

Referência LibreOffice

LibreOffice Basic

Calc

Documentos do LibreOffice

Documento atual

```
Dim Doc As Object
Doc = ThisComponent
```

Open another existing document

Modo visível

```
Dim Doc As Object
Dim PathDoc As String
Dim Props() 'aqui, esta tabela não está inicializada
PathDoc = ConvertToURL("C:\Path\To\CalcFile.ods")
Doc = StarDesktop.loadComponentFromURL(PathDoc, "_blank", 0, Props(0))
```

Modo invisível

```
Dim Doc As Object
Dim PathDoc As String
Dim Props(0) As New com.sun.star.beans.PropertyValue
PathDoc = ConvertToURL("C:\Path\To\CalcFile.ods")
Props(0).Name = "Hidden" 'o documento vai abrir escondido "
Props(0).Value = True
Doc = StarDesktop.loadComponentFromURL(PathDoc, "_blank", 0, Props(0))
```

Tornar visível a posteriori

```
Doc.CurrentController.Frame.ContainerWindow.Visible = True
Doc.CurrentController.Frame.ContainerWindowToFront()
```

Criando um novo documento do Calc

De (1) o modelo padrão ou (2) um modelo específico.

```
Dim Doc As Object
Dim Props() 'aqui, esta tabela não está inicializada
Model = "private:factory/scalc" '(1)
'or
Model = "C:\Path\To\ACalcTemplateFile.ots" '(2)
Doc = StarDesktop.loadComponentFromURL(Model, "_blank", 0, Props())
```

Salvar um documento

O documento já existe

(equivalente a Arquivo > **Salvar**)

Use o armazenamento de métodos do objeto de documento. EX: `ThisComponent.store`

O documento ainda não foi salvo (equivalente a **Arquivo > Salvar como**)

```
Dim Doc As Object 'o documento objeto a ser armazenado
Dim PathDoc As String 'o caminho para salvar
Dim Props() 'as propriedades de salvamento. (esvaziar)
PathDoc = ConvertToURL("C:\Path\To\CalcFile.ods")
Doc.storeAsURL(PathDoc, Props())
```

☞ Se uma cópia, ele se **transforma** em documento ativo

Salvar uma cópia...

Como acima, mas com `Doc.storeToURL(PathDoc, Props())`

☞ A cópia **não** se torna o documento ativo.

Fechar um documento

Use o método `close` do objeto do documento: `ThisComponent.close(True)`

Informação do documento

O objeto de documento expor propriedades

Location	A pasta do documento. ☞ Cadeia vazia ainda não está salva
DocumentProperties (Object)	Propriedades adicionais (abaixo).

Propriedades do documento

Author	Nome do autor	ModifyDate	Data última modificação
CreationDate	Data da criação.	Subject	Assunto do documento (string).
Description	Descrição do documento	Title	Título do documento
ModifiedBy	Nome do usuário que modificou o documento.	UserDefinedProperties	Propriedades personalizadas (Object).

É um documento do Calc?

O objeto `Doc` aponta para o documento (ex: `Doc=ThisComponent`).

```
CalcOK = Doc.SupportsService("com.sun.star.sheet.SpreadsheetDocument")
```

Calc - Funcionalidades gerais

O objeto `Doc` aponta para o documento (ex: `Doc=ThisComponent`).

Cálculo Automático

Ativo? (Boolean)	<code>Auto = Doc.isAutomaticCalculationEnabled</code>
Desativar	<code>Doc.enableAutomaticCalculation(False)</code>
Ativar	<code>Doc.enableAutomaticCalculation(True)</code>
Recalculo forçado	<code>Doc.calculate</code> (apenas p/ fórmulas não atualizadas)
	<code>Doc.calculateAll</code> (todas as fórmulas)

Proteger documento

O documento está protegido?	<code>Test = Doc.isProtected</code>
Proteger documento	<code>Doc.protect(password)</code> [senha pode estar vazia]
Desproteger documento	<code>Doc.unprotect(password)</code>

Planilhas (Sheets)

O objeto Doc aponta para o documento (ex: Doc=ThisComponent).

Acesso a planilhas

Trabalhar com objetos Planilhas:

Planilha ativa	<code>MySheet = Doc.CurrentController.ActiveSheet</code>
Listar planilhas	<code>AllSheets = Doc.Sheets</code>
Número de planilhas	<code>NumberSheets = Doc.Sheets.Count</code>
Objeto de planilha (pelo índice [base 0])	<code>MySheet = Doc.Sheets(index)</code>
Objeto de planilha (por nome)	<code>MySheet = Doc.Sheets.getByName("SheetName")</code>
Verificar existência (nome)	<code>Exist = Doc.Sheets.hasByName("SheetName")</code>
Índice de planilha	<code>Index = MySheet.RangeAddress.Sheet</code>

Modificar planilhas

Adicione planilha chamada Name e posição p (base 0)	<code>Doc.Sheets.insertNewByName(Name, p)</code>
Excluir uma planilha	<code>Doc.Sheets.removeByName("SheetName")</code>
Duplicar uma planilha para a posição p (base 0)	<code>Doc.Sheets.copyByName("SourceName", "TargetName", p)</code>
Mover planilha para a posição p (base 0)	<code>Doc.Sheets.moveByName(SheetName, p)</code>

Gerenciar planilhas

MySheet é um objeto de pasta.

Ativar planilha	<code>Doc.CurrentController.ActiveSheet = MySheet</code>
Proteger planilha (senha pode estar vazia)	<code>MySheet.protect(password)</code>
Desproteger planilha	<code>MySheet.unprotect(password)</code>
Cor da guia	<code>MySheet.tabColor = RGB(255, 255, 0)</code>

Vincular uma folha

Link para um arquivo (ex: CSV)	<code>MySheet.link(URL, "", "Text - txt - csv (StarCalc)", _ Filter, com.sun.star.sheet.SheetLinkMode.VALUE)</code>
Quebrar um link	<code>MySheet.setLinkMode(com.sun.star.sheet.SheetLinkMode.NONE)</code>

Encontrar a última linha / coluna usada

MySheet é o objeto de pasta a ser explorado. Row e Col são informações para buscar.

<code>Dim Cur As Object' cursor na célula</code>
<code>Dim Range As Object' o alcance usado</code>
<code>Dim Row As Long</code>
<code>Dim Col As Long</code>
<code>Cur = MySheet.createCursorByRange(MySheet.getCellRangeByName("A1"))</code>
<code>Cur.gotoEndOfUsedArea(True)</code>
<code>Range = MySheet.getCellRangeByName(Cur.AbsoluteName)</code>
<code>Row = Range.RangeAddress.EndRow</code>
<code>Col = Range.RangeAddress.EndColumn</code>

Células (cells)

Abaixo Cel é um objeto de célula.

Acesso às células

MySheet é um objeto de pasta. Acesso ao objeto da célula:

Por notação padrão de célula	<code>Cel = MySheet.getCellRangeByName("A4")</code>
Por nome	<code>Cel = MySheet.getCellRangeByName("TVA")</code>
Por coordenadas X e Y	<code>Cel = MySheet.getCellByPosition(0,3)</code> Wih X=0 (col.A) ; Y=3 (row.4)

Acesso a célula ativa

Doc é um objeto de documento e ActiveCel o objeto de célula ativa.

```
If Doc.currentSelection.supportsService _
("com.sun.star.sheet.SheetCell") Then
'É uma célula
ActiveCel = Doc.currentSelection
End If
```

Selecione uma célula

`ThisComponent.CurrentController.select(Cel)`

Coordenadas de células

Coordenadas (Object)	<code>Coord = Cel.CellAddress</code>
Índice de planilha (Integer)	<code>NumS = Cel.CellAddress.Sheet</code>
Índice de colunas (Long)	<code>NumC = Cel.CellAddress.Column</code>
Índice de linhas (Long)	<code>NumL = Cel.CellAddress.Row</code>
Objeto de recipiente de planilha	<code>MySheet = Cel.Spreadsheet</code>
Coordenadas Absolutas (String)	<code>Coord = Cel.AbsoluteName</code>

(des) proteger as células

`Cel.CellProtection` can take boolean values:

Impedir modificação	<code>CellProtection.IsLocked = True</code>
Esconder a fórmula da célula	<code>CellProtection.IsFormulaHidden = True</code>
Esconder célula	<code>CellProtection.IsHidden = True</code>
Não imprimir célula	<code>CellProtection.IsPrintHidden = True</code>

Acesse o conteúdo da célula

Propriedades

Ler o conteúdo do texto	<code>MyText = Cel.String</code>
Ler o conteúdo numérico	<code>aNumber = Cel.Value</code>
Fórmula de célula lida (nomes en-US)	<code>TheFormula = Cel.Formula</code>
Fórmula de célula de leitura (nomes localizados)	<code>LaFormule = Cel.FormulaLocal</code>
Tipo de célula	<code>TheType = Cel.Type</code>
Esvazie uma célula	<code>Cel.String = ""</code>

Tipo de conteúdo (Type property)

As constantes `com.sun.star.table.CellContentType.XXX` representam o tipo de informação da célula (`Cel.Type`, above) :

EMPTY	Célula vazial	VALUE	Valor numerico
TEXT	Conteúdo de texto	FORMULA	Conteúdo de fórmula

Escrever em uma célula

Substituir texto existente	<code>Cel.String = "Hello !"</code>
Substituir um valor existente	<code>Cel.Value = 1.234</code>
Substituir uma fórmula existente (localizada)	<code>Cel.Formula = "=AND(A1="YES";A2="OK")"</code>

Intervalos (ranges)

Intervalo = conjunto de células, (incluindo uma única): `Dim MyRange As Object`

Acesso a intervalos

`MySheet` é um objeto de pasta. Obter um objeto de intervalo `Ran`>:

Por notação padrão de célula	<code>Ran = MySheet.getCellRangeByName("C2:G14")</code>
Por nome	<code>Ran = MySheet.getCellRangeByName("RangeName")</code>
Por coordenadas (X1, Y1, X2, Y2)	<code>Ran = MySheet.getCellRangeByPosition(2, 1, 6, 13)</code>
Aleatoriamente (ex terceira planilha)	<code>Ran = ThisComponent.Sheets.getCellRangeByPosition(2, 2, 1, 6, 13)</code>

Intervalo ativo

Como célula ativa, mas verifique `"com.sun.star.sheet.SheetCellRange"` or `"[...]SheetCellRanges"`.

Seleção de intervalo

`ThisComponent.CurrentController.select(MyRange)` where `MyRange` is an object.

Coordenadas de alcance

Coordenadas (Object)	<code>Coord = MyRange.RangeAddress</code>
Índice de planilhas (Integer)	<code>Ran = MyRange.RangeAddress.Sheet</code>
Classificação da Coluna (Long) canto superior esquerdo	<code>NumCHG = MyRange.RangeAddress.StartColumn</code>
Classificação da linha (Long) canto superior esquerdo	<code>NumLHG = MyRange.RangeAddress.StartRow</code>
Classificação da coluna (Long) canto inferior direito	<code>NumCBD = MyRange.RangeAddress.EndColumn</code>
Classificação da linha (Long) canto inferior direito	<code>NumLBD = MyRow.RangeAddress.EndRow</code>
Objeto de recipiente de planilha	<code>MySheet = MyRange.Spreadsheet</code>
Coordenadas Absolutas (String)	<code>Coord = MyRange.AbsoluteName</code>

Intervalos nomeados

O objeto `Doc` aponta para o documento. Com `Dim TheRanges As Object`

Intervalos nomeados	<code>TheRanges = Doc.NamedRanges</code>
Número (Long)	<code>Nb = TheRanges.Count</code>
Obter um intervalo (por índice)	<code>MyRange = TheRanges(index)</code>
Verificar a existência (nome)	<code>Exist = TheRanges.hasByName("RangeName")</code>
Obter intervalo (por nome)	<code>MyRange = TheRanges.getByName("RangeName")</code>
Adicionar	
<code>Coord</code> : coordenadas de intervalo	<code>TheRanges.addNewByName("Rangename", Coord, _</code>
<code>CellRef</code> : objeto de célula de referência	<code>CellRef.CellAddress, 0)</code>
Excluir (por nome)	<code>TheRanges.removeByName("RangeName")</code>

Apagar um intervalo

Apagar o conteúdo `MyRange` `MyRange.ClearContents(EraseMode)`

`EraseMode` é um valor que define o tipo de limpeza. Usar `com.sun.star.sheet.CellFlags.XXX` e combine-os com +):

ANNOTATION	Comentários	STRING	Texto
DATETIME	Números formatados data/hora	VALUE	Numeros (exeto data/hora)
FORMULA	Fórmula		

Obter o conteúdo da célula em um intervalo

`MyRange.DataArray` é uma tabela de valores de células para `MyRange`.

Copiar o conteúdo do intervalo em outro intervalo

Tem 2 intervalos `Source`(origem) e `Target`(destino), com as mesmas dimensões.

Copia o conteúdo (valores) de <code>Source</code> (origem) para <code>Target</code> (destino).	<code>Target.DataArray = Source.DataArray</code>
---	--

Escrever valores em um intervalo

`MyRange` é um objeto de intervalo `MyTable` uma tabela, **com as mesmas dimensões**, onde os valores devem ser transferidos para o intervalo.

```
Dim MyTable As Variant
MyTable = MyRange.DataArray 'MyTable toma as dimensões do intervalo
' (dar valores aos elementos da tabela)
MyRange.DataArray = MyTable
```

`.DataArray` é uma tabela incorporada: use `.DataArray(i)(j)`

Atravessar células em um intervalo

De uma coleção (`MyRanges.Cells`) criar uma enumeração. Atravessar o intervalo chamando suas propriedades `hasMoreElements` and `NextElement`:

```
Dim Plages As Object
MyRanges = ThisComponent.CreateInstance("com.sun.star.sheet.SheetCellRanges")
MyRanges.InsertByName("", MyRange)
LEnum = MyRanges.Cells.CreateEnumeration
Do While LEnum.hasMoreElements
MyCell = LEnum.NextElement
' aplicar instruções ao objeto celular
Loop
```

células vazias não são atravessadas !

Intervalos: Diversos

Mesclar células de `MyRange` `MyRange.Merge`

Tipos de intervalo

Dependendo do modo de acesso, os intervalos implementam um desses serviços:

① <code>com.sun.star.sheet.SheetCell</code>	④ <code>com.sun.star.sheet.SheetCellRange</code>
② <code>com.sun.star.table.CellRange</code>	⑤ <code>com.sun.star.sheet.SheetCellRanges</code>
③ <code>com.sun.star.sheet.NamedRange</code>	

Dependendo do serviço implementado, os intervalos devem ser empregados de maneira diferente. Teste através do método `supportsService()` (ex. abaixo)

Célula ou alcance?

Para conhecer o tipo de objeto, teste `supportsService()` com `service_name` `service_name` abaixo (boolean) no objeto (intervalo ou célula):

```
If MyObj.supportsService(service_name) Then ...
```

Replace `service_name` by :

Célula ?	"com.sun.star.sheet.SheetCell" ①
Intervalo simples ?	"com.sun.star.sheet.SheetCellRange" ④
Multiplo Intervalo ?	"com.sun.star.sheet.SheetCellRanges" ⑤

Sempre teste uma célula **antes** de um intervalo único, porque uma célula **também** é um intervalo único!

Linhas / Colunas (rows/columns)

Linhas e colunas são propriedades de objetos Planilha e intervalo.

Geral

Linhas (TheRows object)	TheRows = MyRange.Rows
Colunas (TheCols object)	TheCols = MyRange.Columns
Contando	NbL = MyRange.Rows.Count NbC = MyRange.Columns.Count
Uma linha (TheRow object (base 0))	TheRow = MyRange.Rows(index)
Uma coluna (TheCol object (base 0))	TheCol = MyRange.Columns(index)

Row/Columns properties

Aplica-se a linha ou linhas (resp. Coluna ou colunas).

Visível / Oculto (Boolean)	IsVisible = True
Largura ótima (Boolean)	OptimalWidth = True

Inserir / excluir - linhas / colunas

Defina o object `RorC`, e `FirstPos` e `LastPos` as posições do início e fim do conjunto de linhas (ou colunas) para adicionar/excluir (Long).

Inserir	<code>RorC.insertByIndex(FirstPos, LastPos)</code>
Excluir	<code>RorC.removeByIndex(FirstPos, LastPos)</code>

Congelar linhas / colunas

Use o objeto `Controller:MyController = ThisComponent.CurrentController`

Existe um?	<code>Freeze = MyController.hasFrozenPanels</code>
Congelar	<code>MyController.freezeAtPosition(1, 2)</code>
Excluir	<code>MyController.freezeAtPosition(0, 0)</code>

Chame uma função Calc

Use serviço "com.sun.star.sheet.FunctionAccess"

Uso

```
Dim FCalc As Object
Dim Result As (context dependent)
Dim Params As (context dependent)
Dim FunctionName As String
FCalc = CreateUnoService("com.sun.star.sheet.FunctionAccess")
Results = FCalc.callFunction(FunctionName, Params)
```

☞ Nome da função, parâmetros e tipo de resultados dependem da função selecionada
O nome da função deve ser seu nome em **inglês**.

Para obter a função Nome em inglês, mude temporariamente para a exibição de nomes de função do English Calc, em **Ferramentas> Opções> LibreOffice Calc> Fórmula, Usar nomes de funções em inglês**.

Exemplo 1 (function SOMA())

```
Dim FCalc As Object
Dim Results As Long
FCalc = CreateUnoService("com.sun.star.sheet.FunctionAccess")
Results = FCalc.callFunction("SUM", Array(1, 55, 321, 8))
```

Exemplo 2 (function ALEATÓRIO())

```
Dim FCalc As Object
Dim Results As Double
Dim Params(1) As Variant
Params(0) = 1,2345 'number to round
Params(1) = 3      '3 places
FCalc = CreateUnoService("com.sun.star.sheet.FunctionAccess")
Results = FCalc.callFunction("ROUND", Params())
```

Criar uma função Calc

Criar

Exemplo : calcular a área de um trapézio ($S = ((B + b) / 2) \times H$)

```
Function AreaTrapezio(GB As Double, PB As Double, H As Double) As Double
AreaTrapezio = ((GB + PB) / 2) * H
End Function
```

Uso no Calc

Se A2 é a base grande, A3 a base pequena e A4 a altura, a área do trapézio é obtida inserindo a seguinte fórmula em uma célula: =AREATRAPEZIO(A2;A3;A4)

- ☞ A macro recebe os **valores** dos argumentos e não o objeto da célula.
- ☞ A macro **retorna um valor**. **Não se aplica** a uma célula.
- ☞ ☞ A função deve estar localizada em uma biblioteca acessível em tempo de execução (e.g. Standard biblioteca do documento ou usuário) (caso contrário, produzirá o erro #VALUE! error).

Creditos

Author : Jean-François Nifenecker – jean-francois.nifenecker@laposte.net

Tradução para pt_br: Gilberto Schiavinatto – gilberto@schiavinatto.com

Baseado na versão 1.11 de 13/01/2018 / Tradução: 02/04/2019