Cobos Quick Start

version 3.5

Revision: 018
Date: 10/20/2014

Reference

Date	Description	Writer
	Cobos User Guide	Metrixware
1st Edition, 17 September 2010	OpenCOBOL-1.1-06FEB2009-Programmers-Guide	Gary Cutler

Copyright

Contents

1. Introduction

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

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 Cobos 3.5:

- Cobos 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 Cobos 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)[1].

Resources:

You downloaded **Cobos_3.5.x_Release-demo.zip** or **Cobos_3.5.x_Essentials-demo.zip**rom the Cobos 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
- **Cobos_V3.5_DemoReadme.txt**
- **Cobos_V3.5_Quickstart.pdf**: this file!
- **Cobos_V3.5_User_Guide.pdf**
- **releaseNotes.txt**

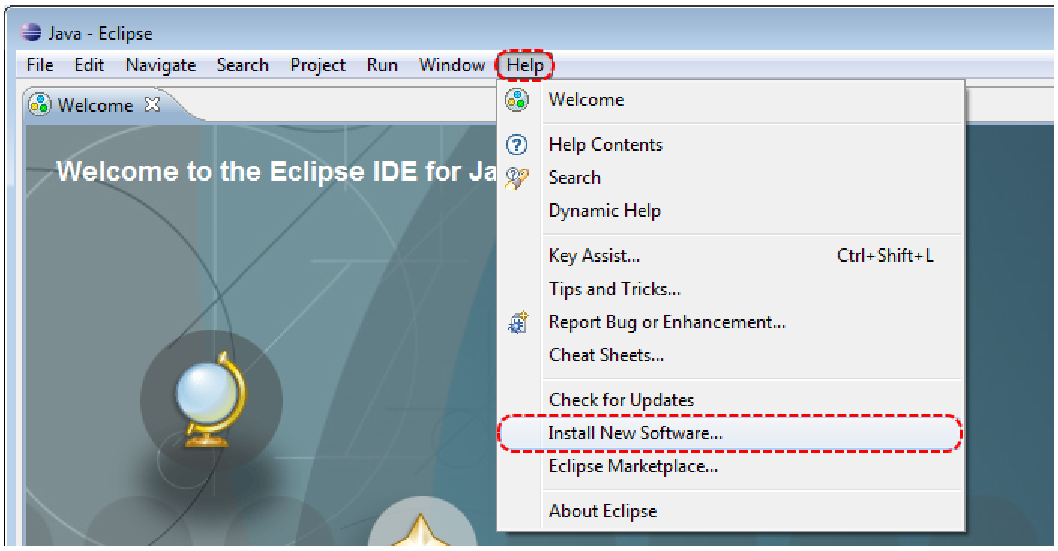
[1] For the users of Open Object Rexx (ooRexx), some scripts of Cobos are not fully compatible.

2. Installation

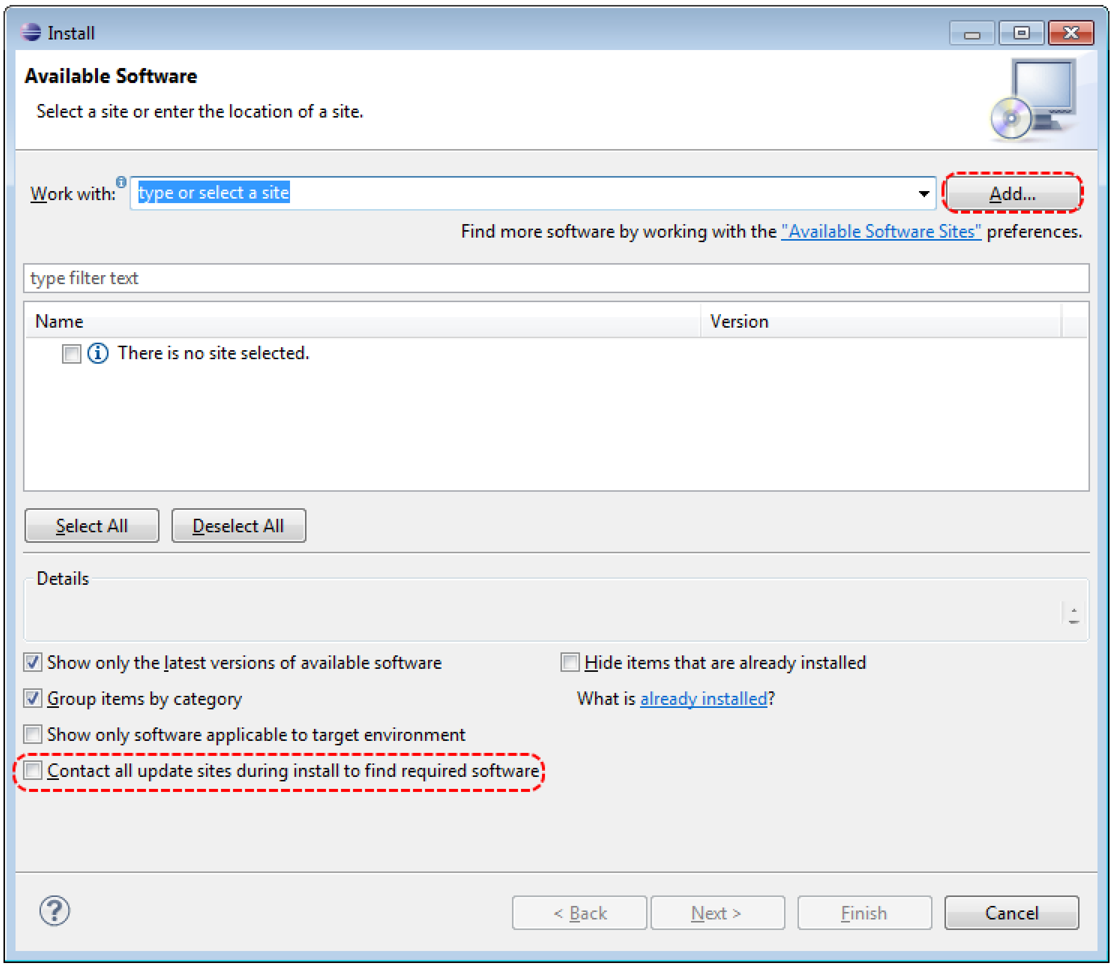
Retrieve Eclipse for Windows from http://www.eclipse.org/downloads.
Note: This document has been produced with Eclipse Indigo. The examples also work with Eclipse versions listed in the prerequisites.

1 Launch Eclipse and select a new workspace (temporary used for Cobos 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.



Install

Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☐ There is no site selected.	

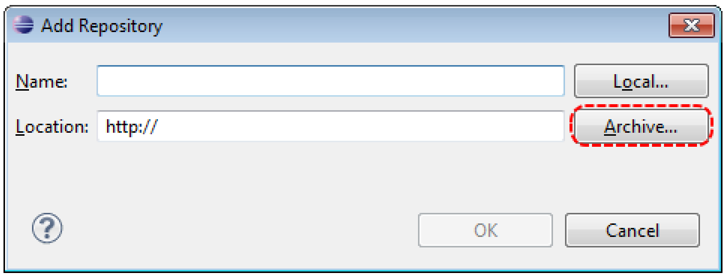
Details

- [✓] Show only the latest versions of available software
- [✓] Group items by category
- [ ] Show only software applicable to target environment
- [ ] Contact all update sites during install to find required software
- [ ] Hide items that are already installed

What is already installed?

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 Cobos plug-in are stored and choose one archive file containing plug-ins to be installed.

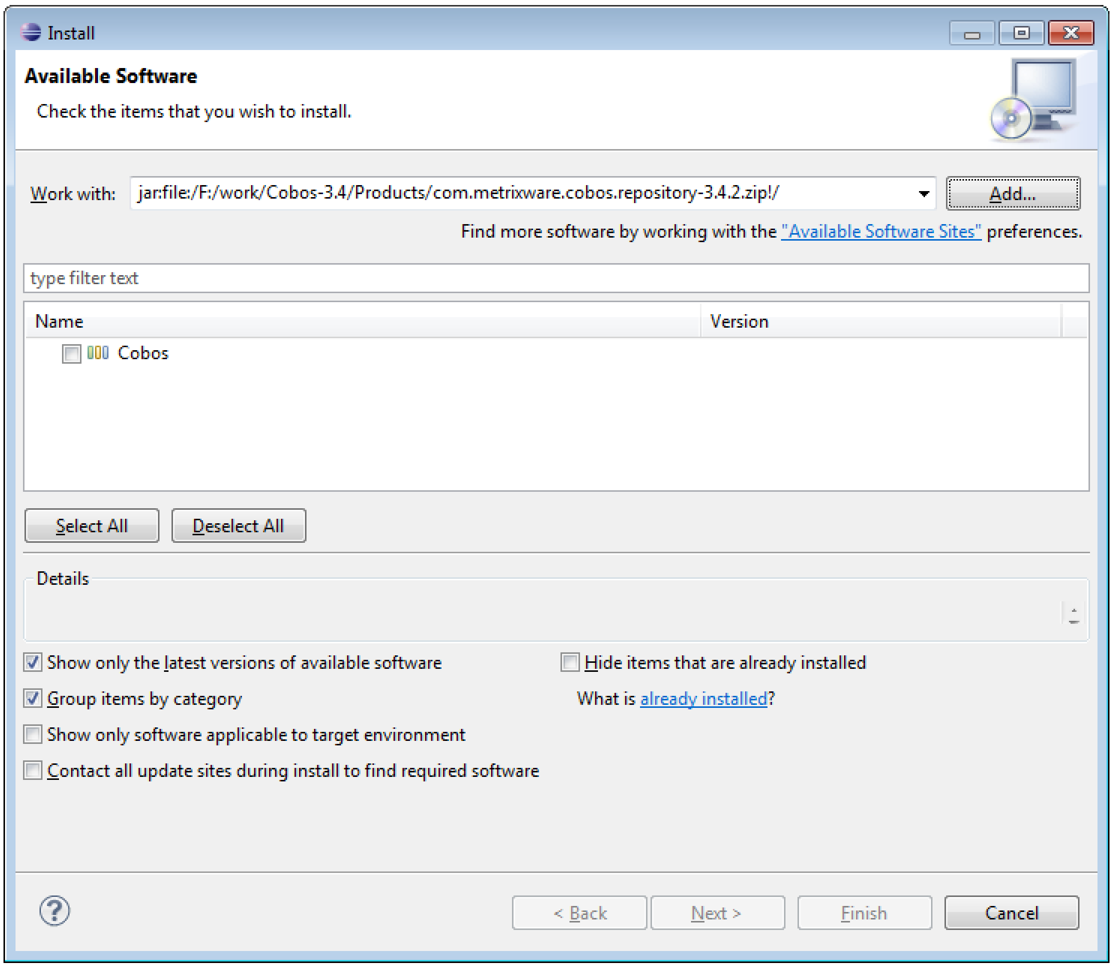


Add Repository

Name:

Location: http://

The component appear alone in the window:


Install

Available Software

Check the items that you wish to install.

Work with: jar:file:/F:/work/Cobos-3.4/Products/com.metrixware.cobos.repository-3.4.2.zip!/ [Add...]

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☐ Cobos	

[Select All] [Deselect All]

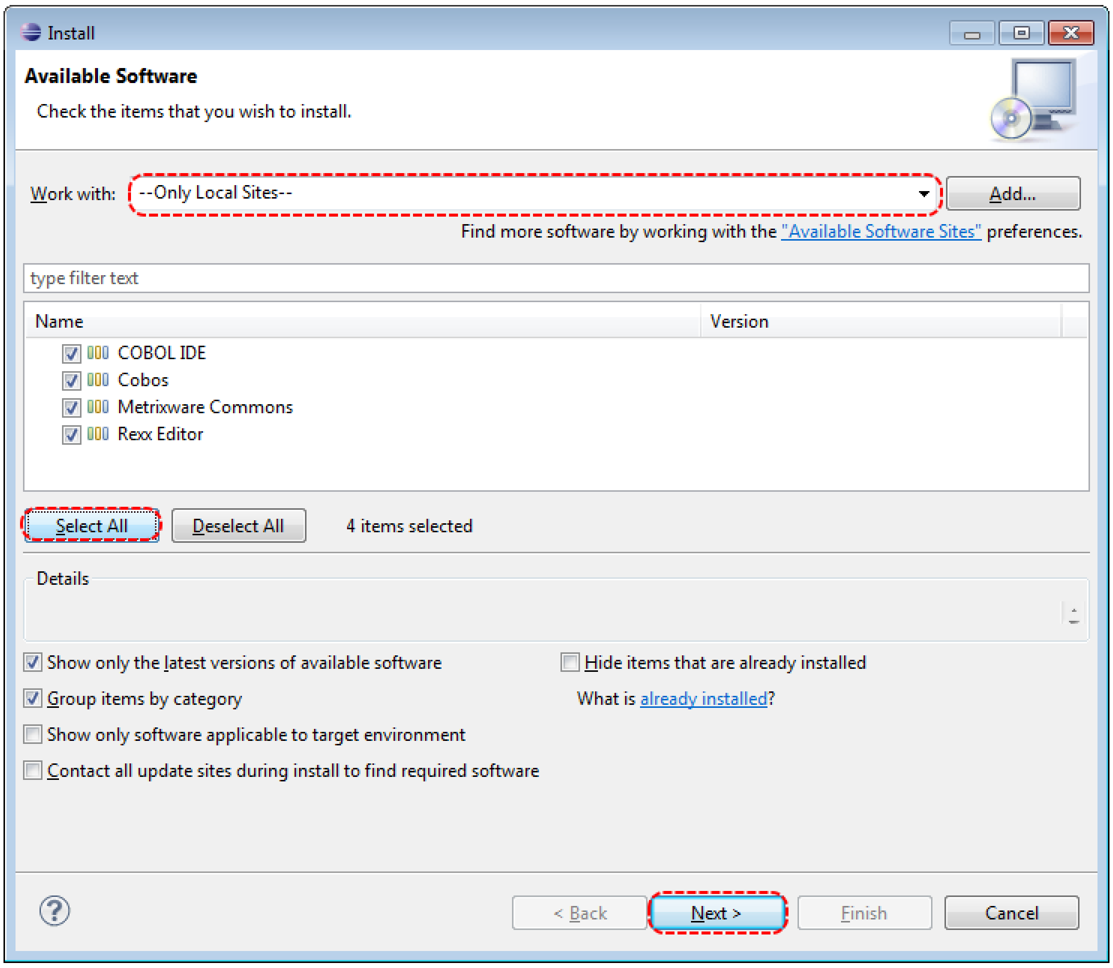
Details

☑ Show only the latest versions of available software
☐ Hide items that are already installed
☑ Group items by category
What is already installed?
☐ Show only software applicable to target environment
☐ Contact all update sites during install to find required software

[< Back] [Next >] [Finish] [Cancel]

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.



Install

Available Software

Check the items that you wish to install.

Work with: --Only Local Sites--

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ COBOL IDE	
☑ Cobos	
☑ Metrixware Commons	
☑ Rexx Editor	

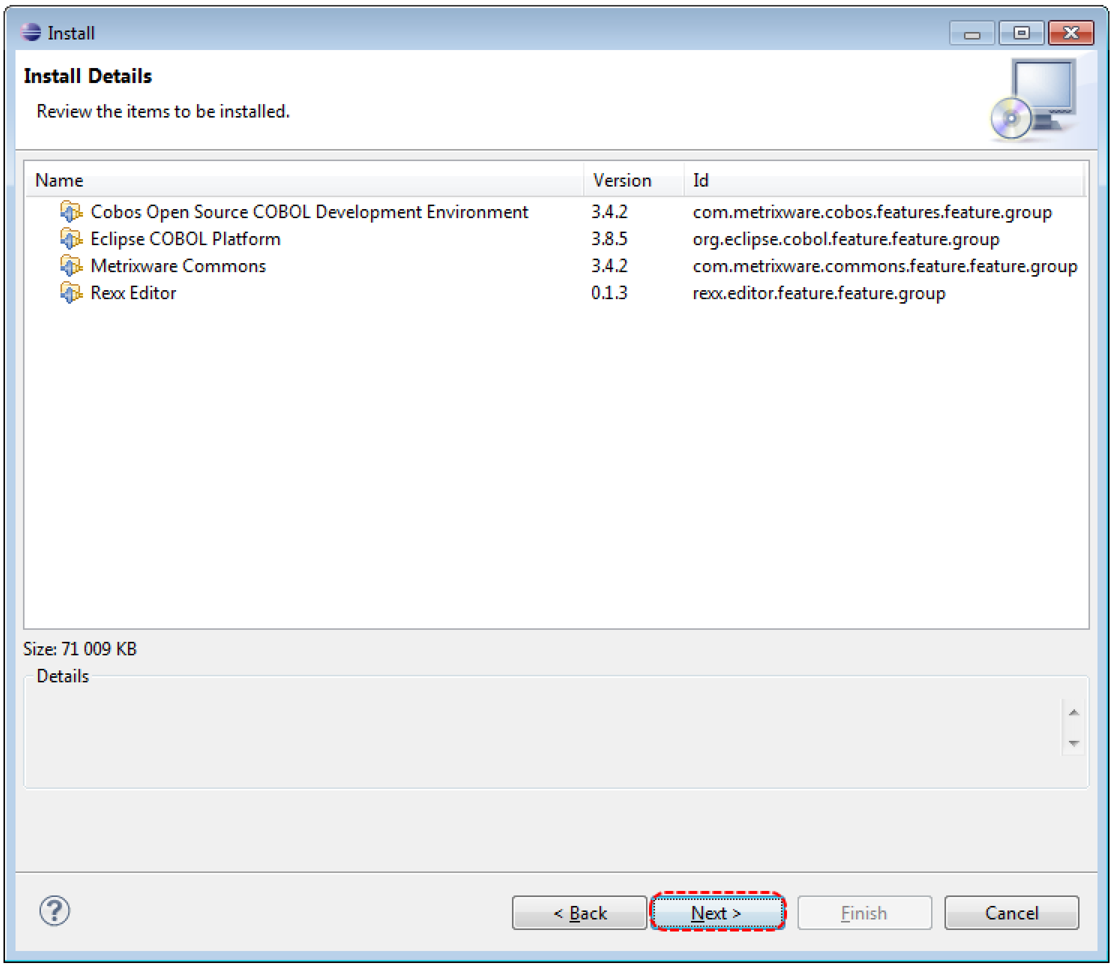
4 items selected

Details

☑ Show only the latest versions of available software
☐ Hide items that are already installed
☑ Group items by category
What is already installed?
☐ Show only software applicable to target environment
☐ Contact all update sites during install to find required software

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:


Install

Install Details

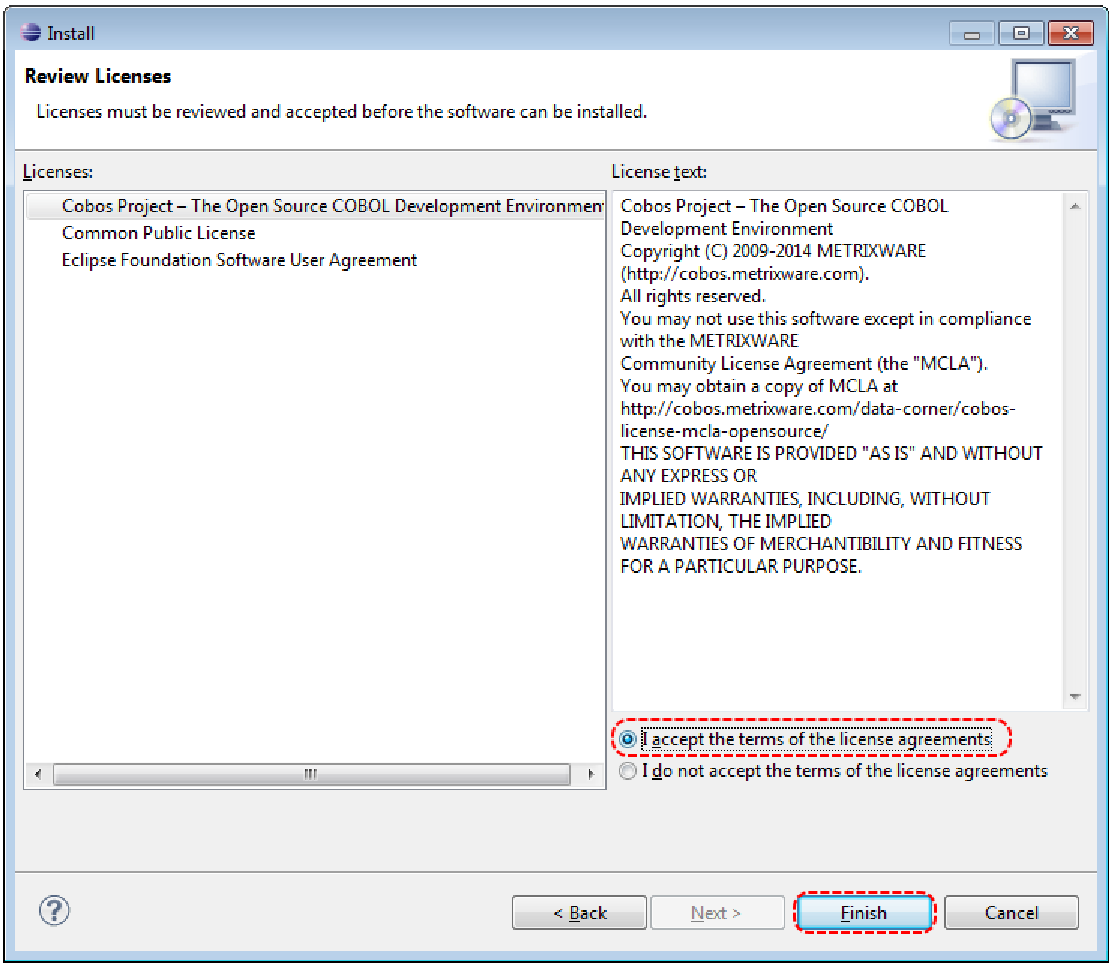
Review the items to be installed.

Name	Version	Id
Cobos Open Source COBOL Development Environment	3.4.2	com.metrixware.cobos.features.feature.group
Eclipse COBOL Platform	3.8.5	org.eclipse.cobol.feature.feature.group
Metrixware Commons	3.4.2	com.metrixware.commons.feature.feature.group
Rexx Editor	0.1.3	rexx.editor.feature.feature.group

Size: 71 009 KB

Details

7 Accept the terms of the license agreements and click on **“Finish”** button.



Install

Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses:

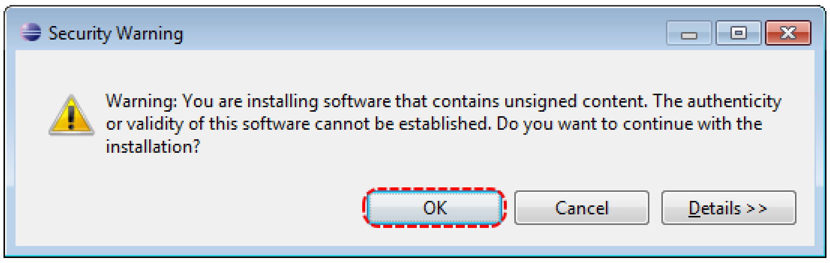
- Cobos Project – The Open Source COBOL Development Environment
- Common Public License
- Eclipse Foundation Software User Agreement

License text:

Cobos Project – The Open Source COBOL Development Environment
Copyright (C) 2009-2014 METRIXWARE
(http://cobos.metrixware.com).
All rights reserved.
You may not use this software except in compliance with the METRIXWARE
Community License Agreement (the "MCLA").
You may obtain a copy of MCLA at
http://cobos.metrixware.com/data-corner/cobos-license-mcla-opensource/
THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

[✓ I accept the terms of the license agreements]
I do not accept the terms of the license agreements

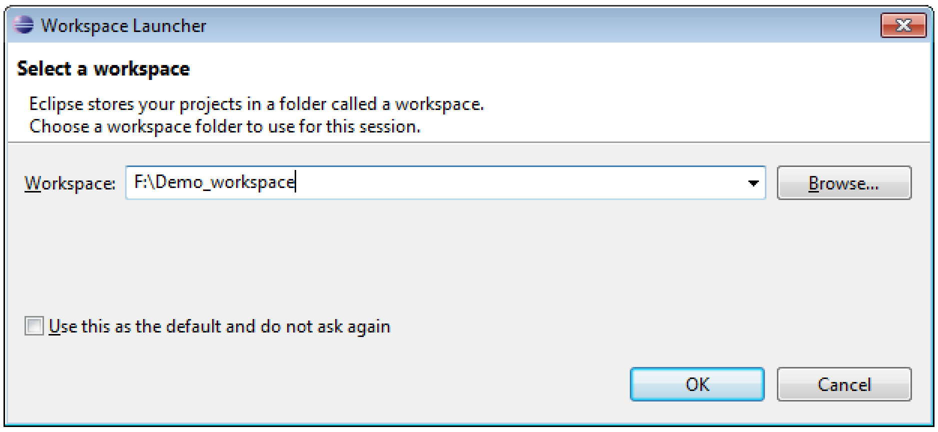
8 It is possible that a security warning appears during the installation phase. Click on **"OK"** button to continue the installation.



Security Warning

Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?

9 When the installation is finished, restart Eclipse and **select the Demo_workspace directory that you have unzipped from Cobos_3.5.x_Release-demo.zip file**.

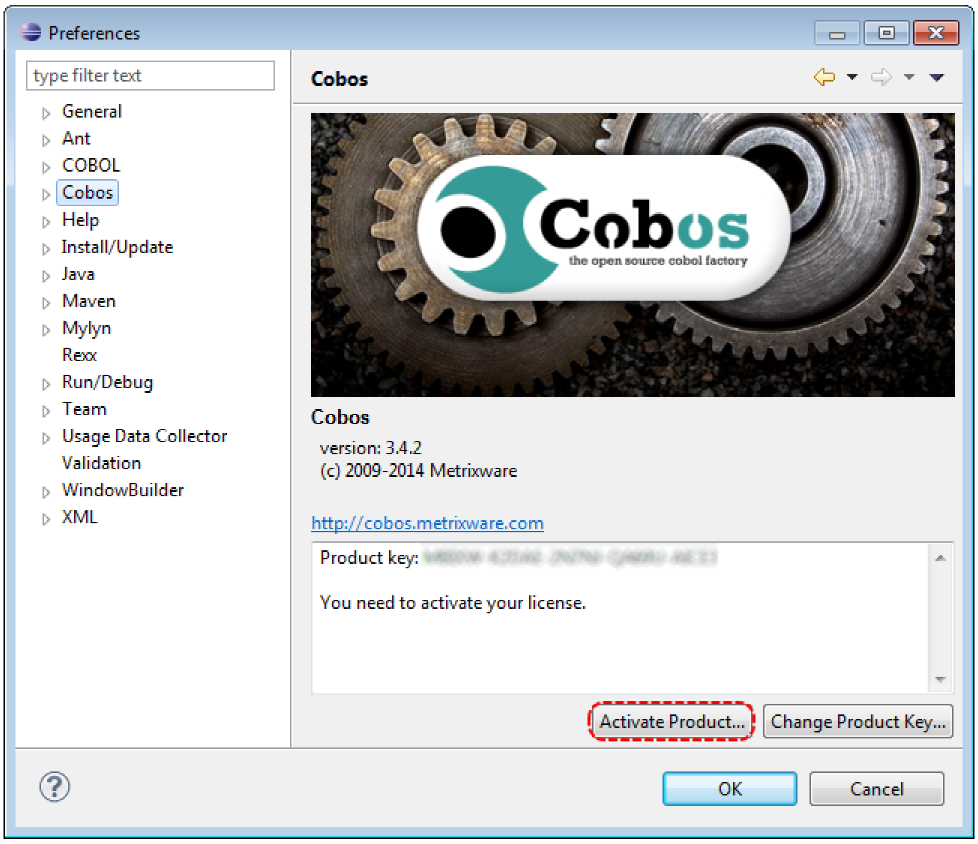


Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: F:\Demo_workspace

10 Go to **"Window ► Preferences ► Cobos"** and click on the **"Activate Product..."** button.



Cobos

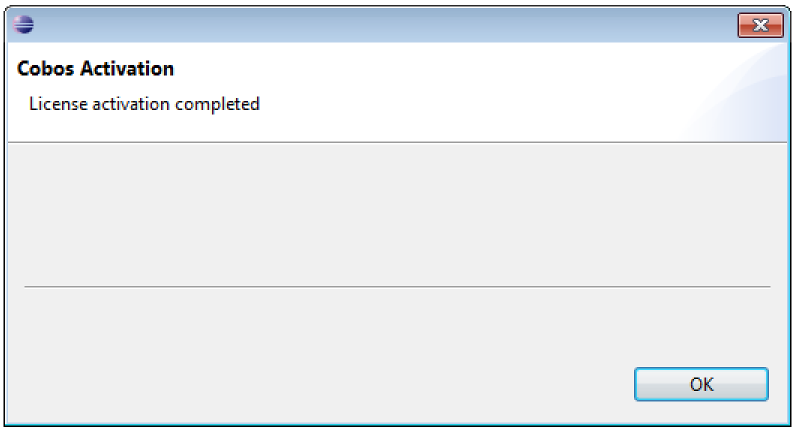
version: 3.4.2
(c) 2009-2014 Metrixware

http://cobos.metrixware.com

Product key: [redacted]

You need to activate your license.

11 A popup indicates that the license is activated

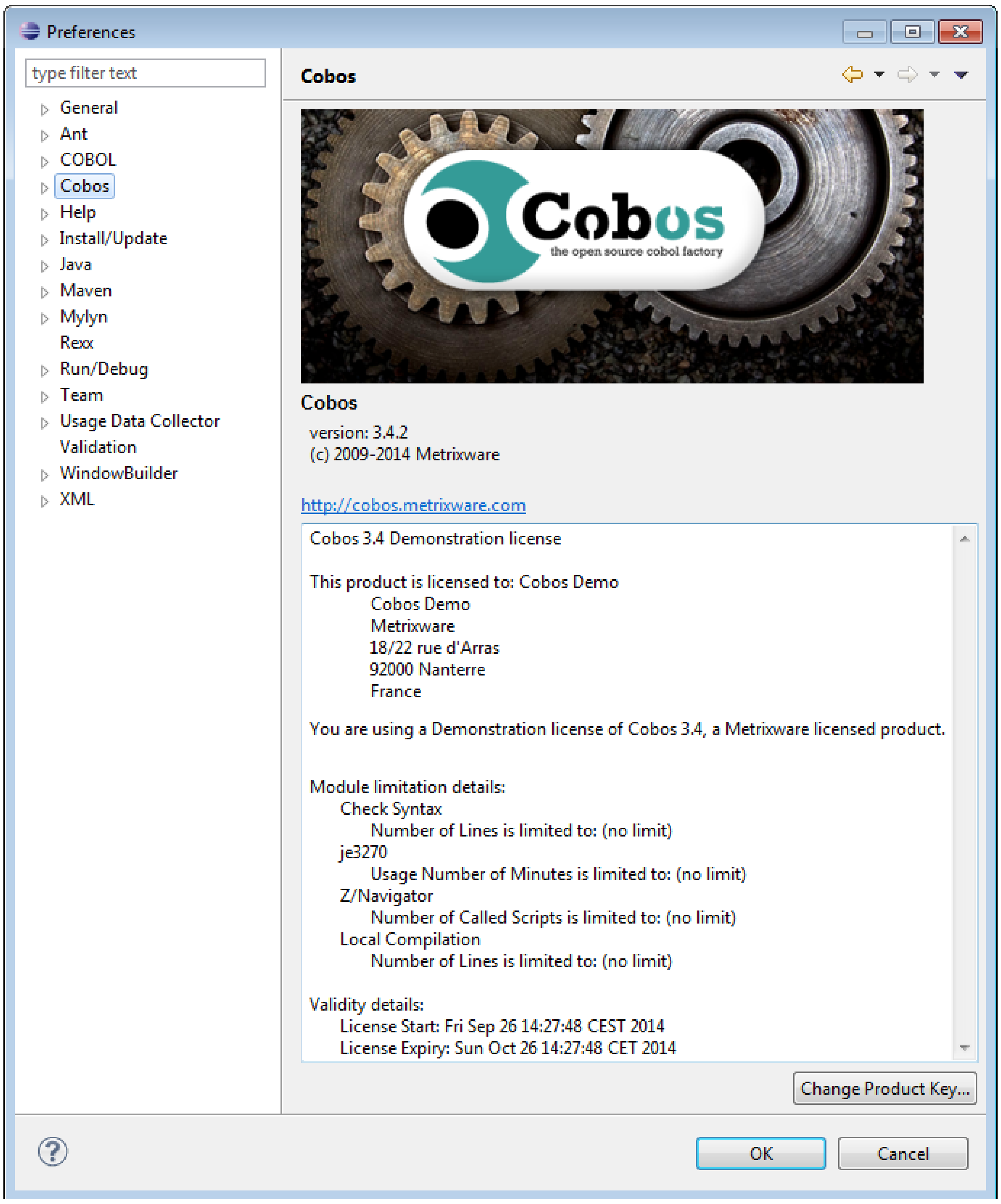


Cobos Activation

License activation completed

Just click OK

12 The Demo license is displayed.



Cobos

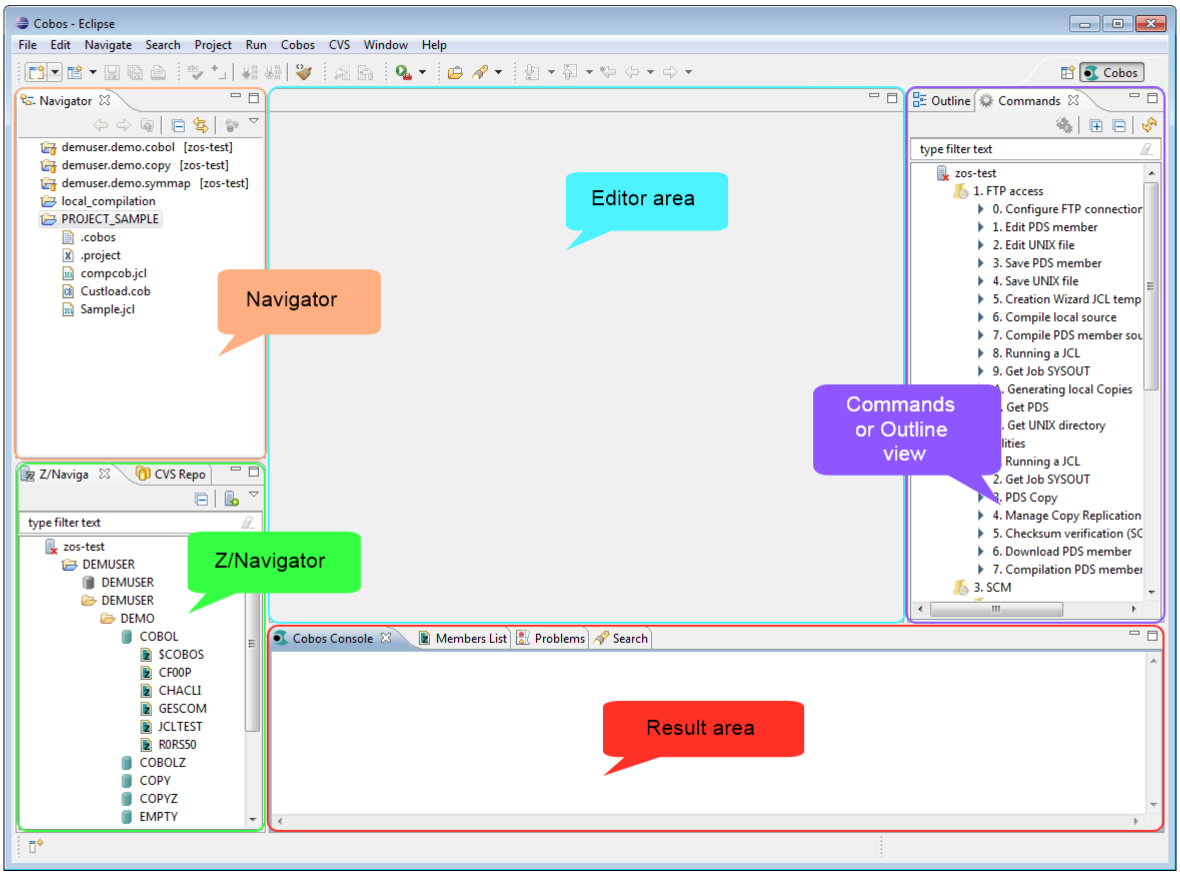
version: 3.4.2
(c) 2009-2014 Metrixware

http://cobos.metrixware.com

```
Cobos 3.4 Demonstration license

This product is licensed to: Cobos Demo
            Cobos Demo
            Metrixware
            18/22 rue d'Arras
            92000 Nanterre
            France

You are using a Demonstration license of Cobos 3.4, a Metrixware licensed product.


Module limitation details:
      Check Syntax
            Number of Lines is limited to: (no limit)
      je3270
            Usage Number of Minutes is limited to: (no limit)
      Z/Navigator
            Number of Called Scripts is limited to: (no limit)
      Local Compilation
            Number of Lines is limited to: (no limit)

Validity details:
      License Start: Fri Sep 26 14:27:48 CEST 2014
      License Expiry: Sun Oct 26 14:27:48 CET 2014
```

Click OK.

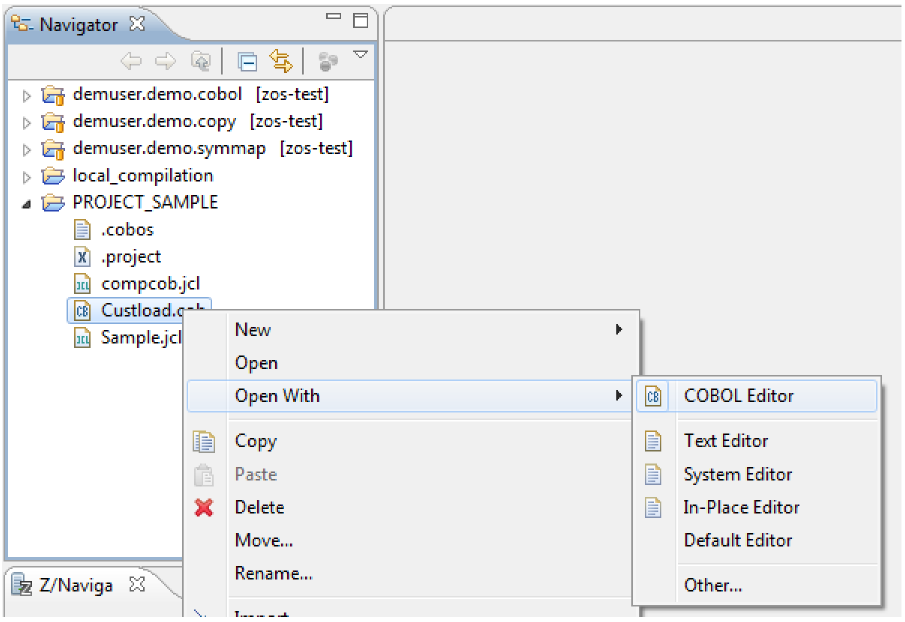
13 You are ready to use Cobos in standalone configuration (without installing Cobos components on the mainframe).



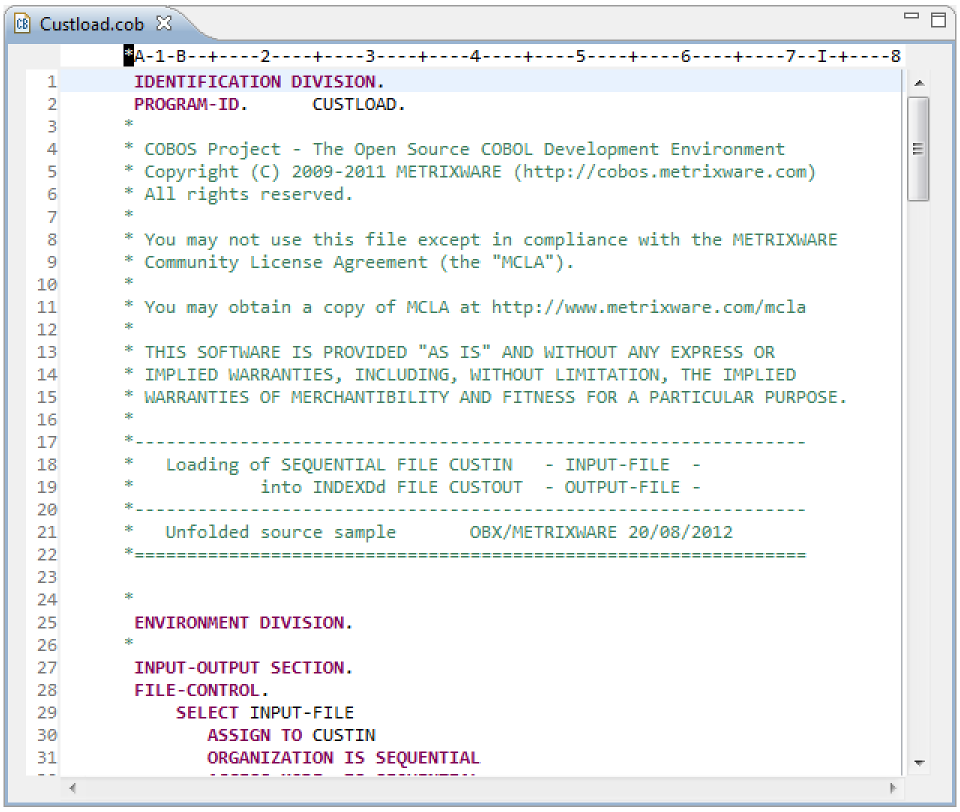
Cobos perspective

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”**

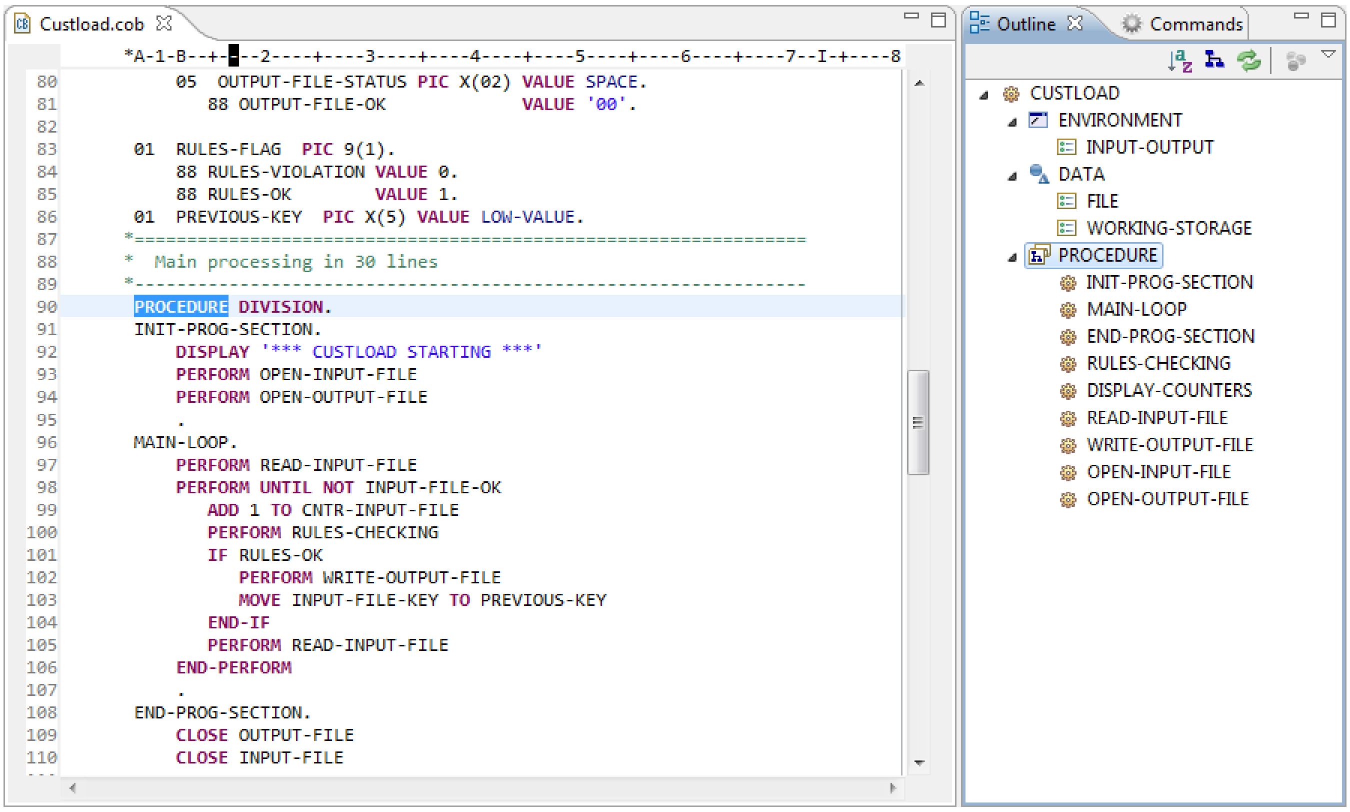


Custload.cob shows off in the Editor.



```
*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--I-+----8
 IDENTIFICATION DIVISION.
 PROGRAM-ID.      CUSTLOAD.
*
* COBOS Project - The Open Source COBOL Development Environment
* Copyright (C) 2009-2011 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 at http://www.metrixware.com/mcla
*
* THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
* WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
*
*----------------------------------------------------------------
*   Loading of SEQUENTIAL FILE CUSTIN    - INPUT-FILE  -
*            into INDEXDd FILE CUSTOUT   - OUTPUT-FILE -
*----------------------------------------------------------------
*   Unfolded source sample       OBX/METRIXWARE 20/08/2012
*================================================================

*
 ENVIRONMENT DIVISION.
*
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT INPUT-FILE
        ASSIGN TO CUSTIN
        ORGANIZATION IS SEQUENTIAL
```

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



Custload.cob

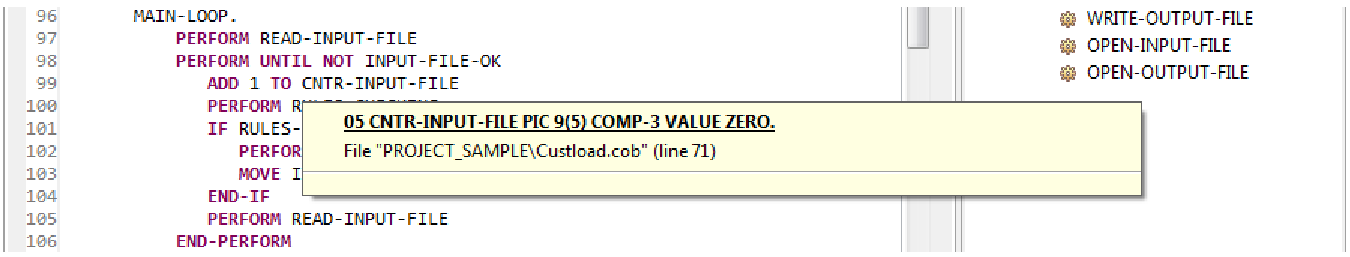
```
   *A-1-B--+--+--2----+----3----+----4----+----5----+----6----+----7--I-+----8
80          05  OUTPUT-FILE-STATUS PIC X(02) VALUE SPACE.
81              88 OUTPUT-FILE-OK            VALUE '00'.
82
83      01  RULES-FLAG  PIC 9(1).
84          88 RULES-VIOLATION VALUE 0.
85          88 RULES-OK        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
```

Outline | Commands

- CUSTLOAD
  - ENVIRONMENT
    - INPUT-OUTPUT
  - DATA
    - FILE
    - WORKING-STORAGE
  - PROCEDURE
    - INIT-PROG-SECTION
    - MAIN-LOOP
    - END-PROG-SECTION
    - RULES-CHECKING
    - DISPLAY-COUNTERS
    - READ-INPUT-FILE
    - WRITE-OUTPUT-FILE
    - OPEN-INPUT-FILE
    - OPEN-OUTPUT-FILE

(for "mainframers") Forget ISPF!

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



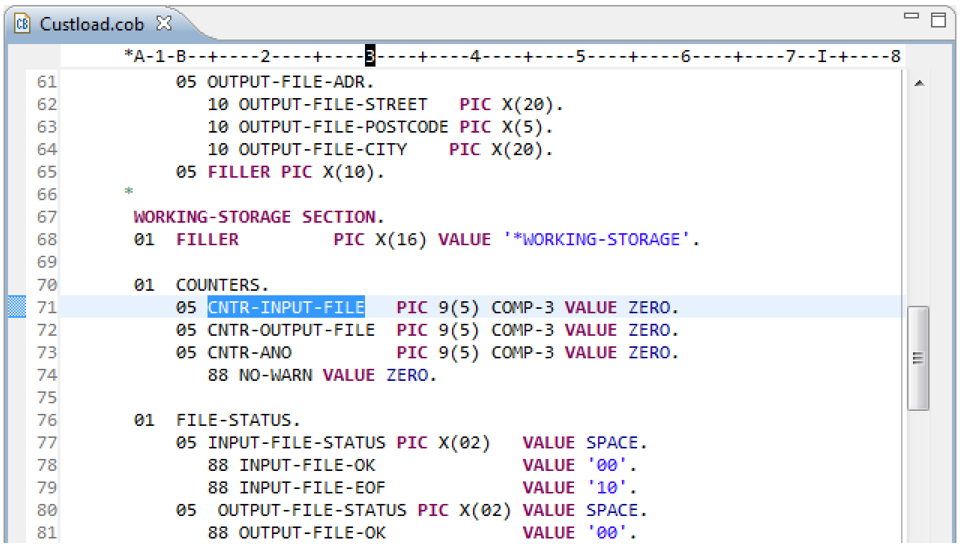
```
96      MAIN-LOOP.
97          PERFORM READ-INPUT-FILE
98          PERFORM UNTIL NOT INPUT-FILE-OK
99             ADD 1 TO CNTR-INPUT-FILE
100            PERFORM R
101            IF RULES-
102               PERFOR
103               MOVE I
104            END-IF
105            PERFORM READ-INPUT-FILE
106         END-PERFORM
```

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

- WRITE-OUTPUT-FILE
- OPEN-INPUT-FILE
- OPEN-OUTPUT-FILE

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**

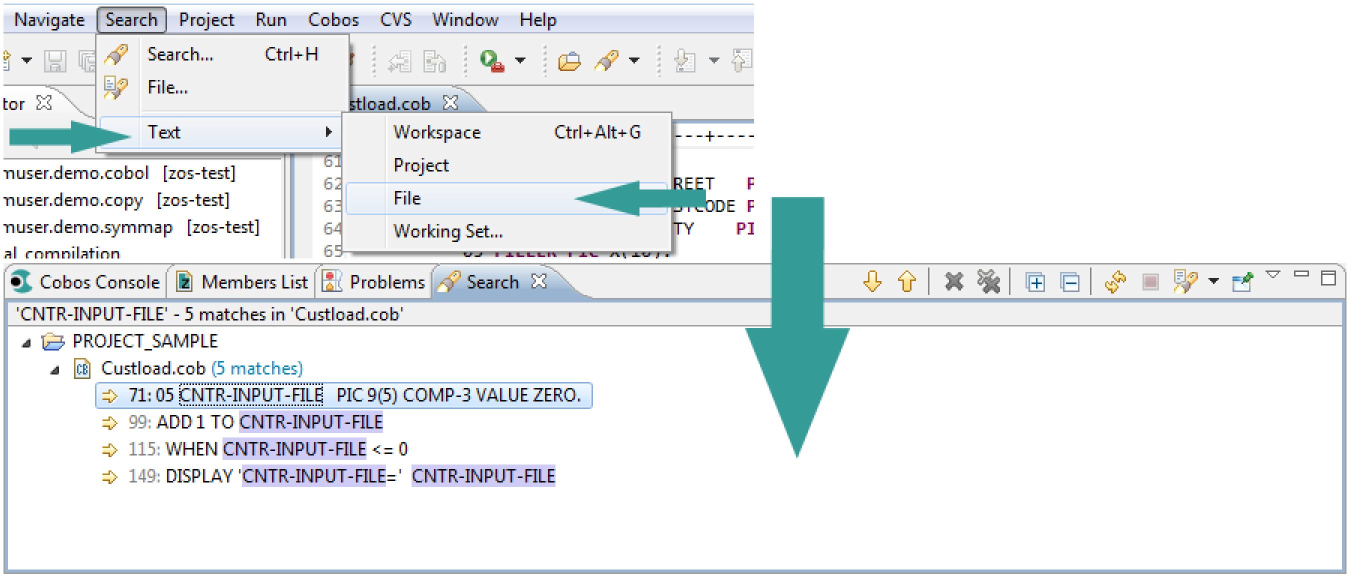


Custload.cob

```
   *A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--I-+----8
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)

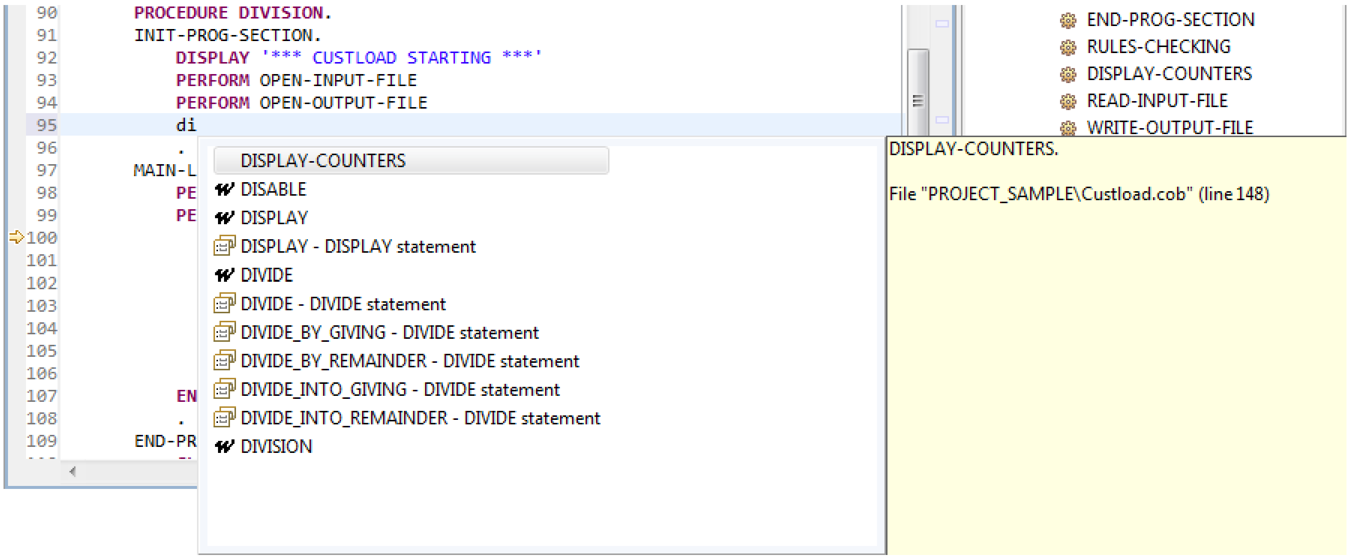
Tip: *To view all occurrences of this variable, select menu **“Search ► Text ► File”**.*



```
'CNTR-INPUT-FILE' - 5 matches in 'Custload.cob'
PROJECT_SAMPLE
  Custload.cob (5 matches)
    71: 05 CNTR-INPUT-FILE   PIC 9(5) COMP-3 VALUE ZERO.
    99: ADD 1 TO CNTR-INPUT-FILE
    115: WHEN CNTR-INPUT-FILE <= 0
    149: DISPLAY 'CNTR-INPUT-FILE=' CNTR-INPUT-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**.



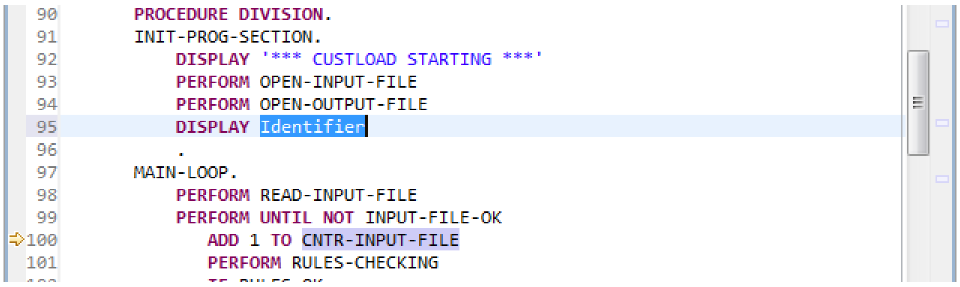
```
 90        PROCEDURE DIVISION.
 91        INIT-PROG-SECTION.
 92            DISPLAY '*** CUSTLOAD STARTING ***'
 93            PERFORM OPEN-INPUT-FILE
 94            PERFORM OPEN-OUTPUT-FILE
 95            di
 96            .
 97        MAIN-L
 98            PE
 99            PE
100
101
102
103
104
105
106
107            EN
108            .
109        END-PR
```

```
DISPLAY-COUNTERS
DISABLE
DISPLAY
DISPLAY - DISPLAY statement
DIVIDE
DIVIDE - DIVIDE statement
DIVIDE_BY_GIVING - DIVIDE statement
DIVIDE_BY_REMAINDER - DIVIDE statement
DIVIDE_INTO_GIVING - DIVIDE statement
DIVIDE_INTO_REMAINDER - DIVIDE statement
DIVISION
```

```
DISPLAY-COUNTERS.

File "PROJECT_SAMPLE\Custload.cob" (line 148)
```

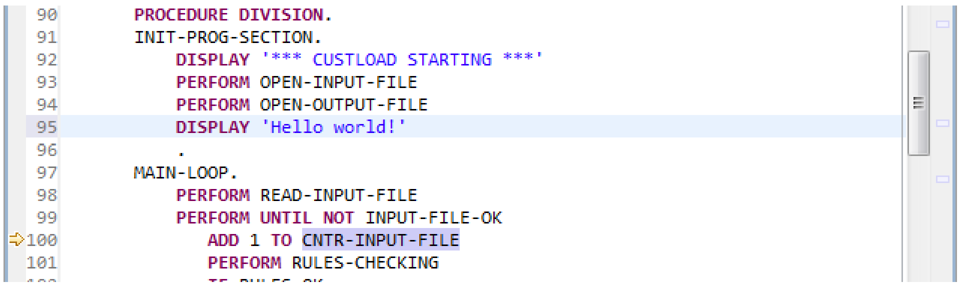
Variables and labels along with their definition line are shown first, then COBOL keywords with or with-out pattern.

6 Double Click on DISPLAY statement pattern



```
 90        PROCEDURE DIVISION.
 91        INIT-PROG-SECTION.
 92            DISPLAY '*** CUSTLOAD STARTING ***'
 93            PERFORM OPEN-INPUT-FILE
 94            PERFORM OPEN-OUTPUT-FILE
 95            DISPLAY Identifier
 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
```

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



```
       PROCEDURE DIVISION.
       INIT-PROG-SECTION.
           DISPLAY '*** CUSTLOAD STARTING ***'
           PERFORM OPEN-INPUT-FILE
           PERFORM OPEN-OUTPUT-FILE
           DISPLAY 'Hello world!'
           .
       MAIN-LOOP.
           PERFORM READ-INPUT-FILE
           PERFORM UNTIL NOT INPUT-FILE-OK
              ADD 1 TO CNTR-INPUT-FILE
              PERFORM RULES-CHECKING
```

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 "Cobos ► Check Syntax")
- **Ctrl + Shift + U**: unfold of the code (shortcut for menu "Cobos ► Unfold COBOL source")
- **Ctrl + Shift + C**: remote compilation[1] (shortcut for menu "Cobos ► Compilation")

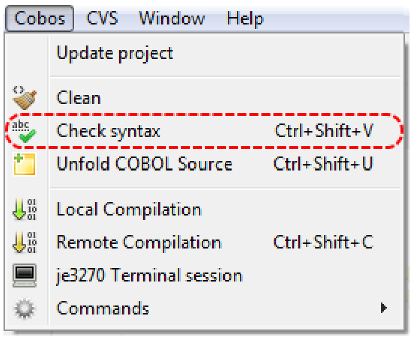
[1] Need a Mainframe.

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 **“Cobos ► Check syntax”**

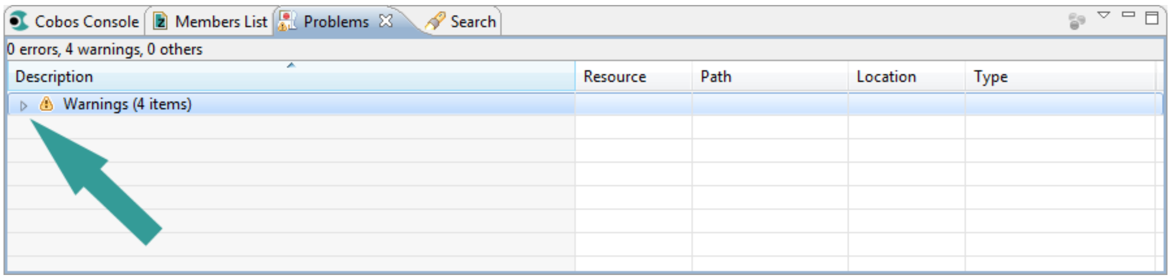

Cobos
CVS
Window
Help
Update project
Clean
Check syntax Ctrl+Shift+V
Unfold COBOL Source Ctrl+Shift+U
Local Compilation
Remote Compilation Ctrl+Shift+C
je3270 Terminal session
Commands

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.

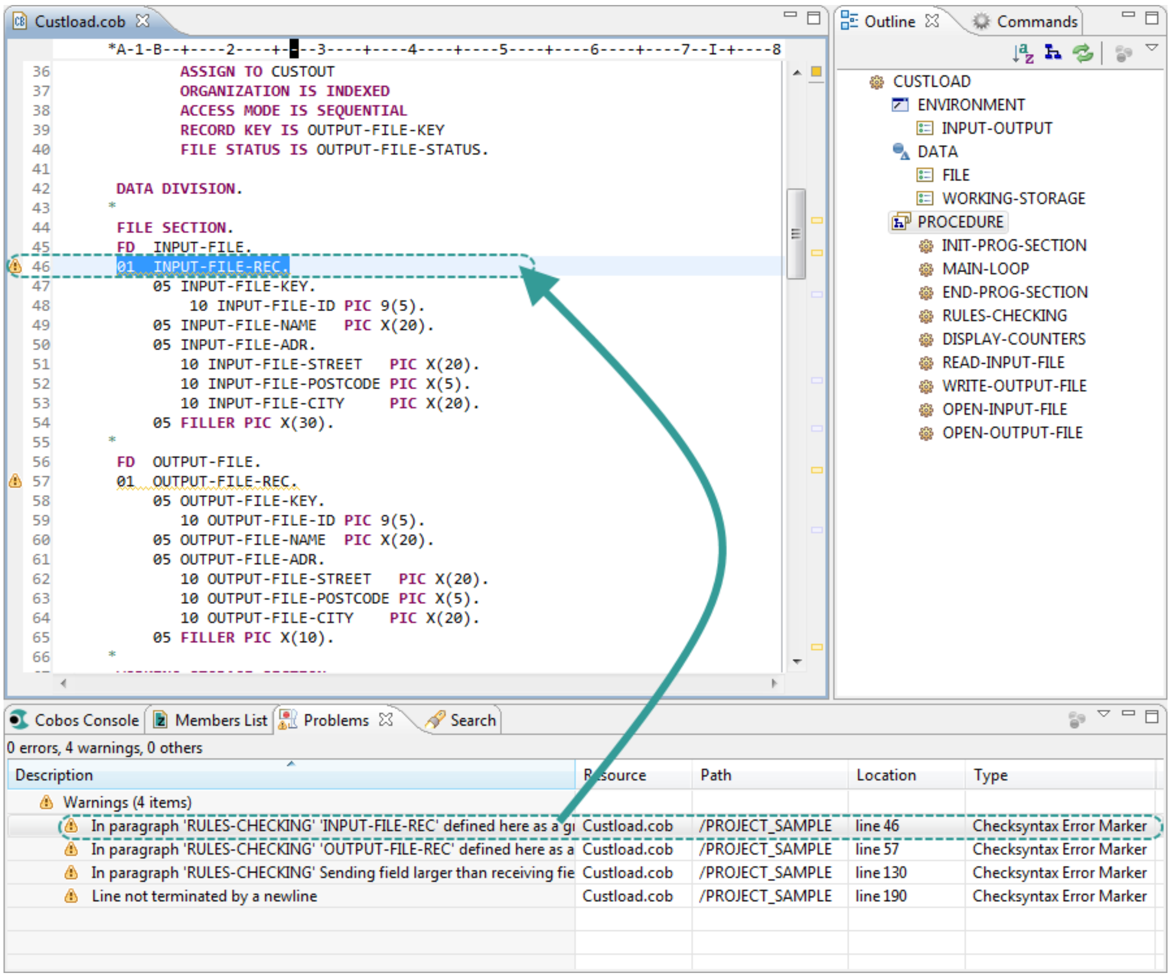
4 In the Problems view, expand the Warnings line to see the messages.



0 errors, 4 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (4 items)				

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



```
        *A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--I-+----8
                   ASSIGN TO CUSTOUT
                   ORGANIZATION IS INDEXED
                   ACCESS MODE IS SEQUENTIAL
                   RECORD KEY IS OUTPUT-FILE-KEY
                   FILE STATUS IS OUTPUT-FILE-STATUS.

            DATA DIVISION.
           *
            FILE SECTION.
            FD  INPUT-FILE.
            01  INPUT-FILE-REC.
                05 INPUT-FILE-KEY.
                    10 INPUT-FILE-ID PIC 9(5).
                05 INPUT-FILE-NAME   PIC X(20).
                05 INPUT-FILE-ADR.
                   10 INPUT-FILE-STREET   PIC X(20).
                   10 INPUT-FILE-POSTCODE PIC X(5).
                   10 INPUT-FILE-CITY     PIC X(20).
                05 FILLER PIC X(30).
           *
            FD  OUTPUT-FILE.
            01  OUTPUT-FILE-REC.
                05 OUTPUT-FILE-KEY.
                   10 OUTPUT-FILE-ID PIC 9(5).
                05 OUTPUT-FILE-NAME  PIC X(20).
                05 OUTPUT-FILE-ADR.
                   10 OUTPUT-FILE-STREET   PIC X(20).
                   10 OUTPUT-FILE-POSTCODE PIC X(5).
                   10 OUTPUT-FILE-CITY    PIC X(20).
                05 FILLER PIC X(10).
           *
```

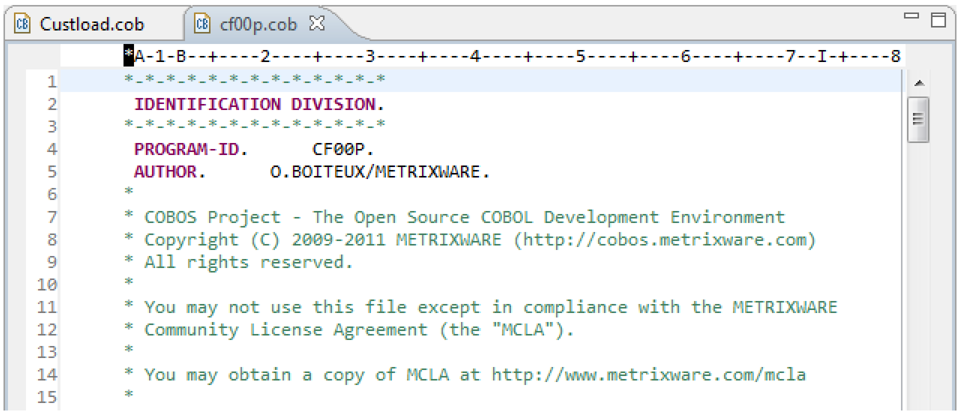
0 errors, 4 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (4 items)				
In paragraph 'RULES-CHECKING' 'INPUT-FILE-REC' defined here as a gı	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 Cobos.

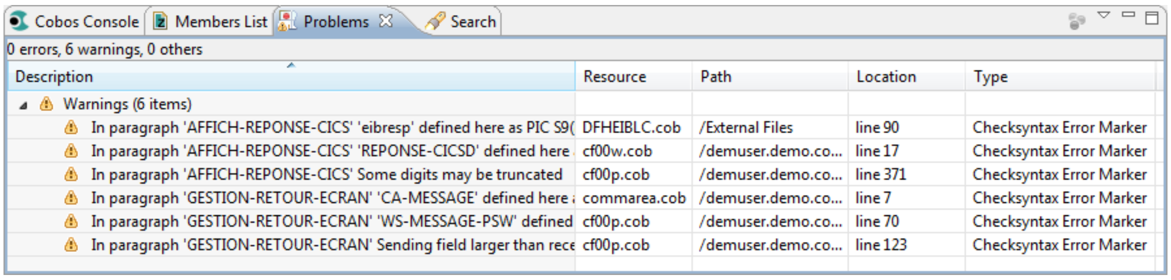
1 Open the program "cf00p.cob" from demuser.demo.cobol project with the COBOL Editor.



```
        *A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--I-+----8
        *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
         IDENTIFICATION DIVISION.
        *-*-*-*-*-*-*-*-*-*-*-*-*-*-*
         PROGRAM-ID.      CF00P.
         AUTHOR.      O.BOITEUX/METRIXWARE.
        *
        * COBOS Project - The Open Source COBOL Development Environment
        * Copyright (C) 2009-2011 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 at http://www.metrixware.com/mcla
        *
```

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

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

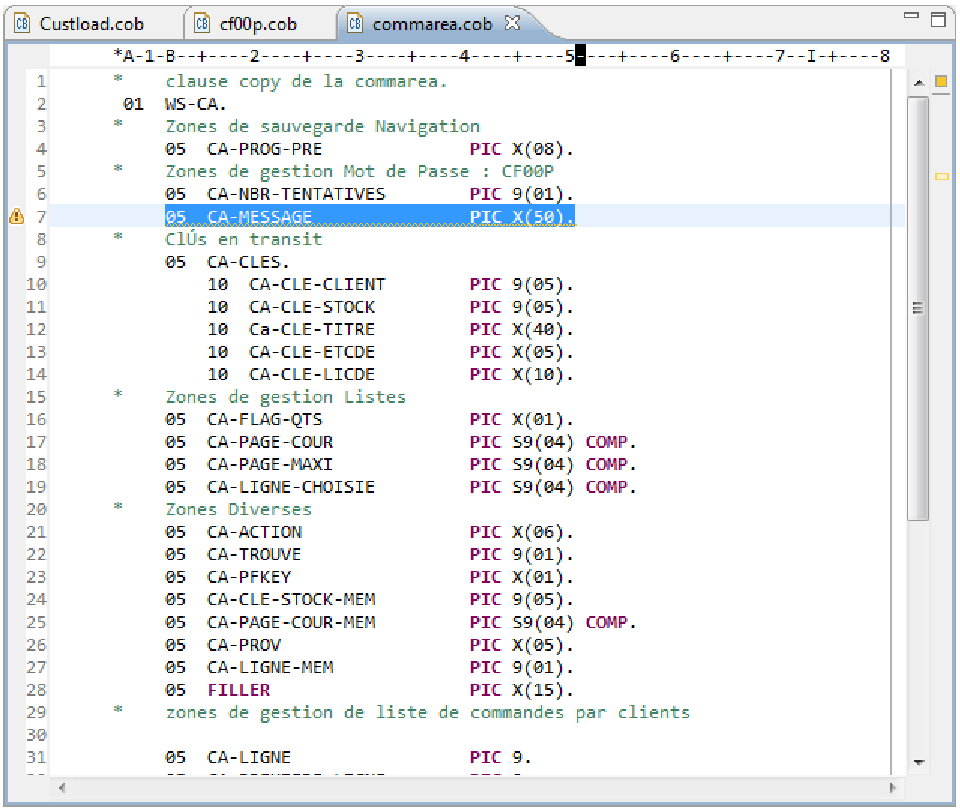


0 errors, 6 warnings, 0 others

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 a	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     *     ClÛs 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.
```

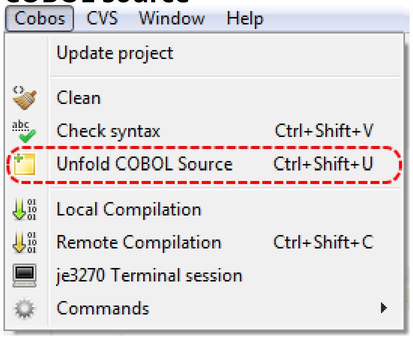
A marker is placed in front of the line addressed by the warning and the line is highlighted.

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".

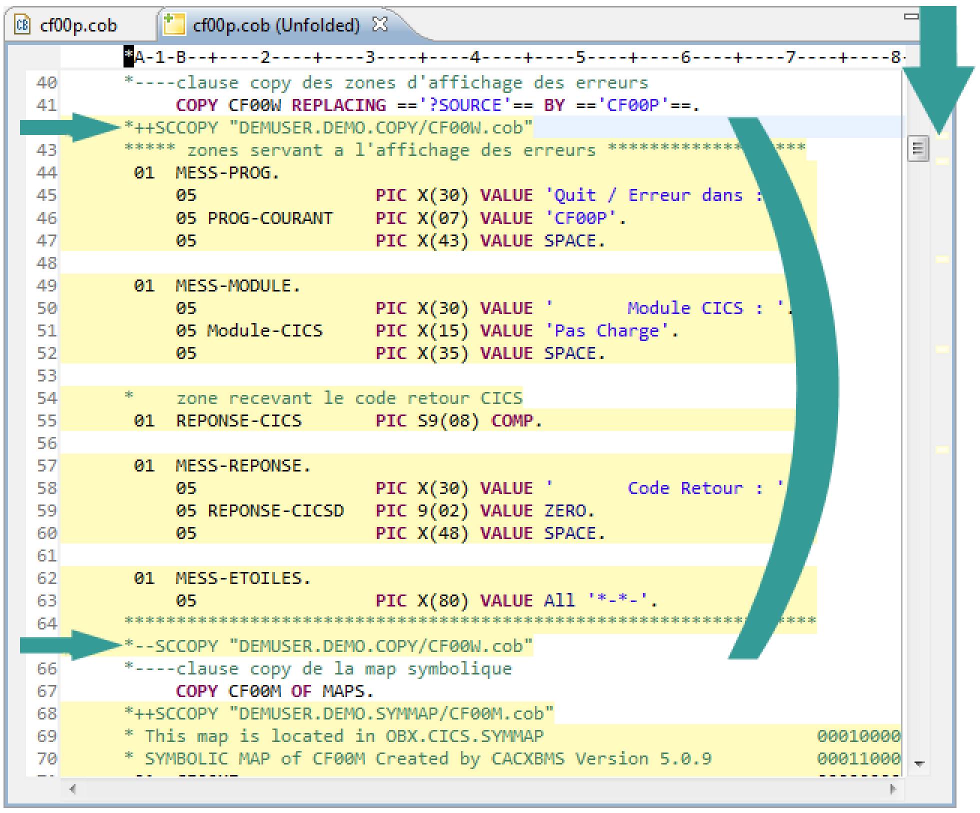
5.1 CF00P program

1 Check that the program "cf00p.cob" is active in the COBOL Editor, select menu **"Cobos ► Unfold COBOL source"**


Cobos
CVS
Window
Help
Update project
Clean
Check syntax Ctrl+Shift+V
Unfold COBOL Source Ctrl+Shift+U
Local Compilation
Remote Compilation Ctrl+Shift+C
je3270 Terminal session
Commands

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)".



```
       *A-1-B--+----2----+----3----+----4----+----5----+----6----+----7----+----8
       *----clause copy des zones d'affichage des erreurs
            COPY CF00W REPLACING =='?SOURCE'== BY =='CF00P'==.
       *++SCCOPY "DEMUSER.DEMO.COPY/CF00W.cob"
       ***** zones servant a l'affichage des erreurs ************   ***
        01  MESS-PROG.
            05                   PIC X(30) VALUE 'Quit / Erreur dans :
            05 PROG-COURANT      PIC X(07) VALUE 'CF00P'.
            05                   PIC X(43) VALUE SPACE.

        01  MESS-MODULE.
            05                   PIC X(30) VALUE '       Module CICS : '.
            05 Module-CICS       PIC X(15) VALUE 'Pas Charge'.
            05                   PIC X(35) VALUE SPACE.

       *    zone recevant le code retour CICS
        01  REPONSE-CICS         PIC S9(08) COMP.

        01  MESS-REPONSE.
            05                   PIC X(30) VALUE '       Code Retour : '
            05 REPONSE-CICSD     PIC 9(02) VALUE ZERO.
            05                   PIC X(48) VALUE SPACE.

        01  MESS-ETOILES.
            05                   PIC X(80) VALUE All '*-*-'.
       ***********************************************************   ****
       *--SCCOPY "DEMUSER.DEMO.COPY/CF00W.cob"
       *----clause copy de la map symbolique
            COPY CF00M OF MAPS.
       *++SCCOPY "DEMUSER.DEMO.SYMMAP/CF00M.cob"
       * This map is located in OBX.CICS.SYMMAP                        00010000
       * SYMBOLIC MAP of CF00M Created by CACXBMS Version 5.0.9        00011000
```

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"***.

▸If you installed the Cobos Essentials version you stop here.◂
To continue, you must have the Cobos Release version.

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.

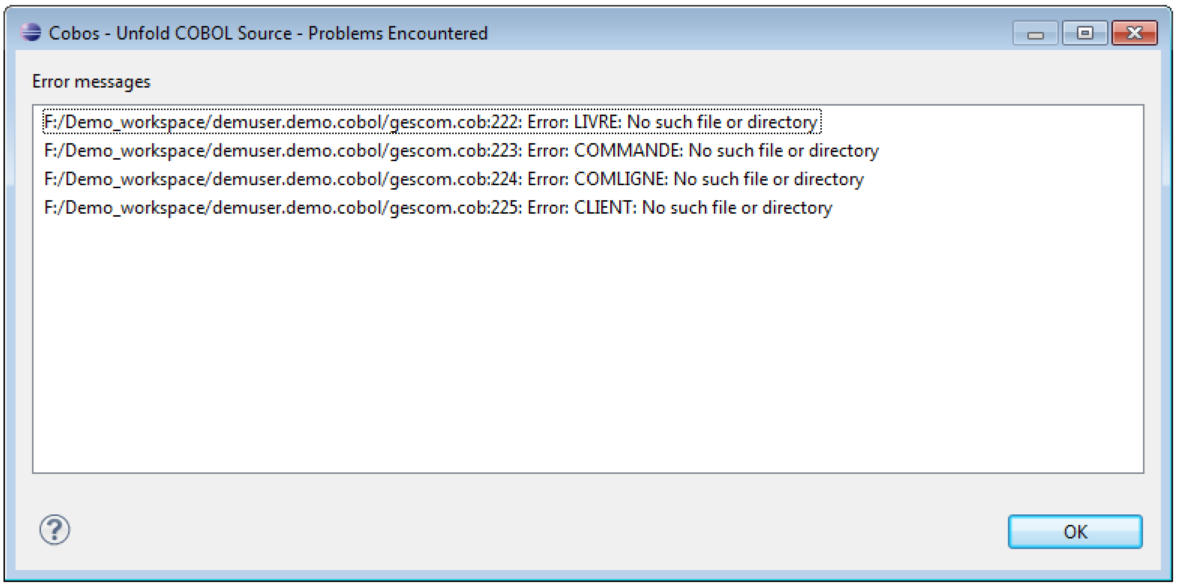
One solution is to replicate the copybooks in a network place, and configure the .cobos file.
Note: if you want to understand how copybooks paths are specified, open the .cobos file. For more information, select menu **"Help ▸ Help Contents ▸ Cobos ▸ Chapter 5 Viewing and ... ▸ Configuring a project"**.

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 Cobos (and sysout is still available).

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

2 Unfold (select menu **"Cobos ▸ Unfold COBOL source"** or push button).

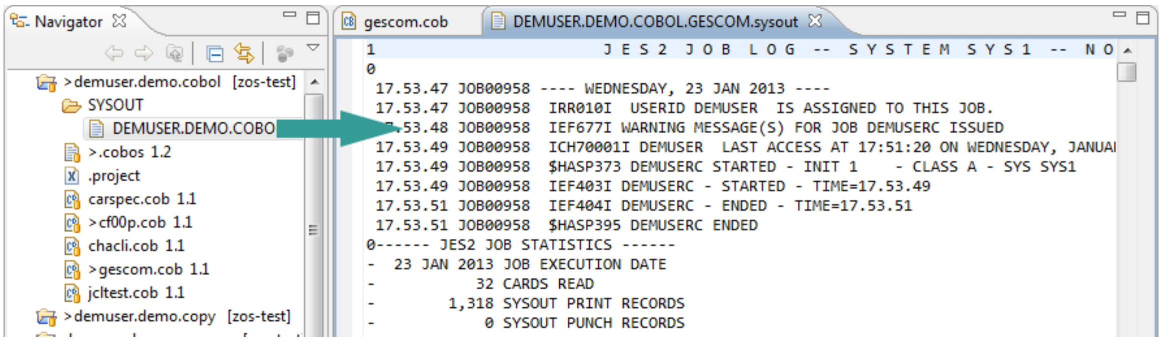
3 After Unfold, you will get a new tab with the name "gescom.cob (Unfolded)" but a popup shows off because copybooks are not available according to the paths specified in .cobos file:



```
F:/Demo_workspace/demuser.demo.cobol/gescom.cob:222: Error: LIVRE: No such file or directory
F:/Demo_workspace/demuser.demo.cobol/gescom.cob:223: Error: COMMANDE: No such file or directory
F:/Demo_workspace/demuser.demo.cobol/gescom.cob:224: Error: COMLIGNE: No such file or directory
F:/Demo_workspace/demuser.demo.cobol/gescom.cob:225: Error: CLIENT: No such file or directory
```

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.



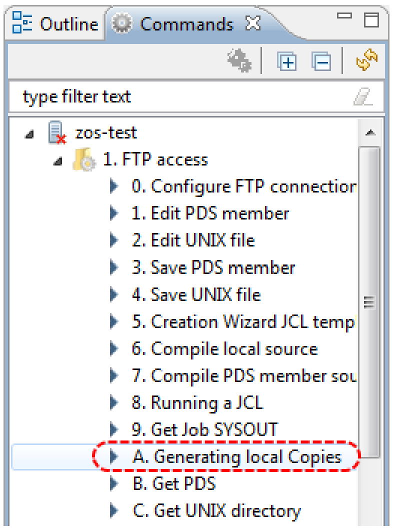
- >demuser.demo.cobol [zos-test]
  - SYSOUT
    - DEMUSER.DEMO.COBO
  - >.cobos 1.2
  - .project
  - carspec.cob 1.1
  - >cf00p.cob 1.1
  - chacli.cob 1.1
  - >gescom.cob 1.1
  - jcltest.cob 1.1
- >demuser.demo.copy [zos-test]

```
1                          J E S 2  J O B  L O G  --  S Y S T E M  S Y S 1  --  N O
0
 17.53.47 JOB00958 ---- WEDNESDAY, 23 JAN 2013 ----
 17.53.47 JOB00958  IRR010I  USERID DEMUSER  IS ASSIGNED TO THIS JOB.
 17.53.48 JOB00958  IEF677I WARNING MESSAGE(S) FOR JOB DEMUSERC ISSUED
 17.53.49 JOB00958  ICH70001I DEMUSER  LAST ACCESS AT 17:51:20 ON WEDNESDAY, JANUAI
 17.53.49 JOB00958  $HASP373 DEMUSERC STARTED - INIT 1    - CLASS A - SYS SYS1
 17.53.49 JOB00958  IEF403I DEMUSERC - STARTED - TIME=17.53.49
 17.53.51 JOB00958  IEF404I DEMUSERC - ENDED - TIME=17.53.51
 17.53.51 JOB00958  $HASP395 DEMUSERC ENDED
0------ JES2 JOB STATISTICS ------
-  23 JAN 2013 JOB EXECUTION DATE
-           32 CARDS READ
-        1,318 SYSOUT PRINT RECORDS
-            0 SYSOUT PUNCH RECORDS
```

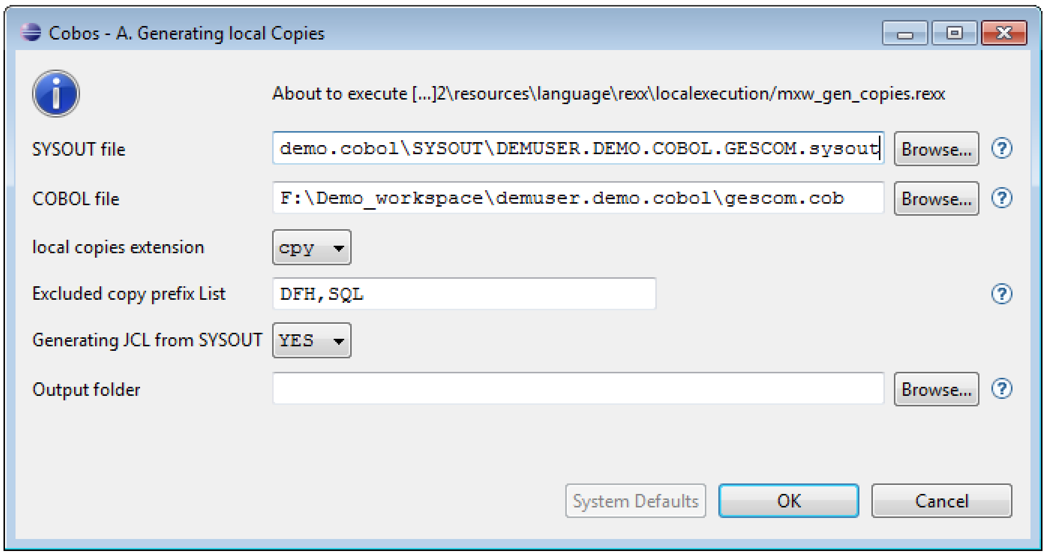
The sysout is opened in the Text Editor.

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 Cobos command **"1. FTP access ► A. Generating local Copies"** from the Command view.



7 Select the sysout and click OK.

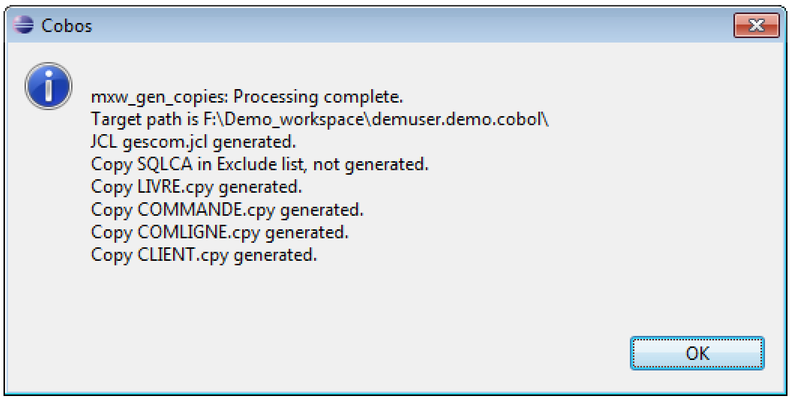


About to execute [...]2\resources\language\rexx\localexecution/mxw_gen_copies.rexx

SYSOUT file	demo.cobol\SYSOUT\DEMUSER.DEMO.COBOL.GESCOM.sysout
COBOL file	F:\Demo_workspace\demuser.demo.cobol\gescom.cob
local copies extension	cpy
Excluded copy prefix List	DFH,SQL
Generating JCL from SYSOUT	YES
Output folder	

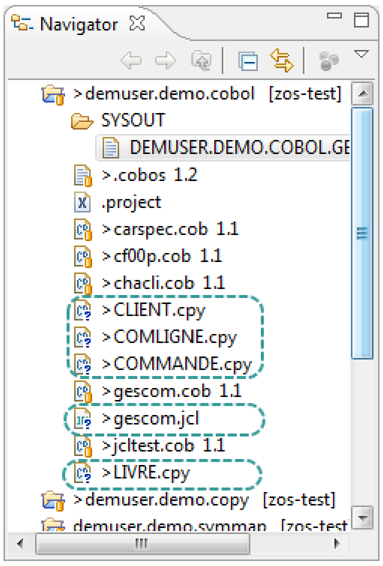
Note: in this example, we exclude CICS and DB2 copybooks.

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



```
mxw_gen_copies: Processing complete.
Target path is F:\Demo_workspace\demuser.demo.cobol\
JCL gescom.jcl generated.
Copy SQLCA in Exclude list, not generated.
Copy LIVRE.cpy generated.
Copy COMMANDE.cpy generated.
Copy COMLIGNE.cpy generated.
Copy CLIENT.cpy generated.
```

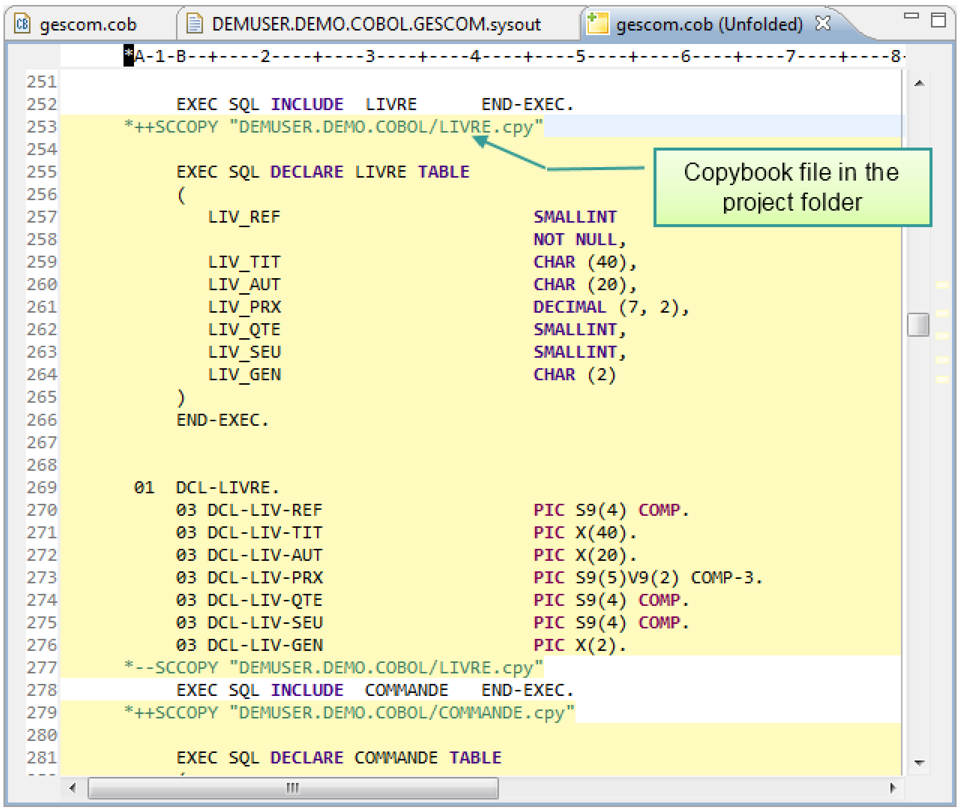
9 The copies and the JCL are stored along with the source file.



```
Navigator
>demuser.demo.cobol  [zos-test]
  SYSOUT
    DEMUSER.DEMO.COBOL.GE
  >.cobos 1.2
  .project
  >carspec.cob 1.1
  >cf00p.cob 1.1
  >chacli.cob 1.1
  >CLIENT.cpy
  >COMLIGNE.cpy
  >COMMANDE.cpy
  >gescom.cob 1.1
  >gescom.jcl
  >jcltest.cob 1.1
  >LIVRE.cpy
>demuser.demo.copy  [zos-test]
demuser.demo.symman  [zos-test
```

Note: Cobos 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.


gescom.cob
DEMUSER.DEMO.COBOL.GESCOM.sysout
gescom.cob (Unfolded)
Copybook file in the project folder

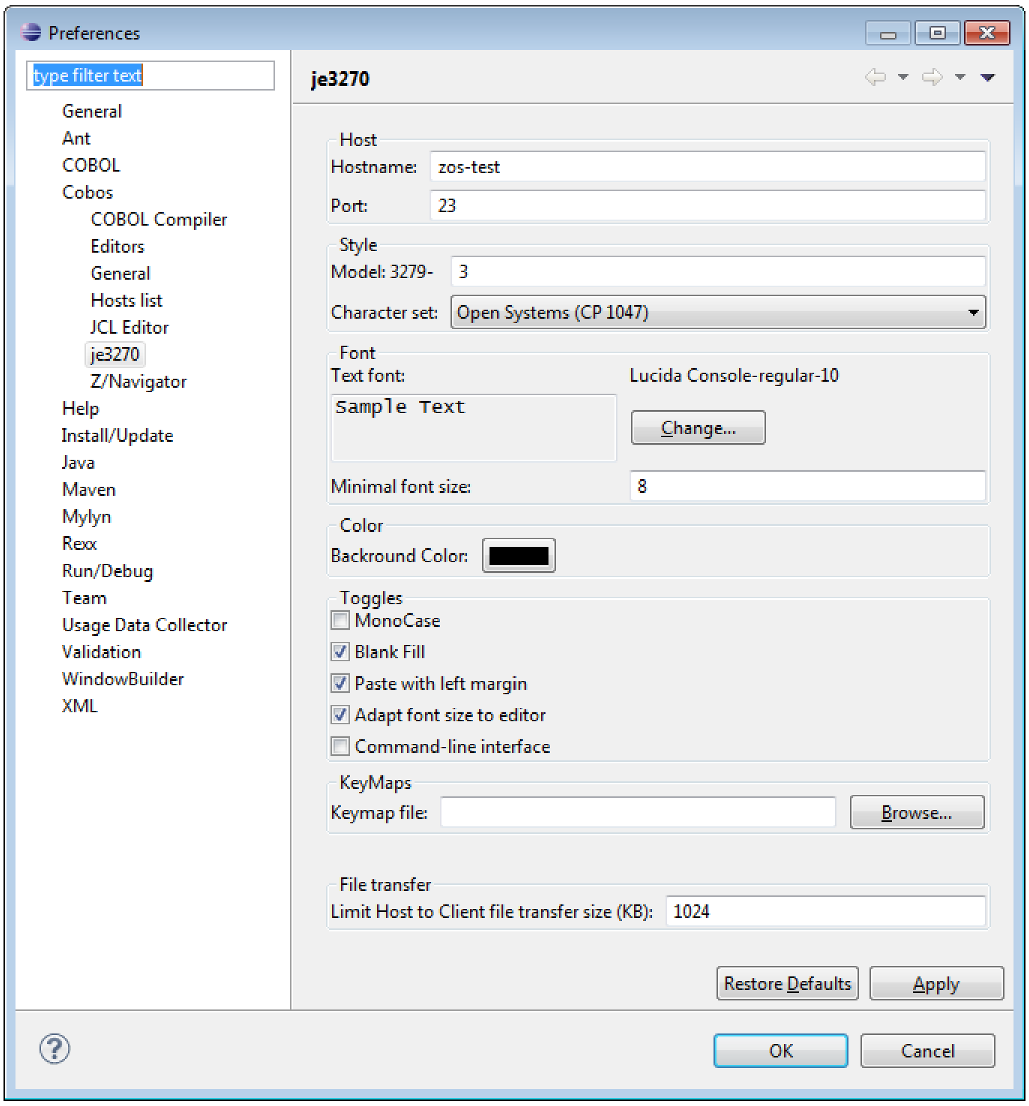
```
*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7----+----8

          EXEC SQL INCLUDE  LIVRE      END-EXEC.
     *++SCCOPY "DEMUSER.DEMO.COBOL/LIVRE.cpy"

          EXEC SQL DECLARE LIVRE TABLE
          (
             LIV_REF                        SMALLINT
                                            NOT NULL,
             LIV_TIT                        CHAR (40),
             LIV_AUT                        CHAR (20),
             LIV_PRX                        DECIMAL (7, 2),
             LIV_QTE                        SMALLINT,
             LIV_SEU                        SMALLINT,
             LIV_GEN                        CHAR (2)
          )
          END-EXEC.


      01  DCL-LIVRE.
          03 DCL-LIV-REF                    PIC S9(4) COMP.
          03 DCL-LIV-TIT                    PIC X(40).
          03 DCL-LIV-AUT                    PIC X(20).
          03 DCL-LIV-PRX                    PIC S9(5)V9(2) COMP-3.
          03 DCL-LIV-QTE                    PIC S9(4) COMP.
          03 DCL-LIV-SEU                    PIC S9(4) COMP.
          03 DCL-LIV-GEN                    PIC X(2).
     *--SCCOPY "DEMUSER.DEMO.COBOL/LIVRE.cpy"
          EXEC SQL INCLUDE  COMMANDE   END-EXEC.
     *++SCCOPY "DEMUSER.DEMO.COBOL/COMMANDE.cpy"

          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 Cobos 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 ► Cobos ► 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)



je3270

Host
Hostname: zos-test
Port: 23

Style
Model: 3279- 3
Character set: Open Systems (CP 1047)

Font
Text font: Lucida Console-regular-10
Sample Text
Minimal font size: 8

Color
Backround Color: ■

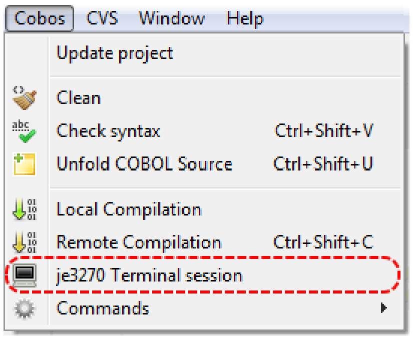
Toggles
☐ MonoCase
☑ Blank Fill
☑ Paste with left margin
☑ Adapt font size to editor
☐ Command-line interface

KeyMaps
Keymap file:

File transfer
Limit Host to Client file transfer size (KB): 1024

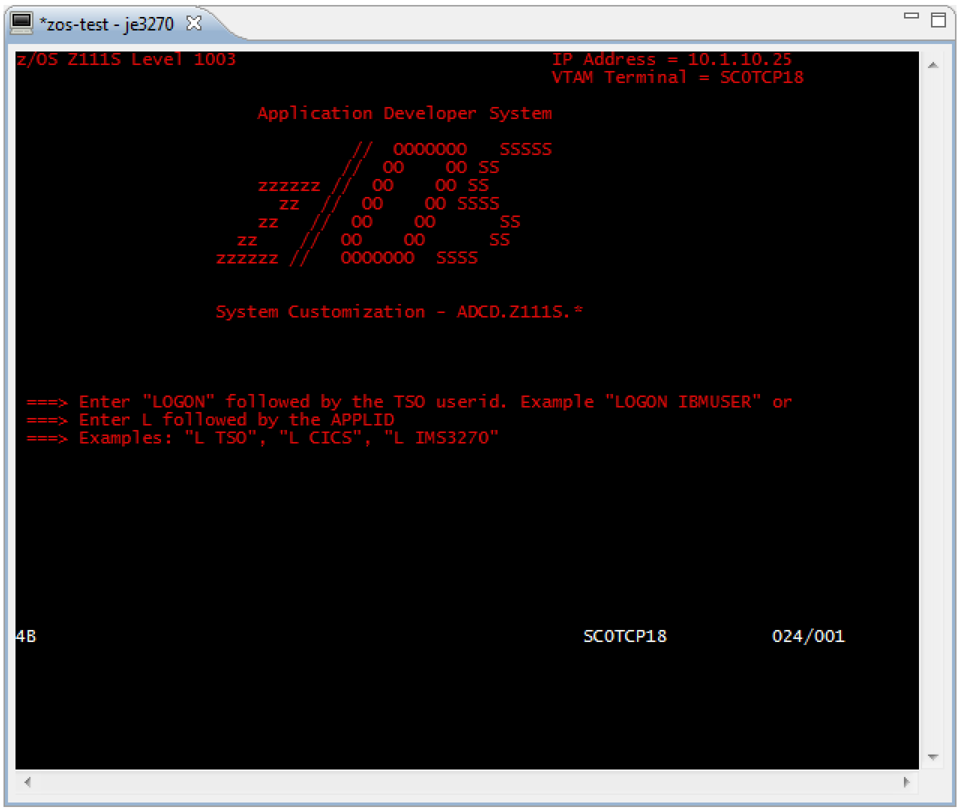
You will be able to define a customized key map.

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



Cobos
CVS
Window
Help
Update project
Clean
Check syntax Ctrl+Shift+V
Unfold COBOL Source Ctrl+Shift+U
Local Compilation
Remote Compilation Ctrl+Shift+C
je3270 Terminal session
Commands

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



```
z/OS Z111S Level 1003                                  IP Address = 10.1.10.25
                                                       VTAM Terminal = SCOTCP18

                        Application Developer System

                                  //  OOOOOOO   SSSSS
                                 //  OO    OO  SS
                        zzzzzz  //  OO    OO  SS
                          zz   //  OO    OO  SSSS
                        zz    //  OO    OO      SS
                      zz     //  OO    OO      SS
                    zzzzzz  //   OOOOOOO   SSSS


                    System Customization - ADCD.Z111S.*



 ===> Enter "LOGON" followed by the TSO userid. Example "LOGON IBMUSER" or
 ===> Enter L followed by the APPLID
 ===> Examples: "L TSO", "L CICS", "L IMS3270"









4B                                                      SCOTCP18          024/001
```

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!

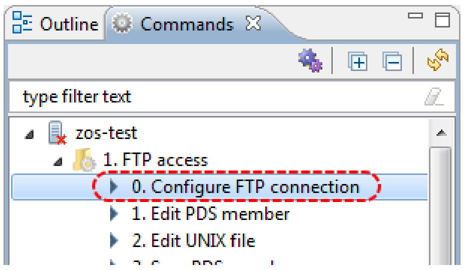
7. Configure FTP access

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

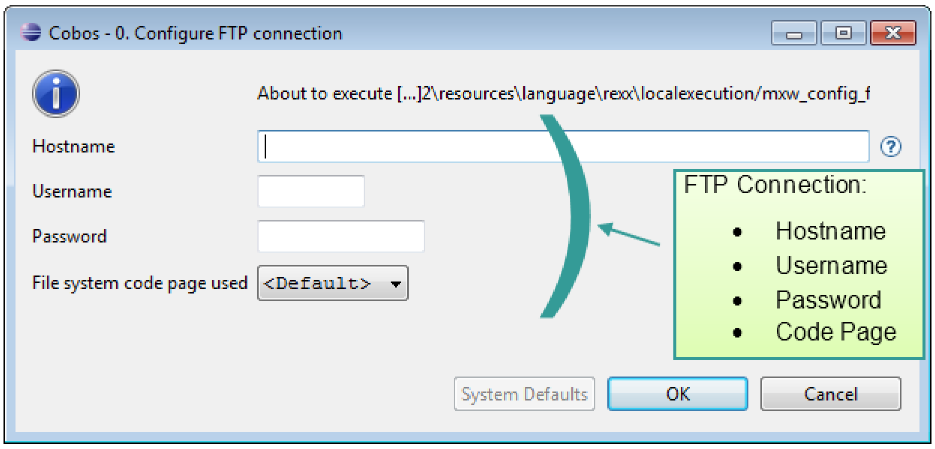
"FTP Access" module provides another way of working with mainframe sources without any mainframe installation. This module requires installation of a REXX interpreter on the workstation such as Regina REXX Interpreter.

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:


Cobos - 0. Configure FTP connection

About to execute [...]2\resources\language\rexx\localexecution/mxw_config_f

Hostname

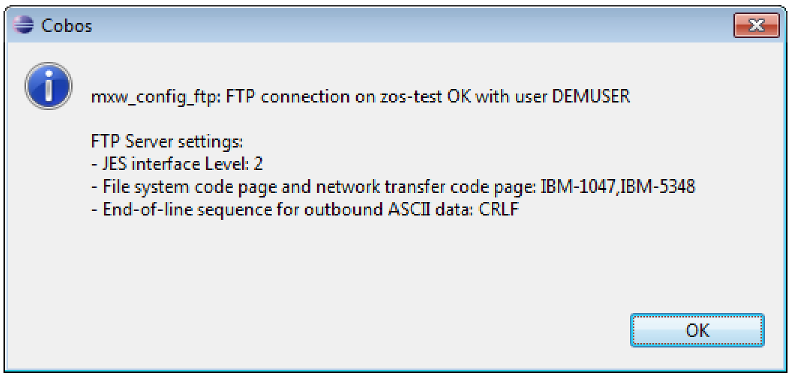
Username

Password

File system code page used <Default>

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.



```
mxw_config_ftp: FTP connection on zos-test OK with user DEMUSER

FTP Server settings:
- JES interface Level: 2
- File system code page and network transfer code page: IBM-1047,IBM-5348
- End-of-line sequence for outbound ASCII data: CRLF
```

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

Cobos 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.

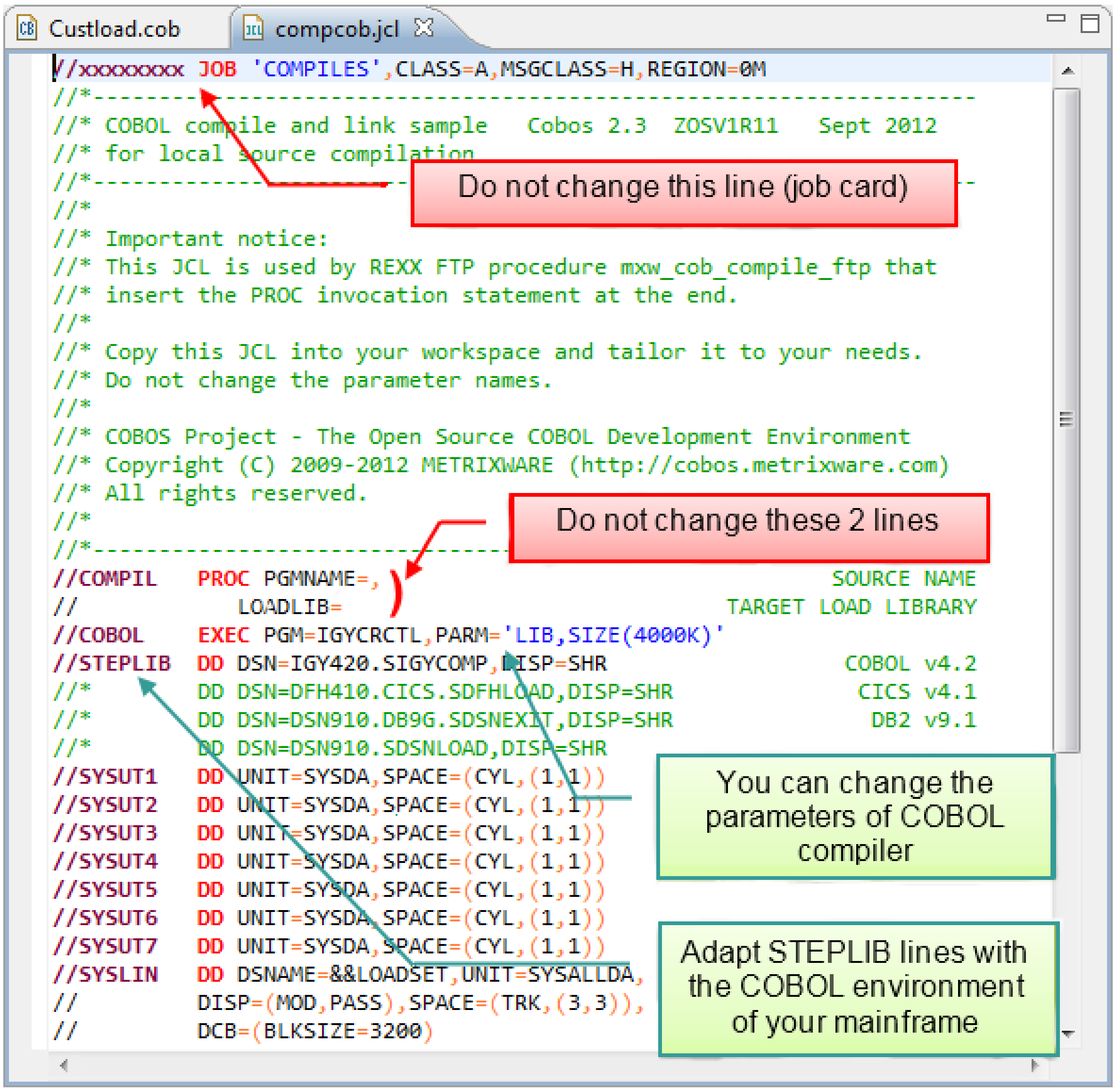
Z/Navigator plug-in is designed to process mainframe files directly from eclipse but requires scripts installation on mainframe.

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

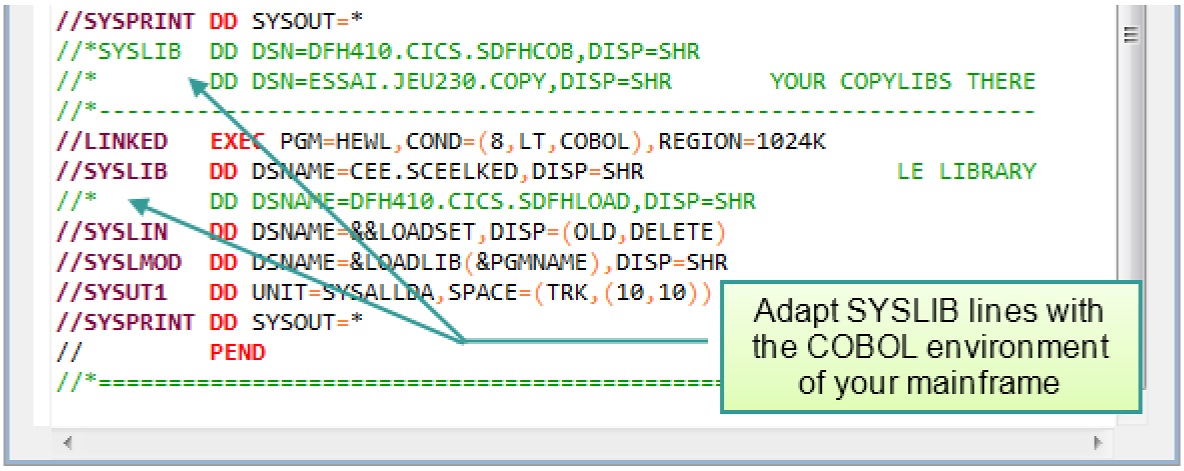


Custload.cob
compcob.jcl
Do not change this line (job card)
Do not change these 2 lines
You can change the parameters of COBOL compiler
Adapt STEPLIB lines with the COBOL environment of your mainframe

```
//xxxxxxxx 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 DSNAME=&&LOADSET,UNIT=SYSALLDA,
//         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 DSNAME according with the version of Enterprise

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



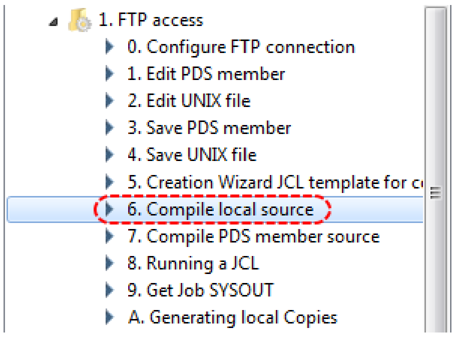
```
//SYSPRINT DD SYSOUT=*
//*SYSLIB  DD DSN=DFH410.CICS.SDFHCOB,DISP=SHR
//*        DD DSN=ESSAI.JEU230.COPY,DISP=SHR       YOUR COPYLIBS THERE
//*-------------------------------------------------------------------
//LINKED   EXEC PGM=HEWL,COND=(8,LT,COBOL),REGION=1024K
//SYSLIB   DD DSNAME=CEE.SCEELKED,DISP=SHR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

Adapt SYSLIB lines with the COBOL environment of your mainframe

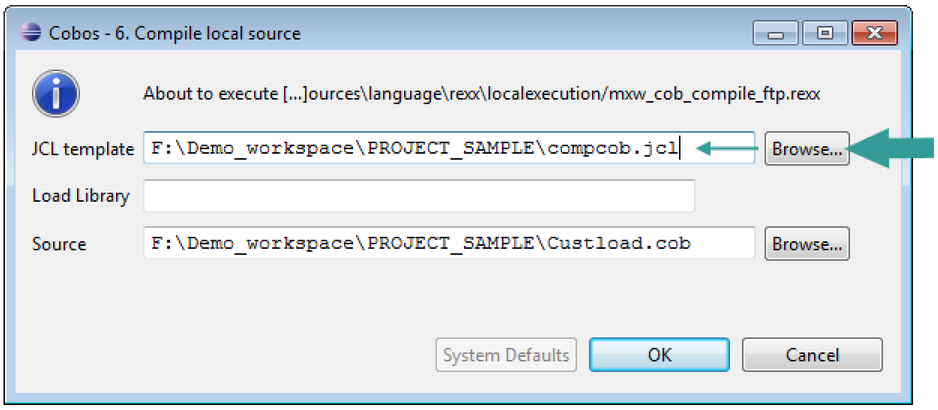
Custload program does not require copies and object modules. So, leave the comment lines in SYSLIB (COBOL and LINKED).

3 Select Custload.cob tab (Custload.cob).

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:



Cobos - 6. Compile local source

About to execute [...]ources\language\rexx\localexecution/mxw_cob_compile_ftp.rexx

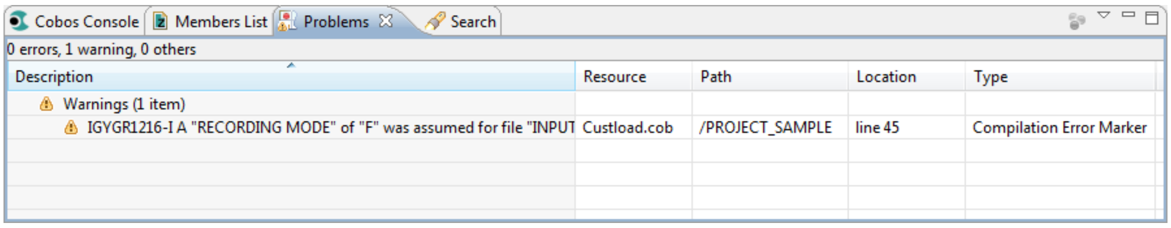
JCL template: F:\Demo_workspace\PROJECT_SAMPLE\compcob.jcl

Load Library:

Source: F:\Demo_workspace\PROJECT_SAMPLE\Custload.cob

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

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

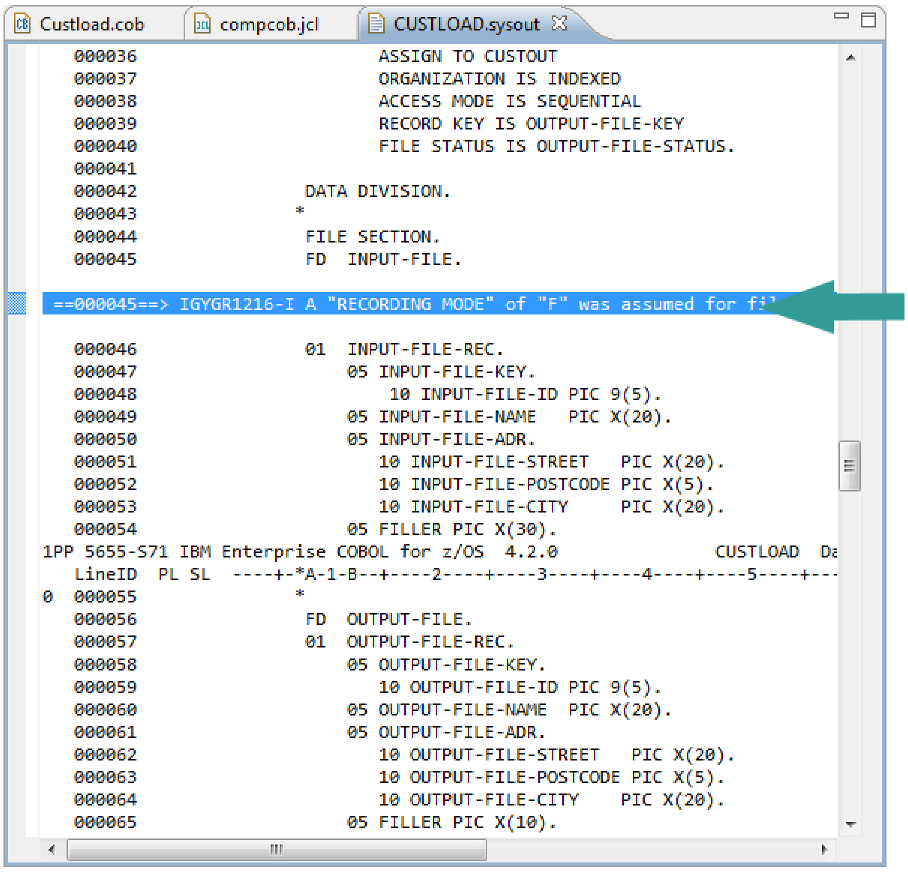


0 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
Warnings (1 item)				
IGYGR1216-I A "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-I A "RECORDING MODE" of "F" was assumed for fi

   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  Da
   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).
```

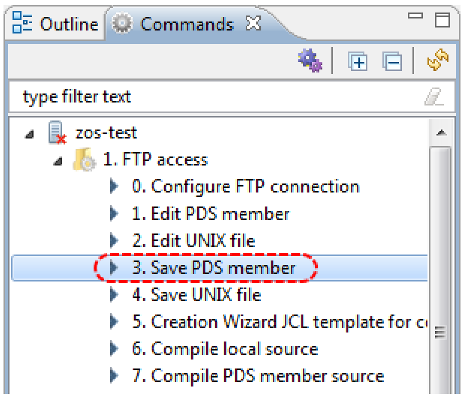
The sysout opens in the text editor and the warning line is selected.

Editing a PDS member

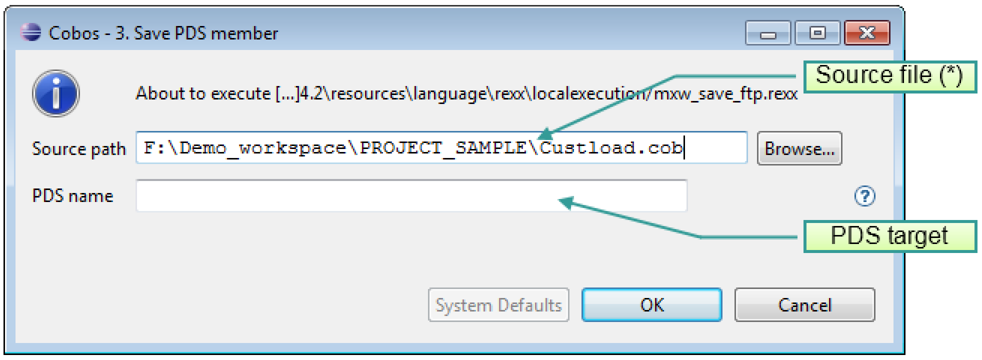
8.1 Saving a PDS member

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

1 Save *Custload.cob* in a PDS using the Cobos command **“1. FTP access ▸ 3. Save PDS member**.



2 Fill the form and press **“OK”**.


Cobos - 3. Save PDS member

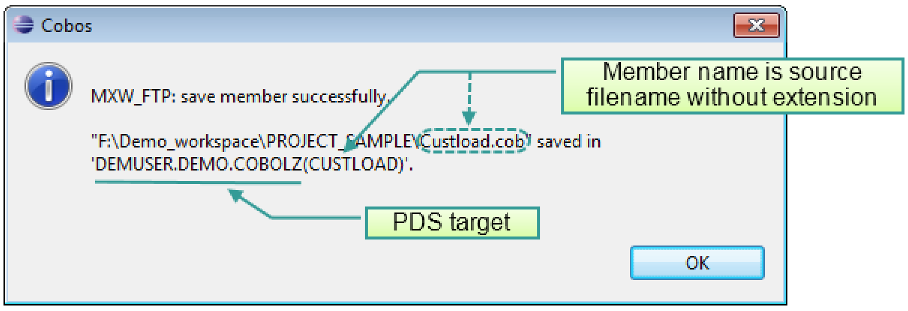
About to execute [...]4.2\resources\language\rexx\localexecution/mxw_save_ftp.rexx

Source path: F:\Demo_workspace\PROJECT_SAMPLE\Custload.cob

PDS name:

(*) 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.


Cobos

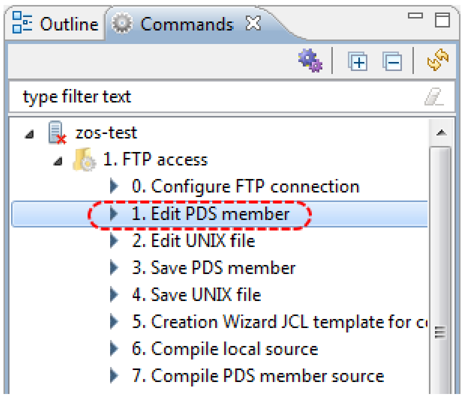
MXW_FTP: save member successfully.

"F:\Demo_workspace\PROJECT_SAMPLE\Custload.cob' saved in 'DEMUSER.DEMO.COBOLZ(CUSTLOAD)'.

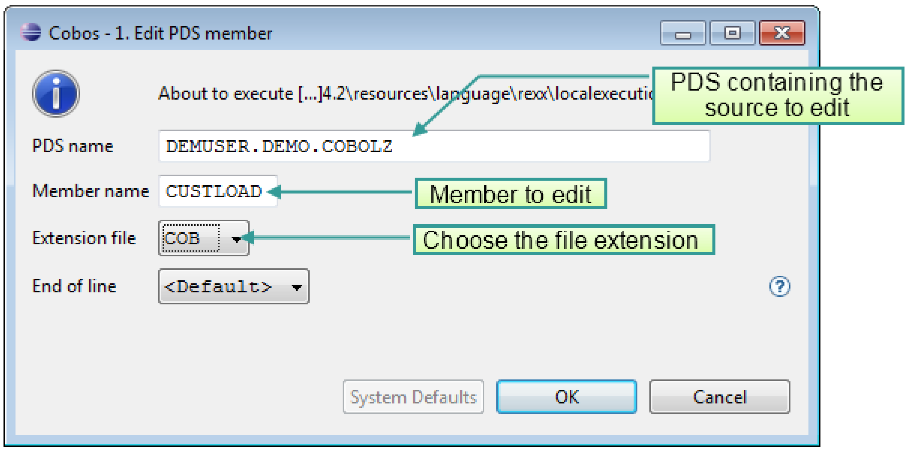
Here PDS target is DEMUSER.DEMO.COBOLZ and the member is CUSTLOAD.

8.2 Opening a PDS member

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



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


Cobos - 1. Edit PDS member

About to execute [...]4.2\resources\language\rexx\localexecuti

PDS name: DEMUSER.DEMO.COBOLZ

Member name: CUSTLOAD

Extension file: COB

End of line: <Default>

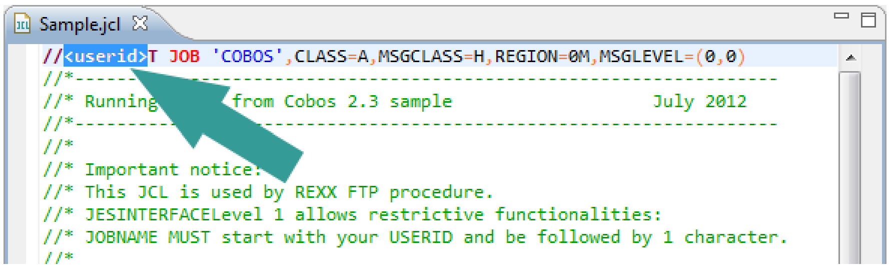
Here we edit the member "CUSTLOAD" in the PDS "DEMUSER.DEMO.COBOLZ" previously saved.

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.

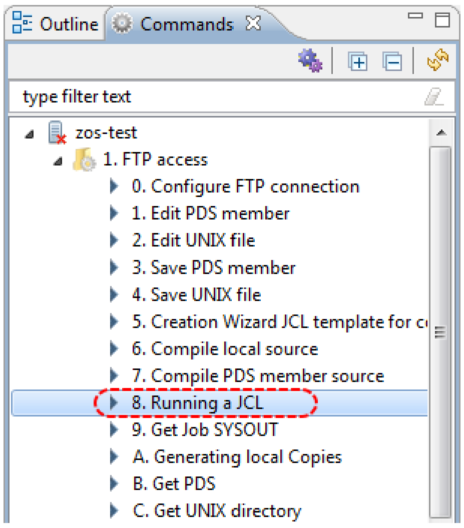
9.1 Submitting a jcl

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

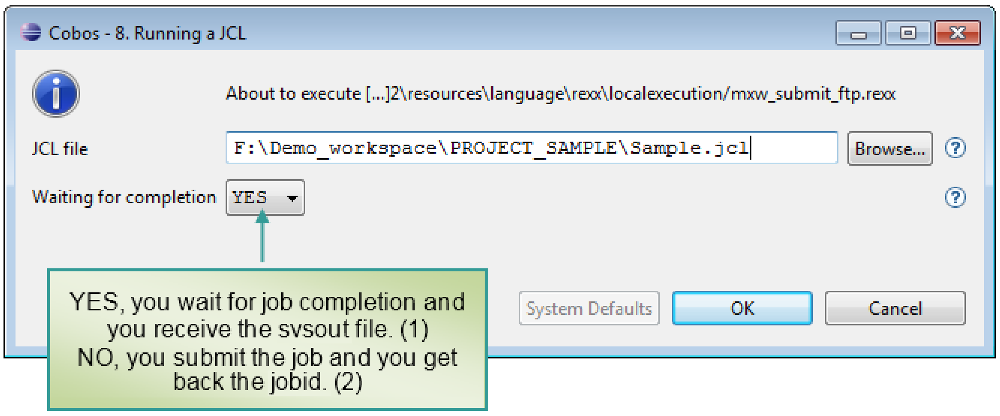


```
//<userid>T JOB 'COBOS',CLASS=A,MSGCLASS=H,REGION=0M,MSGLEVEL=(0,0)
//*------                   ---------------------------------------------------
//* Running              from Cobos 2.3 sample                    July 2012
//*---------                 ---------------------------------------------------
//*
//* 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 Cobos command **"1. FTP access ► 8. Running a JCL"**.



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



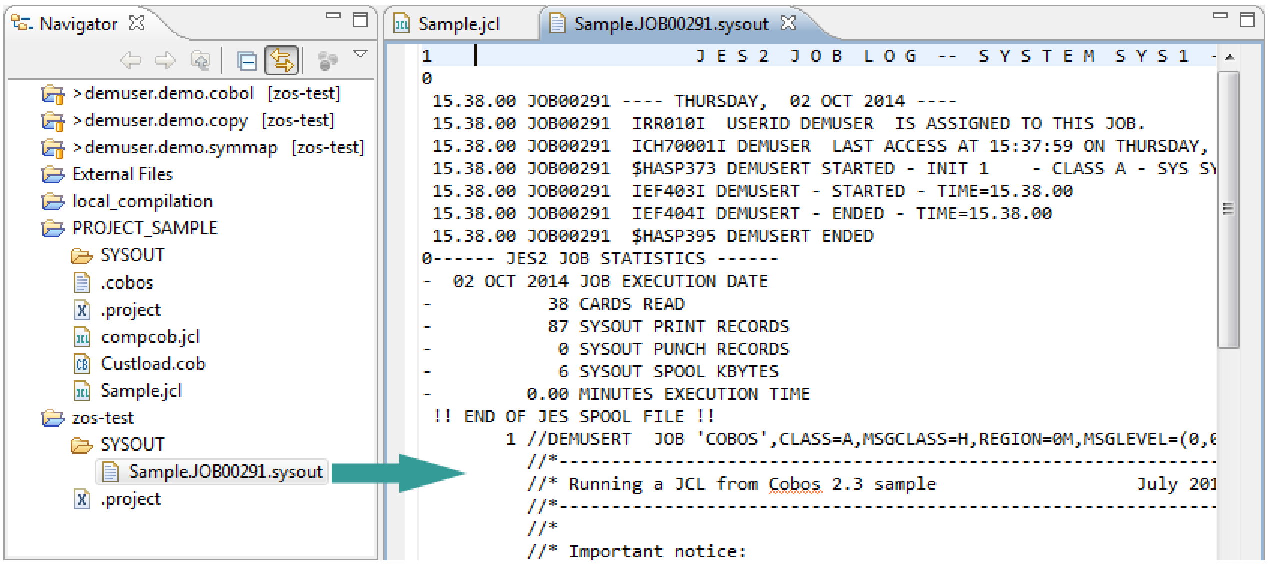
About to execute [...]2\resources\language\rexx\localexecution/mxw_submit_ftp.rexx

JCL file: F:\Demo_workspace\PROJECT_SAMPLE\Sample.jcl

Waiting for completion: YES

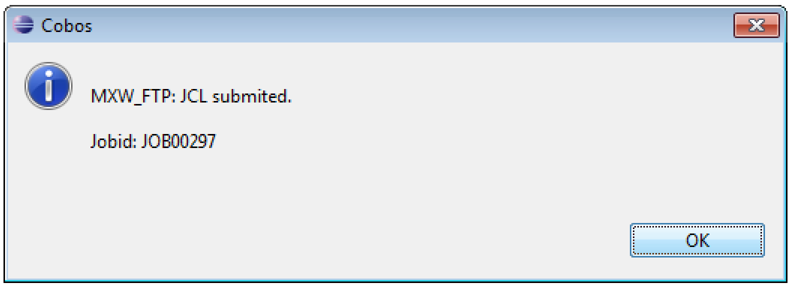
YES, you wait for job completion and you receive the svsout file. (1)
NO, you submit the job and you get back the jobid. (2)

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



```
1                       J E S 2  J O B  L O G  --  S Y S T E M  S Y S 1  -
0
 15.38.00 JOB00291 ---- THURSDAY,  02 OCT 2014 ----
 15.38.00 JOB00291  IRR010I  USERID DEMUSER  IS ASSIGNED TO THIS JOB.
 15.38.00 JOB00291  ICH70001I DEMUSER  LAST ACCESS AT 15:37:59 ON THURSDAY,
 15.38.00 JOB00291  $HASP373 DEMUSERT STARTED - INIT 1    - CLASS A - SYS SY
 15.38.00 JOB00291  IEF403I DEMUSERT - STARTED - TIME=15.38.00
 15.38.00 JOB00291  IEF404I DEMUSERT - ENDED - TIME=15.38.00
 15.38.00 JOB00291  $HASP395 DEMUSERT ENDED
0------ JES2 JOB STATISTICS ------
-   02 OCT 2014 JOB EXECUTION DATE
-            38 CARDS READ
-            87 SYSOUT PRINT RECORDS
-             0 SYSOUT PUNCH RECORDS
-             6 SYSOUT SPOOL KBYTES
-          0.00 MINUTES EXECUTION TIME
 !! END OF JES SPOOL FILE !!
        1 //DEMUSERT  JOB 'COBOS',CLASS=A,MSGCLASS=H,REGION=0M,MSGLEVEL=(0,0
          //*-----------------------------------------------------------------
          //* Running a JCL from Cobos 2.3 sample                   July 201
          //*-----------------------------------------------------------------
          //*
          //* Important notice:
```

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



MXW_FTP: JCL submited.

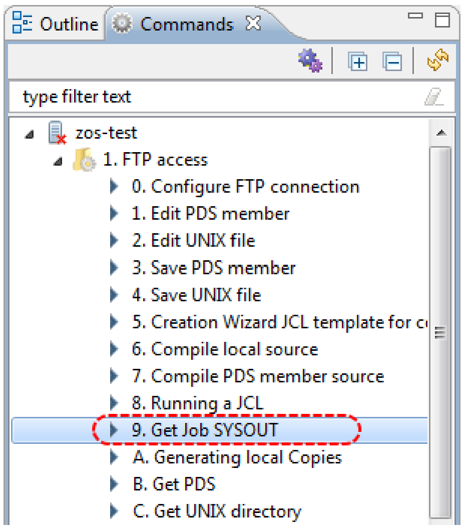
Jobid: JOB00297

Here Jobid is JOB00297.

9.2 Retrieving a sysout

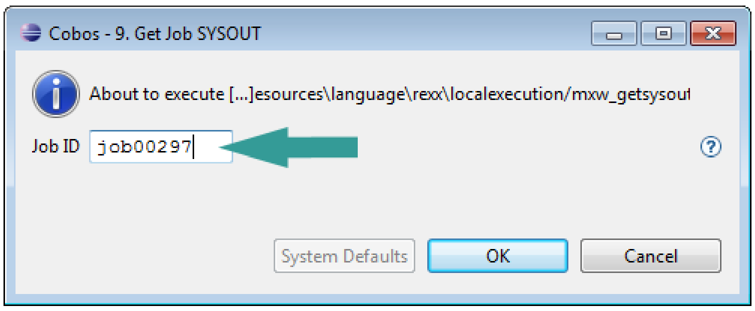
We'll get the sysout of the previously submitted job.

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



- zos-test
  - 1. FTP access
    - 0. Configure FTP connection
    - 1. Edit PDS member
    - 2. Edit UNIX file
    - 3. Save PDS member
    - 4. Save UNIX file
    - 5. Creation Wizard JCL template for c
    - 6. Compile local source
    - 7. Compile PDS member source
    - 8. Running a JCL
    - 9. Get Job SYSOUT
    - A. Generating local Copies
    - B. Get PDS
    - C. Get UNIX directory

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

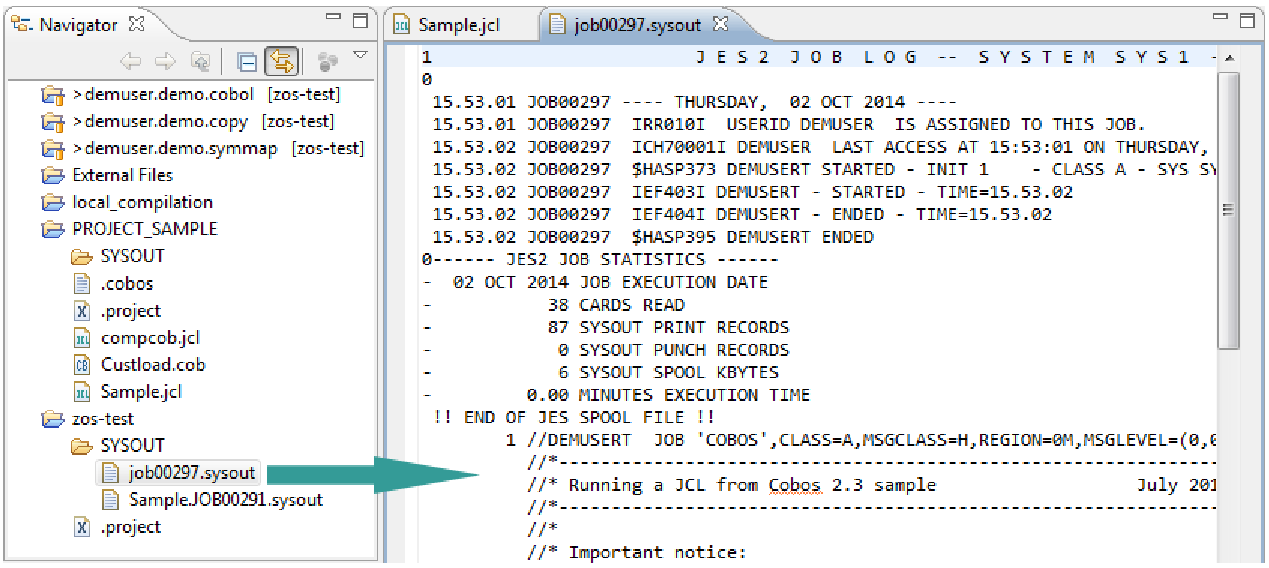


About to execute [...]esources\language\rexx\localexecution/mxw_getsysout

Job ID: job00297

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*.



- >demuser.demo.cobol [zos-test]
- >demuser.demo.copy [zos-test]
- >demuser.demo.symmap [zos-test]
- External Files
- local_compilation
- PROJECT_SAMPLE
  - SYSOUT
  - .cobos
  - .project
  - compcob.jcl
  - Custload.cob
  - Sample.jcl
- zos-test
  - SYSOUT
    - job00297.sysout
    - Sample.JOB00291.sysout
  - .project

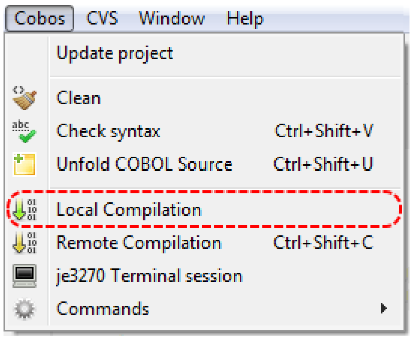
```
1                       J E S 2  J O B  L O G  --  S Y S T E M  S Y S 1  -
0
 15.53.01 JOB00297 ---- THURSDAY,  02 OCT 2014 ----
 15.53.01 JOB00297  IRR010I  USERID DEMUSER  IS ASSIGNED TO THIS JOB.
 15.53.02 JOB00297  ICH70001I DEMUSER  LAST ACCESS AT 15:53:01 ON THURSDAY,
 15.53.02 JOB00297  $HASP373 DEMUSERT STARTED - INIT 1    - CLASS A - SYS SY
 15.53.02 JOB00297  IEF403I DEMUSERT - STARTED - TIME=15.53.02
 15.53.02 JOB00297  IEF404I DEMUSERT - ENDED - TIME=15.53.02
 15.53.02 JOB00297  $HASP395 DEMUSERT ENDED
0------ JES2 JOB STATISTICS ------
-   02 OCT 2014 JOB EXECUTION DATE
-           38 CARDS READ
-           87 SYSOUT PRINT RECORDS
-            0 SYSOUT PUNCH RECORDS
-            6 SYSOUT SPOOL KBYTES
-        0.00 MINUTES EXECUTION TIME
 !! END OF JES SPOOL FILE !!
        1 //DEMUSERT  JOB 'COBOS',CLASS=A,MSGCLASS=H,REGION=0M,MSGLEVEL=(0,0
          //*-------------------------------------------------------------
          //* Running a JCL from Cobos 2.3 sample                    July 201
          //*-------------------------------------------------------------
          //*
          //* Important notice:
```

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”**.

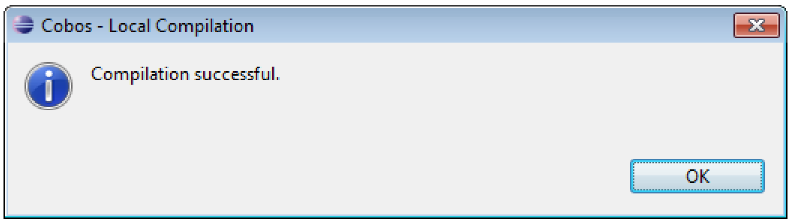
2 Select menu **“Cobos ► Local Compilation”**



Cobos
CVS
Window
Help
Update project
Clean
Check syntax Ctrl+Shift+V
Unfold COBOL Source Ctrl+Shift+U
Local Compilation
Remote Compilation Ctrl+Shift+C
je3270 Terminal session
Commands

or click on the “Local Compilation” button placed in the toolbar .

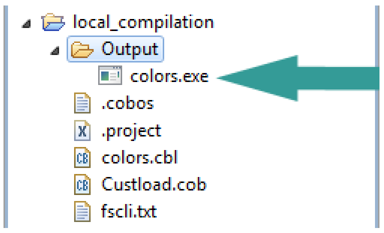
3 Once the compilation has been achieved, this popup is displayed.



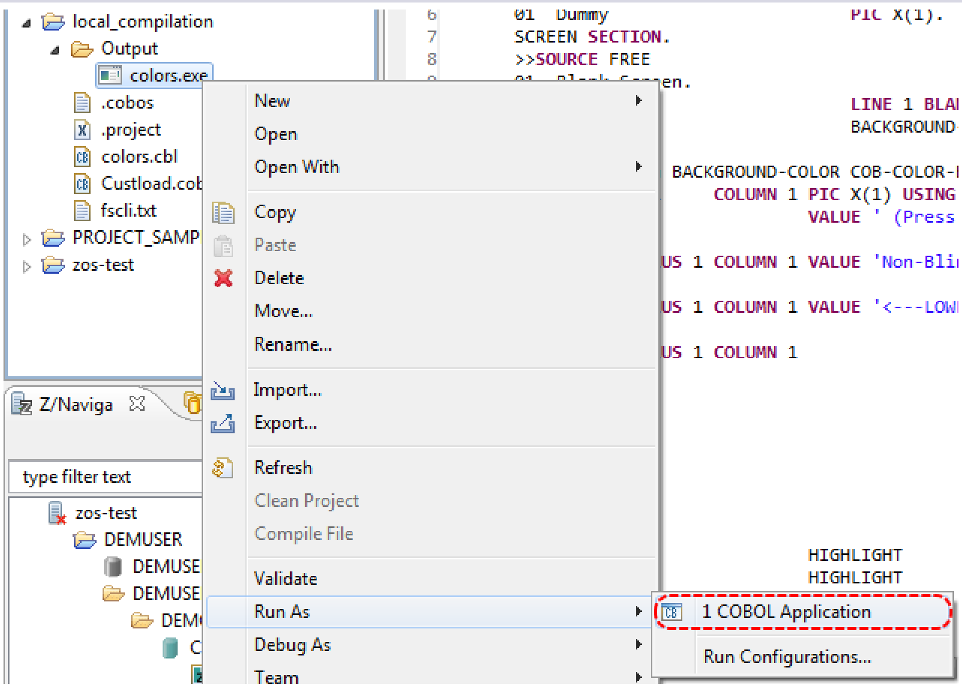
Compilation successful.

Just click on “OK” button.

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

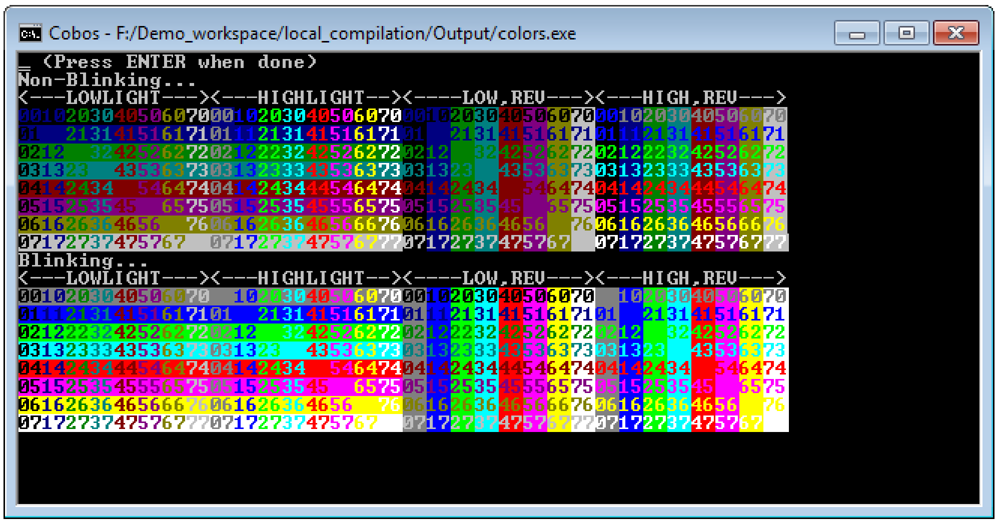


5 Right-click on “colors.exe” file and select **“Run As ► COBOL Application”**



```
       01  Dummy                          PIC X(1).
       SCREEN SECTION.
       >>SOURCE FREE
       01 Blank-Screen.
                                          LINE 1 BLA
                                          BACKGROUND
                         BACKGROUND-COLOR COB-COLOR-
                            COLUMN 1 PIC X(1) USING
                                   VALUE ' (Press
                        US 1 COLUMN 1 VALUE 'Non-Bli
                        US 1 COLUMN 1 VALUE '<---LOW
                        US 1 COLUMN 1
                                   HIGHLIGHT
                                   HIGHLIGHT
```

6 A console opens in which the program runs [1].


Cobos - F:/Demo_workspace/local_compilation/Output/colors.exe

```
_ (Press ENTER when done)
Non-Blinking...
<---LOWLIGHT---><---HIGHLIGHT--><----LOW,REV---><---HIGH,REV--->
0010203040506070001020304050607000102030405060700010203040506070
01  21314151617101112131415161710 1  2131415161710111213141516171
0212  32425262720212223242526272 0212  32425262720212223242526272
031323  4353637303132333435363730 31323  435363730313233343536373
04142434  5464740414243444546474 04142434  5464740414243444546474
0515253545  657505152535455565750 515253545  65750515253545556575
061626364656  760616263646566676 061626364656  760616263646566676
07172737475767  071727374757677707172737475767  0717273747576777
Blinking...
<---LOWLIGHT---><---HIGHLIGHT--><----LOW,REV---><---HIGH,REV--->
0010203040506070  102030405060700010203040506070  10203040506070
011121314151617101  2131415161710111213141516171 01  213141516171
02122232425262720212  324252627202122232425262720212  3242526272
0313233343536373031323  43536373031323334353637303 1323  43536373
041424344454647404142434  546474041424344454647404142434  546474
05152535455565750515253545  65750515253545556575 0515253545  6575
0616263646566676061626364656  7606162636465666760616263646  56  76
071727374757677707172737475767  071727374757677707172737475767
```

Press ENTER to quit the program.

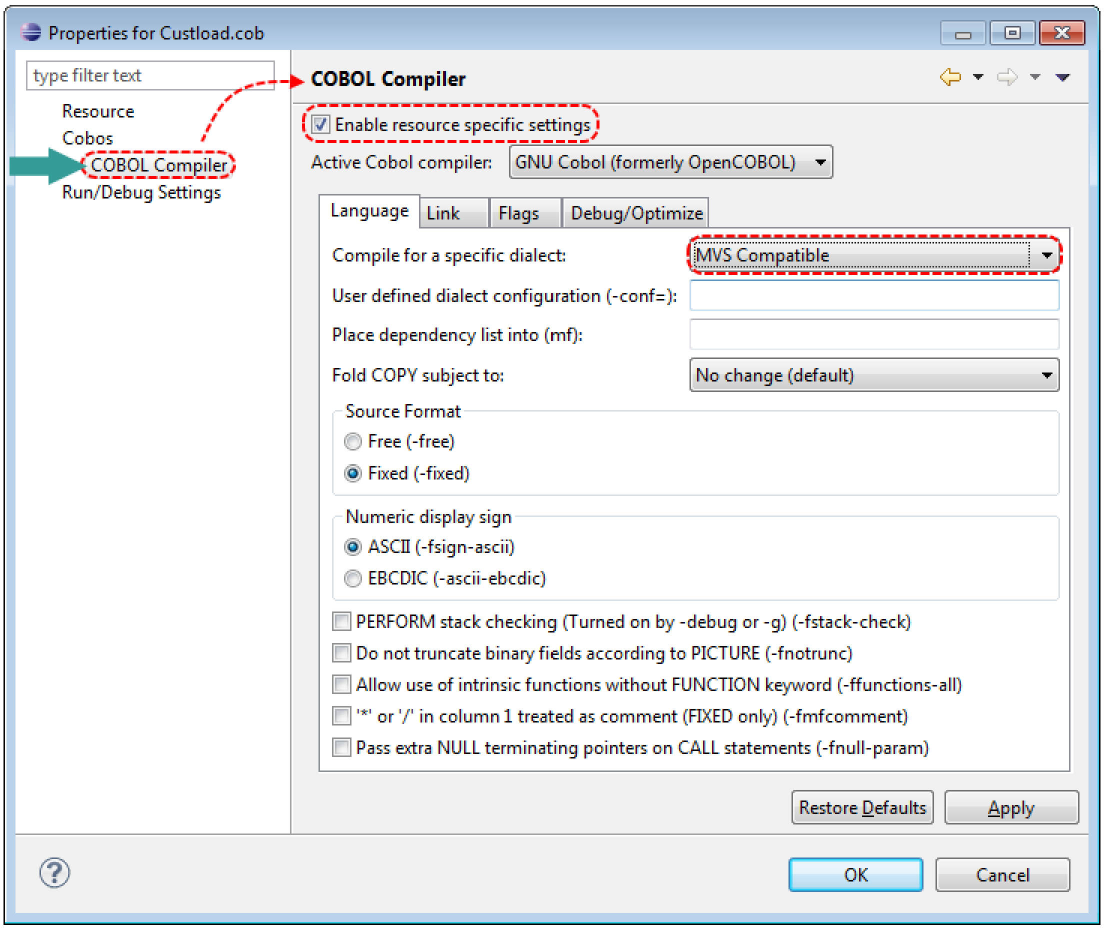
7 Press any key to close the console.

10.2 CUSTLOAD program

1 In “Navigator” view, select “Custload.cob” from “Local_compilation” project, right-click and select **“Properties”**

[1] “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 **"Cobos ► COBOL Compiler"** and check **"Enable resource specific settings"**



COBOL Compiler

☑ Enable resource specific settings

Active Cobol compiler: GNU Cobol (formerly OpenCOBOL)

Compile for a specific dialect: MVS Compatible

User defined dialect configuration (-conf=):

Place dependency list into (mf):

Fold COPY subject to: No change (default)

Source Format
- ○ Free (-free)
- ● Fixed (-fixed)

Numeric display sign
- ● ASCII (-fsign-ascii)
- ○ EBCDIC (-ascii-ebcdic)

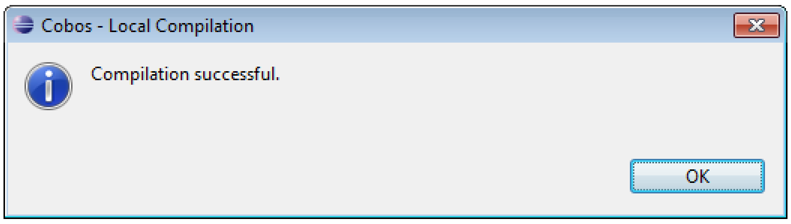
- ☐ PERFORM stack checking (Turned on by -debug or -g) (-fstack-check)
- ☐ Do not truncate binary fields according to PICTURE (-fnotrunc)
- ☐ Allow use of intrinsic functions without FUNCTION keyword (-ffunctions-all)
- ☐ '*' or '/' in column 1 treated as comment (FIXED only) (-fmfcomment)
- ☐ Pass extra NULL terminating pointers on CALL statements (-fnull-param)

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 **"Cobos ► Local Compilation"**.

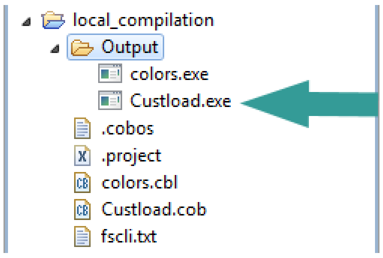
5 Once the compilation has been achieved, this popup is displayed.



Compilation successful.

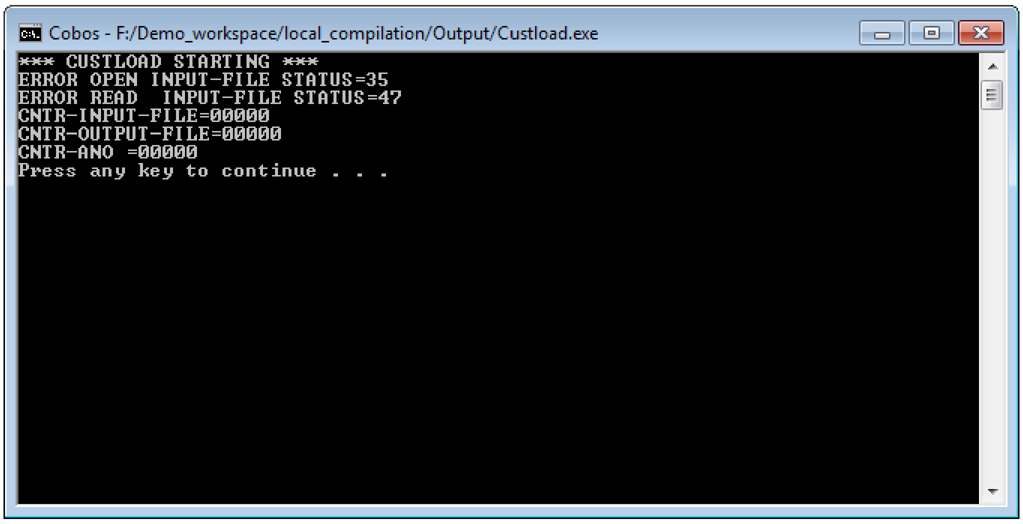
Just click on "OK" button.

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.

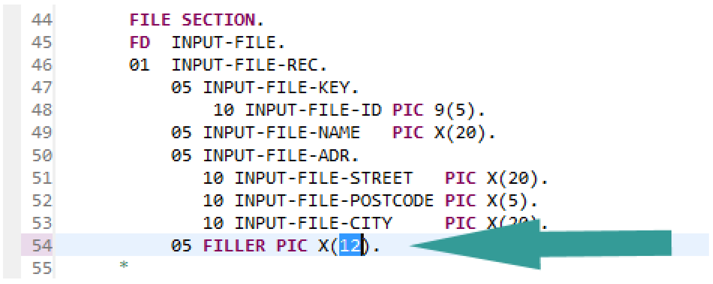

Cobos - F:/Demo_workspace/local_compilation/Output/Custload.exe

```
*** CUSTLOAD STARTING ***
ERROR OPEN INPUT-FILE STATUS=35
ERROR READ  INPUT-FILE STATUS=47
CNTR-INPUT-FILE=00000
CNTR-OUTPUT-FILE=00000
CNTR-ANO =00000
Press any key to continue . . .
```

The program exits[2] 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)**.

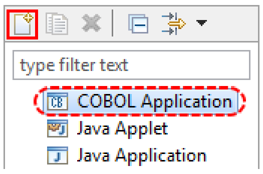


```
44        FILE SECTION.
45        FD  INPUT-FILE.
46        01  INPUT-FILE-REC.
47            05 INPUT-FILE-KEY.
48               10 INPUT-FILE-ID PIC 9(5).
49            05 INPUT-FILE-NAME    PIC X(20).
50            05 INPUT-FILE-ADR.
51               10 INPUT-FILE-STREET    PIC X(20).
52     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

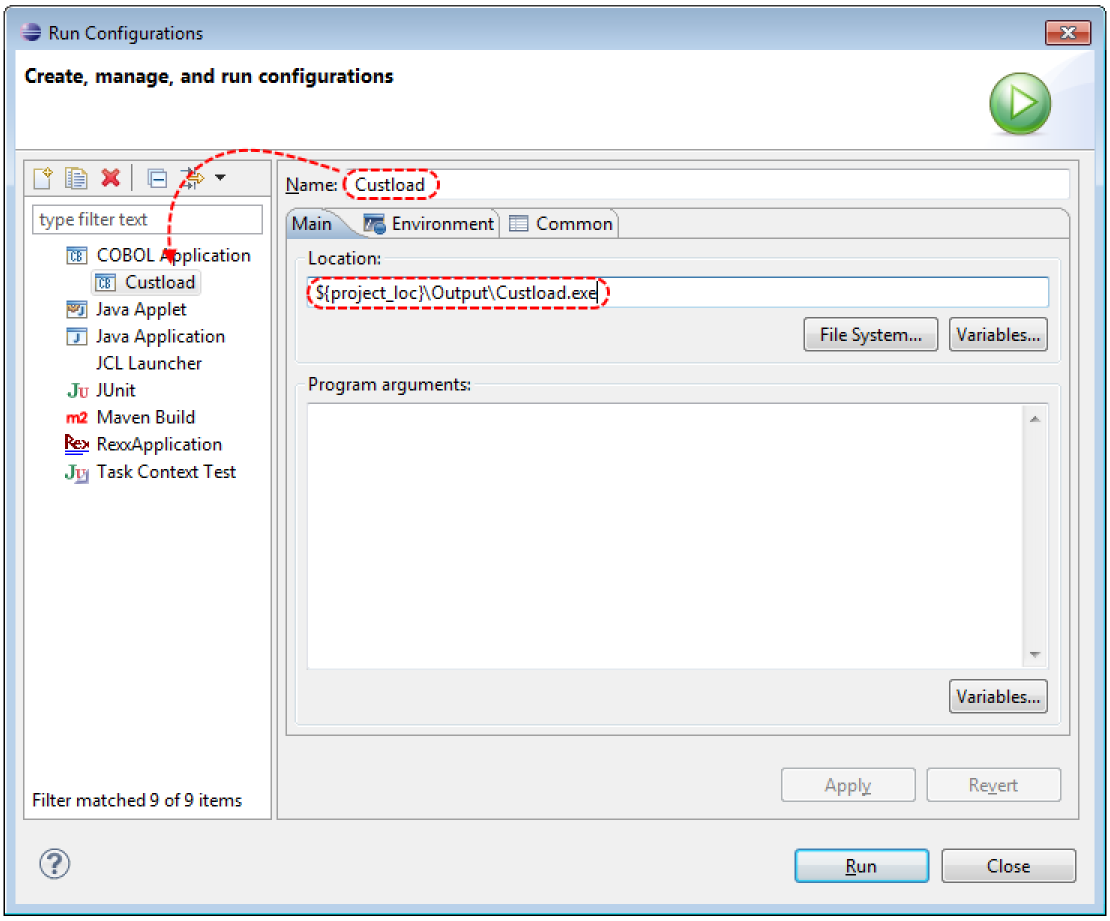
12 Right-click on “Custload.exe” file and select **“Run As ► Run Configuration...”**.

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.

[2] Sometimes the program hangs. Press Ctrl+C to stop it immediately.


Run Configurations

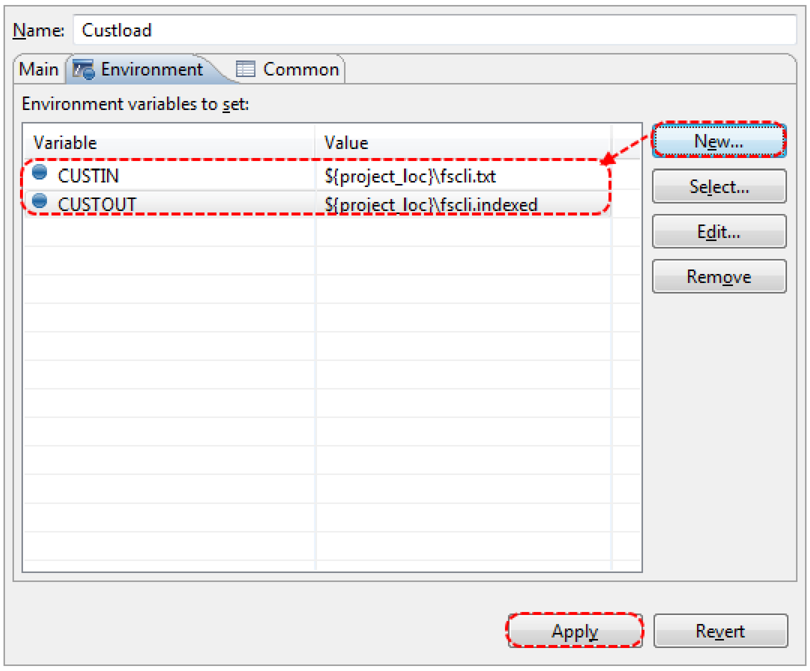
Create, manage, and run configurations

Name: Custload

Location: ${project_loc}\Output\Custload.exe

Program arguments:

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

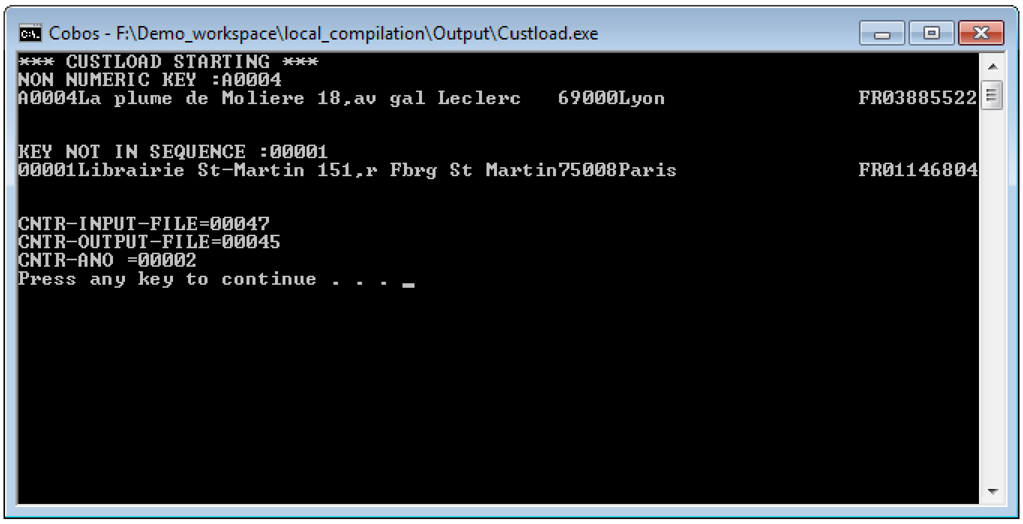

Name: Custload

Environment variables to set:

Variable	Value
CUSTIN	${project_loc}\fscli.txt
CUSTOUT	${project_loc}\fscli.indexed

Click on **“Apply”** button to save the configuration.

16 Push **“Run”** button to execute the program.

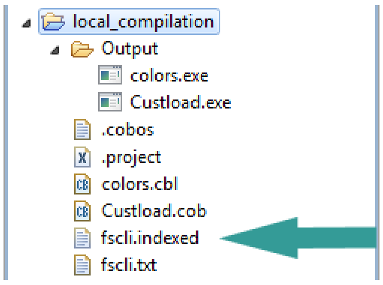

Cobos - F:\Demo_workspace\local_compilation\Output\Custload.exe

```
*** 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’


```
local_compilation
  Output
    colors.exe
    Custload.exe
  .cobos
  .project
  colors.cbl
  Custload.cob
  fscli.indexed
  fscli.txt
```