2023

ISSUE

09

BIANNUAL JOURNAL OF VERYANT AND isCOBOL

veryant NEWS

THIS ISSUE

PLEASE JOIN US ON LinkedIn or Facebook to stay up-to-date with us.

Watch our demonstration videos and Subscribe to our YouTube Channel

TELL US:

We want to spread the word about how wonderful Veryant is. And we think you can help us.
If you know of another company "ISV or End User" running applications in COBOL, send us their contact information, and we'll send both you and the customer your refer a $50.00 pre-paid Visa Credit Card.
Just like that!

2023 R2 is here

isCOBOL 2023 Release 2 was released September 6, 2023.

sCOBOL 2023 Release 2 was released September 6, 2023. It includes:

- **Support for drag-and-drop for GUI controls to let developers imagine new ways to interact with their application.**
- **A new IDE feature that lets you run or debug programs using specific JRE versions for each project.**
- **A new TurboRun feature that improves performance of COBOL batch programs that are executed sequentially by running them in the Application Server Java Virtual Machine, saving the cost of initializing a JVM for each application.**

2023 R2 is here

The newest isCOBOL Evolve includes exciting new features for GUI controls, the IDE lets you select a specific Java version for each project, and TurboRun speeds up COBOL batch execution.

GUI Control Enhancements
isCOBOL Evolve 2023R2 introduces support for drag-and-drop to and from GUI controls to let developers imagine new ways to interact with their application.

The grid now supports load-on-demand for its content, delivering faster load times and a smoother user experience.

Filters have been improved and you now have the option of using entry fields or check boxes for filtering data more effectively.

Java Version Selection
The isCOBOL IDE 2023 R2 includes the ability to specify the Java version at the project level.

Applications may need to be run using different Java releases for multiple reasons; for example when using Object Oriented Programming with dependencies on classes compiled with specific JDK releases. You can now choose the JRE version for each project from the list of installed JREs.

TurboRun Feature
The new TurboRun feature improves the performance of COBOL batch programs that are executed sequentially by running them in the Application Server's Java Virtual Machine. This saves the cost of initializing a JVM for each application. Using the Application Server's JVM boosts performance even further, since it uses the cache and the JIT (Just-In-Time) optimizations that are already running.

Get more information about these and other changes in the Release Overview, or watch a demonstration in our newest video.

DBMAKER 5.4.5 HAS BEEN RELEASED

DBMaker 5.4.5 has been released and it includes a type 4 JDBC driver.
No more native dependences for your Java-JDBC application.

The new driver can be used on every platform where Java can run, even if a DBMaker client porting is not available for that platform.

DBMaker 5.4.5 is available for download here.

Automatic validation of entry-field data using control properties

You would validate an entry in an entry-field by adding some code in the entry-field's AFTER procedure. It's often necessary to access data to check a record. But there are other cases where you need a simpler validation, without the need for data access. In these cases it's easier to use entry-field properties. Here are two examples:

- A code that has a format like 1 alphabetic letter and 1 numeric value can be automatically checked using these properties:

validation-regexp	"[A-z]+[0-9]"
validation-errmsg	"Invalid code"

  In this case the error message shown when the validation fails is set in the property "validation-errmsg". Though this is a simple example, with more complex regular expression syntaxes you can check almost any situation.

- A numeric value that accepts values between 9 and 100 can be automatically checked using these properties:

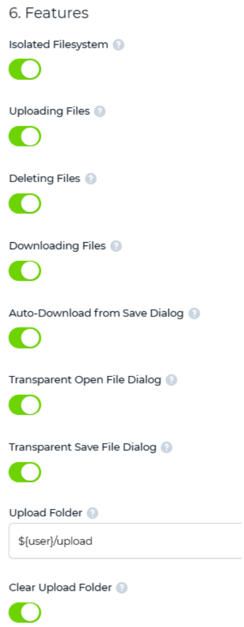
  min-val 9 max-val 100

  In this case the error message shown is "Please enter a value between 9 and 100" in the localized language.

For more information and more examples, check out the KB article here.

Uploading and Downloading files in Webclient environment

To transfer files between the end user's client PC and the server in a WebClient environment, you can use C$OPENSAVEBOX to automatically upload from the PC where the browser runs to the Webclient server, and C$DESKTOP with the CDESKTOP-OPEN op-code to automatically download a file to that PC.



6. Features

Isolated Filesystem ☑

Uploading Files ☑

Deleting Files ☑

Downloading Files ☑

Auto-Download from Save Dialog ☑

Transparent Open File Dialog ☑

Transparent Save File Dialog ☑

Upload Folder: ${user}/upload

Clear Upload Folder ☑

You will also need to enable the "Uploading files" and "Downloading files" properties in the WebServer application setup. Other WebClient setup options to consider are to:

- enhance security with "Isolated Filesystem" and "Upload folder," setting a controlled directory on the Webclient server for uploads.
- streamline the upload/download process by enabling "Auto-Download from Save Dialog" and "Transparent Open File Dialog."
- use the "Clear Upload Folder" to keep the designated folder free of transferred files on your application's termination.

Have You Seen This?

New YouTube videos:

2023R1 New Features

isCOBOL Integrated Debugger

COBOL-IT isCOBOL Conversion

isCOBOL Remote Debugging

Starting the Application Server as a service on Linux

New Knowledgebase articles:

How to view two different consoles in the IDE?

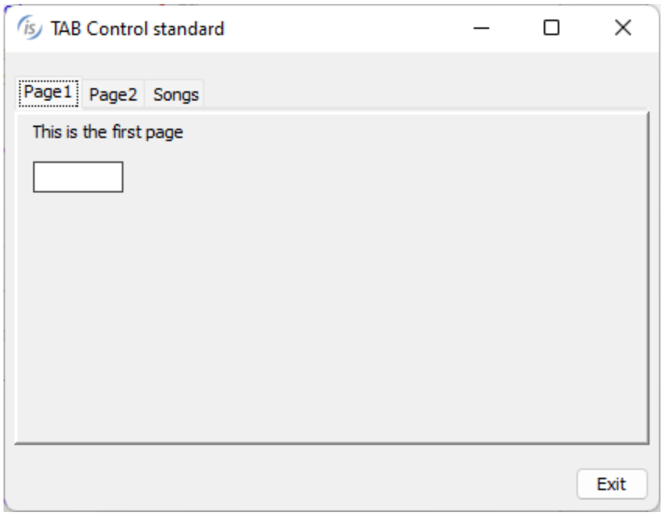
How to set switches with isCOBOL?

How to find the smaller setup to install just the c-treeRTG's ODBC client?

Managing the order of the alternate keys in an indexed file

Modernize Your Tab Control

Does your tab control look dated? Tabs are one of the controls that frequently change their look and behavior. The standard tab-control screenshot below shows the old look and feel. For instance, notice the 3-d look of the tabs that separates them from the page. Also, the tabs have no color, and the dotted line around the tab page number looks very dated.



This is the first page

Another problem with this old style of tab-control is the multiple steps required to change from one tab to another. Each change requires you to:

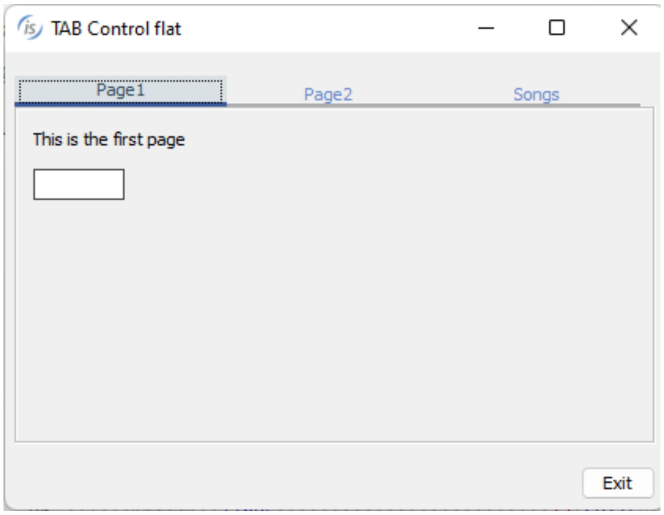
- Intercept the "CMD-TABCHANGED" event.
- Make the screen section of the old page invisible
- Make the screen section of the new page visible

To simplify these coding steps, you can add the "allow-container" style. The look and feel is very similar, but managing the tab changes is much easier. The three steps required above are done automatically. You'll see a performance improvement too, especially on older machines.

To modernize the look and feel, you can add the "tab-flat" style. This more up-to-date look also opens up other properties that can improve the look of your tab-control by adding color to your tab background and foreground and tab borders, as well as determining different widths of your tabs.

This screen is defined with these properties:

```
03 Tb1-container
   tab-control
   line                          2
   col                           2
   lines                         17 cells
   size                          68 cells

   allow-container
   tab-flat

   active-tab-border-width       (0 0 3 0)
   active-tab-border-color       rgb x#395A9D
   active-tab-background-color   rgb x#dae1e5
   active-tab-foreground-color   rgb x#354c5c

   tab-border-width              (0 0 2 0)
   tab-foreground-color          rgb x#6C8BCA
   tab-border-color              rgb x#ACACAC

   tab-widths                    (22 22 22)
```



This is the first page

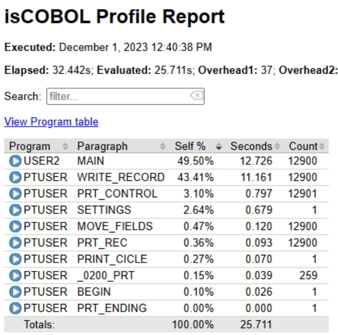
All of these properties can be added to your existing programs using the code injection feature by adding the following entry to the configuration file used during your compile.

```
iscobol.compiler.gui.tab_control.defaults= \
    tab-flat \
    active-tab-border-width        (0 0 3 0) \
    active-tab-border-color        rgb x#395A9D \
    active-tab-background-color    rgb x#dae1e5 \
    active-tab-foreground-color    rgb x#354c5c \
    tab-border-width               (0 0 2 0) \
    tab-foreground-color           rgb x#6C8BCA \
    tab-border-color               rgb x#ACACAC \
    tab-widths                     (22 22 22)
```

A small isCOBOL tool gives you a huge helping hand.

Why should you use the isCOBOL Profiler? Using the isCOBOL Profiler to do a performance analysis of your COBOL programs can tell you which paragraphs in your COBOL program are using the most CPU time. We included the profiler in the runtime framework in version 2015 R1, but in 2020 R2 we made it easy to use, just add "-profile" to your command line. The reports are put in a folder called "hprofHtmlReport".

We recently used the profiler to help a customer cut a batch printing program time in half.   The print file needed data from two c-treeRTG files; the first accessed internally and the second in a called program with linkage passing the required record information.
We created files with test data and ran the programs (PTUSER calls USER2) simply with iscrun, adding "-profile" to the command line. Here are the results:



isCOBOL Profile Report

Executed: December 1, 2023 12:40:38 PM

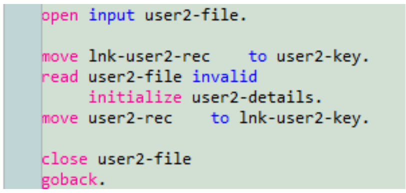
Elapsed: 32.442s; **Evaluated:** 25.711s; **Overhead1:** 37; **Overhead2:**

Search: filter...

View Program table

Program	Paragraph	Self %	Seconds	Count
USER2	MAIN	49.50%	12.726	12900
PTUSER	WRITE_RECORD	43.41%	11.161	12900
PTUSER	PRT_CONTROL	3.10%	0.797	12901
PTUSER	SETTINGS	2.64%	0.679	1
PTUSER	MOVE_FIELDS	0.47%	0.120	12900
PTUSER	PRT_REC	0.36%	0.093	12900
PTUSER	PRINT_CICLE	0.27%	0.070	1
PTUSER	_0200_PRT	0.15%	0.039	259
PTUSER	BEGIN	0.10%	0.026	1
PTUSER	PRT_ENDING	0.00%	0.000	1
Totals:		100.00%	25.711	

Notice that the report shows both the % of total time, and the specific seconds for each paragraph. The bulk of the time was spent in the second program, USER2. That surprised us, because the USER2 program simply reads a smaller file record and returns it in linkage to the main program. Here's the code in USER2:



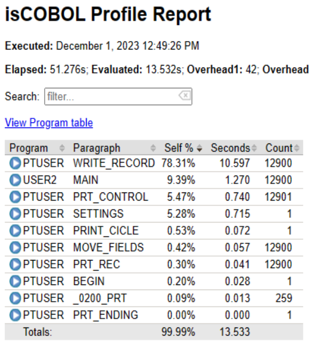
```
open input user2-file.

move lnk-user2-rec    to user2-key.
read user2-file invalid
     initialize user2-details.
move user2-rec    to lnk-user2-key.

close user2-file
goback.
```

We realized all the time was being spent opening and closing the file repeatedly. Fortunately, this customer was using c-treeRTG, which has an option called <filepool> that would help. With file pooling, even though the COBOL program closes the file c-tree keeps it open (pooled) so the next open can get a handle to the file immediately.

We added these options to the customer's configuration:

```
iscobol.file.index.filepool=true
iscobol.file.index.inpool=true
```

and ran the profiler again. Here are the new results:



isCOBOL Profile Report

Executed: December 1, 2023 12:49:26 PM

Elapsed: 51.276s; **Evaluated:** 13.532s; **Overhead1:** 42; **Overhead2:** 55

Search: filter...

View Program table

Program	Paragraph	Self %	Seconds	Count
PTUSER	WRITE_RECORD	78.31%	10.597	12900
USER2	MAIN	9.39%	1.270	12900
PTUSER	PRT_CONTROL	5.47%	0.740	12901
PTUSER	SETTINGS	5.28%	0.715	1
PTUSER	PRINT_CICLE	0.53%	0.072	1
PTUSER	MOVE_FIELDS	0.42%	0.057	12900
PTUSER	PRT_REC	0.30%	0.041	12900
PTUSER	BEGIN	0.20%	0.028	1
PTUSER	_0200_PRT	0.09%	0.013	259
PTUSER	PRT_ENDING	0.00%	0.000	1
Totals:		99.99%	13.533	

Notice the time for the test program decreased from 25.7 seconds to 13.5 seconds, and the bulk of the time happens where it should, in the paragraph that assembles and writes the print lines.

You can run the profiler this way, with iscrun, or while in the Application Server, or even other scenarios (Java calling isCOBOL, EIS in Apache Tomcat, etc.)
This test was all batch processing – meaning no user interface was needed. When you profile a program with user input, it will include the time it took the user to click on things, which won't help you assess the performance of the program.

The solution for this is to use C$PROFILER to disable and enable the profiler around the ACCEPT statements. C$PROFILER is also helpful to limit the profiling when you suspect the problem to be in a specific set of code or set of programs.
The profiler is accessible from the IDE too, with a specific Profiler view to apply the profiler to one, some, or all of the classes in a project.

Your performance problem probably won't be solved the same way as this customer's was, but there are a lot of performance tuning tips in our documentation, and we're always available at support@veryant.com to help you identify and solve any difficulties. Here's a link to the documentation for the profiler in 2023R2.

DOCUMENTATION HIGHLIGHT

You might need to start more than one isCOBOL Server or more than one c-tree Server on the same machine.

For instance, you could use a second session for debugging, testing, or to better distribute the workload.

What's the proper way to start multiple server processes? You'll find instructions for the isCOBOL Server at

Starting multiple services on the same machine (Windows)

and

Starting multiple daemons on the same machine (Unix).

You will find instructions for the c-tree Server at

Starting multiple services on the same machine (Windows)

and

Starting multiple daemons on the same machine (Unix)

The Path to True Interoperability

Davide Spizzi, Technical Director of EMEA, talks about Web Services. It makes true interoperability attainable. It opens up a world of possibilities for your application, lets you keep the code you already have and expand it without locking you into one programming language

Is your business run with programs written in different languages (COBOL, C++, Java, etc.)? If so, you've probably struggled with getting these systems to interact.

There are a lot of reasons to pursue interoperability. Interoperability means all the business stakeholders have access to the same information, duplicate data is minimized, data quality is improved, customer-supplier management is expanded, security protocols and standards are consistent for all your data, and integration efforts are minimized.

Fortunately interoperability between disparate systems is almost always limited to data sharing. In the 1980's and 1990's data sharing was done with file transfers; saving and sending whole files of data, formatted programmatically for the receiving program. Then relational databases became dominant, and many companies moved their data into relational databases. This meant improved collaboration between teams, departments, and vendors, added value to ISVs with recognizable database brand names, and easy reporting and business intelligence gathering. For isCOBOL customers, it also meant keeping their COBOL programs unchanged by adding the DatabaseBridge product to their development system.

However, sometimes keeping data in a relational database isn't the best solution. It's a complex format that requires database management, often means additional costs to your IT load, and there are security concerns if you want to share data with sources outside your network. So a new alternative to databases is becoming popular: Web services

Most software languages understand XML or JSON tags. Web Services use these formats to exchange data, usually with APIs using REST or SOAP frameworks. Using Web Services means that applications written in different programming languages and running on different platforms can seamlessly exchange data over intranets or the Internet. It's a standardized, efficient way to connect your systems, as well as expose them in and outside of your network

.

At Veryant we're focused on making it easier to keep up with technology, and for Web Services, we've included the Service Bridge and Stream2Wrk tools in our EIS development system. ServiceBridge generates a bridge program that will convert your COBOL program's linkage section to an XML or JSON stream for outgoing and incoming data. The result is a COBOL program that can be used as a Web Service. Stream2Wrk generates the COBOL working-storage definitions for incoming XML, XDS, WSDL, and JSON data, so your COBOL program can consume an existing Web Service.

veryant

Evolution, without revolution

Contact Us

For supported customer email us at support@veryant.com

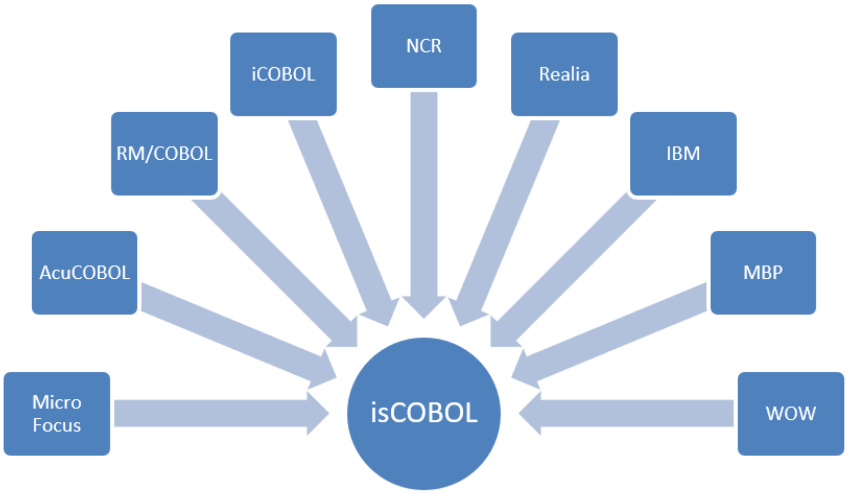
If you would like Veryant to contact you to schedule a technical product briefing, email us at info@veryant.com

If you would like Veryant to contact you for special quote or sales assistants email us at sales@veryant.com

Corporate Headquarters
6390 Greenwich Dr., Suite 225
San Diego, CA 92122 - USA
Tel (English): +1 619 797 1323
Tel (Español): +1 619 453 0914

European Headquarters
Via Pirandello, 29
29121 - Piacenza - Italy
Tel: +39 0523 490770
Fax: +39 0523 480784
emea@veryant.com

As always, 2023R2 contains multiple compatibility additions – as we continue to make your conversion process as smooth, quick, and pain-free as possible.



veryant.com

Follow **Veryant** on

veryant.com