

# JAR Packaging report

Workgroup number: E8.01

Repository: <https://github.com/javiervaz01/Acme-Toolkits>

Date: 2022/06/01

Name	Corporate e-mail
Carrasco Núñez, Alejandro	<a href="mailto:alecarnun@alum.us.es">alecarnun@alum.us.es</a>
Durán Terrero, Andrés	<a href="mailto:anddurter@alum.us.es">anddurter@alum.us.es</a>
López Benítez, Pablo Delfín	<a href="mailto:pablopben@alum.us.es">pablopben@alum.us.es</a>
Núñez Moreno, Pablo	<a href="mailto:pabnunmor@alum.us.es">pabnunmor@alum.us.es</a>
Robledo Campa, Pablo José	<a href="mailto:pabrobcam@alum.us.es">pabrobcam@alum.us.es</a>
Vázquez Monge, Francisco Javier	<a href="mailto:fravazmon@alum.us.es">fravazmon@alum.us.es</a>

# Table of Contents

<b>Executive summary</b>	<b>3</b>
<b>Revision table</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>Contents</b>	<b>6</b>
<b>Conclusions</b>	<b>12</b>
<b>Bibliography</b>	<b>13</b>

## Executive summary

The following document contains a report detailing, step-by-step, the approach that we have followed to package the Acme Framework (In this case the 22.7 version, the latest one until the date of the creation of the report). Note that such a packaging procedure has not resulted in a completely standalone version of the Acme-Toolkits project, given the relatively complex dependencies with the Acme-Framework project.

## Revision table

Revision number	Date	Description
v1	2022/05/27	Initial version
v2	2022/06/01	Add disclaimer

## Introduction

The contents of this report detail the required steps for exporting the Acme Framework project as a standalone .jar file. This results in great benefits from the point of view of portability and compatibility, since we can ensure that we have a working version of the software in a compact file that we can deliver to the end user.

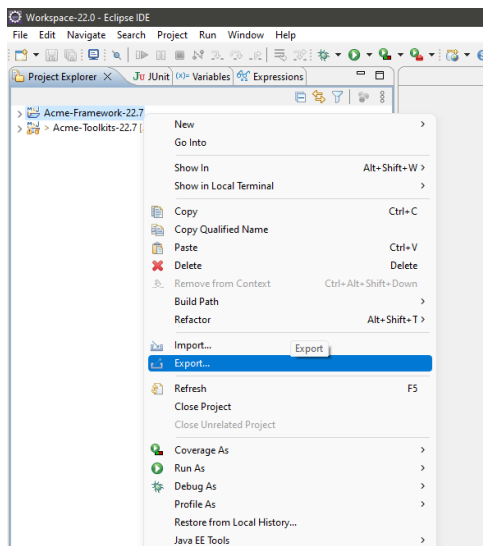
# Contents

The procedure for packaging and using the Acme Framework project as a .jar is as follows:

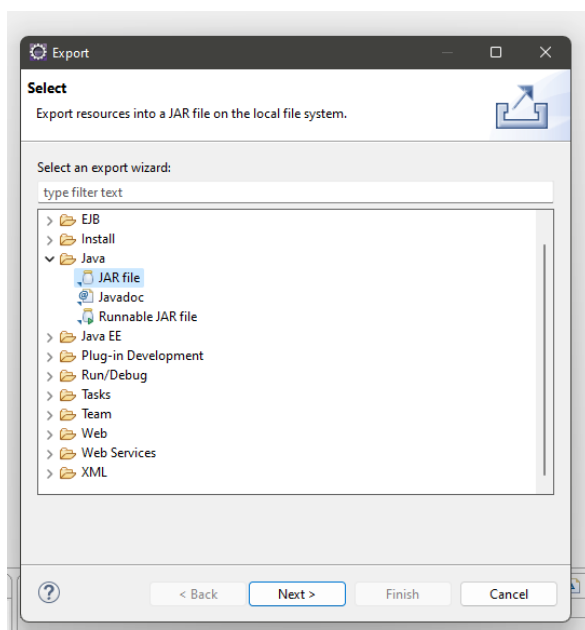
1. Open the Eclipse IDE



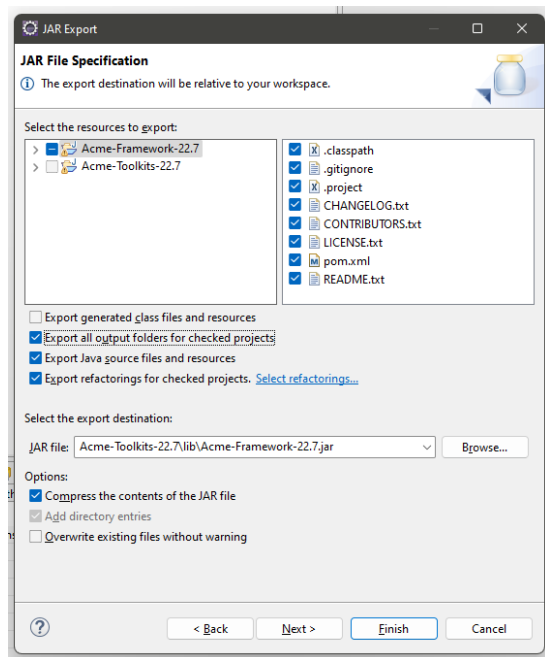
2. Right click on the corresponding Acme-Framework project and click on "Export"



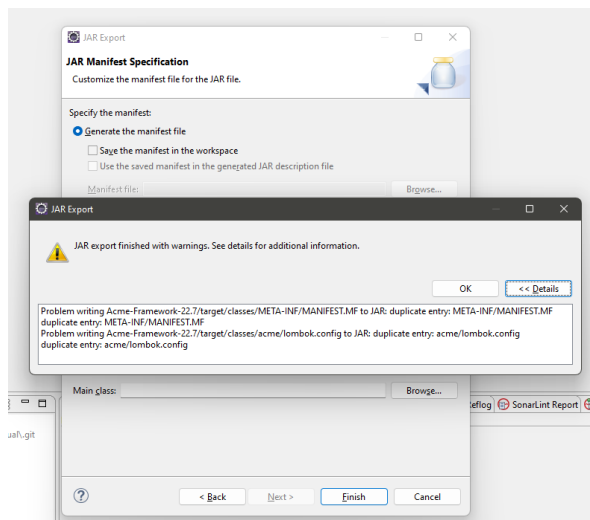
3. Click on "JAR file"



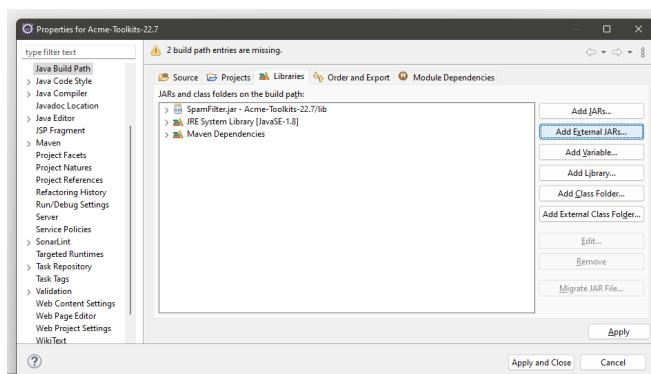
4. Leave the “JAR File Specification” as it is. Specify an export destination as a local route



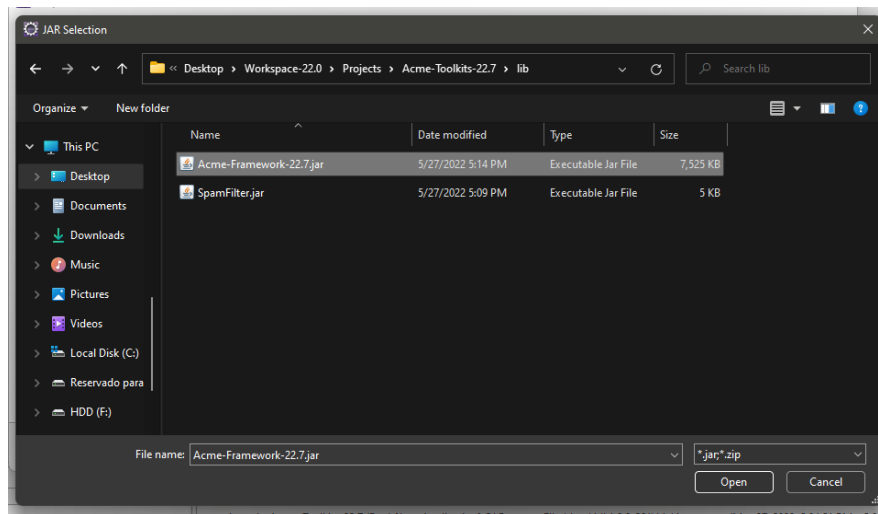
5. Click on “Finish”. Ignore the following warnings on duplicate entries.



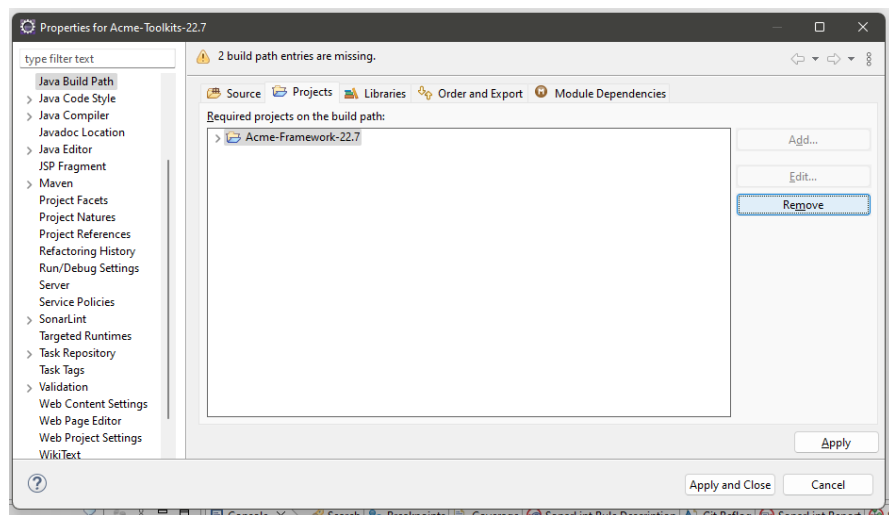
6. Click on “Libraries” > “Add External JARs...”



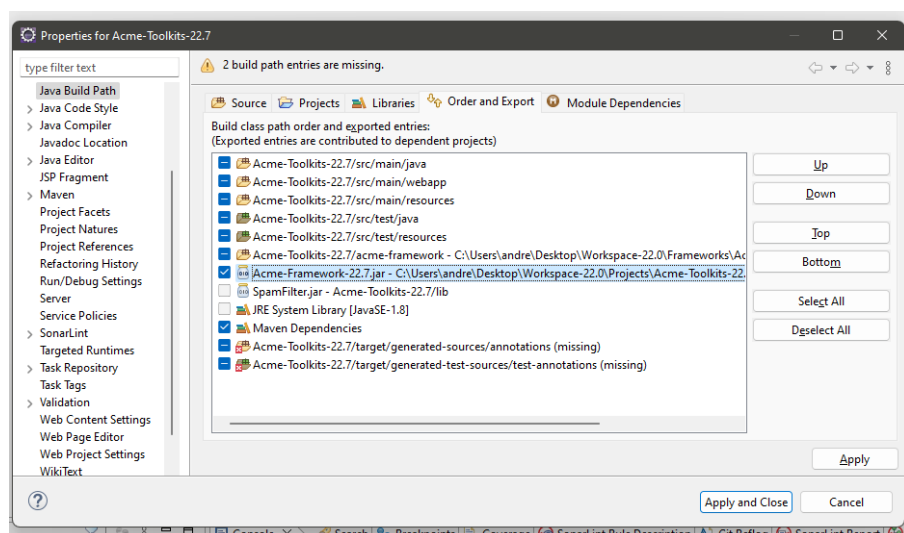
7. Select the Acme-Framework JAR file, which should be inside the project's folder



8. Click on “Projects” > “Remove” to remove the dependency to the Acme-Framework project

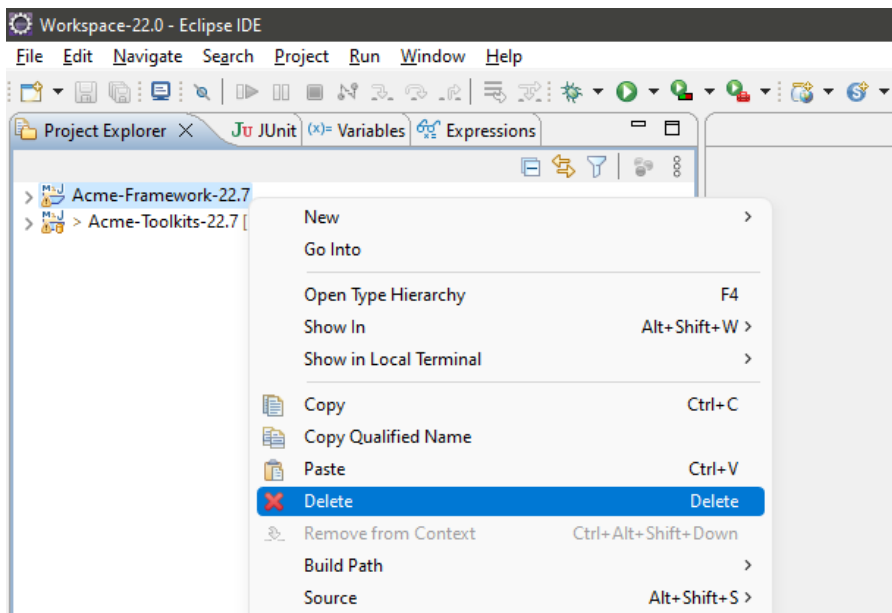


9. Click on “Order and export”, check the Acme-Framework JAR entry and, preferably, place it under the rest of Acme-Toolkits entries. Finally, click on “Apply and Close”

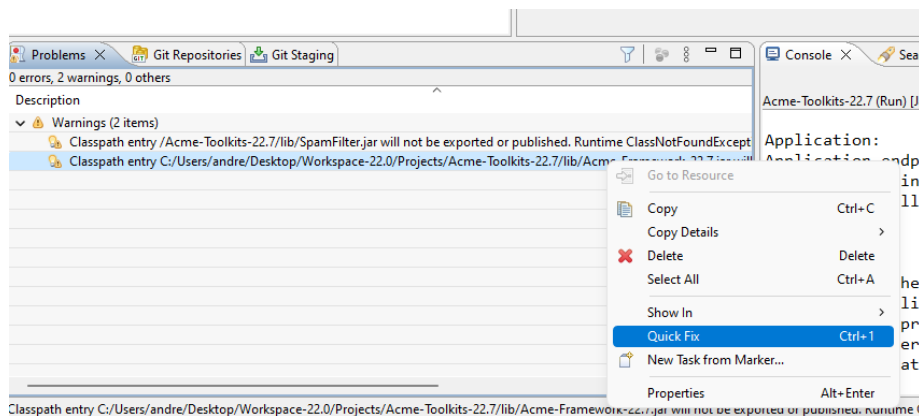




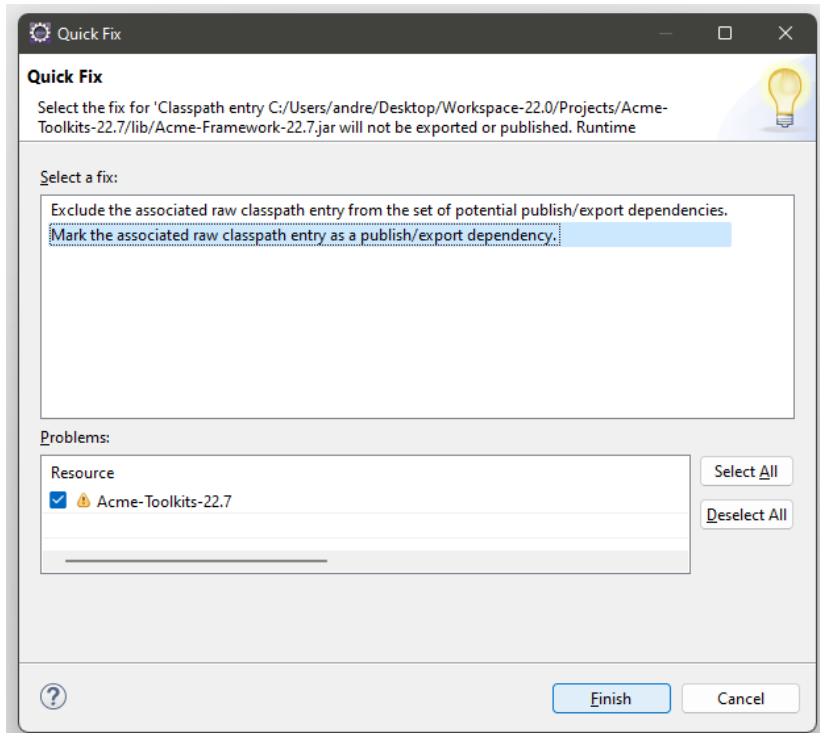
## 10. Delete the Acme-Framework project from Eclipse



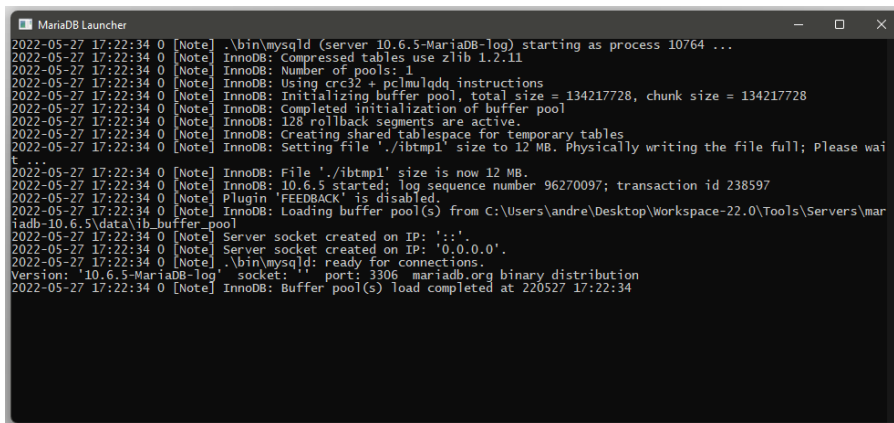
## 11. (Optional) Right click on the warnings and then on "Quick fix"



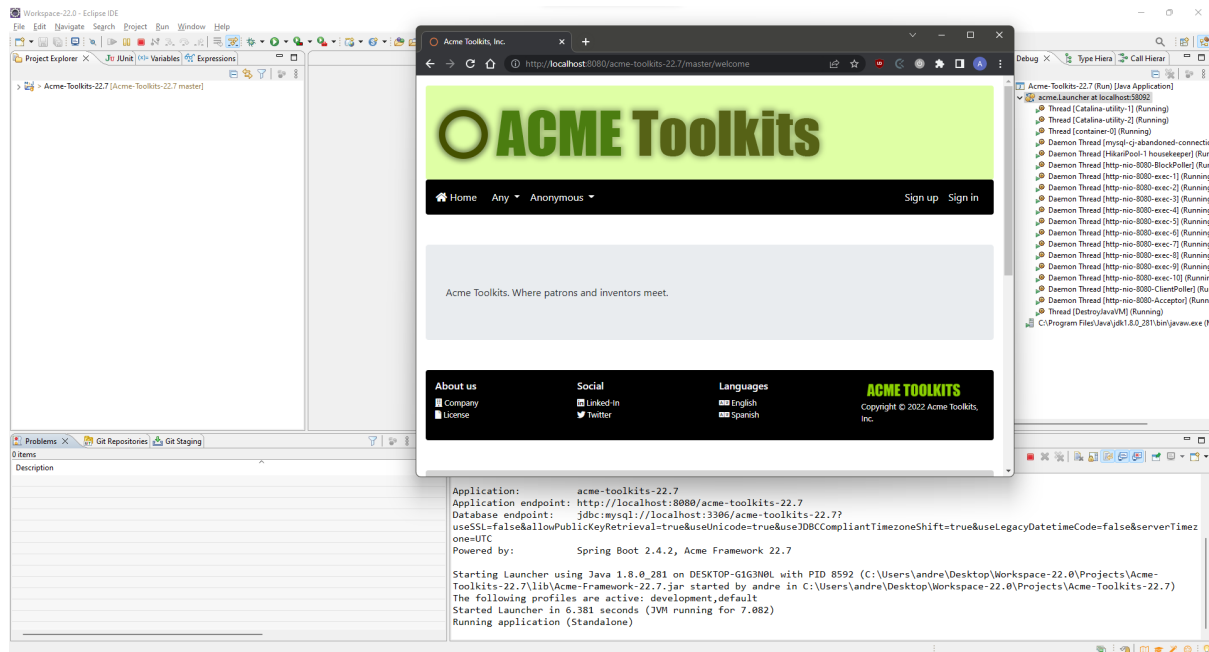
## 12. (Optional) Click on "Mark the associated raw classpath entry as publish/export dependency" and then "Finish" to mend the warning



### 13. Open the provided MariaDB Launcher



### 14. Run the application and ensure that everything is working as expected



### Disclaimer:

*Packaging the Acme Framework as an independent .jar component has been relatively doable. However, and even though we have been able to minimize the cohesion between the framework and the project, the Acme Toolkits project still depends on the existence of the “Frameworks/Acme-Framework-22.7” folder. Even if the project is executed correctly, removing this dependency results in a crash when a page is requested from the server. For documentation purposes, the log when entering the main page is the following:*

```
java.lang.AssertionError: Couldn't locate view ['/master'] with root cause at
acme.framework.components.tiles.DynamicTilesView.renderMergedOutputModel(DynamicTil
esView.java:171): Couldn't locate view ['/master']
```

*The amount of time that has been dedicated to this task has been fixed to 2 hours. Given that this provides little-to-no value to the client, spending more time on this requirement would result in a lower benefit for the organization without a justified cause. Since the value-time ratio that we provide to the client is so low for this requirement, the most intelligent decision from a managerial point of view is to provide this partial solution to the problem and focus on the most valuable requirements.*

## Conclusions

In conclusion, the amount of effort put in exporting the framework has not been outstanding. We have fulfilled this requirement by obtaining, as we are requested, a .jar file of the Acme-Framework project, minimizing the cohesion between our project and the Framework folder. However, we have had to learn this from the external sources mentioned in the bibliography due to the lack of explanations on this topic in the course slides.

## Bibliography

*Importing and Exporting Projects as JAR Files.* (s. f.). University at Albany. Fetched on May 27, 2022, at <https://www.albany.edu/faculty/jmower/geog/gog692/ImportExportJARFiles.htm>