Common Jenkins failures

Summary

This page lists typical non-code related failures we got on jenkins, tries to explain them and should hence help to fix them.

Maven Dependency plugin

example

```
[INFO] Searching repository for plugin with prefix: 'dependency'.
[INFO] org.sonatype.plugins: checking for updates from central
[INFO] org.apache.maven.plugins: checking for updates from central
[INFO] org.codehaus.mojo: checking for updates from central
[INFO] artifact org.codehaus.mojo:dependency-maven-plugin: checking for updates from central
[INFO] ------------------------------------------------------------------------
[ERROR] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Required goal not found: dependency:analyze in org.codehaus.mojo:dependency-maven-plugin:1.0
```

There's two dependency plugins - For some reason, Maven (or Hudson's integration thereof) is looking for the old mojo one. In case the new Apache one is (temporarily) not available, mvn will pick the old one but this has no goal analyze.

Solution: redefine your job's goal - instead of `dependency:analyze`, use the more specific `org.apache.maven.plugins:maven-dependency-plugin:analyze`.

Re-triggering the build normally fixes that problem.

Missing Artifact

example

```
[ERROR] BUILD ERROR
[INFO] ------------------------------------------------------------------------
[INFO] Failed to resolve artifact.

Missing:
----------
1) info.magnolia:magnolia-module-standard-templating-kit:jar:2.0.7-SNAPSHOT

  Try downloading the file manually from the project website.

  Then, install it using the command:
      mvn install:install-file -DgroupId=info.magnolia -DartifactId=magnolia-module-standard-templating-kit -
Dversion=2.0.7-20121025.150418-11 -Dpackaging=jar -Dfile=/path/to/file

  Alternatively, if you host your own repository you can deploy the file there:
      mvn deploy:deploy-file -DgroupId=info.magnolia -DartifactId=magnolia-module-standard-templating-kit -
Dversion=2.0.7-20121025.150418-11 -Dpackaging=jar -Dfile=/path/to/file -Durl=[url] -DrepositoryId=[id]

  Path to dependency:
        1) info.magnolia:magnolia-module-commenting:jar:1.2.3-SNAPSHOT
        2) info.magnolia:magnolia-module-standard-templating-kit:jar:2.0.7-SNAPSHOT

----------
1 required artifact is missing.
```

So obviously there's an artifact missing, but why?

Potential reasons:

- there's a typo -> fix it
- that version of the referenced module had never been build and hence it was never deployed to nexus -> either build that other module or get rid of the dependency to it
- the artifact has been removed from the cache (happens every few months) -> rebuild that version of that module to get it re-installed

... isn't a file

log fragment

```
[ERROR] BUILD ERROR
[INFO] ------------------------------------------------------------------------
[INFO] Failed to create assembly: Error adding file 'info.magnolia:magnolia-module-standard-templating-kit:jar:
2.5-SNAPSHOT' to archive: /usr/local/hudson/maven-local-repo/info/magnolia/magnolia-module-standard-templating-
kit/2.5-SNAPSHOT/magnolia-module-standard-templating-kit-2.5-SNAPSHOT.jar isn't a file.
```

- Looks like this happens, when a module is build but one of its dependencies is built at the same time (so its jar is already 'deleted') -> waiting until the dependency build had finished, then re-triggering the module build fixed it

Clover Version conflict

So if you have something like

log fragment

```
[CLOVER] FATAL ERROR: Clover could not be initialised because of an unexpected error.
(class java.lang.NoSuchMethodError:com_cenqua_clover.CloverVersionInfo.
An_old_version_of_clover_is_on_your_compilation_classpath___Please_remove___Required_version_is___3_1_7()V)
{code}
```

in your log, you should try to align parent pom versions of the involved modules.

Here's a recent example of a full log.

Potential reasons:

- Module "a" is making use of module "b" but that one was instrumented with older clover version because both use different versions of parent pom's and clover versions has changed in between

Invalid Java version

log fragment

```
[INFO] ------------------------------------------------------------------------
[ERROR] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Compilation failure
Failure executing javac, but could not parse the error:
javac: invalid target release: 1.6
Usage: javac <options> <source files>
```

Reason:

- Each hudson job has a jdk to be used - if this is e.g. set to 1.5 but the build would actually require 1.6 you'll see the above error -> get the job properly configured

Related:

See Failing Builds and Kanban Board Incidents