



# isCOBOL™ Evolve

## isCOBOL Evolve 2013 Release 1 (R1) Overview

Copyright © 2013 Veryant LLC.

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and recompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Veryant and its licensors, if any.

Veryant and isCOBOL are trademarks or registered trademarks of Veryant LLC in the U.S. and other countries. All other marks are property of their respective owners.

## **isCOBOL Evolve 2013 Release 1 (R1) Overview**

### Introduction

Veryant is pleased to announce the latest release of isCOBOL™ Evolve, isCOBOL Evolve 2013 R1. isCOBOL Evolve provides a complete environment for the development, deployment, maintenance, and modernization of COBOL applications. isCOBOL 2013 R1 includes a new product called isCOBOL Mobile, several enhancements to the isCOBOL IDE, to the Utilities and to the isCOBOL Server product; isCOBOL 2013 R1 also includes many Debugger improvements, and other new features.

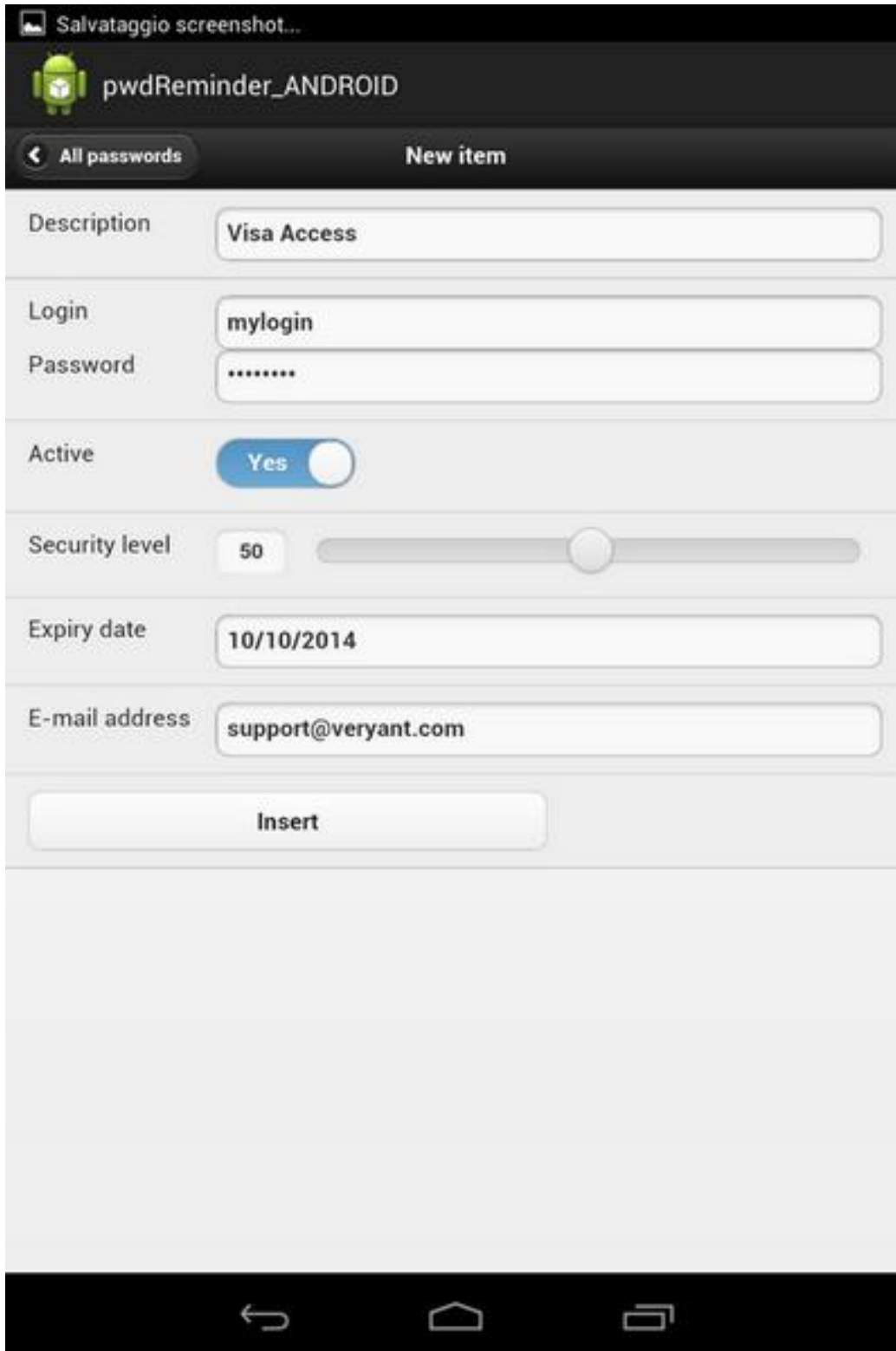
Details on these enhancements and updates are included below.

### isCOBOL Mobile

The isCOBOL Mobile product allows to bring COBOL code on mobile devices. The goal is to reuse the existing backend COBOL logic as well as sequential, relative and indexed files (in Jlsam format) on a mobile application while the UI is rewritten using HTML and JavaScript. IsCOBOL 2013 R1 reaches the objective through two new features:

- a new internal class that allows to communicate with HTML pages retrieving data and printing results
- a framework library that runs backend COBOL code on mobile devices (currently only Android devices are supported)

This is the screenshot taken during the execution of isCOBOL sample App on Android Tablet:



## isCOBOL 2013 R1 Enhancements

### Compatibility with other COBOLs

The isCOBOL Framework is now able to read RM indexed files natively, this feature is particularly useful during the conversion from RM/COBOL to easily convert data files to one of the file systems supported by isCOBOL.

In terms of compatibility with Acucobol-GT, isCOBOL 2013 R1 includes these features:

- Compiler option -dcn to use NCR sign encoding
- Compiler option -fm for an implicit LOCK MODE IS MANUAL
- New library routines:
  - C\$GETLASTFILEOP to retrieve information about the last file operation performed
  - R\$IO to access relative files in a dynamic way
- New WIN\$PRINTER op-code WINPRINT-UPDATE-PRINTERS to update the printers list within the runtime session
- The ability to specify in Declaratives the procedure USE AT PROGRAM START executed the first time the program is loaded and USE AT PROGRAM END when the program is canceled. Code example:

```
DECLARATIVES.  
my-start section.  
    use at program start.  
my-start-para.  
    display message "program loaded".  
my-end section.  
    use at program end.  
my-end-para.  
    display message "program canceled".  
END DECLARATIVES.
```

In terms of compatibility with ICOBOL, isCOBOL 2013 R1 provides:

- A new compiler option -ci to have implicit LOCK MODE IS MANUAL WITH MULTIPLE RECORDS of files and COL + 1 is assumed on COL without value

In terms of compatibility with RM/COBOL, isCOBOL 2013 R1 provides:

- New library routines:

- C\$GETLASTFILENAME and C\$GETLASTFILEOP to retrieve information about the last file name and file operation performed
- C\$GUICFG, C\$SETDEVELOPMENTMODE, P\$CLEARDIALOG P\$CLEARFONT P\$DISABLEDIALOG P\$DISPLAYDIALOG, P\$DRAWBITMAP, P\$DRAWBOX, P\$DRAWLINE, P\$DRAWROUNDBOX, P\$ENABLEDIALOG, P\$GETDEVICECAPABILITIES, P\$GETDIALOG, P\$GETFONT, P\$GETTEXTMETRICS, P\$NEWPAGE, P\$SETDEFAULTMODE, P\$SETDEFAULTUNITS, P\$SETDIALOG, P\$SETDOCUMENTNAME, P\$SETFONT, P\$SETPEN, P\$SETPOSITION, P\$SETTEXTCOLOR, P\$SETTEXTPOSITION, P\$SETTOPMARGIN and P\$TEXTOUT to manage printer jobs
- The support for BINARY variable declaration specifying the size in bytes. Code example:

```
77 my-var Pic S9(5) Binary(2).
```
- Exponentiation syntax in data division. Code example:

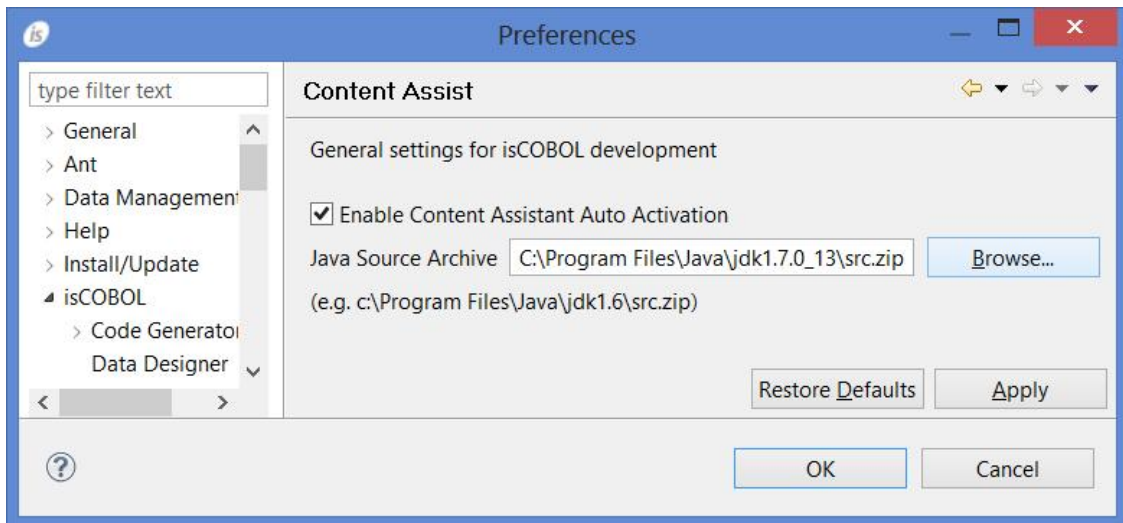
```
78 my-const value 2 ** 32.
```
- The support for LIKE condition. In addition isCOBOL can use the power of java regular expressions. See details in the "Improvements in the Compiler" session.
- The ability to point to first or last record with the extended syntax in the START statement:

```
START filename KEY [IS] FIRST / LAST keyname
```
- The support for WHILE condition in the START. In addition isCOBOL supports this syntax on any type of data file (sequential, relative, indexed). See details in the "Improvements in the Compiler" session.

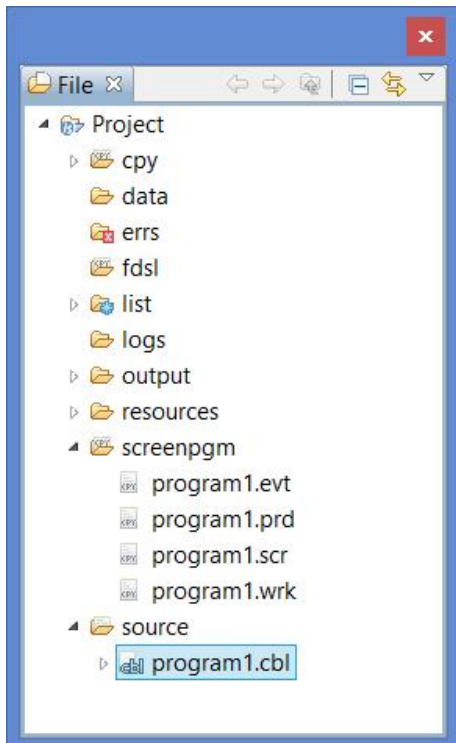
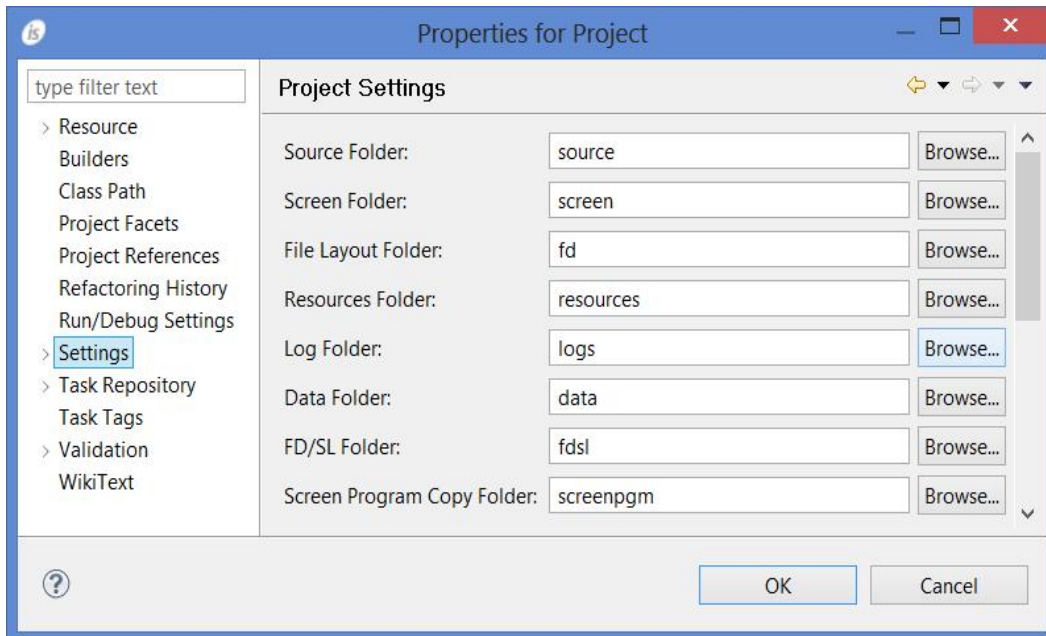
### isCOBOL IDE Enhancements

Numerous enhancements were made to the isCOBOL IDE in the 2013 R1 release, including:

- Added the option “Enable Content Assistant Auto Activation” in the isCOBOL Editor Content Assist (by default it is enabled). When this option is disabled, the Content Assistant will be available only typing Ctrl+Space.

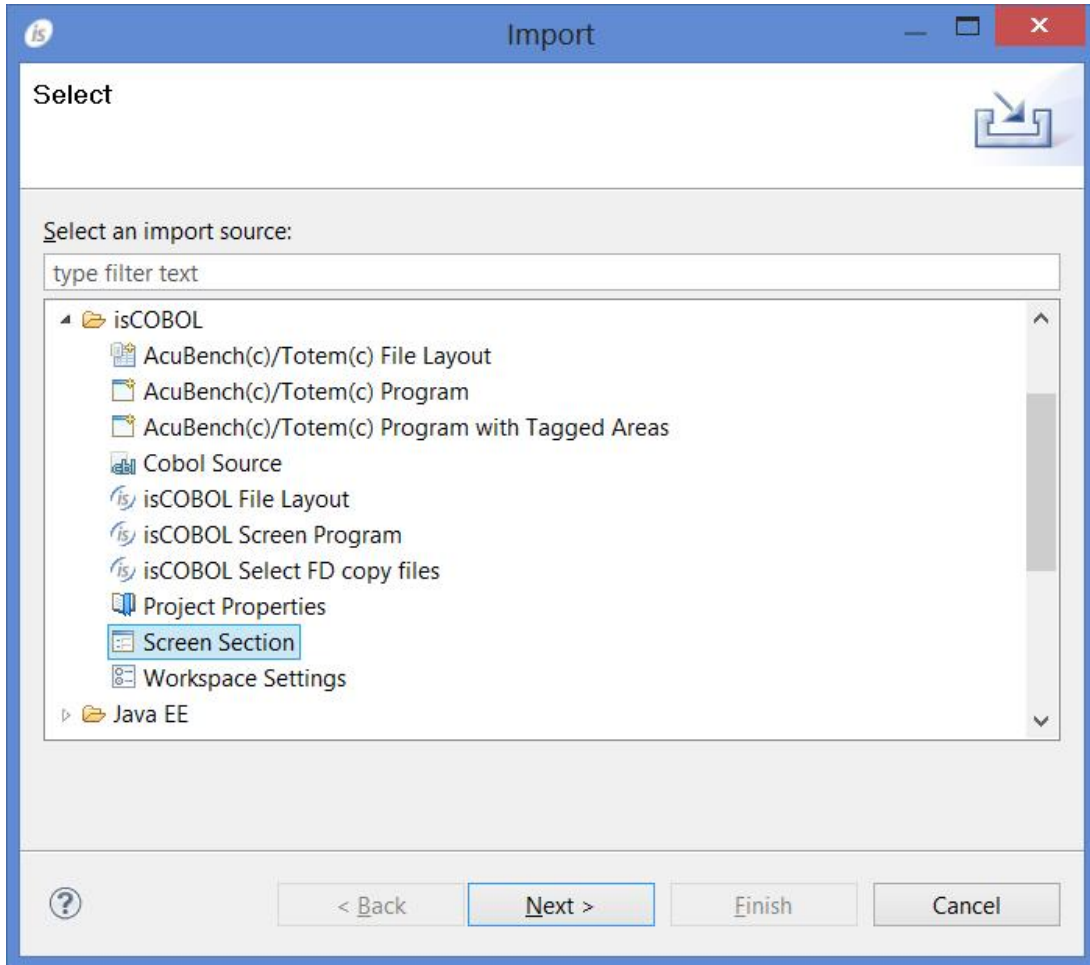


- Created specific folders for files generated by the Screen Designer to have a clearer structure in the File View:





- Added the ability to export/import one specific screen from a program to a different one



### Improvements in the Compiler

Improvements to the isCOBOL Compiler in 2013 R1 include:

- new statement ASSERT (cond) OTHERWISE "Exception message to raise" to raise exception when executing with java option -ea. This is specially useful for Debugging purpose as VisualC / Java languages. Code example:

```
assert (var1 = 1 or var2 = 2)
    otherwise "Exception message to raise, " VAR1, VAR2.
```

- The support for LIKE condition in IF statement to verify if a variable match with a regular expressions. Code example:

```
move "There are three cats" to var1
move "The.*ats" to var-reg-exp
if var1 is like trimmed right var-reg-exp
    display "match".
```

- The ability to filter records through a regular-expression set on the START statement. Code example:

```
move "^0[24][a-z][a-z][a-z]" to var-reg-exp
start file1 key first key1 while like var-reg-exp
perform until exit
    read file1 next at end exit perform end-read
    add 1 to cnt
    move arc-file1 to filtered-rec(cnt)
end-perform.
```

## Debugger Improvements

Improvements to the isCOBOL Debugger in 2013 R1 include:

- New "Variable area" to show the variables created with display -tree syntax and variables better shown in 2 new columns.

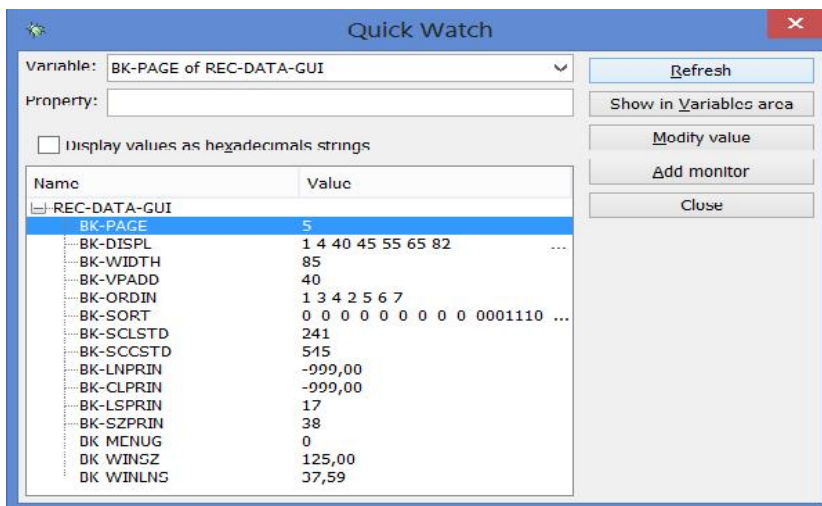
This is the main window with new Variable area open to show two group variable in tree structure:

The screenshot shows the isCOBOL Graphic Debugger interface. The main window displays the source code of a COBOL program. The current line of execution is line 60, which is highlighted in yellow. The code includes several COBOL statements such as `perform LOAD-FONT`, `perform LOAD-BMP`, `open input data-gui`, `if file-status not = "35"`, `perform READ-DATA-GUI`, `if current-page = 7`, `move 1 to jbflag`, `end-if`, `close data-gui`, `end-if`, `accept today from century-date`, `accept wtime from time`, `accept system-information from system-info`, `accept terminal-abilities from terminal-info`, `if win-col = -99999 and win-line = -99999`, `compute win-col = (physical-screen-width - 829) / 2`, `compute win-line = (physical-screen-height - 597) / 2`, and `end-if`.

The 'Variable area' on the right shows a tree structure of variables. The root node is 'RECORD-H-GRID', which contains four records: 'REC (1)', 'REC (2)', 'REC (3)', and 'REC (4)'. Each record contains several fields: 'CODE', 'ART', 'QTY', 'PRICE', 'DISP', 'EDATE', and 'AVAILABLE'. The values for these fields are displayed in the 'Value' column.

The 'Perform stack' window at the bottom shows the current stack of operations. The stack contains two entries: 'CURRENT-PAGE [ISCONTROLSET]' with a value of 1, and 'FILE-STATUS [ISCONTROLSET]'.

This is the updated Quick Watch window with button to shown in Variables area:



- The ability to monitor an occurs item whose index is a variable. Command example: MON var1(counter)
- The ability to skip all statements of the paragraph or program with the option - paragraph or -program on JUMP statement. There are also new menu items and new buttons in the toolbar to make the same functions

### isCOBOL DataBase Bridge Improvements

Improvements to the isCOBOL Database Bridge in 2013 R1 include:

- New option `-cc` to generate code that manages the `COMMIT_COUNT` feature. The feature allows to have a Commit statement automatically performed after a given number of successful Write, Rewrite and Delete statements. The number can be configured in the property `iscobol.easydb.commit_count.<connection-name>` or `iscobol.easydb.commit_count`.
- New option `-Owfl` to generate code that allows dynamic setting of wait for locks. It return record lock condition under Oracle depending on `iscobol.easydb.wait_for_locks` setting that can be set dinamically from program through `SET ENVIRONMENT` statement.

### New graphical user interface features

Enhancements to user interface features include:

- The ability to scale automatically an image with new property BITMAP-SCALE in BITMAP control. Code Example:

```
03 b1 bitmap
    bitmap-handle h-bmpapp
    bitmap-scale 1
    line 1 col 1 size 100 pixels lines 100 pixels.
```

- The ability to use HIDDEN-DATA property also for COMBO-BOX and LIST-BOX controls. Code Example:

```
modify list-box1 item-to-add "My-List-Box-Item"
    hidden-data "123"
modify list-box1 query-index 1
inquire list-box1 hidden-data var-hidd-list
modify combo-box1 item-to-add "My-Combo-Box-Item"
    hidden-data "456"
modify combo-box1 item 1
inquire combo-box1 hidden-data var-hidd-combo
```

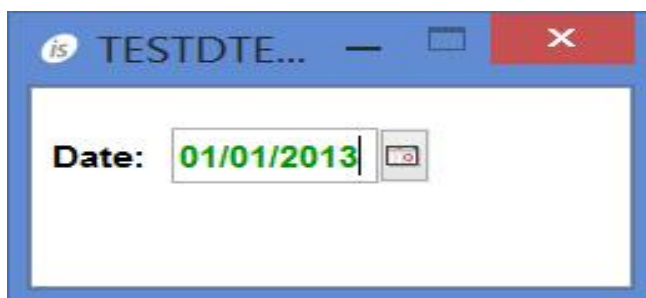
- The ability to intercept new events MSG-ICONIFIED and MSG-DEICONIFIED for window control when user reduces the window to task bar

- The ability to have more row headings with new property NUM-ROW-HEADINGS for GRID CONTROL



- The ability to input only numeric letters in date-entry control setting the style NUMERIC. In addition the new style ALLOW-EMPTY allows to leave empty the control. Code Example:

```
03 del date-entry
    3, col 3, size 13 cells
    value w-data NUMERIC ALLOW-EMPTY
    display-format "dd/MM/yyyy".
```



### Improvements in the Framework

Improvements to the isCOBOL Framework in 2013 R1 include:

- The ability to work with sequential files in memory with standard syntax for sequential files. Code Example:

```
select filemem assign to address memory-file
      organization line sequential.
fd filemem.
01 filemem-rec pic x(80).
working-storage section.
77 memory-file pic x any length.
```

- The ability to create temporary files with new library routine C\$CREATE\_TMP\_FILE. It is useful in particular for web application needing to send printings, images, audio files etc. to the web browser. Code Example:

```
call "c$create_tmp_file" using tmp-path prefix suffix.
```

- The ability to replace parts of text in an alphanumeric variable looking for a regular expression with new library routine C\$REPLACE-ALL. Code Example:

```
call "c$replace-all" using variable, regex, replacement
                        [w-opts, w-error]
```

### Improvements in the isCOBOL Server and File Server

Enhancements to isCOBOL Server and File Server include:

- The ability to configure the algorithm for password encryption with the new configuration property iscobol.as.digest. Configuration example:

```
iscobol.as.digest=SHA-1
```

- The ability to run program with a particular configuration file in Thin Client mode through aliases. Configuration example:

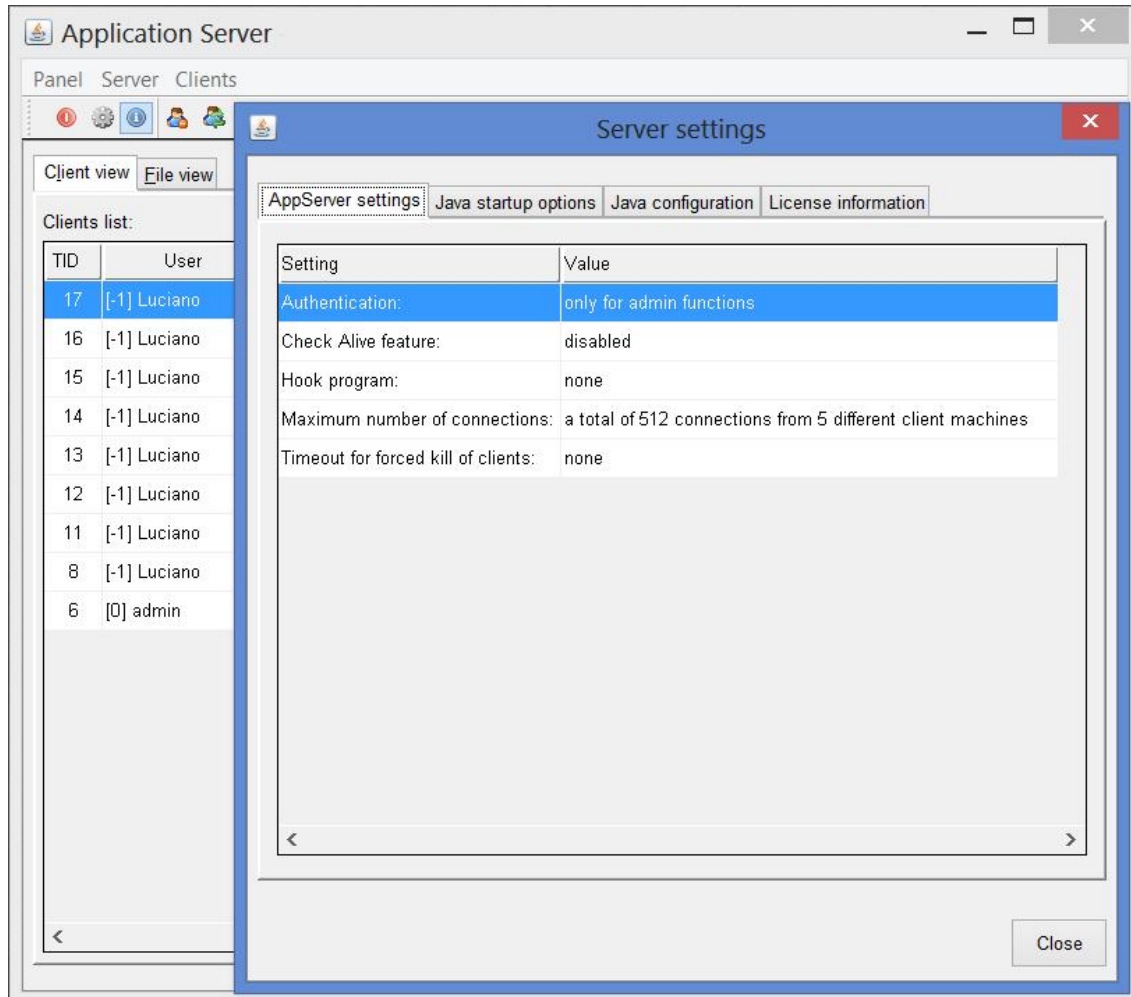
```
iscobol.as.use_aliases=true
iscobol.as.alias.myalias=PROGRAM1,myconfig
```

Command line:

```
iscclient -hostname ipserver -port 10999 myalias
```



- The Server Panel enhanced to monitor the state of remote calls and it is now able to show the “settings” (Java options, configuration, license info)



### Enhancements to isCOBOL utilities

IsCOBOL 2013 R1 contains several new features in utilities including:

- New options in JUTIL (-getimg and -makeimg) utility to create the definition string from an existing file and to create a new file from the definition string. In addition also the -rebuild function has been enhanced with a new parameter F to force rebuild in Jlsam .dat file if necessary. Command line examples:

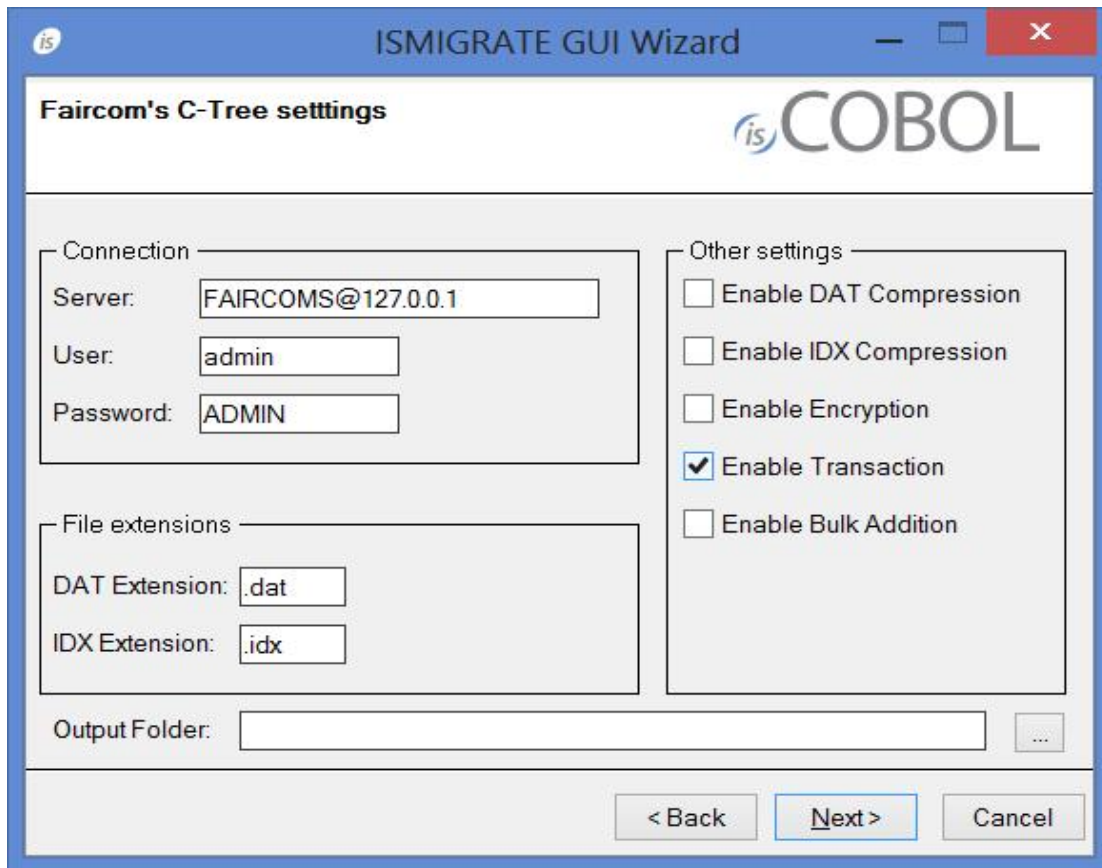
```
jutil -getimg file1
00044,00044,001,01,0,004,00000
```

```
jutil -getimg file2 00044,00044,001,01,0,004,00000
```

```
file created
```

```
jutil -rebuild file1 F
```

- Additional settings in ISMIGRATE utility to easily configure C-Tree migrations allowing customizations for extensions, compression, encryption and the ability to use a remote server connection.



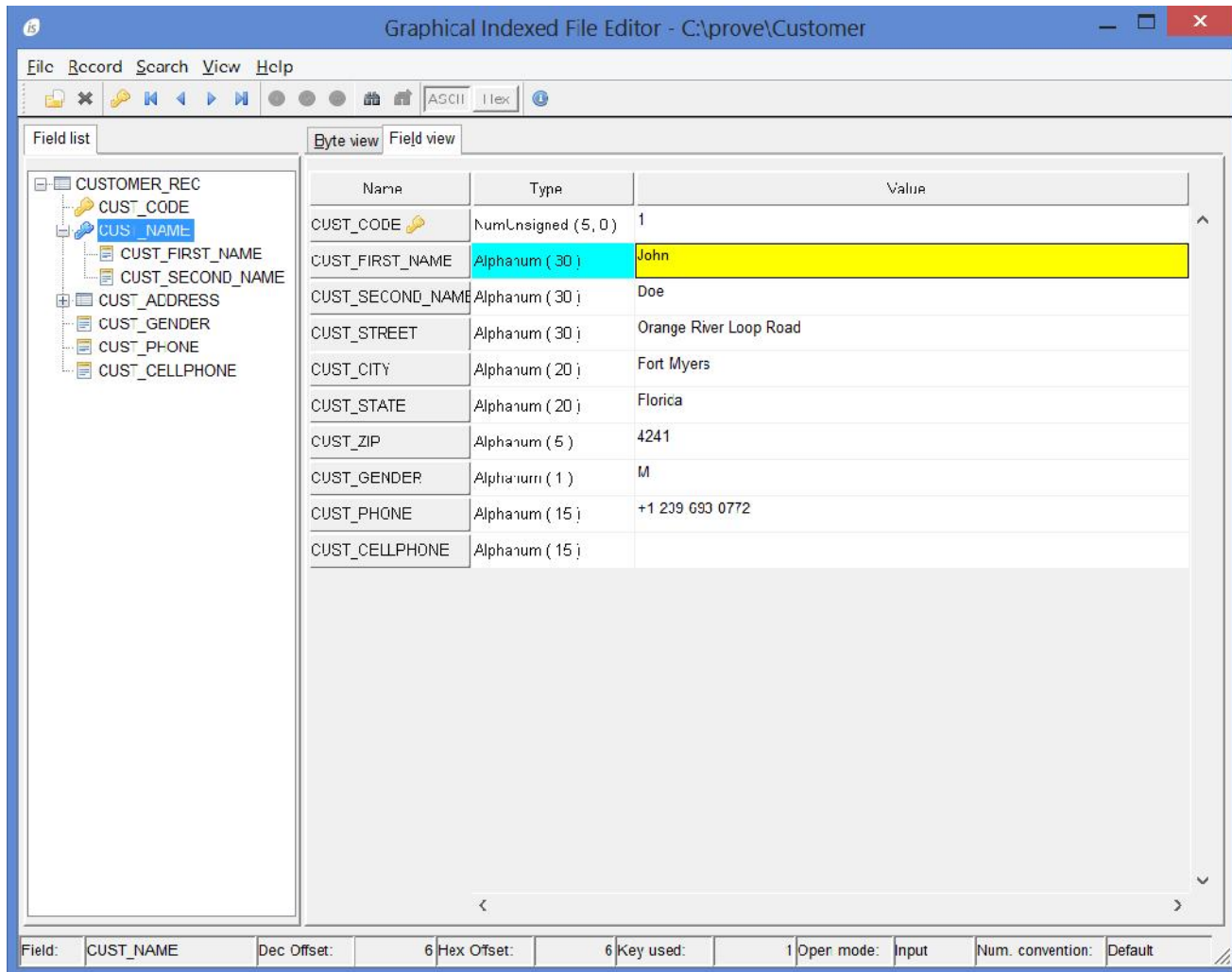
- New "Field view" in GIFE utility to easily see the content of each field with the supports of EFD file to list the fields described in the file.

This is the updated "Byte View":

The screenshot shows the Graphical Indexed File Editor (GIFE) window titled "Graphical Indexed File Editor - C:\prove\Customer". The window has a menu bar (File, Record, Search, View, Help) and a toolbar with icons for file operations and viewing modes (ASCII, Hex). On the left, a "Field list" pane shows a tree structure for "CUSTOMER\_REC" with fields: CUST\_CODE, CUS\_NAME (expanded to show CUST\_FIRST\_NAME and CUST\_SECOND\_NAME), CUST\_ADDRESS (expanded to show CUST\_GENDER, CUST\_PHONE, and CUST\_CELLPHONE). The main area is split into "Byte view" and "Field view" tabs. The "Byte view" is active, showing a table with columns for "Offset", "ASCII view", and "Hex view". The data represents a customer record for "John Doe" at "Orange River Loop Road Fort Myers Florida" with phone number "239 693 0772". The status bar at the bottom shows: Field: CUST\_FIRST\_NAME, Dec Offset: 6, Hex Offset: 6, Key used: 6, Oper mode: Input, Num. convention: Default.

Offset	ASCII view	Hex view
C0001	0 0 0 0 1 J o h n	30 30 30 30 31 4A 6F 68 6E 20 20 20 20 20 20 20 20 20 20 20 20
C0021	D o e	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 44 6F 65 20 20
C0041		20 20
C0061	o r a n g e R i v e r L o	20 20 20 20 20 20 4F 72 61 6E 67 65 20 52 39 76 65 72 20 4C 6F
C0081	o p R o a d F o r t	6F 70 20 52 6F E1 64 20 20 20 20 20 20 20 20 20 20 46 6F 72 74 20
C0101	M y e r s F l o r i	4D 79 65 72 73 20 20 20 20 20 20 20 20 20 20 20 46 6C 6F 72 69
C0121	d a 4 2 4 1	64 61 20 20 20 20 20 20 20 20 20 20 20 20 20 20 34 32 34 31 20
C0141	M + 1 2 3 9 6 9 3 0 7 7 2	4D 23 31 20 32 E3 39 20 36 39 33 20 30 37 37 32 20 20 20 20
C0161		20 20 20 20 20 20 20 20 20 20

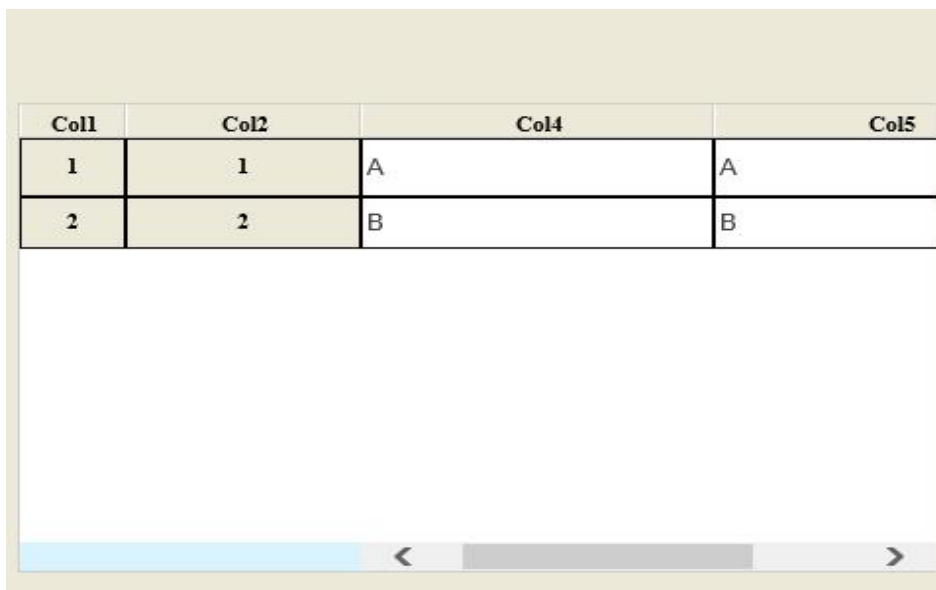
This is the new “Field View”:



### Improvements to Web Direct 2.0

Web Direct 2.0 has been improved and now it offers:

- The ZK framework updated to latest version including more java beans
- The ability to uses row heading through properties ROW-HEADINGS and NUM-ROW-HEADINGS



The image shows a screenshot of a web browser displaying a table. The table has four columns: 'Col1', 'Col2', 'Col4', and 'Col5'. The first two columns are used for row headings. The first row has '1' in Col1 and '1' in Col2, with 'A' in Col4 and 'A' in Col5. The second row has '2' in Col1 and '2' in Col2, with 'B' in Col4 and 'B' in Col5. Below the table is a scrollbar with a light blue track and a grey slider, indicating that the table content can be scrolled horizontally.

Col1	Col2	Col4	Col5
1	1	A	A
2	2	B	B