TscExcelExport

VCL component for Delphi


- Version 4.4
- March 2019

- Stefan Cruysberghs
- Flanders, Belgium
- http://www.scip.be
# Contents

Contents ........................................................................................................................................... 2  
Description ....................................................................................................................................... 4  
Properties ........................................................................................................................................ 4  
Events ............................................................................................................................................... 5  
Methods .......................................................................................................................................... 5  
Component editor ............................................................................................................................ 5  
Live templates .................................................................................................................................. 5  
Type library / COM ............................................................................................................................. 5  
Delphi / Office ......................................................................................................................................... 5  
Copyrights & registration .................................................................................................................... 6  
Prices .................................................................................................................................................. 6  
Files included ..................................................................................................................................... 7  
Installation .......................................................................................................................................... 8  
  Run-time and design-time packages ................................................................................................. 8  
  Build and install packages in Delphi 7 .............................................................................................. 9  
  Build and install packages in Delphi XE3. ....................................................................................... 10  
  Build run-time package in Delphi 10 Seattle .................................................................................... 11  
  Build and install design-time package in Delphi 10.3 Rio ............................................................... 11  
  Library Path ...................................................................................................................................... 12  
  Library path in Delphi XE3 ............................................................................................................... 12  
  Library Path in Delphi 10.3 Rio ......................................................................................................... 13  
Type libraries and Excel versions ........................................................................................................ 13  
  What is a type library? ..................................................................................................................... 13  
  Configure a type library in scExcelExportConfig.inc ................................................................. 14  
  Supported type libraries ................................................................................................................... 15  
  Supported Excel versions .................................................................................................................. 15  
Known issues ........................................................................................................................................ 16  
  Component can’t be registered ........................................................................................................ 16  
  OLE error / Interface not supported ............................................................................................... 16  
Demo application ................................................................................................................................. 17
Error reading form in older Delphi versions................................................................. 18
Loading MIDAS.DLL in demo application.................................................................... 18
Examples.......................................................................................................................... 19
Example 1: Easiest way to export dataset to Excel......................................................... 19
Example 2: Using layout properties, adding summary cells and save file...................... 19
Example 3: Export multiple datasets.............................................................................. 20
Example 4: Change background color and font style in the OnGetCellStyle event ...........21
Example 5: Grouping........................................................................................................ 22
Example 6: Export worksheets....................................................................................... 23
Example 8: Open existing worksheets............................................................................ 24
Example 9: Access ExcelWorksheet object.................................................................... 24
More examples................................................................................................................ 25
Version history............................................................................................................... 27
Description

This TscExcelExport component is an advanced, powerful but easy to use component which enables you to export all records of a dataset from Embarcadero Delphi to Microsoft Excel. Many features are provided to change the layout, use conditional formatting, to add totals, to create groups, to set a filter, ...


Properties

- Name of worksheet and file
- Header and footer texts
- Begin row and column of header, footer, titles (=fieldnames) and data (fieldvalues)
- Width of columns or autofit
- Font of header, footer, titles, summary and data (Alignment, HorizontalAlignment, WrapText, Orientation, MergeCells)
- BackgroundColor and borders of header, titles, summary and data
- Summaries for numeric or given fields (SUM, MIN, MAX, AVG, COUNT)
- Create groups with given fields
- AutoFilter for titles
- A lot of other options to customize the result
Events

- To define the background color and font color, size, name and bold style of each cell
- For exporting data without using a TDataset

Methods

- Export to Excel
- Save worksheet as XLSX (Open XML), XLS, HTML, XML or CSV
- Show print preview

Component editor

- The component editor can be used to change some settings on an easy way.

Live templates

- Live templates since Delphi 2006: ExcelExportUse and ExcelExportCreate

Type library / COM

- This component uses the Microsoft Office server components from the COM type library. Each Delphi version supports different Excel type libraries.
- Microsoft Excel should be installed on the PC when using this component.

Delphi / Office

- This component will also give you access to the Excel Application, Workbook and Worksheet objects so you can access all VBA properties and methods.
Copyrights & registration

- All copyrights to this component are owned by the author Stefan Cruysberghs.
- This component is freeware for non-commercial use only and it can be freely distributed.
- The author doesn’t give a warranty for error free running of this component.
- If you like this component then you can register it to encourage the author to further develop and improve this component.
- **If you are using the component in a commercial environment/application then you are obligated to register it!**
- Benefits of registering
  - Site license for unlimited developers.
  - Unlimited deployment license.
  - Full source code for Delphi version 5 to XE10 and 10.3.
  - Support will be provided by e-mail.
  - Bugs will be solved as soon as possible.
  - You will receive an email when new versions or upgrades are available.

Prices

- I am using the MyCommerce eSellerate service (http://www.mycommerce.com/esellerate) for payment processing. You can pay safely by credit card online, by fax or by phone.
- Single developer license: $35:
- Site license (unlimited developers): $130:
- Please send an email if you want to pay via PayPal or need an invoice
- After registering you do not need a registration key. The version which can be downloaded at [www.scip.be](http://www.scip.be) is fully functional.
Files included

- **scExcelExport.pas**: component
- **scExcelExportReg.pas**: component editor and registration of component
- **scExcelExportConfig.inc**: configuration of type library (97, 2000, XP, 2010)
- **scExcelExport.dcr**: component icon
- **ExcelExportPackx.dproj**: run-time package for Delphi 5, 6, 7, 9 (2005), 10 (2006 & Turbo), 11 (2007), 12 (2009), 14 (2010), XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10 Seattle and 10.1 Berlin
- **dclExcelExportPackx.dproj**: design-time package for Delphi 5, 6, 7, 9 (2005), 10 (2006 & Turbo), 11 (2007), 12 (2009), 14 (2010), XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10 Seattle, 10.1 Berlin, 10.2 Tokyo, 10.3 Rio
- **TscExcelExport readme.pdf**: this file
- **DemoExcelExport.exe**: demonstration application
- **/Live templates**: the 2 live templates (ExcelExportCreate.xml and ExcelExportUse.xml) for Delphi (BDS) 2006, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10, 10.2, 10.3 should be copied to
  - Delphi 2006
    - C:\Program Files\Borland\BDS\4.0\Objrepos\Code_Templates\Delphi
  - Delphi 2007
    - C:\Program Files\CodeGear\RAD Studio\5.0\ObjRepos\Code_Templates\Delphi
  - Delphi 2009
    - C:\Program Files\CodeGear\RAD Studio\6.0\ObjRepos\Code_Templates\Delphi
  - Delphi 2010
    - C:\Program Files\Embarcadero\RAD Studio\7.0\ObjRepos\Code_Templates\Delphi
  - Delphi XE
    - C:\Program Files\Embarcadero\RAD Studio\8.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi XE2
    - C:\Program Files\Embarcadero\RAD Studio\9.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi XE3
    - C:\Program Files\Embarcadero\RAD Studio\10.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi XE4
    - C:\Program Files\Embarcadero\RAD Studio\11.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi XE5
    - C:\Program Files\Embarcadero\RAD Studio\13.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi XE6
    - C:\Program Files\Embarcadero\RAD Studio\14.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi XE7
    - C:\Program Files\Embarcadero\RAD Studio\15.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi XE8
    - C:\Program Files\Embarcadero\RAD Studio\16.0\ObjRepos\en\Code_Templates\Delphi
  - Delphi 10 Seattle
    - C:\Program Files\Embarcadero\RAD Studio\17.0\ObjRepos\en\Code_Templates\Delphi
• Delphi 10.1 Berlin
  C:\Program Files\Embarcadero\RAD Studio\18.0\ObjRepos\en\Code_Templates\Delphi
• Delphi 10.2 Tokyo
  C:\Program Files\Embarcadero\RAD Studio\19.0\ObjRepos\en\Code_Templates\Delphi
• Delphi 10.3 Rio & Community Edition
  C:\Program Files\Embarcadero\RAD Studio\20.0\ObjRepos\en\Code_Templates\Delphi
• /Source demo: demo project to demonstrate all features of the TscExcelExport component.

Installation

Run-time and design-time packages

• Step 1: Open the run-time package ExcelExportPackX.dproj and Compile or Build it
  o Borland Delphi 5: ExcelExportPack5
  o Borland Delphi 6: ExcelExportPack6
  o Borland Delphi 7: ExcelExportPack7
  o Borland Delphi 2005: ExcelExportPack9
  o Borland Delphi 2006: ExcelExportPack10
  o Codegear Delphi 2007: ExcelExportPack11
  o Codegear Delphi 2009: ExcelExportPack12
  o Embarcadero Delphi 2010: ExcelExportPack14
  o Embarcadero Delphi XE: ExcelExportPackXE
  o Embarcadero Delphi XE2: ExcelExportPackXE2
  o Embarcadero Delphi XE3: ExcelExportPackXE3
  o Embarcadero Delphi XE4: ExcelExportPackXE4
  o Embarcadero Delphi XE5: ExcelExportPackXE5
  o Embarcadero Delphi XE6: ExcelExportPackXE6
  o Embarcadero Delphi XE7: ExcelExportPackXE7
  o Embarcadero Delphi XE8: ExcelExportPackXE8
  o Embarcadero Delphi 10 Seattle: ExcelExportPackD10
  o Embarcadero Delphi 10.1 Berlin: ExcelExportPackD101
  o Embarcadero Delphi 10.2 Tokyo: ExcelExportPackD102
  o Embarcadero Delphi 10.3 Rio & Community Edition: ExcelExportPackD103

• Step 2: Open the design-time package dclExcelExportPackX.dproj (prefix dcl) and Compile it.

• Step 3: Install it. The install option is only available in the context popup menu when you right click
• If you already installed this component previously then it is better to do an Uninstall first.
- The TscExcelExport component can be found in the tabsheet 'SC' of the component palette.

Build and install packages in Delphi 7
Build and install packages in Delphi XE3
Build run-time package in Delphi 10 Seattle

Build and install design-time package in Delphi 10.3 Rio
• When you like to add the component to an existing package, add the unit scExcelExport to a run-time package. Make sure the DCP file dclOffice is added as required. This file can be found in the Delphi Lib folder. The unit scExcelExportReg.pas contains the registration and property and component editor. This unit should be included in a design-time package.

Library Path
Make sure to add a path to the folder with the scExcelExport.pas file in the global Library path of Delphi (Tools >> Options >> Delphi Options >> Library) before compiling the demo project.

Library path in Delphi XE3
Library Path in Delphi 10.3 Rio

Type libraries and Excel versions

What is a type library?

- A type library is a binary file that contains the description about a COM or DCOM object’s properties and methods in a form that is accessible to other applications at runtime. Microsoft calls an OCX an ActiveX control of which the fundamental concept is the Component Object Model (COM) and, in a network, the Distributed Component Object Model (DCOM). These type libraries with the interfaces of Office are provided by Microsoft and are included in the Office installation.

- Delphi provides libraries where the interfaces of these Office objects are converted into an Object Pascal unit files. These Delphi type libraries for the Microsoft Office applications can be found in the **OCX\Servers** folder of your installed Delphi version.

- The TscExcelExport component uses the type library for Excel.

- Each Delphi version supports different Excel type libraries.

- Microsoft Excel should be installed on the PC when using this component.
Configure a type library in scExcelExportConfig.inc

- For each Delphi version the latest type library is defined. This is implemented with compiler directives in the `scExcelExportConfig.inc` file. Since Delphi XE2 the default is **EXCEL2010**. There is no newer type library for Delphi since 2010.
- To override the default type library, you have to uncomment the `$UNDEF` and `$DEFINE` lines at the bottom of the `scExcelExportConfig.inc` file.

```plaintext
(* Description : scExcelExportConfig.inc - compiler directives

($IFDEF VER330) // Delphi 10.3 Rio : Default Excel2010, ExcelXP can also be used
  ($DEFINE EXCEL2010)
  ($DEFINE DELPHI103)
  ($DEFINE DELPHI100CORNEWER)
  ($DEFINE DELPHIXE2CORNEWER)
  ($DEFINE DELPHIXE2CORNEWER)
($ENDIF)

($IFDEF VER320) // Delphi 10.2 Tokyo : Default Excel2010, ExcelXP can also be used
  ($DEFINE EXCEL2010)
  ($DEFINE DELPHI102TOKYO)
  ($DEFINE DELPHI100CORNEWER)
  ($DEFINE DELPHIXE2CORNEWER)
  ($DEFINE DELPHIXE2CORNEWER)
($ENDIF)
```
Supported type libraries

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Excel2010.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Excel2010</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExcelXP.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>EXCELXP</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excel2000.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EXCEL2000</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Excel97.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EXCEL97</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Supported Excel versions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Excel2010.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EXCEL2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExcelXP.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EXCELXP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excel2000.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EXCEL2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excel97.pas</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EXCEL97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Known issues

Component can’t be registered

This error can occur when installing the design-time package (dclExcelExport). It has something to do with another package of another component that has already registered one of the Office type libraries provided by Delphi.

➢ The solution is to open the TscExcelExport packages but remove all the required libraries of Delphi. Then build and install it. Delphi will add all required packages again and it will skip the ones that are already loaded by other packages.

OLE error / Interface not supported

If your application that exports data to Excel is running perfectly on one computer, but it throws an OLE-error (e.g. 800A03EC) or “Interface not supported” or “Client not registered” exception on another computer, then there is something wrong with the registration of the COM/OLE libraries (registry reference or missing DLL) of Microsoft Office.
➢ You can try to force an Excel version to register without running the setup. This can be done by passing the `/regserver` option. It only works for Excel XP and 2003. Since 2007 this option has been removed.
  - XP: `C:\Program Files\Microsoft Office\Office10\EXCEL.EXE /regserver`
  - 2003: `C:\Program Files\Microsoft Office\Office11\EXCEL.EXE /regserver`

➢ For Excel versions 2007 and up, you have to repair or reinstall Microsoft Office via the setup.

**Demo application**

A full demo application with 18 examples is provided to test all features of the component (and Excel automation). The project file and source code is located in the **Source demo** folder. The executable **DemoExcelExport.exe** is also included in this folder.

Make sure to add a path to the folder with the scExcelExport.pas file in the global Library path of Delphi (Tools >> Options >> Delphi Options >> Library) before compiling the demo project.
Error reading form in older Delphi versions

Error reading PersistDataPacket, ExplicitTop, ExplicitLeft, ExplicitWidth and ExplicitHeight properties when opening the demo application in older Delphi versions.

- Just choose "Ignore all". These properties will be removed from the DFM and the application will run fine.

Loading MIDAS.DLL in demo application

Error loading MIDAS.DLL when running the demo application.

- Make sure the MIDAS.DLL assembly is in the same folder as the Demo application. The provided MIDAS.DLL is version 17.
Examples

Example 1: Easiest way to export dataset to Excel

```delphi
scExcelExport1.Dataset:=MyDataset;
scExcelExport1.ExportDataset;
scExcelExport1.Disconnect;
```

Example 2: Using layout properties, adding summary cells and save file

```delphi
scExcelExport1.WorksheetName := 'MyDataset';
scExcelExport1.Dataset:=MyDataset;
scExcelExport1.StyleColumnWidth:=cwOwnerWidth;
scExcelExport1.ColumnHeader := 20;
scExcelExport1.HeaderText.Text := 'Header';
scExcelExport1.BeginRowHeader := 2;
scExcelExport1.FontTitles := LabelTitle.Font;
scExcelExport1.FontTitles.Orientation := 45;
scExcelExport1.BorderTitles.LineStyle := blLine;
scExcelExport1.BeginRowTitles := 5;
scExcelExport1.FontData := LabelData.Font;
scExcelExport1.SummarySelection := ssValues;
scExcelExport1.SummaryCalculation := scMAX;
scExcelExport1.ExcelVisible:=False;
try
  scExcelExport1.ExportDataset;
  if Assigned(scExcelExport1.ExcelWorkSheet) then
  scExcelExport1.SaveAs('c:\test.xls',ffXLS);
finally
  scExcelExport1.Disconnect;
end;
```
Example 3: Export multiple datasets

```
scExcelExport1.ExcelVisible:=True;
try
  scExcelExport1.Dataset:=MyDataset;
  scExcelExport1.WorksheetName:='1';
  scExcelExport1.ConnectTo := ctNewExcel;
  scExcelExport1.ExportDataset;
  scExcelExport1.Disconnect;
  scExcelExport1.Dataset:=Table2;
  scExcelExport1.WorksheetName:='2';
  scExcelExport1.ConnectTo := ctNewWorkbook;
  scExcelExport1.ExportDataset;
  scExcelExport1.Disconnect;
  scExcelExport1.Dataset:=Table3;
  scExcelExport1.WorksheetName:='3';
  scExcelExport1.ConnectTo := ctNewWorksheet;
  scExcelExport1.ExportDataset;
finally
  scExcelExport1.Disconnect;
end;
```
procedure scExcelExportGetCellStyleEvent(Sender: TObject; Field: TField; var ColorBackground : TColor; FontCell : TxlFont);
begin
  if Field.FieldName = 'CustNo' then
  begin
    if Field.Value > 2000 then
    begin
      FontCell.Color := clRed;
      FontCell.Name := 'Times New Roman';
      FontCell.Size := 14;
    end;
    if Field.Value > 3000 then
    begin
      FontCell.Style := [fsBold];
    end;
  end;
  if Field.FieldName = 'EmpNo' then
  begin
    if Field.Dataset.FieldByName('CustNo').Value > 2000 then
    begin
      ColorBackground := clRed;
    end;
    if Field.Dataset.FieldByName('EmpNo').Value > 100 then
    begin
      ColorBackground := clYellow;
    end;
  end;
end;
Example 5: Grouping
See full example 13 in demo application

```delphi
try
    scExcelExport1.Dataset:=DatasetOrders;
    scExcelExport1.GroupFields.Add('ShipVia');
    scExcelExport1.GroupFields.Add('Terms');
    scExcelExport1.ExportDataset;
finally
    scExcelExport1.Disconnect;
end;
```
try
  scExcelExport1.LoadDefaultProperties;
  scExcelExport1.ExcelVisible:=False;
  scExcelExport1.WorksheetName := 'TscExcelExport DEMO 9';
  scExcelExport1.Dataset:=DatasetOrders;
  StatusBar.Panels[1].Text := '';
  scExcelExport1.ExportDataset;

  scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName) + 'ExcelExportDefault',ffDefault); //without file extension

  if scExcelExport1.ExcelVersion = 12 then
    scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName) + 'ExcelExport2007.xlsx',ffXLSX);

  scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName) + 'ExcelExport2003.xls',ffXLS);

  if scExcelExport1.ExcelVersion <> 12 then
    scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName) + 'ExcelExport97.xls',ffXL97);

  scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName) + 'ExcelExportCSV.csv',ffCSV);

  if scExcelExport1.ExcelVersion >= 10 then
    scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName) + 'ExcelExportHTM.htm',ffHTM);

  if scExcelExport1.ExcelVersion >= 11 then
    scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName) + 'ExcelExportXML.xml',ffXML);

finally
  scExcelExport1.Disconnect(True);
end;
Example 8: Open existing worksheets

See full example 12 in demo application

```delphi
scExcelExport1.LoadDefaultProperties([pgPositions,pgText]);
scExcelExport1.ExcelVisible:=False;
scExcelExport1.Dataset:=DatasetBiolife;
scExcelExport1.WorksheetName:='Biolife';
scExcelExport1.Filename:=ExtractFilePath(Application.ExeName)+'ExcelExport.xls';
scExcelExport1.ExportDataset;
scExcelExport1.SaveAs(ExtractFilePath(Application.ExeName)+'ExcelExport.xls',ffXLS);
scExcelExport1.Disconnect;
```

Example 9: Access ExcelWorksheet object

See full example 14 in demo application

```delphi
try
  scExcelExport1.ExcelVisible:=True;
  scExcelExport1.LoadDefaultProperties;
  scExcelExport1.Dataset:=DatasetAnimals;
  scExcelExport1.WorksheetName:='TscExcelExport DEMO 14';
  scExcelExport1.ConnectTo := ctNewExcel;
  scExcelExport1.Connect;
  scExcelExport1.ExcelWorkSheet.Range;'B2','B2'.AddComment('This is comment for a cell');
finally
  scExcelExport1.Disconnect;
end;
```
More examples
Check out the demo application with 18 examples which show all the features of the component and how to use the Excel COM type library.

- 1: Simple export
- 2: Calculated fields, display format of numbers and dates
- 3: Memo fields and AutoFilter property
- 4: Memo field and OnGetText event
- 5: Summaries for numeric fields (SummarySelection & SummaryCalculation properties)
- 6: Headers, footers, summaries, fonts & borders (HeaderText, FooterText, BorderHeader, BorderTitles, BorderSummary, BorderData, BorderFooter properties)
- 7: Conditional background color and fonts (OnGetCellStyleEvent)
- 8: Visible fields (VisibleFieldsOnly property)
- 9: Save worksheet to file (XLSX, XLS, CVS, XML and HTML)
- 10: Show print preview
- 11: Add workbooks and worksheets to active Excel instance (ConnectTo & WorksheetName properties)
- 12: Reuse existing files and worksheets (WorksheetName & FileName properties)
- 13: Grouping of data (GroupFields, HeaderText, MergeHeaderCells, FooterText, MergeFooterCells, BorderHeader, BorderTitles, BorderSummary, BorderData, BorderFooter, AutoFilter properties)
- 14: Access ExcelWorksheet object and use all Excel methods and properties (Range, Cells, EntireColumn, Font, AddComment, ...)
- 15: Access ExcelApplication and ExcelWorkbook objects and use all Excel methods and properties (AutoFilter, FreezePanes, WebPagePreview, PivotTableWizard, ...)
- 16: Export data using events instead of dataset (OnGetFieldName, OnGetFieldValue, OnGetFieldCount, OnGetFieldDisplayName, OnGetFieldDataType, OnGotoNextRecord, OnGetFieldCellStyleEvent)
- 17: Close all active Excel instances
More information about the Microsoft Excel objects can be found at the Microsoft website: https://docs.microsoft.com/en-us/dotnet/api/microsoft.office.tools.excel
Version history

Version 4.4 (March 2019)

- Added Font.VertialAlign (vaTop, vaCenter, vaBottom)
- Extended Font.Align (=Horizontal align) (haDistributed, haFill, haJustify)
- Fixed missing Font properties in OnGetCellStyle and OnGetFieldCellStyle
- Updated demo application

Version 4.31 (January 2019)

- Added support for Delphi 10.3 Rio & Community Edition
- Added ftByte field type

Version 4.3 (September 2017)

- Added support for Excel XLSM (Open XML format macro enabled)
- Bugfixes in some packages

Version 4.29 (July 2017)

- Added support for Delphi 10.2 Tokyo

Version 4.28 (July 2016)

- Added support for Delphi 10.1 Berlin
- Bugfix for WideString

Version 4.27

- Added support for Delphi 10 Seattle
- Added support for Excel 2016

Version 4.26

- Added support for Delphi XE8

Version 4.25

- Added support for Delphi XE7

Version 4.24

- Added support for Delphi XE6
- Bugfix time formatting with AM/PM
• Bugfix currency fields when there are hidden fields

Version 4.23

• Added support for Delphi XE5

Version 4.22

• Bugfix for Delphi 2010
• Added support for Delphi XE4
• Unknown Office versions will be handled as new version (number=99)

Version 4.21

• Bugfixes for Delphi XE3
• Added support for Office 2013

Version 4.2

• Added support for Delphi XE3

Version 4.1

• Added support for Delphi XE2. Installing update 1 and 2 of Delphi XE2 is required
• Added support for Excel2010 type library
• Bugfix exporting as XLSX in Office 2010

Version 4.0

• Added support for Delphi 2010 and XE (2011)
• Added support for Office 2010
• Added support for ftSingle field type
• Bugfix SetFormat when there are invisible fields
• Added COUNT as summary calculation
• Added BackAlternateColor to CellBorder

Version 3.9

• Added support for Delphi 2010

Version 3.81

• Support for Interbase TSQLTimeStampField
Version 3.8

- Improvements to SetAutoFilter method. Now this will also work properly in Excel 97 and 2000
- GetDateTimeFormat has become a public function
- GetCurrencyFormat has become a public function
- Important modifications and bugfixes for exporting dates and times. Now these values are always exported as Delphi datetime (=float) which is the same as the internal Excel datetime value (1 = 1 January 1900). Formatting is done afterwards in the SetFormat function. This should solve all date problems in international Excel versions.
- Added support for Delphi 2009

Version 3.7

- Added ftWideMemo datatype in IsMemoField function (>= Delphi 2006)
- Added ftFixedWideChar, ftWideMemo, ftOraTimeStamp and ftOraInterval datatype to CanConvertFieldToCell function (>= Delphi 2006)
- Added internal record with Excel restrictions (max rows, max columns, max string length) which will be initialized after checking the Excel type library and version
- Added EMaxColumnsReached and EMaxRowsReached exceptions and resource strings
- Added support for more than 256 columns in Excel 2007
- Bugfix when exporting strings larger than 910 characters in Excel 2003

Version 3.6

- Added support for Delphi 2007
- Retested support for Excel 2007
- Tested with Windows Vista
- Started with converting comments to XML documentation
- Added public method FindFirstEmptyRow
Version 3.5 (September 2006)

- Added properties for merging of header and footer cells (MergeHeaderCells, MergeFooterCells)
- Added AutoFilter property for titles of dataset
- Added support for Excel 2007 (bêta)
- Added ffXLSX for saving files in Excel 2007 Open XML format
- Added support for OnGetText event for numeric fields
- Added public property for LCID so it can be used after exporting
- Using default font and size of Excel by setting Size = -1 and Font.Name = ‘MS Sans Serif’
- Tested with Turbo Delphi Explorer (Win32)
- Renewed demo application !
- Bugfix when setting border and font when using OnGetStyle events

Version 1.0 (February 2000)

- See the source code for information about older versions.