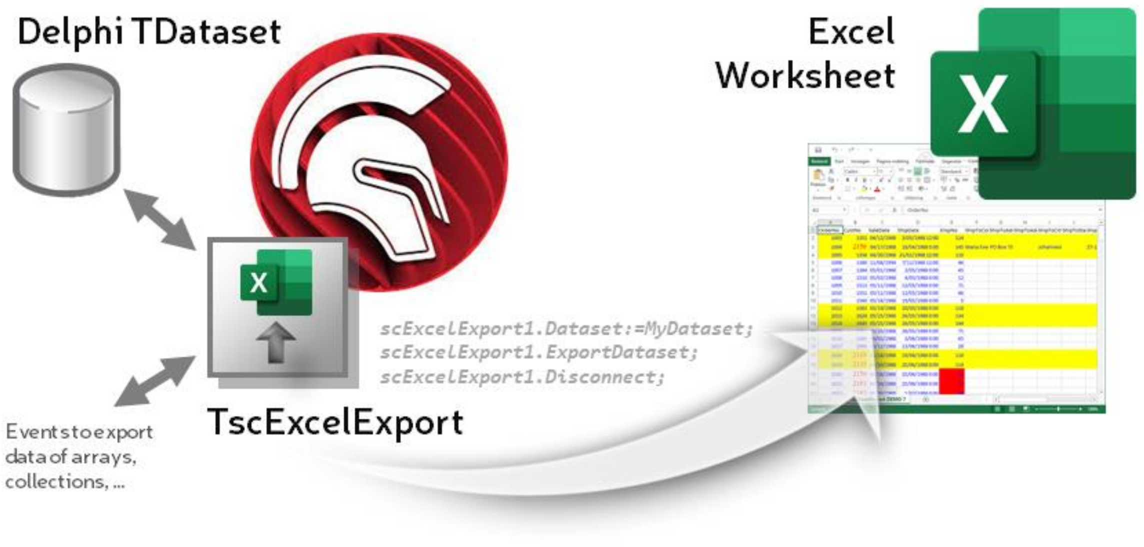
TscExcelExport

VCL component for Delphi

- Support for Delphi 5, 6, 7, 2005, 2006, Turbo Delphi, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10 Seattle, 10.1 Berlin, 10.2 Tokyo, 10.3 Rio, 10.4 Sydney, 11 Alexandria & Community Edition
- Support for Excel 97, 2000, XP, 2003, 2007, 2010, 2013, 2015 and 2016/2019/365

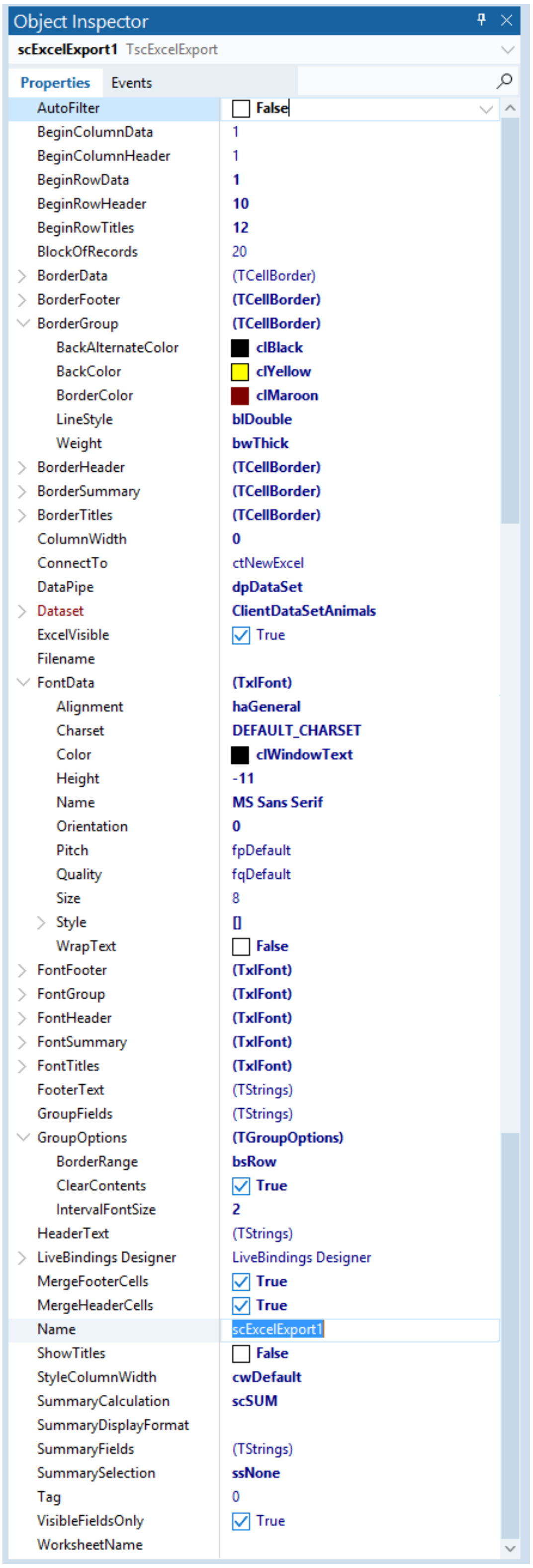
- Version 4.42
- October 2021

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


Delphi TDataset
Excel Worksheet
scExcelExport1.Dataset:=MyDataset;
scExcelExport1.ExportDataset;
scExcelExport1.Disconnect;
TscExcelExport
Events to export data of arrays, collections, ...

Contents

Description



Object Inspector

scExcelExport1 TscExcelExport

Properties Events

AutoFilter	☐ False
BeginColumnData	1
BeginColumnHeader	1
BeginRowData	1
BeginRowHeader	10
BeginRowTitles	12
BlockOfRecords	20
BorderData	(TCellBorder)
BorderFooter	(TCellBorder)
BorderGroup	(TCellBorder)
BackAlternateColor	⬛ clBlack
BackColor	🟨 clYellow
BorderColor	🟥 clMaroon
LineStyle	blDouble
Weight	bwThick
BorderHeader	(TCellBorder)
BorderSummary	(TCellBorder)
BorderTitles	(TCellBorder)
ColumnWidth	0
ConnectTo	ctNewExcel
DataPipe	dpDataSet
Dataset	ClientDataSetAnimals
ExcelVisible	☑ True
Filename	
FontData	(TxlFont)
Alignment	haGeneral
Charset	DEFAULT_CHARSET
Color	⬛ clWindowText
Height	-11
Name	MS Sans Serif
Orientation	0
Pitch	fpDefault
Quality	fqDefault
Size	8
Style	[]
WrapText	☐ False
FontFooter	(TxlFont)
FontGroup	(TxlFont)
FontHeader	(TxlFont)
FontSummary	(TxlFont)
FontTitles	(TxlFont)
FooterText	(TStrings)
GroupFields	(TStrings)
GroupOptions	(TGroupOptions)
BorderRange	bsRow
ClearContents	☑ True
IntervalFontSize	2
HeaderText	(TStrings)
LiveBindings Designer	LiveBindings Designer
MergeFooterCells	☑ True
MergeHeaderCells	☑ True
Name	scExcelExport1
ShowTitles	☐ False
StyleColumnWidth	cwDefault
SummaryCalculation	scSUM
SummaryDisplayFormat	
SummaryFields	(TStrings)
SummarySelection	ssNone
Tag	0
VisibleFieldsOnly	☑ True
WorksheetName	

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, ...

This VCL component works in Delphi 5, 6, 7, 2006, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10 Seattle, 10.1 Berlin, 10.2 Tokyo, 10.3 Rio, 110.4 Sydney and 11 Alexandria & Community Edition. It uses the Microsoft Office type library and it supports Excel 97, 2000, XP, 2003, 2007, 2010, 2013, 2016 and 2019/365.

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 VCL component was tested in Delphi 5 (SP1), Delphi 6, Delphi 7, Delphi (BDS) 2005, Delphi (BDS) 2006, Turbo Delphi, Delphi 2007, Delphi 2009, Delphi 2010, Delphi XE, Delphi XE2, Delphi XE3, Delphi XE4, Delphi XE5, Delphi XE6, Delphi XE7, Delphi XE8, Delphi 10 Seattle, Delphi 10.1 Berlin, Delphi 10.2 Tokyo, Delphi 10.3 Rio, Delphi 10.4 Sydney, Delphi 11 Alexandria & Community Edition

- The TscExcelExport component works with Excel 97, Excel 2000, Excel XP, Excel 2003, Excel 2007, Excel 2010, Excel 2013, Excel 2016, Excel 2019 and Excel 365 and it has been tested with Dutch and some English Office versions.
- 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 XE8 and to 11.
  - Bugs will be solved as soon as possible.
  - I try to provide support by e-mail.

Prices

- You can pay via PayPal. Please send me an email if you need an invoice
- Single developer license : **35** euro: https://www.paypal.com/paypalme/StefanCruysberghs/35EUR
- Site license (unlimited developers) : **130** euro: https://www.paypal.com/paypalme/StefanCruysberghs/130EUR
- After registering you do not need a registration key. The version which can be downloaded at 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
- **dclExcelExportPackx.dproj**: design-time package
- **TscExcelExport readme.pdf**: this file
- **DemoExcelExport.exe**: demonstration application
- **/Live templates**: the 2 live templates (ExcelExportCreate.xml and ExcelExportUse.xml) 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

- Delphi 10.4 Sydney
  C:\Program Files\Embarcadero\RAD Studio\21.0\ObjRepos\en\Code_Templates\Delphi
- Delphi 11 Alexandria
  C:\Program Files\Embarcadero\RAD Studio\22.0\ObjRepos\en\Code_Templates\Delphi
- **/Source demo**: demo project to demonstrate all features of the TscExcelExport component.

Installation

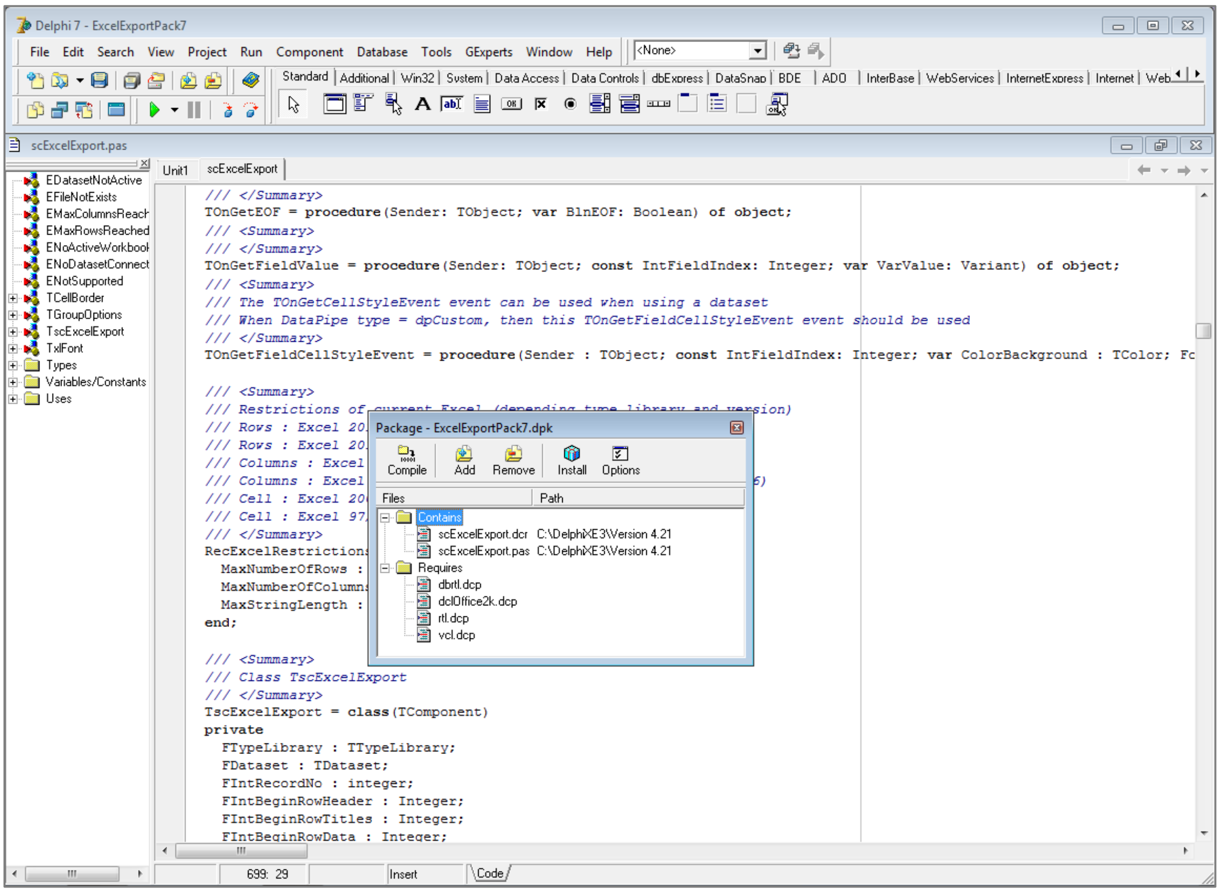
Run-time and design-time packages

- Step 1: Open the run-time package **ExcelExportPackX.dproj** and **Compile** or **Build** it
  - Borland Delphi 5: ExcelExportPack5
  - Borland Delphi 6: ExcelExportPack6
  - Borland Delphi 7: ExcelExportPack7
  - Borland Delphi 2005: ExcelExportPack9
  - Borland Delphi 2006: ExcelExportPack10
  - Codegear Delphi 2007: ExcelExportPack11
  - Codegear Delphi 2009: ExcelExportPack12
  - Embarcadero Delphi 2010: ExcelExportPack14
  - Embarcadero Delphi XE: ExcelExportPackXE
  - Embarcadero Delphi XE2: ExcelExportPackXE2
  - Embarcadero Delphi XE3: ExcelExportPackXE3
  - Embarcadero Delphi XE4: ExcelExportPackXE4
  - Embarcadero Delphi XE5: ExcelExportPackXE5
  - Embarcadero Delphi XE6: ExcelExportPackXE6
  - Embarcadero Delphi XE7: ExcelExportPackXE7
  - Embarcadero Delphi XE8: ExcelExportPackXE8
  - Embarcadero Delphi 10 Seattle: ExcelExportPackD10
  - Embarcadero Delphi 10.1 Berlin: ExcelExportPackD101
  - Embarcadero Delphi 10.2 Tokyo: ExcelExportPackD102
  - Embarcadero Delphi 10.3 Rio & Community Edition: ExcelExportPackD103
  - Embarcadero Delphi 10.4 Sydney: ExcelExportPackD104
  - Embarcadero Delphi 11 Alexandria: ExcelExportPackD11

- 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.

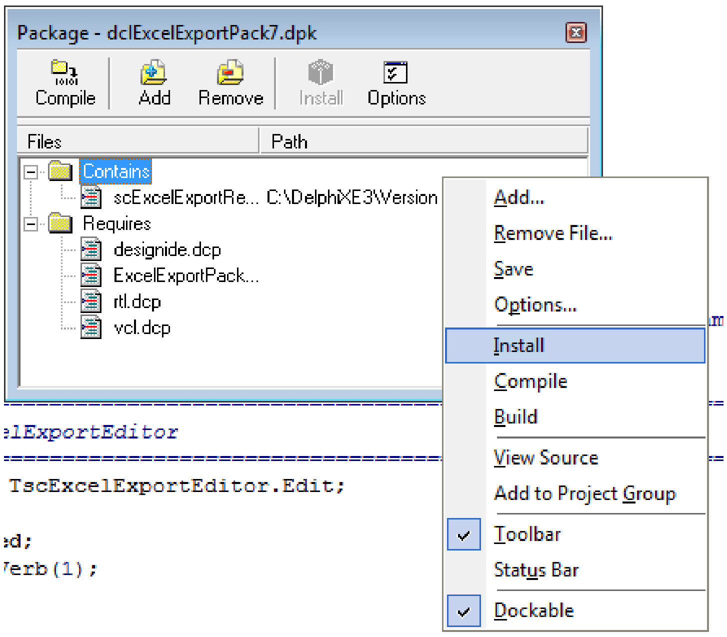
- The TscExcelExport component can be found in the tabsheet '**SC**' of the component palette.

Build and install packages in Delphi 7


Delphi 7 - ExcelExportPack7
File Edit Search View Project Run Component Database Tools GExperts Window Help
<None>
Standard Additional Win32 System Data Access Data Controls dbExpress DataSnap BDE ADO InterBase WebServices InternetExpress Internet Web
scExcelExport.pas
EDatasetNotActive
EFileNotExists
EMaxColumnsReach
EMaxRowsReached
ENoActiveWorkbook
ENoDatasetConnect
ENotSupported
TCellBorder
TGroupOptions
TscExcelExport
TxlFont
Types
Variables/Constants
Uses
Unit1
scExcelExport
Package - ExcelExportPack7.dpk
Compile
Add
Remove
Install
Options
699: 29
Insert
Code

```
/// </Summary>
TOnGetEOF = procedure(Sender: TObject; var BlnEOF: Boolean) of object;
/// <Summary>
/// </Summary>
TOnGetFieldValue = procedure(Sender: TObject; const IntFieldIndex: Integer; var VarValue: Variant) of object;
/// <Summary>
/// The TOnGetCellStyleEvent event can be used when using a dataset
/// When DataPipe type = dpCustom, then this TOnGetFieldCellStyleEvent event should be used
/// </Summary>
TOnGetFieldCellStyleEvent = procedure(Sender : TObject; const IntFieldIndex: Integer; var ColorBackground : TColor; Fc

/// <Summary>
/// Restrictions of current Excel (depending type library and version)
/// Rows : Excel 20
/// Rows : Excel 20
/// Columns : Excel
/// Columns : Excel                                              6)
/// Cell : Excel 20
/// Cell : Excel 97,
/// </Summary>
RecExcelRestriction
  MaxNumberOfRows :
  MaxNumberOfColumn
  MaxStringLength :
end;

/// <Summary>
/// Class TscExcelExport
/// </Summary>
TscExcelExport = class(TComponent)
private
  FTypeLibrary : TTypeLibrary;
  FDataset : TDataset;
  FIntRecordNo : integer;
  FIntBeginRowHeader : Integer;
  FIntBeginRowTitles : Integer;
  FIntBeginRowData : Integer;
```

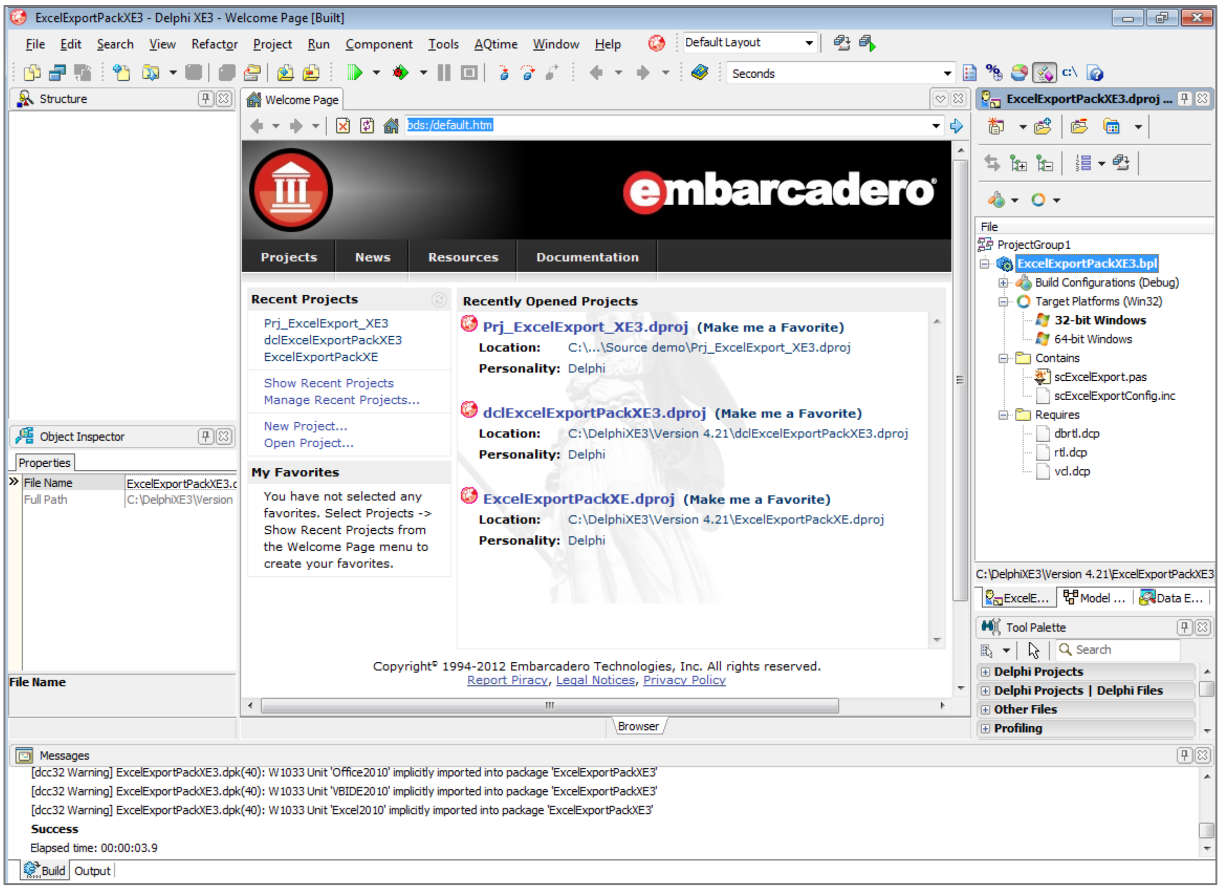
Files	Path
Contains	
scExcelExport.dcr	C:\DelphiXE3\Version 4.21
scExcelExport.pas	C:\DelphiXE3\Version 4.21
Requires	
dbrtl.dcp	
dclOffice2k.dcp	
rtl.dcp	
vcl.dcp	



Files	Path
Contains	
scExcelExportRe...	C:\DelphiXE3\Version
Requires	
designide.dcp	
ExcelExportPack...	
rtl.dcp	
vcl.dcp	

```
elExportEditor
 TscExcelExportEditor.Edit;

ed;
erb(1);
```

Build and install packages in Delphi XE3



Recent Projects
Prj_ExcelExport_XE3
dclExcelExportPackXE3
ExcelExportPackXE
Show Recent Projects
Manage Recent Projects...
New Project...
Open Project...

My Favorites
You have not selected any favorites. Select Projects -> Show Recent Projects from the Welcome Page menu to create your favorites.

Recently Opened Projects

Prj_ExcelExport_XE3.dproj (Make me a Favorite)
Location: C:\...\Source demo\Prj_ExcelExport_XE3.dproj
Personality: Delphi

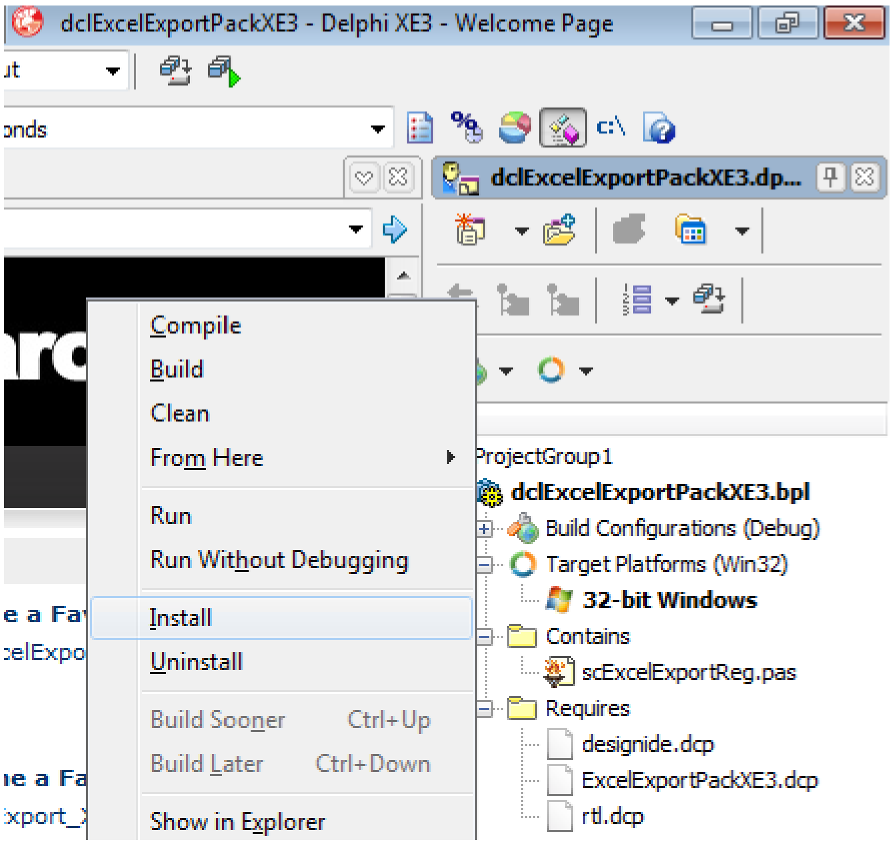
dclExcelExportPackXE3.dproj (Make me a Favorite)
Location: C:\DelphiXE3\Version 4.21\dclExcelExportPackXE3.dproj
Personality: Delphi

ExcelExportPackXE.dproj (Make me a Favorite)
Location: C:\DelphiXE3\Version 4.21\ExcelExportPackXE.dproj
Personality: Delphi

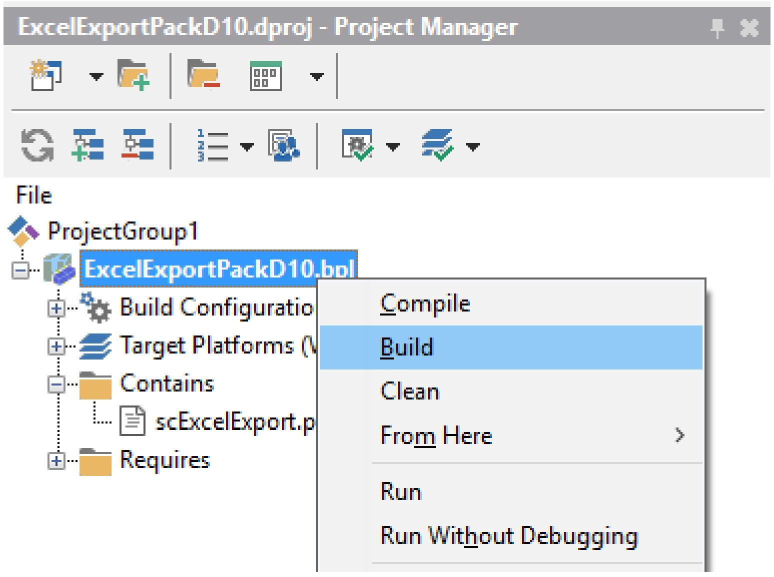
Copyright© 1994-2012 Embarcadero Technologies, Inc. All rights reserved.
Report Piracy, Legal Notices, Privacy Policy

```
[dcc32 Warning] ExcelExportPackXE3.dpk(40): W1033 Unit 'Office2010' implicitly imported into package 'ExcelExportPackXE3'
[dcc32 Warning] ExcelExportPackXE3.dpk(40): W1033 Unit 'VBIDE2010' implicitly imported into package 'ExcelExportPackXE3'
[dcc32 Warning] ExcelExportPackXE3.dpk(40): W1033 Unit 'Excel2010' implicitly imported into package 'ExcelExportPackXE3'
Success
Elapsed time: 00:00:03.9
```

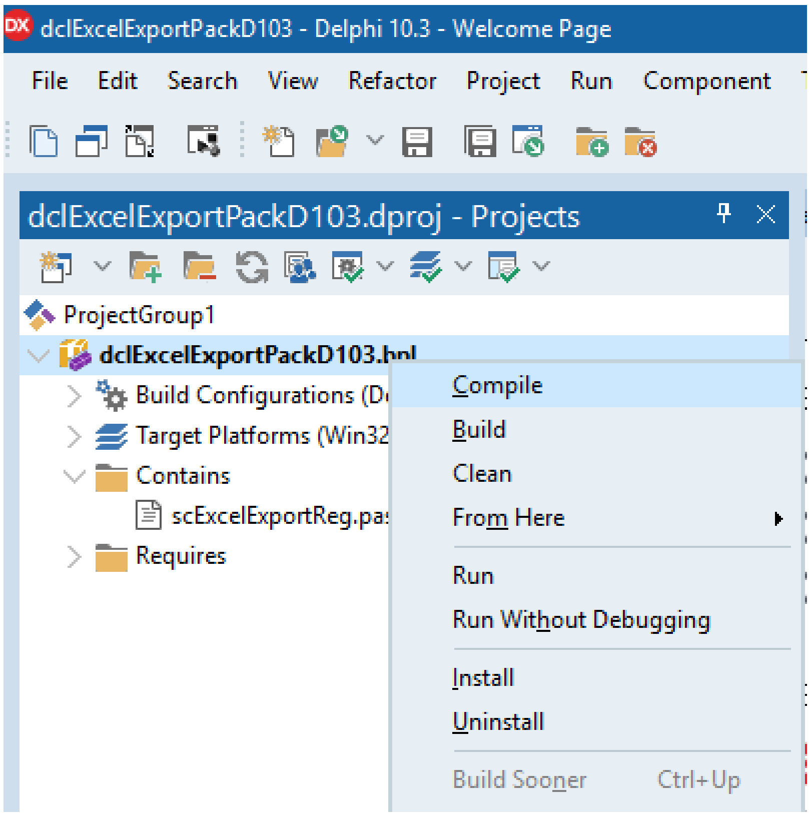
C:\DelphiXE3\Version 4.21\ExcelExportPackXE3



Build run-time package in Delphi 10 Seattle



Build and install design-time package in Delphi 10.3 Rio



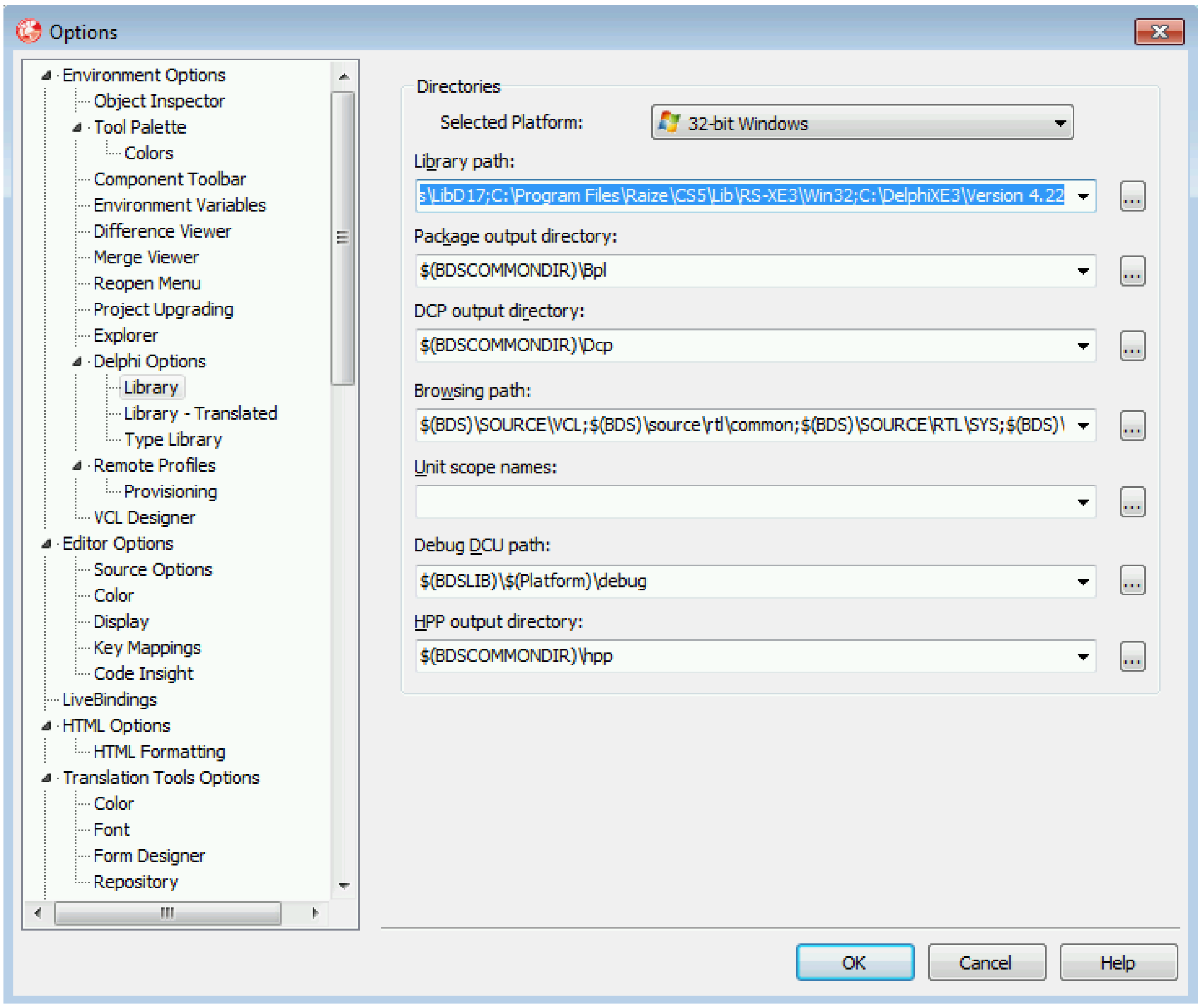
- When you like to add the component to an existing package, add the unit scExcelExport to a run-time package. Make sure the DCP file dclOffice is added as required. This file can be found in the Delphi Lib folder. The unit scExcelExportReg.pas contains the registration and property and component editor. 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 before compiling the demo project.

- **Tools >> Options >> Delphi Options >> Library**
- **Tools >> Options >> Language >> Delphi Options >> Library**
- **Tools >> Options >> Language >> Delphi >> Library**

Library path in Delphi XE3



Directories

Selected Platform: 32-bit Windows

Library path:
s\LibD17;C:\Program Files\Raize\CS5\Lib\RS-XE3\Win32;C:\DelphiXE3\Version 4.22

Package output directory:
$(BDSCOMMONDIR)\Bpl

DCP output directory:
$(BDSCOMMONDIR)\Dcp

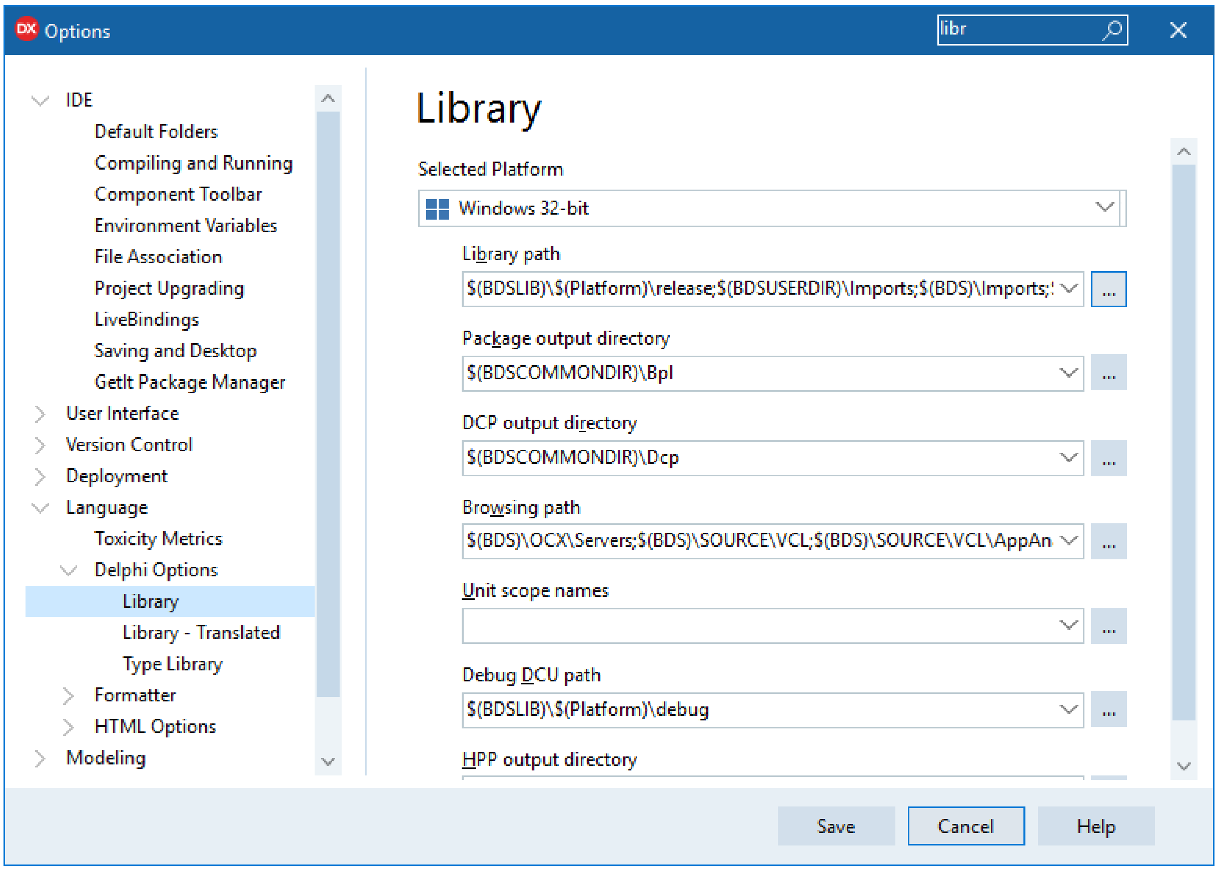
Browsing path:
$(BDS)\SOURCE\VCL;$(BDS)\source\rtl\common;$(BDS)\SOURCE\RTL\SYS;$(BDS)\

Unit scope names:

Debug DCU path:
$(BDSLIB)\$(Platform)\debug

HPP output directory:
$(BDSCOMMONDIR)\hpp

Library Path in Delphi 10.3 Rio



Library

Selected Platform: Windows 32-bit

Library path: $(BDSLIB)\$(Platform)\release;$(BDSUSERDIR)\Imports;$(BDS)\Imports;

Package output directory: $(BDSCOMMONDIR)\Bpl

DCP output directory: $(BDSCOMMONDIR)\Dcp

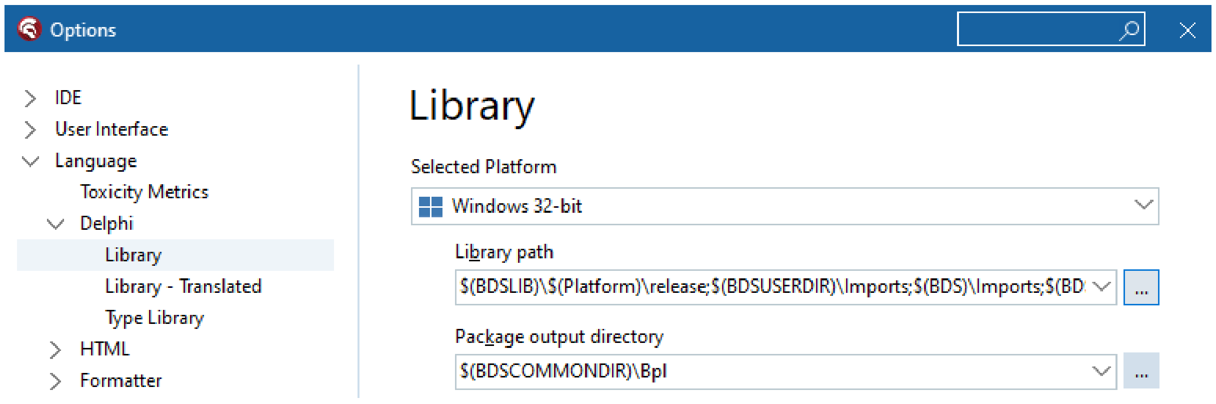
Browsing path: $(BDS)\OCX\Servers;$(BDS)\SOURCE\VCL;$(BDS)\SOURCE\VCL\AppAn

Unit scope names:

Debug DCU path: $(BDSLIB)\$(Platform)\debug

HPP output directory

Library Path in Delphi 11 Alexandria



Library

Selected Platform: Windows 32-bit

Library path: $(BDSLIB)\$(Platform)\release;$(BDSUSERDIR)\Imports;$(BDS)\Imports;$(BD

Package output directory: $(BDSCOMMONDIR)\Bpl

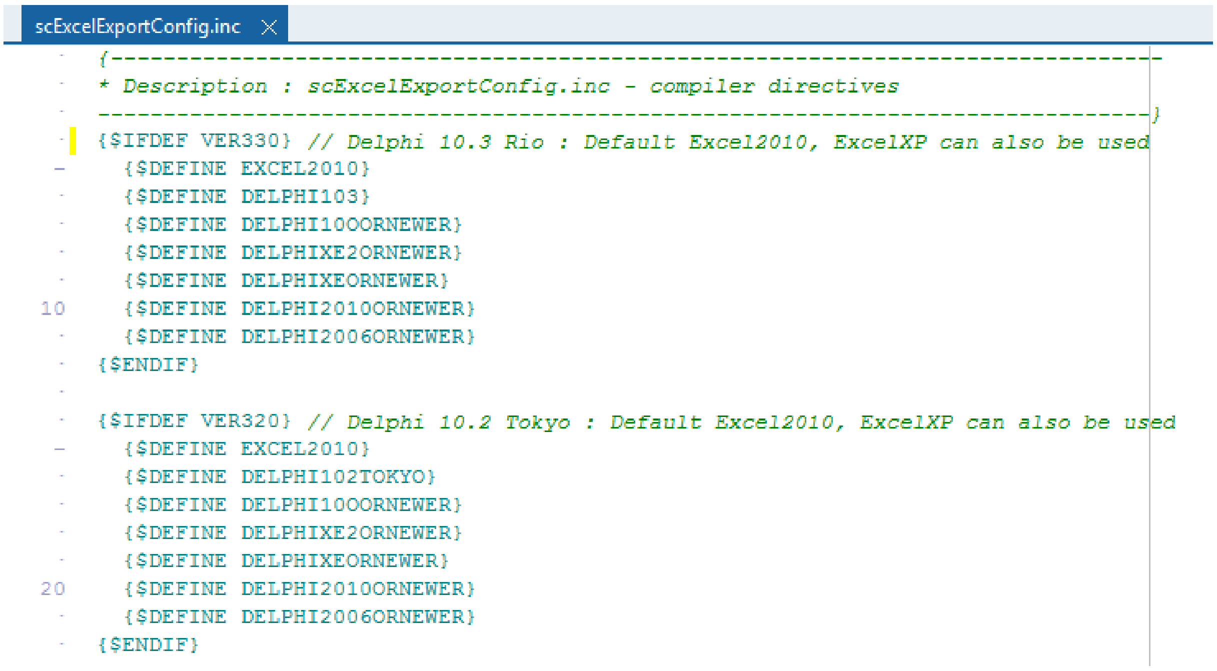
Type libraries and Excel versions

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.

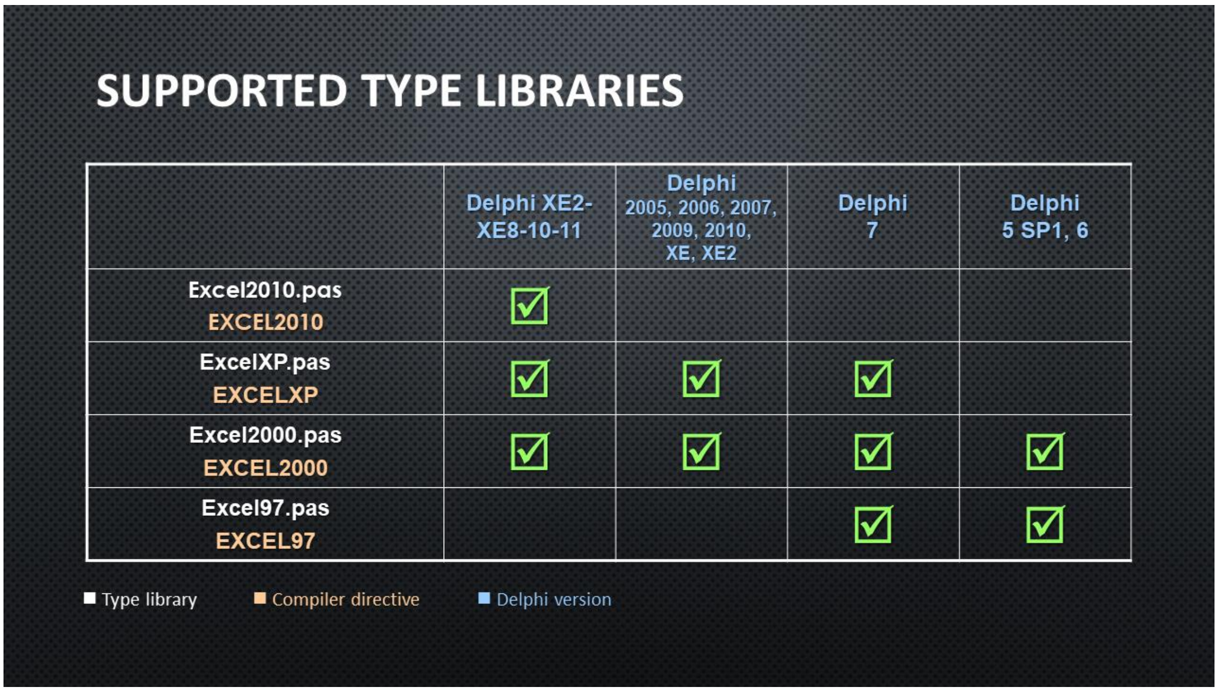
Configure a type library in scExcelExportConfig.inc

- 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.



```
{---------------------------------------------------------------------------------
* Description : scExcelExportConfig.inc - compiler directives
---------------------------------------------------------------------------------}
{$IFDEF VER330} // Delphi 10.3 Rio : Default Excel2010, ExcelXP can also be used
  {$DEFINE EXCEL2010}
  {$DEFINE DELPHI103}
  {$DEFINE DELPHI10ORNEWER}
  {$DEFINE DELPHIXE2ORNEWER}
  {$DEFINE DELPHIXEORNEWER}
  {$DEFINE DELPHI2010ORNEWER}
  {$DEFINE DELPHI2006ORNEWER}
{$ENDIF}

{$IFDEF VER320} // Delphi 10.2 Tokyo : Default Excel2010, ExcelXP can also be used
  {$DEFINE EXCEL2010}
  {$DEFINE DELPHI102TOKYO}
  {$DEFINE DELPHI10ORNEWER}
  {$DEFINE DELPHIXE2ORNEWER}
  {$DEFINE DELPHIXEORNEWER}
  {$DEFINE DELPHI2010ORNEWER}
  {$DEFINE DELPHI2006ORNEWER}
{$ENDIF}
```

Supported type libraries

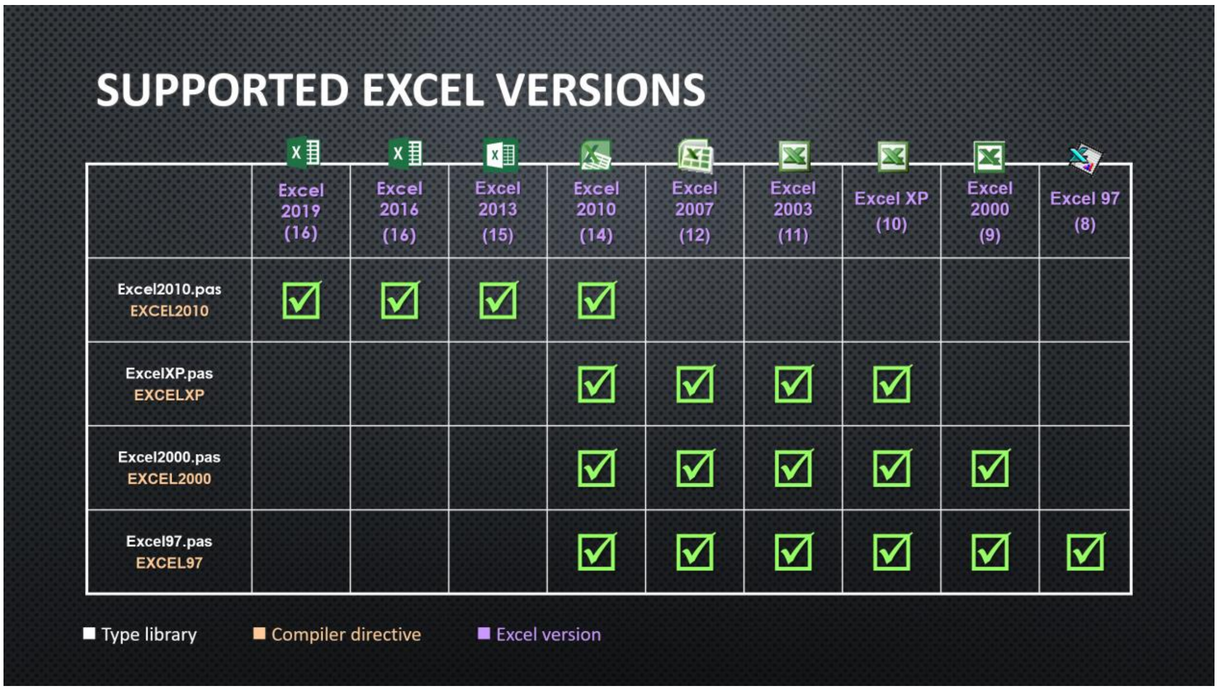


SUPPORTED TYPE LIBRARIES

	Delphi XE2-XE8-10-11	Delphi 2005, 2006, 2007, 2009, 2010, XE, XE2	Delphi 7	Delphi 5 SP1, 6
Excel2010.pas EXCEL2010	✓			
ExcelXP.pas EXCELXP	✓	✓	✓	
Excel2000.pas EXCEL2000	✓	✓	✓	✓
Excel97.pas EXCEL97			✓	✓

■ Type library ■ Compiler directive ■ Delphi version

Supported Excel versions



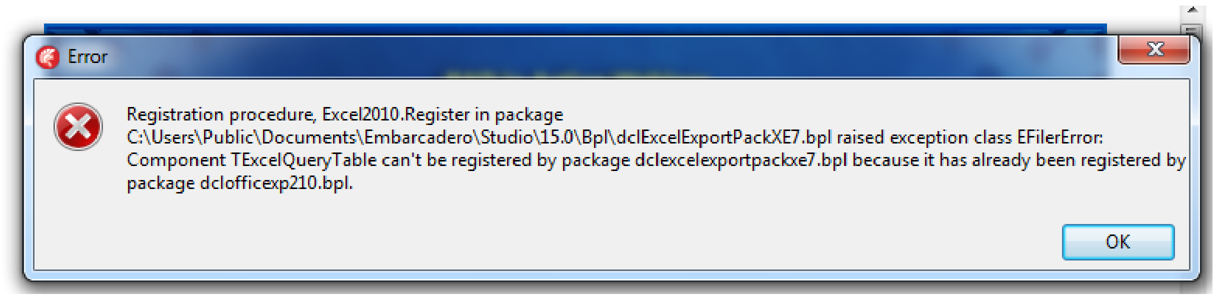
SUPPORTED EXCEL VERSIONS

	Excel 2019 (16)	Excel 2016 (16)	Excel 2013 (15)	Excel 2010 (14)	Excel 2007 (12)	Excel 2003 (11)	Excel XP (10)	Excel 2000 (9)	Excel 97 (8)
Excel2010.pas EXCEL2010	✓	✓	✓	✓					
ExcelXP.pas EXCELXP				✓	✓	✓	✓		
Excel2000.pas EXCEL2000				✓	✓	✓	✓	✓	
Excel97.pas EXCEL97				✓	✓	✓	✓	✓	✓

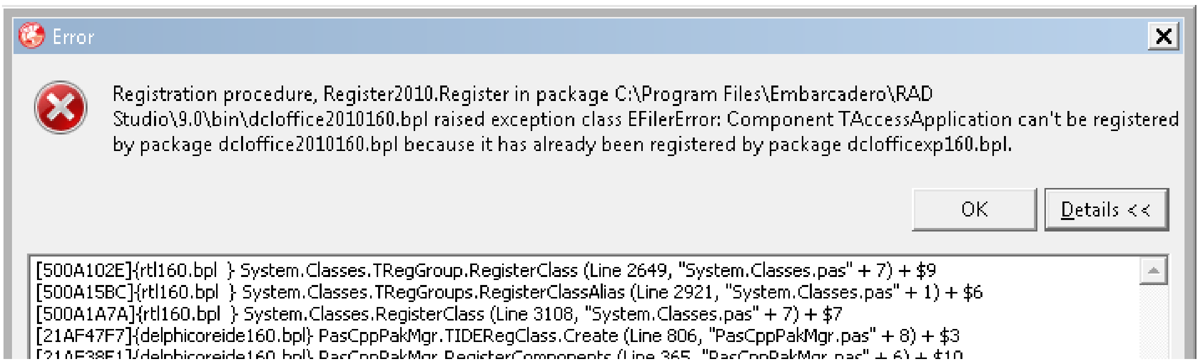
■ Type library ■ Compiler directive ■ Excel version

Known issues

Component can’t be registered



Registration procedure, Excel2010.Register in package C:\Users\Public\Documents\Embarcadero\Studio\15.0\Bpl\dclExcelExportPackXE7.bpl raised exception class EFilerError: Component TExcelQueryTable can't be registered by package dclexcelexportpackxe7.bpl because it has already been registered by package dclofficexp210.bpl.



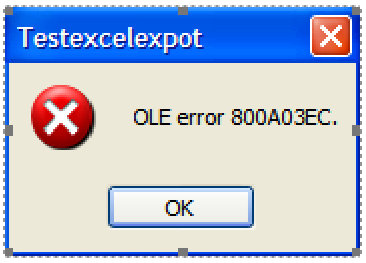
Registration procedure, Register2010.Register in package C:\Program Files\Embarcadero\RAD Studio\9.0\bin\dcloffice2010160.bpl raised exception class EFilerError: Component TAccessApplication can't be registered by package dcloffice2010160.bpl because it has already been registered by package dclofficexp160.bpl.

```
[500A102E]{rtl160.bpl  } System.Classes.TRegGroup.RegisterClass (Line 2649, "System.Classes.pas" + 7) + $9
[500A15BC]{rtl160.bpl  } System.Classes.TRegGroups.RegisterClassAlias (Line 2921, "System.Classes.pas" + 1) + $6
[500A1A7A]{rtl160.bpl  } System.Classes.RegisterClass (Line 3108, "System.Classes.pas" + 7) + $7
[21AF47F7]{delphicoreide160.bpl} PasCppPakMgr.TIDERegClass.Create (Line 806, "PasCppPakMgr.pas" + 8) + $3
[21AE38E1]{delphicoreide160.bpl} PasCppPakMgr.RegisterComponents (Line 365, "PasCppPakMgr.pas" + 6) + $10
```

This error can occur when installing the design-time package (dclExelExport).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



OLE error 800A03EC.

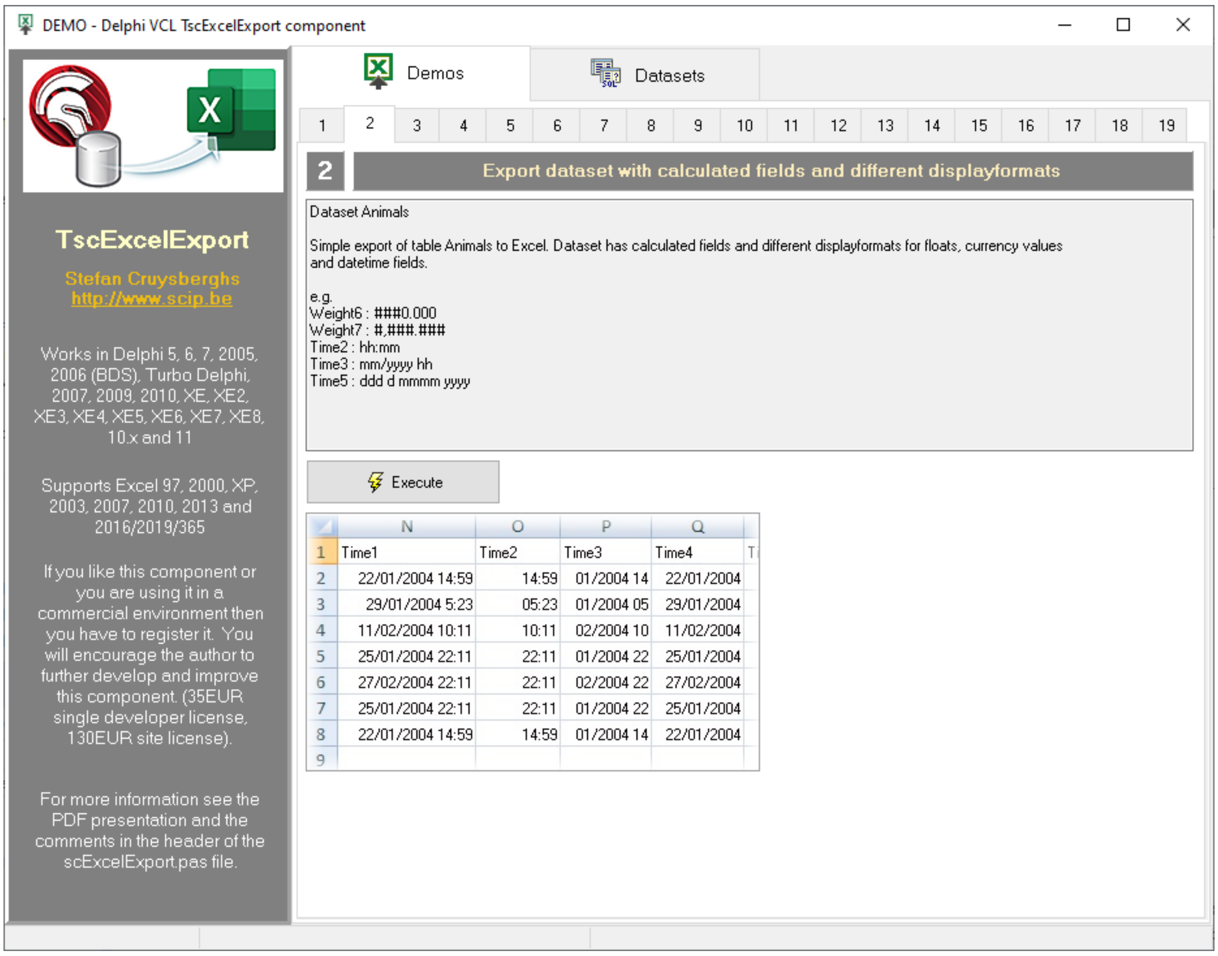
If your application that exports data to Excel is running perfectly on one computer, but it throws an OLE-error (e.g. 800A03EC) or "Interface not supported" or “Client not registered” exception on an another computer, then there is something wrong with the registration of the COM/OLE libraries (registry 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 options 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.



TscExcelExport

Stefan Cruysberghs
http://www.scip.be

Works in Delphi 5, 6, 7, 2005, 2006 (BDS), Turbo Delphi, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10.x and 11

Supports Excel 97, 2000, XP, 2003, 2007, 2010, 2013 and 2016/2019/365

If you like this component or you are using it in a commercial environment then you have to register it. You will encourage the author to further develop and improve this component. (35EUR single developer license, 130EUR site license).

For more information see the PDF presentation and the comments in the header of the scExcelExport.pas file.

2 Export dataset with calculated fields and different displayformats

Dataset Animals

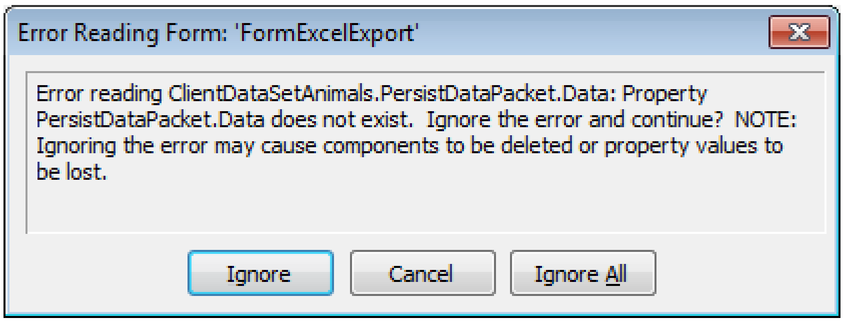
Simple export of table Animals to Excel. Dataset has calculated fields and different displayformats for floats, currency values and datetime fields.

e.g.
Weight6 : ###0.000
Weight7 : #,###.###
Time2 : hh:mm
Time3 : mm/yyyy hh
Time5 : ddd d mmmm yyyy

	N	O	P	Q
1	Time1	Time2	Time3	Time4
2	22/01/2004 14:59	14:59	01/2004 14	22/01/2004
3	29/01/2004 5:23	05:23	01/2004 05	29/01/2004
4	11/02/2004 10:11	10:11	02/2004 10	11/02/2004
5	25/01/2004 22:11	22:11	01/2004 22	25/01/2004
6	27/02/2004 22:11	22:11	02/2004 22	27/02/2004
7	25/01/2004 22:11	22:11	01/2004 22	25/01/2004
8	22/01/2004 14:59	14:59	01/2004 14	22/01/2004
9				

Error reading form in older Delphi versions

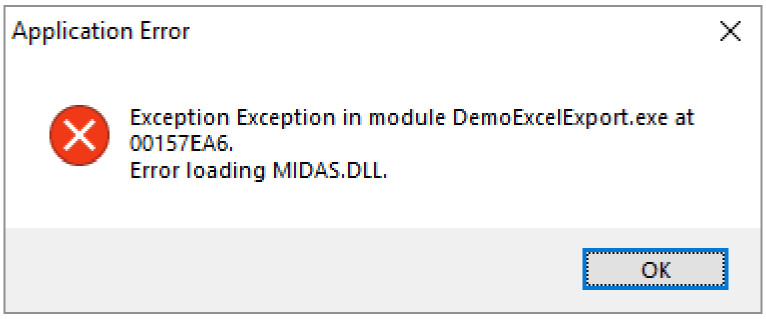
Error reading PersistDataPacket, ExplicitTop, ExplicitLeft, ExplicitWidth and ExplicitHeight properties when opening the demo application in older Delphi versions.


Error Reading Form: 'FormExcelExport'

Error reading ClientDataSetAnimals.PersistDataPacket.Data: Property PersistDataPacket.Data does not exist. Ignore the error and continue? NOTE: Ignoring the error may cause components to be deleted or property values to be lost.

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

Loading MIDAS.DLL in demo application


Application Error

Exception Exception in module DemoExcelExport.exe at 00157EA6.
Error loading MIDAS.DLL.

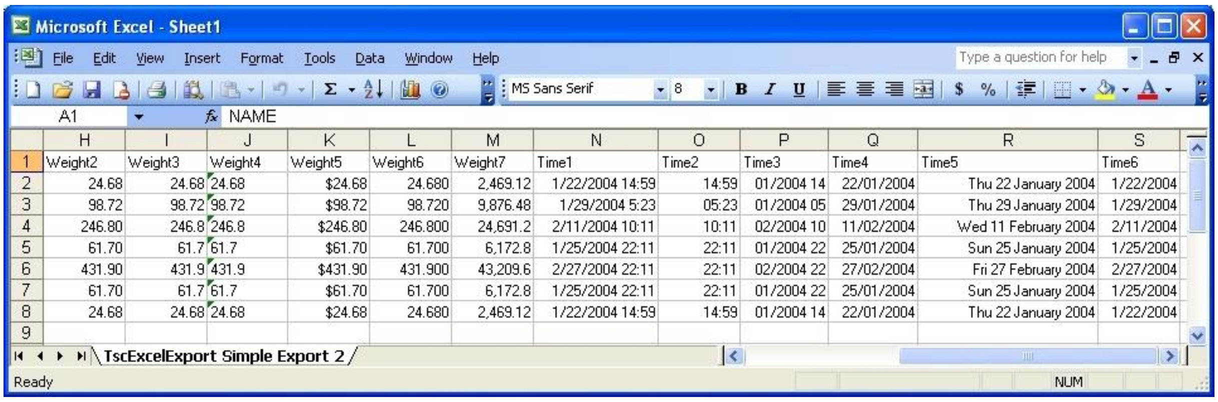
Error loading MIDAS.DLL when running the demo application.

- 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

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


Microsoft Excel - Sheet1

A1 NAME

	H	I	J	K	L	M	N	O	P	Q	R	S
1	Weight2	Weight3	Weight4	Weight5	Weight6	Weight7	Time1	Time2	Time3	Time4	Time5	Time6
2	24.68	24.68	24.68	$24.68	24.680	2,469.12	1/22/2004 14:59	14:59	01/2004 14	22/01/2004	Thu 22 January 2004	1/22/2004
3	98.72	98.72	98.72	$98.72	98.720	9,876.48	1/29/2004 5:23	05:23	01/2004 05	29/01/2004	Thu 29 January 2004	1/29/2004
4	246.80	246.8	246.8	$246.80	246.800	24,691.2	2/11/2004 10:11	10:11	02/2004 10	11/02/2004	Wed 11 February 2004	2/11/2004
5	61.70	61.7	61.7	$61.70	61.700	6,172.8	1/25/2004 22:11	22:11	01/2004 22	25/01/2004	Sun 25 January 2004	1/25/2004
6	431.90	431.9	431.9	$431.90	431.900	43,209.6	2/27/2004 22:11	22:11	02/2004 22	27/02/2004	Fri 27 February 2004	2/27/2004
7	61.70	61.7	61.7	$61.70	61.700	6,172.8	1/25/2004 22:11	22:11	01/2004 22	25/01/2004	Sun 25 January 2004	1/25/2004
8	24.68	24.68	24.68	$24.68	24.680	2,469.12	1/22/2004 14:59	14:59	01/2004 14	22/01/2004	Thu 22 January 2004	1/22/2004
9												

TscExcelExport Simple Export 2

Ready

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

```
scExcelExport1.WorksheetName := 'MyDataset';
scExcelExport1.Dataset:=MyDataset;
scExcelExport1.StyleColumnWidth:=cwOwnerWidth;
scExcelExport1.ColumnWidth := 20;
scExcelExport1.HeaderText.Text := 'Header';
scExcelExport1.BeginRowHeader := 2;
scExcelExport1.FontTitles := LabelTitle.Font;
scExcelExport1.FontTitles.Orientation := 45;
scExcelExport1.BorderTitles.BackColor := clYellow;
scExcelExport1.BorderTitles.BorderColor := clRed;
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.ExcelWorkSheet.Range['A1','A10'].Value := 'Delphi';
  scExcelExport1.SaveAs('c:\test.xls',ffXLS); finally
  scExcelExport1.Disconnect;
end;
```

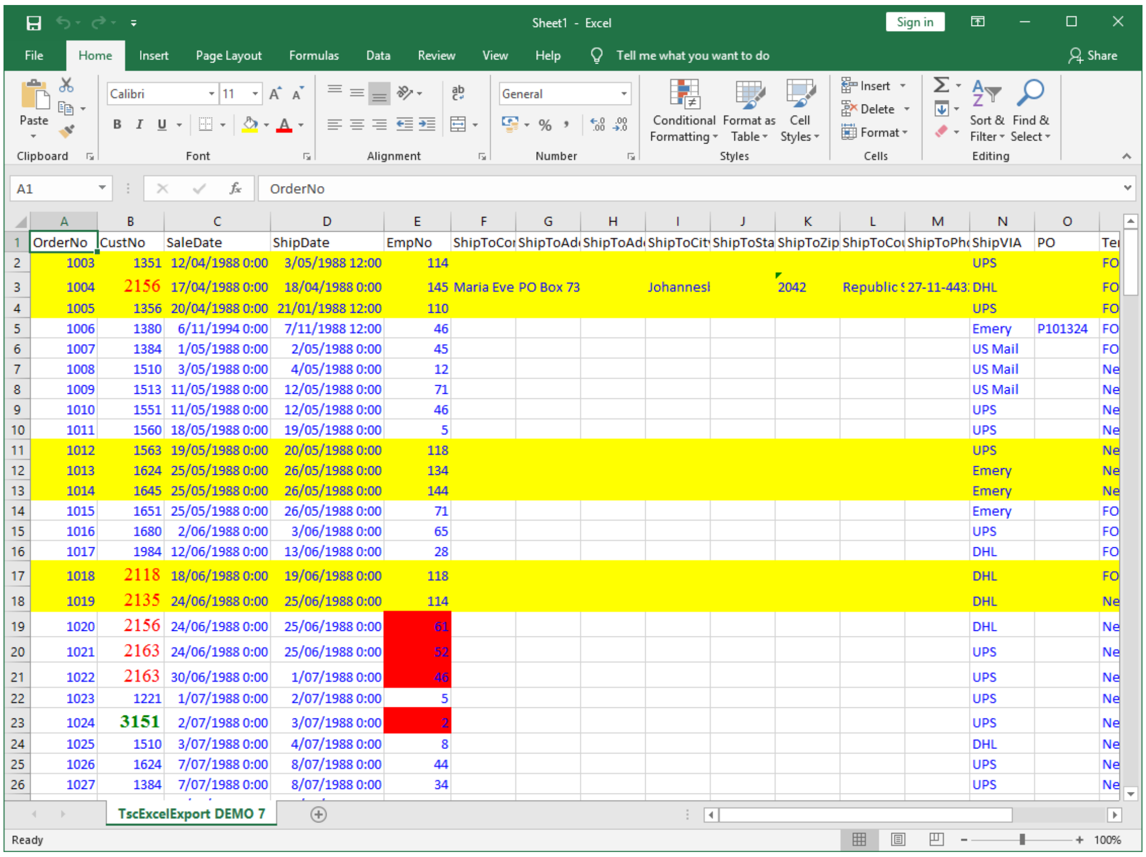
Example 3: Export multiple datasets

```
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;
```

Example 4: Change background color and font style in the OnGetCellStyle event

See full example 7 in demo application

```
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
      ColorBackground := clRed;
  end;
  if Field.DataSet.FieldByName('EmpNo').Value > 100 then
    ColorBackground := clYellow;
end;
```

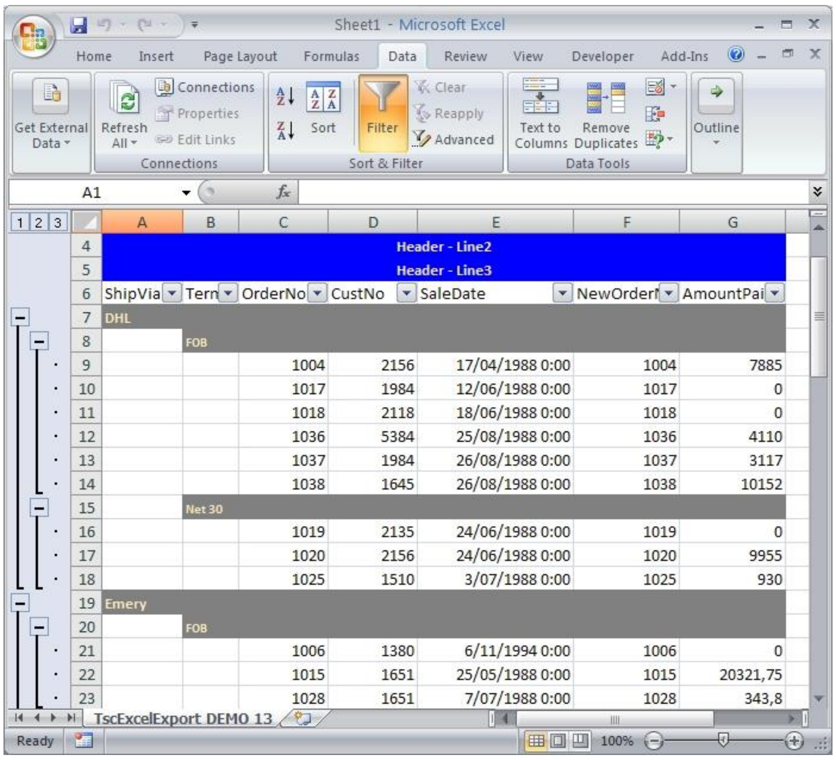


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	OrderNo	CustNo	SaleDate	ShipDate	EmpNo	ShipToCoı	ShipToAd	ShipToAd	ShipToCit	ShipToSta	ShipToZip	ShipToCoı	ShipToPh	ShipVIA	PO	Te
2	1003	1351	12/04/1988 0:00	3/05/1988 12:00	114									UPS		FO
3	1004	2156	17/04/1988 0:00	18/04/1988 0:00	145	Maria Eve	PO Box 73		Johannesl		2042	Republic !	27-11-443:	DHL		FO
4	1005	1356	20/04/1988 0:00	21/01/1988 12:00	110									UPS		FO
5	1006	1380	6/11/1994 0:00	7/11/1988 12:00	46									Emery	P101324	FO
6	1007	1384	1/05/1988 0:00	2/05/1988 0:00	45									US Mail		FO
7	1008	1510	3/05/1988 0:00	4/05/1988 0:00	12									US Mail		Ne
8	1009	1513	11/05/1988 0:00	12/05/1988 0:00	71									US Mail		Ne
9	1010	1551	11/05/1988 0:00	12/05/1988 0:00	46									UPS		Ne
10	1011	1560	18/05/1988 0:00	19/05/1988 0:00	5									UPS		Ne
11	1012	1563	19/05/1988 0:00	20/05/1988 0:00	118									UPS		Ne
12	1013	1624	25/05/1988 0:00	26/05/1988 0:00	134									Emery		Ne
13	1014	1645	25/05/1988 0:00	26/05/1988 0:00	144									Emery		Ne
14	1015	1651	25/05/1988 0:00	26/05/1988 0:00	71									Emery		FO
15	1016	1680	2/06/1988 0:00	3/06/1988 0:00	65									UPS		FO
16	1017	1984	12/06/1988 0:00	13/06/1988 0:00	28									DHL		FO
17	1018	2118	18/06/1988 0:00	19/06/1988 0:00	118									DHL		FO
18	1019	2135	24/06/1988 0:00	25/06/1988 0:00	114									DHL		Ne
19	1020	2156	24/06/1988 0:00	25/06/1988 0:00	61									DHL		Ne
20	1021	2163	24/06/1988 0:00	25/06/1988 0:00	52									UPS		Ne
21	1022	2163	30/06/1988 0:00	1/07/1988 0:00	46									UPS		Ne
22	1023	1221	1/07/1988 0:00	2/07/1988 0:00	5									UPS		Ne
23	1024	**3151**	2/07/1988 0:00	3/07/1988 0:00	2									UPS		Ne
24	1025	1510	3/07/1988 0:00	4/07/1988 0:00	8									DHL		Ne
25	1026	1624	7/07/1988 0:00	8/07/1988 0:00	44									UPS		Ne
26	1027	1384	7/07/1988 0:00	8/07/1988 0:00	34									UPS		Ne

Example 5: Grouping

See full example 13 in demo application

```
try
    scExcelExport1.Dataset:=DatasetOrders;
    scExcelExport1.GroupFields.Clear;
    scExcelExport1.GroupFields.Add('ShipVia');
    scExcelExport1.GroupFields.Add('Terms');
    scExcelExport1.ExportDataset;
  finally
    scExcelExport1.Disconnect;
  end;
```



	A	B	C	D	E	F	G
4	Header - Line2						
5	Header - Line3						
6	ShipVia	Tern	OrderNo	CustNo	SaleDate	NewOrderN	AmountPai
7	DHL						
8		FOB					
9			1004	2156	17/04/1988 0:00	1004	7885
10			1017	1984	12/06/1988 0:00	1017	0
11			1018	2118	18/06/1988 0:00	1018	0
12			1036	5384	25/08/1988 0:00	1036	4110
13			1037	1984	26/08/1988 0:00	1037	3117
14			1038	1645	26/08/1988 0:00	1038	10152
15		Net 30					
16			1019	2135	24/06/1988 0:00	1019	0
17			1020	2156	24/06/1988 0:00	1020	9955
18			1025	1510	3/07/1988 0:00	1025	930
19	Emery						
20		FOB					
21			1006	1380	6/11/1994 0:00	1006	0
22			1015	1651	25/05/1988 0:00	1015	20321,75
23			1028	1651	7/07/1988 0:00	1028	343,8

Example 6: Export worksheets

See full example 9 in demo application

```
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

```
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

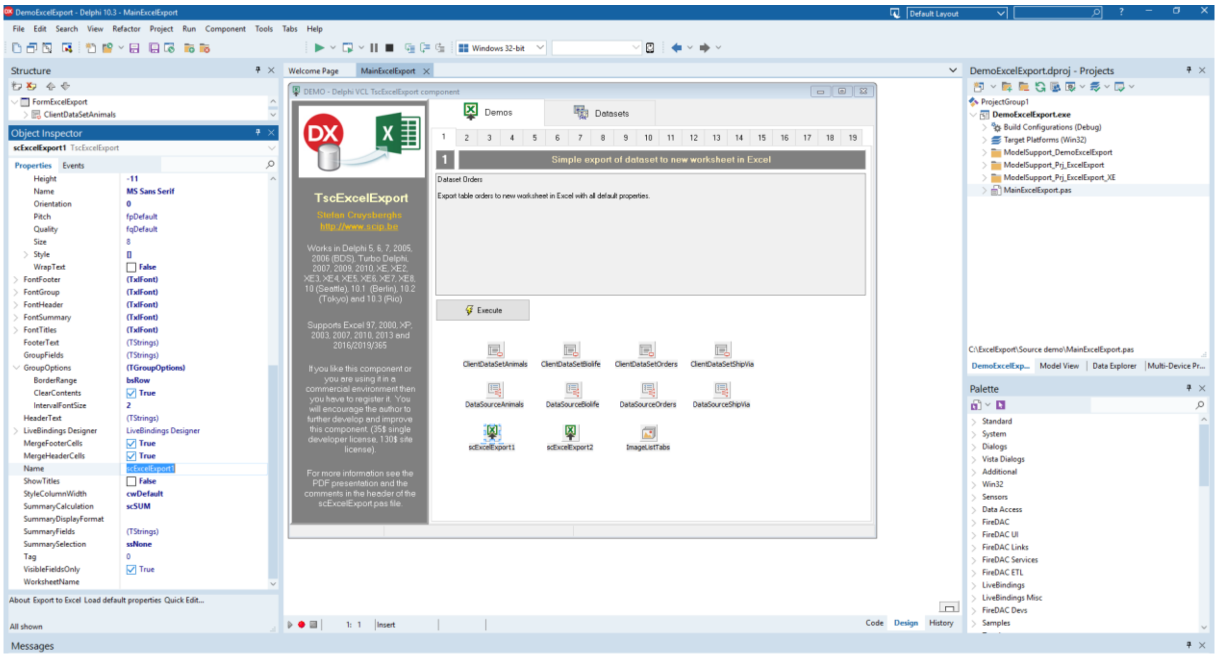
```
try
    scExcelExport1.ExcelVisible:=True;
    scExcelExport1.LoadDefaultProperties;
    scExcelExport1.Dataset:=DatasetAnimals;
    scExcelExport1.WorksheetName:='TscExcelExport DEMO 14';
    scExcelExport1.ConnectTo := ctNewExcel;
    scExcelExport1.Connect;
    scExcelExport1.ExcelWorkSheet.Range['A2','C8'].Borders.Color := clRed;

    scExcelExport1.ExportDataset;

    scExcelExport1.ExcelWorkSheet.Range['B5','E7'].Cells.Clear;

    scExcelExport1.ExcelWorkSheet.Range[Format('A%d',
      [scExcelExport1.EndRowData+3]),
      Format('A%d',[scExcelExport1.EndRowData+3])].Font.Size := 16;
    scExcelExport1.ExcelWorkSheet.Range[Format('A%d',
      [scExcelExport1.EndRowData+3]),
      Format('A%d',[scExcelExport1.EndRowData+3])].Value2 :=
      'Adding extra information to Excel worksheet';

    scExcelExport1.ExcelWorkSheet.Range['M1','M1'].Value2 := 10;
    scExcelExport1.ExcelWorkSheet.Range['M2','M2'].Value2 := 5;
    scExcelExport1.ExcelWorkSheet.Range['M3','M3'].Value2 := '=M1+M2';
    scExcelExport1.ExcelWorkSheet.Range['M3','M3'].Font.Color := clRed;

    scExcelExport1.ExcelWorkSheet.Range['N1','N20'].Value2 :=
      'Filling extra column with autofit';
    scExcelExport1.ExcelWorkSheet.Range['N1','N20'].Font.Size := 12;
    scExcelExport1.ExcelWorkSheet.Range['N1','N20'].Font.Color := clBlue;
    scExcelExport1.ExcelWorkSheet.Range['N1','N20'].EntireColumn.Autofit;

    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


DemoExcelExport - Delphi 10.3 - MainExcelExport
Structure
Object Inspector
scExcelExport1 TscExcelExport
Welcome Page
MainExcelExport
DEMO - Delphi VCL TscExcelExport component
Demos
Datasets
Execute
DemoExcelExport.dproj - Projects
Palette
Messages

1 Simple export of dataset to new worksheet in Excel

Dataset Orders

Export table orders to new worksheet in Excel with all default properties.

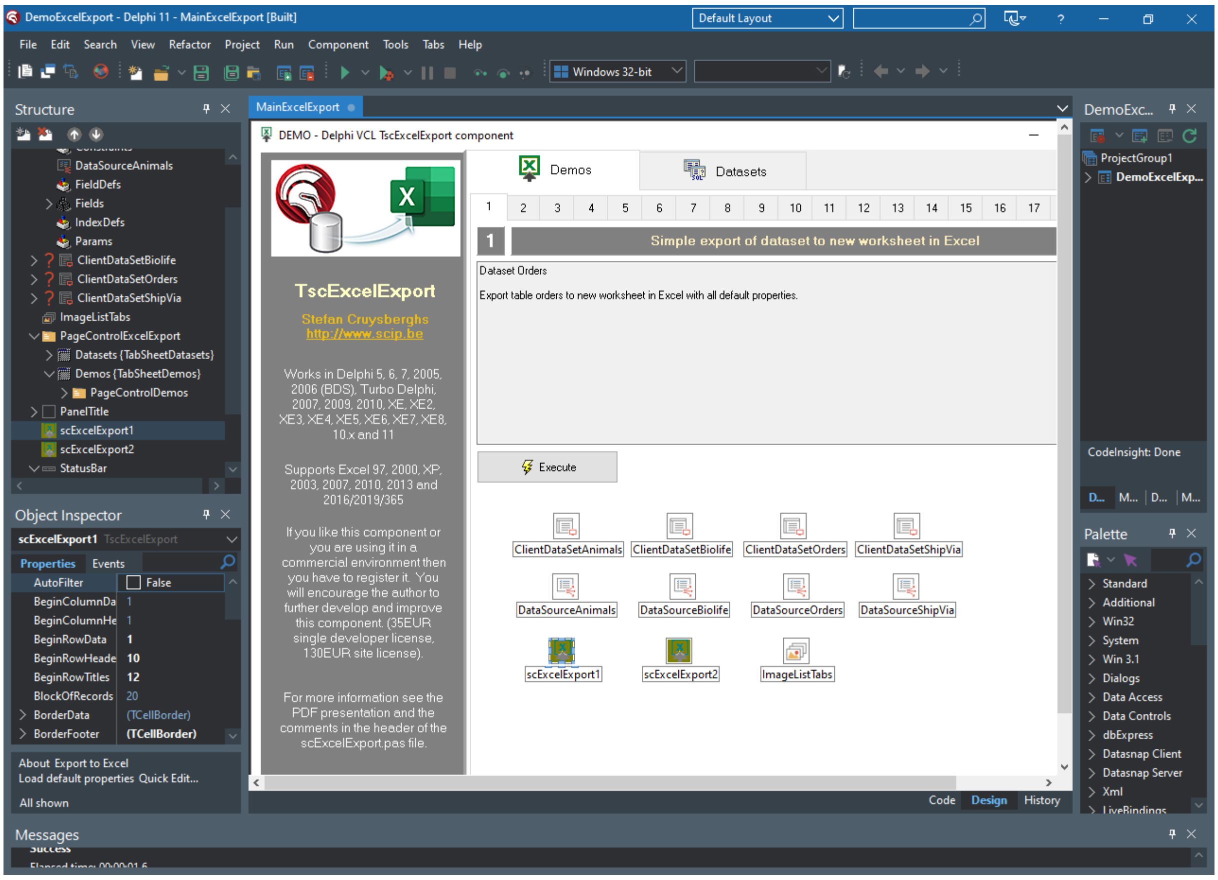
TscExcelExport
Stefan Cruysberghs
http://www.scip.be

Works in Delphi 5, 6, 7, 2005, 2006 (BDS), Turbo Delphi, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10 (Seattle), 10.1 (Berlin), 10.2 (Tokyo) and 10.3 (Rio)

Supports Excel 97, 2000, XP, 2003, 2007, 2010, 2013 and 2016/2019/365

If you like this component or you are using it in a commercial environment then you have to register it. You will encourage the author to further develop and improve this component. (35$ single developer license, 130$ site license).

For more information see the PDF presentation and the comments in the header of the scExcelExport.pas file.



AutoFilter	False
BeginColumnDa	1
BeginColumnHe	1
BeginRowData	1
BeginRowHeade	10
BeginRowTitles	12
BlockOfRecords	20
BorderData	(TCellBorder)
BorderFooter	(TCellBorder)

1 Simple export of dataset to new worksheet in Excel

Dataset Orders

Export table orders to new worksheet in Excel with all default properties.

TscExcelExport
Stefan Cruysberghs
http://www.scip.be

Works in Delphi 5, 6, 7, 2005, 2006 (BDS), Turbo Delphi, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8, 10.x and 11

Supports Excel 97, 2000, XP, 2003, 2007, 2010, 2013 and 2016/2019/365

If you like this component or you are using it in a commercial environment then you have to register it. You will encourage the author to further develop and improve this component. (35EUR single developer license, 130EUR site license).

For more information see the PDF presentation and the comments in the header of the scExcelExport.pas file.

Success
Elapsed time: 00:00:01.6

Version history

Version 4.42 (October 2021)

- Added support for Delphi 11 Alexandria

Version 4.41 (June 2020)

- Added support for Delphi 10.4 Sydney

Version 4.4 (March 2019)

- Added Font.VerticalAlign (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 (bèta)
- 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' -> Excel97/2003 : Arial size 10, Excel 2007 : Calibri size 11
- 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.