nuts, the Java Package Manager

nuts
Java Package Manager
~~~~~~~~~~~~~~~~~~~~~~~
https://thevpc.github.io/nuts

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

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

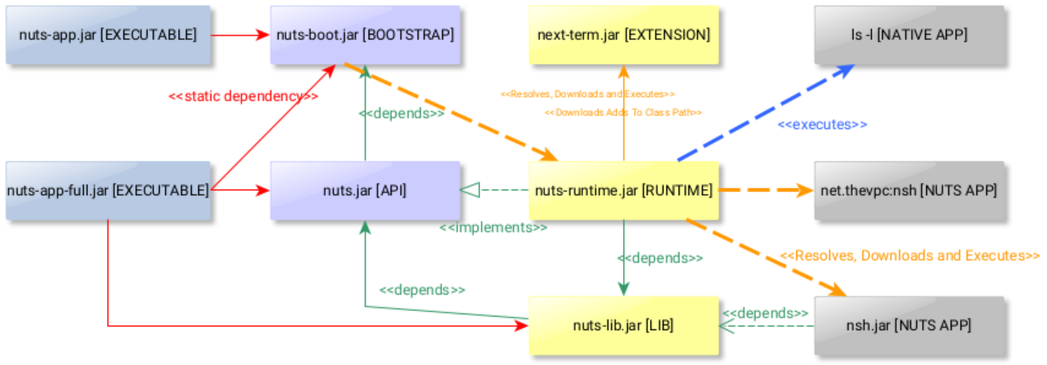
nuts.packagemanager@gmail.com

thevpc, 2025-01-04
~~~~~~~~~~~~~~~~~~~~~~~

Plan

1. API
2. Nuts as Library
3. Nuts as a Framework
4. Spring Integration

1. Main Components


nuts-app.jar [EXECUTABLE]
nuts-boot.jar [BOOTSTRAP]
next-term.jar [EXTENSION]
ls -l [NATIVE APP]
<<static dependency>>
<<depends>>
<<Resolves, Downloads and Executes>>
<<Downloads Adds To Class Path>>
<<executes>>
nuts-app-full.jar [EXECUTABLE]
nuts.jar [API]
nuts-runtime.jar [RUNTIME]
net.thevpc:nsh [NUTS APP]
<<implements>>
<<depends>>
<<Resolves, Downloads and Executes>>
<<depends>>
nuts-lib.jar [LIB]
<<depends>>
nsh.jar [NUTS APP]

2. Nuts as A Library

- Simply add nuts to your dependencies
- Compatible with java 1.8+

```
<dependency>
```

```
    <groupId>net.thevpc.nuts</groupId>
    <artifactId>nuts-lib</artifactId>
    <version>0.8.5</version>
</dependency>
```

3. Nuts as A Library

- You can also add runtime to force the runtime version

```
<dependency>
    <groupId>net.thevpc.nuts</groupId>
    <artifactId>nuts-runtime</artifactId>
    <version>0.8.5.0</version>
</dependency>
```

3.1. Session API

```
    // setSharedInstance allows NSession/NWorkspace to be accessible globally as a
singleton
    Nuts.openWorkspace("-Z","-S","y","--json").setSharedInstance();
    // then you can get the current session anywhere in your code
    NSession session=NSession.of();
    session.setConfirm(NConfirmationMode.ASK);
    session.setOutputFormat(NContentType.XML);

    session.out().println("Hello");
    session.out().printlnf("Hello");

    session.out().println(Arrays.asList("Hello"));
    session.out().printlnf("Hello %s","world");
    session.out().println(NMsg.ofC("Hello %s","world"));
    session.out().println(NMsg.ofJ("Hello {0}","world"));
    session.out().println(NMsg.ofV("Hello $v",NMaps.of("v","world"));
```

3.2. Messages and text formatting

```
Nuts Library allows multiple variants of string interpolation
```

```
    NSession session=NSession.of();
    session.setConfirm(NConfirmationMode.ASK);
    session.setOutputFormat(NContentType.XML);

    session.out().println("Hello");
    session.out().printlnf("Hello");
    session.out().println(Arrays.asList("Hello"));
    session.out().printlnf("Hello %s","world");
    session.out().println(NMsg.ofC("Hello %s","world"));
    session.out().println(NMsg.ofJ("Hello {0}","world"));
    session.out().println(NMsg.ofV("Hello $v",NMaps.of("v","world"));
```

3.3. Std In/Out/Err API

```
Nuts.openWorkspace("-Z","-S","y","--json").setSharedInstance();
NSession session=NSession.of();
session.out().println("Hello");
session.out().printlnf("Hello");

session.out().println(Arrays.asList("Hello"));
session.out().printlnf("Hello");

session.err()....;
session.in()....;
```

3.4. Find API

```
NSession session=...;
NStream<NId> ids=NSearchCmd.of()
    .addId("org.jedit:jedit")
    .setLatest(true)
    .setDistinct(true).getResultIds();
for(NId id:ids){
   ...
}
NStream<NDefinition> defs=NSearchCmd.of()
    .addId("org.jedit:jedit")
    .setLatest(true)
    .setDistinct(true).getResultDefinitions();
for(NDefinition d:defs){
   session.out().println(d.getInstallInformation()
   .getInstallFolder());
}
```

3.5. ClassPath API

```
NSession session=...;
ClassLoader loader=NSearchCmd.of()
    .addId("org.jedit:jedit")
    .addId("org.spring.framework:spring-context")
    .setLatest(true)
    .setDistinct(true).getResultClassLoader();
```

3.6. NTF API

```
NSession session=...;
session.out().printlnf("#Hello1# ##Hello2## ##:_:Hello3## ");
session.out().printlnf("```java public static class MyClass {}```");
session.out().printlnf("```js public static class MyClass {}```");
session.out().printlnf("```xml <a>hello</a>```");
session.out().printlnf("```json {a:'hello'}```");
```

3.7. Format API

```
    NSession session=...;
    class Customer{String id;String name;}
    Customer customer1,customer2,customer3; ...
    //
    session.setOutputFormat(NContentType.JSON).out().printlnf(Arrays.
asList(customer1,customer2,customer3))
    session.setOutputFormat(NContentType.TREE).out().printlnf(Arrays.
asList(customer1,customer2,customer3))
    session.setOutputFormat(NContentType.PLAIN).out().printlnf(Arrays.
asList(customer1,customer2,customer3))
    session.setOutputFormat(NContentType.XML).out().printlnf(Arrays.
asList(customer1,customer2,customer3))
    session.setOutputFormat(NContentType.PROPS).out().printlnf(Arrays.
asList(customer1,customer2,customer3))
    session.out().printlnf(Arrays.asList(customer1,customer2,customer3))
```

3.8. Format API

```
NSession session=...;
Object a,b,c,d; ...
NMutableTableModel m = NMutableTableModel.of();
m.newRow().addCells(a,b,c,d);
session.out().printlnf(m);
```

3.9. Exec API

```java
NSession session=Nuts.openWorkspace("-Z","-S");
int code=NExecCmd.of().addCommand("ls", "-l").getResult();
String out=NExecCmd.of().addCommand("nsh", "ls","--table")
    .grabOutputString()
    .getOutputString();
```

3.10. IO API

```
NCp.of()
    .from("http://my-server.com/file.pdf")
    .to("/home/my-file")
    .setProgressMonitor(true)
    .setValidator((in)->checkSHA1Hash(in))
    .run();

NPs ps=NPs.of()
if(ps.isSupportedKillProcess()){
    ps.killProcess("1234");
}
```

4. Nuts as a Framework

- Nuts Application Framework
    - Add support for Base Directory API
        - API to manage per application directories (log, cache, config,...)
    - Add support for Base Commandline API
        - standardized commandline options
        - inherit common options (--table, --json, ...)

5. Nuts as a Framework

- Add support for Application Lifecycle (Hooks for install, update, uninstall)
- Add support for auto update
- Add support for isolated input/output (via session in/out)
- Add support for Desktop Integration
    - Add Shortcuts, Menus
    - Add Aliases

6. Nuts Application Framework

- Implement NApplication
- Add Description Properties in pom.xml

7. NAF Example

```
public class Main implements NApplication {
    public static void main(String[] args) {
        new Main.runAndExit(args);
    }
    @Override
    public void run() {
        NCmdLine cmd=NApp.of().getCmdLine();
        ...
    }
}
```

8. NAF Example

```java
public class Main implements NApplication {
    public static void main(String[] args) {new Main().runAndExit(args);}
    @Override
    public void run() {
        NCmdLine cmd=NApp.of().getCmdLine();
        ...
    }
    @Override
    public void onInstallApplication() {}
    @Override
    public void onUpdateApplication() {}
    @Override
    public void onUninstallApplication() {}
}
```

9. NAF + Spring

```java
@SpringBootApplication
@Import(NutsSpringBootConfig.class)
public class AppExample implements NApplication {
    public static void main(String[] args) {
        SpringApplication.run(AppExample.class, args);
    }

    @Override
    public void run() {
        NPrintStream out = NSession.of().out();
        out.println("Hello ##World##");
    }
}
```

10. NAF + Spring

while adding the following maven dependency

```
        <dependency>
            <groupId>net.thevpc.nuts</groupId>
            <artifactId>nlib-spring-boot</artifactId>
            <version>0.8.5.0</version>
        </dependency>
```

10.1. Conclusion

- `nuts` can be used as a library or as a framework
- Using `nuts` provides many valuable features
- 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