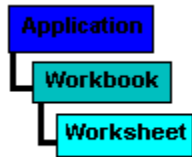


Hierarchy

JVExcel Components Hierarchy



An Application component can have multiple workbooks (not tested and will need methods/properties to switch between workbooks). And a workbook can have multiple worksheet components.



TExcelApp - Excel Application Component

[Hierarchy](#)

[Properties](#)

[Methods](#)

This component is the OLE object that controls the excel application. Place this component on a form that will have the same or greater life span desired for the Excel Application. The excel application will automatically close when this component (or the form that its on) is destroyed.



TExcelBook - Excel Workbook Component

[Hierarchy](#)

[Properties](#)

[Methods](#)

This component represents the Excel File that is being utilized. It must be assigned to a TExcelApp component. Although conceptually it would be possible to setup up multiple workbooks per TExcelApp component that has not been tested.

TExcelSheet - Excel Worksheet Component

[Hierarchy](#)

[Properties](#)

[Methods](#)

This component represents the Excel worksheet within a workbook.

TExcelApp Properties

ExcelStarted

ExcelShowAlerts

Visible

TExcelApp.ShowExcelAlerts

property ShowExcelAlerts : [Boolean](#) ;

Description

This property determines whether to display the Excel prompts or to suppress the Excel prompts such as “*Do you want to save this file before closing Excel?*”. When this property is set to true all Excel prompts will be shown.

Caution. The prompt may be hidden behind your application form. If this property is set to false the prompt will be suppressed and the default Excel push button will be executed.

TExcelApp.ExcelStarted

property ExcelStarted : [Boolean](#) ;

Description

This property indicates whether the Excel OLE Application has been started. The Excel Application is started via the TExcelApp.[Start](#) method. This is a read only property.

TExcelApp Methods

[Start](#)

[Quit](#)

TExcelApp.Start

procedure Start;

Description

The Start procedure creates the OLE Object and sets the TExcelApp.Started property to true

TExcelApp.Visible

property visible : Boolean;

Description

This property determines whether the Excel Application is visible to the user. If it is set to false (the default) the application is not visible. If set to true the application is visible; The visible property is automatically set to true for a print preview.

TExcelApp.Quit

procedure Quit;

Description

The Quit procedure sets the TExcelApp.Started property to false, Quits the Excel Application, and destroys the OLE Object

TExcelBook properties

[FileName](#)

[ExcelApp](#)

[PasswordRead](#)

[PasswordWrite](#)

TExcelBook Methods

[Open](#)

[Close](#)

[Save](#)

[GetActiveSheet](#)

[SetActiveSheet](#)

TExcelBook.FileName

property FileName : TFileName

Description

The TFileName is a string indicating the drive, path, and filename of the Excel workbook to be assigned to the [TExcelBook](#) component. If the “.xls” extension is omitted it will be appended by the component.

TExcelBook.ExcelApp

[TExcelBook](#)

property ExcelApp : TExcelApp

Description

The ExcelApp points to the parent TExcelApp of the worksheet. It allows you to access the [TExcelApp](#) properties and methods. E.g. if ExcelApp.[ExcelStarted](#) <> true then ExcelApp.[Start](#) ;

The ExcelApp property must be set before accessing any of the properties or methods of the TExcelBook component. The easiest way to do this is once you have dropped both a TExcelApp and a TExcelBook component, use the object inspector and update the property by selecting the down arrow. The TExcelApp should be there for selection.

TExcelBook.PasswordRead

[TExcelBook](#)

property PasswordRead : **string**

Description

This property is a string containing the read password for the Excel Workbook; Used for both the [TExcelBook.Open](#) and the [TExcel.Save](#)

TExcelBook.PasswordWrite

[TExcelBook](#)

property PasswordWrite : **string**

Description

This property is a string containing the write password for the Excel Workbook; Used for both the [TExcelBook.Open](#) and the [TExcel.Save](#)

TExcelBook.Open

[TExcelBook](#)

procedure Open;

Description

This procedure Opens the Excel Worksheet indicated in the TExcelBook.[FileName](#) property

TExcelBook.Close

[TExcelBook](#)

Procedure Close(SaveChanges : boolean);

Description

This procedure Closes the open workbook. If the SaveChanges parameter is true the file will be saved prior to closing. If the SaveChanges is false the file changes will be discarded.

TExcelBook.Save

TExcelBook

procedure Save;

Description

This procedure will save changes made to disk. If you want to do a “SaveAs” just change the TExcelBook.FileName property property prior to executing this method.

TExcelBook.GetActiveSheet

[TExcelBook](#)

function GetActiveSheet : string;

Description

This function returns the name of the currently Active Excel worksheet;

TExcelBook.SetActiveSheet

[TExcelBook](#)

procedure SetActiveSheet(SheetName : string);

Description

This procedure makes the worksheet indicated in the SheetName parameter as the active worksheet

TExcelSheet Properties

ExcelBook

SheetName

TExcelSheet Methods

[InsertData](#)

[AutoFitColumns](#)

[SetFont](#)

[PositionToCell](#)

[SetFormula](#)

[PrintPreview](#)

[PrintHardCopy](#)

TExcelSheet.ExcelBook

TExcelSheet

property ExcelBook : TExcelBook;

Description

The ExcelBook property points to the parent TExcelBook of the worksheet. It allows you to access the TExcelBook properties and methods. E.g. ExcelBook.Save

The ExcelBook property must be set before accessing any of the properties or methods of the TExcelSheet component. The easiest way to do this is once you have dropped both a TExcelBook and a TExcelSheet component, use the object inspector and update the property by selecting the down arrow. The TExcelBook should be there for selection.

TExcelSheet.SheetName

[TExcelSheet](#)

property SheetName : string

Description

This property identifies the name of the worksheet. In Excel the name of the worksheet normally defaults to “sheet1”, sheet2”, etc. In Excel these sheets are renamed to reflect more meaningful names. When you add the TExcelSheet component and associate it with a workbook the worksheets will automatically be added to the workbook with the property of SheetName as its name.

TExcelSheet.InsertData

TExcelSheet

```
procedure InsertData ( InsertText :      pchar;  
                       StartRow   :      integer;  
                       StartColumn : integer;  
                       EndRow     : Integer;  
                       EndColumn  : integer);
```

Description

The InsertData procedure takes the data from the InsertText paramater and places it on the worksheet starting on the StartRow:StartColumn cell. If the data exceeds the range of the EndRow:EndColumn cell it will be truncated. The cell values are delimited by the Tab (#9) character and rows are delimited by the carriage return (#10).

E.g. InsertText := '1234' + #9 + '3457' + #10 + '6780'+ #9 + '3889'; would yield

1234	3457
6780	3889

TExcelSheet.AutoFitColumns

[TExcelSheet](#)

Procedure AutoFitColumns(StartRow : integer;
StartColumn : integer;
EndRow : Integer;
EndColumn : integer);

Description

This method will expand the Excel cell width to the largest cell in a column and the height to the largest in a cell row.

TExcelSheet.SetFont

[TExcelSheet](#)

```
procedure SetFont(StartRow      : integer  
                  StartColumn   : integer  
                  EndRow        : Integer;  
                  EndColumn     : integer;  
                  FontInfo      : TFont);
```

Description

This method takes a TFont object and assigns it to the Excel text within the passed range of cells.

TExcelSheet.PositionToCell

[TExcelSheet](#)

```
procedure PositionToCell(row    : integer;  
                        column  : integer);
```

Description

This method will make the Row:Column combination the current cell.

TExcelSheet.SetFormula

[TExcelSheet](#)

```
procedure SetFormula(Cells      : string;  
                     formula    : string);
```

Description

This method will put whatever formula identified in the formula parameter in the cells passed through the Cells parameter. To Only apply the formula to one cell pass the Cell name e.g. A5. To apply the formula to multiple cells enter a range e.g. A5:A8.

TExceSheet.PrintPreview

[TExcelSheet](#)

Procedure PrintPreview;

Description

This method will make the sheet the active or current sheet then it will call the Excel print preview. Note that if there is no data within the worksheet, this method has no effect.

TExcelSheet.PrintHardCopy

TExcelSheet

```
procedure PrintHardCopy(PageFrom      : integer;  
                        PageTo        : integer;  
                        Copies         : integer);
```

Description

This method will print a hard copy of the sheet to the default printer. Set the TExcelApp.ShowExcelAlerts property to true if you want to select a printer other than the default.

