

Creating a Maven proxy (mvnproxy)

Creating a Maven proxy

A Maven proxy can be thought of as a Maven mirror that aggregates one or more Maven repositories.

This document will walk through the setup of maven-proxy <http://maven-proxy.codehaus.org/>.

Additional installation instructions can be found at <http://docs.codehaus.org/display/MAVEN/MAVEN-PROXY>

Aggregation of proxies may not be your friend

A word of warning. Some proxies allow to aggregate sites (ie. you can tell Maven to download from proxy X and X will check the real repositories A, B and C for the files).

In your pom you have specified the repository A and in your settings have configured A to point to your maven-proxy installation X. Because the proxy aggregates the repositories, the artifacts from the B and C repositories are available even though you have not defined them in your pom.

Normally this would not show up as an issue. The problem manifests as soon as you start using SNAPSHOT version of artifacts, perhaps as part of doing some work on maven itself. Due to the aggregation you will find that you are using SNAPSHOT versions instead of the more stable released versions.

The recommendation is to not put SNAPSHOT repositories into your maven-proxy configuration.

If you want to proxy SNAPSHOT repositories then create a separate maven-proxy installation that just contains the SNAPSHOTs.

I think maven-proxy should have used the repository id as part of the mirror URL so that aggregation was never a side effect.

Prerequisites

Shell access to NUCLEUS

Shell access is needed to execute commands and to configure the maven proxy. This may be done in conjunction with a sysadmin if you do not have the required privileges.

Build 0.2 of maven-proxy

See <http://maven-proxy.codehaus.org/Downloads> for access to <http://dist.codehaus.org/maven-proxy/distributions/maven-proxy-standalone-0.2.zip>

maven-proxy.properties configuration file.

Up to date configuration properties file can be found at <http://maven-proxy.codehaus.org/Configuration>.

An already configured instance of the maven-proxy.properties for maven-proxy 0.2 configuration file is as below.

The only changes needed are:

- replace WEB_ROOT with the correct path details
- decide whether to allow non-local access to web configuration pages. If yes, then change serverName to the host details of the machine instead of localhost.

```
#####  
# This file IS NOT DOCUMENTED, it is taken from  
#  
maven-proxy/core/src/test/resources/org/apache/maven/proxy/config/PropertyLoaderTest1.properties  
# and configured to match  
http://docs.codehaus.org/display/MAVENUSER/Using+Maven+in+a+corporate+environment
```

```
#
# You will need to globally replace WEB_ROOT to be the full path to
where
# the maven2 repositories are located. As per the above document
these directories
# live under the web server root directory.
#
# If you want remote browsing of the maven-proxy repository then you
must
# configure serverName to be something other than localhost.
#####

##### GLOBAL SETTINGS
# This is where maven-proxy stores files it has downloaded
repo.local.store=WEB_ROOT/maven2_repositories/maven_proxy

#The port to listen on - not used if loaded as a webapp
port=9999

#This is the base area that all files are loaded from. While it is
possible to leave this blank, this behaviour
#is deprecated and will be disabled in version 2.0. There are too
many namespace conflicts caused by not using
#a prefix.
#The repository will be shown at http://localhost:9999/repository/
#for the .war loaded into a webapp server, the default prefix is
"repository" (edit the web.xml to change)
# As maven doesn't like a trailing slash, this address shouldn't have
one either.
prefix=repository

#This is the simple date format used to display the last modified date
while browsing the repository.
lastModifiedDateFormat=yyyy/MM/dd HH:mm:ss

##### SNAPSHOT HANDLING
#If you want the proxy to look for newer snapshots, set to true
snapshot.update=true

##### M2 METADATA HANDLING
#If you want the proxy to prevent looking for newer metadata, set to
false (default is true)
#metadata.update=false

##### M2 POM HANDLING
#If you want the proxy to look for newer POMs, set to true (default is
false)
pom.update=true

##### PROMOTION HANDLING
# **** NOT CURRENTLY IMPLEMENTED ****
#Promotion describes the process by which new artifacts are loaded to
global maven-proxy repository. It
```

```
# is designed to be used by "higher security installations" that do
not want to acquire artifacts from
# remote repositories without approval.
#
#If promotion handling is enabled, then the proxy will not download
remote artifacts without permission
# (local repositories with copy=false are considered to be local)
#
#Permission to download is granted via the Promotion menu which will
be enabled
# when promotion handling is enabled.
#
#If promotion is false, artifacts are sourced from any repository as
per normal.
#
#Promotion and snapshots: If promotion is enabled, snapshots are not
downloadable. The concept of using
# a snapshot in a production build (which is primarily what promotion
is for) is counterintuitive.
##
promotion=false

##### WEB INTERFACE
# This defines the absolute URL the server should use to identify
itself.
# This can often be determined automatically, but we recommend you
specify
# it explicitly to prevent problems during startup.
# The prefix will be added to this for the actual repository
# i.e. proxy available at http://localhost:9999/, repository at
http://localhost:9999/repository
serverName=http://localhost:9999

#If true, the repository can be browsed
browsable=true

#If true, the repository can be searched
searchable=true

#Not currently implemented. Will allow webdav access to the repository
at some point.
webdav=true

#Stylesheet - if configured, will override the default stylesheet
shipped with maven-proxy - absolute URLs only
#eg. /maven-proxy/style.css, http://www.example.com/style.css
stylesheet=/maven-proxy/style.css

#bgColor / bgColorHighlight are replaced in the built in stylesheet to
produce a simple color scheme.
#If a stylesheet is set, these are not used.
bgColor=#14B
bgColorHighlight=#94B
```

```
#rowColor / rowColorHighlight are replaced in the built in stylesheet
to produce a simple color scheme.
#If a stylesheet is set, these are not used.
rowColor=#CCF
rowColorHighlight=#DDF

##### PROXIES
#Note: Proxies are only needed for external repositories.
#If you need to use a proxy to get Internet access to an external
repository
#then in the repository definition section you must also specify which
proxy that repository should use.
#e.g. repo.www-ibiblio-org.proxy=one

#This is just a hack, it should auto discover them (but it currently
doesn't)
#proxy.list=one,two,three

#Unauthenticated proxy
#proxy.one.host=proxy1.example.com
#proxy.one.port=3128

#Authenticated proxy
#proxy.two.host=proxy2.example.org
#proxy.two.port=80
#proxy.two.username=username2
#proxy.two.password=password2

#Authenticated proxy
#proxy.three.host=proxy3.example.net
#proxy.three.port=3129
#proxy.three.username=username3
#proxy.three.password=password3

##### REPOSITORIES
#This is not just a hack, it specifies the order repositories should
be checked
#Note that the proxy adds a "/" which is why the urls aren't suffixed
with a "/"
repo.list=inhouse,inhouse_snapshot,external_free,external_non_free,ww
w-ibiblio-org,dist-codehaus-org

#inhouse
repo.inhouse.url=file:///WEB_ROOT/maven2_repositories/inhouse
repo.inhouse.description=inhouse released artifacts
#If copy is true, jars are copied from the store to the proxy-repo.
Only configurable for file:/// repos
repo.inhouse.copy=false
#If hardfail is true, any unexpected errors from the repository will
cause
```

```
#the client download to fail (typically with a 500 error)
repo.inhouse.hardfail=true
#Don't cache a file repository
repo.inhouse.cache.period=0

#inhouse_snapshot
repo.inhouse_snapshot.url=file:///WEB_ROOT/maven2_repositories/inhouse_snapshot
repo.inhouse_snapshot.description=inhouse snapshot artifacts
#If copy is true, jars are copied from the store to the proxy-repo.
Only configurable for file:/// repos
repo.inhouse_snapshot.copy=false
#If hardfail is true, any unexpected errors from the repository will cause
#the client download to fail (typically with a 500 error)
repo.inhouse_snapshot.hardfail=true
#Don't cache a file repository
repo.inhouse_snapshot.cache.period=0

#external_free
repo.external_free.url=file:///WEB_ROOT/maven2_repositories/external_free
repo.external_free.description=external free artifacts, but with annoying license restrictions so they can not be housed on ibiblio.
#If copy is true, jars are copied from the store to the proxy-repo.
Only configurable for file:/// repos
repo.external_free.copy=false
#If hardfail is true, any unexpected errors from the repository will cause
#the client download to fail (typically with a 500 error)
repo.external_free.hardfail=true
#Don't cache a file repository
repo.external_free.cache.period=0

#external_non_free
repo.external_non_free.url=file:///WEB_ROOT/maven2_repositories/external_non_free
repo.external_non_free.description=external non free artifacts
#If copy is true, jars are copied from the store to the proxy-repo.
Only configurable for file:/// repos
repo.external_non_free.copy=false
#If hardfail is true, any unexpected errors from the repository will cause
#the client download to fail (typically with a 500 error)
repo.external_non_free.hardfail=true
#Don't cache a file repository
repo.external_non_free.cache.period=0

#www.ibiblio.org
#Maven1 Repository
#repo.www-ibiblio-org.url=http://www.ibiblio.org/maven
#Maven2 Repository
repo.www-ibiblio-org.url=http://www.ibiblio.org/maven2
```

```
repo.www-ibiblio-org.description=www.ibiblio.org
#If hardfail is true, any unexpected errors from the repository will
cause
#the client download to fail (typically with a 500 error)
repo.www-ibiblio-org.hardfail=true
#Cache this repository for 1 hour
repo.www-ibiblio-org.cache.period=3600
repo.www-ibiblio-org.cache.failures=true
#If a Proxy is needed, which one?
repo.www-ibiblio-org.proxy=one

#dist.codehaus.org
repo.dist-codehaus-org.url=http://dist.codehaus.org
#If hardfail is true, any unexpected errors from the repository will
cause
#the client download to fail (typically with a 500 error)
repo.dist-codehaus-org.hardfail=false
#Cache this repository for 1 hour
repo.dist-codehaus-org.cache.period=3600
repo.dist-codehaus-org.cache.failures=true
```

```
#If a Proxy is needed, which one?  
#repo.dist-codehaus-org.proxy=one
```

Install maven-proxy on NUCLEUS

Get your systems administrator to install maven-proxy.

See <http://docs.codehaus.org/display/MAVEN/MAVEN-PROXY> for additional instructions on how to do this.

Run maven-proxy

Run maven proxy with your maven-proxy.properties file specified.

Update settings.xml

```

<mirrors>
  ...
  <mirror>
    <id>maven-proxy</id>
    <name>Maven-Proxy Mirror</name>
    <url>http://NUCLEUS:9999/repository/</url>
    <mirrorOf>central</mirrorOf>
  </mirror>
  <mirror>
    <id>maven-proxy</id>
    <name>Maven-Proxy Mirror</name>
    <url>http://NUCLEUS:9999/repository/</url>
    <mirrorOf>snapshots</mirrorOf>
  </mirror>
  <!-- Include these if you would also like to proxy internal
repositories -->
  <mirror>
    <id>maven-proxy</id>
    <name>Maven-Proxy Mirror</name>
    <url>http://NUCLEUS:9999/repository/</url>
    <mirrorOf>inhouse</mirrorOf>
  </mirror>
  <mirror>
    <id>maven-proxy</id>
    <name>Maven-Proxy Mirror</name>
    <url>http://NUCLEUS:9999/repository/</url>
    <mirrorOf>inhouse_snapshot</mirrorOf>
  </mirror>
  <mirror>
    <id>maven-proxy</id>
    <name>Maven-Proxy Mirror</name>
    <url>http://NUCLEUS:9999/repository/</url>
    <mirrorOf>external_free</mirrorOf>
  </mirror>
  <mirror>
    <id>maven-proxy</id>
    <name>Maven-Proxy Mirror</name>
    <url>http://NUCLEUS:9999/repository/</url>
    <mirrorOf>external_non_free</mirrorOf>
  </mirror>
</mirrors>

```

Run maven

Maven should now connect to your installed maven-proxy to download artifacts. Remember on the first run the proxy's cache will be empty so the first run will be slower than usual.

Why proxy internal repositories?

See [Why proxy internal repositories](#)