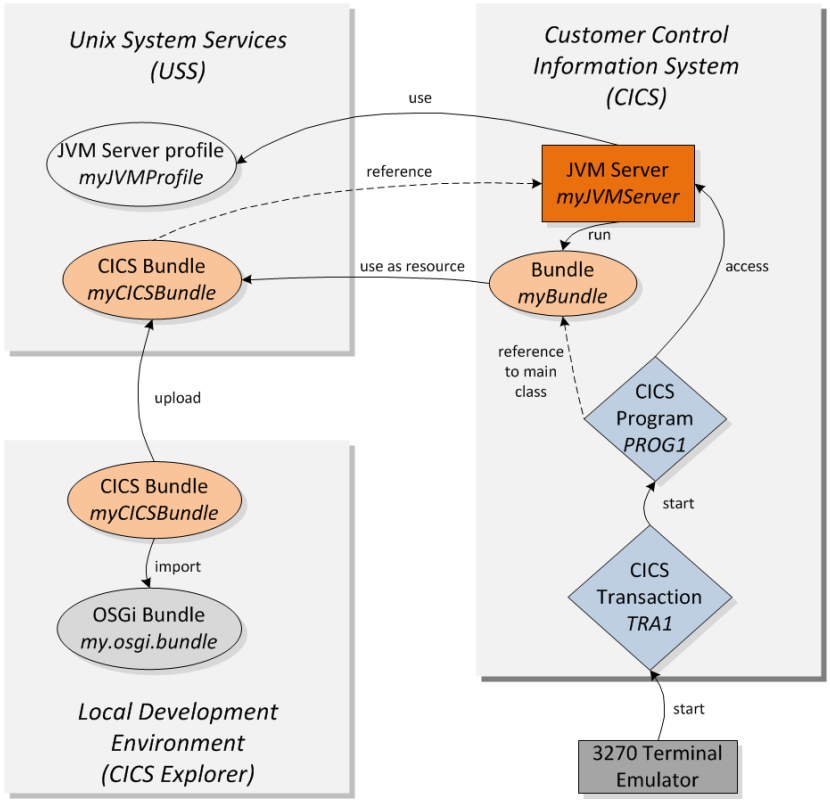
Practical Application Development with the CICS JVM Server

This tutorial practically shows the installation and configuration of the CICS Explorer as well as an example for the development of Java applications for the JVM Server.
The CICS Explorer is an interface for the local Eclipse IDE that is used for the management of CICS JVM Servers. It enables to communicate with the USS file system and with the CICS region.
In order to create a Java application for the JVM Server the following tasks have to be carried out (excluding the configurational tasks). Note that the basic procedure for this tutorial has been adapted from Transaction Server for z/OS Version 4 Release 2 - Java Applications in CICS.

1. Create a **Plug-in Project** in Eclipse that represents the OSGi bundle
2. Create a CICS bundle
3. Include the OSGi bundle into the CICS bundle and assign the JVM Server name needed for its execution
4. Export the CICS bundle into the USS file system
5. Create the following definitions in the CICS SM Perspective
   (a) JVM Server
   (b) Bundle
   (c) Program
   (d) Transaction
6. Install all definitions into the CICS region

A step by step procedure for all tasks is provided within this tutorial.

The interaction of OSGi bundles, the JVM Server, CICS programs and transactions is illustrated in the following



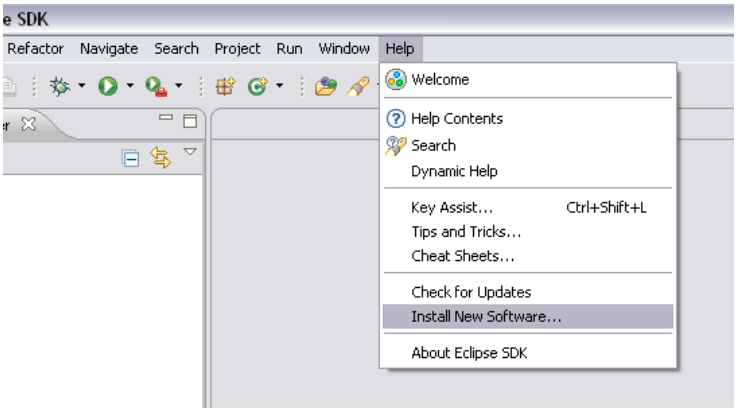
Install the CICS Explorer SDK into Eclipse

Requirements

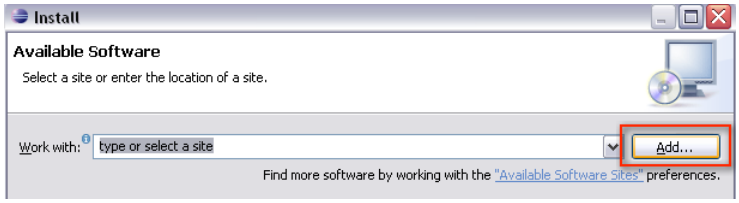
- Eclipse IDE v. 3.7 (and higher).
- CICS Explorer SDK (download from `http://www-01.ibm.com/software/htp/cics/explorer/`).

1. Start your Eclipse IDE.

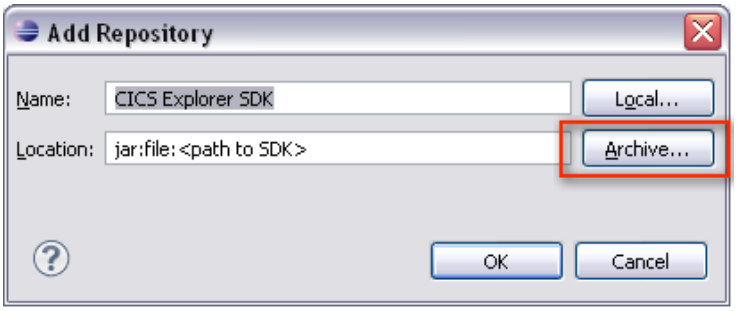
2. Go to **Window → Install New Software...**.



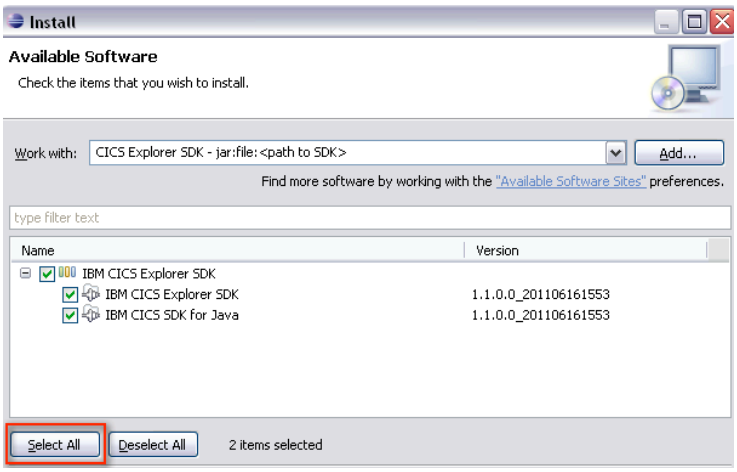
3. Click on **Add...** in the **Install** panel.



4. Name the repository, click on **Archive...** and browse to the CICS Explorer SDK *zip* file.



5. Select all options in the **Install** panel and click **Next**.

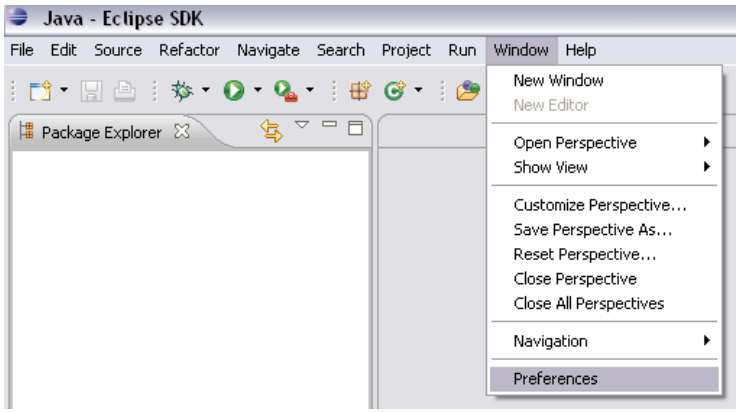


6. Accept the license agreement and finish the installation.
After restarting Eclipse several new Perspectives will be available. The ones important for this tutorial are the **CICS SM** and the **z/OS**.

Configure required Connections

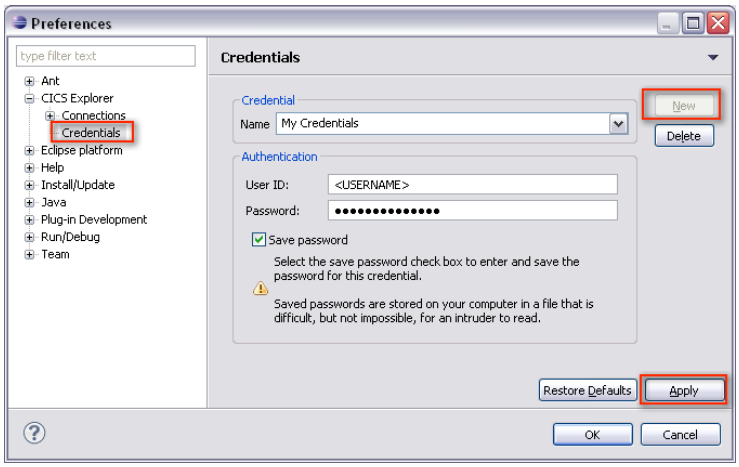
In order to run applications the connections to the CICS Management Interface and the z/OS interface need to be configured.

1. Go to **Window** → **Preferences**.

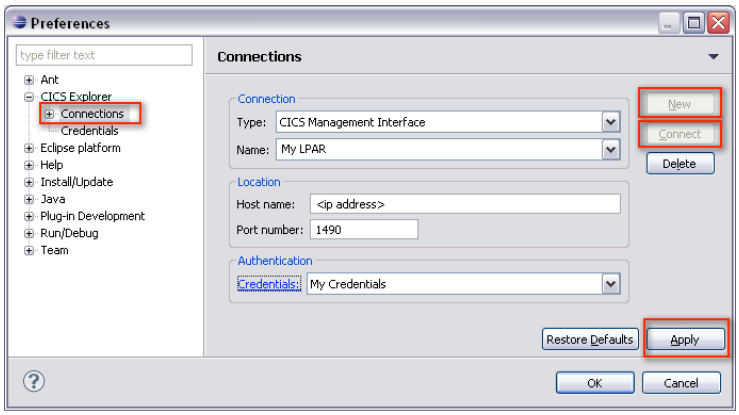


2. In the **Preferences** panel expand the **CICS Explorer**menu, click on the **Credentials** node and click on **New**.

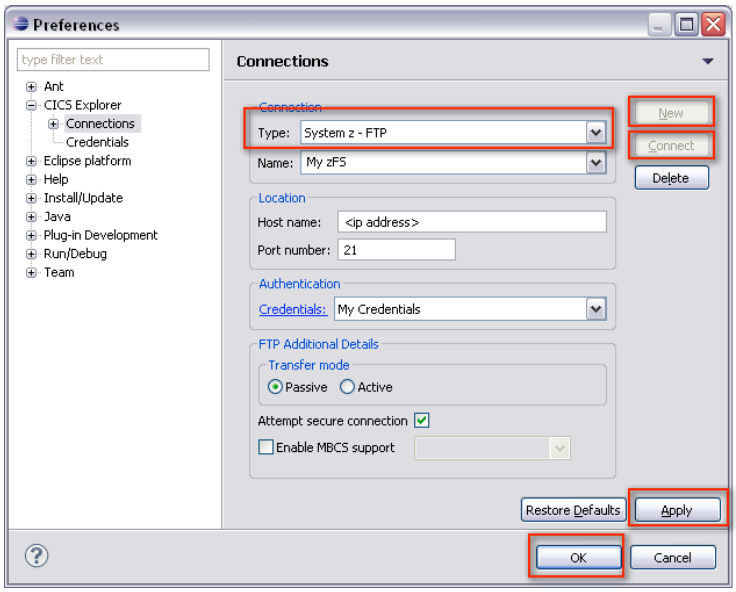
3. Name the credentials and enter your username and password and click on **Apply**. Keep the **Preferences** panel open.



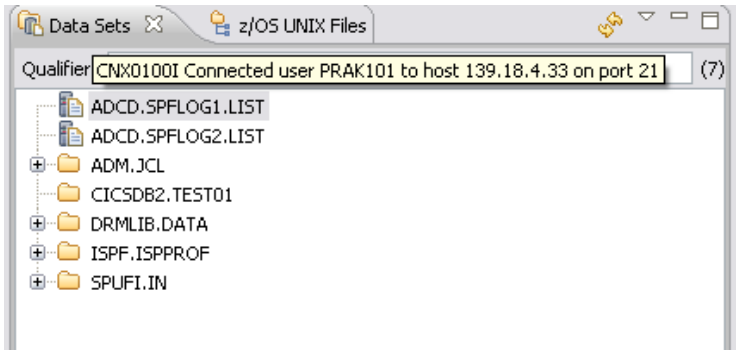
4. Click on the **Connections** node and create a new **CICS Management Interface** connection using the previously defined credentials. The default port for the CICS Management Interface is **1490**. Click on **Apply** and click on **Connect**. Keep the **Preferences** panel open.

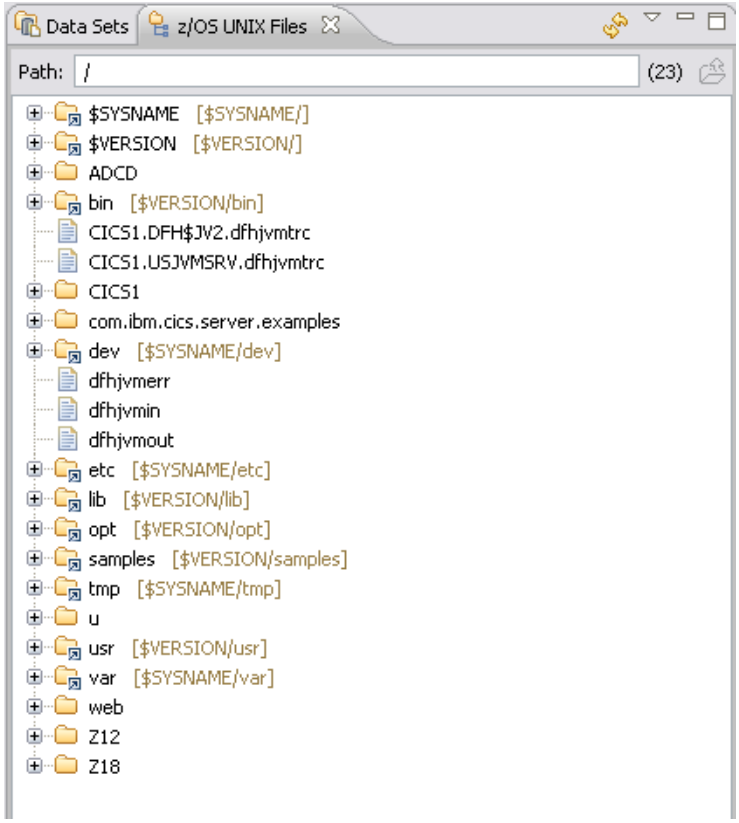


5. Choose **System z - FTP** as connection type and click on **New** to create a new FTP connection using the previously defined credentials. The default port for the **System z - FTP** is **21**. Apply the changes, click on **Connect** and click on **OK** to close the **Preferences** panel.



After successfully connecting to the System Z - FTP, you will be able to see all your z/OS and USS files within the **z/OS** Perspective.



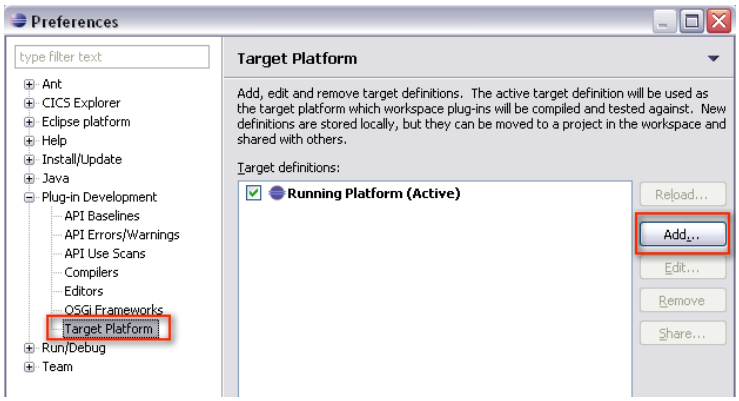


Configure the Target Platform

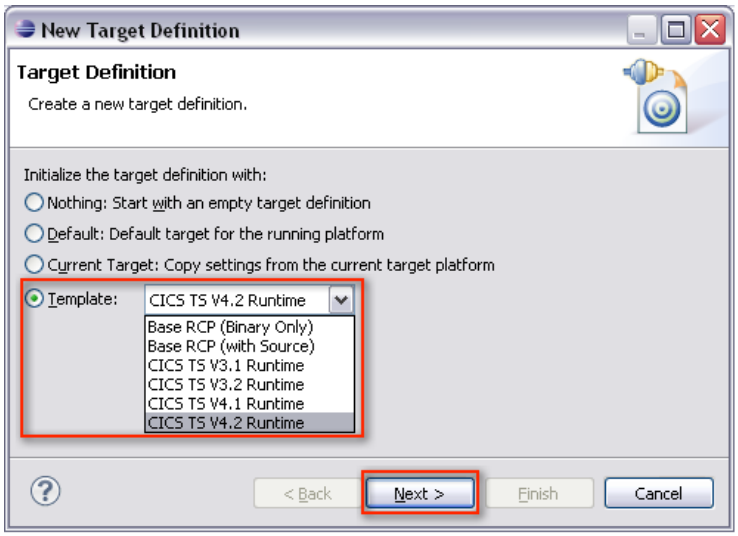
In order to develop applications for the CICS JVM Server, the target platform needs to be configured for the CICS TS 4.2 environment.

1. Go to **Window** → **Preferences**.

2. In the **Preferences** panel expand the **Plug-in Development** menu, click on the **Target Platform** node and click on **Add**.



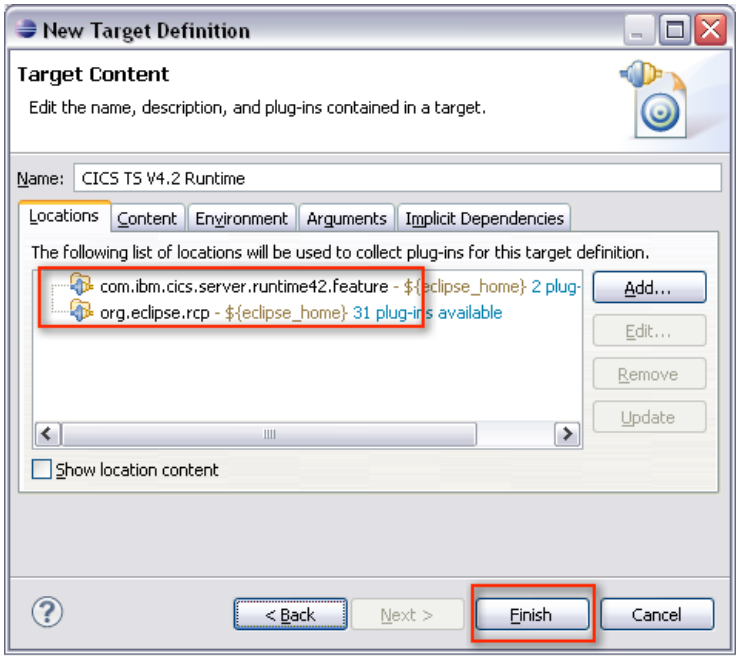
3. Choose **CICS TS 4.2 Runtime** as the template in the new window and click **Next**.



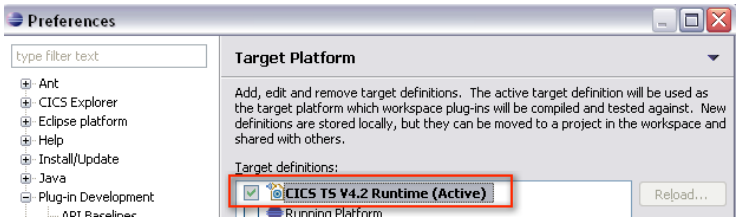
4. Review if the locations

- **com.ibm.cics.server.runtime42.feature** and
- **org.eclipse.rcp**

are included in the **Locations** tab of the **Target Content** panel and click on **Finish**.



5. Choose **CICS TS V4.2 Runtime** as the standard target platform and klick **OK**.

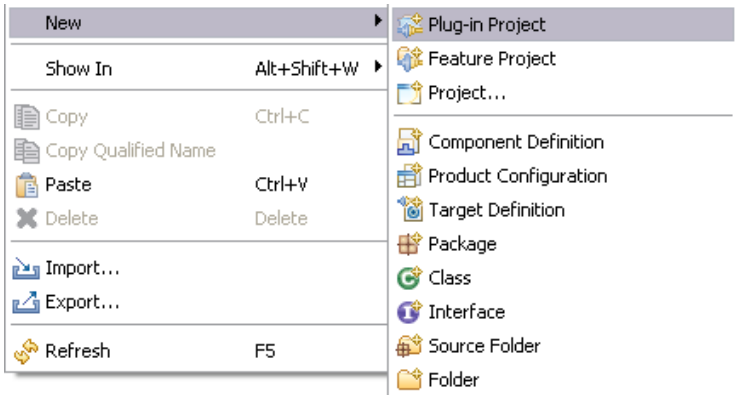


Create a HelloWorld Application

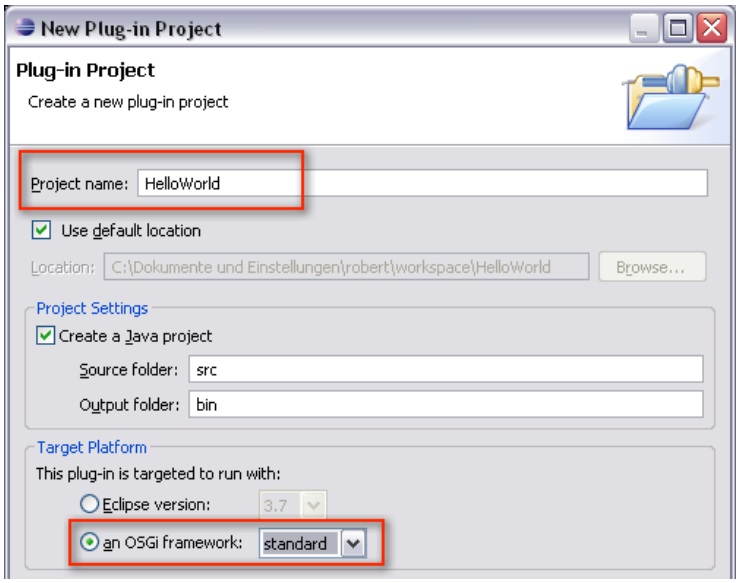
Since Eclipse is based on the OSGi Equinox Framework, OSGi bundles in Eclipse are designated as **Plug-ins**.

1. Open the **Plug-in Development** Perspective.

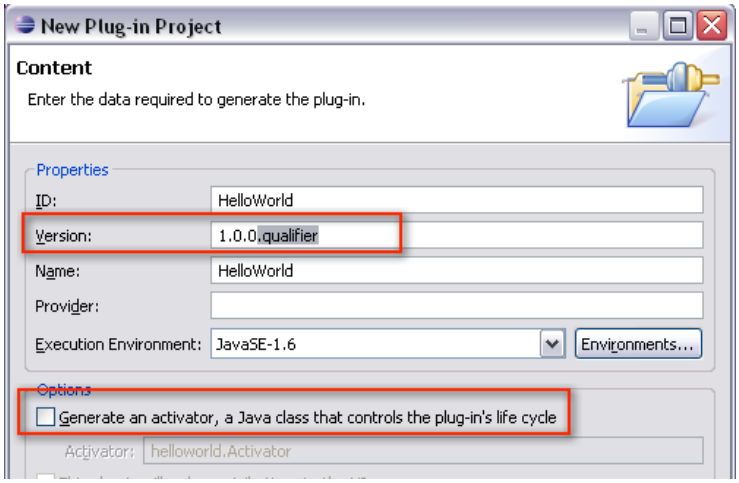
2. Create new Plug-in Project.



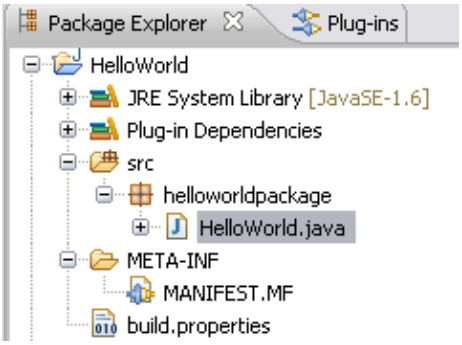
3. Enter the name of the project, choose **standard** as the target platform and click **Next**.



4. Remove the *qualifier*, uncheck the option for the generation of an activator and click **Next**.



5. Create a package **helloworldpackage** and a `HelloWorld` class in the previously created project.



6. Add the following code into the `HelloWorld` class (adapted from the **Hello CICS World** CICS Explorer Example).

```
package helloworldpackage;

import com.ibm.cics.server.*;

public class HelloWorld {

    public static void main(CommAreaHolder CAH)
    {
        Task t = Task.getTask();
        if ( t == null )
            System.err.println("HelloCICSWorld example: Can't get
                Task");
        else
                    t.out.println(" transaction started");
            t.out.println("Hello from a Java CICS application");
    }
}
```

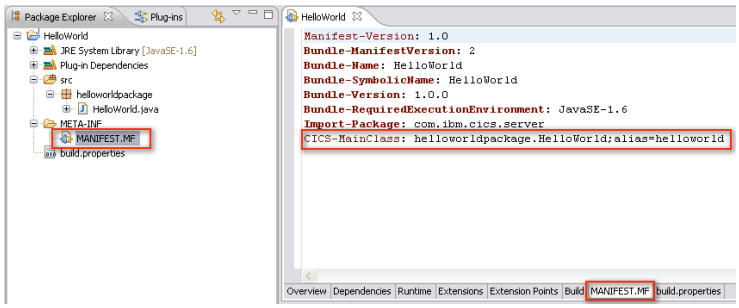
7. Open the MANIFEST.MF and add

`Import-Package: com.ibm.cics.server`

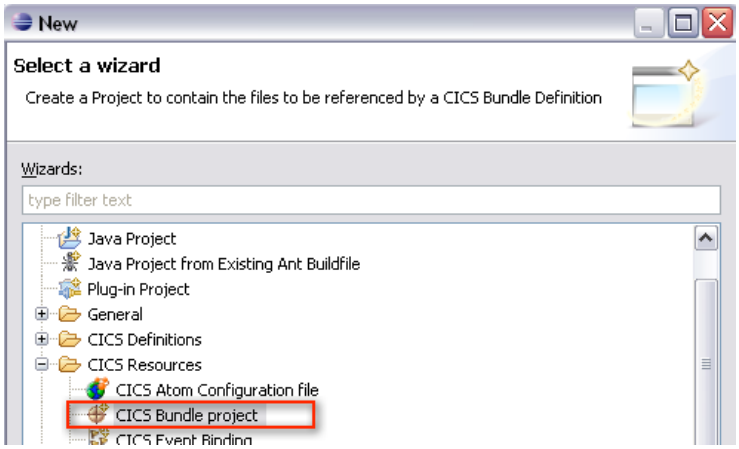
to enable the package import carried out in the `HelloWorld` class and

`CICS-MainClass: helloworldpackage.HelloWorld;alias=helloworld`

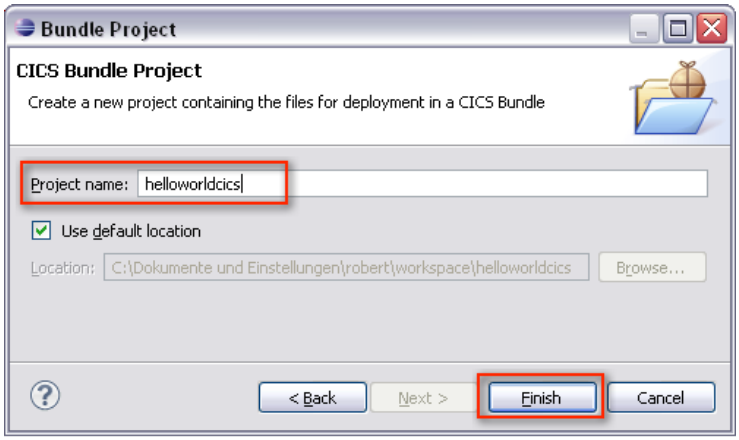
to identify the bundle and its package holding the main class. Press the key combination <**CTRL**>+**S** on your keyboard to save your changes. Note that the `alias=helloworld` statement assigns an optional alias for the main class. Also note that the last statement of the header file should include a line break (in this case a line break should appear in the CICS-MainClass line).



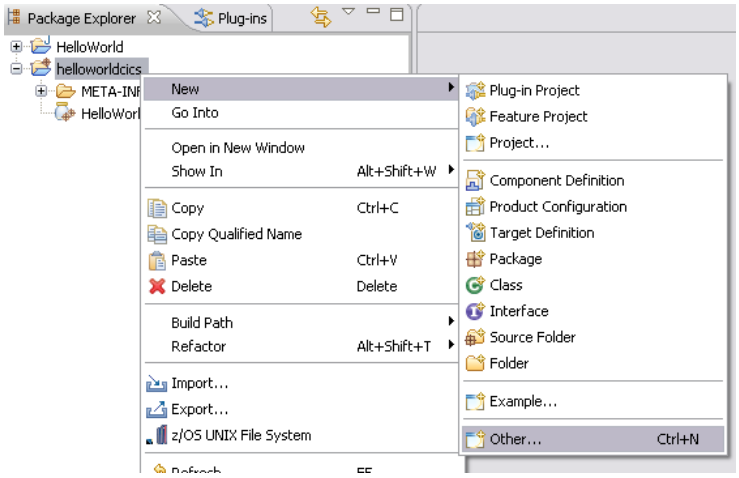
8. Go to **File** → **New** → **Other..** → **CICS Resources**, choose **CICS Bundle project** and click **Next** to create a CICS Bundle.



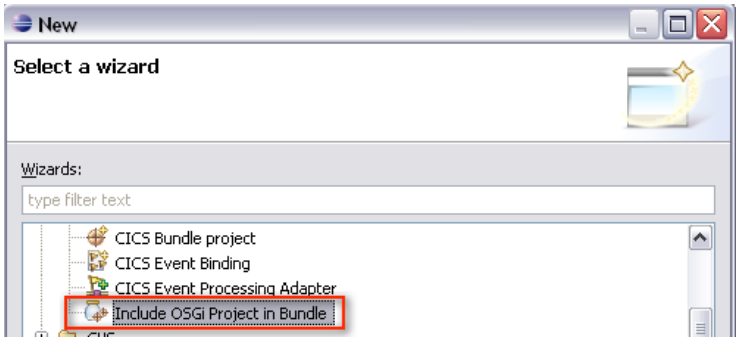
9. Name the CICS bundle and click **Finish**.



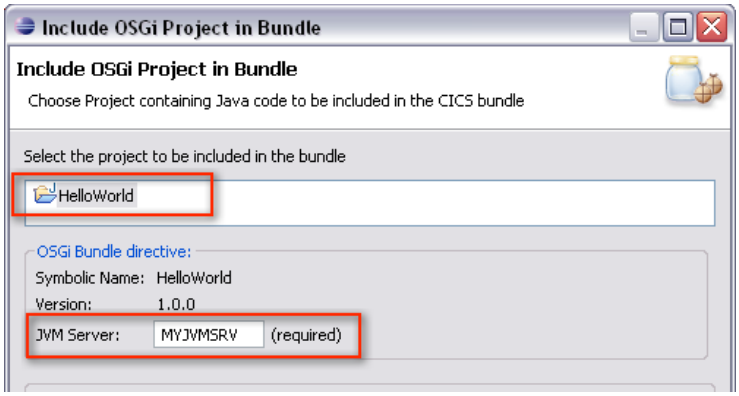
10. Perform a right-click on the previously created CICS bundle and go to **New → Other...**.



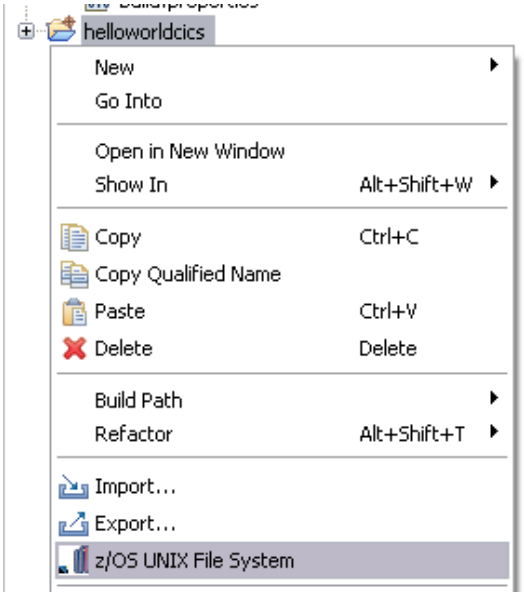
11. Choose **CICS Resources → Include OSGi Project in Bundle** and click **Next**.



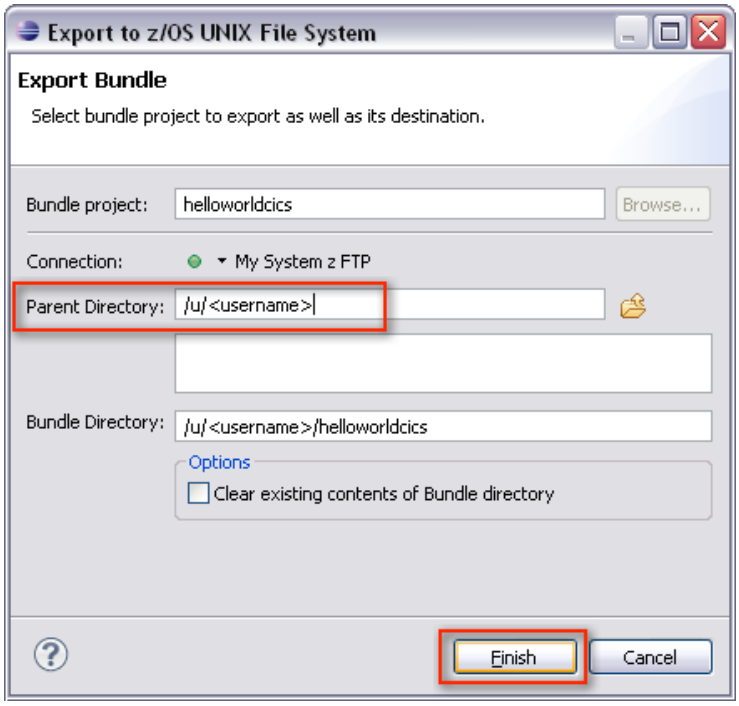
12. Choose the **HelloWorld** Plug-in project (OSGi bundle), enter the name of the JVM Server your application will use and click **Finish**. Note that a definition of a JVM Sever will be described step 18.



13. Click right on the previously created CICS bundle and go to **z/OS UNIX File System** to export the CICS Bundle into the USS file system.

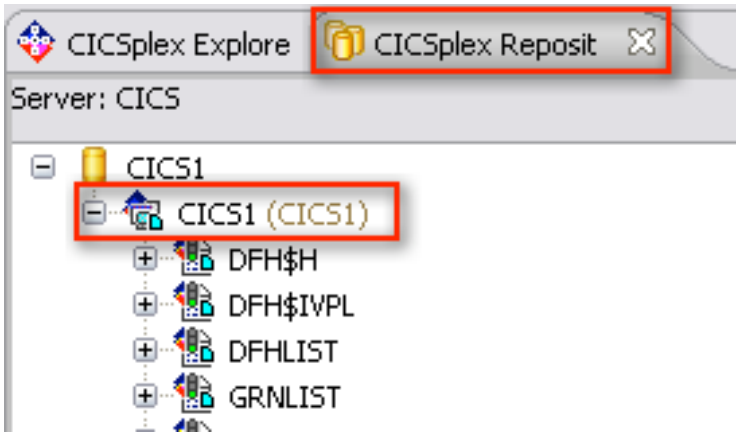


14. Choose the export directory and click **Finish**.



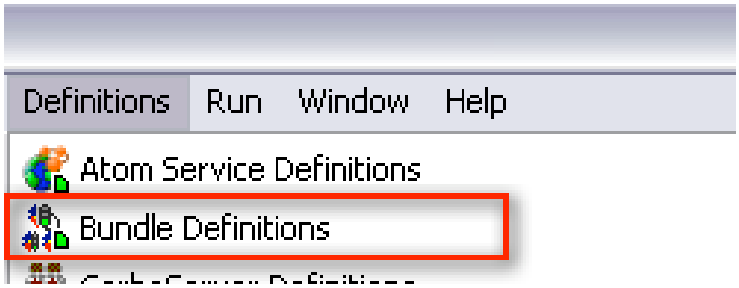
15. Open the **CICS SM** Perspective.

16. Click on the **CICSPlex Repositories** view, expand your CICS repository and select your CICS region

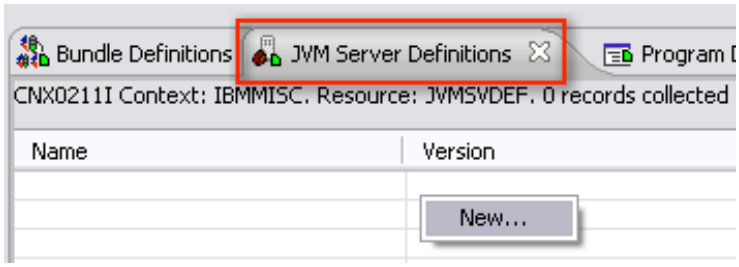


17. Since CICS requires a definition for each resource, it is mandatory to define a JVM Server, an OSGi bundle a program and a transaction.
Go to **Definitions** → **Bundle Definitions** / **JVM Server Definitions** /

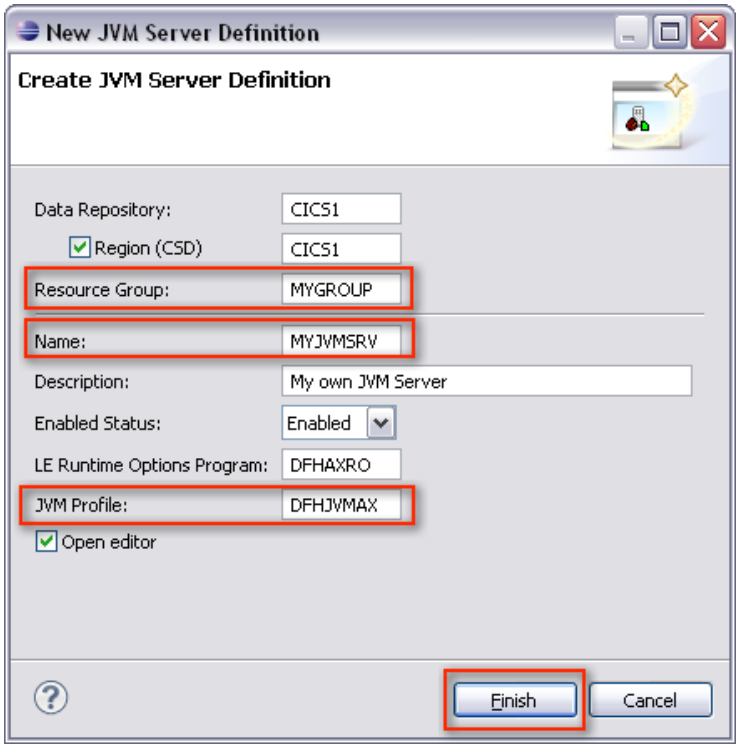
Program Definitions / **Transaction Definitions** to open all views needed for the definitions.



18. Activate the **JVM Server Definitions** view and perform a right-click → **New...** to create a new JVM Server definition. Skip this step if you want to use an existing JVM Server definition.



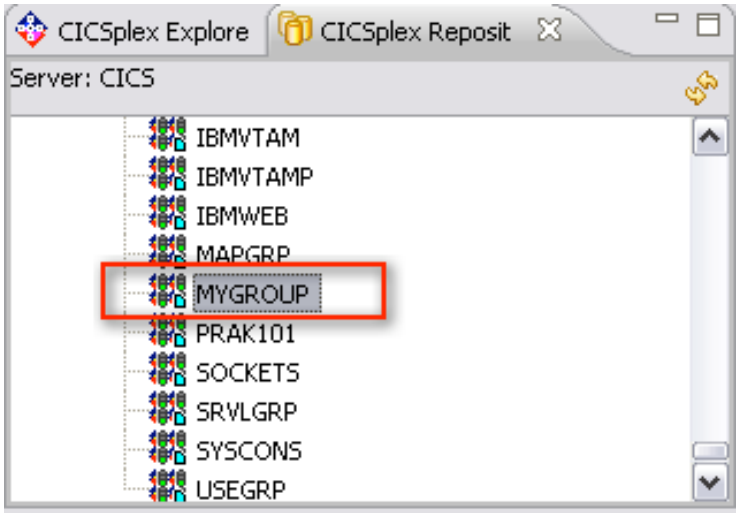
19. Assign the group (MYGROUP), define a unique name (MYJVMSRV) and assign the JVM profile (DFHJVMAX) to the JVM Server and click **Finish**.



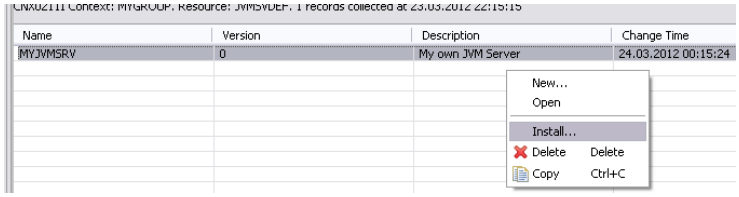
Description of options:

- **Resource Group:** Resources in CICS are grouped. Therefore, one needs to assign a group to the JVM Server. If the entered group is not defined, it will be created automatically.
- **Name:** The name is a unique identifier for the JVM Server. An OSGi bundle requires to be assigned to the JVM Server (refer step 12).
- **LE Runtime Option Program:** Assigns the program that defines the options for Language Environment. By default set to DHFAXRO. Do not alter this field.
- **JVM Profile:** The profile defines necessary parameters such as the working directory or optional startup parameters for the JVM. By default profiles are located in the USS file system directory:
  `/usr/lpp/cicsts/cicsts42/JVMProfiles`

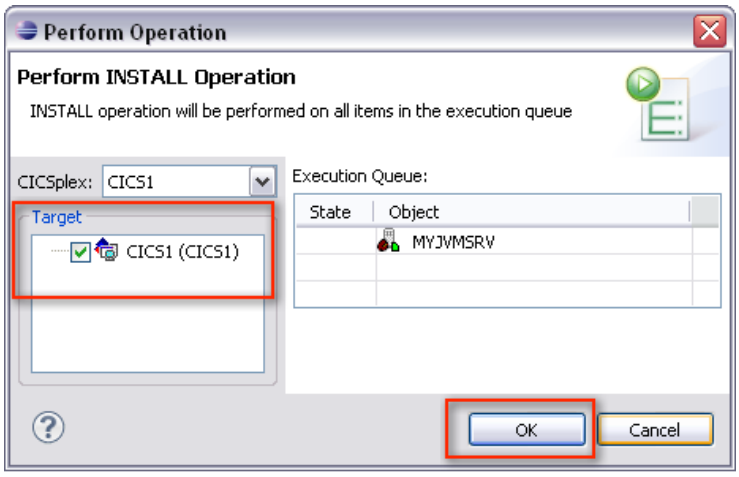
To get a user-friendlier look on your definitions, find your resource group and click in it. With it being activated, only your personal definitions will be displayed.



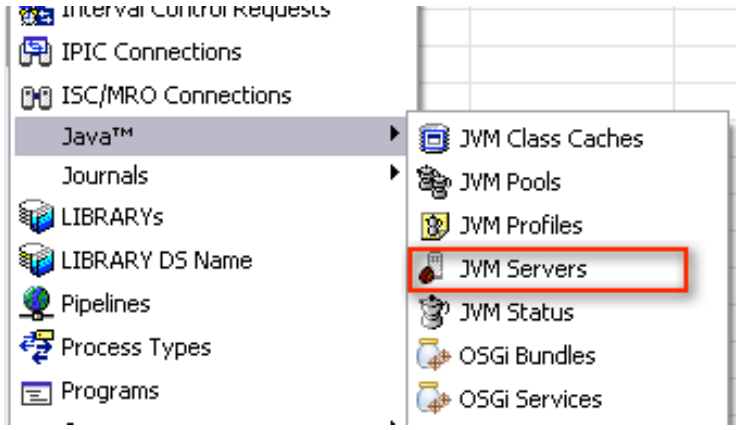
20. Perform a right-click → **Install...** on the new created JVM Server definition.



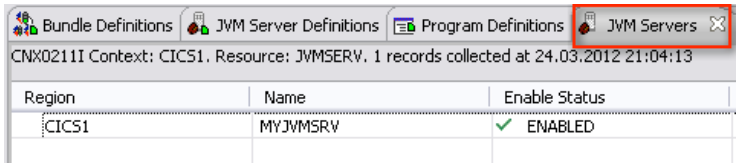
21. Choose the CICS region for the installation and click **OK**.



22. Go to **Operations** → **Java** → **JVM Servers** to open the **JVM Server** view.

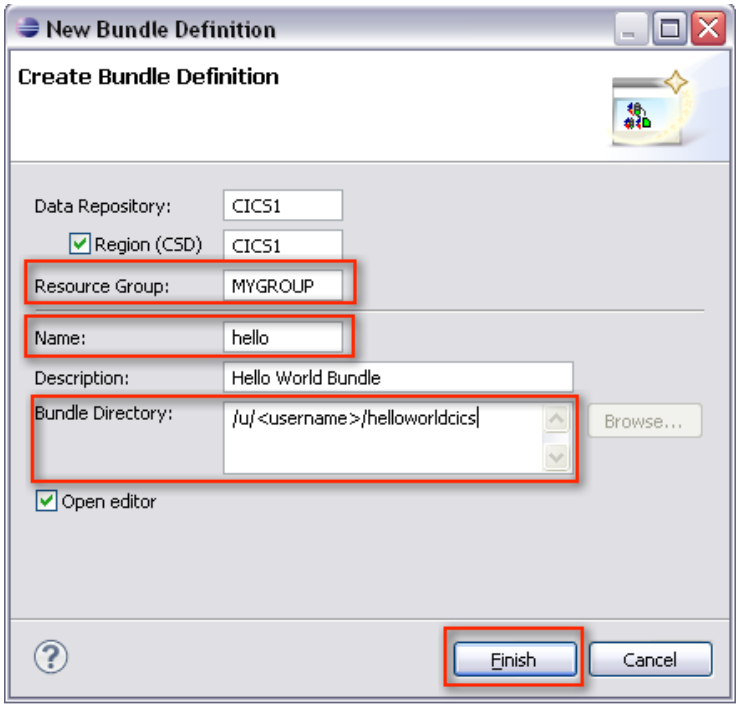


23. Activate the **JVM Servers** view, refresh the view and verify if the JVM Server installation was successful (Enable Status = ENABLED).



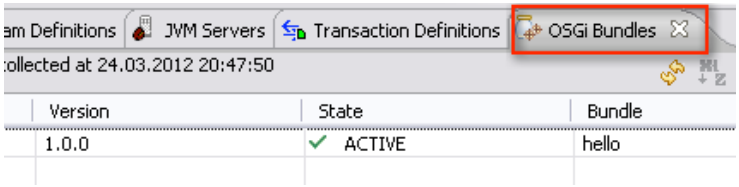
24. Activate the **Bundle Definitions** view and perform a right-click → **New...** to create a new bundle definition.

25. Assign the group (MYGROUP), define a unique name (hello) and the directory of the bundle chosen in step 14 and click on **Finish**.



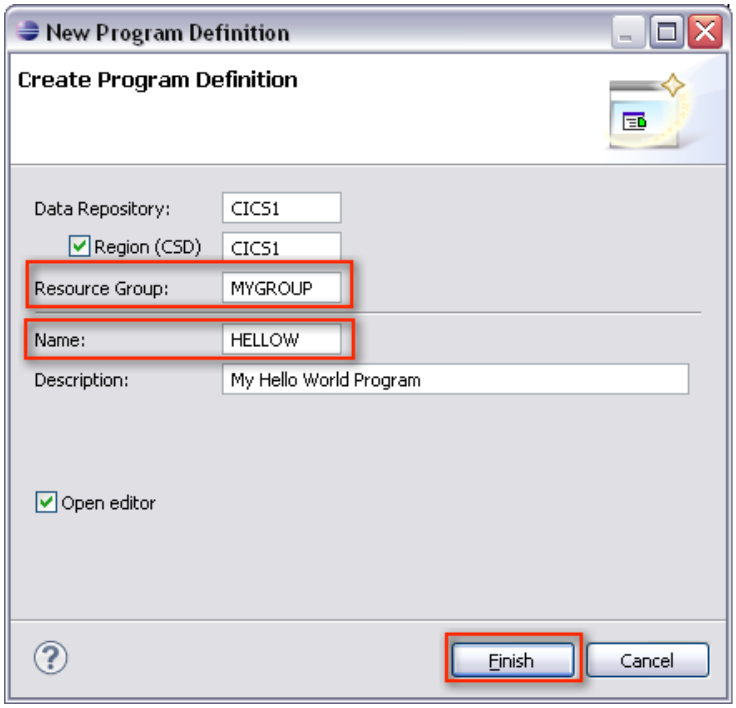
26. Perform a right-click → **Install...** on the new created OSGi bundle definition. Choose the CICS region for the installation and click **OK**.

27. Go to **Operations → Java → OSGi Bundles** to open the **OSGi Bundles** view. Activate the **OSGi Bundles** view, refresh the view and verify if the OSGi bundle installation was successful (State = Active).



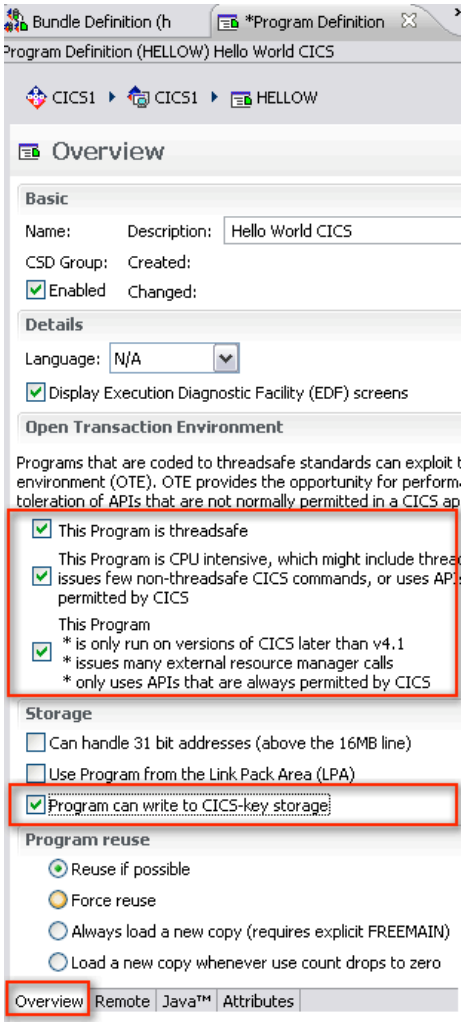
28. Activate the **Program Definitions** view and perform a right-click → **New...** to create a new program definition.

29. Assign the group (MYGROUP), define a unique name (HELLOW) to and click on **Finish**. A new view opens that enables to specify parameters for the program.

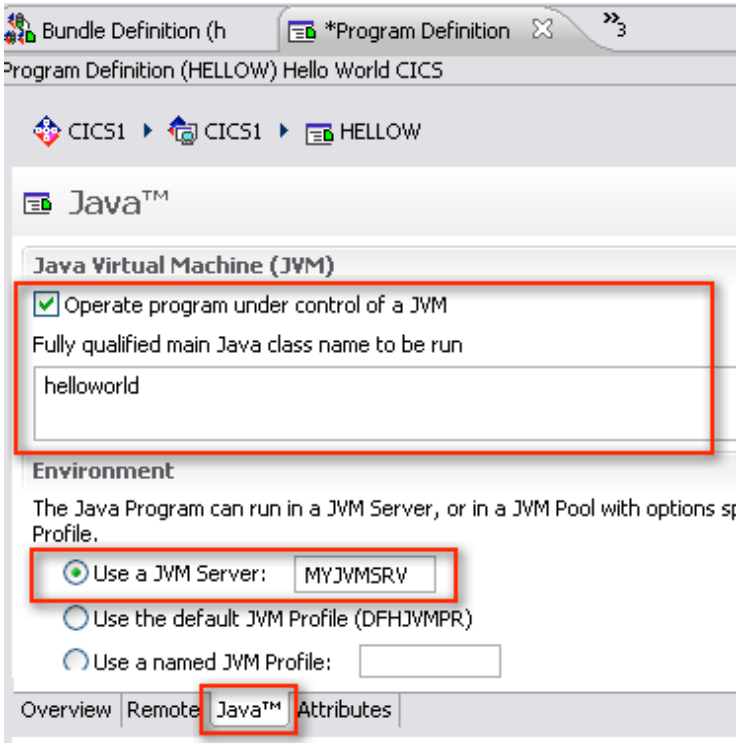


30. Activate the options for

- This program is threadsafe
  → Since the JVM Server is a multi application environment, all application are required to be threadsafe.
- This program is CPU intensive...
  → Enables higher priority scheduling.
- This program only runs on versions of CICS later that 4.1...
  → Since the JVM Server is introduced for CICS version 4.2, this option is mandatory.
- Program can write to CICS-key storage
  → Since the JVM Server uses a T8 TCB and since the storage protection key 8 is assigned to CICS, the program needs to be able to write to the CICS-key storage.



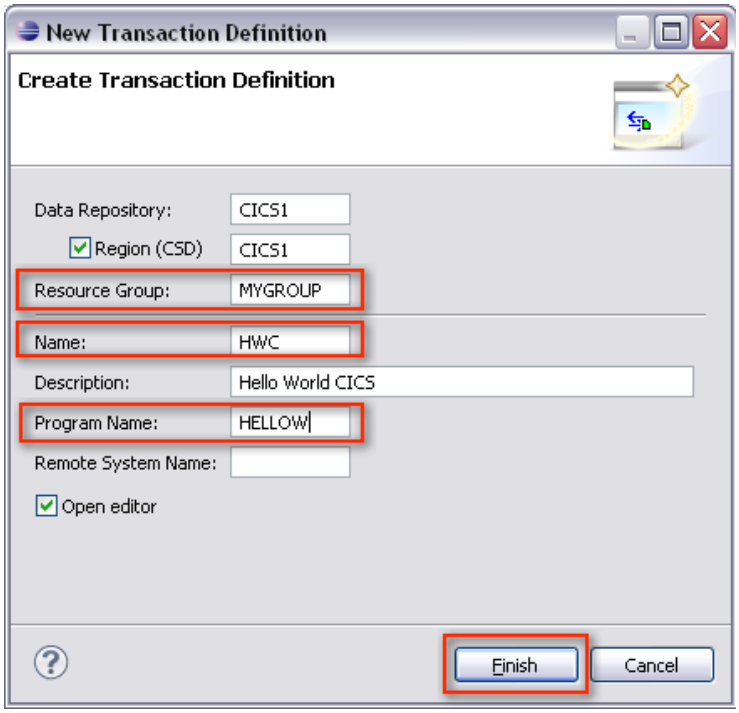
31. Activate the **Java** tab, check the option **Operate program under...**, insert the alias defined in step 7 (or the string `helloworldpackage.HelloWorld`), assign the JVM Server to the program and press the key combination **CTRL**+**S** on your keyboard to save your changes.



32. Activate the **Program Definitions** view and perform a right-click → **Install...** on the new created program definition. Choose the CICS region for the installation and click **OK**.

33. Activate the **Transaction Definitions** view and perform a right-click → **New...** to create a new transaction definition.

34. Assign the group (MYGROUP), define a unique name (HWC), assign the previously defined program (HELLOW) and click **Finish**.



35. Activate the **Transaction Definitions** view and perform a right-click → **Install...** on the new created program definition. Choose the CICS region for the installation and click **OK**.

36. Connect to CICS using a *3270* emulator and execute the previously installed transaction. The output should show the text

```
HWC transaction started
Hello World from CICS.
```