



isCOBOL™ Evolve

isCOBOL Evolve 2012 Release 1 (R1) Overview

Copyright © 2012 Veryant LLC.
4241 Orange River Loop Road, Fort Myers, FL 33905 U.S.A.

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.

Table of Contents

Introduction.....	4
Veryant UDBC Server.....	4
isCOBOL IDE Enhancements	5
Enhanced compatibility with Micro Focus® extend® (formerly ACUCOBOL-GT)	6
Debugger Improvements	7
New graphical and character user interface features.....	10
Enhancements to XML2WRK utility and the XMLStream class	11
The ability to load printer fonts from the server side when using isCOBOL Server and thin client..	13

isCOBOL Evolve 2012 Release 1 (R1) Overview

Introduction

Veryant is pleased to announce the latest release of isCOBOL™ Evolve, isCOBOL Evolve 2012 R1. isCOBOL Evolve provides a complete environment for the development, deployment, maintenance, and modernization of COBOL applications. isCOBOL 2012 R1 includes a new product (Veryant UDBC Server), several enhancements to the isCOBOL Integrated Development Environment (IDE), debugger improvements, and other new features.

Details on these enhancements and updates are included below.

isCOBOL 2012 R1 Enhancements

Veryant Universal Database Connector (UDBC) Server

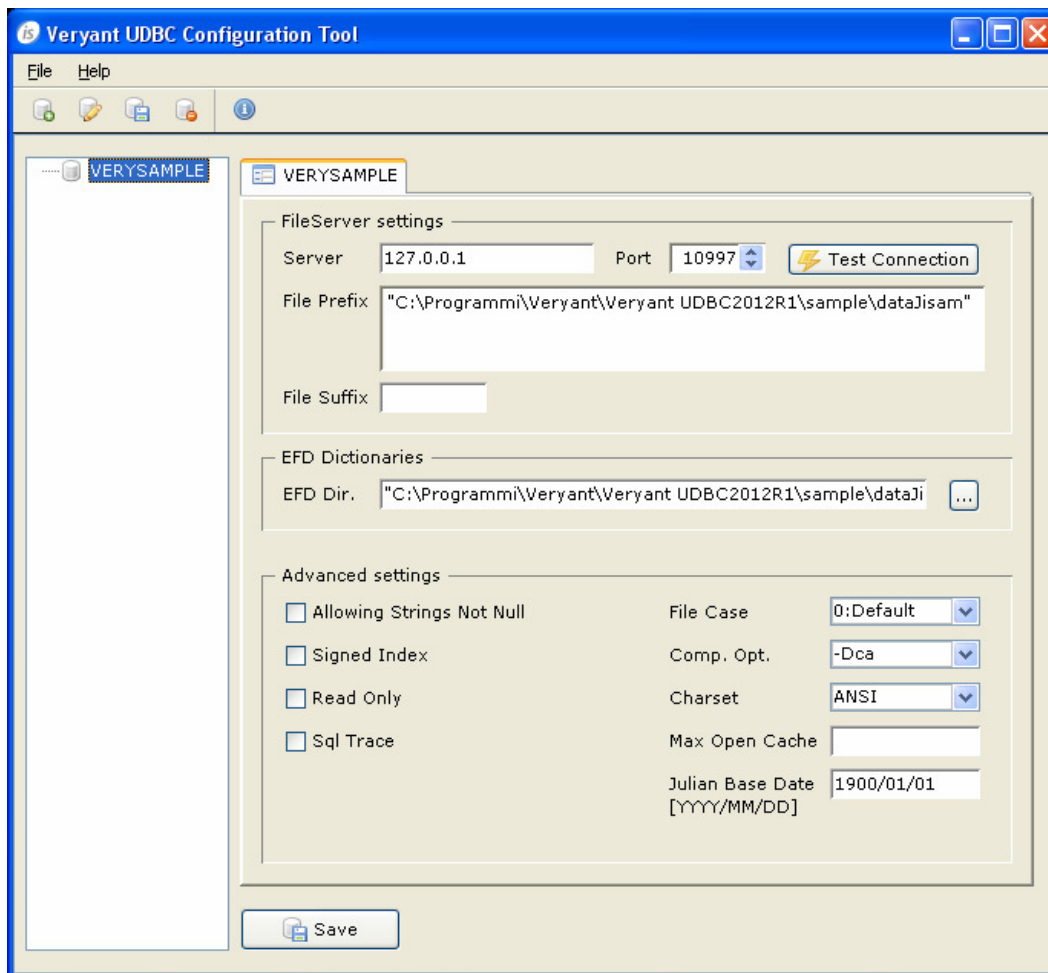
The Veryant UDBC Server product allows developers to access indexed files hosted by the isCOBOL File Server as if they were a relational database. This enables organizations to benefit from modern productivity tools designed for relational databases without having to restructure or migrate existing data.

Veryant UDBC Server includes an ODBC and JDBC driver for all supported isCOBOL file systems, including Veryant JISAM. The product also offers server-side SQL processing and includes a GUI utility to manage configurations and a command-line SQL interpreter utility.

The basic steps required to manage data through Veryant UDBC Server are:

1. Configure the database
2. Start Veryant UDBC Server and the isCOBOL File Server*
3. Connect with a ODBC or JDBC client and run SQL queries

The following screenshot shows the Veryant UDBC Server configuration GUI.

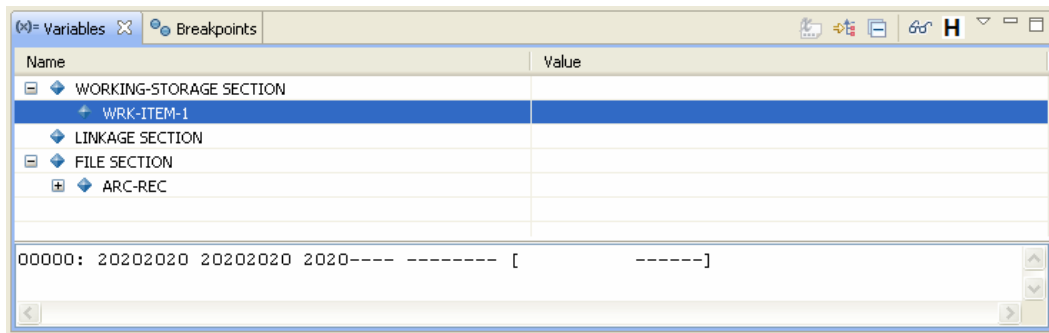


* Note, Veryant UDBC requires isCOBOL File Server to be up and running. The isCOBOL File Server is provided along with isCOBOL.

isCOBOL IDE Enhancements

Numerous enhancements were made to the isCOBOL IDE in the 2012 R1 release, including:

- The ability to show working-storage and file section variables automatically in the Variables view. This is illustrated in the following screen shot:



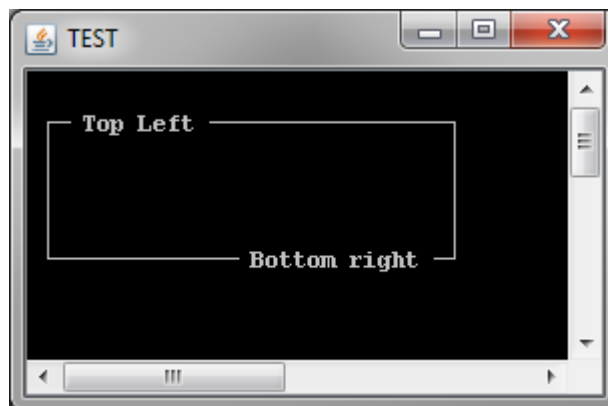
- The ability to add and modify customize keyboard accelerators for actions such as toggling comments, compiling and running. This is done by selecting "Window" in the menu bar, then "Preferences". In the tree on the left expand "General" and select "Keys". There a developer can manage shortcuts.
- The ability to integrate with and generate Veryant Database Bridge programs from the isCOBOL IDE.
- The ability to start with the last property edited in the previous control when editing the properties for a new control.

Enhanced compatibility with Micro Focus® extend® (formerly ACUCOBOL-GT)

Additional compatibility with Micro Focus® extend® (formerly ACUCOBOL-GT) was added in isCOBOL 2012 R1) to further simplify transitions to the isCOBOL platform, including:

- The ability to display two titles in the same DISPLAY BOX. For example:

```
display box line 2 col 2 lines 5 size 30
top left title "Top Left"
bottom right title "Bottom right".
```

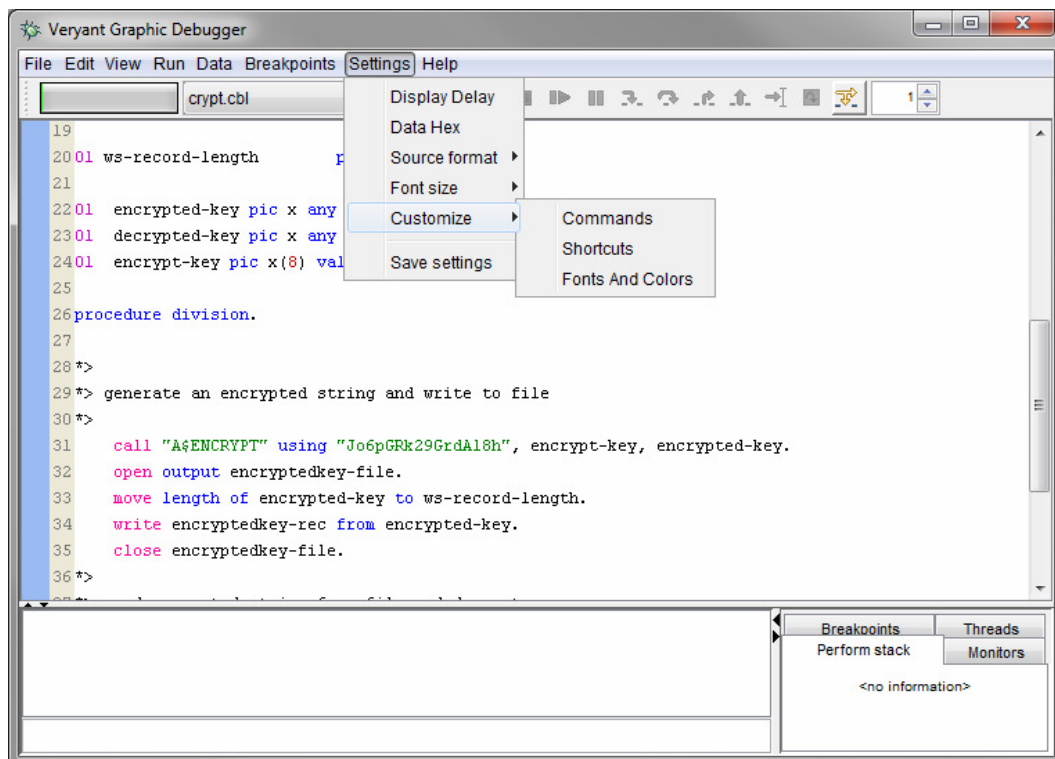


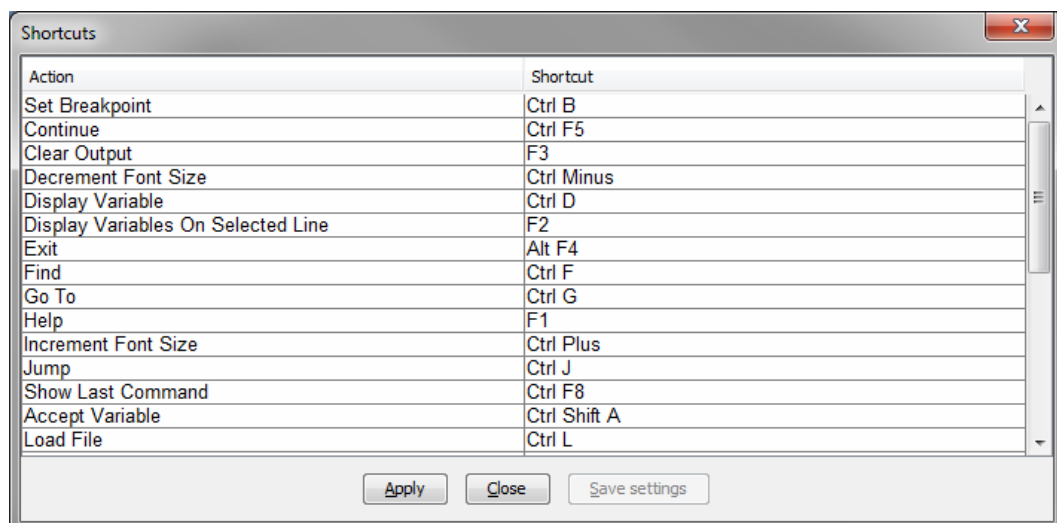
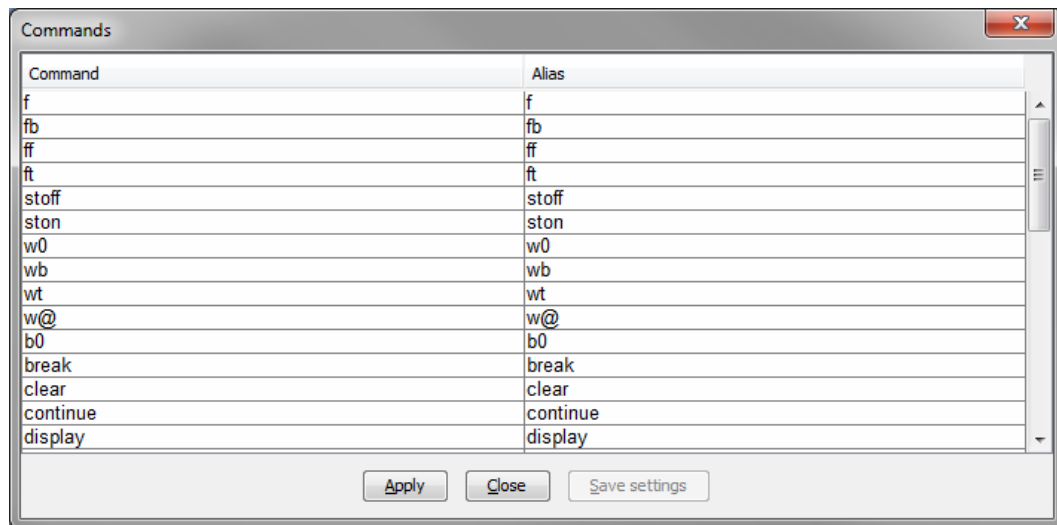
- A new compiler option, -sevc , to expand environment variables in COPY path-names such as in COPY library-name OF "\$LIB/\$COPYLIB/\$SUBDIR"
- A new compiler option, -dcdm, that causes any data item whose underlying type is binary to be stored in the minimum number of bytes needed to hold it, like ACUCOBOL's -dm option

Debugger Improvements

Improvements to the isCOBOL Debugger in 2012 R1 include:

- The ability to customize command names and keyboard accelerators, as seen in the series of images below:





- Support for complex conditions in breakpoints

```
BREAK [-d] line-number [filename] [progrname] [WHEN [-x|-env]
conditional expression]
```

set a breakpoint at a line.

```
BREAK [-d] paragraph-name [filename] [progrname] [WHEN [-x|-env]
conditional expression]
```

set a breakpoint on a paragraph.

```
BREAK -l
list breakpoints.
```

```
[-d: disable the breakpoint]
```


when-phrase specifies a break condition

the conditional expression has the following format:

```
{ value1 > value2 }
    >=
    <
    <=
    =
    !=

{ ! cond1          }

{ cond1  && cond2  }
    ||
```

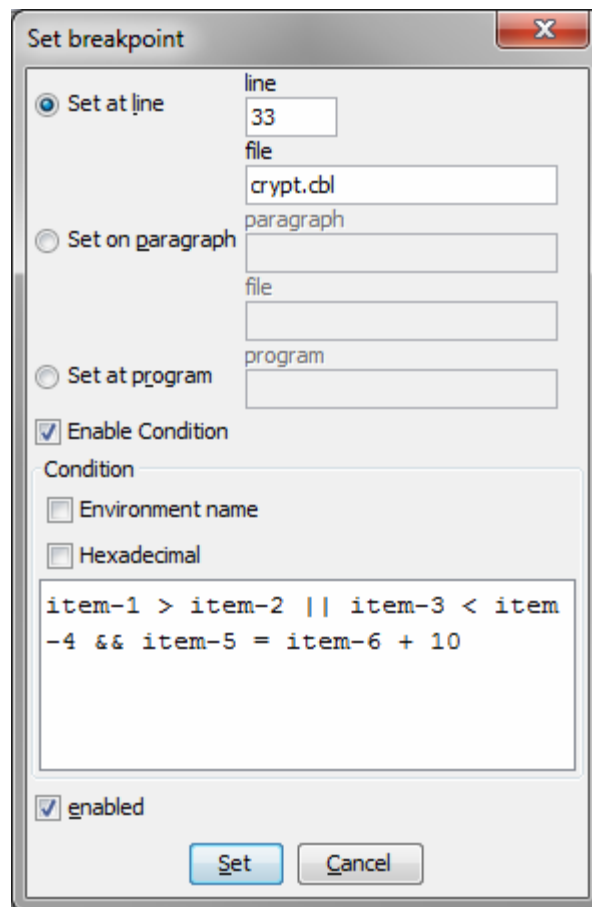
where *value1* and *value2* are variable names, arithmetical expressions or constants, *cond1* and *cond2* are conditional expressions

The string constants must be delimited by "".

If the *-env* option is specified, the variable names must be related to environment variables.

If the *-x* option is specified, the constants must be hexadecimal strings.

The supported arithmetical operators are +, -, *, /



- Version checking between client and remote debugger

New graphical and character user interface features

Enhancements to graphical and character user interface features include:

- The ability to copy/paste with a mouse, similar to putty

To copy text from a character-based screen into the clipboard users can click the left mouse button and drag the mouse to select text. The act of selection copies the text into the clipboard. There is no need to press a control key.

To paste text from the clipboard into an active field accepting input users can press the middle mouse button (usually the scroll wheel). The pasted text is put into the keyboard buffer as if the user had typed it into the field.

- The ability to configure whether docking windows are boxed

- New List-Box Properties -- Row-Color-Pattern, Row-Foreground-Color-Pattern, Row-Background-Color-Pattern
- New properties, `iscobol.gui.disabled_field_color` and `iscobol.gui.entryfield.read_only_color`, to set the color of disabled and read-only fields

Enhancements to XML2WRK utility and the XMLStream class

isCOBOL 2012 R1 contains several new features in the XMLStream class.

- *iscobol.xmlstream.indent_number*

The XMLStream write methods will indent the output according to the value of this property. The default is -1 which disables indentation.

For example, to indent each level 2 columns set this property as follows:

```
iscobol.xmlstream.indent_number=2
```

- *iscobol.xmlstream.omit_empty_elements*

To omit empty elements from the XML output, set this property to true. The default value is false.

```
iscobol.xmlstream.omit_empty_elements=true
```

- *setFileName, getPrintWriter and new write methods*

It is now possible to set the filename, get the PrintWriter and write the XML output in 3 separate steps. This enables you to print lines such as processing instructions to the XML output before writing the XML record. For example, to add a line specifying an XML stylesheet you could use the following code:

```
objXmlStream::"setFileName"("xmltest.xml").  
objXmlStream::"getPrintWriter"::"println"  
    ('<?xml-stylesheet type="text/xsl" href="ccr.xsl"?>'  
    as string).  
objXmlStream::"write"().
```

Also, you can now read and write from/to streams and strings using the new `readFromStream`, `readFromString`, `writeToStream` and `writeToStringBuffer` methods. For example, to write XML to and read XML from a PIC X item you could use the following code:

```
77 XML-STRING PIC X ANY LENGTH.  
...  
set objStringBuffer to stringBuffer::"new" ().
```

```
objXmlStream::"writeToStringBuffer" (objStringBuffer).  
set XML-STRING to objStringBuffer::"toString" ().  
...  
objXmlStream::"readFromString" (XML-STRING as string).
```

The XML2WRK utility has 2 additional command line parameters; *prefix* and *disambiguate_flag*. *prefix* allows you to specify a string to prepend to each element name. Pass 1 for *disambiguate_flag* to generate long names which are guaranteed to be unique.

usage: java XML2WRK xmlfile[wrkfile] [prefix] [disambiguate_flag]

For example, to prepend "o-" to each element name and to disambiguate the names you could run the following command:

```
java XML2WRK .\SampleCCRDDocument.xml ccr.wrk o- 1
```

Here is an example of an indented output with empty tags omitted and a stylesheet processing instruction at the top:

```
<?xml-stylesheet type="text/xsl" href="ccr.xsl"?>  
<?xml version="1.0" encoding="UTF-8"?>  
<ContinuityOfCareRecord>  
  <Body>  
    <SocialHistory>  
      <SocialHistoryElement>  
        <Episodes>  
          <Episode>  
            <DateTime>  
              <DateTimeRange>  
                <BeginRange>  
                  <ExactDateTime>20101205</ExactDateTime>  
                </BeginRange>  
              </DateTimeRange>  
            </DateTime>  
          </Episode>  
        </Episodes>  
      </SocialHistoryElement>  
      <SocialHistoryElement>  
        <Episodes>
```

```
<Episode>
  <DateTime>
    <DateTimeRange>
      <BeginRange>
        <ExactDateTime>20101201</ExactDateTime>
      </BeginRange>
    </DateTimeRange>
  </DateTime>
</Episode>
</Episodes>
</SocialHistoryElement>
</SocialHistory>
</Body>
</ContinuityOfCareRecord>
```

The ability to load server-side printer fonts when using isCOBOL Server

isCOBOL 2012 R1 offers the ability to load server-side printer fonts when using isCOBOL Server and thin client.