Using Properties in Deployment and Resolution

Introducing Matrix Parameters

Matrix parameters key-value pairs parameters separated by a semicolon (;) that you can place anywhere on a URI.

This is a standard method for specifying parameters in HTTP (in addition to querying parameters and path parameters).

For example:

`http://repo.jfrog.org/artifactory/libs-releases-local/org/libs-releases-local/org/jfrog/build-info-api/1.3.1/build-info-api-1.3.1.jar`**`;status=DEV;rating=5`**

Artifactory makes use of matrix parameters for:

1. Adding properties to artifacts as part of deployment
2. Controlling artifact resolution using matrix parameters

Page Contents

Dynamically Adding Properties to Artifacts on Deployment

You can add key-value matrix parameters to deploy (PUT) requests and those are automatically transformed to properties on the deployed artifact.

Since matrix parameters can be added on any part of the URL, not just at the end, you can add them to the target deployment base URL. At the time of deployment, the artifact path is added after the matrix parameters and the final deployed artifact will be assigned the defined properties.

You can even use dynamic properties, depending on our deployment framework.

When using Maven, for instance, you can add two parameters to the deployment URL: `buildNumber` and `revision`, which Maven replaces at deployment time with dynamic values from the project properties (e.g. by using the Maven build-number plugin).

So, if you define the distribution URL as:

```
http://myserver:8081/artifactory/qa-releases;buildNumber=${buildNumber};revision=${revision}
```

And deploy to the `qa-releases` repository a jar with the following path:

```
/org/jfrog/build-info-api/1.3.1/build-info-api-1.3.1.jar
```

Upon deployment the URL is transformed to:

```
http://myserver:8081/artifactory/qa-releases;buildNumber=249;revision=1052/org/jfrog/build-info-api/1.3.1/build-info-api-1.3.1.jar
```

And the deployed `build-info-api-1.3.1.jar` has two new properties:

```
buildNumber=249
revision=1052
```


Permissions to attach properties

You must have the 'Annotate' permission in order to add properties to deployed artifacts.

Controlling Artifact Resolution with Matrix Parameters Queries

Matrix parameters can also be used in artifact resolution, to control how artifacts are found and served.

There is currently support for two types of queries:

- Non-conflicting values
- Mandatory values.

Non-mandatory Properties

Resolved artifacts may either have no property with the key specified, or have the property with the key specified and the exact value specified (i.e. the artifact is resolved if it has a property with a non-conflicting value).

Non-mandatory properties are identified by a simple `key=value` parameter.

For example:

Current Artifact Property	Matrix Parameter	Resolution Result
color=black	color=black	OK (200)
None or height=50	color=black	OK (200)
color=red	color=black	NOT_FOUND (404)

Mandatory Properties

Resolved artifacts must have a property with the key specified and the exact value specified.

Mandatory properties are identified with a plus sign (+) after the property key: `key+=value`.

For example:

Current Artifact Property	Matrix Parameter	Resolution Result
color=black	color+=black	OK (200)
None or height=50	color+=black	NOT_FOUND (404)
color=red	color+=black	NOT_FOUND (404)