



Advanced PDF Generator for RAD Studio User's Manual

© 1999-2025 EMS Software Development



Advanced PDF Generator for RAD Studio User's Manual

© 1999-2025 EMS Software Development

All rights reserved.

This manual documents EMS Advanced PDF Generator for RAD Studio, version 1.3.

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: 20.10.2025

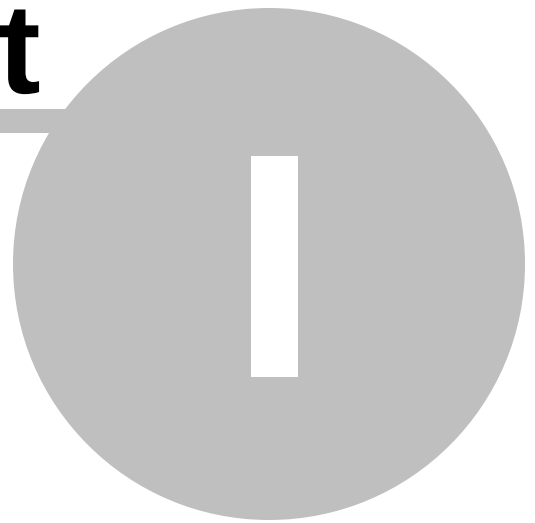
Table of Contents

Part I Welcome to Advanced PDF Generator for RAD Studio!	7
Overview	7
What's new	8
Installation	9
Registration	11
How to register Advanced PDF Generator	12
Other EMS Products	13
Part II Advanced PDF Generator	20
TQuickPdf	21
TQuickPdf Reference	21
Properties	22
Contents.....	23
Document.....	24
EmbeddedFonts.....	25
FileName.....	26
PageInfo.....	27
PDFAuthor.....	28
PDFCreator.....	29
PDFKeywords.....	30
PDFSubject.....	31
PDFTitle.....	32
Security.....	33
SecurityOptions.....	34
SecurityOwnerPassword.....	35
SecurityUserPassword.....	36
Methods	37
Clear.....	38
GenerateToFile.....	39
GenerateToString.....	40
GenerateToStream.....	41
TQuickPdfContentsItem	42
TQuickPdfContentsItem Reference	42
Properties	43
Count.....	44
Items.....	45
Link.....	46
Parent.....	47
Title.....	48
Methods	49
Add.....	50
Create.....	51

Clear.....	52
TQuickPdfDocument	53
TQuickPdfDocument Reference	53
Properties	54
Footer.....	55
Header	56
MainTable.....	57
Section.....	58
Width.....	59
Methods	60
SetMargins.....	61
TQuickPdfPageInfo	62
TQuickPdfPageInfo Reference	62
Properties	63
Height.....	64
MarginBottom.....	65
MarginLeft.....	66
MarginRight.....	67
MarginTop.....	68
Size.....	69
Width.....	70
TRenderAtom	71
TRenderAtom Reference	71
Properties	72
BgColor.....	73
Height.....	74
LinkName.....	75
LinkTarget.....	76
Width.....	77
Methods	78
Create.....	79
TRenderCell	80
TRenderCell Reference	80
Properties	81
BorderBottom.....	82
BorderLeft.....	83
BorderRight.....	84
BorderTop.....	85
Indent.....	86
Padding.....	87
Methods	88
AddItem.....	89
AddParagraph.....	90
AddText.....	91
AddTextItem.....	92
InternalWidth.....	93
TRenderImage	94
TRenderImage Reference	94
Methods	95
Create.....	96
TRenderRow	97

TRenderRow Reference	97
Properties	98
Cell.....	99
CellCount.....	100
TRenderSequence	101
TRenderSequence Reference	101
Methods	102
AddItem.....	103
AddText.....	104
Create.....	105
New Line.....	106
TRenderTable	107
TRenderTable Reference	107
Properties	108
Border.....	109
Cell.....	110
HasFooter	111
HasHeader.....	112
Padding.....	113
Row Count.....	114
Row s	115
Methods	116
Create.....	117
AddRow	118
LastRow	119
TRenderWord	120
TRenderWord Reference	120
Methods	121
Create.....	122
Part III Units	124
QuickPdf unit	124
QPDFRender unit	125
TRenderTextAlign type	126
Part IV Appendix	128
LinkName and LinkTarget Example	128

Part



1 Welcome to Advanced PDF Generator for RAD Studio!

1.1 Overview

EMS Advanced PDF Generator for RAD Studio is a component that gives you an opportunity to create PDF documents with your Delphi and C++ Builder applications in the simplest and easiest way. There is no need to know PDF specification - you can generate PDF files without any knowledge of PDF format using our Advanced PDF Generator. It allows you to break tables into several parts automatically in order to fit the pages, supports embedded tables, gives you an opportunity to add JPG pictures and much more!

Visit our web-site for details: <https://www.sqlmanager.net/>

Key features

- Easy-to-use component API gives you the easiest way to create PDF documents
- Support for 64-bit Windows target platform
- Automatic division of tables into several parts to fit the pages
- Full support of embedded tables
- Adding JPG, BMP pictures to PDF documents
- Adding EMF or WMF files to PDF in vector format
- Adding any other pictures that can be loaded in the TGraphic component
- Generating PDF documents with internal and external links
- Unicode characters full support
- RAD Studio XE2-XE8, 10 Seattle, 10.1 Berlin, 10.2 Tokyo, 10.3 Rio, 10.4 Sydney, 11 Alexandria, 12 Athens, 13 Florence
- Two encryption methods support: 40-bit and 128-bit
- Defining document properties: author, keywords, producer, subject, title

Product information

Homepage	https://www.sqlmanager.net/products/tools/pdfgenerator
Support Ticket System	https://www.sqlmanager.net/support
Register on-line at	https://www.sqlmanager.net/products/tools/pdfgenerator/buy

1.2 What's new

Version**Advanced PDF Generator for RAD Studio 2.1****Release date**

October 20, 2025

What's new in Advanced PDF Generator for RAD Studio?

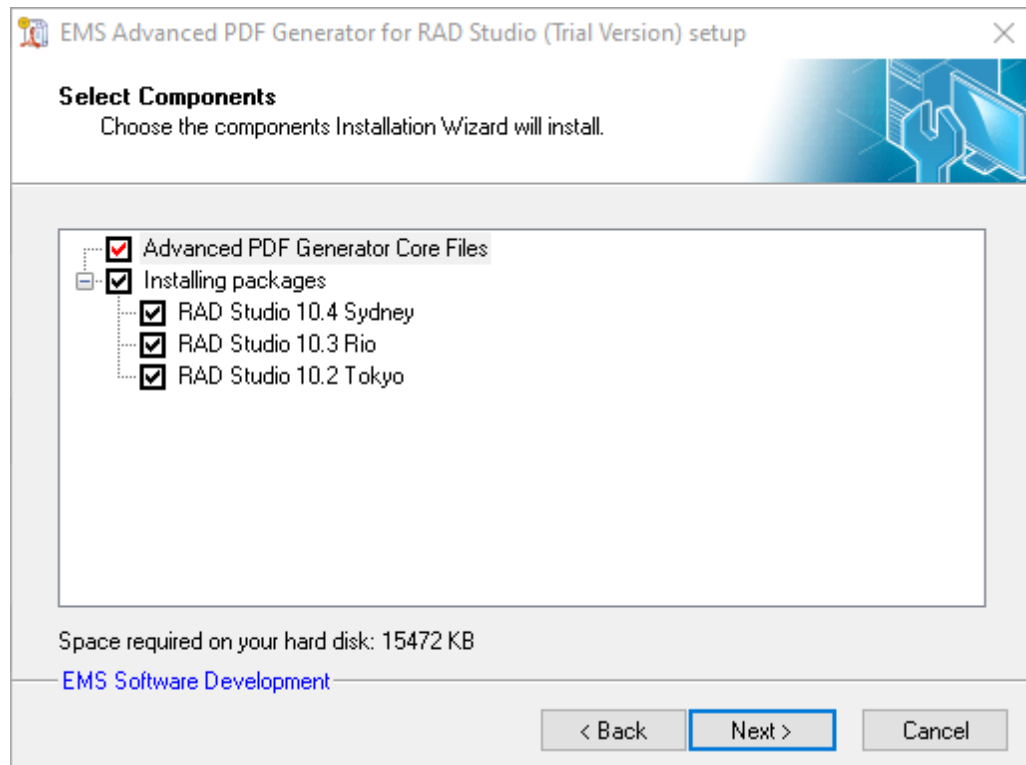
- Implemented support for RAD Studio 13 Florence.
- Implemented support for Windows 11 ARM.
- End of support for RAD Studio 2010 & XE.

1.3 Installation

To install the **trial version** of **Advanced PDF Generator for RAD Studio** onto your system:

- download the distribution package of **Advanced PDF Generator for RAD Studio** from the [download page](#) available at our website;
- unzip the downloaded file to any local directory, e.g. *C:\unzipped*;
- close all currently opened Delphi and/or C++ Builder IDEs, if any;
- run the executable setup file from the local directory and follow the instructions of the installation wizard.

During the installation you will need to select the packages to install:



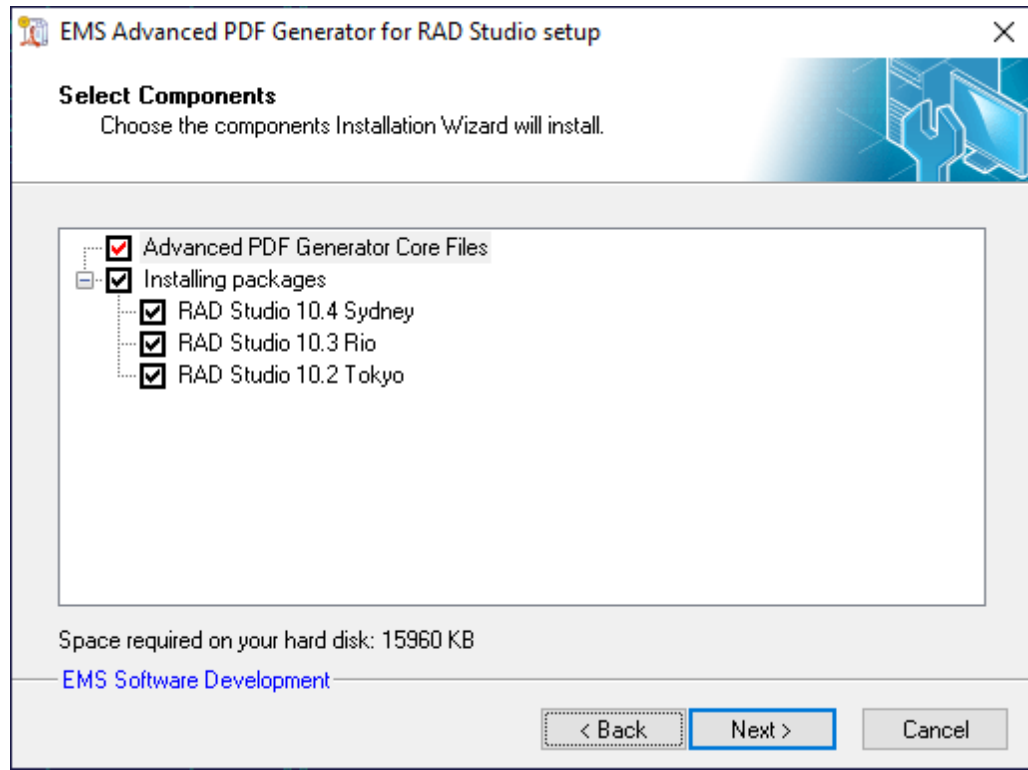
When you are done, you can finish installation of the **trial version** of **Advanced PDF Generator for RAD Studio**.

To install the **full version** of **Advanced PDF Generator for RAD Studio** onto your system:

- download the distribution package of **Advanced PDF Generator for RAD Studio** from the [download page](#) available at our website;
- unzip the downloaded file to any local directory, e.g. *C:\unzipped*;
- close all currently opened Delphi and/or C++ Builder IDEs, if any;
- run the executable setup file from the local directory and follow the instructions of the installation wizard.

Enter valid registration information in the appropriate boxes: **Registration name** and **Registration Key**. See [details](#) on getting this information.

During the installation you will need to select the packages to install:



When you are done, you can finish installation of the **full version** of **Advanced PDF Generator for RAD Studio**.

Note: If the above given instructions have been insufficient for successful installation of the component suite, please refer to the *readme.1st* file distributed with the product.

1.4 Registration

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.

Product distribution	PayPro Global
Advanced PDF Generator for RAD Studio Component Full version (with sources)*	Register Now!
Advanced PDF Generator for RAD Studio Component Trial version	Download Now!

* **EMS Maintenance Program** provides the following benefits:

- Free software bug fixes, enhancements, updates and upgrades during the maintenance period
- Free unlimited communications with technical staff for the purpose of reporting Software failures
- Free reasonable number of communications for the purpose of consultation on operational aspects of the software

After your maintenance expires you will not be able to update your software or get technical support. To protect your investments and have your software up-to-date, you need to renew your maintenance.

You can easily reinitiate/renew your maintenance with our on-line, speed-through Maintenance Reinstatement/Renewal Interface. After reinitiating/renewal you will receive a confirmation e-mail with all the necessary information.

1.5 How to register Advanced PDF Generator

To register your newly purchased copy of **EMS Advanced PDF Generator for RAD Studio**, perform the following steps:

- receive the notification letter from **PayPro Global** with the registration info;
- enter the **Registration Name** and the **Registration Key** from this letter while [installing](#) the **full version** of the product.

See also:

[Registration](#)

1.6 Other EMS Products

Quick navigation

[MySQL](#)[Microsoft SQL Server](#)[PostgreSQL](#)[InterBase / FireBird](#)[Oracle](#)[IBM DB2](#)[Tools & components](#)

MySQL



[SQL Management Studio for MySQL](#)

EMS SQL Management Studio for MySQL is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



[SQL Manager for MySQL](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



[Data Export for MySQL](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more.



[Data Import for MySQL](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



[Data Pump for MySQL](#)

Migrate from most popular databases (MySQL, PostgreSQL, Oracle, DB2, InterBase/Firebird, etc.) to MySQL.



[Data Generator for MySQL](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



[DB Comparer for MySQL](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



[DB Extract for MySQL](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



[SQL Query for MySQL](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



[Data Comparer for MySQL](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

Microsoft SQL Server



[SQL Management Studio for SQL Server](#)

EMS SQL Management Studio for SQL Server is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



[EMS SQL Backup for SQL Server](#)

Perform backup and restore, log shipping and many other regular maintenance tasks on the whole set of SQL Servers in your company.



[SQL Administrator for SQL Server](#)

Perform administrative tasks in the fastest, easiest and most efficient way. Manage maintenance tasks, monitor their performance schedule, frequency and the last execution result.



[SQL Manager for SQL Server](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



[Data Export for SQL Server](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more



[Data Import for SQL Server](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



[Data Pump for SQL Server](#)

Migrate from most popular databases (MySQL, PostgreSQL, Oracle, DB2, InterBase/Firebird, etc.) to Microsoft® SQL Server™.



[Data Generator for SQL Server](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



[DB Comparer for SQL Server](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



[DB Extract for SQL Server](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



[SQL Query for SQL Server](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



[Data Comparer for SQL Server](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

PostgreSQL



[SQL Management Studio for PostgreSQL](#)

EMS SQL Management Studio for PostgreSQL is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



[EMS SQL Backup for PostgreSQL](#)

Creates backups for multiple PostgreSQL servers from a single console. You can use automatic backup tasks with advanced schedules and store them in local or remote folders or cloud storages



[SQL Manager for PostgreSQL](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



[Data Export for PostgreSQL](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more



[Data Import for PostgreSQL](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



[Data Pump for PostgreSQL](#)

Migrate from most popular databases (MySQL, SQL Server, Oracle, DB2, InterBase/Firebird, etc.) to PostgreSQL.



[Data Generator for PostgreSQL](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



[DB Comparer for PostgreSQL](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



[DB Extract for PostgreSQL](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



[SQL Query for PostgreSQL](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



[Data Comparer for PostgreSQL](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

InterBase / Firebird



[SQL Management Studio for InterBase/Firebird](#)

EMS SQL Management Studio for InterBase and Firebird is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



[SQL Manager for InterBase/Firebird](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



[Data Export for InterBase/Firebird](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more



[Data Import for InterBase/Firebird](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



[Data Pump for InterBase/Firebird](#)

Migrate from most popular databases (MySQL, SQL Server, Oracle, DB2, PostgreSQL, etc.) to InterBase/Firebird.



[Data Generator for InterBase/Firebird](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



[DB Comparer for InterBase/Firebird](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



[DB Extract for InterBase/Firebird](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



[SQL Query for InterBase/Firebird](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



[Data Comparer for InterBase/Firebird](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

Oracle



[SQL Management Studio for Oracle](#)

EMS SQL Management Studio for Oracle is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



[SQL Manager for Oracle](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



[Data Export for Oracle](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more.



[Data Import for Oracle](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via

user-friendly wizard interface.



[Data Pump for Oracle](#)

Migrate from most popular databases (MySQL, PostgreSQL, MySQL, DB2, InterBase/Firebird, etc.) to Oracle



[Data Generator for Oracle](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



[DB Comparer for Oracle](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



[DB Extract for Oracle](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



[SQL Query for Oracle](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



[Data Comparer for Oracle](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

IBM DB2



[SQL Manager for DB2](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



[Data Export for DB2](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more.



[Data Import for DB2](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



[Data Pump for DB2](#)

Migrate from most popular databases (MySQL, PostgreSQL, Oracle, MySQL, InterBase/Firebird, etc.) to DB2



[Data Generator for DB2](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



[DB Extract for DB2](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



[SQL Query for DB2](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts

based on retrieved data quickly and more.

[Scroll to top](#)

Tools & components



[Advanced Data Export for RAD Studio VCL](#)

Advanced Data Export for RAD Studio VCL allows you to save your data in the most popular office programs formats.



[Advanced Data Export .NET](#)

Advanced Data Export .NET is a component for Microsoft Visual Studio .NET that will allow you to save your data in the most popular data formats for the future viewing, modification, printing or web publication. You can export data into MS Access, MS Excel, MS Word (RTF), PDF, TXT, DBF, CSV and more! There will be no need to waste your time on tiresome data conversion - Advanced Data Export will do the task quickly and will give the result in the desired format.



[Advanced Data Import for RAD Studio VCL](#)

Advanced Data Import for RAD Studio VCL will allow you to import your data to the database from files in the most popular data formats.



[Advanced PDF Generator for RAD Studio](#)

Advanced PDF Generator for RAD Studio gives you an opportunity to create PDF documents with your applications written on Delphi or C++ Builder.



[Advanced Query Builder for RAD Studio VCL](#)

Advanced Query Builder for RAD Studio VCL is a powerful component for Delphi and C++ Builder intended for visual building SQL statements for the SELECT, INSERT, UPDATE and DELETE clauses.



[Advanced Excel Report for RAD Studio](#)

Advanced Excel Report for RAD Studio is a powerful band-oriented generator of template-based reports in MS Excel.

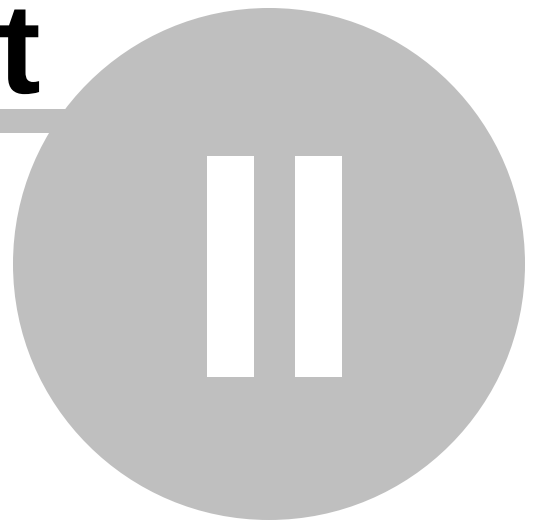


[Advanced Localizer for RAD Studio VCL](#)

Advanced Localizer for RAD Studio VCL is an indispensable component for Delphi for adding multilingual support to your applications.

[Scroll to top](#)

Part



2 Advanced PDF Generator

EMS Advanced PDF Generator for RAD Studio represents a set of tools for generating PDF documents with text, links, images, tables, etc. None of the mechanisms of cooperation between applications in Microsoft Windows environment (DDE, OLE) is used when generating PDF documents, which ensures an extremely high speed of work.

The suite consists of the main [TQuickPdf](#) with basic properties, and a set of various objects describing PDF elements.

Advanced PDF Generator for RAD Studio provides a collection of the following components:

Component	Unit	Brief description
TQuickPDF	QuickPDF	The main component of the suite
TQuickPDFContentsItem	QuickPDF	An item of the document contents
TQuickPDFDocument	QuickPDF	An entire PDF document
TQuickPDFPageInfo	QuickPDF	Information about document page sizes
TRenderAtom	QPDFRender	A basic class for all document objects
TRenderCell	QPDFRender	A table cell
TRenderImage	QPDFRender	An image
TRenderRow	QPDFRender	A table row
TRenderSequence	QPDFRender	A sequence of objects
TRenderTable	QPDFRender	A table
TRenderWord	QPDFRender	An unbreakable sentence

2.1 TQuickPdf

2.1.1 TQuickPdf Reference

Unit

[QuickPdf](#)

Description

The *TQuickPDF* component is the main component of the suite. In this class the basic properties of the document are determined, and so are the three basic methods - [GenerateToString](#), [GenerateToFile](#), and [GenerateToStream](#).

2.1.2 Properties

▶ Run-time only

 Key properties

- ▶ [Contents](#)
- ▶ [Document](#)
- [EmbeddedFonts](#)
- [FileName](#)
- [PageInfo](#)
- [PDFAuthor](#)
- [PDFCreator](#)
- [PDFKeywords](#)
- [PDFSubject](#)
- [PDFTitle](#)
- [Security](#)
- [SecurityOptions](#)
- [SecurityOwnerPassword](#)
- [SecurityUserPassword](#)

2.1.2.1 Contents

property Contents: [TQuickPdfContentsItem](#);

Description

The *Contents* property is used to define the document contents. The document contents is formed up as a tree of [TQuickPdfContentsItem](#) objects. Each node in a contents consists of a label and a name of the link which the contents item is associated with. Each node can have a list of subnodes associated with it. This property is accessible only at runtime.

See also:

[TQuickPdfContentsItem Class](#)

2.1.2.2 Document

property Document: [TQuickPdfDocument](#);

Description

The *Document* property describes the entire PDF document as a set of objects, such as tables, text, images, etc. This property is read-only and accessible only at runtime.

See also:

[TQuickPdfDocument Class](#)

2.1.2.3 EmbeddedFonts

property EmbeddedFonts: **Boolean**;

Description

Set this option to True to embed all the fonts used in the document to the result PDF file.

2.1.2.4 FileName

property FileName: **String**;

Description

The *FileName* property defines the name of the result PDF document. The property must not be empty; if it is empty in the moment of the [GenerateToFile](#) method invocation, the component will stop being executed having generated an exception with the message 'FileName property is not assigned'.

2.1.2.5 PageInfo

property PageInfo: [TQuickPdfPageInfo](#);

Description

The *PageInfo* property contains the information about the document page size.

2.1.2.6 PDFAuthor

property PDFAuthor: **String**;

Description

The *PDFAuthor* property defines an author of the result PDF document.

See also:

[PDFCreator](#)

[PDFKeywords](#)

[PDFSubject](#)

[PDFTitle](#)

2.1.2.7 PDFCreator

property PDFCreator: **String**;

Description

The *PDFCreator* property defines a name of the program that generated the result PDF document.

See also:

[PDFAuthor](#)

[PDFKeywords](#)

[PDFSubject](#)

[PDFTitle](#)

2.1.2.8 PDFKeywords

property PDFKeywords: **String**;

Description

The *PDFKeywords* property defines a list of keywords that relates to the result PDF document.

See also:

[PDFAuthor](#)

[PDFCreator](#)

[PDFSubject](#)

[PDFTitle](#)

2.1.2.9 PDFSubject

property PDFSubject: **String**;

Description

The *PDFSubject* property defines a subject of the result PDF document.

See also:

[PDFAuthor](#)

[PDFCreator](#)

[PDFKeywords](#)

[PDFTitle](#)

2.1.2.10 PDFTitle

property PDFTitle: **String**;

Description

The *PDFTitle* property defines a title of the result PDF document.

See also:

[PDFAuthor](#)

[PDFCreator](#)

[PDFKeywords](#)

[PDFSubject](#)

2.1.2.11 Security

property Security: TQuickPdfSecurityMode;

type

TQuickPdfSecurityMode = (secDisabled, sec40bit, sec128bit);

Description

The *Security* property defines an encoding method that is used to crypt the result document.

See also:

[SecurityOptions](#)

[SecurityOwnerPassword](#)

[SecurityUserPassword](#)

2.1.2.12 SecurityOptions

property SecurityOptions: TQuickPdfSecurityOptions;

type

TQuickPdfSecurityOptions = **set of** (secPrint, secExtraction, secModify);

Description

The *SecurityOptions* property is a set of abilities that are given to user who accessed the document using a password that defined in the [SecurityUserPassword](#). Use this property to limit rights to print, modify, and extract a document content.

See also:

[SecurityOptions](#)

[SecurityOwnerPassword](#)

[SecurityUserPassword](#)

2.1.2.13 SecurityOwnerPassword

property SecurityOwnerPassword: **String**;

Description

The *SecurityUserPassword* property defines a user password to access the document. This password gives an ability to use the document with unlimited rights.

See also:

[PDFAuthor](#)

[PDFCreator](#)

[PDFKeywords](#)

[PDFTitle](#)

2.1.2.14 SecurityUserPassword

property SecurityUserPassword: **String**;

Description

The *SecurityUserPassword* property defines a user password to access the document. This password gives an ability to use the document with rights that are defined by the [SecurityOptions](#) property.

See also:

[PDFAuthor](#)

[PDFCreator](#)

[PDFKeywords](#)

[PDFTitle](#)

2.1.3 Methods

 Key methods

[Clear](#)

[GenerateToString](#)

[GenerateToFile](#)

[GenerateToStream](#)

2.1.3.1 Clear

procedure Clear;

Description

Use this method to clear the document.

2.1.3.2 GenerateToFile

procedure GenerateToFile;

Description

Use *GenerateToFile* to generate a document to the file specified by the [FileName](#) property.

See also:

[GenerateToStream Method](#)

[GenerateToString Method](#)

2.1.3.3 GenerateToString

function GenerateToString: **String**;

Description

Use *GenerateToString* to generate a document to a string that is returned as a result of the function execution.

See also:

[GenerateToFile Method](#)

[GenerateToStream Method](#)

2.1.3.4 GenerateToStream

procedure GenerateToStream(Stream: TStream);

Description

Use *GenerateToStream* to generate a document to the stream specified by the Stream parameter. This can be any TStream descendant.

See also:

[GenerateToFile Method](#)

[GenerateToStream Method](#)

2.2 TQuickPdfContentsItem

2.2.1 TQuickPdfContentsItem Reference

Unit

[QuickPdf](#)

Description

The *TQuickPDFContentsItem* class represents an item of a contents. Use this class to create your contents menu for the document. Each item in a contents consists of a label and a name of the link which the contents item is associated with. Each item can have a list of subitems associated with it.

2.2.2 Properties

▶ Run-time only

 Key properties

- ▶ [Count](#)
- ▶ [Items](#)
- ▶ [Link](#)
- ▶ [Parent](#)
- ▶ [Title](#)

2.2.2.1 Count

property Count: **Integer**;

Description

The *Count* property returns an amount of subitems for the item. This property is read-only and accessible only at runtime.

2.2.2.2 Items

property Items[Index: **Integer**]: [TQuickPdfContentsItem](#);

Description

Use *Items* to access subitems for the item by index. This property is read-only and accessible only at runtime.

2.2.2.3 Link

property `Link`: **String**;

Description

The *Link* property defines a name of the link in the document which the current item points to. Link names are specified by the [LinkName](#) property of all objects in the document.

2.2.2.4 Parent

property Parent: [TQuickPdfContentsItem](#);

Description

The *Parent* property returns the item which the current item is associated with. This property is accessible only at runtime.

2.2.2.5 Title

property Title: **String**;

Description

The *Title* property defines the displayable text for the contents item. This text is displayed in PDF contents.

See also:

[LinkTarget](#)

2.2.3 Methods

 Key methods

[Add](#)

[Clear](#)

2.2.3.1 Add

function Add(ATitle, ALink: **String**): [TQuickPdfContentsItem](#);

Description

The *Add* method creates a subitem for the item and adds it to a list of subitems. The *ATitle* parameter specifies a title of the new subitem, and the *ALink* property specifies a link which the new subitem is associated with.

See also:

[Create](#)

[Clear](#)

2.2.3.2 Create

constructor Create(ATitle, ALink: **String**; AParent: [TQuickPDFContentsItem](#) = nil); over

Description

Use this method to create new instances of the [TQuickPDFContentsItem](#) class.

See also:

[Add](#)

[Clear](#)

2.2.3.3 Clear

procedure Clear;

Description

Use the *Clear* method to delete all children of an item, freeing all associated memory.

See also:

[Add](#)

[Create](#)

2.3 TQuickPdfDocument

2.3.1 TQuickPdfDocument Reference

Unit

[QuickPdf](#)

Description

The *TQuickPDFDocument* class describes the PDF document as a [header](#), [footer](#), and a set of [document sections](#). This class cannot be created by the user, but can be accessed at runtime through the [Document](#) property of the [TQuickPDF](#) component.

See also:

[TQuickPDF Component](#)

2.3.2 Properties

▶ Run-time only

 Key properties

- ▶ [Footer](#)
- ▶ [Header](#)
- ▶ [MainTable](#)
- ▶ [Section](#)
- ▶ [Width](#)

2.3.2.1 Footer

property Footer: [TRenderCell](#);

Description

The *Footer* property defines the footer for each page of the document. As a matter of fact, this property returns the first cell in the last row of the main document table defined by the [MainTable](#) property. The *Footer* property is read-only and accessible only at runtime.

See also:

[Header Property](#)

[MainTable Property](#)

[TRenderCell Class](#)

2.3.2.2 Header

property Header: [TRenderCell](#);

Description

The *Header* property defines the header for each page of the document. As a matter of fact, this property returns the first cell in the first row of the main document table defined by the [MainTable](#) property. The *Header* property is read-only and accessible only at runtime.

See also:

[Footer Property](#)

[MainTable Property](#)

[TRenderCell Class](#)

2.3.2.3 MainTable

property MainTable: [TRenderTable](#);

Description

Technically, all the document is represented as a one-column table. The first row is the document *header*, the last is the *footer*, and all the intervening rows are document *sections*. The *MainTable* property defines the main table of the document. Through this property you can work with an entire document in the same way as with a simple table. The *MainTable* property is read-only and accessible only at runtime.

See also:

[Footer Property](#)

[Header Property](#)

[TRenderTable Class](#)

2.3.2.4 Section

property Section[Index: **Integer**]: [TRenderCell](#);

Description

Technically, all the document is represented as a one-column table. The first row is the document *header*, the last is the *footer*, and all the intervening rows are document *sections*. The *Section* property is an array of all the document sections. You can access them by index using this property. As a matter of fact, the *Section[Index]* returns the first cell in the *Index+1* row of the main document table defined by the [MainTable](#) property. Each section begins from the new page, so you can use sections to manually break your document to pages. The Sections property is read-only and accessible only at runtime.

See also:

[Footer](#)

[Header](#)

[MainTable](#)

[TRenderCell Class](#)

2.3.2.5 Width

property Width: **Integer**;

Description

The *Width* property returns the width of the work area in points (1/72 inches). This property is accessible only at runtime.

2.3.3 Methods

 Key methods

[SetMargins](#)

2.3.3.1 SetMargins

procedure SetMargins(left, top, right, bottom: **Integer**);

Description

Use this method to set the document margins in points (1/72 inches).

2.4 TQuickPdfPageInfo

2.4.1 TQuickPdfPageInfo Reference

Unit

[QuickPdf](#)

Description

The *TQuickPDFContentsItem* class contains all the properties needed to customize a document page size and layout.

2.4.2 Properties

▶ Run-time only

 Key properties

[Height](#)
[MarginBottom](#)
[MarginLeft](#)
[MarginRight](#)
[MarginTop](#)
[Size](#)
[Width](#)

2.4.2.1 Height

property Height: **Integer**;

Description

The *Height* property defines the page height in points (1/72 inches).

See also:

[Size](#)

[Width](#)

2.4.2.2 MarginBottom

property MarginBottom: **Integer**;

Description

Use this property to define a height of the bottom margin of the document pages. A value of the *MarginBottom* property is measured in points (1/72 inches).

See also:

[MarginBottom](#)

[MarginLeft](#)

[MarginRight](#)

2.4.2.3 MarginLeft

property MarginLeft: **Integer**;

Description

Use this property to define a width of the left margin of the document pages. A value of the *MarginLeft* property is measured in points (1/72 inches).

See also:

[MarginBottom](#)

[MarginRight](#)

[MarginTop](#)

2.4.2.4 MarginRight

property MarginRight: **Integer**;

Description

Use this property to define a width of the right margin of the document pages. A value of the *MarginRight* property is measured in points (1/72 inches).

See also:

[MarginBottom](#)

[MarginLeft](#)

[MarginTop](#)

2.4.2.5 MarginTop

property MarginTop: **Integer**;

Description

Use this property to define a height of the top margin of the document pages. A value of the *MarginTop* property is measured in points (1/72 inches).

See also:

[MarginBottom](#)

[MarginLeft](#)

[MarginRight](#)

2.4.2.6 Size

property Size: TQuickPdfPageSize;

type TQuickPdfPageSize = (psCustom, psA4, psLetter, psLegal);

Description

Use this property to define standard page size for the document. The following page sizes are available:

psA4 (8.27" X 11.69")

Letter (8.5" X 11")

Legal (8.5" X 14")

The *Size* is set to *psCustom* if the page size differs from a standard page size.

2.4.2.7 Width

property Width: **Integer**;

Description

The *Width* property defines the page width in points (1/72 inches).

See also:

[Height](#)

[Size](#)

2.5 TRenderAtom

2.5.1 TRenderAtom Reference

Unit

[QPDFRender](#)

Description

The *TRenderAtom* class is the basic class for all the PDF objects available in the suite. This class contains most common properties, such as width and height.

2.5.2 Properties

▶ Run-time only

 Key properties

▶ [Height](#)
▶ [Width](#)
[BgColor](#)
[LinkName](#)
[LinkTarget](#)

2.5.2.1 BgColor

property BgColor: **Integer**;

Description

The *BgColor* property defines a background color for the object.

See also:

[Height](#)

2.5.2.2 Height

property Height: **Integer**;

Description

The *Height* property returns the calculated height of the object in points (1/72 inches). This property is read-only and accessible only at runtime.

See also:

[Width](#)

2.5.2.3 LinkName

property LinkName: **String**;

Description

The *LinkName* property is intended to define a link name in the document. To create a link to the current object from any other object:

- 1) set a value of this property for the current object;
- 2) set a value of the *LinkTarget* property for the source object to a value of the *LinkName* property for the current object.

[Example](#)

See also:

[Height](#)

2.5.2.4 LinkTarget

property LinkTarget: **String**;

Description

The *LinkTarget* property is intended to define a link target for the object. To create a link from the current object to any other object:

- 1) set a value of the *LinkName* property for the destination object;
- 2) set a value of the *LinkTarget* property for the current object to a value of the *LinkName* property for the destination object.

[Example](#)

See also:

[Height](#)

2.5.2.5 Width

property Width: **Integer**;

Description

The *Width* property returns the calculated width of the object in points (1/72 inches). This property is read-only and accessible only at runtime.

See also:

[Height](#)

2.5.3 Methods

▶ Run-time only

 Key properties

[Create](#)

2.5.3.1 Create

constructor `Create(_Width: Integer); overload;`

Description

Use this method to create new instances of the [TRenderAtom](#) class. The `_Width` parameter defines a width of the object in points (1/72 inches).

2.6 TRenderCell

2.6.1 TRenderCell Reference

Unit

[QPDFRender](#)

Description

The *TRenderCell* is a class representing a table cell in the PDF document. It contains all the methods and properties needed to manage table cells. The *TRenderCell* class is a [TRenderAtom](#) descendant.

See also:

[TRenderRow Class](#)

[TRenderTable Class](#)

2.6.2 Properties

▶ Run-time only

 Key properties

[BorderBottom](#)

[BorderLeft](#)

[BorderRight](#)

[BorderTop](#)

[Indent](#)

[Padding](#)

2.6.2.1 BorderBottom

property BorderBottom: **Integer**;

Description

The *BorderBottom* property defines a thickness of the bottom border of the cell in points (1/72 inches). To define a one-pixel thickness of the border, set this property to 0. Set it to -1 to make the border invisible. By default it is set to 0.

See also:

[BorderLeft](#)

[BorderRight](#)

[BorderTop](#)

2.6.2.2 BorderLeft

property BorderLeft: **Integer**;

Description

The *BorderLeft* property defines a thickness of the left border of the cell in points (1/72 inches). To define a one-pixel thickness of the border, set this property to 0. Set it to -1 to make the border invisible. By default it is set to 0.

See also:

[BorderBottom](#)

[BorderRight](#)

[BorderTop](#)

2.6.2.3 **BorderRight**

property BorderRight: **Integer**;

Description

The *BorderRight* property defines a thickness of the right border of the cell in points (1/72 inches). To define a one-pixel thickness of the border, set this property to 0. Set it to -1 to make the border invisible. By default it is set to 0.

See also:

[BorderBottom](#)

[BorderLeft](#)

[BorderTop](#)

2.6.2.4 BorderTop

property BorderTop: **Integer**;

Description

The *BorderTop* property defines a thickness of the top border of the cell in points (1/72 inches). To define a one-pixel thickness of the border, set this property to 0. Set it to -1 to make the border invisible. By default it is set to 0.

See also:

[BorderBottom](#)

[BorderLeft](#)

[BorderRight](#)

2.6.2.5 Indent

property Indent: **Integer**;

Description

The *Indent* property defines an indentation interval for the first row of the cell in points (1/72 inches). By default it is set to 0.

See also:

[BorderLeft](#)

[BorderRight](#)

[BorderTop](#)

2.6.2.6 Padding

property Padding: TRect;

Description

The *Padding* property defines cell padding intervals in points (1/72 inches). Use this property to customize left, right, top, and bottom padding intervals separately, or at once using the Rect function. If any of padding intervals is set to -1, then this interval would be inherited from the parent table.

See also:

[BorderLeft](#)

[BorderRight](#)

[BorderTop](#)

2.6.3 Methods

 Key methods

[AddItem](#)
[AddParagraph](#)
[AddText](#)
[AddTextItem](#)
[InternalWidth](#)

2.6.3.1 AddItem

procedure AddItem(X, Y: **Integer**; Item: [TRenderAtom](#));

Description

The *AddItem* method places the Item object to the current cell by specified coordinates. Use this method to place objects (tables, images, words, etc.) to a definite place in the cell.

Note: The cell becomes solid if any coordinate differs from 0. A solid cell cannot be split into pages, and no word wrapping is available.

See also:

[AddParagraph](#)

[AddText](#)

[AddTextItem](#)

[TRenderAtom Class](#)

2.6.3.2 AddParagraph

function AddParagraph: [TRenderCell](#);

Description

Use this method to add a paragraph to a cell. Technically, if the cell has no paragraphs, this method creates a table within the cell and returns the first cell in the first row of the created table. If the cell already has paragraphs, the method adds a row to the inner table and returns the first cell in its last row.

Note: If you want to add a mixed content to a cell, always use *AddParagraph* before each new object, i.e. if you want to add two tables and a text to the table cell, use the following code:

```
Cell.AddParagraph;  
Cell.AddTextItem(Table1);  
Cell.AddParagraph;  
Cell.AddText('Text', Font, TextAlignment);  
Cell.AddParagraph;  
Cell.AddTextItem(Table2);
```

If you will skip the first row, then the second *AddParagraph* will create a new row in the Table1 table and the document appearance will become incorrect. So be careful when using mixed content in a single cell.

See also:

[AddItem](#)

[AddText](#)

[AddTextItem](#)

2.6.3.3 AddText

procedure AddText(_Text: WideString; _Font: TFont; _Align: [TRenderTextAlign](#) = rtaLeft,

Description

Use this method to add a text to the last paragraph of the cell. The *_Font* parameter defines a text font, the *_Align* parameter defines a text alignment. The *_Link* parameter defines a target link for the added text. For more information about linking see the [TRenderAtom.LinkName](#) and the [TRenderAtom.LinkTarget](#) topics.

See also:

[AddItem](#)

[AddParagraph](#)

[AddTextItem](#)

[TRenderAtom Class](#)

[TRenderAtom.LinkName](#)

[TRenderAtom.LinkTarget](#)

[TRenderTextAlign Type](#)

2.6.3.4 AddTextItem

```
procedure AddTextItem(_Item: TRenderAtom);
```

Description

Use this method to add any *TRenderAtom* object (or its descendant) to the last paragraph of the cell as a text word. This object would be processed by the laying out system as a part of text.

See also:

[AddItem](#)

[AddParagraph](#)

[AddText](#)

[TRenderAtom Class](#)

2.6.3.5 InternalWidth

function InternalWidth: **Integer**;

Description

Use this function to collect an internal width of the cell - a width of a cell work area excluding padding intervals.

See also:

[Width](#)

2.7 TRenderImage

2.7.1 TRenderImage Reference

Unit

[QPDFRender](#)

Description

The *TRenderImage* class represents an image in a PDF document. Use it to add your images to documents. This class is a [TRenderAtom](#) descendant.

2.7.2 Methods

 Key methods

[Create](#)

2.7.2.1 Create

```
constructor Create(_graphic: TGraphic; _Raster: Boolean = false); overload;  
constructor Create(_Width: Integer; _graphic: TGraphic; _Raster: Boolean = false); overload;  
constructor Create(_Width, _Height: Integer; _graphic: TGraphic; _Raster: Boolean = false);
```

Description

Use this method to create new instances of the [TRenderImage](#) class. The *_graphic* parameter is a *TGraphic* object containing an image you want to add. If the *_Raster* parameter is set to *True*, then any JPEG or meta images would be converted to bitmap images. You can also define the image size with *_Width* and *_Height* parameters. If only the *_Width* parameter is set, then the image would be scaled automatically.

2.8 TRenderRow

2.8.1 TRenderRow Reference

Unit

[QPDFRender](#)

Description

The *TRenderRow* class represents a table row. This class is a collection of the [TRenderCell](#) objects. Use the [Cell](#) property to access these objects by their indices. This class is a [TRenderAtom](#) descendant.

See also:

[TRenderCell Class](#)

[TRenderTable Class](#)

2.8.2 Properties

▶ Run-time only  Key properties

▶ [Cell](#)

▶ [CellCount](#)

2.8.2.1 Cell

property Cell[Index: **Integer**]: [TRenderCell](#);

Description

Use the *Cell* property to access the row cells by *Index*. this property is read-only and accessible only at runtime.

See also:

[CellCount](#)

2.8.2.2 CellCount

property CellCount: **Integer**;

Description

This property returns an amount of elements in the [Cell](#) array. This property is read-only and accessible only at runtime.

See also:

[Cell](#)

2.9 TRenderSequence

2.9.1 TRenderSequence Reference

Unit

[QPDFRender](#)

Description

The *TRenderSequence* class represents a sequence of different objects ([TRenderAtom](#) descendants). This class can be used to place a sequence of objects with fixed width and height by defined coordinates. This class is a [TRenderAtom](#) descendant.

2.9.2 Methods

 Key methods

[Create](#)
[AddItem](#)
[AddText](#)
[NewLine](#)

2.9.2.1 AddItem

```
procedure AddItem(_Item: TRenderAtom);
```

Description

Use this method to add items to a sequence.

See also:

[AddText](#)

[NewLine](#)

2.9.2.2 AddText

```
procedure AddText(_Text: WideString; _Font: TFont; _Link: String = '');
```

Description

Use this method to add a text to a sequence. The *_Font* parameter defines a text font. The *_Link* parameter defines a target link for the added text. For more information about linking see the [TRenderAtom.LinkName](#) and the [TRenderAtom.LinkTarget](#) topics.

See also:

[AddItem](#)

[NewLine](#)

2.9.2.3 Create

constructor `Create(_Text: WideString; _Font: TFont; _Align: TRenderTextAlign = rtaLeft)`

Description

Use this method to create a new instance of the *TRenderSequence* class. The *_Text* parameter defines an initial text in the sequence; this parameter can be set to an empty string if you create a sequence of non-textual objects. The *_Font* parameter defines a text font, the *_Align* parameter defines a text alignment. The *_Link* parameter defines a target link for the added text. For more information about linking see the [TRenderAtom.LinkName](#) and the [TRenderAtom.LinkTarget](#) topics.

2.9.2.4 NewLine

procedure NewLine;

Description

This method inserts line break to a sequence.

See also:

[AddItem](#)

[AddText](#)

2.10 TRenderTable

2.10.1 TRenderTable Reference

Unit

[QPDFRender](#)

Description

The *TRenderTable* represents a table in the PDF document. This class contains all the methods and properties needed to manage table rows and cells.

2.10.2 Properties

▶ Run-time only

 Key properties

- ▶ [Cell](#)
- ▶ [RowCount](#)
- ▶ [Rows](#)
- ▶ [Border](#)
- ▶ [HasFooter](#)
- ▶ [HasHeader](#)
- ▶ [Padding](#)

2.10.2.1 Border

property Border: **Integer**;

Description

The *Border* property defines a thickness of the border for the table in points (1/72 inches). To define a one-pixel thickness of the border, set this property to 0. Set it to -1 to make the border invisible. By default it is set to 0.

See also:

[BorderLeft](#)

[BorderRight](#)

[BorderTop](#)

2.10.2.2 Cell

property Cell[x,y: **Integer**]: [TRenderCell](#);

Description

Use the *Cell* property to access table cells by their coordinates. This property is read-only and accessible only at runtime.

See also:

[Rows](#)

[TRenderCell](#)

2.10.2.3 HasFooter

property HasFooter: **Boolean**;

Description

If this property is set to *True*, then the first row of the table would be processed as a table footer. This row would be copied to each page where the table is located.

See also:

[HasFooter](#)

2.10.2.4 HasHeader

property HasHeader: **Boolean**;

Description

If this property is set to *True*, then the first row of the table would be processed as a table header. This row would be copied to each page where the table is situated.

See also:

[HasFooter](#)

2.10.2.5 Padding

property Padding: TRect;

Description

The *Padding* property defines padding intervals for the table in points (1/72 inches). Use this property to customize left, right, top, and bottom padding intervals separately, or at once using the Rect function. By default these parameters are set to 0.

See also:

[BorderLeft](#)

[BorderRight](#)

[BorderTop](#)

2.10.2.6 RowCount

property RowCount: **Integer**;

Description

This property returns an amount of items in the [Rows](#) array. This property is read-only and accessible only at runtime.

See also:

[Rows](#)

2.10.2.7 Rows

property Rows[Index: **Integer**]: [TRenderRow](#);

Description

Use this property to access the table rows by *Index*. The *Rows* property is read-only and accessible only at runtime.

See also:

[Cell](#)

[RowCount](#)

2.10.3 Methods

 Key methods

[Create](#)

[AddRow](#)

[LastRow](#)

2.10.3.1 Create

constructor Create(_Widths: **array** of **Integer**); **overload**;

Description

Use this method to create new instances of the [TRenderTable](#) class. The *_Width* parameter set widths of table columns in points (1/72 inches).

2.10.3.2 AddRow

function AddRow: [TRenderRow](#);

Description

This method adds a new row in the table and returns it as a result of execution.

See also:

[LastRow](#)

[TRenderRow](#)

2.10.3.3 LastRow

function LastRow: [TRenderRow](#);

Description

This method returns the last row of the table.

See also:

[AddRow](#)

[TRenderRow](#)

2.11 TRenderWord

2.11.1 TRenderWord Reference

Unit

[QPDFRender](#)

Description

The *TRenderWord* class represent a single word in a document. This word cannot be wrapped. Use this class to create unbreakable sentences or single words. This class is a [TRenderAtom](#) descendant.

2.11.2 Methods

 Key methods

[Create](#)

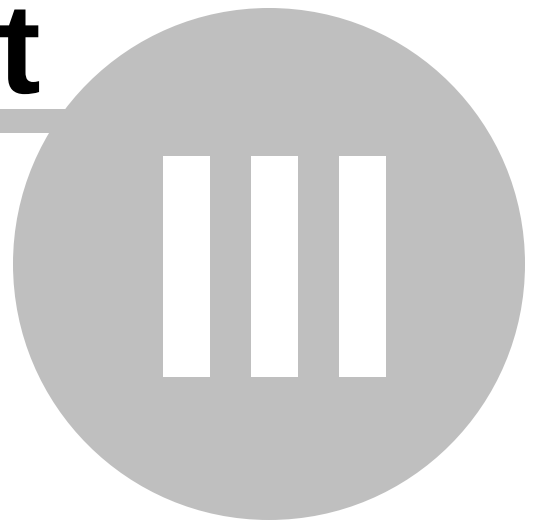
2.11.2.1 Create

constructor `Create(_Word: WideString; _Font: TFont);`

Description

Use this method to create a new instance on the [TRenderWord](#) class. The *_Word* parameter defines a text, and a *_Font* parameter defines a text font in the document.

Part



3 Units

3.1 QuickPdf unit

Components

[TQuickPDF](#)

Classes

[TQuickPDFDocument](#)

[TQuickPDFContentsItem](#)

3.2 QPDFRender unit

Classes

[**TRenderAtom**](#)

[TRenderCell](#)

[TRenderRow](#)

[TRenderTable](#)

[TRenderWord](#)

[TRenderSequence](#)

[TRenderImage](#)

Types

[**TRenderTextAlign**](#)

3.2.1 TRenderTextAlign type

Unit[QPDFRender](#)**Declaration**

```
type TRenderTextAlign = (rtaLeft, rtaRight, rtaCenter, rtaJustify, rtaTerminal);
```

Description

This type defines the text alignment. The following values are available:

Value	Alignment
<i>rtaLeft</i>	Text aligned to left
<i>rtRight</i>	Text aligned to right
<i>rtaCenter</i>	Centered text
<i>rtaJustify</i>	Justified text
<i>rtaTerminal</i>	Spaces are processed as symbols. Use this alignment type to print a space-formatted text

Part

IV

4 Appendix

4.1 LinkName and LinkTarget Example

This example displays the usage of the **LinkName** and the **LinkTarget** properties of the [TRenderAtom](#) class and its descendants. The following code creates a link from the 'Click Here' text to the 'Link Position' text in the document footer, and a link from the document footer to the *EMS Software Development* site.

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    LinkPosition: TRenderWord;  
begin  
    LinkPosition := TRenderWord.Create('Link Position', TFont.Create());  
    LinkPosition.LinkName := 'position';  
    QuickPDF1.Document.Footer.AddItem(0, 0, LinkPosition);  
    QuickPDF1.Document.Section[0].AddText('Click Here', TFont.Create(), rtaLeft, 'position');  
    QuickPDF1.Document.Footer.LinkTarget := 'http://www.sqlmanager.net/';  
    QuickPDF1.GenerateToFile;  
end;
```


Credits

Software Developers:

Dmitry Goldobin

Vadim Vinokur

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