

[Home](#)

Específicos

Flex

Getting Started

Java

Processo Datasul

Progress

Tools

Trilhas

User

[Page Operations](#)

[Browse Space](#)

SDK - Software Development Kit - v.1.0

Como utilizar paginação em um FreeForm?

Added by [Diego Silvio Zimmermann](#), last edited by [Tajana Maira Werlich](#) on Jun 01, 2011 ([view change](#))

Labels
(None)

Para utilizar paginação em um FreeForm, deve ser adicionado o componente **GridContainer**, que automaticamente já adicionar o componente DataGrid juntamente à ele.

Para informar a quantidade de registros que devem ser mostrados por páginas, deve ser preenchida a propriedade **pageSize** do componente GridContainer. O valor atribuído como default é 100.

Abaixo segue os exemplos de códigos ABLScript's que devem ser adicionados para o funcionamento da paginação:

No evento de **creationComplete** e no **pageChange** do container principal, deve ser adicionada a seguinte lógica:

```

/*Definição das variáveis*/
DEFINE VARIABLE grid AS WIDGET-HANDLE.
DEFINE VARIABLE gridContainer AS WIDGET-HANDLE.
DEFINE VARIABLE vHandle AS HANDLE.
DEFINE VARIABLE currentPage AS INTEGER.
DEFINE VARIABLE pageSize AS INTEGER.
DEFINE TEMP-TABLE ttPaginada
    FIELD codigo AS INTEGER LABEL "Código"
    FIELD nome AS CHARACTER LABEL "Nome"
    FIELD obs AS CHARACTER LABEL "Observação".

DEFINE VARIABLE totalRecord AS INTEGER.

/*Captura do componente GridContainer a página atual*/
ASSIGN currentPage = GETPROPERTY(gridContainer, "currentPage").

/*Captura do componente GridContainer a quantidade de registros que serão mostrados por página*/
ASSIGN pageSize = GETPROPERTY(gridContainer, "pageSize").

PROCESS-WAIT(true).

/*Executa a procedure responsável por retornar os registros paginados e o seu total*/
RUN testes/retornaPaginado.p PERSISTENT SET vHandle.
RUN retornaRegistros IN vHandle (INPUT currentPage, INPUT pageSize, OUTPUT totalRecord, OUTPUT ttPaginada).

PROCESS-WAIT(false).

/*Popula o grid*/
grid:QUERY-OPEN(ttPaginada).

/*Atribue o número de registros ao componente GridContainer*/
SETPROPERTY(gridContainer, "totalRecords", totalRecord).
    
```

Atenção

O tratamento de paginação será feito na API Progress que o desenvolvedor utilizar. Utilizar a técnica de **QUERY-OPEN** e **REPOSITION-TO-ROW**. Abaixo segue um exemplo do funcionamento da paginação e anexo está o programa para testes.

```

/* Parametros fornecidos pelo Programa chamador */
DEFINE VARIABLE iCurrentPage AS INTEGER NO-UNDO.
DEFINE VARIABLE iPageSize AS INTEGER NO-UNDO.

DEFINE VARIABLE iInitialPage AS INTEGER NO-UNDO.

DEFINE TEMP-TABLE ttReturn LIKE mguni.pais.

/* Verifica qual será o primeiro registro da próxima pagina*/
ASSIGN iInitialPage = 1 + (iCurrentPage - 1) * iPageSize.

RUN retornaRegistro IN THIS-PROCEDURE.

PROCEDURE retornaRegistro:

    DEFINE VARIABLE qhTeste AS HANDLE NO-UNDO.
    DEFINE VARIABLE hTTReturn AS HANDLE NO-UNDO.
    DEFINE VARIABLE hTable AS HANDLE NO-UNDO.
    DEFINE VARIABLE hbTTReturn AS HANDLE NO-UNDO.
    DEFINE VARIABLE qhTemp AS HANDLE NO-UNDO.
    DEFINE VARIABLE hField AS HANDLE NO-UNDO.
    
```

```

DEFINE VARIABLE i          AS INTEGER NO-UNDO.
DEFINE VARIABLE j          AS INTEGER      NO-UNDO.

ASSIGN hTable = BUFFER mguni.pais:HANDLE.

ASSIGN hTTReturn = TEMP-TABLE ttReturn:HANDLE.
hbTTReturn = hTTReturn:DEFAULT-BUFFER-HANDLE.

CREATE QUERY qhTeste.
qhTeste:SET-BUFFERS(hTable).
qhTeste:QUERY-PREPARE("FOR EACH pais" ).
qhTeste:QUERY-OPEN.

/*Posiciona no registro previamente calculado */
qhTeste:REPOSITION-TO-ROW(iInitialPage).

REPEAT i = 1 TO iPageSize:
  qhTeste:GET-NEXT().
  IF qhTeste:QUERY-OFF-END THEN LEAVE.

  hbTTReturn:BUFFER-CREATE().
  hbTTReturn:BUFFER-COPY(hTable).
  hbTTReturn:BUFFER-RELEASE().

END.
qhTeste:QUERY-CLOSE().
DELETE OBJECT qhTeste.

IF VALID-HANDLE(qhTemp) THEN
  DELETE OBJECT qhTemp.
IF VALID-HANDLE(hTTReturn) THEN
  DELETE OBJECT hTTReturn.
IF VALID-HANDLE(hTable) THEN
  DELETE OBJECT hTable.
IF VALID-HANDLE(hbTTReturn) THEN
  DELETE OBJECT hbTTReturn.

END.

```

 [Add Comment](#)