



---

# nuts, the Java Package Manager



~~~~~  
<https://thevpc.github.io/nuts>

<https://github.com/thevpc/nuts> (git repo)

<https://thevpc.github.io/nuts> (website)

[nuts.packagemanager@gmail.com](mailto:nuts.packagemanager@gmail.com)

thevpc, 2022-01-08



## 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
  - **go**lang package manager (modules)
- **Java** ecosystem already have more than 7M packages deployed



## 1.1. Java Package Manager?

- **maven, gradle**
  - Build tools
  - Dependency-management tools
  - Poor package/deployment management (**maven** 's **deploy** is a build time stage)
  - Lack of deployment lifecycle (install/uninstall/update)



## 1.2. Example

Let's take a `hello-world` example with dependencies :

```
package net.thevpc.nuts.doc.baseproject;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Main {
    private static final Logger LOG = LoggerFactory.getLogger(Main.class);
    public static void main(String[] args) {
        LOG.debug("A simple app with dependencies. Won't work out of the box!,
unless...");
    }
}
```



## 1.3. pom.xml

A minimal `pom.xml` is:

```
<?xml version="1.0" encoding="UTF-8"?><project xmlns="..."><modelVersion>
4.0.0</modelVersion>
<groupId>net.thevpc.nuts.doc</groupId>
<artifactId>base-project</artifactId>
<version>1.0-SNAPSHOT</version>
<dependencies>
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>1.2.10</version>
</dependency>
</dependencies>
</project>
```



## 1.4. Example

- With a minimal `pom.xml` we cannot execute unless we add the transitive `dependencies` to the classpath
- We also need to adjust the `pom.xml` to include the main class too!

```
target: fish — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master)> java -jar ./simple-project-1.0-SNAPSHOT.jar
no main manifest attribute, in ./simple-project-1.0-SNAPSHOT.jar
vpc@linux-rogue /d/g/n/t/s/target (master) [1]> java -classpath ./simple-project-1.0-SNAPSHOT.jar net.
thevpc.nuts.tutorial.simpleproject.Main
Exception in thread "main" java.lang.NoClassDefFoundError: org/slf4j/LoggerFactory
    at net.thevpc.nuts.tutorial.simpleproject.Main.<clinit>(Main.java:11)
Caused by: java.lang.ClassNotFoundException: org.slf4j.LoggerFactory
    at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:581)
    at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:178)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:522)
    ... 1 more
vpc@linux-rogue /d/g/n/t/s/target (master) [1]> █
```



## 1.5. Alternatives for deployment

- Java Web Start
- System PM / Installers
- jpackage, jlink
- Portable Installers
- Custom Deployments
- Build time Processors (Fat Jars)





## 1.6. Java Web start

- Run Remote App using **jnlp** file (with all of its dependencies)
- Special packaging
- Execution Sandbox : More Limitations
- Deprecated!! since Java9
- No Shared Dependencies / Centralized Dep Mgt
- The same applies to alternatives : **trivrost**, **OpenJNLP**



## 1.7. System PM / Installers

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



## 1.8. JPackage jlink

- **rpm, deb, dmg, msi**
  - All System PM / Installers applies
  - Not portable
  - java 8- not supported
  - requires all dependencies to be packaged as rpm/deb or be bundled for each app
  - JRE bundled each time!



## 1.9. 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 (most of the time)
  - Still Manual



## 1.10. 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



## 1.11. 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**)



```

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>3.2.0</version>
      <configuration>
        <archive>
          <manifest>
            <addClasspath>true</addClasspath>
            <classpathPrefix>lib/</classpathPrefix>
            <mainClass>net.thevpc.nuts.doc.mvndepproject.Main</mainClass>
          </manifest>
        </archive>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-dependency-plugin</artifactId>
      <executions>
        <execution>
          <id>copy-dependencies</id>
          <phase>prepare-package</phase>
          <goals>
            <goal>copy-dependencies</goal>
          </goals>
          <configuration>
            <outputDirectory>${project.build.directory}/lib</outputDirectory>
          </configuration>
        </execution>
      </executions>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-antrun-plugin</artifactId>
      <version>3.0.0</version>
      <executions>
        <execution>
          <id>antrun-archive</id>
          <phase>package</phase>
          <goals>
            <goal>run</goal>
          </goals>
          <configuration>
            <target>
              <property name="final.name" value="${project.build.directory}/${project.build.finalName}"/>
              <property name="archive.includes" value="${project.build.finalName}.${project.packaging} lib/*"/>
              <property name="tar.destfile" value="${final.name}.tar"/>
              <zip basedir="${project.build.directory}" destfile="${final.name}.zip" includes="{archive.includes}" />
            </target>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>

```



File Explorer (Left):

- antrun
- classes
- lib
- maven-archiver
- maven-status
- dependency-project-1.0-SNAPSHOT.jar
- dependency-project-1.0-SNAPSHOT.zip

Zip Archive Viewer (Right):

dependency-project-1.0-SNAPSHOT.zip — Ark

Archive File Settings Help

Extract Preview Open Find... Add Files... Delete

| Name                                | Size      | Compressed | M   |
|-------------------------------------|-----------|------------|-----|
| lib                                 | 3 Files   |            |     |
| logback-classic-1.2.10.jar          | 228.3 KiB | 201.4 KiB  | -rv |
| logback-core-1.2.10.jar             | 443.6 KiB | 390.1 KiB  | -rv |
| slf4j-api-1.7.32.jar                | 40.5 KiB  | 35.9 KiB   | -rv |
| dependency-project-1.0-SNAPSHOT.jar | 4.1 KiB   | 2.3 KiB    | -rv |





## 1.12. Fat Jars : Uber Jar

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



```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-assembly-plugin</artifactId>
      <executions>
        <execution>
          <phase>package</phase>
          <goals>
            <goal>single</goal>
          </goals>
          <configuration>
            <archive>
              <manifest>
                <mainClass>net.thevpc.nuts.doc.mvnassproject.Main</mainClass>
              </manifest>
            </archive>
            <descriptorRefs>
              <descriptorRef>jar-with-dependencies</descriptorRef>
            </descriptorRefs>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```



assembly-project-1.0-SNAPSHOT-jar-with-dependencies.jar — Ark

Archive File Settings Help

Extract Preview Open Find... Add Files... Delete

| Name        | Size               | Compressed | Mode       | CRC ch |
|-------------|--------------------|------------|------------|--------|
| META-INF    | 2 Folders, 1 File  |            | drwxr-xr-x |        |
| ch          | 1 Folder           |            | drwxr-xr-x |        |
| qos         | 1 Folder           |            | drwxr-xr-x |        |
| logback     | 2 Folders          |            | drwxr-xr-x |        |
| net         | 1 Folder           |            | drwxr-xr-x |        |
| thevpc      | 1 Folder           |            | drwxr-xr-x |        |
| org         | 1 Folder           |            | drwxr-xr-x |        |
| slf4j       | 4 Folders, 9 Files |            | drwxr-xr-x |        |
| logback.xml | 549 B              | 285 B      | -rw-r--r-- | D4F05E |

assembly-proj...



## 1.13. 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



```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <version>2.5.6</version>
      <executions>
        <execution>|
          <goals>
            <goal>repackage</goal>
          </goals>
          <configuration>
            <classifier>spring-boot</classifier>
            <mainClass>net.thevpc.nuts.doc.springbootproject.Main</mainClass>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```



The screenshot displays the file explorer of an IDE. On the left, a project tree shows the following structure:

- classes
- maven-archiver
- maven-status
- springboot-project-1.0-SNAPSHOT-spring-boot.jar** (selected)
- springboot-project-1.0-SNAPSHOT.jar

The main window, titled 'springboot-project-1.0-SNAPSHOT-spring-boot.jar — Ark', shows the internal structure of the selected jar file:

- BOOT-INF
  - classes
    - net
      - logback.xml
  - lib
    - logback-classic-1.2.10.jar
    - logback-core-1.2.10.jar
    - slf4j-api-1.7.32.jar
    - spring-boot-jarmode-layertools-2.5.6.jar
  - classpath.idx
  - layers.idx
- META-INF
  - org
    - springframework
      - boot
        - loader
          - archive
          - data
          - jar



## 1.14. 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:

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





## 2.1. nuts: A Package Manager for Java

- Centralized package manager for Java Apps and Libs (not only)
  - **install**, **uninstall**, **update**, **search** and **exec** for packages
  - Optimized dependency resolution solver
  - Cache for dependencies across installed apps
- Automation/devops friendly commandline tool
- Portable across Architectures, OSes, OS Distibs, Desktop Environments, Platforms (Java versions)
- Libre and Open Source, developed in java



## 2.2. nuts: A Package Manager for Java

Is Not:

- a replacement for **maven**, **gradle** or any build tool (used at deploy time)
- a plugin for **maven**, **gradle** or any build tool (do not change the build process)
- 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: Maven & Gradle

- Integrates seamlessly with **maven**
  - No required modification of the build process
  - Does not alter/rewrite the package
  - No special **maven/gradle** plugin needed
- Supports local Jars, public packages (maven central), and private packages (local .m2, nexus repos,...)
- Solves at runtime what **maven/gradle** solve at build time
  - Supports **maven** and **gradle** dependency resolution algorithms, scopes, ...



## 2.4. nuts: Dependency Optimization

- Downloads, Caches and Installs only relevant dependencies according to
  - **arch** (hardware architecture: x86, x64, relevant for native dependencies)
  - **os** (operating system: Win/Linux/Mac, relevant for specific tasks)
  - **osDist** (operating distribution : Ubuntu/OpenSuse,...)
  - **desktop** (desktop environment, relevant for icon/shortcut creation and environment integration)
  - **platform** (java SE versions installed to know what dependencies to use)



## 2.5. nuts: Integration

- Solid integration with environments
  - Uses OS's File System Layouts (XDG for Linux, ...)
    - separate folders per app
    - separate folders for log, config, lib, cache, etc.
    - portable across OSes (~/.config versus ~/AppData)
  - Supports cmdline and gui apps (installs scripts, icons, menus, ...)
  - Supports **jar** and **zip** based apps



## 2.6. nuts: Toolbox

- Terminal Coloring on Linux/Windows
- Supports Windows `cmd/PowerShell` and \*NIX `sh`, `bash`, `csch`, `zsh` and `fish` and their relative `rcfiles`
- Bundles a `bash/GNU binutils` compatible (still incomplete) but enhanced java implementations
  - `ls`, `cp`, `touch`, `mkdir`, `rmdir`, ...
  - works on windows
  - adds some extra goodies (`ssh`, `json`, support ...)



## 2.7. nuts: Existing Apps

- Supports out of the box
  - **maven** 's repos (including central, spring, google, ...), more than 7M dependencies
  - Apache repos (**netbeans**, **tomcat**, **derby**, etc...)



## 2.8. nuts: Automation

- Powerful toolbox with customizable output formats
  - `props`
  - `xml`
  - `json`
  - `yaml`
  - `table`
  - `tree`





## 2.9. nuts: Unique features

- Is statically built and has (almost) no dependencies
- Can be used as a library to support transitive ClassPath resolution
- Has a clean and rich API



## 2.10. nuts: Stability

- Tested:
  - over 160 regression tests with 3500+ lines of test-code in the repository.
  - **opensuse, ubuntu, docker, windows7, windows10**
  - **sh, bash, csh, zsh, fish**



## 2.11. 'nuts'... really?

- **N**etwork **U**pdatable **T**hings **S**ervices
- The **nuts** (fool) companion for the **maven** (sage) in the Software Kingdom's Palace!



## 3. Demonstration

### 3.1. Install Nuts

1. Download `nuts.jar`
2. Run `java -jar nuts.jar -Zy`
3. Restart your terminal



## 3.2. Install Nuts (Linux)

- Install for Preview/Evaluation, most **recent**

```
$ wget https://thevpc.net/nuts-preview.jar -o nuts.jar  
$ java -jar nuts.jar -Zy -r=+dev  
$ exit
```

- Install for Production, most **stable**

```
$ wget https://repo.maven.apache.org/maven2/net/thevpc/nuts/nuts/0.8.3/nuts-0.8.3.jar  
-O nuts.jar  
$ java -jar nuts.jar -Zy  
$ exit
```

- In all cases, do not forget to **restart** your terminal



### 3.3. Run the app

- We just run the app
- No modification is required
- We use the already built (by maven) jar
- The "artifactId" is (almost) sufficient to resolve the application to install



```
target: fish — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master)> java -jar ./simple-project-1.0-SNAPSHOT.jar
no main manifest attribute, in ./simple-project-1.0-SNAPSHOT.jar
vpc@linux-rogue /d/g/n/t/s/target (master) [1]> java -classpath ./simple-project-1.0-SNAPSHOT.jar net.
thevpc.nuts.tutorial.simpleproject.Main
Exception in thread "main" java.lang.NoClassDefFoundError: org/slf4j/LoggerFactory
    at net.thevpc.nuts.tutorial.simpleproject.Main.<clinit>(Main.java:11)
Caused by: java.lang.ClassNotFoundException: org.slf4j.LoggerFactory
    at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:581)
    at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:178)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:522)
    ... 1 more
vpc@linux-rogue /d/g/n/t/s/target (master) [1]> nuts ./simple-project-1.0-SNAPSHOT.jar
22:54:33.963 [main] DEBUG net.thevpc.nuts.tutorial.simpleproject.Main - A simple app with dependencies
. Won't work out of the box!, unless...
vpc@linux-rogue /d/g/n/t/s/target (master)> █
```



### 3.4. Demonstration : Install Application

- Or we can install the app
  - All **required** dependencies will be resolved and downloaded
  - dependencies are shared across multiple apps
  - multiple versions of the same dependencies can coexist (required by different apps)
- And then we run it





```
target: fish — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master) [245]> nuts install simple-project
the following new artifacts are going to be installed : net.thevpc.nuts.doc:simple-project#1.0-SNAPSHOT, net.the
vpc.nuts.tutorial:simple-project#1.0-SNAPSHOT
should we proceed?
(default is y, accepts y, n) ? : y
install net.thevpc.nuts.doc:simple-project#1.0-SNAPSHOT ...
require ch.qos.logback:logback-classic#1.2.10 from local repository (maven-local).
require ch.qos.logback:logback-core#1.2.10 from local repository (maven-local).
require org.slf4j:slf4j-api#1.7.32 from local repository (maven-local).
install net.thevpc.nuts.doc:simple-project#1.0-SNAPSHOT from local repository (maven-local). set as default.
install net.thevpc.nuts.tutorial:simple-project#1.0-SNAPSHOT ...
install net.thevpc.nuts.tutorial:simple-project#1.0-SNAPSHOT from local repository (maven-local). set as default
.
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts simple-project
22:58:56.722 [main] DEBUG net.thevpc.nuts.tutorial.simpleproject.Main - A simple app with dependencies. Won't wo
rk out of the box!, unless...
vpc@linux-rogue /d/g/n/t/s/target (master)> █
```



### 3.5. Install Gui App

- We can run a gui app of course
- **nuts** will create for it
  - a Desktop Shortcut (Icon)
  - a Menu Item



```
target: fish — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts -y install netbeans-launcher
install net.thevpc:netbeans-launcher#1.3.0 ...
require net.thevpc:common:thevpc-common-plaf#1.1.0 from local repository (maven-local).
require com.formdev:flatlaf#1.1.2 from local repository (maven-local).
update file /home/vpc/.bashrc
force updating .nuts-env.sh, .nuts-env.fish, .nuts-init.sh, .nuts-init.fish, nuts, .nuts-term-init.sh, .nuts-term-in
it.fish, nuts-term, .bashrc, .profile, .zshenv, .cshrc, .kshrc, config.fish, net.thevpc.nuts-nuts.desktop, net.thevc
c.nuts-nuts.menu to point to workspace /home/vpc/.config/nuts/default-workspace
create file /home/vpc/.config/nuts/default-workspace/apps/id/net/thevpc/nuts/nuts/0.8.3-rc1/bin/netbeans-launcher
install net.thevpc:netbeans-launcher#1.3.0 from local repository (maven-local). set as default.
vpc@linux-rogue /d/g/n/t/s/target (master)> []
```

The screenshot shows a Linux desktop environment. On the left, the application menu is open, displaying a search bar and various application categories like Favorites, All Applications, Development, Education, Games, Graphics, Internet, Multimedia, Office, Science & Math, and Settings. The main area shows a grid of application icons including Vivaldi, System Settings, Dolphin, Kate, Discover, Help, digiKam, Konsole, Kontact, Florence Virtual Keyboard, SUSE Studio Imagewriter, Eekboard, OBS Studio, Kamoso, KSysGuard, and Netbeans Launcher 1.3.0.

On the right, a window titled "Netbeans Launcher" is open, showing a table of installed and available Netbeans IDE versions. The table has columns for Name, Status, and Release Date.

| Name                    | Status        | Release D... |
|-------------------------|---------------|--------------|
| Netbeans IDE 12.6       | Not Installed | 2021 Nov 22  |
| Netbeans IDE 12.5       | Not Installed | 2021 Sep 08  |
| Netbeans IDE 12.4       | Not Installed | 2021 May 14  |
| Netbeans IDE 12.4-be... | Not Installed | 2021 Apr 07  |
| Netbeans IDE 12.3       | Not Installed | 2021 Feb 25  |
| Netbeans IDE 12.2       | Not Installed | 2020 Nov 21  |

At the bottom right of the window, the version "v1.3.0" is displayed.



### 3.6. Search for available applications

- We can search for installed or available (local/remote) apps
- We can search for apps and/or libs

```
target: fish — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search -l
d-- 2022-01-04 23:06:04.694 anonymous <main> com.formdev:flatlaf#1.1.2
I-X 2022-01-04 23:06:04.668 anonymous <main> net.thevpc.netbeans-launcher#1.3.0
i-x 2022-01-04 23:05:47.821 admin <main> net.thevpc.nuts:nuts#0.8.3-rc1
ir- 2022-01-04 23:05:48.558 admin <main> net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1
d-- 2022-01-04 23:06:04.666 anonymous <main> net.thevpc.common:thevpc-common-plaf#1.1.0
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --apps
net.thevpc.netbeans-launcher#1.3.0
net.thevpc.nuts:nuts#0.8.3-rc1
  browse ~/.config/.../thevpc-common-plaf/1.1.0  eol

vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --lib
com.formdev:flatlaf#1.1.2
net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1
net.thevpc.common:thevpc-common-plaf#1.1.0
vpc@linux-rogue /d/g/n/t/s/target (master)> █
```



## 3.7. Repositories

- We can configure Repositories used to install/update packages
- We can list Repositories used to install/update packages
- Supports
  - Standard Maven Repositories
  - Plain Folders
  - Browsable HTTP folders (Parses HTML for common Webserver Directory Lists)



```
target: fish — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts settings list repos --tree
├── name=maven-local
│   ├── type=maven
│   ├── location=/home/vpc/.m2/repository
│   ├── enabled=enabled
│   └── mirrors
├── name=system
│   ├── type=nuts
│   ├── location=/opt/nuts/lib/default-workspace/id
│   ├── enabled=enabled
│   └── mirrors
├── name=maven-central
│   ├── type=maven
│   ├── location=htmlfs:https://repo.maven.apache.org/maven2
│   ├── enabled=enabled
│   └── mirrors
└── name=local
    ├── type=nuts
    ├── location=/home/vpc/.config/nuts/default-workspace/local
    ├── enabled=enabled
    └── mirrors
vpc@linux-rogue /d/g/n/t/s/target (master)> █
```



## 3.8. Integration and Formats

- Customize any command's output to use structured/parsable or user friendly output formats
- All Commands support options!
  - structured (parsable) : `--json`, `--xml`, `--props`, `--yaml`
  - unstructured : `--plain`, `--table` , `--tree`



```
target: fish -- Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --long
d-- 2022-01-04 23:06:04.694 anonymous <main> com.formdev:flatlaf#1.1.2
I-X 2022-01-04 23:06:04.668 anonymous <main> net.thevpc:netbeans-launcher#1.3.0
i-x 2022-01-04 23:05:47.821 admin <main> net.thevpc.nuts:nuts#0.8.3-rc1
ir- 2022-01-04 23:05:48.558 admin <main> net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1
d-- 2022-01-04 23:06:04.666 anonymous <main> net.thevpc.common:thevpc-common-plaf#1.1.0
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --json
[
  "com.formdev:flatlaf#1.1.2"
  , "net.thevpc:netbeans-launcher#1.3.0"
  , "net.thevpc.nuts:nuts#0.8.3-rc1"
  , "net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1"
  , "net.thevpc.common:thevpc-common-plaf#1.1.0"
]
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --xml
<?xml version="1.0" encoding=?>
<root>
<string value="com.formdev:flatlaf#1.1.2"/>
<string value="net.thevpc:netbeans-launcher#1.3.0"/>
<string value="net.thevpc.nuts:nuts#0.8.3-rc1"/>
<string value="net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1"/>
<string value="net.thevpc.common:thevpc-common-plaf#1.1.0"/>
</root>
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --table


| id                                         |
|--------------------------------------------|
| com.formdev:flatlaf#1.1.2                  |
| net.thevpc:netbeans-launcher#1.3.0         |
| net.thevpc.nuts:nuts#0.8.3-rc1             |
| net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1   |
| net.thevpc.common:thevpc-common-plaf#1.1.0 |


vpc@linux-rogue /d/g/n/t/s/target (master)>
```





### 3.9. Companions

- We can use **nsh** instead of **bash** / **cmd**
- Implements common internal bash commands (**cd**,...) and constructs (**pipes**,...)
- Implements common **binutils** commands (**ls**,**mkdir**,....)
- All commands support **json** (and **yaml**, ...) out of the box
- All commands support **ssh** and extended path format (including URLs) out of the box, so that **cp** can be used as a simple alternative to **wget**



```
core: fish — Konsole
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/core (master)> nsh -c ls
nuts
nuts-runtime
pom.xml
vpc@linux-rogue /d/g/n/core (master)> nsh -c ls --json
{
  "/data/git/nuts/core": [
    {
      "name": "nuts",
      "path": "/data/git/nuts/core/nuts",
      "type": "d",
      "upermis": "rwxrwxrwx",
      "jpermis": "rwx",
      "owner": "vpc",
      "group": "users",
      "length": 4096,
      "modified": "2022-01-04T21:05:25.439838Z",
      "created": "2022-01-04T21:05:25.4392",
      "accessed": "2022-01-04T21:05:26.539Z"
    },
    {
      "name": "nuts-runtime",
      "path": "/data/git/nuts/core/nuts-runtime",
      "type": "d",
      "upermis": "rwxrwxrwx",
      "jpermis": "rwx",
      "owner": "vpc",
      "group": "users",
      "length": 4096,
      "modified": "2022-01-04T21:49:07.938503Z",
      "created": "2022-01-04T21:49:07.9382Z",
      "accessed": "2022-01-04T21:49:08.954Z"
    },
    {
      "name": "pom.xml",
      "path": "/data/git/nuts/core/pom.xml",
      "type": "-x",
      "upermis": "rwxrwxrwx",
    }
  ]
}
```



## 3.10. Bot Mode

- Running with `--bot` will disable all interaction and terminal coloring

```
target : fish — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --json
[
  "com.formdev:flatlaf#1.1.2"
  , "net.thevpc.netbeans-launcher#1.3.0"
  , "net.thevpc.nuts:nuts#0.8.3-rc1"
  , "net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1"
  , "net.thevpc.common:thevpc-common-plaf#1.1.0"
]
vpc@linux-rogue /d/g/n/t/s/target (master)> nuts search --json --bot
[
  "com.formdev:flatlaf#1.1.2"
  , "net.thevpc.netbeans-launcher#1.3.0"
  , "net.thevpc.nuts:nuts#0.8.3-rc1"
  , "net.thevpc.nuts:nuts-runtime#0.8.3.0-rc1"
  , "net.thevpc.common:thevpc-common-plaf#1.1.0"
]
vpc@linux-rogue /d/g/n/t/s/target (master)> █
```



## 3.11. Help

- An extensive help is available from within the command line

```
core: fish — Konsole
File Edit View Bookmarks Plugins Settings Help
vpc@Linux-rogue /d/g/n/core (master)> nuts help search
search :
search for artifacts
SYNOPSIS:
nuts search [--options-]... <ids> ... <args> ...
search for <ids>
OPTIONS:
--lenient
when an id is found but its descriptor and/or its file are not found, do not raise an error and continue. default no
--all-versions
return all versions of the same ids. if no will always return the latest one. default yes
--duplicates
return the same version from distinct repositories if found. default yes
--distinct
remove duplicates . default false
-L | --latest | --latest-versions
return latest version of each searched id. equivalent to --all-versions=no
-S | --single | --single-versions
return one instance of each version even if found in distinct repositories. equivalent to --duplicate=no
-s | --sort
sort result. default no
--main
include main result. This is meaningful when one needs to find only dependencies, and hence turns this flag to no. default yes
--dependencies
include dependencies in result. default no
```



### 3.12. Conclusion

- **nuts** tries to be for **java** what **npm** is for javascript
- **nuts** is a versatile toolbox
- **nuts** is **2800+** classes, **600ko+** boot jar
- I invite you to
  - Take a shot, try to use it and give feedback
  - **Star(\*)** the repository <https://github.com/thevpc/nuts>
  - Spread the word
  - Join the Core Team to enhance **nuts**



---

Thank you

please support us by starring our repo at

<https://github.com/thevpc/nuts> (git repo)

<https://thevpc.github.io/nuts> (website)

[nuts.packagemanager@gmail.com](mailto:nuts.packagemanager@gmail.com)