MavenBuilder

A simple Builder to interface with Maven

Table of contents

1. Overview

MavenBuilder: *A simple Builder to interface with Maven*

1.1. Intent

Provide a standard interface to automate <u>Maven</u> integrated builds.

- Automate enviornment and configuration setup.
- Expose maven options and arguments into the ControlTier object model for automating each build.
- Support automated builds within a Continuous Integration (CI) Loop such as <u>Hudson</u> or <u>CruiseControl</u>.
- Support additional <u>Continuous Deployment</u> as part of a CI Loop in order to automate end to end build and deployment within a development environment.

1.2. Problem

Automating builds require command line parameters, environment, supporting infrastructure, and configuration. Setting up an automated build is different than a developer's enviornment in that pure command line and environment must be explicit and managed for change. Exposing all of these configurations including any referring CI deployed tool must be coherently managed.

2. Design

Super Type
Builder

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

3. Constraints

3.1. Allowed Child Dependencies

- BuilderAutoUpdate1
- BuilderBuildFile1
- BuilderBuildstamp1
- BuilderBuildTarget1
- BuilderImportMax1
- BuilderImportMin1
- BuilderJavaHome1
- [BuilderMavenHome](#) 1
- [BuilderMavenOpts](#) 1
- [BuilderNotificationList](#) 1
- BuilderPackageBuildtimePattern1
- BuilderPackageExtension1
- BuilderPackageFilebase1
- BuilderPackageInstallRank1
- BuilderPackageInstallroot1
- BuilderPackageRequireBuildstamp1
- BuilderPackageSeparator1
- BuilderPackageType1
- BuilderPackageVendor1
- BuilderPackageVersion1
- BuilderPath1
- BuilderScmBinding1
- [BuilderScmCheckExternal](#) 1
- BuilderScmConnection1
- BuilderScmLabel1
- BuilderScmModule1
- BuilderScmUser1
- BuilderScmView1
- [BuilderVersionBuild](#) 1
- [BuilderVersionMajor](#) 1
- [BuilderVersionMinor](#) 1
- [BuilderVersionRelease](#) 1
- BuilderVersionTag1
- CruiseControlArtifact1
- CruiseControlInterval1
- CruiseControlMailHost1
- CruiseControlReturnAddress1
- HudsonPollSpec1

- [PlatformZip](#)

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

3.2. Allowed Parent Dependencies

- [CruiseControl](#)
- [Hudson](#)
- Node
- Updater

Limit referring [CruiseControl](#) and [Hudson](#) objects for Continous Integration (CI) automated builds.

Allow referring Updater's for optional automatic deployments within a CI loop.

Support Node based deployments

4. Attributes

4.1. Exported Attributes

Name	Property	Description
basedir	deployment-basedir	the "checkout" workspace of the maven enabled build
targetdir	deployment-install-root	the "target" directory containing artifact(s) produced by maven enabled build

4.2. Defaults for Imported Attributes

Name	Default	Description
autoUpdate	false	turn on automatic deployments when integrated within a Continuous Integration Loop (typical for development environments), defaults to false
buildFile	pom.xml	maven buildfile, defaults to "pom.xml"
buildPath	/bin:/usr/bin	PATH environment used for the build, defaults to a typical unix

		environment
`buildTarget`	all	maven build target, defaults to "all"
`cruiseControlInterval`	300	Continuous Integration build interval via cruisecontrol, defaults to 300 seconds
`hudsonPollSpec`	*/5 * * * *	Continuous Integration build interval hudson, defaults to 5 minutes (cron based format)
`importMax`	1	maximum number of packages to import into the controltier package repository, defaults to 1
`importMin`	1	minimum number of packages to import into the controltier package repository, defaults to 1
`javaHome`	${env.JAVA_HOME}	value for JAVA_HOME (root of the jdk install), defaults to JAVA_HOME environment variable
`mavenHome`	${env.MAVEN_HOME}	value for MAVEN_HOME (root of the maven install), defaults to MAVEN_HOME environment variable
`mavenOpts`		optional value for MAVEN_OPTS passed into the maven build
`packageBuildtimePattern`	yyyyMMdd'T'HHmmss	
`packageExtension`	war	package filename extension of the produced artifact, defaults to the "war"
`packageFilebase`	.*?	basename of the produced artifact filename (precedes the packageSeparator character), defaults to the ".*?" regular expression
`packageInstallRank`	10	default installation for

		artifactobject's package object and used at deployment time.
`packageInstallroot`	${entity.attribute.catalinaBase}/w	target installation root for the produced artifact when deployed, defaults to the webapps directory of a tomcat container
`packageSeparator`	-	separator in produced filename of the artifact differentiating the package name from the version of the package, defaults to the "-" character
`packageType`	war	package type for the produced artifact when imported into the controltier package repository, defaults to the war package type
`requireBuildstamp`	false	enforce use of the buildstamp command line parmeter, defaults to false
`scmBinding`	cvs	source code management binding, defaults to cvs
`scmLabel`		optional label to apply prior to checkout from source
`scmUser`	${user.name}	user name for interacting with the source code managmement system, defaults to the effective user

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. runBuildScript

runs the build script

Execute the maven pom (such as pom.xml) with defined parameters

Usage

```
runBuildScript [-basedir <>] [-buildstamp <>] [-javahome <>]
[-mavenhome <>] [-mavenopts <>] -packageBuildtimePattern
[-path <>] [-target <>]
```

Options

Option	Description
basedir	*directory where build resources reside*
buildstamp	*build identifier*
	defaults to the buildstamp attribute which can be dynamically determined at build time if not provided and not required
javahome	*Java installation to use for build*
mavenhome	*Maven installation to use for build*
mavenopts	*Maven options to use for build*
packageBuildtimePattern	*Build Time date format*
path	*System path to use for build*
target	*build target to evaluate*

5.2. installPlugin

Install the ControlTier Maven plugin in the local Maven repository

Usage

```
installPlugin [-javahome <>] [-mavenhome <>] [-path <>]
```

Options

Option	Description
javahome	*Java installation to use for build*
mavenhome	*Maven installation to use for build*
path	*System path to use for build*

5.3. generateProject

Generate a CruiseControl project definition for inclusion in config.xml

Generates the cruisecontrol configuration if such a referrer to this build is defined.

Usage
```
generateProject [-artifactname <>] [-artifacttype <>]
[-autoupdate <>] [-basedir <>] [-buildfile <>] [-buildtarget
<>] -file <> [-interval <>] [-mailhost <>]
[-notificationlist <>] [-project <${context.name}>]
[-returnaddress <>]
```

Options

Option	Description
artifactname	*CruiseControl artifact name*
artifacttype	*CruiseControl artifact type*
autoupdate	*Set to true to automatically update after build*
basedir	*directory where build resources reside*
buildfile	*Maven pom file for the build (specified relative to the base directory)*
buildtarget	*Maven goal(s) for the build*
file	*The absolute path of the project 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*

5.4. generateHudsonConfig

Generate a Hudson job (project) config.xml file

Generates the hudson configuration if such a referrer to this build is defined.

Usage

```
generateHudsonConfig [-autoupdate <>] [-buildFile <>]
[-buildTarget <>] -jobdir <> [-mavenHome <>] [-mavenOpts <>]
[-notificationlist <>] [-project <${context.name}>]
```

Options

Option	Description
autoupdate	*Set to true to automatically update after build*
buildFile	*Maven pom file for the build (specified relative to the base directory)*
buildTarget	*Maven goal(s) for the build*
jobdir	*Absolute path to Hudson jobs folder*
mavenHome	*Maven installation to use for build*
mavenOpts	*Maven options to use for build*
notificationlist	*Comma separated list of email addresses to be notified about the build*
project	*Name of the project to generate*

5.5. generateHudsonJdkInstallation

Generate a Hudson JDK installation xml fragment

Usage
```
generateHudsonJdkInstallation -file <> [-javaHome <>]
```

Options

Option	Description
file	*Absolute path to the Hudson xml fragment temp file*
javaHome	*JavaHome*

5.6. writeBuildProperties

Write out properties file for use in Hudson

Usage
```
writeBuildProperties [-buildstamp <>] -file <>
```

Options

Option	Description
`buildstamp`	*build identifier*
`file`	*Absolute path to the Hudson xml fragment temp file*

6. Related Types

The following types are defined for use with MavenBuilder.

6.1. BuilderMavenHome

6.1.1. Overview

BuilderMavenHome: *Maven instance that should be used*

supports the "mavenHome" attribute setting, where settingValue will override the referring attribute-default (if defined). See use of "MAVEN_HOME" environment variable and [Maven](#) documentation.

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
`mavenHome`	settingValue

6.2. BuilderMavenOpts

6.2.1. Overview

BuilderMavenOpts: *Maven options that should be used*

supports the "mavenOpts" attribute setting, where settingValue will override the referring attribute-default (if defined). See use of "MAVEN_OPTS" environment variable and [Maven](#) documentation.

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
mavenOpts	settingValue

6.3. BuilderScmCheckExternal

6.3.1. Overview

BuilderScmCheckExternal: *Subversion specific external definition to check for last changed revision*

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
scmCheckExternal	settingValue

6.4. BuilderVersionMajor

6.4.1. Overview

BuilderVersionMajor: *Major version number to use to generate a buildstamp*

6.4.2. Design

Super Type
BuilderSetting

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

6.4.3. Attributes

6.4.3.1. Exported Attributes

Name	Property
versionMajor	settingValue

6.5. BuilderVersionMinor

6.5.1. Overview

BuilderVersionMinor: *Minor version number to use to generate a buildstamp*

6.5.2. Design

Super Type
BuilderSetting

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

6.5.3. Attributes

6.5.3.1. Exported Attributes

Name	Property
versionMinor	settingValue

6.6. BuilderVersionRelease

6.6.1. Overview

BuilderVersionRelease: *Release version number to use to generate a buildstamp*

6.6.2. Design

Super Type
BuilderSetting

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

6.6.3. Attributes

6.6.3.1. Exported Attributes

Name	Property
versionRelease	settingValue

6.7. BuilderVersionBuild

6.7.1. Overview

BuilderVersionBuild: *Build version number to use to generate a buildstamp (often pulled from Subversion, etc)*

6.7.2. Design

Super Type
BuilderSetting

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

6.7.3. Attributes

6.7.3.1. Exported Attributes

Name	Property

versionBuild	settingValue

6.8. BuilderNotificationList

6.8.1. Overview

BuilderNotificationList: *Comma separated list of email addresses to be notified about the build*

6.8.2. Design

Super Type
BuilderSetting

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

6.8.3. Attributes

6.8.3.1. Exported Attributes

Name	Property
notificationList	settingValue