

How is metadata about an archetype stored?

原型项目相关的元数据存储在archetype-metadata.xml文件中，其位于JAR文档META-INF/maven目录下

The metadata about an archetype is stored in the archetype-metadata.xml file located in the directory META-INF/maven of its JAR file, see the [reference documentation](#).

The metadata file stores the additional properties, with corresponding default values.

It also stores the project's generated files in filesets.

Finally it also stores inner modules of the archetype, which enable the creation of multi-module projects using a single archetype.

A minimal archetype-metadata.xml file looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<archetype-descriptor name="basic">
  <fileSets>
    <fileSet filtered="true" packaged="true">
      <directory>src/main/java</directory>
      <includes>
        <include>**/*.java</include>
      </includes>
    </fileSet>
  </fileSets>
</archetype-descriptor>
```

This example above shows:

- the archetype name is basic
- the archetype defines a single fileset:
 - the fileset will take all the files in archetype-resources/src/main/java that match the **/*.java pattern
 - the selected files will be generated using the Velocity engine (filtered=true)
 - the files will be generated in the src/main/java directory of the generated project in the same directory as in the JAR file, but with that directory prepended by the package property.

Defining additional properties 定义其他属性

The main properties that are used by the Velocity engine during a project's file generation are groupId, artifactId, version and package.

It is possible to define additional properties that must be valued before the file generation.

These additional properties can be provided with default values, which enable not to ask the user for there values.

Additional properties are defined in the archetype-metadata.xml file with:

```
<archetype-descriptor name="basic">
  <requiredProperties>
    <requiredProperty key="property-with-default">
      <defaultValue>default-value</defaultValue>
    </requiredProperty>
  </requiredProperties>
```

```

    <requiredProperty key="property-without-default"/>
  </requiredProperties>
  ...
</archetype-descriptor>

```

Here two additional properties are defined: `property-with-default` and `property-without-default`.

Note: The property keys can not contain any dots as they are Velocity properties.

Defining specific filesets 定义特殊的文件集

The `filesets` contained in the `archetype-metadata.xml` file defines the way the project's files located in the JAR file are used by the Archetype Plugin to generate a project.

Filesets must define the directory where the files will be searched for which is also the directory where the project's files will be generated. The first is the directory inside the archetype JAR file, the second is the directory in the generated project's tree.

Filesets also defines the inclusion/exclusion of files "à la" Ant. This provide a powerful way to describe a large set of files to be selected for the generation process.

Filesets can be filtered, which means the selected files will be used as Velocity templates. They can be non-filtered, which means the selected files will be copied without modification.

Filesets can be packaged, which means the selected files will be generated/copied in a directory structure that is prepended by the package property. They can be non-packaged, which means that the selected files will be generated/copied without that prepend.

A fileset is defined in the `archetype-metadata.xml` with this fragment:

```

...
<fileSets>
  <fileSet filtered="true" packaged="true">
    <directory>src/test/java</directory>
    <includes>
      <include>**/*.java</include>
    </includes>
    <excludes>
      <exclude>AllTest.java</exclude>
    </excludes>
  </fileSet>
</fileSets>
...

```

This example shows a fileset that will select all the java files in the `src/test/java` directory of the archetype resources, except the `AllTest.java` file that is located at the root of this directory.

This fileset is also packaged and filtered.

定义多模块项目

Defining multiple modules in the archetype metadata

Inner modules of an archetype are used to create a multi-module Maven project from a single archetype.

Modules in the `archetype-metadata.xml` file are defined like this:

```

<archetype-descriptor name="multi-module">
  <fileSets>
    ...
  </fileSets>
  <modules>
    <module name="SubProject" id="subproject" dir="sub-project">
      <fileSets>
        ...
      </fileSets>
    </module>
  </modules>
</archetype-descriptor>

```

In the example above, the archetype `multi-module` contains a module named `SubProject`. This module is located in the `sub-project` directory of the archetype. It also has the artifactId `subproject`.

The attributes `name`, `id` and `dir` of the module are used to determine the directory where the generated module's files will appear. They also are used to determine the artifactId of the Maven project corresponding to this module.

Putting it all together 综合上述功能

The `<requiredProperties>` element is only allowed as a child of `<archetype-descriptor>`.

Modules are allowed in `<archetype-descriptor>` and in `<modules>` (no limit is given).

`<archetype-descriptor>` and `<modules>` must define at least one `<fileSet>` each to be valid.

It is possible to define default values for required properties by defining say the `groupId` and giving it a default:

```

...
<requiredProperties>
  <requiredProperty key="groupId">
    <defaultValue>com.company.department</defaultValue>
  </requiredProperty>
</requiredProperties>

```

Filesets can be defined from the root directory of the module/project by having an empty `<directory>` element:

```

...
<fileSet filtered="true" packaged="true">
  <directory></directory>
  <includes>
    ...
  </includes>
  <excludes>
    ...
  </excludes>
</fileSet>

```