



Felicitações ao COBOL pelos 60 anos. Ele nunca esteve melhor!

ESTAMOS ANIMADOS COM AS NOVIDADES DA 2020R1.



Você pode

- inserir novos controles GUI em seu programa, como o SCROLL-PANE e o novo estilo do TREE-VIEW chamado TABLE-VIEW
- aprimorar a interface do usuário e o gerenciamento de eventos do mouse usando o estilo NOTIFY-MOUSE suportado em todos os controles
- mostrar miniaturas de página na visualização de impressão, permitindo que o usuário mude rapidamente para uma página específica em um relatório de várias páginas

Maurizio Girometti, nosso CEO, diz que a Veryant começou com a missão de "revitalizar o COBOL e transformá-lo em uma linguagem de tecnologia de ponta" e, com os aprimoramentos da 2019R2 e 2020R1, continuamos com essa missão.

veryant

NEWS

TÓPICOS

1. Novidades da versão 2020R1
2. COBOL Hoje
3. Code Completion e Unit Testing
3. Exemplos Multithreading
4. Destaques na Documentação
4. LAF: Escolhendo um design para suas telas GUI
4. Você já viu isso?
5. Destaques 2019 R2
5. Layout Manager - Zoom
5. Mensagens de erros de compilação
6. JOE script
6. Opções de Dados
7. Última página

NOVIDADES

2020R1 chegou

A Veryant tem o prazer de anunciar a versão mais recente do isCOBOL™ Evolve, isCOBOL Evolve 2020 R1. O isCOBOL Evolve fornece um ambiente completo para o desenvolvimento, implantação, manutenção e modernização de aplicações COBOL

A partir da versão 2020R1, o isCOBOL está apresentando recursos para os usuários corporativos: Code Coverage e Test Unit. Esses recursos estão disponíveis em outras linguagens, como Java, e agora também estão disponíveis para os desenvolvedores do isCOBOL. Eles ajudarão você a produzir testes de melhor qualidade para as aplicações COBOL.

Esses novos recursos também estão integrados na IDE, bem como um novo recurso de ocultar trechos de código que pode ser personalizado. Adicionamos alguns novos controles GUI, scroll-pane e novo estilo do tree-view chamado table-view, que ajudarão a manter seus aplicativos com uma aparência nova e atualizada.

Outras melhorias incluem um controle mais preciso do mouse, miniaturas de visualização do preview e, é claro, as constantes melhorias no desempenho e na compatibilidade.

Esta edição incluirá informações sobre a 2019 R2. Normalmente, novos recursos importantes são introduzidos nas versões R1, no início do ano, mas há algumas mudanças na 2019 R2 que são bastante empolgantes, como o LM-ZOOM que permite redimensionar a tela inteira com uma única variável e a possibilidade de aplicar um controle usando o arquivo de configuração em vez de alterar seu código.

COBOL HOJE



Maurizio Girometti, CEO da Veryant, fala sobre suas experiências com o COBOL e como ele se encaixa no mundo de hoje

Começou a escrever aplicações COBOL em 1982 em um minicomputador Texas TI-990. Foi uma ótima experiência depois de praticar com Assembler, Fortran e Basic. Achei o COBOL fácil, estável e, é claro, absolutamente adequado para aplicações de negócios.

Com a chegada de novas linguagens, o COBOL começou a ter uma má reputação na comunidade de TI. No começo, havia a ilusão de linguagens 4GL: em teoria, qualquer programador era capaz de escrever aplicações complexas em minutos. *Não é verdade.*

Então outras linguagens prometeram fornecer tecnologia de ponta e realmente o fizeram. Isso definitivamente contribuiu para aumentar a má reputação do COBOL, já que a maioria do mundo de TI achava que, naquele momento, o COBOL estava apenas relegado para aplicações antigas de mainframe.

Então, é verdade que o COBOL é antigo e não suporta as novas tecnologias ou isso é apenas uma "lenda urbana"?

Em 2005 decidimos iniciar a Veryant com o objetivo de revitalizar o COBOL e transformá-lo em uma linguagem de tecnologia de ponta, pois ainda há muitos softwares no mundo escritos em COBOL. Como resultado, o isCOBOL® nasceu e hoje é utilizado em todo o mundo por milhares

de desenvolvedores.

Assim, supor que o COBOL é uma linguagem antiga é uma "lenda urbana", uma vez que o isCOBOL oferece todas as funcionalidades fornecidas pelas linguagens modernas, incluindo uma interface gráfica com o usuário, serviços Web, thin client remoto, interface Web, OOP, uma poderosa IDE baseada no Eclipse e muito mais.

"Em 2005, decidimos iniciar a Veryant com o objetivo de revitalizar o COBOL e transformá-lo em uma linguagem de tecnologia de ponta."

E, tudo isso é portátil de mainframes para dispositivos móveis sem alterações, pois no final do processo de compilação o que você obtém é um objeto Java. Isso significa que, se necessário, você pode interagir com qualquer outro objeto Java, aproveitando facilmente o poder do mundo Java.

Se você quiser testar nosso produto, ligue para nossos escritórios e teremos o maior prazer em fornecer mais informações técnicas e de vendas e entregar uma cópia de avaliação de nossos produtos. Nossa equipe técnica o ajudará durante todo o período de avaliação.

If you were a farmer would you still plow the ground in this way?

veryant

Evolution without revolution



Experimente o poder do COBOL no mundo moderno!

Code Completion e Unit Testing

O novo recurso Code Coverage informará quanto do seu código está sendo utilizado durante uma execução. Isso é útil quando você está configurando um ambiente de teste e deseja ter certeza de que está testando o máximo de código possível.

Facilitamos a implementação (adicione -coverage à sua linha de comando de execução) e verifique os resultados com um "Relatório global de cobertura" usando gráficos para mostrar a cobertura nos níveis do programa e do parágrafo, além de fornecer acesso a seu código, ressaltado por cores para mostrar linha por linha o que foi executado e o que não foi.

Sabemos que nem sempre foi possível criar um ambiente de teste automatizado e adicionamos algo para ajudá-lo. O recurso Unit Test permite que os desenvolvedores criem conjuntos de testes automatizados, projetados para testar se as seções do código são executadas conforme

o esperado. O objetivo do Unit Test é isolar seções de um programa e garantir que eles estejam funcionando corretamente.

Para configurar esse teste, basta acrescentar a instrução ASSERT às partes do seu código que deseja testar, semelhante a esta:

```
assert string1 = "my string"
otherwise "Test string
manipulation: Error"
```

Quando você compila e executa seu programa com as opções corretas, o Unit Test é ativado, sem essas opções as instruções são ignoradas. O recurso Unit Test gera relatórios semelhantes à Cobertura de código e, quando executados juntos, os relatórios são combinados.

E a melhor parte? Tudo isso também está integrado na IDE. Entre em contato com o suporte ou seu representante de vendas para obter mais informações ou uma demonstração.

The screenshot shows the IDE interface with a code completion menu open over a COBOL program. Below the code editor, a coverage report table is displayed for the file 'io-performance'.

| Element | Statements Coverage | Missed Stmtms | Total Stmtms | Paragraphs Coverage | Missed Pars | Total Pars |
|-------------------|---------------------|---------------|--------------|---------------------|-------------|------------|
| io-performance | 92.34% | 17 | 222 | 96.15% | 1 | 26 |
| IO_LINESEQUENTIAL | 97.30% | 1 | 37 | 100.00% | 0 | 5 |
| IO_INDEXED | 81.82% | 14 | 77 | 87.50% | 1 | 8 |
| MAIN_LOGIC | 78.57% | 3 | 14 | | | |
| START_FILE1_TEST | 0.00% | 10 | 10 | | | |
| LOAD_FILE1_TEST | 100.00% | 0 | 11 | | | |
| READ_FILE1_TEST | 100.00% | 0 | 13 | | | |
| UPDATE_FILE1_TEST | 100.00% | 0 | 10 | | | |
| DELETE_FILE1_TEST | 100.00% | 0 | 10 | | | |



EXEMPLOS MULTI THREADING

Você já viu nossos samples utilizando Multi Threading? É uma maneira interessante e fácil de agregar valor ao seu aplicativo. Os exemplos exibem como:

- executar um relógio em segundo plano
- evitar fechar uma janela antes da outra
- verificar se o usuário não executa um programa mais de uma vez
- contar os programas em execução
- executar vários programas no mesmo ou em ambientes separados
- deixar o usuário interromper um processo em execução
- executar uma barra de progresso



```
255. PERFORM READ-FILE1-TEST.
256. IF TEST-START = 1
257.   PERFORM START-FILE1-TEST
258.   END-IF
259. PERFORM UPDATE-FILE1-TEST.
260. PERFORM DELETE-FILE1-TEST.
261.
262. MOVE TOTAL-TIME TO TIME-DISP
263. DISPLAY "Total Time: " TIME-DISP
264.
265. GOBACK
266.
267.
268. START-FILE1-TEST.
269. OPEN INPUT FILE1.
270. PERFORM START-TIMER.
271. PERFORM VARYING IND FROM 1 BY 1 UNTIL IND IS > NUM-TIMES
272.   MOVE IND TO key-FILE1
273.   START FILE1 KEY IS = key-FILE1
274. END-PERFORM.
275. PERFORM STOP-TIMER.
276. CLOSE FILE1.
277. ADD TIME-DIFF TO TOTAL-TIME.
278. MOVE TIME-DIFF TO TIME-DISP
279. DISPLAY "START: " TIME-DISP
280.
281.
282. LOAD-FILE1-TEST.
283. INITIALIZE D-A01
284. OPEN OUTPUT FILE1.
285. PERFORM START-TIMER.
286. PERFORM VARYING IND FROM 1 BY 1 UNTIL IND > NUM-TIMES
287.   MOVE IND TO KEY-FILE1
```

DESTAQUES NA DOCUMENTAÇÃO



Nova Documentação

A documentação do isCOBOL foi redesenhada com uma nova aparência e agora está acessível aos desenvolvedores através de dispositivos móveis.

Isso também soluciona o problema que você pode ter notado na documentação de 2019 - alguns navegadores (como Chrome e Firefox) não exibem a documentação instalada devido a um novo recurso de segurança que eles introduziram.

Experimente e informe ao nosso departamento de documentação o que você acha. Basta clicar no ícone de e-mail no canto inferior direito de cada página.

Clique [aqui](#) para conferir.

Você viu isso?

Novos vídeos do YouTube

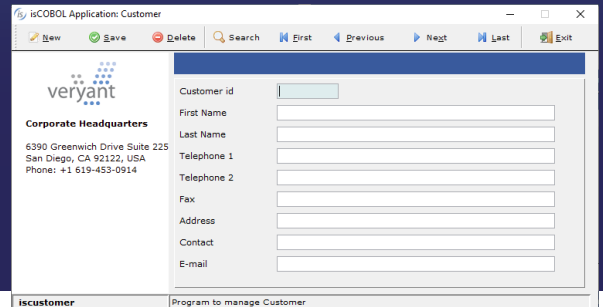
- [REST/SOAP Webservices](#)
- [Intalando o isCOBOL SDL para iniciantes](#)
- [Novos recursos da isCOBOL 2019R1](#)

Novos artigos Knowledgebase

[Como utilizar os copybooks FD/SL existentes nos Screen Programs sem gerar FD/SL a partir da IDE Data View](#)

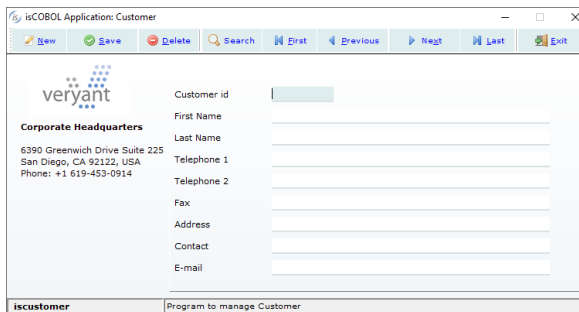
LAF: Escolhendo um design para suas telas GUI

Tela Original:



Compilado com estas definições de configuração:

```
iscobol.compiler.gui.window.defaults=gradient-color-1 rgb x#ffffff, gradient-color-2 rgb x#F2F6F9
iscobol.compiler.gui.label.defaults=transparent
iscobol.compiler.gui.tool_bar.defaults=Background-Color rgb x#FFFFFF,Foreground-Color rgb x#F2F6F9
iscobol.compiler.gui.push_button.defaults=flat, Background-color -14675438, Foreground-color x#F2F6F9
iscobol.compiler.gui.entry_field.defaults=border-color rgb x#dae1e5, border-width (0 0 2 0 )
```



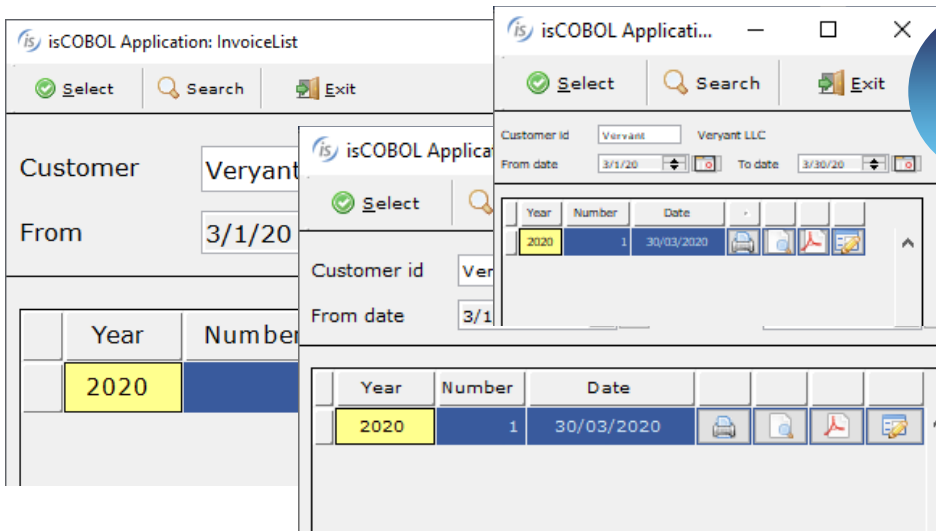
Mesmo programa compilado com a configuração acima

Uma linguagem de design é uma definição de um conjunto de estilos e princípios de design para uma aparência coesa em um aplicativo ou plataforma. [Apple Human Interface](#), introduzidas com Apple no final de 1977, [Google's Material Design language](#), desenvolvida em 2014, e o [Fluent Design System](#) da Microsoft lançado em 2017, são exemplos comuns de linguagens de design e descrevem a aparência e o comportamento que os usuários costumam ver todos os dias.

Se você deseja que seu aplicativo pareça familiar e atualizado, provavelmente acabará imitando uma ou mais dessas linguagens de design.

Na 2019R2, adicionamos alguns recursos para ajudar a manter as telas GUI atualizadas: bordas personalizadas para entry fields, para que você possa copiar o design do Google apenas com uma borda inferior visível e o aprimoramento dos push-buttons e também das abas do controle accordion

E para facilitar, você pode aplicar esses e outros estilos no arquivo de configuração em vez de adicionar individualmente à definição de cada controle em cada programa.



DESTAQUES 2019 R2

Aqui estão alguns dos destaques:

- Insira estilos GUI no seu programa a partir do arquivo de configuração, sem alterar seu código
- Melhoramos a manipulação dinâmica de variáveis
- Aumentamos o desempenho das instruções CALL e das operações c-treeRTG.
- Nossa compatibilidade com RM/COBOL foi aprimorada para oferecer suporte a v8 ou superior
- O painel de gerenciamento do WebClient se tornou mais informativo e seguro com a gravação automática de sessões e notificações push para o client.



Zoom Layout Manager

No último boletim informativo, falamos sobre o Responsive Layout Manager - permitindo aplicar o comportamento de redimensionamento a cada controle na tela para responder a diferentes ambientes de exibição.

Na 2019R2, ampliamos isso com o estilo lm-zoom. Definindo esta configuração no seu arquivo de propriedades:

iscobol.gui.layout_manager=lm-zoom

Isso possibilita que seu aplicativo se adapte automaticamente a vários

dispositivos, tamanhos e resoluções de tela. Ele permite que o usuário redimensione uma janela do aplicativo ou gire uma tela, um comportamento necessário na era dos dispositivos móveis. Suas janelas são redimensionadas automaticamente - a propriedade size dos controles se ajusta à largura da janela e a propriedade size da fonte dos controles se ajusta ao comprimento da janela.

Isso fornece uma maneira fácil de acomodar diferentes tamanhos e resoluções de tela sem alterar ou recompilar seu código.

Tipos de Mensagens para Erros de Compilação

iscc prog1.cbl

- S: #36 Ambiguous identifier A; file = prog1.cbl, line = 17, col 11
- E: #154 End statement required END-PERFORM; file = prog1.cbl, line = 21, col 19
- W: #254 Procedure name same as data name: A; file = prog1.cbl, line = 23

Você sabe o que S, E e W significam acima? Quando você compila seu código, existem tipos de mensagens. S é um erro grave, portanto, seu código não será compilado em uma classe. E é um erro regular e W é um aviso - ambos não impedirão o compilador de compilar e criar sua classe.

Se, por exemplo, você não quiser ver o aviso acima toda vez que compilar, poderá criar uma propriedade de configuração para ocultá-lo:

iscobol.compiler.messagelevel.254=0

[Leia mais sobre isso em nossa documentação.](#)



JOE - O Poder Por Trás da Linguagem Script da Veryant

Tradicionalmente os desenvolvedores de software misturam linguagens de desenvolvimento com utilitários feitos em linguagens de script para unir seus módulos de aplicativos. Essas linguagens de script (shell script) oferecem uma estrutura mais flexível, nenhuma fase de compilação envolvida e uma maneira flexível e simples de interagir com o sistema operacional que geralmente não possui linguagens de desenvolvimento mais formais. Geralmente, os aplicativos legados combinam o desenvolvimento COBOL com as linguagens de script baseadas no SO e algumas funções principais dos aplicativos podem depender muito dele. Se a plataforma de desenvolvimento não possuísse o equivalente a essa linguagem de script, isso complicaria bastante a migração. Considerando tudo isso, a Veryant oferece nossa própria linguagem de script, chamada JOE.

Além dos benefícios da linguagem de script padrão, o JOE também pode:

- Interagir diretamente com classes Java
- Interagir com classes do isCOBOL
- Acessar e atualizar arquivos ISAM
- Acessar e atualizar qualquer

- RDBMS via driver JDBC e SQL
- Acessar e definir variáveis de ambiente
- Oferecer fluxo lógico básico, mas poderoso, e manipulação de expressões
- Interagir com aplicativos do Office (como Excel) por meio de classes de Apache POI ou similares
- Trabalhar perfeitamente com a arquitetura do servidor de aplicativos

E como qualquer outra linguagem de script, JOE:

- É fácil de entender, usar e estender (além da fase de migração)
- Pode ser personalizado para lidar com operações frequentes
- Assemelha-se a qualquer outra linguagem de script em termos de capacidade e legibilidade

Dado tudo isso, o JOE é a resposta perfeita do que fazer com seu legado de linguagem de procedimentos. É a escolha perfeita para modernização e desenvolvimento contínuo para implementar operações frequentes com poderosa simplicidade e elegância, unindo os componentes de seu aplicativo.

```

1  /** isCOBOL exec JOE */
2
3  checkSurname := {a_value.
4    a_surname := a_value toString ; toUpperCase ; trim.
5    !if ((a_surname = "DOE") or (a_surname = "SMITH")), {
6      "Yes, he's a member of Veryant staff".
7    } ; {
8      "Veryant staff only allowed here!".
9    }
10 } .
11
12 checkCompany := {a_value.
13   a_company := a_value toString ; toUpperCase ; trim.
14   !if (a_company = "VERYANT"), {
15     "Yes".
16   } ; {
17     "Only Veryant is allowed!".
18   }
19 } .

```



OPÇÕES DE DADOS

Com o isCOBOL, você tem uma ampla variedade de opções de armazenamento de dados, que incluem:

JISAM

Este é o formato de arquivo indexado padrão no runtime do isCOBOL.

c-tree

Essa é outra opção inclusa no seu runtime. Em vez de usar diretórios compartilhados para seus arquivos de dados, como o JISAM faz, o c-tree é um servidor de arquivos, adicionando velocidade e segurança aos seus dados. C-TREE pode ser expandido para ser um banco de dados e permitir acesso aos seus dados com SQL.

RDBMS com Sintaxe COBOL

Nosso produto Database Bridge, também conhecido como EasyDB, converte suas instruções I/O de arquivo COBOL em instruções SQL e se conecta ao seu sistema de banco de dados relacional

RDBMS com ESQL

O suporte ESQL está embutido no compilador isCOBOL, sem necessidade de pré-compilação

File Connectors

Acesse seus dados nos arquivos Vision, c-tree, DBMaker ou Micro Focus ISAM sem convertê-los.

DBMaker e Pervasive databases

As interfaces para acesso a dados como DCI e Btrieve são incorporados ao isCOBOL, para que você armazene seus dados nesses sistemas de banco de dados sem o EasyDB ou outros custos adicionais.



Evolution without revolution

Veryant LLC

A equipe de suporte dedicada e experiente da Veryant está comprometida em fornecer os mais altos níveis de atendimento ao cliente

Para suporte, envie um email para support@veryant.com

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

Se você desejar que a Veryant entre em contato com você para obter cotações específicas ou assistência à vendas, envie um e-mail para sales@veryant.com

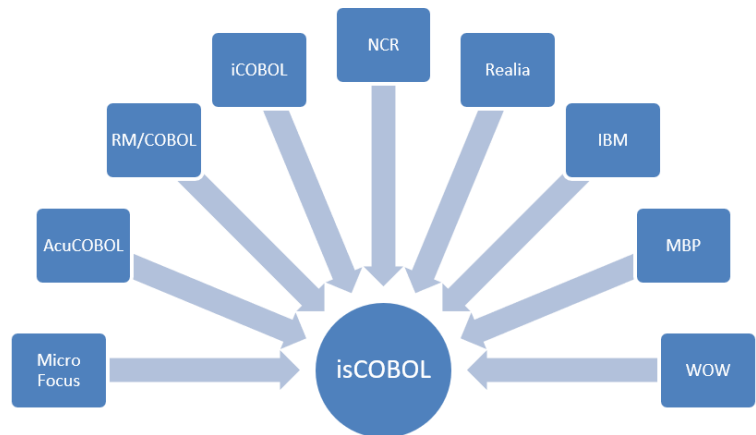
Sede Principal

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

Sede Europeia

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

Como sempre, a 2020R1 contém várias adições de compatibilidade - enquanto continuamos a tornar seu processo de conversão o mais simples e rápido possível.



veryant.com

Oferecendo opções de escolha, flexibilidade e soluções econômicas para organizações com ativos COBOL valiosos

CONTATE-NOS PARA UMA DEMONSTRAÇÃO

CLIQUE AQUI



Cool Business Oriented Language

Follow Veryant on



veryant.com