

TMS W_{EB} Core
Web Application Development with Delphi
2nd Edition
Foreword by Bruno Fierens, TMS Software

Dr. Holger Flick

April 2023



Published by FlixEngineering LLC, Estero, FL, USA.

Copyright © 2023, Dr. Holger Flick, FlixEngineering LLC.

World rights reserved.

The source code examples shown here and the example database are fictitious and were created for this book as copyright protected freeware for use by the readers of this book. The information and source code examples contained or referenced in this book are provided "as is" without expressed, statutory, or implied warranties. Neither the author nor FlixEngineering LLC shall be liable in any manner whatsoever for damages caused or alleged to be caused directly or indirectly by this book or its contents, the source code examples and the sample database or other related online content and/or files for a particular purpose.

Copyright law prohibits you from republishing the content, source code, sample database and all accompanying online content in print, electronic or any other form without the prior and express written permission of the author. No part of this publication may be stored in a retrieval system, transmitted or reproduced in any form or by any means, in the original or translated language, including, but not limited to, photocopying, photography, digital, magnetic or other recording, without the prior and express written permission of the author.

ISBN-13: 979-8377953722

Author: Dr. Holger Flick

Technical Review: Bruno Fierens, Roman Kassebaum, Wagner Landgraf



Cover Design: Kristin Lee Flick

Typesetting System: pdfL^AT_EX (T_EXLive 2023)

Printed copies of this book are available for sale on Amazon through Kindle Direct Publishing (KDP). Any other download or sale is probably illegal. This is not a free book - do not distribute it.

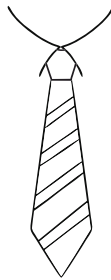
Further information and links to purchase this book can be found at:

<https://www.flixengineering.com/books>

Delphi is a trademark of Embarcadero Technologies.  Windows is a trademark of Microsoft Corporation. iPhone, iPad, macOS are trademarks of  Apple Inc. Other product and company names mentioned herein or in accompanying online materials may be the trademarks of their respective owners.

For my wife Kristin – I ♥ u.

'You either get better, or you don't progress.'
In memory of Lance Reddick.



Contents

1	Introduction	19
1.1	A mystery unveiled	19
1.2	Why Web Applications?	22
1.3	What is TMS WEB Core?	24
1.4	Target audience	29
1.5	Required skills	30
1.6	Downloading source code	30
1.7	Notation	31
1.7.1	Screenshots	31
1.7.2	Related videos	31
1.7.3	Source code	32
1.7.4	Keyboard	32
1.7.5	Examples and exercises	32
1.7.6	Development environment	33
1.8	Technical review	33
1.9	Author	37
2	Basics	39
2.1	System requirements	42
2.1.1	Web server	42
2.1.2	Web browser	43
2.2	Installing and uninstalling	44
2.2.1	Installing TMS WEB Core	45
2.2.2	Uninstalling TMS WEB Core	55
2.2.3	Installation of additional components	57
2.3	The first web application	58
2.3.1	Creating an application	59
2.3.2	Displaying time and date	65
2.3.3	Display the time with automatic updates	66
2.4	Debugging	67
2.4.1	Providing debug information	67

2.4.2	Using the web browser developer console	68
2.4.3	Define breakpoints in the IDE	70
2.4.4	Breakpoints in JavaScript	72
2.5	Settings	72
2.5.1	Global settings	73
2.5.2	Installed building blocks	78
2.5.3	Project-specific settings	78
2.5.4	Using the Delphi compiler (DCC)	80
2.6	Online documentation	81
2.6.1	Access from the IDE	81
2.6.2	Full-text search	81
2.6.3	Light and dark theme	83
2.7	Building blocks	83
2.8	Pascal to JavaScript Transpiler	87
2.8.1	Unsupported Delphi language features	87
2.8.2	Runtime library (RTL)	88
2.8.3	Preprocessor	90
2.8.4	asm/end: Importing JavaScript source code	90
2.8.5	Logging with the browser console	94
2.8.6	JavaScript Arrays: TJSArray	94
2.8.7	JavaScript Objects: TJSObject	97
2.8.8	Procedures and functions with variable number of parameters . . .	100
2.8.9	Custom Attributes	102
2.8.10	Generics	103
2.8.11	Asynchronous language extensions	106
2.8.12	Summary	113
2.9	Functionality in detail	114
3	Application Development	117
3.1	Creating applications	123
3.2	Controls for the design of web pages	129
3.3	Displays and HiDPI	134
3.4	Using the Application object	137
3.4.1	Creating forms	137
3.4.2	Navigating and routing	138
3.4.3	Default appearance	139
3.4.4	Downloading data	140
3.4.5	Information about the run-time environment	141
3.4.6	Running JavaScript code	144
3.4.7	Hint windows with Bootstrap	144
3.4.8	Events	145
3.5	Light and dark themes in your application	149
3.6	Right-to-left written languages (BiDiMode)	150
3.7	Designing web pages with forms	151
3.7.1	FormContainer	153
3.7.2	Creating forms at run-time	153

3.7.3	Replace current window with new window	155
3.7.4	Pop-up window	157
3.7.5	Creating forms using async/await	157
3.7.6	Creating pop-up forms with await	160
3.7.7	Embedding forms in other controls	160
3.7.8	Examples	161
3.7.9	Inheritance	169
3.7.10	Frames	173
3.8	Design options for forms	177
3.8.1	Remarks on WYSIWYG	178
3.8.2	Positioning	179
3.8.3	Design with CSS and Bootstrap	181
3.8.4	Design with HTML	193
3.8.5	RAD Studio: Preview of layout at run-time	206
3.8.6	Scaling	208
3.8.7	Alignment, groups, and anchors	209
3.9	Web Browser Cache: Forms not updating?!	217
3.10	Interaction between controls	218
3.10.1	Example	218
3.10.2	Exercise	222
3.11	Dialog boxes	223
3.11.1	Display messages with ShowMessage	223
3.11.2	Confirm or deny with 'Confirm'	224
3.11.3	Custom messages with MessageDlg	225
3.11.4	Non-visual component for MessageDlg	229
3.11.5	User input with InputBox and InputQuery	230
3.11.6	Async-variations for use with await	233
3.12	Base64 encoding and decoding	237
3.13	Data URLs	238
3.13.1	A practical example with icons	238
3.13.2	Constructing data URLs in code	239
3.13.3	Summary	240
3.14	Downloading data from the Web	240
3.14.1	Component support	241
3.14.2	HTTP verb	241
3.14.3	URL	242
3.14.4	Request Header	242
3.14.5	Basic access/user authentication	244
3.14.6	Timeout	245
3.14.7	Sending data	245
3.14.8	Specifying a ResponseType	245
3.14.9	Executing requests	247
3.14.10	Reading response data	250
3.14.11	Implementing a progress bar	253
3.14.12	Assigning data to an image control	254
3.14.13	Example	256

3.15	Using data modules	259
3.15.1	Creating data modules	259
3.15.2	Examples	260
3.16	Numbers & date and time values	263
3.16.1	Usage of standard components	264
3.16.2	Configuration of the web browser	266
3.16.3	Date settings	266
3.16.4	Currency floating point values	270
3.16.5	Separators and other symbols	271
3.16.6	Examples	271
3.16.7	Retrieving the localization of a specific region	272
3.17	WebTools: Utilities	272
3.18	Drawing in the web browser: TWebPaintBox	277
3.19	(HTTP) Cookies	279
3.20	Working with JSON	282
3.20.1	Serializing Pascal objects	283
3.20.2	Deserialization	286
3.20.3	Creating JSON with helper classes	287
3.20.4	Retrieving information from web services	289
3.21	Google Maps: Integrate maps into web applications	294
3.21.1	Application key	294
3.21.2	Map view	296
3.21.3	Map position and zoom	296
3.21.4	Adding markers	300
3.21.5	Drawing on the map	302
3.21.6	Displaying routes	305
3.21.7	Determine locations on the map (Geocoding)	309
3.21.8	Important events	310
3.21.9	Determine the current location	312
3.21.10	Using Map Themes	314
3.21.11	Example	316
3.22	Embedding YouTube content in websites	316
3.23	Unit Test Applications	319
3.23.1	Our scenario	319
3.23.2	Creating a new Unit Test application	321
3.23.3	Writing unit tests	321
3.23.4	Running the test application	326
3.24	Icons: Using vector paths	327
3.24.1	Find icons	327
3.24.2	Usage options	328
3.24.3	SVG tags	330
3.24.4	Summary	330
3.25	Working with HTML elements	332
3.25.1	Getting access to the DOM	332
3.25.2	Finding elements by their ID	333
3.25.3	Body of a document	334

3.25.4	Modifying elements in code	336
3.25.5	Adding events	337
3.25.6	Dynamic creation of a navigation bar	339
3.25.7	Essential classes in detail	345
4	Web-, Mobile- and Desktop Applications	353
4.1	Progressive Web Applications	355
4.1.1	Creating a PWA	355
4.1.2	Project Options	357
4.1.3	Behavior on mobile devices	358
4.1.4	Note on local databases	359
4.1.5	Querying the online status	359
4.1.6	Example application	361
4.2	Electron	364
4.2.1	Creating an application	366
4.2.2	Target platforms	367
4.2.3	Project options	371
4.2.4	Forms	372
4.2.5	Drag & Drop	384
4.2.6	Fonts	390
4.2.7	Utility functions	391
4.2.8	Accessing the clipboard	392
4.2.9	Accessing the operating system	393
4.2.10	Information about the local file system	394
4.2.11	Reading and writing data	395
4.2.12	Accessing the current window/form	396
4.2.13	Dialog boxes for opening and saving	397
4.2.14	Standard dialogs for user interaction	400
4.2.15	Menus	403
4.2.16	Defining a tray icon	404
4.2.17	Monitoring the local file system	405
4.3	TMS Miletus	406
4.3.1	Supported Platforms	407
4.3.2	Miletus Packager (miletus.org)	409
4.3.3	Creating an application	410
4.3.4	Working with multiple forms	413
4.3.5	Developer Console in debug and release mode	416
4.3.6	Migrating an existing application	416
4.4	Responsive components	418
4.4.1	TWebResponsiveGrid	419
4.4.2	TWebResponsiveGridPanel	425

5	Databases and web services	433
5.1	TWebDataset: A TClientDataSet for web applications	435
5.1.1	Using TWebClientConnection	435
5.1.2	Creating fields during design time	439
5.1.3	Using TXDataWebConnection	441
5.2	Local Databases	441
5.2.1	TWebLocalStorage: Permanent local data storage	442
5.2.2	TWebSessionStorage: Local, transient data storage	443
5.2.3	IndexedDB: Local NoSQL-Database	444
5.2.4	IndexedDB: Image data	450
5.3	Database providers	456
6	Hands-on: Comprehensive example	459
6.1	Example 1: Web service	462
6.1.1	Creating the application	462
6.1.2	Building the user interface	464
6.1.3	Requesting data from the web service	465
6.1.4	Displaying data in a table	472
6.1.5	Displaying the date	476
6.1.6	Pointing out an error state	477
6.1.7	Adding local storage and query string	477
6.1.8	Improve keyboard support	478
6.2	Example 2: Using TWebSocketClient	479
6.2.1	User interface changes	479
6.2.2	Adding TWebSocketClient	480
6.2.3	Updating the button appearance	482
6.2.4	Subscribing to a symbol	483
6.2.5	Receiving data	484
6.2.6	Display data in a table	486
6.3	Example 3: Transfer the WebSocket app to the desktop	491
7	Multi-tier with XData	497
7.1	Notes on the database	500
7.2	Creating the web service	503
7.2.1	Creating the application	504
7.2.2	Supported databases	509
7.2.3	Connecting to the database	513
7.2.4	Importing database tables	514
7.2.5	Starting and stopping the web service	518
7.2.6	Testing the service	518
7.2.7	XData in a nutshell	521
7.2.8	Testing with Swagger UI	525
7.2.9	Modifications to the main form	526
7.3	Retrieve information with a web application	527
7.4	Map markers from a database	530
7.4.1	Basic application	531

7.4.2	Web Design with Bootstrap	533
7.5	Adding special functions to the web service	538
7.5.1	Defining a service	540
7.5.2	Retrieving data from a service method	544
7.5.3	Implementation of the web client	545
7.5.4	User interaction: Retrieving detailed information	547
7.5.5	Design with Bootstrap	551
7.6	Building an XData server for Linux	552
7.6.1	Creating a console application	554
7.6.2	A custom server container class	554
7.6.3	Looking at the TUri data type	555
7.6.4	Implementing the Singleton pattern	557
7.6.5	Creating the HTTP dispatcher	558
7.6.6	Starting the XData server	558
7.6.7	Stopping the XData server	560
7.6.8	Implementing the service	561
7.6.9	Adding user interaction	564
7.6.10	Preparing Linux	571
7.6.11	Selecting Linux	571
7.6.12	Hints on PAServer	571
7.6.13	Starting the services	572
7.6.14	Summary	572
8	Deploying your TMS WEB Core application	575
8.1	Common steps	576
8.2	Windows: Internet Information Services (IIS)	580
8.3	Linux: Nginx	585
8.4	Provider: Ionos/1&1	588
8.5	HTTPS: Secure data transmission and 'those darn' certificates	590
8.6	Summary	592
9	Docker: Making deployment fun	593
9.1	Installing Docker	597
9.1.1	■ Microsoft Windows (10 and 11)	597
9.1.2	🍏 Apple macOS	614
9.2	Images and Containers	618
9.2.1	Downloading a Docker Image	619
9.2.2	Getting help	620
9.2.3	Creating a container	621
9.2.4	Inspecting containers	624
9.2.5	Managing containers in Docker Desktop	625
9.2.6	Mapping a port and starting in the background	626
9.2.7	Naming a container	629
9.2.8	Starting and stopping containers	631
9.2.9	Removing a container	631
9.2.10	Finding Docker images and getting to know them	632

9.2.11	File system of containers	636
9.3	Deploying a web application to a container	637
9.3.1	Creating a new web application	637
9.3.2	Creating a container to host the application	644
9.3.3	Building a custom image	646
9.3.4	Exporting an image as an archive	650
9.3.5	Importing an image from an archive	651
9.4	Database servers in Docker	652
9.4.1	Using client libraries with FireDAC	654
9.4.2	MariaDB in a container	655
9.4.3	Introducing Docker Compose	656
9.4.4	MariaDB environment variables	668
9.4.5	Loading values from a file	670
9.4.6	Executing scripts to import data	670
9.4.7	Using a Dockerfile in Docker Compose	671
9.4.8	Mapping a folder	673
9.4.9	Remarks on initializing containers	675
9.4.10	Using a custom configuration file	675
9.4.11	MySQL	676
9.4.12	Note about ARM processors	677
9.4.13	Summary	678
9.5	Building a multi-tier Docker application	678
9.5.1	Description of the application	679
9.5.2	Creating a multi-tier Docker application	688
9.5.3	Summary	692
10	Appendix	695
10.1	References	696
10.2	Example data	697
10.2.1	cars.json	697
10.2.2	states.json	698
10.2.3	Schools in Lee County, FL	699
10.2.4	Points of Interest in Lee County, FL	699

Preface



At the time Dr. Holger Flick was writing the very first TMS WEB Core book, the TMS WEB Core product was at version *v1.4*. The first edition of the book helped numerous Delphi developers to make the first steps into web client application development with Object Pascal from the Delphi IDE. The inputs and feedback from Holger as well as so many developers using TMS WEB Core, helped shape the product to what it is today with the *v2.1* release just behind us. A lot has happened between TMS WEB Core *v1.4* and the release *v2.1*. In numerous ways, the product evolved, among which:

- Delphi language compatibility substantially increased via the *pas2js* transpiler and its RTL
- TMS WEB Core framework was significantly extended with new components and new features
- Support for easier asynchronous programming via promises
- More browser APIs and DOM access was exposed via the framework via easy-to-use Object Pascal classes
- New Delphi IDE integrations targeted at increasing productivity, especially in the area of using HTML templates or JavaScript libraries
- A new framework Miletus was added to offer lightweight cross-platform desktop application development
- TMS WEB Core can now also be used from the free & cross-platform Visual Studio Code IDE

So many improvements in every area of the product cried for a new edition of the book. You now hold in your hands the result of this enormous work from Dr. Holger Flick. No part of TMS WEB Core is left uncovered in this new book and its sheer size immediately tells it. And that is not all. The book also goes further on several topics related to web development in general. There is a new chapter on deployment and introduces how you can use Docker in connection with TMS WEB Core web apps for easy deployment.

I'm truly happy, thankful and delighted there is now a new and up-to-date resource to take the Delphi developer by the hand and introduce the many aspects of web client application development step by step by means of a book. I still strongly believe in the concept and added value of a book written by a knowledgeable author.

It allows to bring content gradually in well thought out chapters so its readers can really understand and learn the whole process of creating web client applications with Object Pascal. It is a completely different process from just starting to work with a project and jump almost randomly from one online component reference page to another. With the book from Holger, you, the reader gets the full story, the ideas behind it and the why and how of development techniques possible with TMS WEB Core. This way, I'm sure you'll be productive quicker, write more solid code easier and create quality web client apps faster.

It is my hope and wish that the new TMS WEB Core book will help making the learning curve for TMS WEB Core less steep and enjoyable at the same time. I hope it will spin off a new and large group of Delphi developers no longer afraid to choose a web development project and take advantage of the many benefits it has: no-deploy, cross-platform, one code-base, no software update hassles, usable from almost any device on the planet, reusing easy-to-get existing HTML templates to offer beautiful UIs and much more...

Finally, I want to thank all people involved in the TMS WEB Core project:

- **Mattias Gaertner & Michael Van Canneyt** responsible for the pas2js transpiler & RTL
- **Roman Kassebaum** for Delphi IDE integration
- **José Leon Serna** for Visual Studio Code integration
- **Pieter Scheldeman** for TMS WEB Core support for the entire TMS FNC component suite
- **Tünde Keller & Bart Holvoet** for several TMS WEB Core framework components
- **Wagner Landgraf** for creating the XData backend integration
- **Holger Flick** for QA, brainstorming, documentation, videos, consulting, and also this book

This book is a cherry on the cake for all the hard work that went into TMS WEB Core v2.1. It is at the same time an inspiring milestone to continue the road to many more versions!

Bruno Fierens

Wevelgem, Belgium
April 2023

It's been a loooong time!



A while ago, I looked at pictures of me presenting in front of Delphi developers. Due to recent world events, the last presentation was at the end of November 2019 in Düsseldorf, Germany. If you compare it to recent pictures of me, my hairline and simple arithmetic will yield that it has been a long time; almost 4 years ago when you read this!

Writing this, TMS and I are in the process of planning an in-person event again. Finally! We will be able to show the magic that goes on in the TMS world every single day.

Before reading this book, you will certainly browse the content first. You will notice a lot of passages that are marked with the *New* icon you see here. All content that has been added or changed with regard to the first edition of this book has been highlighted that way. This makes it accessible for existing readers to glimpse at the new content.



Believe me, it has not been an easy decision whether to update or add a second book with just the new features. After grinding our brains for multiple days, we decided to update the existing book. To be more precise, this book does not only come with new content that has been added, a large chunk of the text has been rephrased to be current with the latest release. This means you have the complete foundation for software development with TMS WEB Core in one book. You might even say 'big' book now. I am quite sure though, the added weight is worth it! A lot of things have happened in the world of TMS WEB Core. Even though a lot of things have been at a standstill the last couple of years, TMS were definitely not. To me personally, these are the two biggest additions to TMS WEB Core:

- The ability to create cross-platform desktop applications for Microsoft Windows, Apple macOS, Linux, and the Raspberry Pi using a new technology called **TMS Miletus**. From a single IDE, based on one web application, you are able to create binaries for all the platforms. No additional hardware is required on your part to create the versions for the target system.
- **Asynchronous programming** has become much easier using the newly introduced function *await*. In other programming languages *await* has become a keyword as part of parallel programming. TMS were only able to add it as a function to a language that does not support this concept. Not only new components make TMS WEB Core attractive. The team's continuously ongoing dedication to enhance the Object Pascal language to make development



FIGURE 1: *Presenting at the TMS Days in Düsseldorf, Germany on November 15, 2019. It was the last in-person, multi-day TMS event before the COVID-19 pandemic. The next event after that one took place on May 11 and 12, 2023, in Bruges, Belgium.*

of web applications easier is also key to make the latest versions of TMS WEB Core better and better (→ section 2.8.11, on page 106).

But those are just two aspects. As I did not have to travel to present in person, I had more time to be creative and create new exciting hands-on examples. There have been lots of video tutorials and even multi-hour video courses in the meantime. Still, there was plenty of time to write down my thoughts about the new versions of TMS WEB Core which resulted in this very book you hold in your hands now. Thus, the last chapters presenting examples have been filled with significantly more content and hands-on use cases. Even deployment using Linux and Docker containers is covered now.

Hopefully, you are just as excited as I was learning about all the new things that are TMS WEB Core. Maybe, we will even meet during the event that we are planning right now... best of all: it will be in person!

A handwritten signature in black ink that reads "Holger Flick". The script is fluid and cursive, with the first name "Holger" and the last name "Flick" written in a single continuous line.

Estero, USA
April 2023

Preface from the first edition, March 2020

More than 15 years ago, at an EKON Delphi conference in Mörfelden, nearby Frankfurt in Germany, I met Holger Flick for the first time. The interest and curiosity of Holger Flick with respect to Delphi software development immediately struck me. It goes without saying we talked for hours and hours about Delphi, the language, the components, its ecosystem...

During all these years, we stayed in contact and it was meanwhile almost 3 years ago that our conversations led to a more intense collaboration. It was around the time the first internal TMS WEB Core versions saw the light. Seeing the first experimental versions of TMS WEB Core, Holger was immediately hooked and saw its huge potential. Since its architecture is based on the single-page application (SPA) model and thus radically different from former Delphi single-stack web solutions, we both had the vision that a lot of education and evangelism would be highly desirable to make Delphi developers efficiently productive with the new tool. And thus the idea was born to produce videos for TMS WEB Core and write blog articles. The idea worked so well that meanwhile many areas of our component products are covered by videos and blog articles written by Holger. But the role of Holger Flick at tmssoftware.com is much more than just producing video content or writing articles and now writing this book. Maybe even more important than the attention to detail with the production of videos itself, two key characteristics of Holger play an instrumental role in the development of TMS WEB Core and meanwhile many other things we do here on a daily basis. Holger is foremost curious and critical. Holger not only wants to know how something works but also why it was made to work the way it works. And he is very critical and will always raise questions if the way it works is really the best and most logical and intuitive way. Either Holger will learn and appreciate how something works or he will send the engineers back to the drawing board.

It is exactly this attitude that steered the development of TMS WEB Core itself as well as the writing of the book you are holding in your hands. It is quite unique to read a book written by someone who not only deeply and thoroughly understands the topic, but who also played and continues to play a crucial role in shaping this product.

It goes without saying that I'm truly blessed to have the luck and the opportunity to work together with Holger. It is amazing how well our visions align. Time and time again, Holger comes with ideas where I say, sure, this is exactly what we need to do. Time and time again Holger comes with critical remarks where I say, sure, this is exactly what we need to fix or improve. Quite some time ago, Holger suggested that TMS WEB Core deserved a book and sure, Holger was so right.

And now you are holding this result in your hands and are about to embark on a journey reading and discovering a wealth of new possibilities in new territories for Delphi developers. I wish you as much fun reading and discovering new capabilities & directions for your software applications with TMS WEB Core as Holger and I had together during the past years shaping the product!

- Bruno Fierens, December 2019