

Build parameters can be used to store configuration options or data that should not live in source code (e.g. Mobile Cloud credentials). This example will explain how to use build parameters in Jenkins when building a Maven enabled Java project.

Example

Java Code

In order to accept the build parameter from Jenkins, use the `System.getProperty` object. For example, to accept a property called "PerfectoUsername" use the following code:

```java
String user = System.getProperty("PerfectoUsername");
```

System properties can be set in the Eclipse project for testing/debugging purposes. In a Maven enabled project, a Maven Build Configuration can be created or edited in the Run Configurations dialog. To open Run Configurations, choose the Run menu, then Run Configurations. Parameters can be added in the Main tab.
Jenkins Configuration

In order to create the parameters, open a Jenkins build job and check the “This build is parameterized” checkbox.
Then, click the Add Parameter button, and select the parameter type to add. Note, that when storing a password, choose "Password Parameter". The value will then be masked. The actual values can be provided in the configuration, or runtime values can be provided when creating the build.

The build parameters are then passed to the build in the Build > Goals and options configuration. For every parameter, append a string with the following syntax:

-D= dollar sign

Example: -DPerfectoUsername=$PERFEETOUSERNAME
<table>
<thead>
<tr>
<th>Build</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Root POM</td>
<td>pom.xml</td>
</tr>
<tr>
<td>Goals and options</td>
<td><code>test -Dtest=$CLASSPATH -DperfectoDeviceID=$PERFECTODEVICEID -DperfectoUsername=$PERFECTOUSERNAME -DperfectoPassword=$PERFECTOPASS</code></td>
</tr>
</tbody>
</table>