

SDK - Software Development Kit - v.1.0

## Customização

Added by [Endrigo Antonini](#), last edited by [Endrigo Antonini](#) on Ago 24, 2011 ([view change](#))  
Labels  
(None)

## Objetivo

Ensinar como realizar a customização dos dados que serão exibidos e também armazenados no banco de dados através de uma camada intermediária, assim sendo, podendo ajustar valores dos campos ou até mesmo passar valores que não encontram-se na tabela física através dos chamados campos virtuais.



É aconselhado que seja realizado a leitura também dos documentos:

- [Pontos de UPC](#): Trata referente a como realizar UPC em programas que pertencem a camada de customização;
- [Query's muito extensa](#): Documentação da técnica aplicada para corrigir problemas que estavam sendo gerados devido a execução de uma query maior que o banco de dados suporta;

## Estrutura

O programa destinado a realizar estas customizações é dividido em 4 (quatro) procedures. São elas e seus pontos de execução:

- [afterAllSelectedRecords](#): Esta procedure é executada após ter lido todos os registros da página.
- [afterPersistRecord](#): Esta procedure é executada após realizar qualquer tipo de persistência (CREATE, DELETE, UPDATE).
- [afterSelectedRecord](#): Esta procedure é executada para cada registro que é encontrado na página.
- [beforePersistRecord](#): Esta procedure é executada antes de realizar qualquer tipo de persistência (CREATE, DELETE, UPDATE).
- [prepareQuery](#): Esta procedure é executada antes de realizar a pesquisa dos registros para que se possa customizar a query.

Também disponibiliza algumas FUNCTIONS e PROCEDURES auxiliares para facilitar a utilização. São elas:

- [prepareDatasetFields](#);
- [prepareRowObjectFields](#);
- [getDatasetFieldHandle](#);
- [getRowObjFieldHandle](#);

## Procedures para implementação

### afterAllSelectedRecords

**Objetivo:** Tem por objetivo ser executada após todos os registros (apenas os registros da página corrente) terem sido lidos para que o usuário possa realizar customizações em registros específicos (navegar para um determinado registro, alterar informação de um registro com base no valor de outro, etc)

**Parâmetros:**

Parâmetro	I/O	Tipo	Conteúdo
tableName	Input	Character	Nome da tabela em questão
ttDataset	Input-Output	Handle	Handle da temp-table do dataset

**Exemplo:** O programa abaixo verifica o nome da tabela que está sendo lida para não realizar o processo abaixo para as tabelas de referência. Também lê todos os registros e altera a coluna "teste" com um contador que é incrementado de 9 em 9.

```

*****/
DEFINE INPUT PARAMETER tableName AS HANDLE.
DEFINE INPUT-OUTPUT PARAMETER ttDataset AS HANDLE.

DEFINE VARIABLE queryHandle AS HANDLE NO-UNDO.
DEFINE VARIABLE bufferHandle AS HANDLE NO-UNDO.
DEFINE VARIABLE contador AS INTEGER NO-UNDO.
DEFINE VARIABLE hTeste AS HANDLE NO-UNDO.

IF tableName <> "banco.tabela" THEN DO:
    return "ok":u.
END.

CREATE QUERY queryHandle.

ASSIGN bufferHandle = ttDataset:DEFAULT-BUFFER-HANDLE.

queryHandle:SET-BUFFERS(bufferHandle).
queryHandle:QUERY-PREPARE("for each ttDataset").
queryHandle:QUERY-OPEN().
queryHandle:GET-FIRST(NO-LOCK).

DO WHILE(bufferHandle:AVAILABLE):

    RUN prepareDatasetFields (INPUT-OUTPUT ttDataset).

    ASSIGN hTeste = getDatasetFieldHandle("teste").
    IF VALID-HANDLE(hTeste) THEN DO:
        ASSIGN
            contador = contador + 9
            hTeste:BUFFER-VALUE = contador.
    END.

    queryHandle:GET-NEXT(NO-LOCK).
END.

```

### afterPersistRecord

**Objetivos:** Tem por objetivo ser executada após ter realizado qualquer tipo de persistência (CREATE, UPDATE e DELETE) para que possa ser realizado customizações após a realização da persistência e/ou correções na RowErrors.

**Parâmetros:**

Parâmetro	I/O	Tipo	Conteúdo
pType	Input	Character	"CREATE", "UPDATE" ou "DELETE"
ttDataset	Input-Output	Handle	Handle da temp-table do dataset
RowObject	Input-Output	Handle	Handle da temp-table do RowObject
hBoHandle	Input	Handle	Handle da BO
RowErrors	Input-Output	Table	Tabela RowErrors

**Exemplo:** O programa abaixo altera o valor da coluna "name" da tabela adicionando os valores das colunas "teste" e "teste2" que eram do Dataset para a tabela física.

```

*****/
DEFINE INPUT PARAMETER pType AS CHARACTER NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER ttDataset AS HANDLE.
DEFINE INPUT-OUTPUT PARAMETER RowObject AS HANDLE.
DEFINE INPUT PARAMETER hBoHandle AS HANDLE.
DEFINE INPUT-OUTPUT PARAMETER TABLE FOR RowErrors.

/* captura os campos do Dataset */
RUN prepareDatasetFields (INPUT-OUTPUT ttDataset).
RUN prepareRowObjectFields (INPUT-OUTPUT RowObject).

/*:T--- Utilize o parâmetro pType para identificar quais as validações a serem executadas ---*/
/*:T--- Os valores possíveis para o parâmetro são: Create, Delete e Update ---*/

DEFINE VARIABLE hRNome AS HANDLE NO-UNDO.
DEFINE VARIABLE hDatasetNome AS HANDLE NO-UNDO.
DEFINE VARIABLE hDatasetTeste AS HANDLE NO-UNDO.
DEFINE VARIABLE hDatasetTeste2 AS HANDLE NO-UNDO.

IF pType = "CREATE":U THEN DO:
    hRNome = getRowObjFieldHandle("nome").
    hDatasetNome = getDatasetFieldHandle("nome").
    hDatasetTeste = getDatasetFieldHandle("teste").
    hDatasetTeste2 = getDatasetFieldHandle("teste2").

    IF VALID-HANDLE(hRNome) THEN DO:
        ASSIGN hRNome:BUFFER-VALUE = hRNome:BUFFER-VALUE + " - " + pType.

        IF VALID-HANDLE(hDatasetTeste) THEN
            ASSIGN hRNome:BUFFER-VALUE = hRNome:BUFFER-VALUE + " - DATASET " + hDatasetTeste:BUFFER-VALUE.

        IF VALID-HANDLE(hDatasetTeste2) THEN

```

```

        ASSIGN hRNome:BUFFER-VALUE = hRNome:BUFFER-VALUE + " - DATASET " + hDatasetTeste2:BUFFER-VALUE.
    END.
END.

```

## afterSelectedRecord

**Objetivo:** Tem por objetivo ser executada após cada registro ter sido lido para que o usuário possa realizar customizações no registro (gerar informação de um campo virtual baseado nos valores do registro, validar a informação de uma coluna da tabela, etc).

### Parâmetros:

Parâmetro	I/O	Tipo	Conteúdo
tableName	Input	Character	Nome da tabela em questão
ttDataset	Input-Output	Handle	Handle da temp-table do dataset

**Exemplo:** O programa abaixo verifica o nome da tabela que está sendo lida para não realizar o processo abaixo para as tabelas de referência. E também altera o valor da coluna "teste2" para o valor 15 e também adiciona na coluna "nome" o prefixo "[CUSTOM]".

```

*****/
DEFINE INPUT PARAMETER tableName AS CHARACTER.
DEFINE INPUT-OUTPUT PARAMETER ttDataset AS HANDLE.

DEFINE VARIABLE hDSTeste2      AS HANDLE      NO-UNDO.
DEFINE VARIABLE hDSNome        AS HANDLE      NO-UNDO.

IF tableName <> "banco.tabela" THEN DO:
    return "ok":u.
END.

RUN prepareDatasetFields (INPUT-OUTPUT ttDataset).

hDSTeste2 = getDatasetFieldHandle("teste2").
IF VALID-HANDLE(hDSTeste2) THEN DO:
    hDSTeste2:BUFFER-VALUE = 15.
END.

hDSNome = getDatasetFieldHandle("nome").
IF VALID-HANDLE(hDSNome) THEN DO:
    hDSNome:BUFFER-VALUE = "[CUSTOM] " + hDSNome:BUFFER-VALUE.
END.

```

## beforePersistRecord

**Objetivo:** Tem por objetivo ser executada antes de realizar qualquer tipo de persistência (CREATE, UPDATE e DELETE) para que possa ser realizado a conversão dos valores virtuais para colunas físicas.

### Parâmetros:

Parâmetro	I/O	Tipo	Conteúdo
pType	Input	Character	"CREATE", "UPDATE" ou "DELETE"
ttDataset	Input-Output	Handle	Handle da temp-table do dataset
RowObject	Input-Output	Handle	Handle da temp-table do RowObject
hBoHandle	Input	Handle	Handle da BO

**Exemplo:** O programa abaixo altera o valor da coluna "name" da tabela adicionando os valores das colunas "teste" e "teste2" que eram do Dataset para a tabela física.

```

*****/
DEFINE INPUT PARAMETER pType AS CHARACTER NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER ttDataset AS HANDLE.
DEFINE INPUT-OUTPUT PARAMETER RowObject AS HANDLE.
DEFINE INPUT      PARAMETER hBoHandle AS HANDLE.

/* captura os campos do Dataset */
RUN prepareDatasetFields (INPUT-OUTPUT ttDataset).
RUN prepareRowObjectFields (INPUT-OUTPUT RowObject).

/*:T--- Utilize o parâmetro pType para identificar quais as validações a serem executadas ---*/
/*:T--- Os valores possíveis para o parâmetro são: Create, Delete e Update ---*/

DEFINE VARIABLE hRNome      AS HANDLE      NO-UNDO.
DEFINE VARIABLE hDatasetNome AS HANDLE      NO-UNDO.
DEFINE VARIABLE hDatasetTeste AS HANDLE      NO-UNDO.
DEFINE VARIABLE hDatasetTeste2 AS HANDLE      NO-UNDO.

IF pType = "CREATE":U THEN DO:
    hRNome = getRowObjFieldHandle("nome").

```

```

hDatasetNome = getDatasetFieldHandle("nome").
hDatasetTeste = getDatasetFieldHandle("teste").
hDatasetTeste2 = getDatasetFieldHandle("teste2").

IF VALID-HANDLE(hRNome) THEN DO:
    ASSIGN hRNome:BUFFER-VALUE = hRNome:BUFFER-VALUE + " - " + pType.

    IF VALID-HANDLE(hDatasetTeste) THEN
        ASSIGN hRNome:BUFFER-VALUE = hRNome:BUFFER-VALUE + " - DATASET " + hDatasetTeste:BUFFER-VALUE.

    IF VALID-HANDLE(hDatasetTeste2) THEN
        ASSIGN hRNome:BUFFER-VALUE = hRNome:BUFFER-VALUE + " - DATASET " + hDatasetTeste2:BUFFER-VALUE.

END.
END.

```

## prepareQuery

**Objetivo:** Tem por objetivo disponibilizar um meio de customizar a query que será executada para realizar a busca de dados.

**Parâmetros:**

Parâmetro	I/O	Tipo	Conteúdo
queryHandler	Input-Output	Handle	Handle da Query

**Exemplo:** O programa abaixo altera a query para adicionar a na cláusula a empresa corrente do usuário.

```

*****/

DEFINE INPUT-OUTPUT PARAMETER queryHandle AS HANDLE NO-UNDO.

DEFINE VARIABLE cTemp AS CHARACTER NO-UNDO.
DEFINE VARIABLE queryString AS CHARACTER NO-UNDO.
DEFINE NEW GLOBAL SHARED VARIABLE v_cod_empres_usuar AS
    CHARACTER FORMAT "x(3)":U LABEL "Empresa"
    COLUMN-LABEL "Empresa" NO-UNDO.

queryString = queryHandle:PREPARE-STRING.

ASSIGN cTemp = " WHERE ".

IF INDEX(queryString, "WHERE") > 0 THEN
    ASSIGN cTemp = "AND".

queryString = queryString + cTemp +
    " tabela.cod_empresa = '" + v_cod_empres_usuar + "'".

queryHandle:QUERY-PREPARE(queryString).

```

## Functions e Procedures auxiliares

### prepareDatasetFields

**Tipo:** PROCEDURE

**Objetivo:** Esta PROCEDURE realiza o mapeamento de todos os campos que vieram no Dataset para que possa facilitar a forma de trabalhar com estas handles. Deve ser utilizada junto com a FUNCTION [getDatasetFieldHandle](#).

**Parâmetros:**

Parâmetro	I/O	Tipo	Conteúdo
ttDataset	Input-Output	Handle	Handle da temp-table do dataset

**Exemplo:** Vide exemplos das procedures [afterAllSelectedRecords](#), [afterSelectedRecord](#) e [beforePersistRecord](#).

### prepareRowObjectFields

**Tipo:** PROCEDURE

**Objetivo:** Esta PROCEDURE realiza o mapeamento de todos os campos que vieram do RowObject para que possa facilitar a forma de trabalhar com estas handles. Deve ser utilizada junto com a FUNCTION [getRowObjFieldHandle](#).

**Parâmetros:**

Parâmetro	I/O	Tipo	Conteúdo
RowObject	Input-Output	Handle	Handle da temp-table do RowObject

**Exemplo:** Vide exemplos das procedures [afterAllSelectedRecords](#), [afterSelectedRecord](#) e [beforePersistRecord](#).

## getDatasetFieldHandle

**Tipo:** FUNCTION

**Objetivo:** Esta FUNCTION tem por objetivo retornar a handle de um determinado campo do dataset. Sua utilização está vinculada ao uso correto da procedure [prepareDatasetFields](#).

**Parâmetros:**

Parâmetro	I/O	Tipo	Conteúdo
pName	Input	Character	Nome do campo que deseja buscar
retorno	Return	Handle	Handle do campo desejado

**Exemplo:** Vide exemplos das procedures [afterAllSelectedRecords](#), [afterSelectedRecord](#) e [beforePersistRecord](#).

## getRowObjFieldHandle

**Tipo:** FUNCTION

**Objetivo:** Esta FUNCTION tem por objetivo retornar a handle de um determinado campo da RowObject. Sua utilização está vinculada ao uso correto da procedure [prepareRowObjectFields](#).

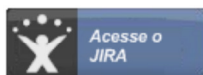
**Parâmetros:**

Parâmetro	I/O	Tipo	Conteúdo
pName	Input	Character	Nome do campo que deseja buscar
retorno	Return	Handle	Handle do campo desejado

**Exemplo:** Vide exemplo da procedure [beforePersistRecord](#).

### Informações sobre este Documento

Versão	Estado	Compatibilidade	Criador	Criação	Último Revisor	Última Revisão
<a href="#">26</a>		\$param0	<a href="#">Endrigo Antonini</a>	Ago 23, 2011 15:13	<a href="#">Endrigo Antonini</a>	Ago 24, 2011 17:56



Your Rating:

Results:

1 rates

[O que é RSS?](#)

**Children (2)** [Hide Children](#) | [View in Hierarchy](#)

- [Pontos de UPC](#)
- [Query's muito extensa](#)

[Add Comment](#)