nuts

The Java Package Manager

http://github.com/thevpc/nuts

Plan

1. Why a package manager
2. `nuts` features
3. Demo

1. Why a Package Manager

- Popularity of a language is proportional to popularity of its PM
  - `Javascript`: `npm`/`npx`/`yarn`
  - `Python`: `pip`, `conda`
  - `Ruby`: `rubygems`
- Newcomer languages already include a PM
  - `golang` package manager (modules)
- `Java` ecosystem already have more that 7M packages deployed

1.1. Java Package Manager?

- `maven`, `gradle`
    - Build tools
    - Manage packaging and dependencies
    - Poor deployment management (`maven` 's `deploy` is a build time stage)
    - Lack of deployment lifecycle (install/uninstall/update)

1.2. Alternatives for deployment

- System PM / Installers
- Portable Installers
- Custom Deployements
- Build time Processors (Fat Jars)

1.3. System PM / Installers

- `rpm`, `deb`, `dmg`, `msi`
  - Native integration with OS/Env
  - Centralized management
  - Automatable (cmdline)
  - Not portable
  - Multiple deployment packages
  - Problem with installing multiple versions of the same package

1.4. Portable Installers

- `InstallAnywhere`, `GetDown`, `IzPack`, BitRock `InstallBuilder`
  - Good integration with OS/Env
  - No centralized management
  - Disk and network overload of dependencies
  - Graphical! not suitable for automation
  - Still Manual

1.5. Custom Deployers

- Custom (tomcat, netbeans) with multiple formats (tarball, zip)
  - Manual
  - No centralized management
  - Difficult to automate
  - Lack of integration with environment
  - Disk and network overload of dependencies

1.6. Fat Packages: maven-dependency-plugin

- `maven-dependency-plugin`
  - Maven plugin
  - Jars included in the "lib" folder
  - Still need to bundle the jar and the lib folder (zip with `maven-antrun-plugin`)

1.7. Fat Jars : Uber Jar

- `maven-assembly-plugin`
  - Jars deflated into the same jar
  - Can rewrite classes/resources
- `maven-shade-plugin`
  - Jars deflated into thesame jar
  - Rewrites classes/resources
  - Simpler than `maven-assembly-plugin`

1.8. Fat Jars : Jar Jar

- `onejar-maven-plugin`
  - Rewrites jar to include dependencies as jars!
  - Adds bootstrap classes
  - Changes classloader
- `spring-boot-maven-plugin`
  - Rewrites jar to include dependencies as jars!
  - Adds bootstrap classes
  - Changes classloader

1.9. So…

- All alternatives are poor and/or ugly
- `pom.xml` polluted with +16-20 lines of code
- ~~Why do we need a package manager for `Java`~~
- Why don’t we already have a package manager for `Java`!

2. nuts Package Manager for Java

Main Idea:

- Open Source
- Simple but extensible Package Manager for Java
- Good Integration with Java ecosystem and popupar build/deploy/devops tools
- Little to no Intrusion and Backward compatibility to support existing apps and repos
- Solid enough to support multiple platforms

2.1. nuts: A Package Manager for Java

Is:

- Centralized package manager for Java Apps and Libs (not only)
  - `install`, `uninstall`, `update` and `search` for packages
  - Optimized dependency resolution solver
  - Cache for dependencies across installed apps
- Automation/devops friendly commandline tool
- Developed in java
- Portable across OSes, Architectures, Desktop Environements
- Libre and Open Source

2.2. nuts: A Package Manager for Java

Is Not:

- a replacement for `maven`, `gradle` or any build tool
- a plugin for `maven`, `gradle` or any build tool
- a replacement for `spring` framework or any other framework
- a replacement for `IzPack` or `InstallAnywhere` (but can do pretty much of it)
- a replacement for `ansible` or `chef` (but is conceptually driven by automation)
- a mere download tool

2.3. nuts: Unique features

- Solid integration with environments
  - Uses OS's File System Layouts (XDG for Linux, …)
  - Supports cmdline and gui apps (installs scripts, icons, menus, …)
  - Download/Caches/Installs only relevant dependencies (according to arch …)
- Integrates seamlessly with `maven`
  - No required modification of the build process
  - No special `maven`/`gradle` plugin needed
  - Does not alter/rewrite the package

2.4. nuts: Unique features

- Solves at runtime what `maven`/`gradle` solve at build time
  - Supports `maven` and `gradle` dependency resolution algorithms, scopes, ...
- Supports out of the box
  - `maven` 's repos (including central, spring, google, ...)
  - Apache repos
  - Powerful toolbox with `props`, `xml`, `json`, `yaml`, `table`, `tree`, ... output formats
  - Bundles a `bash`/`GNU binutils` compatible (still incomplete) but enhanced java implementations
- Is statically built and has (almost) no dependencies

2.5. 'nuts'... really?

- N etwork U pdatable T hings S ervices
- The `nuts` (fool) companion for the `maven` (sage)

2.6. nuts stability

- Tested:
  - over 100 regression tests with 3500+ lines of test-code in the repository.
  - `opensuse`, `ubuntu`, `docker`, `windows7`, `windows10`
  - `sh`, `bash`, `csh`, `zsh`, `fish`