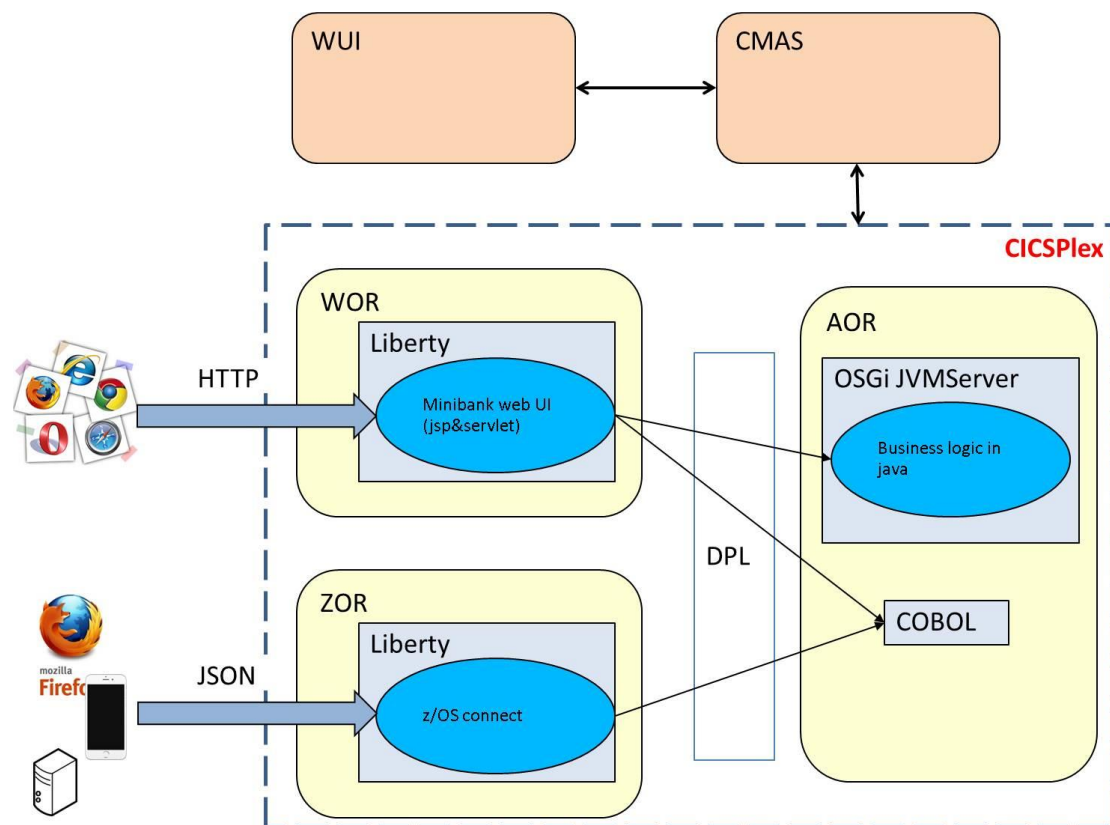

How to setup MiniBank in your environment

Jiang Chen (jiangcbj@cn.ibm.com), IBM CICS
12, Nov, 2015

Requirements



System requirement:

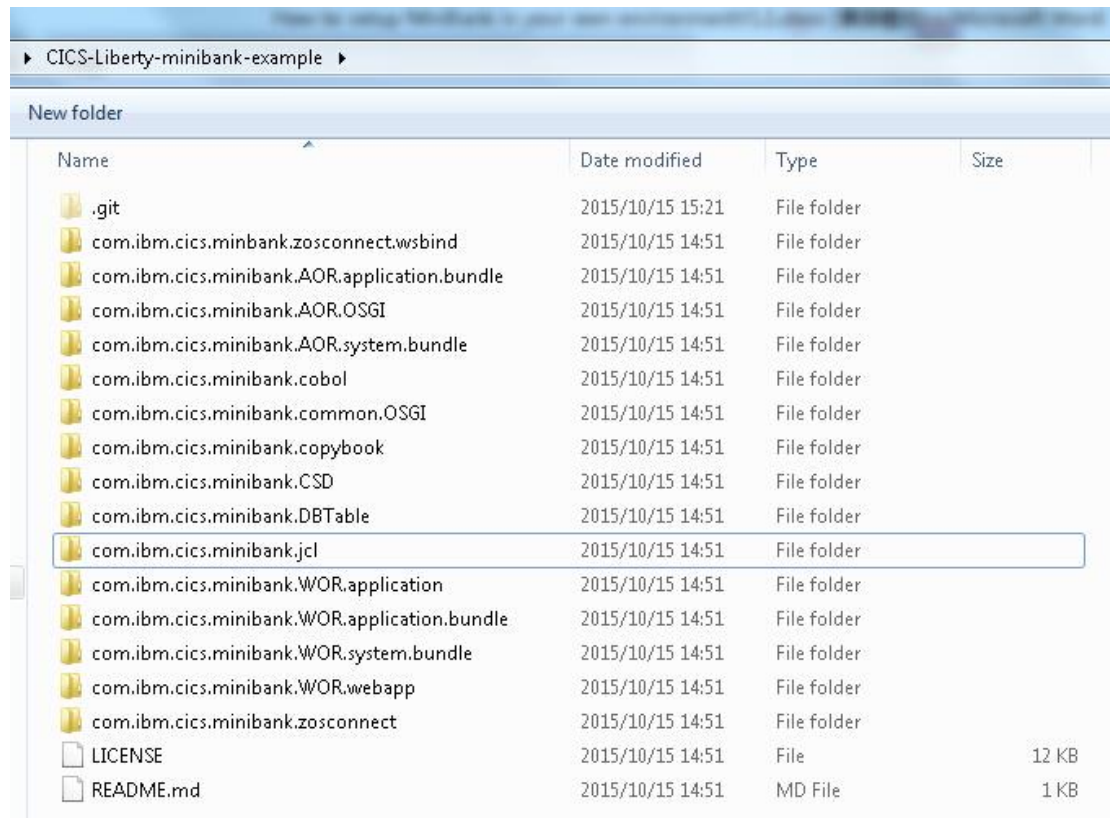
1. We need a CPSM for the Minibank. The requirement for the 5 regions
CMAS: CICS V5.2 with APAR PI25503, or later
WUI: CICS V5.2 or later
AOR: CICS V5.2 or later
WOR: CICS V5.2 or later
ZOR: CICS V5.2 with APAR PI25503, or later. Please set SEC=YES for this region.
Notes: In this document, we use CICS V5.3
2. IBM Java SDK for z/OS V7.0 for SR6 (or later)
3. Eclipse with the CICS Explorer SDK and Websphere Application Server Developer Tool plug-in installed
4. DB2 for z/OS V10 (or later)

Setup Steps – prepare for the environment

Get the Minibank sources from the Github

1. You can get all the codes for MiniBank in this link:

<https://github.com/cicsdev/CICS-Liberty-minibank-example>



Name	Date modified	Type	Size
.git	2015/10/15 15:21	File folder	
com.ibm.cics.minibank.zosconnect.wsbind	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.AOR.application.bundle	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.AOR.OSGI	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.AOR.system.bundle	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.cobol	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.common.OSGI	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.copybook	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.CSD	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.DBTable	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.jcl	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.WOR.application	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.WOR.application.bundle	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.WOR.system.bundle	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.WOR.webapp	2015/10/15 14:51	File folder	
com.ibm.cics.minibank.zosconnect	2015/10/15 14:51	File folder	
LICENSE	2015/10/15 14:51	File	12 KB
README.md	2015/10/15 14:51	MD File	1 KB

2. Below is brief introduction on these projects.

com.ibm.cics.minibank.AOR.application.bundle:

CICS bundle project for AOR. Application resource definitions, includes CICS program definitions, OSGI bundles where Java programs reside in.

com.ibm.cics.minibank.AOR.OSGI :

OSGI bundle project for AOR. Back-end mini bank Java programs.

com.ibm.cics.minibank.AOR.system.bundle :

CICS bundle project for AOR. System resource definitions, Includes CICS transactions, CICS JVMprofile, CICS JVMserver definitions.

com.ibm.cics.minibank.cobol:

The source code for the Transfer transaction

com.ibm.cics.minibank.copybook

The copybook used in Minibank

com.ibm.cics.minibank.csd

The csd for the three regions

com.ibm.cics.minibank.DBTable

The jcl to create the tables used in Minibank

com.ibm.cics.minibank.jcl

the jcl to generate the files used in zosConnect

com.ibm.cics.minibank.common.OSGI:

OSGI bundle project. It is common utility which will be included in both AOR and TOR.

com.ibm.cics.minibank.WOR.application:

OSGI application bundle.

com.ibm.cics.minibank.WOR.application.bundle:

CICS bundle project for WOR(WOR). Application resource definitions, Includes CICS program definitions. Ebabundle which will run in Liberty server.

com.ibm.cics.minibank.WOR.system.bundle

CICS bundle project for WOR(WOR). System resource definitions, Includes CICS transactions, URIMAPS, JVM server and JVMprofile definitions.

com.ibm.cics.minibank.WOR.webapp:

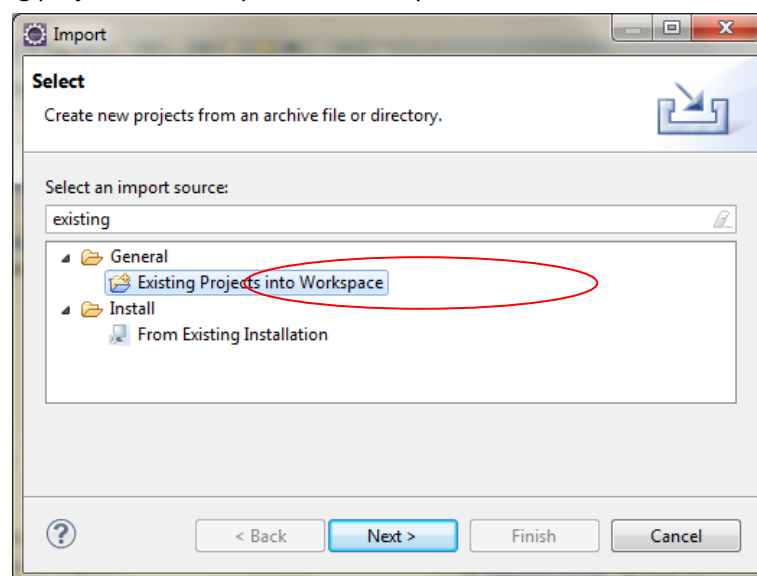
Web enabled OSGI bundle project. JSP, java code which will run in Liberty reside here.

com.ibm.cics.minibank.zosconnect:

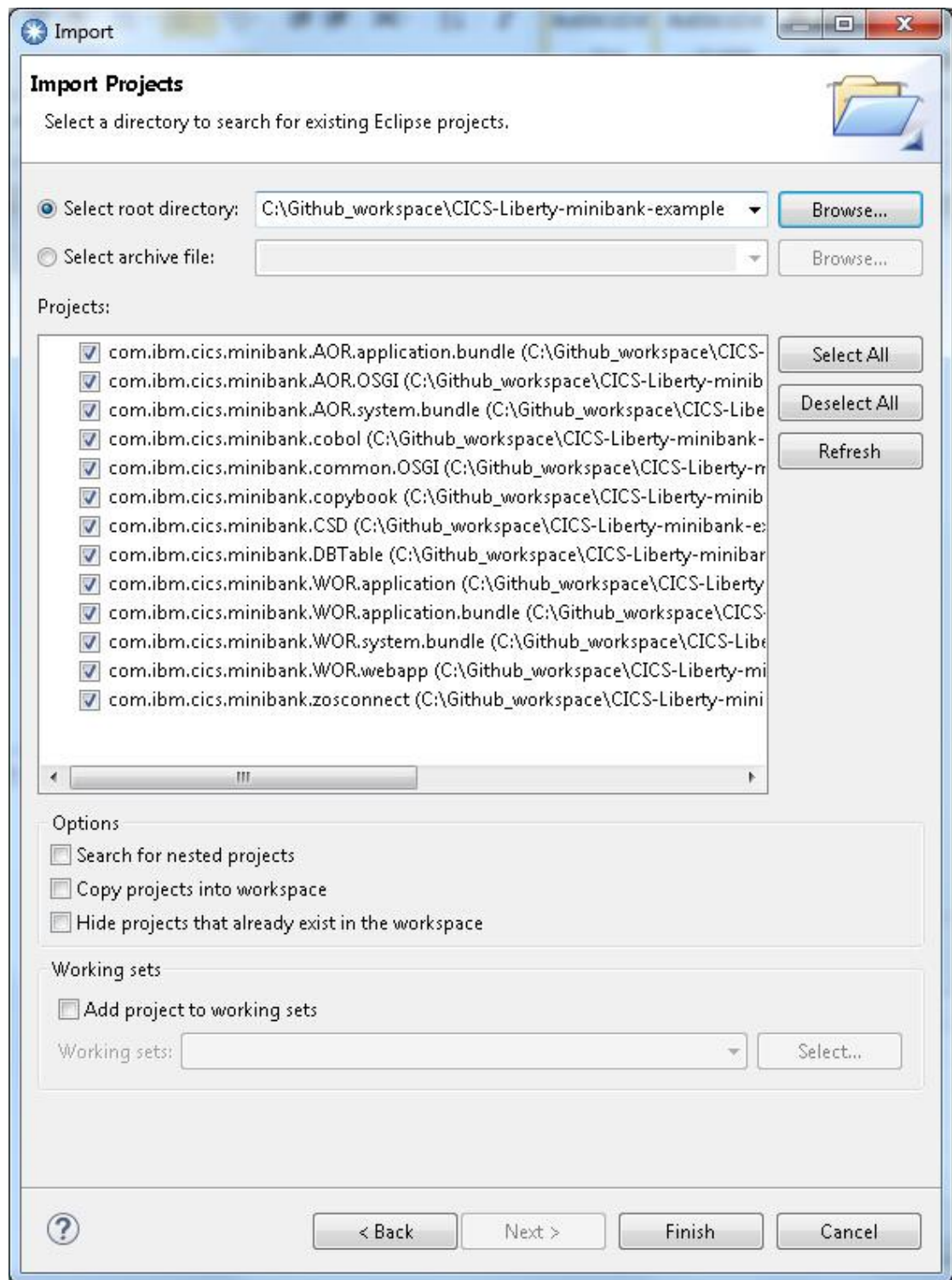
CICS bundle project for zos connect region (ROR). Includes CICS JVMsrever, JVMprofile, Pipeline definitions.

Import the projects to CICS Explorer

1. In Eclipse, Open J2EE Perspective and open "Project Explorer" view by Windows -> Show view -> Project Explorer., and right-click the blank area in "Project Explorer" view and choose "Import" in the pop-up menu.
2. In the pop-up wizard, choose General -> Existing projects into Workplace. You can enter "Existing projects" in the input box to easily locate the item. Click Next.

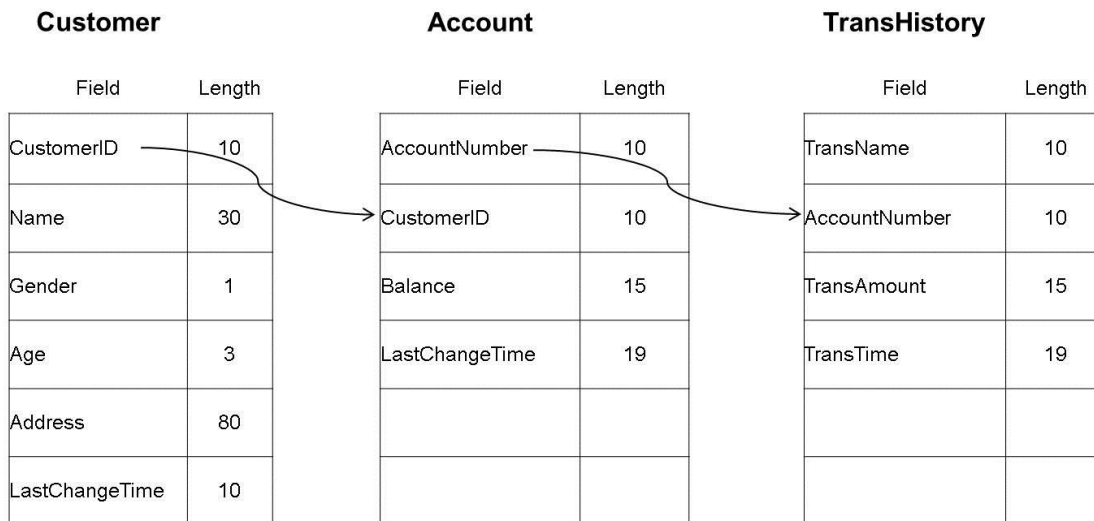


3. Click the **Browse** button on the right to import the **projects** from the minibank resources you get from Github



Setup DB2 Related things

1. Please create the DB2 tables using the JCL in the project **com.ibm.cics.minibank.DBTable** ,
`//JOB LIB DD DSN=SYS2.DB2.V10.SDSNLOAD,DISP=SHR`
`LIB('SYS2.DB2.V10.DSNT.RUNLIB.LOAD') +`
Please change the dataset name to your own environment name.
You can also change the schema name in the JCL. But please remember the schema.



- In CICS Explorer, find the file MinibankAOR.properties in the project com.ibm.cics.minibank.AOR.OSGI by open the src/properties. Change YANFENG to your schema name. Then save it.

```
#####
# DB2 table names
#####
TABLE_CUSTOMER=YANFENG.CUSTOMER
TABLE_ACCOUNT=YANFENG.ACCOUNT
TABLE_TRANHIST=YANFENG.TRANSISTORY
```

- Setup DB2CONN in your AOR region, set the DB2ID of your DB2 system, then install and connect the DB2CONN

Compile the COBOL program

- Please compile the cobol program in the project **com.ibm.cics.minibank.cobol**. Please remember the load dataset name of the cobol program.
- Find the MINIBANK.library file in project com.ibm.cics.minibank.AOR.application.bundle. Change the value of DSNAMES01 to the cobol program's load dataset name.

Setup JVMServer

- In CICS Explorer, find the file DFHOSGI.jvmprofile in the project com.ibm.cics.minibank.AOR.system.bundle.

Change the values of the below keys to your own values.

```
WORK_DIR=/u/cntest7/liberty/workdir
JAVA_HOME=/java/J7.0_64
LIBPATH_SUFFIX=/usr/lpp/db2v10/jdbc/lib/
OSGI_BUNDLES=/usr/lpp/db2v10/jdbc/classes/db2jcc4.jar,\
```

/usr/lpp/db2v10/jdbc/classes/db2jcc_license_cisuz.jar

2. In CICS Explorer, find the file DFHWLP.jvmprofile in the project `com.ibm.cics.minibank.WOR.system.bundle`.

Change the values of the below keys to your own values.

WORK_DIR=/u/cntest7/liberty/workdir

JAVA_HOME=/java/J7.0_64

LIBPATH_SUFFIX=/usr/lpp/db2v10/jdbc/lib/

-Dcom.ibm.cics.jvmserver.wlp.server.http.port=30095

-Dcom.ibm.cics.jvmserver.wlp.server.https.port=30096

-Dcom.ibm.cics.jvmserver.wlp.server.host=winmvs2d.hursley.ibm.com

3. In CICS Explorer, find the file DFHWLP.jvmprofile in the project `com.ibm.cics.minibank.zosconnect`.

Change the values of the below keys to your own values.

WORK_DIR=/u/cntest7/liberty/workdir

JAVA_HOME=/java/J7.0_64

-Dcom.ibm.cics.jvmserver.wlp.server.http.port=30095

-Dcom.ibm.cics.jvmserver.wlp.server.https.port=30096

-Dcom.ibm.cics.jvmserver.wlp.server.host=winmvs2d.hursley.ibm.com

Generate the WSBIND file

1. Upload the file TRANSFER in the project **com.ibm.cics.minibank.copybook** to a dataset, the JCL will use the copybook to generate wsbind, schemas used in zosConnect lab. You can also copy the file content to a dataset member
2. Upload the file LS2JS.jcl in the project **com.ibm.cics.minibank.jcl** to a dataset, or copy the file content to a dataset member.

Update the red places. Please create the uss directory used in the JCL.

/u/amilybj/zosConnect/ls2js/log/

/u/amilybj/zosConnect/ls2js/wsbind/

/u/amilybj/zosConnect/ls2js/schema/

//L2JB02 JOB (MYSYS,AUSER),MSGCLASS=H,

// CLASS=A,NOTIFY=&SYSUID,REGION=0M

//LS2JS JCLLIB ORDER=(ANTZ.CICS.TS.DEV.INTEGRAT.SDFHINST)

//WSBIND EXEC DFHLS2JS,

// JAVADIR='/java/java71_bit64_sr1/J7.1_64',

// USSDIR='devint',

// PATHPREF='',

// TMPDIR='/tmp',

// TMPFILE='I'

*//INPUT.SYSUT1 DD **

LOGFILE=/u/amilybj/zosConnect/ls2js/log/ls2js.log

PDSLIB=//AMILYBJ.CICS.MINIBANK

REQMEM=TRANSFER

RESPMEM=TRANSFER

```
LANG=COBOL
WSBIND=/u/amilybj/zosConnect/ls2js/wsbind/trasferJSON.wsbind
JSON-SCHEMA-REQUEST=/u/amilybj/zosConnect/ls2js/schema/transfQ.json
JSON-SCHEMA-RESPONSE=/u/amilybj/zosConnect/ls2js/schema/transfP.json
MAPPING-LEVEL=3.0
PGMNAME=TRANSFER
PGMINT=COMMAREA
URI=/minibankAPI/transfer
DATETIME=PACKED15
SYNCONRETURN=YES
CCSID=935
/*
```

3. Submit the JCL and get MACC=0
4. In the project **com.ibm.cics.minibank.zosconnect**, update the file ZOSCONN.pipeline
wsdir="/u/yanfeng/zosConnect/ls2js/wsbind". Change the ussdir to the uss dir
you used in the JCL.

Use CICS Explorer to deploy Minibank

Export the projects to USS

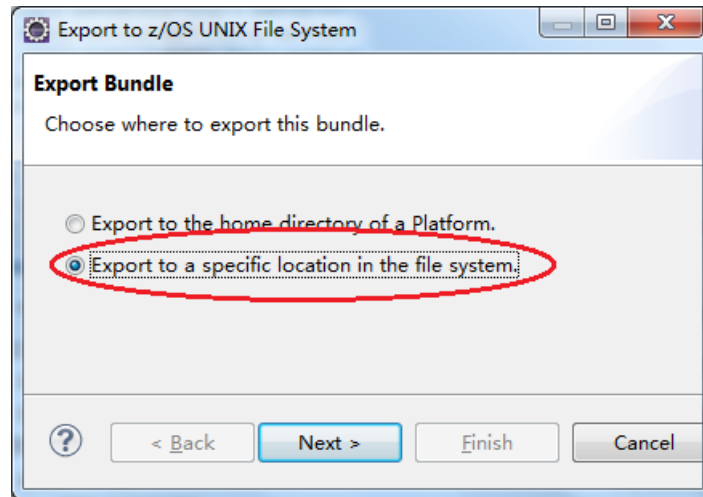
Before this step, make sure you have configured FTP connection to MVS and configured CMCI connection to MVS. If not, reference Appendix B.

Make sure you have setup the target platform, reference to Appendix A.

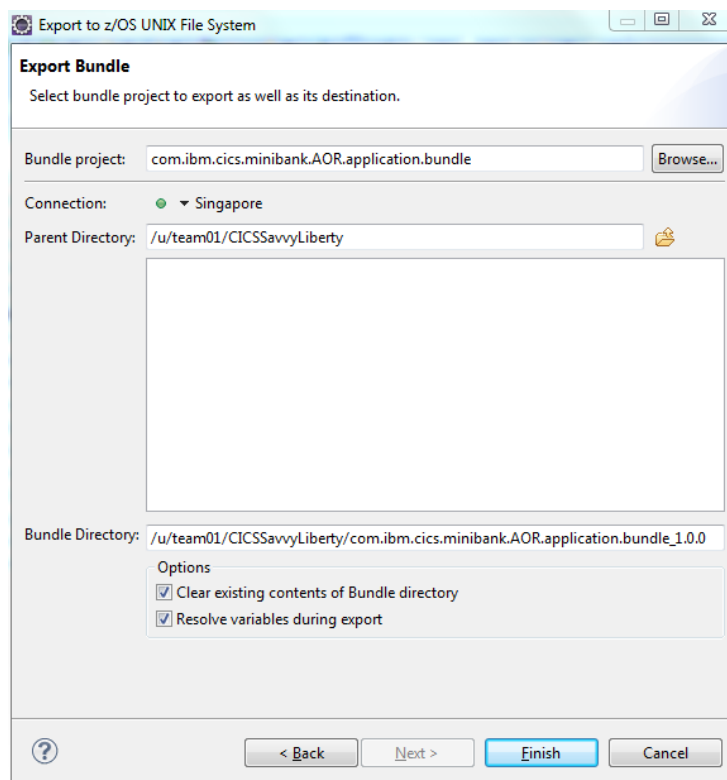
A CICS bundle is a collection of resources, artifacts, references, and a manifest file that you can deploy into a CICS region to represent all or part of an application. In this step, you need to upload the below four CICS bundle projects to USS.

```
com.ibm.cics.minibank.AOR.application.bundle;  
com.ibm.cics.minibank.AOR.system.bundle ;  
com.ibm.cics.minibank.WOR.application.bundle;  
com.ibm.cics.minibank.WOR.system.bundle;
```

1. Right click the **com.ibm.cics.minibank.AOR.application.bundle**, choose the menu "Export Bundle Project to z/OS UNIX File System..."
2. Select the "Export to specific location in the file system"



Enter your USS folder (**/u/ ****) as Parent Directory to export the bundle. It does not matter the directory is not there because the tool will create one for you. if you already have old content in the folder, for example the old version of the bundle, you can enable the check box “Clear existing contents of Bundle directory” to clean the folder before exporting. Click **Finish** to export the **com.ibm.cics.minibank.AOR.application.bundle** bundle.



3. After you press finish, you will see the similar output like:(*note: please record down the bundle directory before you press Finish, it will be used later when you define the bundle resource*)

```
z/OS
z/OS UNIX folder=/u/team01/CICSSavvyLiberty/ - created
z/OS UNIX folder=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/META-INF/ - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/RETACT.program - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/RETUSER.program - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/DEPOSIT.program - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/META-INF/cics.xml - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/MINIBANK.library - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/QRVACT.program - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/QRVUSER.program - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/TRANSFER.program - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/WITHDRAW.program - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/com.ibm.cics.minibank.AOR.OSGI.osgibundle - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/com.ibm.cics.minibank.common.OSGI.osgibundle - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/com.ibm.cics.minibank.AOR.OSGI.1.0.0.jar - created
z/OS UNIX file=/u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.AOR.application.bundle.1.0.0/com.ibm.cics.minibank.common.OSGI.1.0.0.jar - created
```

4. Repeat step 1-3 to upload below left 3 bundle projects.

com.ibm.cics.minibank.AOR.system.bundle ;
com.ibm.cics.minibank.WOR.application.bundle;
com.ibm.cics.minibank.WOR.system.bundle;

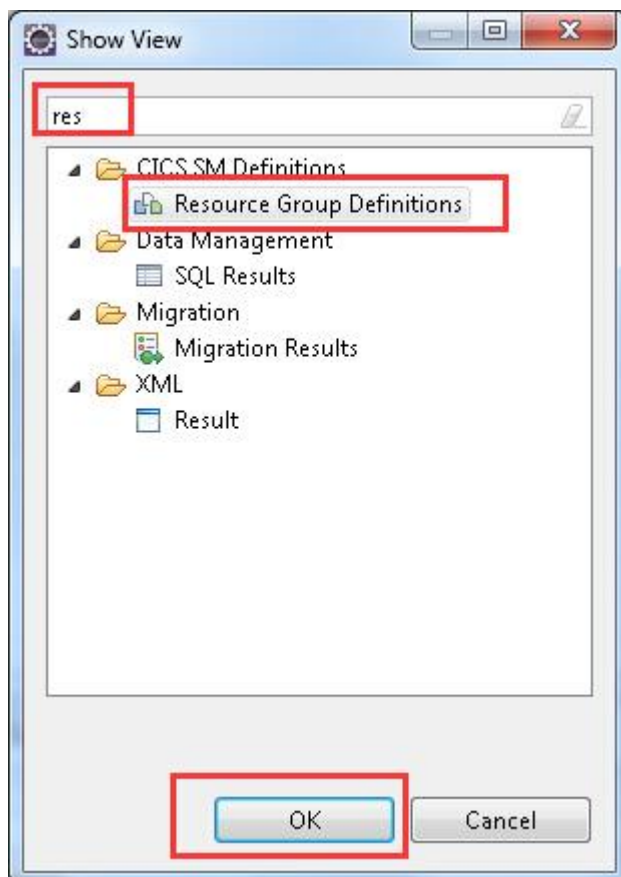
Define system connections

WOR <-> AOR used for Minibank

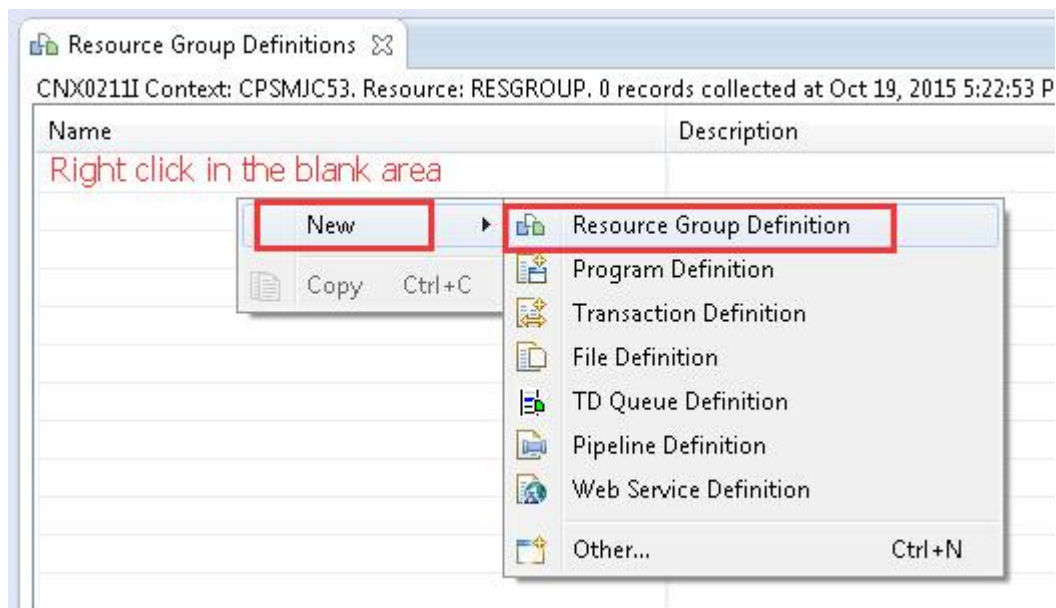
ZOR <-> AOR used for zosConnect

Create resource group definition

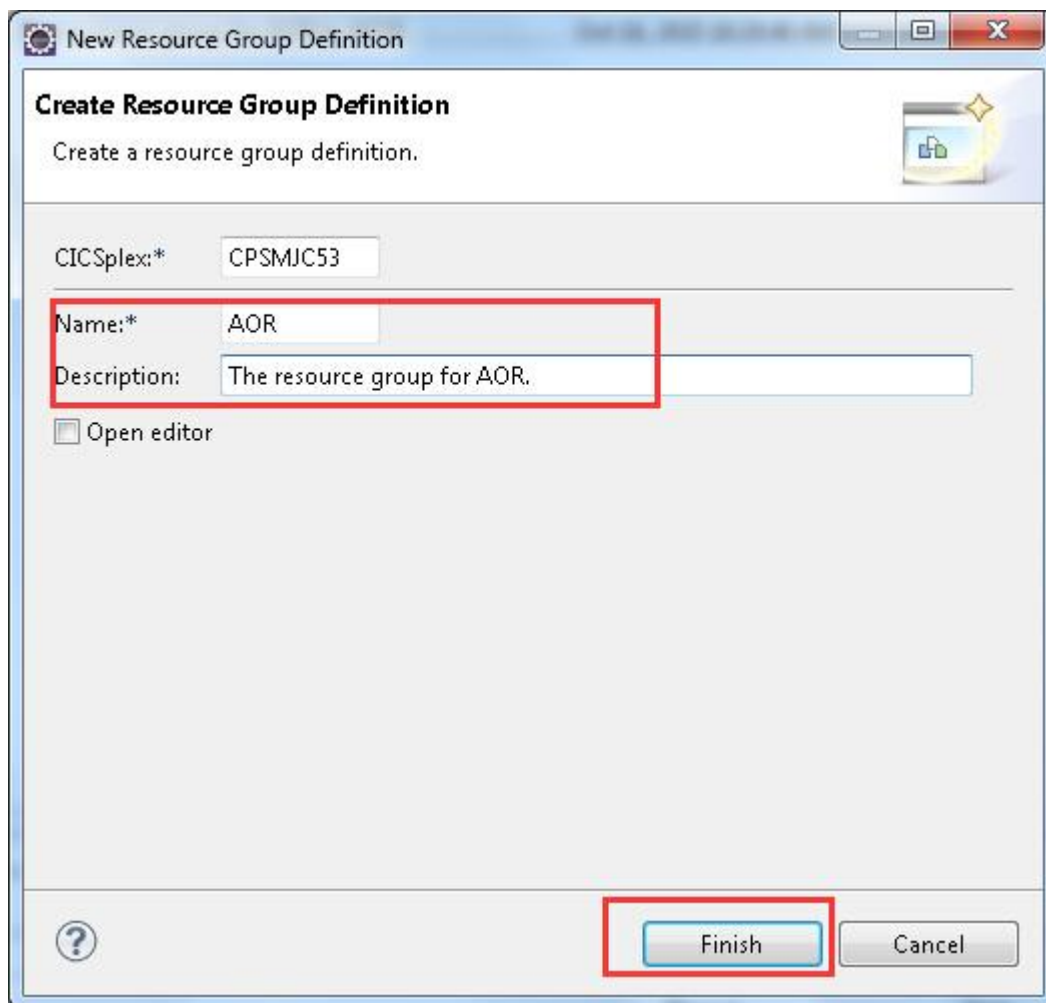
1. In the CICS SM perspective, click Window->Show View -> Other., input res, then choose resource group definition, click OK.



2. In the resource group view, right click the blank area, then click new.



3. Input the group name and description



4. Repeat the 1~3 steps to add other three group, WOR, ZOR. After these steps, you can see 3 groups.

Create TCIPService

1. In the CICS SM perspective, click Definitions-> TCP/IP Service Definitions
2. Right click the blank area in the I TCP/IP Service Definitions, then click new
3. Input the group, name, description port number and choose IPIC in the protocol, then click finish.

Create TCP/IP Service Definition
Create a TCP/IP service definition.

CICSplex:* CPSMJC53

☐ Region (CSD):

Resource/CSD Group: AOR

Name:* AOR

Description: The TCP/IP Service of AOR.

Port Number:* 30031 *The port of AOR*

Protocol: IPIC URM:

☐ Open editor

Finish **Cancel**

4. Repeat the 2~3 steps to add other three connections, please note:
ZOR: GROUP(ZOR), Port(The port of the ZOR)
WOR: GROUP(WOR), Port(The port of the WOR)

Create IPIC Connection

1. In the CICS SM perspective, click Definitions-> IPIC Connection Definitions
2. Right click the blank area in the IPIC Connection Definitions, then click new
3. Input the attributes as the picture showed.

New IPIC Connection Definition

Create IPIC Connection Definition
Create a IPIC connection definition.

CICSplex:* CPSMJC53

☐ Region (CSD):

Resource/CSD Group: AOR

Name:* ATOW

Description: The IPIC Connection from AOR to TOR.

TCP/IP Service:* AOR

Outgoing

☒ This connection can send data

Host:* winmvs2c.hursley.ibm.com *Change to your host*

Port:* 30032 *The WOR's TCP/IP Service port*

Application ID:* IYK3Z0A4 *The WOR's application id*

☐ Open editor

Finish **Cancel**

4. Double the ATOW to open the editor, change the AUTOCONNECT to YES

***IPIC Connection Definition (ATOR)**

Attributes

CPSMJC53 ▶ ATOR ▼

type here to filter on Name and CICS Name

Name	CICS Name	Value
Basic		
Application ID	APPLID	IYK3Z0A5
> Autoconnect	AUTOCONNECT	YES
Certificate	CERTIFICATE	

5. Repeat the 2~4 steps to add other three sessions: WTOA, ATOR, RTOA. Don't forget to change the AUTOCONNECT to YES.

New IPIC Connection Definition

Create IPIC Connection Definition
Create a IPIC connection definition.

CICSplex:* CPSMJC53

☐ Region (CSD):

Resource/CSD Group: AOR

Name:* ATOR

Description:

TCP/IP Service:* AOR

Outgoing

☒ This connection can send data

Host:* winmvs2c.hursley.ibm.com

Port:* 30034

Application ID:* IYK3Z0A5

☐ Open editor

Finish **Cancel**

ZOR's port and application id

New IPIC Connection Definition

Create IPIC Connection Definition
Create a IPIC connection definition.

CICSplex:* CPSMJC53

☐ Region (CSD):

Resource/CSD Group: WOR

Name:* WTOA

Description:

TCP/IP Service:* WOR

Outgoing

☒ This connection can send data

Host:* winmvs2c.hursley.ibm.com

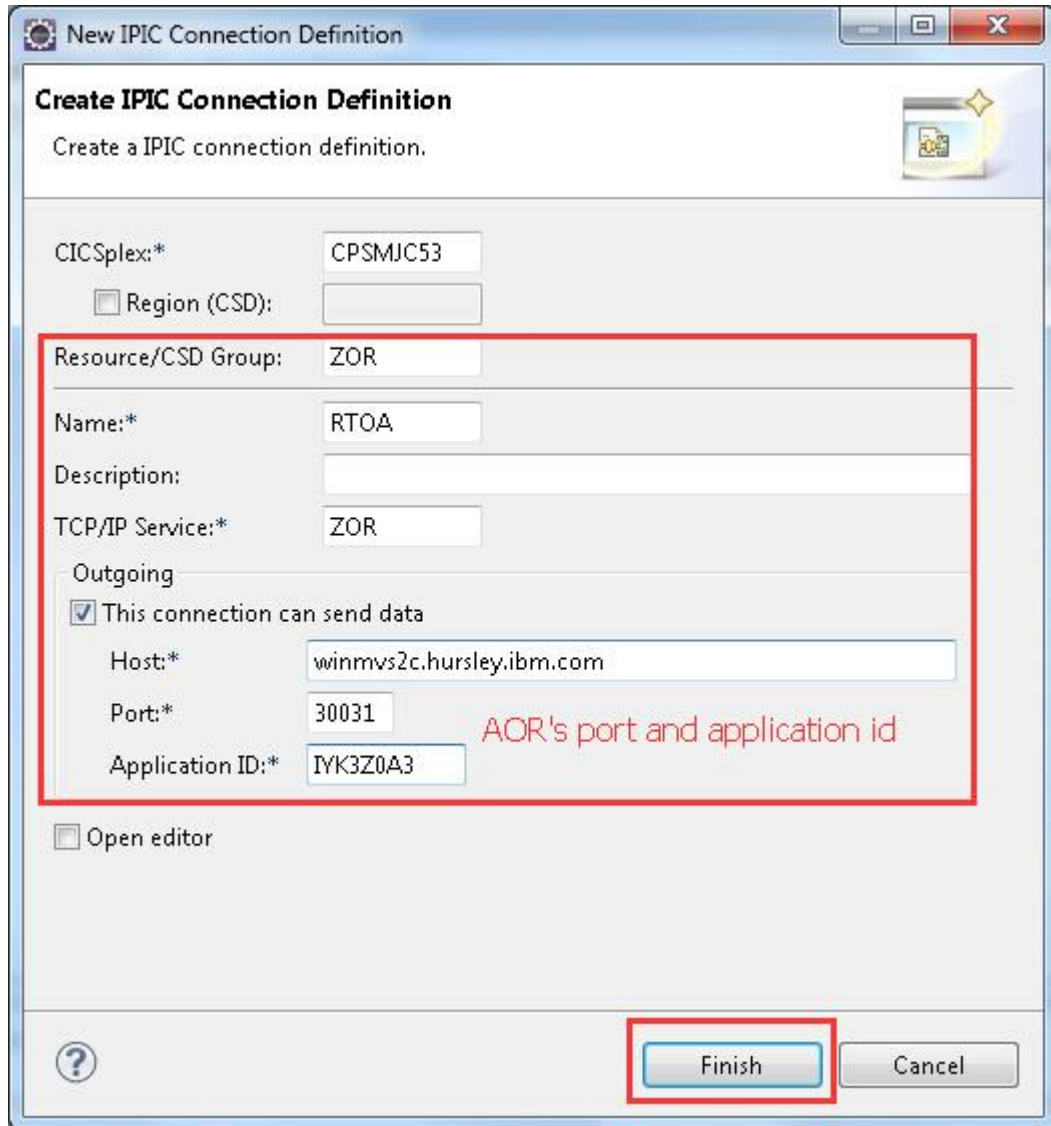
Port:* 30031

Application ID:* IYK3Z0A3

☐ Open editor

Finish Cancel

AOR's port and application id



The image shows a Windows-style dialog box titled "New IPIC Connection Definition". The main heading is "Create IPIC Connection Definition" with a sub-instruction "Create a IPIC connection definition." and a small icon of a folder with a star. The dialog contains several input fields and checkboxes. A red rectangular box highlights a section containing the following fields: "Resource/CSD Group:" with the value "ZOR", "Name:*" with the value "RTOA", "Description:" (empty), "TCP/IP Service:*" with the value "ZOR", a section header "Outgoing", a checked checkbox "This connection can send data", "Host:*" with the value "winmvs2c.hursley.ibm.com", "Port:*" with the value "30031", and "Application ID:*" with the value "IYK3Z0A3". A red handwritten note "AOR's port and application id" is next to the port and application ID fields. Below the highlighted section is an unchecked checkbox "Open editor". At the bottom right, there are two buttons: "Finish" (highlighted with a red box) and "Cancel". A help icon (?) is at the bottom left.

New IPIC Connection Definition

Create a IPIC connection definition.

CICSplex:* CPSMJC53

☐ Region (CSD):

Resource/CSD Group: ZOR

Name:* RTOA

Description:

TCP/IP Service:* ZOR

Outgoing

☒ This connection can send data

Host:* winmvs2c.hursley.ibm.com

Port:* 30031

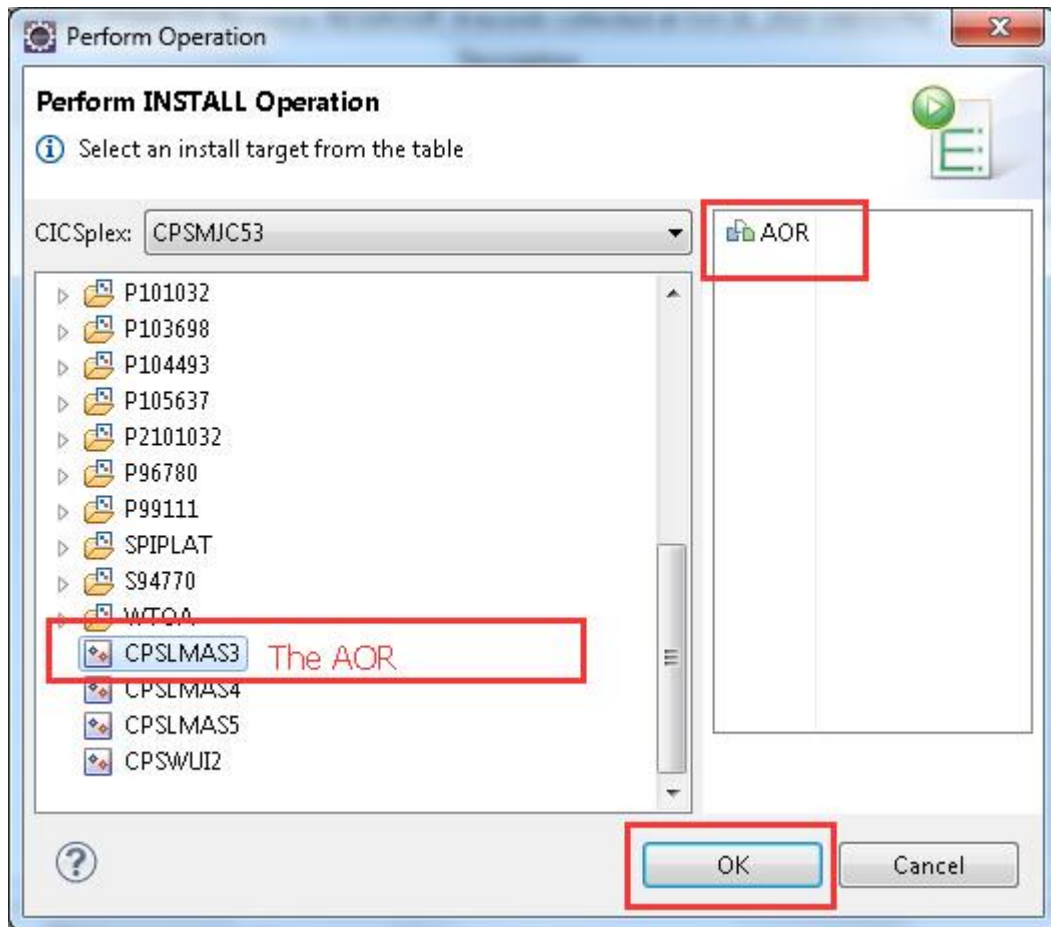
Application ID:* IYK3Z0A3

☐ Open editor

Finish Cancel

Install the resource group

1. In the Resource Group Definitions View, right the AOR, then click the Install. Select the AOR region, then click OK.



2. Install WOR and ZOR resource group to the right region.
3. Click Operations -> IPIC Connection. If you can't see data, please click the CPSM name in the left. You can also click the refresh button to get the latest data.

Region	Name	Port	TCP/IP Service	Appl ID	Host	Connection Status	Service Status
CPSLMAS3	ATOR	30034	AOR	IVK3Z0A5	winmvs2c.hursle...	ACQUIRED	INSERVICE
CPSLMAS3	ATOW	30032	AOR	IVK3Z0A4	winmvs2c.hursle...	ACQUIRED	INSERVICE
CPSLMAS4	WTOA	30031	WOR	IVK3Z0A3	winmvs2c.hursle...	ACQUIRED	INSERVICE
CPSLMAS5	RTOA	30031	ZOR	IVK3Z0A3	winmvs2c.hursle...	ACQUIRED	INSERVICE

Define resources for WOR and AOR

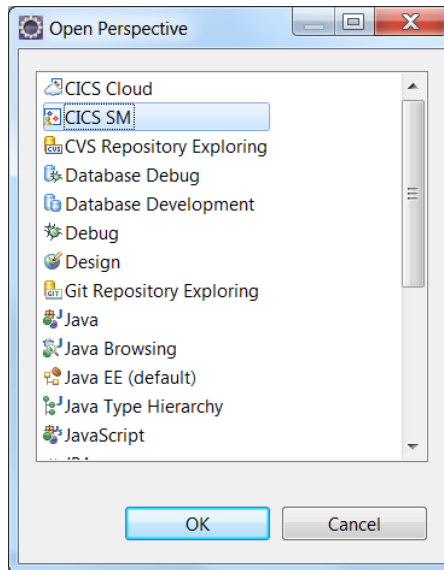
Before define resources, you need to set up CMCI connection to your CPSM environment. If not, please refer to Appendix B.

All bundle artifacts are at USS now. In this step, you need to define and install CICS bundle resource by CICS explorer.

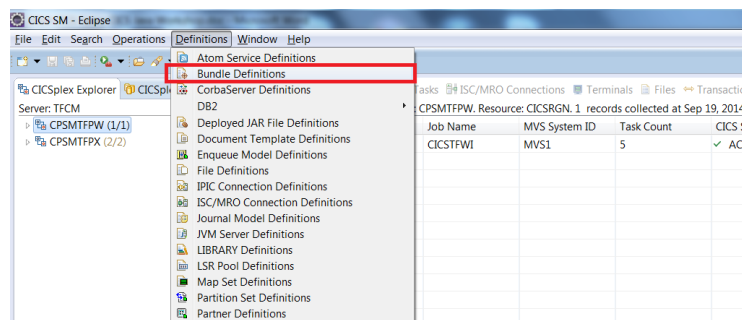
- For WOR:
 - BUNDLE: WAPPBUND – for WORapplication bundle
 - BUNDLE: WSYSBUND – for WORsystem bundle
- For AOR:

- BUNDLE: AAPPBUND – for AOR application bundle
- BUNDLE: ASYSBUND – for AOR system bundle

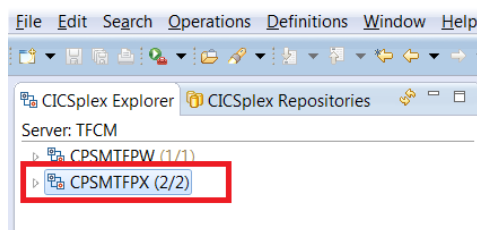
1. Open perspective and choose **CICS SM**.



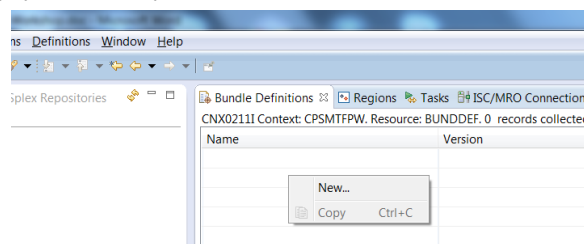
2. Go to Definitions -> Bundle Definitions



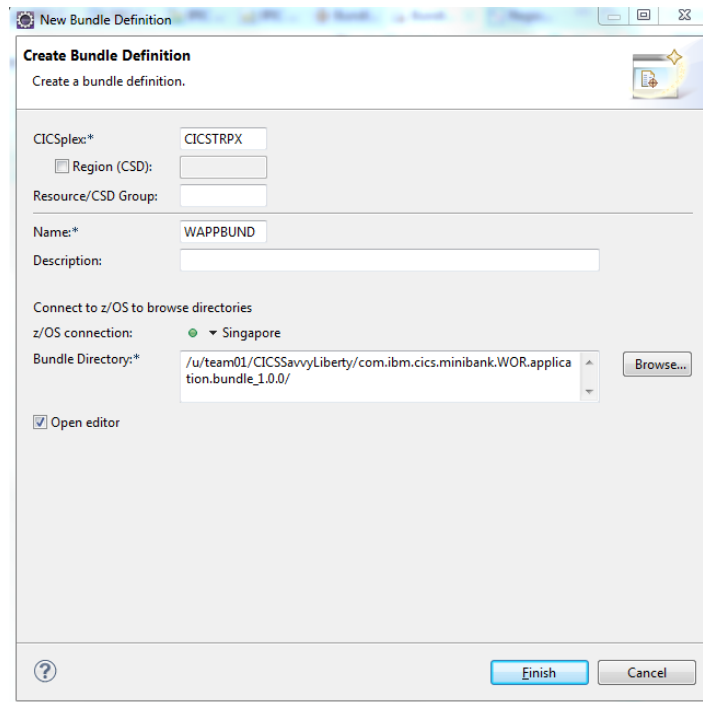
3. Click **CPSMxxPX** in CICSplex Explorer, it will be the CONTEXT for the resource definitions.



4. Right-click the empty area, and press New.



5. Input Name and Bundle Directory for the definition. (*note: Bundle Directory is the directory where you uploaded your Bundle project in USS*)



Create Bundle Definition
Create a bundle definition.

CICSplex*: CICSTRPX

☐ Region (CSD):

Resource/CSD Group:

Name*: WAPPBUND

Description:

Connect to z/OS to browse directories

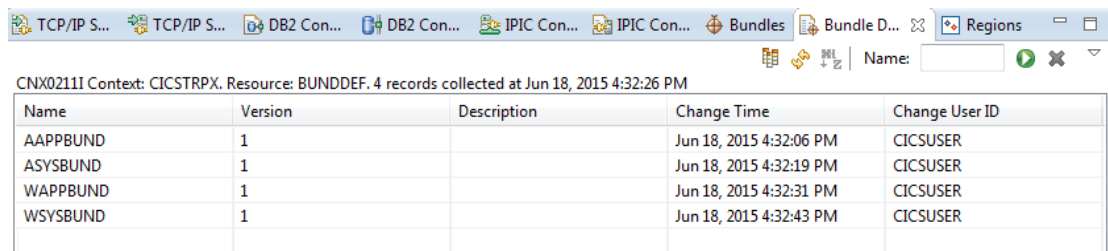
z/OS connection: Singapore

Bundle Directory*: /u/team01/CICSSavvyLiberty/com.ibm.cics.minibank.WOR.application.bundle_1.0.0/ Browse...

☒ Open editor

Finish Cancel

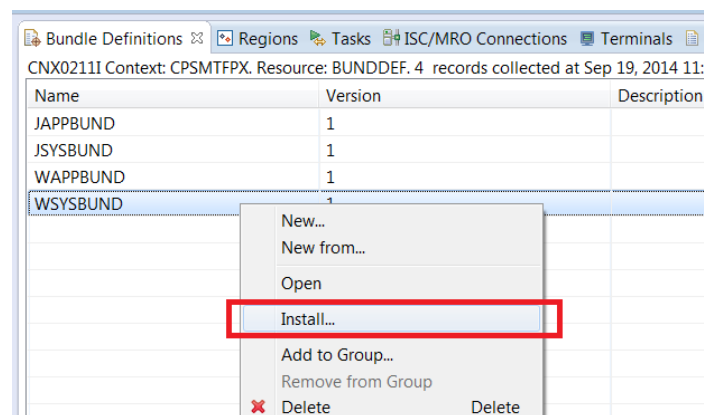
6. Repeat step 4-5 to recreate bundle definitions for other three bundles, and finally you will have four bundle definitions.



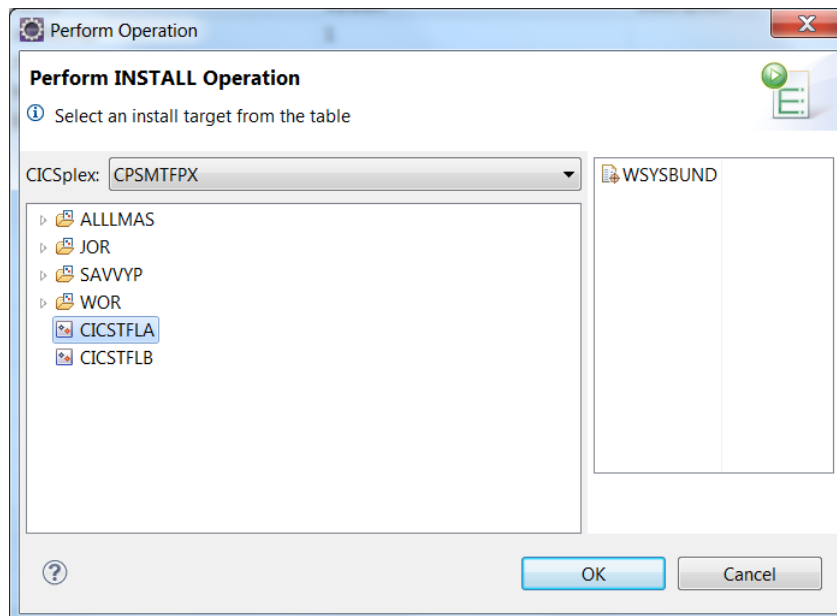
CNX0211I Context: CICSTRPX. Resource: BUNDDEF. 4 records collected at Jun 18, 2015 4:32:26 PM

Name	Version	Description	Change Time	Change User ID
AAPPBUND	1		Jun 18, 2015 4:32:06 PM	CICSUSER
ASYSBUND	1		Jun 18, 2015 4:32:19 PM	CICSUSER
WAPPBUND	1		Jun 18, 2015 4:32:31 PM	CICSUSER
WSYSBUND	1		Jun 18, 2015 4:32:43 PM	CICSUSER

7. Install BUNDLE definitions WSYSBUND to WOR. Firstly, right-click on WSYSBUND, and press Install.



8. Select the target region, WOR REGION



9. Repeat step 7-8 to install below bundles **in sequence**:

- WAPPBUND to WOR REGION.
- ASYSBUND to AOR REGION.
- AAPPBUND to AOR REGION.

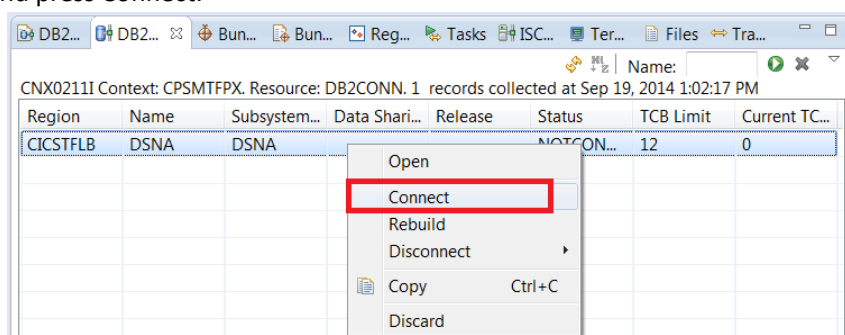
Finally, you have four bundle definitions installed, two for WOR REGION, and two for AOR REGION.

10. Click Operations -> Bundle to review all the bundles installed on the system. You will be able to see:

Bundles										
CNX0211I Context: CPSMTFPX. Resource: BUNDLE. 12 records collected at Sep 19, 2014 12:06:10 PM										
Region	Name	Bundle ID	Major Version	Minor Version	Micro Version	Status	Availability	Install Time	Enat	
CICSTFLA	WSYSBUND	CICSSavvyWeb1...	1	0	0	✓ ENABLED	NONE	Sep 19, 2014 11:...	1	
CICSTFLA	WAPPBUND	CICSSavvyWeb1...	1	0	0	✓ ENABLED	NONE	Sep 19, 2014 11:...	8	
CICSTFLB	JYSBUND	CICSSavvy15Bu...	1	0	0	✓ ENABLED	NONE	Sep 19, 2014 11:...	1	
CICSTFLB	JAPPBUND	CICSSavvy15Bu...	1	0	0	✓ ENABLED	NONE	Sep 19, 2014 11:...	8	

Following resources were already setup, please check if they are installed correctly.

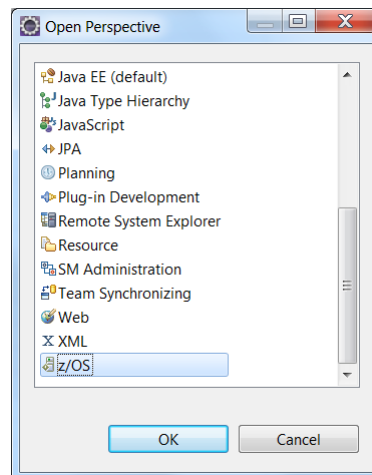
11. Go to Operations -> DB2 -> DB2 Connections, and check the DB2 definition status. Right-click on it and press Connect.



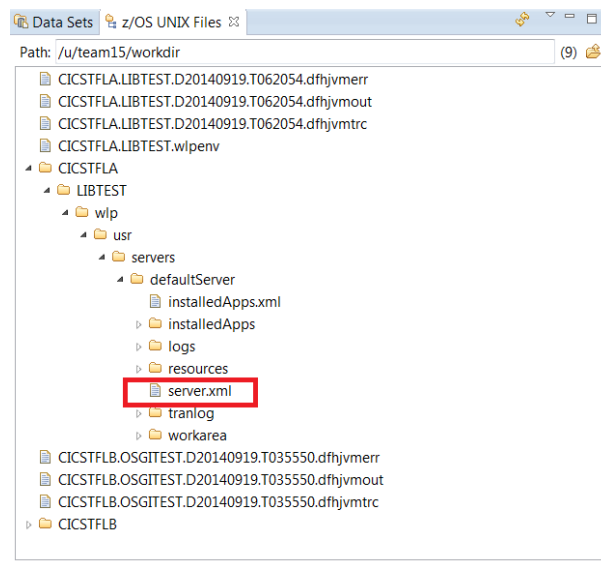
Update configurations to support JDBC Type 4:

We need to update one file to support security and JDBC type 4 in Liberty.

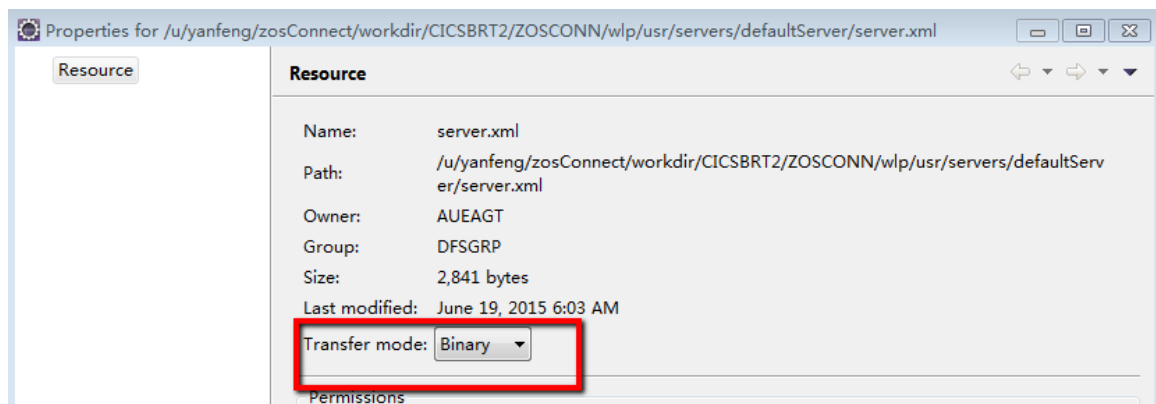
1. Open Perspective z/OS



2. In z/OS UNIX Files view,



3. Set server.xml property to **binary** for FTP save purpose, binary transfer won't corrupt the data format. Right click on server.xml, then click on Properties, change transfer mode to Binary. Save "OK".



4. You can copy and paste the following parts to your server.xml

- Add new feature to feature manager

```
<feature>jdbc-4.0</feature>
```

- Add the following data source and library to support JDBC type 4
Change the databasename, serverName, portNumber, user and password.
Change the db2 library directory.

```
<dataSource jndiName="jdbc/CICSType4DataSource">  
  <jdbcDriver libraryRef="db2jdbc4Lib"/>  
  <properties.db2.jcc databaseName="TSTDB202" driverType="4"  
    password="{xor}NW4+LDduLjY=" portNumber="448" serverName="192.168.42.79"  
    user="YANFENG"/>  
</dataSource>
```

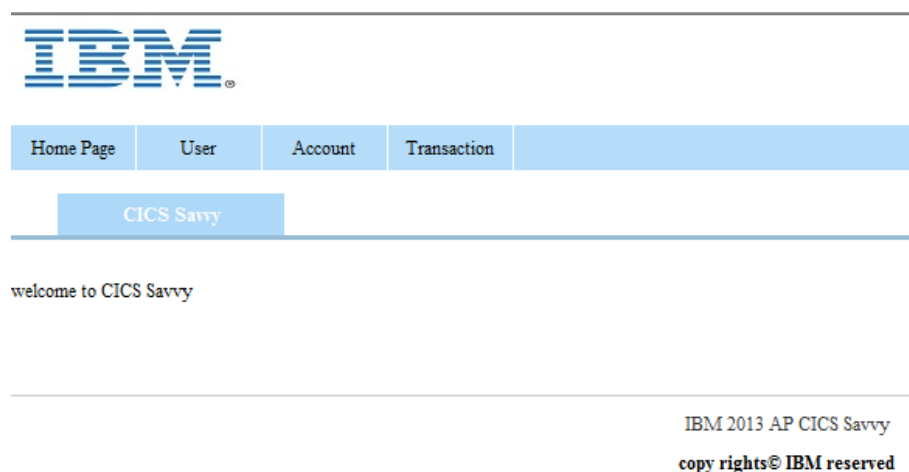
```
<library id="db2jdbc4Lib">  
  <fileset dir="/usr/lpp/db2/db2a10/jdbc/classes" includes="db2jcc4.jar  
    db2jcc_license_cisuz.jar"/>  
  <fileset dir="/usr/lpp/db2/db2a10/jdbc/lib"/>  
</library>
```

Run Mini Bank programs:

Mini bank is Web based applications as we deployed in above steps. We will expose one of its programs as JSON web services in zos connect in late part of this workshop. Now you can play with mini bank. Have a fun!

Visit the Liberty program with URL

Go to <http://your> host:<your http port for the WOR liberty >/Minibank/toMenuPage.action. It will get page as below.



Click the User, Account, Transaction, you will get drop-down menu. From the menu you can run the operations.

User	Create User:	Create user with basic info
	Query User:	Query the user basic info and all the accounts of this user
Account	Create Account:	Create account with account number and initial balance. Associate the account to a specific user. The account number needs to be a 10 digits number.
	Query Account:	Query the account info and balance, and list all the historical transaction records of this account. Transaction type is deposit, withdraw or transfer
Transactions	Deposit:	Deposit to the specified account
	Withdraw:	Withdraw from the specified account
	Transfer:	Money transfer from source account to target account

Set up CICS zosConnect support in ROR

Brief introduction:

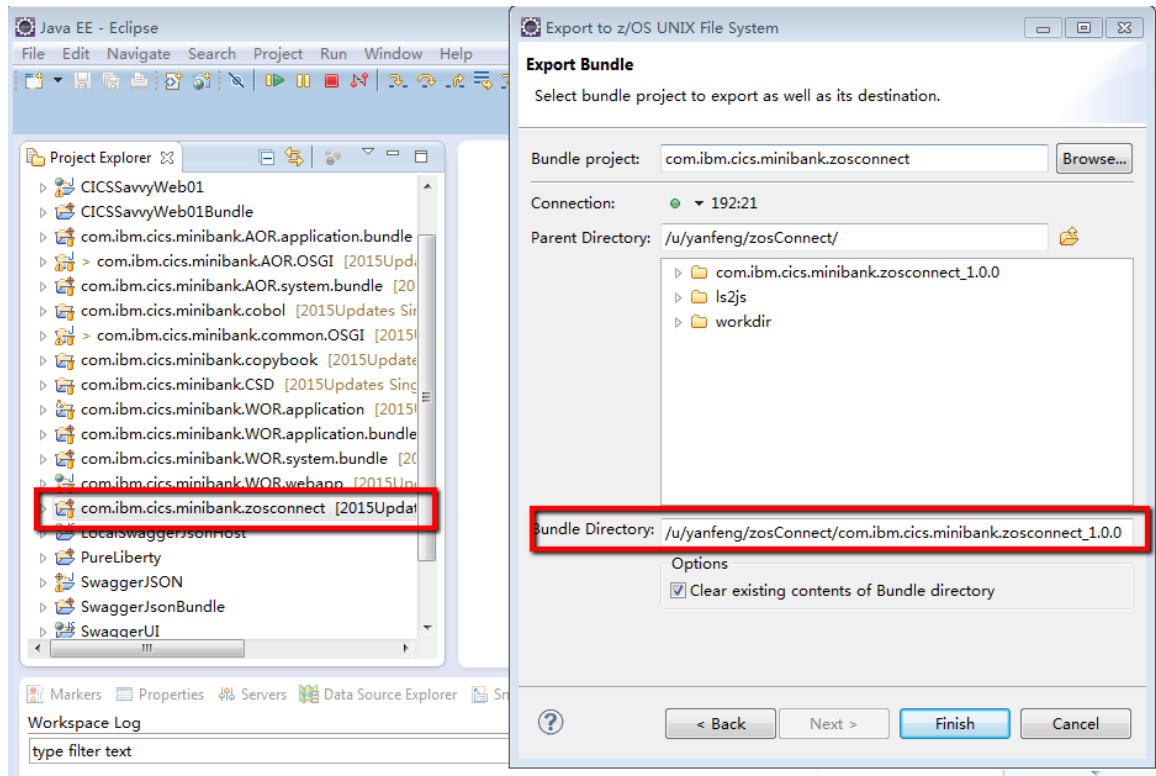
To enable zosConnect feature in CICS, we need to set up a Liberty JVMserver. It is a good practice to configure a separate JVM server for the sole use of z/OS Connect. It is also a good practice to only have a single WebSphere Liberty Profile JVM server that is configured in any single CICS region, so we use a new TOR to host the service (ROR REGION).

A pipeline resource is also needed to set up the JSON webservice for program "TRANSFER". We had prepared the program "TRANSFER" in AOR and wsbind file for the pipeline resource. You can just use the created materials.

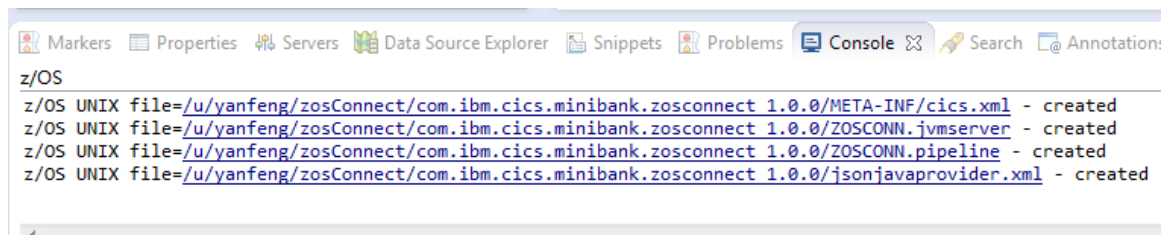
We packaged all the required resources into one CICS bundle project called "**com.ibm.cics.minibank.zosconnect**" which you already imported in page 3, follow below instructions to finish setting up your first zosConnect environment.

Check the project is exported

Follow the instruction in Page 5. Make sure your **com.ibm.cics.minibank.zosconnect** project is exported to HFS, like the screenshot below:



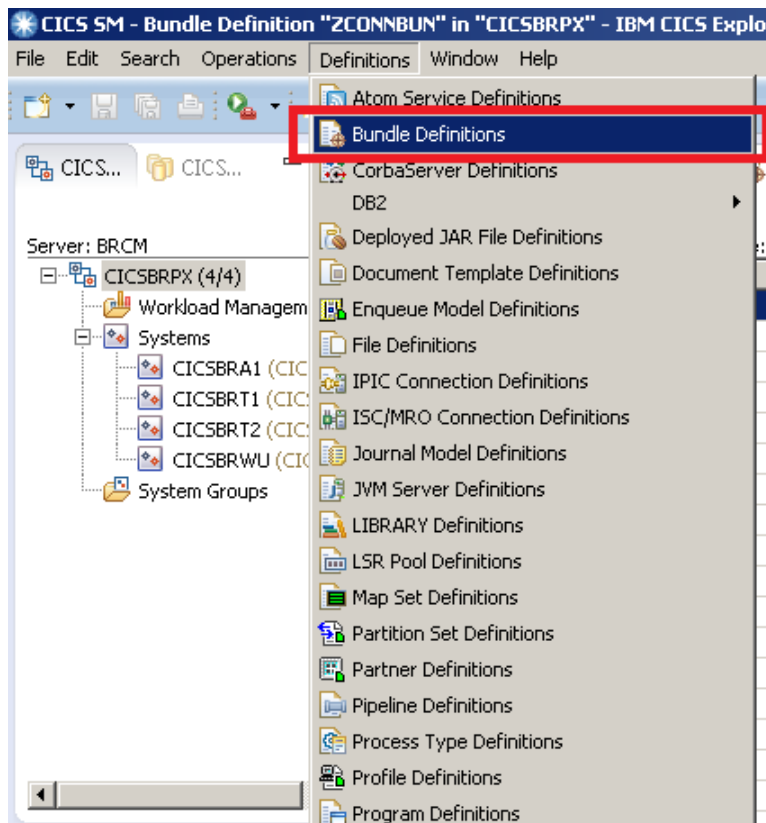
Do not forget to copy the “Bundle Directory” where you upload the bundle files, this directory will be used in next step for bundle definition.



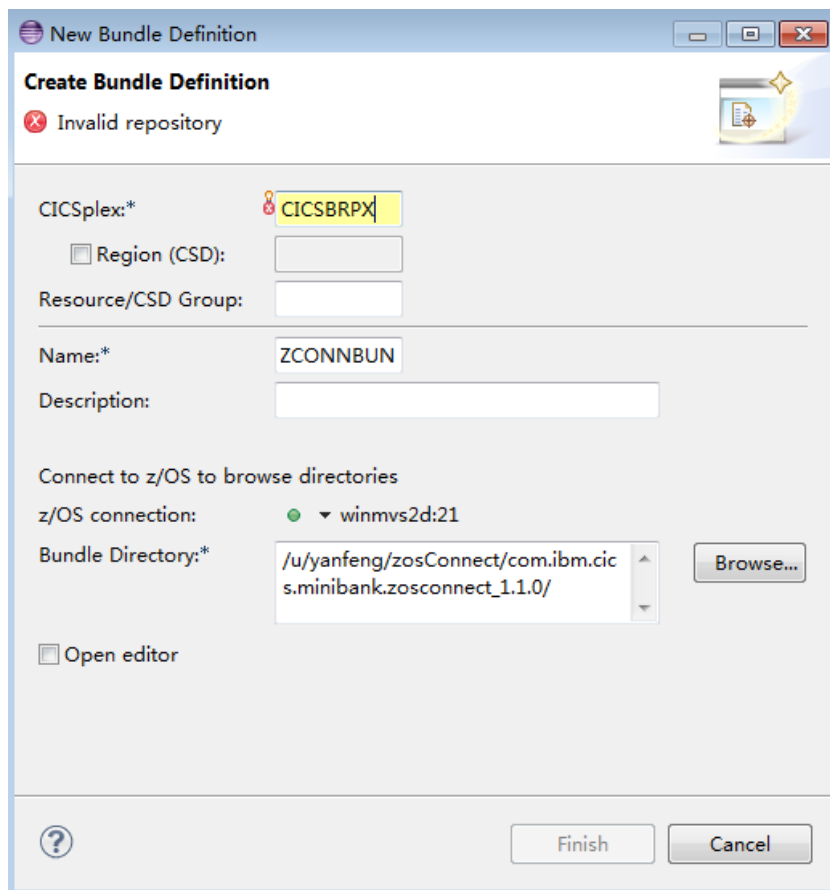
Check the export is successful.

Define bundle resource for Liberty server and pipeline

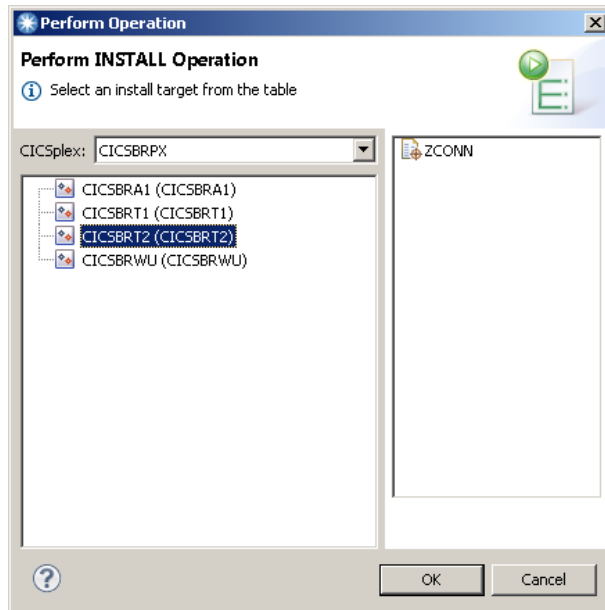
1. Make sure your CICS Explorer can connect to your CPSM environment with CMCI connection. Select 'Definitions-> Bundle Definition' from menu.



2. Right click the 'Bundle Definition' view and select 'New...' to startup Bundle definition wizard. You need give bundle definition a name “**ZCONNUN**” and change **bundle directory** the folder where you exported the bundle project under your USS folder.



3. Right click the new created 'Bundle Definition' and select 'Install' item to install it into your second TOR ROR REGION.



4. You can verify the installed bundle enabled in your ROR by select 'Operations-> Bundle' from CICS Explorer menu.

Region	Name	Bundle ID	Major Vers...	Minor Version	Micro Version	Status	Install Time	Enabled C...
CICSBRT2	ZCONNBUN	com.ibm.cics.minibank.zosconnect	1	0	0	✓ ENABLED	Jun 17, 20...	2

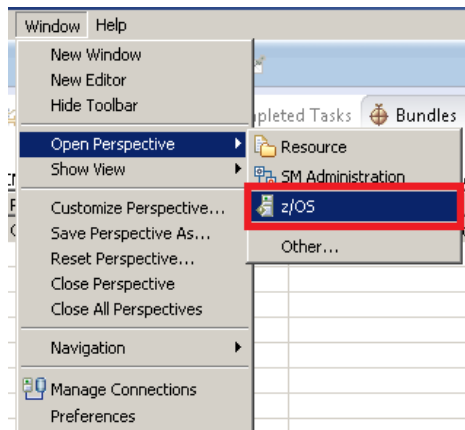
5. Connection between ROR to AOR is installed in advance. Go to Operations->ISC/MRO connections, check your ROR to AOR connection is in acquired status.

What's contained in this bundle?

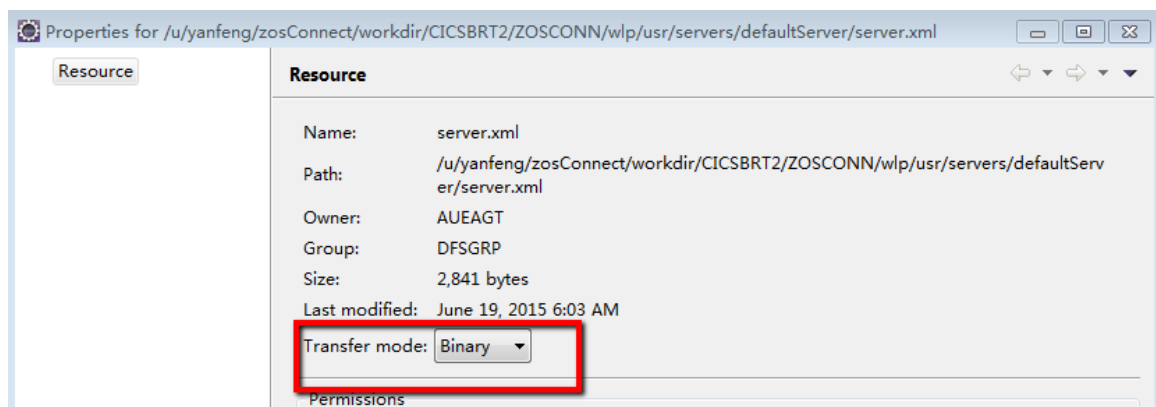
- One JVMserver as zosConnect service host
- One pipeline resource for the json webservice. The wsbind file referenced in pipeline is generated from DFHLS2JS utility.
- A remote program 'TRANSFER', which is the end program to provide the function logic.

Customize server.xml to enable zosConnect feature

1. Switch to 'z/OS' perspective in CICS Explorer.

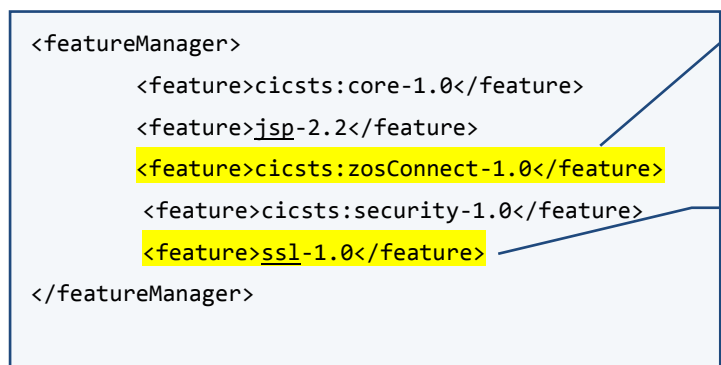


2. Locate the Liberty server.xml under your WORKDIR. The Liberty server.xml will under:
 "/u/ **/zosConnect/workdir/ROR REGION/ZOSCONN/wlp/usr/servers/defaultServer". Please replace ** with your team number.
3. Set server.xml property to **binary** for FTP save purpose, binary transfer won't corrupt the data format. Right click on server.xml, then click on Properties, change transfer mode to Binary. Save "OK".



4. Open server.xml by double click:
 Copy the following statements in yellow highlight to your server.xml

Add support for zosConnect feature in liberty featureManager section.



Enable the zosConnect feature in Liberty server

zosConnect feature requires SSL to be enabled. Double check SSL-1.0 feature is present

Add following statements under featureManager tag.

<pre> <webAppSecurity allowFailOverToBasicAuth="true"/> <com.ibm.cics.wlp.zosconnect.CICSEndpoint id="com.ibm.cics.wlp.zosconnect.CICSEndpointService"/> <authorization-roles id="zos.connect.access.roles"> <security-role name="zosConnectAccess"> <user name="your tso id"/> </security-role> </authorization-roles> </pre>	<p>We will allow failover to HTTP authentication if SSL client certificate authentication is not configured, so add the webAppSecurity element to server.xml</p>
	<p>Define the z/OS Connect for CICS resource program</p>
	<p>Add a security role to access zosConnect component</p>

5. Save your server.xml. You should get a file like the following example:

```

/u/yanfeng/zosConnect/workdir/CICSBRT2/ZOSCONN/wlp/usr/servers/defaultServer/server.xml
<?xml version="1.0" encoding="UTF-8"?><server description="CICS Liberty profile sample configuration">

  <!-- Enable features -->
  <featureManager>
    <feature>cicsts:core-1.0</feature>
    <feature>isp-2.2</feature>
    <feature>cicsts:zosConnect-1.0</feature>
    <feature>ssl-1.0</feature>
    <feature>cicsts:security-1.0</feature>
  </featureManager>

  <!-- Default HTTP End Point -->
  <httpEndpoint host="*" httpPort="6302" httpsPort="6303" id="defaultHttpEndpoint"/>

  <!-- CICS Bundle Installed Applications -->
  <include location="${server.output.dir}/installedApps.xml"/>

  <!-- CICS Default Application -->
  <cicsts_defaultApp deployed="true"/>

  <!-- The following configuration controls how often server.xml

```

```

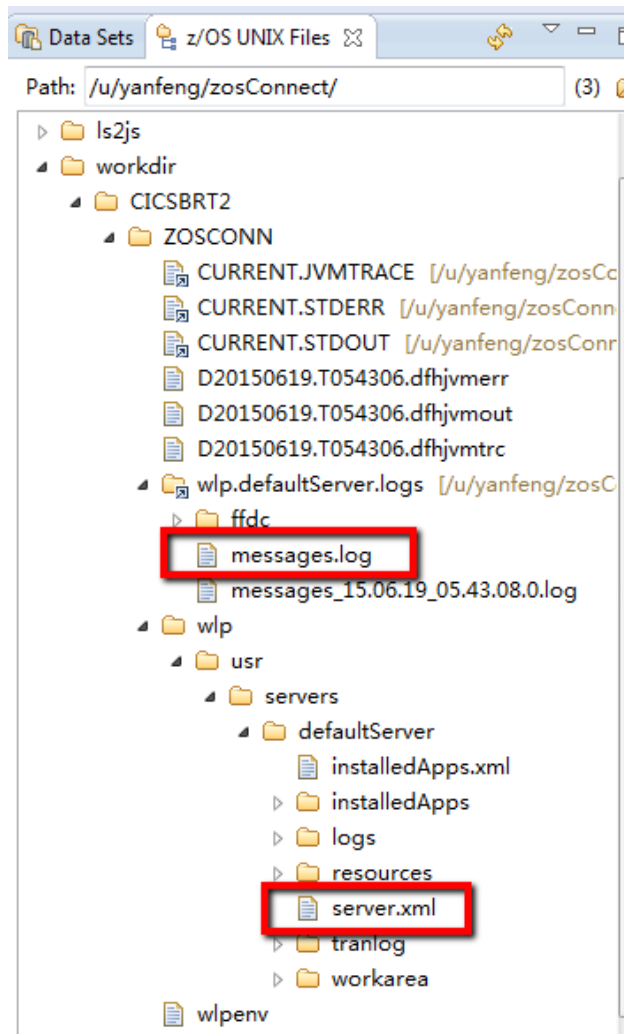
<!-- The following configuration controls how often server.xml
is scanned for updates. The default is every 500ms which may
cause excessive I/O and CPU cost on z/OS.
The values shown below reduce the overhead while still
providing a relatively timely detection of new applications
that have been installed/removed via a CICS Bundle
(WAR bundlepart). If you use CICS bundles to install Web
Applications (WAR files) do not disable the polling.
-->
<config monitorInterval="5s" updateTrigger="polled"/>

<!-- Further scanning is performed to detect application updates or
addition/removal of applications to the dropins directory. If
you are using CICS Bundles as the vehicle for Application
deployment you should disable the dropins directory as shown.
Further I/O and CPU reduction can be achieved by disabling
the application scan. To effect changes to your applications
while the server is still running, you should disable and
re-enable the CICS bundle that contains the Web application.
The pollingRate is only applicable when the updateTrigger is
set to the 'polled' value.
Consult the WebSphere Application Server Liberty Profile
documentation for further information on these parameters.
-->
<applicationMonitor dropins="dropins" dropinsEnabled="false" pollingRate="5s" updateTrigger="disabled"/>
<webAppSecurity allowFailoverToBasicAuth="true"/>
<com.ibm.cics.wlp.zosconnect.CICSEndpoint id="com.ibm.cics.wlp.zosconnect.CICSEndpointService"/>
<authorization-roles id="zos.connect.access.roles">
  <security-role name="zosConnectAccess">
    <user name="YANFENG"/>
  </security-role>
</authorization-roles>

<ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLS"/>
<keyStore id="defaultKeyStore" password="defaultPassword"/>
<zosConnectService invokeURI="/minibankAPI/transfer" serviceName="transferJSON" serviceRef="com.ibm.cics.wlp.zosconnect.CICSEndpointService"/>
</server>

```

6. Check log for zosConnect feature enabled message. Find your liberty server log: The directory should be `"/u/team**/zosConnect/workdir/ROR REGION/ZOSCONN/"`. Example in following picture:



Open message.log file, you should get

“Web Module z/OS Connect has been bound to default_host” message printed indicating the z/OS Connect feature is successfully deployed.

```
I SRVE0169I: Loading Web Module: com.ibm.cics.wlp.defaultapp.
I SRVE0250I: Web Module com.ibm.cics.wlp.defaultapp has been bound to default_host.
A CWWKT0016I: Web application available (default_host): http://testmvs:6302/com.ibm.cics.wlp.defaultapp.
I CWRLS0010I: Performing recovery processing for local WebSphere server (Cell\Node\Server).
I CWRLS0012I: All persistent services have been directed to perform recovery processing for this WebSphere server.
I WTRN0135I: Transaction service recovering no transactions.
I SRVE0169I: Loading Web Module: z/OS Connect.
I SRVE0250I: Web Module z/OS Connect has been bound to default_host.
A CWWKT0016I: Web application available (default_host): http://testmvs:6302/
```

7. Add the TRANSFER service to server.xml. We have created a pipeline resource in the bundle. This pipeline creates a “transferJSON” webservice in our CICS region, we need to register this service to zosConnect.

Still open server.xml, add the following statements

```
<zosConnectService invokeURI="/minibankAPI/transfer"
serviceName="transferJSON"
serviceRef="com.ibm.cics.wlp.zosconnect.CICSEndpointService"/>
```

An entry is needed to associate the U Service with the z/OS Connect receiver.

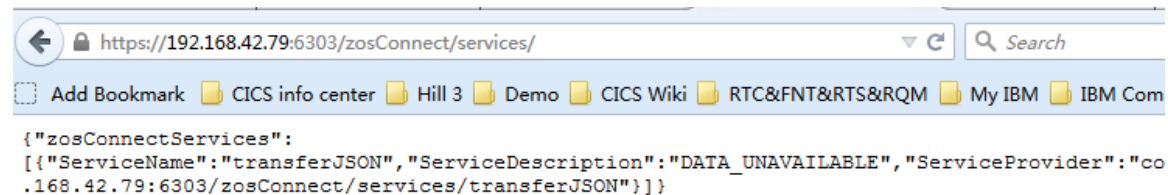
8. Check the “transferJSON” service is successfully deployed. Go to your browser (don’t forget to change the port):

https://your host:<Your zosConnect jvmserver https port, 6**5>/zosConnect/services/ this link can help you to list all available services in this server.

From browser, if there is certificate exception, accept certificate exception.

If a pop up window requires userid and pwd, specify your credential (TEAM**) specified in the security role (Note, the **userid is case sensitive**, you need to keep consistent with the ID your specified in server.xml security role)

Then you should get:



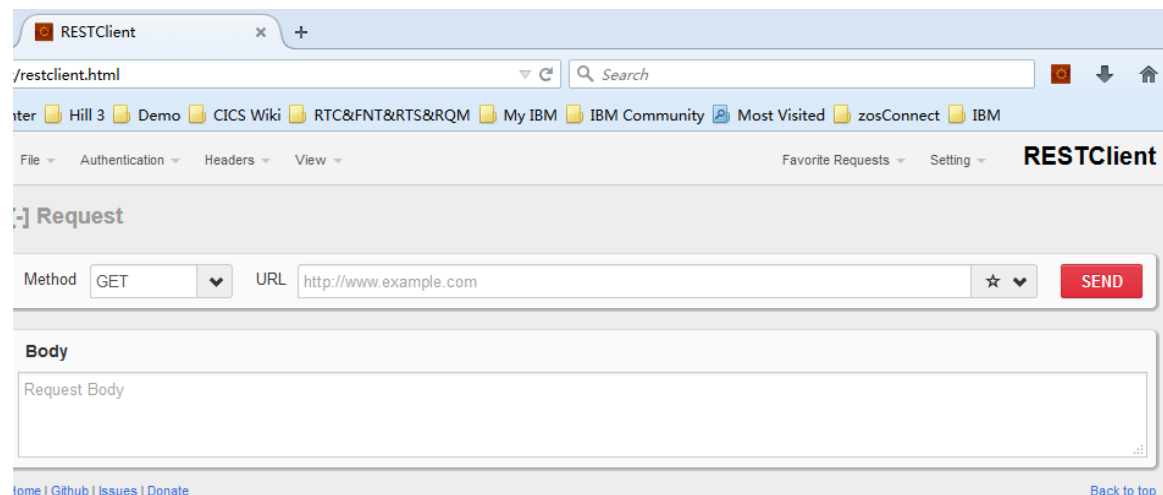
Congratulations!

The service is already registered and can be used now.

Optional: Test the transferJSON API service.

1. We will use Firefox add-on “RESTClient” to do the testing. If you have similar tools can send HTTP rest request, you can change to use your own one.

Search in Firefox Add-On lib, install “RESTClient”, after install and restart, it should be like this:



2. Config parameters:

Method: POST

URL: https://your host:<Your zosConnect jvmserver https port, 6**5>/minibankAPI/transfer

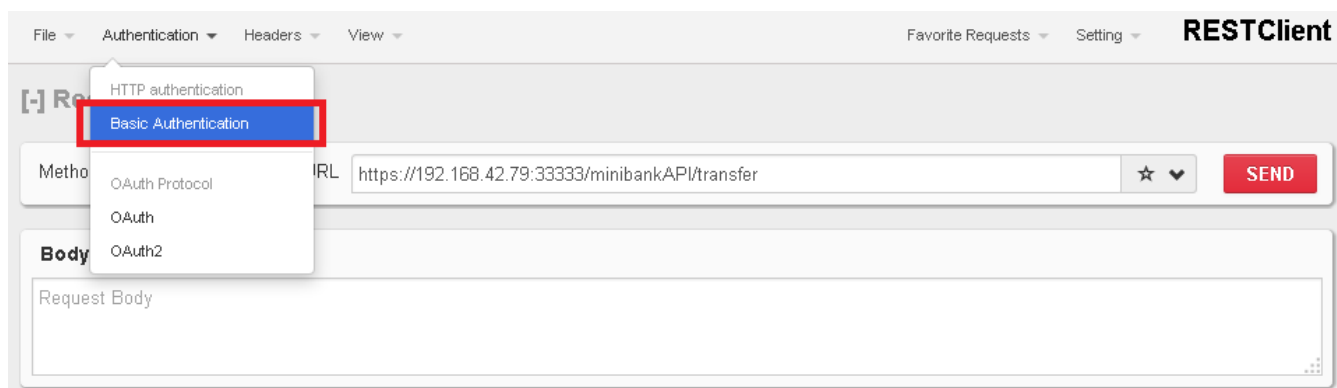
(Replace your own HTTPS port)

Body: Give the following JSON data.

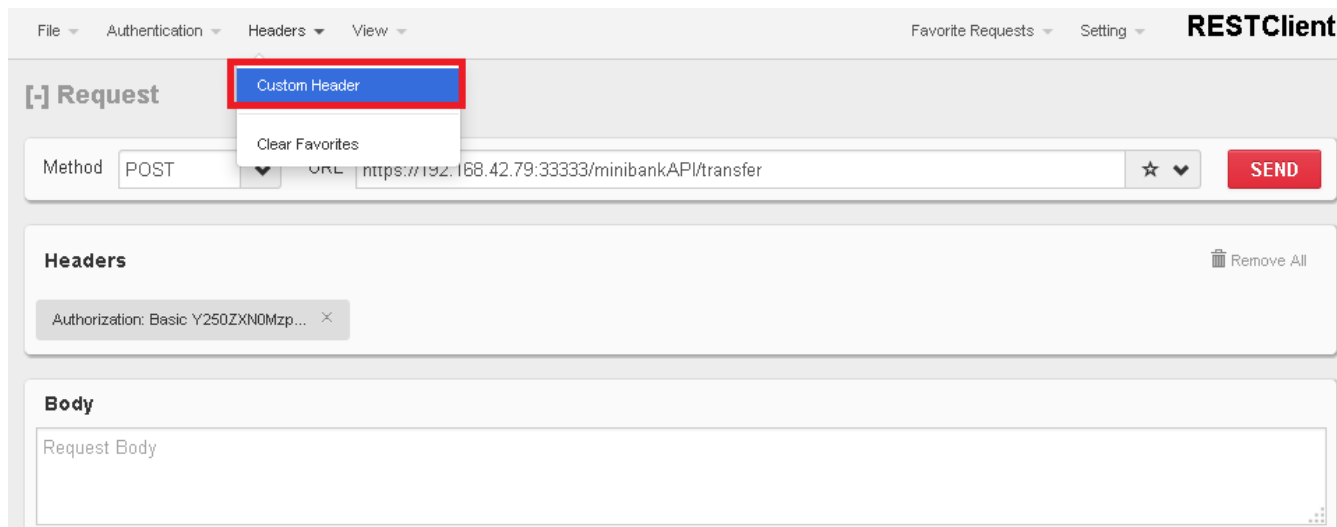
(Note: the account number need to be created first, use the number we provide is advised)

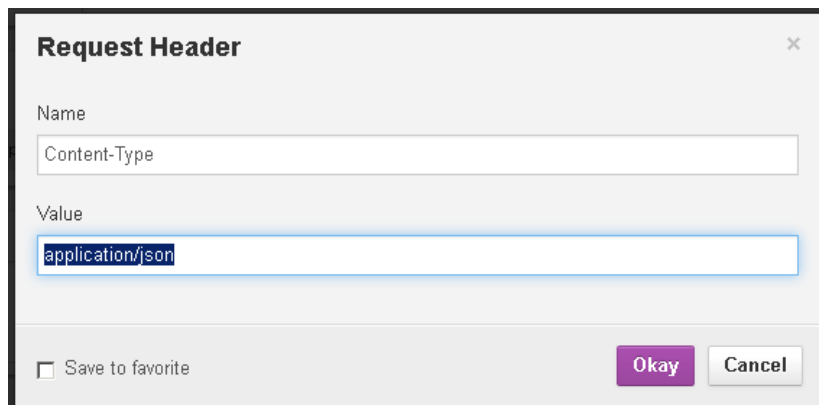
```
{
  "TRANSFEROperation": {
    "source_account": "0123456789",
    "target_account": "0123456788",
    "trans_amount": "100.25",
    "trans_time": "2015-05-15",
    "tran_result": "1",
    "error_msg": "transfer fail",
    "source_balance": "0",
    "target_balance": "0"
  }
}
```

Specify Basic Authentication and content-type Header:
You need input your MVS ID/PWD as basic authorization.



Add JSON content-type header into request in RESTClient as below:





A dialog box titled "Request Header" with a close button (X) in the top right corner. It contains two input fields: "Name" with the text "Content-Type" and "Value" with the text "application/json". At the bottom, there is a checkbox labeled "Save to favorite" which is unchecked, and two buttons: "Okay" and "Cancel".

3. Test! Press SEND button. You should get:

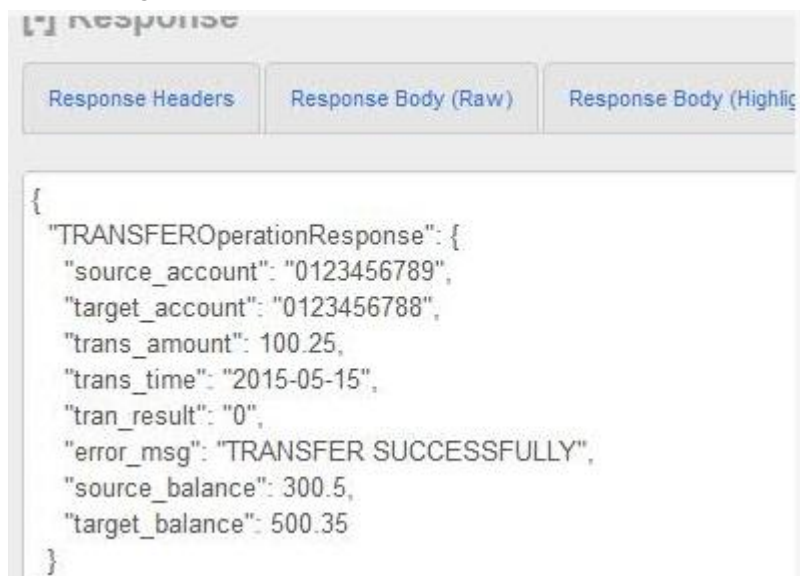


The "Response" window with the "Response Headers" tab selected. It displays a list of response headers:

No.	Header	Value
1.	Status Code	200 OK
2.	Content-Language	en-US
3.	Content-Type	application/json
4.	Date	Fri, 19 Jun 2015 08:22:31 GMT
5.	Transfer-Encoding	chunked
6.	X-Powered-By	Servlet/3.0

Check "Response Body":

You should get "TRANSFER SUCCESSFULLY" result.



The "Response" window with the "Response Body (Raw)" tab selected. It displays a JSON object:

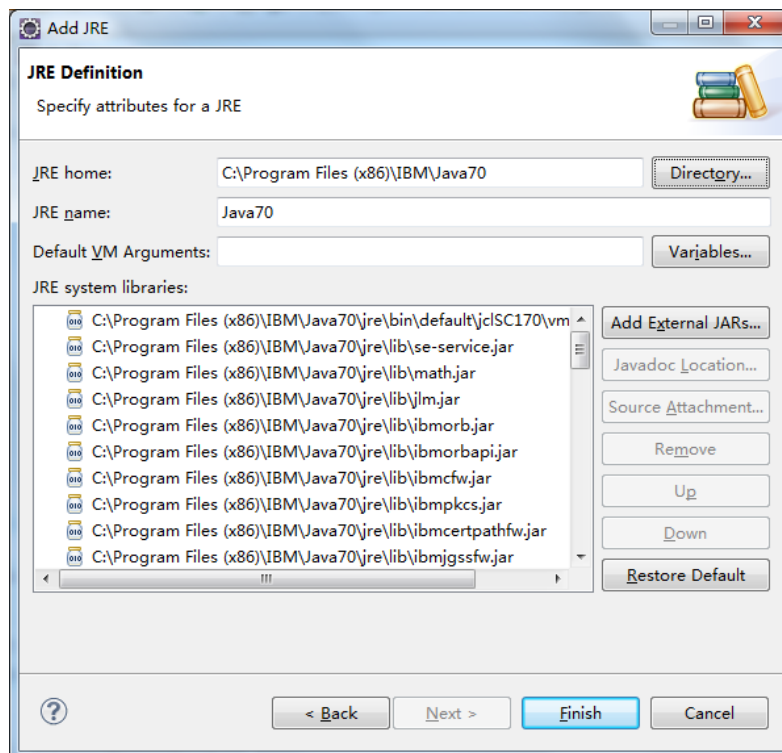
```
{
  "TRANSFEROperationResponse": {
    "source_account": "0123456789",
    "target_account": "0123456788",
    "trans_amount": 100.25,
    "trans_time": "2015-05-15",
    "tran_result": "0",
    "error_msg": "TRANSFER SUCCESSFULLY",
    "source_balance": 300.5,
    "target_balance": 500.35
  }
}
```

Congratulations!

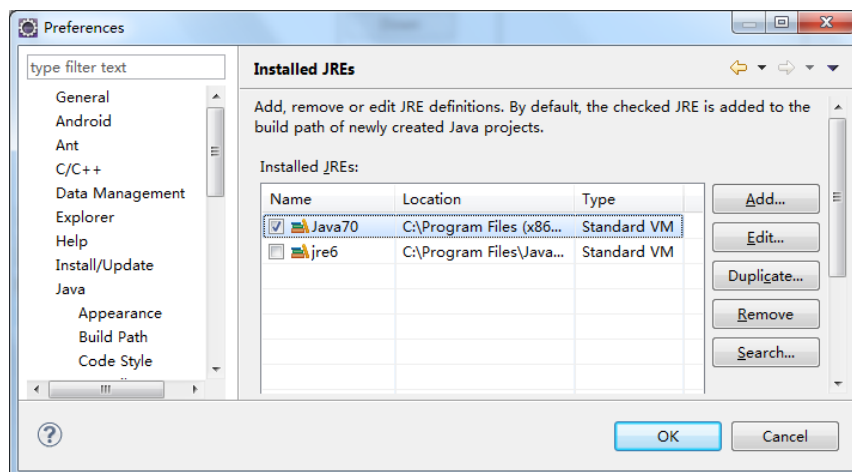
Appendix A: Set up Eclipse with CICS Explore SDK for CICS Java development

Setup JRE

1. Go to “Window -> Preferences -> Java -> Installed JREs”, to add a JRE definition pointing to your local JRE folder. Please note this workshop need JRE 1.7. Like below:



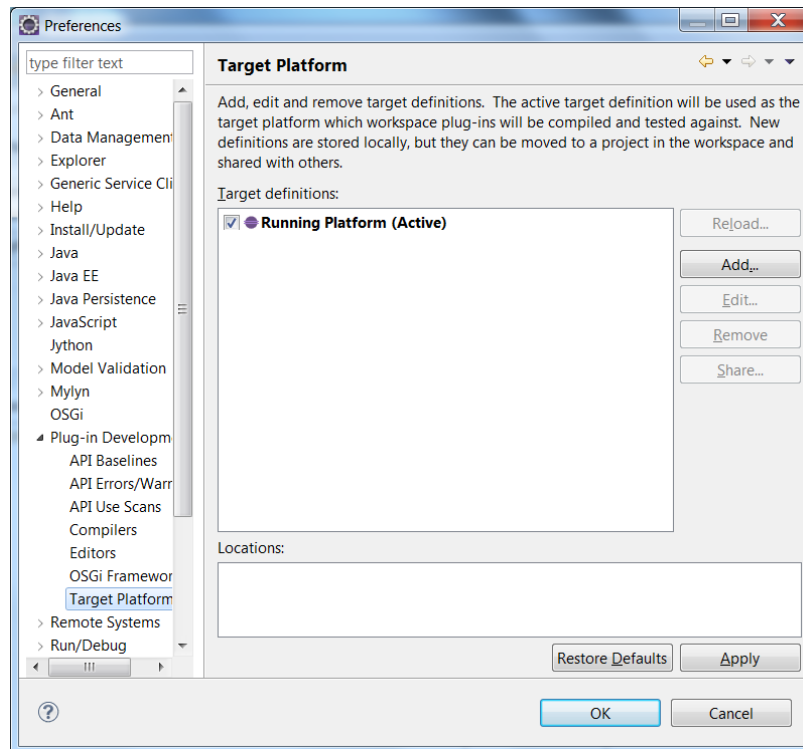
2. Click the “Finish” button and you will see the JVM named Java70 been added to the available JVM list:



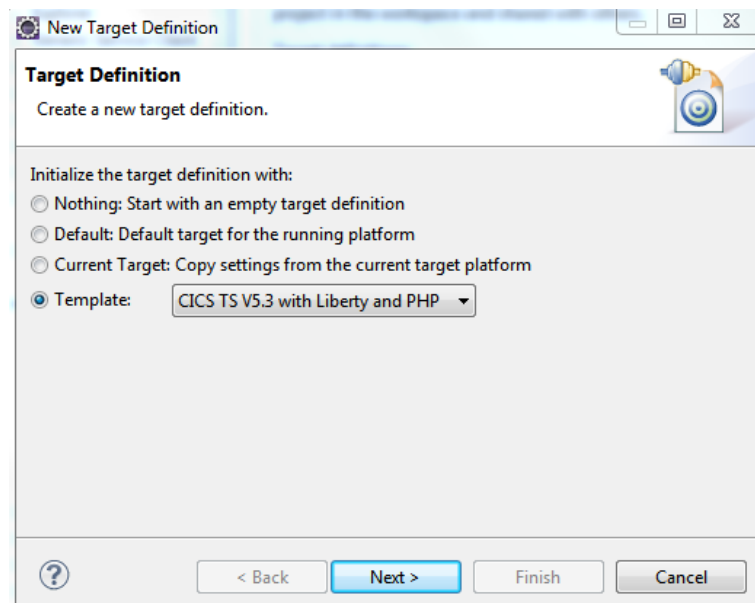
Click the **Java70 JVM** and click “OK” button

Setup Plug-in Target Platform

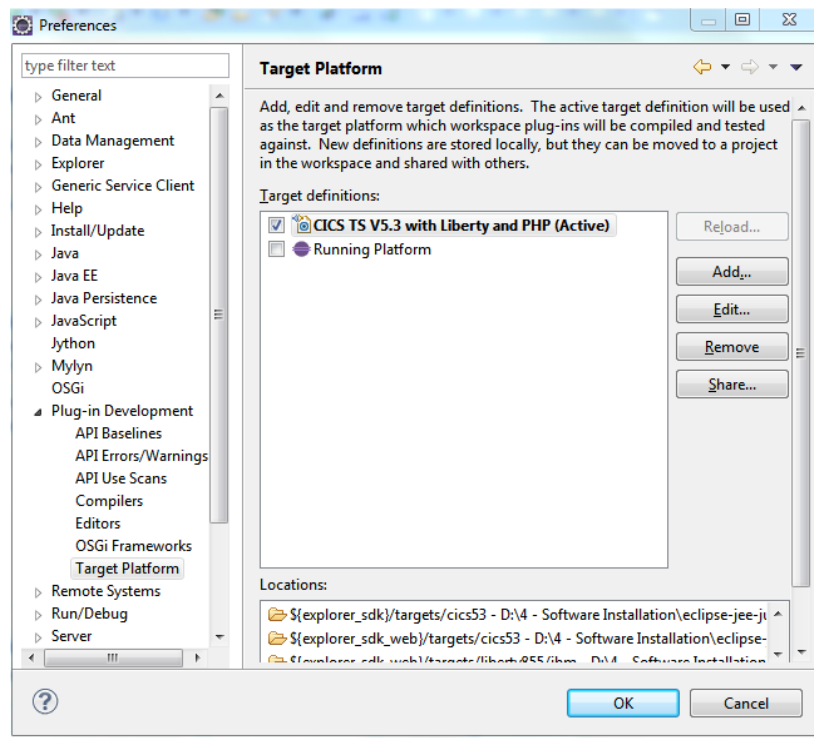
1. Go to “Window -> Preferences”, in the left list, jump to “Plug-in Development -> Target Platform”, click “Add”:



2. Select “Template”, in the drop-down list, and select “CICS TS V5.3 with Liberty and PHP”, click Next and then Finish to close the Target Definition window.



You will see:



3. Click the “CICS TS V5.3 with Liberty and PHP” to active it, and then click OK to exit the Preference window.

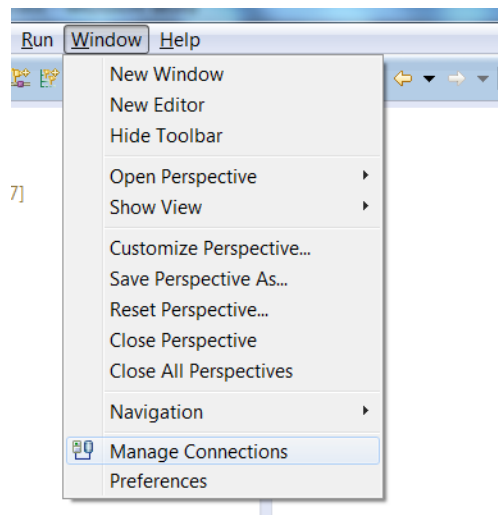
Appendix B: Set up connection between CICS Explorer and mainframe for the workshop

To support this workshop, you need to set up two types of connection for CICS Explorer:

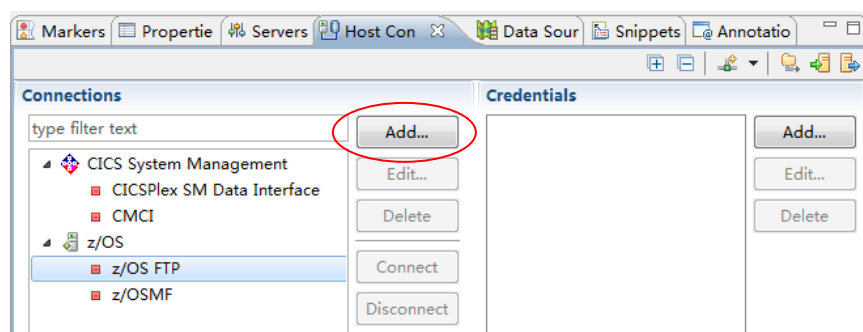
1. z/OS FTP connection
2. CMCI connection

Configure FTP connection to MVS

1. Here we need to establish a FTP connection to MVS so that we can easily deploy CICS bundles into MVS USS. In Eclipse, go to Window -> Show View -> Other -> Host Connections



You will see the Host Connection view as below:



2. Select the z/OS FTP, and click the Add button to configure the z/OS FTP connection to 192.168.42.79:

Add z/OS FTP Connection

Specify the host, port, and any additional details for the new connection

Name: SINGAPORE_FTP

Location

Host name: 192.168.42.79

Port number: 21 ☐ Secure connection (TLS/SSL)

FTP Additional Details

☐ Active transfer mode

MBCS support: (none)

SBCS local character set: GB18030

SBCS remote character set:

- Click "Save and Close", in the list you can see a z/OS FTP connection configuration is created:

Connections

type filter text

- ▲ CICS System Management
 - CICSplex SM Data Interface
 - CMCI
- ▲ z/OS (1)
 - ▲ z/OS FTP (1)
 - SINGAPORE_FTP
 - z/OSMF

- In the Credentials panel, click Add button to add a credential to connect MVS:

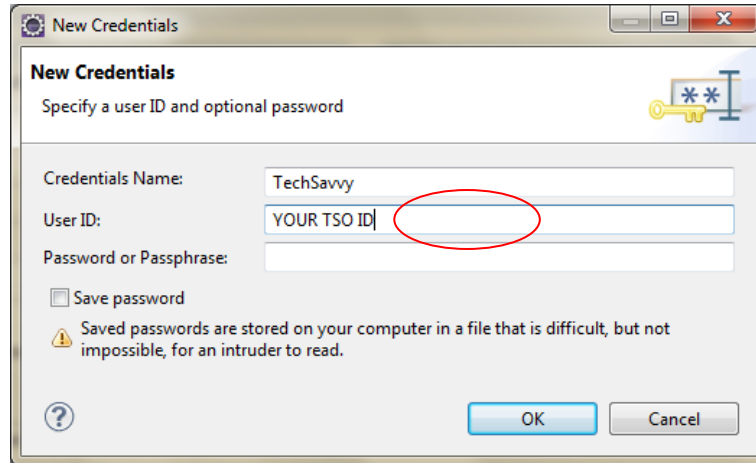
Connections

type filter text

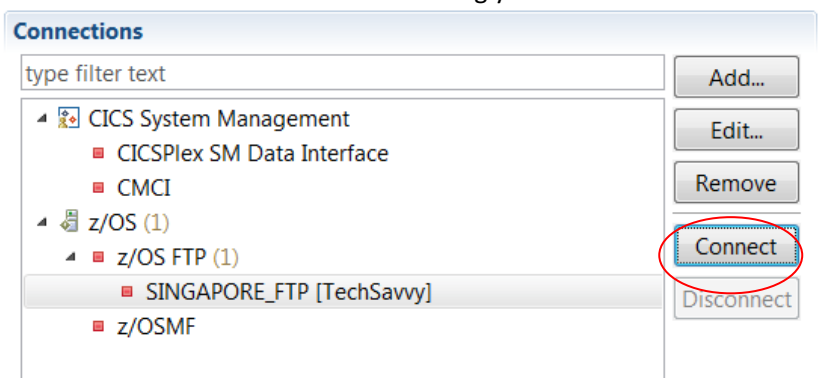
- ▲ CICS System Management
 - CICSplex SM Data Interface
 - CMCI
- ▲ z/OS (1)
 - ▲ z/OS FTP (1)
 - SINGAPORE_FTP
 - z/OSMF

Credentials

Click **OK** to create the credential.

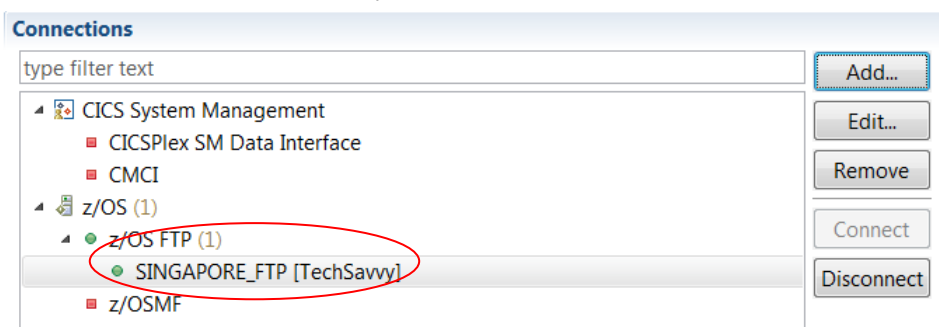


5. Connect to the 192.168.42.79 using your credential:



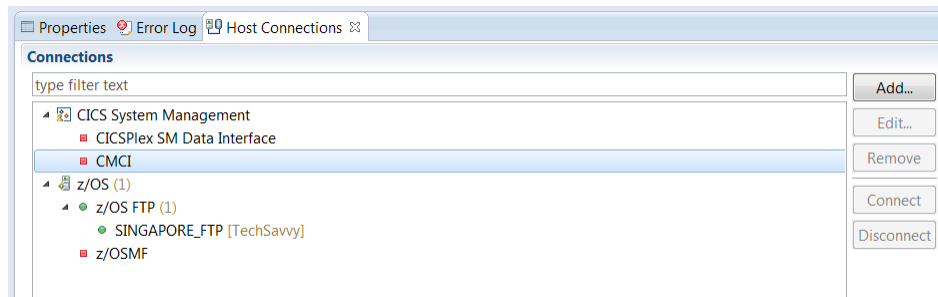
Using the credential we just created to connect to MVS

6. If successfully connected, you will see the TechSavvy connection turns to green and Connect button turns Gray:

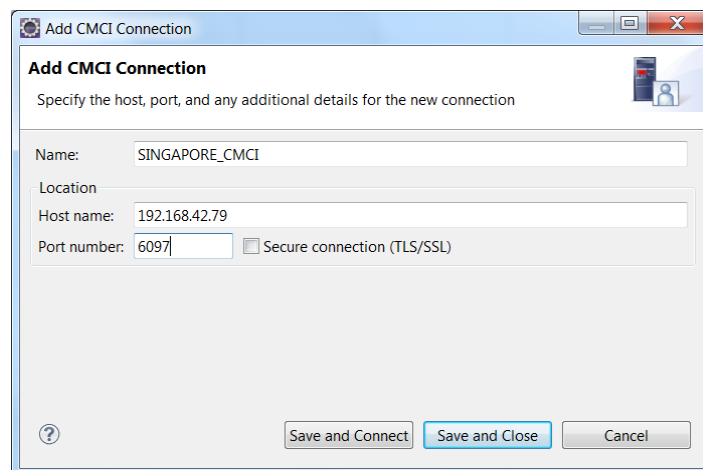


Configure CMCI connection to MVS

1. Select CMCI as connection type, and press Add.



2. Input Name, Host name and port number. This port number is CMCIPORT (6**1)assigned for your WUI region.



3. Save and connect using your own credential. You will be able to see the CMCI connection is configured successfully.

