



PUBLICACIÓN SEMESTRAL DE VERYANT E ISCOBOL

QUE MÁS HAY DE NUEVO EN 2023R1?



- Mayor compatibilidad con Micro Focus, Acucobol, IBM, DB2, ProCobol, y VAX/Open-VMS
- Linkage sections con número variable de argumentos, simplificadas
- Conversión de su juego de caracteres de datos y compatibilidad con caracteres nacionales para aumentar la compatibilidad internacional
- MASS-UPDATE para redibujar ventanas sin problemas
- Bitmaps en el control tree-view utilizando el estilo table-view



veryant

NEWS

EN ESTE NÚMERO

1. Lo nuevo de la versión 2023R1 **2.** Modificando su Ui para dispositivos móviles y tablets **4.** Replicación de Datos con c-treeRTG **8.** – Cuarta parte de Docker Containers **10** Documentation Highlights **11.** Es solo COBOL, no programación de Mainframe **12.** Tips & Tricks/Documentación Destacados/Nuevos Videos y KB's **13.** Ultima página

2023 R1 : Nuevas Opciones para el Desarrollador

La nueva versión 2023 R1 brinda a los desarrolladores más opciones con una extensión VSCode, un controlador de archivos Vision nativo, la capacidad de cargar WebClient de prueba y una mejor manera de administrar la replicación de datos de c-tree.

V Visual Studio Code es un entorno de desarrollo popular para COBOL, y Veryant ahora tiene una extensión isCOBOL que los programadores pueden usar para facilitar la edición de su código fuente en un entorno familiar.

El formato de archivo Vision de ACUCOBOL es popular y puede ser difícil dejarlo por varias razones. Ahora isCOBOL puede manejar archivos Vision tan fácilmente como Jisam y c-tree, con su nuevo controlador de archivos integrado, VisionJ..

WebClient continúa madurando hasta convertirse en una herramienta robusta para ejecutar sus aplicaciones COBOL en un navegador web o encapsuladas en una página web con cambios mínimos o nulos en los programas. Ahora hemos agregado una herramienta de prueba que puede automatizar sus casos de prueba y ejecutar múltiples ejecuciones para pruebas de carga.

La replicación de datos de c-treeRTG es fácil de usar y proporciona la estabilidad de datos necesaria y la capacidad de continuar ejecutando su aplicación cuando su servidor deja de funcionar. Hemos agregado un administrador de replicación gráfico basado en el navegador llamado Memphis (ver "Replicación de datos con c-treeRTG")

Modificando su UI para dispositivos Móviles y Tablets

Ahora que ya usted puede utilizar [WebClient](#) para ejecutar sus aplicaciones dentro de un browsers, hay algunos cambios que usted puede implementar para mejorar la experiencia del usuarios con pantallas mas pequeñas.

Cuando observa su aplicación ejecutándose en una pantalla más pequeña, como una tableta o un teléfono inteligente, puede parecer abarrotada y diminuta, lo que dificulta su uso. Pero con isCOBOL le facilitamos la realización de algunas modificaciones para que su aplicación se vea bien en diferentes tamaños de pantalla. Estos incluyen cambiar a un menú de hamburguesa, eliminar la barra de título que hace que parezca una aplicación y no una página web, aplicar un layout manager con una variable de configuración y obtener información sobre su entorno.

Revisa tu entorno

Si tu aplicación está corriendo con WebClient, puede tratar de desplegar en una pantalla mas pequeña Usa la rutina [C\\$GETRUNENV](#) rpara saber en que ambiente esta corriendo tu aplicación (thin-client, Webclient, standalone)

```
call "C$GETRUNENV" GIVING env-code  
if env-code = runenv-web-client
```

Quita la decoración de tu pantallas

Si está ejecutando WebClient, le recomendamos que configure los estilos [undecorated](#) y [maximized](#). Esto eliminará el marco y la barra de título de su aplicación y los expandirá para llenar la ventana del navegador sin importar el tamaño de la ventana del navegador. Su aplicación se parecerá más a una verdadera aplicación web que a una aplicación de escritorio.

Obten información del monitor

[C\\$MONITOR](#) rdevuelve el tamaño de los monitores en píxeles, lo que le indicará el tamaño del entorno en el que se va a mostrar su programa. Para clientes ligeros y clientes pesados, también le indicará cuántos monitores hay disponibles y cuál es el monitor principal. Esto puede ayudarlo a ahorrar recursos descargando soloel font correcto y le brinda control sobre dónde se muestra su aplicación.

Use el menú hamburger en lugar del menú clasico

Ahora es más común ver menús de hamburguesas en lugar de barras de menú en las aplicaciones web. Un menú de hamburguesas consta de tres líneas que pueden ocultar el menú y expandirse cuando se hace clic en ellas. En isCOBOL, es fácil cambiar sus menús..

El código y la lógica para declarar un menú de hamburguesas es el mismo que el de una barra de menú; la única diferencia es el código de operación utilizado para crearlo. La barra de menú usa el op-code WMENU-NEW, el menú de hamburguesas usa el op-code [WMENU-NEW-HAMBURGER](#).

Una vez que cambia el op- code al menú de hamburguesas, tiene muchas opciones para personalizar el diseño usando el op-code [WMENU-SET-ATTRIBUTE](#), incluyendo colores, fonts, tamaño, posición, e íconos.

Aplica un layout manager

Un layout manager cambiará el tamaño de su pantalla y controles con el tamaño de la ventana. Sugerimos una de dos opciones: [LM-ZOOM](#) o [LM-RESPONSIVE](#).

LM-ZOOM se puede configurar en el archivo de configuración para que se aplique a cada ventana que aún no tenga un layout manager asignado. Cambiará el tamaño y moverá los controles y los fonts para que quepan en la ventana

LM-RESPONSIVE requiere un poco más de esfuerzo, porque debe aplicarse a cada control. Pero hay algunas pantallas que se verían mucho mejor con este tipo de administrador de diseño. Para ver esto en acción, mire el ejemplo "RESPONSIVE" en issamples (en la sección de nuevas características de 2019R1)..

Aligera tus pantallas

A veces, una pantalla es demasiado compleja para mostrarse bien en una pantalla pequeña. Este es el momento de modificar estas pantallas y usar LM-RESPONSIVE para ayudar a rediseñarlas para diferentes tamaños de pantalla.

Hazlo de la manera fácil

En Veryant, queremos minimizar los cambios de código que debe realizar para actualizar su aplicación. Aquí le mostramos cómo realizar rápidamente algunos de los cambios descritos anteriormente:

1. Use code injection para que el compilador haga los cambios por usted. Por ejemplo, configurar la siguiente variable en el archivo de configuración de su compilador hará que cada ventana no esté decorada y cambie de tamaño y usará el layout manager LM-ZOOM

```
6 iscobol.compiler.gui.window.defaults= \  
7   undecorated, \  
8   resizable, \  
9   layout-manager lm-zoom
```

2. Cambia cada menú a un menú hamburger con la siguiente línea agregada al principio de tu aplicación

```
call "w$menu" using wmenu-set-attribute  
                        "menu-bar-flavor"  
                        "hamburger"
```

3. Maximize su ventana con esta línea después de desplegarla ventana (cambiandola al nombre de su window handle)

```
modify win-handle action action-maximize
```

Replicación de datos con c-treeRTG



Por *Fabrizio Cesura*
Senior Support Engineer

Mantener los datos sincronizados en varios servidores le permite tener copias de seguridad consistentes de los datos y poder cambiar a un servidor alternativo en caso de que el servidor actual no esté disponible.

El servidor c-treeRTG incluye un plugin para la replicación de datos. Puede habilitar este plugin en el archivo de configuración ctsrvr.cfg eliminando el punto y coma que comenta la siguiente línea:

```
;PLUGIN ctagent;<libraryname>
```

La configuración de este plugin es manejado con los siguientes archivos instalados con el producto en el folder config: Archivo

Archivo	Alcance	Ejemplo
ctagent.json	Principal archivo de configuración. Incluye la lista de archivos de configuración del agente de replicación	<pre>{ "detached": true, "configurationFileList": ["../config/replication/ctreplagent1. cfg"] }</pre>
replication/ctreplagent1.cfg	Archivo de configuración del agente de replicación	<pre>unique_id REPLAGENT1 target_authfile ../config/replication/target_auth.set target_server FAIRCOMS@localhost source_authfile ../config/replication/source_auth.set source_server FAIRCOMS@SourceIP batch_size 8192 read_timeout_ms 5000 log_file_name ../config/replication/replagent1.log exception_mode transaction file_filter <../config/replication/filter1.xml replicate_data_definitions yes</pre>

Archivo	Alcance	Ejemplo
replication/filter1.xml	Lista de archivos a replicar Se permiten Wildcards	<pre><replfilefilter version="1" persistent="y"> <file status="include">*.dat</file> <purpose>create_file</purpose> <purpose>read_log</purpose> </replfilefilter></pre>
replication/source_auth.cf	Credenciales para conectar el servidor origen	USERID ADMIN PASSWD FAIRCOM
replication/target_auth.cf	Credenciales para conectar el servidor destino	USERID ADMIN PASSWD FAIRCOM

El proceso de replicación se basa en archivos de registro de transacciones: el plugin de replicación lee los archivos de registro de transacciones para comprender qué modificación de datos se ha realizado en un servidor y solicita la misma modificación de datos en el otro servidor. Mientras que los programas realizan i-o en el servidor c-tree, el mismo i-o se replica en el otro servidor c-tree.

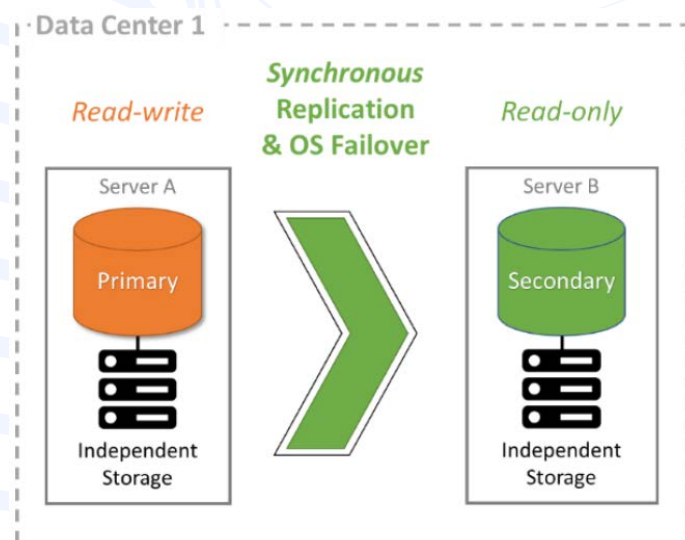
Solo se replican los archivos con el registro de transacciones habilitado.

Hay dos tipos de replicación de datos:

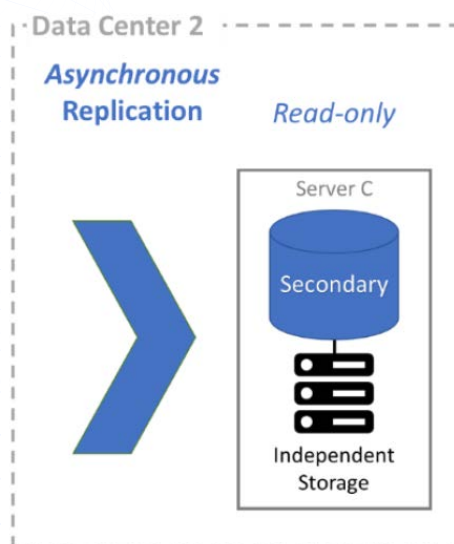
Replicación Asíncrona – replica datos entre dos servidores a través de un proceso sin bloqueo. Esto permite que el servidor principal procese los datos lo más rápido posible sin que un servidor secundario o la red lo ralenticen.

Replicación Síncrona – utiliza un proceso de bloqueo para replicar datos entre servidores para garantizar que ambos servidores hayan conservado una transacción en sus registros de transacciones antes de que el servidor principal informe a la aplicación que la transacción se ha confirmado. La replicación síncrona es necesaria para la alta disponibilidad porque solo ella puede garantizar que los datos sean siempre los mismos en ambos servidores en todo momento. Debido a que está bloqueando, no es tan rápido como la replicación asíncrona.

High Availability



Disaster Recovery



El tipo de replicación se establece en el archivo de configuración `ctreplagent1.cfg`. De forma predeterminada, se realiza una replicación asíncrona. Para realizar una replicación síncrona, elimine el punto y coma que comenta la siguiente línea:

```
;syncagent yes
```

Una vez configurado, la replicación de datos comienza junto con el servidor c-tree.

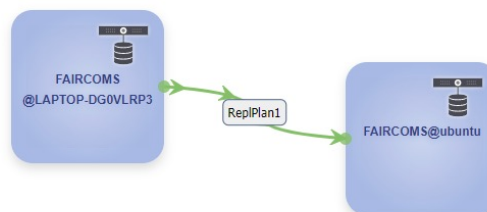
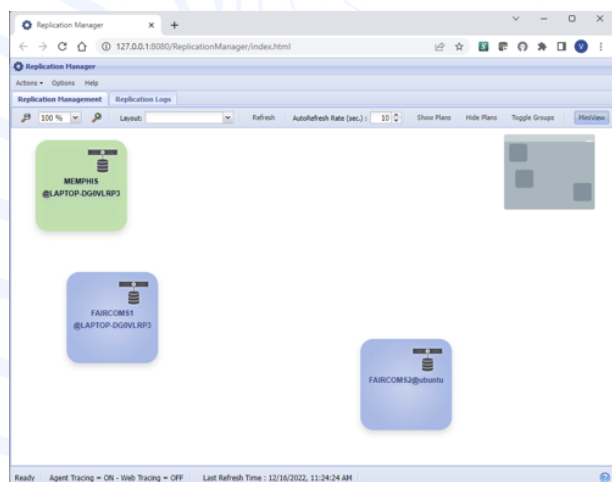
La actividad de replicación de datos se rastrea en un archivo de registro llamado **ctreplagent.log**.

Utilice Memphis para la administración

Desde la versión 3.0.2.783 distribuida junto con isCOBOL 2023, la mejor manera de configurar y monitorear la replicación de datos es con la nueva interfaz gráfica de usuario, llamada “Memphis”, implementada en HTML5..

Después de iniciar el archivo ejecutable llamado “Memphis” y los servidores c-treeRTG en la red, puede abrir un navegador e ingresar la URL: <http://server-ip:8080/ReplicationManager> (o <https://server-ip:8443/ReplicationManager> para usar el protocolo HTTPS). Verá la aplicación de Replication Manager con una lista de servidores c-tree disponibles.

En la aplicación del Replication Manager, puede arrastrar un servidor de “origen” sobre un servidor de “destino” para crear un plan de replicación, basado en una publicación donde puede definir si la replicación es para tablas SQL, archivos ISAM o ambos. También puede filtrar archivos ISAM incluidos en carpetas específicas que deben incluirse o excluirse de la replicación. Los últimos pasos del proceso de configuración son: Suscribirse a la Publicación, eligiendo si la replicación es unidireccional o bidireccional, y Desplegar el plan de replicación. Después de la implementación, si la replicación se inició correctamente, la línea que conecta los dos servidores c-tree se vuelve verde.



Se puede activar una descripción general del estado de replicación actual y la cantidad total de confirmaciones, registros agregados, actualizados o eliminados desde el elemento del menú “Activity” del menú pop-up del plan implementado..

Plan activity for plan : ReplPlan1

Options

Time Frame: 1 Hour Start Date: 12/16/2022 End Date: 12/16/2022 Start Time: 11:17 AM End Time: 12:17 PM Refresh Auto Display Refresh Rate: 0 sec. Dump

Details

Statistics Exceptions Function Timing

Page 1 of 1 Snapshot Status: Running Stop Collection Rate: 5 Displaying 1 - 28 of 28

Type	Identification	Time	Replication				Thread Counts			Transaction Throughput			
			Paused	Aborted	Failures	Latency	Analyze	Apply	Commits	Add	Update	Delete	
20	sourceToTarget	12/16/2022 12:15:20 PM				0	1	8	3	2	0	1	
21	sourceToTarget	12/16/2022 12:15:30 PM				0	1	8	3	2	0	1	
22	sourceToTarget	12/16/2022 12:15:40 PM				0	1	8	3	2	0	1	
23	sourceToTarget	12/16/2022 12:15:51 PM				0	1	8	3	2	0	1	
24	sourceToTarget	12/16/2022 12:16:01 PM				0	1	8	3	2	0	1	
25	sourceToTarget	12/16/2022 12:16:11 PM				0	1	8	3	2	0	1	
26	sourceToTarget	12/16/2022 12:16:21 PM				0	1	8	3	2	0	1	
27	sourceToTarget	12/16/2022 12:16:31 PM				0	1	8	3	2	0	1	
28	sourceToTarget	12/16/2022 12:16:41 PM				0	1	8	3	2	0	1	

Close



La parte final de una serie de cuatro partes sobre contenedores isCOBOL y Docker, este artículo es una guía paso a paso para acceder a la memoria compartida mientras se ejecuta más de un contenedor.

Por Valerio Biolchi Senior Support Engineer

Un segmento de memoria compartida es un fragmento de memoria que se comparte entre varios procesos.

A los contenedores Docker se les asignan 64 MB de memoria compartida de forma predeterminada

Comunicación entre el servidor isCOBOL y el servidor c-treeRTG con memoria compartida ejecutándose en diferentes contenedores

Prerequisitos

Los requisitos previos en una computadora Linux de 64 bits (donde se construyen los Dockers) son:

- El motor Docker (<https://docs.docker.com/engine/install/>) instalado desde un repositorio
- Un ambiente isCOBOL SDK con su respectiva licencia
- Un servidor Docker isCOBOL en una máquina Linux 64bit (ver instrucciones en el primer artículo de la serie, [aquí](#)).
- Un servidor Docker c-treeRTG en una máquina Linux 64bit (ver instrucciones en el tercer artículo de la serie, [aquí](#))

Ejecuta el servidor Docker c-treeRTG

```
docker run --name ctreeserver -d -v shm:/tmp/ctreedbs -v /isapplication/data:/isapplication/data -p 5597:5597 --ipc=shareable ctree
```

`-v shm:/tmp/ctreedbs`

Los tokens de protocolos de memoria compartida se comparten

`--ipc=shareable`

Habilita el uso compartido de memoria en todo el contenedor

`/isapplication/data:/isapplication/data`

Monta el directorio host, /isapplication/data, en /isapplication/data en el contenedor

El servidor c-treeRTG iniciado por este contenedor permite conexiones TCP/IP a través del puerto predeterminado expuesto 5597, además de memoria compartida.

Comunicación entre el servidor isCOBOL y el servidor c-treeRTG con memoria compartida ejecutándose en diferentes contenedores

Ejecuta el servidor Docker isCOBOL

```
docker run --name iscobolserver -dit -v shm:/tmp/ctreedbs -v /isapplication/data:/isapplication/data --ipc=container:ctreeserver -p 10999:10999 iscobol
```

`-v shm:/tmp/ctreedbs`

Los mismos tokens de protocolos de memoria compartida compartidos por el contenedor “ctreeserver” anterior

`--ipc=container:ctreeserver`

EHabilita la posibilidad de unirse al IPC de otro contenedor “compartible”.

El servidor de aplicaciones isCOBOL iniciado por este contenedor permite conexiones TCP/IP a través del puerto predeterminado expuesto 10999, además de la memoria compartida.

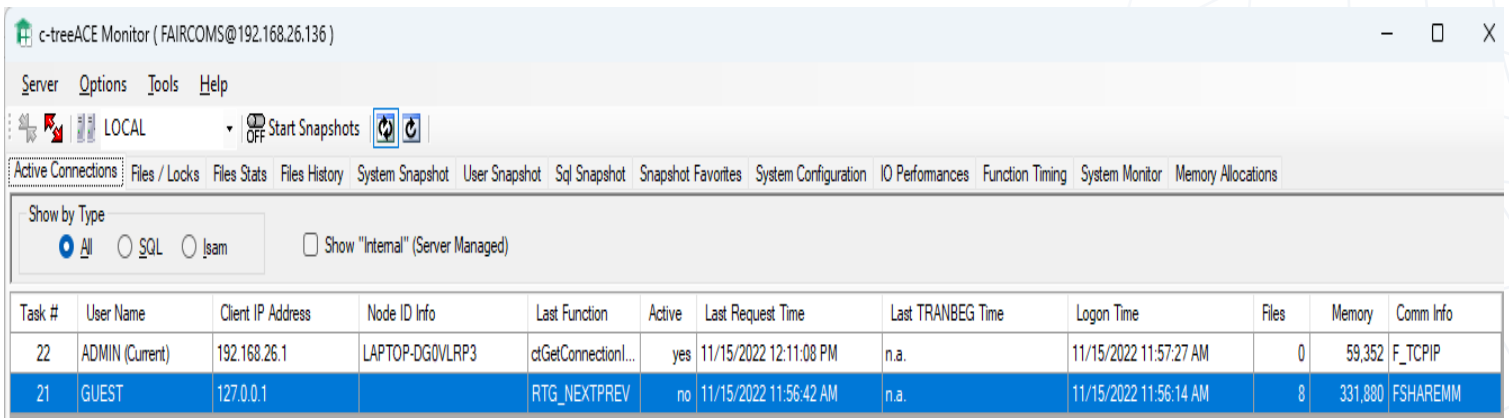
Ejecute el programa ejemplo IO_INDEXED desde una PC cliente

```
iscllient -c runtime.properties -hostname 192.168.26.136 IO_INDEXED
```

Este comando ejecutará el programa IO_INDEXED en el contenedor “iscobolserver” y trabajará con los archivos de datos indexados c-tree del contenedor “ctreeserver” a través del protocolo de memoria compartida.

Ejecute el monitor c-treeACE

Ejecutando el monitor c-treeACE desde un cliente de Windows mostrará el protocolo de comunicación utilizado y los archivos abiertos:



The screenshot shows the c-treeACE Monitor application window. The title bar reads "c-treeACE Monitor (FAIRCOMS@192.168.26.136)". The menu bar includes "Server", "Options", "Tools", and "Help". The toolbar shows "LOCAL" and "Start Snapshots" (OFF). The main menu includes "Active Connections", "Files / Locks", "Files Stats", "Files History", "System Snapshot", "User Snapshot", "Sql Snapshot", "Snapshot Favorites", "System Configuration", "IO Performances", "Function Timing", "System Monitor", and "Memory Allocations". Below the menu is a "Show by Type" section with radio buttons for "All" (selected), "SQL", and "Isam", and a checkbox for "Show 'Internal' (Server Managed)". The main area displays a table of active connections.

Task #	User Name	Client IP Address	Node ID Info	Last Function	Active	Last Request Time	Last TRANBEG Time	Logon Time	Files	Memory	Comm Info
22	ADMIN (Current)	192.168.26.1	LAPTOP-DG0VLRP3	ctGetConnection...	yes	11/15/2022 12:11:08 PM	n.a.	11/15/2022 11:57:27 AM	0	59,352	F_TCPIP
21	GUEST	127.0.0.1		RTG_NEXTPREV	no	11/15/2022 11:56:42 AM	n.a.	11/15/2022 11:56:14 AM	8	331,880	FSHAREMM

Comunicación entre el servidor isCOBOL y el servidor c-treeRTG con memoria compartida ejecutándose en diferentes contenedores

Mejoras en el rendimiento

```
iscclient -c runtime.properties -hostname 192.168.26.136 IO_INDEXED
```

Trabajar con memoria compartida le brinda un rendimiento significativamente mejor que trabajar con TCP/IP. Esta tabla compara los resultados de rendimiento trabajando con TCP/IP y con memoria compartida.

IO_INDEXED NUM-TIMES=10000	TCP-IP	SHARED MEMORY
WRITE	4.40	0.85
READ	3.94	0.17
REWRITE	4.36	0.41
DELETE	3.35	0.58
TOTAL TIME	16.05	2.01

DESTACADOS DE LA DOCUMENTACIÓN

Opciones para implementar su aplicación de cliente ligero

Existen diferentes posibilidades para el despliegue de una aplicación de cliente ligero. Mientras que el back-end del servidor está instalado en su servidor, la parte del cliente ligero se puede distribuir de diferentes maneras dependiendo de si

desea una instalación de cliente cero, si prefiere usar configuraciones o si está apegado a la tecnología heredada como JWS (Java Web Start).

Todas estas posibilidades se explican en detalle en la Documentación de isCOBOL en

[isCOBOL Application Server : Client deployment.](#)

Es solo COBOL, No programación de Mainframe



Por Massimo Bertoli, President of Veryant,
Sobre la importancia para una casa de software
que utiliza el sistema abierto COBOL de enseñar
COBOL a los desarrolladores nuevos y jóvenes.

Recientemente se ha hablado mucho sobre algunos desarrolladores de COBOL que están envejeciendo y jubilándose, y cómo eso debería afectar sus planes futuros para sus programas de COBOL. Sin embargo, no se ha discutido una diferencia importante: el programador COBOL de mainframe frente al programador COBOL de sistema abierto (Linux, Unix, Windows y Mac).

La mayoría de los desarrolladores de COBOL tienen una amplia experiencia en áreas que no son necesarias para los programas COBOL de sistema abierto; CICS y JCL son dos de esas habilidades que no tienen relevancia para su aplicación. El sistema abierto COBOL es más fácil de desarrollar y comprender, y requiere un conjunto de habilidades diferente al de un programador de Mainframe.

Si recientemente buscó un desarrollador de COBOL para contratar, probablemente descubrió que los solicitantes disponibles son mayores, tienen experiencia en tecnología Mainframe que no necesita y se encuentran en los rangos de salarios más altos. En su lugar, **¿por qué no enseñar COBOL a sus desarrolladores existentes?** Es posible que conozcan un lenguaje de programación orientado a objetos como Java o C++, y considerarían que aprender un lenguaje de procedimientos como COBOL es una experiencia nueva e interesante.

A estos empleados típicamente más jóvenes les resultará fácil aprender COBOL específico de la forma en que se ha utilizado en su aplicación. Por ejemplo, podrían pasar algún tiempo aprendiendo SQL incorporado si su aplicación lo usa. Puede aprovechar a sus expertos COBOL salientes de una manera centrada en la aplicación y acortar el tiempo de aprendizaje al eliminar las lecciones innecesarias.

De hecho, capacitar a los empleados actuales le ahor-

rará tiempo y dinero, brindará a sus desarrolladores oportunidades de desarrollo y crecimiento, y hará que su departamento de TI esté más integrado, menos aislado por el lenguaje.

Otro beneficio importante de enseñar COBOL a sus programadores existentes es el camino fácil hacia la interoperabilidad que ofrece. El futuro de COBOL normalmente no implica una nueva programación de COBOL. Su COBOL será solo una pieza de una aplicación multilingüe. Veryant se centra en esta interoperabilidad como un camino hacia adelante para los sistemas abiertos COBOL.

“No se ha discutido una diferencia importante: el programador COBOL de mainframe frente al programador COBOL de sistema abiertos.”

Una vez que sus desarrolladores conozcan COBOL, serán un gran recurso para agregar nuevas funciones e integrar su aplicación COBOL con nuevos lenguajes. isCOBOL ya lo coloca en este camino porque compila y se ejecuta como un programa Java. Es un ajuste natural para integrarse con el nuevo desarrollo de Java.

Las habilidades de los programadores de mainframe pueden ser difíciles de aprender, pero todo lo que necesita es el lenguaje COBOL, y probablemente ya tenga los recursos que necesita en su departamento de TI. Si su aplicación utiliza alguna sintaxis isCOBOL específica (como GUI, subprocesos u OOP) además de COBOL básico, Veryant se complace en realizar la capacitación o proporcionar material de apoyo para su capacitación interna sobre estos temas. Estamos aquí para ayudarlo a tener éxito.

COBOL

Consejos y Trucos

Una excelente manera de modernizar sus programas es cambiar el color de las ventanas, por ejemplo, de gris a blanco.

Probablemente ya sepa que podría usar la inyección de código para hacer esto en todas sus pantallas a la vez desde el archivo de propiedades, sin cambiar su código, usando esta propiedad:

```
iscobol.compiler.gui.window.defaults=color rgb x#FFFFFF
```

PERO - si ya tiene establecida la propiedad COLOR en la declaración de la ventana de visualización, esta configuración del programa anulará cualquier configuración de COLOR en su archivo de propiedades, por lo que la inyección de código no funcionará.

Aquí hay otra opción: establecer colores degradados.

Las propiedades de degradado de color no se sobrescriben con la configuración de COLOR del programa. Si configura los dos colores de degradado en el mismo color (por ejemplo, blanco), aún puede usar la inyección de código para cambiar el color de la ventana sin modificar su código.

Por ejemplo, puede usar las siguientes propiedades durante la compilación de los programas::

```
iscobol.compiler.gui.window.defaults=gradient-color-1 rgb x#FFFFFF gradient-color-2 rgb x#FFFFFF
```

Has Visto esto?



Nuevos vídeos de YouTube

[Starting the Application Server as a Windows service](#)

[ISMIGRATE from the Command Prompt](#)

[ISMIGRATE log files](#)

[c-treeRTG's Replication Manager - Memphis](#)

[WebClient Test Tool](#)

Nuevos artículos de la base de conocimientos

[Send SMS from isCOBOL using Twilio](#)



Evolution, without revolution



Contáctenos

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

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

Si desea que Veryant se comunique con usted para obtener una cotización especial o asistentes de ventas, envíenos un correo electrónico a sales@veryant.com

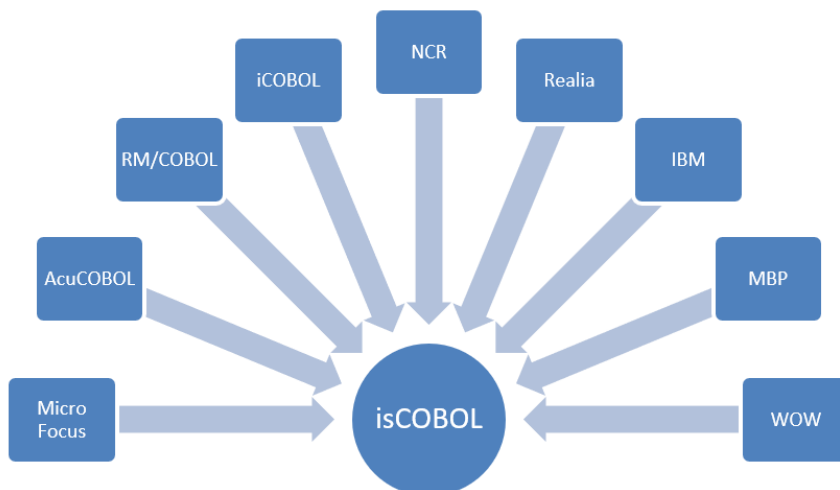
Corporate Headquarters

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

European Headquarters

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

Como siempre, 2023R1 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

Follow Veryant on



veryant.com

©2023 Veryant - All Rights Reserved

Veryant Newsletter Issue 08 2023