

**NEWS
SEMESTRAL
VERYANT E
isCOBOL**

veryant

NOTÍCIAS**ESSA EDIÇÃO**

1. 2023 R2 – Ajude a espalhar a notícia e ganhe US\$ 50,00
2. Validação automática dos dados dos campos de entrada | Upload e download de arquivos no Webclient | Você viu isso?
3. Validação automática dos dados dos campos de entrada | Upload e download de arquivos no Webclient | Você viu isso?
4. Modernize seu controle de abas
5. O caminho para a verdadeira interoperabilidade
6. Destaque da documentação.

A versão 2023R2 chegou

A Publicação 2 de 2023 do isCOBOL foi lançada em 6 de setembro de 2023.

JUNTE-SE A NÓS

LinkedIn ou Facebook
para se atualizar



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

**CONTE PARA NÓS:**

Queremos compartilhar a notícia sobre o quão importante é a parceria cliente e a Veryant. E achamos que você pode nos ajudar.

Se você conhece outra empresa “ISV ou usuário final” executando aplicativos em COBOL, envie-nos suas informações de contato e enviaremos a você e ao cliente indicado um cartão de crédito Visa pré-pago de US\$ 50,00. Simples assim!



A versão 2023R2 do isCOBOL foi lançada em 6 de setembro de 2023 e inclui:

- Suporte para drag-and-drop para controles GUI, permitindo que os desenvolvedores imaginem novas maneiras de interagir com seus aplicativos.
- Novo recurso que permite executar ou depurar programas usando versões específicas do JRE para cada projeto.
- O novo recurso TurboRun melhora o desempenho de programas batch COBOL que são executados sequencialmente, executando-os na Java Virtual Machine do servidor de aplicativos, economizando o custo de inicialização de uma JVM para cada aplicativo.



A versão 2023R2 chegou



Um mais novo isCOBOL Evolve inclui novos recursos interessantes para controles GUI, a IDE permite selecionar uma versão do Java específica para cada projeto e o TurboRun acelera a execução em programas batch.

Melhorias no controle de GUI

A 2023R2 do isCOBOL Evolve introduz suporte para drag-and-drop e para controles GUI permitindo que os desenvolvedores imaginem novas maneiras de interagir com seus aplicativos.

O grid agora oferece suporte ao carregamento sob demanda de seu conteúdo, proporcionando tempos de carregamento mais rápidos e uma experiência de usuário mais tranquila.

Os filtros foram aprimorados e agora você tem a opção de usar campos de entrada ou caixas de seleção para filtrar dados de forma mais eficaz.

Seleção de versão Java

A versão 2023R2 do isCOBOL IDE inclui a capacidade de especificar a versão Java no nível do projeto.

Os aplicativos podem ter que ser executados usando versões Java diferentes por diversos motivos; por exemplo, ao usar Programação Orientada a Objetos com dependências de classes compiladas com versões específicas do JDK. Agora você pode escolher a versão do JRE para cada projeto na lista de JREs instalados.

Recurso TurboRun

O novo recurso TurboRun melhora o desempenho de programas batch COBOL que são executados sequencialmente, executando-os na Java Virtual Machine do servidor de aplicativos. Isso economiza o custo de inicialização de uma JVM para cada aplicativo. A utilização da JVM do Servidor de Aplicativo aumenta ainda mais o desempenho, pois utiliza o cache e as otimizações JIT (Just-In-Time) que já estão em execução.

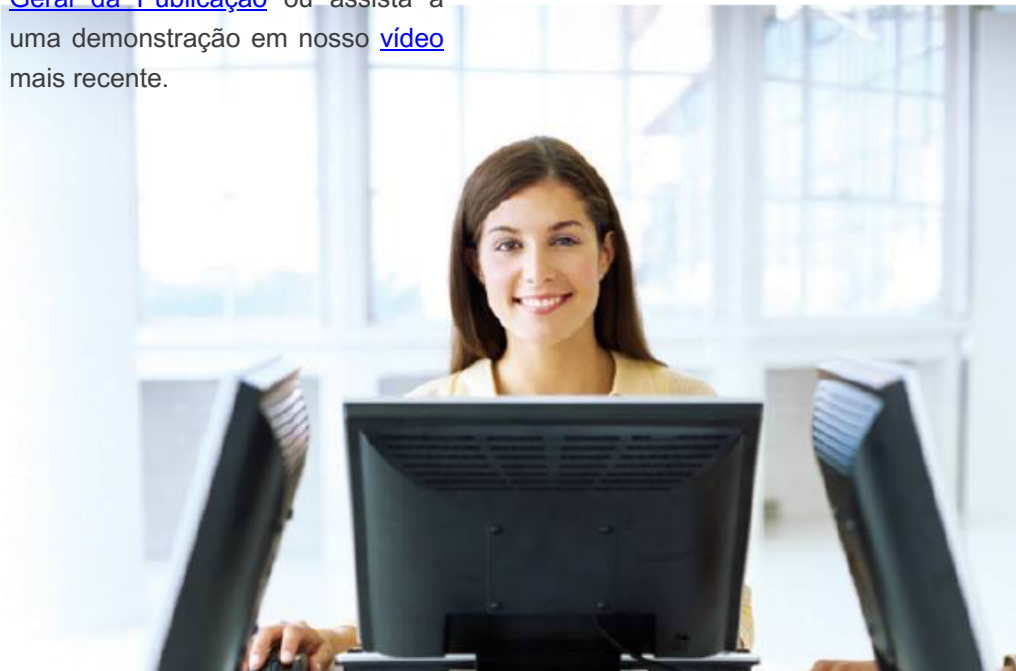
Obtenha mais informações sobre essas e outras mudanças na [Visão Geral da Publicação](#) ou assista a uma demonstração em nosso [vídeo](#) mais recente.

O DBMAKER 5.4.5 FOI LANÇADO

O DBMaker 5.4.5 foi lançado e inclui um driver JDBC tipo 4. Chega de dependências nativas para seu aplicativo Java-JDBC.

O novo driver pode ser usado em todas as plataformas onde o Java pode ser executado, mesmo se a portabilidade do cliente DBMaker não estiver disponível para essa plataforma.

O DBMaker 5.4.5 está disponível para download [aqui](#).



Validação automática de dados de entry field usando propriedades de controle

Você validaria uma entrada em um campo de entrada adicionando algum código no procedimento AFTER Procedure do Entry Field. Muitas vezes é necessário acessar dados para verificar se há um registro. Mas há outros casos em que é necessária uma validação mais simples, sem necessidade de acesso aos dados. Nestes casos é mais fácil usar as propriedades do entry field. Aqui estão dois exemplos:

- Um código que possui um formato como 1 letra alfabética e 1 valor numérico pode ser verificado automaticamente usando estas propriedades:

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

neste caso a mensagem de erro mostrada quando a validação falha é definida na propriedade "validation-errmsg". Embora este seja um exemplo simples, com sintaxes de expressões regulares mais complexas você pode verificar praticamente qualquer situação.

- Um valor numérico que aceita valores entre 9 e 100 pode ser verificado automaticamente usando estas propriedades:

```
min-val "9" max-val "10"
```

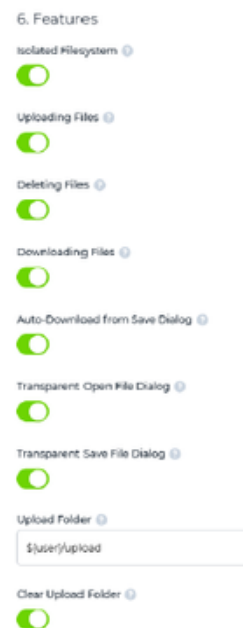
Neste caso a mensagem de erro mostrada é "Por favor insira um valor entre 9 e 100" no idioma localizado.

Para obter mais informações e mais exemplos, confira o artigo da base de conhecimento [aqui](#).

Upload e download de arquivos no ambiente Webclient

Para transferir arquivos entre o PC cliente do usuário final e o servidor em um ambiente WebClient, você pode usar C\$OPENSABEBOX para fazer upload automaticamente do PC onde o navegador é executado para o servidor WebClient, e C\$DESKTOP com o código operacional CDESKTOP-OPEN para baixar automaticamente um arquivo para esse PC.

Você também precisará habilitar as propriedades "Upload de arquivos" e "Download de arquivos" na configuração da aplicação WebServer. Outras opções de configuração do WebClient a serem consideradas são:



- Aprimore a segurança com "Sistema de arquivos isolado" e "Pasta de upload", definindo um diretório controlado no servidor WebClient para uploads.
- Agilize o processo de upload/download ativando "Download automático da caixa de diálogo Salvar" e "Diálogo transparente de abertura de arquivo".
- Use "Limpar pasta de upload" para manter a pasta designada livre de arquivos transferidos ao encerrar seu aplicativo.

Você viu isso?

Novos vídeos do YouTube:
[Novos recursos da 2023 R2](#)

[Depurador Integrado isCOBOL](#)

[Conversão COBOL-IT to isCOBOL](#)

[Depuração remota isCOBOL](#)

[Iniciando o Servidor de Aplicativos como um Serviço no Linux](#)

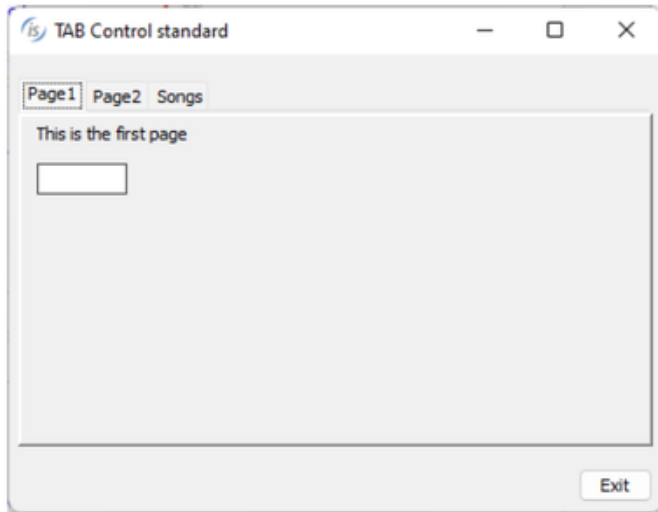
Novos artigos da base de conhecimento:
[Como visualizar dois consoles diferentes no IDE?](#)

[Como configurar switches com isCOBOL?](#)

[Como encontrar a configuração menor para instalar apenas o cliente ODBC do c-treeRTG?](#)

Modernize seu controle de abas

Seu controle de abas parece desatualizado? As abas são um dos controles que mudam frequentemente de aparência e comportamento. A captura de tela padrão do controle de abas abaixo mostra a aparência antiga. Por exemplo, observe a aparência 3D das guias que os separam da página. Além disso, as abas não têm cor e a linha pontilhada ao redor do número da página da aba parece muito desatualizada.



Outro problema com esse estilo antigo de controle de abas são as múltiplas etapas necessárias para mudar de uma aba para outra. Cada mudança exige que você:

- Intercepte o evento "CMD-TABCHANGED".
- Torne a seção da tela da página antiga invisível
- Torne a seção da tela da nova página visível

Para simplificar essas etapas de codificação, você pode adicionar o estilo "allow-container" [permitir container]. A aparência é muito semelhante, mas gerenciar as alterações nas abas é muito mais fácil. As três etapas exigidas acima são realizadas automaticamente. Você verá uma melhoria de desempenho também, especialmente em máquinas mais antigas. Para modernizar a aparência, você pode adicionar o estilo "tab-flat" [aba plana]. Essa aparência mais atualizada também abre outras propriedades que podem melhorar a aparência do seu controle de guias, adicionando cor ao plano de fundo e ao primeiro plano e às bordas da guia, além de determinar diferentes larguras de suas guias.

Esta tela é definida com estas propriedades:

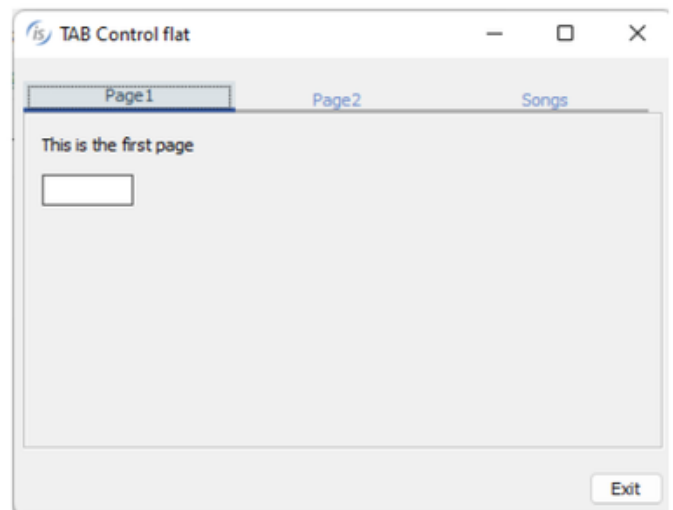
```
03 Tb1-container
  tab-control
  line                2
  col                 2
  lines              17 cells
  size               68 cells

  allow-container
  tab-flat

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

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

  tab-widths            (22 22 22)
```



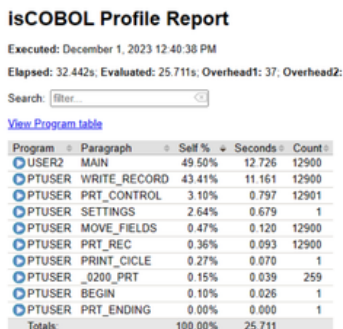
Todas essas propriedades podem ser adicionadas aos seus programas existentes usando o recurso de injeção de código, adicionando a seguinte entrada ao arquivo de configuração usado durante a compilação.

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


Uma pequena ferramenta isCOBOL oferece uma grande ajuda.

Por que você deve usar o isCOBOL Profiler? Usar o isCOBOL Profiler para fazer uma análise de desempenho de seus programas COBOL pode informar quais parágrafos em seu programa COBOL estão usando mais tempo de CPU. Incluímos o criador de perfil na estrutura de tempo de execução na versão 2015R1, mas na 2020R2 facilitamos seu uso – basta adicionar “-profile” à sua linha de comando. Os relatórios são colocados em uma pasta chamada “hprofHtmlReport”.

Recentemente, usamos o profiler para ajudar um cliente a reduzir pela metade o tempo de um programa de impressão em lote. O arquivo de impressão precisava de dados de dois arquivos c-treeRTG; o primeiro acessado internamente e o segundo em um programa chamado com call passando as informações do registro requeridas. Criamos arquivos com dados de teste e executamos os programas (PTUSER chama USER2) simplesmente com iscrun, adicionando “-profile” à linha de comando. Aqui estão os resultados:



isCOBOL Profile Report
Executed: December 1, 2023 12:40:38 PM
Elapsed: 32.442s, Evaluated: 25.711s, Overhead1: 37, Overhead2:

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

Observe que o relatório mostra a % do tempo total e os segundos específicos para cada parágrafo. A maior parte do tempo foi gasta no segundo programa, USER2. Isso nos surpreendeu, porque o programa USER2 simplesmente lê um registro de arquivo menor e o retorna em ligação com o programa principal. Aqui está o código em USER2:

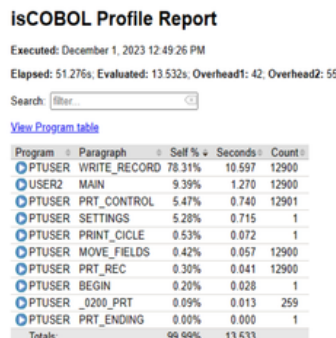
```
open input user2-file.  
  
move lnk-user2-rec to user2-key.  
read user2-file invalid  
initialize user2-details.  
move user2-rec to lnk-user2-key.  
  
close user2-file  
goback.
```

Percebemos que todo o tempo estava sendo gasto abrindo e fechando o arquivo repetidamente. Felizmente, esse cliente estava usando c-treeRTG, que possui uma opção chamada <filepool> que ajudaria. Com o agrupamento de arquivos, mesmo que o programa COBOL feche o arquivo, o c-tree o mantém aberto (agrupado) para que a próxima abertura possa obter um identificador para o arquivo imediatamente.

Adicionamos estas opções à configuração do cliente:

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

e executamos o criador de perfil novamente. Aqui estão os novos resultados:



isCOBOL Profile Report
Executed: December 1, 2023 12:49:26 PM
Elapsed: 51.276s, Evaluated: 13.532s, Overhead1: 42, Overhead2: 55

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

Observe que o tempo do programa de teste diminuiu de 25,7 segundos para 13,5 segundos e a maior parte do tempo aconteceu onde deveria, no parágrafo que monta e escreve as linhas de impressão.

Você pode executar o criador de perfil desta forma, com iscrun, ou enquanto estiver no Application Server, ou mesmo em outros cenários (Java chamando isCOBOL, EIS no Apache Tomcat, etc.)

Este teste foi todo processado em lote, o que significa que nenhuma interface de usuário foi necessária. Ao criar o perfil de um programa com a entrada do usuário, isso incluirá o tempo que o usuário levou para clicar nas coisas, o que não ajudará a avaliar o desempenho do programa.

A solução para isso é usar C\$PROFILER para desabilitar e habilitar o criador de perfil em torno das instruções ACCEPT. C\$PROFILER também é útil para limitar a criação de perfil quando você suspeita que o problema esteja em um conjunto específico de códigos ou programas.

O criador de perfil também pode ser acessado a partir do IDE, com uma visualização específica do Profiler para aplicar o criador de perfil a uma, algumas ou todas as classes em um projeto. Seu problema de desempenho provavelmente não será resolvido da mesma forma que o deste cliente, mas há muitas dicas de ajuste de desempenho em nossa documentação e estamos sempre disponíveis em support@veryant.com para ajudá-lo a identificar e resolver quaisquer dificuldades. [Aqui está um link](#) para a documentação do criador de perfil na P2 de 2023 .

DESTAQUE DA DOCUMENTAÇÃO



Pode ser necessário iniciar mais de um servidor isCOBOL ou mais de um servidor c-tree na mesma máquina.

Por exemplo, você pode usar um segunda sessão para depuração, teste ou para distribuir melhor a carga de trabalho.

Qual é a maneira correta de iniciar vários processos de servidor? Você encontrará instruções para o servidor isCOBOL em

[Iniciando vários serviços na mesma máquina \(Windows\)](#)

e

[Iniciando vários daemons na mesma máquina \(Unix\)](#)

Você encontrará instruções para o Servidor c-tree em

[Iniciando vários serviços na mesma máquina \(Windows\)](#)

e

[Iniciando vários daemons na mesma máquina \(Unix\)](#)



O caminho para a verdadeira interoperabilidade

Webservices tornam possível a verdadeira interoperabilidade. Abre um mundo de possibilidades para sua aplicação, permite manter o código que você já possui e expandi-lo sem ficar preso a uma linguagem de programação



Seu negócio é administrado com programas escritos em diferentes linguagens (COBOL, C++, Java, etc.)?

Se sim, você provavelmente teve dificuldade em fazer com que esses sistemas interajam.

Existem muitos motivos para buscar a interoperabilidade. A interoperabilidade significa que todas as partes interessadas do negócio têm acesso às mesmas informações, os dados duplicados são minimizados, a qualidade dos dados é melhorada, a gestão cliente-fornecedor é expandida, os protocolos e padrões de segurança são consistentes para todos os seus dados e os esforços de integração são minimizados.

Felizmente, a interoperabilidade entre sistemas díspares é quase sempre limitada à partilha de dados. Nas décadas de 1980 e 1990 o compartilhamento de dados era feito com transferências de arquivos; salvar e enviar arquivos inteiros de dados, formatados programaticamente para o programa receptor. Então os bancos de dados relacionais tornaram-se dominantes e muitas empresas transferiram seus dados para bancos de dados relacionais. Isso significou maior colaboração entre equipes, departamentos e fornecedores, valor agregado aos ISVs com nomes de marcas de banco de dados reconhecíveis e fácil geração de relatórios e coleta de inteligência de negócios. Para os clientes isCOBOL, isso também significou manter seus programas COBOL inalterados, adicionando o produto DatabaseBridge ao seu sistema de desenvolvimento.

Entretanto, às vezes manter os dados em um banco de dados relacional não é a melhor solução.

É um formato complexo que requer gerenciamento de banco de dados, muitas vezes significa custos adicionais para sua carga de TI e há preocupações de segurança se você quiser compartilhar dados com fontes fora de sua rede. Assim, uma nova alternativa aos bancos de dados está se tornando popular: Web Services

A maioria das linguagens de software entende tags XML ou JSON. Webservices usam esses formatos para trocar dados, geralmente com APIs usando estruturas REST ou SOAP. Usar Web Services significa que aplicativos escritos em diferentes linguagens de programação e executados em diferentes plataformas podem trocar dados sem problemas através de intranets ou da Internet. É uma maneira padronizada e eficiente de conectar seus sistemas, bem como de expô-los dentro e fora da sua rede

Na Veryant estamos focados em facilitar o acompanhamento da tecnologia e, para Webservices, incluímos as ferramentas Service Bridge e Stream2Wrk em nosso sistema de desenvolvimento EIS. O ServiceBridge gera um programa bridge que converterá a seção de ligação do seu programa COBOL em um fluxo XML ou JSON para dados de entrada e saída. O resultado é um programa COBOL que pode ser usado como Serviço Web. Stream2Wrk gera as definições de armazenamento de trabalho COBOL para dados XML, XDS, WSDL e JSON recebidos, para que seu programa COBOL possa consumir um Serviço Web existente.



Fale conosco

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

Se desejar que a Veryant entre em contato com você para agendar um briefing técnico do produto, envie um e-mail para info@veryant.com

Se desejar que a Veryant entre em contato com você para obter um orçamento especial ou assistentes de vendas, envie um e-mail para sales@veryant.com

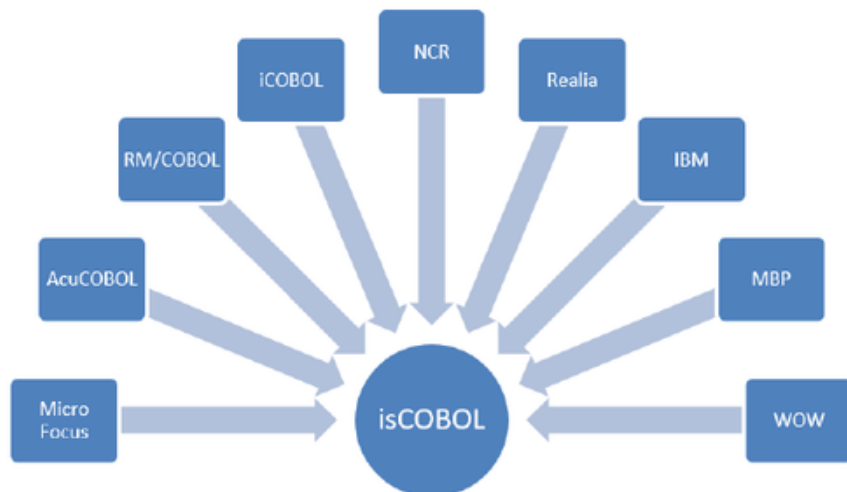
Sedes da empresa

6390 Greenwich Dr., Suite 225
San Diego, CA 92122 – EUA
Fone (Inglês): +1 619 797 1323
Fone (Espanhol): +1 619 453 0914

Sedes Europeias

Via Pirandello, 29 29121 -
Piacenza – Itália
Fone: +39 0523 490770
Fax: +39 0523 480784
emea@veryant.com

Como sempre, a versão 2023R2 contém vários acréscimos de compatibilidade, à medida que continuamos a tornar seu processo de conversão o mais suave, rápido e simples possível.



veryant.com



veryant.com

©2023 Veryant - Todos os direitos reservados