RpmBuilder

An RPM package builder.

Table of contents

1. Overview

RpmBuilder: *An RPM package builder.*

This builder contains additional functionality in the Build workflow to generate the RPM's manifest and create the package. Builds the rpm type by default (which is a noarch spec file). Use normal Builder settings and the BuilderPackageIncludes, BuilderPackageExcludes, and BuilderPackageManifest settings below to customize.

This type can be added to a CruiseControl object for continuous integration.

2. Design

Super Type
Builder

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

3. Constraints

3.1. Allowed Child Dependencies

- BuilderBuildFile1
- BuilderBuildstamp1
- BuilderBuildTarget1
- BuilderExportDestination1
- BuilderImportMax1
- BuilderImportMin1
- BuilderNotificationList1
- BuilderPackageBuildtimePattern1
- [BuilderPackageExcludes](#) 1
- BuilderPackageExtension1
- BuilderPackageFilebase1
- [BuilderPackageIncludes](#) 1

- BuilderPackageInstallRank1
- BuilderPackageInstallroot1
- [BuilderPackageManifest](#) 1
- BuilderPackagePurgeRegex1
- BuilderPackageRequireBuildstamp1
- BuilderPackageSeparator1
- BuilderPackageType1
- BuilderPackageVendor1
- BuilderPackageVersion1
- BuilderScmBinding1
- BuilderScmConnection1
- BuilderScmLabel1
- BuilderScmModule1
- BuilderScmQuiet1
- BuilderScmUser1
- BuilderScmView1
- BuilderVersionBuild1
- BuilderVersionMajor1
- BuilderVersionMinor1
- BuilderVersionRelease1
- BuilderVersionTag1
- CruiseControlArtifact1
- CruiseControlInterval1
- CruiseControlMailHost1
- CruiseControlReturnAddress1

1: These types have a *Singleton* constraint. Only one instance may be added as a resource.

3.2. Allowed Parent Dependencies

- [CruiseControl](#)
- Node
- Updater

4. Attributes

4.1. Defaults for Imported Attributes

Name	Default
cruiseControlInterval	300

`packageExcludes`	
`packageExtension`	noarch.rpm
`packageIncludes`	.*
`packageManifest`	manifest.txt
`packageSeparator`	-
`packageType`	rpm
`versionRelease`	1

5. 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.

5.1. Build

Run the build cycle.

This Build workflow has additional RPM-specific steps generateManifest and createPackage before repoImport.

Usage
```
Build [-buildstamp <>]
```

5.1.1. Workflow

1. scmCheckout
2. setBuildstamp
3. runBuildScript
4. generateManifest
5. createPackage
6. repoImport

Options

Option	Description
`buildstamp`	*build identifier*

5.2. setBuildstamp

Optionally automatically generate the buildstamp, updating the model if required

Updates the builder's BuilderBuildstamp resource value with the provided buildstamp or with a value automatically generated from the builder's attributes so that "entity.attribute.buildstamp" reflects the current value.

Usage
```
setBuildstamp [-binding <>] [-build <>] [-buildstamp <>]
[-connection <>] [-major <>] [-minor <>] [-release <>] [-tag
<>]
```
Options

Option	Description
binding	*SCM type*
build	*Version build number*
buildstamp	*build identifier*
connection	*SCM connection string*
major	*Version major number*
minor	*Version minor number*
release	*Version release number*
tag	*Version tag*

5.3. createPackage

Create package(s)

Implements the means of creating one or more package files of the format associated with the builder package type. The file base, name separator, buildstamp/version, and extension attributes are available for use in naming the package. The location of the list of files and directories to include in the package is made available via the package manifest attribute. The target directory attribute is used to determine where the package files are placed in preparation for importing into the package repository.

Usage
```
createPackage [-buildstamp <>] [-extension <>] [-filebase
<>] [-manifest <>] [-packagetype <>] [-targetdir <>]
```
Options

Option	Description
`buildstamp`	*build identifier*
`extension`	*package file name extension*
`filebase`	*package file base name*
`manifest`	*package manifest file name*
`packagetype`	*package type*
`targetdir`	*directory build artifacts will be written*

5.4. generateManifest

Generate list of files to package.

Uses the base directory, includes and excludes attributes/options to create a package type independent list of files and directories to include in the package(s). The list is stored in the file designated by the manifest option/attribute.

Usage
```
generateManifest [-basedir <>] [-buildstamp <>] [-excludes
<>] [-includes <>] [-manifest <>] [-targetdir <>]
```

Options

Option	Description
`basedir`	*directory where build resources reside*
`buildstamp`	*build identifier*
`excludes`	*directories and files to exclude from the package*
`includes`	*directories and files to include in the package*
`manifest`	*package manifest file name*
`targetdir`	*directory build artifacts will be written*

5.5. generateProject

Generate a CruiseControl project definition for inclusion in config.xml

Usage
```
generateProject [-artifactname <>] [-artifacttype <>]
[-basedir <>] -file <> [-interval <>] [-mailhost <>]
```

```
[-notificationlist <>] [-project <${context.name}>]
[-returnaddress <>]
```

Options

Option	Description
artifactname	*CruiseControl artifact name*
artifacttype	*CruiseControl artifact type*
basedir	*directory where build resources reside*
file	*The project absolute path of the file to generate for inclusion by CruiseControl*
interval	*Minimum duration between builds in seconds*
mailhost	*CruiseControl mail server address*
notificationlist	*Comma separated list of email addresses to be notified about the build*
project	*Name of the CruiseControl project to generate*
returnaddress	*CruiseControl returned mail address*

6. Related Types

The following types are defined for use with RpmBuilder.

6.1. BuilderPackageIncludes

6.1.1. Overview

BuilderPackageIncludes: *comma separated list of Ant-style paths relative to the basedir to include in the package*

Default: .*

6.1.2. Design

Super Type
BuilderSetting

Role	**Concrete**. (Objects can be created.)
Instance Names	**Unique**

6.1.3. Attributes

6.1.3.1. Exported Attributes

Name	Property
`packageIncludes`	settingValue

6.2. BuilderPackageExcludes

6.2.1. Overview

BuilderPackageExcludes: *comma separated list of Ant-style paths relative to the basedir to exclude from the package*

Default: blank (nothing excluded)

6.2.2. Design

Super Type
BuilderSetting

Role	**Concrete**. (Objects can be created.)
Instance Names	**Unique**

6.2.3. Attributes

6.2.3.1. Exported Attributes

Name	Property
`packageExcludes`	settingValue

6.3. BuilderPackageManifest

6.3.1. Overview

BuilderPackageManifest: *Package manifest file path*

Default: manifest.txt

6.3.2. Design

Super Type
BuilderSetting

Role	**Concrete**. (Objects can be created.)
Instance Names	**Unique**

6.3.3. Attributes

6.3.3.1. Exported Attributes

Name	Property
packageManifest	settingValue