Configuration Files

Overview

All Artifactory configuration files are located under the `$ARTIFACTORY_HOME/etc` folder.

On Linux, Solaris and MacOS `$ARTIFACTORY_HOME` is usually a soft link to `/etc/artifactory`.

Global Configuration Descriptor

The global Artifactory configuration file is used to provide a default set of configuration parameters.

The file is located in *`$ARTIFACTORY_HOME/etc/artifactory.config.xml`* and is loaded by Artifactory at initial startup. Once the file is loaded, Artifactory renames it to *`artifactory.config.bootstrap.xml`* and from that point on, the configuration is stored internally in Artifactory's storage. This ensures Artifactory's configuration and data are coherently stored in one place making it easier to back up and move Artifactory when using direct database backups. On every startup, Artifactory also writes its current configuration to *`$ARTIFACTORY_HOME/etc/artifactory.config.latest.xml`* as a backup.

At any time, the default configuration can be changed in the Artifactory UI **Admin** module.

There are two ways to directly modify the Global Configuration Descriptor:

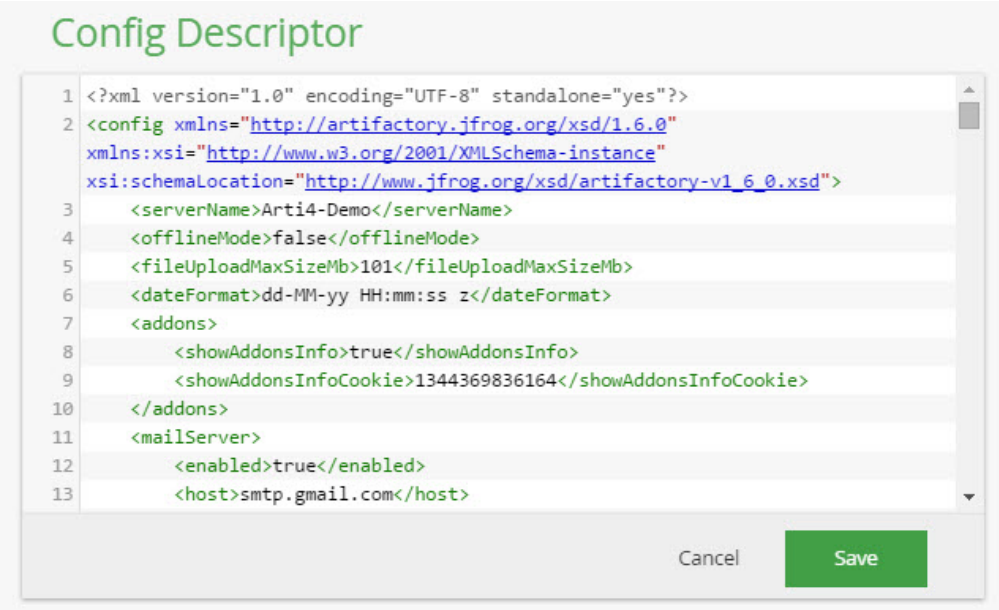
1. Using the Artifactory UI
2. Using the REST API

Page Contents

Modifying Configuration Using the UI

You can access the Global Configuration Descriptor in the **Admin** module under **Advanced | Config Descriptor.** There you can modify the file's contents directly or copy the contents from the entry field.

> Direct modification of the global configuration descriptor is an advanced feature, and if done incorrectly may render Artifactory in an undefined and unusable state. We strongly recommend backing up the configuration before making any direct changes, and taking great care when doing so.



Modifying Configuration Using the REST API

You can retrieve or set the global configuration by sending a GET or POST request to *http://<host>:<port>/artifactory/api/system/configuration*. For example:

Retrieving and Setting the Global Configuration Descriptor

```
curl -u admin:password -X GET -H "Accept: application/xml" http://localhost:8081/artifactory/api/system
/configuration
curl -u admin:password -X POST -H "Content-type:application/xml" --data-binary @artifactory.config.xml
http://localhost:8081/artifactory/api/system/configuration
```

Bootstrapping the Global Configuration

You can bootstrap Artifactory with a predefined global configuration by creating an *$ARTIFACTORY_HOME/etc/artifactory.config.import.xml* file containing the Artifactory configuration descriptor.

If Artifactory detects this file at startup, it uses the information in the file to override its global configuration. This is useful if you want to copy the configuration to another instance of Artifactory.

After you have restart and Artifactory did the import, the old file *$ARTIFACTORY_HOME/etc/artifactory.config.import.xml* will be moved to a file called *$ARTIFACTORY_HOME/etc/artifactory.config.bootstrap.xml*.

Security Configuration Descriptor

You can modify the Security Configuration Descriptor Using the REST API.

> Direct modification of the security descriptor is an advanced feature, and if done incorrectly may render Artifactory in an undefined and unusable state. We strongly recommend backing up the configuration before making any direct changes, and taking great care when doing so.

Modifying Security Using the REST API

You can retrieve or set the security configuration by sending a GET or POST request to *http://<host>:<port>/artifactory/api/system/security*. For example:

Modifying the Security Descriptor

```
curl -u admin:password -X GET -H "Accept: application/xml" http://localhost:8081/artifactory/api/system
/security
curl -u admin:password -X POST -H "Content-Type: application/xml" --data-binary @security.xml
http://localhost:8081/artifactory/api/system/security
```

> **Admin privileges**
>
> You must supply a user with **Admin** privileges to modify the security descriptor through the REST API

Bootstrapping the Security Configuration

Artifactory stores all security information as part of its internal storage.You can bootstrap Artifactory with a predefined security configuration by creating an *$ARTIFACTORY_HOME/etc/security.import.xml* file containing the Artifactory exported security configuration information.

If Artifactory detects this file at startup, it uses the information in the file to override all security settings. This is useful if you want to copy the security configuration to another instance of Artifactory.

Content Type/MIME Type

Artifactory provides a flexible mechanism to manage content type/MIME Type. You can define system-wide MIME types for common usage, but you can also overwrite the MIME types for specific files as needed. The list of default MIME types can be found in *$ARTIFACTORY_HOME/etc/mimetypes.xml* and can be edited in order to add, remove or change MIME types. If a file has an extension that is not supported by any of the MIME types, or does not have an extension at all, Artifactory will use the default MIME type of *application/octet-stream*. To determine an artifact's MIME type, Artifactory compares its extension with the those in the *mimetype.xml* file, and applies the MIME type of the first extension that matches.

MIME Type Attributes

Each MIME type may have the following attributes:

type	The MIME type unique name (mandatory)
extensions	A comma separated list of file extensions mapped to this MIME type (mandatory)
index	True if this MIME type should be indexed for archive searching (valid only for supported archive files)
archive	True if this MIME type is a browsable archive
viewable	True if this MIME type can be viewed as a text file inside Artifactory UI
syntax	The UI highlighter syntax to for this MIME type (only relevant if this is a `viewable` type)
css	The css class of a display icon for this mime type

Example of mimetype.xml

```
<mimetypes version="4">
    <mimetype type="text/plain" extensions="txt, properties, mf, asc" viewable="true" syntax="plain"/>
    <mimetype type="text/html" extensions="htm, html" viewable="true" syntax="xml"/>
    <mimetype type="text/css" extensions="css" viewable="true" syntax="css"/>
    <mimetype type="text/xsl" extensions="xsl" viewable="true" syntax="xml"/>
    <mimetype type="text/xslt" extensions="xslt" viewable="true" syntax="xml"/>
    <mimetype type="text/x-java-source" extensions="java" viewable="true" syntax="java"/>
    <mimetype type="text/x-javafx-source" extensions="fx" viewable="true" syntax="javafx"/>
</mimetypes>
```

For example, from the extensions parameter in the above *mimtypes.xml* file sample we can conclude that:

- *test.properties* is a *text/plain* MIME type
- *test.css* is a *text/css* MIME type
- *test.doc* is an *application/octet-stream* MIME type since "*doc*" is not included in any of the other MIME types).

IMPORTANT: Make sure you restart Artifactory for your changes to take affect.

Artifactory MIME Types

Some of the Mime-Types specified in *mimetypes.xml (e.g. application/x-checksum)* are used by Artifactory. Great care should be taken before changing these Mime-Types to ensure Artifactory continues to function correctly.

Setting Content-Type During Download

Using Artifactory, when downloading files you can override the *Content-Type* HTTP header by setting the *artifactory.content-type* property.

If the *artifactory.content-type* property is not explicitly set, Artifactory will use the default mechanism of matching the artifact name extension to the extensions in the *mimtypes.xml file* to apply the Content-Type

This feature is only available with Artifactory Pro.

System Properties

Rather than configuring properties in the JVM runtime configuration of the hosting container, you can edit *$ARTIFACTORY_HOME/etc/artifactory.system.properties* file and restart Artifactory.

The Artifactory system properties are documented within this file.

Since these settings impact the entire container VM, we recommend using this feature primarily for specifying Artifactory-related properties only (such as changing the database used by Artifactory, etc.).

Setting properties in *artifactory.system.properties* is an advanced feature and is typically not required.

Do not confuse these setting with those in the *$ARTIFACTORY_HOME/data/artifactory.properties* file, which are for internal use.

Logging Configuration Files

Artifactory uses the Logback Framework to manage logging and lets you configure the verbosity of log files. For details please refer to Configuring Log Verbosity

---

Storage Properties

Artifactory provides you with a *binarystore.xml* file so that you can configure the specific storage solution used in your system. For details please refer to Configuring the Filestore.