

ABAPZombie Guide to ABAP – Parte 7 – ASSIGN

10/02/2011 08:59

ASSIGN! Dá pra falar bastante coisa desse comando que é uma mão na roda para qualquer programador.

*** Atenção, este é um post longo porque contém explicações de diversas técnicas com o ASSIGN. Eu achei que fosse ficar menor.., mas enfim 😊**

Não tem como falar de ASSIGN sem falar de FIELD-SYMBOLS, por isso eu irei adiantar o tópico dos [FIELD-SYMBOLS](#) (clique para ler, caso não saiba do que se trata!).

O que ele faz: você cria uma referência para uma área de memória, e associa essa refência a um Field-Symbol através do ASSIGN.

Isso que dizer que, após dar um ASSIGN, você pode mudar o valor da variável X diretamente pelo Field-Symbol, como no exemplo abaixo:

```
REPORT zombie_assign.

* Declaração dos Field-Symbols
*-----
FIELD-SYMBOLS: <fs_teste> TYPE ANY.

* Variável Qualquer
*-----
DATA: v_matnr TYPE mara-matnr.

* Cria a referência da área de memória no Field-Symbol. Note que
* é necessário colocar o nome da variável entre ( ), além de o nome
* estar em letras maiúsculas.
ASSIGN ('V_MATNR') TO <fs_teste>.

IF IS ASSIGNED.
  <fs_teste> = 'MATERIALZUMBI'.
ENDIF.

WRITE v_matnr.
```

A vantagem principal é que você pode usar um mesmo field-symbol para alterar variáveis diferentes com a **mesma linha de código**. Como no exemplo abaixo:

Sempre que você for atribuir valores a um Field-Symbol verifique antes se o mesmo foi realmente referenciado, através da consistência **IS ASSIGNED**. Se você tentar usar um Field Symbols que não foi corretamente referenciado, o programa irá gerar um DUMP! Você também pode desfazer a referência através do comando **UNASSIGN <fs>**.

Mas qual a vantagem disso tudo? Elementar meu caro leitor: **usar o mesmo field-symbol para alterar valores de campos diferentes!** 😊

```
REPORT zombie_assign.

* Declaração dos Field-Symbols
*-----
FIELD-SYMBOLS: <fs_teste> TYPE ANY.

* Note que cada variável é de um tipo diferente!
*-----
DATA: v_matnr TYPE mara-matnr,
```

```

v_mtart TYPE mara-mtart,
v_labst TYPE mard-labst.

* Cria a referência da área de memória no Field-Symbol. Note que
* é necessário colocar o nome da variável entre ( ), além de o nome
* estar em letras maiúsculas.
ASSIGN ('V_MATNR') TO <fs_teste>.

IF <fs_teste> IS ASSIGNED.
  <fs_teste> = 'MATERIALZUMBI'.
ENDIF.

WRITE <fs_teste>.

ASSIGN ('V_MTART') TO <fs_teste>.

IF <fs_teste> IS ASSIGNED.
  <fs_teste> = 'FERT'.
ENDIF.

WRITE <fs_teste>.

ASSIGN ('V_LABST') TO <fs_teste>.

IF <fs_teste> IS ASSIGNED.
  <fs_teste> = '1234'.
ENDIF.

WRITE <fs_teste>.

```

O ASSIGN ainda tem mais coisas que podem ajudá-lo em diversos momentos. Vamos lá:

– **ASSIGN COMPONENT OF STRUCTURE:** Ele permite que você associe num Field-Symbol um campo qualquer de uma estrutura. E você nem precisa saber o nome exato do campo, só a posição dele na sequência de declaração da estrutura. vejamos:

```

REPORT zombie_assign.

* Declaração dos Field-Symbols
*-----
FIELD-SYMBOLS: <fs_wa> TYPE ANY.

* Work Area
*-----
DATA: wa_mara TYPE mara.

* Assign de estrutura com o nome do campo da estrutura
ASSIGN COMPONENT 'MATNR' OF STRUCTURE wa_mara
              TO <fs_wa>.

IF <fs_wa> IS ASSIGNED.
  <fs_wa> = 'MATERIALZUMBI'.
ENDIF.

WRITE wa_mara-matnr.

* O campo 10 é o campo MTART
ASSIGN COMPONENT 10 OF STRUCTURE wa_mara
              TO <fs_wa>.

IF <fs_wa> IS ASSIGNED.
  <fs_wa> = 'FERT'.
ENDIF.

```

```
WRITE <fs_wa>.
```

– **ASSIGN CASTING:** Ele faz a mesma referência dos outros, mas o field-symbol mostra o valor do tipo que você usou no casting. Ou seja, você pode associar uma variável char à um field-symbol que irá mostrar o seu valor em hexadecimal.

Isso é útil para achar e manipular caracteres especiais em strings, como quebras de linha e tabs indesejados (aqueles que aparecem como # no debug).

```
REPORT zombie_assign.

* Declaração dos Field-Symbols
*-----
FIELD-SYMBOLS: <fs_x> TYPE ANY.

* Variável CHAR
*-----
DATA: v_char TYPE char1.

* Referenciando o valor de V_CHAR trocando o tipo
ASSIGN ('V_CHAR') TO <fs_x> CASTING TYPE x.

* Se movermos AB para V_CHAR..
v_char = 'A'.

*... o WRITE do Field-Symbol escreverá 41, que é o código em hexa do
* A maiusculo :)
WRITE <fs_x>.
```

– **ASSIGN de variáveis de outros programas:** Você pode associar a memória não só do seu programa, mas de qualquer outro programa carregado na memória da execução. Isso é muito usado em EXITS, BADIs e Enhancement Points, para a leitura de dados que nem sempre estão disponíveis na interface do FORM, Função ou método:

```
* Um exemplo simples de cada capturar valores de outros programas carregados na memória.
* Neste caso, estou capturando o valor de uma tabela que está declarada no
* programa SAPMV45A e associando a referência no field-symbol.
ASSIGN ('(SAPMV45A)XVBAP') TO <fs_tabela>.

* Note que você pode usar as dicas do tópico do Novo Debugger
* para navegar entre a pilha de programas carregados e verificar
* as variáveis que estão na memória.
```

– **LOOP AsSIGNING / READ ASSIGNING:** Você pode ainda associar uma estrutura durante um loop ou um read, economizando uma work area e alterando os valores direto na tabela. Eu uso muito isso para alterar valores na tabela em que estou dando um Loop:

```
REPORT zombie_assign.

* Nesta declaração é necessário definir o tipo exato da Work Area para
* o Field-Symbol, caso contrário o comando <fs_mara>-matnr não irá
* ativar...
FIELD-SYMBOLS: <fs_mara> TYPE mara.

* Tabela
DATA: t_mara TYPE TABLE OF mara.

* Linha vazia, só para mostrar o Loop
APPEND INITIAL LINE TO t_mara.

LOOP AT t_mara ASSIGNING <fs_mara>.
```

** Associando o valor direto na linha*

```
<fs_mara>-matnr = 'MATERIALZUMBI'.
```

ENDLOOP.

** Analise a tabela. Sem MODIFY nem nada, os valores da linha foram*

** alterados :) Você pode usar isto com o o READ TABLE também!*

BREAK-POINT.

Phew, teminei. Aho que estes são os mais utilizados.. 😊 Qualquer dúvida é só comentar!

Abraço!

Comentários

Julio Cesar Marques Pires — 17/05/2021 21:58

Valeu Muricio mto obrigado pelas dicas ! foi de grandre vali !

Hilda Araujo — 21/08/2017 15:25

Boa tarde.

Isso só funciona para programa??

E se a tabela/variavel estiver num classe/método, como faço?

Vinicius Ostan — 02/08/2013 16:35

Grande Maurício,

Cara, que tópico excelente.

Seu site é minha wiki ABAP, mas sobre ASSIGN, tem um ponto que não encontrei aqui, talvez fosse legal incluir-lo, segue abaixo:

ASSIGN ->* TO .

Eu costumo usar muito para trabalhar com tabela dinâmica, mas confesso que não sei bem como funciona esse " ->* " .

Abraço

Rafael — 10/07/2013 15:57

Olá,

Parabéns pelo post, muito bem explicado. No exemplo com a tabela interna, o field symbol foi usado para modificar os dados da mesma. Como trata-se de um ponteiro para area de memória da linha da tabela acessada, não necessita fazer o "modify".

Queria saber se o field symbol também é recomendado para popular uma tabela interna, e caso seja, como poderia ser feito isso?

Mauricio Cruz — 10/07/2013 16:06

Cuidado... se você sair falando por aí que um field-symbol é um ponteiro, alguém algum dia irá reclamar 😊
 Leia esta discussão: <http://scn.sap.com/thread/3227484> .

E o último exemplo do post é um exemplo de preenchimento de tabela interna com field-symbol. Mas cara, usa uma WA para adicionar dados em uma tabela que fica bem mais claro no código o que você está fazendo.

Abs!

Rafael — 10/07/2013 16:19

Ok, obrigado pelas dicas e pela rapida resposta!

Jahniffer Santos — 06/06/2013 14:28

Olá, Maurício. Tenho gostado muito de usar field-symbol e adotado-o sempre que possível em meus códigos, principalmente o do último caso de suas dicas. Adorei o resumo de como utilizá-lo ficou fácil de entender. Usarei mais vezes.

Sobre seu comentário, concordo plenamente sobre ser dever do programador utilizar os diversos artifícios da linguagem da melhor maneira possível. E sempre manter-se atualizado para conhecer os artifícios oferecidos. Acredito que a escolha de uso entre um artifício e outro deve ser determinada por aquele que oferece mais benefícios.

Muito obrigada por compartilhar os conhecimentos.

Tim — 09/03/2012 10:02

Muito bom esse tópico e muito bem explicado!!

Estão de parabéns pelo ótimo conteúdo do site.

Abs.

Mauricio Cruz — 09/03/2012 10:06

Muito Obrigado!

Alex Ferreira — 30/01/2012 09:24

Obrigado Mauricio!!

O post ficou muito bom, bem explicado e muito didático. Sanou várias dúvidas que eu tinha a respeito de como utilizar o FIELD-SYMBOL e ASSIGN.

É muito bom ter pessoas como você, dispostas a compartilhar o conhecimento. Valeu.

Pedro Paulo Costa — 03/01/2012 08:40

Estava me perguntando se realmente era mais rápido loop/assign x loop/work area/modify usando sy-tabix... Ampliei o código acima, e a t_mara ficou com 10000 registros. Alterando o campo matnr de cada registro usando

field-symbols, obtive 28.259 microseconds.

Agora,

Fazendo a mesma alteração usando work area e o comando modify/sy-tabix, obtive 41.829 microseconds.

Realizando outros testes com diferentes quantidades de registros na t_mara, percebe-se que há uma melhora na performance somente para tabelas acima de 5000 registros.

Ex: 1000 registros:

field-symbols: 2.167 microseconds

work-area/modify = 2.966 microseconds.

Gostei dessa idéia de usar field-symbols no loop, mas acredito que quanto mais field-symbols tem no código mais fica difícil de entendê-lo....Creio que seja bom estudar se é realmente necessário no programa e usá-lo com consciência para aproveitar as vantagens desse comando e não deixar o código difícil para que até nas próximas manutenções do código o consultor que ler, conseguir entender da melhor maneira possível.

Concorda, Mauricio?

Mauricio Cruz — 03/01/2012 09:46

Olá Pedro! Obrigado pela comentário!

Concordo plenamente que devemos avaliar a viabilidade não só do LOOP ASSIGNING, mas de qualquer técnica de programação antes de sair utilizando.

Interessante sua análise.. mas eu achei os valores meio estranhos.. eu explico: na comparação que fiz no programa [deste post](#), o LOOP ASSIGNING sempre foi mais rápido, e mesmo assim a diferença é mínima, ou seja, tanto faz usar um ou outro pois não vai causar impacto significativo na performance do código. Se quiser fazer mais alguns testes e encontrar algo diferente, deixe o código aqui nos comentários para eu poder rodar e verificar também! 😊

Discordo que o Field-Symbol deixe o código difícil: o que deixa o código difícil é o jeito com que as pessoas o utilizam. Se a linguagem tem diversos artifícios, é dever do programador utilizá-los da melhor maneira possível, sempre tentando (e conseguindo na maior parte das vezes 😊) deixar o código performático, estruturado e objetivo.

Um abraço!
Mauricio

Allan Araujo — 20/10/2011 13:35

No post acima tentei colocar o sinal de 'Maior que' e 'Menor que' rs.

Allan Araujo — 20/10/2011 13:34

Obrigado pela dica!

E só atentando pra quem ficar perdido à 1ª vista com o uso de < e >, eles podem ser substituídos por , certo?

Abs

Mauricio Cruz — 20/10/2011 13:37

Olá Allan! Sim, pode usar como maior e menor.. isso é bug aqui do nosso site. Mas vamos acabar com essa coisa logo logo 😊

Arrumei o post, bele?

Abraços!

Bruno Cappellini — 26/04/2011 19:49

Muito obrigado pelas dicas e pelo conteúdo irado do site!

To estudando muito por aqui! Abraço!