USANDO ASSIGN E FIELD-SYMBOLS PARA ALTERAR DADOS NA MÉMORIA

O commando FIELD-SYMBOLS é usada para declarar uma referência à uma área de memória.

A declaração dessas refência pode ou não ter um tipo definido. Você pode declarar um field-symbol type ANY, ou type ANY TABLE, indicando que ele estará pronto para referênciar qualquer tipo de variável ou tabela.

Para associar uma área de memória à um field-symbol, utilizando o comando ASSIGN.  
  
Exemplos de Declaração:  
  REPORT  zteste.  
  
FIELD-SYMBOLS: <fs\_any>   TYPE ANY,       *"Qualquer coisa menos tabelas*  
               <fs\_table> TYPE ANY TABLE, *"Qualquer tabela*  
               <fs\_vbak>  TYPE SPFLI.      *"Tipo definido*  *Após dar um ASSIGN, você pode mudar o valor da variável X diretamente pelo Field-Symbol isto é muito útil quando se quer mudar valores de campo em tela ou variáveis em um programa standard por exemplo é sempre bom validar se a variável esta referenciado verificando se ela IS ASSIGNED para evitar Dump por falta de referência*, como no exemplo abaixo:  
*REPORT  zteste.   
  
\* Declaração dos Field-Symbols  
\*-----------------------------------  
FIELD-SYMBOLS: <fs\_teste> TYPE any.  
  
\* Variável Qualquer  
\*-----------------------------------  
DATA: v\_teste TYPE char30.  
  
" RECEBENDO O VALOR NA VARIAVEL  
v\_teste = 'TESTE'.  
  
\* 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 maíusculas.  
ASSIGN ('V\_TESTE') TO <fs\_teste>.  
  
"MODIFICANDO O VALOR DO ENDEREÇO DE MEMORIA  
IF  <fs\_teste> IS ASSIGNED.  
  <fs\_teste>  = 'TESTE MEMORIA'.  
ENDIF.  
  
WRITE v\_teste.    
- 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, ou seja, você consegue alterar variáveis de outros programas na memória se souber usar este recurso:*  
*REPORT  zteste.  
  
FIELD-SYMBOLS <fs\_tabela> TYPE ANY TABLE. "Qualquer tabela  
\* 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.*