

Tela de Seleção Turbinada – Parte 1 – Botões

21/12/2010 11:49

A tela de seleção no SAP contém inúmeras funções que poucas pessoas utilizam. É possível criar subscreens, botoes, tabstrips... A idéia dessa série de posts é tentar **prevenir que você crie um online ou uma tela sem necessidade**, além de **ajudar você a deixar sua tela de seleção mais “gringa”**, e quem sabe surpreender algum usuário por aí 😊

Deixa sua tela de seleção **no modo turbo** após essa sequência de posts!

Dica de hoje... **como adicionar um botão na tela de seleção?**

Veja o código abaixo. Você pode também usar o velho e sempre bom CTRL+C CTRL+V e testar num programa local num ambiente que você tenha acesso:

```
REPORT z_teste_botao.

* Estrutura necessária para manipulação do elemento na Tela
*-----
TABLES: sscrfields.

* Type Pools com os Ícones (duplo clique para visualiza-los)
*-----
TYPE-POOLS: icon.

* Declaração do Botão
* Você pode utilizar da função 1 até a função 5
* O atalho será Ctrl+Fx onde X é o número da Function Key
*-----
SELECTION-SCREEN: FUNCTION KEY 1. "Declaração do Botão

* O Botão por sí só não faz uma tela de seleção.
* Outros elementos de tela devem existir, e o botão irá aparecer ao lado
* do botão executar.
*-----
PARAMETERS: p_dummy TYPE char10.

* Descrevendo o botão no evento Initialization
*-----
INITIALIZATION.

    PERFORM cria_botao.

* Manipulação do Botão
*-----
AT SELECTION-SCREEN.

    IF sy-ucomm = 'FC01'.
        BREAK-POINT.
    ENDIF.

*&-----*
*&      Form  cria_botao
*&-----*
FORM cria_botao .

* Estrutura para descrever o botão
DATA: wa_button TYPE smp_dyntxt.

* Nome do Botão
wa_button-text   = 'Botao Log de Erros'.
```

```

* Ícone do Botão
wa_button-icon_id    = icon_error_protocol.

* Texto que aparecerá ao lado do ícone (pode ser vazio)
wa_button-icon_text  = 'Log de Erros'.

* Quickinfo (aparece quando o user passar o mouse sobre o botao)
wa_button-quickinfo  = 'Visualizar Log de Erros'.

* Associa essas propriedades com a função 1
sscrfields-functxt_01 = wa_button.

ENDFORM.                  " cria_botao

```

Você pode atribuir pop-ups nos botões, outros programas através de um submit, mostrar uma SM30, chamar alguma transação com call transaction.. as opções são inúmeras.

A tela de seleção vai ficar assim com o código acima. Se liga ali no botão!

Clicando no botão, o programa cai nesse evento.
Aí você pode colocar a chamada que você quiser!

Continue seguindo os posts para descobrir como fazer uma tela de seleção turbinada. No último post irei mostrar uma tela que una todos os códigos de tela de seleção listados aqui no blog.

Abraços!

Comentários

Danilo Arguelho — 28/07/2022 15:23

perfeição de conteúdo, valeu!

LUIZ CARLOS DE SOUZA — 10/05/2016 15:31

Daria para utilizar ele como Exit-Command ? Por exemplo, ele executar a função dele ignorando os campos obrigatórios de uma tela.

E parabéns pelo trabalho!

rafael — 14/08/2019 16:29

estou tentando fazer isso, você descobriu?

André Pampolin — 16/07/2013 18:00

Mauricio... descobri... é que o meu programa estava com o campo "Banco de dados lógico = PNP", e assim ele sempre é o primeiro botão a aparecer... bastou eu trocar no código onde estava botão 1 para 2 que deu certo...

Valeu.

André Pampolin — 16/07/2013 17:18

Mauricio, fiz o teste do botão acima Ctrl+C Ctrl+V, porém não apareceu o botão... será que tem a ver com a versão do SAP, atualmente estamos na 6.0

Amanda PF — 11/08/2011 18:23

Boa noite Maurício!

Estou iniciando em abap e estou adorando seu blog! Parabéns!

Estou com algumas dúvidas a respeito desse código...

1- "IF sy-ucomm = 'FC01'."

Nessa linha, esse sy-ucomm é por exemplo, uma variável para validar a execução e entrada de dados? Por que no caso do botão seu valor é igual a 'FC01' ?

2- E no código de apagar o botão:

"não entendi o pq da criação desse campo"

FIELD-SYMBOLS: TYPE ANY.

"por que concatenar esses dados?"

CONCATENATE 'SSCRFIELDS-FUNCTXT_0' p_number
INTO v_field.

"aqui não entendi nd rs"

ASSIGN (v_field) TO .

```
IF sy-subrc = 0.
CLEAR .
ENDIF.
```

Obrigado pela atenção! 😊

Henrique Dias — 28/02/2011 07:27

Eu fiz um teste aqui e não precisa usar esse field symbol maroto q vc usou aí no exemplo.

Só isso já resolve:

* Exclui o botão associado ao FUNCTION KEY 1.

```
CLEAR sscrfields-functxt_01.
```

Ah, e se alguém quiser excluir o botão executar é só usar o código abaixo:

* Include de Selection Screen Standard

```
INCLUDE rsdbc1xx.
```

* Exclui o botão Executar da tela de seleção (a estrutura current_scr pode variar dependendo da versão do SAP)

```
APPEND 'ONLI' TO current_scr-excl.
```

raoni — 27/04/2012 16:49

Henrique,
excelente.

Mauricio Cruz — 28/02/2011 07:48

Heouaheouah

Curti o fato de vc ter sido simplista. Na verdade eu vou usar o código ali para colocar num método da ABAPZombie Gear, daí eu fiz de uma vez.

Valeu pelo esquema de excluir o executar 😊

Abraço!

Henrique — 23/02/2011 14:27

Se eu quiser colocar o botão blz, mas e se eu quiser dar um fim nele quando mudar um radio button por exemplo?

Mauricio Cruz — 25/02/2011 10:51

É só limpar a work area SSCRFIELDS para apagar este tipo de botão.

Assim:

```
<pre lang="abap" escaped="true">
```

```
*&_____*
```

```
* & Form APAGA_BOTAO
```

```
* & _____ *
```

```
FORM apaga_botao USING p_number TYPE c.
```

```
* Variáveis
```

```
DATA: v_field(21) TYPE c.
```

```
* Field Symbols
```

```
FIELD-SYMBOLS: <fs_field> TYPE ANY.
```

```
CONCATENATE 'SSCRFIELDS-FUNCTXT_0' p_number  
INTO v_field.
```

```
ASSIGN (v_field) TO <fs_field>.
```

```
IF sy-subrc = 0.
```

```
CLEAR <fs_field>.
```

```
ENDIF.
```

```
ENDFORM. " APAGA_BOTAO
```

```
</pre>
```

É só chamar o comando `PERFORM apaga_botao USING '1'`, se você quiser apagar o botão associado a function key 1, por exemplo. Faça isso no AT SELECTION SCREEN OUTPUT e seja feliz 😊

Abraço!