

SQL Manager.net™

EMS® Software Development



Advanced Data Export .NET User's Manual

© 1999-2025 EMS Software Development

Advanced Data Export .NET User's Manual

© 1999-2025 EMS Software Development

All rights reserved.

This manual documents EMS Advanced Data Export .NET, version 1.6.

No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Use of this documentation is subject to the following terms: you may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way.

Document generated on: 15.01.2025

Table of Contents

Part I Welcome to EMS Advanced Data Export .NET Suite!	14
Suite overview	14
What's new	15
Product installation	16
Register EMS Advanced Data Export .NET	17
Product licensing information	18
Part II EMS Advanced Data Export .NET Component Suite	25
SqlManager.QExport.Access namespace	25
SqlManager.QExport.Access namespace Reference	25
QExportAccess component	26
QExportAccess component Reference.....	26
QExportAccess Properties.....	27
AutoCreateDatabase.....	28
AutoCreateTable.....	29
DatabaseName	30
TableName	31
Provider	32
SqlManager.QExport.Base namespace	33
SqlManager.QExport.Base namespace Reference	33
QExportBase component	34
QExportBase component Reference.....	34
Properties.....	35
Version	36
Aborted	37
About	38
Allow Captions	39
AutoCalcColWidth.....	40
AutoCalcStrType.....	41
Captions	42
ColumnsAlign	43
ColumnsWidth	44
Command	45
DataTable	46
EncodingType	47
ExportedFields	48
ExportRecCount	49
ExportSource	50
Footer	51
Formats	52
Header	53

ListView	54
NotTruncatableFields	55
SkipRecCount	56
Title	57
UserFormats	58
Methods	59
Abort	60
Execute	61
ExportToStream	62
LoadPropertiesFromFile	63
SavePropertiesToFile	64
Events	65
BeforeExportRow	66
BeginExport	67
EndExport	68
ExportedRecord	69
FetchedRecord	70
GetCellParams	71
GetExportText	72
SkippedRecord	73
StopExport	74
QExportText component	75
QExportText Properties	76
FileName property	77
PrintFile property	78
Show File property	79
QExportCol object	80
QExportCol Properties	81
Name property	82
Value property	83
QExportFormats object	84
QExportFormats Properties	85
BooleanFalse property	86
BooleanTrue property	87
CurrencyFormat property	88
DateTimeFormat property	89
FloatFormat property	90
IntegerFormat property	91
NullString property	92
TimeFormat property	93
QExportFormats Methods	94
ResetFormats method	95
QExportRow object	96
QExportRow Properties	97
QExportRow.this[]	98
SqlManager.QExport.Clipboard namespace	99
SqlManager.QExport.Clipboard namespace Reference	99
QExportClipboard component	100
QExportClipboard component Reference	100
QExportClipboard Properties	101
ClipboardViewer	102
ExportType	103
Separator	104

Show Content	105
Spacing	106
SqlManager.QExport.DBF namespace	107
SqlManager.QExport.DBF namespace Reference	107
QExportDBF component	108
QExportDBF component Reference.....	108
QExportDBF Properties.....	109
ColumnsPrecision.....	110
DefaultFloatSize.....	111
DefaultFloatDecimal.....	112
NullValue	113
SqlManager.QExport.HTML namespace	114
SqlManager.QExport.HTML namespace Reference	114
QExportHTML component	115
QExportHTML component Reference.....	115
QExportHTML Properties.....	116
BoolAsCheckBox.....	117
CSSFileName	118
GenerateIndex	119
HTMLOptions	120
HTMLTemplate	121
InterpretTags	122
MaxRecords	123
Navigation	124
TableOptions	125
OverwriteCSSFile.....	126
UsingCSS	127
QExportHTML Methods.....	128
LoadTemplateFromFile.....	129
SaveTemplateToFile.....	130
QExportHTML Events.....	131
TableOptions object	132
TableOptions Properties.....	133
AdvancedAttributes property.....	134
BackgroundFileName.....	135
Border property.....	136
BorderColor property.....	137
CellPadding property.....	138
CellSpacing property.....	139
HeadersRow BgColor property.....	140
HeadersRow FontColor property.....	141
OddRow BgColor property.....	142
TableBgColor property.....	143
TableFontColor property.....	144
Color Using Notes	145
HtmOptions object	146
HtmOptions Properties.....	147
AdvancedAttributes property.....	148
ALinkColor property.....	149
BackgroundColor property.....	150
BackgroundFileName property.....	151
DefaultOptions property.....	152
LinkColor property.....	153

TextFont property	154
VLinkColor property	155
Color Using Notes	156
QExportHtmlNavigation object	157
QExportHtmlNavigation Properties	158
FirstLinkTitle property	159
IndexLinkTemplate property	160
IndexLinkTitle property	161
LastLinkTitle property	162
NextLinkTitle property	163
OnBottom property	164
OnTop property	165
PriorLinkTitle property	166
HtmlTemplate enum	167
UsingCSS enum	168
SqlManager.QExport.LaTeX namespace	169
SqlManager.QExport.LaTeX namespace Reference	169
QExportLaTeX component	170
QExportLaTeX component Reference	170
QExportLaTeX Properties	171
Options	172
Preamble	173
QExportLaTeX Methods	174
LaTeXOptions component	175
TLaTeXOptions - Properties	176
CodePage property	177
DocumentParams property	178
DocumentStyle property	179
Languages property	180
LaTeXVersion property	181
SqlManager.QExport.PDF namespace	182
SqlManager.QExport.PDF namespace Reference	182
QExportPDF component	183
QExportPDF component Reference	183
QExportPDF Properties	184
Options	185
PdfFont object	186
PdfFont Properties	187
BaseFont property	188
FontColor property	189
FontEncoding property	190
FontSize property	191
PdfOptions object	192
PdfOptions Properties	193
CaptionFont property	194
ColSpacing property	195
DataFont property	196
FooterFont property	197
GridLineColor property	198
GridLineWidth property	199
HeaderFont property	200
Row Spacing property	201
SqlManager.QExport.RTF namespace	202

SqlManager.QExport.RTF namespace Reference	202
QExportRTF component	203
QExportRTF component Reference.....	203
QExportRTF Properties.....	204
Options	205
QExportRTF Methods.....	206
QExportRTF Events.....	207
GetCaptionStyle.....	208
GetDataStyle	209
GetFooterStyle	210
GetHeaderStyle.....	211
RtfOptions object	212
RtfOptions Properties.....	213
PageOrientation property.....	214
CaptionAligns property.....	215
CaptionStyle property.....	216
DataStyle property.....	217
FooterStyle property.....	218
HeaderStyle property.....	219
StripStyles property.....	220
StripType property.....	221
RtfStyle object	222
RtfStyle - Properties.....	223
Alignment property.....	224
Allow Background property.....	225
Allow Highlight property.....	226
BackgroundColor property.....	227
Font property	228
HighlightColor property.....	229
RtfStyles object	230
RtfStyles - Properties.....	231
RtfStyles indexer.....	232
SqlManager.QExport.SQL namespace	233
SqlManager.QExport.SQL namespace Reference	233
QExportSQL component	234
QExportSQL component Reference.....	234
QExportSQL Properties.....	235
CommitAfterScript.....	236
CommitRecCount.....	237
CommitStatement.....	238
CreateTable	239
FormatValues	240
StatementTerm	241
TableName	242
SqlManager.QExport.TXT namespace	243
SqlManager.QExport.TXT namespace Reference	243
QExportTXT component	244
QExportTXT component Reference.....	244
QExportTXT Properties.....	245
CSVComma	246
CSVQuote	247
CSVQuoteStrings.....	248
ExportType	249

TXTSpacing	250
QExportTXT Methods.....	251
SqlManager.QExport.XLS namespace	252
SqlManager.QExport.XLS namespace reference	252
QExportXLS component	253
QExportXLS component Reference.....	253
QExportXLS Properties.....	254
Cells	255
Charts	256
ExportStage	257
FieldFormats	258
FooterRow s	259
HeaderRow s	260
HyperLinks	261
Images	262
MergedCells	263
Notes	264
Options	265
Pictures	266
Sheets	267
StartDataCol	268
StripStyles	269
StripType	270
QExportXLS Methods.....	271
AddBooleanCell.....	272
AddDateTimeCell.....	273
AddMergedCells.....	274
AddNumericCell.....	275
AddStringCell	276
QExportXLS Events.....	277
AdvancedBeforeExportRow	278
AdvancedExportedRecord.....	279
AdvancedGetExportText.....	280
AfterExportSheet.....	281
BeforeExportSheet.....	282
GetAggregateParams	283
GetBeforeDataParams.....	284
OnGetCaptionParams.....	285
GetDataParams	286
GetFooterParams.....	287
GetHeaderParams.....	288
XlsFormat object	289
XlsFormat Properties.....	290
Alignment property.....	291
Borders property.....	292
Fill property	293
Font property	294
Wrap property	295
XlsFormats object	296
XlsFormats Properties.....	297
Items property	298
XlsFieldFormat object	299
XlsFieldFormat - Properties.....	300

Aggregate property	301
Width property	302
XlsFieldFormats object	303
XlsFieldFormats Properties	304
XlsFieldFormats indexer.....	305
XlsFont object	306
TxIsFont Properties.....	307
Charset property.....	308
Color property	309
Name property	310
Script property	311
Size property	312
Bold property	313
Italic property	314
Strikeout property.....	315
Underline property	316
XlsBorder object	317
TxIsBorder Properties.....	318
Color property	319
Style property	320
XlsBorders object	321
XlsBorders - Properties	322
Bottom property.....	323
DiagDown property.....	324
DiagUp property.....	325
Left property	326
Right property	327
Top property	328
XlsFill object	329
XlsFill Properties.....	330
Background property.....	331
Foreground property.....	332
Pattern property.....	333
XlsAlignment object	334
TxIsAlignment Properties.....	335
Horizontal property.....	336
Vertical property.....	337
XlsHyperLink object	338
XlsHyperLink Properties.....	339
Col property	340
Format property.....	341
Row property	342
ScreenTip property.....	343
Style property	344
Target property.....	345
Title property	346
XlsHyperLinks object	347
XlsHyperLinks Properties.....	348
XlsHyperLinks indexer.....	349
XlsNote object	350
XlsNote Properties	351
Col property	352
Format property.....	353
Lines property	354

Row property	355
XlsNotes object	356
XlsNotes Properties	357
XlsNotes indexer	358
XlsChartSeries object	359
XlsChartSeries Properties	360
Color property	361
DataColumn property	362
DataRange property	363
DataRangeType property	364
Title property	365
XlsChartSeriesList object	366
XlsChartSeriesList Properties	367
XlsChartSeriesList indexer	368
XlsChart object	369
XlsChart Properties	370
AutoColor property	371
CategoryLabels property	372
CategoryLabelsType property	373
CategoryLabelsColumn property	374
LegendPlacement property	375
Position property	376
Series property	377
Show Legend property	378
Style property	379
Title property	380
XlsCharts object	381
XlsCharts Properties	382
XlsCharts indexer	383
XlsPicture object	384
XlsPicture Properties	385
Name property	386
FileName property	387
XlsPictures object	388
XlsPictures Properties	389
XlsPictures indexer	390
XlsImage object	391
XlsImage Properties	392
Col property	393
PictureName property	394
Row property	395
Title property	396
Zoom property	397
XlsImages object	398
XlsImages Properties	399
XlsImages indexer	400
XlsCell object	401
XlsCell Properties	402
CellType property	403
Col property	404
DateTimeFormat property	405
Format property	406
IsBoolean property	407
IsDateTime property	408

IsNumeric property.....	409
IsString property.....	410
NumericFormat property.....	411
Row property.....	412
Value property.....	413
XlsCells object.....	414
XlsCells Properties.....	415
XlsCells indexer.....	416
XlsMergedCells object.....	417
XlsMergedCells Properties.....	418
FirstCol property.....	419
FirstRow property.....	420
LastCol property.....	421
LastRow property.....	422
XlsMergedCellList object.....	423
XlsMergedCellList Properties.....	424
XlsMergedCellList indexer.....	425
XlsNoteFormat object.....	426
XlsNoteFormat Properties.....	427
Alignment property.....	428
BackgroundColor property.....	429
FillType property.....	430
Font property.....	431
ForegroundColor property.....	432
Gradient property.....	433
Orientation property.....	434
Transparency property.....	435
XlsDataRange object.....	436
XlsDataRange Properties.....	437
Col1 property.....	438
Col2 property.....	439
Row 1 property.....	440
Row 2 property.....	441
XlsChartPosition object.....	442
XlsChartPosition Properties.....	443
X1 property.....	444
X2 property.....	445
Y1 property.....	446
Y2 property.....	447
XlsSheet object.....	448
XlsSheet Properties.....	449
DefColWidth property.....	450
DefRow Height property.....	451
XlsSheets object.....	452
XlsSheets Properties.....	453
XlsSheets indexer.....	454
XlsOptions object.....	455
XlsOptions Properties.....	456
AggregateFormat property.....	457
CaptionsFormat property.....	458
DataFormat property.....	459
FooterFormat property.....	460
HeaderFormat property.....	461
PageFooter property.....	462

PageHeader property.....	463
SheetTitle property.....	464
HyperLinkFormat property.....	465
NoteFormat property.....	466
XIsColor enum	467
XIsBorderStyle enum	469
XIsPattern enum	470
SqlManager.QExport.XML namespace	471
SqlManager.QExport.XML namespace Reference	471
QExportXML component	472
QExportXML component Reference.....	472
QExportXML Properties.....	473
Options	474
XmIOptions object	475
TXMLOptions Properties.....	476
Encoding property.....	477
StandAlone property.....	478
Version property.....	479

Part



1 Welcome to EMS Advanced Data Export .NET Suite!

1.1 Suite overview

EMS Advanced Data Export .NET represents a set of tools for exporting data from any Command objects, DataTable and ListView descendants to different popular formats, such as: Microsoft Excel, RTF, HTML, PDF, LaTeX, CSV, DIFF, SYLK, Plain text, Windows Clipboard, XML, DBF, SQL and Microsoft Access format. During the export process none of the mechanisms of cooperation between applications in Microsoft Windows environment (DDE, OLE) is used (except Microsoft Access export), which ensures an extremely high speed of work.

From a programmer's point of view the suite represents a homomorphic hierarchy of classes with the common ancestor [QExportBase](#). Besides the basic properties, methods and events, some characteristics, specific for this type of export are included in some descendant classes.

The suite consists of several components:

Component Brief description

- [QExportXLS](#) Export to Microsoft Excel format
- [QExportRTF](#) Export to RTF format, compatible with Microsoft Word
- [QExportHTML](#) Export to HTML format compatible with 4.0 specification
- [QExportPDF](#) Export to PDF format
- [QExportXML](#) Export to XML format
- [QExportSQL](#) Export to SQL script file as INSERT statement
- [QExportDBF](#) Export to DBF Format
- [QExportTXT](#) Export to CSV, DIFF, SYLK, Plain text formats
- [QExportLaTeX](#) Export to LaTeX format versions 2.09 and 2e X
- [QExportClipboard](#) Export to Windows Clipboard
- [QExportAccess](#) Export to MS Access

To learn more, follow the link below

<http://sqlmanager.net/products/tools/advancedexportdotnet>

1.2 What's new

Advanced Data Export .NET

December 25, 2018

What's new in Advanced Data Export .NET 1.8.1?

1. Support for MS Visual Studio 2017 implemented.
2. WordWrap function added for PDF.
3. Data containing leading spaces is exported correctly now.
4. Other minor fixes.

1.3 Product installation

- Download the Advanced Data Export .NET distribution package from the download page at our site;
- Run setup.exe from the local folder and follow the instructions of installation wizard;
- When asked, enter the valid registration key (skip this step if you are using the TRIAL version)
- After installation process is completed, find the Advanced Data Export .NET group in the Windows start menu.

If the installer was unable to add the "EMS Advanced Data Export .NET" tab into MS Visual Studio .NET Toolbox automatically you should install the components manually. To do so, use SqlManager.QExport.dll assembly. To get more information, click [here](#).

1.4 Register EMS Advanced Data Export .NET

All purchases are provided by **PayPro Global** registration service. The **PayPro Global** order process is protected via a secure connection and makes on-line ordering by credit/debit card quick and safe.

PayPro Global is a global e-commerce provider for software and shareware sales via the Internet. It accepts payments in US Dollars, Euros, Pounds Sterling, Japanese Yen, Australian Dollars, Canadian Dollars or Swiss Franks by Credit Card (Visa, MasterCard/EuroCard, American Express, Diners Club), Bank/Wire Transfer.

If you want to review your order information, or you have questions about ordering or payments please visit our [PayPro Global Shopper Support](#), provided by **PayPro Global**.

Please note that all of our products are delivered via ESD (Electronic Software Delivery) only. After purchase you will be able to immediately download the registration keys. Also you will receive a copy of registration keys by email. Please make sure to enter a valid email address in your order. If you have not received the keys within 2 hours, please, contact us at sales@sqlmanager.net.

To obtain more information on this product, visit us at www.sqlmanager.net/en/products/tools/advancedexportdotnet

Product description	Price	PayPro Global
Advanced Data Export .NET Component Suite (with sources)	\$480	Buy Now!
Advanced Data Export .NET Component Suite (without sources)	\$240	Buy Now!
Advanced Data Export .NET Component Suite (trial version)	Free!	Download Now!

1.5 Product licensing information

Copyright (C) 1999-2020 EMS Software Development

END-USER LICENSE AGREEMENT-SINGLE DEVELOPER LICENSE FOR ALL SOFTWARE COMPONENT PRODUCT(S)

IMPORTANT- READ CAREFULLY: This EMS Software Development ("EMS") End-User License Agreement ("EULA") is a legal agreement between you, an individual developer of software applications, ("Developer End User") and EMS for all EMS software components, source code, demos, intermediate files, media, printed materials, and "online" or electronic documentation ("SOFTWARE COMPONENT PRODUCT(S)") contained in this installation file. EMS grants to you as an individual, a personal, nonexclusive license to install and use the SOFTWARE COMPONENT PRODUCT(S) for the sole purposes of designing, developing, testing, and deploying application programs which you create. If you are an entity, you must designate one individual within your organization to license the right to use the SOFTWARE COMPONENT PRODUCT(S) in the manner provided herein.

By installing, copying, or otherwise using the SOFTWARE COMPONENT PRODUCT(S), you agree to be bound by the terms of this EULA. If you do not agree to any part of the terms of this EULA, DO NOT INSTALL, USE, DISTRIBUTE IN ANY MANNER, OR REPLICATE IN ANY MANNER, ANY PART, FILE OR PORTION OF THE SOFTWARE COMPONENT PRODUCT(S). Immediately return it to place of purchase for a full refund.

All SOFTWARE COMPONENT PRODUCT(S) is licensed, not sold.

RIGOROUS ENFORCEMENT OF INTELLECTUAL PROPERTY RIGHTS.

If the licensed right of use for this SOFTWARE COMPONENT PRODUCT(S) is purchased by you with any intent to reverse engineer, decompile, and the exploitation or unauthorized transfer of any EMS intellectual property and trade secrets, to include any exposed methods or source code where provided, no licensed right of use shall exist, and any PRODUCT(s) created as a result shall be judged illegal by definition of all applicable law. Any sale or resale of intellectual property or created derivatives so obtained will be prosecuted to the fullest extent of all local, federal and international law.

1. GRANT OF LICENSE. This EULA, if legally executed as defined herein, licenses and so grants the single individual Developer End User the following rights:

SOFTWARE COMPONENT PRODUCT(S).

You may install and use one copy of the SOFTWARE COMPONENT PRODUCT(S), including any and all source code if provided, or any prior version legally licensed for the same operating system, on a single computer. The primary user of the computer on which the SOFTWARE COMPONENT PRODUCT(S) is installed may make a second copy for his or her exclusive use on a portable computer. You acknowledge and agree that the SOFTWARE COMPONENT PRODUCT(S) in source code form remains a confidential trade secret of EMS.

(a) Storage/Network Use.

You, as Developer End User, may also store or install a copy of the SOFTWARE COMPONENT PRODUCT(S) on a storage device, such as a network server, used only to install or run the SOFTWARE COMPONENT PRODUCT(S) on your other computers over an

internal network; however, Developer End User must acquire and dedicate a license for each separate computer on which the SOFTWARE COMPONENT PRODUCT(S) is installed or run from the storage device. A license for the SOFTWARE COMPONENT PRODUCT(S) may not be shared or used concurrently on different computers.

DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

2. Not for Resale Software.

If the SOFTWARE COMPONENT PRODUCT(S) is labeled and provided as "Not for Resale" or "NFR", then, notwithstanding other sections of this EULA, you may not resell, distribute, or otherwise transfer for value or benefit in any manner, the SOFTWARE COMPONENT PRODUCT(S) or any derivative work using the SOFTWARE COMPONENT PRODUCT(S). You may not transfer, rent, lease, lend, copy, modify, translate, sublicense, time-share or electronically transmit the SOFTWARE COMPONENT PRODUCT(S), media or documentation. This also applies to any and all intermediate files, source code, and compiled executables.

Limitations on Reverse Engineering, Decompilation, and Disassembly.

You may not reverse engineer, decompile, or disassemble the SOFTWARE COMPONENT PRODUCT(S), and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation. The provision of source code, if included with the SOFTWARE COMPONENT PRODUCT(S), does not constitute transfer of any legal rights to such code, and resale or distribution of all or any portion of all source code and intellectual property will be prosecuted to the fullest extent of all applicable local, federal and international laws. You agree to take all reasonable, legal and appropriate measures to prohibit the illegal dissemination of the SOFTWARE COMPONENT PRODUCT(S) or any of its constituent parts and Distributables to the fullest extent of all applicable local, US and International Laws and Treaties regarding anti-circumvention, including but not limited to, the Geneva and Berne World Intellectual Property Organization (WIPO) Diplomatic Conferences.

Separation of Components, their constituent parts and Redistributables.

The SOFTWARE COMPONENT PRODUCT(S) is licensed as a single PRODUCT(s). The SOFTWARE COMPONENT PRODUCT(S) and its constituent parts and any provided Distributables may not be reverse engineered, decompiled, disassembled or separated for use on more than one computer, nor placed for distribution, sale, or resale as individual creations by Developer End User. The provision of source code, if included with the SOFTWARE COMPONENT PRODUCT(S), does not constitute transfer of any legal rights to such code, and resale or distribution of all or any portion of all source code and intellectual property will be prosecuted to the fullest extent of all applicable local, federal and international laws. All EMS libraries, source code, Redistributables and other files remain EMS's exclusive property. Regardless of any modifications that you make, you may not distribute any files (particularly EMS source code and other non-executable files) except those that EMS has expressly designated as Redistributable. EMS PRODUCT(s) may include certain files ("Redistributables") intended for distribution by you to the users of programs you create. Redistributables include, for example, those files identified in printed or on-line documentation or identified by EMS as redistributable files, or those files preselected for deployment by an install utility provided with the SOFTWARE COMPONENT PRODUCT(S) (if any). In any event, the Redistributables for the SOFTWARE COMPONENT PRODUCT(S) are only those files specifically designated as such by EMS. From time to time, EMS may designate other files as Redistributables. You must contact EMS to obtain a list of for additional information on redistributable files. Subject to all of the terms and conditions in this EULA, you may reproduce and distribute exact copies of the Redistributables, provided that such copies are made from the original copy of the

SOFTWARE COMPONENT PRODUCT(S) or the copy transferred to the single hard disk. Copies of Redistributables may only be distributed with and for the sole purpose of executing application programs permitted under this EULA that you have created using the SOFTWARE COMPONENT PRODUCT(S). Under no circumstances may any copies of Redistributables be distributed separately.

3. Rental.

You may not rent, lease, or lend the SOFTWARE COMPONENT PRODUCT(S). SOFTWARE COMPONENT PRODUCT(S) Transfer. You may NOT permanently or temporarily transfer ANY of your rights under this EULA to any individual or entity. Regardless of any modifications which you make and regardless of how you might compile, link, and/or package your programs, under no circumstances may the libraries, code, Redistributables, and/or other files of the SOFTWARE COMPONENT PRODUCT(S) (including any portions thereof) be used for developing programs by anyone other than you. Only you as the licensed Developer End User have the right to use the libraries, code, Redistributables, or other files of the SOFTWARE COMPONENT PRODUCT(S) (or any portions thereof) for developing programs created with the SOFTWARE COMPONENT PRODUCT(S). In particular, you may not share copies of the Redistributables with other co-developers. You may not reproduce or distribute any EMS documentation without EMS's permission.

4. Termination.

Without prejudice to any other rights or remedies, EMS will terminate this EULA upon your failure to comply with all the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE COMPONENT PRODUCT(S) and all of its component parts including any related documentation, and must remove ANY and ALL use of such technology immediately from any applications using technology contained in the SOFTWARE COMPONENT PRODUCT(S) developed by you, whether in native, altered or compiled state.

5. Additional Restrictions.

ONLY THE LEGALLY REGISTERED DEVELOPER END USER IS LICENSED TO DISTRIBUTE ROYALTY-FREE THE EXPRESSNAVBAR, AND ALL ACCOMPANYING VCL CONTROLS AS PART OF AN EXECUTABLE PROGRAM ONLY, subject to all of the conditions of this agreement.

You may use the SOFTWARE COMPONENT PRODUCT(S) and related source code to create new VCL components for your company's internal use ONLY. THE SOURCE CODE, PACKAGES, LIBRARIES AND ALL RESULTING INTERMEDIATE FILES (DCU, OBJ,DLL, ETC.) ARE CONFIDENTIAL AND PROPRIETARY TRADE SECRETS OF EMS.

YOU MAY NOT CREATE NEW 'VCL' COMPONENTS OR 'ACTIVE X' COMPONENTS, or ANY OTHER COMPONENT ARCHITECTURE, INCLUDING BUT NOT LIMITED TO DLLs, FOR DISTRIBUTION OUTSIDE OF YOUR COMPANY IN ANY FORM, MANNER OR MEDIA OR USING ANY DISTRIBUTION CHANNEL, WHICH UTILIZES ALL OR ANY PORTION OF THE SOFTWARE COMPONENT PRODUCT(S) AND ITS RELATED SOURCE CODE.

YOU MAY NOT CREATE ANY TOOL OR SOFTWARE COMPONENT PRODUCT(S) THAT DIRECTLY OR INDIRECTLY COMPETES WITH THE SOFTWARE COMPONENT PRODUCT(S) WHICH UTILIZES ALL OR ANY PORTION OF THE SOFTWARE COMPONENT PRODUCT(S) AND ITS RELATED SOURCE CODE. MODIFICATION OF SOURCE CODE AND SUBSEQUENT REDISTRIBUTION OF ANY SUCH MODIFICATIONS IS PROHIBITED.

6. UPGRADES.

If the SOFTWARE COMPONENT PRODUCT(S) is labeled as an upgrade, you must be

properly licensed to use a PRODUCT(s) identified by EMS as being eligible for the upgrade in order to use the SOFTWARE COMPONENT PRODUCT. A SOFTWARE COMPONENT PRODUCT labeled as an upgrade replaces and/or supplements the SOFTWARE COMPONENT PRODUCT that formed the basis for your eligibility for the upgrade, and together constitute a single PRODUCT. You may use the resulting upgraded PRODUCT only in accordance with all the terms of this EULA.

7. COPYRIGHT.

All title and copyrights in and to the SOFTWARE COMPONENT PRODUCT(S) (including but not limited to any images, demos, source code, intermediate files, packages, photographs, distributables, animations, video, audio, music, text, and "applets" incorporated into the SOFTWARE COMPONENT PRODUCT(S), the accompanying printed materials, and any copies of the SOFTWARE COMPONENT PRODUCT(S) are owned by EMS or its subsidiaries. The SOFTWARE COMPONENT PRODUCT(S) is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE COMPONENT PRODUCT(S) like any other copyrighted material except that you may install the SOFTWARE COMPONENT PRODUCT(S) on a single computer provided you keep the original solely for backup or archival purposes. You may not copy the printed materials accompanying the SOFTWARE COMPONENT PRODUCT(S).

8. Installation and Use.

The license granted in this EULA for you to create your own compiled programs and distribute your programs and the Redistributables (if any), is subject to all of the following conditions:

- (i)** all copies of the programs you create must bear a valid copyright notice, either your own or the EMS copyright notice that appears on the SOFTWARE COMPONENT PRODUCT (S);
- (ii)** you may not remove or alter any EMS copyright, trademark or other proprietary rights notice contained in any portion of EMS libraries, source code, Redistributables or other files that bear such a notice;
- (iii)** EMS provides no warranty at all to any person, other than the Limited Warranty provided to the original purchaser of the SOFTWARE COMPONENT PRODUCT(S), and you will remain solely responsible to anyone receiving your programs for support, service, upgrades, or technical or other assistance, and such recipients will have no right to contact EMS for such services or assistance;
- (iv)** you will indemnify and hold EMS, its related companies and its suppliers, harmless from and against any claims or liabilities arising out of the use, reproduction or distribution of your programs;
- (v)** your programs containing EMS SOFTWARE COMPONENT PRODUCT(S) must be written using a licensed, registered copy of the SOFTWARE COMPONENT PRODUCT(S);
- (vi)** your programs must add primary and substantial functionality, and may not be merely a set or subset of any of the libraries, code, Redistributables or other files of the SOFTWARE COMPONENT PRODUCT(S);
- (vii)** regardless of any modifications which you make and regardless of how you might compile, link, or package your programs, the libraries, code, Redistributables, and/or other files of the SOFTWARE COMPONENT PRODUCT(S) (including any portions thereof) may not be used in programs created by your end users (i.e., users of your programs) and may not be further redistributed by your end users;
- (viii)** you may not use EMS's or any of its suppliers' names, logos, or trademarks to market your programs.

U.S. GOVERNMENT RESTRICTED RIGHTS.

The SOFTWARE COMPONENT PRODUCT(S) is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Products clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer SOFTWARE Products Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is EMS.

EXPORT RESTRICTIONS.

EMS expressly complies with all export restrictions imposed by the government of the United States of America. You, as Developer End User, must agree not to export or re-export the SOFTWARE COMPONENT PRODUCT(S) within any created application to any country, person, entity or end user subject to U.S.A. export restrictions. Restricted countries currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Sudan, and Syria. You warrant and represent that neither the U.S.A. Bureau of Export Administration nor any other federal agency has suspended, revoked or denied your export privileges.

NO WARRANTIES.

EMS expressly disclaims any warranty for the SOFTWARE COMPONENT PRODUCT(S). THE SOFTWARE COMPONENT PRODUCT(S) AND ANY RELATED DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE COMPONENT PRODUCT(S) REMAINS WITH YOU. To the maximum extent permitted by applicable law, in no event shall EMS be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the SOFTWARE COMPONENT PRODUCT(S) or the provision of or failure to provide Support Services, even if EMS has been advised of the possibility of such damages. EMS's entire liability and your exclusive remedy shall be, at EMS's option, either

(a) return of the price paid, or

(b) repair or replacement of the SOFTWARE COMPONENT PRODUCT(S) that does not meet EMS's Limited Warranty and which is returned to EMS with a copy of your receipt. DO NOT RETURN ANY PRODUCT(S) UNTIL YOU HAVE CONTACTED THE EMS CUSTOMER SERVICE DEPARTMENT AND

OBTAINED A RETURN AUTHORIZATION NUMBER. This Limited Warranty is void if failure of the SOFTWARE COMPONENT PRODUCT(S) has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE COMPONENT PRODUCT(S) will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. Outside the United States, neither these remedies nor any PRODUCT(s) support services offered by EMS are available without proof of purchase from an authorized non-U.S. source. Some states and jurisdictions do not allow the exclusion or limitation of liability, so the above limitation may not apply to you.

Support Services.

EMS may provide you with support services related to the SOFTWARE COMPONENT PRODUCT(S) ("Support Services"). Use of Support Services is governed by the EMS policies and programs described in the user manual, in "on line" documentation and/or other EMS provided materials. Any supplemental SOFTWARE COMPONENT PRODUCT(S) code provided to you as part of the Support Services shall be considered part of the SOFTWARE COMPONENT PRODUCT(S) and subject to the terms and conditions of this EULA. With

respect to technical information you provide to EMS as part of the Support Services, EMS may use such information for its business purposes, including for PRODUCT(s) support and development. EMS will not utilize such technical information in a form that personally identifies you.

MISCELLANEOUS.

This EULA shall be construed, interpreted and governed by the laws of the State of Nevada, U.S.A. This EULA gives you specific legal rights; you may have others that vary from state to state and from country to country.

EMS reserves all rights not specifically granted in this EULA.

GENERAL PROVISIONS.

This EULA may only be modified in writing signed by you and an authorized officer of EMS. If any provision of this EULA is found void or unenforceable, the remainder will remain valid and enforceable according to its terms. If any remedy provided is determined to have failed for its essential purpose, all limitations of liability and exclusions of damages set forth in the Limited Warranty shall remain in effect.

If this PRODUCT(s) was acquired outside the United States, then you, as the Developer End User, agree and ascend to the adherence to all applicable international treaties regarding copyright and intellectual property rights which shall also apply. In addition, you, as Developer End User, agree that any local law(s) to the benefit and protection of EMS ownership of, and interest in, its intellectual property and right of recovery for damages thereto will also apply.

[How to Register?](#)

Part



2 EMS Advanced Data Export .NET Component Suite

2.1 SqlManager.QExport.Access namespace

2.1.1 SqlManager.QExport.Access namespace Reference

Components

[QExportAccess](#)

See also:

[QExportAccess](#)

[QExportAccess Properties](#)

2.1.2 QExportAccess component

2.1.2.1 QExportAccess component Reference

Namespace

[SqlManager.QExport.Access](#)

Description

The QExportAccess component is intended for exporting data to MS Access format through an ADO connection. To work with this component you should have ADO installed on your system.

See also:

[QExportAccess Properties](#)

2.1.2.2 QExportAccess Properties

Key properties

[AutoCreateDatabase](#)

[AutoCreateTable](#)

[DatabaseName](#)

[TableName](#)

See also:

[QExportAccess](#)

2.1.2.2.1 AutoCreateDatabase

public new bool AutoCreateDatabase

Description

Setting the AutoCreateDatabase property to True creates the result Access database automatically in case it does not exists.

See also:

[QExportAccess](#)

[QExportAccess Properties](#)

2.1.2.2.2 AutoCreateTable

public new bool AutoCreateTable

Description

Setting the AutoCreateTable property to True creates the result Access table automatically in case it does not exists.

See also:

[QExportAccess](#)

[QExportAccess Properties](#)

2.1.2.2.3 DatabaseName

public new string DatabaseName

Description

The DatabaseName property defines the name of the result Access file to export data to.

See also:

[QExportAccess](#)

[QExportAccess Properties](#)

2.1.2.2.4 TableName

```
public new string TableName
```

Description

The TableName property defines the name of the result Access table to export data to.

See also:

[QExportAccess](#)

[QExportAccess Properties](#)

2.1.2.2.5 Provider

```
public SqlManager.QExport.Access.ProviderType Provider {get; set;}
```

Description

Defines the type of data access OLEDB provider.

See also:

[QExportAccess](#)

[QExportAccess Properties](#)

2.2 SqlManager.QExport.Base namespace

2.2.1 SqlManager.QExport.Base namespace Reference

Components

[QExportBase](#)

Objects

[QExportText](#)

[QExportCol](#)

[QExportFormats](#)

[QExportRow](#)

2.2.2 QExportBase component

2.2.2.1 QExportBase component Reference

Namespace

[SqlManager.QExport.Base](#)

Description

The QExportBase class is the basic class of the collection. In this class the basic properties and events for all the descendant classes are determined, and so are the two basic methods - [Execute](#) and [Abort](#), which are later overridden in each of the components.

See also:

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2 Properties

Key properties

[Version](#)

[Aborted](#)

[About](#)

[AllowCaptions](#)

[AutoCalcColWidth](#)

[AutoCalcStrType](#)

[Captions](#)

[ColumnsAlign](#)

[ColumnsWidth](#)

[Command](#)

[DataTable](#)

[EncodingType](#)

[ExportedFields](#)

[ExportRecCount](#)

[ExportSource](#)

[Footer](#)

[Formats](#)

[Header](#)

[ListView](#)

[NotTruncatableFields](#)

[SkipRecCount](#)

[UserFormats](#)

See also:

[QExportBase](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.1 Version

```
public string Version {get;}
```

Description

The Version property is used in all QExportBase descendant properties and shows the number of the current QuickExport.NET™ version. If you click to change it you will see the "About" window with more detailed information.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.2 Aborted

```
public bool Aborted {get; set;}
```

Description

The Aborted property shows, if method [Abort](#) was called.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.3 About

```
public string About {get;}
```

Description

The About property is used in all QExportBase descendant components and contains information about the current QuickExport.NET™ version and the contact information of its developers. Click the property button to display the "About" window.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.4 Allow Captions

```
protected bool AllowCaptions {get; set;}
```

Description

If AllowCaptions is True, then [Captions](#) are visible in the result file.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.5 AutoCalcColWidth

```
protected bool AutoCalcColWidth {get; set;}
```

Description

The AutoCalcColWidth property is used in [QExportXLS](#), [QExportPDF](#) and [QExportTXT](#) components. If this property is True, then width of each column in the result file is set automatically depending on the maximum number of symbols in the column cells.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.6 AutoCalcStrType

```
public bool AutoCalcStrType {get; set;}
```

Description

The AutoCalcStrType property works only if [ExportSource](#) equals esListView. If this property is True, then after exporting the first record of the source column QuickExport.NET™ tries to determine the data type of the column, and pass this type to the result file. E.g. if the first record value is 1234.78, and AutoCalcStrType = True, then all the column is treated as Float. It makes sense if all the values of the column have the same format, and you export data, e.g. to MS Excel.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.7 Captions

```
public SqlManager.QExport.Collections.StringListCollection Captions {get; set;}
```

Description

The Captions property is used in all QExportBase descendant components, except [QExportDBF](#), [QExportSQL](#) and [QExportXML](#) and determines the column titles in the exported file. The name for each field is set by a separate string which looks as follows

```
<field_name>=<column_title>
```

where <field_name> is the name of one of the source fields, and <column_title> is the text string without such restrictions as quotation marks and apostrophes. The <field_name> is case-insensitive, and the spaces around the mark of equality are not taken into consideration. If the corresponding string for the field is not specified, the the default caption will be used as a column title (e.g. property Column.Caption, if [ExportSource](#) is esDataSet, or Column.Text, if [ExportSource](#) is esListView). The Captions property works only if AllowCaptions is true.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.8 ColumnsAlign

```
protected SqlManager.QExport.Collections.StringListCollection ColumnsAlign {get; set;}
```

Description

Property ColumnsAlign is used in [QExportHTML](#), [QExportRTE](#), [QExportPDF](#) and [QExportTXT](#) (if [ExportType](#) is etTXT), and defines the alignment of the exported cells. Set the alignment in the following format: <Column_Name>=Left|Center|Right, e.g. if COLUMN_1=Center, then column COLUMN_1 will aligned center.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.9 ColumnsWidth

```
protected SqlManager.QExport.Collections.StringListCollection ColumnsWidth {get; set;}
```

Description

Property ColumnsWidth is used in [QExportXLS](#), [QExportRTF](#), [QExportPDF](#) and [QExportTXT](#) (if [ExportType](#) is etTXT), and allows you to set the width of the result table columns in the following way: <field_name>=<value>, where <field_name> is the source field name, and <value> is the corresponding column width in symbols.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.10 Command

```
public System.Data.IDbCommand Command {get; set;}
```

Description

Use this property to determine the exported command, if the ExportSource equals esCommand.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.11 DataTable

```
public System.Data.DataTable DataTable {get; set;}
```

Description

Use DataTable property to determine the exported table, if the [ExportSource](#) property is esDataTable.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.12 EncodingType

protected SqlManager.QExport.Base.EncodingType EncodingType {get; set;}

Description

This property determines the encoding type of the result file.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.13 ExportedFields

```
public SqlManager.QExport.Collections.StringListCollection ExportedFields {get; set;}
```

Description

The ExportedFields property is used in all the QExportBase descendant components and contains the list of the exported fields. If this list is empty (as default), then all the table fields will be exported, except BLOB ones.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.14 ExportRecCount

```
public int ExportRecCount {get; set;}
```

Description

The ExportRecCount property determines the number of records, exported from the source table. If ExportRecCount equals 0, then all the records are exported.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.15 ExportSource

```
public enum QExportSource
{
    // Fields
    esCommand = 0,
    esDataTable = 1,
    esListView = 2
}

public SqlManager.QExport.Base.QExportSource ExportSource {get; set;}
```

Description

The ExportSource property determines the data source for export. The following values are available: esCommand - export from Command, esDataTable - export from DataTable, esListView - export from ListView. Use QExportBase properties to determine the instance of the selected source component: [Command](#), [DataTable](#), or [ListView](#) in accordance.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.16 Footer

```
public SqlManager.QExport.Collections.StringListCollection Footer {get; set;}
```

Description

The Footer property is used in all the descendant components, except [QExportXML](#), [QExportDBF](#) and [QExportAccess](#) and can include a text that will be placed in the export file ([FileName](#)) directly after the exported data.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.17 Formats

```
public SqlManager.QExport.Base.QExportFormats Formats {get; set;}
```

Description

The complex property `Formats` is used in all the descendant components, except [QExportDBF](#), and determines the type of the exported data of the types most frequently used, for which a formatted output might be needed. For the representation of each of these types there exist the string properties `IntegerFormat`, `FloatFormat`, `CurrencyFormat` and `DateTimeFormat`, which have the settings specified in the operation system by default. There are also properties for representing culture name, boolean and Null values.

The settings of these properties will be applied to all source fields of the corresponding types. To set a special format for a separate field (fields), use [UserFormats](#) property.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.18 Header

```
public SqlManager.QExport.Collections.StringListCollection Header {get; set;}
```

Description

The Header property is used in all the descendant components, except [QExportXML](#), [QExportDBF](#) and [QExportAccess](#) and can include a text that will be placed in the export file ([FileName](#)) directly before the exported data.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.19 ListView

```
public System.Windows.Forms.ListView ListView {get; set;}
```

Description

Use ListView property to determine the exported list view, if the [ExportSource](#) property is esListView.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.20 NotTruncatableFields

```
protected SqlManager.QExport.Collections.StringListCollection NotTruncatableFields {get; set;}
```

Description

The NotTruncatableFields property is used in all the QExportBase descendant components and contains the list of the string fields that will not be truncated by occurrences of carriage returns.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.21 SkipRecCount

```
public int SkipRecCount {get; set;}
```

Description

The SkipRecCount property determines the number of records, which are not exported. If SkipRecCount = 0, then all the records are exported.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.22 Title

protected string Title {get; set;}

Description

The Title property determines the title of the exported document.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.2.23 UserFormats

```
public SqlManager.QExport.Collections.StringListCollection  
UserFormats {get; set;}
```

Description

The UserFormats property is used in all QExportBase descendant components, except [QExportDBF](#) and allows to set a special output format for any source field. At the same time, the setting of the strings being a part of the property is set in the following way

<field_name>=<output_format>

where <field_name> is the name of one of the source fields, and <output_format> is the output format for this field. For more details, see the .NET Framework Developer's Guide. The <field_name> is case-insensitive. Note, that there must not be any spaces to the left and to the right of the equality sign.

Use UserFormats property in this and only in this case when you need to set different output formats for two and more fields of the same type, otherwise [Formats](#) property should be used.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.3 Methods

Key methods

[Abort](#)

[Execute](#)

[ExportToStream](#)

[LoadPropertiesFromFile](#)

[SavePropertiesToFile](#)

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Events](#)

2.2.2.3.1 Abort

```
public virtual void Abort()
```

Description

The Abort method stops the current process of export initiated by the [Execute](#) method invocation. At the same time, the [StopExport](#) event takes place.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.3.2 Execute

```
public virtual void Execute()
```

Description

The `Execute` method is the central method of the collection. It executes the export of source data to file. Of course, for each descendant class the method is defined in its own way, but the logics of the work remains the same for all the components. In case of the invocation, the function checks if the [FileName](#) and [ExportSource](#) (and the corresponding [Command](#), [DataTable](#) or [ListView](#)) properties are correct; in case of an error, the exceptions are raised and the method stops executing, returning false as a result. If all the required properties are set correctly, the method starts its work with generating the [BeginExport](#) event. Then the data export to file begins, and after the export of each record the event [ExportedRecord](#) event takes place. On completion of the process, the event [EndExport](#) is invoked. The export can be interrupted by [Abort](#) method invocation, in this case the [StopExport](#) event handler is invoked. At the same time, it is guaranteed that even in case of export interruption the output file will still exist in a readable form for the corresponding software product.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.3.3 ExportToStream

```
public virtual void ExportToStream(System.IO.Stream Stream, System.IO.TextWriter  
Writer)
```

Description

Use `ExportToStream` to export data to the stream, specified by the `Stream` and `Writer` parameters. This can be any `System.IO.Stream` and `System.IO.TextWriter` descendants. This method works in all the `QExportBase` descendants, except [QExportXLS](#).

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.3.4 LoadPropertiesFromFile

```
public void LoadPropertiesFromFile(string FileName)
```

Description

Use this method to restore object properties from file defined by FileName variable, previously saved with the [SavePropertiesToFile](#) method.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.3.5 SavePropertiesToFile

```
public void SavePropertiesToFile(string FileName)
```

Description

Use this method to save all object properties to file defined by the FileName variable. You can restore previously saved properties by calling the [LoadPropertiesFromFile](#) method.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4 Events

Key events

[BeforeExportRow](#)

[BeginExport](#)

[EndExport](#)

[ExportedRecord](#)

[FetchedRecord](#)

[GetCellParams](#)

[GetExportText](#)

[SkippedRecord](#)

[StopExport](#)

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

2.2.2.4.1 BeforeExportRow

```
public class BeforeExportRowEventArgs : EventArgs
{
    // Methods
    public BeforeExportRowEventArgs(QExportRow Row, bool Accept);

    // Properties
    public bool Accept { get; set; }
    public QExportRow Row { get; }

    // Fields
    private bool _accept;
    private QExportRow _row;
}

public delegate void BeforeExportRowEvent(object Sender, BeforeExportRowEventArgs e);

public event SqlManager.QExport.Delegates.BeforeExportRowEvent
BeforeExportRow
```

Description

The BeforeExportRow event allows you to manage data during the export process. The property Row of BeforeExportRowEventArgs object allows you to access data in the current row, and the property Accept allows you to export the row or not. If the Accept is set to True, the row would be exported.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.2 BeginExport

public event System.EventHandler BeginExport

Description

The BeginExport event takes place directly before the export beginning.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.3 EndExport

public event System.EventHandler EndExport

Description

The EndExport event takes place when the export is complete if it wasn't interrupted by the [Abort](#) method. Otherwise the event [StopExport](#) is invoked.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.4 ExportedRecord

```
public delegate void ExportedRecordEvent(object Sender, int RecNo);

public event SqlManager.QExport.Delegates.ExportedRecordEvent
ExportedRecord
```

Description

The ExportedRecord event takes place after the export of each source record. It is most frequently used to transfer to the user the information on the export execution process, such as: display of the number of records exported, the increase of the ProgressBar position, etc.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.5 FetchedRecord

```
public delegate void ExportedRecordEvent(object Sender, int RecNo);
```

protected event SqlManager.QExport.Delegates.ExportedRecordEvent FetchedRecord

Description

Takes place during the calculating of a width for every column when AutoCalcColWidth is True.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.6 GetCellParams

```
public class GetCellParamsEventArgs : EventArgs
{
    // Methods
    public GetCellParamsEventArgs(int RecNo, int ColNo, string Value, QExportColAlign Align, Font
Font, Color Background);

    // Properties
    public QExportColAlign Align { get; set; }
    public Color Background { get; set; }
    public int ColNo { get; }
    public Font Font { get; }
    public int RecNo { get; }
    public string Value { get; }

    // Fields
    private QExportColAlign _align;
    private Color _background;
    private int _colNo;
    private Font _font;
    private int _recNo;
    private string _value;
}

public delegate void GetCellParamsEvent(object Sender, GetCellParamsEventArgs e);

protected event SqlManager.QExport.Delegates.GetCellParamsEvent GetCellParams
```

Description

The GetCellParams event takes place after the parameters of the record (including record value) are received. Depending on the record (GetCellParamsEventArgs.RecNo) and column (GetCellParamsEventArgs.ColNo) number, and on the value (GetCellParamsEventArgs.Value), you can change the format of the result table cell: alignment (GetCellParamsEventArgs.Align), font (GetCellParamsEventArgs.Font), and background (GetCellParamsEventArgs.Background) color.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.7 GetExportText

```
public class GetExportTextEventArgs : EventArgs
{
    // Methods
    public GetExportTextEventArgs(int ColNo, string Text);

    // Properties
    public int ColNo { get; }
    public string Text { get; set; }

    // Fields
    private int _colNo;
    private string _text;
}

public delegate void GetExportTextEvent(object Sender, GetExportTextEventArgs e);

public event SqlManager.QExport.Delegates.GetExportTextEvent GetExportText
```

Description

This event takes place after the export of each source string. Using this event you can replace some definite text of the exported table with another text.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.8 SkippedRecord

```
public delegate void ExportedRecordEvent(object Sender, int RecNo);
```

```
public event SqlManager.QExport.Delegates.ExportedRecordEvent  
SkippedRecord
```

Description

The SkippedRecord event takes place when one of the first source records is skipped. The number of records to skip is determined by the [SkipRecCount](#) property.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.2.4.9 StopExport

```
public class ExportStopEventArgs : EventArgs
{
    // Methods
    public ExportStopEventArgs(bool CanContinue);

    // Properties
    public bool CanContinue { get; set; }

    // Fields
    private bool _canContinue;
}

public delegate void ExportStopEvent(object Sender, ExportStopEventArgs e);

public event SqlManager.QExport.Delegates.ExportStopEvent StopExport
```

Description

The StopExport event takes place when the export process begun by the [Execute](#) method is interrupted with the [Abort](#) method invocation.

See also:

[QExportBase](#)

[QExportBase Properties](#)

[QExportBase Methods](#)

[QExportBase Events](#)

2.2.3 QExportText component

Namespace

[SqlManager.QExport.Base](#)

Description

The QExportText class contains properties, which define the parameters of the result export file: filename, and if file is opened/printed after export.

See also:

[QExportText Properties](#)

2.2.3.1 QExportText Properties

Key properties

[FileName](#)

[PrintFile](#)

[ShowFile](#)

See also:

[QExportText](#)

2.2.3.1.1 FileName property

public string FileName

Description

The FileName property determines the name of the result export file. The property must not be empty; if it is empty in the moment of the [Execute](#) method invocation, the component will stop being executed having generated an exception with the message "Invalid File Name!".

See also:

[QExportText](#)

[QExportText Properties](#)

2.2.3.1.2 PrintFile property

public bool PrintFile

Description

If PrintFile is True then the result file will be sent to printing right after export.

See also:

[QExportText](#)

[QExportText Properties](#)

2.2.3.1.3 Show File property

public bool ShowFile

Description

If ShowFile property is True then the result file will be opened in the appropriate program right after export.

See also:

[QExportText](#)

[QExportText Properties](#)

2.2.4 QExportCol object

Namespace

[SqlManager.QExport.Base](#)

Description

The QExportCol object is an element of the data row, represented by the [QExportRow](#) object. The QExportCol object contains two properties - [Name](#) and [Value](#).

See also:

[QExportCol Properties](#)

2.2.4.1 QExportCol Properties

Key properties

[Name](#)

[Value](#)

See also:

[QExportCol](#)

2.2.4.1.1 Name property

public string Name

Description

The Name property returns column name.

See also:

[QExportCol](#)

[QExportCol Properties](#)

2.2.4.1.2 Value property

public string Value

Description

Use the Value property to change the column value in the current data row.

See also:

[QExportCol](#)

[QExportCol Properties](#)

2.2.5 QExportFormats object

Namespace

[SqlManager.QExport.Base](#)

Description

The complex property Formats is used in all the descendant components and determines the type of the exported data. For representation of each data type there are string properties, such as IntegerFormat, FloatFormat, CurrencyFormat and DateTimeFormat, etc. The default settings of all the properties are those, specified in the operation system.

The settings of these properties will be applied to all the source fields of the corresponding types. To set a special format for a separate field (fields), use [UserFormats](#) property.

See also:

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1 QExportFormats Properties

Key properties

[BooleanFalse](#)

[BooleanTrue](#)

[CurrencyFormat](#)

[DateTimeFormat](#)

[FloatFormat](#)

[IntegerFormat](#)

[NullString](#)

[TimeFormat](#)

See also:

[QExportFormats](#)

[QExportFormats Methods](#)

2.2.5.1.1 BooleanFalse property

public string BooleanFalse

Description

Use property BooleanFalse to set the string value, representing the False values of the source boolean fields in the result table, e.g. "False" or "-".

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1.2 BooleanTrue property

public string BooleanTrue

Description

Use property BooleanTrue to set the string value, representing the True values of the source boolean fields in the result table, e.g. "True" or "+".

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1.3 CurrencyFormat property

public string CurrencyFormat

Description

The CurrencyFormat property determines the representation of currency fields in the result file.

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1.4 DateTimeFormat property

public string DateTimeFormat

Description

The DateTimeFormat property determines the representation of date/time fields in the result file.

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1.5 FloatFormat property

public string FloatFormat

Description

The FloatFormat property determines the representation of float fields in the result file.

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1.6 IntegerFormat property

public string IntegerFormat

Description

The IntegerFormat property determines the representation of integer fields in the result file.

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1.7 NullString property

public string NullString

Description

Use property NullString to set the string value, representing the Null values in the result table, e.g. "Null" or "0".

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.1.8 TimeFormat property

public string TimeFormat

Description

The TimeFormat property determines the representation of the time fields in the result file.

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.5.2 QExportFormats Methods

Key methods

[ResetFormats](#)

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

2.2.5.2.1 ResetFormats method

```
public void ResetFormats(object Owner)
```

Description

Use method ResetFormats to set all the QExportFormats properties to their default values.

See also:

[QExportFormats](#)

[QExportFormats Properties](#)

[QExportFormats Methods](#)

2.2.6 QExportRow object

Namespace

[SqlManager.QExport.Base](#)

Description

The QExportRow represents the current data row as a collection of [QExportCol](#) objects. Use the [this\[\]](#) property to access these objects by their indices. This class is used by the [BeforeExportRow](#) event of the [QExportBase](#) class to manage data during the export process.

See also:

[QExportRow Properties](#)

2.2.6.1 QExportRow Properties

Key properties

[this\[\]](#)

See also:

[QExportRow](#)

2.2.6.1.1 QExportRow .this[]

```
public QExportCol this[int Index]
```

Description

Use this indexer to access the [QExportCol](#) objects by Index.

See also:

[QExportRow](#)

[QExportRow Properties](#)

2.3 SqlManager.QExport.Clipboard namespace

2.3.1 SqlManager.QExport.Clipboard namespace Reference

Components

[QExportClipboard](#)

2.3.2 QExportClipboard component

2.3.2.1 QExportClipboard component Reference

Namespace

[SqlManager.QExport.Clipboard](#)

Description

Use QExportClipboard component to export your data to the Windows clipboard. Two export formats are available in this component. See [ExportType](#) property for details.

See also:

[QExportClipboard Properties](#)

2.3.2.2 QExportClipboard Properties

Key properties

[ClipboardViewer](#)

[ExportType](#)

[ShowContent](#)

[Separator](#)

[Spacing](#)

See also:

[QExportClipboard](#)

2.3.2.2.1 ClipboardViewer

```
public string ClipboardViewer
```

Description

ClipboardViewer property determines the clipboard-viewer application to display your data after export. The default .exe file is clipbrd.exe, located in C:\ You can change it if necessary. This property is used only if property [ShowContent](#) is True.

See also:

[QExportClipboard](#)

[QExportClipboard Properties](#)

2.3.2.2.2 ExportType

```
public enum QClipboardExportType
{
    etSeparated,
    etFixed
}
```

```
public SqlManager.QExport.Clipboard.QClipboardExportType ExportType
```

Description

Use this property to select the way of separating columns in the exported table. If ExportType is etFixed then each column of the exported table will have its own fixed size, defined by the [Spacing](#) property. If ExportType is etSeparated then the columns of the table will be separated by a certain character, defined by the [Separator](#) property.

See also:

[QExportClipboard](#)

[QExportClipboard Properties](#)

2.3.2.2.3 Separator

public string Separator

Description

This property determines the character used for separating the exported table columns if property [ExportType](#) is etSeparated. The default value is the current Windows list separator.

See also:

[QExportClipboard](#)

[QExportClipboard Properties](#)

2.3.2.2.4 Show Content

public bool ShowContent

Description

This property allows to show content of the clipboard buffer with help of an assigned clipboard viewer after the export process finished.

See also:

[QExportClipboard](#)

[QExportClipboard Properties](#)

2.3.2.2.5 Spacing

public int Spacing

Description

This property determines the interval between the exported table columns, if property [ExportType](#) is etFixed. The default interval is 2 space characters.

See also:

[QExportClipboard](#)

[QExportClipboard Properties](#)

2.4 SqlManager.QExport.DBF namespace

2.4.1 SqlManager.QExport.DBF namespace Reference

Components

[QExportDBF](#)

2.4.2 QExportDBF component

2.4.2.1 QExportDBF component Reference

Namespace

[SqlManager.QExport.DBF](#)

Description

The QExportDBF component allows you to export your data to the DBF format.

See also:

[QExportDBF Properties](#)

2.4.2.2 QExportDBF Properties

Key properties

[ColumnsPrecision](#)

[DefaultFloatSize](#)

[DefaultFloatDecimal](#)

[NullValue](#)

See also:

[QExportDBF](#)

2.4.2.2.1 ColumnsPrecision

```
public SqlManager.QExport.Collections.StringListCollection ColumnsPrecision
```

Description

The ColumnsPrecision property determines the column precisions in the exported file. The name for each field is set by a separate string which looks as follows

```
<field_name>=<column_precision>
```

where <field_name> is the name of one of the source fields, and <column_precision> is the column precision. The <field_name> is case-insensitive, and the spaces around the mark of equality are not taken into consideration.

See also:

[QExportDBF](#)

[QExportDBF Properties](#)

2.4.2.2.2 DefaultFloatSize

```
public int DefaultFloatSize
```

Description

The DefaultFloatSize property defines the default size for float fields. This value is used when float fields has no defined size.

See also:

[QExportDBF](#)

[QExportDBF Properties](#)

2.4.2.2.3 DefaultFloatDecimal

```
public int DefaultFloatDecimal
```

Description

The DefaultFloatDecimal property defines the default size for the fractional part of float fields. This value is used when float fields has no defined size of the fractional part.

See also:

[QExportDBF](#)

[QExportDBF Properties](#)

2.4.2.2.4 NullValue

public string NullValue

Description

The NullValue property defines the string value, representing the Null values in the result table.

See also:

[QExportDBF](#)

[QExportDBF Properties](#)

2.5 SqlManager.QExport.HTML namespace

2.5.1 SqlManager.QExport.HTML namespace Reference

Components

[QExportHTML](#)

Objects

[TableOptions](#)

[HtmlOptions](#)

[QExportHtmlNavigation](#)

Enumerations

[HtmlTemplate](#)

[UsingCSS](#)

2.5.2 QExportHTML component

2.5.2.1 QExportHTML component Reference

Namespace

[SqlManager.QExport.HTML](#)

Description

The QExportHTML component is used, as it follows from its name, for exporting data to HTML files. The main peculiarities of the component are:

- complete compatibility with HTML 4.0 specification (see <http://www.w3.org>);
 - compatibility with contemporary browsers (Internet Explorer, Netscape Communicator), starting with version 3;
 - external table styles support (see [CSSFileName](#), [UsingCSS](#)), which gives the opportunity of editing simultaneously all the files that exist as a result of the export;
 - export templates support ([HTMLTemplate](#) property), the opportunity of creating your own templates, of saving them to file and loading them from file;
 - the opportunity of creating several HTML files with a definite number of records in each ([MaxRecords](#)) and the automatic generation of the index file ([GenerateIndex](#));
 - the opportunity of setting the align ([ColumnsAlign](#)) and the header for each exported dataset field;
- ... and much more.

See also:

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2 QExportHTML Properties

Key properties

[BoolAsCheckBox](#)
[CSSFileName](#)
[GenerateIndex](#)
[HTMLOptions](#)
[HTMLTemplate](#)
[InterpretTags](#)
[MaxRecords](#)
[Navigation](#)
[TableOptions](#)
[OverwriteCSSFile](#)
[UsingCSS](#)

See also:

[QExportHTML](#)
[QExportHTML Methods](#)
[QExportHTML Events](#)

2.5.2.2.1 BoolAsCheckBox

```
public bool BoolAsCheckBox {get; set;}
```

Description

If property BoolAsCheckBox is True then all the boolean fields of the table are exported as checkboxes.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.2 CSSFileName

```
public string CSSFileName {get; set;}
```

Description

The CSSFilename property may be set only when the property [UsingCSS](#) is set to usExternal position, and it determines the name of the external style file for the document that is included with the href attribute of the link element, i.e. if you have the setting CSSFileName = "Animals.css" (and you haven't forgotten about UsingCSS = usExternal!), the following string will be placed in the result file

```
<link rel="stylesheet" type="text/css" href="Animals.css">
```

More on the use of style tables see the [UsingCSS](#) property description, and also the corresponding part of HTML specification at <http://www.w3.org/TR/html4/present/styles.html>.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.3 GenerateIndex

```
public bool GenerateIndex {get; set;}
```

Description

The property GenerateIndex is responsible for the automatic creation of an HTML page, containing links to files created during a multifile export ([MaxRecords](#) property value is more than zero).

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.4 HTMLOptions

```
public SqlManager.QExport.HTML.HtmlOptions HtmlOptions {get; set;}
```

Description

The HTMLOptions property sets the general parameters of the result HTML document, and it is complex, i.e. it includes several "subproperties" (in fact, the properties of [HTMLOptions](#) class). Besides setting each property by itself, you can set them all at once in accordance with one of the ten pre-defined templates (see [HTMLTemplate](#) property description).

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.5 HTMLTemplate

```
public enum HtmlTemplate
{
    htCustom,
    htBW,
    htClassic,
    htColorFul,
    htGray,
    htMS_Money,
    htMurky,
    htOlive,
    htPlain,
    htSimple
}
```

```
public SqlManager.QExport.HTML.HtmlTemplate HtmlTemplate {get; set;}
```

Description

The HTMLTemplate property is used for setting the parameters of the result document ([HTMLOptions](#) property) and the exported data table ([TableOptions](#) property) simultaneously.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.6 InterpretTags

```
public bool InterpretTags {get; set;}
```

Description

If property InterpretTags is true then all special symbols <, >," & found in exported data (text) will be replaced with corresponding < > " & ones

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.7 MaxRecords

```
public int MaxRecords {get; set;}
```

Description

The MaxRecord property sets the maximum number of the source records placed into one HTML file. If it has a setting different from zero, then during the [Execute](#) method execution files FileName00.html, FileName01.html, ... , FileNameNN.html will be created, in each of them exactly MaxRecord of the source records will be placed (it is assumed that FileName is a setting of the [FileName](#) property). If MaxRecords = 0 (default), all the records will be exported to one HTML file.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.8 Navigation

```
public SqlManager.QExport.HTML.QExportHtmlNavigation Navigation {get; set;}
```

Description

The Navigation property allows you to define various options of the multi-file export.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.9 TableOptions

```
public SqlManager.QExport.HTML.TableOptions TableOptions {get; set;}
```

Description

Similar to the [HTMLOptions](#) property, the TableOptions property is a "complex" property and it includes several subproperties (the properties of the [TableOptions](#) class). Along with setting each property itself, you can set them all at once in accordance with one of the ten pre-defined templates (see [HTMLTemplate](#) property description).

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.10 OverwriteCSSFile

```
public bool OverwriteCSSFile {get; set;}
```

Description

If OverwriteCSSFile is true and [UsingCSS](#) is true and file, specified in [CSSFileName](#) already exists, then it will be rewritten.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.2.11 UsingCSS

```
public enum UsingCSS
{
    usExternal,
    usInternal
}

public SqlManager.QExport.HTML.UsingCSS UsingCSS {get; set;}
```

Description

The UsingCSS property sets the position of the style table (inside or outside the HTML file). With the usInternal setting the table is placed directly in the HTML file that can look, for example, as follows (Template = htClassic):

```
<Style type="text/css">
BODY background: #333399; color: #FFFFFF; font-family: Arial;
A:link color: #69EF7D
A:visited color: #FF00FF
A:active color: #00FF00
.ThRows background-color: #FF0000; color: #FFFFFF; font-weight: bold; text-align:
center
.TrRows background-color: #007AEC; color: #FFFFFF
.TrOdd background-color: #006BCE; color: #FFFFFF
</Style>
```

If UsingCSS is usExternal, the text above will be placed in the file specified by the [CSSFileName](#) property, and in the HTML file itself there will be only one line

```
<link rel="stylesheet" type="text/css" href="Animals.css">
```

The use of the usExternal setting can be very helpful in case you create several HTML files; then to change, for example, the background color of the even table rows in all the created files it is enough to correct only one setting of the style file (in this example it is Animals.css).

More on the use of style tables in HTML documents see <http://www.w3.org/TR/html4/present/styles.html>.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.3 QExportHTML Methods

Key methods

[Abort](#)

[Execute](#)

[LoadTemplateFromFile](#)

[SaveTemplateToFile](#)

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Events](#)

2.5.2.3.1 LoadTemplateFromFile

```
public void LoadTemplateFromFile(string FileName)
```

Description

Method LoadTemplateFromFile loads from the FileName file the template previously saved by the [SaveTemplateToFile](#) procedure.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.3.2 SaveTemplateToFile

```
public void SaveTemplateToFile(string FileName)
```

Description

The procedure saves the current value of the [TableOptions](#) and [HTMLOptions](#) properties to the file specified by the FileName parameter. The saved template can be loaded later using the [LoadTemplateFromFile](#) procedure.

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

[QExportHTML Events](#)

2.5.2.4 QExportHTML Events

Key events

[GetCellParams](#)

See also:

[QExportHTML](#)

[QExportHTML Properties](#)

[QExportHTML Methods](#)

2.5.3 TableOptions object

Namespace

[SqlManager.QExport.HTML](#)

Description

The TableOptions class contains the properties, which determine the table parameters in the result HTML document.

See also:

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1 TableOptions Properties

Key properties

[AdvancedAttributes](#)

[Border](#)

[BorderColor](#)

[CellPadding](#)

[CellSpacing](#)

[HeadersRowBgColor](#)

[HeadersRowFontColor](#)

[OddRowBgColor](#)

[TableBgColor](#)

[TableFontColor](#)

See also:

[TableOptions](#)

[Color Using Notes](#)

2.5.3.1.1 AdvancedAttributes property

```
public SqlManager.QExport.Collections.StringListCollection AdvancedAttributes
```

Description

Some additional attributes of the Table element. Their list can be found in the HTML format specification at <http://www.w3.org>.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.2 BackgroundFileName

`property BackgroundFileName: string;`

Description

The BackgroundFileName property specifies the file with the background picture for the table.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.3 Border property

public int Border

Description

The Border property corresponds to the Border attribute of the Table element and sets the width of the table border in pixels. The default setting of this property is 1.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.4 BorderColor property

public System.Drawing.Color BorderColor

Description

The BorderColor property determines the color of the output table border. The default color is white. See [Color Using Notes](#) to set the colors correctly.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.5 CellPadding property

```
public int CellPadding
```

Description

The CellPadding property corresponds to the CellPadding attribute of the Table element and sets the space between the cell border and its contents. Together with the CellSpacing attribute it manages the process of displaying table cells.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.6 CellSpacing property

```
public int CellSpacing
```

Description

The CellSpacing property corresponds to the Cellspacing attribute of the Table element. This attribute sets the space (in pixels) that the user's agent (browser) must leave between the left side of the table and the left edge of the first column on the left, the top border of the table and the top edge of the very top line, and the same for the right and the bottom borders of the table. This attribute also sets the space between cells.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.7 HeadersRow BgColor property

public System.Drawing.Color HeadersRowBgColor

Description

The HeadersRowBgColor property sets the background color in the table column headers (the default headers are the source field names; they can be changed by setting the [Captions](#) property). See [Color Using Notes](#) to set the colors correctly.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.8 HeadersRow FontColor property

```
public System.Drawing.Color HeadersRowFontColor
```

Description

The HeadersRowFontColor property sets the font color in the table column headers (the default headers are the source field names; they can be changed by setting the [Captions](#) property). See [Color Using Notes](#) to set the colors correctly.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.9 OddRow BgColor property

```
public System.Drawing.Color OddRowBgColor
```

Description

The OddRowBgColor property sets the background color for the odd table rows. See [Color Using Notes](#) to set the colors correctly.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.10 TableBgColor property

```
public System.Drawing.Color TableBgColor
```

Description

The TableBgColor property sets the background color for the even table rows. See [Color Using Notes](#) to set the colors correctly.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.1.11 TableFontColor property

```
public System.Drawing.Color TableFontColor
```

Description

The TableFontColor property sets the font color for all table rows. See [Color Using Notes](#) to set the colors correctly.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[Color Using Notes](#)

2.5.3.2 Color Using Notes

When setting "dangerous" (not included in the common color schemes) color settings, be careful: it may happen that the user viewing your file won't have such a bright color scheme, which may negatively influence the document design.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[HtmlOptions](#)

2.5.4 HtmlOptions object

Namespace

[SqlManager.QExport.HTML](#)

Description

The HtmlOptions class contains properties that set the parameters of the result HTML document.

See also:

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1 HtmlOptions Properties

Key properties

[AdvancedAttributes](#)

[ALinkColor](#)

[BackgroundColor](#)

[BackgroundFileName](#)

[DefaultOptions](#)

[LinkColor](#)

[TextFont](#)

[VLinkColor](#)

See also:

[HtmlOptions](#)

[Color Using Notes](#)

2.5.4.1.1 AdvancedAttributes property

```
public SqlManager.QExport.Collections.StringListCollection AdvancedAttributes
```

Description

Some additional attributes of the Body element. Their list can be found in the HTML format specification at <http://www.w3.org>.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1.2 ALinkColor property

public System.Drawing.Color ALinkColor

Description

The ALinkColor property corresponds to the ALink attribute of the BODY element and sets the color of the "active" hyperlinks (when the mouse pointer is over the link) of the result document. See [Color Using Notes](#) to set the colors correctly.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1.3 BackgroundColor property

public System.Drawing.Color BackgroundColor

Description

The BackgroundColor property sets the background color of the result HTML document (but not of the table). See [Color Using Notes](#) to set the colors correctly.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1.4 BackgroundFileName property

```
public string BackgroundFileName
```

Description

The BackgroundFileName property sets the background image of the result HTML document. If an empty string or not existing file name is specified as a property value, the document won't have a background image. When setting this property, it is recommended to specify a relative file path (for example, `./Img/html.jpg`), that will let you change the location of your documents without making any changes to the HTML code. It is also recommended to select the background image of the page close to color scheme to the background of the document (see [BackgroundColor](#)), otherwise the users that have the image display mode turned off can get not a very correct image of your design.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1.5 DefaultOptions property

```
[Flags]
public enum DefaultOptions : byte
{
    doFontName = 0x01,
    doFontSize = 0x02
}
```

```
public SqlManager.QExport.HTML.DefaultOptions DefaultOptions
```

Description

This property determines the usage of the default font name and size in the result HTML document. If this property includes doFontName (DefaultOptions.doFontName=True) then the default font name will be used; if it includes doFontSize (DefaultOptions.doFontSize=True) then the default font size will be used. It may also include both of none of them.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1.6 LinkColor property

public System.Drawing.Color LinkColor

Description

The LinkColor property corresponds to the Link attribute of the BODY element and sets the color of the "regular" (not having been visited and not being clicked on at the given moment) hyperlinks of the result document. See [Color Using Notes](#) to set the colors correctly.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1.7 TextFont property

public System.Drawing.Font TextFont

Description

The TextFont property sets the common font of the HTML document. When selecting the font name, don't forget that in case the selected font is missing in the user system, the browser will automatically select the default font, which can affect the design of your document. That is why it is better to use only those fonts that are present at all the platforms popular nowadays.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.1.8 VLinkColor property

public System.Drawing.Color VLinkColor

Description

The VLinkColor property corresponds to the VLink attribute of the BODY element and sets the color of the "visited" hyperlinks of the output document, i.e. of the links that had already been clicked on (Visited Link). See [Color Using Notes](#) to set the colors correctly.

See also:

[HtmlOptions](#)

[HtmlOptions Properties](#)

[Color Using Notes](#)

2.5.4.2 Color Using Notes

When setting "dangerous" (not included in the common color schemes) color settings, be careful: it may happen that the user viewing your file won't have such a bright color scheme, which may negatively influence the document design.

See also:

[TableOptions](#)

[TableOptions Properties](#)

[HtmlOptions](#)

2.5.5 QExportHtmlNavigation object

Namespace

[SqlManager.QExport.HTML](#)

Description

The QExportHtmlNavigation object allows you to customize the multi-file export to HTML. It contains properties for defining the number and the appearance of the navigation links in the result HTML documents.

See also:

[QExportHtmlNavigation Properties](#)

2.5.5.1 QExportHtmlNavigation Properties

Key properties

[FirstLinkTitle](#)

[IndexLinkTemplate](#)

[IndexLinkTitle](#)

[LastLinkTitle](#)

[NextLinkTitle](#)

[OnBottom](#)

[OnTop](#)

[PriorLinkTitle](#)

See also:

[QExportHtmlNavigation](#)

2.5.5.1.1 FirstLinkTitle property

public string FirstLinkTitle

Description

Use FirstLinkTitle to define the caption of the link navigating to the first document of the collection.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.5.1.2 IndexLinkTemplate property

```
public string IndexLinkTemplate
```

Description

The IndexLinkTemplate property defines the template string for generating links on the index page to other pages in the collection.

E.g. if IndexLinkTemplate='part' then the index page will contain the following links: "part1", "part2", "part3", etc.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.5.1.3 IndexLinkTitle property

public string IndexLinkTitle

Description

Use IndexLinkTitle to define the caption of the link navigating to the "home" ("Index") page of the collection.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.5.1.4 LastLinkTitle property

public string LastLinkTitle

Description

Use LastLinkTitle to define the caption of the link navigating to the last document of the collection.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.5.1.5 NextLinkTitle property

public string NextLinkTitle

Description

Use NextLinkTitle to define the caption of the link navigating to the next document in the collection.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.5.1.6 OnBottom property

public bool OnBottom

Description

The OnBottom property defines if there are navigation links (First, Prior, Next, Last) on the bottom of each page in the collection.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.5.1.7 OnTop property

public bool OnTop

Description

The OnTop property defines if there are navigation links (First, Prior, Next, Last) on the top of each page in the collection.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.5.1.8 PriorLinkTitle property

public string PriorLinkTitle

Description

Use PriorLinkTitle to define the caption of the link navigating to the previous document in the collection.

See also:

[QExportHtmlNavigation](#)

[QExportHtmlNavigation
Properties](#)

2.5.6 HtmlTemplate enum

Namespace

[SqlManager.QExport.HTML](#)

```
public enum HtmlTemplate
{
    htCustom,
    htBW,
    htClassic,
    htColorFul,
    htGray,
    htMS_Money,
    htMurky,
    htOlive,
    htPlain,
    htSimple
}
```

Description

Defines the pre-defined templates of the result files (the settings of the [HtmlOptions](#) and [TableOptions](#) properties). Along with the standard templates, you can create your own using the [SaveTemplateToFile](#) and [LoadTemplateFromFile](#) methods.

See also:

[HtmlTemplate property](#)

2.5.7 UsingCSS enum

Namespace

[SqlManager.QExport.HTML](#)

```
public enum UsingCSS
{
    usExternal,
    usInternal
}
```

Description

The variants of positioning the HTML file style table (internally [default] and externally accordingly).

See also:

[UsingCSS property](#)

2.6 SqlManager.QExport.LaTeX namespace

2.6.1 SqlManager.QExport.LaTeX namespace Reference

Components

[QExportLaTeX](#)

[LaTeXOptions](#)

2.6.2 QExportLaTeX component

2.6.2.1 QExportLaTeX component Reference

Namespace

[SqlManager.QExport.LaTeX](#)

Description

The QExportLaTeX component is used for the data export to LaTeX format, which is a popular (especially among mathematicians and physicists) macroextension by of TeX pack developed by D.Knut. For more info on TeX, its various modifications, extensions and Net resources, see TeX Users Group (TUG) at <http://www.tug.org>.

The Russian-speaking users can also visit the site of the Cyrillic TeX Users Group (CyrTUG) at <http://www.cemi.rssi.ru/cyrtug/>.

See also:

[QExportLaTeX Properties](#)

[QExportLaTeX Methods](#)

[LaTeXOptions component](#)

2.6.2.2 QExportLaTeX Properties

Key properties

[Options](#)

[Preamble](#)

See also:

[QExportLaTeX](#)

[QExportLaTeX Methods](#)

[LaTeXOptions component](#)

2.6.2.2.1 Options

public SqlManager.QExport.LaTeX.LaTeXOptions Options

Description

The Options is "complex" and contains a number of subproperties - properties of the class [LaTeXOptions](#).

See also:

[QExportLaTeX](#)

[QExportLaTeX Properties](#)

[QExportLaTeX Methods](#)

[LaTeXOptions component](#)

2.6.2.2.2 Preamble

```
public SqlManager.QExport.Collections.StringListCollection Preamble
```

Description

The Preamble property determines all the additional LaTeX commands that you would like to place before document. The commands already defined by the subproperties of [LaTeXOptions component](#) property shouldn't be included in this property. For more info on LaTeX commands can be found at [TUG](#).

See also:

[QExportLaTeX](#)

[QExportLaTeX Properties](#)

[QExportLaTeX Methods](#)

[LaTeXOptions component](#)

2.6.2.3 QExportLaTeX Methods

Key methods

[Abort](#)

See also:

[QExportLaTeX](#)

[QExportLaTeX Properties](#)

[LaTeXOptions component](#)

2.6.3 LaTeXOptions component

Namespace

[SqlManager.QExport.LaTeX](#)

Description

The LaTeXOptions class contains the properties which determine the parameters of the result document.

See also:

[LaTeXOptions Properties](#)

[QExportLaTeX](#)

2.6.3.1 TLaTeXOptions - Properties

Key properties

[CodePage](#)

[DocumentParams](#)

[DocumentStyle](#)

[Languages](#)

[LaTeXVersion](#)

See also:

[LaTeXOptions](#)

[QExportLaTeX](#)

2.6.3.1.1 CodePage property

```
public int CodePage
```

Description

The CodePage property determines in which encoding LaTeX will perceive the exported file. The default setting of this property is equal to the active code page.

See also:

[LaTeXOptions](#)

[LaTeXOptions Properties](#)

[QExportLaTeX](#)

2.6.3.1.2 DocumentParams property

```
public string DocumentParams
```

Description

The DocumentParams property sets the unrequired parameters of the output document, which are placed in the (2e) or (2.09) commands. The default setting of this property is a4paper, which shows to LaTeX that the document will be printed on paper of A4 format. For more info see [TUG](#).

See also:

[LaTeXOptions](#)

[LaTeXOptions Properties](#)

[QExportLaTeX](#)

2.6.3.1.3 DocumentStyle property

```
public enum LaTeXDocStyle
{
    dsArticle,
    dsBook
}
```

```
public SqlManager.QExport.LaTeXLaTeXDocStyle DocumentStyle
```

Description

The DocumentStyle property sets the style of the LaTeX document. The following values are available: dsArticle - article style (default) and dsBook - book style. The corresponding string with the same name is placed in the (2e) or (2.09) command.

See also:

[LaTeXOptions](#)

[LaTeXOptions Properties](#)

[QExportLaTeX](#)

2.6.3.1.4 Languages property

public string Languages

Description

The Languages property determines the languages which will be used in the result file. The default is English. To add other languages just type their English names through a comma, for example, for the Russian language support the property should have the setting "english,russian" (without quotes).

See also:

[LaTeXOptions](#)

[LaTeXOptions Properties](#)

[QExportLaTeX](#)

2.6.3.1.5 LaTeXVersion property

```
public enum LaTeXVersion
{
    LaTeX209,
    LaTeX2e
}
```

```
public SqlManager.QExport.LaTeX.LaTeXVersion LaTeXVersion
```

Description

The LaTeXVersion determines the version of the output file. The following values are available: LaTeX209 and LaTeX2e (default).

See also:

[LaTeXOptions](#)

[LaTeXOptions Properties](#)

[QExportLaTeX](#)

2.7 SqlManager.QExport.PDF namespace

2.7.1 SqlManager.QExport.PDF namespace Reference

Components

[QExportPDF](#)

Objects

[PdfFont](#)

[PdfOptions](#)

2.7.2 QExportPDF component

2.7.2.1 QExportPDF component Reference

Namespace

[SqlManager.QExport.PDF](#)

Description

The QExportPDF component is intended for exporting data to PDF format.

See also:

[QExportPDF Properties](#)

2.7.2.2 QExportPDF Properties

Key properties

[Options](#)

See also:

[QExportPDF](#)

2.7.2.2.1 Options

public SqlManager.QExport.PDF.PdfOptions Options

Description

Use Options to define customize the result PDF file - tune its fonts, grid, etc.

See also:

[QExportPDF](#)

[QExportPDF Properties](#)

2.7.3 PdfFont object

Namespace

[SqlManager.QExport.PDF](#)

Description

The PdfFont object contains properties for defining various fonts for use in the result PDF file.

See also:

[PdfFont Properties](#)

[QExportPDF](#)

2.7.3.1 PdfFont Properties

Key properties

[BaseFont](#)

[FontColor](#)

[FontEncoding](#)

[FontSize](#)

See also:

[PdfFont](#)

[QExportPDF](#)

2.7.3.1.1 BaseFont property

```
public enum PdfFontName
{
    poHelvetica,
    poHelveticaBold,
    poHelveticaOblique,
    poHelveticaBoldOblique,
    poCourier,
    poCourierBold,
    poCourierOblique,
    poCourierBoldOblique,
    poTimesRoman,
    poTimesBold,
    poTimesItalic,
    poTimesBoldItalic,
    poSymbol,
    poZapfDingbats
}
```

```
public SqlManager.QExport.PDF.PdfFontName BaseFont
```

Description

The BaseFont property defines the name of the font in the result PDF document.

See also:

[PdfFont](#)

[PdfFont Properties](#)

[QExportPDF](#)

2.7.3.1.2 FontColor property

public System.Drawing.Color FontColor

Description

The FontColor property defines the color of the customized PDF font.

See also:

[PdfFont](#)

[PdfFont Properties](#)

[QExportPDF](#)

2.7.3.1.3 FontEncoding property

```
public enum PdfFontEncoding
{
    poStandardEncoding,
    poWinAnsiEncoding,
    poMacRomanEncoding,
    poPDFDocEncoding
}
```

```
public SqlManager.QExport.PDF.PdfFontEncoding FontEncoding
```

Description

The FontEncoding property is used for setting the character encoding in the result PDF document.

See also:

[PdfFont](#)

[PdfFont Properties](#)

[QExportPDF](#)

2.7.3.1.4 FontSize property

```
public int FontSize
```

Description

The FontSize property defines the size of the customized PDF font.

See also:

[PdfFont](#)

[PdfFont Properties](#)

[QExportPDF](#)

2.7.4 PdfOptions object

Namespace

[SqlManager.QExport.PDF](#)

Description

The PdfOptions class contains properties for defining the parameters of the result PDF document.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1 PdfOptions Properties

Key properties

[CaptionFont](#)

[ColSpacing](#)

[DataFont](#)

[FooterFont](#)

[GridLineColor](#)

[GridLineWidth](#)

[HeaderFont](#)

[RowSpacing](#)

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.1 CaptionFont property

```
public SqlManager.QExport.PDF.PdfFont CaptionFont
```

Description

Use CaptionFont to define the properties of the font for the result PDF captions.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.2 ColSpacing property

public double ColSpacing

Description

The ColSpacing property defines the column spacing in the result PDF table.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.3 DataFont property

public SqlManager.QExport.PDF.PdfFont DataFont

Description

Use DataFont to define the properties of the font for displaying data in the result PDF table.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.4 FooterFont property

```
public SqlManager.QExport.PDF.PdfFont FooterFont
```

Description

Use FooterFont to define the properties of the font for the result PDF footers.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.5 GridLineColor property

public System.Drawing.Color GridLineColor

Description

The GridLineColor property defines the color of the grid lines in the result PDF table.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.6 GridLineWidth property

```
public int GridLineWidth
```

Description

The GridLineWidth property defines the width of the grid lines in the result PDF table.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.7 HeaderFont property

```
public SqlManager.QExport.PDF.PdfFont HeaderFont
```

Description

Use HeaderFont to define the properties of the font for the result PDF headers.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.7.4.1.8 Row Spacing property

public double RowSpacing

Description

The RowSpacing property defines the row spacing in the result PDF table.

See also:

[PdfOptions](#)

[PdfOptions Properties](#)

[QExportPDF](#)

2.8 SqlManager.QExport.RTF namespace

2.8.1 SqlManager.QExport.RTF namespace Reference

Components

[QExportRTF](#)

Objects

[RtfOptions](#)

[RtfStyle](#)

[RtfStyles](#)

2.8.2 QExportRTF component

2.8.2.1 QExportRTF component Reference

Namespace

[SqlManager.QExport.RTF](#)

Description

The QExportRTF component is used for data export to RTF (Rich Text Format) format supported by many text processing programs (e.g. Microsoft Word).

See also:

[QExportRTF Properties](#)

[QExportRTF Methods](#)

[QExportRTF Events](#)

2.8.2.2 QExportRTF Properties

Key properties

[Options](#)

See also:

[QExportRTF](#)

[QExportRTF Methods](#)

[QExportRTF Events](#)

2.8.2.2.1 Options

```
public SqlManager.QExport.RTF.RtfOptions Options
```

Description

The Options property is "complex" and contains some subproperties -- properties of the [RtfOptions](#) class.

See also:

[QExportRTF](#)

[QExportRTF Properties](#)

[QExportRTF Methods](#)

[QExportRTF Events](#)

[RtfOptions](#)

2.8.2.3 QExportRTF Methods

Key methods

[Abort](#)

See also:

[QExportRTF](#)

[QExportRTF Properties](#)

[QExportRTF Events](#)

2.8.2.4 QExportRTF Events

Key events

[GetCaptionStyle](#)

[GetDataStyle](#)

[GetHeaderStyle](#)

[GetFooterStyle](#)

See also:

[QExportRTF](#)

[QExportRTF Properties](#)

[QExportRTF Methods](#)

2.8.2.4.1 GetCaptionStyle

```
public class RtfGetCaptionStyleEventArgs : EventArgs
{
    // Methods
    public RtfGetCaptionStyleEventArgs(int ColNo, RtfStyle Style);

    // Properties
    public int ColNo { get; }
    public RtfStyle Style { get; }

    // Fields
    private int _colNo;
    private RtfStyle _style;
}

public delegate void RtfGetCaptionStyleEvent(object Sender, RtfGetCaptionStyleEventArgs e);

public event SqlManager.QExport.Delegates.RtfGetCaptionStyleEvent GetCaptionStyle
```

Description

The GetCaptionStyle event takes place when the style of the caption is received. Depending on the column number (RtfGetCaptionStyleEventArgs.ColNo) you can edit the caption appearance (RtfGetCaptionStyleEventArgs.Style) style.

See also:

[QExportRTF](#)

[QExportRTF Properties](#)

[QExportRTF Methods](#)

[QExportRTF Events](#)

2.8.2.4.2 GetDataStyle

```
public class RtfGetDataStyleEventArgs : EventArgs
{
    // Methods
    public RtfGetDataStyleEventArgs(int Row, int Col, RtfStyle Style);

    // Properties
    public int Col { get; }
    public int Row { get; }
    public RtfStyle Style { get; }

    // Fields
    private int _col;
    private int _row;
    private RtfStyle _style;
}

public delegate void RtfGetDataStyleEvent(object Sender, RtfGetDataStyleEventArgs e);

public event SqlManager.QExport.Delegates.RtfGetDataStyleEvent GetDataStyle
```

Description

The GetDataStyle event takes place when the style of data is received. Depending on the column number (RtfGetDataStyleEventArgs.Col) you can edit the data appearance style (RtfGetDataStyleEventArgs.Style).

See also:

[QExportRTF](#)

[QExportRTF Properties](#)

[QExportRTF Methods](#)

[QExportRTF Events](#)

2.8.2.4.3 GetFooterStyle

```
public delegate void RtfGetStyleEvent(object Sender, RtfStyle Style);
```

```
public event SqlManager.QExport.Delegates.RtfGetStyleEvent GetFooterStyle
```

Description

The GetFooterStyle event takes place when the style of footer is received. Type code here to change the document's footer appearance style.

See also:

[QExportRTF](#)

[QExportRTF Properties](#)

[QExportRTF Methods](#)

[QExportRTF Events](#)

2.8.2.4.4 GetHeaderStyle

```
public delegate void RtfGetStyleEvent(object Sender, RtfStyle Style);
```

```
public event SqlManager.QExport.Delegates.RtfGetStyleEvent GetHeaderStyle
```

Description

The GetHeaderStyle event takes place when the style of header is received. Type code here to change the document's header appearance style.

See also:

[QExportRTF](#)

[QExportRTF Properties](#)

[QExportRTF Methods](#)

[QExportRTF Events](#)

2.8.3 RtfOptions object

Namespace

[SqlManager.QExport.RTF](#)

Description

The RtfOptions class contains the properties which determine the parameters of the result RTF document.

See also:

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1 RtfOptions Properties

Key properties

[PageOrientation](#)

[CaptionAligns](#)

[CaptionStyle](#)

[DataStyle](#)

[FooterStyle](#)

[HeaderStyle](#)

[StripStyles](#)

[StripType](#)

See also:

[RtfOptions](#)

[RtfStyles](#)

2.8.3.1.1 PageOrientation property

```
public enum QExportPageOrientation
{
    poPortrait,
    poLandscape
}
```

```
public SqlManager.QExport.RTF.QExportPageOrientation PageOrientation
```

Description

The PageOrientation defines the orientation of the result document. The following values are available: poPortrait - vertical page orientation, and poLandscape - horizontal page orientation.

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1.2 CaptionAligns property

```
public SqlManager.QExport.Collections.StringListCollection CaptionAligns
```

Description

The CaptionAligns property defines the captions' alignments for the columns. The alignments are stored as strings in "<Name>=<Alignment>" format.

Example:

```
Column1=Left  
Column2=Center  
Column3=Right
```

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1.3 CaptionStyle property

```
public SqlManager.QExport.RTF.RtfStyle CaptionStyle
```

Description

The CaptionStyle property contains parameters, which define the captions appearance in the result document, such as colors and font.

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1.4 DataStyle property

```
public SqlManager.QExport.RTF.RtfStyle DataStyle
```

Description

The DataStyle property contains parameters, which define the data appearance in the result document, such as colors and font.

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1.5 FooterStyle property

```
public SqlManager.QExport.RTF.RtfStyle FooterStyle
```

Description

Use this property to define the document's footer style.

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1.6 HeaderStyle property

```
public SqlManager.QExport.RTF.RtfStyle HeaderStyle
```

Description

Use this property to define the document's header style.

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1.7 StripStyles property

```
public SqlManager.QExport.RTF.RtfStyles StripStyles
```

Description

Use this property to define the strips styles. The StripStyles property contains a collection of styles, which would be used one by one.

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.3.1.8 StripType property

```
public enum RtfStripType
{
    stNone,
    stCol,
    stRow
}
```

```
public SqlManager.QExport.RTF.RtfStripType StripType
```

Description

This property defines strips direction - horizontal or vertical. If StripType has the stCol value, the strips are vertical. If the stRow value, the strips are horizontal. If the stNone value is set, the strips are disabled.

See also:

[RtfOptions](#)

[RtfOptions Properties](#)

[RtfStyles](#)

2.8.4 RtfStyle object

Namespace

[SqlManager.QExport.RTF](#)

Description

The RtfStyle class contains parameters for defining the text appearance, which can be applied to some of the result document parts.

See also:

[RtfStyle Properties](#)

[RtfStyles](#)

2.8.4.1 RtfStyle - Properties

Key properties

[Alignment](#)

[AllowBackground](#)

[AllowHighlight](#)

[BackgroundColor](#)

[Font](#)

[HighlightColor](#)

See also:

[RtfStyle](#)

[RtfStyles](#)

2.8.4.1.1 Alignment property

```
public enum RtfTextAlignment
{
    talLeft,
    talRight,
    talCenter,
    talFill
}
```

```
public SqlManager.QExport.RTF.RtfTextAlignment Alignment
```

Description

The Alignment property defines the text alignment for the current style. The following values are available: palLeft - text is left-justified, palRight - text is right-justified, palCenter - text is centered, and palFill - text is distributed over the paragraph width.

See also:

[RtfStyle](#)

[RtfStyle Properties](#)

[RtfStyles](#)

2.8.4.1.2 Allow Background property

public bool AllowBackground

Description

The AllowBackground option enables using the background color in the current style. The default value is True.

See also:

[RtfStyle](#)

[RtfStyle Properties](#)

[RtfStyles](#)

2.8.4.1.3 Allow Highlight property

public bool AllowHighlight

Description

The AllowHighlight option enables highlighting text in the current style. The default value is False.

See also:

[RtfStyle](#)

[RtfStyle Properties](#)

[RtfStyles](#)

2.8.4.1.4 BackgroundColor property

public System.Drawing.Color BackgroundColor

Description

Use the BackgroundColor property to define the background color in the current style. This color is applied only if the [AllowBackground](#) property is True.

See also:

[RtfStyle](#)

[RtfStyle Properties](#)

[RtfStyles](#)

2.8.4.1.5 Font property

```
public System.Drawing.Font Font
```

Description

Use the Font property to define text font parameters for the current style.

See also:

[RtfStyle](#)

[RtfStyle Properties](#)

[RtfStyles](#)

2.8.4.1.6 HighlightColor property

public System.Drawing.Color HighlightColor

Description

Use the HighlightColor property to define the color of the text highlighting in the current style. This color is applied only if the [AllowHighlight](#) property is True.

See also:

[RtfStyle](#)

[RtfStyle Properties](#)

[RtfStyles](#)

2.8.5 RtfStyles object

Namespace

[SqlManager.QExport.RTF](#)

Description

The RtfStyles is a collection of the objects.

See also:

[RtfStyles Properties](#)

[RtfStyle](#)

2.8.5.1 RtfStyles - Properties

Key properties

[this\[\]](#)

See also:

[RtfStyles](#)

[RtfStyle](#)

2.8.5.1.1 RtfStyles indexer

```
public RtfStripStyle this[int Index]
```

Description

Use this indexer to access the [TrtfStyle](#) objects by Index.

See also:

[RtfStyles](#)

[RtfStyles Properties](#)

[RtfStyle](#)

2.9 SqlManager.QExport.SQL namespace

2.9.1 SqlManager.QExport.SQL namespace Reference

Components

[QExportSQL](#)

2.9.2 QExportSQL component

2.9.2.1 QExportSQL component Reference

Namespace

[SqlManager.QExport.SQL](#)

Description

The QExportSQL component allows you to export your data to the SQL Script file as SQL statement INSERT.

See also:

[QExportSQL Properties](#)

2.9.2.2 QExportSQL Properties

Key properties

[CommitAfterScript](#)

[CommitRecCount](#)

[CommitStatement](#)

[CreateTable](#)

[FormatValues](#)

[StatementTerm](#)

[TableName](#)

See also:

[QExportSQL](#)

2.9.2.2.1 CommitAfterScript

public bool CommitAfterScript

Description

This property allows you to insert the commit statement to the bottom line of the output script. You can define this statement, using the [CommitStatement](#) property.

See also:

[QExportSQL](#)

[QExportSQL Properties](#)

2.9.2.2.2 CommitRecCount

```
public int CommitRecCount
```

Description

This property allows you to define a number of records, after which the commit statement is inserted to the output script. The default number is 0, i.e. no commits are inserted. You can define the commit statement using the [CommitStatement](#).

See also:

[QExportSQL](#)

[QExportSQL Properties](#)

2.9.2.2.3 CommitStatement

public string CommitStatement

Description

This property determines the commit statement to insert to the result script after a definite number of records (see [CommitRecCount](#) property) or at the end of the script (see [CommitAfterScript](#) property).

See also:

[QExportSQL](#)

[QExportSQL Properties](#)

2.9.2.2.4 CreateTable

public bool CreateTable

Description

If this property is true then the CREATE TABLE statement is inserted to the output script. You can define the table name, using the [TableName](#) property.

See also:

[QExportSQL](#)

[QExportSQL_Properties](#)

2.9.2.2.5 FormatValues

property FormatValues: boolean;

Description

This property permits or forbids formatting of exported values according to the content of [Formats](#) property.

See also:

[QExportSQL](#)

[QExportSQL Properties](#)

2.9.2.2.6 StatementTerm

public char StatementTerm

Description

This property determines the character to denote the end of each statement. The default character is semicolon - ";".

See also:

[QExportSQL](#)

[QExportSQL Properties](#)

2.9.2.2.7 TableName

public string TableName

Description

This property allows you to set the name of the table to use in the INSERT statement and CREATE TABLE statement, if the [CreateTable](#) property is True.

See also:

[QExportSQL](#)

[QExportSQL Properties](#)

2.10 SqlManager.QExport.TXT namespace

2.10.1 SqlManager.QExport.TXT namespace Reference

Components

[QExportTXT](#)

2.10.2 QExportTXT component

2.10.2.1 QExportTXT component Reference

Namespace

[SqlManager.QExport.TXT](#)

Description

The QExportTXT component is used for the data export to formats that are usually used as working or interchange formats, i.e. Comma Separated Value (CSV), Data Interchange File Format (DIFF), Symbolic Links (SYLK) and Plain Text Format.

The output file format is set by the [ExportType](#) property.

See also:

[QExportTXT Properties](#)

[QExportTXT Methods](#)

2.10.2.2 QExportTXT Properties

Key properties

[CSVComma](#)

[CSVQuote](#)

[CSVQuoteStrings](#)

[ExportType](#)

[TXTSpacing](#)

See also:

[QExportTXT](#)

[QExportTXT Methods](#)

2.10.2.2.1 CSVComma

```
public string CSVComma {get; set;}
```

Description

The CSVComma property is used only during the export to .csv format ([ExportType](#) is etCSV) and is created to install the symbol-delimiter in the result file. The default property value is the current Windows list separator.

See also:

[QExportTXT](#)

[QExportTXT Properties](#)

[QExportTXT Methods](#)

2.10.2.2.2 CSVQuote

```
public char CSVQuote {get; set;}
```

Description

The CSVQuote property is used only during the export to .csv format ([ExportType](#) is etCSV) and is created to install the quotation symbol in the result file. This symbol is used only if the [CSVQuoteStrings](#) property is True. The default property value is "".

See also:

[QExportTXT](#)

[QExportTXT Properties](#)

[QExportTXT Methods](#)

2.10.2.2.3 CSVQuoteStrings

```
public bool CSVQuoteStrings {get; set;}
```

Description

The CSVQuoteStrings property is used only during the export to .csv format ([ExportType](#) is etCSV). If this property is True, then all the source strings will be exported as quotations.

For example, string

```
QuickExport.NET is a nice set of components, isn't it?
```

is exported as

```
"QuickExport.NET is a nice set of components, isn't it?"
```

See also:

[QExportTXT](#)

[QExportTXT Properties](#)

[QExportTXT Methods](#)

2.10.2.2.4 ExportType

```
public enum QExportTextType
{
    // Fields
    etCSV = 1,
    etDIF = 2,
    etSYLK = 3,
    etTXT = 0
}
```

```
public SqlManager.QExport.TXT.QExportTextType ExportType {get; set;}
```

Description

The ExportType property sets the format of the output export file. The relations between the property setting and the file type are as follows:

Property setting	File type
etCSV	Comma Separated Values, .csv
etDIF	Data Interchange File Format, .dif
etSYLK	Symbolic Links Format, .slk
etTxt	Plain Text Format, .txt

See also:

[QExportTXT](#)

[QExportTXT Properties](#)

[QExportTXT Methods](#)

2.10.2.2.5 TXTSpacing

```
public int TXTSpacing {get; set;}
```

Description

The TXTSpacing property sets the distance (in symbols) between the data columns in the result file. The default value is 1.

See also:

[QExportTXT](#)

[QExportTXT Properties](#)

[QExportTXT Methods](#)

2.10.2.3 QExportTXT Methods

Key methods

[Abort](#)

See also:

[QExportTXT](#)

[QExportTXT Properties](#)

2.11 SqlManager.QExport.XLS namespace

2.11.1 SqlManager.QExport.XLS namespace reference

Components

[QExportXLS](#)

Objects

[XlsFormat](#)

[XlsFormats](#)

[XlsFieldFormat](#)

[XlsFieldFormats](#)

[XlsFont](#)

[XlsBorder](#)

[XlsBorders](#)

[XlsFill](#)

[XlsAlignment](#)

[XlsHyperLink](#)

[XlsHyperLinks](#)

[XlsNote](#)

[XlsNotes](#)

[XlsChartSeries](#)

[XlsChartSeriesList](#)

[XlsChart](#)

[XlsCharts](#)

[XlsPicture](#)

[XlsPictures](#)

[XlsImage](#)

[XlsImages](#)

[XlsCell](#)

[XlsCells](#)

[XlsMergedCells](#)

[XlsMergedCellList](#)

[XlsNoteFormat](#)

[XlsDataRange](#)

[XlsChartPosition](#)

[XlsOptions](#)

Enumerations

[XlsColor](#)

[XlsBorderStyle](#)

[XlsPattern](#)

2.11.2 QExportXLS component

2.11.2.1 QExportXLS component Reference

Namespace

[SqlManager.QExport.XLS](#)

Description

The QExportXLS component is used to export data to the most popular e-table format - Microsoft Excel. The result files are fully compatible with the versions 95, 97, 2000 and XP.

See also:

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2 QExportXLS Properties

Key properties

[Cells](#)

[Charts](#)

[ExportStage](#)

[FieldFormats](#)

[FooterRows](#)

[HeaderRows](#)

[HyperLinks](#)

[Images](#)

[MergedCells](#)

[Notes](#)

[Options](#)

[Pictures](#)

[Sheets](#)

[StartDataCol](#)

[StripStyles](#)

[StripType](#)

See also:

[QExportXLS](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.1 Cells

```
public SqlManager.QExport.XLS.XlsCells Cells {get; set;}
```

Description

The Cells property allows you to insert values to the definite places in the result Excel document. [XlsCells](#) is a collection of [XlsCell](#) objects, which define the parameters of each single cell.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.2 Charts

```
public SqlManager.QExport.XLS.XlsCharts Charts
```

Description

The Charts property allows you to insert charts based on data from the [ExportSource](#) to the result Excel document. [XlsCharts](#) is a collection of [XlsChart](#) objects, which define the parameters of each single chart.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.3 ExportStage

```
public enum XlsExportStage
{
    // Fields
    esAggregate = 4,
    esCaption = 2,
    esData = 3,
    esFooter = 5,
    esHeader = 1,
    esNone = 0
}

public SqlManager.QExport.XLS.XlsExportStage ExportStage {get; set;}
```

Description

The ExportStage property indicates the stage of the export process. The following values are available: esNone - no export, esHeader - header exported, esCaption - captions exported, esData - data exported, esAggregate - aggregate functions exported, esFooter - footer exported.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.4 FieldFormats

```
public SqlManager.QExport.XLS.XlsFieldFormats FieldFormats {get; set;}
```

Description

The FieldFormats property allows you to set the formats for each source field separately.

Note: StripStyles property has higher priority than the FieldFormats property. If the StripStyles property is defined and the StripType property is set to ssCol or ssRow, all the field styles defined in FieldFormats property takes no effect.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.5 FooterRow s

```
public int FooterRows {get; set;}
```

Description

The FooterRows property defines the number of rows in the [Footer](#) property to include in the result file. If the FooterRows value is 0 then all the rows from the [Footer](#) property will be included in the result file.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.6 HeaderRows

```
public int HeaderRows {get; set;}
```

Description

The HeaderRows property defines the number of rows in the [Header](#) property to include in the result file. If the HeaderRows value is 0 then all the rows from the [Header](#) property will be included in the result file.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.7 HyperLinks

```
public SqlManager.QExport.XLS.XlsHyperlinks Hyperlinks {get; set;}
```

Description

The HyperLinks property allows you to insert hyperlinks to the result Excel document. [XlsHyperLinks](#) is a collection of [XlsHyperLink](#) objects, which define the parameters of each single hyperlink.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.8 Images

```
public SqlManager.QExport.XLS.XlsImages Images {get; set;}
```

Description

The Images property allows you to insert images in the result Excel document. [XlsImages](#) is a collection of [XlsImage](#) objects, which define the parameters of each single image.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.9 MergedCells

```
public SqlManager.QExport.XLS.XlsMergedCellList MergedCells {get; set;}
```

Description

The MergedCells property allows you to merge a few cells in the result Excel document in one cell. [XlsMergedCellList](#) is a collection of [XlsMergedCells](#) objects, which define the parameters of merging.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.10 Notes

```
public SqlManager.QExport.XLS.XlsNotes Notes {get; set;}
```

Description

The Notes property allows you to insert notes to the result Excel document. [XlsNotes](#) is a collection of [XlsNote](#) objects, which define the parameters of each single note.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.11 Options

```
public SqlManager.QExport.XLS.XlsOptions Options {get; set;}
```

Description

The Options property is "complex" and contains some subproperties -- properties of the XlsOptions class.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.12 Pictures

```
public SqlManager.QExport.XLS.XlsPictures Pictures {get; set;}
```

Description

The Pictures property allows you to add pictures to the result Excel document. [XlsPictures](#) is a collection of [XlsPicture](#) objects, which define the parameters of each single picture.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.13 Sheets

```
public SqlManager.QExport.XLS.XlsSheets Sheets {get; set;}
```

Description

Use property Sheets to export data to several Excel worksheets with possibility to set a specific format and a separate data source for each of them. Sheets is a collection of XlsSheet objects, properties of which totally correspond to the properties of QExportXLS component, including properties [Command](#), [DataTable](#) and [ListView](#), which allow you to set data source for each sheet; [Options](#) property to set all the sheet options; [FieldFormats](#) to set format for each field separately; and more. Also XlsSheet class has a boolean property Exported, which allows you to define, if current sheet is exported or not.

Note, that if you keep the Sheets collection empty, the result file consists of only one sheet, which uses options, set in the component properties, but if you add at least one sheet to the collection, the sheet options are used, and component properties are not taken into consideration.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.14 StartDataCol

```
public byte StartDataCol {get; set;}
```

Description

The StartDataCol property defines the number of column to start insert data from.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.15 StripStyles

```
public SqlManager.QExport.XLS.XlsStripStyles StripStyles {get; set;}
```

Description

Use StripStyles do define repeating styles for columns or rows of the result sheet. StripStyle determines font size, color and other attributes, border and fill styles and more. To apply this styles to columns or rows use property [StripType](#).

The pictures below show the examples of the result Excel sheet with two styles applied.

The styles were defined in the following way:

```
StripStyles[0].Fill.Background = clrPaleBlue;  
StripStyles[0].Fill.Pattern = ptSolid;  
StripStyles[1].Fill.Background = clrLightTurquoise;  
StripStyles[1].Fill.Pattern = ptSolid;
```

```
StripType == ssCol.
```

```
StripType == ssRow.
```

Note: StripStyles property has higher priority than the FieldFormats property. If the StripStyles property is defined and the StripType property is set to ssCol or ssRow, all the field styles defined in FieldFormats property takes no effect.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.2.16 StripType

```
public enum XlsStripType
{
    ssNone,
    ssCol,
    ssRow
}
```

```
public SqlManager.QExport.XLS.XlsStripType StripType {get; set;}
```

Description

The StripType property defines, if property [StripStyles](#) should be applied to columns or rows of the result Excel sheet. If StripType is ssNone, then [StripStyles](#) are not used, if StripType is ssCol, then [StripStyles](#) are applied to the sheet columns, and if StripType is ssRow, then [StripStyles](#) are applied to the sheet rows. Note, that if StripType is not ssNone, then all the [FieldFormats](#) are ignored, except the [Aggregate](#) property.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.3 QExportXLS Methods

Key methods

[AddBooleanCell](#)

[AddDateTimeCell](#)

[AddMergedCells](#)

[AddNumericCell](#)

[AddStringCell](#)

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Events](#)

2.11.2.3.1 AddBooleanCell

```
public XlsCell AddBooleanCell(ushort Col, ushort Row, bool Value)
```

Description

The AddBooleanCell adds the boolean value defined in the Value parameter to the specified position. The position is defined by the Col parameter as a column number and the Row parameter as a row number. The Row and the Col parameters are 1-based. The returned value is the created [XlsCell](#) object. You can define the properties of this object to customize the created cell.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.3.2 AddDateTimeCell

```
public XlsCell AddDateTimeCell(ushort Col, ushort Row, string DateTimeFormat, DateTime Value)
```

Description

The AddDateTimeCell adds the date/time value defined in the Value parameter to the specified position. The position is defined by the Col parameter as a column number and the Row parameter as a row number. The Row and the Col parameters are 1-based. The DateTimeFormat parameter is a formatting string for the value. The returned value is the created [XlsCell](#) object. You can define the properties of this object to customize the created cell.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.3.3 AddMergedCells

```
public XlsMergedCells AddMergedCells(ushort FirstRow, ushort LastRow, ushort FirstCol, ushort LastCol)
```

Description

The AddMergedCells method allows you to merge cells in the defined ranges to one cell. The row ranges are set by the FirstRow and the LastRow parameters. The column ranges are set by the FirstCol and the LastCol parameters. These parameters are 1-based. The returned value is the [XlsMergedCells](#) object.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.3.4 AddNumericCell

```
public XlsCell AddNumericCell(ushort Col, ushort Row, string NumericFormat, double Value)
```

Description

The AddNumericCell adds the numeric value defined in the Value parameter to the specified position. The position is defined by the Col parameter as a column number and the Row parameter as a row number. The Row and the Col parameters are 1-based. The NumericFormat parameter is a formatting string for the value. The returned value is the created [XlsCell](#) object. You can define the properties of this object to customize the created cell.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.3.5 AddStringCell

```
public XlsCell AddStringCell(ushort Col, ushort Row, string Value)
```

Description

The AddStringCell adds the string value defined in the Value parameter to the specified position. The position is defined by the Col parameter as a column number and the Row parameter as a row number. The Row and the Col parameters are 1-based. The returned value is the created [XlsCell](#) object. You can define the properties of this object to customize the created cell.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4 QExportXLS Events

Key events

[AdvancedBeforeExportRow](#)

[AdvancedExportedRecord](#)

[AdvancedGetExportText](#)

[AfterExportSheet](#)

[BeforeExportSheet](#)

[GetAggregateParams](#)

[GetBeforeDataParams](#)

[GetCaptionParams](#)

[GetDataParams](#)

[GetFooterParams](#)

[GetHeaderParams](#)

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

2.11.2.4.1 AdvancedBeforeExportRow

```
public class XlsBeforeExportRowEventArgs : EventArgs
{
    // Methods
    public XlsBeforeExportRowEventArgs(int Sheet, QExportRow Row, bool Accept);

    // Properties
    public bool Accept { get; set; }
    public QExportRow Row { get; }
    public int Sheet { get; }

    // Fields
    private bool _accept;
    private QExportRow _row;
    private int _sheet;
}

public delegate void XlsBeforeExportRowEvent(object Sender, XlsBeforeExportRowEventArgs e);

public event SqlManager.QExport.Delegates.XlsBeforeExportRowEvent AdvancedBeforeExportRow
```

Description

This event is the expanded analogue of the BeforeExportRow event of the [QExportBase](#) class.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.2 AdvancedExportedRecord

```
public class XlsExportedRecordEventArgs : EventArgs
{
    // Methods
    public XlsExportedRecordEventArgs(int Sheet, int RecNo);

    // Properties
    public int RecNo { get; }
    public int Sheet { get; }

    // Fields
    private int _recNo;
    private int _sheet;
}

public delegate void XlsExportedRecordEvent(object Sender, XlsExportedRecordEventArgs e);

public event SqlManager.QExport.Delegates.XlsExportedRecordEvent
AdvancedExportedRecord
```

Description

The AdvancedExportedRecord event is the expanded analogue of the [ExportedRecord](#) event of the [QExportBase](#) class. The only difference in these events is the XlsExportedRecordEventArgs.Sheet parameter, which defines the result Excel sheet to apply the formatting to.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.3 AdvancedGetExportText

```
public class XlsGetExportTextEventArgs : EventArgs
{
    // Methods
    public XlsGetExportTextEventArgs(int Sheet, int ColNo, string Text);

    // Properties
    public int ColNo { get; }
    public int Sheet { get; }
    public string Text { get; set; }

    // Fields
    private int _colNo;
    private int _sheet;
    private string _text;
}

public delegate void XlsGetExportTextEvent(object Sender, XlsGetExportTextEventArgs e);

public event SqlManager.QExport.Delegates.XlsGetExportTextEvent
AdvancedGetExportText
```

Description

The AdvancedGetExportText event is the expanded analogue of the [GetExportText](#) event of the [QExportBase](#) class. The only difference in these events is the XlsGetExportTextEventArgs.Sheet parameter, which defines the result Excel sheet to apply the formatting to.

Note: the AdvancedGetExportText event takes place AFTER the GetExportText event.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.4 AfterExportSheet

```
public delegate void XsExportSheetEvent(object Sender, int SheetIndex);
```

```
public event SqlManager.QExport.Delegates.XsExportSheetEvent AfterExportSheet
```

Description

This event takes place after exporting every sheet.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.5 BeforeExportSheet

```
public delegate void XlsExportSheetEvent(object Sender, int SheetIndex);
```

Description

This event takes place before exporting every sheet.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.6 GetAggregateParams

```
public class GetAggregateParamsEventArgs : EventArgs
{
    // Methods
    public GetAggregateParamsEventArgs(int Sheet, int Col, XlsFormat Format, string FormatText,
    string Value);

    // Properties
    public int Col { get; }
    public XlsFormat Format { get; }
    public string FormatText { get; set; }
    public int Sheet { get; }
    public string Value { get; set; }

    // Fields
    private int _col;
    private XlsFormat _format;
    private string _formatText;
    private int _sheet;
    private string _value;
}

public delegate void GetAggregateParamsEvent(object Sender, GetAggregateParamsEventArgs e);

public event SqlManager.QExport.Delegates.GetAggregateParamsEvent
GetAggregateParams
```

Description

The GetAggregateParams event takes place, when the parameters of the cell, containing aggregate function, are received. Depending on the column number (GetAggregateParamsEventArgs.Col), you can change the format (GetAggregateParamsEventArgs.Format) of the cell value (GetAggregateParamsEventArgs.Value), or, if the value is empty, set a string value to the cell, e.g.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.7 GetBeforeDataParams

```
public class GetHeaderFooterParamsEventArgs : EventArgs
{
    // Methods
    public GetHeaderFooterParamsEventArgs(int Sheet, int Col, int Row, XlsFormat Format, string Str);

    // Properties
    public int Col { get; }
    public XlsFormat Format { get; }
    public int Row { get; }
    public int Sheet { get; }
    public string Str { get; set; }

    // Fields
    private int _col;
    private XlsFormat _format;
    private int _row;
    private int _sheet;
    private string _str;
}

public delegate void GetHeaderFooterParamsEvent(object Sender, GetHeaderFooterParamsEventArgs e);

public event SqlManager.QExport.Delegates.GetHeaderFooterParamsEvent GetBeforeDataParams
```

Description

The GetBeforeDataParams event takes place, when an empty cell should be inserted to the result table (e.g. if [StartDataCol](#)>0). Depending on the cell position (GetHeaderFooterParamsEventArgs.Col, GetHeaderFooterParamsEventArgs.Row) and format (GetHeaderFooterParamsEventArgs.Format) you can set a string value to the cell.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.8 OnGetCaptionParams

```
public class GetCaptionParamsEventArgs : EventArgs
{
    // Methods
    public GetCaptionParamsEventArgs(int Sheet, int Col, XlsFormat Format, string Caption);

    // Properties
    public string Caption { get; set; }
    public int Col { get; }
    public XlsFormat Format { get; }
    public int Sheet { get; }

    // Fields
    private string _caption;
    private int _col;
    private XlsFormat _format;
    private int _sheet;
}
```

```
public delegate void GetCaptionParamsEvent(object Sender, GetCaptionParamsEventArgs e);
```

```
public event SqlManager.QExport.Delegates.GetCaptionParamsEvent GetCaptionParams
```

Description

The GetCaptionsParams event takes place when the parameters of the cells with column captions are received. Depending on the column number (GetCaptionParamsEventArgs.Col) and cell format (GetCaptionParamsEventArgs.Format) you can edit the cell value.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.9 GeDataParams

```
public class GetDataParamsEventArgs : EventArgs
{
    // Methods
    public GetDataParamsEventArgs(int Sheet, int Col, int Row, XlsFormat Format, string FormatText);

    // Properties
    public int Col { get; }
    public XlsFormat Format { get; }
    public string FormatText { get; set; }
    public int Row { get; }
    public int Sheet { get; }

    // Fields
    private int _col;
    private XlsFormat _format;
    private string _formatText;
    private int _row;
    private int _sheet;
}

public delegate void GetDataParamsEvent(object Sender, GetDataParamsEventArgs e);

public event SqlManager.QExport.Delegates.GetDataParamsEvent GetDataParams
```

Description

The GetDataParams event takes place, when the parameters of the data cell are received. Depending on the column and row number (GetDataParamsEventArgs.Col, GetDataParamsEventArgs.Row), you can change the format (GetDataParamsEventArgs.Format) of the cell value.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.10 GetFooterParams

```
public class GetHeaderFooterParamsEventArgs : EventArgs
{
    // Methods
    public GetHeaderFooterParamsEventArgs(int Sheet, int Col, int Row, XlsFormat Format, string Str);

    // Properties
    public int Col { get; }
    public XlsFormat Format { get; }
    public int Row { get; }
    public int Sheet { get; }
    public string Str { get; set; }

    // Fields
    private int _col;
    private XlsFormat _format;
    private int _row;
    private int _sheet;
    private string _str;
}

public delegate void GetHeaderFooterParamsEvent(object Sender, GetHeaderFooterParamsEventArgs e);

public event SqlManager.QExport.Delegates.GetHeaderFooterParamsEvent GetFooterParams
```

Description

The GetFooterParams event takes place when the parameters of the footer cell are received. Depending on the cell position (GetHeaderFooterParamsEventArgs.Col, GetHeaderFooterParamsEventArgs.Row) and format (GetHeaderFooterParamsEventArgs.Format) you can edit the cell value.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.2.4.11 GetHeaderParams

```
public class GetHeaderFooterParamsEventArgs : EventArgs
{
    // Methods
    public GetHeaderFooterParamsEventArgs(int Sheet, int Col, int Row, XlsFormat Format, string Str);

    // Properties
    public int Col { get; }
    public XlsFormat Format { get; }
    public int Row { get; }
    public int Sheet { get; }
    public string Str { get; set; }

    // Fields
    private int _col;
    private XlsFormat _format;
    private int _row;
    private int _sheet;
    private string _str;
}

public delegate void GetHeaderFooterParamsEvent(object Sender, GetHeaderFooterParamsEventArgs e);

public event SqlManager.QExport.Delegates.GetHeaderFooterParamsEvent GetHeaderParams
```

Description

The GetHeaderParams event takes place when the parameters of the header cell are received. Depending on the cell position (GetHeaderFooterParamsEventArgs.Col, GetHeaderFooterParamsEventArgs.Row) and format (GetHeaderFooterParamsEventArgs.Format) you can edit the cell value.

See also:

[QExportXLS](#)

[QExportXLS Properties](#)

[QExportXLS Methods](#)

[QExportXLS Events](#)

2.11.3 XlsFormat object

Namespace

[SqlManager.QExport.XLS](#)

Description

XlsFormat describes the style characteristics used when displaying text in cells. XlsFormat defines font (name, style, size etc.), borders (left, right, top, bottom etc), fill and alignment (horizontal and vertical).

See also:

[XlsFormat Properties](#)

2.11.3.1 XlsFormat Properties

Key properties

[Alignment](#)

[Borders](#)

[Fill](#)

[Font](#)

[Wrap](#)

See also:

[XlsFormat](#)

2.11.3.1.1 Alignment property

public SqlManager.QExport.XLS.XlsAlignment Alignment

Description

The Alignment property defines the text alignment in the cells.

See also:

[XlsFormat](#)

[XlsFormat Properties](#)

2.11.3.1.2 Borders property

```
public SqlManager.QExport.XLS.XlsBorders Borders
```

Description

The Borders property defines the appearance of the cell borders.

See also:

[XlsFormat](#)

[XlsFormat Properties](#)

2.11.3.1.3 Fill property

```
public SqlManager.QExport.XLS.XlsFill Fill
```

Description

The Fill property defines the cell filling.

See also:

[XlsFormat](#)

[XlsFormat Properties](#)

2.11.3.1.4 Font property

```
public SqlManager.QExport.XLS.XlsFont Font
```

Description

The Font property defines the cell font.

See also:

[XlsFormat](#)

[XlsFormat Properties](#)

2.11.3.1.5 Wrap property

public bool Wrap

Description

If Wrap property is true, then text is automatically wrapped in the cells.

See also:

[XlsFormat](#)

[XlsFormat Properties](#)

2.11.4 XlsFormats object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsFieldFormats is a collection of [XlsFormat](#) objects.

See also:

[XlsFormats Properties](#)

2.11.4.1 XlsFormats Properties

Key properties

[this\[\]](#)

See also:

[XlsFormats](#)

2.11.4.1.1 Items property

```
public new SqlManager.QExport.XLS.XlsFormats this[int Index]
```

Description

Use this indexer to access the [XlsFormats](#) objects by Index.

See also:

[XlsFormats](#)

[XlsFieldFormats Properties](#)

2.11.5 XlsFieldFormat object

Namespace

[SqlManager.QExport.XLS](#)

Description

XlsFieldFormat inherits from [XlsFormat](#). It allows you to set the format of the certain table column by setting the source field name in the FieldName property. Along with all the settings, available in XlsFormat, you can set the column width ([Width](#) property) and/or add one of four aggregate functions below the column: sum, average, minimum or maximum value of the column.

See also:

[XlsFieldFormat Properties](#)

2.11.5.1 XlsFieldFormat - Properties

Key properties

[Aggregate](#)

[Width](#)

See also:

[XlsFieldFormat](#)

2.11.5.1.1 Aggregate property

```
public enum XlsAggregate
{
    aggNone,
    aggSum,
    aggAvg,
    aggMin,
    aggMax
}
```

```
public SqlManager.QExport.XLS.XlsAggregate Aggregate
```

Description

Use Aggregate to add an aggregate function to the cell below the column. The following values are available: aggNone - no function, aggSum - sum of the column values, aggAvg - average column value, aggMin - minimum value of the column, and aggMax - maximum value of the column. The default value is aggNone.

See also:

[XlsFieldFormat](#)

[XlsFieldFormat Properties](#)

2.11.5.1.2 Width property

```
public int Width
```

Description

Use Width to set the column width in symbols.

See also:

[XlsFieldFormat](#)

[XlsFieldFormat Properties](#)

2.11.6 XlsFieldFormats object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsFieldFormats is a collection of [XlsFieldFormat](#) objects.

See also:

[XlsFieldFormats Properties](#)

2.11.6.1 XlsFieldFormats Properties

Key properties

[this\[\]](#)

See also:

[XlsFieldFormats](#)

2.11.6.1.1 XlsFieldFormats indexer

```
public new SqlManager.QExport.XLS.XlsFieldFormat this[int Index]
```

Description

Use this indexer to access the [XlsFieldFormat](#) objects by Index.

See also:

[XlsFieldFormats](#)

[XlsFieldFormats Properties](#)

2.11.7 XlsFont object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsFont class contains properties, which define font in the result Excel cells.

See also:

[XlsFont Properties](#)

2.11.7.1 TxIsFont Properties

Key properties

[Charset](#)

[Color](#)

[Name](#)

[Script](#)

[Size](#)

[Bold](#)

[Italic](#)

[Strikeout](#)

[Underline](#)

See also:

[XlsFont](#)

2.11.7.1.1 Charset property

public byte Charset

Description

The CharSet property defines the font charset. Default value is 1 (DEFAULT_CHARSET).

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.2 Color property

```
public SqlManager.Xls.Common.XlsColor Color
```

Description

The Color property defines the font color. Default color is clrBlack. See [XlsColor](#) type for details.

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.3 Name property

public string Name

Description

The Name property defines the font name. Default value is "Arial".

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.4 Script property

```
public enum XlsFontScript
{
    fscNone,
    fscSuperscript,
    fscSubscript
}
```

```
public SqlManager.QExport.XLS.XlsFontScript Script
```

Description

The Script property defines the font script. The following values are available: fscNone - no script, fscSuperScript - superscript, fscSubScript - subscript. The default value is fscNone.

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.5 Size property

public float Size

Description

The Size property sets the font size in points. Default value is 10.

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.6 Bold property

public bool Bold

Description

The Bold property defines the bold font style.

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.7 Italic property

public bool Italic

Description

The Italic property defines the italic font style.

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.8 Strikeout property

public bool Strikeout

Description

The Strikeout property defines the strikeout font style.

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.7.1.9 Underline property

```
public enum XlsFontUnderline
{
    fulNone,
    fulSingle,
    fulDouble,
    fulSingleAccounting,
    fulDoubleAccounting
}
```

```
public SqlManager.QExport.XLS.XlsFontUnderline Underline
```

Description

The Underline property defines the font underlining. The following values are available: fulNone, fulSingle, fulDouble, fulSingleAccounting, fulDoubleAccounting. The default value is fulNone.

See also:

[XlsFont](#)

[XlsFont Properties](#)

XlsFont.Strikeout

[XlsFont](#)

[See also](#)

```
public bool Strikeout
```

Description

The Strikeout property defines the strikeout font style.

See also:

[XlsFont](#)

[XlsFont Properties](#)

2.11.8 XlsBorder object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsBorder class contains properties, which define the color and the style of the cell borders in the result Excel table.

See also:

[XlsBorder Properties](#)

2.11.8.1 TxIsBorder Properties

Key properties

[Color](#)

[Style](#)

See also:

[XlsBorder](#)

2.11.8.1.1 Color property

```
public SqlManager.Xls.Common.XlsColor Color
```

Description

The Color property defines the border color. Default value is clrBlack. See [XlsColor](#) type for details.

See also:

[XlsBorder](#)

[XlsBorder Properties](#)

2.11.8.1.2 Style property

```
public enum XlsBorderStyle
{
    bstNone,
    bstThin,
    bstMedium,
    bstDashed,
    bstDotted,
    bstThick,
    bstDouble,
    bstHair,
    bstMediumDashed,
    bstDashDot,
    bstMediumDashDot,
    bstDashDotDot,
    bstMediumDashDotDot,
    bstSlantedDashDot
}
```

```
public SqlManager.QExport.XLS.XlsBorderStyle Style
```

Description

The Style property defines the border style. Default value is bstNone. See [XlsBorderStyle](#) type for details.

See also:

[XlsBorder](#)

[XlsBorder Properties](#)

2.11.9 XlsBorders object

Namespace

[SqlManager.QExport.XLS](#)

Description

Each property of the XlsBorders object correspond to the separate border of the Excel cell (bottom, left, etc.). The values of these properties are of the same type - [XlsBorder](#), which allows you to set the appearance of the certain border.

See also:

[XlsBorders Properties](#)

[XlsBorder](#)

2.11.9.1 XlsBorders - Properties

Key properties

[Bottom](#)

[DiagDown](#)

[DiagUp](#)

[Left](#)

[Right](#)

[Top](#)

See also:

[XlsBorders](#)

[XlsBorder](#)

2.11.9.1.1 Bottom property

```
public SqlManager.QExport.XLS.XlsBorder Bottom
```

Description

The Bottom property defines the bottom border of the cell.

See also:

[XlsBorders](#)

[XlsBorders Properties](#)

[XlsBorder](#)

2.11.9.1.2 DiagDown property

```
public SqlManager.QExport.XLS.XlsBorder DiagDown
```

Description

The DiagDown property defines the diagonal down border of the cell.

See also:

[XlsBorders](#)

[XlsBorders Properties](#)

[XlsBorder](#)

2.11.9.1.3 DiagUp property

```
public SqlManager.QExport.XLS.XlsBorder DiagUp
```

Description

The DiagUp property defines the diagonal up border of the cell.

See also:

[XlsBorders](#)

[XlsBorders Properties](#)

[XlsBorder](#)

2.11.9.1.4 Left property

```
public SqlManager.QExport.XLS.XlsBorder Left
```

Description

The Left property defines the left border of the cell.

See also:

[XlsBorders](#)

[XlsBorders Properties](#)

[XlsBorder](#)

2.11.9.1.5 Right property

```
public SqlManager.QExport.XLS.XlsBorder Right
```

Description

The Right property defines the right border of the cell.

See also:

[XlsBorders](#)

[XlsBorders Properties](#)

[XlsBorder](#)

2.11.9.1.6 Top property

```
public SqlManager.QExport.XLS.XlsBorder Top
```

Description

The Top property defines the top border of the cell.

See also:

[XlsBorders](#)

[XlsBorders Properties](#)

[XlsBorder](#)

2.11.10 XlsFill object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsFill object contains properties, which determine the cell fill in the result Excel table.

See also:

[XlsFill Properties](#)

2.11.10.1 XlsFill Properties

Key properties

[Background](#)

[Foreground](#)

[Pattern](#)

See also:

[XlsFill](#)

2.11.10.1.1 Background property

```
public SqlManager.Xls.Common.XlsColor Background
```

Description

The Background property defines the background color. Default value is clrWhite.

See also:

[XlsFill](#)

[XlsFill Properties](#)

2.11.10.1.2 Foreground property

```
public SqlManager.Xls.Common.XlsColor Foreground
```

Description

The Foreground property defines the foreground color. Default value is clrBlack.

See also:

[XlsFill](#)

[XlsFill Properties](#)

2.11.10.1.3 Pattern property

```
public enum XlsPattern
{
    ptNone,
    ptSolid,
    ptChess,
    ptWhiteSpots,
    ptBlackSpots,
    ptBoldHorizontal,
    ptBoldVertical,
    ptBoldDiagRight,
    ptBoldDiagLeft,
    ptBoldChess,
    ptRingMail,
    ptThinGorizental,
    ptThinVertical,
    ptThinDiagLeft,
    ptThinDiagRight,
    ptCells,
    ptCrissCross,
    ptThinSpots,
    ptThinThinSpots
}
```

```
public SqlManager.QExport.XLS.XlsPattern Pattern
```

Description

The Pattern property defines the filling pattern. Default value is ptNone. See [XlsPattern](#) type for details.

See also:

[XlsFill](#)

[XlsFill Properties](#)

2.11.11 XlsAlignment object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsAlignment object allows you to define horizontal and vertical alignment of the result Excel cells.

See also:

[XlsAlignment Properties](#)

2.11.11.1 TxIsAlignment Properties

Key properties

[Horizontal](#)

[Vertical](#)

See also:

[XlsAlignment](#)

2.11.11.1.1 Horizontal property

```
public enum XlsHorizontalAlignment
{
    halGeneral,
    halLeft,
    halCenter,
    halRight,
    halFill
}
```

```
public SqlManager.QExport.XLS.XlsHorizontalAlignment Horizontal
```

Description

The Horizontal property defines the horizontal alignment of the cell. The following values are available: halGeneral, halLeft, halCenter, halRight, and halFill. The default value is halGeneral.

See also:

[XlsAlignment](#)

[XlsAlignment Properties](#)

2.11.11.1.2 Vertical property

```
public enum XlsVerticalAlignment
{
    valTop,
    valCenter,
    valBottom,
    valJustify
}
```

```
public SqlManager.QExport.XLS.XlsVerticalAlignment Vertical
```

Description

The Vertical property defines the vertical alignment of the cell. The following values are available: valTop, valCenter, valBottom, and valJustify. The default value is valBottom.

See also:

[XlsAlignment](#)

[XlsAlignment Properties](#)

2.11.12 XlsHyperLink object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsHyperLink class contains properties, which define the hyperlink parameters in the result Excel document.

See also:

[XlsHyperLink Properties](#)

2.11.12.1 XlsHyperLink Properties

Key properties

[Col](#)

[Format](#)

[Row](#)

[ScreenTip](#)

[Style](#)

[Target](#)

[Title](#)

See also:

[XlsHyperLink](#)

2.11.12.1.1 Col property

public int Col

Description

The Col property defines the horizontal position of the link.

See also:

[XlsHyperLink](#)

[XlsHyperLink Properties](#)

2.11.12.1.2 Format property

```
public SqlManager.QExport.XLS.XlsFormat Format
```

Description

The Format property defines parameters of displaying the hyperlink in the result document.

See also:

[XlsHyperLink](#)

[XlsHyperLink Properties](#)

2.11.12.1.3 Row property

public int Row

Description

The Row property defines the vertical position of the link.

See also:

[XlsHyperLink](#)

[XlsHyperLink Properties](#)

2.11.12.1.4 ScreenTip property

public string ScreenTip

Description

The ScreenTip property defines the text of the hint to display in Excel for the link.

See also:

[XlsHyperLink](#)

[XlsHyperLink Properties](#)

2.11.12.1.5 Style property

```
public enum XlsHyperlinkStyle
{
    hlsURL,
    hlsLocalFile
}
```

```
public SqlManager.QExport.XLS.XlsHyperlinkStyle Style
```

Description

The Style property defines the type of the hyperlink target. The following values are available: hlsURL - the global URL, hlsLocalFile - link to local file.

See also:

[XlsHyperLink](#)

[XlsHyperLink Properties](#)

2.11.12.1.6 Target property

public string Target

Description

Use this property to define the hyperlink target.

See also:

[XlsHyperLink](#)

[XlsHyperLink Properties](#)

2.11.12.1.7 Title property

public string Title

Description

The Title property defines the hyperlink text.

See also:

[XlsHyperLink](#)

[XlsHyperLink Properties](#)

2.11.13 XlsHyperLinks object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsHyperLinks is a collection of [XlsHyperLink](#) objects.

See also:

[XlsHyperLinks Properties](#)

2.11.13.1 XlsHyperLinks Properties

Key properties

[this\[\]](#)

See also:

[XlsHyperLinks](#)

2.11.13.1.1 XlsHyperLinks indexer

```
public new SqlManager.QExport.XLS.XlsHyperlink this[int Index]
```

Description

Use this indexer to access the [XlsHyperLink](#) objects by Index.

See also:

[XlsHyperLinks](#)

[XlsHyperLinks Properties](#)

2.11.14 XlsNote object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsNote class contains properties, which define the note parameters in the result Excel document.

See also:

[XlsNote Properties](#)

2.11.14.1 XlsNote Properties

Key properties

[Col](#)

[Format](#)

[Lines](#)

[Row](#)

See also:

[XlsNote](#)

2.11.14.1.1 Col property

public int Col

Description

The Col property defines the horizontal position of the note.

See also:

[XlsNote](#)

[XlsNote Properties](#)

2.11.14.1.2 Format property

```
public SqlManager.QExport.XLS.XlsNoteFormat Format
```

Description

The Format property defines parameters of displaying the note in the result document.

See also:

[XlsNote](#)

[XlsNote Properties](#)

2.11.14.1.3 Lines property

```
public SqlManager.QExport.Collections.StringListCollection Lines
```

Description

The Lines property contains the note text.

See also:

[XlsNote](#)

[XlsNote Properties](#)

2.11.14.1.4 Row property

public int Row

Description

The Row property defines the vertical position of the note.

See also:

[XlsNote](#)

[XlsNote Properties](#)

2.11.15 XlsNotes object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsNotes is a collection of [XlsNote](#) objects.

See also:

[XlsNotes Properties](#)

2.11.15.1 XlsNotes Properties

Key properties

[this\[\]](#)

See also:

[XlsNotes](#)

2.11.15.1.1 XlsNotes indexer

```
public new XlsNote this[int Index]
```

Description

Use this indexer to access the [XlsNote](#) objects by Index.

See also:

[XlsNotes](#)

[XlsNotes Properties](#)

2.11.16 XlsChartSeries object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsChartSeries class contains properties, which define the chart series in the chart, which is defined in the [XlsChart](#) object.

See also:

[XlsChartSeries Properties](#)

2.11.16.1 XlsChartSeries Properties

Key properties

[Color](#)

[DataColumn](#)

[DataRange](#)

[DataRangeType](#)

[Title](#)

See also:

[XlsChartSeries](#)

2.11.16.1.1 Color property

```
public SqlManager.Xls.Common.XlsColor Color
```

Description

The Color property defines the chart series color.

See also:

[XlsChartSeries](#)

[XlsChartSeries Properties](#)

2.11.16.1.2 DataColumn property

public string DataColumn

Description

The DataColumn property defines the data column name for the result chart series. This property is used only if [DataRangeType](#) is rtColumn.

See also:

[XlsChartSeries](#)

[XlsChartSeries Properties](#)

2.11.16.1.3 DataRange property

```
public SqlManager.QExport.XLS.XlsDataRange DataRange
```

Description

The DataRange property allows you to define custom data range for the series. This property is used only if [DataRangeType](#) is rtCustom.

See also:

[XlsChartSeries](#)

[XlsChartSeries Properties](#)

2.11.16.1.4 DataRangeType property

```
public enum XlsRangeType
{
    rtColumn,
    rtCustom
}
```

```
public SqlManager.QExport.XLS.XlsRangeType DataRangeType
```

Description

The DataRangeType property defines the data range type for the series. If DataRangeType value is rtColumn, data range is defined by the data column. The column name for the range is defined by the [DataColumn](#) property. If DataRangeType value is rtCustom, the data range is defined by the [DataRange](#) property.

See also:

[XlsChartSeries](#)

[XlsChartSeries Properties](#)

2.11.16.1.5 Title property

public string Title

Description

The Title property defines the title of the result chart series.

See also:

[XlsChartSeries](#)

[XlsChartSeries Properties](#)

2.11.17 XlsChartSeriesList object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsChartSeriesList is a collection of [XlsChartSeries](#) objects.

See also:

[XlsChartSeriesList Properties](#)

2.11.17.1 XlsChartSeriesList Properties

Key properties

[this\[\]](#)

See also:

[XlsChartSeriesList](#)

2.11.17.1.1 XlsChartSeriesList indexer

```
public new XlsChartSeries this[int Index]
```

Description

Use Items to access the [XlsChartSeries](#) objects by Index.

See also:

[XlsChartSeriesList](#)

[XlsChartSeriesList Properties](#)

2.11.18 XlsChart object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsChart class contains properties, which define the chart parameters in the result Excel document.

See also:

[XlsChart Properties](#)

2.11.18.1 XlsChart Properties

Key properties

[AutoColor](#)

[CategoryLabels](#)

[CategoryLabelsColumn](#)

[CategoryLabelsType](#)

[LegendPlacement](#)

[Position](#)

[Series](#)

[ShowLegend](#)

[Style](#)

[Title](#)

See also:

[XlsChart](#)

2.11.18.1.1 AutoColor property

public bool AutoColor

Description

The AutoColor property enables or disables the automatic defining colors of the chart series in the result Excel document.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.2 CategoryLabels property

```
public SqlManager.QExport.XLS.XlsDataRange CategoryLabels
```

Description

The CategoryLabels property allows you to define the data range for the horizontal axis labels of the chart. This property is used only if [CategoryLabelsType](#) property is rtCustom.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.3 CategoryLabelsType property

```
public enum XlsRangeType
{
    rtColumn,
    rtCustom
}
```

```
public SqlManager.QExport.XLS.XlsRangeType CategoryLabelsType
```

Description

Use the CategoryLabelsType property to define the type of data range for marking the horizontal axis of the chart. If CategoryLabelsType value is rtColumn, data range is defined by the data column. The column name for the range is defined by the [CategoryLabelsColumn](#) property. If CategoryLabelsType value is rtCustom, the data range is defined by the [CategoryLabels](#) property.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.4 CategoryLabelsColumn property

```
public string CategoryLabelsColumn
```

Description

The CategoryLabelsColumn property defines the data column name for the horizontal axis labels of the chart. This property is used only if the [CategoryLabelsType](#) property is set to Horizontal.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.5 LegendPlacement property

```
public enum XlsChartLegendPlacement
{
    clpBottom,
    clpCorner,
    clpTop,
    clpRight,
    clpLeft
}
```

```
public SqlManager.QExport.XLS.XlsChartLegendPlacement LegendPlacement
```

Description

The LegendPlacement property defines the position of the chart legend.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.6 Position property

```
public SqlManager.QExport.XLS.XlsChartPosition Position
```

Description

Use this property to define the chart position in the result Excel document.

See also:

[XlsChart](#)

[XlsChart Properties](#)

[XlsChartPosition object](#)

2.11.18.1.7 Series property

```
public SqlManager.QExport.XLS.XlsChartSeriesList Series
```

Description

This property contains the collection of the chart series, which belongs to this chart.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.8 Show Legend property

public bool ShowLegend

Description

Use the ShowLegend option to enable or disable displaying of the chart legend.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.9 Style property

```
public enum XlsChartStyle
{
    xcsColumn,
    xcsColumn3d,
    xcsBar,
    xcsBar3d,
    xcsLine,
    xcsLineMark,
    xcsLine3d,
    xcsPie,
    xcsPie3d,
    xcsArea,
    xcsArea3d,
    xcsSurface,
    xcsSurface3d,
    xcsRadar,
    xcsRadarArea
}
```

```
public SqlManager.QExport.XLS.XlsChartStyle Style
```

Description

Use the Style property to define the chart style.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.18.1.10 Title property

public string Title

Description

The Title property defines the chart title in the result Excel document.

See also:

[XlsChart](#)

[XlsChart Properties](#)

2.11.19 XlsCharts object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsCharts is a collection of [XlsChart](#) objects.

See also:

[XlsCharts Properties](#)

2.11.19.1 XlsCharts Properties

Key properties

[this\[\]](#)

See also:

[XlsCharts](#)

2.11.19.1.1 XlsCharts indexer

```
public new XlsChart this[int Index]
```

Description

Use Items to access the [XlsChart](#) objects by Index.

See also:

[XlsCharts](#)

[XlsCharts Properties](#)

2.11.20 XlsPicture object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsPicture class defines the picture in the result Excel document. Picture is not a displayable object in itself. It is used to create [XlsImage](#) objects and add them to the document.

See also:

[XlsPicture Properties](#)

2.11.20.1 XlsPicture Properties

Key properties

[Name](#)

[FileName](#)

See also:

[XlsPicture](#)

2.11.20.1.1 Name property

public new string Name

Description

This property defines the name of the picture in the result Excel file.

See also:

[XlsPicture](#)

[XlsPicture Properties](#)

2.11.20.1.2 FileName property

```
public string FileName
```

Description

This property defines the name of the file that contains the picture.

See also:

[XlsPicture](#)

[XlsPicture Properties](#)

2.11.21 XlsPictures object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsPictures is a collection of [XlsPicture](#) objects.

See also:

[XlsPictures Properties](#)

2.11.21.1 XlsPictures Properties

Key properties

[this\[\]](#)

See also:

[XlsPictures](#)

2.11.21.1.1 XlsPictures indexer

```
public new XlsPicture this[int Index]
```

Description

Use Items to access the [XlsPicture](#) objects by Index.

See also:

[XlsPictures](#)

[XlsPictures Properties](#)

2.11.22 XlsImage object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsImage class contains properties, which define the visible image in the result Excel document.

See also:

[XlsImage Properties](#)

2.11.22.1 XlsImage Properties

Key properties

[Col](#)

[PictureName](#)

[Row](#)

[Title](#)

[Zoom](#)

See also:

[XlsImage](#)

2.11.22.1.1 Col property

public int Col

Description

This property defines the horizontal position of the image in the result Excel file.

See also:

[XlsImage](#)

[XlsImage Properties](#)

2.11.22.1.2 PictureName property

```
public string PictureName
```

Description

This property defines the name of the picture that the image uses. The picture is defined by the respective [TxlsPicture](#) object.

See also:

[XlsImage](#)

[XlsImage Properties](#)

2.11.22.1.3 Row property

public int Row

Description

This property defines the vertical position of the image in the result Excel file.

See also:

[XlsImage](#)

[XlsImage Properties](#)

2.11.22.1.4 Title property

public string Title

Description

This property defines the title of the image that would be displayed in the result Excel file.

See also:

[XlsImage](#)

[XlsImage Properties](#)

2.11.22.1.5 Zoom property

public int Zoom

Description

This property defines zooming ratio for the image in the result Excel file in percentage wise.

See also:

[XlsImage](#)

[XlsImage Properties](#)

2.11.23 XlsImages object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsImages is a collection of [XlsImage](#) objects.

See also:

[XlsImages Properties](#)

2.11.23.1 XlsImages Properties

Key properties

[this\[\]](#)

See also:

[XlsImages](#)

2.11.23.1.1 XlsImages indexer

```
public new XlsImage this[int Index]
```

Description

Use Items to access the [XlsImage](#) objects by Index.

See also:

[XlsImages](#)

[XlsImages Properties](#)

2.11.24 XlsCell object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsCell class contains properties, which allow you to place a value to the definite place in the result Excel document.

See also:

[XlsCell Properties](#)

2.11.24.1 XlsCell Properties

Key properties

[CellType](#)

[Col](#)

[DateTimeFormat](#)

[Format](#)

[IsBoolean](#)

[IsDateTime](#)

[IsNumeric](#)

[IsString](#)

[NumericFormat](#)

[Row](#)

[Value](#)

See also:

[XlsCell](#)

2.11.24.1.1 CellType property

```
public enum XlsCellType
{
    ctBoolean,
    ctDateTime,
    ctNumeric,
    ctString
}
```

```
public SqlManager.QExport.XLS.XlsCellType CellType
```

Description

This property defines the type of the cell value.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.2 Col property

public int Col

Description

This property defines the horizontal position of the cell in the result Excel file.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.3 DateTimeFormat property

public string DateTimeFormat

Description

This property defines the formatting string for the date/time values of the cell.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.4 Format property

```
public SqlManager.QExport.XLS.XlsFormat Format
```

Description

This property defines the formatting options for the cell in the result Excel document.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.5 IsBoolean property

public bool IsBoolean

Description

This property returns True if the [CellType](#) property is set to ctBoolean.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.6 IsDateTime property

public bool IsDateTime

Description

This property returns True if the [CellType](#) property is set to ctDateTime.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.7 IsNumeric property

```
public bool IsNumeric
```

Description

This property returns True if the [CellType](#) property is set to ctNumeric.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.8 IsString property

public bool IsString

Description

This property returns True if the [CellType](#) property is set to ctString.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.9 NumericFormat property

public string NumericFormat

Description

This property defines the formatting string for the numeric values of the cell.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.10 Row property

public int Row

Description

This property defines the vertical position of the cell in the result Excel file.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.24.1.11 Value property

public object Value

Description

This property defines the value of the cell.

See also:

[XlsCell](#)

[XlsCell Properties](#)

2.11.25 XlsCells object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsCells is a collection of [XlsCell](#) objects.

See also:

[XlsCells Properties](#)

2.11.25.1 XlsCells Properties

Key properties

[this\[\]](#)

See also:

[XlsCells](#)

2.11.25.1.1 XlsCells indexer

```
public new XlsCell this[int Index]
```

Description

Use Items to access the [XlsCell](#) objects by Index.

See also:

[XlsCells](#)

[XlsCells Properties](#)

2.11.26 XlsMergedCells object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsMergedCells class contains properties, which allow you to merge cells in the result Excel document.

See also:

[XlsMergedCells Properties](#)

2.11.26.1 XlsMergedCells Properties

Key properties

[FirstCol](#)

[FirstRow](#)

[LastCol](#)

[LastRow](#)

See also:

[XlsMergedCells](#)

2.11.26.1.1 FirstCol property

```
public int FirstCol
```

Description

This property defines the first column of the cell range to merge.

See also:

[XlsMergedCells](#)

[XlsMergedCells Properties](#)

2.11.26.1.2 FirstRow property

```
public int FirstRow
```

Description

This property defines the first row of the cell range to merge.

See also:

[XlsMergedCells](#)

[XlsMergedCells Properties](#)

2.11.26.1.3 LastCol property

```
public int LastCol
```

Description

This property defines the last column of the cell range to merge.

See also:

[XlsMergedCells](#)

[XlsMergedCells Properties](#)

2.11.26.1.4 LastRow property

```
public int LastRow
```

Description

This property defines the last row of the cell range to merge.

See also:

[XlsMergedCells](#)

[XlsMergedCells Properties](#)

2.11.27 XlsMergedCellList object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsMergedCellList is a collection of [XlsMergedCells](#) objects.

See also:

[XlsMergedCellList Properties](#)

2.11.27.1 XlsMergedCellList Properties

Key properties

[this\[\]](#)

See also:

[XlsMergedCellList](#)

2.11.27.1.1 XlsMergedCellList indexer

```
public new XlsMergedCells this[int Index]
```

Description

Use Items to access the [XlsMergedCells](#) objects by Index.

See also:

[XlsMergedCellList](#)

[XlsMergedCellList Properties](#)

2.11.28 XlsNoteFormat object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsNoteFormat object contains properties which describe style characteristics used for displaying Excel notes. XlsNoteFormat defines font (name, style, size etc.), fill type, transparency and alignment (horizontal and vertical).

See also:

[XlsNoteFormat Properties](#)

2.11.28.1 XlsNoteFormat Properties

Key properties

[Alignment](#)

[BackgroundColor](#)

[FillType](#)

[Font](#)

[ForegroundColor](#)

[Gradient](#)

[Orientation](#)

[Transparency](#)

See also:

[XlsNoteFormat](#)

2.11.28.1.1 Alignment property

public SqlManager.QExport.XLS.XlsAlignment Alignment

Description

The Alignment property defines the text alignment in the notes.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.28.1.2 BackgroundColor property

```
public System.Drawing.Color BackgroundColor
```

Description

The Background property defines the target color for the gradient fill.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.28.1.3 FillType property

```
public enum XlsNoteFillType
{
    nftSolid,
    nftGradient
}
```

```
public SqlManager.QExport.XLS.XlsNoteFillType FillType
```

Description

The FillType property defines the type of filling the note background. The following values are available: nftSolid - solid fill, and nftGradient - gradient fill. Default is nftSolid.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.28.1.4 Font property

```
public SqlManager.QExport.XLS.XlsFont Font
```

Description

The Font property defines the properties of the note text font to use them in the result document.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.28.1.5 ForegroundColor property

```
public System.Drawing.Color ForegroundColor
```

Description

The ForegroundColor defines the background color for the notes when the FillType property is set to solid fill type, and the source color when the gradient fill is selected.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.28.1.6 Gradient property

```
public enum XlsNoteGradient
{
    ngrHorizontal,
    ngrVertical,
    ngrDiagonalUp,
    ngrDiagonalDown,
    ngrFromCorner,
    ngrFromCenter
}
```

```
public SqlManager.QExport.XLS.XlsNoteGradient Gradient
```

Description

The Gradient property defines the type of the gradient fill. The following gradient types are available: ngrHorizontal - horizontal gradient fill, ngrVertical - vertical gradient fill, ngrDiagonalUp - diagonal gradient from the top left corner to the bottom right corner, ngrDiagonalDown - diagonal gradient from the top right corner to the bottom left corner, ngrFromCorner - gradient fill from corner, and ngrFromCenter - gradient fill from center.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.28.1.7 Orientation property

```
public enum XlsOrientation
{
    xrtNoRotation,
    xlrTopToBottom,
    xlrCounterClockWise,
    xlrClockWise
}
```

```
public SqlManager.QExport.XLS.XlsOrientation Orientation
```

Description

Use the Orientation property to define the note orientation. The following values are available: `xrtNoRotation`, `xlrTopToBottom`, `xlrCounterClockWise`, and `xlrClockWise`. The `xlrClockWise` and `xlrCounterClockWise` values rotate the note to 90 degrees, `xlrTopToBottom` orients the note text vertically.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.28.1.8 Transparency property

public byte Transparency

Description

The Transparency property defines the percentage of the note transparency.

See also:

[XlsNoteFormat](#)

[XlsNoteFormat Properties](#)

2.11.29 XlsDataRange object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsDataRange class describes the data range. This class is used to define the chart data range in the [XlsChartSeries](#) class.

See also:

[XlsDataRange Properties](#)

2.11.29.1 XlsDataRange Properties

Key properties

[Col1](#)

[Col2](#)

[Row1](#)

[Row2](#)

See also:

[XlsDataRange](#)

2.11.29.1.1 Col1 property

public byte Col1

Description

The Col1 property defines the left side of the data range.

See also:

[XlsDataRange](#)

[XlsDataRange Properties](#)

2.11.29.1.2 Col2 property

public byte Col2

Description

The Col2 property defines the right side of the data range.

See also:

[XlsDataRange](#)

[XlsDataRange Properties](#)

2.11.29.1.3 Row 1 property

```
public int Row1
```

Description

The Row1 property defines the top of the data range.

See also:

[XlsDataRange](#)

[XlsDataRange Properties](#)

2.11.29.1.4 Row 2 property

```
public int Row2
```

Description

The Row2 property defines the bottom of the data range.

See also:

[XlsDataRange](#)

[XlsDataRange Properties](#)

2.11.30 XlsChartPosition object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsChartPosition class describes the chart position. This class is used to define the chart position in the [XlsChart](#) class.

See also:

[XlsChartPosition Properties](#)

2.11.30.1 XlsChartPosition Properties

Key properties

[X1](#)

[X2](#)

[Y1](#)

[Y2](#)

See also:

[XlsChartPosition](#)

2.11.30.1.1 X1 property

public byte X1

Description

The X1 property defines the horizontal position of the top left corner of the chart.

See also:

[XlsChartPosition](#)

[XlsChartPosition Properties](#)

2.11.30.1.2 X2 property

public byte X2

Description

The X2 property defines the horizontal position of the bottom right corner of the chart.

See also:

[XlsChartPosition](#)

[XlsChartPosition Properties](#)

2.11.30.1.3 Y1 property

public int Y1

Description

The Y1 property defines the vertical position of the top left corner of the chart.

See also:

[XlsChartPosition](#)

[XlsChartPosition Properties](#)

2.11.30.1.4 Y2 property

public int Y2

Description

The Y2 property defines the vertical position of the bottom right corner of the chart.

See also:

[XlsChartPosition](#)

[XlsChartPosition Properties](#)

2.11.31 XlsSheet object

Namespace

[SqlManager.QExport.XLS](#)

The properties of this class are almost totally corresponded to the properties of QExportXLS component, including properties [Command](#), [DataTable](#) and [ListView](#), which allow you to set data source for each sheet; [Options](#) property to set all the sheet options; [FieldFormats](#) to set format for each field separately; and more. Also XlsSheet class has a boolean property Exported, which allows you to define, if current sheet is exported or not.

See also:

[XlsSheet Properties](#)

2.11.31.1 XlsSheet Properties

Key properties

[DefColWidth](#)

[DefRowHeight](#)

See also:

[XlsSheet](#)

2.11.31.1.1 DefColWidth property

```
public int DefColWidth
```

Description

Use the DefColWidth property to set the column width height for the current sheet.

See also:

[XlsSheet](#)

[XlsSheet Properties](#)

2.11.31.1.2 DefRow Height property

```
public double DefRowHeight
```

Description

Use the DefRowHeight property to set the default row height for the current sheet.

See also:

[XlsSheet](#)

[XlsSheet Properties](#)

2.11.32 XlsSheets object

Namespace

[SqlManager.QExport.XLS](#)

XlsSheets class is a collection of [XlsSheet](#) objects.

See also:

[XlsSheets Properties](#)

2.11.32.1 XlsSheets Properties

Key properties

[this\[\]](#)

See also:

[XlsSheets](#)

2.11.32.1.1 XlsSheets indexer

```
public new XlsSheet this[int Index]
```

Description

Use this indexer to access the [XlsSheet](#) objects by Index.

See also:

[XlsSheets](#)

[XlsSheets Properties](#)

2.11.33 XlsOptions object

Namespace

[SqlManager.QExport.XLS](#)

Description

The XlsOptions class contains properties that set the parameters of the result Excel document.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1 XlsOptions Properties

Key properties

[AggregateFormat](#)

[CaptionsFormat](#)

[DataFormat](#)

[FooterFormat](#)

[HeaderFormat](#)

[PageFooter](#)

[PageHeader](#)

[SheetTitle](#)

[HyperLinkFormat](#)

[NoteFormat](#)

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.1 AggregateFormat property

```
public SqlManager.QExport.XLS.XlsFormat AggregateFormat
```

Description

The AggregateFormat property defines the style of the cells, containing aggregate functions. To add an aggregate function use property [Aggregate](#) of the [XlsFieldFormat](#). You can also write code in the [GetAggregateParams](#) event handler to tune the aggregate in runtime.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.2 CaptionsFormat property

```
public SqlManager.QExport.XLS.XlsFormat CaptionsFormat
```

Description

The CaptionsFormat property defines the style of the cells, containing captions. You can also write code in the [OnGetCaptionParams](#) event handler to tune the [Captions](#) in runtime.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.3 DataFormat property

```
public SqlManager.QExport.XLS.XlsFormat DataFormat
```

Description

The DataFormat property defines the style of the cells, containing the exported data. This style will be used in all data columns as default. If you want to change these settings then you can define property [FieldFormats](#) or write code in the [GetDataParams](#) event handler to tune the data cells in runtime.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.4 FooterFormat property

```
public SqlManager.QExport.XLS.XlsFormat FooterFormat
```

Description

The FooterFormat property defines the footer style in the result file. You can also write code in the [GetFooterParams](#) event handler to tune the [Footer](#) in runtime.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.5 HeaderFormat property

```
public SqlManager.QExport.XLS.XlsFormat HeaderFormat
```

Description

The HeaderFormat property defines the header style in the result file. You can also write code in the [GetHeaderParams](#) event handler to tune the [Header](#) in runtime.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.6 PageFooter property

public string PageFooter

Description

The PageFooter property sets the footer of the result file page. The default setting of this property is Page &P of &N, which puts at the top of every page its number and the total quantity of pages in a book (for more details on the use of special symbols see the note on Microsoft Excel).

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.7 PageHeader property

public string PageHeader

Description

The PageHeader property sets the header of the output file page. The default setting of this property is an empty string.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.8 SheetTitle property

public string SheetTitle

Description

The SheetTitle property sets the title of the result worksheet.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.9 HyperLinkFormat property

```
public SqlManager.QExport.XLS.XlsFormat HyperlinkFormat
```

Description

The HyperLinkFormat property defines the hyperlink style in the result Excel document.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.33.1.10 NoteFormat property

```
public SqlManager.QExport.XLS.XlsNoteFormat NoteFormat
```

Description

The NoteFormat property defines the note style in the result Excel document.

See also:

[XlsOptions](#)

[XlsOptions Properties](#)

2.11.34 XlsColor enum

Namespace

[SqlManager.QExport.XLS](#)

```
public enum XlsColor
{
    clrBlack,
    clrBrown,
    clrOliveGreen,
    clrDarkGreen,
    clrDarkTeal,
    clrDarkBlue,
    clrIndigo,
    clrGray80Percent,
    clrDarkRed,
    clrOrange,
    clrDarkYellow,
    clrGreen,
    clrTeal,
    clrBlue,
    clrBlueGray,
    clrGray50Percent,
    clrRed,
    clrLightOrange,
    clrLime,
    clrSeaGreen,
    clrAqua,
    clrLightBlue,
    clrViolet,
    clrGray40Percent,
    clrPink,
    clrGold,
    clrYellow,
    clrBrightGreen,
    clrTurquoise,
    clrSkyBlue,
    clrPlum,
    clrGray25Percent,
    clrRose,
    clrTan,
    clrLightYellow,
    clrLihtGreen,
    clrLightTurquoise,
    clrPaleBlue,
    clrLavender,
    clrWhite,
    clrColor1,
    clrColor2,
    clrColor3,
    clrColor4,
    clrColor5,
```

```
clrColor6,  
clrColor7,  
clrColor8,  
clrColor9,  
clrColor10,  
clrColor11,  
clrColor12,  
clrColor13,  
clrColor14,  
clrColor15,  
clrColor16  
}
```

2.11.35 XlsBorderStyle enum

Namespace

[SqlManager.QExport.XLS](#)

```
public enum XlsBorderStyle
{
    bstNone,
    bstThin,
    bstMedium,
    bstDashed,
    bstDotted,
    bstThick,
    bstDouble,
    bstHair,
    bstMediumDashed,
    bstDashDot,
    bstMediumDashDot,
    bstDashDotDot,
    bstMediumDashDotDot,
    bstSlantedDashDot
}
```

2.11.36 XlsPattern enum

Namespace

[SqlManager.QExport.XLS](#)

```
public enum XlsPattern
{
    ptNone,
    ptSolid,
    ptChess,
    ptWhiteSpots,
    ptBlackSpots,
    ptBoldHorizontal,
    ptBoldVertical,
    ptBoldDiagRight,
    ptBoldDiagLeft,
    ptBoldChess,
    ptRingMail,
    ptThinGorizontal,
    ptThinVertical,
    ptThinDiagLeft,
    ptThinDiagRight,
    ptCells,
    ptCrissCross,
    ptThinSpots,
    ptThinThinSpots
}
```

2.12 SqlManager.QExport.XML namespace

2.12.1 SqlManager.QExport.XML namespace Reference

Components

[QExportXML](#)

Objects

[XmlOptions](#)

2.12.2 QExportXML component

2.12.2.1 QExportXML component Reference

Namespace

[SqlManager.QExport.XML](#)

Description

The QExportXML component allows you to export your data to the XML format.

See also:

[QExportXML Properties](#)

2.12.2.2 QExportXML Properties

Key properties

[Options](#)

See also:

[QExportXML](#)

2.12.2.2.1 Options

```
public SqlManager.QExport.XML.XmlOptions Options
```

Description

The Options property is "complex" and contains some subproperties -- properties of the XmlOptions class.

See also:

[QExportXML](#)

[QExportXML Properties](#)

2.12.3 XmlOptions object

Namespace

[SqlManager.QExport.XML](#)

Description

The XmlOptions class contains the properties that set the parameters of the result document.

See also:

[XmlOptions Properties](#)

2.12.3.1 TXMLOptions Properties

Key properties

[Encoding](#)

[StandAlone](#)

[Version](#)

See also:

[XmlOptions](#)

2.12.3.1.1 Encoding property

public string Encoding

Description

This property allows you to set the encoding of the result document.

See also:

[XmlOptions](#)

[XmlOptions Properties](#)

2.12.3.1.2 StandAlone property

public bool StandAlone

Description

If this option is true, then the result document will be standalone.

See also:

[XmlOptions](#)

[XmlOptions Properties](#)

2.12.3.1.3 Version property

public string Version

Description

This property displays the current XML version.

See also:

[XmlOptions](#)

[XmlOptions Properties](#)

Credits

Software Developers:

Alex Paclin

Alexey Butalov

Technical Writers:

Dmitry Doni

Semyon Slobodenyuk

Olga Ryabova

Cover Designer:

Tatyana Makurova

Translators:

Anna Shulkina

Sergey Fominykh

Team Coordinators:

Alexey Butalov

Alexander Chelyadin

Roman Tkachenko