2020/11

ISSUE 03

BOLETÍN TRIANUAL DE VERYANT E ISCOBOL

veryant NEWS

EN ESTE NÚMERO

EN VERYANT, NOS PREOCUPAMOS POR HACERLE LA VIDA MÁS FÁCIL.

Este número destaca 2020R2, con muchas formas de trabajar de manera más inteligente, que incluyen

- Un Profiler más fácil,
- Un depurador reducido
- Un llamado a las armas para usar el IDE

COBOL
Cool Business Oriented Language

Nuestros pensamientos están con todos aquellos que has sido impactados por el coronavirus.

WHAT'S NEW

Ya está aquí 2020 R2

Veryant se complace en anunciar la última versión de isCOBOL ™ Evolve, isCOBOL Evolve 2020 Release 2.

Cada nuevo release es emocionante para nosotros, porque demostramos a nuestros clientes que los escuchamos y que estamos trabajando arduamente para hacer que nuestros productos sean más ricos, más fáciles y más relevantes para las tendencias actuales. Esta versión no es una excepción.

El Profiler de 2020R2 facilita el establecimiento de aspectos particulares de su aplicación, tanto para ejecutar como para ajustar a sus necesidades. La depuración es más fácil y limpia: consulte el artículo titulado "Adiós a iscobol.debug.code_prefix" a continuación.

El IDE tiene funciones más integradas, por lo que no tiene que dejarlo para utilizar el Profiler, el "code coverage" o el "unit testing". Ustedes solicitaron poder modificar sus variables fuera de los editores gráficos, y en esta versión lo hemos hecho. ¿No usas nuestro IDE? Vea las razones por las que deberías hacerlo en nuestro artículo de Davide Spizzi, Gerente de Ingeniería de Soporte de Veryant.

En 2020R2 se han agregado cambios gráficos como un menú de hamburguesas, la creación de bitmaps con símbolos de glifos, una nueva clase de navegador web y un mejor rendimiento del compilador ha sido agregado a esta versión. Lea más aquí y busque una próxima demostración en video en nuestro canal de You Tube.

DESTACADOS DE LA DOCUMENTACION

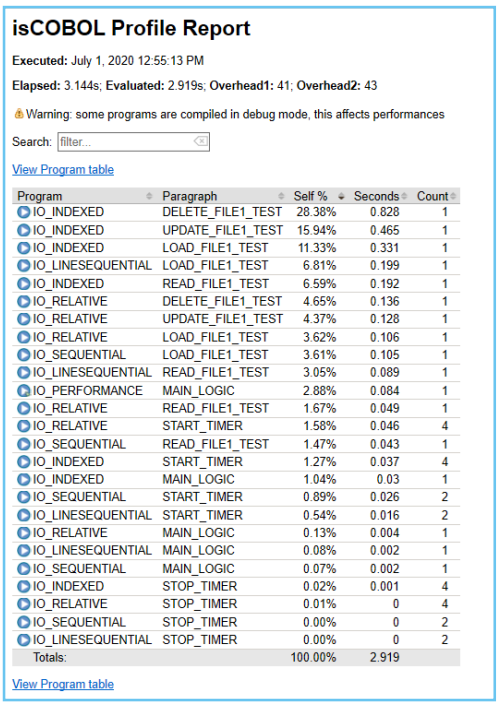
Optimización del Performance

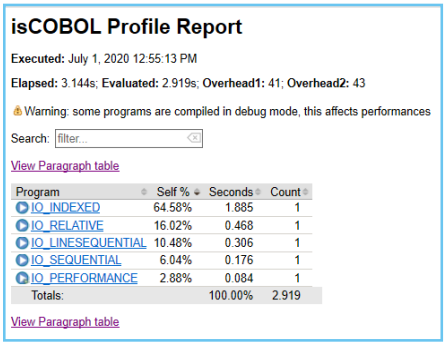
Hemos organizado todas nuestras sugerencias para mejorar el rendimiento en un solo lugar para que le sea más fácil ajustar la velocidad de su aplicación.
En los Apéndices de la documentación de isCOBOL Evolve hay un capítulo llamado Performance Tuning.
Hay tres capítulos , “Guidelines for faster compilation”, “Guidelines for better runtime performance”, y “Profiling COBOL programs”

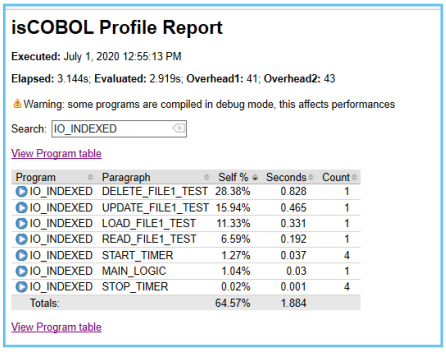
Encontrará sugerencias para una compilación más rápida desde la línea de comandos y el IDE, optimizaciones durante la compilación y ejecución de su aplicación, y sugerencias para Thin Client, acceso a datos y mejora del rendimiento de impresión.

El capítulo de Profiling tiene algunos cambios significativos en 2020R2; lea más al respecto en esta página.

Conozca el nuevo isCOBOL Profiler







2020R2 incluye mejoras al isCOBOL Profiler, lo que facilita la creación y visualización del reporte. La ejecución de sus programas con la opción “-profiler” creará un reporte HTML que muestra el tiempo y el porcentaje de tiempo total para cada programa, así como cada párrafo de los programas.
Se han agregado nuevas opciones para ayudarlo a obtener mas y mejor información, que incluyen:
La rutina C$PROFILER: para activar y desactivar el Profiler programáticamente, especialmente en las declaraciones ACCEPT. Esto elimina el tiempo que el usuario tarda en actuar.
Iscobol.profiler.excludes e iscobol.profiler. includes - Para restringir las clases perfiladas. Esto es útil si su interfaz de usuario está separada o si ha identificado un problema en un programa.
El reporte muestra una advertencia si los programas perfilados se compilaron con “debug”. Solo debe perfilar los programas NO compilados para depuración.

Has visto esto?

New YouTube videos:

New Features of isCOBOL Version 2020R1

isCOBOL for the Technologist

Modernization Methods with isCOBOL

isCOBOL and REST Services Demo

Mainframe Rehosting with HTWC

isCOBOL and zOS

Nuevos artículos en nuestra Knowledgebase:

How do I point my installed isCOBOL to a different Java location?

How to extend the length of AREA B in a program source code

How to run batch programs on a Linux server and compile and debug them from the IDE

How can I control or restrict access to my WebClient applications and see the user name displayed in the WebClient console

How to copy data from the clipboard

COBOL y nuestras Universidades

Marilyn Prince, Ingeniera de Ventas para Norteamérica, habla sobre la necesidad de instrucción COBOL en nuestras universidades, cómo mantener a los estudiantes interesados y qué está haciendo Veryant para ayudar.

La escasez de programadores COBOL ha estado en las noticias mucho últimamente, ya que estos desarrolladores se están retirando de la fuerza laboral. COBOL se enseña en muy pocas universidades y, a menudo, como una asignatura optativa. Pero el código COBOL todavía está funcionando en nuestros bancos, compañías de seguros y gobiernos. Para solucionar el problema, debemos analizar por qué existe una gran brecha entre la necesidad de programadores COBOL y la falta de prioridad para capacitarlos.

Podría deberse a que se esperaba que la mayoría de los programas COBOL fueran reemplazados para el año 2000 (lo que condujo al problema del Y2K). O tal vez a medida que los programas COBOL ronronean alegremente, la gerencia se vuelve complaciente con la necesidad de un cambio: la vieja máxima de “si no está roto, no lo arregle”. Quizás las universidades, compañías con fines de lucro sujetas a los caprichos de su cuerpo estudiantil, encontraron que COBOL estaba pasado de moda entre sus estudiantes, siguiendo el camino de los pantalones de campana y la Beatlemanía

Las causas probablemente sea una combinación de las tres mencionadas o tal vez otras mas. Pero hemos demostrado que COBOL todavía está aquí y seguramente para quedarse de una forma u otra. No solo necesitamos más programadores COBOL jóvenes, sino también programadores COBOL que puedan llevar nuestras aplicaciones al siglo XXI

Los programadores de COBOL ya no pueden salirse con la suya simplemente conociendo COBOL. Por ejemplo, COBOL y Java se utilizan juntos con mayor frecuencia, en parte porque las recientes mejoras de IBM permiten que COBOL y Java se ejecuten juntos en mainframes. Los dos lenguajes encajan bien, por lo que isCOBOL está escrito en Java y compila el código COBOL en clases de Java.

Las capacidades de COBOL como servicios web REST o SOAP significan que COBOL puede continuar funcionando fácilmente como un lenguaje de procesamiento de datos, mientras que los lenguajes más nuevos pueden proporcionar interfaces más llamativas que muestran fácilmente los datos en pantallas con gráficos, tablas y paneles. Incluso una aplicación estrictamente COBOL se puede mejorar con la programación OO y la inclusión de JavaBeans, por ejemplo. Es común ver una implementación COBOL con otros lenguajes o herramientas mezclados.
Un comentarista de Hacker News señaló que un programador de COBOL pasa su carrera “haciendo trabajos de mantenimiento en lugar de un desarrollo nuevo”. Eso podría ser cierto en algunos casos, pero para avanzar hacia el futuro, los programas COBOL tendrán que evolucionar, y el nuevo desarrollo es parte de eso, ya sea que tome la forma de pantallas gráficas COBOL, programación COBOL orientada a objetos o incluyendo COBOL en servicios web.
Pero, ¿cómo conseguir que los estudiantes se interesen lo suficiente como para pagar a las universidades por enseñar COBOL? Los nuevos estudiantes están alienados por el entorno de desarrollo de “pantalla verde”, pero ¿por qué no probar un IDE? Recientemente hablé con Joel Sweatte de la Universidad de East Carolina sobre la enseñanza de COBOL con el IDE de isCOBOL. Veryant se complace en proporcionar a sus estudiantes licencias gratuitas para su clase. Él dice que usar el entorno basado en Eclipse para sus programadores principiantes no solo facilita el aprendizaje, sino que eleva COBOL al mismo nivel que otros lenguajes en los que quieren trabajar, como Java y C ++.

En Veryant estámos haciendo nuestra parte para abordar la escasez de programadores COBOL ofreciendo licencias gratuitas a las universidades

No solo los estudiantes prefieren un entorno de desarrollo gráfico. Estamos viendo un mayor interés en isCOBOL IDE. Incluso hay algunos desarrolladores de mainframe que utilizan el IDE para desarrollar sus programas COBOL y luego los mueven al mainframe para compilarlos y ejecutarlos.
En Veryant estámos haciendo nuestra parte para abordar la escasez de programadores COBOL al ofrecer licencias temporales gratuitas a universidades, como East Carolina University y otras a través de nuestra conexión con COBOL Cowboys. También nos aseguramos de mantener nuestro producto tecnológicamente avanzado y fácil de usar, para que pueda conservar su COBOL, mantenerlo fácilmente y desarrollando aplicaciones modernas.

Russel Kirsch

El inventor del píxel de imágenes digitales murió el 11 de agosto de 2020. Tenía 91 años. Kirsch usó una fotografía digital de su hijo de 3 meses en 1957 para demostrar cómo las computadoras podían mirar imágenes.

Páginas Web Seguras

ahora https:// páginas seguras, incluyendo

www.veryant.com

y

support.veryant.com

CONTRATANDO AHORA

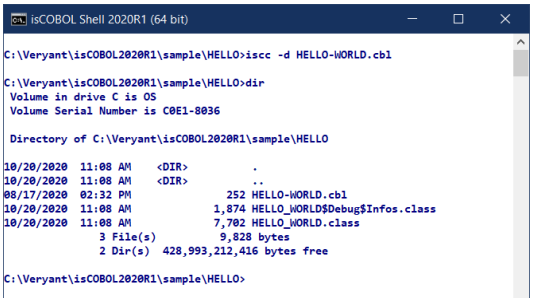
Si está buscando programadores COBOL, pruebe la página de la comunidad de IBM "Calling all COBOL programmers". Encontrará una combinación de programadores de mainframe y no mainframe enumerados con credenciales e información de contacto.

YouTube Scription Drive

Subscríbete a nuestro

Canal You Tube

Diciendole Adios a iscobol.debug. code_prefix

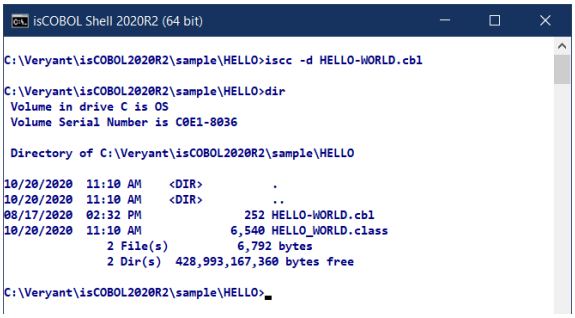


En 2020 R1 y versiones anteriores, la depuración de los programas isCOBOL solo se podía realizar si una copia del código fuente estaba disponible para el runtime Las variables de configuración se establecían para apuntar a estos archivos fuente.

`iscobol.debug.code_prefix` (to find the files locally)
`iscobol.debug.remote_source` (when debugging remotely)

La compilación con -d daba como resultó una clase adicional, en el formato de <class name>$Debug$Infos.class.
Hemos hecho que la depuración sea mucho más fácil en 2020 R2 al eliminar la necesidad de archivos class y configuración adicionales. Ahora, cuando compila con -do -dx, el código fuente se incluye con el archivo de clase compilado. A continuación, se incluyen algunos consejos para recordar cuando utilice esta nueva función de depuración:

- Para ver si una clase está compilada para depuración, aún puede ejecutarla con "isrun -info <nombre de clase>"
- La clase compilada para depuración será más grande y se ejecutará más lentamente que una no compilada para depuración.
- El código fuente de la clase está encriptado
- Puede usar los mismos comandos del depurador ("br 7 HELLO-WORLD" aún establece un punto de interrupción en la línea 7 del programa HELLO-WORLD).
- Aún puede usar las variables de configuración para cargar la fuente desde el disco para compatibilidad con versiones anteriore



No usas el IDE?, es fácil de usar

¿Todavía usa la línea de comando para compilar sus programas? ¿Notepad o vi para codificar? Creo que le sorprendería lo fácil que es pasar al IDE de isCOBOL y lo útiles que le resultarán sus funciones. Estas son algunas de las funciones del IDE que creo que le encantarán:

- Mantenga su estructura de almacenamiento de archivos actual y vincule los archivos a las carpetas del proyecto IDE - No es necesario cambiar su estructura o ubicación de origen actual.
- La función de verificación de sintaxis en tiempo real del IDE le permite ver inmediatamente los errores en el código fuente sin tener que compilar. Cuando lo haga, sabrá que no hay errores de sintaxis.
- Utilice la función Crear proyecto para asegurarse de haber compilado todo. Por ejemplo, si tiene un proyecto con 1,000 programas y cambia un archivo de copia usado por 50 de esos programas (algunos con archivos de copia anidados), Build Project recompilará solo aquellos programas que usan el archivo de copia. Cuando se completa la compilación, es fácil actualizar su entorno de producción ordenando la carpeta de salida por fecha y moviendo solo esos 50 programas.

Puede crear fácilmente un proceso para recompilar automáticamente todos sus programas durante la noche, por ejemplo, para las pruebas internas que necesita el departamento de pruebas. Una línea de comando de ejemplo para esto sería:

isIDE -data workspaceLocation -nosplash --launcher. suppressErrors -application com.iscobol.plugins.screenpainter.IscobolScreenPainter. builderApplication [project projectName] actualizar limpiar build logfile logFilename

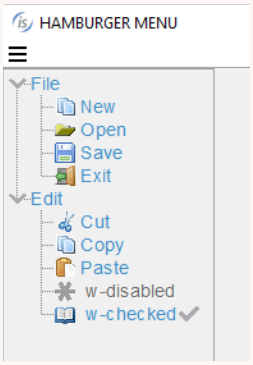
- El IDE se integra fácilmente con software de control de versiones de código como SVN, por lo que finalmente puede mantener sus cambios organizados y su código seguro.
- No es necesario que trabaje fuera del IDE para acceder a las actividades específicas de la líena de comando. Las Terminal View se conecta localmente o mediante SSH / Telnet desde el IDE
- Los hipervínculos pueden moverlo a travéz de su código sin tener que desplazarse constantemente
- Es útil ver su código de diferentes maneras: el IDE le muestra una vista de esquema, le permite expandir o contraer diferentes secciones de código y dividir su pantalla para ver dos partes diferentes de su código al mismo tiempo.
- El IDE incluye una búsqueda avanzada y un campo de acceso rápido para que pueda encontrar cualquier cosa dentro o fuera del IDE.
- Utilice el IDE en Mac y Graphical Linux, así como en Windows.

El IDE está incluido en su sistema de desarrollo, sin costo adicional. El equipo de soporte se complacerá en ayudarlo a pasar al IDE isCOBOL. Creo que una vez que se acostumbre, ¡se preguntará cómo ha podido vivir sin él!

Davide Spizzi,
Support Engineering Manager

NUEVO MENU HAMBURGUESA

Las pantallas GUI solían considerarse un entorno estático: solo tenía que dibujar con un tamaño de pantalla en mente.. En 1999 los ingenieros de Microsoft dijeron que las computadoras iban a "volverse invisibles", ocultándose en objetos comunes, y eso es muy cierto para muchos programas COBOL.. Ahora es más común tener que escribir pantallas receptivas para que se muestren en monitores, teléfonos inteligentes y tabletas. En Veryant, nos enfocamos en hacer que eso sea lo más fácil posible para usted



Version 2020R2 introduce un hamburger menu que puede ser implementado con una variable de configuración en lugar de cambiar su código.

COMO CAMBIAR EL ÍCONO DE TUS APLICACIONES

Es posible que esté utilizando la sintaxis de "icono" en la declaración DISPLAY WINDOW para mostrar un icono en la ventana. Ese icono también se convierte en el icono de la barra de tareas.

Otra forma de personalizar su icono es usando la variable de configuración:

```
iscobol.gui.icon_file=<filename>
```

No necesita cargar o descargar el ícono con W$BITMAP, y actualizar su logotipo es tan fácil como reemplazar el archivo del ícono.

Evolution, without revolution

Veryant LLC

El equipo de soporte técnico dedicado y experimentado de Veryant está comprometido a proporcionar los más altos niveles de atención al cliente

Para clientes con soporte, envíenos un correo a support@veryant.com

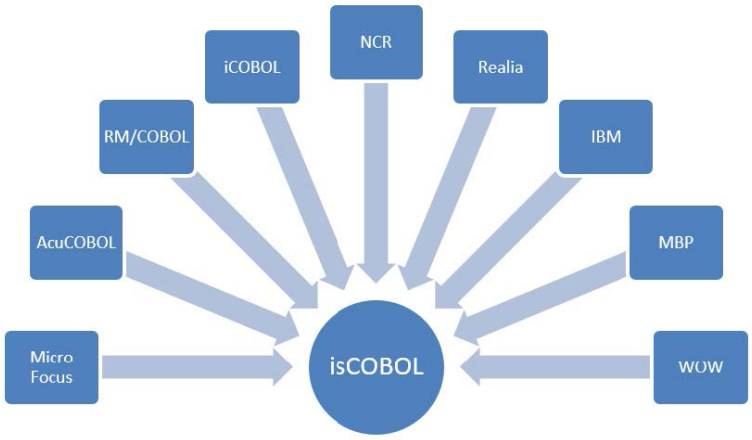
Si desea que Veryant se comunique con usted para programar una reunión técnica sobre el producto, envíenos un correo a info@veryant.com

Si desea que Veryant se comunique con usted para solicitar un presupuesto especial o asistencia de ventas, envíenos un correo a 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

Como siempre, 2020R2 contiene múltiples adiciones de compatibilidad, a medida que continuamos haciendo que su proceso de conversión sea lo más fluido, rápido y sencillo posible.



veryant.com

Ofreciendo opciones, flexibilidad y soluciones rentables para organizaciones con valiosos activos de COBOL

CONTACTENOS PARA UNA DEMOSTRACION

CLICK HERE

Follow Veryant on

veryant.com