Deploying the Payment and Customer Services Spring Boot apps

Introduction:

CBSA has two Spring Boot applications provided, for the Customer Services and Payment interfaces. These applications allow easy use of the restful API in CBSA with simple clean interfaces. There are multiple steps (listed below) that need to be actioned in order to get them up and running.

Assumptions:

- Maven is utilised during deployment (instructions are included below).
- Follow all of the steps listed below.
- The screenshots in this document reflect installation/deployment into a host called nazare-cbsa-test1.fyre.ibm.com utilising a port number of 19080. Please adjust these to reflect your installation's host and port number.

Changing the port to match z/OS Connect EE

The Spring Boot applications run inside a WebSphere Liberty Profile JVM server inside CICS, but also communicate with z/OS Connect EE.

It is important to make sure that the connection information for these is correct. As a default, these are set to port 30701 and host localhost.

Should you need to change these, they are configured in

```
/customerservices/src/main/java/com/ibm/cics/cip/bank/springboot/customerservices/ConnectionInfo.java
```

```
/paymentinterface/src/main/java/com/ibm/cics/cip/bank/springboot/paymentinterface/ConnectionInfo.java
```

Both need to be changed and both currently contain the following lines.

```
private static int port = 30701;
```

```
private static String address = "localhost";
```

Maven:

With z/OS Explorer, you need to install the Maven for Eclipse plug-in. This allows you to run Maven commands from within Eclipse, instead of using the command line.

https://github.com/eclipse-m2e/m2e-core/blob/master/README.md#-installation

You also need Maven installed on your laptop. Maven is a dependency management tool which is provided by Apache.

https://maven.apache.org/install.html

Start the z/OS Explorer. You can now import the Spring Boot applications into the workspace.

File → Import → Existing Projects into Workspace

"customerservices" and "paymentinterface"

These are "Maven" projects, denoted by the small M to the top left.

For each of the Maven projects, right click and select Run As → Maven Install. This will build and package the applications as "war" files.

IMPORTANT NOTE: If you need to repeat the step (for example, after changing the code, you need to Run As → **Maven Clean** first).

Troubleshooting:

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.8.1:compile (default-compile) on project paymentinterface: Compilation failure
```

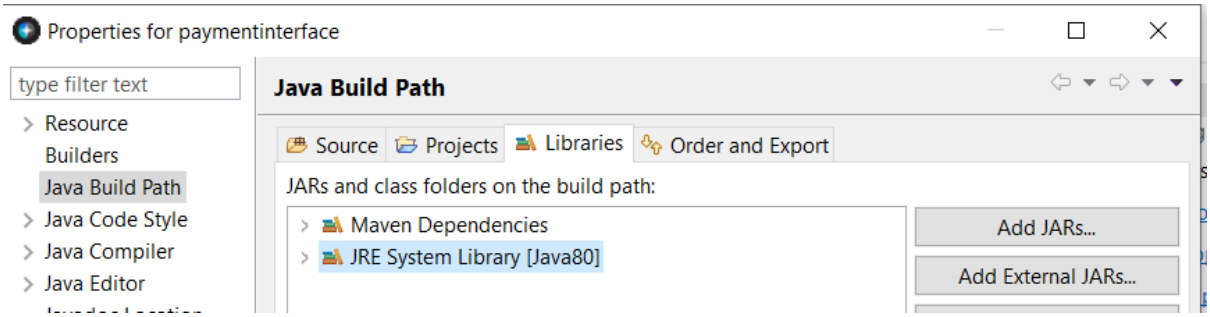
```
[ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?
```

If you receive this error, then you need to configure the build path of the project. The Explorer comes with a Java Runtime Environment (JRE) but **NOT** a Java Development Kit (JDK). You will need a JDK to compile Java classes. If you already have one installed, you can select that instead, otherwise you need to install one from "IBM Java Information Manager" or JIM.

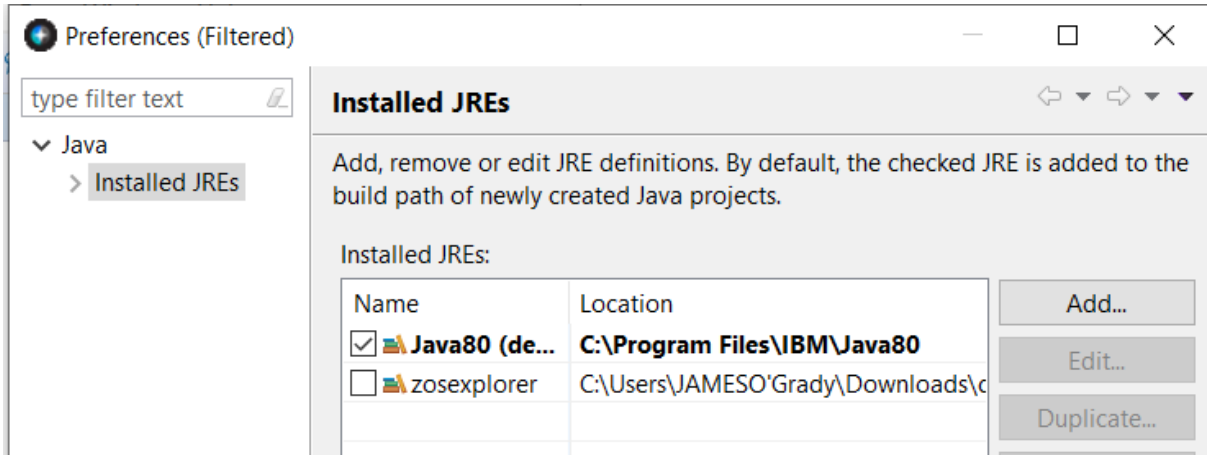
https://w3.hursley.ibm.com/java/jim/#content-main

Right click the project (the Maven folder with the J) and select Build Path → Configure Build Path

In Libraries, select the JRE and the select Edit. You may need to scroll down to find it.

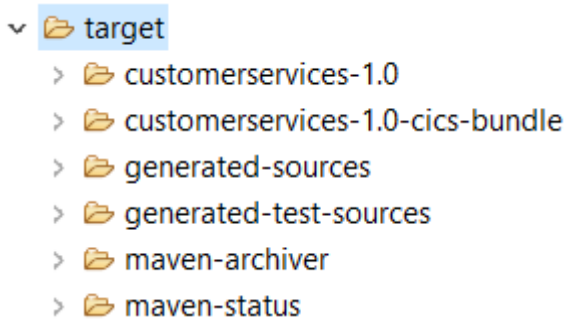


Eclipse lets you specify various JREs.

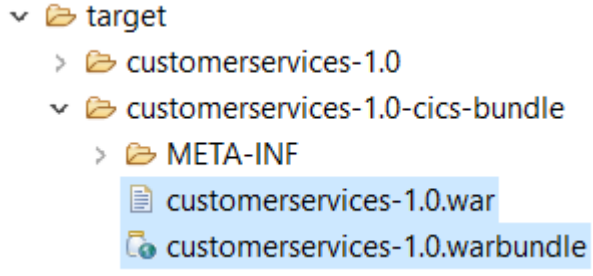


The "target" folder:

Within each project will be a folder called "target". This is populated by the "Maven install" command.

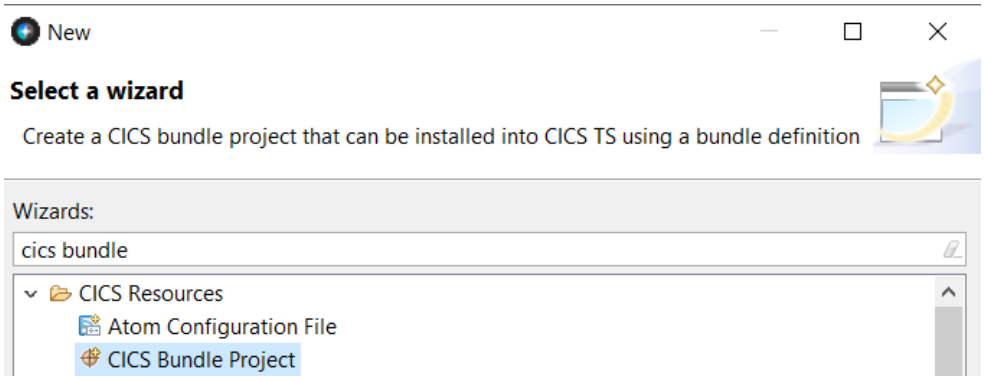


Expand the "*-cics-bundle" folder.

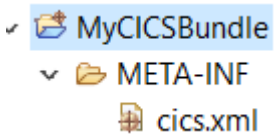


There are two files here that are required, in EACH project.

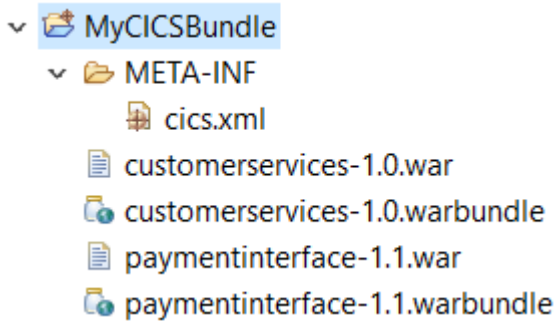
Now create a CICS Bundle project. Do this with File->New->CICS Bundle Project



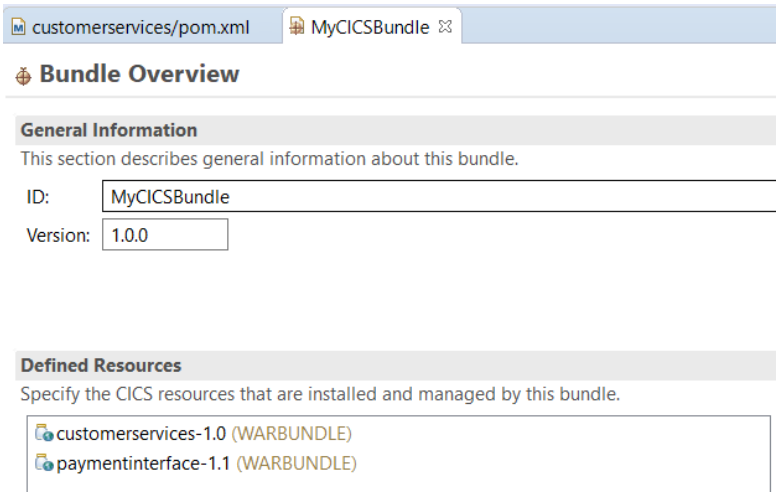
Give the bundle a name, for example “MyCICSBundle”



To begin with this is empty. We need to copy and paste the “war” and “warbundle” files from the cics-bundle folder inside the target folders… into the MyCICSBundle project.

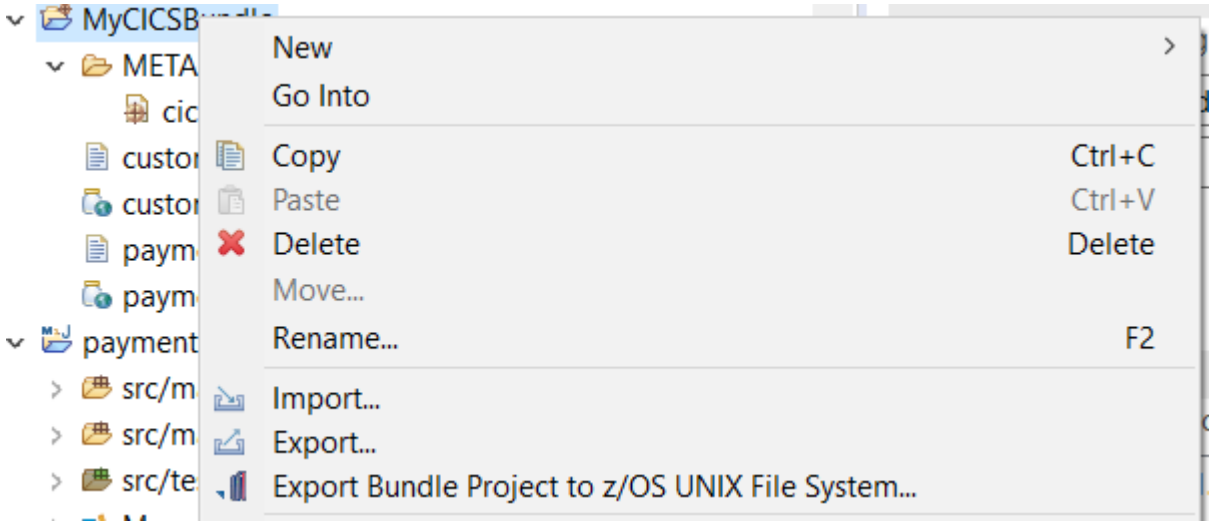


There should now be a CICS bundle project containing our two Spring boot applications.

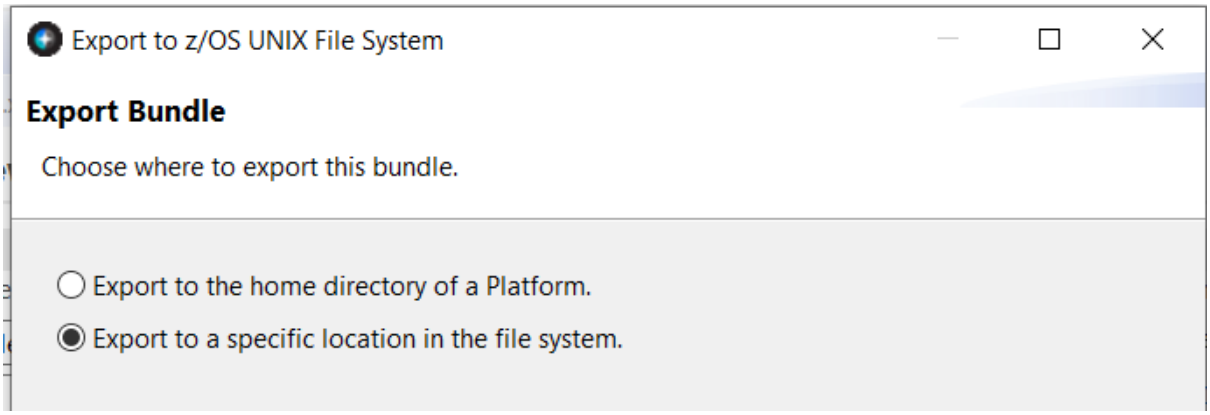

customerservices/pom.xml
MyCICSBundle
Bundle Overview
General Information
This section describes general information about this bundle.
ID:
MyCICSBundle
Version:
1.0.0
Defined Resources
Specify the CICS resources that are installed and managed by this bundle.
customerservices-1.0 (WARBUNDLE)
paymentinterface-1.1 (WARBUNDLE)

Exporting to z/OS:

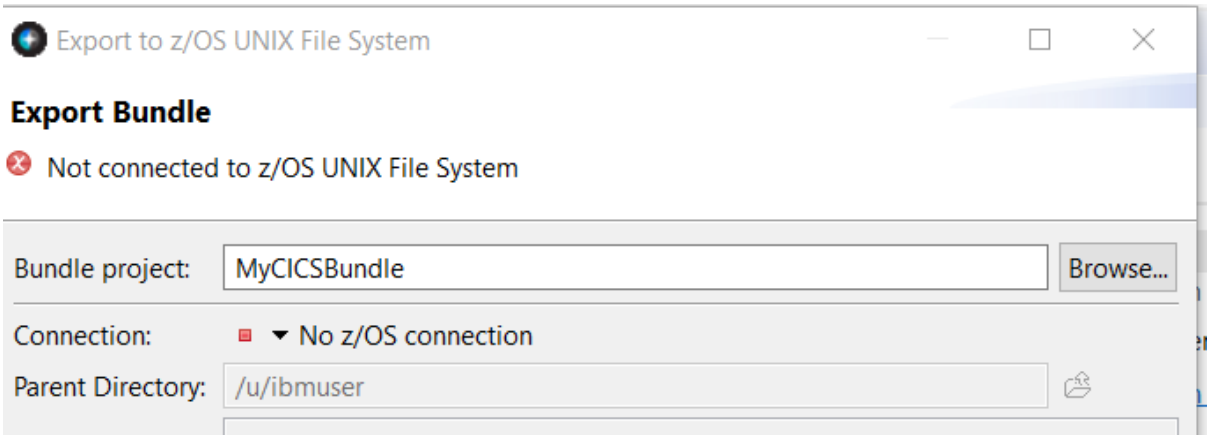
Right click the CICS Bundle project and select Export Bundle Project to z/OS UNIX File System



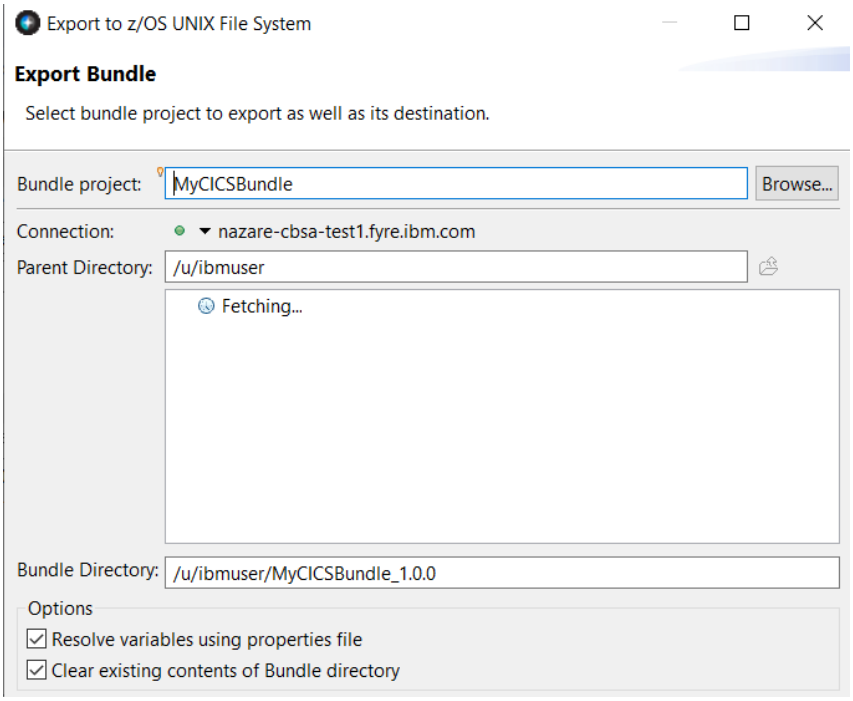
Select “Export to a specific location in the file system”.



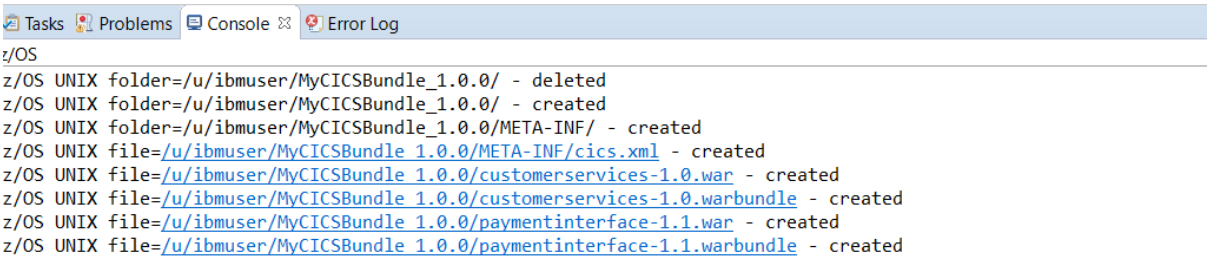
You need a z/OS connection to do this, so select the drop down by connection and fill in the hostname and port of your system.



We have successfully connected and are exporting to/u/ibmuser/MyCICSBundle_1.0.0. We have selected the “Clear existing contents” checkbox in case we want to do this again.



The export is successful, and we have created a folder on the z/OS UNIX Systems Services file system.



Updating the JVM server:

The JVM server may need to be updated to provide more time for the application to start. The JVM profile lives in the JVMPROFILEDIR specified on CICS start-up, in this example it is this one:

/S0W1/var/cics/JVMProfiles/DFHWLP.jvmprofile

At the bottom, add the following timeout overrides.

```
-Dcom.ibm.cics.jvmserver.controller.timeout=900000
-Dcom.ibm.cics.jvmserver.wlp.bundlepart.timeout=900000
```

Then update the server.xml which controls the operation of the WebSphere Liberty Application Server. In this example, it is here, and this is an ASCII file. You need to use 3.17 and the EA option to edit it.

```
/u/cicsuser/CICSTS56/CBSAWLP/wlp/usr/servers/defaultServer/server.xml
```

In the <featureManager> section, add support for Spring Boot by adding this line

```
<feature>springBoot-2.0</feature>
```

Restart the JVM server to allow these new changes to come into effect.

Creating a CICS Bundle Definition:

CICS Bundles can be defined using the CICS Explorer or the CEDA transaction. The important attribute is the “Bundledir” which is where we exported the bundle to earlier.

In this example it is in /u/ibmuser/MyCICSBundle_1.0.0/

```
OBJECT CHARACTERISTICS
 CEDA  View Bundle( MYBUNDLE )
  Bundle          : MYBUNDLE
  Group           : MYBUNDLE
  DEScription     :
  Status          : Enabled            Enabled | Disabled
  BUndledir       : /u/ibmuser/MyCICSBundle_1.0.0/
```

After installing the bundle, wait for it to become enabled, which may take up to 5 minutes.

Use the CEMT INQUIRE BUNDLE command to monitor this.

```
I BUND
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Bun(MYBUNDLE) Dis         Par(00002) Tar(00002)             BEING ENABLED
     Enabledc(00000) Bundlei(MyCICSBundle              )
```

It is now enabled… but you may still have to wait.

```
I BUND
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Bun(MYBUNDLE) Ena         Par(00002) Tar(00002)
     Enabledc(00002) Bundlei(MyCICSBundle              )
```

The WAR files have installed and now we must wait for the Spring Boot applications to start. Check the messages.log file in, in this example these are found in:

```
/u/cicsuser/CICSTS56/CBSAWLP/wlp/usr/servers/defaultServer/logs
```

This is a sign that things are working, and the applications are starting.

Message CWWKZ0001I is issued several minutes later (3 on our test machine) at which point you can use the applications.

Root WebApplicationContext: initialization completed in 180700 ms

Root WebApplicationContext: initialization completed in 180395 ms

Access to the application/interfaces:

To access the Customer Services interface utilise the following URL, note that you will need to amend the URL in this example to reflect your own hostname and port number:

http://nazare-cbsa-test1.fyre.ibm.com:19080/customerservices-1.0/

A user guide for this interface ("CBSA Customer Services Interface User Guide") is located in the /doc folder in the repo.

Access to the Payment interface is via the following URL, note that you will need to amend the URL in this example to reflect your own hostname and port number:

http://nazare-cbsa-test1.fyre.ibm.com:19080/paymentinterface-1.1/

A user guide for the Payment interface ("CBSA Payment Interface User Guide") is located in the /doc folder in the repo.