

# PEX - Criar arquivo (Excel, TXT)

Versão homologada: 12.10.5

## Criando arquivo Excel:

Iremos focar mais no PEX neste exemplo. O processo contém apenas dois formulários

- Grade de dados:
  - Coluna 1 "Cliente";
  - Coluna 2 "Valor"
- Botão: Evento PEX que irá gerar o Excel

image1640624301764.png

A grade será carregada com um SELECT utilizando o evento de **Estrutura de repetição - Carregar** ao entrar, com o seguintes SQL:

```
SELECT CLIFOREMP. RAZAO
        CLIENTE,
        SUM( RECEBER. VALORSALDO) VALOR
FROM RECEBER
JOIN CLIFOREMP ON ( CLIFOREMP. IDCLIFOREMP = RECEBER. IDCLIFOREMP)
WHERE ROWNUM <= 15
        AND RECEBER. VALORSALDO > 0
GROUP BY CLIFOREMP. RAZAO
HAVING SUM( RECEBER. VALORSALDO) > 0
ORDER BY 2 DESC
```

image1640625041311.png

Será necessário o formulário do tipo botão pois é necessário acessar algumas funcionalidades da grade, e na própria grade ao sair não é possível.

Além do formulário do tipo botão é necessário criar um arquivo modelo que o sistema irá se basear para criar o novo arquivo com as informações que a grade estará preenchida, crie uma pasta Temp no C: com esse arquivo modelo.

image-1640625316311.png

Image not found or type unknown

PEX ao sair do botão:

```
Const
    // Abaixo está os formatos da células que você poderá utilizar.
    csMascaraFormatoInteiro : String = '#0;[RED]-#0';
    csMascaraFormatoFloat : String = '#,##0.00';
    csMascaraSemFormato : String = '#0';

    csMascaraData : String = 'DDD DD/MM/YY';
    csMascaraDataSimples : String = 'DD/MM/YYYY';

    // Neste código é definido qual caminho do Arquivo modelo que o sistema irá se basear para
    a criação do novo arquivo
    csCaminhoArquivoModelo : String = 'C:\TEMP\MODELO.XLS';

var
    // Criando as variáveis necessárias para o funcionamento
    liIdRegistro: Integer;
    loRegistro: TJSONObject;
    loCDS : TLibCDS;
    loPlanilha : TEMAPlanilha;
    lsNomeArquivo,
    lsColuna : String;
    liLinha : Integer;
    loStringList : TStringList;

begin

    // liLinha é a linha que o sistema irá começar a preencher as informações, como na primeira
    definimos o cabeçalho iniciaremos os dados na segunda.
    liLinha := 2;
    csCaminhoArquivoModelo := 'C:\TEMP\MODELO.XLS';

    //lsNomeArquivo é aonde o arquivo será salvo e com nome definido.
    lsNomeArquivo := 'c:\temp\processo.xls';
    loPlanilha := nil;
```

```

// Neste comando abaixo ele irá extrair o nome do arquivo para logo abaixo fazer uma
validação se o arquivo existe ou não.

lsNomeArquivo := ExtractFilePath(lsNomeArquivo) +
                StringReplace(ExtractFileName(lsNomeArquivo), '/', ' ',
[rfrReplaceAll]);

// Valida se o arquivo existe, caso exista ele deleta o arquivo para criar com as novas
informações.
if FileExists(lsNomeArquivo) then
    DeleteFile(lsNomeArquivo);

// Se o arquivo não existe ele cria no caminho indicado.
if not FileExists(ExtractFilePath(lsNomeArquivo)) then
    ForceDirectories(ExtractFilePath(lsNomeArquivo));
    TARQ.of_CopyArquivoXXX(csCaminhoArquivoModelo, lsNomeArquivo,
ExtractFileExt(lsNomeArquivo));

//Começa a criar o arquivo.
loPlanilha := TEmaPlanilha.Create(nil);
loPlanilha.of_ArquivoAbrir(lsNomeArquivo);

//Local aonde é definido o nome da aba da planilha.
loPlanilha.of_PastaSelecionar(' PROCESSO'); //Nome da ABA no Excel

// Começa a estrutura de repetição para cada registro da grade ser inserida.
for liIdRegistro := 0 to Pred(aoFormularios.GetJSON('1').GetArrayJSON(' DADOS').Count) do
begin
    //Pega o id do registro que ele está selecionando
    loRegistro :=
aoFormularios.GetJSON('1').GetArrayJSON(' DADOS').GetItemAsJson(liIdRegistro);

    // Cliente -> É informado qual célula é iniciado, neste caso está sendo concatenado
A2, logo após é informado o registro da grade que será preenchido
    // após é definido o nil da célula e o formato da célula
    loPlanilha.op_Cell(' A' + IntToStr(liLinha),
                        loRegistro.GetStr(' CLIENTE'),
                        nil,
                        csMascaraFormatoInteiro);

    // Valor

```

```

        loPlanilha.op_Cell(' B' + IntToStr(liLinha),
                           loRegistro.GetStr(' VALOR' ),
                           nil,
                           csMascaraSemFormato);

        // Incrementa na
linha.
        Inc(liLinha);
    end;
    // Salva a planilha.
    loPlanilha.of_ArquivoSalvar;
end;

```

## Resultado final:

image-1640625520013.png

Image not found or type unknown

## Criando arquivo TXT:

Um dos casos de uso dessa função seria guardar informações tais como observações, texto que venham junto ao um anexo além de salvar arquivos e imagens via evento também criando um arquivo .TXT

Segue o código abaixo, colocando no botão ou ao sair de algum outro formulário.

```

var
    loArquivo : TStringList;
    lsArquivo,
    lsSQL      : String;
    loCDS      : TLibCDS;
begin
    loArquivo := nil;
    lsSQL := ' SELECT * FROM CRM_PROCESSO WHERE IDPROCESSO = ' +
TSTR.AsPa(aoFormularios.GetJSON(' 1' ).GetStr(' TEXTO' ));
    loCDS := of_CriaCDSporSQL( lsSQL);

    if  aoFormularios.GetJSON(' 1' ).GetInt(' TEXTO' ) > 0 then
    begin
        try
            loArquivo := TStringList.Create;
            lsArquivo := ' C:\Teste ema\IDPROCESSO_' + aoFormularios.GetJSON(' 1' ).GetStr(' TEXTO' )

```

```

+ '.txt';          // <--- Pasta com o nome do Anexo que será criado

    loCDS.of_IniciaWhile; // Cada loArquivo.add é uma linha adicionada no txt
    loArquivo.Add(' Dados Exportados' );
    loArquivo.Add(' IDPROCESSO: ' + loCDS.GetStr([' IDPROCESSO' ]));
    loArquivo.Add(' DESCRIÇÃO: ' + loCDS.GetStr([' DESCRICAO' ]));
    loArquivo.Add(' ASSUNTO: ' + loCDS.GetStr([' ASSUNTO' ]));
    loArquivo.Add(' USUARIO: ' + loCDS.GetStr([' USUARIO' ]));
    loCDS.of_Primeiro;
    while not loCDS.of_FimDS do
    begin
        loCDS.of_ProxReg;          // <--- Obrigatorio para percorrer os proximos registros
caso tenha para não entrar em looping
        end;
        aoMensagem.SetStr(' MENSAGEM' , ' Anexo Exportado' );
        aoMensagem.SetStr(' TIPO' , ' INFO' );
        aoMensagem.SetInt(' TIMEOUT' , 5000{Milissegundos});
    finally
        loCDS.Free;
        loArquivo.SaveToFile(lsArquivo); // Salva o arquivo.
        loCDS.of_FinalizaWhile;
    end;
end;
end;
end;

```

---

Revisão #5

Criado 27 December 2021 13:45:30 por Nicolly Andrielly

Atualizado 9 May 2022 16:12:03 por Nicolly Andrielly