

## ABAPZombie Guide to ABAP – Parte 6 – ASSERT

09/02/2011 09:00

Táí um comando que pode ajudar muito o seu trabalho. Sabe quando você tem aquele processo chato que só roda em BACKGROUND, ou em UPDATE TASK, e você não consegue debuggar nem por decreto? Ou por falta de permissões, ou por falta de acessos, falta de conhecimento mesmo...

**E o pior:** às vezes é só para saber se a variável X está igual a Y, e consequentemente está ferrando toda a lógica!

Então eu te pergunto: **Porque não usar o ASSERT para gravar as expressões** lógicas do seu programa?

O ASSERT funciona como uma espécie de "IF", ou seja, ele é um comando que valida se uma expressão lógica é falsa ou não. Veja o código abaixo:

```
REPORT    zombie_assert.

DATA: v_mat1 TYPE mara-matnr,
      v_mat2 TYPE mara-matnr.

DATA: t001a TYPE TABLE OF t001,
      t001b TYPE TABLE OF t001.

DATA: o_cont1 TYPE REF TO cl_gui_container,
      o_cont2 TYPE REF TO cl_gui_container.

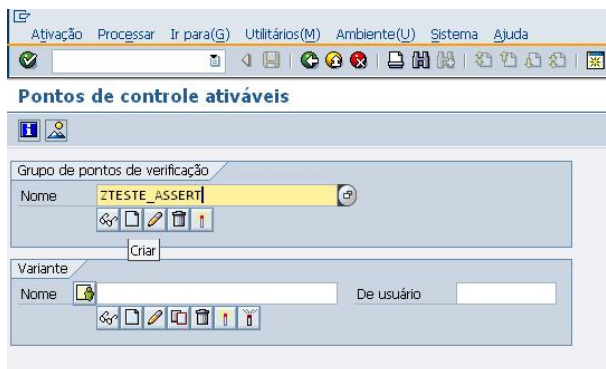
START-OF-SELECTION.

  v_mat1 = '123'.
  v_mat2 = '234'.

  * Assert que irá disparar um DUMP, por não ter nenhum controle
  * externo
  ASSERT v_mat1 = v_mat2.
```

Neste exemplo acima, a expressão lógica é falsa, e o ASSERT irá gerar um DUMP por conta disso – mas, como escrito no comentário, isso só acontece porque não tem nenhum controle externo no ASSERT.

**Como fazer isso?** É simples! Primeiro acesse a transação SAAB:



Transação SAAB. Coloque um novo nome para o "Grupo de pontos de Verificação" e clique na página em branco, para criá-lo.

Depois de criado, vão aparecer diversas opções para o seu grupo. Deixe do mesmo jeito que na foto abaixo:

Escolhendo "reg.em log", você permite que o comando ASSERT guarde um log de suas validações lógicas neste grupo.

O flag **"Interromper"** na foto acima, irá fazer o programa parar no debug quando a condição lógica do ASSERT falhar, e **"Cancelar"** fará com o que o programa dispare um DUMP sempre que a condição lógica do ASSERT falhar.

Pois bem, salve o grupo, e execute o seguinte código:

```
REPORT  zombie_assert.

DATA: v_mat1 TYPE mara-matnr,
      v_mat2 TYPE mara-matnr.

DATA: t001a TYPE TABLE OF t001,
      t001b TYPE TABLE OF t001.

*-----
* START-OF-SELECTION
*-----
START-OF-SELECTION.

  v_mat1 = '123'.
  v_mat2 = '234'.

* Assert com log de comparação de variáveis
*-----
  ASSERT ID zteste_assert FIELDS v_mat1 v_mat2
  CONDITION v_mat1 = v_mat2.

* Select em uma tabela qualquer
  SELECT *
    FROM t001
    INTO TABLE t001a.

  APPEND LINES OF t001a TO t001b.

* Fazendo com que as tabelas fiquem diferentes
  DELETE t001b INDEX 1.
  DELETE t001b INDEX 1.

* Assert com log de comparação de tabelas
*-----
```

```
ASSERT ID zteste_assert FIELDS t001a t001b  
CONDITION t001a = t001b.
```

Nada vai acontecer quando você rodar o programa, mas se voltarmos na transação SAAB, escolhermos o grupo ZTESTE\_ASSERT e verificarmos a aba “Log”:

Vamos ver que as desigualdades ficaram logadas aqui no grupo. Animal, não é?

Agora você não tem mais desculpa para não conseguir avaliar direito seus programas.

Como você perceberá, o grupo na SAAB tem várias opções, inclusive para logs de ASSERTs em BACKGROUND. O negócio agora é fuçar.

Eu, particularmente, descobri este comando enquanto estudava para fazer este post, e achei muito bom! Espero que você também tenha gostado!

Abraço!

---

## Comentários

**Rogério Coimbra** — 27/08/2015 16:57

Muito bom!