

PlatformJar

A platform jar package

Table of contents

1 Overview.....	2
1.1 Intent.....	2
1.2 Problem.....	2
2 Design.....	2
3 Constraints.....	2
3.1 Allowed Property Values.....	2
4 Commands.....	3
4.1 extract.....	3
4.2 finish.....	3

1. Overview

PlatformJar: *A platform jar package*

1.1. Intent

- Automates the installation of a jar to an application platform such as an appserver. This jar is meant to be treated as a library rather than a package that extracts itself at deployment time. Therefore this package gets deployed as is to the desired installation directory (installroot).

1.2. Problem

Often packages such as a jar library, need to be installed onto an existing software deployment. For example, a Tomcat server needs a jar to be deployed into its lib directory since this jar may not be part of the default installation. This package module solves this problem.

2. Design

Super Type
jar

Role	Concrete. (Objects can be created.)
Instance Names	Unique
Notification	false
Template Directory	
Data View	Children, proximity: 1
Logger Name	PlatformJar

3. Constraints

3.1. Allowed Property Values

Property	Allowed Values	Default	Enforced
package-filetype	• jar	• jar	true
package-repo-url	• <code>\${framework.pkg}</code>	• <code>\${framework.pkg}</code>	false

package-vendor	• acme	• acme	false
----------------	--------	--------	-------

4. Commands

Note:

Commandline options displayed in square brackets "[]" are optional. If an option expects arguments, then angle brackets are shown after the option "<>". Any default value is shown within the brackets.

4.1. extract

Extract the package archive.

overrides the jar extract command to ensure that no extraction actually occurs.

Usage

```
extract [-base <>] -filename <> [-filtersfile <>]
-installroot <> -url <>
```

Options

Option	Description
base	<i>package base name</i>
filename	<i>the filename</i>
filtersfile	<i>filtersfile</i>
installroot	<i>the install root</i>
url	<i>the repo url</i>

4.2. finish

finish the extract of jar

overrides the Package finish command to ensure no post processing occurs.

Usage

```
finish -filename <> [-filtersfile <>] [-filtersfile <>]
-installroot <> [-script <>] -url <>
```

Options

Option	Description
--------	-------------

filename	<i>the filename</i>
filtersfile	<i>filtersfile</i>
filtersfile	<i>property file</i>
installroot	<i>the install root</i>
script	<i>script to execute</i>
url	<i>the repo url</i>