

Getting more out of Jenkins than the UI intended





Create a plugin

- Dev overhead
- Needs restarts
- Performance footprint

Groovy Scripts!

- Available from plugins
- Powerful
- No building or installation
- No dependency management

Global Post Build Script

Global Post Script

Script File Name

Run when

Just The Script File Name, in \$JENKINS_HOME/global-post-script

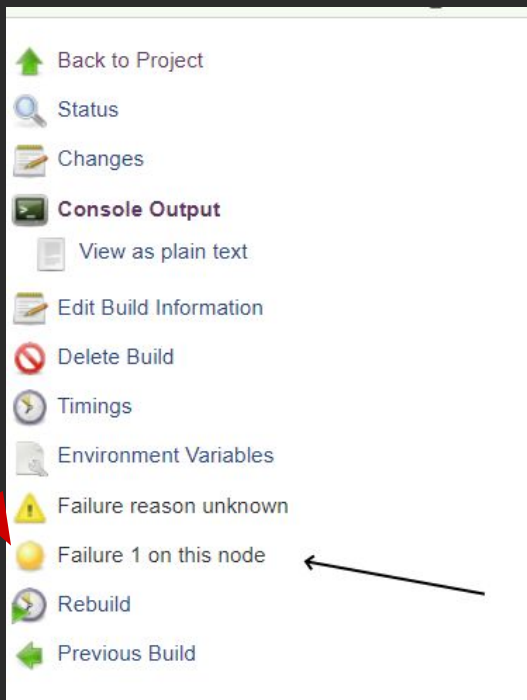
Even if the build is unstable
Only if build is stable
Even if the build is unstable
Even if the build fails
Always run, even if the job was aborted

Global Post Build Script

```
payload = [  
    "data":buildData.toJson(),  
    "buildProperties":buildVariablesString,  
    "@buildTimestamp":buildData.getTimestamp(),  
    "message":manager.run.result.toString(),  
    "changes":manager.run.changeSet.items.collect{change -> change.commitId},  
    "p4port":manager.run.getEnvironment(TaskListener.NULL)["P4_PORT"],  
    "@metadata":[id:manager.run.externalizableId],  
    "cause":manager.run.causes.collect{it.class.name+' : '+it.shortDescription},  
    "timeInQueue":timeInQueueAction ? timeInQueueAction.getQueuingDurationMillis() : 0,  
    "reconcileDuration": GetReconcileDurationFromLog.invoke(null, manager.run),  
]  
  
def extraPropertiesFile = new FilePath(manager.run.workspace, "analytics.json");  
if (extraPropertiesFile.exists()) {  
    def extraProperties = new JsonSlurper(type: JsonParserType.CHARACTER_SOURCE).  
        parse( extraPropertiesFile.read() );  
    payload.put("misc", extraProperties);  
    extraPropertiesFile.delete();  
}  
  
try {  
    dao.push(JsonOutput.toJson(payload));  
} catch (IOException e) {
```

Global Post Build Script

```
if (failedGoodCount < 3 && (matchedFailurePattern == null || !matchedFailurePattern.triggerFailsafeImmediately)) {  
    manager.addBadge("yellow.gif", 'Failure ' + failedCount + ' on this node');  
    return;  
}
```



Global Post Build Script

```
// If there are any files in the artifacts folder, artifact then delete them
def artifactsFolder = "CustomArtifactsFolder";
if (new FilePath(manager.run.workspace, artifactsFolder).exists()) {
    try {
        new hudson.tasks.ArtifactArchiver("${artifactsFolder}\\**").
            perform(
                manager.run,
                manager.run.workspace,
                null,
                manager.listener);
        new FilePath(manager.run.workspace, artifactsFolder).deleteRecursive();
    } catch (Exception e) {
        manager.listener.logger.println("WARNING: Automatic artifact step failed with this exception:");
        manager.listener.logger.println(e.toString());
    }
}
```

Global Post Build Script

```
if (deleteWorkspace) {  
    try {  
        manager.run.workspace.deleteContents()  
    } catch (Exception e) {  
        manager.listener.logger.println(e.toString());  
    }  
}
```

```
// If the agent is low on space (<10GB), take it offline immediately  
computer = manager.run.getExecutor() ? manager.run.getExecutor().getOwner() : manager.run.getBuiltOn().toComputer();  
diskSpaceMonitor = computer.monitorData["hudson.node_monitors.DiskSpaceMonitor"];  
if ( diskSpaceMonitor != null ) {  
    println(" > Agent has ${diskSpaceMonitor.gbLeft}GB space left");  
    if (diskSpaceMonitor.freeSize < 10000000000 )  
    {  
        manager.addBadge("red.gif", 'Agent out of space!');  
        println('NO SPACE: Taking slave ' + ModelHyperlinkNote.encodeTo(computer.getNode()) + ' offline...');  
        computer.setTemporarilyOffline(true, new OfflineCause.ByCLI("Agent has low disk space: "+diskSpaceMonitor.gbLeft));  
        return;  
    }  
}
```

Global Post Build Script

```
new FailurePattern(  
    match: $/*.debugging information corrupt; recompile module.*/$,  
    reason: "Program database corrupted",  
    retryOnMatch: true,  
    noFailsafeTrigger: true,  
    deleteWorkspace: true,  
),  
new FailurePattern(  
    match: $/*.P4 Task: failed: java.lang.NullPointerException.*/$,  
    reason: "Jenkins P4 connection failed",  
    retryOnMatch: true,  
    noFailsafeTrigger: true,  
    retryOnSameNode: true,  
),  
new FailurePattern(  
    match: $/*.java.nio.channels.ClosedChannelException.*/$,  
    reason: "Agent shut down during build",  
    retryOnMatch: true,  
    noFailsafeTrigger: true,  
),  
..
```


System Groovy Scripts

Build

Execute system Groovy script

Groovy command

1

Check syntax

Advanced...

Add build step

System Groovy Scripts

```
22
23 // Columns
24 column(80, "PROJECT");
25 column(12, "AVERAGE");
26
27 // Loop through the projects
28 for (project in Jenkins.instance.projects) {
29     for (build in project.builds.limit(100)) {
30         if (build.result != Result.SUCCESS)
31             continue;
32
33         durationSum += build.duration;
34         buildCount++;
35     }
36
37     averageDuration = durationSum / buildCount;
38     if (averageDuration < 6 * 60 * 1000)
39         continue;
40
41     column(80, project.getDisplayName());
42     column(12, formatDuration(averageDuration));
43
44     println();
45 }
46
47 String pad(String text, int length)
48 {
49     return String.format('%' + length + 's', text);
50 }
51
52 void column(int width, String text)
53 {
54     print(pad(text, width - 1) + " ");
55 }
```

System Groovy Scripts

```
4
5 parameters = build.getActions(hudson.model.ParametersAction).get(0);
6
7 nodes = Hudson.getInstance().getNodes();
8
9 ComputerListener[] listeners = ComputerListener.all()
10     .findAll{listener -> listener.class == org.jenkinsci.plugins.mail
11
12     listeners.each{listener -> listener.unregister()};
13
14 nodes.each { node ->
15     print(node.getNodeName() + ": ");
16
17     if (!node.labelString.matches('.*\\b' + parameters.getParameter('
18         println("[SKIP]");
19         return;
20     }
21
22     computer = node.toComputer();
23     if (!computer.isOnline()) {
24         println('Already offline');
25         return;
26     }
27
28     println('Taking agent offline for reboot');
29     computer.setTemporarilyOffline(true, new OfflineCause.ByCLI("[REB
30 }
```

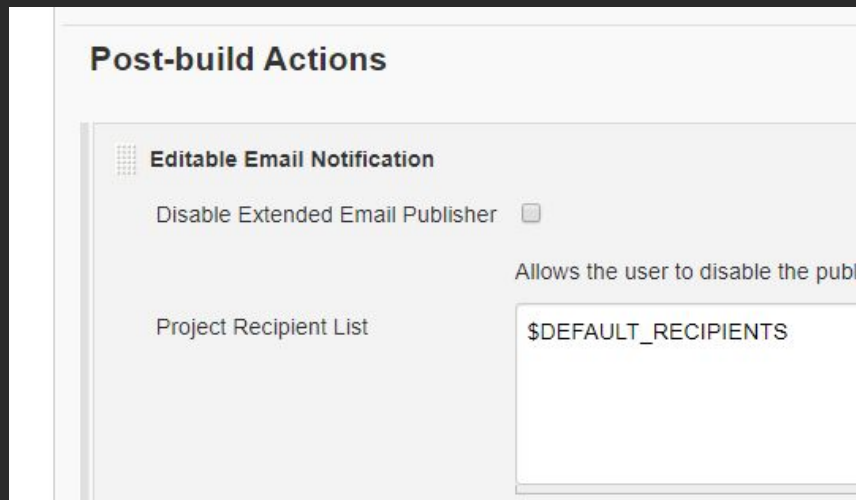
System Groovy Scripts

```
11
12 for (item in Jenkins.instance.items.findAll{!(it instanceof WorkflowJob)}) {
13     out.println("Checking job $item.name")
14
15     // Discard build history
16     discardsHistory = item.getProperty(BuildDiscarderProperty) != null;
17     if (!discardsHistory) {
18         out.println(">> Adding build discarder to $item.name")
19         // Discard logs after 14 days
20         item.setBuildDiscarder(new LogRotator(14, -1, -1, -1));
21         item.save()
22     }
23
24
25     // Job group
26     hasJobGroup = item.getProperty(JobInclusionJobProperty) != null;
27     if (!hasJobGroup) {
28         out.println(">> Adding job group to $item.name")
29         // Add the job to the default priority group
30         item.addProperty(new JobInclusionJobProperty(true, "3"));
31         item.save()
32     }
33
34
35     // Has assigned label
36     if (item.getAssignedLabel() == null) {
37         out.println("Assigning label to $item.name")
38         // If job can run anywhere, it should run on a flyweight
39         item.setAssignedLabel(new LabelAtom("flyweight"));
40         item.save()
41     }
42
43
44     // Has useful description
45     if (item instanceof JobGenerator) {
46         if (!item.description.contains("Generated from ")) {
47             out.println("Adding default generator description text to $item.name");
48             item.setDescription("Generated from ${item.name}\n\n" + item.description);
49         }
50     }
51 }
```

System Groovy Scripts

```
1 def build = Thread.currentThread().executable
2 def resolver = build.buildVariableResolver
3 def jobnameregex = resolver.resolve("JOBNAME_REGEX")
4 def testmode = resolver.resolve("TESTMODE")
5
6 for(job in jenkins.model.Jenkins.theInstance.getProjects()) {
7     String jobName = job.toString();
8     if (jobName =~ /${jobnameregex}/){
9         if ("${testmode}" != "false") {
10             println("will delete job " + jobName);
11         } else {
12             println("deleting job" + jobName);
13
14             // clear the SCM because P4 SCM has a crash bug when you delete it if it has never run
15             job.scm = new hudson.scm.NullSCM();
16             job.delete();
17         }
18     }
19 }
```


Email-ext plugin





Email-ext plugin


```
40 <p>Build log output</p>
41 <table>
42 <%if (!warnings.any() && !logErrors.any()) {%>
43 <tr><th>Could not find any errors in the build log.</th></tr>
44 <%} else {
45     warnings.take(100).each{warning ->%>
46 <tr><th>${warning.category}</th><td><a href="${getWarningUrl.invoke(null, warning, build)}">${warning.getShortt
47 <%}
48     logErrors.take(100).each{ error->%>
49 <tr><th>${error.reason}</th><td>${error.matchedLine}</td></tr>
50 <%}
51     if (warnings.size() > 100 || logErrors.size() > 100) {%>
52 <tr><th /><td>Warning list truncated, see build page for more</td></tr>
53 <%}
54 }%>
55 </table>
56
57 <p>Changes since last successful build</p>
58 <table>
59 <%if (build.changeSet.any()) {
60 <%><tr><th colspan="2">Changes for Build #${(build.number)}</th></tr><%
61     build.changeSet.each{change ->%>
62 <tr><td>${change.author}</td><td>${change.msg}</td></tr>
```


Email-ext plugin


 Timings


 Environment Variables

 Label This Build

 **MSBuild Warnings**

 Built with parameters.

 Compile error

 Failure 1 on this node

Details

FoldersFilesCategoriesWarningsOrigin**Details**New

[descriptor.h:560](#), MSBuild, Priority: High

'uIndex': undeclared identifier

[descriptor.h:562](#), MSBuild, Priority: High

'uIndex': undeclared identifier

Build #21359: FAILURE

| | |
|--------------|---------|
| New warnings | 20 |
| Built by | Agent01 |

Build log output

| | |
|-------|--|
| C2065 | descriptor.h:562 - 'uIndex': undeclared identifier |
| C2065 | descriptor.h:560 - 'uIndex': undeclared identifier |

Email-ext plugin

Post-build Actions



Editable Email Notification

Disable Extended Email Publisher ☐

Allows the user to disable the publisher, while maintaining the settings

Attach Build Log

Do Not Attach Build Log ▼

Content Token Reference

Pre-send Script

`$DEFAULT_PRESEND_SCRIPT`

Post-send Script

`$DEFAULT_POSTSEND_SCRIPT`

Additional groovy classpath

Add

Admin Script Console



Script Console

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

1

Run

The sky's the limit!

Nick.Ridgway@rebellion.co.uk

REBELLION®



www.rebellion.co.uk | @Rebellion