

Novo ABAP Debugger – Porque largar o velho? – Parte 1

14/01/2011 16:42

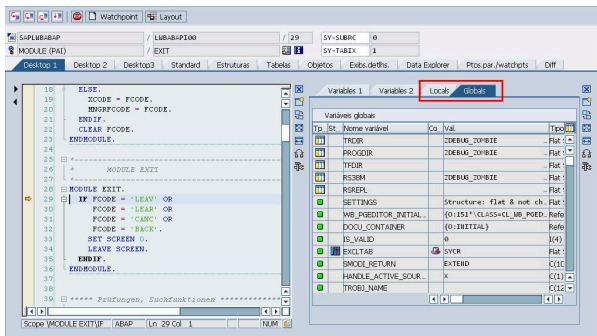
Porque largar o debugger clássico? Eu também fazia esta pergunta quando trabalhei na 6.0 pela primeira vez. Depois de um tempo, dei uma chance para o debugger novo, e agora não largo mais.

Sempre tive vontade de aprender tudo que o debug novo pode fazer, portanto, decidi fazer uma série de posts para explicá-lo.

Só tem **1** problema: eu não sei usar ele por completo :???:. Mas tranquilo, eu pesquiso e vou postando conforme for descobrindo.

Hoje vou explicar algumas coisas que eu sei e acho muito úteis. Vamos lá!

1 – Acesso rápido e fácil à **memória carregada** em tempo de execução.



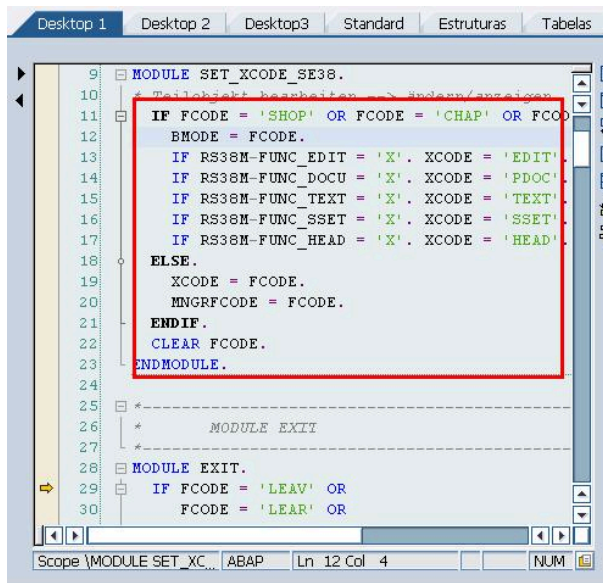
Clicando em Locais/Globais, você vê tudo que o SAP carregou na memória para aquele programa.

Em Desktop 2, você pode navegar na pilha, e indo em Locais/Globais denovo, pode ver a memória de cada programa sendo executado.

2 – Layout interativo.

Divirta-se fuçando na tela, colocando e tirando o que você quiser (sem malícias! 🤪). Se você gravar o layout, ele grava o layout (duh) e mostra denovo quando você debuggar outra coisa.

3 – Se você utilizar também o editor novo, o código é mostrado em formato texto, permitindo que você navegue no código da mesma maneira que você faria na se38. No antigo, tinha que ficar dando page down, page up...



Beeeem mais fácil para analisar POFs (programas orientados a flags), já que ele deixa em negrito IFs, LOOPS, DOs e etc's, em que você clicar.

4 – Análise mais fácil de dados em tabelas, além de otimização na hora de preencher tabelas ou estrutura manualmente durante um debug.

Aff, era muito chato preencher tabelas no Debugger antigo. Não é perfeito no novo, mas é bem menos chato.

5 – Você pode usar o go-to livremente pelo atalho shift+f12

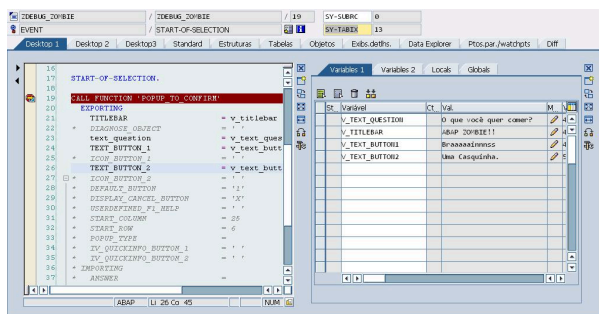
É só clicar numa linha anterior da que você está debuggando e clicar SHIFT+F12, que o cursor vai

voltar. Parece até VB 😬 haha. Mas cuidado, como não é um processo normal, às vezes isso destrói a execução do programa!

6 – Download de tabelas internas para arquivos.

Útil para achar registros ou entender/comparar os dados.

7 – **Esse é animal**: salvar parâmetros de uma função para executar depois na se37. Não é mágica, é tecnologia haha. Chega de se matar para testar BAPIs!



Fiz um programa simples, que chama uma função de popup. Vejam os valores ali nas variáveis. Agora, é só apertar F5 para entrar na função...

... e uma vez dentro da função, clique com o botão direito na área de variáveis (não precisa

mostrar as variáveis, coloquei ali só para vocês verem os valores), e escolha a opção da figura.

Diretório de dados teste: testes individuais

Teste de regressão | Exibir resultado de teste

ID registro dados	Data	Hora	Texto breve	S
1	14.01.2011	16:23:25	TESTE_ZOMBIE	

Testar módulo de função: tela de entrada

Depuração | Diretório de dados de teste

Teste p/grupo de funções: SP01
Módulo de função: POPUP_TO_CONFIRM
Maiúsculas/minúsculas: 02

Parâmetro importação	Valor
TITLEBAR	ABAP ZOMBIE!!
DIAGNOSTIC_OBJECT	
TEXT_QUESTION	O que você quer fazer?
TEXT_BUTTON_1	Braaaaainms
ICON_BUTTON_1	
TEXT_BUTTON_2	Uma Casquinha.
ICON_BUTTON_2	
DEFAULT_BUTTON	1
DISPLAY_CANCEL_BUTTON	X
USERDEFINED_F1_HELP	
START_COLUMN	25
START_ROW	6
POPUP_TYPE	
TV_CHECKINFO_BUTTON_1	
TV_CHECKINFO_BUTTON_2	

Se você for na Se37, você vai ver que ele grava uma variável nova no diretório de testes da função, com os valores do debug. Muito louco!

Acho que está bom para começar. Tem mais um montão de coisas, eu vou postando aos poucos, senão fica muuuuito longo (mais ainda!)



) e ninguém assimila nada.

Se você souber de algo legal e quiser compartilhar, deixe um comentário! Vamos “dissecar” esse debugger novo 😊

Abraços!

Comentários

Marcelo — 21/08/2012 10:25

Criar os dados para testes de funções foi muito válido!

Valeu!

Vinicius Ostan — 13/03/2012 15:58

Vou encaminhar esses posts para uns amigos que “odeiam” o Debugger novo... hehehe

Parabéns!

Bruno Cappellini — 20/12/2011 15:34

Muito obrigado pelas dicas!

Estou estudando muito pelo seu site!