isCOBOL^{TM} Evolve

Cobol-WOW

Veryant's Cobol-WOW replacement

Introduction

Cobol-WOW is Liant Software Corporation's graphical user interface development tool for RM/COBOL. The goal of Veryant's Cobol-WOW replacement is to provide a new and modern environment based on Java. The WOW developer will find the same concepts used in Cobol-WOW designer in Veryant's solution.

All the standard GUI widgets were rewritten in Java and are supported. All the GUI widget behaviors were implemented so the WOW migrated application behaves as originally designed by the COBOL developer.

Veryant's IDE is able to import a Cobol-WOW project and allows the user to maintain the programs via the WOW screen designer and Cobol Source Editor.

Developer productivity has been enhanced by providing improved code navigation while editing in the IDE and debugging programs, and new debugger features have been implemented.

isCOBOL IDE-generated source code mimics the code you are familiar with from the RM/COBOL WOW Designer. COBOL developers have a choice between the WOW GUI and the isCOBOL GUI when designing new application features. WOW Screens and standard isCOBOL Screens can work together as long as they're handled by separate programs.

The isCOBOL runtime provides most of the WOW library routines to manage GUI widgets.

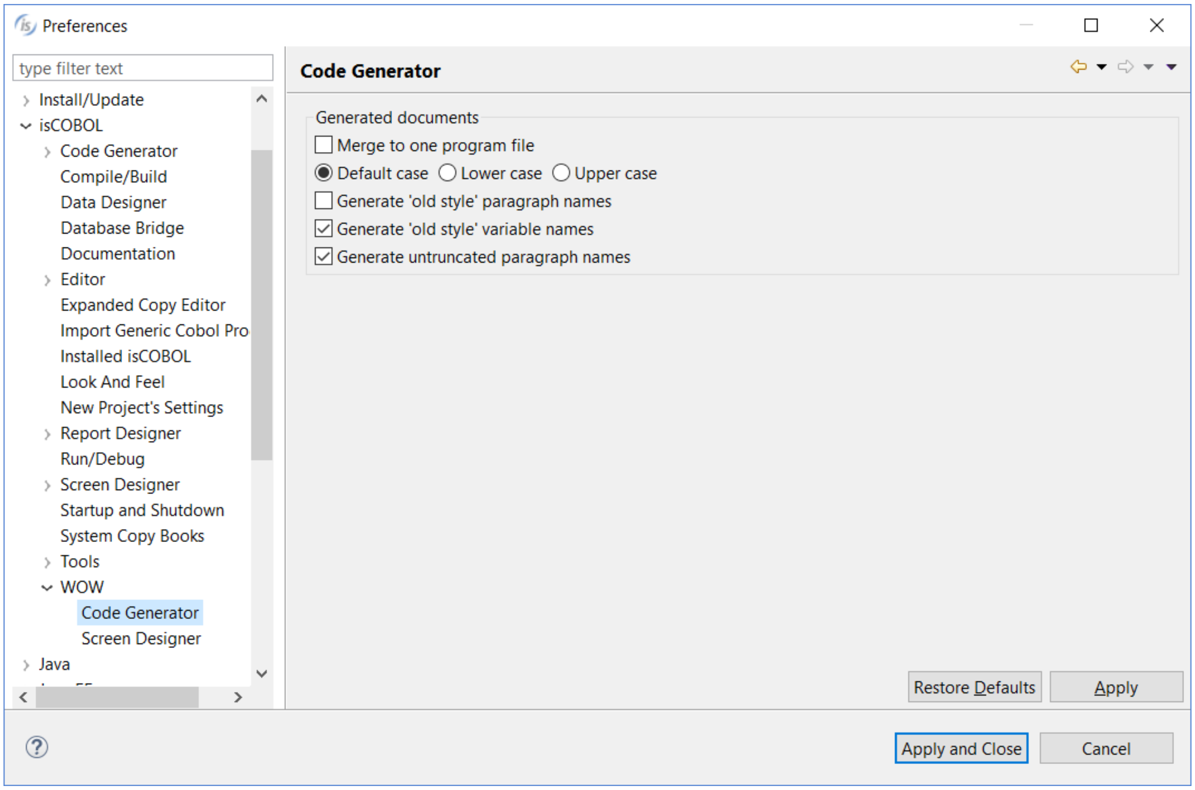
Cobol-WOW programs converted to isCOBOL can also run in thin client mode in an Application Server environment.

Before you start – Initial IDE configuration and creation of a new project

Before importing your Cobol-WOW projects in Veryant's IDE you should review the global preferences set.

1. Click on *Window* in the IDE menu bar and choose *Preferences*
2. Expand *isCOBOL* in the tree, then expand *WOW*

Code Generator



Code Generator

Generated documents

☐ Merge to one program file

● Default case ○ Lower case ○ Upper case

☐ Generate 'old style' paragraph names

☑ Generate 'old style' variable names

☑ Generate untruncated paragraph names

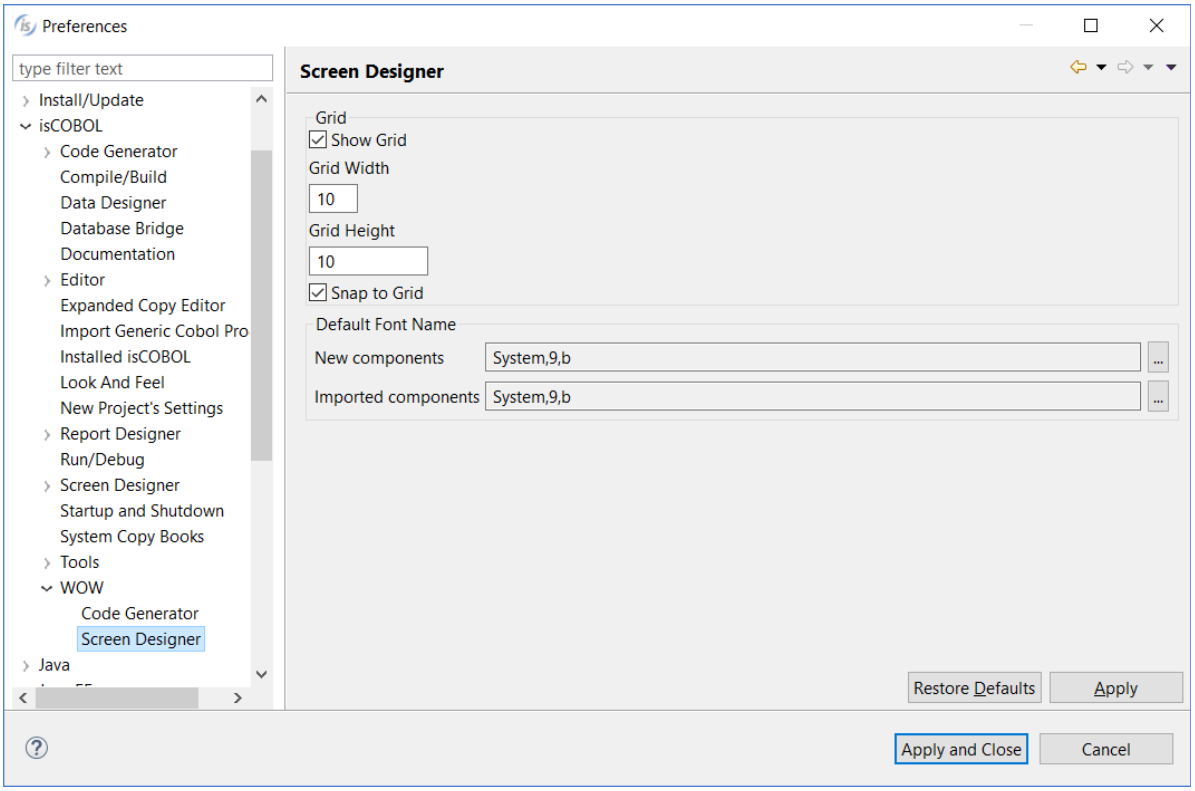
Here you can customize the way isCOBOL IDE generates the source code of your Cobol-WOW programs:

Merge to one program file	If checked, a single CBL file is generated. Otherwise, a CBL file and multiple copybooks are generated.
Default case | Lower case | Upper case	The case used for COBOL identifiers and

	keywords. By default, a lower case is used.
Generate 'old style' paragraph names	This setting reflects the same setting available in Cobol-WOW.[1]
Generate 'old style' variable names	This setting reflects the same setting available in Cobol-WOW.[1]
Generate untruncated paragraph names	This setting reflects the same setting available in Cobol-WOW.[1]

[1]These settings affect only new programs created in the isCOBOL IDE. Programs imported from existing Cobol-WOW projects maintain the settings they had in Cobol-WOW.

Screen Designer



Grid

☑ Show Grid

Grid Width: 10

Grid Height: 10

☑ Snap to Grid

Default Font Name

New components: System,9,b

Imported components: System,9,b

Here you can configure the Screen Designer:

Show Grid	If checked, a grid is shown as

	background of the window in the Screen Designer. The width and height in pixels can be configured in the fields below. The grid helps in dimensioning controls when you draw them with the mouse.
Snap to Grid	If checked, the Screen Designer automatically aligns controls to the nearest cell. It helps in having controls aligned to each other.
Default font name	Specifies the font that is used by default for the new controls drawn in the Screen Designer as well as for the controls imported from existing Cobol-WOW projects if they lack font attributes.

Before importing Cobol-WOW projects you should create an empty isCOBOL Project in the IDE.

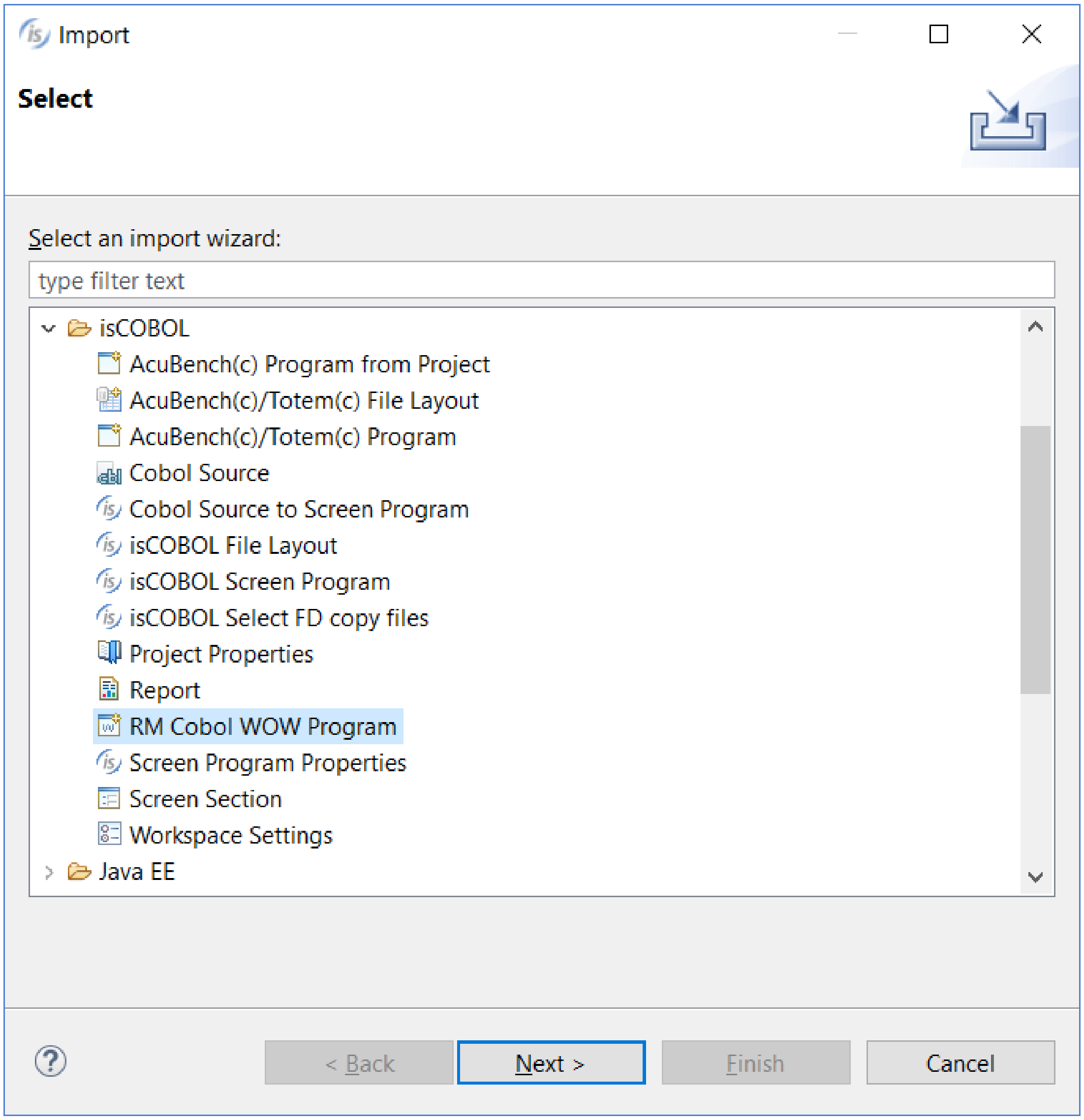
1. Click on *File* in the menu bar and choose *New*
2. Select *isCOBOL Project* from the sub menu
3. Follow the wizard procedures to until the end.

Importing projects from Cobol-WOW in isCOBOL IDE

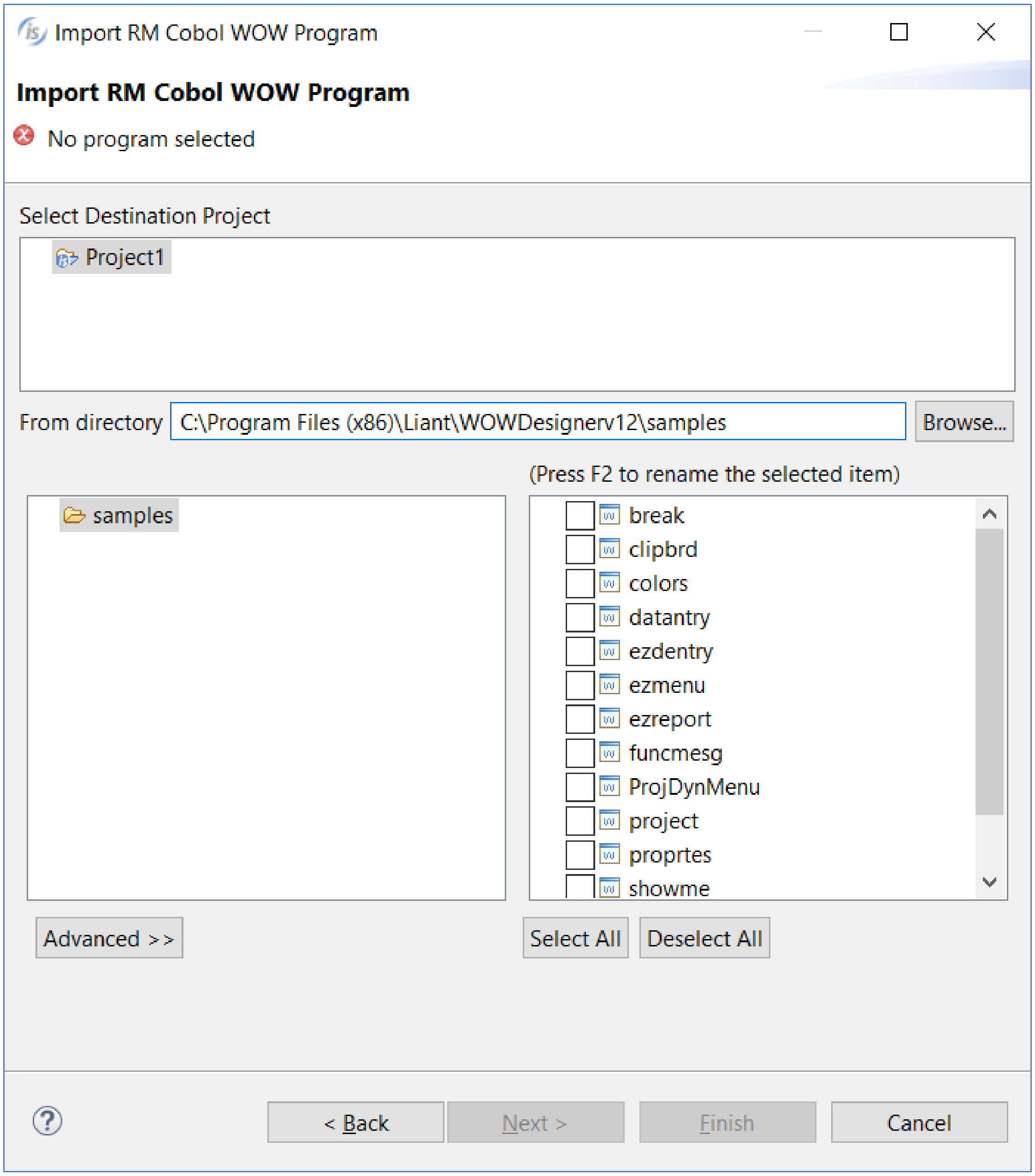
Veryant's IDE is able to import programs from existing Cobol-WOW projects.

In order to import projects from Cobol-WOW in the IDE:

1. Click on *File* in the menu bar and choose *Import*
2. Expand the *isCOBOL* group and select *RM Cobol WOW Program*



3. Browse for the folder where the Cobol-WOW projects are stored and press Enter to load the list of available programs:



Import RM Cobol WOW Program

Import RM Cobol WOW Program

No program selected

Select Destination Project

Project1

From directory C:\Program Files (x86)\Liant\WOWDesignerv12\samples

(Press F2 to rename the selected item)

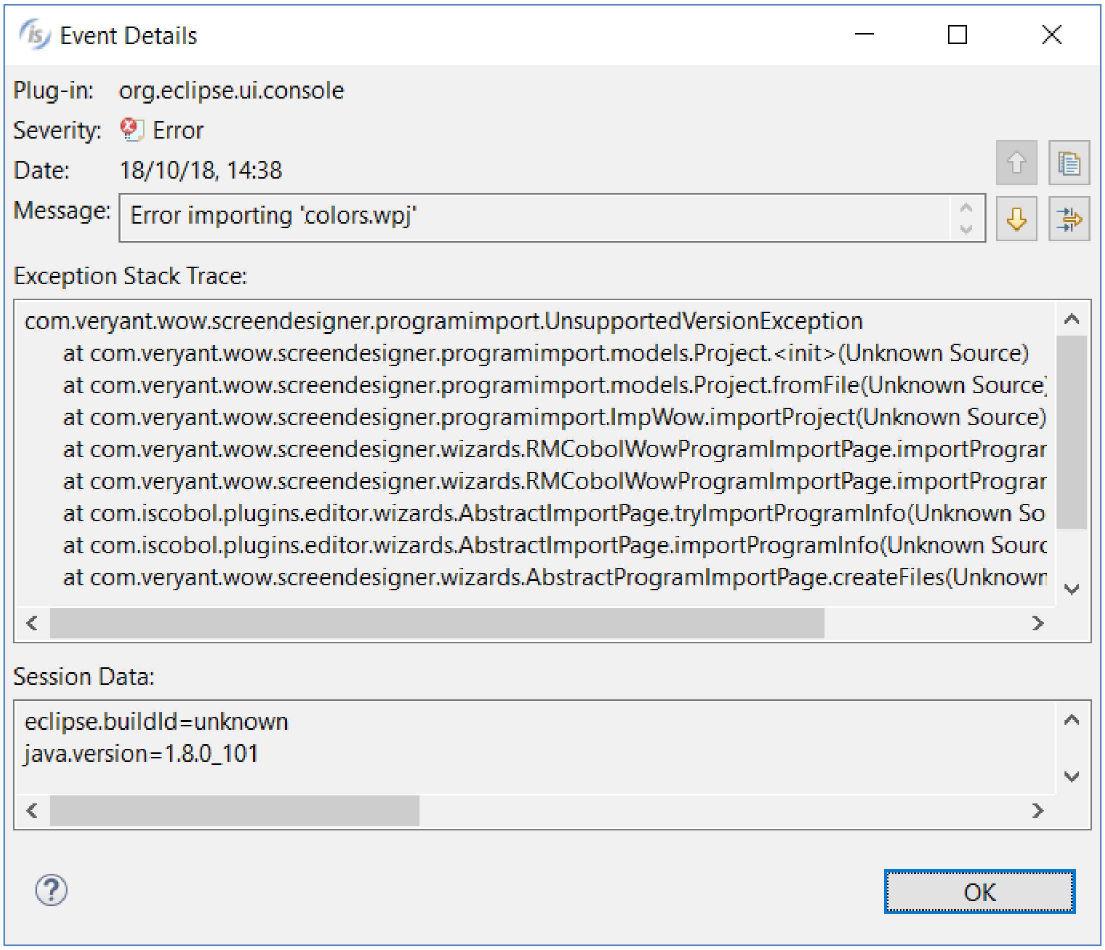
samples

- □ break
- □ clipbrd
- □ colors
- □ datantry
- □ ezdentry
- □ ezmenu
- □ ezreport
- □ funcmesg
- □ ProjDynMenu
- □ project
- □ proprtes
- □ showme

The Advanced button allows you to choose which character set was used in the code of the Cobol-WOW projects. Use it if you notice wrongly decoded characters on the screen after the import.

4. Select the desired programs and click *Finish*.

Note that only projects of version 12 are supported. If you try to import a Cobol-WOW project whose version is unsupported, then the program doesn't appear in the IDE and the following exception is shown in the Error Log view:



Plug-in: org.eclipse.ui.console
Severity: Error
Date: 18/10/18, 14:38
Message: Error importing 'colors.wpj'

Exception Stack Trace:

```
com.veryant.wow.screendesigner.programimport.UnsupportedVersionException
    at com.veryant.wow.screendesigner.programimport.models.Project.<init>(Unknown Source)
    at com.veryant.wow.screendesigner.programimport.models.Project.fromFile(Unknown Source
    at com.veryant.wow.screendesigner.programimport.ImpWow.importProject(Unknown Source)
    at com.veryant.wow.screendesigner.wizards.RMCobolWowProgramImportPage.importPrograr
    at com.veryant.wow.screendesigner.wizards.RMCobolWowProgramImportPage.importPrograr
    at com.iscobol.plugins.editor.wizards.AbstractImportPage.tryImportProgramInfo(Unknown So
    at com.iscobol.plugins.editor.wizards.AbstractImportPage.importProgramInfo(Unknown Sourc
    at com.veryant.wow.screendesigner.wizards.AbstractProgramImportPage.createFiles(Unknowr
```

Session Data:

```
eclipse.buildId=unknown
java.version=1.8.0_101
```

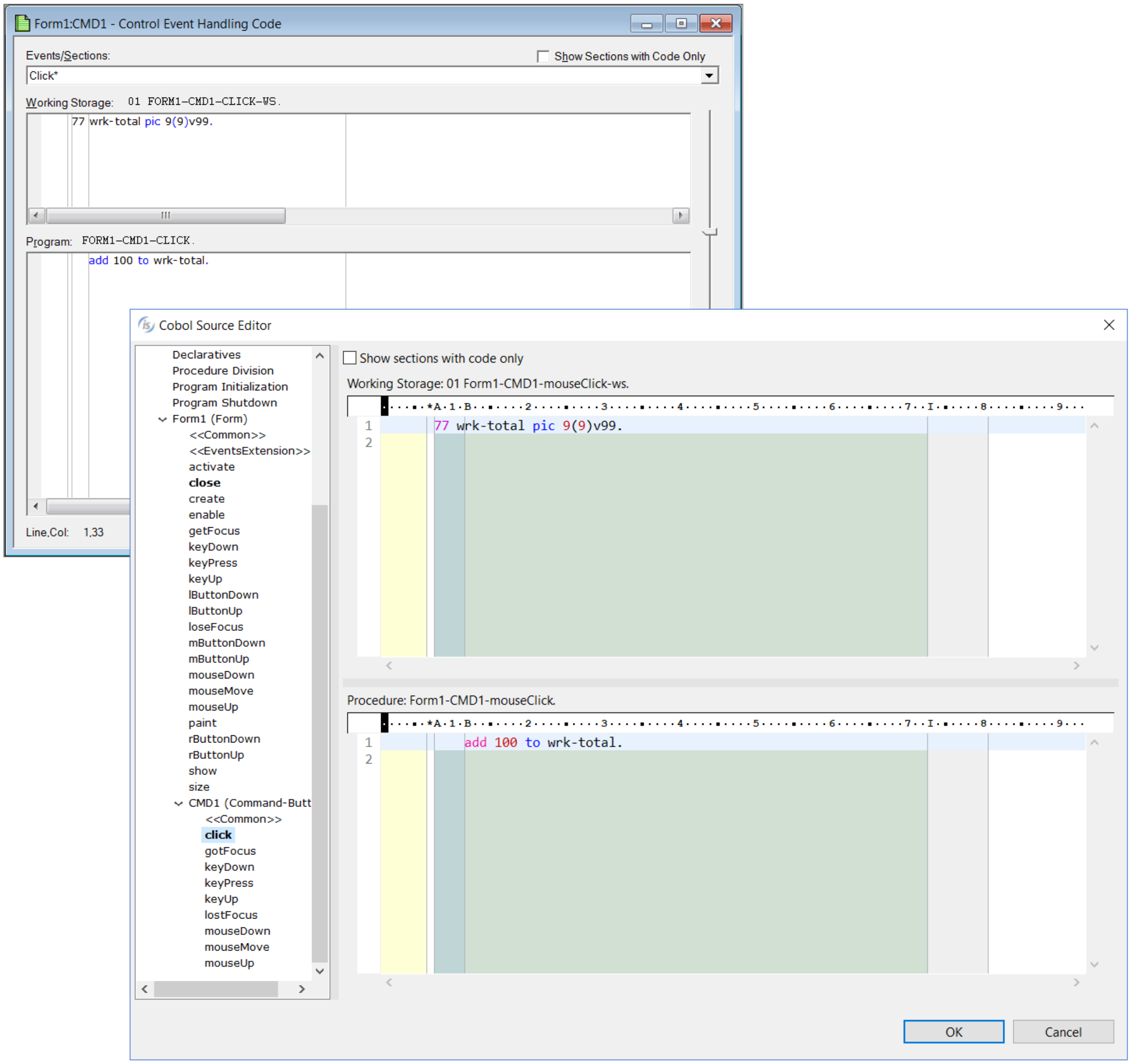
In order to get rid of the error, you can open the Cobol-WOW project with a Cobol-WOW designer version 12, make a small useless modification and save. In this way the project files will be compatible with isCOBOL IDE.

Editing, compiling and running Cobol-WOW programs

Once the Cobol-WOW project has been imported in the IDE, the user can edit it in the same way he did with the Cobol-WOW Designer: draw controls by dragging them from the palette to the screen and edit the code by double clicking on the control.

The code editor in isCOBOL's IDE is slightly different from Cobol-WOW's Designer.

Cobol-WOW allows you to choose the section from a combo box, while isCOBOL IDE shows all the sections in a tree view on the left:



Form1:CMD1 - Control Event Handling Code

Events/Sections: Click*

□ Show Sections with Code Only

Working Storage: 01 FORM1-CMD1-CLICK-WS.

```
77 wrk-total pic 9(9)v99.
```

Program: FORM1-CMD1-CLICK.

```
add 100 to wrk-total.
```

Line,Col: 1,33

Cobol Source Editor

□ Show sections with code only

Working Storage: 01 Form1-CMD1-mouseClick-ws.

```
77 wrk-total pic 9(9)v99.
```

Procedure: Form1-CMD1-mouseClick.

```
add 100 to wrk-total.
```

In order to generate the source code of your programs:

1. Right click on the program name in the Project Explorer's Structural View
2. Select *Generate Program* from the menu

Or

1. Left click on the program name in the Project Explorer's Structural View
2. Click on the *Generate Program* button (⚙) in the IDE's tool-bar

In order to compile a program:

1. Right click on the program name in the Project Explorer's Structural View
2. Select *Compile* from the menu

Or

1. Left click on the program name in the Project Explorer's Structural View
2. Click on the *Compile* button (Ⓒ) in the IDE's tool-bar

Note that, if you compile a program without generating it first, the IDE generates it automatically before compiling.

In order to execute a program:

1. Right click on the program name in the Project Explorer's Structural View
2. Choose *Run As* from the menu
3. Choose *isCOBOL Application* from the sub menu[1]

Or

1. Left click on the program name in the Project Explorer's Structural View
2. Click on the *Run* button (▶) in the IDE's tool-bar

[1] Note that Cobol-WOW programs can't run in WD2 environment. If you select *isCOBOL EIS WD2* from the *Run As* menu, the effects are unpredictable.

In order to debug a program:

1. Right click on the program name in the Project Explorer's Structural View
2. Choose *Debug As* from the menu
3. Choose *isCOBOL Application* from the sub menu[1]

Or

3. Left click on the program name in the Project Explorer's Structural View
4. Click on the *Debug* button (🐞) in the IDE's tool-bar

[1] Note that Cobol-WOW programs can't run in WD2 environment. If you select *isCOBOL EIS WD2* from the *Debug As* menu, the effects are unpredictable.

The isCOBOL Debugger is different than the one in Cobol-WOW. Refer to the isCOBOL documentation for details about the usage of the isCOBOL Debugger.

Note that pressing the *Pause* button on the keyboard while debugging a WOW program doesn't enter the debugger like it does with standard GUI COBOL programs. In order to enter the debugger, you need to click on the *Pause* button (⏸) in the Debugger tool-bar.

Known differences and things to know

Concepts

Cobol-WOW project management is very simple: all files are generated in the same folder, there is no structure. There is a project for each GUI program. Each project references one or more forms and menus.

The Eclipse-based isCOBOL IDE from Veryant instead provides a structured management of the projects. There is a workspace, a disk folder where project files are physically saved. The workspace can host one or more projects. Each project can host one or more programs. All the program items (e.g. form and menu) are included in the program instead of being referenced as separate files.

	Cobol-WOW	isCOBOL IDE
Programs container	Disk folder	Project
Program entity	Project	Program
Form entity	Form	n/a[1]
Menu entity	Menu	n/a[1]

[1] The form and the menu are included in the program.

Project files

Veryant’s IDE uses different file formats and extensions for the project’s file.

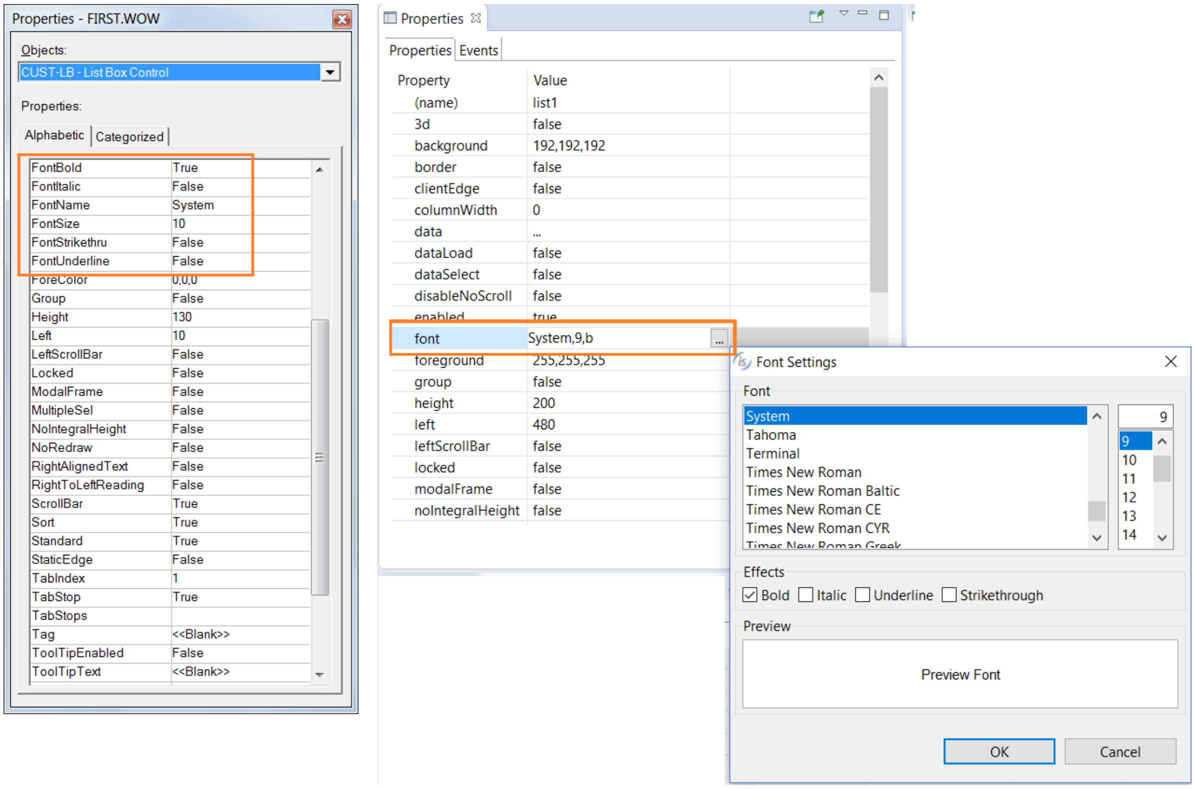
	Cobol-WOW	isCOBOL IDE
Program file extension	Wjp	wsp
Form file extension	Wow	n/a[1]
Menu file extension	wmn	n/a[1]

[1] The information is included in the wsp file.

Screen Designer

There are some differences between Veryant's IDE screen painter and Cobol-WOW's screen designer.

The main difference is in the list of attributes of controls. When you select a graphical control on the screen, the list of attributes is populated in the Properties view. Cobol-WOW shows several attributes while Veryant's IDE tries to group them where applicable. For example, for most of the controls you can set the font. The screenshot below shows a comparison between the multiple font properties shown by Cobol-WOW (on the left) and the single font property shown by Veryant's IDE (on the right).



Properties - FIRST.WOW

Objects: CUST-LB - List Box Control

Properties:

FontBold	True
FontItalic	False
FontName	System
FontSize	10
FontStrikethru	False
FontUnderline	False
ForeColor	0,0,0
Group	False
Height	130
Left	10
LeftScrollBar	False
Locked	False
ModalFrame	False
MultipleSel	False
NoIntegralHeight	False
NoRedraw	False
RightAlignedText	False
RightToLeftReading	False
ScrollBar	True
Sort	True
Standard	True
StaticEdge	False
TabIndex	1
TabStop	True
TabStops	
Tag	<<Blank>>
ToolTipEnabled	False
ToolTipText	<<Blank>>

Property	Value
(name)	list1
3d	false
background	192,192,192
border	false
clientEdge	false
columnWidth	0
data	...
dataLoad	false
dataSelect	false
disableNoScroll	false
enabled	true
font	System,9,b
foreground	255,255,255
group	false
height	200
left	480
leftScrollBar	false
locked	false
modalFrame	false
noIntegralHeight	false

Font Settings

Font: System (Tahoma, Terminal, Times New Roman, Times New Roman Baltic, Times New Roman CE, Times New Roman CYR, Times New Roman Greek); size: 9 (9, 10, 11, 12, 13, 14)

Effects: ☑ Bold ☐ Italic ☐ Underline ☐ Strikethrough

Preview: Preview Font

List-Box and Status-Bar have properties to define the multiple items in WOW designer. In Veryant's IDE instead there is an item designer that shows a pop-up dialog in which the user can configure the items contained in the control.

Bitmaps

When you run a WOW program from Veryant's IDE, bitmaps are loaded from the working directory, which is the "output" folder of the IDE project. The bitmaps are not searched for in the Java Classpath as normally happens with standard isCOBOL programs. If you run a WOW program in thin client mode, then bitmaps are searched for on the client machine and not on the server machine as normally happens with standard isCOBOL programs.

Form and menu sharing

In Cobol-WOW forms and menus are identified by separate files and they're referenced by the Cobol-WOW project. This means that multiple projects can reference the same form or the same menu and altering such form or menu affects all the projects in which the form or menu is referenced.

The same concept is not applicable to the isCOBOL IDE. The IDE keeps the information about form and menu inside the program file. If you edit a form or a menu, only the program you're editing is affected. However, when the source code is generated, a copybook with the same name of the form is generated on disk, so if multiple programs include a form with the same name, an overwrite of the copybook may occur. The same happens with menus.

Menu editor

The maintenance of the menu bar of the form in Cobol-WOW is done via the menu designer, activated by a dedicated button in the Designer tool-bar. In isCOBOL IDE instead the menu editor is activated by editing the menu property of the form.

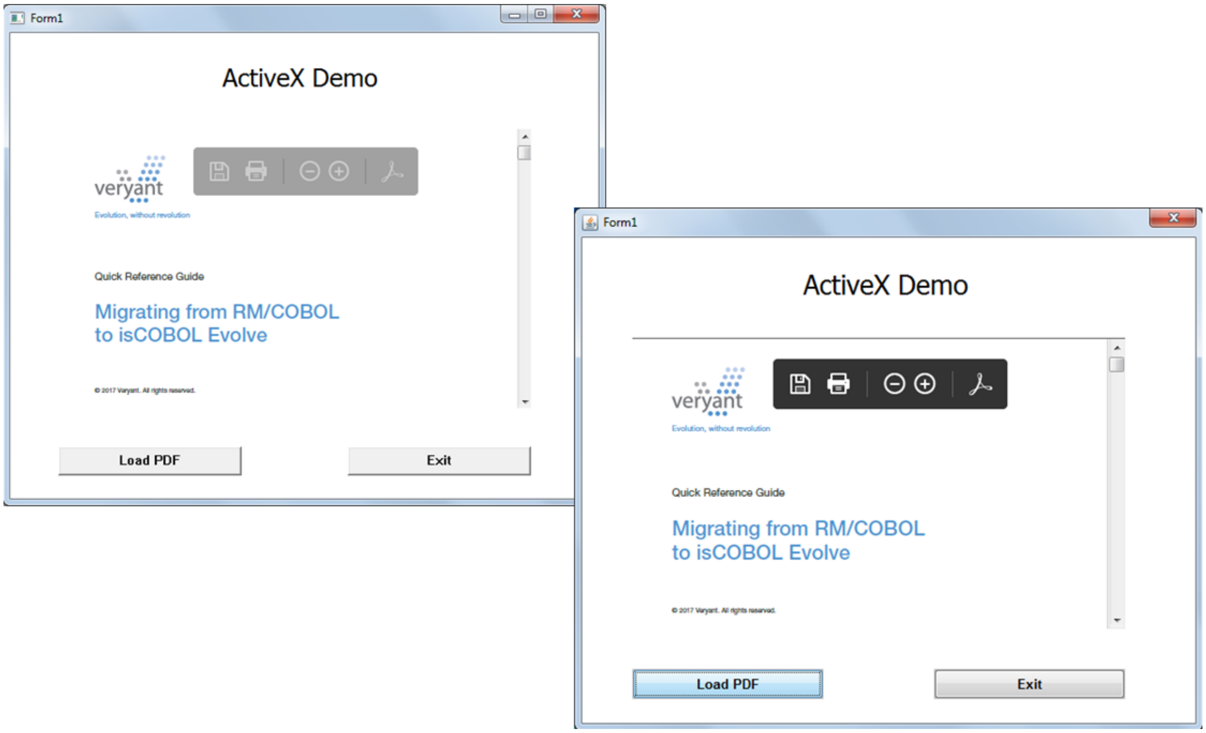
ActiveX

The ActiveX technology is in legacy status and is no longer supported by Microsoft.

Veryant's approach on ActiveX is to provide tools and migration service to re-implement all ActiveX controls as Java components

isCOBOL IDE's import wizard reads the ActiveX resources used in a WOW project and imports all its properties with the correct values. The IDE relies on the isCOBOL utility AxTools to generate a Java template for each ActiveX control to be implemented. AxTools reads all properties, events and methods available from ActiveX and OCX controls and generates a Java source skeleton that users can modify in order to implement the properties, methods and events of the ActiveX in Java.

The screenshot below shows how the IExplorer ActiveX used in a Cobol-WOW form (on the left) was reimplemented using the Java component DJBrowser (on the right):



Veryant will provide some source code examples of ActiveX controls from DBI-Technologies rewritten in Java.

Contact Veryant for more information about ActiveX replacement.

Supported library routines

isCOBOL supports the following library routines of Cobol-WOW:

- AXBINDEVENTARGUMENTS
- AXDOMETHOD
- AXUNBINDEVENTARGUMENTS
- CHECKMENUITEM
- CLOSEWINDOW
- DELETEMENU
- DRAWMENUBAR
- ENABLEMENUITEM
- ENABLEWINDOW
- FINDWINDOW
- GETACTIVEWINDOW
- GETCURSORPOS
- GETENVIRONMENTVARIABLE
- GETFOCUS
- GETMENU
- GETSUBMENU
- GETWINDOWSDIRECTORY
- ISCHILD
- ISWINDOW
- MESSAGEBEEP
- MESSAGEBOX
- MODIFYMENU
- OPENICON

- SENDMESSAGE
- SETACTIVEWINDOW
- SETFOCUS
- SHOWWINDOW
- WINEXEC
- WOWADDITEM
- WOWCLEAR
- WOWCLEARWAITCURSOR
- WOWCREATEWINDOW
- WOWDESTROYWINDOW
- WOWDISCARDEVENTS
- WOWGETFOCUS
- WOWGETINDEXPROP
- WOWGETMESSAGE
- WOWGETNUM
- WOWGETPROP
- WOWINITALLCONTROLS
- WOWINITCONTROL
- WOWMESSAGEBOX
- WOWMOVE
- WOWMULTICONTROLGETPROP
- WOWMULTICONTROLSETPROP
- WOWPEEKMESSAGE
- WOWREFRESH
- WOWREMOVEITEM
- WOWRESETWAITCURSOR

- WOWSETFOCUS
- WOWSETINDEXPROP
- WOWSETNEXTCTRL
- WOWSETNUM
- WOWSETPREVCTRL
- WOWSETPROP
- WOWSETSTRIPTRAILING
- WOWSETWAITCURSOR
- WOWVERSION[1]

[1] WOWVERSION always returns 0.