Configuring source code delivery for Maven for CAST Imaging

Introduction

When **adding a new version** to analyze an Application that includes **Maven based source code**, you have several choices with regard to specifying where the required Maven repositories are located. The location of the repository is crucial to ensure that any associated JAR files can be automatically discovered and that POM dependencies can also be located. You can do as follows:

- You can include the Maven repository when you deliver the source code (i.e. in the ZIP or in the designated source code folder). Place the contents of the Maven repository (using the same file structure) at the root of the ZIP, for example:

```
D:\temp
        |-----JEE-Java
        |-----MavenRepo
        |-----OtherTechno2
```

- You can define local Maven repositories for a single Application
- You can define remote HTTP/S Maven repositories for a single Application
- You can define local Maven repositories for all AIP Nodes/Applications
- You can define remote HTTP/S Maven repositories for all AIP Nodes/Applications

AIP Console will also use the **above order to prioritise the various repositories**. In other words, if you include a repository in the ZIP or in the designated source code folder this will be used instead of any local or remote repositories that have been defined at global or Application level.

Define a repository at Application level - local .m2 and remote HTTP/S repositories

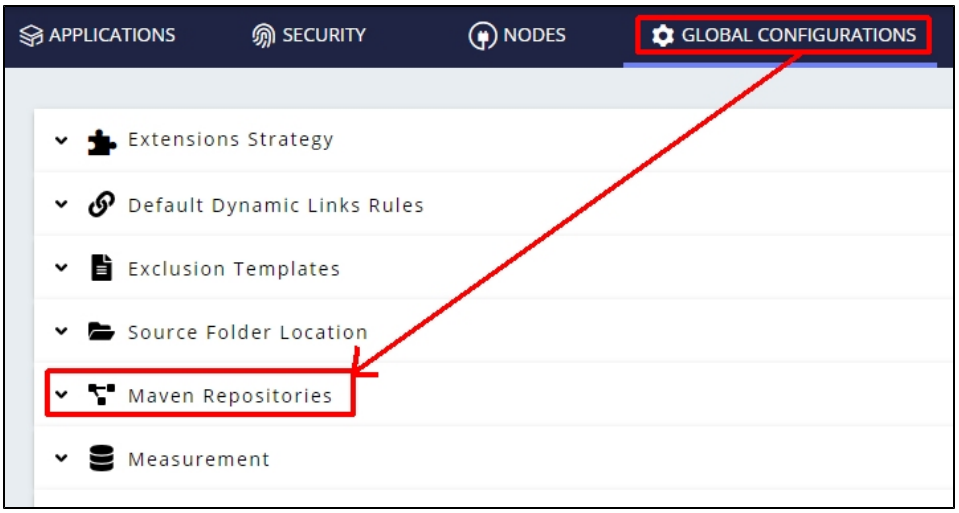
> Repositories discovered in the uploaded source code and any local repositories **always take priority** over any local/remote HTTP/S repositories.

Local and **remote repositories** can be defined at **Application level** and are **valid only for the specific Application**:

Define a repository at global level for all AIP Nodes/Applications - local .m2 and remote HTTP/S repositories

Repositories discovered in the uploaded source code, any repositories defined at Application level and any local repositories **always take priority** over any local/remote HTTP/S repositories.

Local and **remote repositories** can be defined using the **Administration Center** and are **valid for all AIP Nodes/Applications managed by AIP Console**:



Format of the various folders created after delivering the source code

When you configure a local or remote repository, the folders that are created in the Deployment folder after delivering the source code will be as follows:

Repo type	Folders
local (i.e. path of ".m2" repository)	• First folder with name "mainSource". • Second folder with name "mavenPck" containing the jar files.
http (local/remote)	• First folder with name "mainSource". • Second folder with name "mavenHTTPPck" containing the jar files.
both local and http	• First folder with name "mainSource". • Second folder with name "mavenPck" containing the jar files. • Second folder with name "mavenHTTPPck" containing the jar files.

Troubleshooting issues accessing remote HTTPS repositories

In certain situations, an error may be registered in the Delivery log when the AIP Node attempts to access an HTTPS repository. For example, in the log located at `delivery\{app-guid}\data\{guid}\{guid}\{guid}\DMTDeliveryReport.CastLog2:`

```
ERROR cast.dmt.engine.extractor.jee.maven.http.connectionFailed Unkown format id: cast.dmt.engine.extractor.jee.
maven.http.connectionFailed =>  %URL%="https://my.maven.repo/artifactory/maven-release/"
%MESSAGE%="sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.
SunCertPathBuilderException: unable to find valid certification path to requested target"
javax.net.ssl.SSLHandshakeException:sun.security.validator.ValidatorException: PKIX path building failed: sun.
security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested
target at sun.security.ssl.Alerts.getSSLException(:0).
```

The error reported in the log is generated by the **DeliveryManagerTool-CLI.exe** tool located on the **AIP Node**. This tool uses the Java JRE delivered with AIP Core in the following location:

```
%PROGRAMFILES%\CAST\<release>\jre\
```

This error usually occurs if the remote HTTPS repository you have defined is using an SSL certificate:

- where the signing authority is not listed in the Java JRE **cacerts** file located at `%PROGRAMFILES%\CAST\<release>\jre\security\cacerts`
- that is **self-signed**

Resolving the issue involves importing the required SSL certificates into the Java JRE delivered with AIP Core - this is out of the scope of this document.