ISSUE 07
2022
DIÁRIO SEMESTRAL DA VERYANT E isCOBOL

veryant NEWS

NESTA EDIÇÃO

POR FAVOR, JUNTE-SE A NÓS NO

LinkedIn ou Facebook para se atualizar

Assista nossos vídeos de demonstração e se inscreva no nosso canal no YouTube

CONTE PARA NÓS:

Queremos espalhar a palavra sobre o quão maravilhoso a Veryant é. E achamos que você pode nos ajudar.

Se você souber de outra empresa – ISV ou usuário final – executando aplicativos em COBOL, envie-nos suas informações de contato, e nós lhe enviaremos um Crédito Visa pré-pago de US$ 50,00 no Cartão de Crédito. Fácil assim!

2022 R2 está aqui

iisCOBOL 2022 Release 2 foi lançado em 12 de setembro de 2022.

sCOBOL 2022 Versão 2 foi lançado 12 de setembro de 2022. Inclui:

- **Telas GUI mais nítidas com alto reconhecimento de DPI controles e janelas**
- **Um novo compilador capaz de superar todos os tempo de processamento em lote de nossos concorrentes e níveis de instruções por segundo**
- **Um recurso de pré-processador do compilador para que você pode alterar seu código COBOL instantaneamente – enquanto está sendo compilado.**

2022 R2 está aqui

Ter consciência de alto DPI é um recurso importante se você escreveu suas telas GUI em COBOL e planeja executar seus aplicativos em telas de tamanhos diferentes. O velho método de presumir que todas telas têm o mesmo número de pontos, ou pixels, por polegada (DPI) e em seguida, esticando-os se o DPI for maior está OK quando tudo está rodando monitores aproximadamente do mesmo tamanho.

Mas quando você adiciona smartphone e telas de tablets, bem como aqueles laptops com resolução enorme que estãosaindo agora, suas telas podem ter começado a parecer confusas e datada. Nós resolvemos isso em 2022R2 com janelas e Controles GUI que se ajustam ao DPI corretamente, tornando suas telas tão afiadas e atualizadas como você originalmente os programou. Você não precisa fazer algo como recompilar ou adicionar uma variável para o seu arquivo de propriedades – simplesmente atualize seu tempo de execução para o nova versão e execute-a!

Temos trabalhado em um projeto completamente novo de compilador usando tecnologia mais recente para que seus aplicativos sejam executados mais rapidamente com menos memória. E esta versão é nosso primeiro lançamento desse compilador. Você pode usá-lo para fazer seu lote, não-ui, os programas voam!

Remova, adicione ou altere seu código usando nosso novo compilador linha por linha recurso de pré-processador. Escreva um programa para fazer algo como mover linhas gravado no console em um arquivo de log, adicione linhas de comentário no início de cada programa para imprimir ou qualquer mudança que você quiser fazer em seus programas sem realmente mudar o código. Escreva um programa para fazer a mudança, e o compilador irá ler seu programa e fazedr o que você precisa antes de criar o Java código e compila para uma classe Java.

CONTE PARA NÓS:

Se você não é um cliente da Veryant, e esta interessado em mudar para isCOBOL, agora é o melhor momento para fazê-lo.

Temos ofertas especiais para ISVs e usuários finais corporativos até o final do ano para tornar mais fácil de pagar, e a compatibilidade para facilitar a mover seu código para isCOBOL.
Saiba mais sobre este negócio aqui.

Eventos Mouse

Você sabia que pode gerenciar eventos de mouse em todos os controles?

Alguns controles retornam um evento ao parágrafo de procedimento de evento quando é importante interceptar eventos de mouse como o “clique” ou “clique duplo” em um ícone ou célula. Por exemplo, os eventos MSG-GOTO-CELL-MOUSE em grades, MSG BITMAP-CLICKED e MSG-BITMAP-DBLCLICK em grades e campos de entrada com bitmaps.

Você também pode receber eventos em outros controles ativando o estilo NOTIFY-MOUSE, para que esses eventos sejam retornados ao procedimento de evento: MSG-MOUSE-ENTER, MSG MOUSE-EXIT, MSG-MOUSE-CLICKED, MSG-MOUSE DBCLICK.

Um uso típico desses eventos é em controles como bitmaps, rótulos e quadros. Esses controles não disparam nenhum evento por padrão, portanto, definir o estilo NOTIFY-MOUSE oferece aos desenvolvedores um controle mais preciso sobre o design da interface do usuário.

Por exemplo, aqui está um trecho de código para gerenciar o evento MSG MOUSE-CLICKED em um rótulo:

```
...
03 l1 label line 2 col 2 size 10
      title "Click me"
      notify-mouse event label-events
...
label-events.
    if event-type = msg-mouse-clicked
       perform do-action
    end-if.
```

O que são Diretivas do Compilador?

As diretivas do compilador são uma sintaxe especial escrita no código-fonte para declarar algo que é específico para esse código-fonte. Ao colocar as opções do compilador específicas do código-fonte no código-fonte, você pode compilar todos os seus programas com um conjunto de opções padrão definidas no script SDK ou no nível do projeto IDE.

Aqui estão alguns exemplos:

- Se apenas alguns de seus programas precisam usar linhas longas, você pode definir o formato apenas para esses programas adicionando esta diretiva ao topo dos programas que precisam dela:
  >> SOURCE FORMAT FREE

- Para programas de formato Fixo (também conhecido como ANSI) que precisam escrever linhas longas, uma boa alternativa ao formato Livre é usar esta diretiva (equivalente à opção –sl):
  >> IMP MARGIN-R IS AFTER END OF RECORD

Se apenas um programa precisar da opção do compilador –big para ter uma compilação limpa, evitando o erro do compilador java “too many constants”, você pode definir: >> IMP OPTION “-big”

Outro benefício de usar diretivas de compilador é escrever uma fonte que precisa incluir ou excluir alguma parte do código, dependendo de uma condição específica. Mais informações sobre isso são explicadas neste artigo da base de conhecimento:

https://support.veryant.com/support/phpkb/question.php?ID=27

DESTAQUES DA DOCUMENTAÇÃO

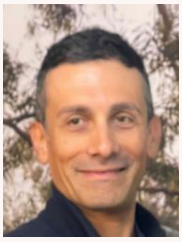
isCOBOL suporta SQL embutido sem a necessidade de pré-compilar as fontes. O código SQL incorporado é compilado junto com outras instruções COBOL em uma etapa. O programa COBOL resultante se conectará a bancos de dados com tecnologia JDBC.

A abordagem isCOBOL tem algumas vantagens se comparada com uma abordagem de pré-compilador:

- Processo de compilação mais simples, pois a compilação é feita em uma etapa
- Implementação mais simples, pois você precisa apenas alterar a biblioteca do driver JDBC para acessar diferentes bancos de dados, mantendo a mesma classe compilada
- Segurança de thread do driver JDBC que permite executar os programas em ambientes de servidor de aplicativo como o Tomcat

Como posso mover meus programas ESQL de um ambiente de pré-compilador como o Oracle. Pré-processador Pro*COBOL ou DB2 para JDBC? Existe algum problema que eu deva estar ciente? A resposta a essas perguntas pode ser encontrada no seguinte Transiting Guias:

Transitioning from Pro*COBOL
Transitioning from the IBM DB2 Preprocessor



A terceira parte de uma série de quatro partes sobre isCOBOL e Docker Containers, este artigo é um guia passo a passo para executar mais de um container. Do engenheiro de suporte sênior, Valerio Biolchi

Os aplicativos modernos consistem em diferentes componentes que precisam se comunicar entre si.

Comunicação entre o isCOBOL Server e o c-treeRTG em execução em diferentes contêineres do Docker

Pré-requisitos

Os pré-requisitos em um computador Linux de 64 bits (onde o docker é construído) são:

- O mecanismo do Docker (https://docs.docker.com/engine/install/) instalado a partir de um repositório

- Um ambiente isCOBOL SDK licenciado
- Um servidor isCOBOL em execução no Docker com configuração e execução de amostra ISAPPLICATION. Veja as instruções no primeiro artigo da série, aqui.

No mundo real, além do domínio do simples tutorial hello-world, executar apenas um contêiner não é suficiente para a maioria dos aplicativos. Um aplicativo moderno geralmente consiste em vários componentes, como um banco de dados, um servidor da Web e alguns microsserviços. Portanto, se você deseja executar todos os seus componentes em containers, como os aplicativos podem se comunicar?

Como os contêineres se comunicam entre si, se deveriam estar isolados?

Vejamos uma comunicação simples entre contêineres do Docker, quando eles estão sendo executados no mesmo host (às vezes chamado de rede de host único).

Como os contêineres se comunicam?

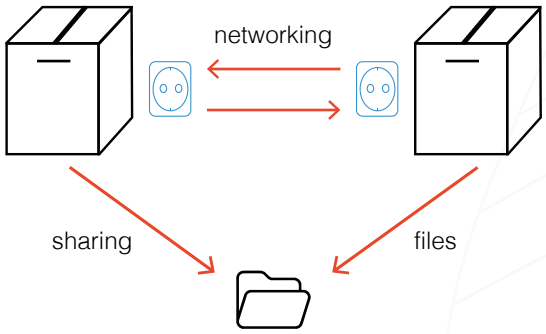
Primeiro, uma rápida visão geral! Embora os containers tenham um nível de isolamento do ambiente ao seu redor, eles geralmente precisam se comunicar uns com os outros e com o mundo externo.

Você leu os outros artigos e executou seus primeiros containers Docker. Mas agora você está lutando para entender como executar mais de um container ao mesmo tempo. Se os containers do Docker estiverem isolados, como diabos eles se comunicam entre si?

Dois contêineres podem se comunicar de duas maneiras, geralmente:

- Comunicação por rede: os containers são projetados para serem isolados. Mas eles podem enviar e receber solicitações para outros aplicativos usando a rede.

- Compartilhamento de arquivos em disco: alguns aplicativos se comunicam lendo e gravando arquivos. Esses tipos de aplicativos podem se comunicar gravando seus arquivos em um volume, que também pode ser compartilhado com outros containers.



Construindo sua rede (virtual)

Uma rede Docker permite que seus contêineres se comuniquem entre si

Se você estiver executando mais de um contêiner, poderá permitir que seus contêineres se comuniquem conectando-os à mesma rede.

Em uma rede, um contêiner possui um endereço IP e, opcionalmente, um nome de host.

Você pode criar diferentes tipos de redes dependendo do que você gostaria de fazer. Abordaremos as opções mais fáceis:

- A rede de ponte padrão, que permite comunicação simples entre contêineres por endereço IP e é criada por padrão.

- Uma rede de ponte definida pelo usuário, que você mesmo cria, que permite que seus contêineres se comuniquem entre si usando o nome do container como um nome de host.

Construindo uma rede de ponte

A rede mais simples no Docker é a rede de ponte. Também é o driver de rede padrão do Docker. Uma rede de ponte oferece comunicação simples entre contêineres no mesmo host..

Quando o Docker for inicializado, ele criará uma rede padrão chamada… bridge.
Ele deve iniciar automaticamente, sem nenhuma configuração exigida por você. A partir desse ponto, todos os contêineres são adicionados à rede de pontes, a menos que você diga o contrário. Em uma rede de ponte, cada container recebe seu próprio endereço IP. Assim, os contêineres podem se comunicar uns com os outros por IP.

A digitação da docker network deve mostrar a rede da ponte na lista

```
root@ubuntu:/home/valerio/myDocker/myApp4# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
6e432c2bd969   bridge    bridge    local
a11a782681ea   host      host      local
8eb1f36b9925   none      null      local
```

Crie e inicie o c-treeRTG Container do servidorr

Conforme explicado no primeiro artigo, começamos com o arquivo de instruções “Dockerfile”. Estes são os diretórios mínimos que precisamos copiar para iniciar corretamente o servidor c-treeRTG:

```
COPY c-treeRTG_v3.0.2/config ${CTREE}/config
COPY c-treeRTG_v3.0.2/data ${CTREE}/data
COPY c-treeRTG_v3.0.2/server ${CTREE}/server
COPY c-treeRTG_v3.0.2/tranlogs ${CTREE}/tranlogs
```

Aqui está o conteúdo completo do Dockerfile que criamos para ter um contêiner funcionando totalmente com o c-treeRTG Server:

```
# Dockerfile
FROM  openjdk:11
MAINTAINER  Veryant
ENV CTREE=/var/c-treeRTG3.0.2
RUN mkdir ${CTREE}
COPY c-treeRTG_v3.0.2/config ${CTREE}/config
COPY c-treeRTG_v3.0.2/data ${CTREE}/data
COPY c-treeRTG_v3.0.2/server ${CTREE}/server
COPY c-treeRTG_v3.0.2/tranlogs ${CTREE}/tranlogs
WORKDIR “${CTREE}/server”
CMD ./ctreesql run
```

Crie a imagem c-treeRTG

Agora podemos construir a imagem do docker ctree a partir da pasta que contém o Dockerfile. O nome do repositório (ctree neste exemplo) deve estar em letras minúsculas.

```
root@ubuntu:/home/valerio/myDocker/myApp4# ls
c-treeRTG_v3.0.2 Dockerfile
root@ubuntu:/home/valerio/myDocker/myApp4# docker build -t ctree
```

Iniciar o container c-treeRTG normalmente adiciona-o à rede bridge:

```
docker run –name ctreeserver -d -v /isapplication/data:/isapplication/data -p 5597:5597 ctree
```

`/isapplication/data:/isapplication/data` monta o diretório do host `/ispplication/data` em `/isapplication/data` no container. Para encontrar os endereços IP de um contêiner, observe a saída do comando docker inspect:

```
root@ubuntu:/home/valerio/myDocker/myApp4# docker inspect ctreeserver | grep IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.2",
                    "IPAddress": "172.17.0.2",
```

Agora podemos verificar se a comunicação TCP/IP rodando um container 'iscobolserver' permitirá a execução do exemplo de benchmark IO_INDEXED. O arquivo Docker usado para criar a imagem iscobol para executar o IO_INDEXED conterá:

```
FROM openjdk:11
MAINTAINER Veryant
ENV ISCOBOL=/var/isCOBOL2022R2
ENV ISCOBOL_CLASSPATH=${ISCOBOL}:${ISCOBOL}/io-performance
RUN mkdir ${ISCOBOL}
COPY iscobol.properties ${ISCOBOL}/iscobol.properties
COPY isCOBOL_SDK2022R2/lib ${ISCOBOL}/lib
COPY isCOBOL_SDK2022R2/native/lib ${ISCOBOL}/native/lib
COPY isCOBOL_SDK2022R2/bin/isserver ${ISCOBOL}/bin/isserver
COPY isCOBOL_SDK2022R2/sample/io-performance ${ISCOBOL}/io-performance
WORKDIR "${ISCOBOL}/io-performance"
CMD ${ISCOBOL}/bin/isserver run
```

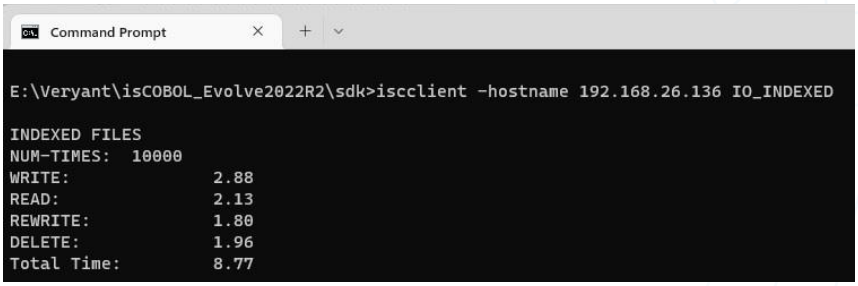
O iscobolserver é iniciado com este comando:

```
docker run –name iscobolserver -e ISSERVER_OPTS='-c /isapplication/data/runtime.properties'-
dit -v /isapplication/data:/isapplication/data -p 10999:10999 iscobol
```

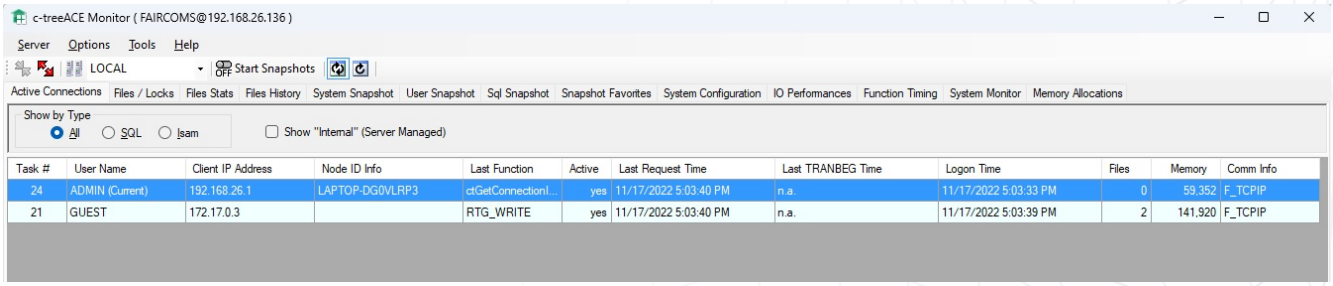
`-e <env_variable>` é usado para definir uma variável de ambiente para o deamon 'isserver' iniciado pelo container. O `/isapplication/data/runtime.properties` contém as propriedades de configuração adequadas para funcionar e se comunicar adequadamente com o servidor c-treeRTG:

```
iscobol.file.index=ctreej
iscobol.file.index.server=FAIRCOMS@172.17.0.2
```

Finalmente executando o IO_INDEXED através de um cliente isCOBOL vemos a execução trabalhando com TCP/IP:



Durante a execução do isCOBOL Client, também podemos ver no utilitário c-treeACE Monitor quais clientes estão conectados ao c-treeRTG Server:



O cliente c-tree conectado com o endereço IP igual a 172.17.0.3 corresponde ao endereço atribuído ao container ‘iscobolserver’ pela rede Docker bridge:

```
root@ubuntu:/home/valerio/myDocker/myApp3# docker inspect iscobolserver |grep IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.3",
                    "IPAddress": "172.17.0.3",
```

A segunda opção, a ponte definida pelo usuário, permite que você tenha um pouco mais de controle.

Obtenha mais controle, com uma ponte definida pelo usuário

Para permitir que os contêineres do Docker se comuniquem por nome, você pode criar uma ponte definida pelo usuário. Em uma rede de ponte definida pelo usuário, você pode ser mais explícito sobre quem se junta à rede e obtém um bônus adicional:

...contêineres podem ser endereçados por seu nome ou alias.

Criar uma rede de ponte definida pelo usuário
Crie sua própria rede de ponte personalizada primeiro usando o comando docker network create. Sob o capô, o Docker configura as tabelas de rede relevantes em seu sistema operacional.

```
docker create network veryant
```

Conecte o contêiner ctreeserver à sua ponte definida pelo usuário
Inicie o contêiner ctreeserver e conecte-o à ponte usando a opção —net.

```
docker run –net veryant –name ctreeserver -d ...
```

Comunicação entre o isCOBOL Server e o c-treeRTG em execução em diferentes containers do Docker

Conecte o contêiner iscobolserver à sua ponte definida pelo usuário
Inicie o contêiner iscobolserver usando ctreeserver como o nome do host.
Quando dois contêineres são unidos à mesma rede de ponte definida pelo usuário, um contêiner é capaz de endereçar outro usando seu nome (como o nome do host)

```
docker run –net veryant –name iscobolserver -e ISSERVER_OPTS='-c /isapplication/data/
runtime.properties ...
```

Conecte-se através da memória compartilhada

O próximo artigo do Docker mostrará como configurar a comunicação entre os contêineres do Docker por meio da memória compartilhada. Fique atento!

Quando os programadores de computador foram treinados no trabalho, a maioria dos programadores de computador eram mulheres como Kathleen Booth. Então os computadores ficaram pequenos e acessíveis o suficiente para dar de presente às crianças, permitindo que aprendessem a codificar na infância. E os pais quase sempre davam esses presentes aos meninos. Quando chegou a hora de escolher uma carreira, os homens tinham a experiência com computadores que as mulheres não tinham, então os homens se tornaram a maioria dos programadores. Comemorando Kathleen Booth!

THEREGISTER.COM
RIP: Kathleen Booth, the inventor of assembly language
Builder and programmer of the ARC and SEC turned 100 this year

Leia mais sobre aqui:: RIP: Kathleen Booth, the inventor of assembly language • The Register

Ainda devemos usar COBOL?

Essa foi a pergunta feita e respondida por Tom Ross em seu blog na comunidade da IBM. . Ele diz que o COBOL foi declarado morto pela primeira vez em 1969, quando o PL/1 foi introduzido, e novamente várias vezes depois disso. Mas como os pontos fortes do COBOL não foram bem replicados em outras linguagens, o COBOL continua a dominar bancos, seguradoras, manufatura em larga escala e aplicativos governamentais..

O COBOL se destaca na análise de números e no processamento de dados do cliente, com seu manuseio especializado de strings de caracteres, registros de comprimento variável e conversão de caracteres em dados numéricos.

O COBOL é melhor em autodocumentação com uma sintaxe semelhante ao inglês que é altamente legível. É prolixo, mas Ross conta isso como uma vantagem, tornando difícil escrever um código misterioso (e, portanto, menos sustentável).

COBOL é fácil de aprender sem nenhum treinamento especial, então não há necessidade de mudar para outro idioma só porque seus programadores não conhecem COBOL. Ross sugere ficar com o idioma que você tem – seja Java ou COBOL.

Isso coloca o isCOBOL em uma posição única, com nossa estreita conexão Java e COBOL. Você não precisa jogar fora seus programas COBOL; com Veryant você pode transformá-los e integrá-los com novas e modernas linguagens e interfaces.with new and modern languages and interfaces.

Onde os wrappers isCOBOL procuram por Java?

O compilador, iscc.exe, usa o valor de ISCOBOL_JDK_ROOT e todos os outros wrappers usam a variável ISCOBOL_JRE_ROOT.

No Windows, essas variáveis são definidas em dois arquivos no Windows: “pref_jre.cfg” na pasta .install4j e isshell.bat (chamado por cobcmd.bat) na pasta bin.

Onde os wrappers isCOBOL procuram por Java?

Eles são lidos nessa ordem, com a última configuração tendo precedência. Ambos são gerados durante a instalação.

No Linux, Unix e MacOSX, essas variáveis são definidas em um arquivo chamado default_java.conf na pasta bin. Este arquivo é gerado durante a instalação.

Se ISCOBOL_JDK_ROOT ou ISCOBOL_JRE_ROOT não estiver definido, os wrappers procurarão a instalação Java padrão. No Windows, isso é retornado pelo comando “where java”. No Linux, Unix e Mac OSX, o Java padrão é retornado pelo comando “which java”.

VOCÊ JÁ VIU ISSO?

NOVOS VÍDEOS NO YOUTUBE

All About Configuration Files

Configuring the isCOBOL Application Server

Handling Multiple Monitors in your program

Using ISMIGRATE to convert your files easily

New Features in 2022R2

Veryant End of Year Campaign

Introducing Veryant and isCOBOL Product lines

Code Analysis Reports (CAR) for Prospects

NOVOS ARTIGOS DA BASE DE CONHECIMENTO (KB):

Guide to update the isCOBOL version in your SDK

How to manage a large COBOL OCCURS on a Database with EasyDB

Guide to updaing the isCOBOL software version in production environment

Working with remote projects

Guide to updateing isCOBOL runtime libraries in Tomcat WebApplications

How to Access files on a different server

How to create, write, and read to files without a program

How to adjust program screens to different screen resolutions

Evolution, without revolution

Contact Us

Para clientes com suporte, envie um e-mail para **support@veryant.com**

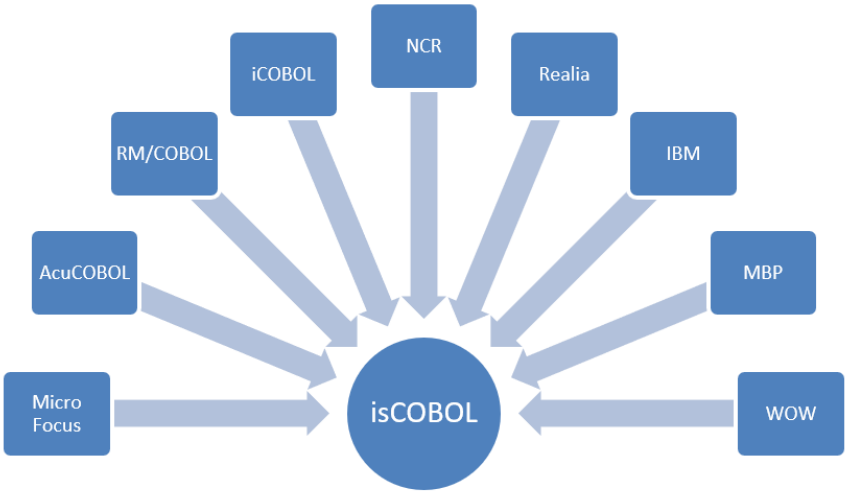
Se você deseja que a Veryant entre em contato com você para agendar um briefing técnico do produto, envie um email para **info@veryant.com**

Se você deseja que a Veryant entre em contato com você para obter uma cotação especial ou assistentes de vendas, envie-nos um e-mail para **sales@veryant.**

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 sempre, o 2022R2 contém várias adições de compatibilidade – enquanto continuamos a tornar seu processo de conversão o mais suave, rápido e descomplicado possível.



veryant.com

Follow **Veryant** on

veryant.com