EMISSÃO
12
2024

DIÁRIO TRIMESTRAL DA VERYANT E isCOBOL

veryant

NOVIDADE

ESTA EDIÇÃO

2024 R2 e Boas Festas

O lançamento do 2024R2 com seu novo controle de Chips box Control tem sido muito popular entre nossos clientes.

Lançamos nosso novo site de suporte ao cliente. É mais intuitivo de usar e tem uma interface mais amigável.

Se você usou nosso site de suporte recentemente, sua conta foi movida para o novo site.
Suas informações de login agora são seu e-mail, com a mesma senha que você sempre usou.

Se você tiver problemas ou quiser acessar, entre em contato com seu gerente de conta em sales@veryant.com ou envie um e-mail para support@veryant.com

Após o recente lançamento do isCOBOL Evolve 2024R2, avançamos diretamente para o próximo 2025R1. Falaremos mais sobre o que está por vir no próximo boletim informativo. Aqui na Veryant estamos entrando no período de férias com desejos calorosos de sucesso e felicidade contínuos aos nossos clientes.

Esta edição do boletim informativo contém a parte 2 do tutorial de depuração de Valerio, informações sobre o uso de JWTs (JSON Web Token) para segurança avançada e o início de uma discussão sobre IA e COBOL.

Esperamos que você tenha uma ótima temporada de férias e um ano novo saudável e próspero!

Você está executando uma versão mais antiga do isCOBOL? Estamos aqui para ajudá-lo a atualizar para ficar com as versões suportadas e aproveitar as novas tecnologias. Consulte a nossa política de fim de vida útil aqui.

Como a inteligência artificial pode beneficiar nossos aplicativos COBOL?

A discussão em torno do papel da IA no futuro do COBOL ressalta tanto seu potencial quanto suas limitações. A IA nunca poderá substituir completamente os desenvolvedores do COBOL, mas provavelmente evoluirá para ser um assistente crítico, reduzindo sua carga de trabalho e permitindo que eles se concentrem em tarefas de alto valor.

A IA está pronta para substituir os desenvolvedores do COBOL?

Ferramentas de IA como o WatsonX da IBM e outros modelos generativos (por exemplo, ChatGPT, GitHub Copilot) são promissoras, mas ainda não estão totalmente equipadas para substituir o desenvolvimento do COBOL. Programadores experientes em COBOL relatam classificação de um código gerado por IA incompleto ou impreciso em discussões de mídia social.

A IA não pode replicar o profundo conhecimento institucional e a criatividade dos desenvolvedores humanos. No entanto, pode ajudar a identificar áreas de interesse dentro dos sistemas legados e fornecer recomendações e insights para agilizar o processo de desenvolvimento.

Que papéis a IA pode desempenhar atualmente?

A força da IA reside no aumento da produtividade do desenvolvedor por meio de funções de apoio:

- **Tradução de Código:** Ferramentas como WatsonX alcançam automação parcial de Traduções de COBOL para Java, lidando com 75-80% do trabalho, mas ainda dependem da experiência humana para refinar os resultados. Embora essas ferramentas estejam focadas no COBOL empresarial para mainframe, a Veryant oferecerá em breve uma conversão de 100% do COBOL para Java para ambientes de sistema aberto.
- **Documentação e compreensão:** a IA pode gerar automaticamente documentação, mapear dependências e conduzir análises de impacto, que são cruciais para a manutenção de sistemas COBOL grandes e opacos.
- **Correção e depuração de bugs:** os desenvolvedores gastam cerca de três quartos de seu tempo localizando o código que contém um bug ou a localização do código adicional afetado por uma alteração em um local. Os sistemas avançados de IA podem identificar seções de código que exigem atualizações ou correções, minimizando o tempo gasto na pesquisa em vastas bases de código.
- **Redução da dívida técnica:** ferramentas de IA estão sendo empregadas para identificar códigos ineficientes ou redundantes, ajudando empresas como a Wayfair a gerenciar dívidas técnicas.

IA simbólica vs. IA generativa

Para fazer tudo isso, precisamos de mais do que uma IA que gera código, chamada de Generative AI. Um novo tipo de IA chamado Symbolic AI está sendo usado para dar uma abordagem mais humana e de senso comum à IA. Aqui está uma descrição das diferenças entre esses dois tipos de IA:

- **Generative AI (e.g., ChatGPT):** concentra-se na criação de conteúdo (código, comentários, etc.). É útil para prototipagem, mas carece de recursos de raciocínio para lidar com sistemas complexos com dependências interconectadas.
- **Symbolic AI (e.g., PhaseChange's COBOL Colleague):** Projetado para raciocinar e entender as relações de causa e efeito no código. A IA simbólica pode ajudar a conceituar o que os sistemas legados estão fazendo e automatizar insights no nível do sistema que permitem que os desenvolvedores façam alterações informadas sem se aprofundar em toda a base de código.

Essa divisão sugere que a combinação de ambas as abordagens poderia produzir os resultados mais produtivos.

Previsões de Longo Prazo

Até 2028, espera-se que a colaboração entre IA e humanos reduza os tempos de desenvolvimento em 30%, de acordo com Gartner. Isso poderia reduzir a barreira à manutenção de sistemas COBOL legados, estender o ciclo de vida desses sistemas enquanto as empresas se modernizam de forma incremental e permitir que os desenvolvedores se concentrem na inovação em vez da manutenção.

Conclusão

A IA, seja generativa ou simbólica, não é uma bala de prata para a manutenção e desenvolvimento de aplicativos COBOL. Em vez disso, é um poderoso aliado que pode agilizar os fluxos de trabalho, reduzir a dívida técnica e aumentar a produtividade dos desenvolvedores. O caminho a seguir está em uma abordagem híbrida - alavancando a experiência humana e as capacidades da IA para garantir que os sistemas legados continuem a operar de forma eficaz enquanto são modernizados quando necessário.

10.000,00 OU 10,000.00? PONTOS DECIMAIS PARA VÁRIOS PAÍSES

Quando você escreve um programa COBOL para ser executado na Europa, você definiria o programa para usar uma vírgula como separador decimal adicionando *DECIMAL-POINT is comma* à seção de nomes especiais.

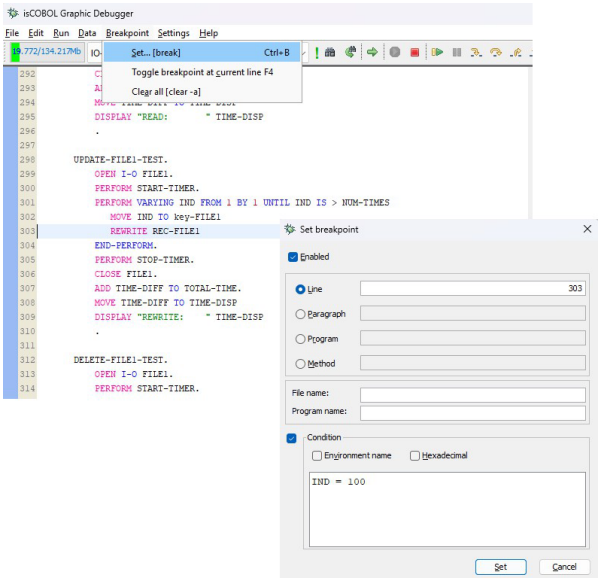
Mas se você precisar executar o mesmo programa nos EUA e na Europa, precisará acomodar os dois tipos de separadores decimais. Isso é fácil no isCOBOL sem alterações no programa. Siga estes passos:

1. Compilar com a opção *-sddp*
2. Ao executar na Europa, altere o comportamento do ponto decimal em tempo de execução definindo a propriedade de configuração *iscobol.runtime.decimal_point_is_comma* para *true*

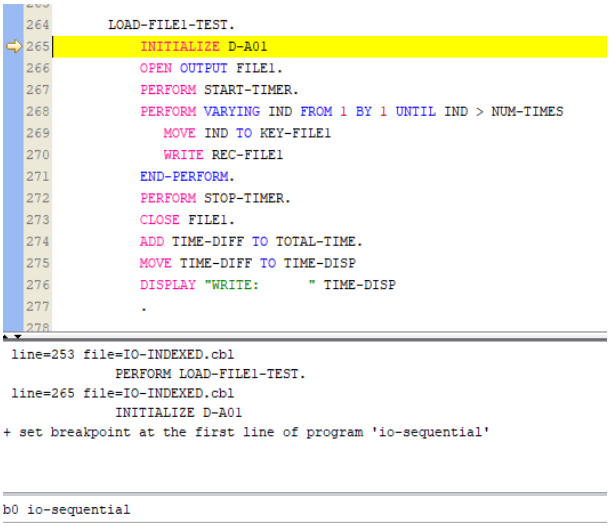
Trabalhando no Debugger

Uma vez que os bugs são encontrados, os codificadores podem iniciar o processo de depuração e trabalhar para livrar o software de quaisquer erros. Vejamos algumas das funções do Debugger inteligente do isCOBOL que permitem que os codificadores identifiquem bugs.

O BREAKPOINT é um ponto de paragem ou de pausa intencional num programa colocado num local para fins de depuração. Na prática, o breakpoint consiste em uma ou mais condições que determinam quando a execução de um programa deve ser interrompida. Você pode definir o breakpoint no menu, digitando Control+B no campo de comando ou clicando com o botão direito do mouse no código no Debugger. Aqui está um exemplo de como parar em uma linha específica quando uma condição foi atendida (neste caso, quando ind = 100)



Você também pode definir o breakpoint no início de um programa chamado, através da área de comando digitando **`b0 program name`**



ESCALABILIDADE PARA APLICAÇÕES COBOL

A escalabilidade garante que a infraestrutura de negócios se adapte ao crescimento, melhorando a eficiência, otimizando recursos e permitindo uma adaptação rápida. Para os sistemas COBOL, isso significa modernizar e ampliar a capacidade sem substituir o código principal.

Soluções isCOBOL da Veryant

A Veryant oferece ferramentas para tornar os aplicativos COBOL escaláveis:

- Suporte em nuvem: o isCOBOL é executado perfeitamente na nuvem ou em qualquer ambiente suportado pelo JRE, permitindo dimensionamento flexível e elástico.
- isCOBOL LoadBalancer: distribui sessões do ThinClient entre servidores para otimizar recursos e evitar gargalos.
- Cliente Web em cluster: Distribui as sessões do Cliente Web por vários pools, garantindo um acesso rápido e fiável durante os picos de utilização.

Passos para Escalar Aplicações COBOL

1. Avaliar Sistemas: Avaliar o desempenho sob várias cargas de trabalho.
2. Adote Soluções de Nuvem: Use as capacidades de nuvem do isCOBOL para flexibilidade.
3. Alavancar Balanceadores de Carga e Clustering: Melhore as interações do usuário com ferramentas como isCOBOL LoadBalancer.
4. Desempenho do Monitor: sistemas ópticos continuamente

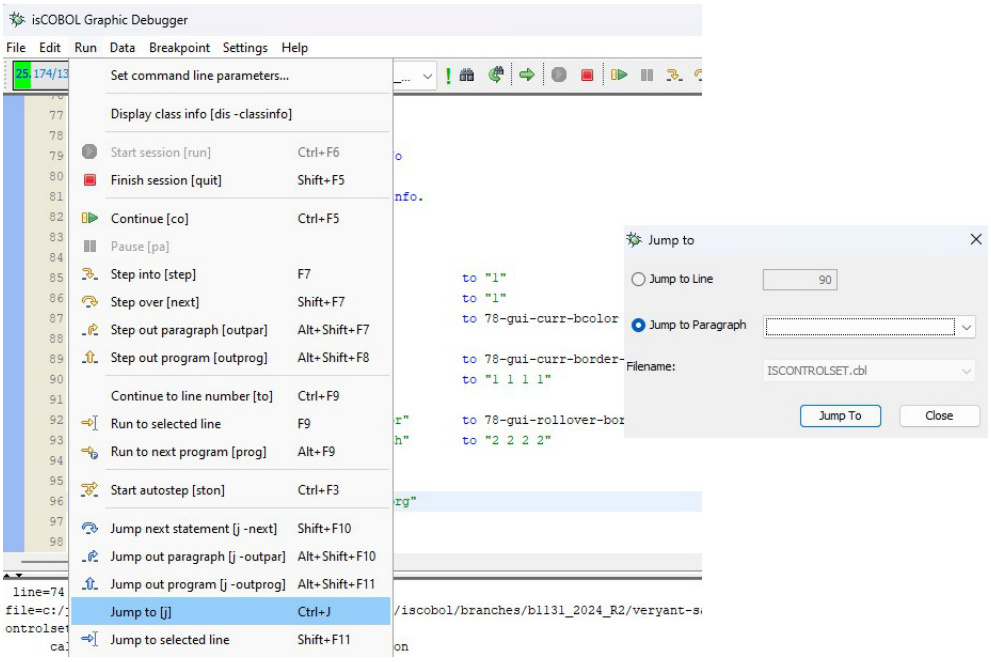
Por que escolher a Veryant?

As ferramentas da Veryant ajudam a modernizar os sistemas COBOL para escalabilidade, apoiando o crescimento sem revisões. Entre em contato com um gerente de contas da Veryant para explorar uma solução personalizada

JUNTE-SE A NÓS PELO

Twitter, LinkedIn ou Facebook para se atualizar com as notícias da Veryant

Assista aos nossos vídeos de demonstração e inscreva-se no nosso Canal do YouTube



A funcionalidade JUMP, disponível apenas em programas compilados com a opção -dx, permite que o usuário pule para uma linha específica, pulando o código entre a linha atual e a linha de destino.

Você Já Viu Isso?

Vídeo mais recente:

O Visual Studio Code ou VS Code, é um editor gratuito da Microsoft. Combinado com a extensão isCOBOL VSCode, você pode obter os benefícios como verificação de sintaxe, digitação preditiva e depuração no aplicativo em um ambiente fácil de usar. Aqui está um vídeo demonstrando como começar.

Novos artigos da KB:

Como encerrar automaticamente uma sessão de thin client que ficou ociosa por um período específico.

Um programa procedural COBOL pode ser INVOCADO com uma matriz de objetos em vez de uma CHAMADA padrão com o USO

Autorização e Assinatura da API Web Avançada

Quando você deseja implementar uma autenticação avançada de API, precisará de um programa que possa validar a autenticidade e a integridade de um JSON Web Token (JWT) assinado com uma chave pública RSA.

As JWTs são uma maneira compacta e autônoma de transmitir informações com segurança entre as partes como um objeto JSON.

Vamos considerar o seguinte código de exemplo:

```
program-id. CobJwtDecode.

 configuration section.
 repository.
     class PublicKey   as “java.security.PublicKey”
     class X509Certificate
                       as “java.security.cert.X509Certificate”
     class RSAPrivateKey
                       as “java.security.interfaces.RSAPrivateKey”
     class RSAPublicKey
                       as “java.security.interfaces.RSAPublicKey”
     class RSAKey      as “java.security.interfaces.RSAKey”
     class JWT           as “com.auth0.jwt.JWT”
     class JWTVerifier   as “com.auth0.jwt.JWTVerifier”
     class Algorithm     as “com.auth0.jwt.algorithms.Algorithm”
     class DecodedJWT    as “com.auth0.jwt.interfaces.DecodedJWT”
     class X509CertUtils as “com.nimbusds.jose.util.X509CertUtils”
 .
 working-storage section.
 77  token       pic x any length.
 77  certificate pic x any length.
 77  cert        object reference X509Certificate.
 77  pubKey      object reference PublicKey.
 77  o-publicKey object reference RSAPublicKey.
 77  o-algorithm object reference Algorithm.
 77  verifier    object reference JWTVerifier.
 77  o-jwt       object reference DecodedJWT.

 procedure division.
 main.
* TODO Auto-generated method stub
     move
     “eyJhbGciOiJSUzI1NiIsImtpZCI6IjYwZTQxMjczMzMwYTg2ZmRjMjh1Mjgz
-    “MDVhNDRkYzlhODgzZTI2YTciLCJ0eXAiOiJKV1QifQ.eyJuYW1lIjoiQnJpY
-    “W4iLCJwaWN0dXJlIjoiaHR0cHM6Ly93d3cudzNzY2hvb2xzLmNvbS9ob3d0b
-    “y9pbWdfYXZhdGFyLnBuZyIsImlzcyI6Imh0dHBzOi8vc2VjdXJ1dG9rZW4uZ
-    “29vZ2xlLmNvbS9jb2xsZWN0aW9uLXBhcnRuZXIiLCJhdWQiOiJjb2xsZWN0a
-    “W9uLXBhcnRuZXIiLCJhdXRoX3RpbWUiOjE1NjMzNzM1MjUsInVzZXJfaWQiO
-    “iJIU0tSWTdrTGxHUU9GSkwzVlUxWXhiWFJNYk0yIiwic3ViIjoiSFNLUlk3a
-    “0xsR1FPRkpMM1ZVMV14YlhSTWJNMiIsImlhdCI6MTU2MzM3MzUyNSwiZXhwI
-    “joxNTYzMzc3MTI1LCJlbWFpbCI6ImJyaWFuLnN1bGxpdmFuMUB5YWhvby5jb
-    “20iLCJlbWFpbF92ZXJpZmllZCI6dHJ1ZSwiZmlyZWJhc2UiOnsiaWRlbnRpd
-    “GllcyI6eyJlbWFpbCI6WyJicm1hbi5zdWxsaXZhbjFAeWFob28uY29tIl19L
-    “CJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0.J6j4b562mHM9U62kZ
-    “F3yMyZhf8AaL3Cum4vGPnFcHc1H9V9dH720v8OURMTCsZKhOrMfO46t8EoTC
-    “EWIJfY-NskA-L9ulzem8lwk5gcac662fam8Jz1HUJGFH4LTIVJbozsWXccIx
-    “kbIm3FrZVWKJAmAE71S_6Id4Nw5Jq5S5gEmi-DBVSIRntieYogYvkSNBmfRV
-    “NiMhR5ixZrNkFOSOQH-RiRGgzTz1VpRT8EwKxlgHmrtiriA7DRtoQJy7jIN0
-    “oGfpYf9aI1pyKMeEAMD3Hi9UXrb_VP0MgO8FP2GbWH4vTGh3MtsX49xcAX-i
-    “B7LIUu5O8Ia0Wt8TUSXqe_Q-A” to token.

     perform check-token.
     goback.
```

```
check-token.
    move
    "-----BEGIN CERTIFICATE----- MIIDHDCCAgSgAwIBAgIIITyq+eI2xDsw
-   "DQYJKoZIhvcNAQEFBQAwMTEvMC0GA1UE AxMmc2VjdXJldG9rZW4uc3lzdGV
-   "tLmdzZXJ2aWNlYWNjb3VudC5jb20wHhcNMTkw NzA3MjEyMDU1WhcNMTkwNz
-   "I0MDkzNTU1WjAxMS8wLQYDVQQDEyZzZWN1cmV0b2tl bi5zeXN0ZW0uZ3Nlc
-   "nZpY2VhY2NvdW50LmNvbTCCASIwDQYJKoZIhvcNAQEBBQAD ggEPADCCAQoC
-   "ggEBALmpZwoVX2b2/nwphVDh+Y3+F5d6EZuk3rgDoP2Vu/Lrqapx ovdxAt0
-   "atmuU+YiyU6wGHBycoMXu0nWKRdArn8VdoDa8vm3RxX2WFwe8u9yIo1Ju Pq
-   "dGOaiIdgbsNSxXofmkepEDoBNdRqdfRfxZ19H+mbOZ6gjb79gHU0n58cb1vP
-   "5C r02RYjiC/Fy7jZ+D0AqODE/e1kU76+SyeAISAKFQnzd62Zg9rIZ0zeZ7F
-   "J5U4BJ5 S/JsoLe+gUtC7ZTZHd9wNc5Ga3+KE7scYEsO18lk6UlxF+VN0R3M
-   "Z/ObJ/q3Q9dR pDQU3OB19xvPrBC5kypiXFPqabg+mHfZUxaSWzECAwEAAaM
-   "4MDYwDAYDVR0TAQH/ BAIwADAOBgNVHQ8BAf8EBAMCB4AwFgYDVR0lAQH/BA
-   "wwCgYIKwYBBQUHAwIwDQYJ KoZIhvcNAQEFBQADggEBAGiVv+cdrAuZrNV5w
-   "VPFaCvUSiGn+LGq/OC6hZWiljt/ AEn6ypoZfkRJEwTv+IrZHKcEBSR7Dees
-   "n0sj+tigL1g37w3I9I1WNEaip/snCCTe BtBm4gmkX9cDv4Ga/rhObGOZAhN
-   "NhEFWY8ofQFuohOxoaZe/Z5fvoDZW6eFKc84a q/tNDaN2Xglyiadccgux9/
-   "7OH2oH+AwoniP1IHmQAaccUsNqOjg/X4WpaIKfBJ2i 9zhevne1IUm0RMxPH
-   "oHSR8iqllu9phpeRX7Po1tRfg3YUY2fyaC4Wad9FBJAGBx7 9YSEIyd0K82o
-   "3h7Y3ITtIGOmiA8NDI9tkNOTVpBTXds= -----END CERTIFICATE-----"
    to certificate.
    Try
       set cert    to X509CertUtils:>parse(certificate)
       set  pubKey to cert:>getPublicKey()
       set o-publicKey to pubKey as RSAPublicKey
       set o-algorithm to Algorithm:>RSA256(o-publicKey
                                                as RSAKey)
       set verifier    to JWT:>require(o-algorithm)
                          :>withIssuer
            ("https://securetoken.google.com/collection-partner")
                          :>build() |build the verifier instance
       set o-jwt to verifier:>verify(token)
       | access jwt fields if needed to verify token data
       display "Token verification successful"
    catch exception
       |Invalid signature/claims
       display "Token verification failed: "
                exception-object:>getLocalizedMessage()

    end-try.
```

Como funciona

O objetivo principal do programa é verificar se uma determinada JWT é genuína e não foi adulterada.

1. O repositório do programa importa as classes necessárias para lidar com chaves RSA, certificados e operações JWT. Essas classes estão incluídas nos seguintes arquivos .jar: jackson-core-2.9.9.jar, jackson-annotations-2.9.9.jar, commons-codec-1.9.jar, nimbus-jose-jwt-6.0.jar, java-jwt-3.8.1.jar, jackson-databind-2.9.9.2.jar, json-smart-2.2.1.jar
2. O parágrafo principal define uma string JWT no item de token para que ela seja usada no próximo parágrafo. No checkToken, um certificado X.509 codificado no formato PEM é analisado para obter um objeto X509Certificate.
3. A chave pública é extraída do certificado e marcada se for uma RSAPublicKey e, em seguida, uma instância Algorithm é criada usando essa chave.
4. Uma instância do JWTVerifier é construída, especificando o emissor (https://securetoken.google.com/collection-partner) e, em seguida, o método verificador: >verify(token) é chamado para verificar o JWT. Uma confirmação ou uma mensagem de erro é exibida dependendo do sucesso ou falha disso.

Este programa fornece um mecanismo para garantir que uma JWT possa ser confiável. Ao verificar a assinatura do token e outros atributos críticos, ele ajuda a impedir o acesso não autorizado e a proteger dados confidenciais.

Evolution, without revolution

Fale Conosco

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

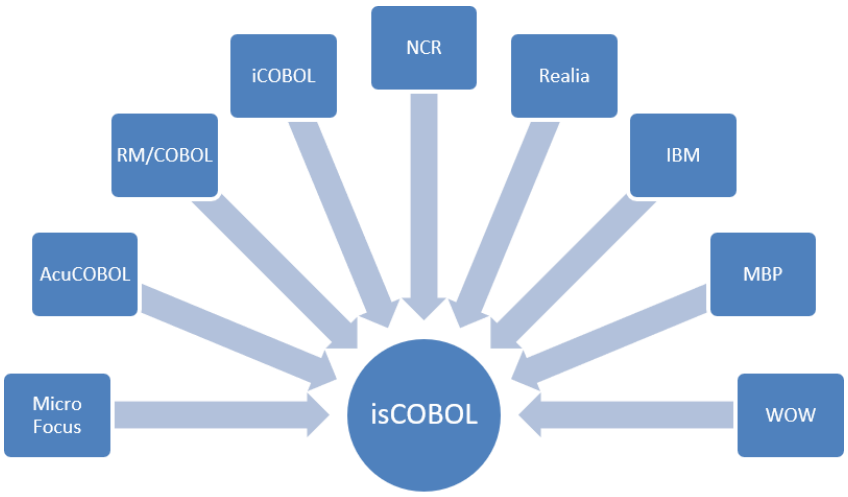
Se você quiser que a Veryant entre em contato com você para agendar um briefing técnico do produto, envie um e-mail para **info@veryant.com**

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

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

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

Como sempre, a mais nova versão do isCOBOL Evolve contém várias adições de compatibilidade - enquanto continuamos a tornar o seu processo de conversão o mais suave, rápido e sem complicações possível.



veryant.com

Siga a **Veryant** no

veryant.com