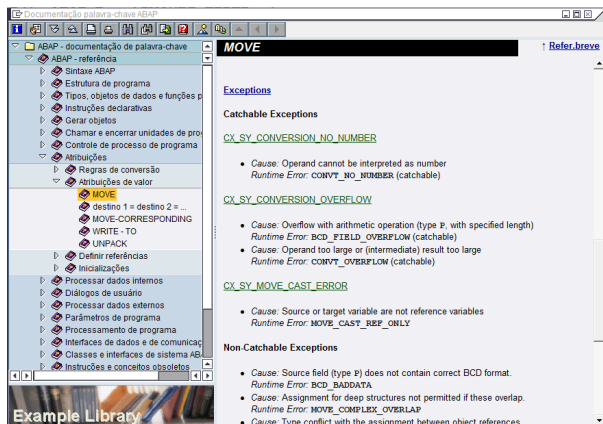


ABAPZombie Guide to ABAP – Parte 29 – CATCH SYSTEM-EXCEPTIONS

12/12/2011 11:00

Você provavelmente já apertou F1 em um comando, e viu no final do help uma lista com as "Exceptions" que o sistema pode disparar caso você não utilize o comando corretamente. Aquela é a famosa lista de *Runtime Errors*(dumps) do comando, com as informações devidas caso você queira prevenir a "Exception" e tratá-la dentro do seu programa (mas nem todas podem ser tratadas!).



Help do comando MOVE. Notem que a descrição mostra a classe de exceção e também o nome dos Runtime Errors (dumps) que você pode tratar (e também os que você não pode tratar).

Nas versões mais novas do AS (*Application Server*), essa lista vai mostrar as Classes de Exceção por onde você deve capturar o erro, como na figura cima. O Mauro [fez um post](#) muito bom explicando como as Classes de Exceção funcionam na Orientação à Objetos, inclusive mostrando um caso de capturar uma exceção do sistema para o comando *MOVE*.

Mas aqui eu quero mostrar o jeito mais antigo, velho, mais ROOTS e **obsoleto** de capturar exceções disparadas pelo sistema. Tipo um trabalho histórico, manja? 😊

Na figura ali de cima, além de mostrar as Classes de Exceção, o sistema também nos mostra o nome do *Runtime Error* de cada exceção. Vamos forçar um deles:

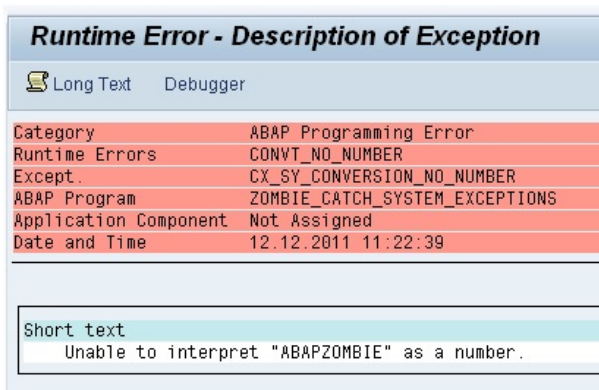
```
REPORT  zombie_catch_system_exceptions.

DATA:  vl_string TYPE char10,
       vl_numc   TYPE i.

vl_string = 'ABAPZOMBIE.BLOG.BR'.

MOVE vl_string TO vl_numc.
```

Se você executar o comando acima, o sistema vai disparar o seguinte *Runtime Error*:



Tela do DUMP para o comando MOVE, usado de forma errada.

Lembram-se que esse *Runtime Error* aparecia como “Catchable” no help do comando? Isso indica que podemos tratá-lo utilizando o **CATCH-SYSTEM EXCEPTIONS**. Veja só:

```
REPORT  zombie_catch_system_exceptions.

DATA: vl_string TYPE char10,
      vl_numc   TYPE i.

vl_string = 'ABAPZOMBIE.BLOG.BR'.

CATCH SYSTEM-EXCEPTIONS convt_no_number = 7.

  MOVE vl_string TO vl_numc.

ENDCATCH.

IF sy-subrc = 7.
  WRITE 'Erro no Move!'.
ENDIF.
```

Ou seja, você vai associar um número para o return code (sy-subrc), exatamente como você já faz quando chama alguma função. Após o Catch, você trata o erro do jeito que achar melhor.

Eu usei o mesmo exemplo do [post do Mauro](#) de propósito. A idéia é que você possa comparar as duas formas de capturar a exceção, a antiga e a nova. Mas lembre-se que esta forma apresentada neste post, é obsoleta nas versões mais novas do SAP.

É importante dizer para os mais novos que as exceções existem por um motivo: você usou o comando de uma forma **que não deveria ter usado**. O correto é você entender o porque isso aconteceu e corrigir o código, e **não sair jogando CATCHs para todos os comandos**.

Abraços!

Comentários

Ivan — 10/06/2013 10:47

Valeu mesmo!

Eu estou trabalhando com ABAP a 8 meses (Junior) e tenho aprendido muito com esse site.

Recentemente eu dei manutenção num código que tinha uma situação muito parecida com essa e eu não sabia muito bem o que significava. Como ficar perguntando tudo para os mais sabidos aqui do trampo é meio

chato....sempre venho navegar por esses lados e dessa vez foi bem no alvo!
Estive pensando em colaborar de alguma maneira...se tiver como...avise.
Abraços e obrigado.

Mauro Laranjeira — 12/12/2011 16:22

Muito bom Mauricio.

Eu não conhecia essa forma XD

Abs,

Mauricio Cruz — 13/12/2011 10:49

É nós!

Abraços a todos aqueles que podiam me ajudar com essa saga. 🙄