2022

ISSUE 06

PUBLICACIÓN TRIANUAL DE VERYANT E isCOBOL

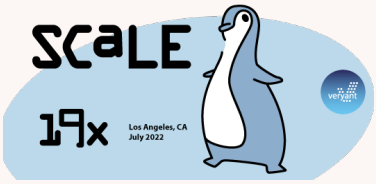
veryant NEWS

THIS ISSUE

ÚNETE A NOSOTROS EN SCALE 19X EN LOS ANGELES

JULIO 28-31, 2022

La conferencia de código abierto administrada por la comunidad más grande de América del Norte está de regreso y tenemos nuestro stand como patrocinador. ¡Visítanos para recibir un regalo! Más información aquí.



ÚNETE A NOSOTROS EN

Twitter, LinkedIn o Facebook para estar al día con las novedades de Veryant

Mira nuestros Videos, y Subscríbete a nuesto canal de YouTubel

2022 Release 1 y Más

2022R1 ha llevado a isCOBOL en un gran paso adelante para mantenerse actualizado y relevante en entornos más grandes y modernos

Ahora soportamos todas las versiones LTS (soporte a largo plazo) de Java; Java 1.8, Java 11 y Java 17. Puede ver la fecha de lanzamiento y las fechas de finalización del soporte para las versiones de Java aquí. El IDE de isCOBOL se actualiza para usar la versión más nueva y estable de Eclipse. 2022R1 está certificado para ejecutarse en Windows 11 con Java 11 o 17. Hemos agregado la agrupación en clústeres al WebClient, por lo que ahora puede equilibrar la carga de procesamiento del WebClient en varios servidores. Vea nuestro video que demuestre cómo hacer esto en nuestro sitio de YouTube,aquí. 2022R1 tiene algunas formas diferentes de expandir el manejo de su archivo de configuración. Puede combinar un archivo con otro con iscobol.conf. copy=<filename>. Puede codificar para restablecer todas las variables a sus valores originales, o establecerlos en valores almacenados en un archivo, con C$CONFIG.

El manejo de bitmaps y fonts que van con su aplicación es más fácil ahora con la sintaxis de COPY RESOURCE para incluirlos en sus clases. Evite procesos colgados en segundo plano configurando iscobol.display.message.timeout para que un messagebox se cierre automáticamente. Como de costumbre, tenemos muchas mejoras en el control de la GUI, incluidas muchas para el menú de hamburguesas, agregando búsquedas, su propio administrador de diseño y varias otras especializaciones para este nuevo tipo de menú que funciona bien en pantallas pequeñas.Puede configurar la ventana principal para que se oscurezca cuando muestra una ventana modal que exige la atención del usuario.

Estas son solo algunas de las nuevas funciones de 2022R1. Puedes obtener más información sobre todos ellas en release documentation y el video.

¿Qué vas a hacer con tu COBOL?

Un argumento para mantener su COBOL en lugar de empezar de nuevo, por Daniel Cardenas, Business, Development Director of Latin America

Recientemente, muchas personas han hablado sobre COBOL y las aplicaciones “antiguas”, y muchos fuera de la industria de COBOL no entienden por qué no se hizo algo sobre la situación antes de la crisis actual.

Pero aquellos de nosotros que usamos COBOL todos los días sabemos que es “más fácil decirlo que hacerlo”. Su aplicación COBOL no solo funciona, trabaja duro todos los días. Pensar en cambiar un sistema esencial con décadas de modificaciones, aplicaciones específicas de la industria, con millones de líneas de código, no es tarea fácil. Decidir qué dirección tomar es desalentador, y desafiaría a cualquier tomador de decisiones a decir que no ha sucumbido a la “parálisis de análisis” sobre el tema al menos una vez.

Los tomadores de decisiones responsables de decidir qué hacer con sus programas COBOL heredados se encuentran en una encrucijada. Hay dos caminos a seguir: mantener el COBOL y actualizarlo, o desecharlo y comenzar de nuevo con un paquete genérico o una reescritura total. Una vez que empiezas por un camino, es muy difícil cambiar sin tener que volver a la encrucijada y empezar de nuevo.r

Una vez, mi hijo limpió su habitación sacándolo todo y luego volviendo a colocarlo todo. Su teoría era que solo devolvería exactamente lo que necesita, y estaría mejor organizado. Y eso es básicamente lo que pasó. Sin embargo, le tomó todo un fin de semana, por lo que tuvo que dormir en el sofá dos noches durante el proceso y toda la casa fue un desastre durante ese tiempo. Y finalmente, su habitación se volvió tan desordenada como antes porque las cosas que pensó que no necesitaba seguían volviendo a su habitación con el tiempo.

Reemplazar su COBOL con un paquete genérico o reescribir totalmente su aplicación es un poco como su método de limpieza de habitaciones “cortar y quemar”. Sí, puedes terminar con algo claro y brillante. Pero aún tendrás que tocar cada parte de tu aplicación COBOL para intentar copiar su funcionalidad.

Y luego pague una gran cantidad de dinero a un proveedor externo para que modifique su paquete genérico y que se adapte a su negocio, o espere años para que sus programadores escriban una aplicación desde cero que reemplace algo que tomó décadas para afinar.

La actualización de su COBOL existente también conlleva problemas, por ejemplo, ¿quién mantendrá el COBOL si todos sus programadores de COBOL se jubilan? ¿Y cómo les dices a tus clientes que tienes una aplicación de última generación escrita en COBOL? ¿Cómo elige al proveedor de COBOL correcto con el conjunto de herramientas correcto para que su actualización sea exitosa?

“Su aplicación COBOL no solo funciona, trabaja duro todos los días.”

Ahí es donde Veryant encaja en la decisión. Nuestro IDE se basa en un IDE estándar de la industria: Eclipse, por lo que la mayoría de los programadores ya están familiarizados con el entorno. COBOL es un lenguaje relativamente fácil de aprender, y dibujar pantallas y codificar en COBOL es muy fácil cuando usan las herramientas IDE integradas que hemos creado.

Si desea minimizar las interacciones de su programador con COBOL, es fácil crear servicios REST individuales y colocar su COBOL en una “caja negra” de fondo, y usar un lenguaje más moderno que se adapte mejor a las pantallas gráficas, como Java, JavaScript o Python para su código de interfaz de usuario.

isCOBOL convierte su COBOL en código Java, luego compila nuevamente en una clase Java. Por lo que sus clientes o usuarios pueden ver, usted tiene una aplicación Java.

Quizás la decisión más importante de su empresa sea qué hacer con su COBOL. Espero que considere seriamente quedárselo. Llámenos para una discusión, demostración o análisis de código.

Iconos en el IDE de isCOBOL

El IDE utiliza diferentes iconos en las 3 secciones del explorador IDE para representar el estado del proyecto. Aquí está un mapa para comprender esos íconos.

Icons used in all sections


Poyecto isCOBOL


Libreria isCOBOL: lista la versión de la librería de isCOBOL utilizada en el proyectot

Icons in the Structural view


isCOBOL screen program


isCOBOL WOW program


Libreria isCOBOL: lista la versión de la librería de isCOBOL utilizada en el proyecto


Diseñador de pantalla para el programa o de formularios para el programa WOW


Sección de Diseñador de Reportes, conteniendo todos los reportes de cada programa .r


Diseñador de Reportes


Diseñador de Working storage y Local storage


Diseñador de Linkage section


Sección de archivo que contiene todos los conjuntos de datos (archivos) de cada programa.


Conjunto de datos (la representación de un archivo utilizado dentro del programa)


Párrafo de evento para el programa Screen y el programa WOW

Iconos de decoración

Cada icono se puede decorar con algún pequeño símbolo para mostrar información adicional.

Una advertencia del compilador agregará un triángulo amarillo en la esquina inferior izquierda. Por ejemplo, el icono de la fuente se convierte en.  y el folder 

Un error del compilador agregará una cruz roja en la esquina inferior izquierda. Por ejemplo, el icono de la fuente se convierte en  y el folder 

El icono de error anulará el icono de advertencia..

File view

- Carpeta de proyecto vacía
- Carpeta de proyecto con algunos archivos.
- Carpeta de lista con algunos archivos en ella
- Copy file
- List file
- Código fuente del programa
- Identificación del programa: debajo de cada código fuente hay este icono. Este representa la clase generada
- ID de clase: este icono tiene el mismo significado que el anterior, pero para un ID de clase
- El contenedor de los métodos de una identificación de clase.
- Un método del class-id

Data view

- Un método del class-id
- Diseñador de descripción de archivo (FD)
- Diseñador de llaves de archivor
- Diseñador de manejo de i-o de archivosr
- Diseñador de archivos EFD
- Diseñador de párrafos de evento de archivo

Icon decorations

ISi el archivo o la carpeta no están físicamente presentes en la carpeta del proyecto, pero están vinculados desde otra ubicación, se agrega un símbolo de vínculo en la esquina inferior derecha del icono. Por ejemplo, el icono de la fuente se convierte en y el folder

Estas son las decoraciones isCOBOL. Es posible que vea otros agregados por complementos de Eclipse, como SNV.

Has visto esto?

Nuevos Videos en YouTube

isCOBOL's Profiler- how it works and a demonstration
isCOBOL's Code Coverage Utility
What's new in 2022R1 – with demonstration
All about isCOBOL Licensing
Customizing isCOBOL
WebClient Clustering

Nuevos Artículos en nuestra base de conocimiento (KB)

How can you wait for several threads to finish?
How to use font-based icons in isCOBOL GUI programs
Can I use COBFILEIO to give access to my c-tree files to a Java program?
Modernize your character application

Teniendo un volumen persistente para el servidor isCOBOL en un Docker

La segunda de una serie de tres partes sobre isCOBOL y Docker Containers, este artículo le muestra cómo usar datos persistentes en su contenedor. Aquí hay una guía paso a paso del ingeniero senior de soporte, Valerio Biolchi

Prerequisitos

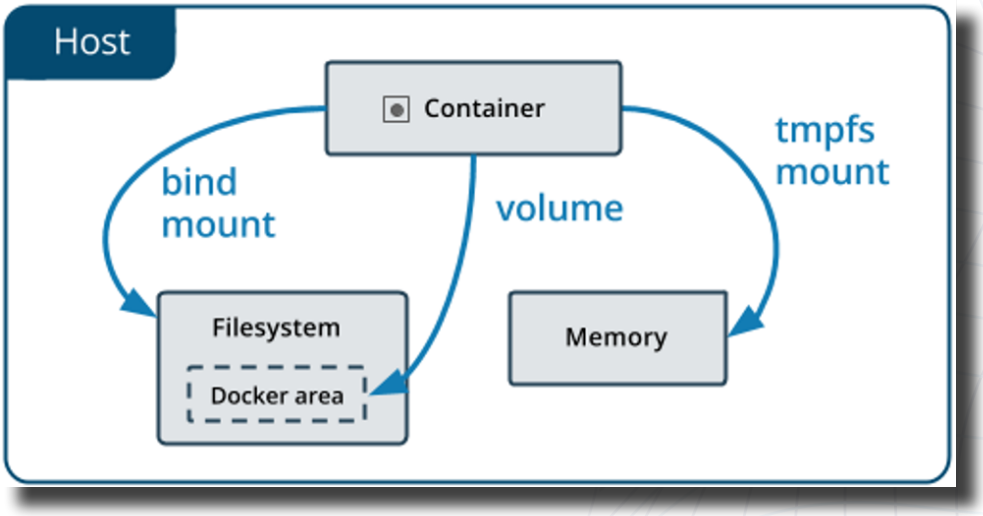
Los prerequisitos en una computadora Linux de 64bit donde el docker es armadot) son :

- El motor Docker(https://docs.docker.com/engine/install/) instalado desde un repositorio
- Un ambiente licenciado de isCOBOL SDK (isCOBOL runtime con licencia válida)
- Un Servidor isCOBOL ejecutando en Docker con el programa ejemplo ISAPPLICATION corriendo. Ver instrucciones en el primer artículo de la serie, aquí.

Administrar datos en un Docker

Como comportamiento predeterminado, todos los archivos creados dentro de un contenedor se almacenan en una capa de escritura. Esto significa que los datos no persisten cuando se elimina el contenedor y es bastante complejo hacer que los datos estén disponibles para otro proceso que se ejecuta fuera del contenedor. Además, la capa de escritura del contenedor está fuertemente acoplada con la máquina host donde se ejecuta el contenedor y no es sencillo mover esos datos a otro host. Por último, pero no menos importante, una abstracción adicional adicional reduce el rendimiento en comparación con el uso de volúmenes de datos, que escriben directamente en el sistema de archivos del host. El sistema Docker proporciona dos opciones para que los contenedores almacenen archivos en la máquina host para que los archivos sean persistentes incluso después de que el contenedor se detenga: volúmenes y montajes de enlace.

Docker también proporciona a los contenedores una forma de almacenar archivos en la memoria de la máquina host llamada “tmpfs”. Dado que se usa la memoria de la máquina host, obviamente estos archivos no son persistentes, pero siguen siendo útiles para obtener el mejor rendimiento o por razones de seguridad para proteger los datos.



Los volúmenes se almacenan en el sistema de archivos del host donde se ejecuta Docker. Por ejemplo, en "/var/lib/docker/volumes/". Como buena práctica, los procesos que no son de Docker deben evitar modificar este sistema de archivos. Los volúmenes son el mecanismo preferido para conservar los datos generados y utilizados por los contenedores de Docker.

Los montajes de enlace eran la opción original para datos persistentes en un Docker. Se pueden almacenar en cualquier parte del sistema de archivos del host. Los procesos que no son de Docker en el host de Docker o en un contenedor de Docker pueden modificarlos en cualquier momento. Como buena práctica, debe usar volúmenes siempre que sea posible; sin embargo, en algunos casos, podría considerar usar montajes de enlace entre los procesos de Docker y la máquina Host; por ejemplo, si necesita compartir un archivo de configuración entre la máquina host y los contenedores.

Usando Volúmenes

Podemos crear un volumen usando el subcomando "crear" y pasando un "nombre" de volumen como argumento:

```
root@ubuntu# docker volume create myAppVolume
```

El subcomando "ls" muestra todos los volúmenes conocidos por Docker:

```
DRIVER                    VOLUME NAMES
local                     myAppVolume
```

Para mostrar información detallada sobre uno o más volúmenes, usamos el subcomando "inspect"::

```
root@ubuntu# docker volume inspect myAppVolume
[
    {
        "CreatedAt": "2022-04-21T01:18:45-07:00",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/myAppVolume/_data",
        "Name": "myAppVolume",
        "Options": {},
        "Scope": "local"
    }
]
```

Debemos tener en cuenta que el controlador del volumen describe cómo el host Docker ubica el volumen. Los volúmenes también se pueden definir para que se ubiquen en almacenamiento remoto a través de NFS. En el ejemplo anterior, el volumen está en el almacenamiento local.

Arrancar un contenedor con un volumen

Si inicia un contenedor con un volumen que aún no existe, Docker crea el volumen por usted. El siguiente ejemplo monta el volumen "myAppVolume" en ":/isapplication/data" en el contenedor

```
docker run --name iscobolserver -v myAppVolume:/isapplication/data -p 10999:10999
-d iscobol
```

La opción -v contiene tres componentes, separados por dos puntos:

Source directory or volume name
Mount point within the container
(Optional) 'ro' if the mount is to be read-only

Dentro del contenedor, debe tener la configuración de ISAPPLICATION con un archivo de configuración "config.properties". El archivo debe incluir iscobol.file_prefix, la propiedad que define el directorio donde se crean y encuentran los archivos COBOL, establecida en el punto de montaje del volumen definido. Por ejemplo:

```
iscobol.file_prefix=/isapplication/data
```

Después de la ejecución del contenedor Docker, encontrará archivos indexados de datos COBOL dentro de la carpeta física denominada "/var/lib/docker/volumes/myAppVolume/_data".

Docker File

Aquí está el contenido del archivo Docker utilizado en el artículo anterior, como referencia:

```
# Dockerfile

FROM  openjdk:11
MAINTAINER  Veryant
ENV ISCOBOL=/var/isCOBOL2022R1
ENV ISCOBOL_CLASSPATH=${ISCOBOL}:${ISCOBOL}/isapplication
ENV ISSERVER_OPTS="-c ${ISCOBOL}/isapplication/config.properties"

RUN mkdir ${ISCOBOL}

COPY iscobol.properties ${ISCOBOL}/iscobol.properties
COPY isCOBOL_SDK2022R1/lib ${ISCOBOL}/lib
COPY isCOBOL_SDK2022R1/bin/isserver ${ISCOBOL}/bin/isserver
COPY isCOBOL_SDK2022R1/sample/isapplication ${ISCOBOL}/isapplication

WORKDIR "${ISCOBOL}/isapplication"

CMD ${ISCOBOL}/bin/isserver run
```

Integración COBOL/Web

El poder del componente IWC-PANEL

El IWC-PANEL es un control de contenedor que puede usar en su aplicación COBOL para incluir componentes web en su pantalla cuando su programa está encapsulado en una página web. El control IWC-PANEL le permite usar los verbos COBOL MODIFY e INQUIRE para interactuar con el componente web.En general, un componente iwc-panel en la screen section tendría el siguiente aspecto:

```
03 f-map iwc-panel
            js-name          "f-map"
            line  5          column  2
            size 68 cells    lines 15 cells
            value            fmap-struct
            event procedure FMAP-PROC.
```

La propiedad JS-NAME contiene un identificador que se enviará a la página web al crearla, para que se pueda crear el componente web correspondiente.
La propiedad VALUE del control contiene la estructura del mensaje que se usa para enviar acciones al panel en la página web, por lo que usaría la declaración MODIFY en esa propiedad para enviar un mensaje a la página web. Por otro lado, si la página web ejecuta un performAction en el panel, se llamará al procedimiento de evento de su programa COBOL y un INQUIRE en la propiedad de valor devolverá el mensaje que se ha enviado.

DÍGALE A LA PAGINA WEB PARA CREAR EL COMPONENTE

Para cada IWC-PANEL en un formulario, se llama a una devolución de llamada en la página web, con los detalles necesarios para realizar la inicialización del componente. El objeto webclientInstance.options .compositingWindowsListener define las devoluciones de llamada para varios eventos, como la creación del IWC-PANEL y la apertura y cierre de ventanas.
Una creación de IWC-PANEL activará la devolución de llamada windowOpened y se pasará una referencia al IWC-PANEL como argumento de función. La devolución de llamada puede comprobar la propiedad .name para determinar qué control se ha creado y reaccionar en consecuencia. Aquí hay un fragmento de código JavaScript que muestra el manejo de windowOpened:

```
compositingWindowsListener:{
   windowOpened: function(win) {
      if (win.name === 'f-map'){
         createMap(win)
      }
   },
},
```

ENVIA INFORMACIÓN A LA PAGINA WEB

El siguiente fragmento de código muestra cómo interactúa el programa COBOL con la página web. Se crea un mensaje usando OOP a un jsonStream, luego se pasa a JavaScript con una simple instrucción COBOL "modify f-map value"

```
SHOW-ON-MAP.
    move "selectOffice"        to fmap-action
    move offices(office-index) to selected-office
    set objJsonStream to jsonStream:>
                         new(selected-office, 1);;
    set strbuffer     to string-buffer:>new
    objJsonStream:>writeToStringBuffer(strbuffer)
    move strbuffer:>toString to fmap-data
    modify f-map value fmap-struct.
```

Javascript recibe y analiza los datos en este fragmento:

```
if (actionName === 'selectOffice'){
    let office = JSON.parse(data);
    selectOffice(office);
 }
```

OBTENER INFORMACIÓN DE LA PAGINA WEB

Un cambio en el mapa desencadena una acción performAction en la página web que es capturada por el programa COBOL. El programa ejecuta el procedimiento de eventos del control, FMAP-PROC, que utiliza una declaración de "inquire f-map value..." para obtener el evento.J
avascript envió la información en este fragmento de código:

```
infoWindow.addListener('closeclick', () =>{
   if (mapControl){
      mapControl.performAction(
        {actionName: 'pinClosed',
        data:marker.title});
  }
 });

if (mapControl){
    mapControl.performAction(
    {actionName: 'pinClicked',
     data:marker.title});
}
```

Y el programa COBOL lo procesó en el procedimiento de eventos del control

```
FMAP-PROC.
     if event-type = ntf-iwc-event
        inquire f-map value in fmap-struct
        evaluate fmap-action
        when "pinClicked"
             move fmap-datato sel-description
        ...
        when "pinClosed"
        ...
        end-evaluate

     end-if.
```

En la carpeta sample/issamples de su instalación de isCOBOL, proporcionamos un proyecto completo que muestra cómo integrar un componente de mapa de Google en una aplicación COBOL y cómo interactuar con él. Las instrucciones para ejecutar el programa SAMPLES encapsuladas en una página web, un requisito para usar el IWC-PANEL, se encuentran en el documento README.md que se encuentra en la carpeta issamples.

Destacados de la Documentación

Configuración común de conexión JDBC

Puede administrar sus datos en cada base de datos compatible con JDBC con isCOBOL

Le ofrecemos dos formas de hacerlo. Puede hacerlo escribiendo código SQL incorporado (ESQL) en su programa o puede usar nuestro producto Database Bridge para escribir el ESQL por usted. En ambos casos, deberá agregar el driver JDBC adecuado a Classpath y configurar la cadena de conexión JDBC.

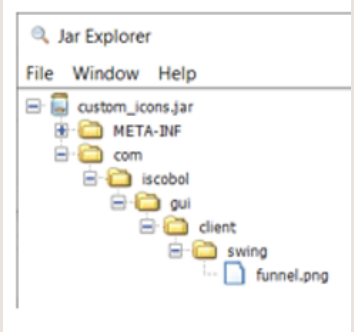
Para que le resulte más fácil hacerlo, la documentación de isCOBOL incluye una colección de drivers JDBC y cadenas de conexión para las bases de datos relacionales más comunes. aquí.

Personalización de iconos predeterminados

El runtime de isCOBOL incluye una serie de iconos predeterminados que se utilizan en varios lugares de la GUI. Estos íconos son archivos PNG y GIF almacenados en el paquete com.iscobol.gui.client.swing en la biblioteca iscobol.jar. Es posible personalizar estos íconos agregando una biblioteca (o una carpeta) con el mismo paquete antes de iscobol.jar en CLASSPATH. Suponga que desea personalizar el ícono de embudo que se muestra en el encabezado de la cuadrícula cuando se establece el estilo COLUMNA FILTRABLE o la propiedad TIPOS DE FILTRO. Solo tiene que colocar un archivo llamado funnel.png en una estructura de carpetas llamada com/iscobol/gui/client/swing y luego crear un jar a partir de él, p.e.

Copie este jar en el directorio “jars” de su SDK de isCOBOL y, a partir de ahora, cuando ejecute un programa COBOL con el SDK de isCOBOL, verá su embudo personalizado en los encabezados de la cuadrícula.

Consulte la documentación en línea aquí para ver la lista de íconos que puede personalizar y para obtener más detalles sobre la creación de su .jar personalizada.



Ventajas de la fuente de símbolos

Usando los opcodes WBITMAP-LOAD-SYSTEM-FONT y WBITMAP-LOAD-SYSTEM-FONT-EX

Los fonts de símbolos son bibliotecas de fonts. Estas son algunas de las ventajas de usar fonts de símbolos para sus imágenes en lugar de archivos de imagen externos e independientes. Los íconos tradicionales pierden calidad cuando aumentas o disminuyes su tamaño, o cuando los abre y los guarda en un programa de edición de fotos. Los fonts de iconos utilizados por estos códigos de operación (opcodes)de la rutina W$BITMAP almacenan imágenes en un formato vectorial y no pierden calidad. Esto significa que con fuentes de símbolos:

- Si necesita más de un tamaño de un ícono (por ejemplo, 32 y 16 píxeles), no tiene que crear y mantener dos archivos de íconos separados para mantener la misma calidad.
- Cuando agrega un ícono al final de una tira, mantiene la calidad de la tira agregando código, en lugar de tener que abrir la tira en un programa de edición de fotos, degradando la calidad.
- Si su aplicación tiene dos esquemas diferentes (por ejemplo, claro y oscuro), puede establecer estas propiedades en su código, en lugar de crear dos conjuntos de imágenes con diferentes colores.

Las fuentes de símbolos ofrecen una gran variedad de imágenes. Por ejemplo, Font Awesome, una de las fuentes de sistema más populares, tiene más de 7000 imágenes.

Aquí hay un ejemplo de código que cambia el tamaño y el esquema de color de una tira definida:

```
If use-big-image
   move 32 to imageWidth
else
   move 16 to imageWidth
end-if

If dark-theme
   move -13421772 to imageColor
else
   move -6710886  to imageColor
end-if

CALL "W$BITMAP" USING
      WBITMAP-LOAD-SYMBOL-FONT
                  fontHandle
                  charactersSequence
                  imageWidth
                  imageColor
            GIVING bitmapHandle.
```

Puede leer más sobre cómo funcionan las fuentes del sistema en nuestro artículo de KB aquí.

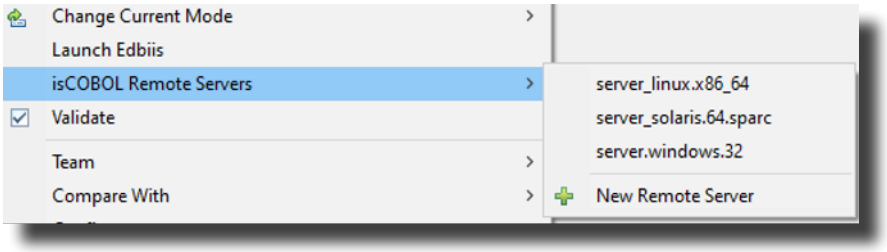
Use el IDE para ejecutar sus programas batch en forma remota

Su IDE puede compilar y ejecutar su código de forma local o remota, con el servidor remoto del IDE. Si tiene procesos por lotes que necesita ejecutar en un servidor remoto, con un poco de configuración puede iniciarlos desde el IDE en lugar de hacerlo directamente en el servidor.

Una vez que inicie el servidor de aplicaciones de isCOBOL con el interruptor especial -ide en el servidor remoto, cree un proyecto y defina el servidor remoto en el espacio de trabajo de su IDE, luego vincúlelos o únalos. Una vez que haya hecho eso, tendrá más modos de compilación y tiempo de ejecución; además de la "ejecución" normal, encontrará "@<nombre de su servidor>.Ejecutar", por ejemplo.

Puede usar el servidor remoto para ejecutar más que programas por lotes. Los programas con una interfaz de usuario también funcionan bien, porque el servidor remoto utiliza tecnología de cliente ligero.

Los detalles está en nuestra documentación aquí, ¡y el soporte está a solo un correo electrónico de distancia!



En donde estan mis licencias!?!

El runtime framework (y otros productos de isCOBOL) busca licencias en un archivo llamado "iscobol.properties" en 4 lugares, y un quinto lugar y el nombre que especifique. Se utiliza la licencia más reciente encontrada. Aquí está la lista de ubicaciones y el orden buscado.

Para ver qué licencias ve el runtime, ejecute

```
iscrun -license
```

Obtendrá una lista de todos los archivos "iscobol.properties" encontrados y qué licencias se recogieron y en qué orden..

1. /etc/iscobol.properties
2. <userhome>/iscobol.properties
3. <java classpath>/iscobol.properties
4. -c <any path and name passed from the command line>
5. $ISCOBOL/iscobol.properties

veryant

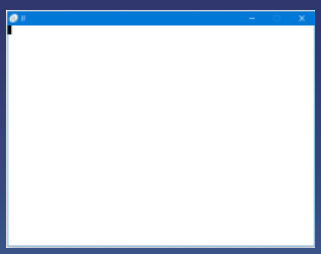
VENTANAS UNDECORATED (SIN DECORAR)

Que son y cuando tiene sentido utilizarlas.

Here's a Cuando se usa el estilo "undecorated" (sin decoración) decoraciones nativas como el marco y la barra de título no se muestran.
Este estilo de ventanas es útil en el entorno de WebClient donde es posible que desee eliminar la barra de título de la ventana para que la pantalla del programa se vea como un formulario HTML común o esté incrustado en una página.

Puede utilizar este estilo en ventanas gráficas flotantes, independientes e iniciales (estándar). No tiene efecto en las ventanas de acoplamiento y MDI, y las ventanas de notificación ya no están decoradas de forma predeterminada.

Aquí hay una ventana decorada:



Y aquí está la misma ventana con la propiedad "undecorated". Note que tanto la barra del título como el borde azul se han ido.

Evolution, without revolution

Contáctenos

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

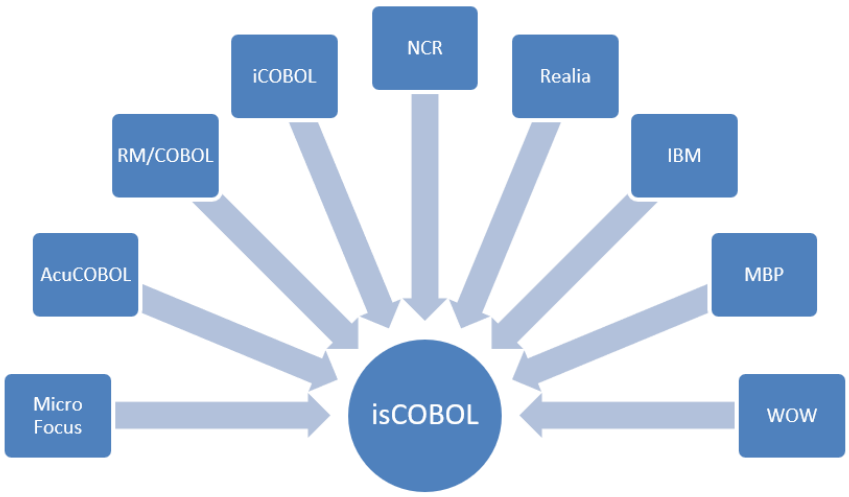
Si desea que Veryant se comunique con usted para programar una sesión informativa técnica sobre el producto, envíenos un correo electrónico **info@veryant.com**

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, 2022R1 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