Artifactory Log Files

Overview

Artifactory uses the Logback Framework to manage logging. Activity is logged according to type in four different log files which can be found under the *ARTIFACTORY_HOME/logs* folder.

The following log files are available:

artifactory.log	The main Artifactory log file containing data on Artifactory server activity.
access.log	Security log containing important information about accepted and denied requests, configuration changes and password reset requests. The originating IP address for each event is also recorded.
request.log	Generic http traffic information similar to the Apache HTTPd request log.
import.export.log	A log used for tracking the process of long-running import and export commands.
sha256_migration.log	Logs status and errors when migrating the Artifactory database to include SHA256 values.

Page Contents

Tomcat/Servlet container-specific log files

When running Artifactory inside an existing servlet container, the container typically has its own log files.

These files normally contain additional information to that in `artifactory.log` or application bootstrapping-time information that is not found in the Artifactory logs.

In Tomcat, these files are *catalina.out* and *localhost.yyyy-mm-dd.log* respectively.

Configuring Log Verbosity

The verbosity of any logger in your system can be configured by entering or modifying the `level` value in the corresponding entry in the Logback configuration file *ARTIFACTORY_HOME/etc/logback.xml.*

For example:

Modifying the verbosity of a logger

```
<logger name="org.apache.wicket">
    <level value="error"/>
</logger>
```

Artifactory loads any changes made to the Logback configuration file within several seconds without requiring a restart.

Minimizing Output to catalina.out

When running Artifactory as a background service, Artifactory log messages are redirected to catalina.out which may cause this file to be over-inflated with content. To reduce the volume of logging to catalina.out we recommend adding a "threshold filter" to the "CONSOLE" appender in logback.xml as follows:

```
...

<appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">

     <!-- Add a Threshold filter to reduce log output that is below the specified threshold. In the example
below, only ERROR level log messages will be added -->
     <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
       <level>ERROR</level>
     </filter>


       <encoder class="ch.qos.logback.core.encoder.LayoutWrappingEncoder">
           <layout class="org.artifactory.logging.layout.BackTracePatternLayout">
               <pattern>%date ${artifactory.contextId}[%thread] [%-5p] \(%-20c{3}:%L\) - %m%n</pattern>
           </layout>
       </encoder>
   </appender>
...
```

Log File Structure

The Request and Access log files each display specific type of activity and as such have a consistent and specific file structure for maximum readability

Request Log

A request log file record has the following structure:

Date and Time stamp | Request time | Request type | IP | User name | Request method | Requested resource path | Protocol version | Response code | Request Content-Length

Note: If not provided by the client, 'Request Content-Length' is initialised as "-1".

Here is a typical example:

Request log file record sample

```
20140508154145|2632|REQUEST|86:12:14:192|admin|GET|/jcenter/org/iostreams/iostreams/0.2/iostreams-0.2.
jar|HTTP/1.1|200|8296
```

Date and time stamp	The date and time the request was completed and entered into the log file. Format is [YYYYMMDDHHMMSS]
Request time	The time in ms taken for the request to be processed
Request type	DOWNLOAD for a download request UPLOAD for an upload request REQUEST for any other request
IP	The requesting user's IP address
User name	The requesting user's user name or "non_authenticated_user" when accessed anonymously
Request method	The HTTP request method. e.g. GET, PUT etc.
Requested resource path	Relative path to the requested resource
Protocol version	The HTTP protocol version
Response code	The HTTP response code

Size (bytes) of request or response	If request method is GET: Size of response If request method is PUT or POST: Size of request

Access Log

An access log file record has the following structure:

Date and Time stamp | Action response and type | Repository path (Optional) | Message (Optional) | User name | IP

Here is a typical example:

Access log file record

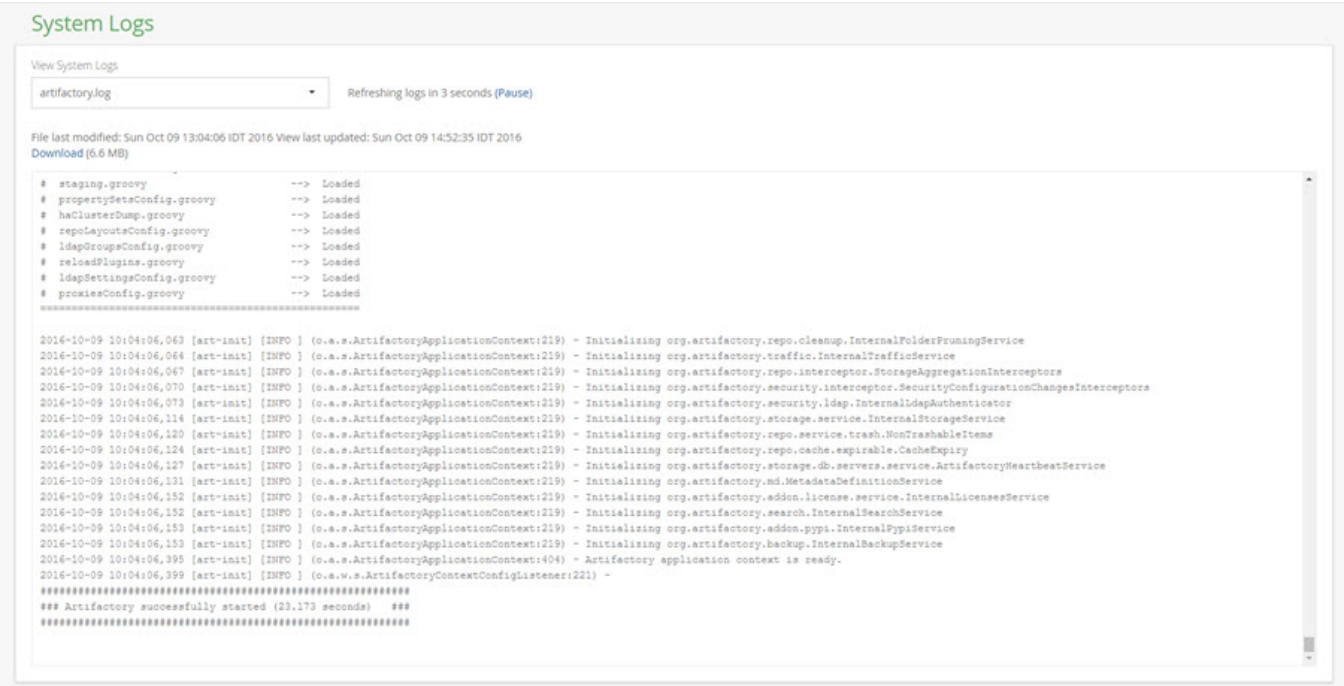
```
2014-05-08 15:52:27,456 [ACCEPTED DOWNLOAD] jcenter-cache:org/iostreams/iostreams/0.2/iostreams-0.2.jar for
anonymous/86:12:14:192.
```

Date and Time stamp	The date and time that the entry was logged. Format is [YYYY-MM-DD HH:MM:SS, milliseconds]
[Action response and type]	The response (ACCEPTED/DENIED) and the action type (e.g. DOWNLOAD, UPLOAD etc.)
Repository path (Optional)	The repository that was accessed
Message (Optional)	An optional system message
User name	The accessing user's user name or "anonymous" when accessed anonymously
IP	The accessing user's IP address

Viewing Log Files from the UI

You can view or download any of the Artifactory log files from the UI.

In the **Admin** module, under **Advanced | System Logs,** select the file you want to view from the drop-list. The log tail view is automatically refreshed every few seconds, however can be paused and resumed if you wish to browse the log.



To save system resources, do not leave the log view open in your browser unnecessarily.

Sending Artifactory Logs to Syslog

Some sites want to consolidate logs into the syslog facility. Switching artifactory to use syslog in addition to, or instead of the standard log files takes a quick edit of a couple of files. Artifactory currently uses the logback library for logging, so that's what needs to be configured.

First edit the $ARTIFACTORY_HOME/etc/logback.xml file to send logs to the syslog facility. You need to add an appender to syslog:

```
<appender name="SYSLOG" class= "ch.qos.logback.classic.net.SyslogAppender">
 <syslogHost>localhost</syslogHost>
 <facility>SYSLOG</facility>
 <suffixPattern>[%thread] %logger %msg</suffixPattern>
 </appender>
```

then you need to add this appender to the output, in the section:

```
<root>
 <level value="info"/>
 <appender-ref ref="CONSOLE"/>
 <appender-ref ref="FILE"/>
 </root>
```

add:

```
 <appender-ref ref="SYSLOG"/>
```

before the `</root>` line.

Save the file, you will not need to restart artifactory for this to take effect.

Since logback is using internet sockets, you have to make sure your syslog facility accepts them. Modern linux distributions are using the rsyslog daemon for syslogging. Ensure that the configuration for internet domain sockets is enabled, either by editing `/etc/rsyslog.conf` and uncommenting:

```
# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
```

or placing it in a file under `/etc/rsyslog.d` ending in .conf

Rsyslog will need restarting with `service rsyslog restart` for this to take effect.