Matrix Reloaded Plugin Design

This page contains the "Matrix Reloaded Plugin" design.

Matrix Reloaded Plugin Design

Design Discussion

The problem which needs to be solved

A matrix project is a kind of project in Jenkins/Hudson which allows the user to configure and parallelize several builds with very similar properties, for instance multiple variants (user/debug/etc) of a build. For various reasons, often related to build environment issues and external dependencies, one or several configurations (which is the technical term for these sub-builds) of a matrix build may fail.

Currently, a failed configuration in a matrix-build implies a total rebuild of all configurations, which is both resource- and time-demanding and also opens up for new failed configurations.

This wiki-page describes a new feature, to be implemented as a Jenkins plug-in, which would allow the user to rebuild only selected configurations of a failed matrix build, and thereby allow quicker completion of a build that was partly failed due to environment issues; i.e. external services were temporarily down or any of the build-machines was broken, badly configured, failed to get a license or what ever caused it.

The core problem

A **MatrixBuild** is part of the Jenkins core. Internally in Jenkins, the configurations (the sub-jobs in the matrix, MatrixRun) are executed as regular builds They are **Run** objects which inherit from **AbstractBuild**.

We have implemented an extension point in the Matrix project called MatrixBuildListener, that is able to determine whether a MatrixRun(sub runs of MatrixBuilds) should be run or not. Furthermore, MatrixBuilds are able to identify which base build they are part of.

The rest of the solution will be implemented in a new plugin called **Matrix Reloaded Plugin** (this).

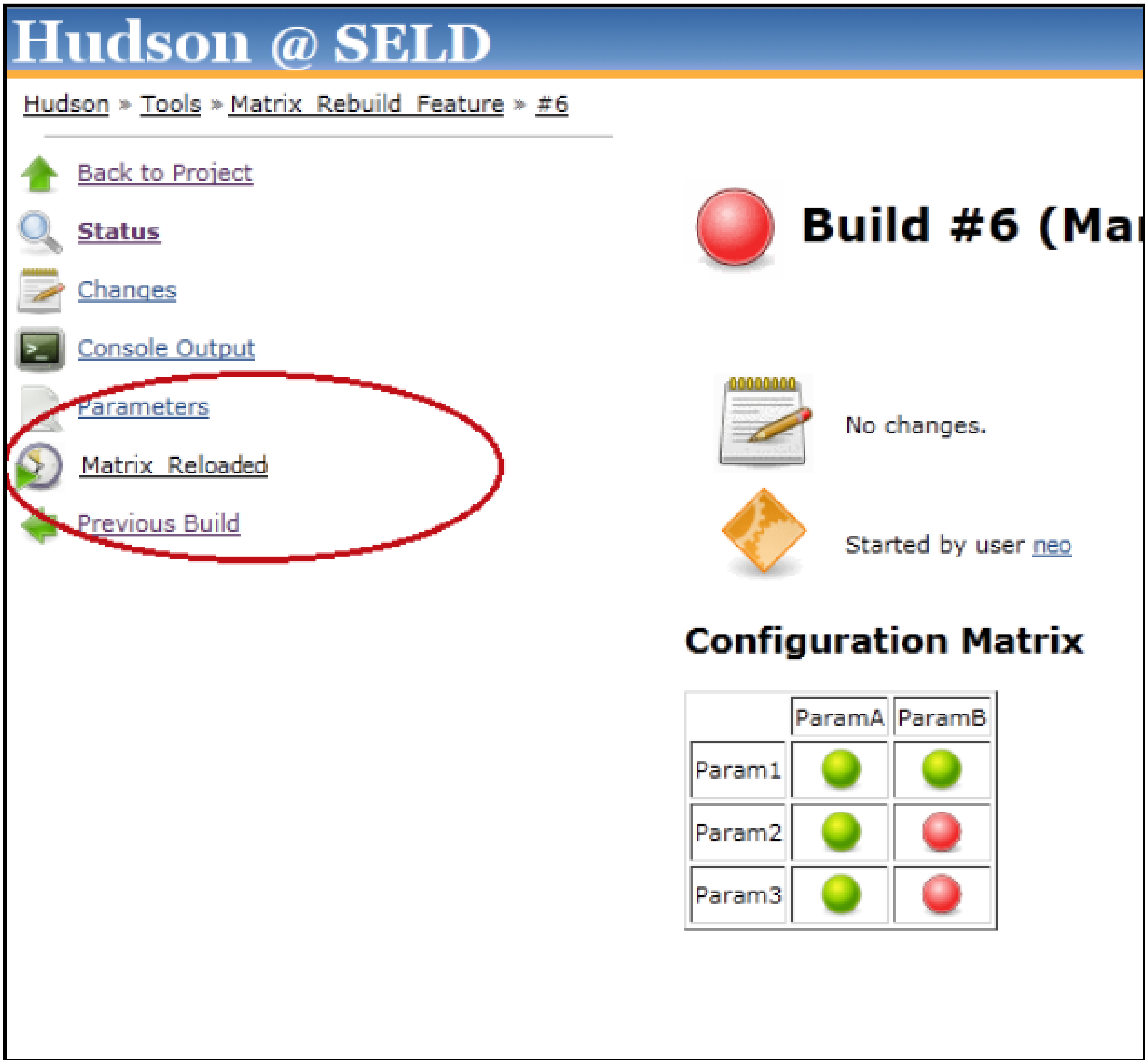
The Matrix Reloaded Plugin

The plugin will eventually be documented at the wiki page you are looking at now:

https://wiki.jenkins-ci.org/display/JENKINS/Matrix+Reloaded+Plugin

The user interface

If you have installed the plugin you will see a "matrix reloaded" link in the left panel of a Matrix Build:



Build #6 (Ma

No changes.

Started by user neo

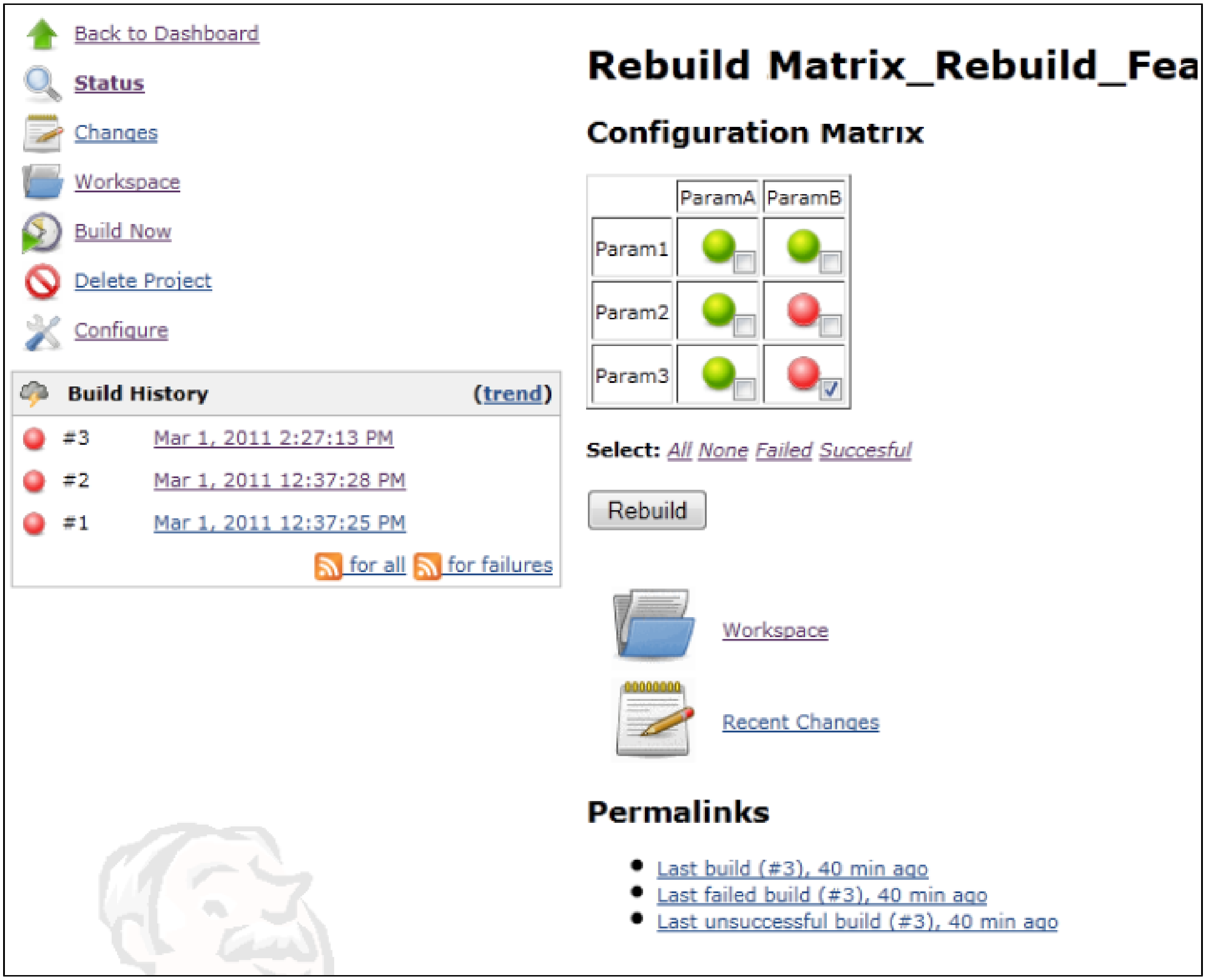
Configuration Matrix

	ParamA	ParamB
Param1	🟢	🟢
Param2	🟢	🔴
Param3	🟢	🔴

When you hit the "Matrix Reloaded" link you will be taken to the plugin's web form showing you the matrix of the build, only this time each cell (configuration) in the matrix is decorated with a check box indicating which configurations should be rebuild (the ones without check marks are reused, the ones that are checked are rebuild).

The matix will have a few useful selction shortcuts like All, None, Failed, Successful ...etc

When the user has made his choice he hits the Rebuild button.



Rebuild Matrix_Rebuild_Fea

Configuration Matrix

	ParamA	ParamB
Param1	🟢 □	🟢 □
Param2	🟢 □	🔴 □
Param3	🟢 □	🔴 ☑

Build History

🔴	#3	Mar 1, 2011 2:27:13 PM
🔴	#2	Mar 1, 2011 12:37:28 PM
🔴	#1	Mar 1, 2011 12:37:25 PM

Permalinks

- Last build (#3), 40 min ago
- Last failed build (#3), 40 min ago
- Last unsuccessful build (#3), 40 min ago

When the users has opened a configuration build (one of the individual sub-builds of the matrix) a line will show in the left panel saying ”Reload Build”. When the user activates this link he's taken to the plugin form, with the configuration pre-selected in the matrix.

Plugin Rebuild Workflow

When the user hits the rebuild button the Matrix Build is rescheduled and the user input form is parsed; for each configuration that isn't checked (must be reused) an **ActionParameter** is set that indicates that is must be *reused*, for the ones that are checked, another **ActionParameter** is set on the **Run** indicating that it must be *rebuild*. When the user inputs are all processed the Matrix Build is handed to the **RunListner**.

The Matrix Reloaded subscribes to the **onStarted** and **onCompleted** Run events, and when the **onStarted** trigger fires the **ActionParamater** on the configuration is transformed; for all the **Runs** that must be reused the **reuse** property on the **Run** object is set (see the section on core changes above) on all the configurations that must be *rebuild* an environment variable named **MRP_REBUILD** is set to indicate that this is actually a rebuild situation.

The Matrix reloaded Plugin also subscribes to the **onCompleted** event which is decorating the web page of the job with the "Reload Build" link.

Considerations

We do not plan that the Matrix Reloaded plugin should have *any* influence on jobs actually being rebuild. This could potentially create a a bit of a paradox: If a build is reusing some configurations and rebuilding others, and the job is configured to get the latest commit from a VSC, and a new commit actually exists, then *some* (the reused ones) configurations are based on one commit, while *others* (the rebuild ones) are actually based on the newer commit.

Potentially a mess!

We are not planning to avoid this potential mess in the first version of the plugin. In the documentation we plan to warn users about this issue and encourage them to make sure that they know what they are doing.

In the builds and configurations where we plan to used this plugin, we will only build named commits.

When used like that, it won't become messy.

currently just a place holder.

We are creating a new plugin which will end up having it's home here.

In the mean time keep youself updated at JENKINS issue # 1636.