

Support for other languages

This page will serve as brainstorm for all the features that maven should provide in order to support builds in other languages than Java.

Initial discussion is for C and C++ builds.

Currently there are two plugins: [NAR Plugin](#) and [native C-build Plugin](#)

Explanation of the NAR plugin

The nar plugin builds a jar with a file that acts as a pointer to other files in the repo:

- one artifactId-version-**x**.jar, where x is common, noarch or something like that, having files that are common among all systems.
- several artifactId-version-**AOL**.jar, where aol is the system identifier, having files that are specific to that system.

AOL stand for Architecture, OS, Linker, but may not be enough, the values needed to identify uniquely a system are

- architecture (i386, powerpc,...)
- operating system (windows, unix, linux,...)
- distribution (suse 9, fedora 4, cygwin 1.1.7,...)
- compiler
- version of the compiler

Distribution may imply the os in all cases but one, under windows you may have cygwin or other environment.

This would be hard to manage, so the proposed system id is:

- architecture
- operating system
- user defined (suse 9)

This new type would be mapped in the repository expanded (at least in the local repository) because the compiler doesn't understand compressed files. It may be stored in the remote repo as a tgz an uncompressed when downloaded, deleting the compressed version and keeping the name.

```
/groupId/artifactId/version/artifactId-version-sources.tgz/...  
/groupId/artifactId/version/artifactId-version-noarch.tgz/...  
/groupId/artifactId/version/artifactId-version-i386-linux-suse9.tgz/...  
/groupId/artifactId/version/artifactId-version-i386-win-cygin.tgz/...
```

NOTE take a look at the naming convention used by linux

This will work for almost any organization, where there isn't a lot of similar environments. eg. for a version of a distribution there's no people with different compiler versions. This won't work for a internet wide binary repository, so the suggestion is to limit it to source artifacts.

At some point in the future we may try to standardize the user defined portion.

BTW this problem also happens in java, where you can have different compiler versions (1.3, 1.4, 1.5,...)

For building all the current plugins use the cpp ant tasks