



# COBEE

## COBEE Quick Start

*version 3.5*

Revision: 018  
Date: 10/20/2014

## Reference

Date	Description	Writer
	COBEE User Guide	EasiRun Europa GmbH
1st Edition, 17 September 2010	OpenCOBOL-1.1-06FEB2009-Programmers-Guide	Gary Cutler

## Copyright

COBEE – The Open Source COBOL Development Environment

Granted License 2014 EasiRun Europa (<http://easirun.com>)

Cobos Project – The Open Source COBOL Development Environment

Copyright © 2009-2014 METRIXWARE (<http://cobos.metrixware.com>)

All rights reserved.

You may not use this file except in compliance with the METRIXWARE Community License Agreement (the "MCLA").

You may obtain a copy of MCLA [here](#).

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	What's new . . . . .	4
1.2	Prerequisites . . . . .	5
<b>2</b>	<b>Installation</b>	<b>6</b>
<b>3</b>	<b>Editing a COBOL program</b>	<b>15</b>
<b>4</b>	<b>Checking the syntax of a COBOL program</b>	<b>20</b>
4.1	CUSTLOAD program . . . . .	20
4.2	CF00P program . . . . .	21
<b>5</b>	<b>Unfolding a COBOL program</b>	<b>23</b>
5.1	CF00P program . . . . .	23
5.2	GESCOM program . . . . .	24
<b>6</b>	<b>Using 3270 Emulator inside eclipse</b>	<b>28</b>
<b>7</b>	<b>Configure FTP access</b>	<b>30</b>
<b>8</b>	<b>Compiling a local source on mainframe</b>	<b>32</b>
8.1	Saving a PDS member . . . . .	35
8.2	Opening a PDS member . . . . .	36
<b>9</b>	<b>Managing a job</b>	<b>37</b>
9.1	Submitting a jcl . . . . .	37
9.2	Retrieving a sysout . . . . .	38
<b>10</b>	<b>Local Compilation</b>	<b>40</b>
10.1	COLORS program . . . . .	40
10.2	CUSTLOAD program . . . . .	41

# 1. Introduction

The purpose of this guide is to explain how to install, configure and use the Eclipse plug-ins COBEE.

You will edit and compile both local COBOL programs and mainframe COBOL programs WITHOUT INSTALLATION OF MAINFRAME RESOURCES.

## 1.1 What's new

### Improvements in COBEE 3.5:

- COBEE 3.5 is fully qualified with the very latest Eclipse version such as Kepler (4.3) and Luna (4.4).
  - Support of listings coming from the remote Micro Focus® compiler:  
the COBOL programs and copybooks are marked with the compilation messages in the Problems view.
  - Preprocessing capabilities:
    - Custom processing can be called before the GNU Cobol Check Syntax and Unfolding.
    - A standard post processing is proposed so that the messages are marked in the right place in the source code.
- (a support of custom macro instructions has been implemented and distributed as a custom plug-in)

### This release solves the following bugs:

**5320** FTP Access supports use of non standard TCP/IP port

## 1.2 Prerequisites

Ensure that the workstation has at least 2GB of RAM.

Supported OS: Windows XP SP3, Windows 7.

Ensure that a Java JRE 6 or 7 is present on the workstation.(JRE 8 is not yet fully qualified.)

This COBEE 3.5 Release must be installed on Eclipse Helios 3.6.2, Indigo 3.7.2, Kepler 4.3.x or Luna 4.4.x (32bits or 64bits).

**FTP Access module requires installation of a REXX interpreter on the workstation** such as [Regina REXX Interpreter](#) (version 3.6 or 3.8.2 recommended)<sup>1</sup>.

Resources:

You downloaded **COBEE\_3.5.x\_Release-demo.zip** from the [COBEE site](#).

Once you have unzipped the downloaded file, you've got the following directories and files:

- **Demo\_workspace** which contains preconfigured resources for this "quickstart"
- **Products** which contains the plug-ins
- **COBEE\_V3.5\_DemoReadme.txt**
- **COBEE\_V3.5\_Quickstart.pdf**: this file!
- **COBEE\_V3.5\_User\_Guide.pdf**
- **releaseNotes.txt**

---

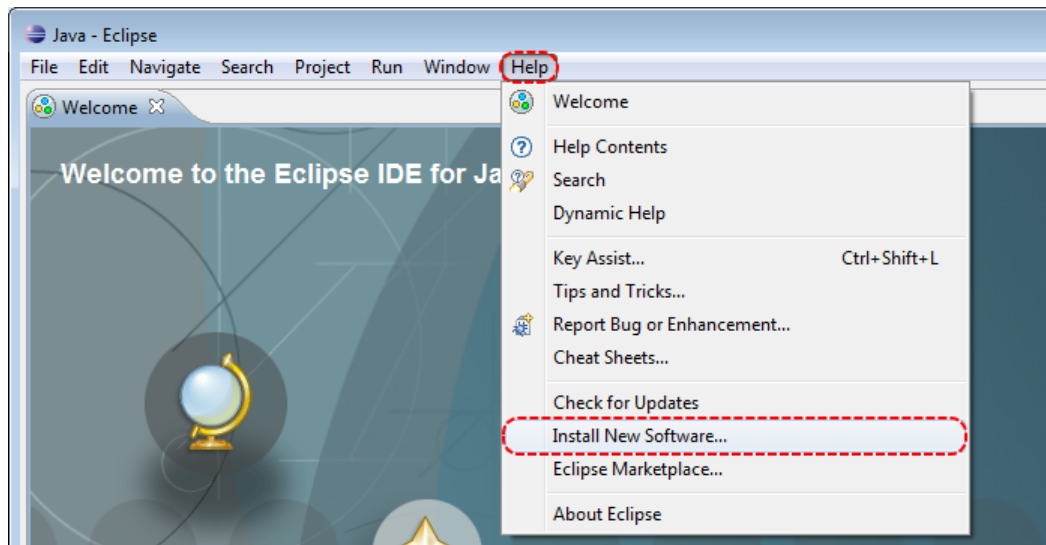
<sup>1</sup> For the users of Open Object Rexx (ooRexx), some scripts of COBEE are not fully compatible.

## 2. Installation

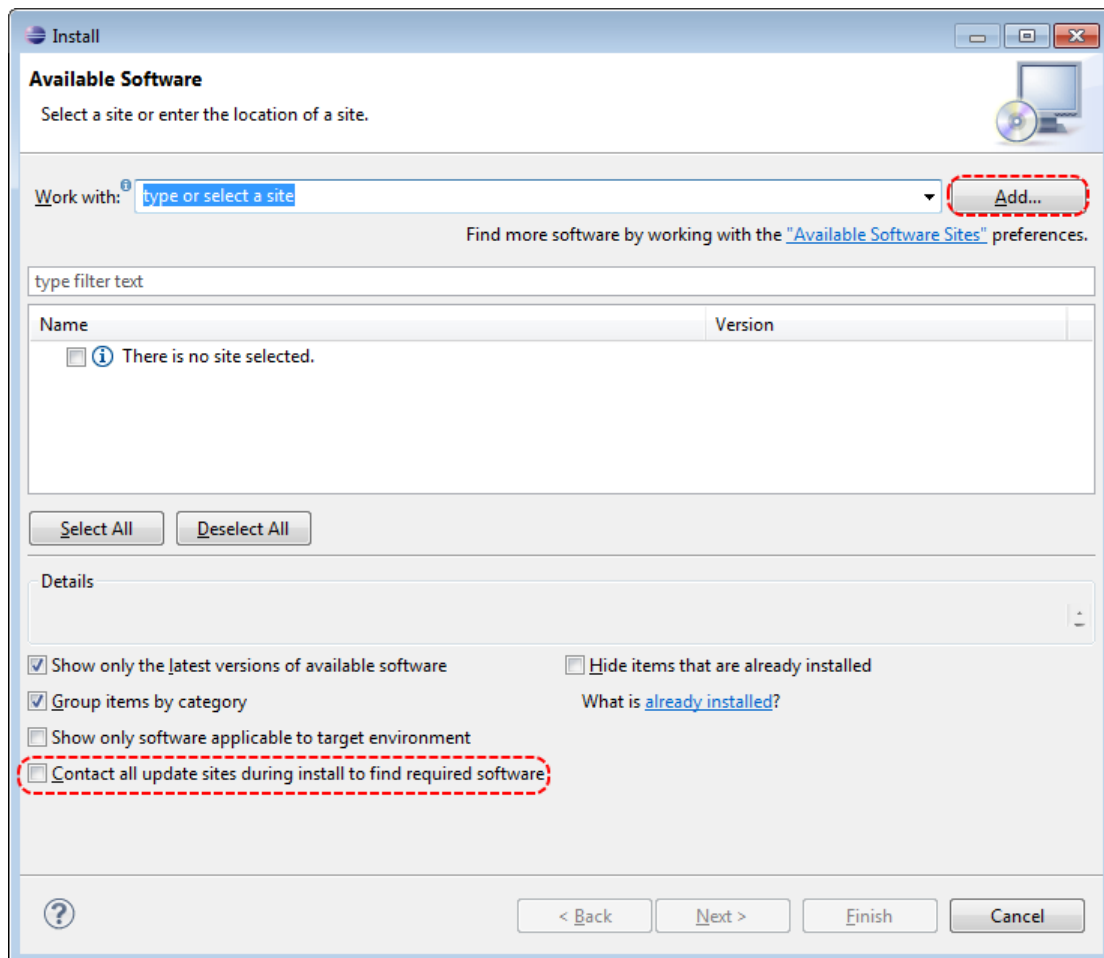
Retrieve Eclipse for Windows from <http://www.eclipse.org/downloads>.

Note: This document has been produced with Eclipse Indigo. The examples also work with Eclipse versions listed in the prerequisites.

- 1 Launch Eclipse and select a new workspace (temporary used for COBEE plug-in installation).
- 2 Installation of plug-ins: Select in the menu **"Help ► Install New Software..."**



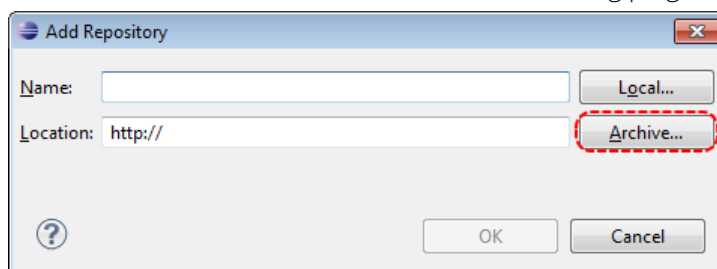
- 3 Add local plug-in archives: Click the **"Add..."** button.



Tip:

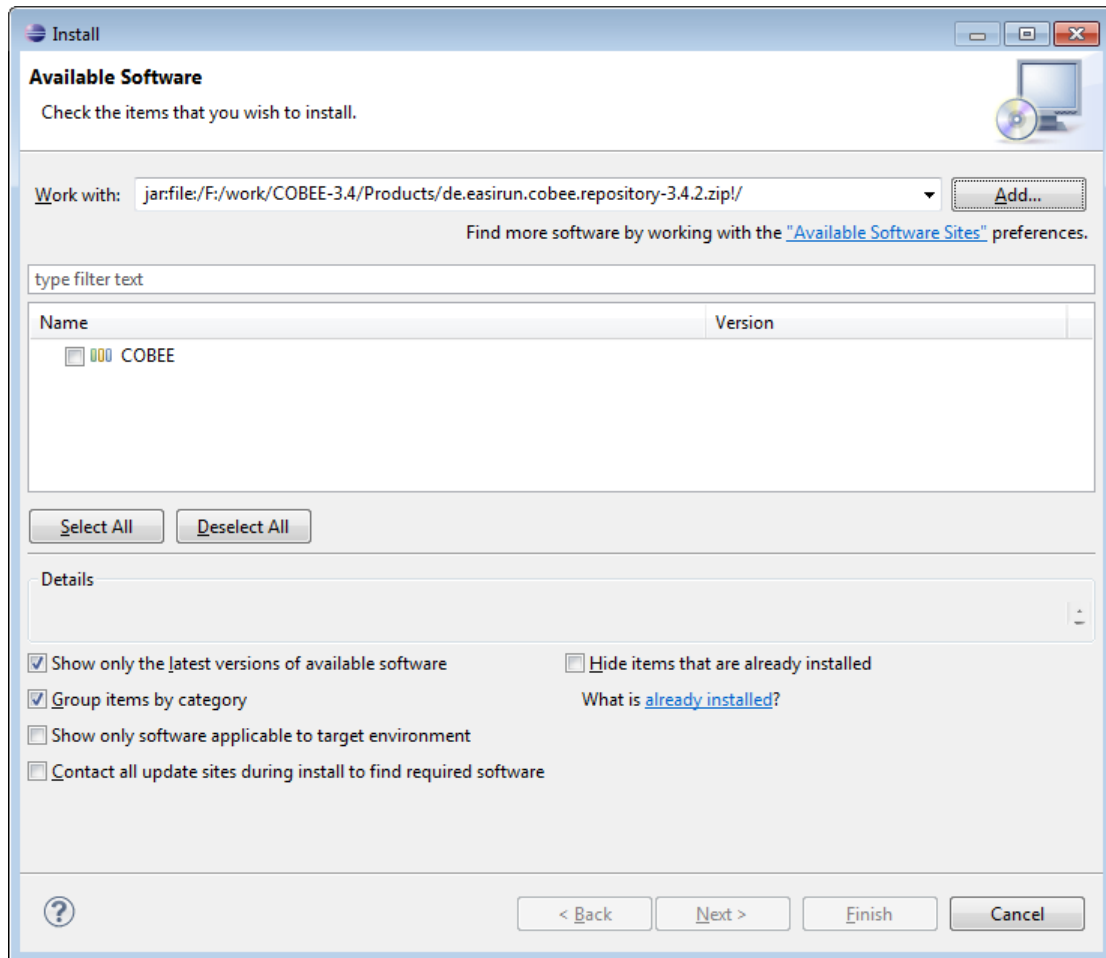
To save time, you should uncheck **"Contact all update sites during install to find required software"**.

- 4 In the dialog box, click on the **"Archive..."** button then select the directory Products where the COBEE plug-in are stored and choose one archive file containing plug-ins to be installed.



The component appear alone in the window:

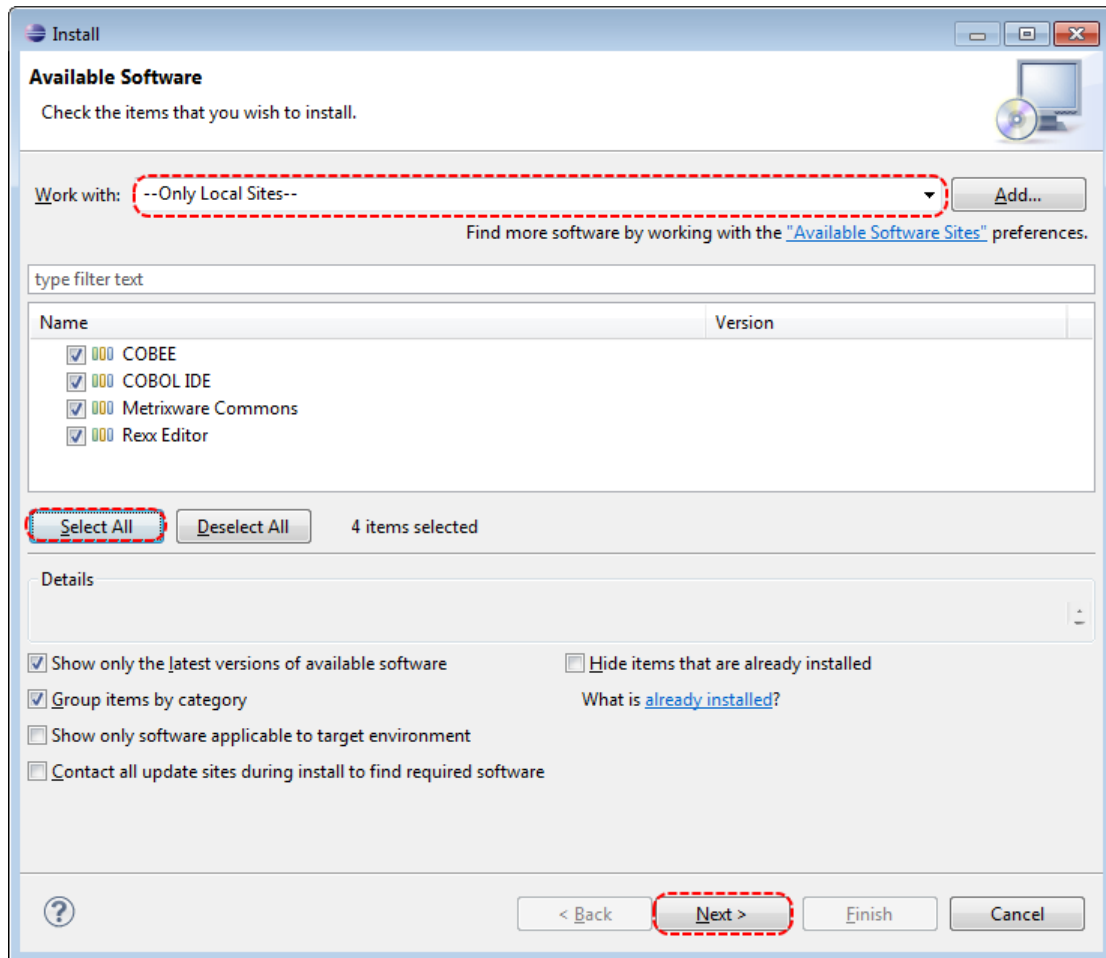




Repeat from step 3 for each archive.

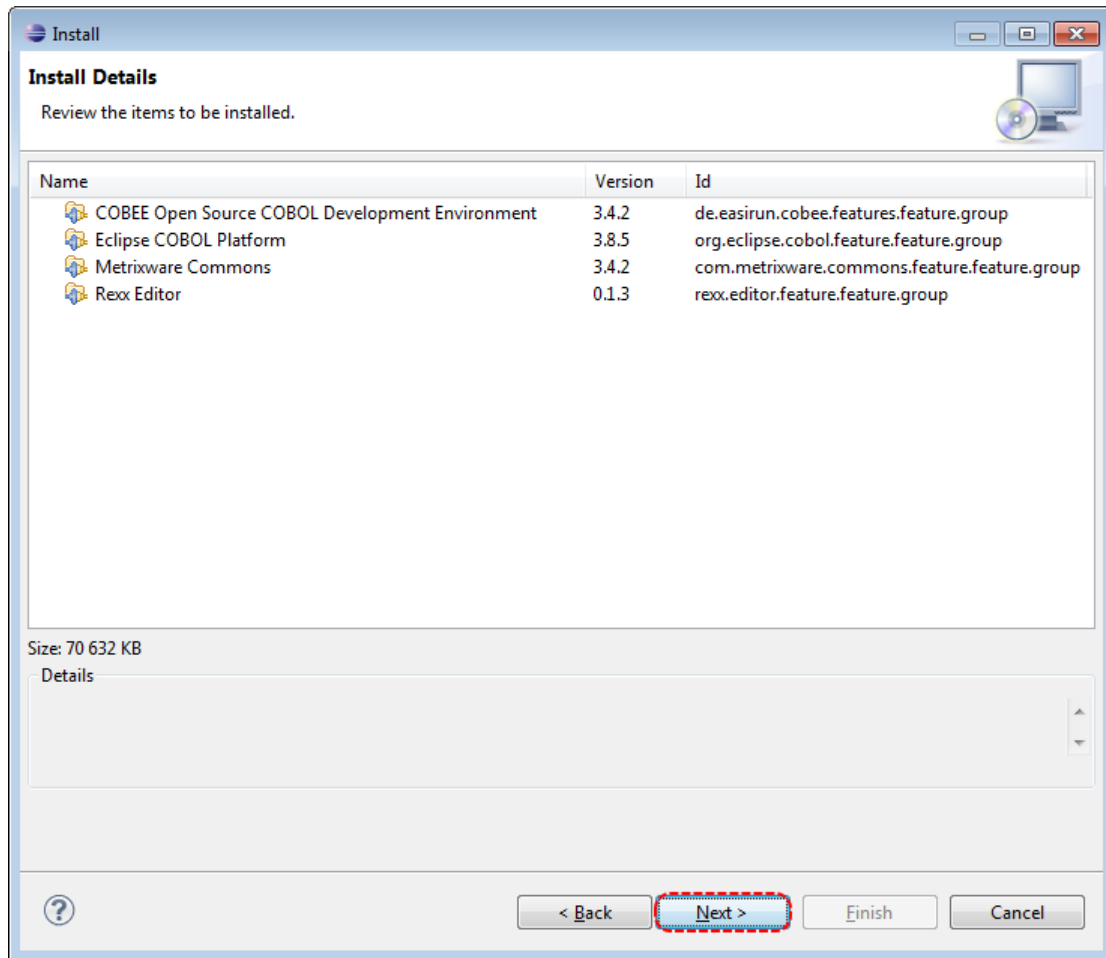
- 5 Check plug-ins: Select **“-Only Local Sites-”**, click on **“Select All”** button and click on **“Next”** button.



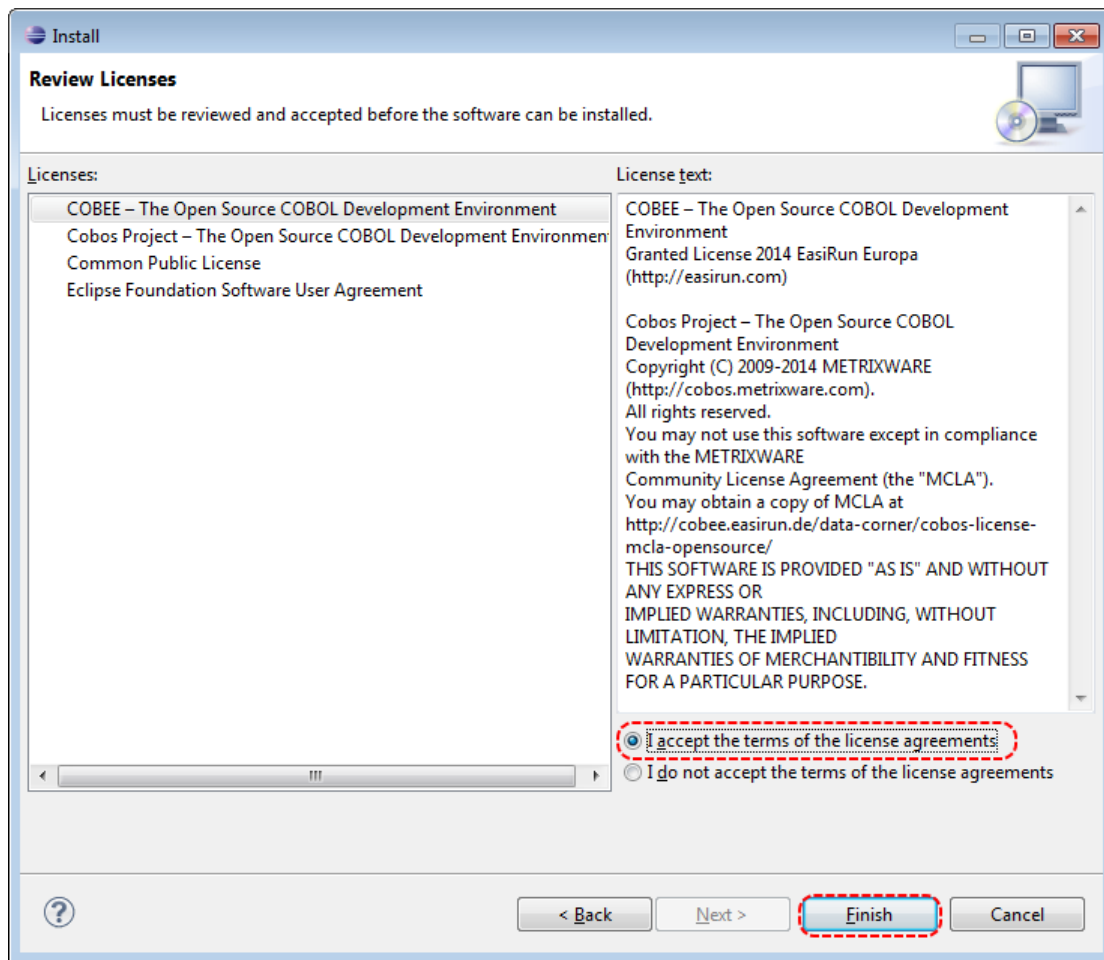


Be patient: "calculating requirements and dependencies" can take a while in some case.

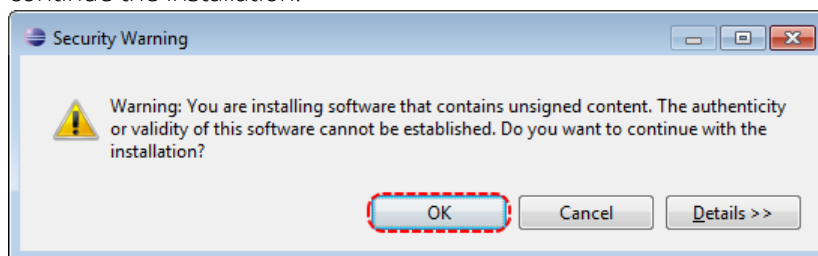
- 6 Review and confirm by clicking on **"Next"** button. Three items are to be installed:



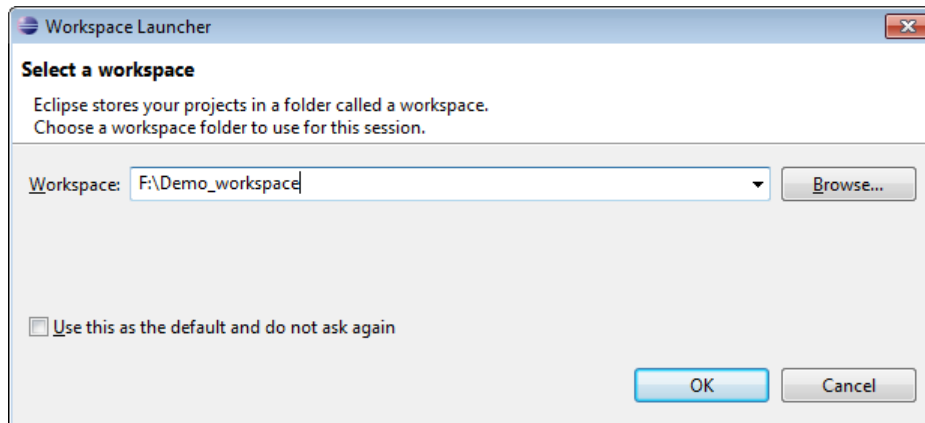
- Accept the terms of the license agreements and click on **"Finish"** button.



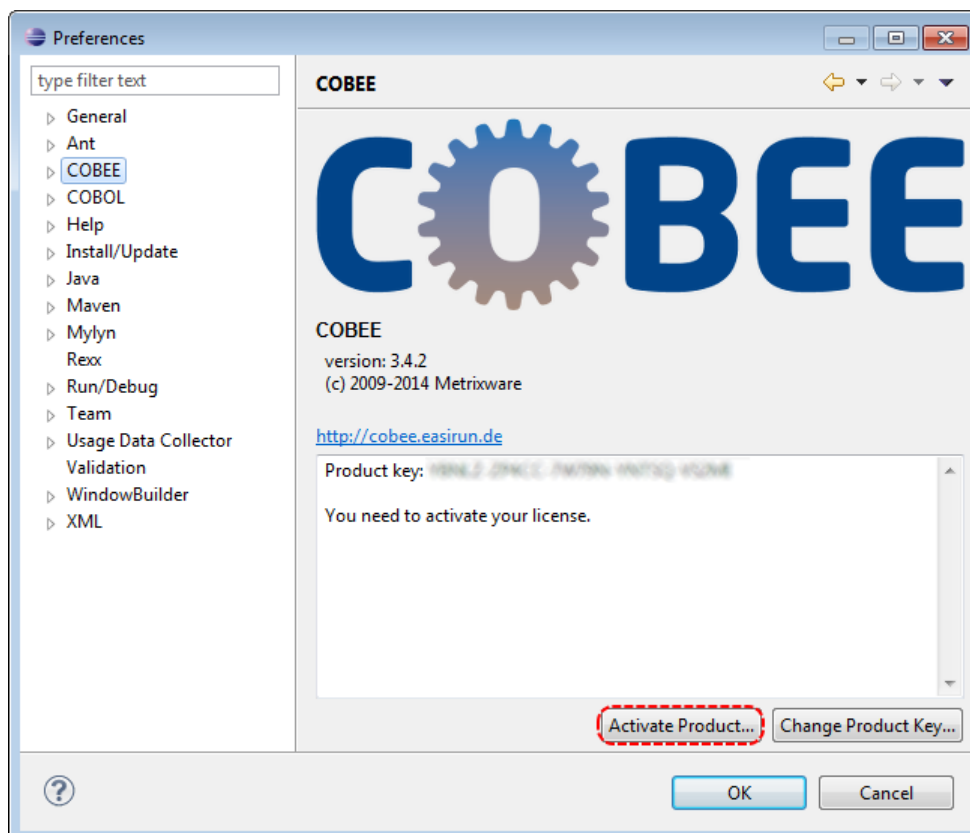
- 8 It is possible that a security warning appears during the installation phase. Click on **"OK"** button to continue the installation.



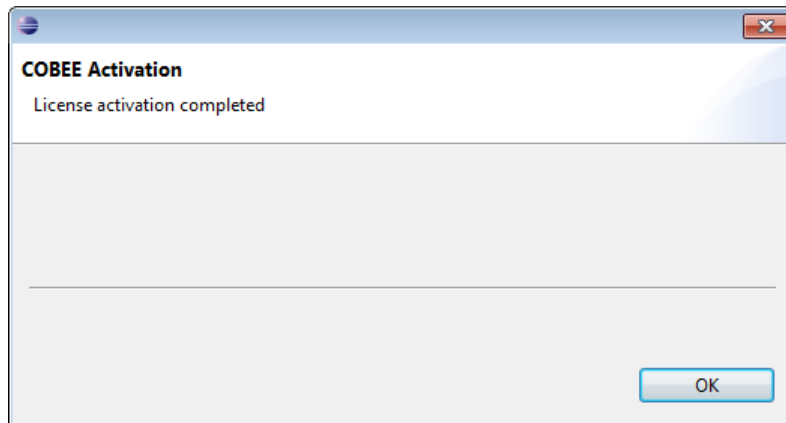
- 9 When the installation is finished, restart Eclipse and **select the Demo\_workspace directory that you have unzipped from COBEE\_3.5.x\_Release-demo.zip file.**



- 10** Go to **“Window ► Preferences ► COBEE”** and click on the **“Activate Product...”** button.



- 11** A popup indicates that the license is activated



Just click OK

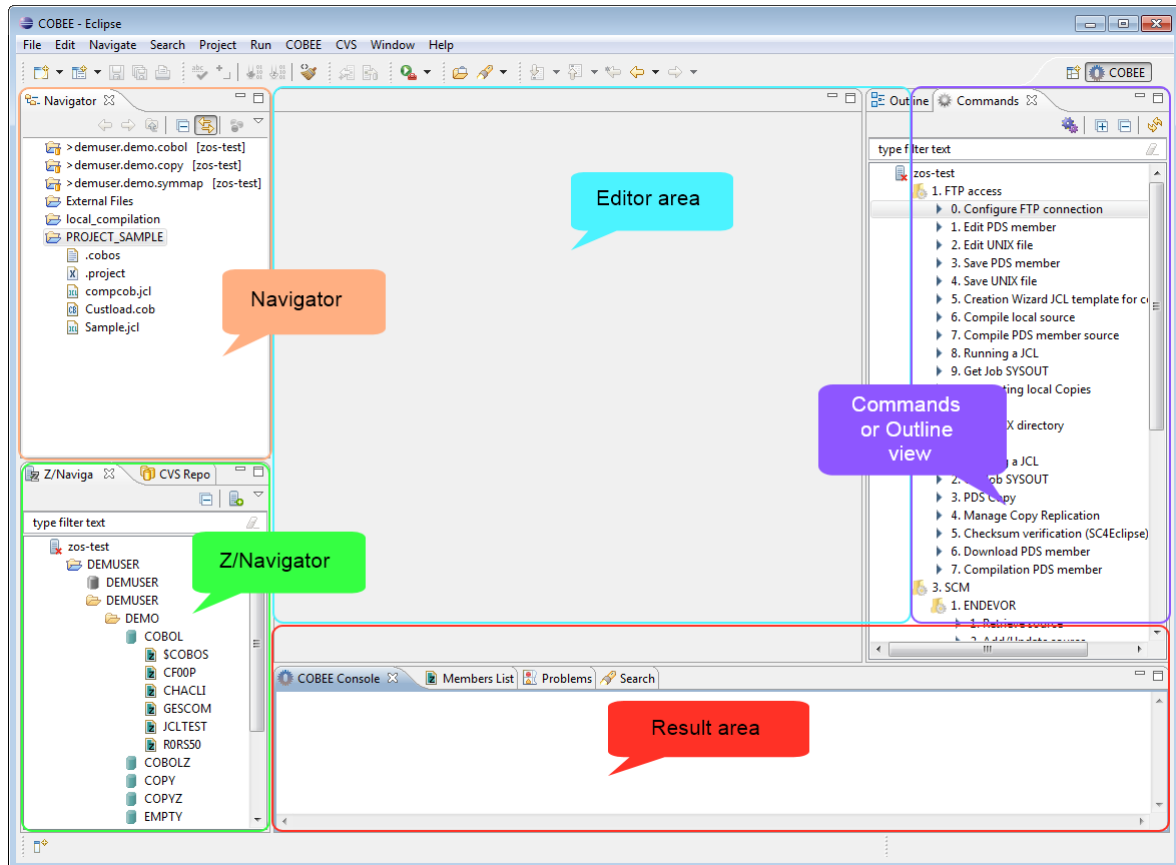
- 12** The Demo license is displayed.



Click OK.

- 13** You are ready to use COBEE in standalone configuration (without installing COBEE components on the

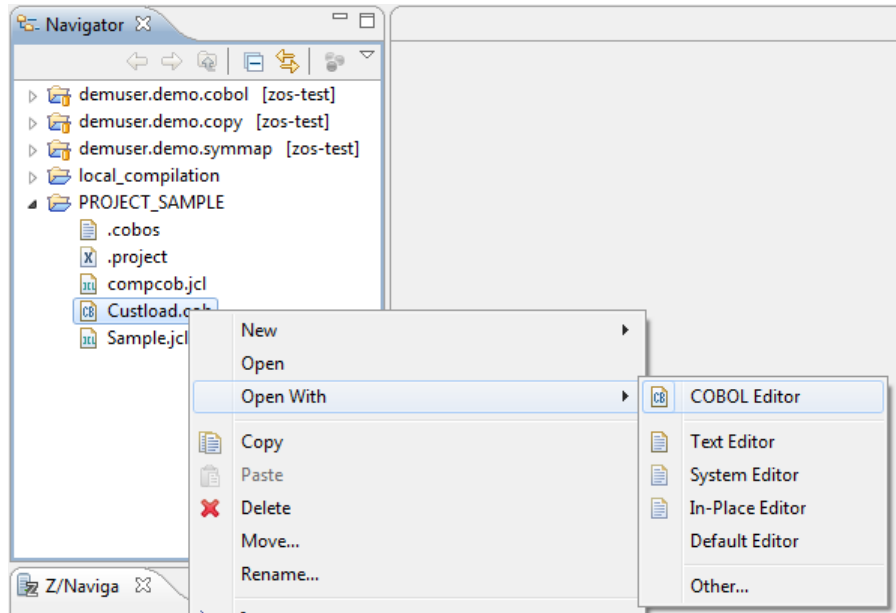
mainframe).



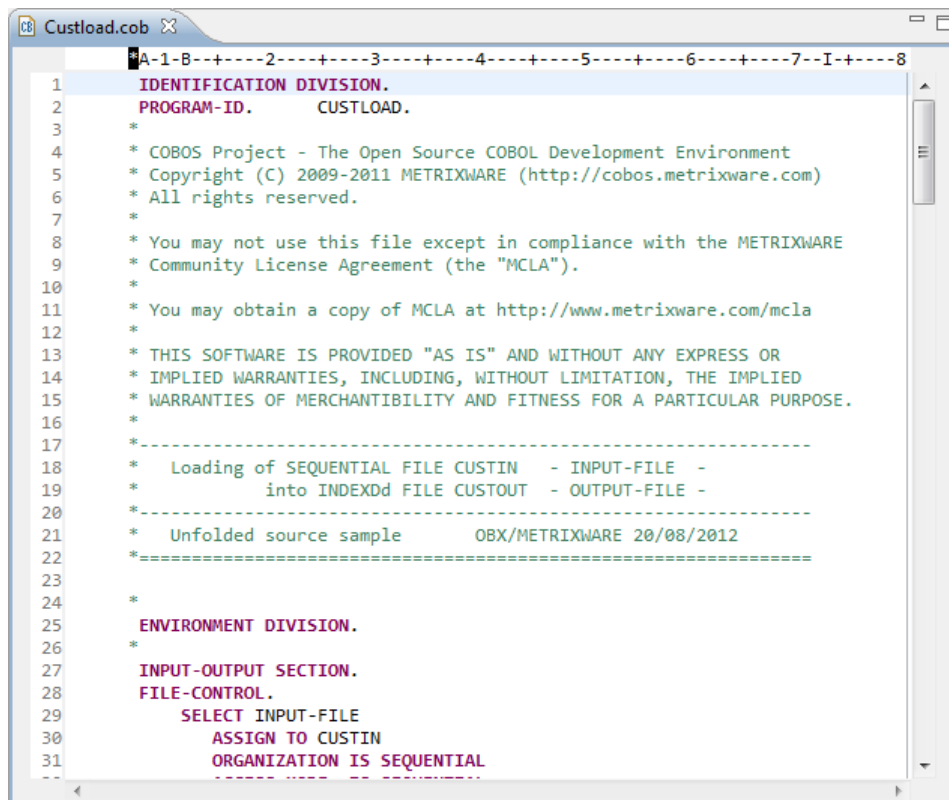
COBEE perspective

### 3. Editing a COBOL program

- 1 Open the program "Custload.cob" from PROJECT\_SAMPLE project with the COBOL Editor by right-clicking on it in the "Navigator" view and by choosing **"Open With ► COBOL Editor"**

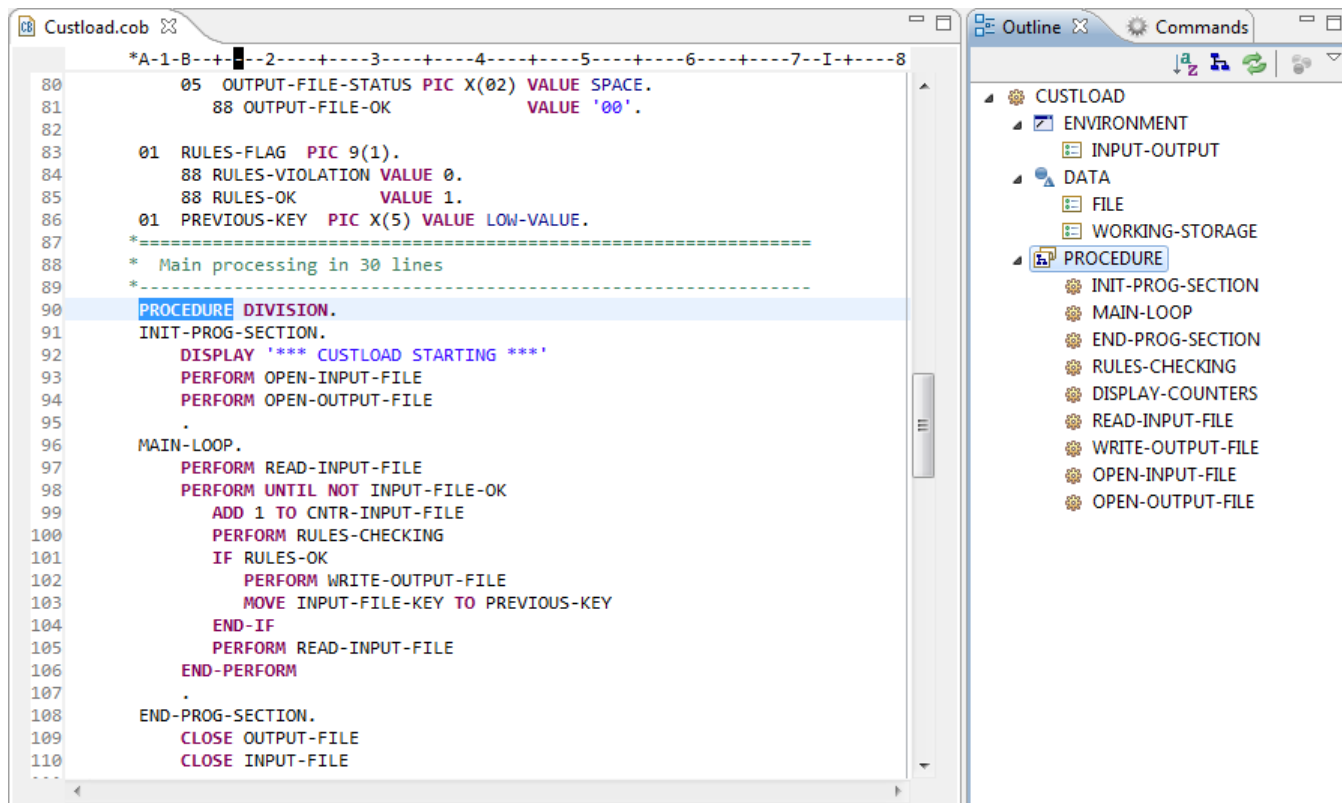


Custload.cob shows off in the Editor.



- 2 Activate the Outline View on the right side. Click on the word PROCEDURE in Outline View.





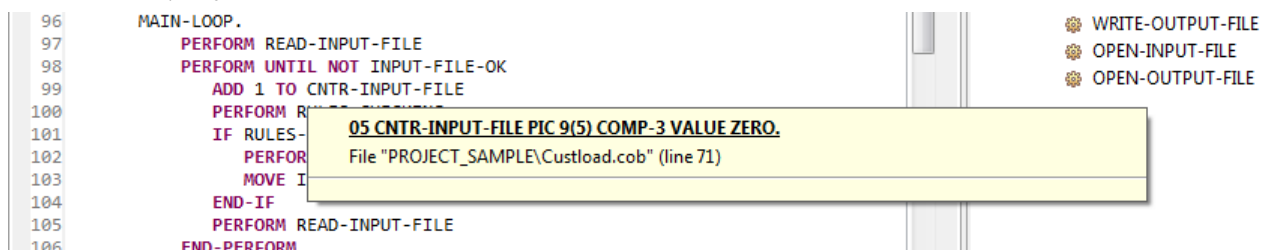
```

80      05 OUTPUT-FILE-STATUS PIC X(02) VALUE SPACE.
81      88 OUTPUT-FILE-OK    VALUE '00'.
82
83      01 RULES-FLAG PIC 9(1).
84      88 RULES-VIOLATION VALUE 0.
85      88 RULES-OK        VALUE 1.
86      01 PREVIOUS-KEY PIC X(5) VALUE LOW-VALUE.
87
88      * Main processing in 30 lines
89
90      PROCEDURE DIVISION.
91      INIT-PROG-SECTION.
92          DISPLAY '*** CUSTLOAD STARTING ***'
93          PERFORM OPEN-INPUT-FILE
94          PERFORM OPEN-OUTPUT-FILE
95
96      MAIN-LOOP.
97          PERFORM READ-INPUT-FILE
98          PERFORM UNTIL NOT INPUT-FILE-OK
99              ADD 1 TO CNTR-INPUT-FILE
100             PERFORM RULES-CHECKING
101             IF RULES-OK
102                 PERFORM WRITE-OUTPUT-FILE
103                 MOVE INPUT-FILE-KEY TO PREVIOUS-KEY
104             END-IF
105             PERFORM READ-INPUT-FILE
106         END-PERFORM
107
108     END-PROG-SECTION.
109     CLOSE OUTPUT-FILE
110     CLOSE INPUT-FILE

```

(for "mainframers") Forget ISPF!

- 3 In the editor, put your mouse above the word CNTR-INPUT-FILE on line 99



```

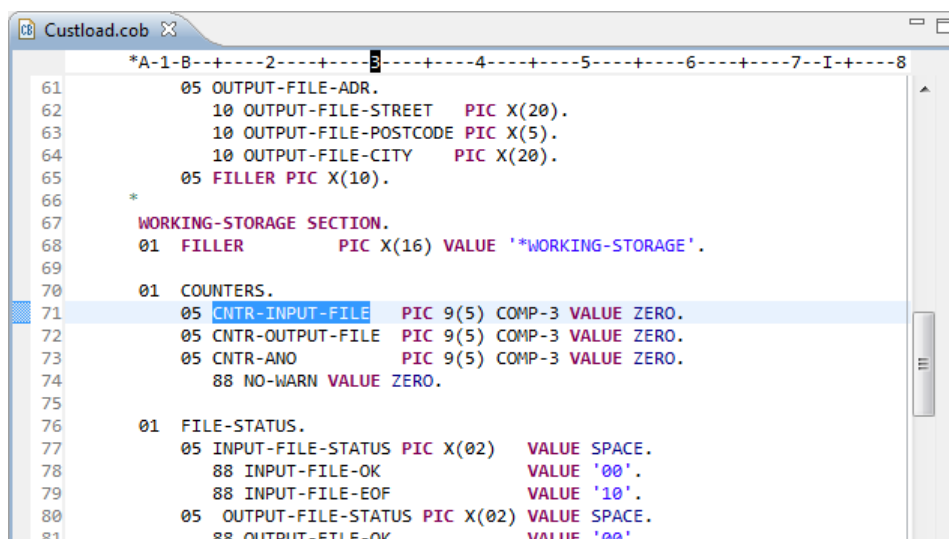
96      MAIN-LOOP.
97          PERFORM READ-INPUT-FILE
98          PERFORM UNTIL NOT INPUT-FILE-OK
99              ADD 1 TO CNTR-INPUT-FILE
100             PERFORM RULES-CHECKING
101             IF RULES-OK
102                 PERFORM WRITE-OUTPUT-FILE
103                 MOVE INPUT-FILE-KEY TO PREVIOUS-KEY
104             END-IF
105             PERFORM READ-INPUT-FILE
106         END-PERFORM

```

05 CNTR-INPUT-FILE PIC 9(5) COMP-3 VALUE ZERO.  
File "PROJECT\_SAMPLE\Custload.cob" (line 71)

You can see line definition of the variable under the mouse pointer.

- 4 Click on variable CNTR-INPUT-FILE on line 99 and hit **F3**



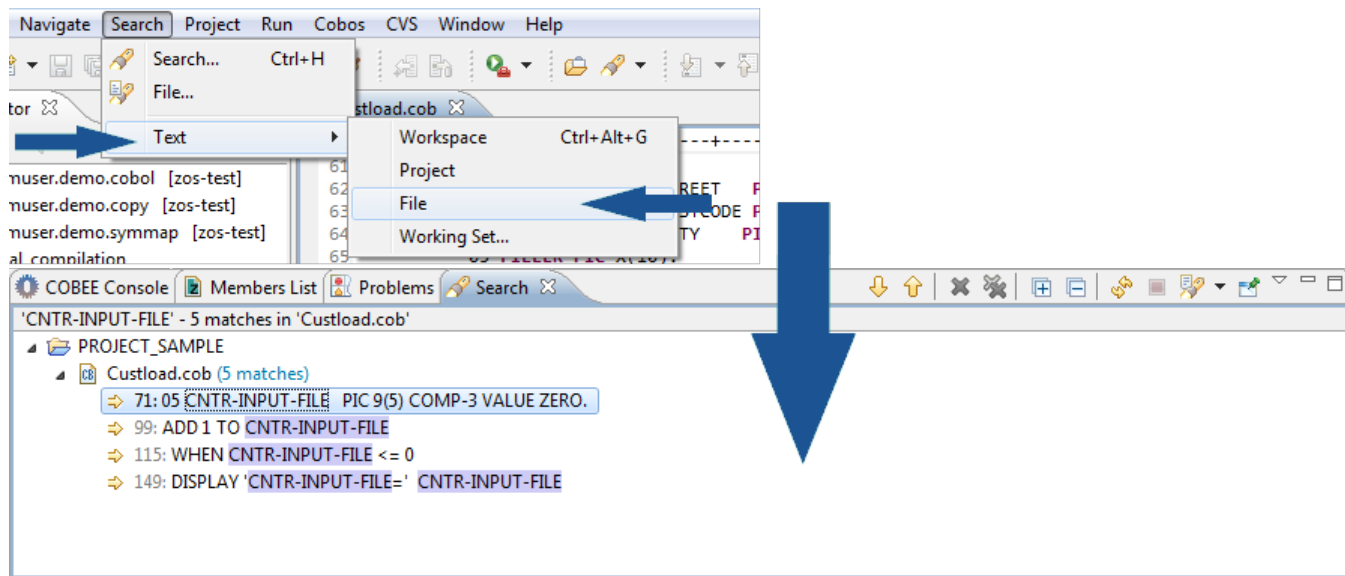
```

61      05 OUTPUT-FILE-ADR.
62      10 OUTPUT-FILE-STREET PIC X(20).
63      10 OUTPUT-FILE-POSTCODE PIC X(5).
64      10 OUTPUT-FILE-CITY PIC X(20).
65      05 FILLER PIC X(10).
66
67      WORKING-STORAGE SECTION.
68      01 FILLER PIC X(16) VALUE '*WORKING-STORAGE'.
69
70      01 COUNTERS.
71      05 CNTR-INPUT-FILE PIC 9(5) COMP-3 VALUE ZERO.
72      05 CNTR-OUTPUT-FILE PIC 9(5) COMP-3 VALUE ZERO.
73      05 CNTR-ANO PIC 9(5) COMP-3 VALUE ZERO.
74      88 NO-WARN VALUE ZERO.
75
76      01 FILE-STATUS.
77      05 INPUT-FILE-STATUS PIC X(02) VALUE SPACE.
78      88 INPUT-FILE-OK VALUE '00'.
79      88 INPUT-FILE-EOF VALUE '10'.
80      05 OUTPUT-FILE-STATUS PIC X(02) VALUE SPACE.
81      88 OUTPUT-FILE-OK VALUE '00'.

```

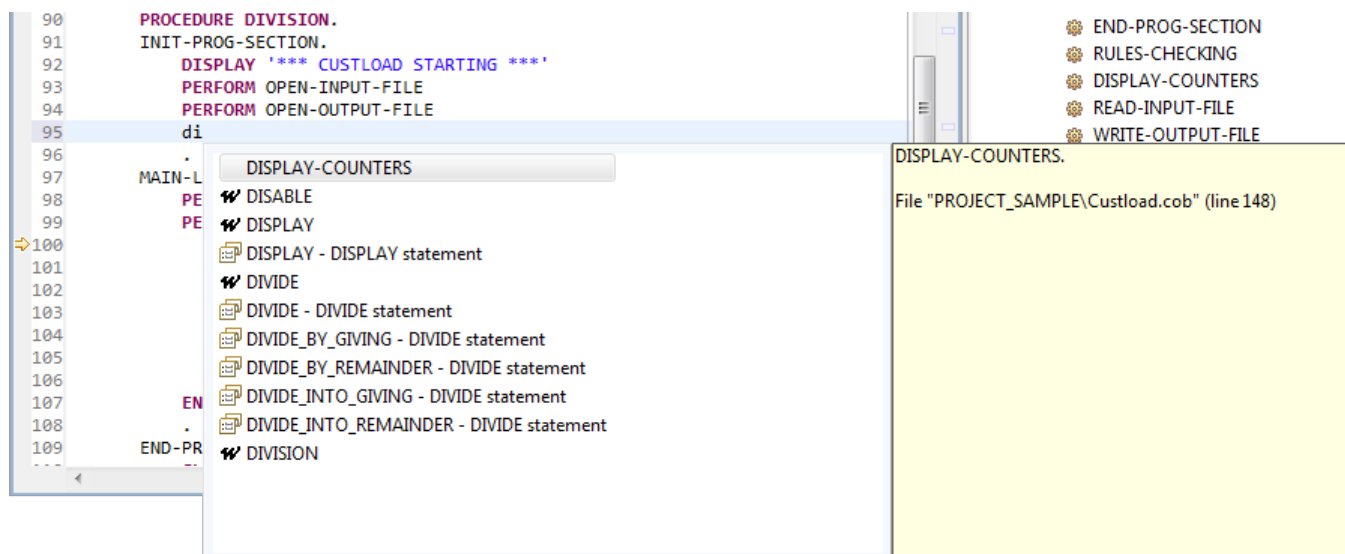
Focus is given to the actual line definition in the program (or in a Copy as well)

 *Tip:* To view all occurrences of this variable, select menu **"Search ► Text ► File"**.



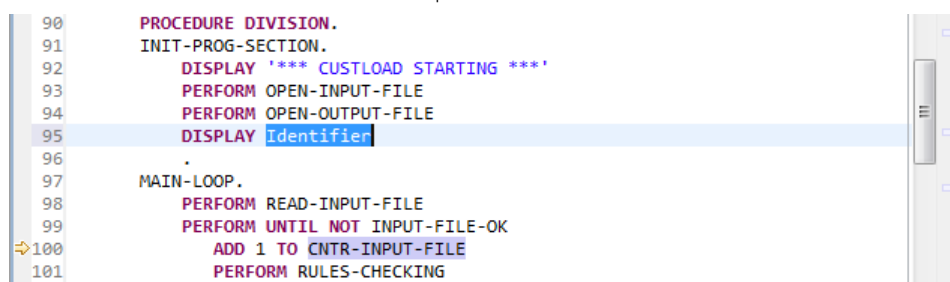
Double clicking on an occurrence in the Search View will position the editor on this occurrence, of course.

**5** Let's insert a statement: place the cursor at the end of line 94 and hit enter. Key 'di' and hit **Ctrl + Space**.



Variables and labels along with their definition line are shown first, then COBOL keywords with or without pattern.

**6** Double Click on DISPLAY statement pattern






- 7 Now, replace identifier by any string you want...e.g.: 'Hello world!'

```
90  PROCEDURE DIVISION.  
91  INIT-PROG-SECTION.  
92      DISPLAY '*** CUSTLOAD STARTING ***'  
93      PERFORM OPEN-INPUT-FILE  
94      PERFORM OPEN-OUTPUT-FILE  
95      DISPLAY 'Hello world!'  
96  .  
97  MAIN-LOOP.  
98      PERFORM READ-INPUT-FILE  
99      PERFORM UNTIL NOT INPUT-FILE-OK  
100     ADD 1 TO CNTR-INPUT-FILE  
101     PERFORM RULES-CHECKING  
102     IF CNTR-INPUT-FILE = 10  
103     GO TO END-MAIN-LOOP  
104     GO TO MAIN-LOOP  
105  END MAIN-LOOP  
106  END PROCEDURE
```

- 8 Finally, save the file (press **Ctrl + S** or push  button from the toolbar).

The main additional keyboard shortcuts are:

- **Ctrl + Space**: invoke auto-completion.
- **Ctrl + Shift + Y**: change selected characters to lowercase
- **Ctrl + Shift + X**: change selected characters to uppercase
- **Ctrl + Shift + Z**: set CAPS ON (like in ISPF Editor)
- **Ctrl + L**: go to the N line in a source file
- **Alt + Shift + A**: Toggle Block Selection (useful for block indentation updating)
- **Ctrl + Q**: return to the last edition in a file
- **Ctrl + E**: go to another opened editor. A pop-up window appears with the choice of opened editors
- **F3**: go to the definition of the variable
- **Ctrl + Shift + V**: check syntax of the code (shortcut for menu "COBEE ►  Check Syntax")
- **Ctrl + Shift + U**: unfold of the code (shortcut for menu "COBEE ►  Unfold COBOL source")
- **Ctrl + Shift + C**: remote compilation<sup>1</sup> (shortcut for menu "COBEE ►  Compilation")

---

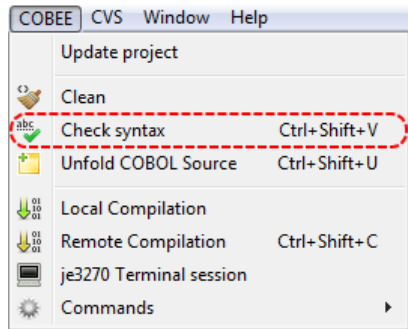
<sup>1</sup> Need a Mainframe.


## 4. Checking the syntax of a COBOL program

### 4.1 CUSTLOAD program

1 Now, we will check the syntax of the program "Custload.cob".

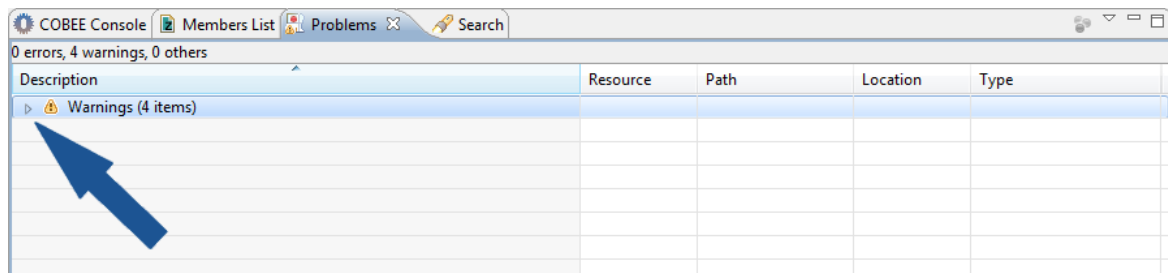
2 Select menu "**COBEE ► Check syntax**"



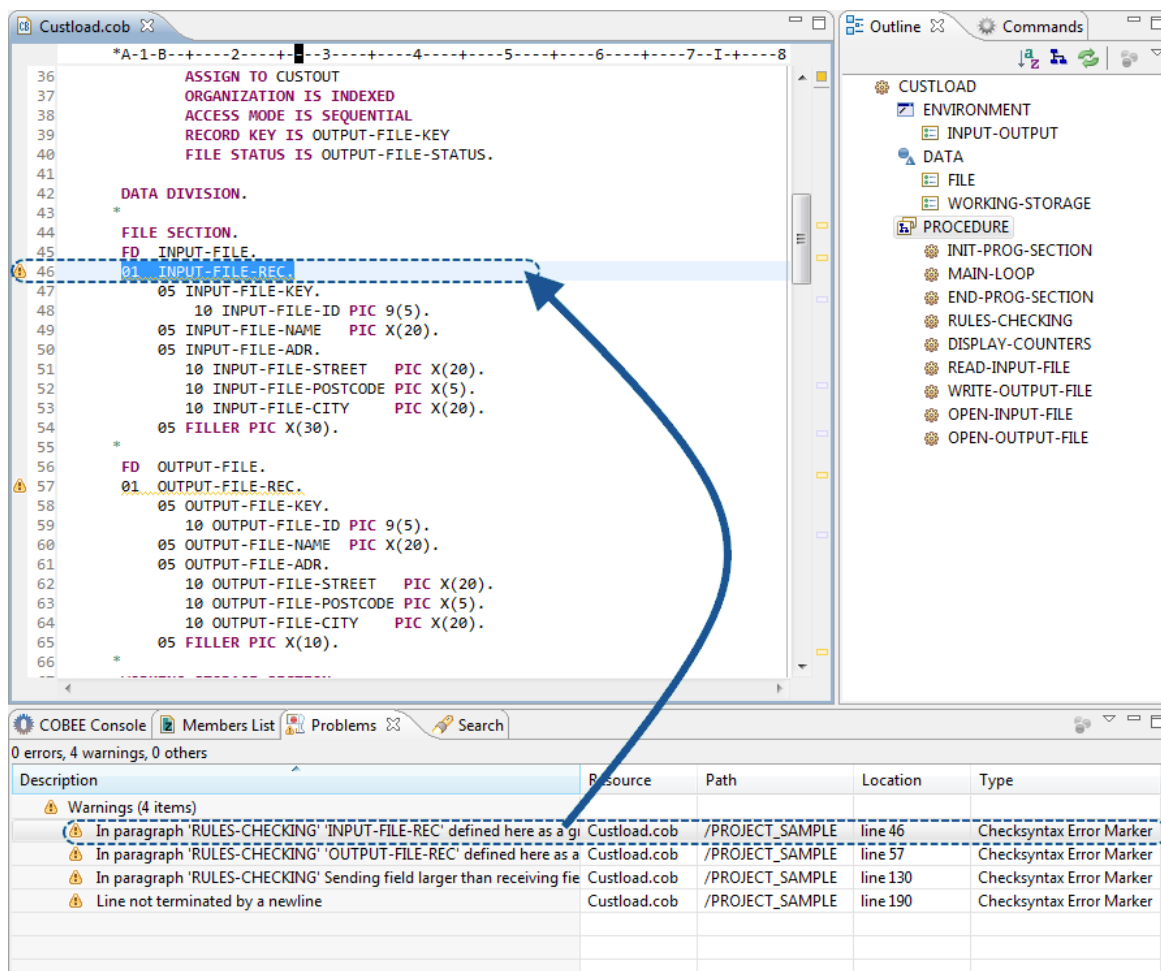
or click on the "check syntax" button placed in the toolbar  or use the keyboard shortcut "**Ctrl + Shift + V**".

3 Once the check syntax has been achieved, 4 warnings are found in this program.

4 In the Problems view, expand the Warnings line to see the messages.



5 Simply double-clicking a message in the Problems view gives focus on the line in error in the COBOL editor.



The screenshot shows the COBEE IDE with the `Custload.cob` program open. The program code is displayed in the main editor, and the COBEE Console at the bottom shows four warnings related to syntax errors. A blue arrow points from the first warning in the console to the corresponding line in the code.

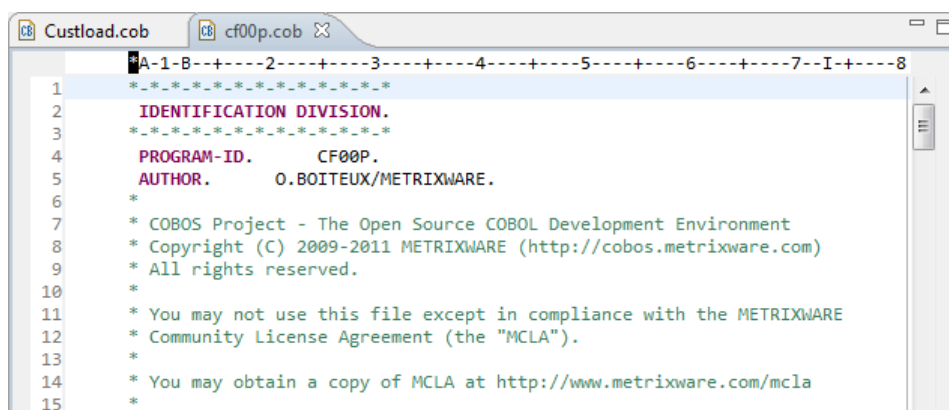
**COBEE Console Warnings:**

Description	Resource	Path	Location	Type
Warnings (4 items)				
In paragraph 'RULES-CHECKING' 'INPUT-FILE-REC' defined here as a gr	Custload.cob	/PROJECT_SAMPLE	line 46	Checksyntax Error Marker
In paragraph 'RULES-CHECKING' 'OUTPUT-FILE-REC' defined here as a	Custload.cob	/PROJECT_SAMPLE	line 57	Checksyntax Error Marker
In paragraph 'RULES-CHECKING' Sending field larger than receiving fie	Custload.cob	/PROJECT_SAMPLE	line 130	Checksyntax Error Marker
Line not terminated by a newline	Custload.cob	/PROJECT_SAMPLE	line 190	Checksyntax Error Marker

## 4.2 CF00P program

In general, COBOL programs contain copybooks. Let's see how it works in COBEE.

- 1 Open the program "cf00p.cob" from demuser.demo.cobol project with the COBOL Editor.



The screenshot shows the COBEE IDE with the `cf00p.cob` program open. The program code is displayed in the main editor, showing the identification division and program details.

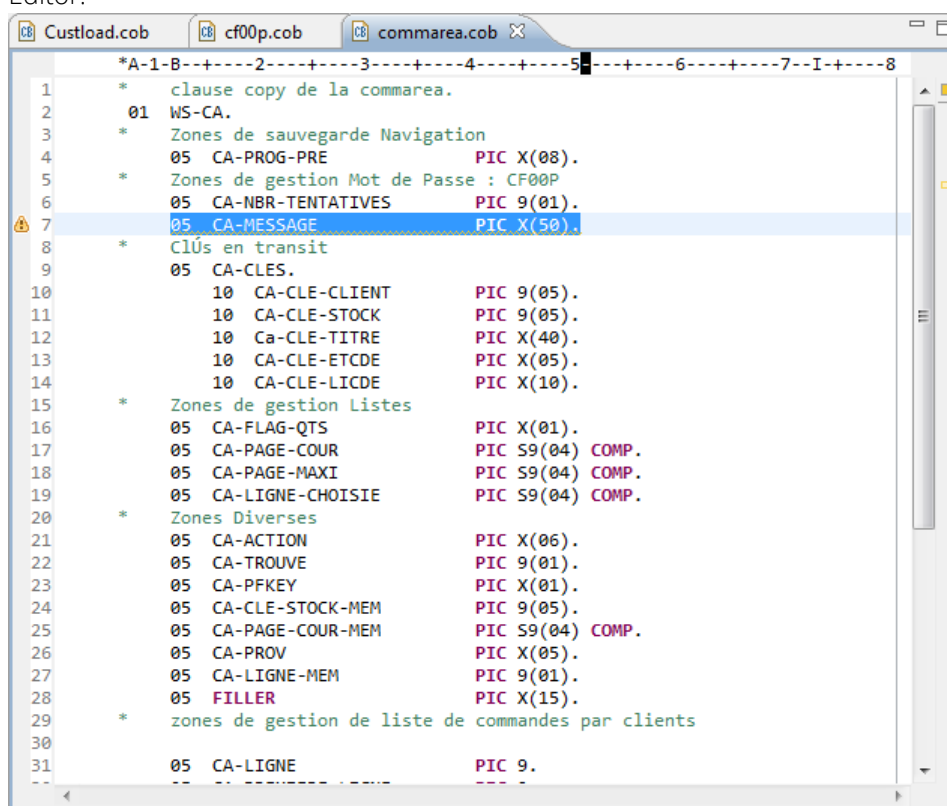
- 2 Check syntax (select menu "COBEE ► Check syntax" or push  or hit "Ctrl + Shift + V").

- 3 Once the check syntax has been achieved, 6 warnings are found in this program.

COBEE Console Members List Problems Search				
0 errors, 6 warnings, 0 others				
Description	Resource	Path	Location	Type
Warnings (6 items)				
In paragraph 'AFFICH-REPONSE-CICS' 'eibresp' defined here as PIC S9(	DFHEIBLC.cob	/External Files	line 90	Checksyntax Error Marker
In paragraph 'AFFICH-REPONSE-CICS' 'REPONSE-CICSD' defined here	cf00w.cob	/demuser.demo.co...	line 17	Checksyntax Error Marker
In paragraph 'AFFICH-REPONSE-CICS' Some digits may be truncated	cf00p.cob	/demuser.demo.co...	line 371	Checksyntax Error Marker
In paragraph 'GESTION-RETOUR-ECRAN' 'CA-MESSAGE' defined here	commarea.cob	/demuser.demo.co...	line 7	Checksyntax Error Marker
In paragraph 'GESTION-RETOUR-ECRAN' 'WS-MESSAGE-PSW' defined	cf00p.cob	/demuser.demo.co...	line 70	Checksyntax Error Marker
In paragraph 'GESTION-RETOUR-ECRAN' Sending field larger than rece	cf00p.cob	/demuser.demo.co...	line 123	Checksyntax Error Marker

In the column "Resource", you can see that some messages refer to files outside the main program: these copybooks are included in the program at syntax checking.

- 4 Double click on the warning of the copybook "commarea.cob". This copybook shows off in the COBOL Editor.



```

*A-1-B-+---2---+---3---+---4---+---5---+---6---+---7-I-+---8
1  * clause copy de la commarea.
2  01 WS-CA.
3  * Zones de sauvegarde Navigation
4  05 CA-PROG-PRE          PIC X(08).
5  * Zones de gestion Mot de Passe : CF00P
6  05 CA-NBR-TENTATIVES    PIC 9(01).
7  05 CA-MESSAGE           PIC X(50).
8  * Clés en transit
9  05 CA-CLES.
10     10 CA-CLE-CLIENT     PIC 9(05).
11     10 CA-CLE-STOCK      PIC 9(05).
12     10 CA-CLE-TITRE      PIC X(40).
13     10 CA-CLE-ETCDE      PIC X(05).
14     10 CA-CLE-LICDE      PIC X(10).
15  * Zones de gestion Listes
16     05 CA-FLAG-QTS       PIC X(01).
17     05 CA-PAGE-COUR      PIC S9(04) COMP.
18     05 CA-PAGE-MAXI      PIC S9(04) COMP.
19     05 CA-LIGNE-CHOISIE  PIC S9(04) COMP.
20  * Zones Diverses
21     05 CA-ACTION         PIC X(06).
22     05 CA-TROUVE         PIC 9(01).
23     05 CA-PFKEY          PIC X(01).
24     05 CA-CLE-STOCK-MEM  PIC 9(05).
25     05 CA-PAGE-COUR-MEM  PIC S9(04) COMP.
26     05 CA-PROV           PIC X(05).
27     05 CA-LIGNE-MEM      PIC 9(01).
28     05 FILLER            PIC X(15).
29  * zones de gestion de liste de commandes par clients
30
31     05 CA-LIGNE          PIC 9.

```

A marker is placed in front of the line addressed by the warning and the line is highlighted.

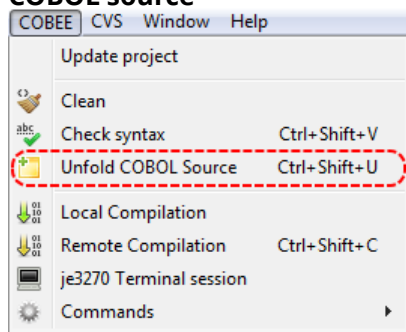


## 5. Unfolding a COBOL program

This function displays the original program including source code, copybooks and SQL includes which are used by the program. The expanded file (Unfolded) is opened in a new tab in read-only mode. The expanded copybooks are displayed on a yellow background, indicating their path on the start line beginning with “++SCCOPY” and ending with “--SCCOPY”.

### 5.1 CF00P program

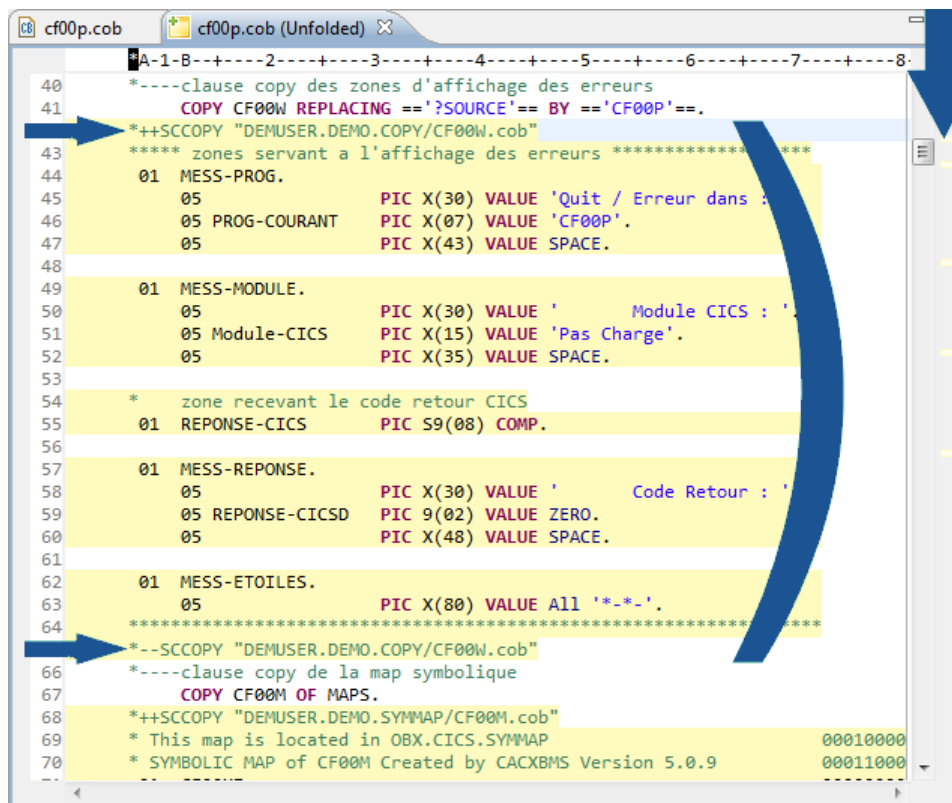
- 1 Check that the program “cf00p.cob” is active in the COBOL Editor, select menu **“COBEE ► Unfold COBOL source”**



or click on the “unfold” button placed in the toolbar

or use the keyboard shortcut **“Ctrl + Shift + U”**.

- 2 After Unfold, you will get a new tab with the name “cf00p.cob (Unfolded)”.



You can browse through the source code by clicking on the yellow markers on the right.



Tip:

To view the list of copybooks, position the cursor over SCCOPY and select menu **“Search ► Text ► File”**.


## 5.2 GESCOM program

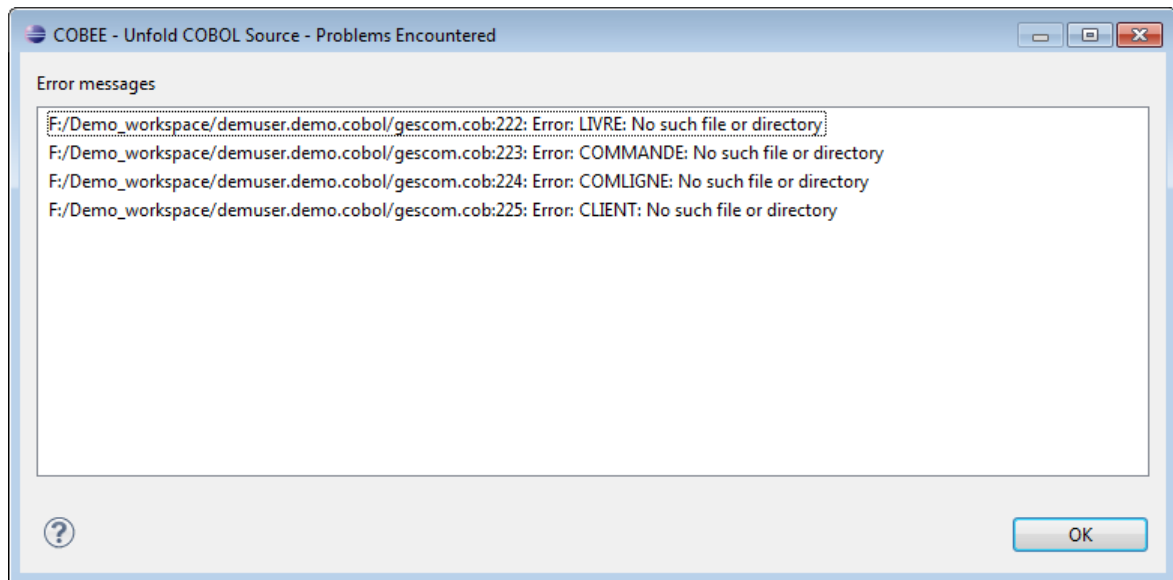
OK, now let's see how we manage to make it with mainframe COBOL programs. For successful **Check syntax**, **Unfold**, **Auto-completion**, etc...we need to access copybooks used by the programs we are working on.

One solution is to replicate the copybooks in a network place, and configure the .cobos file.

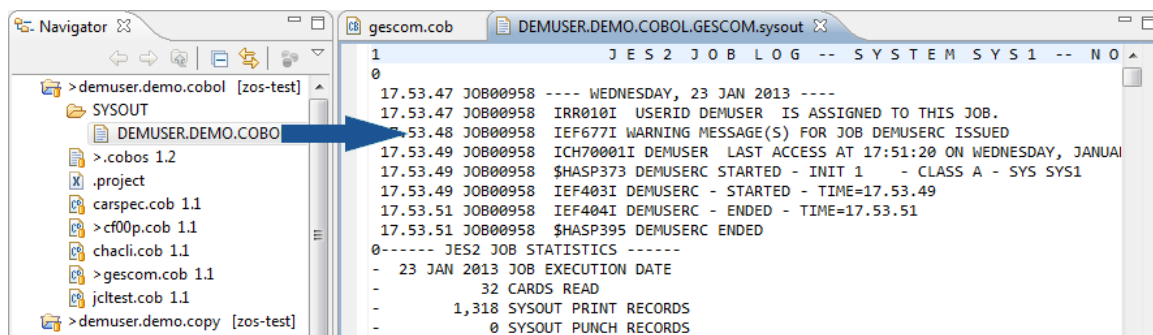
Note: if you want to understand how copybooks paths are specified, open the .cobos file. For more information, select menu **"Help ► Help Contents ► COBEE ► Chapter 5 Viewing and ... ► Configuring a project"**.

One another very simple solution is to retrieve copybooks from a compilation sysout. Let's run this scenario with a COBOL program we have compiled from COBEE (and sysout is still available).

- 1 Open the program "gescom.cob" from demuser.demo.cobol project with the COBOL Editor.
- 2 Unfold (select menu **"COBEE ► Unfold COBOL source"** or push  button).
- 3 After Unfold, you will get a new tab with the name "gescom.cob (Unfolded)" but a popup shows off because copybooks are not available according to the paths specified in .cobos file:



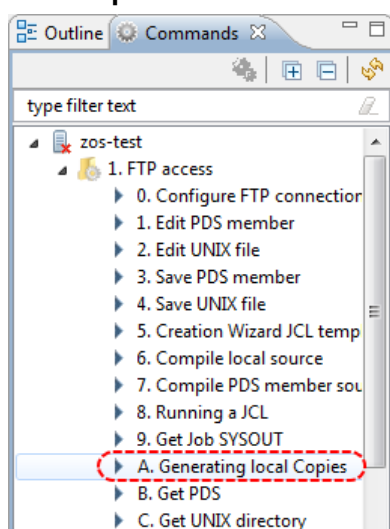
- 4 Close the "gescom.cob (Unfolded)".
- 5 If you want to look at the sysout content, you can retrieve the sysout in the Navigator View in demuser.demo.cobol/SYSOUT and double click on it.



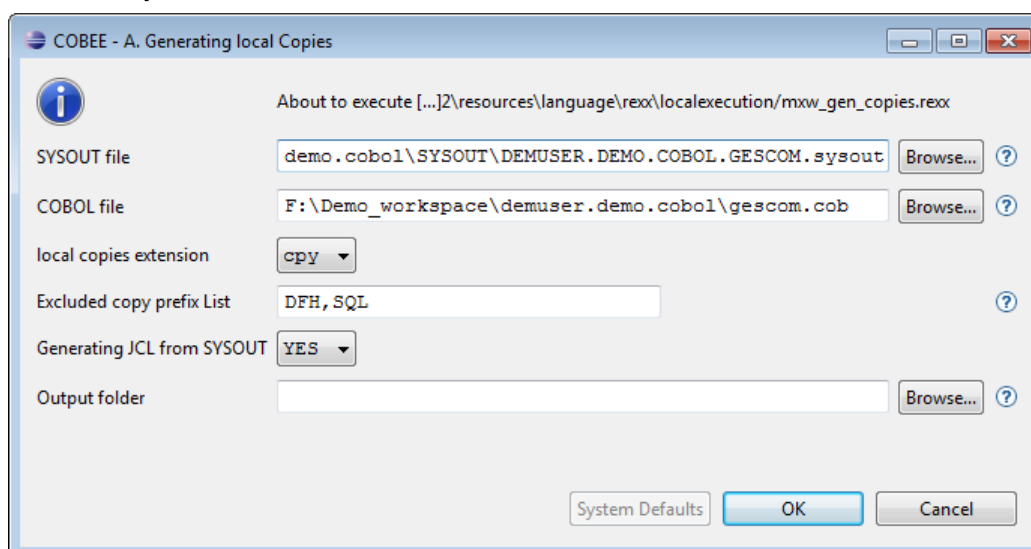
The sysout is opened in the Text Editor.

Warning: If you encounter "resource is out of sync ...", you have to press Refresh key (F5)

- 6 Give back focus to "gescom.cob" and launch the COBEE command **"1. FTP access ► A. Generating local Copies"** from the Command view.

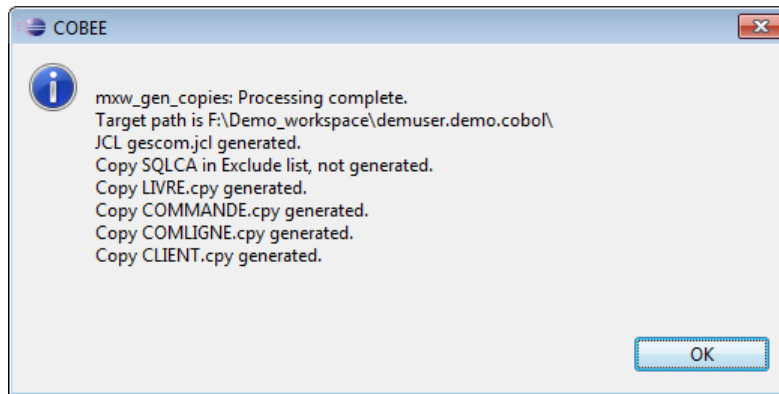


- 7 Select the sysout and click OK.

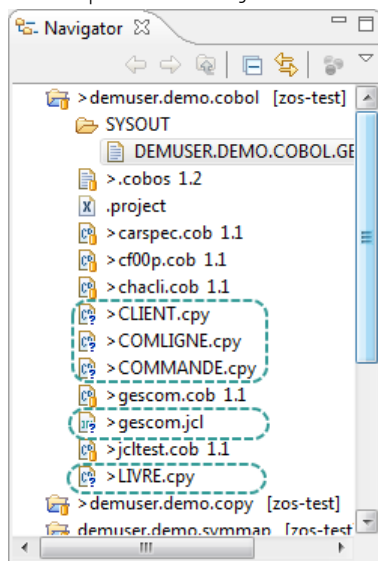


Note: in this example, we exclude CICS and DB2 copybooks.

- 8 A pop-up is displayed showing the list of copybooks and the generated JCL.

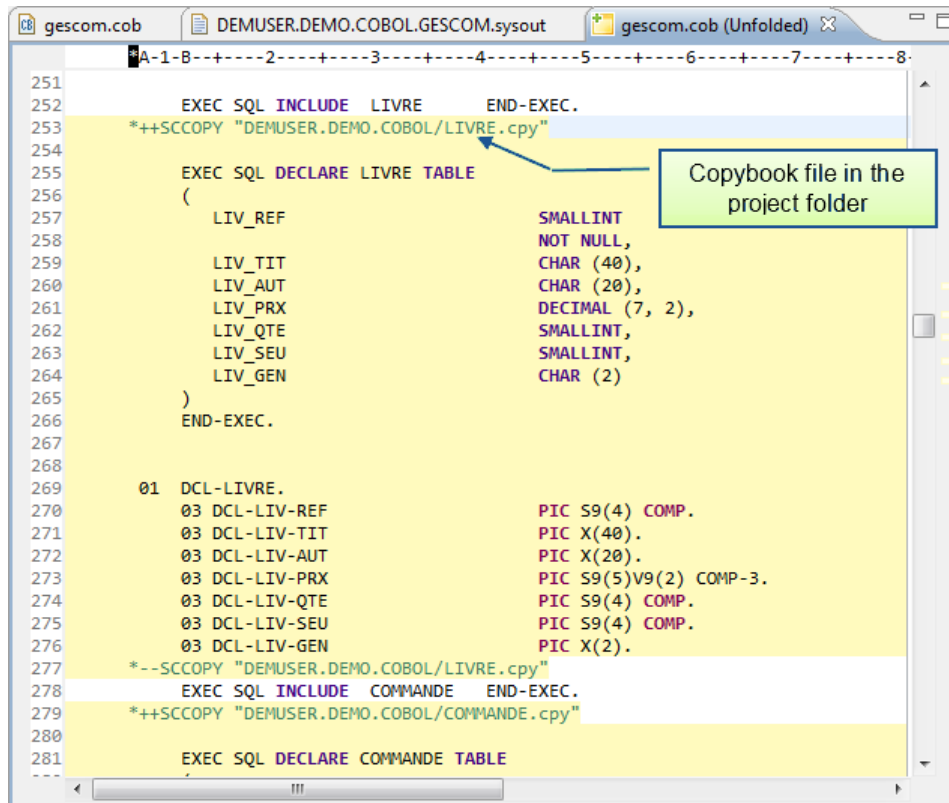


- 9 The copies and the JCL are stored along with the source file.



Note: COBEE don't look for copybooks systematically first in the directory of the program (if you want to explore this directory, you MUST specify it in the .cobos file). You can of course store the copybooks in another place.

- 10 Now, we can **unfold** the source file.



```

251
252 EXEC SQL INCLUDE LIVRE END-EXEC.
253 *++SCCOPY "DEMUSER.DEMO.COBOL/LIVRE.cpy"
254
255 EXEC SQL DECLARE LIVRE TABLE
256 (
257     LIV_REF SMALLINT
258     NOT NULL,
259     LIV_TIT CHAR (40),
260     LIV_AUT CHAR (20),
261     LIV_PRX DECIMAL (7, 2),
262     LIV_QTE SMALLINT,
263     LIV_SEU SMALLINT,
264     LIV_GEN CHAR (2)
265 )
266 END-EXEC.
267
268
269 01 DCL-LIVRE.
270 03 DCL-LIV-REF PIC S9(4) COMP.
271 03 DCL-LIV-TIT PIC X(40).
272 03 DCL-LIV-AUT PIC X(20).
273 03 DCL-LIV-PRX PIC S9(5)V9(2) COMP-3.
274 03 DCL-LIV-QTE PIC S9(4) COMP.
275 03 DCL-LIV-SEU PIC S9(4) COMP.
276 03 DCL-LIV-GEN PIC X(2).
277 *--SCCOPY "DEMUSER.DEMO.COBOL/LIVRE.cpy"
278 EXEC SQL INCLUDE COMMANDE END-EXEC.
279 *++SCCOPY "DEMUSER.DEMO.COBOL/COMMANDE.cpy"
280
281 EXEC SQL DECLARE COMMANDE TABLE

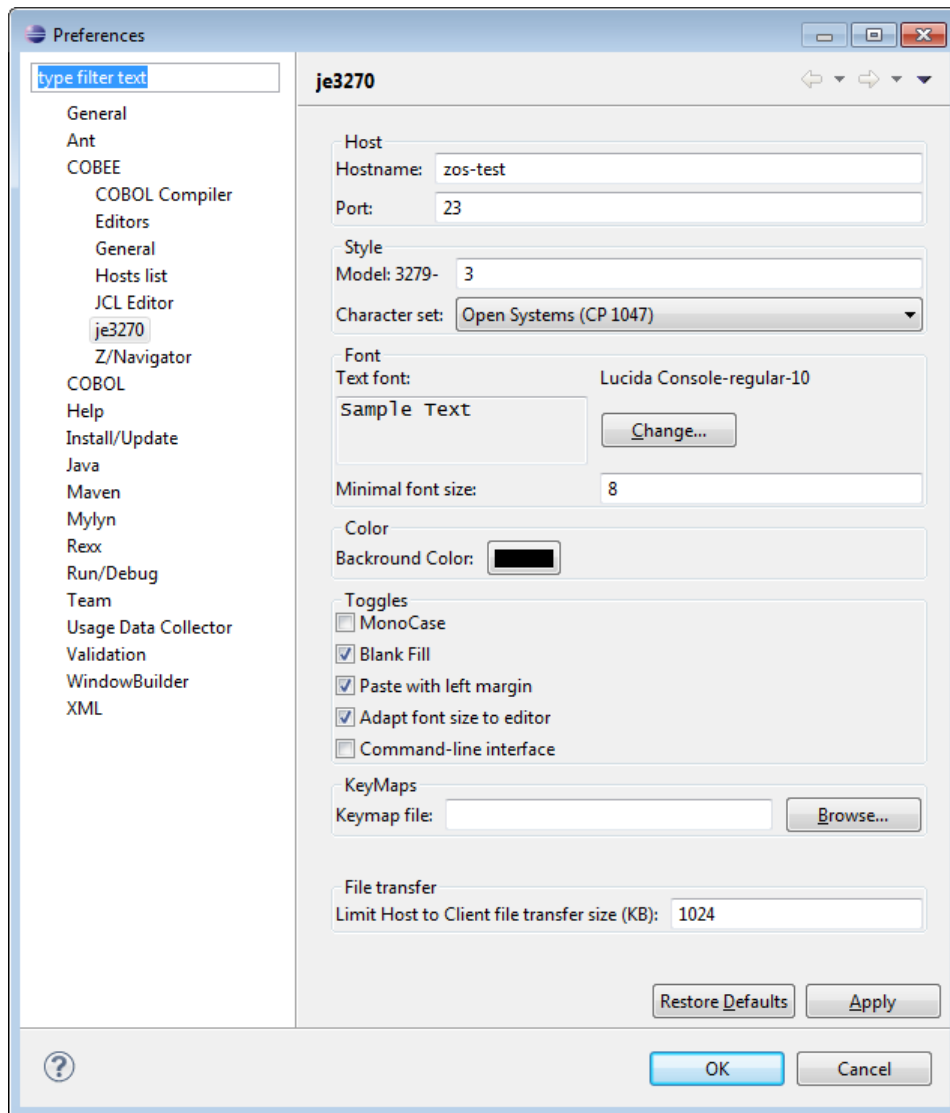
```

Note: Copybooks retrieving is useful also for Check Syntax, auto-completion and Hover.

## 6. Using 3270 Emulator inside eclipse

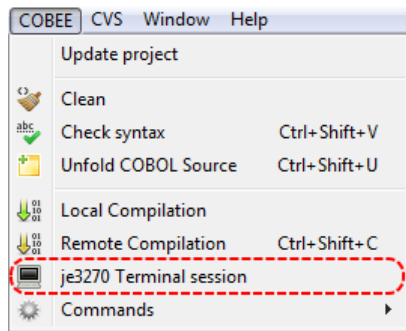
Powerful COBEE extension is designed for mainframe COBOL development enhancement. First of all, 3270 access to mainframe remains the basic way of working with a mainframe (but not the most comfortable, as already shown in this document)

- 1 Go to **“Window ► Preferences ► COBEE ► je3270”** and adjust the values for your environment (at least, Host name field must be filled with IP address or DNS name of your host)

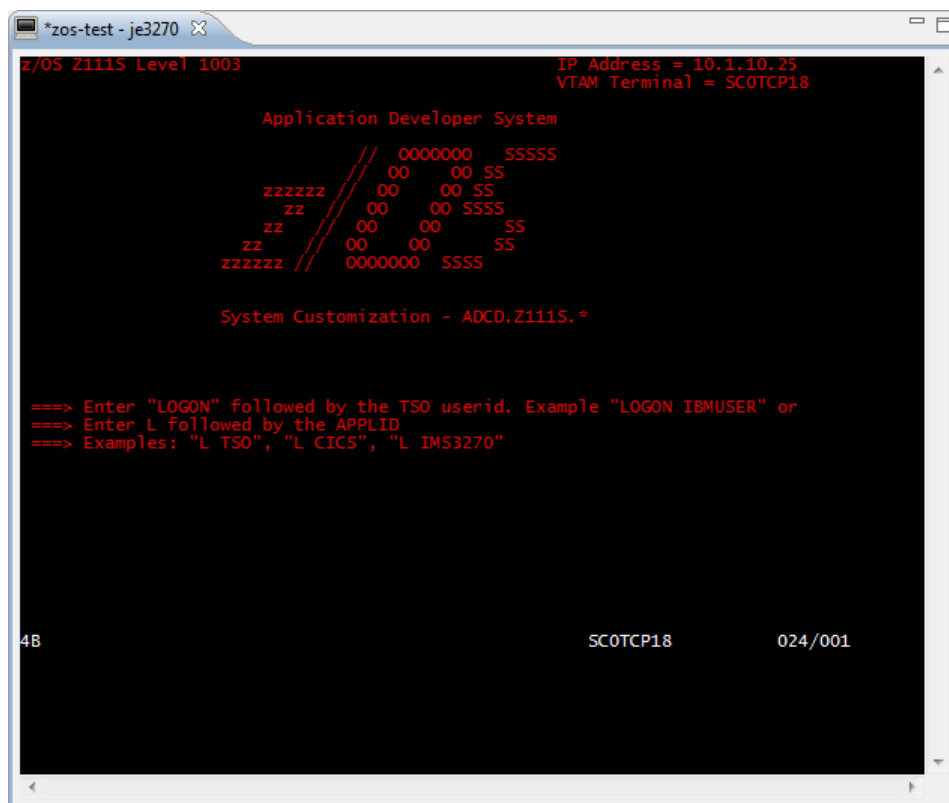


You will be able to define a customized key map.

- 2 Launch the 3270 emulator from menu **“COBEE ► je3270 Terminal session”**:



- 3 Your favorite welcome screen is displayed! Enter your credentials.



You are now at home in your usual mainframe environment BUT inside eclipse!

A button bar appears above the je3270 window with useful functions. Additional shortcuts are defined in keymap file (see Help Contents / 3270 Terminal for further information)

*Note: F10 key is well interpreted by je3270 but in the same time trapped by eclipse, activating File Menu. You have to retype F10 to give focus back to je3270. A bottle of good champagne for a fix of this issue!*



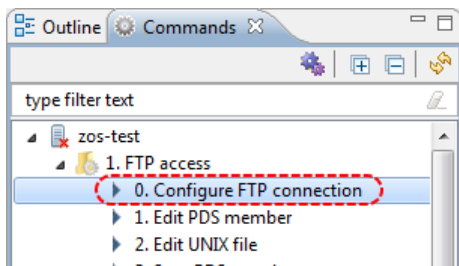
## 7. Configure FTP access

First, you must configure the FTP connection to use “FTP access” commands.

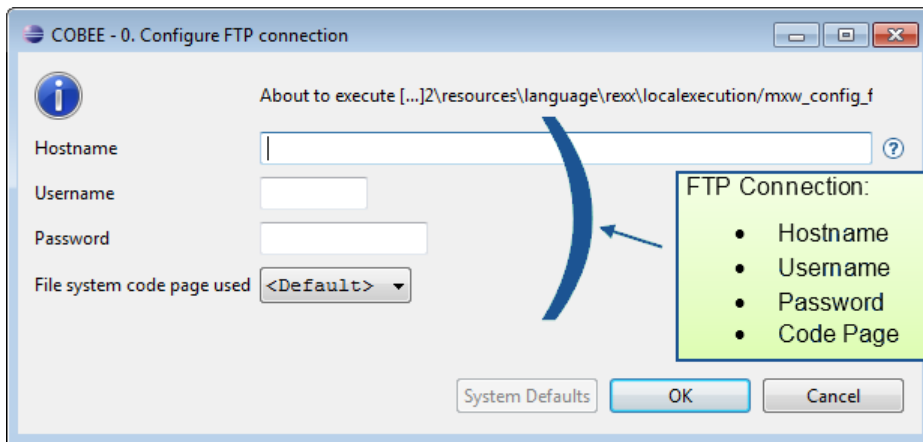
“FTP Access” module provides another way of working with mainframe sources without any mainframe installation. This module requires installation of a REXX interpreter on the workstation such as [Regina REXX Interpreter](#).

**In this quick start, we are going to use for this purpose the “FTP Access” facility. At this time, you should have installed a REXX interpreter (type “rexx -v” from a DOS prompt)**

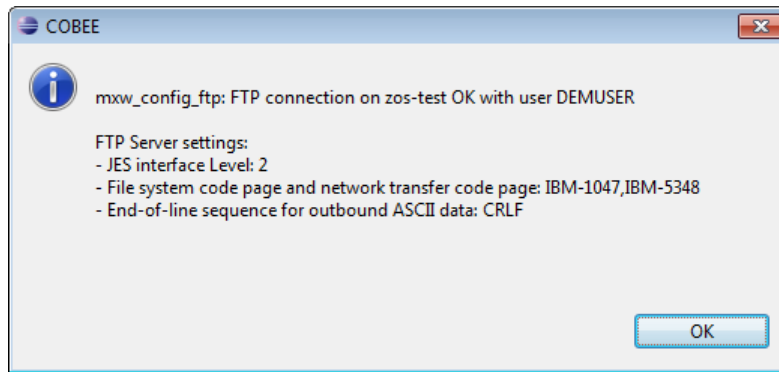
- 1 Double-click on **“1. FTP access ► 0. Configure FTP connection”** in the Commands view:



- 2 Fill in the form fields:



- 3 Click on **“OK”** button to check the data.
- 4 If the connection is successful, data connection is saved in workspace and a popup shows you the JES interface level.



Here JES interface Level is 2.

Note: JES interface Level differences is described in "IP User's Guide and Commands" documentation from IBM.

## 8. Compiling a local source on mainframe

COBEE is specially designed to easily develop mainframe applications. Also, you can compile a local program on the mainframe and retrieve the SYSOUT locally.

The error messages are displayed in the Problems view synchronized with the source in the editor.

Z/Navigator plug-in is designed to process mainframe files directly from eclipse but requires scripts installation on mainframe.

If you are comfortable with JCL editing, try adapting the JCL sample supplied for a batch program. If not, feel free to go to next sequence: Editing a PDS member page 35.

In the PROJECT\_SAMPLE project:

- 1 Open *Custload.cob* in COBOL Editor.
- 2 Open *compcob.jcl* in JCL Editor

```
//xxxxxxx JOB 'COMPILES',CLASS=A,MSGCLASS=H,REGION=0M
/*-----
/* COBOL compile and link sample  Cobos 2.3  ZOSV1R11  Sept 2012
/* for local source compilation
/*-----
/*
/* Important notice:
/* This JCL is used by REXX FTP procedure mxw_cob_compile_ftp that
/* insert the PROC invocation statement at the end.
/*
/* Copy this JCL into your workspace and tailor it to your needs.
/* Do not change the parameter names.
/*
/* COBOS Project - The Open Source COBOL Development Environment
/* Copyright (C) 2009-2012 METRIXWARE (http://cobos.metrixware.com)
/* All rights reserved.
/*-----
//COMPIL  PROC PGMNAME=, ) SOURCE NAME
//          LOADLIB=          TARGET LOAD LIBRARY
//COBOL    EXEC PGM=IGYCRCTL,PARM='LIB,SIZE(4000K)'
//STEPLIB  DD DSN=IGY420.SIGYCOMP,DISP=SHR          COBOL v4.2
/*          DD DSN=DFH410.CICS.SDFHLOAD,DISP=SHR      CICS v4.1
/*          DD DSN=DSN910.DB9G.SDSNEXIT,DISP=SHR      DB2 v9.1
/*          DD DSN=DSN910.SDSNLOAD,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT5   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT6   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT7   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIN   DD DSN=DSN910.SDSNLOAD,UNIT=SYSAALLDA,
//          DISP=(MOD,PASS),SPACE=(TRK,(3,3)),
//          DCB=(BLKSIZE=3200)
```

*Custload* program requires neither CICS nor SQL. So, you can leave the comment lines in STEPLIB. Check the COBOL STEPLIB and, if necessary adapt the DSNAMES according with the version of Enterprise

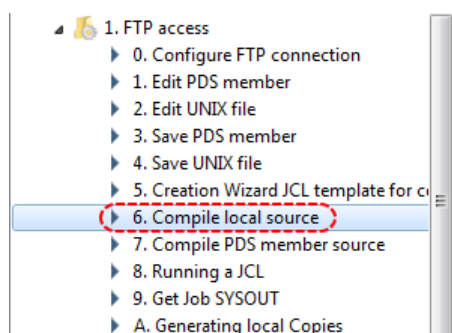
COBOL in use (here IGY420 qualifier for COBOL 4.2 compiler).

```
//SYSPRINT DD SYSOUT=*
//*SYSLIB DD DSN=DFH410.CICS.SDFHCOB,DISP=SHR
//*      DD DSN=ESSAI.JEU230.COPY,DISP=SHR      YOUR COPYLIBS THERE
//*-----
//LINKED EXEC PGM=HEWL,COND=(8,LT,COBOL),REGION=1024K
//SYSLIB DD DSN=CEE.SCEELKED,DISP=SHR          LE LIBRARY
//*      DD DSN=DFH410.CICS.SDFHLOAD,DISP=SHR
//SYSLIN DD DSN=&LOADSET,DISP=(OLD,DELETE)
//SYSLMOD DD DSN=&LOADLIB(&PGMNAME),DISP=SHR
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(TRK,(10,10))
//SYSPRINT DD SYSOUT=*
//      PEND
//*-----
```

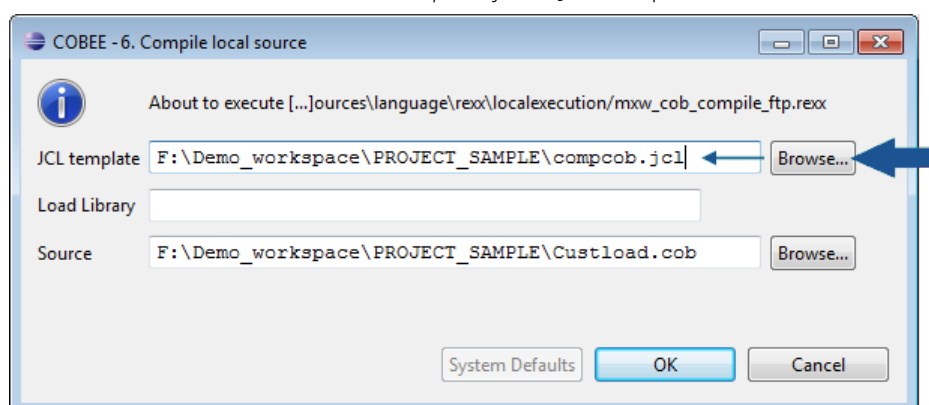
*Custload* program does not require copies and object modules. So, leave the comment lines in SYSLIB (COBOL and LINKED).

3 Select Custload.cob tab ( ).

4 Double-click on **"1. FTP access ► 6. Compile local source"** in the Commands view:



5 Fill in the form fields and select *compcob.jcl* as JCL template with the **"browse"** button:



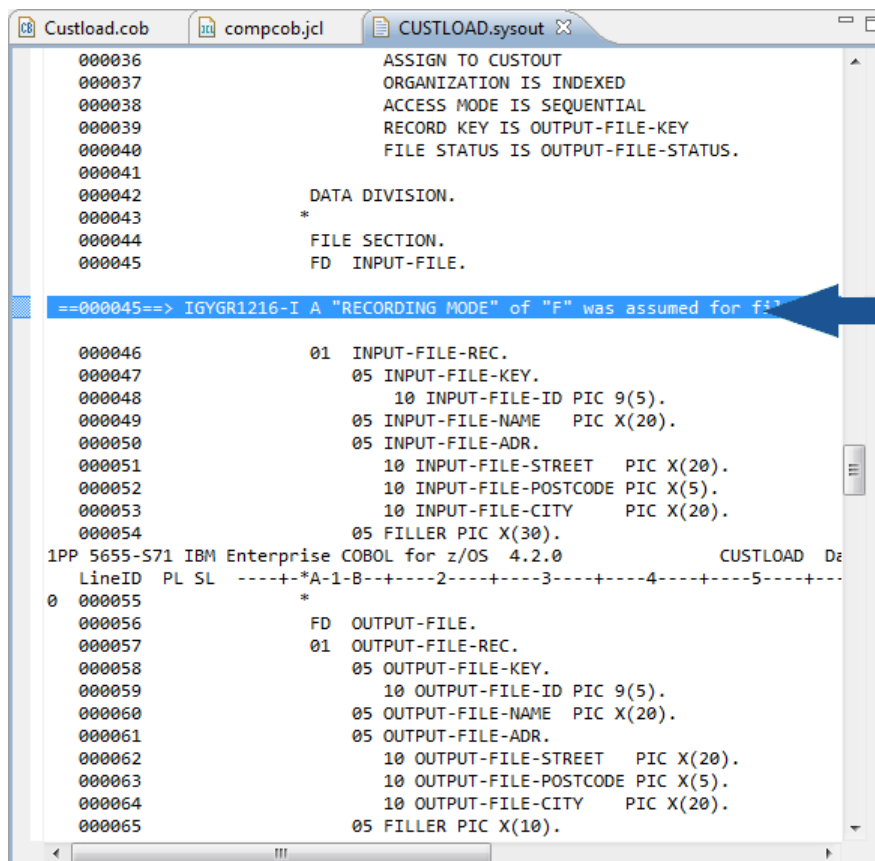
6 Click on **"OK"** button to launch the compilation.

7 See the result in the **"Problems"** view:

COBEE Console Members List Problems X Search				
0 errors, 1 warning, 0 others				
Description	Resource	Path	Location	Type
Warnings (1 item)				
IGYGR1216-IA "RECORDING MODE" of "F" was assumed for file "INPUT	Custload.cob	/PROJECT_SAMPLE	line 45	Compilation Error Marker

You should see a warning at line 45.

- 8 On the warning line, Right-click and select "Open Sysout" in context menu:



```

000036      ASSIGN TO CUSTOUT
000037      ORGANIZATION IS INDEXED
000038      ACCESS MODE IS SEQUENTIAL
000039      RECORD KEY IS OUTPUT-FILE-KEY
000040      FILE STATUS IS OUTPUT-FILE-STATUS.
000041
000042      DATA DIVISION.
000043      *
000044      FILE SECTION.
000045      FD INPUT-FILE.

==000045==> IGYGR1216-IA "RECORDING MODE" of "F" was assumed for f:

000046      01 INPUT-FILE-REC.
000047      05 INPUT-FILE-KEY.
000048          10 INPUT-FILE-ID PIC 9(5).
000049      05 INPUT-FILE-NAME PIC X(20).
000050      05 INPUT-FILE-ADR.
000051          10 INPUT-FILE-STREET PIC X(20).
000052          10 INPUT-FILE-POSTCODE PIC X(5).
000053          10 INPUT-FILE-CITY PIC X(20).
000054      05 FILLER PIC X(30).
1PP 5655-S71 IBM Enterprise COBOL for z/OS 4.2.0 CUSTLOAD De
LineID PL SL ----+*A-1-B--+---2---+---3---+---4---+---5---+---
0 000055      *
000056      FD OUTPUT-FILE.
000057      01 OUTPUT-FILE-REC.
000058      05 OUTPUT-FILE-KEY.
000059          10 OUTPUT-FILE-ID PIC 9(5).
000060      05 OUTPUT-FILE-NAME PIC X(20).
000061      05 OUTPUT-FILE-ADR.
000062          10 OUTPUT-FILE-STREET PIC X(20).
000063          10 OUTPUT-FILE-POSTCODE PIC X(5).
000064          10 OUTPUT-FILE-CITY PIC X(20).
000065      05 FILLER PIC X(10).
    
```

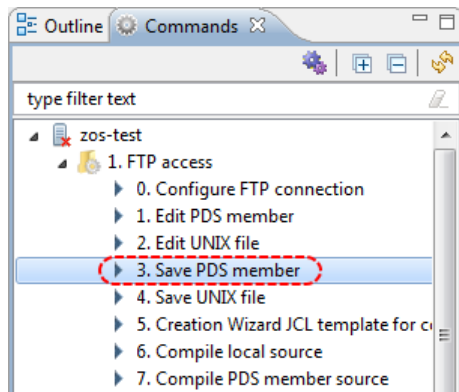
The sysout opens in the text editor and the warning line is selected.

# Editing a PDS member

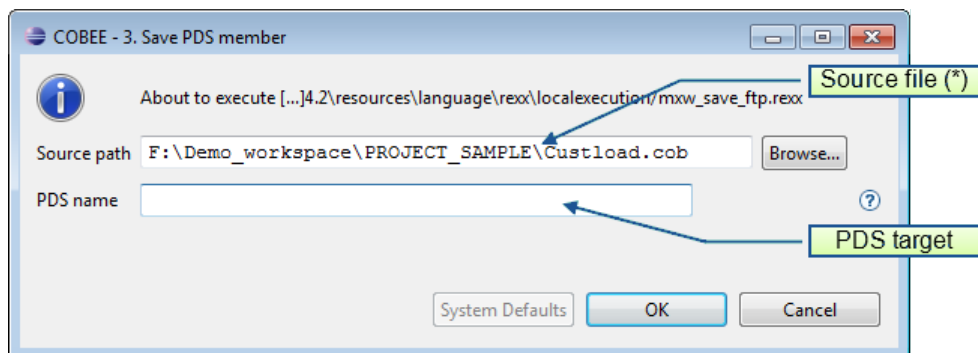
## 8.1 Saving a PDS member

In the PROJECT\_SAMPLE project, open *Custload.cob* in COBOL Editor (if not already done).

- 1 Save *Custload.cob* in a PDS using the COBEE command **"1. FTP access ► 3. Save PDS member"**.

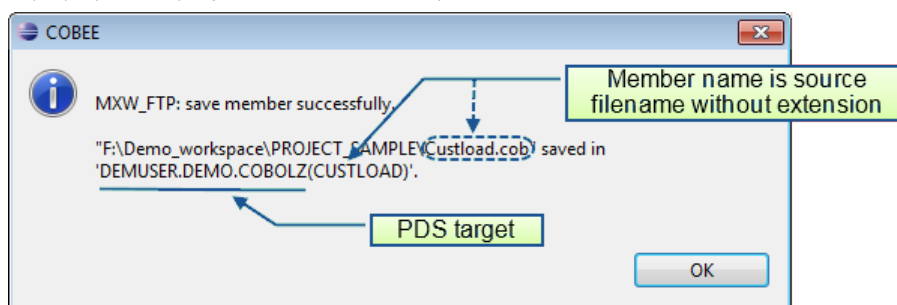


- 2 Fill the form and press **"OK"**.



(\*) Source file name must conform to the naming rules of a PDS member. You can use the "Browse" button to select the file containing the member to save.

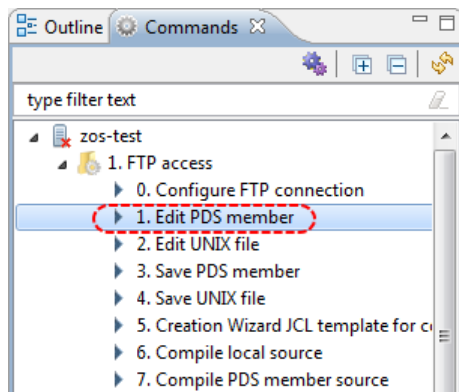
- 3 A popup is displayed at the end of operation.



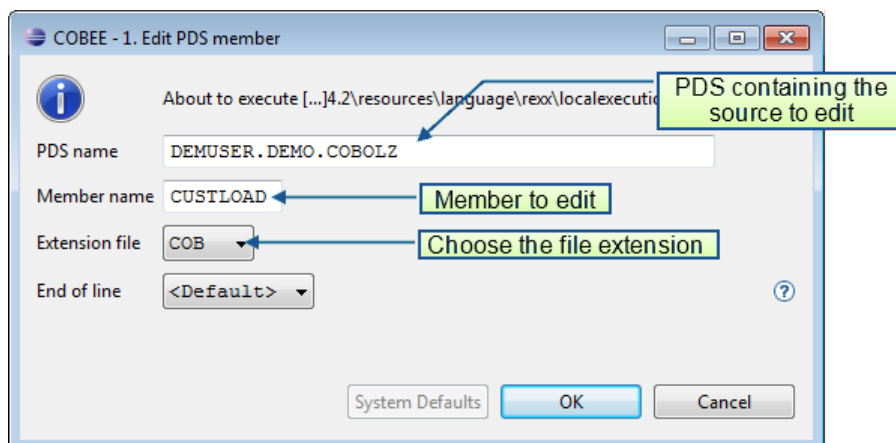
Here PDS target is DEMUSER.DEMO.COBOLZ and the member is CUSTLOAD.

## 8.2 Opening a PDS member

- 1 Open a PDS member of your choice which is a COBOL program using the COBEE command **"1. FTP access ► 1. Edit PDS member"**.



- 2 Fill the form and press **"OK"**:



Here we edit the member "CUSTLOAD" in the PDS "DEMUSER.DEMO.COBOLZ" previously saved.

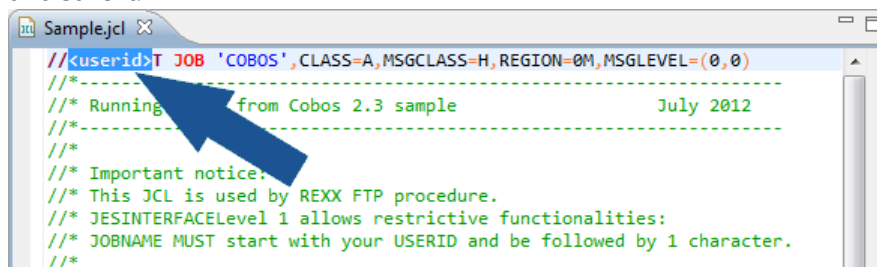


## 9. Managing a job

We need to submit a job and retrieve the result of its execution. We'll show you how to do it with an easy example.

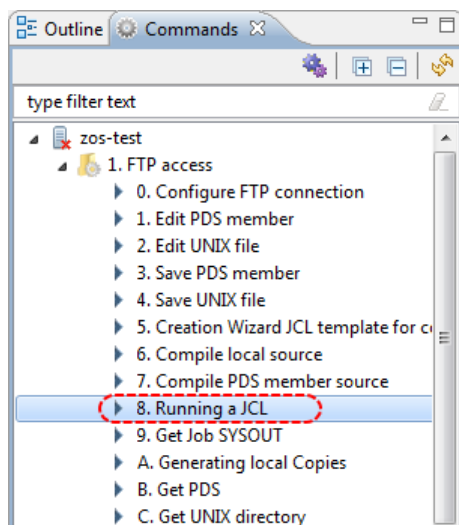
### 9.1 Submitting a jcl

- 1 Edit the file *Sample.jcl*, adapt the JOB card (replace <userid> by your mainframe's user id in uppercase) and save it.

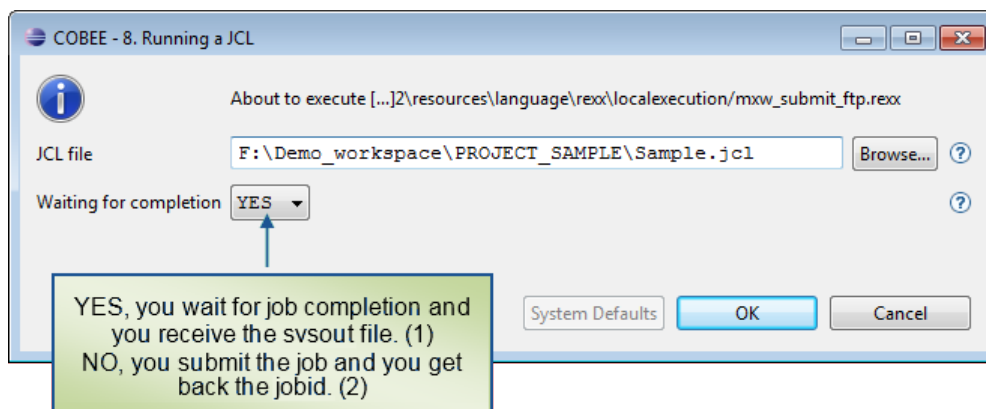


```
//<userid>T JOB 'COBOS',CLASS=A,MSGCLASS=H,REGION=0M,MSGLEVEL=(0,0)
/*-----
/* Running from Cobos 2.3 sample          July 2012
/*-----
/*
/* Important notice:
/* This JCL is used by REXX FTP procedure.
/* JESINTERFACELevel 1 allows restrictive functionalities:
/* JOBNAME MUST start with your USERID and be followed by 1 character.
/*
```

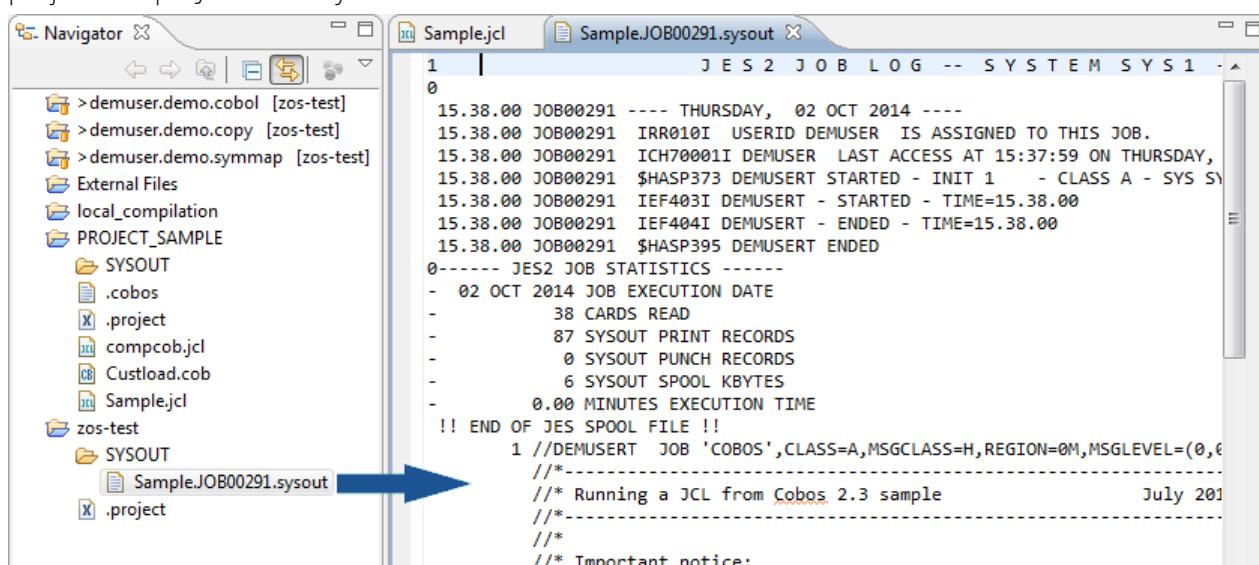
- 2 Submit the JCL by using the COBEE command **"1. FTP access ► 8. Running a JCL"**.



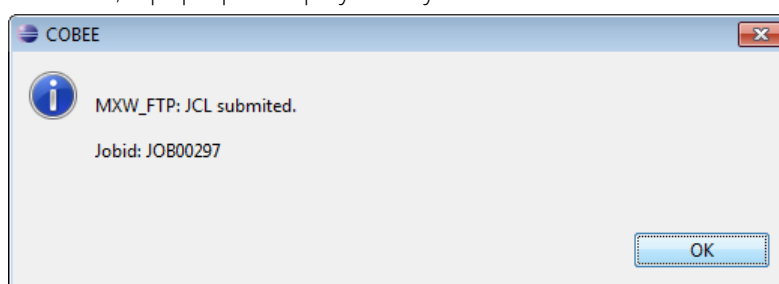
- 3 Fill the form and press **"OK"**:



- (1) The filename is <jclfilename>.<jobid>.sysout in the project <hostname>, folder SYSOUT. E.g.: Sample.jcl → Sample.JOB00291.sysout



- (2) If no wait, a pop-up is displayed so you can check status and retrieve output later (see below).

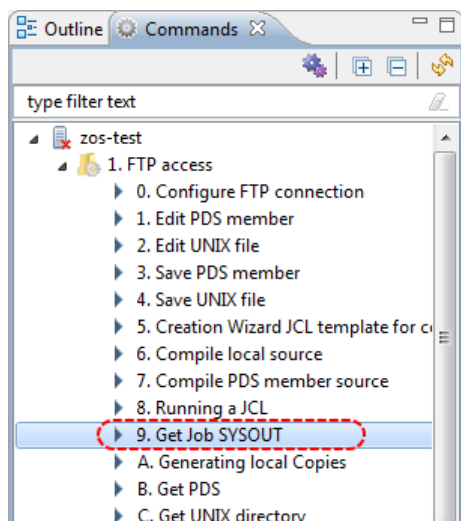


Here Jobid is JOB00297.

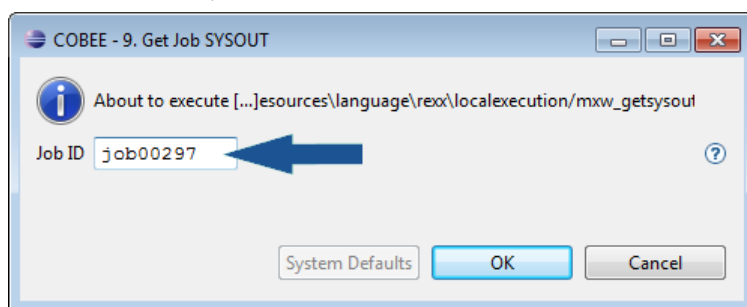
## 9.2 Retrieving a sysout

We'll get the sysout of the previously submitted job.

- 1 Retrieve the sysout of job by using the COBEEcommand **"1. FTP access ► 9. Get Job SYSOUT"**.

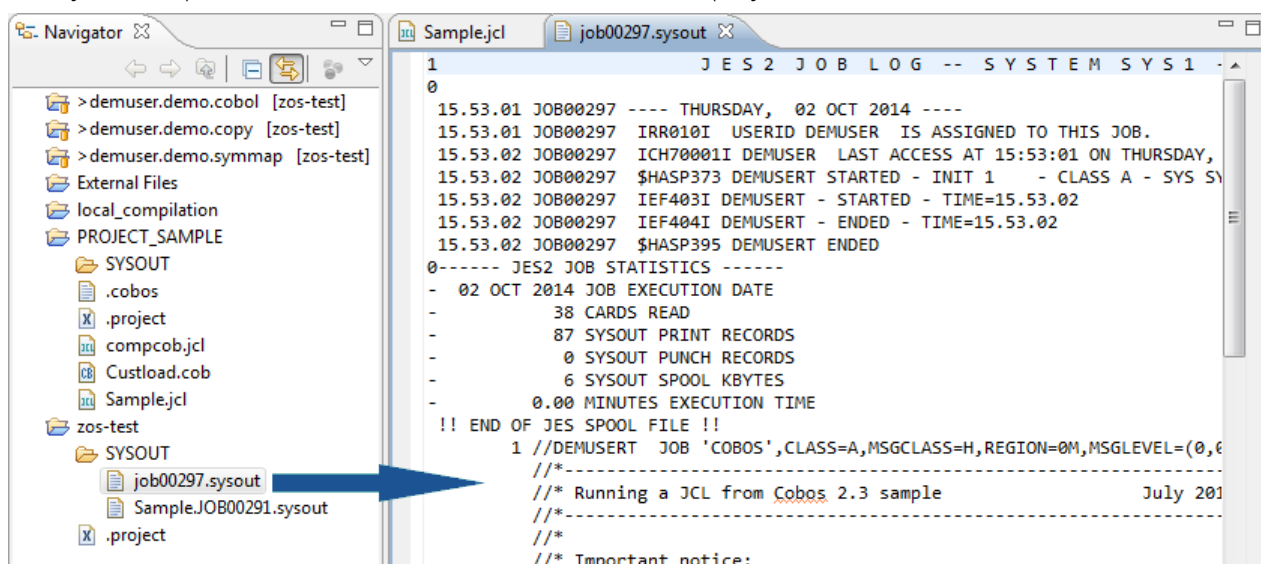


- 2 Fill the form and press "OK":



Here we want to retrieve the sysout of job00297.

- 3 The sysout is opened in the Text Editor and is stored in the project <hostname>, folder *SYSOUT*.

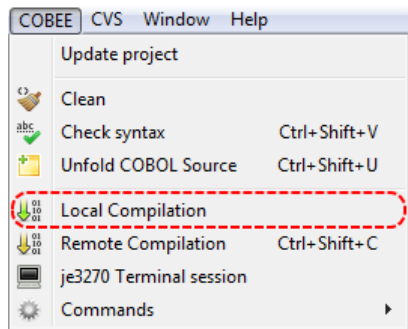


# 10. Local Compilation

## 10.1 COLORS program

- 1 Open the program "colors.cbl" from "Local\_compilation" project with the COBOL Editor by right-clicking on it in the "Navigator" view and by choosing **"Open With ► COBOL Editor"**.

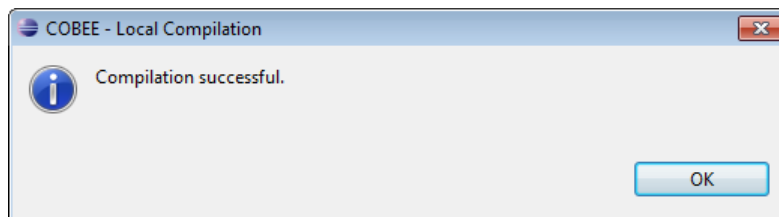
- 2 Select menu **"COBEE ► Local Compilation"**



or click on the "Local Compilation" button placed in the toolbar

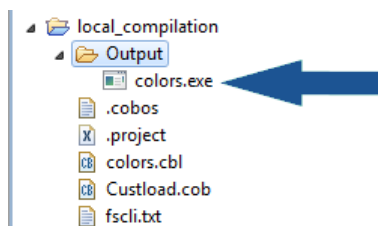


- 3 Once the compilation has been achieved, this popup is displayed.

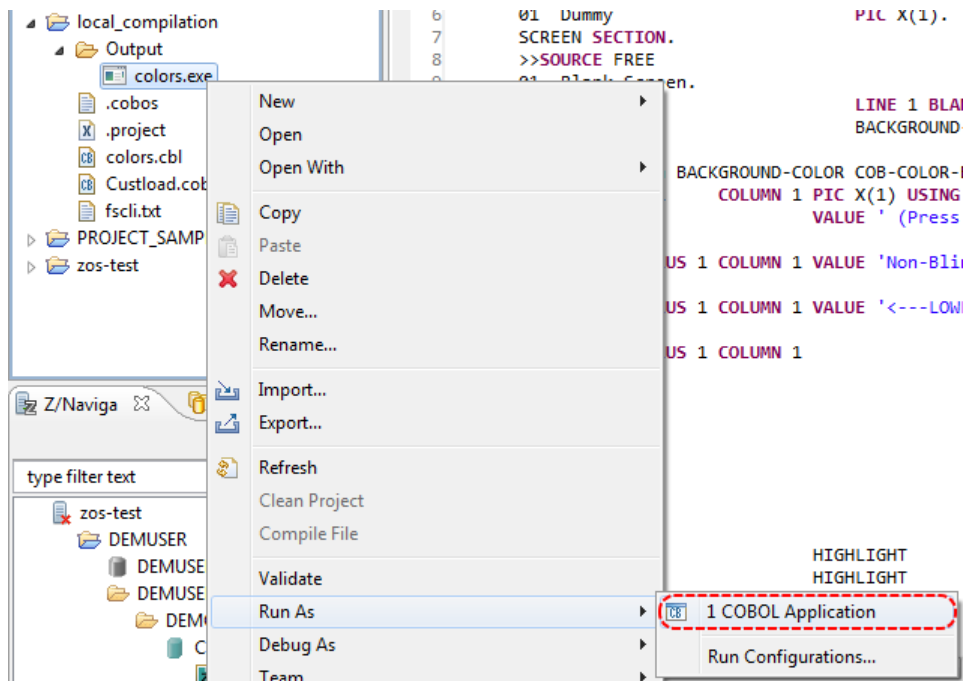


Just click on "OK" button.

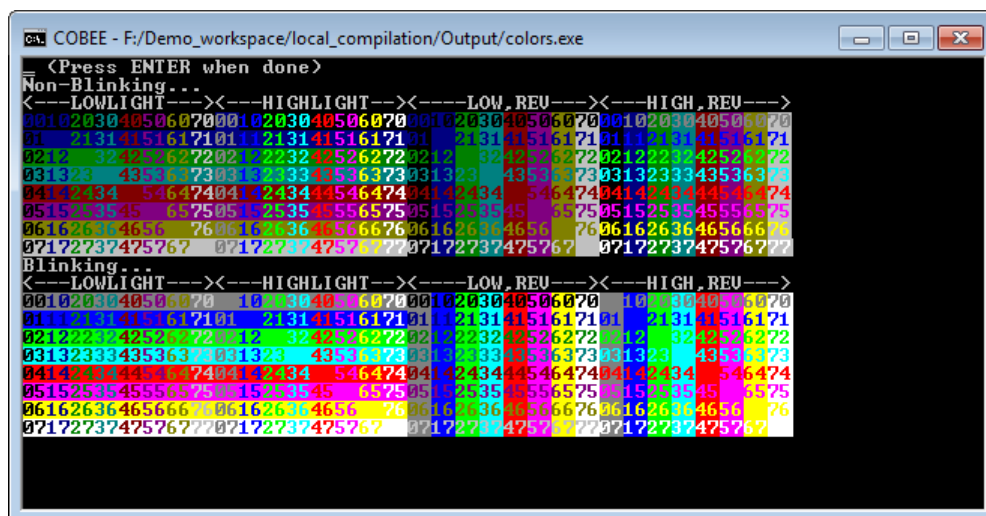
- 4 In the "Navigator" view, you should see the executable "colors.exe" in "Output" folder.



- 5 Right-click on "colors.exe" file and select **"Run As ► COBOL Application"**



6 A console opens in which the program runs <sup>1</sup>.



Press ENTER to quit the program.

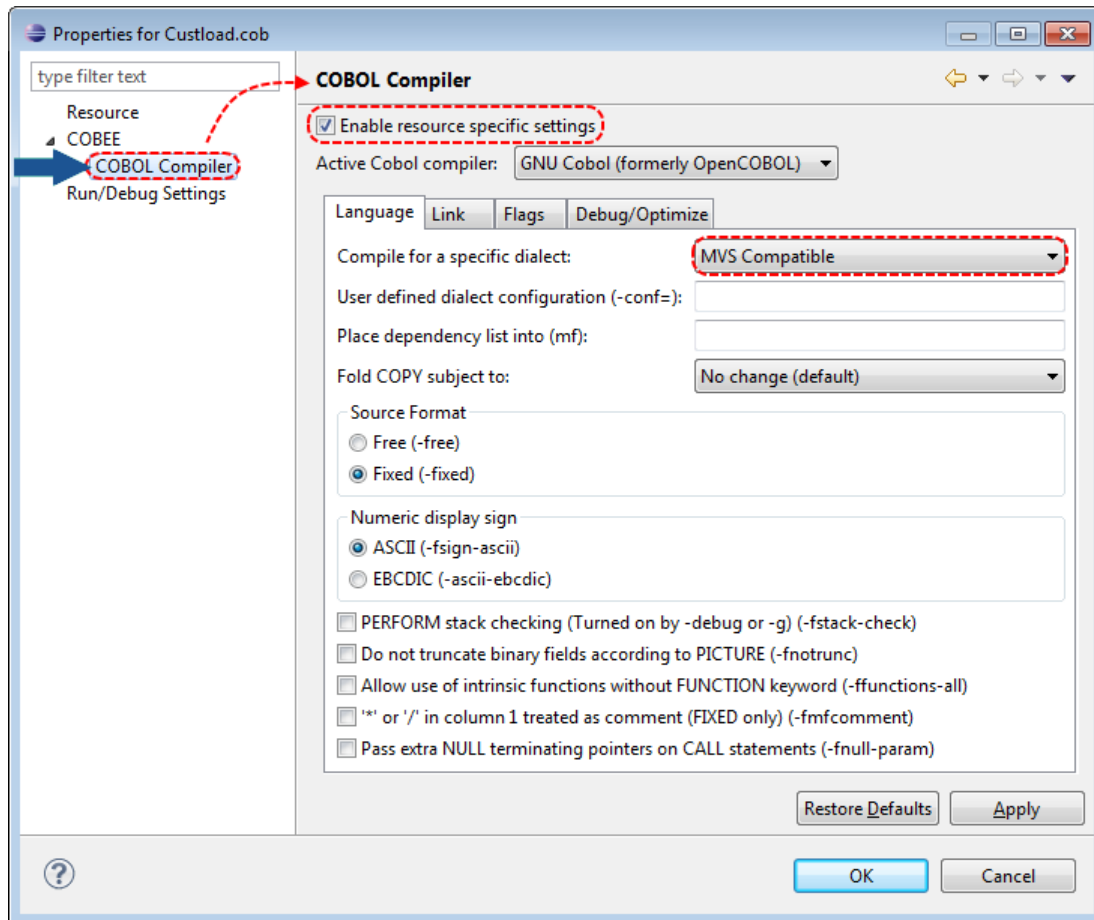
7 Press any key to close the console.

## 10.2 CUSTLOAD program

1 In "Navigator" view, select "Custload.cob" from "Local\_compilation" project, right-click and select "Properties"

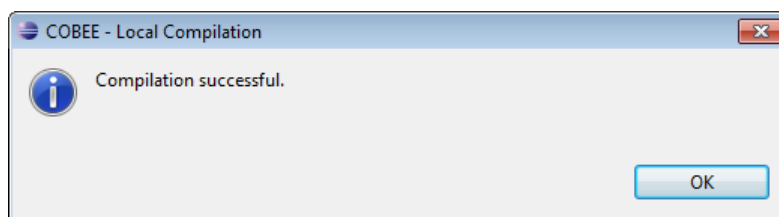
<sup>1</sup>The BLINK attribute modifies the visual appearance of the BACKGROUND-COLOR specification. The Windows console does not support blinking, so the visual effect of BLINK in the Windows version of OpenCOBOL is to provide the same sixteen colors to the BACKGROUND-COLOR palette as are possible with FOREGROUND-COLOR combined with LOWLIGHT/HIGHLIGHT." [OpenCOBOL-1.1-06FEB2009-Programmers-Guide]

- 2 Select **"COBEE ► COBOL Compiler"** and check **"Enable resource specific settings"**



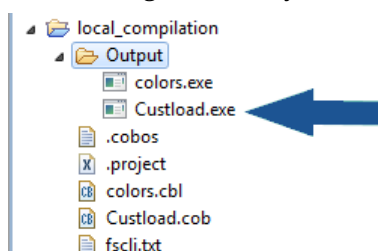
In "Language" tab, select **"MVS Compatible"** for a specific dialect and push "OK" button.

- 3 Open the program "Custload.cob" from "Local\_compilation" project with the COBOL Editor by right-clicking on it in the "Navigator" view and by choosing **"Open With ► COBOL Editor"**.
- 4 In the editor, Right-click and select **"COBEE ► Local Compilation"**.
- 5 Once the compilation has been achieved, this popup is displayed.



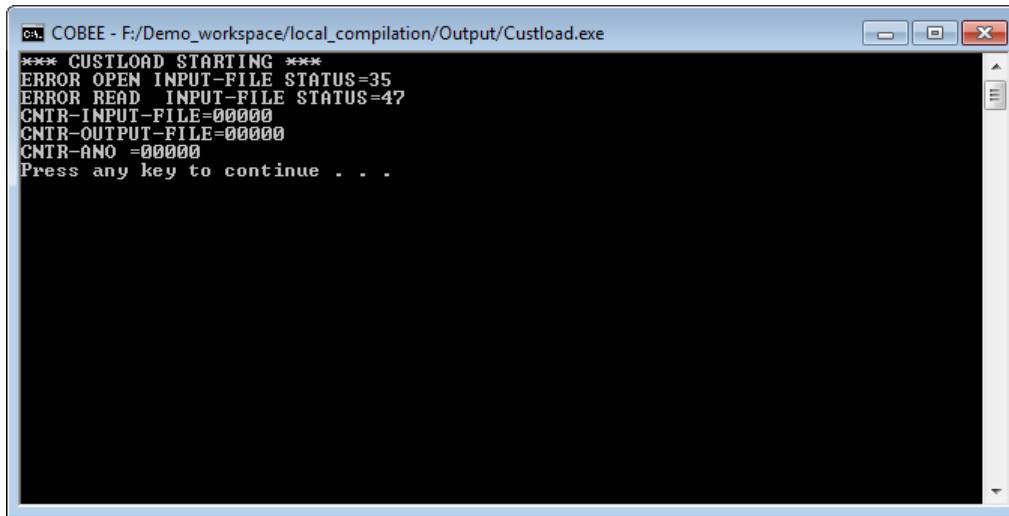
Just click on "OK" button.

- 6 In the "Navigator" view, you should see the executable "Custload.exe" in "Output" folder.



7 Right-click on “Custload.exe” file and select **“Run As ► COBOL Application”**.

8 A console opens in which the program runs.



The program exits<sup>2</sup> with INPUT-FILE STATUS = 35.

9 Press any key to close the console.

10 Modify the program to absorb the end of line characters contained in the input file. Under Windows, the end of line consists of two characters: CR & LF. So, in line 54, replace **X(10)** by **X(12)**.

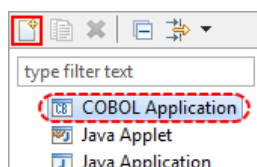
```

44      FILE SECTION.
45      FD INPUT-FILE.
46      01 INPUT-FILE-REC.
47          05 INPUT-FILE-KEY.
48              10 INPUT-FILE-ID PIC 9(5).
49          05 INPUT-FILE-NAME PIC X(20).
50          05 INPUT-FILE-ADR.
51              10 INPUT-FILE-STREET PIC X(20).
52              10 INPUT-FILE-POSTCODE PIC X(5).
53              10 INPUT-FILE-CITY PIC X(20).
54          05 FILLER PIC X(12).
55      *
```

11 Save and compile the program again as in step 4.

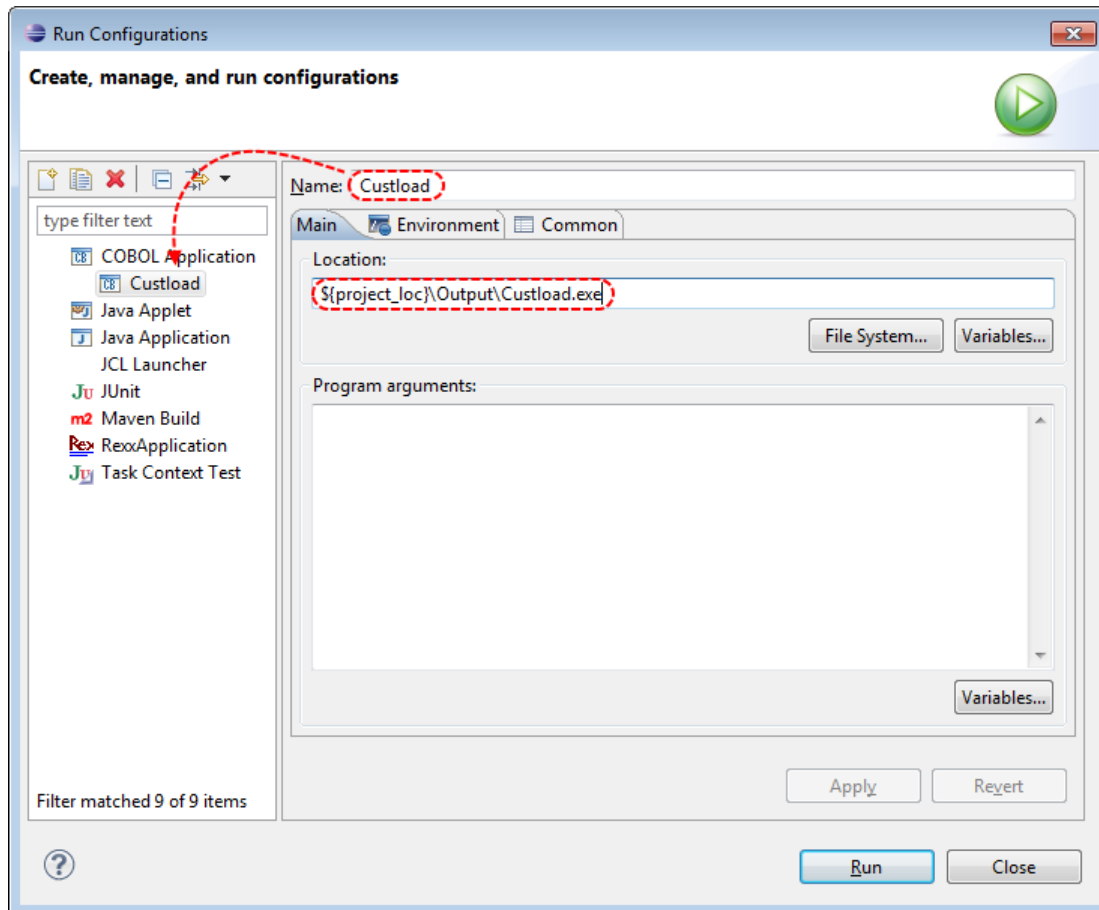
12 Right-click on “Custload.exe” file and select **“Run As ► Run Configuration...”**.

13 Select **“COBOL Application”** and Press the **“New”** button to create a new configuration.

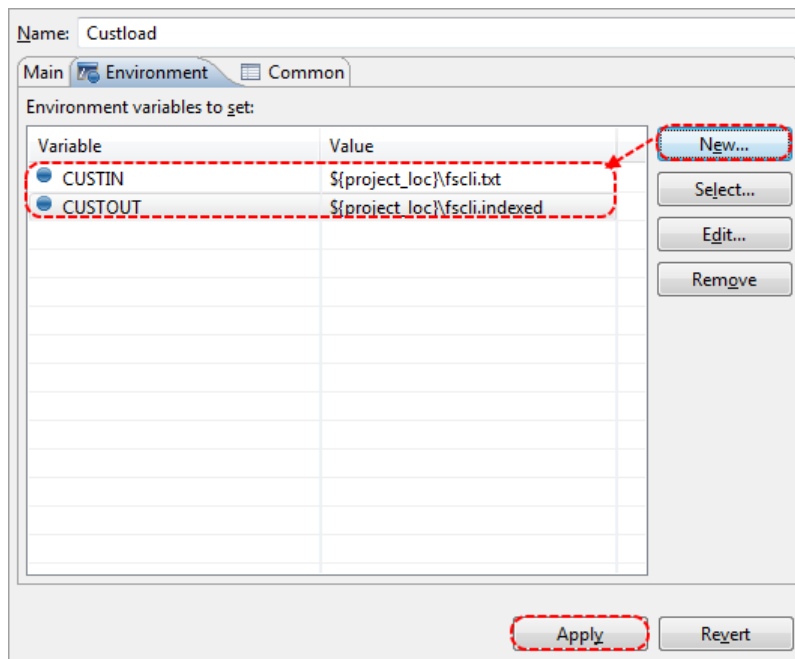


14 Name the configuration and select Custload.exe in **“Main”** tab.

<sup>2</sup> Sometimes the program hangs. Press Ctrl+C to stop it immediately.



- 15 Select **"Environment"** tab and add 2 variables CUSTIN and CUSTOUT as shown in the following window:



Click on **"Apply"** button to save the configuration.

- 16 Push **"Run"** button to execute the program.



```

COBEE - F:\Demo_workspace\local_compilation\Output\Custload.exe
*** CUSTLOAD STARTING ***
NON NUMERIC KEY :A0004
A0004La plume de Moliere 18,av gal Leclerc 69000Lyon FR03885522

KEY NOT IN SEQUENCE :00001
00001Librairie St-Martin 151,r Fbrg St Martin75008Paris FR01146804

CNTR-INPUT-FILE=00047
CNTR-OUTPUT-FILE=00045
CNTR-ANO =00002
Press any key to continue . . . _
    
```

Result: 47 input records, 45 output records, 2 anomalies

**17** Press any key to close the console.

**18** In "Navigator" view, refresh "Local\_compilation" project and check the presence of 'fscli.indexed'

