Document generator plugin

*Document generator* is a Mavan plugin that allows you to generate a *doc* document starting off from an xml descriptor file directly from a maven build life cycle.

The plugin needs:

* *MS Word .docx* template file
* *.xml* schema that contains the project coordinates and the project activity changes

# Create a DOC template

In order to create a doc template you should insert in a standard word document the *placeholders* that will be replaced from the plugin with the actual values.

There are two way to add a placeholder:

* Use standard MS Word ***MergeField***
* Use Macro which display in a dialog box fields available for the model

## Create a merged Field

To add a MergeField placeholder:

* press ***Ctrl+F9*** in the MS word document
* after click with the right mouse button to open the contextual menu and select *Edit Filed*.
* Choose **MergeField in the tree**
* Write the Velocity expression (i.e. $project.name)

## Use Macro

Refer to the library documentation: <https://github.com/opensagres/xdocreport/wiki/DocxDesignReportMacro>

# Model API

Here is all properties that the plugin exposes.

Basically it is possible navigate the XML file entries using this syntax *$<tagName>.<InnerTagName>*

Furthermore the plugin exposes some shortcuts that enables a direct access to all inner values:

|  |  |  |
| --- | --- | --- |
| **Placeholder value** | **Model Value** | **Description** |
| $release | <cms:release> | Root node |
| $release-version | <cms:release>  <cms: release-version/>  </cms:release> | The verison of project released |
| $release-name | <cms:release>  <cms: release-name/>  </cms:release> | The name of project released |
| $contact | <cms:release>  <cms: contact/>  </cms:release> | The contact of releaser. |
| $date | <cms:release>  <cms: date/>  </cms:release> | The release date |
| $group | <cms:release>  <cms: group/>  </cms:release> | The software group.  - Anritsu/MasterClaw/Foundation  - Anritsu/MasterClaw/Acquisition  - Anritsu/MasterClaw/Mediation  - Anritsu/MasterClaw/Application |
| $summary | <cms:release>  <cms: summary/>  </cms:release> | The summary description |
| $arch | <cms:release>  <cms: arch/>  </cms:release> | The architecture of the RPM package.   * i686 * x86\_64 * noarch |
| $description | <cms:release>  <cms: description/>  </cms:release> | The long description |
| $item | <cms:release>  <cms: item/>  </cms:release> | The list of item released |
| $partRelease | <cms:release>  <cms: part-release/>  </cms:release> | The root of part-release tag |
| $purpose | <cms:release>  <cms: part-release>  <cms: purpose>  </cms: part-release>  </cms:release> | The purpose of release.  (Bug Fixing, New Features or Bug Fixing and New Features)  If the value is empty the system calculates the purpose from the current activities. |
| $baseline | <cms:release>  <cms: part-release>  <cms: baseline>  </cms: part-release>  </cms:release> | The list of the baselines |
| $globalAvailabilities | <cms:release>  <cms: part-release>  <cms: availabilities>  </cms: part-release>  </cms:release> | The list of the GLOBAL availabilities  Properties exposed:   * ***mcVersion*** is the Master claw version for this list of dependencies * ***package:*** the name of the package * **level:** The availability level * **risk:** The availability risk * **reason:** The availability reason |
| $featureAvailabilities | <cms:release>  <cms: part-release>  <cms: availabilities>  </cms: part-release>  </cms:release> | The list of the FEATURES availabilities  Properties exposed:   * ***mcVersion*** is the Master claw version for this list of dependencies * ***feature:*** the name of feature * **level:** The availability level * **risk:** The availability risk * **reason:** The availability reason * **customers:** The customers that request this feature * **description:**  a description |
| $otherAvailabilities | <cms:release>  <cms: part-release>  <cms: availabilities>  </cms: part-release>  </cms:release> | Useful if you need to specify a special availability:  Properties exposed:   * ***mcVersion*** is the Master claw version for this list of dependencies * ***name:*** the name of feature * **level:** The availability level * **risk:** The availability risk * **customers:** The customers that request this feature * **description:**  a description |
| $reference | <cms:release>  <cms: part-release>  <cms: reference>  </cms: part-release>  </cms:release> | The list of the referenced documents |
| $dependencies | <cms:release>  <cms: part-release>  <cms: dependencies>  </cms: part-release>  </cms:release> | The list of dependencies:  Properties exposed:   * ***mcVersion*** is the Master claw version for this list of dependencies * ***platformDependencies:*** the list of the platform deps for this mc version * ***thirdParyDependencies:*** the list of the third party deps for this mc version * ***localDependencies:*** the list of local deps for this mc version * ***monitoredDependencies:*** the list of monitored deps for this mc version |
| $platformDependencies | <cms:release>  <cms: part-release>  <cms: dependencies>  <cms: platformDependencies >  </cms: part-release>  </cms:release> | The list of dependencies:  Properties exposed:   * ***component*** the name of this component * ***required-version:*** the required version * ***tested-version:*** the tested version * ***description:*** the description |
| $third-part-dependency | <cms:release>  <cms: part-release>  <cms: dependencies>  <cms: third-part-dependency>  </cms: part-release>  </cms:release> | The list of dependencies:  Properties exposed:   * ***component*** the name of this component * ***required-version:*** the required version * ***tested-version:*** the tested version * ***description:*** the description * **license:** The type of license of this dependency |
| $local-dependency | <cms:release>  <cms: part-release>  <cms: dependencies>  <cms: local-dependency>  </cms: part-release>  </cms:release> | The list of dependencies:  Properties exposed:   * ***component*** the name of this component * ***required-version:*** the required version * ***tested-version:*** the tested version * ***required:*** true if this dependency is required * ***description:*** the description |
| $monitoredDependencies | <cms:release>  <cms: part-release>  <cms: dependencies>  <cms: monitoredDependencies >  </cms: part-release>  </cms:release> | The list of dependencies:  Properties exposed:   * ***component*** the name of this component * ***required-version:*** the required version * ***tested-version:*** the tested version * ***description:*** the description * **license:** The type of license of this dependency |
| $activities | <cms:release>  <cms: part-release>  <cms: activities>  </cms: part-release>  </cms:release> | The list of the activities for this releases:  Properties exposed:   * **releaseVersion:**the release number for these activities * **eprs:** the list of EPRs for this release version * **enhs:** the list of ENCs for this release version * **enhs:** the list of ENCs for this release version |
| $partRelease. CurrentReleaseActivities | <cms:release>  <cms: part-release>  <cms: activities>  </cms: part-release>  </cms:release> | The list of the activities for the current release:  Properties exposed:   * **releaseVersion:**the release number for these activities * **eprs:** the list of EPRs for this release version * **enhs:** the list of ENCs for this release version * **enhs:** the list of ENCs for this release version |
| $ baseline-summary | N/D | A summary for the baseline list. |
| $release.precautions | <cms:release>  <cms: release-version/>  </cms:release> | The list of precautions for this release. |
| $release.distributionList | <cms:release>  <cms: distribution-list/>  </cms:release> | The list of ML for this project. |

# Usage

## Maven plugin

Add this snippet in your pom.xml file:

|  |
| --- |
| <plugin>  <groupId>com.anritsu.maven</groupId>  <artifactId>masterclaw-release</artifactId>  <version> **-- VERSION --** </version>  <executions>  <execution>  <id>generate-releasenote-file</id>  <phase>package</phase>  <goals>  <goal>release-note-generation</goal>  </goals>  </execution>  </executions>  <configuration>  <verbose>false</verbose>  <!-- Generate documentation section -->  <activitiesFilename>changes.xml</activitiesFilename>  <releaseItems>  <releaseItem>  <templateName>finder-rn-template.doc</templateName>  <finalName>${artifactId}-${project.version}.doc</finalName>  </releaseItem>    <releaseItem>  <templateName>finder-mail.doc</templateName>  <finalName>${artifactId}-mail.doc</finalName>  </releaseItem>  </releaseItems>  <!-- Deploy Section -->  <ftpServer>utmnnfs</ftpServer>  <ftpUser>it100018</ftpUser>  <ftpPassword>it100018..</ftpPassword>  <tcUsername>${teamcity.username}</tcUsername>  <tcPassword>${teamcity.password}</tcPassword>  <dryRun>${masterclaw-release.dryRun}</dryRun>  <fileSets>  <fileSet>  <fileName>finder-${project.version}.doc</fileName>  <sourcePath>${teamcity.url}/FINDER\_LINUX\_TRUNK/.lastFinished</sourcePath>  <uploadPaths>  <uploadPath>ftp:///packages/eoFinder</uploadPath>  <uploadPath>file:///R:/eoFinder/client/release/rel\_notes</uploadPath>  </uploadPaths>  </fileSet>    [ … ]  </fileSets>  <!-- RPM generation Section -->  <rpmItems>  <!—Path to qbase packages ->  <rpmItem>  <item>finder-server/target/finder-srv-${finder.package.version}.tar.gz</item>  </rpmItem>    [ … ]  </rpmItems>  </configuration>  </plugin> |

The default source directory is **src/main/change**

The default output directory is **target/release-document**

## CLI program

Unpack the **zip/tar.gz**  and run:

### Document generator:

The CLI program to generate the release document:

|  |
| --- |
| bin/doc-generator <path of xml file> <path of docx template> <path of output file> [path of properties file] |

### Document download:

The CLI program to download the release document from a remote repository (i.e. TeamCity) in a local folder:

|  |
| --- |
| bin/doc-download <configuration file> <changes.xml> <download folder> <property file>  configuration file................... The XML file that contains the configuration  changes.xml.......................... The XML file that contains the list of activities  download folder...................... Path to download folder  property file........................ Property file. |

### Document deploy:

The CLI program to deploy the release document and packages to a remote repository (i.e. MODI) .

|  |
| --- |
| bin/doc-deploy <configuration file> <changes.xml> <path source folder> <path to properties file>  configuration file................... The XML file that contains the configuration  changes.xml.......................... The XML file that contains the list of activities  path source folder................... Path to folder that contains the document and packages to deploy  property file............................ Property file. |

### Configuration file example:

|  |
| --- |
| <configuration>  <verbose>true</verbose>  <dryRun>false</dryRun>  <skip>false</skip>  <!-- Generate documentation section -->  <activitiesFilename>changes.xml</activitiesFilename>  <releaseItems>  <releaseItem>  <templateName>finder-rn-template.doc</templateName>  <finalName>${project.artifactId}-${project.version}.doc</finalName>  </releaseItem>  <releaseItem>  <templateName>finder-str-template.doc</templateName>  <finalName>${project.artifactId}-str-${project.version}.doc</finalName>  </releaseItem>  <releaseItem>  <templateName>finder-mail.doc</templateName>  <finalName>${project.artifactId}-mail.doc</finalName>  </releaseItem>  </releaseItems>  <!-- Deploy Section -->  <ftpServer>utmnnfs</ftpServer>  <ftpUser>it100018</ftpUser>  <ftpPassword>it100018..</ftpPassword>  <tcUsername>${teamcity.username}</tcUsername>  <tcPassword>${teamcity.password}</tcPassword>  <fileSets>  <fileSet retrieve="true">  <fileName>finder-${project.version}.doc</fileName>  <sourcePath>${teamcity.url}/FINDER\_LINUX\_TRUNK/.lastFinished</sourcePath>  <uploadPath>ftp:///packages/eoFinder</uploadPath>  <uploadPath>file:///R:/eoFinder/client/release/rel\_notes</uploadPath>  <uploadPath>file:///C:/Sviluppo/document-generator/target/deploy</uploadPath>  </fileSet>  <fileSet retrieve="true">  <fileName>finder-mail.doc</fileName>  <sourcePath>${teamcity.url}/FINDER\_LINUX\_TRUNK/.lastFinished</sourcePath>  <uploadPaths/>  </fileSet>  <fileSet retrieve="true">  <fileName>finder-str-${project.version}.doc</fileName>  <sourcePath>${teamcity.url}/FINDER\_LINUX\_TRUNK/.lastFinished</sourcePath>  <uploadPath>file:///R:/eoFinder/client/release/str</uploadPath>  </fileSet>  <fileSet retrieve="true">  <fileName>finder-Linux-${project.version}.tar.gz</fileName>  <sourcePath>${teamcity.url}/FINDER\_LINUX\_TRUNK/.lastFinished</sourcePath>  <uploadPath>ftp:///packages/eoFinder</uploadPath>  </fileSet>  <fileSet retrieve="true">  <fileName>finderoff-Linux-${project.version}.tar.gz</fileName>  <sourcePath>${teamcity.url}/FINDER\_LINUX\_TRUNK/.lastFinished</sourcePath>  <uploadPath>ftp:///packages/eoFinder</uploadPath>  </fileSet>  <fileSet retrieve="true">  <fileName>finderoff-Win-${project.version}.exe</fileName>  <sourcePath>${teamcity.url}/FINDER\_WIN\_TRUNK/.lastFinished</sourcePath>  <uploadPath>ftp:///packages/eoFinder</uploadPath>  </fileSet>  </fileSets>  </configuration> |

### RPM file Generation:

The CLI program to generate an RPM file stating from a qbase package.

The program extracts the package name and package version from the name of qbase package: *<package-name>-<version>.<ext>*

|  |
| --- |
| bin/generate-rmp [options] <package name> <changes.xml> <target folder> [property file]  -v, --verbose................................. Print debug information  qbase package name................... The name of the tgz/tar.gz/tar.Z package.  changes.xml……........................... The XML file that contains the list of activities  target folder................................ Path to download folder  property file................................ Property file (used for substituting variables into xml file) |