



Muse LinkResolver Install

30 August 2022

Document Version 0.0.1.1
Muse 2.8.0.1
Muse LinkResolver 1.2
Build 00



Notice

No part of this publication may be reproduced stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of MuseGlobal Inc.

Disclaimer

MUSEGLOBAL, INC. MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

Trademarks

MUSE IS A REGISTERED TRADEMARK OF MUSEGLOBAL, INC. OTHER PRODUCT NAMES AND SERVICE NAMES ARE THE TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNERS AND ARE USED FOR IDENTIFICATION ONLY.

museknowledge.com







Table of Contents

1.0 Overview	5
2.0 Muse LinkResolver Install	7
2.1 Hardware Requirements	7
2.2 Software Requirements	7
2.3 Installation Steps	10
2.3.1 Installing the Muse LinkResolver application as Web Application Archive	11
2.3.2 Installing the Muse LinkResolver application into Muse Embedded Apache Tomcat as external context	13





1.0

Overview

Muse LinkResolver is an application that uses the OpenURL protocol to provide seamless access to electronic full-text content across all databases and vendors for a provided citation. It uses the publisher knowledge base from Muse Central Index for finding the appropriate linking options.





2.0

Muse LinkResolver Install

Muse LinkResolver can run on any platform on which a Java Virtual Machine 8 or 11 can be installed.

2.1 Hardware Requirements

The following are the minimum and recommended hardware specifications for running Muse LinkResolver.

Minimum specification:

- ✎ 120 GB hard disk space;
- ✎ 8 GB RAM;
- ✎ Dual Core CPU or better processor;

Recommended specification:

- ✎ 250 GB hard disk space;
- ✎ 16 GB RAM;
- ✎ Quad Core processor or better processor / multiprocessors;

2.2 Software Requirements

The following environment software products are necessary to operate Muse LinkResolver.

All the environment software listed below has been tested with Muse LinkResolver, and supports it fully. This does not mean that Muse LinkResolver is not working on other environments not listed below. If the environment you need to install Muse LinkResolver on is not listed here, please contact the Muse Support team for further details.



1 Operating system

The following operating systems have been tested with Muse Proxy:

- Windows Windows 10;
- Linux (x64) – various flavors: Red Hat, CentOS, Debian, Ubuntu.

Note: The only shells that are currently supported are: `bash`, `csch` (for UNIX environments) and Windows command prompt (for Microsoft Windows environments). The provided UNIX scripts may also work with `tcsh`, `ash`. However, these are not tested and may not work as expected.

Note: A properly sized swap partition must exist.

2 Java Virtual Machine Software Development Kit (JDK).

It is recommended the installation of a Java Virtual Machine Software Development Kit (SDK, or also referred to as JDK) package because they contain tools not available in the Java Runtime Environment (JRE) package that can be used for monitoring and troubleshooting issues related to the Java Virtual Machine and the Muse Proxy software. For example, the `jstat` tool available in a JDK package displays performance statistics for an instrumented HotSpot Java Virtual Machine (JVM). The Java Virtual Machine Software Development Kit contains the JRE.

The following virtual machines and platform combinations have been tested with Muse Proxy:

- Intel x64 (Windows 10), JDK8, Eclipse Temurin JDK 8, JDK11, Eclipse Temurin JDK 11;
- Intel x64 Linux, JDK8, OpenJDK 8, OpenJDK 11;

Note: Java 8 is the minimum and recommended software specification for running the Muse LinkResolver application.

Note: It is recommended to install the JDK or OpenJDK version that matches the architecture, e.g. if the architecture is on 64 bit then the 64 bit JDK or OpenJDK must be installed.

a. OpenJDK Installation on Windows Platforms

- Download the latest build of the Windows OpenJDK version 8 from:

<https://adoptium.net/temurin/releases?version=8>

according to the architecture of the server, e.g. download the 64 bit version if the server's architecture is on 64 bit.

- Install the OpenJDK according to the installation instructions from here:

<https://adoptium.net/installation/windows/>



After the installation is performed, define the environment variable called `JAVA_HOME` pointing to the OpenJDK installation. Assuming the destination directory is `c:/OpenJDK1.8` set `JAVA_HOME=c:/OpenJDK1.8`.

Also it is recommended to add in the system `Path` environment variable the OpenJDK `bin` folder. For example, if your Java installation folder is `C:/OpenJDK1.8` then add the following string to the `Path` environment variable: `C:/OpenJDK1.8/bin`. Note that the separator between the entries from the `Path` variable is the semicolon (;). Details for how to update the `Path` variable for some Windows distributions can be found here:

<https://www.java.com/en/download/help/path.xml>

b. OpenJDK Installation on Linux Platforms

- Download the latest build of the Linux OpenJDK version 8 from:

<https://adoptium.net/temurin/releases?version=8>

according to the architecture of the server, e.g. download the 64 bit version if the server's architecture is on 64 bit.

The OpenJDK setup is a `tar.gz` archive, download the one which is appropriate.

- Install the OpenJDK according to the installation instructions from here:

<https://adoptium.net/installation/linux/>

After the installation is performed, define the environment variable called `JAVA_HOME` pointing to the JDK installation. Assuming the destination directory is `/usr/java/openjdk` run the following command :

```
export JAVA_HOME=/usr/java/openjdk
```

Add the line above in the file `/etc/profile` to make this setting permanent on Linux. The variable becomes globally available after the next login.

Also it is recommended to add in the system `PATH` environment variable the JDK `bin` folder. Details for how to update the `PATH` variable for Linux can be found here:

<http://www.java.com/en/download/help/path.xml>



3 Web server implementing Java Servlet and JavaServer Pages technologies.

The Muse LinkResolver application is a web application which runs inside a servlets container server such as Apache Tomcat, JBoss, WebLogic, GlassFish.

The Apache Tomcat server is the web server tested with the Muse LinkResolver application and this is further referred into documentation.

Apache Tomcat is developed in an open and participatory environment and released under the [Apache License version 2](#) and it is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies.

Apache Tomcat version 8.5 or higher must be installed for running the Muse LinkResolver application.

Tomcat 8.5 and 9 were also tested and they work fine, other Apache Tomcat versions were not tested, but they may work as well.

- ✎ Download the Apache Tomcat version 8.5 archive from:

<https://tomcat.apache.org/download-80.cgi>

according to the server's platform (e.g. if the server architecture is 64 bit then download the 64 bit archive)

- ✎ Extract the Apache Tomcat archive in the desired installation location. Recommended locations:
 - ✎ `c:` drive on Windows platforms.
 - ✎ `/opt/` folder on Linux platforms.

Note: On Linux platforms make sure that the script files (`*.sh`) from the `bin` folder from the Apache Tomcat installation folder are executable.

2.3 Installation Steps

Muse LinkResolver is installed into the servlets container by 2 methods:

- ✎ Independently of the Muse Federated Search Platform, as Web Application Archive (`.war` file).
- ✎ Part of the Muse Federated Search Platform, into the Embedded Apache Tomcat as external



context.

Note: Before going further with the installation make sure the Hardware and Software requirements are met.

Note: On following sections and chapters you will see `LINKRESOLVER_HOME` very often and represents the location where Muse LinkResolver application is installed, which can be `MUSE_HOME/linkresolver/` if is installed in the Embedded Apache Tomcat from the Muse Federated Search Platform or `CATALINA_HOME/webapps/muse-linkresolver` if is installed as Web Application Archive in `CATALINA_HOME/webapps` folder.

Note: Any modification made in the following files require a server restart:

- 1 `LINKRESOLVER_HOME/WEB-INF/classes/application.properties`
- 2 `LINKRESOLVER_HOME/WEB-INF/classes/linkresolver-application.properties`
- 3 `LINKRESOLVER_HOME/WEB-INF/classes/linkresolver-mapping.properties`
- 4 `LINKRESOLVER_HOME/WEB-INF/classes/authentication.properties`
- 5 `LINKRESOLVER_HOME/WEB-INF/classes/logback.xml`

2.3.1 Installing the Muse LinkResolver application as Web Application Archive

The Muse LinkResolver application is distributed as a Web Application Archive (`.war` file) that is deployed inside a web server implementing the servlets specification.

The steps for installing the Muse LinkResolver application into Apache Tomcat web server using the `war` archive are below:

- 1 Deploy into `CATALINA_HOME/webapps` the Muse LinkResolver Web Application Archive `muse-linkresolver.war` file.
- 2 If Security Manager is enabled for the Apache Tomcat server then the following entry need to be added in `CATALINA_HOME/conf/catalina.policy` at the end of the file:

```
grant codeBase "file:${catalina.base}/webapps/muse-linkresolver/" {  
    permission java.security.AllPermission; };
```

- 3 Stop the Apache Tomcat server:



```
CATALINA_HOME/bin/shutdown.{bat|sh}
```

After the shutdown command was issued check the system's processes to make sure the Apache Tomcat process has really stopped, and only when the process has stopped issue the startup command.

- 4 Make sure that the following structure of properties files with configurations for the application, are properly configured:

✎ Edit the

`CATALINA_HOME/webapps/muse-linkresolver/WEB-INF/classes/linkresolver-application.properties` file and make sure the following properties are configured with correct values:

`spring.data.solr.host` - The URL to the index service of Muse Central Index.

`solr.document.collection` - The name of the collection providing the indexed items level metadata, e.g. articles, bookitems, preprints, proceedings, book chapters.

`solr.document.title.collection` - The name of the collection providing the indexed titles level metadata, e.g. journals, books, conferences.

`spring.data.mongodb.uri` - The URI for the Muse Central Index database.

`spring.data.mongodb.collection` - The name of the Muse Central Index database collection storing the items level metadata.

`spring.data.mongodb.title.collection` - The name of the Muse Central Index database collection storing the titles level metadata.

Important: All the above values for the Muse Central Index components must be confirmed with the Muse Support team.

✎ Edit the

`CATALINA_HOME/webapps/muse-linkresolver/WEB-INF/classes/authentication.properties` file and configure the desired end-user authentication method into the application. More details are in the "2.2 Authentication" chapter from the `Muse LinkResolver.pdf` manual.

- 5 Start the Apache Tomcat server:

```
CATALINA_HOME/bin/startup.{bat|sh}
```



Upon startup the Muse LinkResolver application archive is deployed, this should take a couple of seconds. Verbose information is written in the Apache Tomcat console.

- 6 The access URL for the Muse LinkResolver application is of the form:

```
http://Installation_HOSTNAME:PORT/muse-linkresolver
```

where replace `Installation_HOSTNAME` with the actual server hostname and `PORT` with the actual server port.

A Muse LinkResolver formatted page will display however without any information since an OpenURL URL is expected.

As a good practice for stopping the Muse LinkResolver application follow the rules:

- ✎ Use only the shutdown/startup sequence via Tomcat's scripts, do not terminate the process by kill/end task actions;
- ✎ After the shutdown command was issued wait as long as it takes until the Apache Tomcat has stopped. This may take even a couple of minutes, to verify that it stopped check the system's processes to make sure the Apache Tomcat is gone.

2.3.2 Installing the Muse LinkResolver application into Muse Embedded Apache Tomcat as external context

The Muse LinkResolver application is also distributed with the Muse Federated Search Platform, as external context for the Embedded Apache Tomcat version 8.5.

The steps are:

- 1 Make sure that the Muse LinkResolver application is installed on machine under `MUSE_HOME/linkresolver/` folder.
- 2 Stop the Embedded Apache Tomcat server:

```
CATALINA_HOME/bin/shutdown.{bat|sh}
```

After the shutdown command was issued check the system's processes to make sure the Apache Tomcat process has really stopped, and only when the process has stopped issue the startup command.

- 3 If Security Manager is enabled for Apache Tomcat, then the following entry must be added in



`MUSE_HOME/aas/java.tomcat.policy` at the end of file:

```
grant codeBase "file:${MUSE_HOME}/linkresolver/www/-" { permission
java.security.AllPermission; };
```

- 4 Add following entry in `CATALINA_HOME/conf/server.xml` configuration file:

```
<Context path="/linkresolver" docBase="${MUSE_HOME}/linkresolver/www"
crossContext="true" xmlnsNamespaceAware="false" xmlValidation="false"/>
```

inside the `Host` tag.

- 5 Make sure that the following structure of properties files with configurations for the application, are properly configured:

✎ Edit the

`CATALINA_HOME/webapps/muse-linkresolver/WEB-INF/classes/linkresolver-application.properties` file and make sure the following properties are configured with correct values:

`spring.data.solr.host` - The URL to the index service of Muse Central Index.

`solr.document.collection` - The name of the collection providing the indexed items level metadata, e.g. articles, bookitems, preprints, proceedings, book chapters.

`solr.document.title.collection` - The name of the collection providing the indexed titles level metadata, e.g. journals, books, conferences.

`spring.data.mongodb.uri` - The URI for the Muse Central Index database.

`spring.data.mongodb.collection` - The name of the Muse Central Index database collection storing the items level metadata.

`spring.data.mongodb.title.collection` - The name of the Muse Central Index database collection storing the titles level metadata.

Important: All the above values for the Muse Central Index components must be confirmed with the Muse Support team.

✎ Edit the

`MUSE_HOME/linkresolver/WEB-INF/classes/authentication.properties` file and configure the desired end-user authentication method into the application. More details are in the "2.2 Authentication" chapter from the `Muse LinkResolver.pdf` manual.

- 6 Start the Apache Tomcat server:



```
MUSE_HOME/tomcat/bin/startup.{bat|sh}
```

Upon startup the Muse LinkResolver application is initialized, this should take a couple of seconds. Verbose information is written in the Embedded Apache Tomcat console.

- 7 The access URL for the Muse LinkResolver application is of the form:

```
http://Installation_HOSTNAME:PORT/linkresolver
```

where replace `Installation_HOSTNAME` with the actual server hostname and `PORT` with the actual server port.

A Muse LinkResolver formatted page will display however without any information since an OpenURL URL is expected.

