From zero to zHero:
zOA - z optimized applications
z/VSE - COBOL

Ingo Franzki
Benjamin Storz
Wilhelm Mild
zvse@de.ibm.com

Table of contents

Disclaimer

This document was created as an instruction guide for the “From zero zo zHero: zOA - z optimized applications” workshop in Germany and is no official IBM reference guide.

1 Lab Overview

In this lab you will use WebSphere Developer for zSeries to develop COBOL Batch and Online applications for z/VSE.
You will perform the following steps for a batch and a CICS online z/VSE application:

- Establish a connection to z/VSE
- Create a z/VSE project
- Import sources from z/VSE into the project
- Edit / syntax check source
- Remote compile and link
- Run the application

2 Introduction and general hints

This lab guide is intended as an introduction for people to become familiar with application development on z/VSE.
You should not need any COBOL skills to run through these labs, but basic knowledge of z/VSE and Mainframe technologies are required to understand them.

Here are some general hints that you should read before you begin with this lab:

- This lab guide is also available as PDF-document on your workstation under C:\VSEWorkshop\VSEBatchCICS.pdf.
- When you copy content from the lab guide and paste it into development tools, always be aware that some line breaks in this document might cause failures. In that case, remove the line break.
- Here are some general hints for z/VSE:
  - When you are requested to press <enter>, please press the right <CTRL> key when you are using Personal Communication (3270 Emulator)!
  - You can only enter data in special screen areas. Use the <TAB> key to go to the next typo field
  - If you try to enter data in a non-typo area, your keyboard will be locked (see red sign, last line, left side **ß J à**)

For this lab we will use a VMWare Image with the following software installed:

- WebSphere Developer for zSeries (WDz) V6.0.1.6
- z/VSE Plugin from QGroup V1.0
- CICS Transaction Server for Windows V5 (CICS TS)
- IBM Personal Communication (PCOM)

3 System information and settings

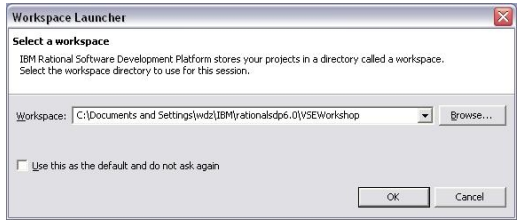
	Value	Comment
VSE userid	TE##	
VSE password	team##	
Hostname/IP	192.168.0.2	
Ports	23 2893	Telnet 3270 VSE Connector Server
Workshop files	c:\VSEWorkshop	
Location of VSE source	PRIMARY.TE##	
Phase name batch Location of batch phase Phase name CICS Location of CICS phase	STARTAPP PRIMARY.TE## TIME## PRIMARY.WKSHOP	 Is included in the CICS Libdef

4 Getting started

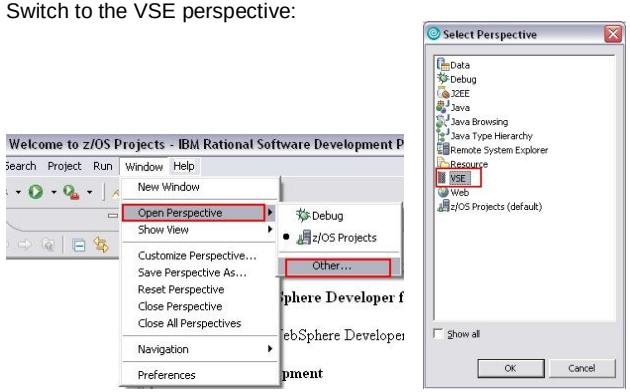
This chapter explains how to launch WDz with the appropriate workspace.

1.) Launch WDz by double-clicking on the following icon on the desktop

2.) Modify the workspace location as follows:
C:\Documents and Settings\wdz\IBM\rationalsdp6.0\VSEWorkshop



3.) Switch to the VSE perspective:

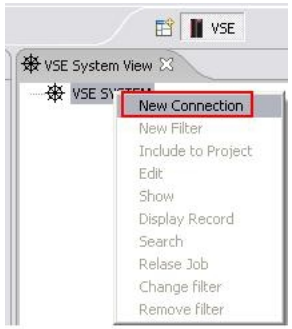


Press OK.

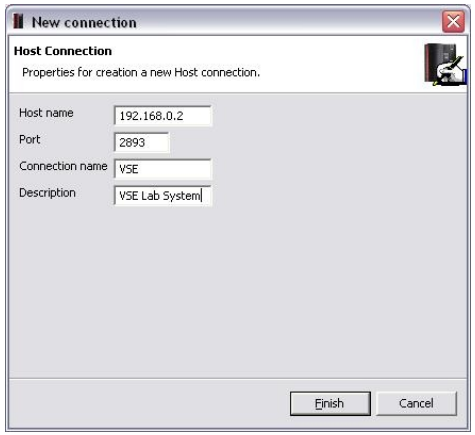
4.) Close the welcome window:

5 Establish a z/VSE connection

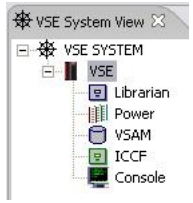
5.) Right-click in the **VSE System View** and press **New Connection**.



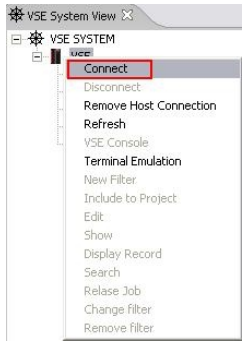
6.) Enter the following values and press finish:



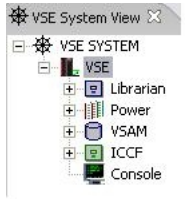
7.) Your connection should look like this:



8.) Right-click on your VSE System and choose **connect**. Logon with your userid (**TE##**) and your password (**team##**).

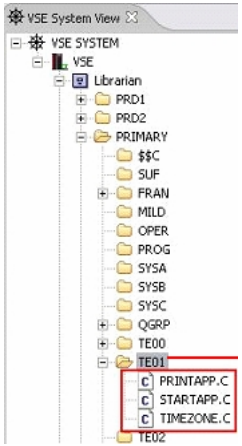


After a successful connection you should see the following:



You should notice the changed icon and the plus signs in front of the subsystems when the connection was successful.

9.) Navigate to your programs by clicking on the + before Librarian, then open the library **PRIMARY** and your sub-library **TE##**

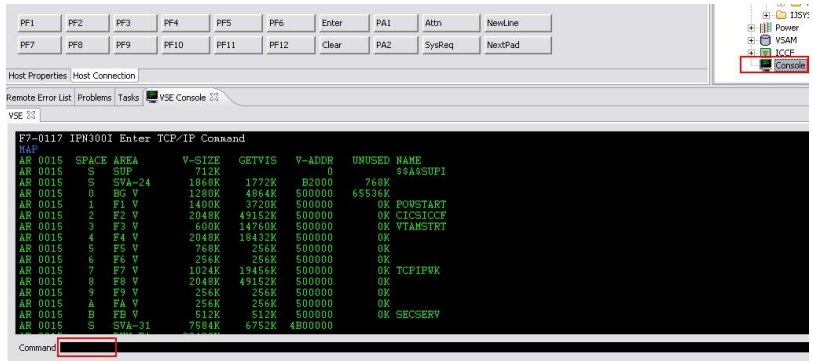


Use your team number!

These 3 members will be used later in the workshop.

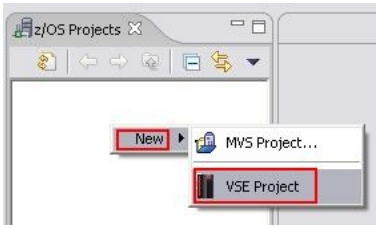
6 Using the VSE Console

10.) To open the VSE console double-click the **console** icon under your VSE connection. You can issue VSE commands in the **Command** line. Press enter to activate your command.

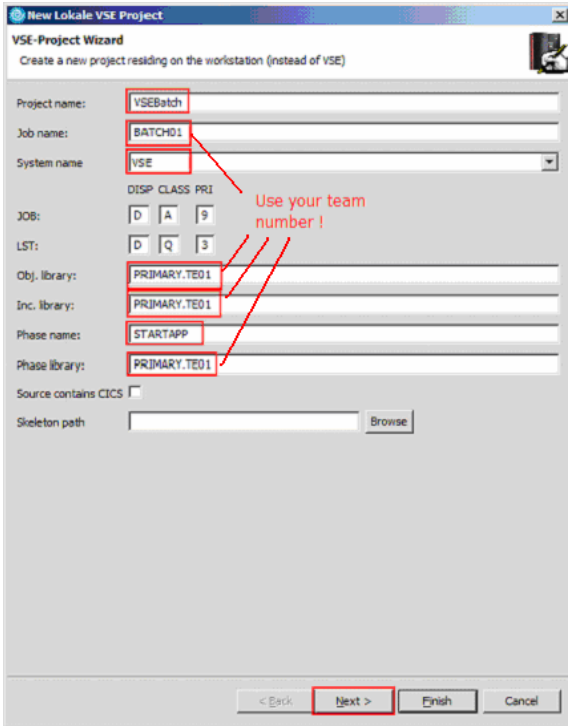


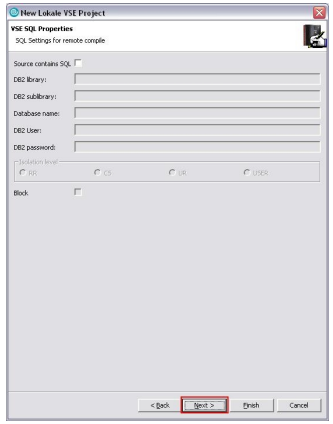
7 Create a VSE project for batch

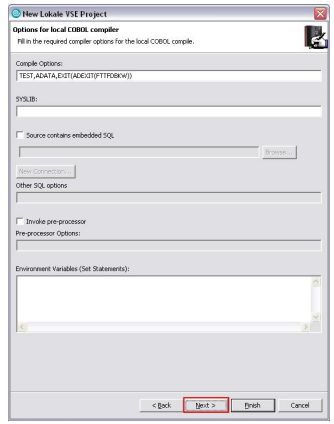
11.) Right-click in the project view and select **New -> VSE Project**.

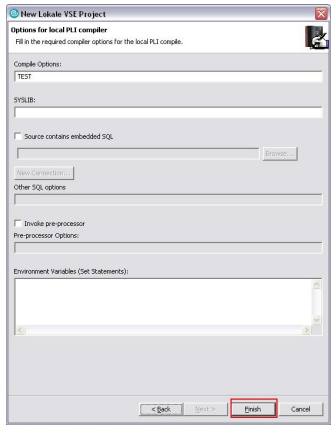


12.) Enter the following values. Choose the VSE system using the drop down box and choose as Job name **BATCH##** and enter Obj. library, Inc. library and Phase library: **PRIMARY.TE##**. Phase name is **STARTAPP**.

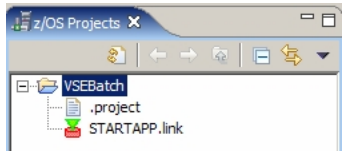






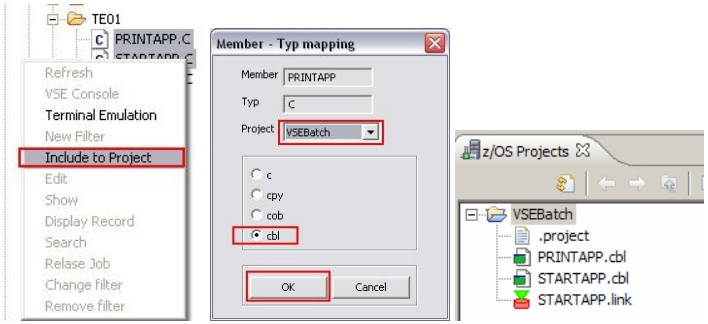


Press **Finish**. You should have the following:

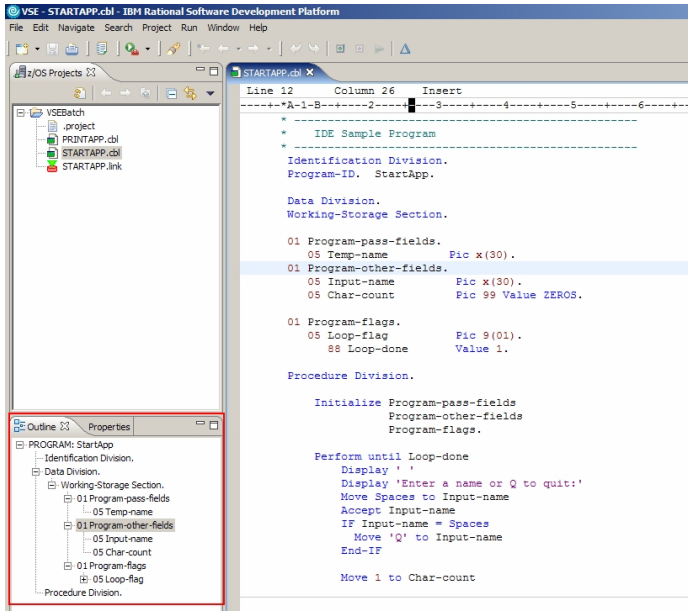


8 Edit, Compile and Link batch program

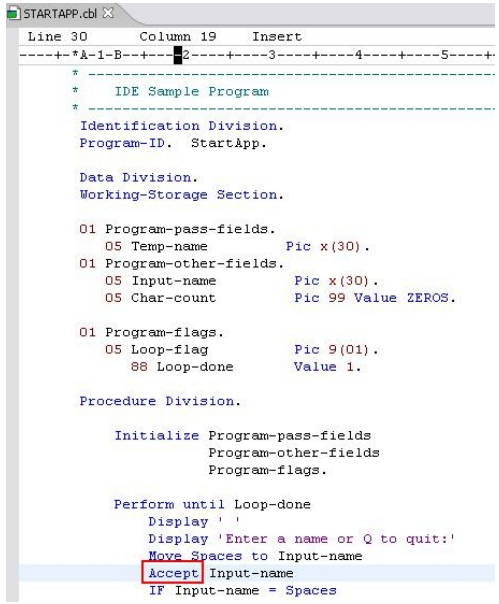
13.) Right-click on the two sources **PRINTAPP.C** and **STARTAPP.C** in your sublibrary **TE##** and choose **Include to Project**.



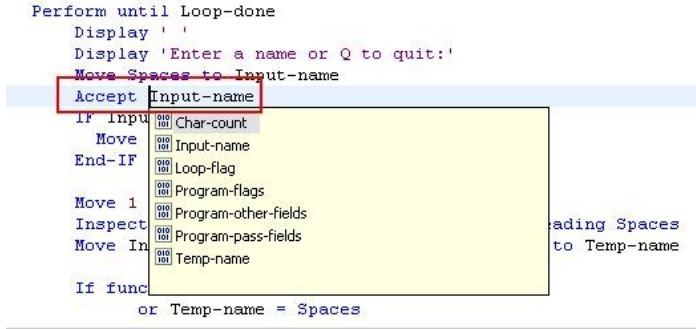
14.) Double-click on STARTAPP.cbl in the project view to open the Live Parsing Extensible Editor (LPEX). You can navigate in the outline view; the position in the outline view will be synchronized with the position of the curser in the editor.



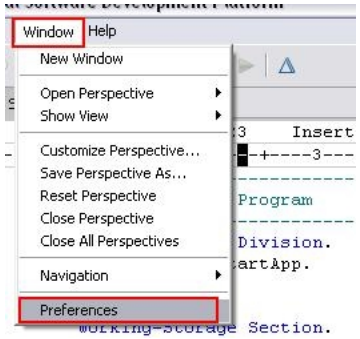
15.) You can open the COBOL language help by selecting a COBOL keyword and pressing **F1**. The opening of the help may take some time, but it is recommended to minimize the help window. The next time you use it, will bring it up much faster.



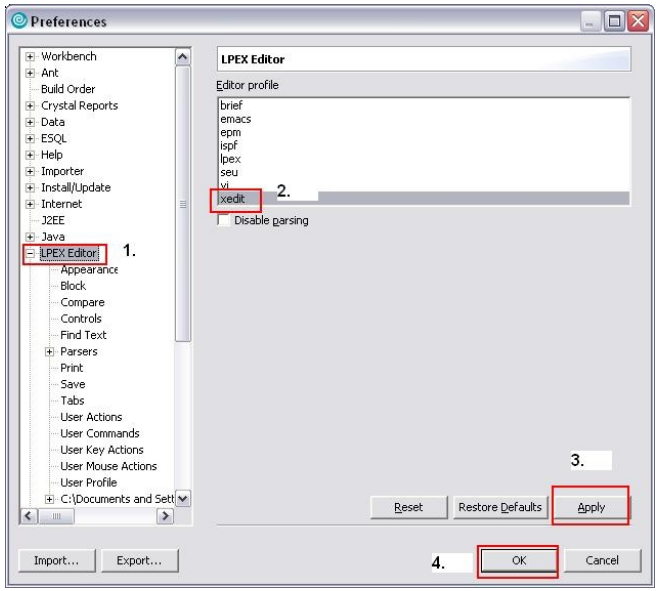
16.) Move the cursor right before the “Input-name” and press **STRG + space**. This will show you the content assist functionality which shows all possible keywords or variables defined in your working storage section.



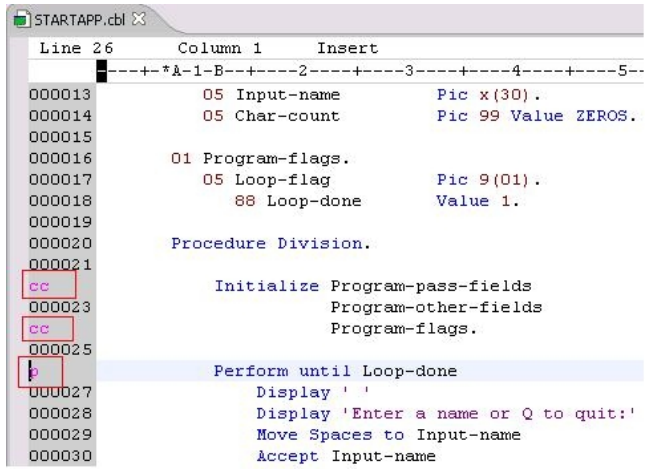
17.) We will now change the profile of the editor to behave like XEDIT on z/VM. Select **Window -> Preferences**.



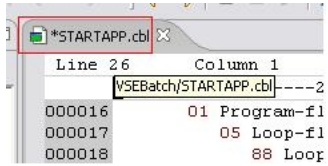
Select **LPEX Editor** then choose **xedit**, press **Apply** and then leave by pressing **OK**.



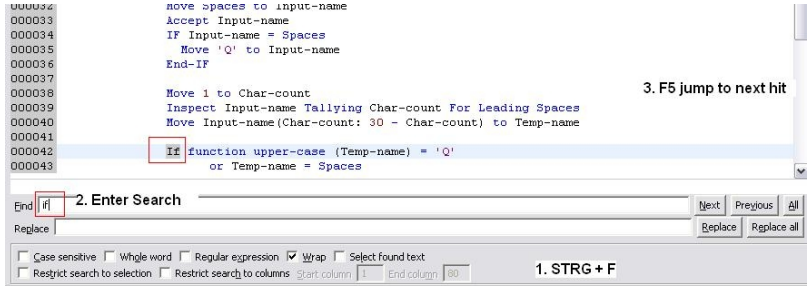
18.) You will see that the editor now shows line numbering on the left; this is prefix area where XEDIT commands can be issued. Try out the following copy command (**CC**). **P** means inserting previously and **F** would insert following, after the character.



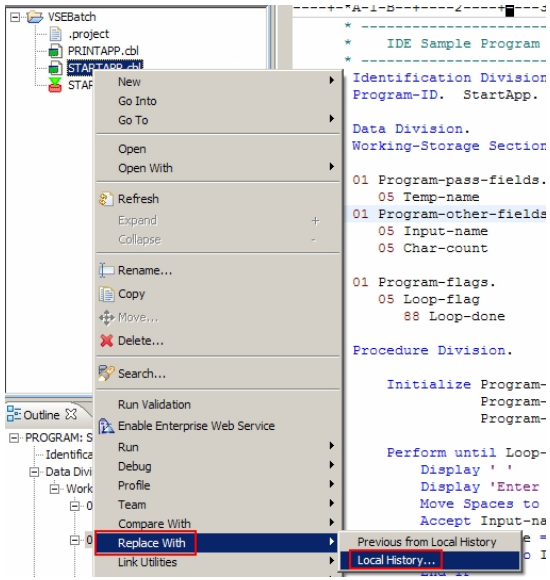
After your change the source has been modified, you will see the asterisk in front of your program name. Save the program with **STRG + S** or using the menu.



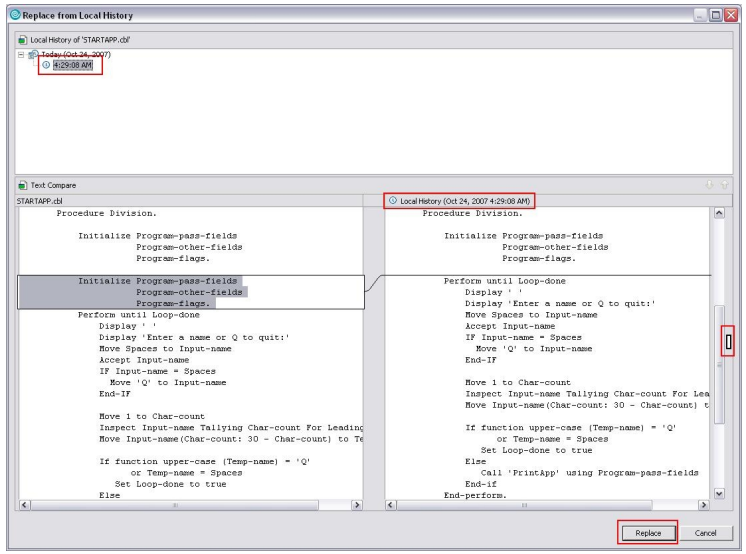
19.) To search for expressions in your source code press **STRG** + **F**. **F5** brings you to next match. With **ESC** the search window will disappear.



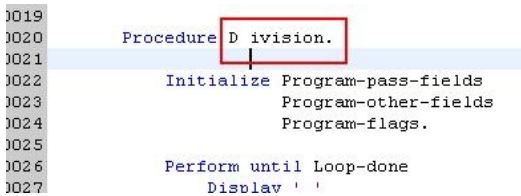
20.) You can go back to any previously saved version of your program. Select your program STARTAPP.cbl and right-click then choose **Replace With -> Local History**.



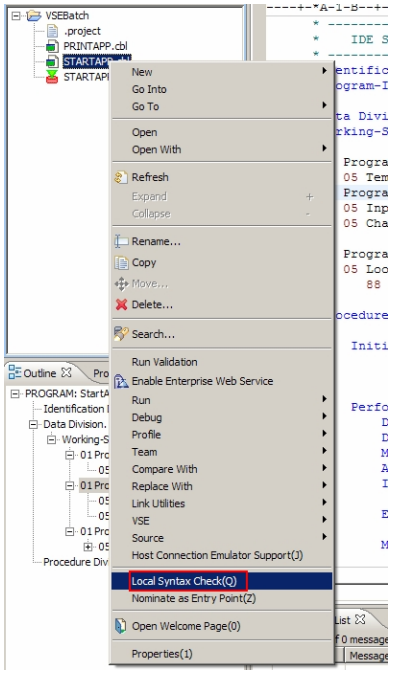
The compare dialog shows you visually the differences. Note the white bar on the right side.

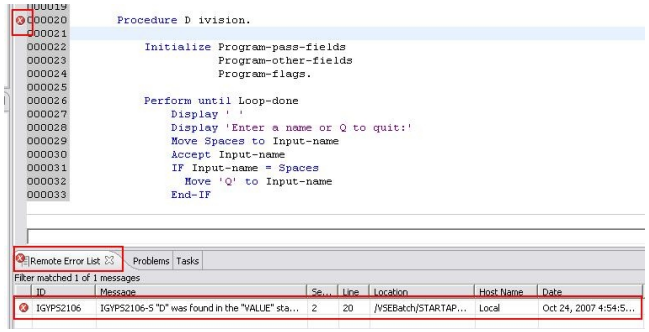


21.) We will now bring an error in the application. E.g. move a space in Division keyword, **don't forget to save**!



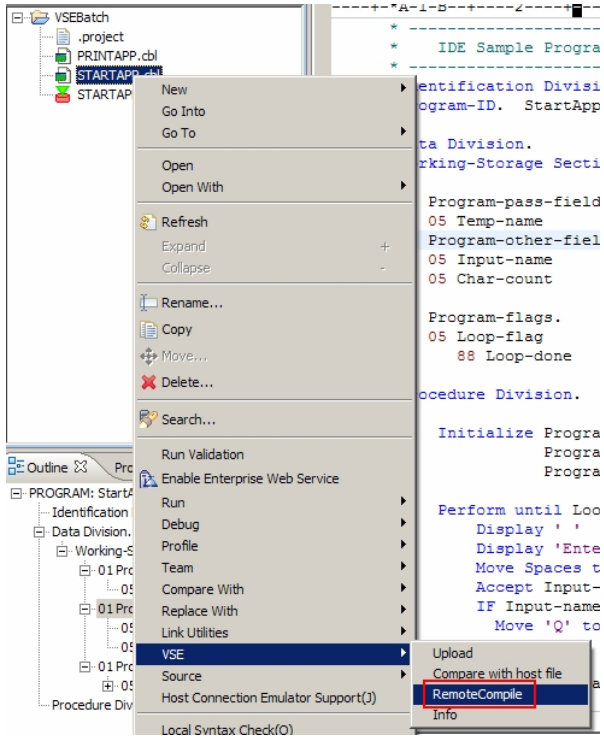
22.) To verify the syntax of your COBOL program right-click it and select **Local Syntax Check**. This will invoke the local COBOL compiler and bring you syntax errors in the **Remote Error List**.



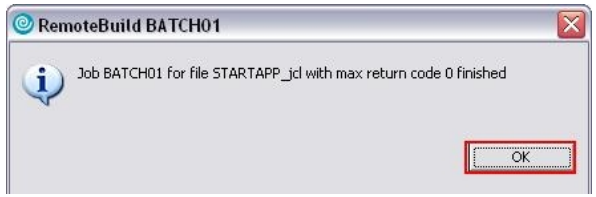


The **Remote Error List** shows local and remote errors. Double-click on the error brings you to the location in the editor where the error occurs. Notice the red bullet in the editor.

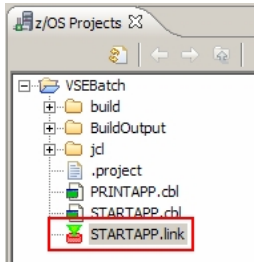
23.) Remote compile STARTAPP.cbl and PRINTAPP.cbl by right-clicking on each source file and the choosing **VSE -> Remote Compile**.

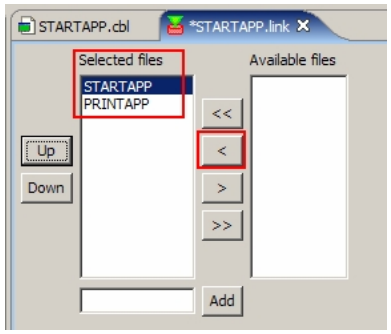


You should get a clean compile with return code 0. Otherwise fix the errors until you get 0.



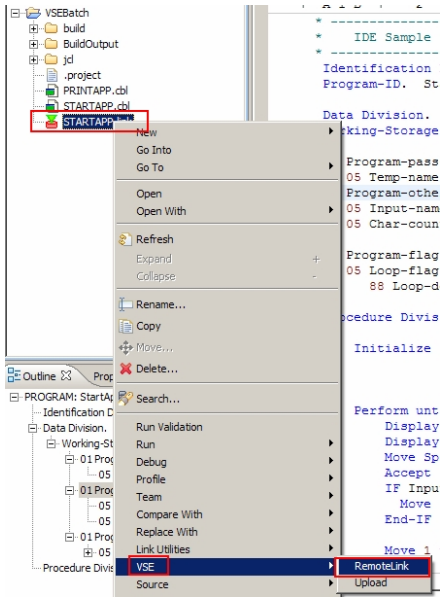
24.) To link your application, double-click on the **STARTAPP.link**.
Move the “available files” to the “selected files” using the left arrow button.
The sequence is important, the first should be STARTAPP.





Save the link book.

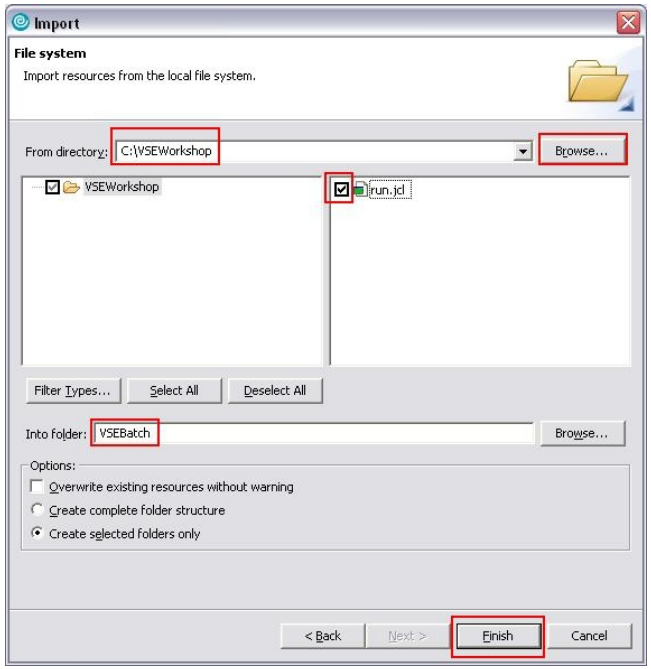
25.) To perform the link right-click on the **STARTAPP.link** and select **VSE -> RemoteLink**. Return code 2 is OK.



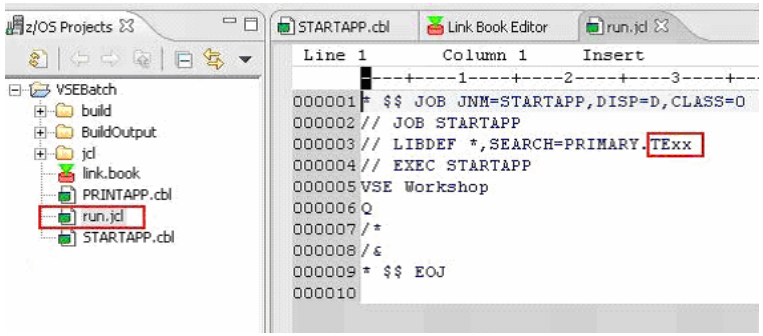
9 Prepare JCL and run batch program

26.) We will import a JCL template to run the STARTAPP batch application.
Choose **File -> Import** and select **File system**.

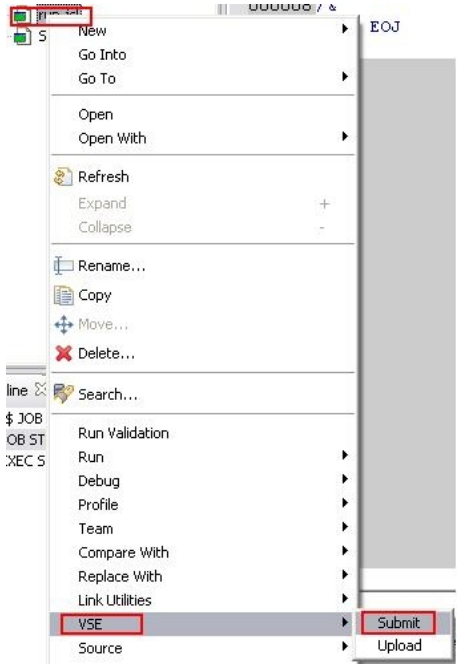
Navigate to C:\VSEWorkshop and select **run.jcl.** Make sure the import folder is **VSEBatch**.

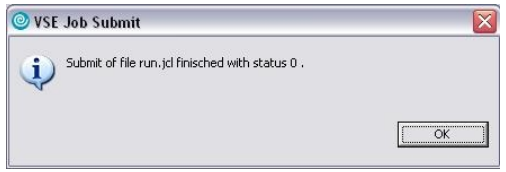


27.) Open your run.jcl and modify the libdef to your team specifications. (**TE##).** Save.

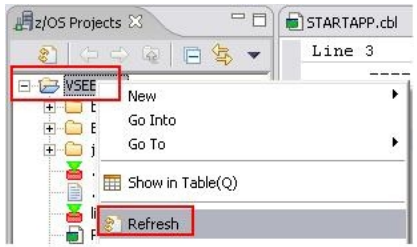


28.) To submit your job right-click on **run.jcl** and select **VSE -> Submit**. Check for the return code.

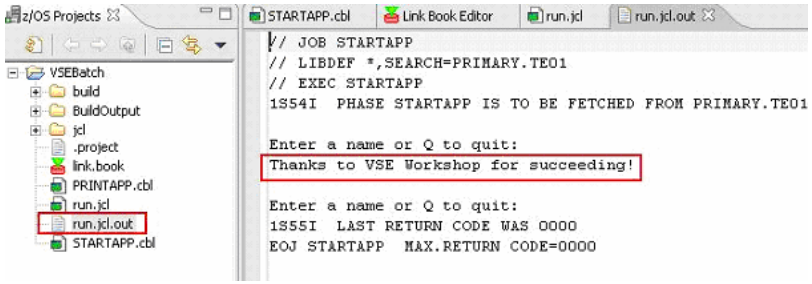




To look at the job output do a Refresh on the project.



Open **run.jcl.out** and check the output of your application.

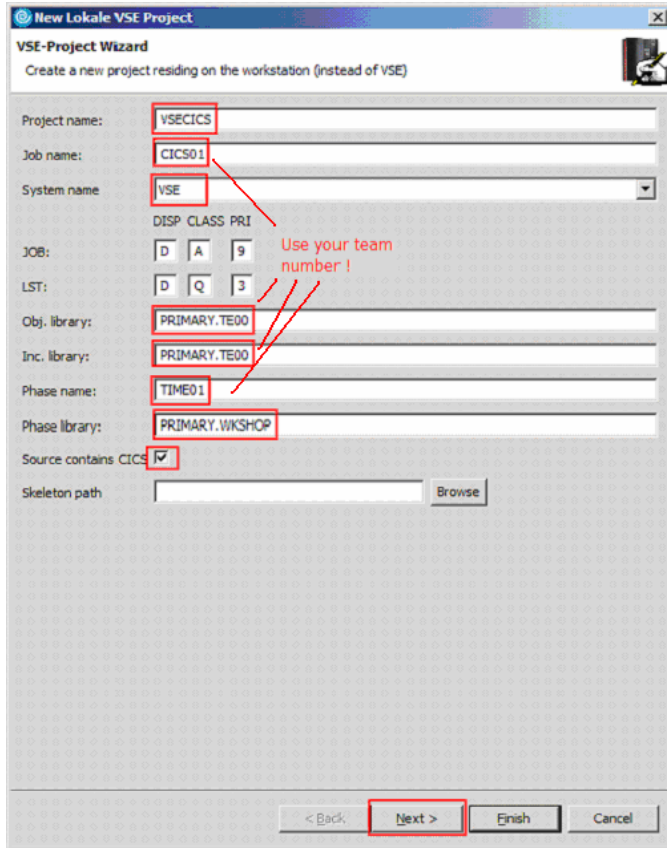


Congratulations you have successfully finished the COBOL batch lab!!!

We will now continue with CICS …

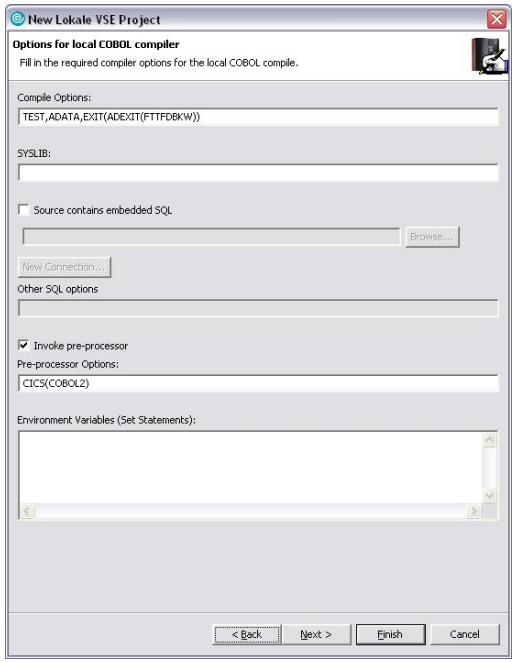
10 Create a VSE project for CICS

29.) Create a new VSE project (same as step 10.)) **VSECICS**. In the VSE-Project Wizard use the following values. For Job name use **CICS##**, for Sublibraries use **TE##**, Phase name use **TIME##** and the Phase sublibrary is the same for every group (**WKSHOP**). Remember to select the **Source contains CICS** checkbox.



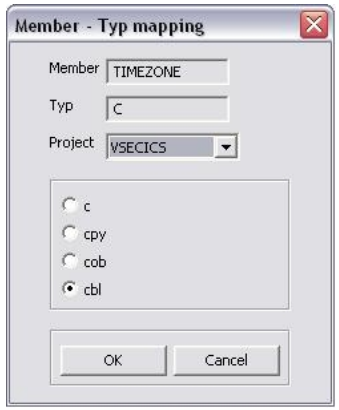
Skip the panel for SQL properties. On the panel for the local COBOL compiler options, enable the project for local syntax check with CICS select

the **Invoke pre-processor** check box. Then press **Finish**.



11 Edit, Compile and Link CICS Program

30.) Include the source TIMEZONE.C from your sublibrary (**TE##**) into your CICS project.

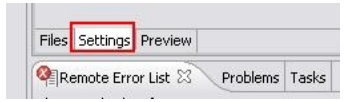


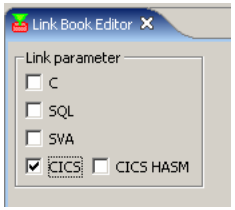
31.) You can use again the LPEX Editor for modifying your source.

32.) Perform a local syntax check to invoke the local COBOL compiler and CICS pre-processor.

33.) Perform a remote compile on your TIMEZONE.cbl source within your VSE project. Verify that you have a clean compile.

34.) To link your application open the **TIME##.link** and move the TIMEZONE to the **selected files** side. Open the **settings** tab and make sure that CICS is selected (but not CICS HASM).





35.) Perform a remote link. Return code 2 is OK.

12 Run the CICS Application

The following CICS resource definitions **have already been done** for you, they are listed for your reference:

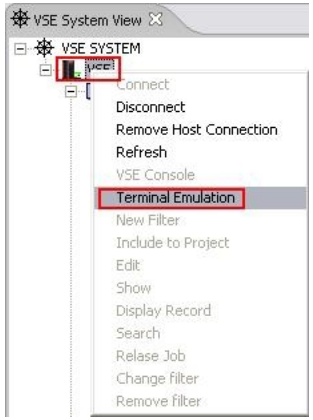
36.) CEDA DEFINE PROGRAM(TIME##)
Language=COBOL
Group=VSESPG

37.) CEDA DEFINE TRANS(TZ##)
Program=TIME##
Group=VSESPG

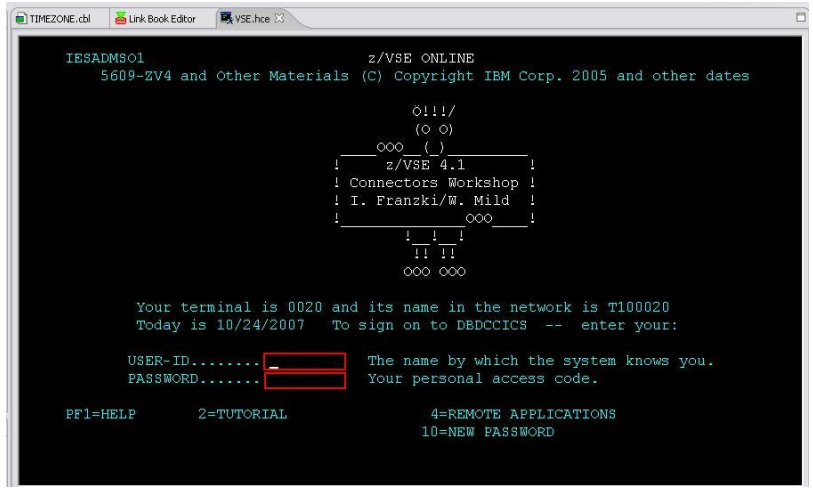
38.) INSTALL GROUP(VSESPG)

39.) Include PRIMARY.WKSHOP in the CICS Libdef.

40.) Open the integrated Terminal Emulation. Right click on your VSE system in the VSE System view and select “Terminal Emulation”.

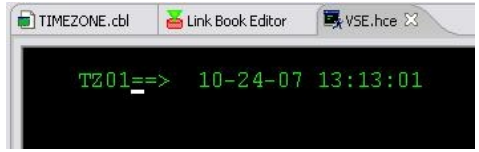


41.) Enter your VSE userid (**TE##**) and password (**team##**) and press **enter**.



42.) In the next screen press **F6** to go to the CICS screen.

43.) Enter **TZ##** to start the CICS Transaction that invokes your program **TIME##**.



Use your team number!

Congratulations you have successfully finished the COBOL CICS lab!!!