BLAISE PASCALCMAGAZINE

Multi platform /Object Pascal / Internet / JavaScript / WebAssembly / Pas2Js / Databases / CSS Styles / Progressive Web Apps Android / IOS / Mac / Windows & Linux

W.M

BLAISE PASCAL MAGAZINE

Multi platform /Object Pascal / Internet / JavaScript / WebAssembly / Pas2Js / Databases / CSS Styles / Progressive Web Apps Android / IOS / Mac / Windows & Linux

CONTENT

ARTICLES

Cartoons from our Technical Advisor	Page 5
By Jerry King	24.2 /
New Books: Hands on with Delphi - a Preview	Page 80
Image detection with Pascal and artificial intelligence	Page 7
By Max Kleiner	_
Speaking Sports Clock	Page 63
By Detlef Overbeek	_
GIT continued: Branching and partial commits	
for Lazarus and Delphi	Page 36
By Michael van Canneyt	
Testing IntraWeb Applications with Delphi	Page 145
By Bob Swart	
Skia for Delphi	Page 30
By Paulo César Botelho Barbosa	
Creating a new Library Program with PDF viewer	Page 157
By Detlef Overbeek	
Introduction to programming the internet with PAS2S	Page 98
By Michael van Canneyt	
Delphi for Raspberry	Page 55
By Bruno Fierens	
Last part of the Webservice (5), Deploy to Apache	Page 124
By Danny Wind	
PDF document in a report – using of new TfrxPDFView object	Page 114
By Den Zubow	
Scaling to a Very Large AI (Artificial Intelligence)	
causes unprecedented insights	Page 169

Back ground by Jim Bahn - Sherman Tree, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=10534735

ADVERTISERS

Barnsten	Page 35
Components4Developers	Page 168/175/176
Delphi Company	Page 123
Fast Report	Page 144
Lazarus Handbook Hardcover Subscription Combi	Page 112
NEW Library Libstick (app)	Page 29/113
Pocket Lazarus Handbook Subscription Combi	Page 62
Pocket Lazarus Handbook	Page 53/54
Superpack	Page 61
TMS	Page 6



Pascal is an imperative and procedural programming language, which Niklaus Wirth designed (left below) in 1968–69 and published in 1970, as a small, efficient language intended to encourage good programming practices using structured programming and data structuring. A derivative known as Object Pascal designed for object-oriented programming was developed in 1985. The language name was chosen to honour the Mathematician, Inventor of the first calculator: Blaise Pascal (see top right).

Niklaus Wirth

Publisher: PRO PASCAL FOUNDATION in collaboration © Stichting Ondersteuning Programmeertaal Pascal - Netherlands



Contributors

Stephen Ball http://delphiaball.co.uk @DelphiABall

David Dirkse www.davdata.nl E-mail: David @ davdata.nl

Holger Flick holger @ flixments.com

Primož Gabrijelčič primoz @ gabrijelcic.org

Max Kleiner www.softwareschule.ch max @ kleiner.com

Vsevolod Leonov vsevolod.leonov@mail.ru

Page

Boian Mitov mitov @ mitov.com

Detlef Overbeek - Editor in Chief www.blaisepascal.eu editor @ blaisepascal.eu

Wim Van Ingen Schenau -Editor wisone @ xs4all.nl

Bob Swart www.eBob42.com Bob @ eBob42.com

Anton Vogelaar ajv @ vogelaar-electronics.com Dmitry Boyarintsev dmitry.living @ gmail.com

Michaël Van Canneyt, michael @ freepascal.org

Benno Evers b.evers @ everscustomtechnology.nl

Mattias Gärtner nc-gaertnma@netcologne.de

John Kuiper john_kuiper @ kpnmail.nl

Paul Nauta PLM Solution Architect CyberNautics paul.nauta @ cybernautics.nl

Howard Page Clark hdpc @ talktalk.net

Peter van der Sman sman @ prisman.nl

B.J. Rao contact @ intricad.com

Danny Wind dwind @ delphicompany.nl Marco Cantù www.marcocantu.com marco.cantu @ gmail.com

Bruno Fierens www.tmssoftware.com bruno.fierens @ tmssoftware.com

Wagner R. Landgraf wagner @ tmssoftware.com

Andrea Magni www.andreamagni.eu andrea.magni @ gmail.com www.andreamagni.eu/wp

Kim Madsen www.component4developers.com

Jeremy North jeremy.north @ gmail.com

Heiko Rompel info @ rompelsoft.de

Rik Smit rik @ blaisepascal.eu

Daniele Teti www.danieleteti.it d.teti @ bittime.it

Siegfried Zuhr siegfried @ zuhr.nl

Ω

donator of WIKIPEDIA

Editor - in - chief

Detlef D. Overbeek, Netherlands Tel.: Mobile: +31 (0)6 21.23.62.68 News and Press Releases email only to editor@blaisepascal.eu

Editors

Correctors

Peter Bijlsma, W. (Wim) van Ingen Schenau, Rik Smit Howard Page-Clark, Peter Bijlsma

Trademarks All trademarks used are acknowledged as the property of their respective owners. Caveat Whilst we endeavour to ensure that what is published in the magazine is correct, we cannot accept responsibility for any errors or omissions.

If you notice something which may be incorrect, please contact the Editor and we will publish a correction where relevant.

Subscriptions (2019 prices)	Internat. excl. VAT	Internat. incl. 9% VAT	Shipment	
Printed Issue ±60 pages	€ 155,96	€ 250	€ 80,00	
Electronic Download Issue 60 pages	€ 64,20	€ 70		Member and
Printed Issue inside Holland (Netherlands) 60 pages		€ 250,00	€ 70,00	Member of th

Subscriptions can be taken out online at www.blaisepascal.eu or by written order, or by sending an email to office@blaisepascal.eu Subscriptions can start at any date. All issues published in the calendar year of the subscription will be sent as well. Subscriptions run 365 days. Subscriptions will not be prolonged without notice. Receipt of payment will be sent by email.

Subscriptions can be paid by sending the payment to: ABN AMRO Bank Account no. 44 19 60 863 or by credit card or Paypal

Name: Pro Pascal Foundation-Foundation for Supporting the Pascal Programming Language (Stichting Ondersteuning Programeertaal Pascal) IBAN: NL82 ABNA 0441960863 BIC ABNANL2A VAT no.: 81 42 54 147 (Stichting Programmeertaal Pascal)

Subscription department

Edelstenenbaan 21 / 3402 XA IJsselstein, The Netherlands

Mobile: + 31 (0) 6 21.23.62.68 office@blaisepascal.eu

Copyright notice

All material published in Blaise Pascal is copyright © SOPP Stichting Ondersteuning Programeertaal Pascal unless otherwise noted and may not be copied, distributed or republished without written permission. Authors agree that code associated with their articles will be made available to subscribers after publication by placing it on the website of the PGG for download, and that articles and code will be placed on distributable data storage media. Use of program listings by subscribers for research and study purposes is allowed, but not for commercial purposes. Commercial use of program listings and code is prohibited without the written permission of the author.



From your editor

Dear Readers,

thank you for making it possible for me to create and design the Magazine in an ongoing series up to issue number hundred. In the latest issue I changed the Layout and I hope you like that.

Please let me know.

In this issue we have a lot surprises for you: large offers for discounts, free version kbmMW community edition and almost each article has a code project which of course is available for you if you are a subscriber. I myself like the Speaking-Sports-Clock the most and use it everyday.

I also created a new version of the library stick that contains all issues and code which was published over time; unfortunately that one is not for free. It contains 100 Issues, 5809 Pages, 873 Articles, 8 Gig of Code and covers an amount of 10 Gig of space.

This issue has 176 pages and will be made available for free for anyone. (except the code – that is only for subscribers).

Now we have achieved some of our plans, I can tell you what is going on for the next year:

We (the Free Pascal and Lazarus team) want to bring Lazarus to the next level:

a Lazarus Web App that will be available in any place on the Web. This is a huge step and it will take a few years and we will do some fundraising.

Details will be made available in the next issue in January 2022.

We also need to finalize some coding: the Extended RTTI, Anonymous functions and WebAssembly. But I think that all will be very soon solved because most of it is in testing phase.

I have one wish and I will start to make it work asap: an interactive page where our readers can communicate and write there opinions. (must be about Pascal of course).

If you have wishes or comments please let me know: editor @ blaisepascalmagazine.eu

Have fun reading or using/recreating the code...

Detlef



From our Technical advisor: Cartoons from Jerry King



"Can you send help? Someone is buried under 100 issues of Blaise Pascal Magazine. He's still alive and holding a glass of champaign."



tmssoftware;com

Accelerate your development for Windows, Web, iOS, Andriod, macOS & Linux

develop • faster





Coupon code: BLAISE100 Valid till Nov 30, 2021

Valid for new licenses of all products!

jupyter colab

AUTHOR: MAX KLEINER Try finally begin. - Max

With the

following report I show how to

host and execute a deep learning project on a cloud. The cloud is hosted by google colab and enables working and testing in teams. Lazarus is also being built in colab and the deep learning network is compiled and trained

too in a Jupyter notebook.

https://gitlab.ti.bfh.ch/knm4/python4delphi/-/blob/master/ EKON24_SimpleImageClassificationCPU_2021.ipynb

So what's Colab? With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code.

Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine. We at BFH also use this service.

The Bern University of Applied Sciences (BFH) is one of the leading application-oriented universities in Switzerland.

With 31 Bachelor's and 25 Master's courses,

we offer a wide range of training and further education.

🔽 Webmail :: Welco× 🔍 🦗 so	oftwareschule.c X 🚦 GitHub import d X 🖸 maxkleiner/pythc X 📑 EKON24_Simpled X 📲 Readme - Ssh - H X 💽 maxkox maxbo X +	ſ
$\leftarrow \rightarrow$ C \textcircled{o} O	D A https://gitlab.ti.bfh.ch/knm4/python4delphi/-/blob/master/EKON24_SimpleImage(☆ ୧ google colab → 🛛 🛃 📗	=
🖁 Projects 🛩 Groups 🛩 🛛	More 🗸 🖸 🗸 Search or jump to Q D 11 v 🗹 💇 v	î
P python4delphi	Kleiner Helmut Max > python4delphi > Repository	
1 Project overview	master \checkmark python4delphi / EKON24_SimpleImageClassificationCPU_2 Find file Blame History Permalink	
Repository	Created using Colaboratory	
Files	Max Kleiner authored 5 days ago	
Commits		
Branches	EKON24_SimpleImageClassificationCPU_2021.ipynb 🏠 647	
Tags	кв 🛱 🔁 💆	
Contributors	CO Open in Colab	
Graph	In [1]:	
Compare	:apt-get upuate	
Locked Files	Out [1]:	
D in a	0% [Working]	
Dy Issues 0	Ign:1 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 InRelease	
Ξ ✓ Requirements	<pre>0% [Connecting to archive.ubuntu.com] [Connecting to security.ubuntu.com] [Wait</pre>	
≪ Collapse sidebar	Get:2 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,626 B]	~





Page 1/22

USING JUPYTER NOTEBOOKS AND GOOGLE COL

From Divya Singh
https://www.datasciencecentral.com/profile/DivyaSingh456

Let's start with an explanation of **Jupyter notebooks** and **Google colab**. We try to wite code in the notebooks and focus on the basic features of notebooks. Before diving directly into writing code, let us familiarise ourselves with writing the code notebook style!

WHAT ARE JUPYTER NOTEBOOKS?

- The **Jupyter Notebook** is an **open source web application** that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.
- **Jupyter Notebooks** are a spin-off project from the **IPython** project, which used to have an **IPython Notebook** project itself.
- The name, **Jupyter**, comes from the core supported programming languages that it supports: Julia, Python, and R.
- **Jupyter** ships with the **IPYTHON** kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.
- Jupyter is a project aiming to standardize interactive computing in any programming languages. The kernel provides interactive environment that executes user code as a server, connected with a frontend through sockets. (A network socket is a software structure within a network node of a computer network that serves as an endpoint for sending and receiving data across the network).
- A Jupyter Notebook is also an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and describing text.

Interactive notebooks are experiencing a rise in popularity. How do we know?

They're replacing PowerPoint in presentations, shared around organizations, and they're even taking workload away from BI (*1) suites<u>, Jupyter, R Markdown</u>. Apache Zeppelin, Spark Notebook and more. There are kernels/backends to multiple languages, such as Python, Julia, Scala, SQL, and others. Notebooks are typically used by data scientists for quick exploration tasks.

B *1 (Microsoft Business Intelligence (BI) is a suite of products and tools that you can use to monitor, analyze and plan your business by using scorecards, dashboards, management reporting and analytics. It contains the following tools: SQL Server Analysis Services (SSAS) SQL Server Integration Services (SSIS)). Today there are many notebooks to choose from)







Page 2/22



THE NOTEBOOK WAY

Traditionally, notebooks have been used to document research and make results reproducible, simply by rerunning the notebook on source data. But why would one want to choose to use a notebook instead of a favorite IDE or command line?

There are many limitations in the current browser-based notebook implementations, but what they do offer is an **environment for exploration, collaboration, and visualization.** Notebooks are typically used by data scientists for quick exploration tasks. In that regard, they offer a number of advantages over any local scripts or tools. Notebooks also tend to be set up in a cluster environment, allowing the data scientist to take advantage of computational resources beyond what is available on her laptop, and operate on the full data set without having to download a local copy.

Why Jupyter Notebooks

Jupyter notebooks are particularly useful as scientific lab books when you are doing computational physics and/or lots of data analysis using computational tools. This is because, with Jupyter notebooks, you can:

- Record the code you write in a notebook as you manipulate your data. This is useful to remember what you've done, repeat it if necessary, etc.
- Graphs and other figures are rendered directly in the notebook so there's no more printing to paper, cutting and pasting as you would have with paper notebooks or copying and pasting as you would have with other electronic notebooks.
 - You can have dynamic data visualizations, e.g. animations, which is simply not possible with a paper lab book.
 - One can update the notebook (or parts thereof) with new data by re-running cells.
 - You could also copy the cell and re-run the copy only if you want to retain a record of the previous attempt.

Google Colab

Colaboratory is a free Jupyter

notebook environment that requires no setup and runs entirely in the cloud. With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser. Colab let's you import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code.

Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine.

Why Google Colab

As the name suggests, Google Colab comes with collaboration backed in the product. In fact, it is a Jupyter notebook that leverages Google Docs collaboration features. It also runs on Google servers and you don't need to install anything.

Moreover, the notebooks are saved to your Google Drive account.





Page 3/22



TPU

Tensor Processing Unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning, particularly using Google's own TensorFlow software. https://en.wikipedia.org/wiki/Tensor_Processing_Unit

LAZBUILD Lazbuild is a



command line utility to compile Lazarus projects and packages, as well as the Lazarus IDE itself. When you built Lazarus yourself you can find the lazbuild executable in the Lazarus source directory together with the lazarus executable. When you find an error like "/bin/bash: lazbuild: command not found" then you missed lapt-get install fpc fpc-source lazarus git subversion

Direct Link to start:

https://colab.research.google.com/github/maxkleiner/python4delphi/blob/master/ EKON24_SimpleImageClassificationCPU_2021.ipynb

/bin/bash: lazbuild: command not found

function GetNumberOfProcessors: longint;
var
SystemInfo: TSystemInfo;
begin
GetSystemInfo(SystemInfo);
Result:= SystemInfo.dwNumberOfProcessors;
end;

In Delphi, you print via the TPrinter object.

- * Add printers to your uses clause
- * Use the Printer function to access the global instance of Tprinter
- * Printer.BeginDoc starts the print job
- * Printer.EndDoc stops the print job and sends it to the printer
- * Printer.NewPage forces a new page
- * Printer.Canvas is used to generate the output page





IMAGE DETECTION WITH LAZARUS Page 5/22 maXbox Starter 87 co ab jupyter 🔽 Webmail :: Welcome 🗴 🛛 💽 maXbox | maxbox 🗴 👷 mXScriptasHTML 🛛 😵 softwareschule.ch/d: X 🖪 (14) (PDF) maXbox_Si X 👿 Tensor Processing U. X CO EKON24_SimpleImag × → C A 🔿 🗛 🔤 https://colab.research.google.com/github/maxkleiner/python4delphi/blob/master/EKON: 🏠 🛛 🤻 lazbuild \rightarrow \bigtriangledown \mathbf{F} CEKON24 SimpleImageClassificationCPU.ipynb GD Share ۵ File Edit View Insert Runtime Tools Help Cannot save changes RAM + Code + Text A Copy to Drive • 🖍 Editina 🛛 🔨 × ⊟ Files Disk 🔳 V 🗢 🏟 🗔 📋 🗄 I apt-get update A Lc 🔺 0 Ign:1 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 InRK Get:2 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,626_B] InRelease 8 bin Ign:3 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 InRelease $\langle \cdot \rangle$ boot Hit:4 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86 64 Release Hit:5 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86 64 Release content Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB] Get:7 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ Packages [67.4 kB] Get:8 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease [15.9 kB] 🖿 datalab

Hit:9 http://archive.ubuntu.com/ubuntu bionic InRelease Get:11 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]

Hit:15 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease

Hit:13 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease Get:14 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]

Hit:17 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease

Get:16 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [581 kB]

Get:18 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1,428 kB] Get:19 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main Sources [1,802 kB]

Get:20 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2,335 kB] Get:21 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2,202 kB]

Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [613 kB] Get:23 http://pa.launchpad.net/c2d4u.team/c2d4u.0+/ubuntu bionic/main amd64 Packages [922 kB] Get:24 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2,770 kB] Fetched 13.0 MB in 4s (3,425 kB/s)

[] !apt-get install fpc fpc-source lazarus git subversion 64.15 GB available

Reading package lists... Done

💟 Webmail :: Welcome 🗶 💽 maXbox maxbo	r 🗙 🕵 mXScriptasHTML X 🧟 softwareschule.ch/dc× 🖪 (14) (PDF) maXbox_Si× 🐨 Tensor Processing UrX 🚥 EKON24_Simple	ielmag × + ×
$\leftarrow \rightarrow$ C \bigcirc O A :	² https://colab.research.google.com/github/maxkleiner/python4delphi/blob/master/EKON. ☆ 🭳 lazbuild	\rightarrow \bigtriangledown \checkmark \parallel
CO CEKON24_SimpleImageC File Edit View Insert Runtime	lassificationCPU.ipynb Tools Help <u>Cannot save changes</u>	🖙 Share 🏚 🌍
\equiv Files $ imes$	+ Code + Text & Copy to Drive V RAM Disk	Editing 🔨
o 🗗 🗔 🖪	lapt-get update	
 bin boot content sample_data README.md anscombe.json california_housing california_housing california_housing mnist_test.csv mnist_test.csv mnist_test.csv mlist_test.csv bib dev etc home lib lib lib32 lib64 media fd 15 GB available 	Ign:1 https://developer.download.nvidia.com/compute/uda/repos/ubunt1804/x86 64 InRelease Get:2 https://developer.download.nvidia.com/compute/uda/repos/ubunt1804/x86 64 InH Hit4 https://developer.download.nvidia.com/compute/uda/repos/ubunt1804/x86 64 Release Hit5 https://developer.download.nvidia.com/compute/uda/repos/ubunt1804/x86 64 Release Get:6 http://ppa.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic-record/0 Fackages [67.4 kB] Get:7 https://lopal.aunchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic InRelease Get:11 http://ppa.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic InRelease Get:16 http://ppa.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic InRelease Get:16 http://ppa.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic InRelease Get:16 http://ppa.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic InRelease Get:16 http://ps.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic InRelease Get:19 http://ps.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic/main Sources [1,428 kB] Get:20 http://ps.launchpad.net/c2d4u.team/c2d4u.for/ubuntu bionic/main Sources [1,802 kB] Get:21 http://ps.launchpad.net/c2d4u.team/c2d4u.for/universe amd64 Fackages [2,202 kB] Get:22 http://archive.ubuntu.com/ubuntu bionic-security/metsra amd64 Fackages [2,202 kB] Get:23 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Fackages [2,202 kB] Get:24 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Fac	Open editor settings IRelease :lease
04,10 OB available		• ×

dev 🕨 🖿 etc

🕨 🖿 lib 🖿 lib32

lib64

media

🖿 opt proc

python-apt m root

mnt

▶ 🖿 run 🖿 sbin

Disk

=

🖿 home





×

11



💟 Webmail :: Welcome 🗶 💽 maXbox maxbox 🛛 🗴 🏨 m	XScriptasHTML × 👷 softwareschule.ch/d: × 🔥 (14) (PDF) maXbox_S × 👿 Tensor Processing U: ×	CO EKON24_SimpleImag × + - C ×
\leftarrow \rightarrow C \textcircled{a} \bigcirc A $\overrightarrow{\sim}$ https://cola	p.research.google.com/github/maxkleiner/python4delphi/blob/master/EKON. 🏠 🔍 lazbuild	\rightarrow \bigtriangledown \checkmark \parallel
CO CEKON24_SimpleImageClassificatio	CPU.ipynb Cannot save changes	🖘 Share 🌣 🍘
\equiv Files $ imes$	+ Code + Text 🔺 Copy to Drive	✓ RAM Disk → Editing ∧
	 Layer 10 Neurons: 10 Max Weight: 0.319 Min Weight: -0.273 Max Output: Layer 11 Starting Testing. Epochs: 50 Examples seen:2000000 Test Accuracy: 0.8393 Test Error: 0.4457 Epoch time: 2.3167 minutes. 50 epochs: 1.9306 hours. 	5.777 Min Output ↑ ↓ ⊕ ✿ ☑ : 0.608 Min Output: 0.000 TNNetSoltMax 7 Test Loss: 0.4874 Total time: 176.12mi
 boot content sample_data README.md 	Epochs: 50. Working time: 2.94 hours. Finished.	×
SimpleImageClassifier-64.csv : anscombe.json california_housing_test(_tast modified Mi california_housing_train.csv	[] from google.colab import files !ls -1 sifier-64.csv(3.31K) n Oct 04 2021 11:5455 GMT+0200 (Central European Summer Time) -rw-rr 1 root root 61 Sep 23 08:51 batches.meta.txt	
 mnist_test.csv mnist_train_small.csv matalab 	drwxr-xr-x 2 2156 1103 4096 Jun 4 2009 citar-10-batches-bin -rw-r-rr-1 root root 30730000 Sep 23 08:51 data_batch_1.bin -rw-rr-rr-1 root root 30730000 Sep 23 08:51 data_batch_2.bin -rw-rr-rr-1 root root 30730000 Sep 23 08:51 data_batch_4.bin	
web run.sh dev dev	-rw-r-r 1 root root 30730000 Sep 23 08:51 data_batch_5.bin -rw-r-r 1 root root 170052171 Sep 23 08:46 file.tar drwxr-xr-x 5 root root 4096 Sep 23 08:39 http: drwxr-xr-x 7 root root 4096 Sep 23 08:42 neural-api -rw-root root 20 Sep 23 08:42 neural-api	
SimpleImageClassifier-64.nn Disk 64.15 GB available	<pre>-rw-r-r 1 root root oo Sep 23 08:51 readme.ntmi drwxr-wr-x 1 root root 4096 Sep 16 13:40 sample_data -rw-rr- 1 root root 3390 Sep 23 11:48 SimpleImageClassifier-64.nn -rw-rr 1 root root 2349757 Sep 23 11:41 SimpleImageClassifier-64.nn -rw-r-r-r- 1 root root 30730000 Sep 23 08:51 test_batch.bin</pre>	57 n
		• ×

Blaise Pascal Magazine 96 2021







jupyter colab

Some Extra Features

System Aliases 1 Jupyter includes shortcuts for common operations, such as Is and others.

- 2 Tab-Completion and Exploring Code Colab provides tab completion to explore attributes of Python objects, as well as to quickly view documentation strings.
- 3 **Exception Formatting** Exceptions are formatted nicely in Colab outputs
- **4** Rich, Interactive Outputs Until now all of the generated outputs have been text, but they can be more interesting.
- Integration with Drive Colaboratory is integrated with Google Drive. It allows you to share, comment, and collaborate on the same document with multiple people.

Differences between Google Colab and Jupyter notebooks

1 Infrastructure

Google Colab runs on Google Cloud Platform (GCP). Hence it's robust, flexible Hardware

Google Colab recently added support for Tensor Processing Unit (TPU) apart from its existing GPU and CPU instances. So, it's a big deal for all deep learning people.

3 Pricing

2

Despite being so good at hardware,

the services provided by Google Colab are completely free.

This makes it even more awesome. 4

Integration with Google Drive

Yes, this seems interesting as you can use your google drive as an interactive file system with Google Colab. This makes it easy to deal with larger files while computing your stuff.

Boon for Research and Startup Community 5

Perhaps this is the only tool available in the market which provides such a good PaaS for free to users.

This is overwhelmingly helpful for startups,

the research ommunity and students in deep learning space





IMAGE DETECTION WITH LAZARUS Page 8/22 maXbox Starter 87



Platform as a service (PaaS) is a cloud computing model where a third-party provider delivers hardware and software tools to users over the internet. Usually, these tools are needed for application development.

A PaaS provider hosts the hardware and software on its own infrastructure.











jupyter colab

Software as a service (SaaS) is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. SaaS is also known as "on-demand software" and Web-based/Web-hosted software.

SaaS is considered to be part of cloud computing, along with infrastructure as a service (IaaS), platform as a service (PaaS), desktop as a service (DaaS), managed software as a service (MsaaS), mobile backend as a service (MBaaS), datacenter as a service (DcaaS), integration platform as a service (iPaaS), and information technology management as a service (ITMaaS).

SaaS apps are typically accessed by users using a thin client, e.g. via a web browser. SaaS became a common delivery model for many business applications, including office software, messaging software, payroll processing software, DBMS software, management software, CAD software, development software, gamification, virtualization, accounting, customer relationship management (CRM), collaboration (MIS), enterprise resource planning (ERP), invoicing, field service management, human resource management (HRM), talent acquisition, learning management systems, content management (CM), geographic information systems (GIS), and service desk management.

SaaS has been incorporated into the strategy of nearly all enterprise software companies.

(Gartner forecasted that software as a service (SaaS) remains the largest market segment on public cloud services and is forecast to reach \$122.6 billion in 2021.)



Infrastructure

as a service (IaaS)

are online services that provide high-level APIs used to dereference various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc.

A hypervisor*, such as Xen, Oracle VirtualBox, Oracle VM, KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.

(*A hypervisor (or virtual machine monitor, VMM, virtualizer) is a kind of emulator; it is computer software, firmware or hardware that creates and runs virtual machines.)

Typically IaaS involves the use of a cloud orchestration technology like **OpenStack, Apache CloudStack** or **OpenNebula**. This manages the creation of a virtual machine and decides on which hypervisor (i.e. physical host) to start it, enables **VM** migration features between hosts, allocates storage volumes and attaches them to **VMs**, tracks usage information for billing and more.

An alternative to hypervisors are Linux containers, which run in isolated partitions of a single Linux kernel running directly on the physical hardware. Linuxcgroups and namespaces are the underlying Linux kernel technologies used to isolate, secure and manage the containers.

Containerisation offers higher performance than virtualization, because there is no hypervisor overhead.





IMAGE DETECTION WITH LAZARUS Page 10/22



IaaS clouds often offer

additional resources such as a virtual-machine disk-image library, raw block storage, file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles.

The **NIST's** (*National Institute of Standards and Technology*) definition of cloud computing defines infrastructure as a service as:

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

According to the **Internet Engineering Task Force (IETF)**, the most basic cloud-service model is that of providers offering IT infrastructure – virtual machines and other resources – as a service to subscribers.

IaaS-cloud providers supply these resources on-demand from their large pools of equipment installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure.

In this model, the cloud user patches and maintains the operating systems and the application software.

Cloud providers typically bill **IaaS services** on a utility computing basis: cost reflects the amount of resources allocated and consumed.



https://www.nist.gov/





IMAGE DETECTION WITH LAZARUS Page 11/22



colab jupyter

Now I want to show step by step (16 steps) how we can organize this Lazarus-Project and build and train an image classifier and detector. You can open the link in Colab and start with:

() apt-get update

Update, as mentioned above, will fetch available software and update the lists while upgrade will install new versions of software installed on your computer or in our case on the colab cloud (actual software updates).

Then it goes like this:

0% [Working] Ign:1 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86 64 InRelease 0% [Connecting to archive.ubuntu.com] [Connecting to security.ubuntu.com] [Wait Get:2 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,626 B]

These steps are done for you with a Jupyter notebook. A Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and describing text. So the second more inter-esting command will be:

(2) !apt-get install fpc fpc-source lazarus git subversion

Reading package lists... Done Building dependency tree Reading state information... Done git is already the newest version (1:2.17.1-1ubuntu0.9).

The following additional packages will be installed:

autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-stripnondeterminism file fp-compiler-3.0.4 fp-docs-3.0.4 fp-ide-3.0.4 fp-unitsbase-3.0.4 fp-units-db-3.0.4 fp-units-fcl-3.0.4 fp-units-fv-3.0.4 fp-unitsgfx-3.0.4 fp-units-gtk2-3.0.4 fp-units-math-3.0.4 ...

So the last entries of install fpc will be:

Processing triggers for man-db (2.8.3-2ubuntu0.1) ... Processing triggers for mime-support (3.60ubuntu1) ... Processing triggers for libvlc-bin:amd64 (3.0.8-Oubuntu18.04.1) ...

Thanks to FPC and git subversion we now can install Lazarus on a Ubuntu Bionic image machine. Ubuntu is distributed on three types of images, but we let colab to choose from. You can check your actual platform with a live python notebook script:

```
import platform
platform.platform()
>>> Linux-5.4.104+-x86_64-with-Ubuntu-18.04-bionic
```

Our next task should get the API for the neural network working on the bionic platform!





IMAGE DETECTION INTERNAL								
maXbox Starter 87								
M 3-								
Fe5			- lah					
			COICID					
	Web	mail 🙆 softw	varesci 🔹 GitHub			Conv of X	Readme S 🖉 m	aXboxLo
M.C.		We Sold		inity () indexternely	T EKONE4_SI	- copy of A		
~	\rightarrow (C 🛆	○ △ = htt	os://colab.research. go	ogle.com/drive/1	A5oayNDOeRVzo	:Sy4LHCXsHCjbbe රි	약 ubu
	^	Copy of EKO	N24_SimpleIm	ageClassificatio	nCPU.ipynb			
	File	Edit View In	sert Runtime T	ools Help				
	+ Cor	de + Text						
≣	1 000					/		
	<pre>/ [1]</pre>	Get:21 http:	//ppa.launchpa	d.net/c2d4u.team/	(c2d4u4.0+/ubu	<u>untu</u> bionic/ma	ain amd64 Packages	[922 kB]
Q		Get:22 http: Get:23 http:	//archive.ubun	tu.com/ubuntu bio	onic-updates/n	nain amd64 Pa	ckages [2,787 kB]	(D)
		Get:24 http:	//archive.ubun	tu.com/ubuntu bio	onic-updates/u	universe amd6	4 Packages [2,202 }	kB]
$\langle \rangle$		Get:25 http:	//ppa.launchpa	d.net/graphics-dr	rivers/ppa/ubu	untu bionic/ma	ain amd64 Packages	[42.0 kB
		Fetched 13.1 Reading pack	. MB in 4s (3,0	b3 kB/s)				
		Redding pack	age 11505 5	511C				
	•	(·					
	•	nlatform.	orm					
		PIGOLOI.	architectu	platform.architect	ure(executable=	'/usr		
		0	collections					
	[]	!apt-get ß	DEV_NULL					
		Beading n	dist					
		Building	java_ver					_
		Reading s 🖗	libc_ver					_
		git is al 🏵	linux_distribu	tion				_
		The follo 🏵	mac_ver					_
		autocon 🕅	machine			conf		
		dh-stri	node			fp-id	e-3.0.4	
		fp-unit {}	03			nits-	20-3.0.4	
		fp-units=m	platform	ILS-MULTIMEGIA-	.U.4 ID-UDITS	s-net-3.0.4		
		fp-units-r	tl-3.0.4 fp-ut	ils-3.0.4 fpc-3.0	.4 fpc-source	-3.0.4 gdb g	dbserver	
		gettext ge	ttext-base gir	1.2-atk-1.0 gir1.	2-freedesktop	gir1.2-gdkp	ixbuf-2.0	
=		gir1.2-gtk	-2.0 gir1.2-pa	ngo-1.0 intltool-	- debian lazarı	us-1.8 lazaru	s-doc-1.8	
					√ 0s	completed at 3:	57 PM	

3 !git clone https://github.com/joaopauloschuler/neural-api.git

Cloning into 'neural-api'... remote: Enumerating objects: 2372, done. remote: Counting objects: 100% (494/494), done. remote: Compressing objects: 100% (370/370), done. remote: Total 2372 (delta 340), reused 236 (delta 124), pack-reused 1878 Receiving objects: 100% (2372/2372), 4.58 MiB | 10.38 MiB/s, done. Resolving deltas: 100% (1585/1585), done.

Blaise Pascal Magazine 96 2021







The git clone is a git command, which creates a clone/copy of an existing repository into a new directory.

It is also used to create remote-tracking branches for each branch in the cloned repository.

It is the most common command which allows users to obtain a development copy of an existing central repository.

Good to know after the clone, a plain git fetch without arguments will update all the remotetracking branches, and a git pull without arguments will in addition merge the remote master branch into the current master branch.

The neural-api or **CAI API (Conscious Artificial Intelligence)** is something like TensorFlow for Pascal and is a platform-independent open source library for artificial intelligence or machine learning in the field of speech recognition, image classification, OpenCL, data science and computer vision.

https://sourceforge.net/projects/cai/files/

It could be that you see some **Pascal dialect** but the other dialects of **Object Pascal** have always aligned themselves closely with the Delphi dialect of the language. Free Pascal/Lazarus, Oxygene, Smart Pascal, maXbox, DWScript, PdScript, PascalABC, etc... all do them. So while there isn't an official standard of Object Pascal, the dialects stay close to each other.

Then we use checkout and Lazbuild to prepare more of the project, above all we compile a package MultiThreadProcsLaz 1.2.1 with in then end with 1215 lines compiled, 0.1 sec and 5 hints issued:

!svn checkout https://svn.code.sf.net/p/
 lazarus-ccr/svn/components/multithreadprocs mtproc

5 !lazbuild mtprocs/multithreadprocslaz.lpk

6 lls -l neural-api/examples/SimpleImageClassifier/SimpleImageClassifier.lpi

Point 6 shows the project we use:

-rw-r--r-- 1 root root 5694 Sep 23 08:37 neural-api/ examples/SimpleImageClassifier/SimpleImageClassifier.lpi





Image Defection With Lazarus Image Page 14/22 Image Defection Image Page 14/22 Image Page 14/2

OPTIONS

-h,--help Displays a short help message.
-B,--build-all build all files of project/package.
We check that **lpi-build** with:

8.ls -l neural-api/bin/x86_64-linux/bin/SimpleImageClassifier -rwxr-xr-x 1 root root 1951024 Sep 23 08:43 neural-api/bin/x86_64linux/bin/SimpleImageClassifier*

We can see we have execution rights rwx on the project-code in our scripts.

Next step 9 is to get the image based data to train and test with it and step 10 checks that download:

```
import os
import urllib.request
```

```
if not os.path.isfile('cifar-10-batches-bin/data_batch_1.bin'):
    print("Downloading CIFAR-10 Files")
    url = 'https://www.cs.toronto.edu/~kriz/cifar-10-binary.tar.gz'
    urllib.request.urlretrieve(url, './file.tar')
```

>>> Downloading CIFAR-10 Files

🕕 ls -l

total 166080 -rw-r--r-- 1 root root 170052171 Sep 23 08:46 file.tar drwxr-xr-x 5 root root 4096 Sep 23 08:39 mtprocs/ drwxr-xr-x 7 root root 4096 Sep 23 08:42 neural-api/ drwxr-xr-x 1 root root 4096 Sep 16 13:40 sample_data/

It's the file.tar we downloaded.

We made a script that executes the whole build script in maXbox imported from a jupyter notebook. This version 4.7.5.80 from July 2021 allows us with the help of Python4Delphi and an environment with modules in site-packages to execute Py-functions. But the most is only available in a 32-bit space as maXbox4 is still 32-bit, possible also with 64-bit Python means calling external shell(ExecuteShell) and **Python4Lazarus**. We have to unpack those files:



IMAGE DETECTION WITH LAZARUS Page 15/22

!tar -xvf ./file.tar

and we get:

cifar-10-batches-bin/ cifar-10-batches-bin/data_batch_1.bin cifar-10-batches-bin/batches.meta.txt cifar-10-batches-bin/data_batch_3.bin cifar-10-batches-bin/data_batch_4.bin cifar-10-batches-bin/test_batch.bin cifar-10-batches-bin/readme.html cifar-10-batches-bin/data_batch_5.bin cifar-10-batches-bin/data_batch_2.bin

12 Copying files to current folder

if not os.path.isfile('./data_batch_1.bin'):
 print("Copying files to current folder")
 !cp ./cifar-10-batches-bin/* ./
In 12, we copy the image files to prepare for running
 the project of image classification
 training step ()

if os.path.isfile('./data_batch_1.bin'):
 print("RUNNING!")
 !neural-api/bin/x86_64-linux/bin/SimpleImageClassifier

Hurray, analyze, build, compile and deploy of the 12 layers neural network with 331 neurons (abcd) is running now for about 3 hours!

> RUNNING! Creating Neural Network... Layers: 12 Neurons:331 Weights:162498 Sum: -19.536575 Learning rate:0.001000 L2 decay:0.000010 Batch size:64 Step size:64 File name is: SimpleImageClassifier-64 Training images: 40000 - Validation images: 10000 - Test images: 10000 Computing...

Imagine the accuracy goes up and the loss-function (error-rate) goes down. The loss function is the bread and butter of modern machine learning; it takes your algorithm from theoretical to practical and transforms neural networks from glorified matrix multiplication into deep learning.

https://algorithmia.com/blog/introduction-to-loss-functions





Loss functions are used in

regression when finding a line of best fit by

minimizing the overall loss of all the points with the prediction from the line. Loss functions are used while training perceptrons and neural networks by influencing how their weights are updated (*our result will be such a file with the trained weights!*). The larger the loss, the larger the update.

NOTE:

Loss functions are different based on a problem statement to which deep learning is being applied. The cost function is another term used interchangeably for the loss function, but it holds a more different meaning.

A loss function is for a single training example, while a cost function is an average loss over the complete train dataset (./data batch 1.bin).

After a certain amount of working time we get: Starting Testing. Epochs: 50 Examples seen:2000000 Test Accuracy: 0.8393 Test Error: 0.4457 Test Loss: 0.4874 Total time: 176.12min Epoch time: 2.3167 minutes. 50 epochs: 1.9306 hours.

Epochs: 50. Working time: 2.94 hours.

Finished.

💟 Webmail := 🔍 softwaresci 🚦 GitHub imp	🔋 EKON24 × 📁 Copy of EK 🚯 Readme - S 💽 EKON 24 K ijhmonypqay31 🧇 Conscious 🕐 loss functic + 👘 🗖
← → C @ O A https://gitlab	دti.b fh.ch /knm4/python4delphi/-/blob/master/EKON24_SimpleImage⊂ 分
🖁 Projects 🛩 Groups 🛩 More 🛩	🖬 🗸 🛛 Search or jump to 🔍 🗘 🗗 🔥 🧐 🗸
P python4delphi	1995840 Examples seen. Accuracy: 0.8091 Error: 0.48418 Loss: 0.61899 Threads: 2 Forward time: 0.68s Back 1996480 Examples seen. Accuracy: 0.8083 Error: 0.53391 Loss: 0.53823 Threads: 2 Forward time: 0.68s Back 1997120 Examples seen. Accuracy: 0.8079 Error: 0.47216 Loss: 0.51692 Threads: 2 Forward time: 0.69s Back
✿ Project overview	1997760 Examples seen. Accuracy: 0.8078 Error: 0.56156 Loss: 0.68294 Threads: 2 Forward time: 0.67s Back 1998400 Examples seen. Accuracy: 0.8047 Error: 0.52817 Loss: 0.55287 Threads: 2 Forward time: 0.71s Back
Repository	1999040 Examples seen. Accuracy: 0.8064 Error: 0.52653 Loss: 0.52688 Threads: 2 Forward time: 0.70s Back 1999680 Examples seen. Accuracy: 0.8079 Error: 0.45205 Loss: 0.43995 Threads: 2 Forward time: 0.70s Back
Files	Starting Validation. Epochs: 50 Examples seen:2000000 Validation Accuracy: 0.8461 Validation Error: 0.4314 Validation Loss: 0
Commits	Layer 0 Max Output: 1.672 Min Output: -1.922 TNNet Layer 1 Neurons: 64 Max Weight: 0.626 Min Weight: -0.628 Max Output: 5.432 Min Output: -5.954 TNNet
Branches	Layer 2 Max Output: 5.432 Min Output: -1.371 TNNet
Tags	Layer 3 Neurons: 1 Max Weight: 0.937 Min Weight: 0.897 Max Output: 5.009 Min Output: -2.571 TNNet Layer 4 Neurons: 64 Max Weight: 0.290 Min Weight: -0.370 Max Output: 10.887 Min Output: 0.000 TNNet
Contributors	Layer 5 Neurons: 64 Max Weight: 0.376 Min Weight: -0.336 Max Output: 11.827 Min Output: 0.000 TNNet Layer 6 Neurons: 64 Max Weight: 0.346 Min Weight: -0.269 Max Output: 7.334 Min Output: 0.000 TNNet
Graph	Layer 7 Neurons: 64 Max Weight: 0.215 Min Weight: -0.224 Max Output: 3.276 Min Output: 0.000 TNNet Layer 8 Max Output: 3.276 Min Output: 0.000 TNNet
Compare	Layer 9 Max Output: 3.276 Min Output: 0.000 TNNet
Locked Files	Layer 10 Neurons: 10 Max Weight: 0.319 Min Weight: -0.273 Max Output: 5.777 Min Output: -10.742 TNNe Layer 11 Max Output: 0.608 Min Output: 0.000 TNNet
D Issues 0	Epochs: 50 Examples seen:2000000 Test Accuracy: 0.8393 Test Error: 0.4457 Test Loss: 0.4874 Total time: Epoch time: 2.3167 minutes. 50 epochs: 1.9306 hours.
Ξ - Requirements	Epochs: 50. Working time: 2.94 hours. Finished.
≪ Collapse sidebar	< >>





Page 16/22

IMAGE DETECTION WITH LAZARUS Page 17/22

Now we

want to export the trained result, means we get 2 files:

from google.colab import files !ls -l

-rw-r--r-- 1 root root 3390 Sep 23 11:48 SimpleImageClassifier-64.csv -rw-r--r-- 1 root root 2349757 Sep 23 11:41 SimpleImageClassifier-64.nn

files.download('SimpleImageClassifier-64.nn')

(6 files.download('SimpleImageClassifier-64.csv')

Note: Probably you get an FileNotFoundError: Cannot find file: SimpleImageClassifier.csv because colab increments files with a postfix in the file name: SimpleImageClassifier-65.csv

In that casse you have to adjust the download command 15 and 16:

files.download('SimpleImageClassifier-65.csv')

So what's the context of the 2 files. The csv is just the log of the training with the hyperparameters such as learning rate:

epoch training accuracy training loss training error validation accuracy validation loss validation error learning rate time test accuracy test loss test error

The *.nn file serves as a pretrained file (FAvgWeight) to classify or predict images we trained on. Also the CIFAR-10 classification examples with experiments/testcnnalgo/testcnnalgo.lpr and a number of CIFAR-10 classification examples are available on /experiments.

Git is a distributed version control system, which means you can work locally, then share or "push" your changes to a server. In our case, the server is GitLab. This fascinating option is to run the entire system as runtime virtualization with the help of a **Jupyter notebook** running on **Ubuntu** in the cloud on **Colab** or an **Colab.research** container.

We step a last step to exec a script in a script! If we call a file or a Python command then we use ExecString (PYCMD) : http://www.softwareschule.ch/examples/pydemo19.txt/





IMAGE DETECTION WITH LAZARUS Page 18/22

At last a minimal configuration called "**Pyonfly**" with a colab platform tester. The minimal configuration depends on your Python-installation and the UseLastKnownVersion property in TDynamicDll and if something went wrong you got a raiseError Py exception:

```
with TPythonEngine.Create(Nil) do begin
pythonhome:= PYHOME;
try
loadDLL;
Println('Colab Platform: '+
EvalStr('_import_("platform").platform()'));
except
raiseError;
finally
free;
end;
end;
```







IMAGE DETECTION WITH LAZARUS Page 19/22



EKON CAI, P4D and Colab topics

https://entwickler-konferenz.de/delphi-innovationsfundamentals/python4delphi/ https://colab.research.google.com/ https://entwickler-konferenz.de/blog/machine-learning-mit-cai/

Learn about Jupyter.org

https://jupyter.org/ https://forum.lazarus.freepascal.org/index.php?topic=38955.0

CONCLUSION SCRIPT:

Here is the final Result of this project seen as an image:



Figure 12: Person : 80.40134906768799; output on colab/content/sample_data

Continuation on the next page







CONCLUSION SCRIPT:

Note: You will need a google account to run a predefined jupyter notebook on Colab; the exported script of the classification:

```
# -*- coding: utf-8 -*-
"""Copy EKON_SimpleImageClassificationCPU.ipynb
Automatically generated by Colaboratory.
                               https://colab.research.google.com/drive/1clvG2uoMGo-
Original file is located at
bfrJnxBJmpNTxjvnsMx9
.......
!apt-get update
!apt-get install fpc fpc-source lazarus git subversion
!git clone https://github.com/joaopauloschuler/neural-api.git
!svn checkout https://svn.code.sf.net/p/lazarus-ccr/svn/components/multithreadprocs mtprocs
!lazbuild mtprocs/multithreadprocslaz.lpk
!ls -l neural-api/examples/SimpleImageClassifier/SimpleImageClassifier.lpi
!lazbuild neural-api/examples/SimpleImageClassifier.lpi
ls -l neural-api/bin/x86 64-linux/bin/SimpleImageClassifier
import os
import urllib.request
if not os.path.isfile('cifar-10-batches-bin/data batch 1.bin'):
 print("Downloading CIFAR-10 Files")
  url = 'https://www.cs.toronto.edu/~kriz/cifar-10-binary.tar.gz'
  urllib.request.urlretrieve(url, './file.tar')
ls -1
!tar -xvf ./file.tar
if not os.path.isfile('./data batch 1.bin'):
 print("Copying files to current folder")
  !cp ./cifar-10-batches-bin/* ./
if os.path.isfile('./data_batch_1.bin'):
 print("RUNNING!")
  !neural-api/bin/x86 64-linux/bin/SimpleImageClassifier
from google.colab import files
!ls -1
files.download('SimpleImageClassifier-66.nn')
files.download('SimpleImageClassifier-66.csv')
```





IMAGE DETECTION WITH LAZARUS Page 21/22



References:

Docs: https://maxbox4.wordpress.com/blog/ http://www.softwareschule.ch/download/maxbox_starter86_3.pdf

EKON

Image Classification with Lazarus

https://colab.research.google.com/github/maxkleiner/maXbox/blob/master/ EKON24_SimpleImageClassificationCPU.ipynb

November 8 – 10, 2021 Dusseldorf

REGISTER NOW

YEARS

UNTIL OCTOBER 7

save up to **200€**

https://entwickler-konferenz.de/location-en/



16 Blog

SEP MACHINE LEARNING MIT CAI

This report visualizes the field of object recognition using computer vision techniques from machine learning. An image classifier from the CAI framework in Lazarus and Delphi, the so-called CIFAR-10 image classifier, is also used.





18 Blog, Interview

SEP "DELPHI DEVELOPMENT IS STILL GOING STRONG"

Marco Cantu talkes about the current status of Delphi, how it has evolved, and what's in store for this language in the future.

MEHR





IMAGE DETECTION WITH LAZARUS Page 22/22

colab

SPEAKERS OF EKON 25



jupyter

Stefan Glienke Aagon



Dr. Cary Jensen Jensen Data Systems, Inc.



Dr. Don Wibier DevExpress



Lisa Moritz INNOQ



Ray Konopka Raize Software



Max Kleiner kleiner kommunikation



Nigel Tavendale All Things Syslog



Markus Humm ebm-papst Group



Bruno Fierens tmssoftware.com bvba



Arnaud Bouchez Developer at TranquillT -Founder of the Open Source mORMot Framework



Andrea Magni Freelance



Jens Fudge Archersoft Aps



Marco Cantu Embarcadero



Serge Pilko Softacom Ltd.



Bernd Ua probucon Business Consulting GmbH&Co KG



Matthias Eißing Embarcadero



Christoph Schneider Schneider Infosystems AG



Dr. Falk W. Müller diconium digital solutions



Dr. Annegret Junker Allianz Deutschland AG



THE LIBRARY APPLICATION 2021

(Former LIB Stick)

🕵 LIBraryForm

ID





 \times

Sh	ow articl	e Go to issue Nr 13 Search	< <first <="" next="" prior=""> Last >> Add Delete Ed</first>	dit Save	Cancel Refresh CS	V
ID	IssueNr	Author	Article	PDF	PageNr	^
150	12	Henk Schreij	SQL produced by the ClientDataSet		1	2
151	12	Marco Cantù	Introducing GoogleDocsApi_SecondPart		1	6
152	12	Benno Evers	The use of kbmMW components in a server application		1	8
153	12	Primoz Gabrijelicic	Inter Process Communication		2	4
154	12	David Dirkse	Peg Solitaire - a single player puzzle		2	7
155	12	Peter Bijlsma	Trans-continental travel using the great circle		3	1
156	12	Fikret Hasovic	Creating gateways and other advanced topics		3	4
157	12	Bruno Fierens	Creating Office 2010 style VCL applications with TMS components		3	5
158	12	Anton Vogelaar	Electronics Delphi Controller and Delphi Developer board		3	7
159	13	Miguel van de Laar	Mini Course SQL		1	0
160	13	Zeljan	Is Lazarus ready for writing commercial applications?		1	3
161	13	Pawel Glowacki	Delphi JSON Viewer		1	6
162	13	Erwin Mouthaan	NexusDB exceptionaly good, a real surprise		2	2
163	13	by Bruno Fierens	The TMS DataModeler		2	7
164	13	Cary Jensen	Introduction to Delphi Database Development: Part 1		3	1 🗸







expert

The **Skia4Delphi** is an open-source cross-platform 2D graphics library for **Delphi** based on **Google's**

Skia Graphics Library, to draw and manipulate text, geometries, images, focused on accurate, high quality and high-performance rendering. Google's Skia Graphics Library serves as the graphics engine for Google Chrome and Chrome OS, Android, Flutter, Xamarin, Mozilla Firefox and Firefox OS, and many other products.

It provides common and comprehensive 2D APIs that work across a variety of hardware and software platforms, abstracting the implementation of low level libraries that it uses behind, like **Vulkan**, **DirectX**, **Metal**, **GLES**, and others.

We can create several projects in **Delphi**, among them are **Console**, **VCL and FMX** (**Firemonkey**), the latest being the new generation of cross-platform GUI applications made to replace **VCL**.

There is no denying the great advances that have been made in the graphics part in relation to **VCL**, the truth is that Firemonkey's graphics engine (drawing, text and codecs) in general is good.

However, when compared to other modern languages we see some important limitations of Firemonkey's graphics engine. Here are some:



 Support for right-to-left languages like Arabic, Hebrew and Persian; Antialiasing support on any platform, but resulting in poor drawing quality; Font-weight support, which is the thickness of the font, is absolutely indispensable in modern designs; In addition to the ability to use installed system fonts, support for custom fonts installed or loaded at run-time; Many advanced text features for writing paragraphs such as choosing the maximum number of lines of text. Currently either you choose to have only 1 line with word wrap disabled, or you choose to have the maximum number of lines with it enabled; Vulkan support (OpenGL/GLES replacement library for low level drawings); DirectX 12 support; Support for parallel drawings. It is possible to draw using the CPU or th GPU easily using multiple threads, without locks; Full Unicode support. Delphi supports Unicode well, but it can't predict exactly how many bytes a grapheme occupies in a string. This is a problem as new emojis can take up to 16 bytes and so, under certain circumstances, there may be a malformed string during string trimming operations; Drawing cache control; Advanced resampling filters to better resize images; Support for SVG, GIF, HEIC and WEBP image formats; Support for SVG, GIF, HEIC and WEBP image formats; PDF creation support. 			
 Is absolutely indispensable in modern designs; In addition to the ability to use installed system fonts, support for custom fonts installed or loaded at run-time; Many advanced text features for writing paragraphs such as choosing the maximum number of lines of text. Currently either you choose to have only 1 line with word wrap disabled, or you choose to have the maximum number of lines with it enabled; Vulkan support (OpenGL/GLES replacement library for low level drawings); DirectX 12 support; Support for parallel drawings. It is possible to draw using the CPU or the GPU easily using multiple threads, without locks; Full Unicode support. Delphi supports Unicode well, but it can't predict exactly how many bytes a grapheme occupies in a string. This is a problem as new emojis can take up to 16 bytes and so, under certain circumstances, there may be a malformed string during string trimming operations; Drawing cache control; Advanced resampling filters to better resize images; Support for SVG, GIF, HEIC and WEBP image formats; Support animated image formats animated GIF, animated WEBP, APNG, but mainly lottie files and tgs files (telegram stickers); PDF creation support. 		Support for right-to-left languages like Arabic, Hebrew Antialiasing support on any platform, but resulting in poor Font-weight support, which is the thickness of the font,	and Persian; or drawing quality;
 Many advanced text features for writing paragraphs such as choosing the maximum number of lines of text. Currently either you choose to have only 1 line with word wrap disabled, or you choose to have the maximum number of lines with it enabled; Vulkan support (OpenGL/GLES replacement library for low level drawings); DirectX 12 support; Support for parallel drawings. It is possible to draw using the CPU or th GPU easily using multiple threads, without locks; Full Unicode support. Delphi supports Unicode well, but it can't predict exactly how many bytes a grapheme occupies in a string. This is a problem as new emojis can take up to 16 bytes and so, under certain circumstances, there may be a malformed string during string trimming operations; Drawing cache control; Advanced resampling filters to better resize images; Support for SVG, GIF, HEIC and WEBP image formats; Support animated image formats animated GIF, animated WEBP, APNG, but mainly lottie files and tgs files (telegram stickers); PDF creation support. 		In addition to the ability to use installed system fonts , support for custom fonts installed or loaded at run-time:	
 Vulkan support (OpenGL/GLES replacement library for low level drawings); DirectX 12 support; Support for parallel drawings. It is possible to draw using the CPU or the GPU easily using multiple threads, without locks; Full Unicode support. Delphi supports Unicode well, but it can't predict exactly how many bytes a grapheme occupies in a string. This is a problem as new emojis can take up to 16 bytes and so, under certain circumstances, there may be a malformed string during string trimming operations; Drawing cache control; Advanced resampling filters to better resize images; Support for SVG, GIF, HEIC and WEBP image formats; Support animated image formats animated GIF, animated WEBP, APNG, but mainly lottie files and tgs files (telegram stickers); PDF creation support. 		Many advanced text features for writing paragraphs such as the maximum number of lines of text . Currently either to have only 1 line with word wrap disabled, or you choose the maximum number of lines with it enabled:	s choosing you choose to have
 Full Unicode support. Delphi supports Unicode well, but it can't predict exactly how many bytes a grapheme occupies in a string. This is a problem as new emojis can take up to 16 bytes and so, under certain circumstances, there may be a malformed string during string trimming operations; Drawing cache control; Advanced resampling filters to better resize images; Support for SVG, GIF, HEIC and WEBP image formats; Support animated image formats animated GIF, animated WEBP, APNG, but mainly lottie files and tgs files (telegram stickers); PDF creation support. 	0	Vulkan support (OpenGL/GLES replacement library for low DirectX 12 support; Support for parallel drawings. It is possible to draw us GPU easily using multiple threads, without locks;	v level drawings); ing the CPU or th
 Drawing cache control; Advanced resampling filters to better resize images; Support for SVG, GIF, HEIC and WEBP image formats; Support animated image formats animated GIF, animated WEBP, APNG, but mainly lottie files and tgs files (telegram stickers); PDF creation support. 		Full Unicode support . Delphi supports Unicode well, but exactly how many bytes a grapheme occupies in a string. This is a problem as new emojis can take up to 16 bytes and under certain circumstances, there may be a malformed str string trimming operations;	it can't predict 1 so, ing during
		Drawing cache control; Advanced resampling filters to better resize images; Support for SVG, GIF, HEIC and WEBP image formats Support animated image formats animated GIF, animate but mainly lottie files and tgs files (telegram stickers); PDF creation support.	; ed WEBP, APNG,



SKIA4DELPHI PROJECT

PAGE 2/5

With that, and given the need to improve the current **Delphi 2D** drawing engine, came the project **Skia4Delphi**, to make all thinks that the current firemonkey's graphic engine does, but also with numerous improvements and features among them, all mentioned above.



Furthermore, the project came to unify/standardize the drawing engine used in these 3 types of projects: **Console, VCL** and **Firemonkey.** With this, both the front-end and the back-end will be able to use **Skia4Delphi for 2D graphics.**

Highlight the server side, that is, be it console, windows service, or another type, as it is an incredible advance, since there are no good alternatives in the standard **Delphi** code.

For example, now a multithreading server will be able to receive images, decode, resize, crop, encode in another format and save to the database, in a completely parallel way, without bottlenecks and with high quality images and in formats that take up less space, like **WEBP**.

No less amazing too is the use of the library for **VCL** and **Firemonkey**.

Applications can add **SVG** (vectored image) instead of static images

and improve the quality of icons at any scale while decreasing the size of the app. They can render text with several new features, such as: font-weight, automatic right-to-left, custom font, max lines and multi-styles.

They can upload animations of their logos, through lottie files,

or render **Telegram stickers**, and both with maximum quality as they are vectorized images and are still extremely small files when compared to videos or animated gifs. And much more.

Currently in **Firemonkey**, the library acts as a helper for the existing graphics engine, without using the **GPU**, but in future versions we plan to optionally completely replace its ***2D (not meant as a dimensional naming it is only a name)** graphics engine, which briefly consists of drawing, text and codecs, to have an automatic gain of resources and even performance in existing applications.



2D computer graphics is the computer-based generation of digital images -mostly from two - dimensional models (such as 2D geometric

WikipediA

models, text, and digital images) and by techniques specific to them. It may refer to the branch of computer science that comprises such techniques or to the models themselves.

2D computer graphics are mainly used in applications that were originally developed upon traditional printing and drawing technologies, such as typography, cartography, technical drawing, advertising, etc. In those applications, the two-dimensional image is not just a representation of a real-world object,





SKIA4DELPHI PROJECT

PAGE 3/5

but an independent artifact with added semantic value; two-dimensional models are therefore preferred, because they give more direct control of the image than 3D computer graphics (whose approach is more akin to photography than to typography).



In many domains, such as desktop publishing, engineering, and business, a description of a document based on 2D computer graphics techniques can be much smaller than the corresponding digital image—often by a factor of 1/1000 or more. This representation is also more flexible since it can be rendered at different resolutions to suit different output devices. For these reasons, documents and illustrations are often stored or transmitted as 2D graphic files.

2D computer graphics started in the 1950s, based on vector graphics devices. These were largely supplanted by raster-based devices in the following decades. The PostScript language and the X Window System protocol were landmark developments in the field.

Skia4Delphi supports all Delphi

versions of the last 7 years (XE6 and newer) in Windows applications, and in the latest versions, **Delphi 10.4 Sydney** and 11 Alexandria, the library is fully cross-platform, that is, it supports all the platforms Delphi supports: Windows, Linux, Android, iOS and MacOS (including MacOS ARM).

To use it, just download it from our website

to use it, just download it from our website

https://skia4delphi.org.

Some samples with the main functionalities will also be installed. The online documentation is on the official github page: https://github.com/viniciusfbb/skia4delphi



A graphics library is a

program library designed to aid in rendering computer graphics to a monitor. This typically involves providing optimized versions of functions that handle common rendering tasks.

This can be done purely in software and running on the CPU, common in embedded systems, or being hardware accelerated by a GPU, more common in PCs. By employing these functions, a program can assemble an image to be output to a monitor. This relieves the programmer of the task of creating and optimizing these functions, and allows them to focus on building the graphics program. Graphics libraries are mainly used in video games and simulations

The **Skia Graphics Engine** is an open-source graphics library written in C++ which abstracts away platform-specific graphics API (which differ from one to another).[1] Skia Inc. originally developed the library; Google acquired it in 2005,[2] and then released the software as open source licensed under the New BSD free software license in 2008.

Scalable Vector Graphics (SVG) is an (XML)-based vector image format for two-dimensional graphics with support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999.

SVG images and their behaviors are defined in XML text files. This means that they can be searched, indexed, scripted, and compressed, and can be scaled in size without loss of quality. As XML files, SVG images can be created and edited with any text editor, as well as with drawing software. The most-used web browsers render SVG files.



SKIA4DELPHI PROJECT Examples

With skia you can create PDF documents and draw anything on them, from text to images. The example below demonstrates how to create an PDF document and draw an SVG inside it:





PAGE 4/5

SKIA4DELPHI PROJECT

Skia developed a way to load and render svgs with the class **TSkSVGDOM**. The code below shows how to load and render a svg into an **ISkCanvas**:

var Lbitmap: TBitmap; begin LBitmap := TBitmap.Create(100, 100); try LBitmap.SkiaDraw(procedure (const ACanvas: ISKCanvas) var FOR GDEL LStream: ISkStream; LDOM: ISkSVGDOM; LSvgRect, LFitDest: TRectF; begin LStream := TSkStreamAdapter.Create(TStringStream.Create(TFile.ReadAllText('Assets\Samples\gorilla.svg')), True); LDOM := TSkSVGDOM.Make(LStream); if Assigned(LDOM) then begin if LDOM.ContainerSize.IsZero then LDOM.ContainerSize := TSizeF.Create(LBitmap.Width, LBitmap.Height); LSvgRect := TRectF.Create(PointF(0, 0), LDOM.ContainerSize); LFitDest := LSvgRect.FitInto(RectF(0, 0, LBitmap.Width, LBitmap.Height)); ACanvas.Translate(LFitDest.Left, LFitDest.Top); ACanvas.Scale(LFitDest.Width / LSvgRect.Width, LFitDest.Height / LSvgRect.Height); LDOM.Render(ACanvas); end; end);

Paragraph

end;

With the skia SkParagraph, you can render texts



PAGE 5/5





Not only the 26th but all November we have secret Black Friday deals to celebrate the 100th edition of Blaise Pascal Magazine!

Promo valid on all

- * Embarcadero/Borland products * FastReport Components
- ***** TMS Components
- * Steema Components
- RemObject Components
- * Gnostice Components
- * Devart Components
- * DevExpress Components

Send your request to blackfriday@barnsten.com and we will send you a super deal back! Only valid in November 2021 and only at Barnsten.

Phone number: +31 23 542 22 27

BRANCHING AND PARTIAL COMMITS PAGE 1/17

By Michaël Van Canneyt starter expert



OINTRODUCTION

Often, several avenues of development occur at the same time: people working on separate features or fixing issues simultaneously. To prevent these separate development efforts from interfering with each other, they are conducted in different branches of the repository. A branch is simply a series of commits which is kept separate from the 'main' series of commits from which the actual project releases are made. A possible branching situation is depicted in figure 1 on page 1. Each dot in the diagram is a commit, and is named. In reality these names, the so-called commit hashes, will be much longer.



The middle line (green commits 1..6) are the main branch: from this branch, releases are made. The bottom line (purple dots: commits X..Z2) are the commits from work on a feature development. The top line (blue dots: commits A-C, joining at commit 6) are from work on a bugfix.

Figure 1: Icons have a marker if they are changed.




BRANCHING AND PARTIAL COMMITS PAGE 2/17

🔶 git

When a line of development is finished (*as for the bugfix in figure* 1 on page 1), the work is reintegrated with the main development branch from which releases are made. This is depicted by commit 6: the work in commits A-C is merged to the main development line, and in the above diagram, this is done through a new commit. This new commit is not a requirement, but more about this later.

Branching is one way to keep various lines of development separate. This is a conscious planning step.

However, imagine you're working on some code: a feature, a bugfix, this does not really matter. You are working, you have changed one or more files, and while working you notice a bug that you quickly fix. When it is time to commit your work, you decide that this fix needs to be committed separately, not together with the work you were originally intending to do. How to separate this fix from the other changes? This is where a partial commit comes in. This is a simple case, which we'll treat first.

PARTIAL COMMIT

Imagine you need to change the following unit, you need to add a second hash function to it:

```
unit myunit;
{$mode objfpc}
interface
Function MyHash(aString: String): Longint;
implementation
Function MyHash(aString: String): Longint;
begin
Result:=Length(aString) div 2;
end;
end.
```

You start working on it and at some point you notice that you should have divided by 3 in MyHash, so you change it. After you finish the MyBetterHash function, the end result of your work is:

```
unit myunit;
{$mode objfpc}
interface
Function MyHash(aString: String): Longint;
Function MyBetterHash(aString:String):Longint;
implementation
Function MyHash(aString: String): Longint;
begin
  Result:=Length(aString) div 2;
  Result:=Length(aString) div 3;
end:
Function MyBetterHash(aString: String): Longint,
Var C: Char;
begin
  Result:=0;
  For C in aString do
     Result:=Result xor Ord(C);
  end:
end.
```

Blaise Pascal Magazine 99/100 2021



BRANCHING AND PARTIAL COMMITS PAGE 3/17

Now it is time to commit. Obviously, you wish to keep the bugfix separate from the new feature.

As explained in a previous article, the add command in Git adds a file to the next commit. This command has an option -p or -patch, which tells Git to analyse the changes in the file and ask you for each change whether you want to add it to the commit or not. git add --patch myunit.pp

This will result in the following output:

```
diff --git a/myunit.pp b/myunit.pp
index bda87cf..e16c96d 100644
--- a/myunit.pp
+++ b/myunit.pp
@@ -3,13 +3,26 @@ unit myunit;
interface
Function MyHash(aString:String):Longint;
+Function MyBetterHash(aString:String):Longint;
implementation
Function MyHash(aString:String):Longint;
begin
- Result:=Length(aString) div 2;
+ Result:=Length(aString) div 3;
end:
-end
\ No newline at end of file
+Function MyBetterHash(aString: String): Longint;
+Var
+ C: Char;
+begin
+ Result:=0;
+ For C in aString do
+ Result:=Result xor Ord(C);
+end:
+
+end.
(1/1) Stage this hunk [y,n,q,a,d,s,e,?]?
```

Clearly, this will add everything (*the whole 'hunk'*) in one big commit. That's not what you want, obviously, so you can give the 's' command, which splits the commit into smaller commits.

Here are all the possible answers to the question Git is asking:

- y Add this hunk to the next commit.
- n Do not add this hunk to the next commit.
- q quit; Do not add this hunk or any of the remaining hunks.
- a Add this hunk and all later hunks in the file.
- d Drop this hunk and all the later hunks in the file.
- g Go to a particular hunk.
- / Search for a hunk: this command must be followed by a regular expression.
- j Jump over this hunk (leaves it undecided) and show the next undecided hunk.
- J Jump over this hunk (leaves it undecided) and show the next hunk.
- k Leave this hunk undecided, show the previous undecided hunk.
- K Leave this hunk undecided, and show the previous hunk.
- s Split the current hunk into smaller hunks, and repeat the process.
- e Edit the current hunk: you replace +/- by # to skip parts of the hunk.
- ? Show a help message.



BRANCHING AND PARTIAL COMMITS PAGE 4/17



After choosing 's', Git will respond with:

```
Split into 3 hunks.
@@ -3,9 +3,10 @@
interface
Function MyHash(aString : String) : Longint;
+Function MyBetterHash(aString : String) :
Longint;
implementation
Function MyHash(aString : String) : Longint;
begin
(1/3) Stage this hunk [y,n,q,a,d,j,J,g,/,e,?]?
```

You can see that this is the first change in the file. Also note that git now offers more possible answers than the first time it asked the question. Depending on what you want: first commit the feature, then the fix or vice versa, you can now answer 'y' or 'n', respectively. We'll first commit the fix, so we answer 'n' because we don't need the new function declaration. **Git** responds with:

```
@@ -6,9 +7,9 @@
implementation
Function MyHash(aString : String) : Longint;
begin
- Result:=Length(aString) div 2;
+ Result:=Length(aString) div 3;
end;
(2/3) Stage this hunk [y,n,q,a,d,K,j,J,g,/,e,?]?
```

This is the part we want because it contains the fix we wish to commit first, so we answer 'y'. **Git** will then ask us what we want to do with the last part:

```
@@ -13,3 +14,15 @@
end;
-end.
\ No newline at end of file
+Function MyBetterHash(aString : String) : Longint;
+
+Var
+ C : Char;
+begin
+ Result:=0;
+ For C in aString do
+ Result:=Result xor Ord(C);
+end:
+end.
+
(3/3) Stage this hunk [y,n,q,a,d,K,g,/,e,?]?
```



BRANCHING AND PARTIAL COMMITS PAGE 5/17

Here we answer with 'n', because this is not part of the fix we want to commit. After this, **Git** is finished. When we ask for the status, we get the following:

```
> git status
On branch main
Your branch is up-to-date with 'origin/main'.
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified: myunit.pp
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: myunit.pp
```

As you can see, the myunit.pp file is listed twice:

- once as modified and ready to be committed.
- once as modified, but not yet staged for commit.
- If we ask to see the changes in our file using the diff command,

Git will respond with the following:

```
git diff myunit.pp
diff -- git a/myunit.pp b/myunit.pp
index c6b8536..e16c96d 100644
--- a/myunit.pp
+++ b/myunit.pp
@@ -3,6 +3,7 @@ unit myunit;
interface
Function MyHash (aString : String) : Longint;
+Function MyBetterHash(aString : String) : Longint;
implementation
@@ -12,4 +13,16 @@ begin
Result:=Length(aString) div 3;
end:
-end.
\ No newline at end of file
+Function MyBetterHash(aString : String) : Longint;
+
+Var
+ C : Char;
+begin
+ Result:=0;
                                      As you can see, only the new functionality is
+ For C in aString do
+ Result:=Result xor Ord(C);
                                     present in the changes. The bugfix change is already in the
+end;
                                    'staging area', the name Git uses for changes that have been
                                   scheduled for a commit.
+end.
                                 A similar situation can arise when you change a file,
                                and add it to the staging area, but then
                               change the file some more, before committing.
                             We can now commit our bugfix change with the commit
                               command:
```

```
> git commit -m '* Bugfix for MyHash: divide by 3 instead of 2'
[main 5b10ccb] * Bugfix for MyHash: divide by 3 instead of 2
1 file changed, 1 insertion(+), 1 deletion(-)
```

Note the commit hash 5b10ccb. It can be used to check what was committed with the diff command of **Git:**



BRANCHING AND PARTIAL COMMITS PAGE 6/17

> git diff 5b10ccb^{...5b10ccb}
diff --git a/myunit.pp b/myunit.pp
index bda87cf..c6b8536 100644
--- a/myunit.pp
+++ b/myunit.pp
@@ -9,7 +9,7 @@ implementation
Function MyHash(aString : String) : Longint;
begin
- Result:=Length(aString) div 2;
+ Result:=Length(aString) div 3;
end;
end.



The sequence 5b10ccb[^]..5b10ccb in this command-line tells git to show the difference between the commit previous to commit 5b10ccb (denoted by 5b10ccb[^]) and commit 5b10ccb. As you can see, only our bugfix was committed. Now, the new feature can be committed directly:

```
> git commit -m '* Add MyBetterHash function' myunit.pp
[main 324ccf6] * Add MyBetterHash function
1 file changed, 14 insertions(+), 1 deletion(-)
```

If you want to split the changes in multiple commits, the above interactive add command must be repeated for every partial commit you wish to make.

The same operation can be done in **TortoiseGit**, but it is implemented in a somewhat roundabout manner – presumably this is a heritage of the **TortoiseSVN** origins, because in Subversion, no native partial commit functionality is available.

- In the commit dialog of **ToirtoiseGit**, there is a context menu item called Restore after commit (*see figure 2 on article-page 7*). What this does is to create a backup of the file as it is, which will be restored after the commit has happened.
- Now you can open, from within the same commit dialog, the file with the diff & merge tool of **TortoiseGit** (see the Compare with base menu item in figure 2 on article-page 8
- In this tool you can then indicate what changes you want to see included in the commit using the Mark this block menu item in **TortoiseGit** (see figure 3 on article-page 7).
- When you have marked all changes, you can discard all other changes with the Leave only marked blocks menu item (see again figure 3 on article-page 7).
- Now, after you save the file in the diff tool, you can commit the resulting file: see *figure 4 on article-page 8* and note the counts of added and removed lines.
- After the commit, **TortoiseGit** will then restore the backup it made and as a result, all the remaining changes will again be present in your working copy.
- The file will still be marked changed in the explorer, and the remaining changes can also be committed. See *figure 5 on article-page* 9, and again note the counts for added and removed lines.

If you want to split the changes in multiple commits, the above procedure must be repeated for every partial commit you wish to make.



BRANCHING AND PARTIAL COMMITS PAGE 7/17

ick on file for diff versioned Vers i	f): ioned Added De	leted	1/1 Add Signed-off-by Modified Files Submodules	
ision Status	Lines added	Lines	removed	
Mar differen	d 13 –		1	
Wodifier		C C C C C C C C C C C C C C C C C C C	mpare with base ow changes as unified diff vert ip worktree	
	lick on file for diff versioned Vers nsion Status	lick on file for diff): versioned Versioned Added De nsion Status Lines added	lick on file for diff): versioned Versioned Added Deleted I nsion Status Lines added Lines	1/1 Add Signed-off-by lick on file for diff): versioned Versioned Added Deleted Modified Files Submodules nsion Status Lines added Lines removed



Figure 3: The 'mark this block' menu item in TortoiseDiff





BRANCHES

A better way to split fixes and features is to use branches. As explained in the introduction, branches are lines of development that do not interfere: a series of commits that you will only see when you are working in the correct branch.

Branches need to be planned: in difference with the above case where you decide to split a commit to clearly separate a fix from a feature at commit time, the decision to use a branch should be made before changing files (*although you can work around this*).

So, when to branch? Clearly, when parallel development of 2 features are planned that you expect will take a long time, developing them in separate branches makes sense.

But for a small bugfix (*a typo or so*), it's a matter of taste whether or not to create a branch for this. One could argue that this is a matter of being consequent: do every new development in a branch, and merge it to the main development line when it is ready.

Creating a branch is easy and can be done in 2 ways:

SULLAS FOR M	yHash: di	vide by 3	instead of 2		
Amend Last Cor	mmit				1/46
Set author date					
Set author					Add Signed-off-by
ianges made (dou heck: All Non e	uble-click on e Unversio	file for diff): ned Versior	n ed Added De	eleted Modified	Files Submodules
ianges made (dou heck: All Non e Path	uble-click on e Unversio Extension	file for diff): ned Versior Status	ned Added De Lines added	eleted Modified Lines removed	Files Submodules
aanges made (dou heck: All Non ∉ Path ☑ 🕵 myunit.pp	uble-click on e Unversio Extension .pp	file for diff): ned Versior Status Modified	ned Added De Lines added 1	eleted Modified Lines removed 1	Files Submodules
anges made (dou heck: All Non Path ? 🚯 myunit.pp Show Unversion	uble-click on e Unversio Extension .pp ned Files	file for diff): ned Versior Status Modified	ned Added De Lines added 1	eleted Modified Lines removed 1	Files Submodules

Figure 4: Committing the first part git branch myfeature

This will create a new branch at the currently checked out commit, which is probably the standard case.

But if you want to create a branch at an earlier (or later) commit, you can instruct **Git** to do so:

git branch myfeature 5b10ccbfdcf73f88a1dd790346e73563788314dd



BRANCHING AND PARTIAL COMMITS PAGE 9/17



You can also use the short form of the hash:

git branch myfeature 5b10ccb

Note that at this point, the current branch has not yet changed: all the above commands do is create a new branch. To make this branch the active branch, the switch command can be used:

D:\FPC\myproject	t\myunit.pp ·	- Commit - T	ortoiseGit		-		×
nmit to: main		new	branch				
essage:							
Add MyBetter	Hash funct	ion					
Amend Last Co	mmit					1	/28
Set author date							
_ Set author date							
					Add Sign	ed. off. by	
Set author	uble-click on	file for diff):			Add Sign	ed-off-by	
Set author hanges made (doo Check: All Non Path	uble-click on e Unversion Extension	file for diff): ned Versior Status	ned Added Do Lines added	eleted Modified Lines removed	Add Sign Files Submo	ed-off-by	
Set author hanges made (doo Check: All Non Path M 🙀 myunit.pp	uble-click on e Unversion Extension .pp	file for diff): ned Versior Status Modified	ned Added Do Lines added 12	eleted Modified Lines removed 0	Add Sign	ed-off-by	
Set author hanges made (doo Check: All Non Path M () () ()	uble-click on le Unversion Extension .pp	file for diff): ned Versior Status Modified	ned Added Do Lines added 12	eleted Modified Lines removed 0	Add Sign	ed-off-by	
Set author hanges made (doo Check: All Non Path M R myunit.pp	uble-click on e Unversion Extension .pp ned Files	file for diff): ned Versior Status Modified	ned Added Do Lines added 12	eleted Modified Lines removed 0	Add Sign Files Subma 1 files select	ed-off-by odules ed, 1 files to	otal
Set author hanges made (dou heck: All Non Path R Myunit.pp Show Unversior Do not autosele	uble-click on e Unversion Extension .pp ned Files ect submodul	file for diff): ned Versior Status Modified	ned Added Do Lines added 12	eleted Modified Lines removed 0	Add Sign Files Submo 1 files select Vie	ed-off-by odules ed, 1 files to w Patch>>	otal
Set author hanges made (dou check: All Non Path Path Show Unversior Do not autosele Show Whole Pro	uble-click on e Unversion Extension .pp ned Files ect submodul oject	file for diff): ned Versior Status Modified	ned Added Do Lines added 12	eleted Modified Lines removed 0	Add Sign Files Submo 1 files select Vie	ed-off-by odules ed, 1 files to w Patch>>	otal

Now, every change you make and commit will be part of the branch 'myfeature'. The second way to create a branch uses a variation of the switch command:

git switch -c myfeature



BRANCHING AND PARTIAL COMMITS PAGE 10/17

The -c switch instructs **Git** to create the branch and immediatly makes it the active branch. If we now commit something in the branch, then git will show us that the branch was updated:



> git commit -m '* Added text 1' README.md
[myfeature 5836f28] * Added text 1
1 file changed, 1 insertion(+)

We can add a second commit:

> git commit -m '* Added text 2' README.md
[myfeature 57a7dbb] * Added text 2
1 file changed, 2 insertions(+)

To create a branch in TortoiseGit happens using the context menu, shown in *figure 6 on* page 11. In the dialog that appears then (*see figure 9 on page 11*), you can enter the details for the new branch:

- The name of the new branch.
- 'Base on' here you indicate what commit the branch should start: the default is HEAD on the current branch, which the name Git uses for the current commit. But you can also specify the tip of an existing branch, a tag (*which is just a symbolic name for a specific commit*), or a specific commit.

🐴 Git Sync	🗼 Resolve
🕐 Git Commit -> "main"	S Revert
Git Check for modifications	🧹 Clean up
TortoiseGit >	Nitch/Checkout
New	Y Merge
Properties	 Create Tag

Figure 6: Tortoisegit 'Create branch' menu

• Whether Git should immediatly switch to the new branch.

Branch	myfeature			
ase On				
HEAD (main)				
OBranch	main		\sim	
◯ Tag			\sim	
◯ Commit			~	
ptions				
Track	Force	Switch to new branch		
escription				
				^
				\sim

Assuming you did not switch to the newly created branch yet, you can still do so after the branch was created. The Switch/Checkout context menu item can be used for this, and it will show the switch dialog as in *figure 8 on article page 11*



BRANCHING AND PARTIAL COMMITS PAGE 11/17

8		
		711
	<u> </u>	5

witch To			
Branch	myfeature		~
🔿 Tag			~
◯ Commit			~
Overwrite wor Track	king tree changes (force)	☐ Merge	
Track			
Override brand	:h if exists		
	ОК	Cancel	Help

Figure 8: Tortoisegit 'Switch/Checkout' dialog

MERGING BRANCHES

Once you are done with the new feature or bugfix on your branch, it is time to merge your changes back into the main development line.

This is done with the Git merge command. This command will take all changes in a branch and merge them into the current branch. The following 2 commands demonstrate this:

```
> git switch main
Switched to branch 'main'
Your branch is up-to-date with
'origin/main'.
> git merge myfeature
Updating 324ccf6..57a7dbb
Fast-forward
README.md | 3 +++
1 file changed, 3 insertions(+)
home: ~/source/articles/git/demo (main)
git merge myfeature
```

The output Fast-forward means that **Git** was able to apply your changes as-is: **Git** stores all commits as diffs between one version and the next of a file. When merging, **Git** attempts to replay all the diffs of the branch you want to merge to the current branch: when the current branch does not have any commits that the branch you're trying to merge has not, then this is called a fast-forward.

This situation is depicted in *figure 9 on page 11*: when branch Bugfix 1 is merged into branch Releases, git will perform a fast-forward. When the fast-forward is done, the history will look like figure 10 on article page 12:





BRANCHING AND PARTIAL COMMITS PAGE 12/17



Figure 10: Fast-forward merged situation

The same is true if you do a git pull: If your local repository does not have any commits that a remote repository has, then git will perform a fast-forward: a pull operation is simply fetching all commits from a remote branch, and then merging them into the locally checked-out branch.

To perform a merge in **TortoiseGit**, the 'Merge...' menu item must be chosen from the **TortoiseGit** popup menu (*figure 11 on article page 12*. When chosen, the Merge dialog appears (*figure 12 on page 13* which offers you a choice which branch to merge, plus some options to observe when merging.



Squash

When you check this, all commits will be squashed into 1 big commit: if you're not interested in the commit history of the branch, but just the result, you can select this. On the command-line you can use -squash.

Messages

the number of messages to include in a merge commit. On the command-line you can use -log=N.

No Fast Forward

will force a merge commit, even if the merge can be performed with a fast-forward. On the command-line you can use -no-ff

Fast Forward Only

only do the merge if the merge can be performed as a Fast-Forward -ff-only

The command-line has more options than shown here, but it would lead too far to discuss them all.

If you do not specify any options, Git will attempt a fast-forward, but if it fails, it will create a merge commit to perform the merge. TortoiseGit will show the result of the operation in the merge result dialog, depicted in figure 13 on article page 13. Note that this dialog offers to delete the merged branch: if the branch is no longer needed, it's a good idea to remove it.



BRANCHING AND PARTIAL COMMITS PAGE 13/17

D:\FPC\myproject - I	Merge - TortoiseGit				×	V	Bir
From							
Branch	myfeature			~			
◯ Tag				\sim			
🔾 Commit				×			
Option Squash		Messages	20				
No Fast Forward		Fast Forward Only	r.				
No Commit							
Strategy	~		~				
Merge Message							
<auto generated<="" td=""><td>by Git></td><td></td><td></td><td></td><td></td><td></td><td></td></auto>	by Git>						
		ОК	Cancel	Help			
	and the last						
Bure 12: Ionoisegit I D:\FPC\myproje	ect - Git Commar	nd Progress - Tortoise	Git		- [ı ×	
		-					
<u>99</u>							
git.exe merge	myfeature						
	ca74f84bc						
Updating 05e19 Fast-forward							
Updating 05e19 Fast-forward README.md 2	++ 2 incontion						
Updating 05e19 Fast-forward README.md 2 1 file changed	++ , 2 insertion	IS(+)					
Updating 05e19 Fast-forward README.md 2 1 file changed Success (125 m	++ , 2 insertion s @ 16/10/202	15(+) 1 17:22:01)					
Updating 05e19 Fast-forward README.md 2 1 file changed Success (125 m	++ , 2 insertion s @ 16/10/202	ns(+) 21 17:22:01)					
Updating 05e19 Fast-forward README.md 2 1 file changed Success (125 m Remove	++ , 2 insertion s @ 16/10/202 branch	15(+) 21 17:22:01)	C	lose	Ab	ort	



BRANCHING AND PARTIAL COMMITS PAGE 14/17

G REBASE

There are developers that do not like the large trees of branches. The situation in *figure 1* on page 1 is actually quite simple, but over time, having many people working on a project simultaneously can create a tangle of branches and commits, for an example see figure 14 on page 15.

By contrast, a situation like *figure 10 on page 12* where a merge can be done without merge commits is more simple, it looks like a linear history, although in reality the commit history is not linear.

In fact, such a situation can be achieved with a git command: **Git** stores commits as diffs. When merging the diffs from a branch are replayed on the branch you're merging into. The use of diffs suggests a mechanism to work towards a 'fast-forward' scenario in the commit history: This mechanism is implemented using the rebase command. What does it do?

The rebase command must be executed in the branch you want to merge, for example in the Bugfix 1 branch as shown in *figure 15 on of article page 16*: this depicts the same repository as *figure 1 on article page 1*, but without the merge commit 6. You must tell the rebase command what branch you want to rebase on (Releases in the case of the figure).

When the rebase command is executed, it will restore the situation of commit 2, apply all the commits in the Releases branch, and rewrite the commits in the BugFix 1 branch. A-C in such a way that they appear to have been applied after commit 5 in the Releases branch.

What the Git rebase command does, looks like this (the releases branch is called 'main' in the example): > git switch Bugfix_1

Switched to branch 'Bugfix_1'
> git rebase main
First, rewinding head to replay your work on top
of it...
Applying: * MyBetterHash2 start
Using index info to reconstruct a base tree...
M myotherunit.pp
Falling back to patching base and 3-way merge...
Auto-merging myotherunit.pp
Applying: * Implement MyBetterHash2

This shows exactly what is being done: it rewinds to the branch point, applies the patches from the main branch and then applies the patches from the Bugfix_1 branch.

Now, when the Bugfix_1 branch is merged to the Releases branch, **Git** can perform a Fast-Forward, and the end result will be a more linear history, as shown in *figure 16 on article page 16*.

It is important to note that the rebase command can fail: when incompatible changes are encountered in the two branches, the process will stop when applying a patch that fails: In that case, you must manually fix the incompatibilities, add the changed file to the staging area and then execute the rebase command again with the -continue option: git will then resume the rebase operation and apply the remaining patches.

To do this in **TortoiseGit**, the 'Rebase' menu item must be chosen from the context menu, as in *figure 17 on article page 16*.



git

BRANCHING AND PARTIAL COMMITS PAGE 15/17



https://blog.carbonfive.com/always-squash-and-rebase-your-git-commits/)

Figure 14: Git tangled branches

When this menu item is chosen, the dialog in *figure 18 on article page 17* is shown. In this dialog, you must select the branch to rebase (*top left*), and the current branch is selected. At the top-right you must select the branch on which to rebase: the 'Onto' button will open a dialog showing a tree of branches. You can also select one from the dropdown list.

The list of commits that will be rebased is then shown, and they are all selected (selected commits have the word PICK before them). You can, if you so desire, using the buttons below the list rearrange or skip some commits. For instance commits that are known to result in conflicts can be removed in this way.

The 'Start rebase' button can be used to start the rebasing process (*remember, it can fail*): The log of what git is doing will then be shown in the bottom half of the dialog. When all has completed successfully, the result will look for example like in *figure 19 on article page 17*.

As noted earlier, branches in a remote repository are branches like any other, and doing a 'git pull' is like doing a merge from a remote branch: the story of entangled branches and a lineair history as explained here with purely local branches, also applies across repositories.





In order to get a lineair commit history, we must enable **Git** to always do a merge from the remote repository using a fast-forward.

To ensure this, you can tell Git to rebase your local changes on top of the upstream branch when doing a git pull. The option to do this is called -rebase, so the command becomes:

git pull --rebase

In **TortoiseGit**, there is an option in the 'git push/pull' dialog to achieve the same effect.

6 CONCLUSION

In this and the two previous articles, we've explained the basic operation of git: basic operation of git is not very different from SubVersion, but Git provides more possibilities, more than we've treated in this article. The Internet is however full of resources that treat git in any desired level of depth.



Figure 17: TortoiseGit rebase menu



BRANCHING AND PARTIAL COMMITS PAGE 17/17

D:\FPC\myproject - Re	ebase - TortoiseGit				- 0	×
anch: Bugfix_1	``	🗸 Upstream: ma	ain	~	0	nto
BASE ID ck 2 ck 1	SHA-1 Messay cca7cec Bugfix c63eda9 * MyBe	ge 1 * Implement MyBetterHa etterHash2 start	Author sh2 Michael Michael	Date 16/10/2021 19:0. 16/10/2021 19:0.		
th Extension Sta	tus Lines added	Lines removed		lerges L		
		Start R	lebase	Abort	Help	
re 18: TortoiseCit re	base dialog	_				
D:\FPC\myproject - R						
	ebase - TortoiseGit				- 0	×
Branch: Bugfix_1	ebase - TortoiseGit	V 🗶 Upstream: ma	ain	~		× Into
Bugfix_1 REBASE ID Vick 2 Vick 1	SHA-1 Messag cca7cec * Imple c63eda9 * MyBe	ge ement MyBetterHash2 etterHash2 start	Author C Michael 1 Michael 1	Date 6/10/2021 19:0 6/10/2021 19:0		X
ranch: Bugfix_1 REBASE ID Pick 2 Pick 1 Pick 1 Pick ALL	SHA-1 Messag cca7cec * Imple c63eda9 * MyBe	Upstream: ma ge ement MyBetterHash2 etterHash2 start Down Add	ain Author E Michael 1 Michael 1 Preserve me	✓ Date 6/10/2021 19:0 6/10/2021 19:0 erges	- C	× Into
Branch: Bugfix_1 REBASE ID Pick 2 Pick 1 Pick 1: c63eda95a03 MyBetterHash2 sta uto-merging myothed detached HEAD df44 Date: Sat Oct 16 3 1 file changed, 6	SHA-1 Messag cca7cec * Imple c63eda9 * MyBe Up Up 31fa0fb4618ea6246i art erunit.pp 4a58] * MyBetterHi 19:00:52 2021 +020 insertions(+)	y terream: ma ge ement MyBetterHash2 etterHash2 start Down Add a11d2557cb835 ash2 start 00	ain Author E Michael 1 Michael 1	✓ Date 6/10/2021 19:0 6/10/2021 19:0 erges	- C	×
Image: State of the state	SHA-1 Messag cca7cec * Imple c63eda9 * MyBe Up Up 31fa0fb4618ea6246a art erunit.pp 4a58] * MyBetterHi 19:00:52 2021 +020 insertions(+) 3f1e55b2d273ebe934 erunat.pp 1707] * Implement 19:01:44 2021 +020 insertions(+)	y the set of the set o	ain Author E Michael 1 Michael 1	Date 6/10/2021 19:0 6/10/2021 19:0 erges	Force Rebase	×
Branch: Bugfix_1 REBASE ID Pick 2 Pick 1 Pick 1 Pick 1 Pick 4 Pick 4 P	SHA-1 Messag cca7cec * Imple c63eda9 * MyBe w Up Up 31fa0fb4618ea6246a art erunit.pp 4a58] * MyBetterHi 19:00:52 2021 +020 insertions(+) 3f1e55b2d273ebe93a erHash2 erunit.pp 1707] * Implement 19:01:44 2021 +020 insertions(+) f -B Bugfix_1 4b3: rd 4b31707741dc890	Upstream: ma ge ement MyBetterHash2 etterHash2 start Down Add alld2557cb835 ash2 start 00 48307a9f5e898 MyBetterHash2 00 1707741dc89daa14ab2be5d	ain Author C Michael 1 Michael 1 Preserve me	Date 6/10/2021 19:0 6/10/2021 19:0 erges	- C	

Figure 19: TortoiseGit rebase dialog showing the result of the operation



SUMMERSALE



<section-header><section-header><text><text><text>

Including the PDF and Code Examples

electronic index

https://www.blaisepascalmagazine.eu/product/lazarus-handbook-pocket/

We have a new service at our website, for shipping you can make three choices: 1.The shipping cost depending on which part of the world you want the book to be shipped: Europe or the other countries: the World

(·······································		[[······	······································
	SHIPPING COST Europe	SHIPPING COST Europe + Registered	SHIPPING COST Europe +	SHIPPING COST World	SHIPPING COST World + Registered	SHIPPING COST World + Track & Trace
1					[mannanana]	000000000000000000000000000000000000000



RASPBERRY PI APPS WITH DELPHI

PAGE 1/6

A new approach to create Raspberry Pi apps with Delphi via MILETUS By Bruno Fierens

MILETUS

In **TMS WEB Core v1.7**, we introduced the **Miletus** technology for the first time as a technology to create cross-platform desktop applications from the **Delphi** IDE based on web technology. So far, with **Miletus**, it was possible to create **Windows** (Win32 & Win64), macOS (Intel & ARM) and **LINUX 64** bit applications from the same codebase. Now, from **TMS WEB CORE V1.9**, it is possible to generate applications from the **Delphi** IDE that can run on a **Raspberry Pi** using the official Raspberry Pi OS. https://arstechnica.com/ information-technology/ 2013/03/how-two-volunteers-built-the -raspberry-pis-operating-system/



GETTING STARTED

To create a **Raspberry Pi** app

from the **Delphi IDE**, follow File | New | Other

and under TMS WEB, you find the application type TMS Miletus app.

After the IDE created the default application, you can start adding your code to the application in pretty much the same way as you would do for a **VCL application or FireMonkey** application.

STREET, STREET

The components you can use are the same components as for a regular **TMS WEB Core** web client application, i.e. the TWEB* components.



RASPBERRY PI APPS WITH DELPHI PAGE 2/6

A new approach to create Raspberry Pi apps with Delphi via MILETUS



After you added the code to your application, you can select from the build targets, the **Raspberry Pi** target. Just compile either in **Raspberry Pi** release mode or **Raspberry Pi** debug mode and the executable will be generated for you.

Project1.dproj - Projecto

in operation of the second sec	
iii × III II C 🖶 0 × ⊙ × ···· ×	# ×
The ProjectGroup 1	
✓	
Build Configurations (Build-Raspherey-Pa)	
Debug	
🌣 Release	
🕸 Build-Win32	
🛱 Build-Win64	
🗱 Build-Linux64	
Build-MacOS64-Intel	
Duild-MacOS64-ARM	
Build-Raspberry-Pi	
🔅 Debug-Win32	
🔅 Debug-Win64	
🔅 Debug-Linux64	
🔅 Debug-MacOS64-Intel	
Debug-MacOS64-ARM	
🛱 Debug-Raspberry-Pi	
> O Target Platforms (Windows 32-bit)	
MiletusLinux.png	
MiletusMac.icns	
MiletusWindows.ico	
Project1.html	
> 🔂 Unit1	



RASPBERRY PI APPS WITH DELPHI PAGE 3/6

A new approach to create Raspberry Pi apps with Delphi via MILETUS

XCOPY DEPLOYMENT

Deploying the applications on your **Raspberry Pi SBC** can't be easier. Just **XCOPY** the executable and the desktop link file to the **Raspberry Pi** and you can start using it.

RASPBERRY PI SPECIFIC COMPONENTS

The **Raspberry Pi** is not only interesting because it is so cheap, it is mostly interesting as it offers various ways to extend it with hardware connected to it. So, it is kind of expected that one can take advantage of the interfaces on board. This includes at this moment the **GPIO**, **i**²**c** ports, the **ISP** ports and the **UART** port (Universal Asynchronous Receiver/Transmitter).



GPIO

The **Raspberry Pi** offers in total **29 GPIO** pins (practically 27 **GPIO** as there are 2 dedicated pins for advanced use). A **GPIO** pin is a pin that can be configured to be used as input or as output pin.

TMiletusRaspberryGPIOMode = (gmRead, gmWrite); function GPIOConfig(AGPIOPin: Integer; AMode: TMiletusRaspberryGPIOMode): TJSPromise; function GPIOWrite(AGPIOPin: Integer; Avalue: Integer): TJSPromise; function GPIORead(AGPIOPin: Integer): TJSPromise;

So, we can use a **GPIO** pin to configure it as an output and control a **LED** connected to it from our **Miletus** application with the following code:

procedure TForm1.FormCreate(Sender: Tobject);

var

res: Integer; begin

// configure a pin as GPIO output pin

```
res := Await(Integer, GPIOConfig(GPIOPINNUMBER, gmWrite));
end;
```

and from a toggle button we can turn the LED on and off.

```
procedure TForm1.WebToggleButton1Click(Sender: TObject);
```

```
var
val: Integer;
begin
if WebToggleButton1.Checked then
val := 0
else
val := 1;
GPIOWrite(GPIOPINNUMBER, val);
```

```
end:
```

```
Blaise Pascal Magazine 99/100 2021
```



RASPBERRY PI APPS WITH DELPHI PAGE 4/6

A new approach to create Raspberry Pi apps with Delphi via MILETUS

Note that these are functions returning a promise as these are asynchronous functions. Fortunately, using the await() for the promise returning function, we can still write code in a sequential way.

i²C

I²C is a standard 2 wire protocol to communicate with devices. The **Raspberry Pi** offers out of the box two **I²C** interfaces. There is **SCLK** and **SDA** signal, i.e. the clock signal and the data signal that is open-collector based input / output. Communicating over **I²C** works via sending first an 8bit address and read or write bit over the **SDA** signal and then either write or read data bits. We exposed this in the **TMiletusRaspberryI2C** component via methods:

General open / close methods:

TMiletusRaspberryI2C.Open;
TMiletusRaspberryI2C.Close;
Read data methods
TMiletusRaspberryI2C.ReadByte(Address:byte): TJSPromise;
TMiletusRaspberryI2C.ReadSmallInt(Address:byte): TJSPromise;
TMiletusRaspberryI2C.ReadBuffer(Address:byte, Size: integer): TJSPromise;
Write data methods
TMiletusRaspberryI2C.WriteByte(Address: byte; AData: byte);
TMiletusRaspberryI2C.WriteByteP(Address: byte; AData: byte): TJSPromise;
TMiletusRaspberryI2C1.WriteAddress(Address: byte);
TMiletusRaspberryI2C1.WriteAddressP(Address: byte): TJSPromise;
TMiletusRaspberryI2C1.WriteBuffer(AData: Array of byte; ASize: integer);

TMiletusRaspberryI2C1.WriteBufferP(AData: Array of byte; ASize: integer): TJSPromise

Just like with the **GPIO** functions, also here it concerns asynchronous functions returning a promise that can be used with await() to still write the code in a sequential way. Here is a code snippet that shows how we can read calibration values from a sensor from 3 registers at addresses specified.

var ac1,ac2,ac3: integer; begin ac1 := await(Integer, MiletusRaspberryI2C1.ReadSmallInt(\$AA)); ac2 := await(Integer, MiletusRaspberryI2C1.ReadSmallInt(\$AC)); ac3 := await(Integer, MiletusRaspberryI2C1.ReadSmallInt(\$AE)); end;



A new approach to create Raspberry Pi apps with Delphi via MILETUS

SPI

The **SPI** protocol (Serial peripheral interface) uses a 3-wire connection, a clock and the data-in and data-out signal. Other than this, it is similar to **I²C**, meaning, it also uses an address and reads and writes data serialized over this 2 data wires.

The methods for the TMiletusRaspberrySPI component are:

TMiletusRaspberrySPI.Open; TMiletusRaspberrySPI.Close; TMiletusRaspberrySPI.ReadTransfer(**var** ABuffer: TBytes; AWSize: integer; ARSize: integer): TJSPromise; TMiletusRaspberrySPI.WriteTransfer(ABuffer: TBytes; AWSize: integer): TJSPromise;

As you can see, also the **SPI** protocol works over asynchronous functions. Reading and writing over **SPI** to a piece of hardware connected to the **Raspberry Pi** can be done via the **ReadTransfer()** and **WriteTransfer()** calls where a buffer of bytes is sent or received.

UART

Out of the box, the **Raspberry Pi** is also equipped with the good old serial port, or called **UART.** For serial communications, the baud rate, the bit count and parity can all be set via properties of TMiletusRaspberryUART. To read and write data, the TMiletusRaspberryUART offers methods:

MiletusRaspberryUART.Open;

MiletusRaspberryUART.Close;

MiletusRaspberryUART.ReadBuffer(var ABuffer: TBytes; ALength: integer): TJSPromise; MiletusRaspberryUART.WriteBuffer(ABuffer: TBytes; ALength: integer): TJSPromise;

In this code snippet, it is demonstrated how a text can be written to a **LED** strip via the **UART**:

Var buf: TBytes; begin // create a buffer of bytes from a string buf := GetTextBuffer('Hello world'); await(Integer, MiletusRaspberryUART1.WriteBuffer(buf, Length(buf))); end;

The GetTextBuffer() call is a call that formats the bytes in a specific way the LED strip driver expects and is thus LED strip hardware specific.



PUTTING THE PIECES TOGETHER

Now, with the **Delphi IDE**, **TMS WEB Core** with **Miletus Raspberry Pi** support and added to that the **TMS FNC** components, we can explore new territories as **Object Pascal** developers to create new types of applications running on the **Raspberry Pi OS** desktop and talking to the connected hardware. In our lab, we created such sample project where a **Bosch** sensor is connected to the **Raspberry Pi**. The **Bosch BPM180** sensor can retrieve the air pressure and temperature over i²c. In addition, we connected a **LED** strip via the **UART** to the **Raspberry Pi** and wrote a little project that reads our the air pressure and temperature from the **Raspberry Pi** and shows the air pressure in an **FNC** chart on the screen and the temperature in an **FNC** gauge control. At the same time, we also visualize the measured values via the

LED strip. You can see the app running on



and the hardware with which this was realized in this picture:



SUMMARY

A whole new world of possibilities open up for Object Pascal developers. We can create apps that run on a low-cost Raspberry Pi SBC via Miletus, we can access optionally connected hardware to the device and we can create stunning Uis with HTML5/CSS3 and even combine this with a whole set of sophisticated TMS FNC components. We are curious to see how you will apply your creativity to bring exciting new solutions!



Advertisement



Hint: Click on the page

- 1. One year **Subscription**
- 2. The newest LIB Stick
 - including Credit Card USB stick
- 3. Lazarus Handbook Personalized -PDF including Code

- 4. Book **Learn To Program** using Lazarus PDF including 19 lessons and projects
- 5. **Book Computer Graphics Math & Games** book + PDF including ±50 projects

https://www.blaisepascalmagazine.eu/product/ bundle-computer-graphics-math-games-pascal-libstick-download-subscription/

ADVERTISEMENT





Summersale Subscription + Lazarus Handbook (hardcover) € 100 Ex Vat 9%

https://www.blaisepascalmagazine.eu/product/lazarus-handbook-hardcover-subscription/

SPEAKING-SPORTS-CLOCK

By Detlef Overbeek

D11 starter



PAGE 1 / 17

ABSTRACT

expert

I created this Speeking-Gymnastic-Clock because I wanted to be able to be reminded in a certain time span to run up, or down the stairs to do my gymnastics and than back to work: Sit!

All of us have the same problem, so I thought of making this **Speeking-Gymnastic-Clock** in a way that it can handle events: You can not only set the time but also enter a text which will be transformed to speech, or just play a short song or sound at a specified time or arrange it according to your own needs.

INTRODUCTION

To be able to do all this we need several components of which one is very new to me: Text to speech.

That means that you can enter a text in a file and then let it be read by a component that will transform it to sound. I didn't find a good one and asked TMS Software if they had anything like this. It is available for browsers, so with an API it might be transformed to make it work on the desktop as well. TMS (Bruno Fierens) was immediately interested and created a Component that is capable of doing this for multiple OS's.

Furthermore I needed to do design Time-Event handling which is rather complex. Once it is explained, you can understand the working of it and reuse it for your own sake. The text to speech component, and the clock-component is available at TMS Software and you can also get the components in a trial pack.

The other component is the Color button which is freely available. I chose the clock of TMS because the clock which is available from the Jedi Component Library is small and does not have the ease to use as the TMS clock. It even looks better. Now the EventClock consists of 4 sections: A digital clock, an Analogue clock, an Agenda and finally Text to Speech component.

The Speeking-Gymnastic-Clock is based on Desktop-Clock Agenda that I created before. All the project code is available for subscribers: https://www.blaisepascalmagazine.eu /your-downloads/



Figure 1: At the top the Analogue clock Figure 2: The digital version



SPEAKING-SPORTS-CLOCK



COMPONENTS USED:

Of course you can run the project in earlier versions of **Delphi**.

It depends very much on the components you use. If you simply want the to try you could use **Community Edition of Delphi 10.4** in combination with the **kbmMW Community Edition**. But the **MemTable Standard** is very inexpensive. (only \in 30,00) There is also a **Pro** version, which covers a lot of extra's. You the need to buy then the **kbmMW development toolkit** for **Delphi and C++Builder** for building native modular and fast multi-tier solutions professional or the enterprise version(*See the advertisement on the last page of this issue*.) So you can if you want to compile the project do that at no cost.

You are free to use and alter this application (EventClock) for your own use.

I will create after this issue a version for Lazarus and than it would become available for all Os's : Windows / Linux / Mac as well FMX, Android and iPhone.

The components are here available:

The clock and the Text to speech component: **TTMSFNCWXSpeechSynthesis** in the group **FNC** https://www.tmssoftware.com/site/tmsfncwxpack.asp

TMS Software (as try or by)

Color Button: Available as component in your download section: (You must have a subscription) https://www.blaisepascalmagazine.eu/your-downloads/

kbmMemTable: (for the ClientDataSet)

http://www.components4developers.com/products_kbmMemTable.html

GOALS:

Create Time events which will be set at a specific time Create Time events that are repeated in a time span Make it possible to enter your own text for transforming to speech Let it play a sound or music





Figure 3: Speaker list

Figure 4: The popup menu



1. THE CALENDERFC



PAGE 3 / 17

Figure 5 The Month calendar form

LblTime 🗶 Labe test Labe LblDate	IS ColorDialog1pupMenu1Timer1	Clock	BLAISE PASCAL MAGAZINE 98 2021		Cla	ickGUI 🗙
October 2021 Mon Tue Wed Thu Fri Sat Sun 7 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 1 2 3 4 5 6 7 10 32 34 5 6 7 8 9 30 31 1 2 3 4 5 6 7 8 29 30 31 1 2 3 4 5 6 7 7 8 3 31 1 2 3 4 5 6 7 32 3 3 3 3 3 3 3 3 34 5 6	October October <t< th=""><th>October 2021 ► Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7</th><th>October 2021 Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 11 2 3 34 5 26 27 28 29 30 31 1 2 3 4 5 6 7 1 2 3 4 5 6 7 20 20 30 31 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 <t< th=""><th></th><th>Timer PopupAction</th><th>Barl Timer2</th></t<></th></t<>	October 2021 ► Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7	October 2021 Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 11 2 3 34 5 26 27 28 29 30 31 1 2 3 4 5 6 7 1 2 3 4 5 6 7 20 20 30 31 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 <t< th=""><th></th><th>Timer PopupAction</th><th>Barl Timer2</th></t<>		Timer PopupAction	Barl Timer2
October 2021 Image: Stress of the stress of th	Mon Tue Wed Thu Fr Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 52 26 27 28 29 30 1 1 2 3 4 5 6 7 October 2021 A 5 6 7	Mon Tue Wed Thu Fil Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 7 28 29 30 31 1 2 3 4 5 6 7 October 2021 A 5 6 7	Mon tue Wed Thu Fit Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 28 29 30 1 2 3 24 5 6 7 20 21 22 23 24 15 16 17 19 20 21 28 29 30 1 1 2 3 4 5 6 7 October 2021 2 28 24 30 3 4 5 6 7	Figure 6: The Aanlog	Appointment 60 30 Open Speech Label1 Action or Appoint Para	15 1 X orm
Mon Tue Wed Thu Fit Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 72 28 29 30 31 1 2 3 4 5 6 7 October 2021 A 5 6 7 50	Mon Tue Wed Thu Fit Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 7 28 29 30 31 1 2 3 4 5 6 7 October 2021 - 26 27 30 31	Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 7 28 30 31 1 2 3 4 5 6 7 1 2 3 4 5 6 7 20 21 22 24 24 24 24 24 24 24 25 6 7 26 29 30 31 1 2 3 4 5 6 7 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 25	Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 CalendarGUI ClockGUI FrmData Change form color DateEvent TimeEvent E	dataGUI × Delete all appointments ventName	- K < Þ þ + - A	□ × ✓ × ⊘
Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 20 21 22 23 24 35 6 7 1 2 3 4 5 6 7 8 9 30 31 1 2 3 4 5 6 7 8 6 7	Mon Tue Wed Thu Fri Sat Sum 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7	Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7	•	ColorDialog1		
START: First of all let us forms. I named	create the nece them:	ssary	¢			>
 Calendard ClockGUI. DataGUI.p MessageGU 	GUI.pas - Cale pas - ClockFr pas - FrmData JI.pas - Messa	ndarFrm n geFrm	CalendarGUI	ClockGUI dataGUI	Figure 7: The Da	ata form
			Type text Speak Ed	it Text Save voice selection	Stop the Form Speak Memo text	
			Pau	ISE Resume	Cancel	

Figure 8: The Speaking Message Form





PAGE 4 / 17

1 THE CALENDERFORM (Calendargui.p

How to use it:

The calendar form is rather complex to make it possible to be sized as very small item, that can collapse if not need otherwise. It has the possibility of showing 1 month or even 16 months. I did this to give the opportunity to have a view on eventual overlapping dates over 1.1/4 years.

On this form is a **Digital clock** created and it has some extra settings:

You can reach the Analogue Clock by pressing the Analogue Clock Button or by clicking on the Blaise Icon at the right top. It shows the **Digital Clock** as well the Analogue Clock. That's done on purpose: the digital clock gives you basic and different information.

You can enable the 1 month form, plan events and save them (See figure 9).

You can even change the forms colour and that is finally set in an ini-file so it remembers the setting next time you start it. That is a basic idea: anything you want to change can be set in an inifile. We need for this for one timer and a Pop-Up menu:

that has a list of events you can trigger or set:

8:14X 1

Start Analoque & Digital Clock Start AnalogueClock Only

Calendar 1 month Calender Year (16 months) Date only, no border

Form view, move the form

Appointment Form

Close App

000000 Form view, move the form

Calendar 1 month

Start the Analogue & Digital clock

Start the AnalogueClock Only

- Appointment Form
- Close App

Figure 9: The popup menu of the Digital Clock

15:43:58 🗙 Sunday 24/10/2021	Analo	gue Clock	BLAISE PASCAL MAGAZINE 98 2021	Ċ
October 2021 🔺 🗸	November 2021 🔺 🗸	December 2021 🔺 🗸	January 2022	^ ~
Mon Tue Wed Thu Fri Sat Sun	Mon Tue Wed Thu Fri Sat Sun	Mon Tue Wed Thu Fri Sat Sun	Mon Tue Wed Thu	Fri Sat Sun

In this form there is code to create the digital clock and to set the days of the week, so it automatically sets the correct date: see next page \rightarrow

Figure 10: opening the month form



in this code section you can see how the days settings are arranged (See annotation)

procedure TCalendarFrm.FormCreate(Sender: TObject); Var day :array[1..7] of string; begin // set all necessary paths // Dir:=GetFolderPath(0,CSIDL_LOCAL_APPDATA)+ 'Delphi_D11_C4D'; Dir:= ExtractFilePath(Application.ExeName); /// to show the path you can enable them Label3.Caption:=Dir;

Label4.Caption:=GetShellFolder(CSIDL LOCAL APPDATA);

Label5.Caption:=Dir; **If not** ForceDirectories(Dir) **then Raise** Exception.CreateFmt('Failed to create settings directory %s',[Dir]);

FN := Dir +'\'+'C4D_RIO_appINI.ini'; kbm := Dir +'\'+'Calendar_kbmMW_Binary.detlef'; CL := Dir +'\'+'AppColor.ini';

///This is all for the smalleste form size
LblTime.Caption:= TimeToStr(Now);
LblDate.Caption:= DateToStr(Now);

day[1] := 'Monday'; day[2] := 'Tuesday'; day[3] := 'Wednesday'; day[4] := 'Thursday'; day[5] := 'Friday'; day[6] := 'Saturday'; day[7] := 'Sunday';

LblWeekday.Caption:= day[DayOfTheWeek(Now)]; /// DayOfTheWeek is ISO 8601 compliant, /// since it uses Monday as the start of the week.

```
///To set all calendars to the tight months
CV1.Visible := False; CV1.Date := Now;
CV2.Visible := False; CV2.Date := incMonth(Date, 1);
CV3.Visible := False; CV3.Date := incMonth(Date, 2);
CV4.Visible := False; CV4.Date := incMonth(Date, 3);
CV5.Visible := False; CV5.Date := incMonth(Date, 4);
CV6.Visible := False; CV6.Date := incMonth(Date, 5);
CV7.Visible := False; CV7.Date := incMonth(Date, 6);
CV8.Visible := False; CV8.Date := incMonth(Date, 7);
CV9.Visible := False; CV9.Date := incMonth(Date, 8);
CV10.Visible := False; CV10.Date:= incMonth(Date, 9);
CV11.Visible := False; CV11.Date:= incMonth(Date,10);
CV12.Visible := False; CV12.Date:= incMonth(Date,11);
CV13.Visible := False; CV13.Date:= incMonth(Date, 12);
CV14.Visible := False; CV14.Date:= incMonth(Date,13);
CV15.Visible := False; CV15.Date:= incMonth(Date,14);
CV16.Visible := False; CV16.Date:= incMonth(Date,15);
```

The code continues on the next page \rightarrow





```
/// Activate the databse
 kbmMemTable1.Open;
 kbmMemTable1.Active;
 /// Set the name of the file where the data in is recorded
 /// These database e;lement are fre available
 kbmMemTable1.LoadFromFile(kbm);
 /// Free gift from Kim Madsen
 ReportMemoryLeaksOnShutdown:=true;
 /// All the settings are recorded in C4D_Rio_appINI
 C4D RIO appINI := TIniFile.Create(ChangeFileExt(FN,'.ini'));
 try
 Top := C4D RIO appINI.ReadInteger('Placement','Top', Top);
 Left := C4D_RIO_appINI.ReadInteger('Placement','Left', Left);
 Width := C4D RIO appINI.ReadInteger('Placement','Width', Width);
 Height := C4D_RIO_appINI.ReadInteger('Placement', 'Height', Height);
 finally
 C4D RIO appINI.Free;
 end;
 AppColor := TIniFile.Create(ChangeFileExt(CL,'.ini'));
 try
 Color := AppColor.ReadInteger('FormColor','Color', Color);
 finally
 AppColor.Free;
 end;
 /// All the settings are recorded in C4D_RIO_appINI
 If ClockFrm = Nil Then ClockFrm := TClockFrm.Create(Self);
 //ClockFrm.Show;
 Height := 70;
 Width := 210;
 Image1.Left := 155;
 if left +215 >= screen.Width
 then left := screen.Width - 240
 else
 If left +215 <= screen.Width</pre>
 then left:=Left;
End
```



SPEAKING-SPORTS-CLOCK 1. THE CALENDERFORM



In this code example there is shown how to set the path for use for ini files or for generally valid settings:

the path set tlike this (*1) Dir:=GetFolderPath(0,CSIDL_LOCAL_APPDATA)+ 'Delphi_D11_C4D';

will refer to something like this:

c:\Users\edito\AppData\Local\Delphi_D11_C4D\

C4D_DI1_appINI.ini AppColor.ini C4D_RI0_appINI.ini Calendar kbmMW Binary.detlef

Where this setting (*2) Dir:= ExtractFilePath (Application.ExeName); will lead you to path ...\ClockAgenda_Txt2Speech\Delphi_11_C4D.exe it is more variable way and it has very different advantages.

*1 is putting the items in more hidden way

- so it is less easy for the eventual user to change or damage it.
- its centralized on every version of Windows.
- But what's to be sure? Windows 11 wasn't expected at all...

*2 If you want to be able to place it in any path on whatever drive you want, I would suggest the second version.

The ini file is put in your project and can be seen immediately not needing to find out where in havens name it might be placed.

procedure TCalendarFrm.FormCreate(Sender: TObject); Var day :array[1..7] of string; begin

/// set all necessary paths

// Dir:=GetFolderPath(0,CSIDL_LOCAL_APPDATA)+ 'Delphi_D11_C4D'; is an other option
Dir:=ExtractFilePath(Application.ExeName);

How to use it:

II.



In this app we make use of the **kbmMemTable**.

In many other articles I have been writing about this.

(Issue 83) (Dataset Change Tool) - (Issue62) (Creating ToDo List)

THE CLOCKFORM (ClockGui.pas - ClockFrm)

If you open the Analogue clock you can choose from various a

A very especial value is that one can use in this **ClientDataSet**, the standard **SQL**. So it gives much better result than the filtering system that a normal **ClientDataSet** uses. The big difference is that you can use a binary file stream or a **CommaSeparatedValue** stream. You need to connect it with a **kbmBinaryStreamFormat** which has an enormous number of settings.

All the details of this can be found in the project it self, by clearly tracing the connections. The specific settings are explained in this issue: page 11/17 figure 13.

	Right click and a menu pops up:		
0	Close: (Double click) this is obvious it close the from but opens or reopens the Move Move means tat you will get a non-transparent form with can drag the form. If you want to return to the transpare and choose Transparency.	Digital cloo a top bar v nt form rig	ck. where you ht click
3	Transparency is set if the form was set to Move.		
4	Set Action, Appointment Here you can precise at a given time (your choice) plan y	<u>ou</u> r appoin	tment.
5	Message every minute Repeating every minute(for testing reasons).		
6	Message 30 minutes Repetition every half hour.	Clock	
7	Message 60 minutes (for testing reasons) Repetition every hour. Standard is set the Nightingale and a spoken message that you can alter or create in the "MessageGuiFrm". This is separately mentioned and explained.	Appointment 60 Open Spec 15:58:12	2 g- Close: (Doubleclick) Move Iransparency Set Action, Appointment Message every minute
8	Stop messaging shuts of the sound and messaging.		Message 15 minutes Message 30 minutes Message <u>6</u> 0 minutes

Figure 11: Move the Clockform



SPEAKING-SPORTS-CLOCK



PAGE 9 / 17

This menu is actually created to let you choose a setting for repeated events. You can create the text that will be spoken and eventually choose a sound or melody. For this reason I added a number of sounds which are hosted in the Dir ...\wav.

Malelminutecall.wav output_2021_09_18_11_58_47_0umz8pe.mp3 speech.mp3 speech.wav boop-boop2.wav NightingaleStory.wav pacman_intro.wav pacman_dies_y.wav pacman2_x.wav pacman_x.wav Teehee.wav shutter.wav Poppeye.wav

as you see it is not dependant on .wav files but other formats can be used as well. It uses the sndPlaySound(SoundStr, SND_FILENAME or SND_ASYNC); The use of it I will explain later. All the buttons have equivalent functions and speak for them selves. See the schema and picture in figure xx.

In the code you will see a section Timers.

// Timer1

procedure TClockFrm.TimerlTimer(Sender: TObject);
begin

sndPlaySound(SoundStr, SND_FILENAME or SND_ASYNC);

If MessageFrm = Nil Then MessageFrm := TMessageFrm.Create(Self);
MessageFrm.ClrBtnEditSpeakClick(Sender);

end;

// Timer2
procedure TClockFrm.Timer2Timer(Sender: TObject);
begin

Timer2.Enabled := false; sndPlaySound(SoundStr, SND_FILENAME or SND_ASYNC);

If MessageFrm = Nil Then MessageFrm := TMessageFrm.Create(Self);

end;

// Timer3
procedure TClockFrm.Timer3Timer(Sender: TObject);
begin

Label1.Caption := TimeToStr(Now)

end;



Figure 12: The components on the clock form



SPEAKING-SPORTS-CLOCK II. THE CLOCKFORM



PAGE 10 / 17

Timer1

handles the sound module and also activates the MessageFrm and sends an clickevent to the form for ClrBtnEditSpeakClick so the editlabel text will be executed and the message text is than altered to voice by the transformer TTMSFNCWXSpeechSynthesis. Details are explained in the section MessageForm.

Timer2

deactivates the timer and makes the sndPlaySound available It doesn't make the form visible but only activates it.

Timer3

creates the digital clock-string on a label, so can see the time setting of that moment.

```
procedure TClockFrm.MessageOneHourClick(Sender: TObject); {1h}
   var TriggerTime : TTime;
   begin
    Timer1.Enabled := True;
    TriggerTime := TTime(Frac(Now)); // when to trigger the timer
    Timer1.Interval := round((TriggerTime) + 3600000);
    SoundStr
                := './wav/Nightingale_song.wav';
    MessageFrm.Edit1.Text := 'This is a one hour reminder!';
   end:
             procedure TClockFrm.Message30MinutesClick(Sender: TObject); {30M}
             var TriggerTime : TTime;
             begin
              Timer1.Enabled := True;
              TriggerTime := TTime(Frac(Now)); // when to trigger the timer
              Timer1.Interval := round((TriggerTime) + 1800000);
              SoundStr
                          := './wav/Poppeye.wav';
              MessageFrm.Edit1.Text := 'This is a half hour reminder!';
             end:
procedure TClockFrm.Message15minutes1Click(Sender: TObject); {15M}
var TriggerTime : TTime;
begin
 Timer1.Enabled := True;
 TriggerTime := TTime(Frac(Now)); // when to trigger the timer
 Timer1.Interval := round((TriggerTime) + 90000);
 SoundStr
              := './wav/Nightingale_song.wav';
 MessageFrm.Edit1.Text := 'This is a quarter hour reminder!';
end:
procedure TClockFrm.MessageEveryMinuteClick(Sender: TObject); {1M}
var TriggerTime : TTime;
begin
 Timer1.Enabled := True;
 TriggerTime := TTime(Frac(Now)); // when to trigger the timer
 Timer1.Interval := round((TriggerTime) + 5000); // kept short fore testing
 // SoundStr
                := './wav/pacman_dies_y.wav';
 // SoundStr
                := './wav/pacman_intro.wav';
 SoundStr := './wav/boop-boop2.wav';
 MessageFrm.Edit1.Text := ";
// MessageFrm.Edit1.Text := 'This is a one minute reminder!';
end:
```
SPEAKING-SPORTS-CLOCK II. THE CLOCKFORM



PAGE 11 / 17

CREATING THE SETTING FOR TRANPARENCY OF THE FORM

You need to set the following properties AlphaBlend = True AlphaBlendmode := 255 //You can alter this but it wi.

TransparentColor := True

TransparentColorValue := ClBtnFace // You actually can choose any colour





procedure TClockFrm.ClrBtn_60_IntervalClick(Sender: TObject); var TriggerTime: TTime; begin

Timer1.Enabled := True; TriggerTime := TTime(Frac(Now)); // when to trigger the timer Timer1.Interval:= round((TriggerTime) + 3600000);

SoundStr := './wav/Nightingale_song.wav'; MessageFrm.Edit1.Text := 'This is a one hour reminder!'; //If you want a pre-heare: sndPlaySound(SoundStr, SND_FILENAME or SND_ASYNC); // Otherwise disable it MessageFrm.ClrBtnEditSpeakClick(Sender); // Otherwise disable it

end;

SPEAKING-SPORTS-CLOCK

II. THE CLOCKFORM



PAGE 12 / 17

SETTINGS FOR THE APPOINTMENT

It is necessary to realise that measuring time is not that simple: If you want to add time or subtract it you have to be very precise.

We need to create these measurements first: and this is why I love pascal you can simply read your code. I created a triggertime (the moment you want to let an event happen). First of all we have to get the -"Now" time: You do that with the code **NOW**. In the system.pas file you will find this code:

function Frac(const X: Extended): Extended;

The system file is usually already in your uses section. A **fraction** (from Latin fractus, "broken") represents a part of a whole or, more generally, any number of equal parts. Good to remember.

You see here the 24 hour clock measure to be multiplied by 8640 (The number 86400 of daily minutes multiplied by seconds 1000)

TriggerTime := TTime(Frac(Now)); // when to trigger the timer Timer1.Interval:= round((TriggerTime) + 3600000); // which means an hour



```
Timer1.Enabled := True;
TriggerTime := TTime(Frac(Now)); // when to trigger the timer
Timer1.Interval:= round((TriggerTime) + 3600000);
```

SoundStr := './wav/Nightingale_song.wav'; MessageFrm.Edit1.Text := 'This is a one hour reminder!'; //If you want a pre-heare: (as example) sndPlaySound(SoundStr, SND_FILENAME or SND_ASYNC); // Otherwise disable it MessageFrm.ClrBtnEditSpeakClick(Sender); // Otherwise disable it

end;

```
procedure TClockFrm.ClrBtnAppointmentClick(Sender: TObject);
var TriggerTime, NowTime: TTime;
begin
if Maskeditl.Text =' : : 'then ShowMessage('You must enter a time')
else
begin
TriggerTime :=StrToTime(Maskeditl.Text); // when to trigger the timer
NowTime:=TTime(Frac(Now));
Timer2.Interval := round((TriggerTime - NowTime) * 86400 * 1000);
Timer2.Enabled := true;
SoundStr := './wav/Nightingale_song.wav';
Editl.Visible := False;
Maskeditl.Text := ' : : ';
end;
end;
```



SPEAKING-SPORTS-CLOCK II. THE CLOCKFORM



		D.exe (Win32 - Debug)	
	 Building Delphi Compiler Resource Compiler Build Events Application Entitlement List Forms Manifest Icons 	Forms Main form CalendarFrm Auto-create forms CalendarFrm MessageFrm FrmData ClockFrm	Available <u>f</u> orms
nd; roced egin	lure TClockFrm.Move1Cl	ick(Sender: TObject);	
Borde Trans color	erStyle:=bsToolWindow parentColor:=False; ::=clred;	·'	
Borde Frans Color Calen	<pre>rStyle := bsToolWindow parentColor := False; := clred; darFrm.WindowState :=</pre>	., TWindowState.wsMinimized <mark>;</mark>	
Borde Frans color Calen nd; roced agin	<pre>rStyle := bsToolWindov parentColor := False; := clred; darFrm.WindowState := dure TClockFrm.transpa</pre>	, TWindowState.wsMinimized; rentForm1Click <mark>(</mark> Sender:TObject);	
Borde Frans Color Calen nd; roced egin Borde Frans	<pre>rrStyle := bsToolWindow parentColor := False; idarFrm.WindowState := dure TClockFrm.transpa rrStyle := bsNone; parentColor := True;</pre>	TWindowState.wsMinimized; .rentForm1Click(Sender:TObject);	



III. THE MESSAGE FORM (MessageGUI.pas - MessageFrm)

How to use it:

At the left top is an edit- label where you can type the text you want to be spoken.

In the middle of the form there is a Memo field where you can enter a large text. You could use this technique to speak out your blog or helpfile(?) or any other text. So after you have selected the voice you would like to read out the text, you need to save the voice selection.

Figure 16: Overview of the message form





Type text		Stop the Form	
Speak Edit Text	Save voice selection	Speak Memo text	
Select voice 🚽		~	
"Microsoft George - En "Microsoft Hazel - Engl "Microsoft Susan - Eng "Microsoft Frank - Dute	glish (United Kingdo lish (United Kingdon lish (United Kingdor ch (Netherlands)"	om)" n)" m)"	
Pause	Resume	Cancel	

Figure 17: The Memo field for speaking

Figure 18: The list of speaker voices





MICROSOFT

The Microsoft text-to-speech voices are speech synthesizers provided for use with applications that use the Microsoft Speech API (SAPI) or the Microsoft Speech Server Platform. There are client, server, and mobile versions of Microsoft text-tospeech voices. Client voices are shipped with Windows operating systems; server voices are available for download for use with server applications such as Speech Server, Lync etc. for both Windows client and server platforms, and mobile voices are often shipped with more recent versions.

Microsoft neural text-to-speech uses deep neural networks to make the voices of computers nearly indistinguishable from recordings of people. With the human-like natural prosody and clear articulation of words, neural text-to-speech has significantly reduced listening fatigue when you interact with AI systems.

The patterns of stress and intonation in spoken language are called prosody. Traditional text-to-speech systems break down prosody into separate linguistic analysis and acoustic prediction steps that are governed by independent models. That can result in muffled, buzzy voice synthesis. Microsoft neural text-to-speech capability does prosody prediction and voice synthesis simultaneously, uses deep neural networks to overcome the limits of traditional text-to-speech systems in matching the patterns of stress and intonation in spoken language, and synthesizes the units of speech into a computer voice. The result is a more fluid and natural-sounding voice.

In this overview, you can see the benefits and capabilities of the text-to-speech service, which enables your applications, tools, or devices to convert text into human-like synthesized speech. Use human-like neural voices, or create a custom voice unique to your product or brand. For a full list of supported voices, languages, and locales, see supported languages:

WINDOWS 10 AND LATER

In Windows 10, Microsoft Hazel (in the TMS version it is still available) was removed from the US English Language Pack and the Microsoft voices for Mobile (Phone/tablet) are available (Microsoft Mark and Microsoft Zira). These are the same voices found on Windows Phone 8, Windows Phone 8.1 and Windows 10 Mobile.

These voices language packs are also available for a variety of voices similar to that of Windows 8 and 8.1. None of these voices match the Cortana text-to-speech voice which can be found on Windows Phone 8.1, Windows 10, and Windows 10 Mobile.

In an attempt to unify its software with Windows 10, all of Microsoft's current platforms use the same textto-speech voices except for Microsoft David and a few others.





MOBILE

Every mobile voice package has the combination of male/female, while most of the desktop voice packages have only female voices. All mobile voices have been made universal and any user who downloads the language pack of that choice will have one extra male and female voice per that package.

A hidden text-to-speech voice in Windows 10 called Microsoft Eva Mobile is present within the system. Users can download a pre-packaged registry file from the **windowsreport.com** website. Microsoft Eva is believed to be the early voice for Cortana until Microsoft replaced her with the voice of Jen Taylor in most areas.

These voices are updated with Windows to sound more natural than in the original version as seen in updated retail builds of Windows 10.

Since I plan to create versions for Android and iPhone, Linux and Mac and Windows I will make the technique available for everybody on each platform. If you are a subscriber, you will have the Windows version for free. The other variations need some extra time to be created. Extra information:

https://docs.microsoft.com/en-us/azure/cognitive-services/ speech-service/index-text-to-speech

15:43:58 🔀 Sunday 24/10/2021	Analo	gue Clock	Blaise Pascal Magazine 98 2021	T
October 2021 🔺 🗸	November 2021 🔺 🐱	December 2021 🔺 🗸	January 2022	^ ~
Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	Mon Tue Wed Thu Fri Sat Sun 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5	Mon Tue Wed Thu Fri Sat Sun 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2	Mon Tue Wed Thu F 27 28 29 30 3 3 4 5 6 10 11 12 13 17 18 19 20 2 24 25 26 27 2	Fri Sat Sun 31 1 2 7 8 9 14 15 16 21 22 23 28 29 30
February 2022 A V	March 2022 A V	April 2022 ^ V	May 2022	4) O
Mon Tue Wed Thu Fri Sat Sun 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 1 2 3 4 5 6 7 8 9 10 11 12 13	Mon Tue Wed Thu Fri Sat Sun 28 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10	Mon Tue Wed Thu Fri Sat Sun 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 32 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10	Mon Tue Wed Thu 1 25 26 27 28 2 2 3 4 5 9 10 11 12 1 16 17 18 19 2 23 24 25 26 2 30 31 1 2	Fri Sat Sun 29 30 1 6 7 8 13 14 15 20 21 22 27 28 29 3 4 5
June 2022 🔺 🗸	July 2022 🔺 🗸	August 2022 🔺 🗸	September 2022	2 ^ ~
Mon Tue Wed Tu Fri Sat Sun 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 66 7 8 9 10	Mon Tue Wed Thu Fri Sat Sun 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 32 24 25 26 7 28 29 30 31 1 2 3 4 5 6 7	Mon Tue Wed Thu Fri Sat Sun 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11	Mon Tue Wed Thu F 29 30 31 1 1 5 6 7 8 12 13 14 15 1 19 20 21 22 2 26 27 28 29 3 3 4 5 6	Fri Sat Sun 2 3 4 9 10 11 16 17 18 23 24 25 30 1 2 7 8 9
October 2022 🔺 🗸	November 2022 🔺 🗸	December 2022 🔺 🗸	January 2023	^ V
Mon Tue Wed Thu Fri Sat Sun 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6	Mon Tue Wed Tu Fri Sat Sun 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4	Mon Tue Wed Thu Fri Sat Sun 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 3 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8	Mon Tue Wed Tue 26 27 28 29 2 2 3 4 5 9 10 11 12 16 17 18 19 2 23 24 25 26 2 30 31 1 2	Fri Sat Sun 30 31 1 6 7 8 13 14 15 20 21 22 27 28 29 3 4 5



IV. THE DATA FORM (dataGUI.pas - FrmData)

How to use it:

the handling of the form is quite simple: You can change the colour of the form and eventually delete all appointments in one event. For all the other handling you need to use the DBNavigator. By coupling it with the CalendarFrm.DataSource1 which is coupled with kbmMemTable1 and that enables the connection. This kind of coupling is very efficient, you could upgrade that to separate buttons. The grid is also coupled through the same settings. Ypu can of course alter this form in any

CalendarGUI 💿 ClockGUI 🛛 dataGUI 🗙		way you want to.
📾 FrmData	– 🗆 ×	
Change form color Delete all appointments		
DateEvent TimeEvent EventName		
FrmData		- 🗆 X
Change form color		
DateEvent TimeEvent Er	ventName	^
11/10/2022 09:00:00		
11/10/2022 09:00:00		
12/10/2022 09:00:00		
13/10/2022 09:00:00		
19/10/2022 09:00:00		
26/10/2021 13:00:00	hel	
		~ \
<		
		⊸ ∧
		>
Figure 20: The Data form		
and runtime Stop the program -		
		So to the Analogue clock
	20:28:37 ×	to the Analogue clock
	Sunday Sunday	
Actual Date	>24/10/2021 🕒	
	October 2021 A V	
	Mon Tue Wed Thu Fri Sat Sun	Select month / year
	27 28 29 30 1 2 3	
	4 5 6 7 8 9 10	
On double click	11 12 13 14 15 16 17	
select and sets	18 19 20 21 22 23 24	— Today
	25 26 27 28 29 30 31	
	1 2 3 4 5 6 7	Opens
Enter the time	00:00 All appointment	Opens the ever issue
	event name	the overview
- Figure 21.	The 1 month calendar	- Enter the text
ngure 21:		for the appointment



PAGE1/17



www.Flixengineering.com/books

Delphi by Example

Learn to build and deploy modern, multi-tier data-driven **Delphi** applications for multiple platforms with the **Visual Component Library (VCL), FireMonkey (FMX)**, and **TMS WEB Core**!

- Become a VCL UI Grid Control Insider! Get to know TAdvStringGrid And its data-aware counterpart TDBAdvGrid in detail. Starting with basic examples, use HTML, insert images, export and import data, customize navigation and mouseinteraction. In-depth look at sorting, editing, filters, and grouping with real-world examples
- Efficiently work with data from text files, embed it in your executable, or use FireDAC with SQLite. Write asynchronous, nonblocking code
- **Use vector images in your grid**, browse **ZIP** archives, and export reports to **Microsoft Exce**l and **Adobe PDF**
- Take the fear out of complex multi-tier **Delphi** projects!
 Step-by-step you will learn to build a database, add a **Web** service, write clients for **Windows**, **Smartphone**s, and the **Web**!
- Learn how to deploy your Web services as Windows Services and manage them with a built-in custom Web application
- Test and add documentation with Swagger UI
- Easily retrieve diagnostic information to monitor your services with Windows Management Instrumentation
- Build Web and cross-platform desktop applications with TMS Web Core and the powerful TMS Miletus sharing the same code base



20

PAGE 2/17

CHAPTER 1. INTRODUCTION

Symbol	Description
ė	click with left mouse button
é	click with right mouse button

1.2.4 References to other books

Other Delphi publications will be referenced using citations with square brackets. You can look up these references in the bibliography (\mapsto section 6.3 on page 461). For example, if the reference [2, p. 75] is given, it means you can find further information on page 75 in book one of the Hands-on series.

1.3 Technical review

Anyone who has tried to write a technical book knows that no matter how good you think you are at something, at some point you have to exchange ideas with other experts. For this reason, I would like to thank my technical reviewers:

Bruno Fierens, TMS Software

Bruno Fierens is owner and CTO of TMS Software. Since the company was founded in 1996, right after the release of Delphi 1, he has been developing VCL components. Meanwhile he and his team create components for Fire-Monkey, LCL, IntraWeb, .NET, IoT, and many other frameworks. He is present at numerous conferences and events, such as Be-Delphi, the Forentage, ITDevCon, CodeWayTour, and SDN. He is especially interested in user interface ergonomics, rapid application development, and the design of interfaces between hardware and electronics.

Wagner Landgraf

Wagner is the brains behind the TMS Business product line. He is actively involved in the development of TMS Logger, RemoteDB, Scripter, Sparkle, Aurelius, and XData. For this reason, it is also clear why these products partly build on each other or interact very well with each other.

Without his technical support and the fast and comprehensive answers to my questions, this book would not have been possible.



81

PAGE 3/17

1.4. AUTHOR

1.4 Author

Dr. Holger Flick studied computer science at the University of Dortmund and received his doctorate from the Faculty of Mechanical Engineering at the Ruhr University in Bochum. He has been programming with Delphi since 1996 and has always been active in the community. During and after his studies, he worked as a freelancer on numerous projects for Borland and was able to exchange ideas directly with many Delphi experts from Scotts Valley, CA. Mainly, he tested Delphi for the QA department, but also programmed database applications and web applications for the Borland Developer Network. Holger has also presented at conferences and seminars on various Delphi topics. His commitment and extensive knowledge of Delphi programming, gained through years of theoretical and practical work in the area of object-oriented programming with Delphi and other programming languages (e.g. C#, Objective-C), led to his appointment as Embarcadero Delphi MVP in 2016. From 2013 to 2018, Dr. Holger Flick was responsible for the entire software and hardware architecture of a medium-sized company in Germany. Among other things, he developed individual software solutions with Delphi. Since 2017, he presents products and solutions of TMS Software as Chief Evangelist in form of numerous technical articles, bilingual video tutorials, and leads through seminars. In 2019, he founded FlixEngineering LLC in the United States of America.

1.5 Required programming skills

This book is written for Delphi developers that have developed Windows applications and are familiar with the VCL component library. Further, elementary understanding of how to use the primary database technology FireDAC that is bundled with Delphi is required.

All the basics concepts presented and explained in [2, 3, 4] with regard to controls, components, databases, web services, web applications, FlexCel, and XData will not be repeated here. The same applies to figures and tables. Instead, this book will give references to other literature when needed.

1.6 Required licenses

You may reproduce all examples using 'Trial' licenses. Of course, you will need to purchase the licenses of the TMS products mentioned for commercial use. As you will learn to make use of the complete TMS toolbox, you might consider purchasing



tmssoftware.com



PAGE 4/17



Contents

1	Intr	oduction	17
	1.1	Structure of the book	18
	1.2	Typesetting	18
		1.2.1 Screenshots	19
		1.2.2 Source code	19
		1.2.3 Keyboard and mouse	19
		1.2.4 References to other books	20
	1.3	Technical review	20
	1.4	Author	21
	1.5	Required programming skills	21
	1.6	Required licenses	21
	1.7	Downloading the source code	23
	1.8	Disclaimer	24
	1.9	Getting help reproducing the examples	25





PAGE 5/17



6				CONTENTS	
I	VCI	Grid	Controls	27	
2	TAd	29	1		
	2.1	Comp	atibility w	ith Delphi versions	r
	2.2	Gettin	g started		1
		2.2.1	Cell ty	pes	
		2.2.2	Dimen	sions	Ē
		2.2.3	Writing	g text into cells	2
		2.2.4	ng mouse clicks		
			2.2.4.1	OnCellClick with row selection	
			2.2.4.2	Determine selected cell(s) without mouse click 37	
			2.2.4.3	Determine selected rows(s) without mouse click 38	
		2.2.5	Changi	ing dimensions 38	
		2.2.6	Quickly	y generating test data	
		2.2.7	Bundle	ed examples	
			2.2.7.1	List of examples 42	
			2.2.7.2	Locating the examples	
		2.2.8	Summa	ary	
	2.3	Static,	virtual, a	nd dynamic cells 45	
		2.3.1	Static o	cell properties	
		2.3.2	Using	virtual cells	
			2.3.2.1	Changing cell data for display	
			2.3.2.2	Provide data on demand using virtual cells 53	
		2.3.3	Dynam	nic cell properties 54	
			2.3.3.1	State of the grid 54	
			2.3.3.2	Color	
			2.3.3.3	Color gradient	
			2.3.3.4	Alignment	
			2.3.3.5	Mouse cursor	
	2.4	Using	HTML		
		2.4.1	Images	stored in PictureContainer and GridImages 63	
		2.4.2	Suppor	rted tags	
		2.4.3	Where	to insert HTML tags 76	



PAGE 6/17

7

HANDS-ON

CONTENTS

2.5	Organizing rows and columns	
	2.5.1 Size	
	2.5.1.1 Persisting column widths	
	2.5.1.2 Stretching 80	
	2.5.2 Reordering (moving)	
	2.5.3 Hidden columns and rows	
	2.5.4 Merging	
	2.5.5 Grouping	
2.6	Loading and writing grid data	
	2.6.1 Customizing saving and loading 100	
	2.6.2 Creating PDF files 101	
	2.6.2.1 Document options	
	2.6.2.2 Document information	
	2.6.2.3 Example 106	
2.7	Navigating	
	2.7.1 Keyboard 110	
	2.7.2 Mouse	
2.8	Sorting	
	2.8.1 Options	
	2.8.2 Storing and loading 131	
	2.8.3 Sort styles	
	2.8.3.1 Manually specifying the cell format	
	2.8.3.2 Custom sorting	
	2.8.4 Events	
	2.8.5 Ignoring columns	
	2.8.6 Row indexes before and after sorting	
	2.8.7 Triggering sorting at run-time	
	2.8.7.1 Single column	
	2.8.7.2 Sort column sequences	
	2.8.8 Example	
2.9	Editing	
	2.9.1 Enabling editing for all cells	
	2.9.2 Read-only cells	
	and a second state of the second second second by the second second second second second second second second s	



8

PAGE 7/17



CONTENTS

		2.9.3	Startin	g the edit process using code			•	149
		2.9.4	Inplace	editors			•	149
			2.9.4.1	Defining an inplace editor	•		÷	151
			2.9.4.2	Restricting the length of input				152
			2.9.4.3	Using masks				152
			2.9.4.4	Spin editors				155
			2.9.4.5	Combo boxes				157
		2.9.5	Valida	ing user input				159
	2.10) Undo	& Redo .		• •		•••	163
3	TDB	AdvGri	id: Data-a	aware TAdvStringGrid				165
	3.1	How i	s data linl	xed to the grid?	.,			169
		3.1.1	Assign	ing the datasource				169
		3.1.2	Definin	ıg columns				169
	3.2	PageM	lode is ke	yl				170
	3.3	EditPo	stMode d	etermines when data is posted	• •		•	171
п	На	nds-or	n Examp	les				173
4	Asso	orted g	rid exam	oles				175
	4.1	VCL: I	oading bi	g CSV data files asynchronously				175
		4.1.1	Downl	oading the data				176
		4.1.2	Why w	e use a resource				178
		4.1.3	Adding	g the resource				180
		4.1.4	Loadin	g the resource at run-time				184
		4.1.5	Indicat	ing activity				186
		4.1.6	Loadin	g data asynchronously		ί.	4	188
	4.2	Brows	ing data f	rom an SQLite database				195
		4.2.1	Creatin	ig an SQLite database with Array DML			4	196
		4.2.2	Chang	ing the browser app to SQLite				202
	4.3	Brows	ing data v	vith a filter				208
		4.3.1	Popula	ting the combo box				210
		4.3.2	Showi	ng only data for one state			ĺ.	211



PAGE 8/17

9

CONTENTS

		4.3.3	Initiali	zing the user interface	211
		4.3.4	Loadin	g data for a state	214
		4.3.5	Adding	g eye candy	216
	4.4	Using v	ector im	ages in TAdvStringGrid	217
		4.4.1	Compo	onents on the main form	219
		4.4.2	Applic	ation settings	227
		4.4.3	TIconI	tem: Container for icons	230
		4.4.4	Fields	and properties of the main form	231
		4.4.5	Setting	g up the grid	234
		4.4.6	Folder	selection	238
		4.4.7	Updati	ng the image list	239
		4.4.8	Loadin	g icons from a folder	240
		4.4.9	Loadin	g icons stored in a ZIP archive	242
		4.4.10	Export	to Microsoft Excel	245
		4.4.11	Reaction	ng to display resolution changes	247
5	Mul	ti-tier, c	ross-pla	tform development	249
	5.1	Buildir	ng the da	tabase	251
		5.1.1	Creatin	ng the database	252
		5.1.2	Postal	code table	254
			5.1.2.1	Source	254
			5.1.2.2	Usage restrictions	254
			5.1.2.3	Attributes	256
			5.1.2.4	Creating the table	256
			5.1.2.5	Sample data	257
		5.1.3	Count	ry codes	257
			5.1.3.1	Source	258
			5.1.3.2	Usage restrictions	258
			5.1.3.3	Attributes	258
			5.1.3.4	Creating the table	260
			5.1.3.5	Sample data	260
	5.2	TMS X	Data: We	b services	260
		5.2.1	Creati	ng the server application	261



tmssoftware;com



10

PAGE 9/17



CONTENTS

	5.2.2	Class for Web service data
	5.2.3	Database access
	5.2.4	Providing a connection
	5.2.5	Using the GetConnection method
	5.2.6	Querying the database
	5.2.7	Mandatory units
	5.2.8	Defining the service interface
	5.2.9	Implementing the service method
	5.2.10	Database client libraries
	5.2.11	Adding a custom route
	5.2.12	Browse your services with Swagger UI
	5.2.13	Adding documentation for Swagger UI
	3	5.2.13.1 XMLDoc in Delphi
		5.2.13.2 Swagger-specific tags 292
		5.2.13.3 Excluding methods from Swagger 294
5.3	VCL: In	plementing a desktop client for lookup
	5.3.1	Designing the main form
	5.3.2	Retrieving data from the Web service
	5.3.3	Implementing the user events
	5.3.4	Customizing the grid control
	5.3.5	A word on T(Virtual)ImageList
5.4	FireMo	nkey: Mobile app to lookup postal codes
	5.4.1	The worstcomes first
	5.4.2	HTTP and HTTPS in Android apps
	5.4.3	Building the user interface
	5.4.4	Retrieving data from the Web service
5.5	TMS W	EB Core: Web client application
	5.5.1	Designing the main form
	5.5.2	Connecting to the XData server
	5.5.3	Calling the service method
	5.5.4	Processing the result and updating the grid 320
5.6	VCL: U	sing the Web service for efficient data entry
	5.6.1	Designing the user interface

PAGE 10/17

HANDS-ON

CONTE	NTS		11
	5.6.2	Fields and constants in the form	326
	5.6.3	Initializing the form and grid	328
	5.6.4	Saving and loading the grid values	329
	5.6.5	Setup for editing	330
	5.6.6	Retrieving data from the Web service	332
	5.6.7	Evaluating the size of Web service results	333
	5.6.8	Validation of postal codes	336
	5.6.9	Selection from a combo box	339
5.7	TMS 2	Data: Deployment as a Windows service	342
	5.7.1	Windows application vs. Windows service	342
	5.7.2	Migrating an existing server project	343
		5.7.2.1 Thanks to Wagner Landgraf and outlook	344
		5.7.2.2 Adding service information to the service container .	344
		5.7.2.3 Starting and stopping the server	347
		5.7.2.4 Adding the Windows service class	348
		5.7.2.5 Helper methods to extend framework classes	351
		5.7.2.6 Updating the project source file	352
		5.7.2.7 Custom Command-line switch	354
5.8	TMS 2	Data: Configuration of Web Services at run-time	355
	5.8.1	Settings for IPostalService	356
		5.8.1.1 Settings for the database connection	356
		5.8.1.2 Settings for the server	357
		5.8.1.3 TServerSettings class	357
		5.8.1.4 Server status	360
	5.8.2	Defining the configuration service	361
	5.8.3	Models, services, and endpoints	362
	5.8.4	Implementing the configuration service	363
	5.8.5	Adding a second XData server	364
		5.8.5.1 Implementing the Singleton pattern	366
		5.8.5.2 Implementing methods to start and stop the data server	368
	5.8.6	Adding services to the second XData server	370
	5.8.7	Updating the postal service	371



12

PAGE 11/17



CONTENTS

	5.8.8	Adding start and stop to the configuration service	374
	5.8.9	Automatic reservation of endpoints	376
	5.8.10	Using the configuration service	379
5.9	TMS WI	EB Core: Building a Web interface	379
	5.9.1	Designing the main form	380
	5.9.2	Data module for XData connectivity	386
	5.9.3	Initiating the connection from the main form	389
	5.9.4	Retrieving the server configuration	390
	5.9.5	Sending settings to the server	396
	5.9.6	Starting the service	399
	5.9.7	Stopping the service	401
5.10	Adding	a Web server to an XData server	403
5.11	VCL: Up	odating the user interface with vectors	405
	5.11.1	Adding vector images	406
	5.11.2	Implementing the config button	408
5.12	TMS XE	Data: Monitoring critical server parameters	408
	5.12.1	Hardware being monitored	409
	5.12.2	Getting the hardware info	410
	5.12.3	Implementing the service methods	416
5.13	TMS W	EB Core: Adding monitoring to the Web client	421
	5.13.1	Updating the service controller	422
	5.13.2	Adding the popup window	427
	5.13.3	Appearance of the caption	436
	5.13.4	Closing the pop-up window	437
	5.13.5	Showing the pop-up window	438
5.14	TMS M	iletus: Creating a Windows client	440
	5.14.1	Looking at the finished application	442
	5.14.2	Create a new TMS Miletus application	446
	5.14.3	Updating the project file	448
	5.14.4	Migrating the service controller	448
	5.14.5	Migrating the main form	449
	5.14.6	Updating the pop-up window	452
	5.14.7	Running the application	454



List of examples

Legal

CONTENTS

6 Appendix

6.1.1

6.1.2

6.1.3

6.1.4

6.2 TColors

6.3 References

PAGE 12/17

13 HAND VITH DELPH 455 455 455



..... 1 . .

. 459

. 461

PAGE 13/17



Chapter 4

TAdvStringGrid

This chapter will explain the control *TAdvStringGrid* in detail. When Delphi 1 arrived on the market, *TAdvStringGrid* set out to be a replacement for the standard VCL control *TStringGrid*. That means the control you drop on the form is the product of 25 years of constant improvement with features steadily being added. For that very reason, it will be impossible to demonstrate each individual feature with an example. Also, not each topic of the documentation will be covered because some features are so specific they do not fit the hands-on approach of this book series.

However, each of the following sections will deal with one major feature of the grid. In comparison to the product documentation, each section will assume that the topics of the preceding section are known. This way, the level of complexity will also rise from section to section. The source code of all examples can be found in the *TAdvStringGrid* folder.

As the control is built with the features of *TStringGrid* in mind, it is helpful to know about the control. TMS uses the same names for common properties, methods, and events. If you know how to use *TStringGrid*, you will be able to use the basic functionality of *TAdvStringGrid*.

2.1 Compatibility with Delphi versions

TAdvStringGrid has been available from the start with Delphi 1 (\mapsto figure 2.1 on page 31). However, even TMS had to give in at some point and the minimum requirement to use *TAdvStringGrid* was raised to Delphi 7. This is still pretty low considering the features that are available. The control is available for every Delphi

29



PAGE 14/17



30

CHAPTER 2. TADVSTRINGGRID

version that has been released after that one. For the record, it also can be used in C++ Builder.

2.2 Getting started

Before certain features of a grid can be looked at in detail, the basic terminology when working with grids needs to be described.

2.2.1 Cell types

A grid consists of cells. Each cell can be identified by a unique coordinate expressed by a tuple of its column and row. Thinking of a 2-dimensional coordinate system, *column* and *row* can be thought of as *x* and *y*.

Figure 2.2 on page 32 illustrates these terms. Each cell in the example shows its coordinate (col/row). In Delphi, arrays are used for tuples which are zero-based. Thus, the upper-left corner has the coordinate [0, 0]. The figure also shows that with a total of 10 rows and 9 columns in the grid, the lower right coordinate is addressed as [8, 9]. Furthermore, 3 cell types are depicted:

- 1. **Fixed cells** that (in general) cannot be resized or edited¹ and do not scroll out of focus. By default, fixed cells cannot be edited even if the editing capabilities of the grid are enabled. Special properties have to be set to enable editing of fixed cells.
- 2. Normal cells that can be edited as long as enabled.
- 3. **Selected cell** or **cells** depending if the selection of multiple cells is enabled. Furthermore, *TAdvStringGrid* can be configured that a whole row is selected.

2.2.2 Dimensions

The number of total cells in a grid is specified using the total number of rows (*Row-Count*) and columns (*ColCount*).

The *CoordinateEx* project demonstrates the minimum amount of steps needed to start using *TAdvStringGrid*:



¹There are special properties available that even allow resizing and editing of fixed cells. This is not very common practice though and is not enabled by default.

PAGE 15/17

2.2. GETTING STARTED

							3	- 11 3	×
File Edit Search Vi	ew Project Run Component D	Database Tools Window Hel	p REGO	-	四次				
*1 m - 0 0 0	Standard Add	itional Win32 Sustem Data Acc	ess Data Contro	s dbExpress]	DataSnap BDE	ADD I InterBase	WebServices Intern	elExpress Intern	1
			1 20 20 5				41 In In In In	编编合量。	
		ula suala finala figula fisika proda pr	at anothe sound and	to succe and a		n	and the state that a		-
Object Inspector		Form1						- 0 ;	×
AdvStringGrid1	TAdvStringGrid			- server and				1.1.2.2.4.4.4.4.4.4.4.4	1
Properties Events		0 1	2	3	4				
8 Grouping	(TGrouping)			-					
Height	321								
HelpContext	0								
HelpKeyword	inglass and an an an an an and a state of the								
HelpType	htContext								
HideFocusHect	Faise								
HighighiLolor			T						
Highlight 1 exil olor	L tourishinghet ext								
HintColor									
HintShowCels	False	11							
HintShowLargeTest	False		1						
HintShowLargeTextPos	hoCel	11							
HintShowSizing	False	1							199
HoverButtons	[THoverButtons]						Ouldk con	110 1V8.5.5.51 6	ŝ
HoverFixedCells	hiNone	and a second sec					L_ Alow	edit	
Hovering	False						Set looks	articitoets	
HoverRow	False	14					Local Month		
HoverRowCells	[hcNormal.hcSelected]	1			Sile and and and		ar an an an an an ar an		11.5
HoverRowColor	Cinlo8k	-				10 1 10 1 10 10 10 10 10 10 10 10 10 10			_
HoverBowColorTo	a Bloost								~
and a rear and along real of the day of the same same	Civone	# Unit1.pas						- 0 ;	~
HTMLHint	False	Unit1.pas	toreto mit					;	^
HTMLHint HTMLKeepLineBreak	False True	Unit1.pas	Unit1					- □) (e ~ e)	*
HTMLHint HTMLKeepLineBreak HTMLSettings	False True (THTMLSettings)	Unit1.pas	Unit Innit Ur	itl;	na ha cata ana ang daga dag	d on the second states		- D) (e v e)	*
HTMLHint HTMLKeepLineBreak HTMLSettings IntegralHeight	True (THTMLSettings) Felse	Unit1.pas	Unit Dr	itl;	diplocation and forbid	gi manunasi ta atau		- D) 40 ~ 00	*
HTMLHint HTMLKeepLineBreak HTMLSettings IntegraHeight IntellPan	False Folse (THTMLSettings) False pVertical	H Unit1.pas S S S S TForm1 Onit1.pas S S TForm1 Onit1.pas S S S S S S	Unit anit Ur interfa	itl;	anna ann ann fostaire	gin an contract and		;	*
HTMLHink HTMLKeepLineBreak HTMLSettings IntegraHeight IntelliPan IntelliZoom IntelliZoom	Faise True (THI NSettings) Faise DVertical True Marring	Unit1.pas	Unit anit Ur interfa	iitl; .ce	anna ann a seo là teile	d re-annessana da ann) *****	*
HTMLHink HTMLKeepLineBreak HTMLSettings IntegraHeight IntellPan IntellZoom InvalidEntryIcon InvalidEntryIcon	False True (TH1 MLSettings) False oVerlical True ieWarning	∰ Unit1.pas ∄:≪ TForm1 ⊕: Vaiables/Constants ⊕: U Vaiables/	Unit Unit Ur janit Ur interfa	iitl; .ce	anna ann a seo is taite	d to an unconstant of a ran) *****	*
HTMLHint HTMLKeepLineBreak HTMLSettings IntegraHeight IntelliZoom IntelliZoom InvalidEntryIcon InvalidEntryIcon InvalidEntryTest InvalidEntryTest	Create True (THTMLSettings) False EVertical True ieWarng	Unit1.pas	Unil panit Ur interfa uses Windo	iitl; .ce ws, Messa	ges, SysUtil	s, Variants.	Classes, Graph	- C :	*
HTMLHint HTMLKeepLineBreak HTMLSettings IntegraHeight IntelliZoom InvalidEnttyToon InvalidEnttyTest InvalidEnttyTest InvalidEnttyTest Left	Rosono False True (If I I M. Settings) False EVertical True eWarning	Unit pas	Unit panit Ur interfa uses Windo Dialo	nitl; nce ws, Messa ga;	ges, SysŪtil	s, Variants,	Classes, Graph	- C :	*
HTMLHink HTMLKepLindBreak HTMLSettings IntegraPright IntellPan IntellCoom InvalidEntryTest InvalidEntryTest InvalidEntryTest Lotk Look	Biological Trus (THTHLSettingt) False Evertical Trus eWarmo, 8 8 a(Custom	Unit1.pas	Unit Init Ur Interfa Uses Windo Dialo	nitl; noe ws, Hessa ggs;	ges, SysUtil	s, Variants,	Classes, Graph	- D) te - ed	»
HTMLHink HTMLKepLineBreak HTMLSettings IntegraPricipit IntegraPricipit IntelfiZoom IntraidEntryTest InvaldEntryTest InvaldEntryTest Lock Lock Lock	Constant Faire True Faire Faire Faire Eventson E	Unit-pas	Unit Ur Innit Ur Interfa Usës Windo Dialo type	nitl; noe nws, Messa ngs;	ges, SysUtil	s, Variants,	Classes, Graph	- D)	> >1
HTMLHrx HTMLKepLinkBrak HTMLSettings IntegraHeight Intelf2om Intel	Crone False True (1111 (L.Settings) False Evention True evention Warring G g g GLutson False False	i Unit.pas ⇒ 5 + 5 Ffem1 ⊕ Uveske/Constants ⊕ Uves	Unit anit Ur interfa uses Windo Dialo type TForm	nitl; noe nys, Messa ngs; nl = class	ges, SysUtil (TFoxm)	s, Variants,	Classes, Grapi	- D)	>
HTMLHrX HTMLKepLiveBreak HTMLSetings Integrateght IntellZoom InvaleEntryTeat InvaleEntryTeat InvaleEntryTeat Lockup Lockup LockupCaseSensitive LockupCaseSensitive	Elson False THTH Nachings) False Evented Evented Evented Evented Evented Evented Evented Evented Evented Evented False False	ill UnitLpss D + to the state Constants D + to Valde Constants D - to Uses	Unit Innit Ur Interfa uses Windo Dialo type TForm Adv	nitl; noe nys, Messa ngs; nl = class StringGri	ges, SysUtil (TForm) dl: TAdvStri	s, Variants,	Classes, Grapi	- D ;	»
HTMLHrx HTMLKeptineBreak HTMLSettings Integraficity Intel®an Intel	Coope False Tour (111 MLSettings) False Civetical Tour Warring 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	i Unit.pas ⇒ 55 Ffem1 ⊕ VaiderConstants ⊕ Uses	Unit Interfa uses Winde Diale type TForm Adt prive	nitl; noe nys, Messa ngs; nl = class "StringGri te	ges, SysUtil (TForm) dl: TAdvStri	.s, Variants, ngGrid;	Classes, Grapt	- 🗆 ;	> >1
HTMLH:rit HTMLSteipLindBreak, HTMLSteitinge Integraf-teight IntelfPan Intelf	Control Contro Control Control Control Control Control Co	ill Unitipes D + to the state and the state	Unit Innit Ur Interfa Usés Windo Dialo Type TForm Adu prive	nitl; noe ws, Hessa gg; hl = class StringGri te private de	ges, SysUtil (TFoxm) dl: TAdvStri clarations)	.s, Variants, ngGrid;	Classes, Graph	- C ;	> >1
HTMLHER HTMLKeptimeBreak HTMLKeptimp Integraficijit Integraficijit IntelliPon Invalidi: ntyTest Invalidi: ntyTest Invalid: ntyT	Coope Faite Tour (111 MLSettings) Faite Civeficial True Warring 8 G G G G G G Custom Faite Faite Faite Faite Custom Faite Custom Faite Custom Faite Custom Faite Custom Faite Custom Faite Custom Custom Faite Custom Faite Custom Custom Faite Custom Custom Faite Custom Custom Faite Custom Cu	Unit.pas ⊕ stor ⊕ stor ⊕ Storent ⊕ Uses ⊕ Uses	Unit nnit Ur interfa uses Windo Dialo type TForm Adv priva fi upphi	nitl; nce nys, Hessa ngs; nl = class StringGri tte Private de g	ges, SysUtil (TFoxm) dl: TAdvStri clarations)	,s, Variants, ngGrid;	Classes, Grapt	- D ;	>>1
HTMLHri HTMLKeptLineBreak HTMLSettings HindSettings HindBethight HealPan HealBethight HealBethight HealBethight Lockup Lockup Lockup Lockup Lockup Lockup Lockup Lockup Lockup Lockup HasCoubCesthikoy Lockup Lockup HasCoubCesthikoy Lockup Lockup HasCoubCesthikoy Lockup	Control Contro Control Control Control Control Control Co	ill Unitipes D + to the state and the stat	Unit nnit Ur interfa uses Windd Diald type TForn Adv prive () publi	nitl; noe nys, Messa ngs; nl = class StringGri te Private de c c bblic dec	ges, SysUtil (TForm) dl: TAdvStri clarations)	.s, Variants, ngGrid;	Classes, Graph	- D 3	*
HTMLHrit HTMLEspLiceBrook HTMLSethge IntegRegalitight Intel®con Intel Inte	Cronge Caste Tour Tour Caste Civotical Tour Civotical Tour Selection False False False False Civotical (Civotical Ci	Unit.pas ⊕ s§ Ffem1 ⊕ VasilerConstants ⊕ Uses	Unit Init Ur interfa uses Windd Diald type TForm Adt prive fi upbli	itl; uce ws, Messa ugs; nl = class StringGri ite orivate de o wublic dec	ges, SysUtil (TForm) dl: TAdvStri clarations) larations)	ngGrid;	Classes, Graph	- D ;	~ »
HTMUHri HTMLGept_neBrak HTMLSept_neBrak HtmLsepthight Instellizion InveldErtylican InveldErtylican InveldErtylican InveldErtylican Lockup Lockup Lockup Lockup Lockup HasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicangh MasCosbicAngh MasC	Crooms False False False False False Guiden Guiden Guiden Guiden Guiden Guiden Guiden Guiden Guiden False Guiden	I Unit Les D ≪ TFom1 D ≪ TFom1 D ≪ Unit Construction D ← Unit C	Unit Ur interfa uses Windo Dialo type TPor Adt prive () usub ()	itl; ice ws, Hessa ggs; hl = class StringGri ite Private de Cublic dec	ges, SysUtil (TForm) dl: TAdvStri clarations)	.s, Variants,	Classes, Grap	- D 3	~ »
HTMLHrit HTMLEspLiceBrook HTMLSelbrogi IntelEcon IntelEc	Cronge Existe Tour Tour Existe Ex	Unit.pas ⊕ s§ Ffem1 ⊕ VasilerConstants ⊕ Uses	Unit Unit Unit Unit Unit Unit Unit Unit	iitl; .cee .cee .ces .ces .ces .ces .ces .ces	ges, SysUtil (TForm) dl: TAdvStri clarations) larations)	ngGrid;	Classes, Graph	- D 3	>>1
HTMLHri HTMLCspLreBinak HTMLSselmga HedgaFidjik HedgaFidjik HedgaFidjik HedgaFidjik HedgaFidjik HedgaFidjik HedgaFidjik HedgaFidjik HedgaFidjik Lockup LockupHino LockupHino LockupHino LockupHino LockupHino LockupHino LockupHino HedgaFidjik HedgaF	Cronge False Tr111 UK-Settings) (Jahan Devening) Intern Intern Methods Setting False (TSungled) 0 (TMouseActions) False 0 (TMouseActions) False Catografiel False Catografiel Catografiel False False Catografiel False False Catografiel False	i Unitipes D ≪ TFem1 0: € ValderZontarts D tes	Unit Interfa uses Windd Diald type TForm Adv prive fi end; var var var	iitl; cce was, Hessa sgg; hl = classs StringGri tte Private de c Public dec	ges, SysUtil (TForm) dl: TAdvStri clarations)	.s, Variants, ngGrid;	Classes, Grapt	- C :	»
HTMLHrt HTMLEsepLiceBrook HTMLSelbrogi HadgeAfright HadgeAfright HadgeTool Invadie frighten Invadie frighten Invadie frighten Lookup Care Gemeilte Lookup Care Gemeilte Maillenoods Name Disvigeton Disvigeton Care Corvett	Cronge False Tour False Evolution Tour Evolution False False False False False False False False False (ThouseActions) False (ThouseActions) False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False False (ThouseActions) False (ThouseActions) False (ThouseActions) False (ThouseActions) False False (ThouseActions) False False False (ThouseActions) False	Unit.pas ⊕ s§ Ffom1 ⊕ VasilerConstants ⊕ Uses	Unit Interfa uses Winddo Dialo type Tforn Africe prive prive f gabli (i end; var Form)	iitl; wes, Messa ggs; al = class StringGri tte Crivate de c Zublic dec :: IForml;	ges, SysUtil (TForm) dl: TadvStri clarations) larations)	ngGrid;	Classes, Graph	- C 3	*
HTMLHri HTMLSeptLrefinak HTMLSeptLrefinak HtmLsepaHojtk HespaHojtk HespaHojtk HespaHojtk HespaHojtk HespaHon HespaHon Lockup Lockup Lockup Lockup LockupHon	Croome Fairs Frain (TSimglui) 0 (TMouseActions) Frain ActomyGott (TMouseActions) Frain G (TMouseActions) Frain G (TAdvision) Frain Frain G (TAdvision) Frain Frain Frain G (TAdvision	ill Unit.pas ⊕ ≪ TFom1 ⊕ UnitslenZontarts ⊕ Unes	Unit Inst Ur interfs uses Rindo Disid type TForn Adv priv fi usef rfor adv fi public (i end; Var Form)	iitl; we ws, Hessa gg; ul = class StringGri te o Private de o Public dec : IForml;	ges, SydUtil (TForm) dl: TAdvStri clarations) larations)	ngGrid;	Classes, Grapt	- C ?	> 1

FIGURE 2.1: As many other TMS controls, TAdvStringGrid is compatible with a lot of older Delphi versions and does not require the latest Delphi version to be used. This capture was taken with version 10.4.1.0 released on 9/17/2020 installed in the Delphi 7 IDE.



tmssoftware.com

31

TH DELPH

PAGE 16/17

CHAPTER 2. TADVSTRINGGRID



Rows Columns **Fixed Rows Fixed Columns** 4 + + + 4 ++ 10 9 1 1 Update 0/0 1/0 2/0 3/0 4/0 5/0 6/0 7/0 8/0 0/1 1 1/1 2/1 3/1 4/1 5/1 6/1 7/1 8/1 0/2 1/2 2/2 3/2 4/2 5/2 6/2 7/2 8/2 0/3 1/3 2/3 1/3 5/3 6/3 7/3 8/3 3 0/4 1/4 2/4 3/4 4 5/4 6/4 7/4 8/4 0/5 1/5 2/5 3/5 4/5 5/5 6/5 715 8/5 0/6 1/6 2/6 3/6 4/6 5/6 6/6 7/6 8/6 0/7 1/7 2/7 3/7 4/7 5/7 6/7 717 8/7 0/8 1/8 2/8 3/8 4/8 5/8 6/8 7/8 8/8 0/9 1/9 2/9 3/9 4/9 5/9 6/9 7/9 8/9

FIGURE 2.2: Addressing cells in a grid. Each cell can be accessed using its column and row index. The coordinate (0,0) is set as the left top corner. Accordingly, the right bottom corner can be addressed using (Number of columns -1, Number of rows -1). The total number of rows and columns each include the number of fixed rows and columns, i.e. if you set the total number of rows to 10 and the number of fixed rows to 2, the grid will contain 8 rows that can be edited.

32



95

2.2. GETTING STARTED

```
1 procedure TFrmMain.InitGrid:
2 begin
    // read values from controls
3
4
   LRows := txtRows.Value;
  LCols := txtCols.Value;
5
6
    LFixedRows := txtFixedRows.Value;
7
    LFixedCols := txtFixedCols.Value;
8
9
10
    // assign values to grid
11
    Grid.RowCount := LRows;
    Grid.ColCount := LCols;
12
13
    Grid.FixedRows := LFixedRows;
14
15
    Grid.FixedCols := LFixedCols;
```

The main form contains *TAdvSpinEdit* controls to specify the number of rows, columns, fixed rows, and fixed columns. These values are transferred into local variables in lines 4 to 8. This step could be omitted as the local variable values are then assigned to the properties of the grid (lines 11-15).

RowCount

Number of total rows in the grid (including fixed rows)

ColCount

Number of total columns in the grid (including fixed columns)

FixedRows

Number of fixed rows at the top of the grid. The number has to be smaller than the number of rows or it will be set to the number of total rows declaring all rows of the grid as fixed.

FixedCols

Number of fixed columns starting at the left of the grid. The number has to be smaller than the number of columns or it will be set to the number of total columns. This would designate all columns as fixed.

Note

With modern styles enabled, fixed cells are no longer drawn with a background and can only be distinguished visually when a different font or font style is used. The typical gray background is no longer used in modern user interface design standards.



33

PAGE 17/17



tmssoftware.com

Summersale



Including the PDF and Code Examples

https://www.blaisepascalmagazine.eu/product/lazarus-handbook-hardcover/

We have a new service at our website, for shipping you can make three choices: 1.The shipping cost depending on which part of the world you want the book to be shipped: Europe or the other countries: the World

C,	www.www.	province of the second	[[vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Number of the second	SHIPPING COST Europe	SHIPPING COST Europe + Registered	SHIPPING COST Europe + Track & Trace	SHIPPING COST World	SHIPPING COST World + Registered	SHIPPING COST World + Track & Trace
r		Innananananan	[accordences]	00000000000000	mannana	Lannananan

REAL-WORLD APPLICATIONS WITH PAS2 JS BY MICHAËL VAN CANNEYT

PAGE 1/14





expert

ABSTRACT

Pas2js is more than just a toy project: as the underlying compiler of **TMSWeb Core**, it is used to create web applications in a RAD manner, in both **Delphi** and **Lazarus**. But it can also be used by itself to create real-world applications. We'll show how in a series of articles.

1 INTRODUCTION

People using **Delphi** and **Lazarus** are very much used to a **RAD** (Rapid Application Development) approach to programming. TMS Web core brings this approach to web development. But a lot (*in fact, most*) web development these days is done using technologies such as **Angular, VueJS** or **React:** they are very successful development models. Indeed, so much so that **React** Native has brought web development techniques to the desktop.

These technologies have in common that they do not use RAD techniques: no point-andclick, no 2-way integrated IDE. They only use HTML and **Javascript** (or transpiled typescript) code.

Such an approach can of course also be achieved with **pas2js**. The superior **Object Pascal** editing skills of the **Lazarus IDE** make creating pascal code a breeze, editing HTML can be done in and outside of the **Lazarus IDE**.

At this moment, HTML and plain pascal code which interact is the most reliable way to create an **Object Pascal** application to run in the web when disregarding TMS Web core.

The **Lazarus** and **Free Pascal** teams are working on bringing **RAD** to native HTML, a concept as can be seen in figure 1 on page 2. The purpose of that development is not to create a **VCL** or **LCL** for the web, but to create and use a series of components that are as close to the underlying HTML as possible: there will be no position and size properties, instead such things are delegated to CSS. But this technology is in a conceptual stage, and not yet ripe for production development.





2 A SERVER-SIDE BACKEND Most Web applications or websites need a back-end: Something to authenticate a user, to fetch data from a database or add data to it. Communication with this backend can be programmed using plain HTTP or WebSockets as a transport mechanism, using several messaging protocols: **SOAP:** an older, **XML** based, protocol. REST: currently the most used protocol, usually using **JSON** as the data exchange format. Very suitable for data exchange. Lazarus IDE v2.3.0 - nativedesigner (debugging ...) File Edit Search View Source Project Run Package Tools Window Help HTML Editor Standard Additional Common Controls Dialogs Data Controls Data Access System SQLdb Misc LazControls Google API 🎦 📩 🗁 🖓 🖶 🖓 🖵 🗸 📱 🖳 🖎 Abc abi. 🛒 🚥 🗹 💿 📰 🗮 📰 🚺 🗮 🔚 🥅 🎆 Ø 1:1 ≤ 5 ■ 11 × 4 × Ø ⊡ 0 **Object Inspector** MainForm: TMainForm 💿 🕑 🚥 🖌 🌑 🗔 🖾 🗗 🗔 🚺 **Properties** (filter) \mathbb{T}_{x} Tx. Design browser Components (filter) Properties Events Favorites Restricted 🖹 🔄 _1: TWebPage 🚳 : TButtonWidget Action ÷ Button #1 Button #2 Button #3 Button #4 Styles: TWebWidgetStyl ActiveControl 🔩 : TButtonWidget Alian alNone Styles: TWebWidgetStyl Button #7 Button #5 Button #6 Button #8 🗄 🚜 : TButtonWidget AllowDropFiles (False) Button #10 AlphaBlend (False) \mathbb{Z}_{\times} AlphaBlendValue 255 Properties (filter) + Anchors [akTop,akLeft] Properties Events AutoScroll (False) Classes WebPage ui-sorta AutoSize (False) ElementID designpage BiDiMode bdLeftToRight Name 1 BorderIcons [biSystemMenu.biMinir ParentID ww-10 BorderStyle bsSizeable srOnElementID StyleRefresh BorderWidth 0 Styles 0 items Caption **IDE Design demo** 0 Tag ChildSizing (TControlChildSizing) Visible False Color clDefault + Constraints (TSizeConstraints) Cursor crDefault DefaultMonitor dmActiveForm DesignTimePP 96

(False) Figure 1: Designing HTML in a RAD manner JSON

(False)

DockSite

DoubleBuffered

ISON RPC: a Remote Procedure call using a more or less standardized ISON format. A server that exchanges data in one of these ways can be programmed in Lazarus or in Delphi. For example Remobjects SDK can be used for RPC and REST-like programming both with Lazarus and Delphi. It supports many transport frameworks: Windows-specific transport, Indy, Synapse, a fast TCP/IP stack.

Created using pas2js. Sources: Program unit

PAGE 2/14

Lazarus and **Free Pasca**l have their own

Web programming interfaces for **RPC** and **REST** mechanisms. There are several:

•	Brook framework	a general web framework built on top of FCL-Web.
•	mORMot	is a complete client server ORM/SOA framework.
٠	Fano Framework	is a web application framework.
•	WST	The Web Services Toolkit is aimed at Webservices: SOAP and JSON-RPC
		messaging protocols are both possible.

In this set of articles we'll concentrate on the frameworks that are delivered with **FPC** and **Lazarus** itself:

•	FCL-Web	this is a low-level technology that allows you to write a HTTP-based application server. This can be a FastCGI server, a CGI, an Apache module or a stand-alone HTTP(s) server. On Windows the Windows-specific HTTP service can also be used.
	fpjsonrpc	Implements a JSON RPC mechanism, which can be used to implement
	SOI DBRestBridge	Implements a REST layer which can be used to exchange information
	SQLDDRestDruge	with a database suppored by FPC.

All three can be accessed easily from within **Pas2JS**. Which mechanism or framework is used is largely a matter of preference. What they all have in common is that communication in the browser is asynchronous.

3 GUI CONSIDERATIONS

When it comes to programming a **GUI** in the browser, there are many choices. There are some general-purpose frameworks, which favour a Web Component based approach. Basically, a web component is small HTML template with (*possibly custom*) HTML tags, and a little piece of **Javascript** (**or typescript**) code to control the HTML template based on available data and user interactions. Here are some of the most popular frameworks:

Angular	Created by and maintained by Google.
React	Created by and maintained by Facebook.
Vue.JS	An alternative to Angular and Bootstrap , but similar in design philosophy.
Svelte	A relatively new kid on the block, which actually compiles HTML and Javascript
	to some very compact applications.
JQuery	One of the first comprehensive Javascript frameworks to manipulate HTML
	and CSS, it is still used widely.

But many more exist. Although it is theoretically possible to use these frameworks, with the exception of **JQuery**, these frameworks are not very suitable for work with **PasJS**: they require extensive tooling and a separate application compile step: the **javascript** and HTML is combined into the final web app.

The above frameworks deal with creating and altering HTML dynamically. They do not define a style for your components: what HTML and CSS you use is entirely up to you.



But there are a huge amount of HTML+CSS frameworks out there which provide visual markup:

	Material	The Material Design UX specifications of Google have been implemented for Angular (for example Angular-Material) and React (Material Kit React).
•	Quasar	Is a rich set of visual components built on top of Vue.JS.
	Bootstrap	Originally created by twitter it can be used with JQuery, React and Jquery, or just like that.
•	Bulma	Similar in design to Bootstrap, but without any Javascript.
•	Semantic-UI	Like Bootstrap, it can be used by itself, but it can also be used
•	JQueryUI	A set of UI components for Jquery.

There are many other frameworks out there, and a lot of plain Javascript components that integrate to a bigger or lesser degree with the above frameworks. Any of these can be used with Pas2js.

4 APPLICATION ROUTING

Basically, **pas2js** allows you to program the browser in **Object Pascal**. That also means that you do not need a server to generate the pages and to rewrite part of a page when the user clicks something or enters data.

Pas2js also does not force you to choose between a multi-page site or a single-page site: you are free to choose this. The advantage of a multi-page site is that there is little or simple logic in each of the pages, so they load fast.

The disadvantage is that you need a way to maintain application state between the pages:

For instance, if you want to display the name of the currently active user, it must be stored somewhere and retrieved whenever a new page loads.

A single-page application is more in line with a desktop application: a single html page is loaded, and the logic in the page is responsible for showing different 'forms': these forms can be built dynamically in code, or it can be just parts of the HTML which are shown and hidden as needed. The advantage of this approach is that the application state is maintained while the user navigates from form to form: there is no page reload.

Since this application runs in the browser, the user will expect to be able to use the back and forward buttons to navigate between the various "forms" in your application. To enable this behaviour, a router can be used. Navigation in your application happens with #hash links in the page URL: The browser provides a mechanism to be notified of hash changes, and you can use this to show the appropriate form in your application.

It is also possible to set the #hash part of the URL in code and the browser will add it to the browser history.

Thus, when you move programmatically to a new form and the user presses 'back' in the browser history, he will go to the previous form.

Pas2JS has a router component that allows you to work with such a routing scheme, we'll explain how to work with it in one of the following articles.



5 LOADING HTML

As written before, when showing forms or pages in the browser, this means changing the HTML in code. Changing the HTML can be done in 2 ways:

- Build the HTML in code using the Javascript DOM classes.
- Pas2js has a unit called web that defines most available classes.
- Load a HTML snippet and insert that in the DOM tree somewhere.

The second way has the advantage that you can design and preview the HTML snippets in the browser. Additionally, ready-to-use HTML snippets can be found all over the web, many of them beautifully styled, so they can be used as-is without the need to tranform them to code that reproduces them.

Because loading HTML in the browser happens asynchronously, it is a good idea to load the HTML snippets in advance. There are several ways to do this; pas2js contains a component TTemplateLoader that can load html snippets (in fact, any text file). This component can be used to load HTML snippets in the background:

The first of these calls returns a **TJSPromise** class, which can be used to wait for the result. The second call uses 2 callbacks: one which is called when the template was loaded, and the second which is called when an error occurs during the loading of the template.

Loading HTML in the background means you must sometimes wait for the snippet to be loaded before you can continue the logic of the application. This means possibly a callback for every snippet you wish to use.

As an alternative mechanism, pas2js also allows you to "link" in the html, much like a form file in Delphi is linked into your application: {\$R mydialog.html}

Depending on the options given to the **pas2js** compiler, this will create a HTML file which contains the HTML as a series of <link /> tags, or a javascript file that contains and registers all the resources as strings with the **pas2js** runtime.

We'll demonstrate both options in future articles.

Does this mean you should always use HTML snippets when changing the HTML ? No, of course not: sometimes you want to fill a table with data, or a <select > tag, and this can just as well be done in code.



PAGE 6/14

6 GETTING STARTED

In a series of articles on pas2js we'll develop an application which allows us to demonstrate some of the techniques explained in the above. The application will be developed using **Bulma** as a CSS framework: this is a CSS framework without need for any javascript framework; the programmer is responsible for setting the necessary CSS classes on the html tags. It is well-documented and can be downloaded from the following website:

http://bulma.io/

So, how to get started with **pas2js** in **Lazarus** ? First of all, download the **pas2js** compiler. The currently latest version can be downloaded from:

https://downloads.freepascal.org/fpc/contrib/pas2js/2.0.6/

A version exists for Windows, Linux and MacOS. It can be unzipped anywhere on your harddisk.

When pas2js is installed, the first thing to do is install the pas2jsdsgn package in the Lazarus IDE. It adds some dialogs and wizards to the Lazarus IDE that help with pas2js applications: in the Packages - Install/Uninstall packages menu, the pas2jsdsgn package can be selected in the right-hand side list of available packages, as shown in *figure 2 on page 7*. After selecting the correct package and clicking the Save and rebuild IDE button, the IDE will prompt you for confirmation (*figure 3 on page 7*) When this is done, the IDE will recompile itself and install the new package. The new package will install some items in the Project - New project dialog, and creates a page in the Tools - Options menu (figure 4 on page 8). This page must be used to tell the Lazarus IDE where it can find several external tools:

- The **pas2js** compiler.
- Simpleserver: A small HTTP server to launch when you want to serve the application files. pas2js distributes a modified version of this server, called compileserver.
- The TCP/IP port where the HTTP server is supposed to listen.
- Where to find the browser executable you wish to use to run the application.
- Where to find the node.js executable you wish to use to run node.js applications.

The default entries are macros that append the correct extension for your platform, and which assume that the binary is in the **PATH** somewhere.

You can write **pas2js** source code if you don't set these options correctly, but you won't be able to compile or run the code.

PAGE 7/14





🕸 Confirm new package set for the IDE 🛛 🗙 🗙				
New package set	Action	Old package set	^	
pas2jsdsgn (>=1.0.1)	new			
FCL (>=1.0.1)	keep	FCL		
LazUtils (>=1.0)	keep	LazUtils		
CodeTools (>=1.0.1)	keep	CodeTools		
LCLBase (>=2.0.10)	keep	LCLBase		
DateTimeCtrls (>=1.5.1)	keep	DateTimeCtrls	~	
Continue Cancel				

Figure 3: Confirm installation of the pas2js package

PAGE 8/14

REAL-WORLD APPLICATIONS WITH **PAS2** JS

 (filter) Environment Files General Window IDE CoolBar Editor ToolBar Component Palette Form Editor 	 Path of pas2js.exe C:\lazarus\pas2js-windows-2.0.6\bin\i386-win32\pas2js.exe Path of simpleserver.exe C:\lazarus\pas2js-windows-2.0.6\bin\i386-win32\compileserver.exe Port numbers to start allocating from simpleserver.exe 4001 Browser to use when opening HTML page SMakeExe(IDE.firefox) 			
 Form Editor Object Inspector Messages Window FPDoc Editor Backup Naming File Filters Pas2JS 	SMakeExe(IDE, firefox) Node.js executable SMakeExe(IDE, nodejs)			
Help OK Cancel				

When the package is installed and configured, the Project - New project dialog gets 2 new options - see figure 5 on page 9:

- Web Browser Application this starts an application designed to run in the browser.
- **Node.js Application** this starts an application designed to run in Node.js.

From the **Lazarus** point of view, there is not much difference between these applications. They both generate some boilerplate project code, and for the Web Browser Application an initial HTML page is generated in which the Web Application will be running.

When you choose for the Web Browser Application project type, an additional dialog appears which allows you to set some options, see figure 6 on page 10. The options have the following meaning:

- **Create initial HTML page** A HTML page is created as part of the project. This is the HTML page in which your code will be run.
- Maintain HTML page this is an experimental option: the IDE will try to change the project filename when you rename the project.
- Run rtl when all page resources are fully loaded This option changes the code in the script tag in the generated HTML. Instead of running the rtl (and hence your program) directly:

rtl.run();

The RTL is only run when all page resources are loaded by the browser:

```
window.onload=function() {
rtl.run();
}
```



🕸 Create a new project		×
🕹 🚞 Project	^	Description
- 🔲 Application		A pas2js program running in the
- 🔟 Simple Program		browser
- 🔟 Program		
- 🔲 Console application		
- 🔟 Library		
- 🖩 FPCUnit Console Test Application		
- 🖩 FPCUnit Test Application		
🔳 InstantFPC program		
- 🔟 Web Browser Application		
🔟 Node.js Application	¥	
< >		
Help		OK Cancel

Figure 5: New project types for developing with pas2js

- Use browser application object. When this is unchecked, the project code generated by the IDE is absolutely minimal: an empty program. When checked, the IDE will define an application object for which you must fill the DoRun method. This application object resembles the TApplication or TCustomApplication objects found in regular desktop applications. The class has some methods for finding elements in the HTML and handling the URL query string (which is treated as the environment variables in a regular application).
- Use browser console unit to display writeln output By default, the output of writeln statements is displayed in the browser debug console. When this option is checked, the browserconsole unit is included in the program uses clause. This will cause the output of writeln to be displayed in the HTML page, in a specially styled DIV element, in addition to being displayed in the browser debug console.
- **Project needs a HTTP server** Some pages can be displayed and run correctly when you open the HTML file on disk by double-clicking it in the file explorer. Pages that load resources in Javascript using Fetch or XMLHTTPRequest will fail, they need a HTTP server to load the page from. Checking this option offers you 2 choices:
 - 1. To start a HTTP server on the specified port, and the page can be loaded by pointing the browser to http://localhost:NNN/yourhtmlfile.html with NNN the indicated port: the IDE will do this when you run the application.
 - 2. You can enter an URL which will be opened using the system browser when you run the application. You can use this when you have a separate HTTP server (apache, NGinx or somesuch) configured to run your application.

The following project source is generated with the default options and **Use browser application object** set:

program Project1;

{\$mode objfpc}



PAGE 10/14

🕸 Pas2JS Browser project options 🛛 🚽 🚽	
 ✓ Create initial HTML page ✓ Maintain HTML Page ✓ Run rtl when all page resources are fully loaded ✓ Use Browser Application object Use Browser Console unit to display writeln() output ✓ Project needs a HTTP Server ④ Start HTTP Server on port ✓ 4001 ✓ ✓ Use this URL to start app 	<pre>uses browserapp, JS, Classes, SysUtils, Web; type TMyApplication = class(TBrowserApplication) procedure doRun; override; end; procedure TMyApplication.doRun; begin // Your code here Terminate; end; var Application: TMyApplication; begin Application:=TMyApplication.Create(nil);</pre>
Figure 6: New project types for developing with pas2js	Application.Initialize; Application.Run; end.
Application.Free; Statement to this code. This statement must be deleted. The follow html<br <html <br="" lang="en
<head>
<meta http-equ
content=" text=""><meta name="vi
initial-scale=
<title>Project
<script src=" p<br=""/> <body> <script> rtl.run(); </script> </body></html> \end{document}	<pre>ing HTML page is generated: > "> iv="Content-type" html; charset=utf-8"> ewport" content="width=device-width, 1"> 1 roject1.js"></pre>
The whole HTML can be changed, but 2 eler the generated Javascript program: <script src="project1.js"></script> The second snippet runs your program:	<pre>ments should remain. The first element includes</pre>
As long as these 2 tags are present, your con is loaded in the browser.	npiled program will run as soon as the HTML



PAGE 11/14

INTERACTING WITH THE HTML

To show how to interact with the HTML from a pas2js program, we'll program a small Login dialog. The html of the page for a login dialog can be designed as follows:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Login demo</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Questrial&display=swap">
<link rel="stylesheet"
href="https://unpkg.com/bulma@0.9.3/css/bulma.min.css">
<link rel="stylesheet"
type="text/css" href="login.css">
<script src="login.js"></script>
</head>
<body>
<section class="hero is-success is-fullheight">
<div class="hero-body">
<div class="container has-text-centered">
<div class="column is-4 is-offset-4">
<h3 class="title has-text-black">Login</h3>
<div class="box">
<form action="javascript:void(0)">
<div class="field">
<div class="control">
<input id="edtEmail"
class="input is-large"
type="email"
placeholder="Your Email"
autofocus="">
</div>
</div>
<div class="field">
<div class="control">
<input id="edtPassword"
class="input is-large"
type="password"
placeholder="Your Password">
 </div>
</div>
<button id="btnLogin"
class="button is-block is-info is-large is-fullwidth">
Login <i class="fa fa-sign-in"
aria-hidden="true"></i>
</button>
</form>
</div>
</div>
</div>
</div>
</section>
<script>
rtl.run();
</script>
</body>
</html>
```

Blaise Pascal Magazine 99/100 2021
REAL-WORLD APPLICATIONS WITH PAS2 / JS

PAGE 12/14



Figure 7: The login page as displayed in the browser

The application declaration contains a variable for every HTML tag that has an ID attribute: 2 edits and one button; the types of the elements are the plain javascript HTML classes that correspond to the used tags (*they are defined in the web unit that comes with* **pas2js**).

The BindElements method called from vareuse the GetHTMLELement method of the TBrowserApplication class to find the element in the HTML page.

To the btnLogin element we attach an event handler that will be called when the 'click' event occurs.



REAL-WORLD APPLICATIONS WITH PAS2 / JS

The AddEventListener method is the standard recommended method to attach an event handler to a HTML element. The DoLoginClick method declaration can be automatically constructed by the Lazarus IDE if you press CTRL-SHIFT-C when the cursor is on the DoLoginClick identifier in the AddEventListener method parameter list.

The method uses the values of the email and password edits to call a hypothetical doServerLogin method. At this point, that method just shows a message to the user:



The rest of the code is the boilerplate code the IDE has generated for you when you created the project. When run, and you click the 'Login' button, the browser will respond with a message, as shown in figure 8 on page 15

A second way to program the Login dialog is to use the webwidget unit. This unit contains a set of components that map to the HTML tags; but they are components, not plain Javascript classes.

So the fields in our application class become:

```
edtEmail: TTextInputWidget;
edtPassword : TTextInputWidget;
btnLogin : TButtonWidget;
```

Since we cannot yet design these components visually, they must be created in code.

This happens in the BindElements call. This code is not different from similar code in a LCL or VCL application:

```
procedure TMyApplication.BindElements;
begin
    edtEmail:=TTextInputWidget.Create(Self);
    edtEmail.elementID:='edtEmail';
    edtPassword:=TTextInputWidget.Create(Self);
    edtPassword.ElementID:='edtPassword';
    edtPassword.Refresh;
    btnLogin:=TButtonWidget.Create(Self);
    btnLogin.ElementID:='btnLogin';
    btnLogin.Text:='Login <i class="fa fa-sign-in" aria-hidden="true"></i>';
    btnLogin.Text:='Login <i class="fa fa-sign-in" aria-hidden="true"></i>';
    btnLogin.Text:='Login <i class="fa fa-sign-in" aria-hidden="true"></i>';
    btnLogin.Refresh;
    end;
```

REAL-WORLD APPLICATIONS WITH PAS2 / JS

PAGE 14/14



The ElementID property tells the component that the HTML tag for this component is already present in the HTML page and has the ID which you assign to the property. The Refresh method will either generate the necessary HTML for the component, or links the component to the existing HTML tag in the page.

For the Button widget, the button Text is set, and TextMode must be set to tmHTML, to allow the button to be correctly rendered: the text must be rendered as HTML, not as-is. To react on the click, the OnClick event handler must be set. It is similar to the one for the plain Javascript classes version of the page:

```
procedure TMyApplication.doLoginClick(sender : TObject; event : TJSEvent);
begin
    DoServerLogin(edtEmail.Value,edtPassword.Value)
end;
procedure TMyApplication.doRun;
begin
    BindElements;
    btnLogin.OnClick:=@DoLoginClick;
end:
```

The page will of course look identical to the page when using plain Javascript classes.

8 CONCLUSION

Programming a basic GUI in a browser application is not difficult and not very different from programming a LCL application. In the next article, we'll show how to actually implement a login call using a server written in Lazarus.





ADVERTISEMENT





Python for Delphi project By Max Kleiner Catseye project By David Dirkse Wrongfully accused of kidnapping his son: Chad Hower Getting started with GIT By Michael van Canneyt TMS FNC components for Lazarus: RichEditor By Detlef Overbeek New components for Lazarus By Detlef Overbeek & Mattias Gaertner

Subscription + Lazarus Handbook (Download) $\in \mathbf{75}$ normal price: 40 + 70 = \in 110

Ex Vat 9% Ex shipment

https://www.blaisepascalmagazine.eu/product/lazarus-handbook-pocket-subscription/



The new Blaise Library App (LIB APP)

Issues 1- 100 / Pages 5809 / 873 / Articles 873 / Size 10 Gig

Price: € 100 including USB Libstick card and 1 year subscription



Page 1/9

First of all, FastReport VCL is a report generator. Based on this conclusion, we can assume that previewing or embedding documents of other formats into a report is excessive.

But according to requests from our customers, we can see the opposite point of view. Many of our customers are solving tasks with the integration of new and old systems. Such an old system may contain already prepared documents that need to connect to the news reports. In other words – embed and view documents of third-party formats into a report adding new pages or fields to it.

For such tasks in FastReport VCL 2021.3, we have added the new report object - TfrxPDFView. This object uses the Open-source library Pdfium and is designed to view PDF documents in a report. You can build this library from the source code or use one ready-to-use from FastReport VCL installation package (frx_pdfium.dll and frx_pdfium_64.dll in Bin folder). After the manual build, you need to rename the library to frx_pdfium.dll for 32-bit system and to frx_pdfium_64.dll for 64-bit system. For the manual build, you need Visual Studio and Google depot_tools. More information can be found here and here.

As was written above you don't need to build Pdfium, you can use prepared libraries from the FastReport VCL installation package, but if your company has a security limitation and requires building all code on your side use links from above.

The TfrxPDFView can draw documents both on one and several report pages. Using the ability of the report engine to split objects. Let's check more closely how to use this object and what ability it provides.

After clear installation latest FastReport VCL 2021.3 version this component should appear on Delphi's components palate in FastReport VCL section.





PDI – U	F DOCUME SING OF N	ENT IN NEW TF	A REP RXPDI	ORT VIEW	OB.	JECT		F	Page	2/9
	-RDemo - RAD St	udio 10.3 - l	Jnit1							
F	ile Edit Searc	h View	Refactor	Project	Run	Compone	ent MMX	Tools	AQtime	Documer
1	ð 📔 ~ 🗄	861	0 0	6 7	6	N 	~ 📭 ~	11 =	9 P	• G
St	ructure					7 ×	Welcome	Page	Docum	entation
老司	🕹 🕹 🚭						FastRep	ort 6.0 De	mo	
	 Form1 frxADOCor frxBarCode frxBarCode frxBIFFExpe frxBIFFExpe frxChartOb frxChartOb frxCheckBe frxCrossOb <	nponents1 Object1 ort1 ject1 oxObject1 ject1 ort1 r1 ontrols1 port1 ixExport1 rt1 npressor1 ivFxnort1				~		Sales World World Porto Porto Count Major Other fea Rotat HTML Keep Repc	Fast (C) 1998-2 (C) 19	Repo 020 by Fast 2 2 2 2 2 2 2 2 2 2 2 2 2
O	oject Inspecto	r				₽ ×		Endle:	geList1 <u></u> ^{SS} frxCSVfr	xGIFExport1
fr	«PDFObject1 Tfr	xPDFObject	1			\sim		Dat	, d RT	ŧ <mark>,</mark> ≝≦ [™]
Pi	roperties Even	ts LivePindi	nas Dosiar	or		Q		PrfrxR	eport1ch0 A3 on A4 (s	bjfrxADOCor cale)
	Name	frxPDFOb	iject1					Print /	A5 on A4 (j	oin)
	Tag	0	-					Encry	ption (type de	'1234')
Q	uick Edit						<	Gauge	e de	_
A	ll shown						۵ 🔶	22	2:56 In	sert
M	essages									
В	uild Output									







R FastReport - Untitled.fr3



Then click on the PDF Object in the report designer objects panel and create a new object on the band in the report. Just like any other FastReport VCL object.





Then click on the PDF Object in the report designer objects panel and create a new object on the band/in the report. Just like any other FastReport VCL object.

The object was created. After that, the object editor should appear which allows loading a PDF document.

IfrxPDFView allows printing multi-page documents. Special for this TfrxPDFView has new properties.

DetailStretchMode property is responsible for displaying the content inside the container of the object and can be one of the following values:

pdOneToOneStrongStretch

– a page is always stretched using report object size. It does not keep the aspect ratio. pdOneToOneNormalize

– a page is always stretched and keeps the aspect ratio of the original page in a PDF document. pdManyToOneNormalize

 allows to fit several pages into the report object area and keeps the aspect ratio of the original page in a PDF document.





PDF DOCUMENT IN A REPORT – USING OF NEW TFRXPDFVIEW OBJECT







Page

Printing of multi-page PDF document

FastReport VCL allows printing PDF documents as one page of a PDF document onto one page of the report for each page of a PDF document.

We already created and load a PDF document on the Report Title band in the report. Let's stretch it to the whole page of the report template page. It should look like the picture below.



If we click on the report preview now, we will see only one page of the load PDF document even when a document has more than one page. To print all the pages, we need to perform a series of steps.

Step 1. Turn on AllowSplit and Stretched properties and set them to True in the band object which has TfrxPDFView as a child. We need this band for stretches and splits.

Step 2. Set StretchMode property to smActualHeight or smMaxHeight on the TfrxPDFView object. This object can stretch now.

Let's run the report preview and check the report output.





PDF DOCUMENT IN A REPORT – USING OF NEW TFRXPDFVIEW OBJECT

Page 8/9

A C S S S S S S S S S S S S S S S S S S		
<text><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/><image/></text>		Î
t 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		~



PDF DOCUMENT IN A REPORT – USING OF NEW TFRXPDFVIEW OBJECT

FastReport VCL prints all pages of PDF document each on the report page.

The TfrxPDFView object has and other properties. The "Password" property sets a PDF document password. "DrawOptions" flags allow controlling document output.

According to the second sec	nd (DrawOptions.Annotations = True) comment (annotation), it conly when tions->Annotations is an annotation, there election of text.The nt is rendered in black te due to DrawOptions-> ale. Lis green and the border CL Developer	
www.fact-report.com	1/45	
Password Password: 123123	FileLink	
FastReport VCL Developer Manual	FastReport VCL Developer Manual	
	-10.00 (pp. 100)	

In addition, you can use a file name as a source for the TfrxPDFView object by using the FileLink property. And that's not all! With the new DataLink property, it's possible to load documents even through HTTP and HTTPS protocols.

Now FastReport VCL can print PDF documents inside a report and not only generate them!



THE DELPHI COMPANY

-est 1998-

OS X Android iOs Windows

Four Platforms One develop environment One Expertise

Vier platforms Eén ontwikkelomgeving Eén expertise

DELPHI

www.delphicompany.nl info@delphicompany.nl

PAGE 1 / 20

By Danny Wind



A expert

starter

As a recap of the previous article, what are the deployment options for a web service written in **Delphi?** If you create a new web application in **Delphi** you have the following options in the wizard



One option is missing from this list; it's also possible to create a **Windows** service application and embed the **Web Broker web serve**r using **Indy** into it. This is mostly interesting for embedded and industrial applications where running a full fledged web

server is no option.

Apache dynamic link module	An Apache module. Apache has support for HTTP and
	HTTPS. The current Apache 2.4 is supported on x64 Linux.
Stand-alone console application	A stand-alone WebBroker console application is a
	web server that has a text-only user interface.
	It supports HTTP using an Indy HTTP server component.
Stand-alone GUI application	A stand-alone WebBroker application is a web server that
	displays a form. It supports HTTP using an Indy HTTP
	server component.
ISAPI dynamic link library	An ISAPI library integrates with IIS. IIS has support for HTTP
	and HTTPS.
CGI stand-alone executable	A CGI executable integrates with a web server.
	Note that CGI is typically slower and more difficult to debug
	than ISAPI or an Anache module





The stand-alone

options and the **Windows** web service possibility do not use the **IIS** or **Apache** web server platform to hook into for receiving and handling HTTP requests, but instead rely on the **Indy HTTP** web server component. This works reliably and can even support **HTTPS** with **OpenSSL**, although the wizard seems to suggest otherwise. However it is less suited for server environments that are constantly maintained, updated and modified and more suited for static installations. System administrators are usually well versed in handling updates and even migrations of **IIS** and **Apache** installations, but are less aware of custom web server solutions such as **Indy**. This is one of the reasons that stand-alone **Indy** web servers are mostly found in self managed server environments or in embedded industrial installations.

> Of these options **ISAPI** and **Apache** make the most sense for deployment on production machines. Both **ISAPI** and **Apache** can be run on default web server installations of **Windows** and **Linux**. You can just order a **Virtual Private Server** with any vendor and have your web service up and running on the internet without any issues. Of course you can also use your own server. Because **IIS** and **Apache** are widely supported you can set up monitoring on the web server with one of the many available tools or delegate maintenance and administration of it to a third-party. Both also support HTTPS encryption, made easy through **Let's Encrypt** certificates or if you need, higher level certificates from one of the other Certificate Authorities.

Let's Encrypt certificates can be created and maintained on your own server with a certificate bot. However if you use a third party provider for your Apache Linux installation they will usually offer their own free solution to installing and maintaining Let's Encrypt certificates.

GETTING STARTED WITH LET'S ENCRYPT: https://letsencrypt.org/getting-started/

In this article we will focus on using **Apache** and **Linux** for our web service.

Apache is widely used as a web server second only to **NGINX** (*pronounced as EngineX*). **NGINX** surpassed **Apache** in global usage somewhere in 2019.

For dynamic modules however **Apache** is very good and also free to use. **NGINX** added dynamic modules to its capabilities in 2016, where Apache has supported them from the beginning and has been designed as such.

Also with **NGINX** you need to move from the free **Open Source** version to its paid **NGINX** Plus to start using modules.

There are other technical differences as well. **NGINX** was created in response to **Apache** limitations on the number of concurrent connections per server, and can handle 10.000+ connections on a single server where **Apache** handles 1000+ connections.



However when

handling dynamic modules, the concurrency

model, **MPM** or **multi-processing model**, of **Apache** assigns one thread to each connection allowing for efficient code execution within a dynamic module, where **NGINX** uses workers, where each worker handles multiple connections with one thread and several dynamic modules will share a thread.

Depending on the complexity of your code, with dynamic modules, either the **Apache** or **NGINX** approach will give you the best performance and the most connections per server. It's not as clear cut as it may seem at first. For us **Apache** is supported in **Delphi** and works very well for most web services.

And Linux? Linux is the second most used web server.

The first web server operating system is actually **Unix**, and **Windows Server** is the third. However **Unix** and **Linux** are very similar in their underlying architecture (50% of the Unix systems are Linux) and as long as the **Unix** you'd like to use is **Intel x86/ AMD64** based and supports running **Apache**, you can also run your **Apache** modules on that **Unix** system. Did you know that **Linux** is also the underlying system for **Google's Android?**

Unix and Linux usage statistics https://w3techs.com/technologies/details/os-unix

To compile to and run our web service on a **Linux** machine we need to perform these three preparation steps:

- A. Install a Linux machine
- B. Install libraries needed for **Delphi** on this machine
- C. Install Apache on this machine

After these steps we will create an **Apache** dynamic link module and deploy and run it on **Apache**. The **Apache** module will use the same source code of our previous article on web services.

The installation of a **Linux** machine has been made a

lot easier with the introduction of **WSL**, **Windows Subsystem** for **Linux**. **WSL** allows you to run a **Linux** machine within your **Windows** machine with a few simple steps. **Microsoft** envisions **WSL** as a tool for developers.

As they state in the **WSL FAQ** "This **(WSL)** is primarily a tool for developers, especially web developers, those working on open source projects, or deploying to **Linux** server environments.".





FAQ for WSL

https://docs.microsoft.com/nl-nl/windows/wsl/faq

What **WSL** does is that: it

makes it easy to install and run a Linux machine on your **Windows** desktop with just a few **PowerShell** commands. The default distribution in **WSL2** is **UBUNTU 20.04 LTS. WSL** has been developed in a joint effort between **Canonical** and **Microsoft**, so the default choice of **Ubuntu** makes sense. However other **Linux** distributions are supported as well. For **Delphi** using **Ubuntu** is a good default so we'll use that, but **Delphi** and **WSL** also support other distributions. **Delphi** 11 also runs fine on **Windows-11** with **WSL2**.

Delphi 11 Alexandria Linux Platforms

Windows 10 computer with WSL2 (Windows Subsystem for Linux) Ubuntu 20.04 LTS Ubuntu 18.04 LTS

RedHat Enterprise Linux (version 8)

Before we go ahead and install our Linux machine with **WSL2** on **Windows**, there are some important considerations you need to be aware of. First let's take a look at the differences between **WSL1** vs **WSL2**

WSL1	WSL2
Uses processes	Uses Hyper-V
Console only	Console and GUI
Windows 10	Windows 11
(and Windows 11)	(and Windows 10 2004)

WSL1 uses processes that are instantiated by the **LXSS** manager service to execute **Linux ELF64** binaries. These **Linux** binaries can then access exposed **Linuxcompatible** kernel interfaces. Not all of **Linux** is supported, for example; you can not run Docker within a **WSL1** instance. **WSL1** is lightweight.

WSL2 uses **Hyper-V** and is essentially running a Virtual Machine. This means more is supported, such as direct access to the **GPU** for acceleration, running **GUI** based **Linux** applications, but also support for running **Docker** instances within your **WSL2** instance. **WSL2** uses more resources.

For this article we will be using **WSL2** which uses **Hyper-V**. If you are running your **Windows** and **Delphi** inside a virtual machine in **VMWare** or **VirtualBox** and would like to run **WSL2** inside this virtual machine as well, please enable **Intel VT-x/EPT** or **AMD-V/RVI** to allow for nested running of virtual machines inside virtual machines. If you are using a Hyper-V virtual machine you can enable nested virtualization with this **PowerShell (Admin)** command.





Set-VMProcessor -VMName <VMName> -ExposeVirtualizationExtensions \$true

If you are running **WSL2** on your bare metal host, but also have **VMWare** or **VirtualBox** installed, then the new **Hyper-V** installation of **WSL2** may conflict with your existing **VMWare** or **VirtualBox** installation. The latest updates of both should fix any issues you may encounter. These issues do not occur with WSL1, as it uses processes to run its Linux binaries and does not rely on **Hyper-V**.

For both running **WSL2** on the bare metal host and as a nested virtual machine the host also needs to have virtualization enabled in the machine's **UEFI/BIOS**.

Let's install a Linux machine by installing WSL2 on Windows.

- Check if your Windows version supports WSL2
 Select the Windows logo key + R, type winver, select OK and check if its Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11.
- If all is OK open an elevated PowerShell
 On Windows 10 right click on the Start menu and Open "Windows PowerShell (Admin)",
 on Windows 11 the entry is called "Windows Terminal (Admin)"
 - 3. On the command line in **PowerShell** type

sl --install -d Ubuntu

Administrator: Windows Power? × +

Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements: https://aka.ms/PSWindows

PS C:\Users\Delphi> #sl --install -d Ubuntu Installing: Virtual Machine Platform Virtual Machine Platform has been installed. Installing: Windows Subsystem for Linux Windows Subsystem for Linux has been installed. Downloading: WSL Kernel Installing: WSL Kernel WSL Kernel has been installed. Downloading: GUI App Support Installing: GUI App Support GUI App Support has been installed. Downloading: Ubuntu The requested operation is successful. Changes will not be effective until the system is rebooted. PS C:\Users\Delphi> |



n

×

After the installation completes reboot your machine. It will then automatically complete the installation and ask you to set up the administrator (root) login for the Linux machine

O Ubuntu

4.

Installing, this may take a few minutes... Please create a default UNIX user account. The username does not need to match your Windows username. For more information visit: https://aka.ms/wslusers

Enter new UNIX username:

- 5. Use a simple user name that's different from your Windows user name and use small caps
- 6. Next make sure the Linux machine is up to date with its package list. It actually asks you to do this, so go ahead and type sudo apt update
- 7. If there are packages that need updating, the number of packages that can be updated will be displayed

💽 delphi@Windows11_Beta: –

Get:42 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [540 B] Fetched 20.4 MB in 8s (2708 kB/s) Reading package lists... Done Building dependency tree Reading state information... Done 69 packages can be upgraded. Run 'apt list --upgradable' to see them.

You can update this list of packages with 8.

sudo apt upgrade

- 9. After the upgrade of our 69 packages we now have a fresh and up to date **Ubuntu Linux** machine, and our first major step, Install a **Linux** machine, is completed.
- 10. We do not need to do a sudo apt dist-upgrade as our Ubuntu image is already up to date when downloaded through the **WSL** installation
- 11. If you reboot your Windows machine the Linux machine will shutdown automatically, but you can also manage your **Linux** machine from within a new PowerShell window

Lists the machines you have installed
Starts your new Ubuntu machine
Shutdown your Ubuntu machine
Manually shuts down all Linux machines
(you can not use sudo shutdown)
List of the wsl commands

More WSL commands here: https://docs.microsoft.com/en-us/windows/wsl/basic-commands





12. Starting your **Ubuntu** after a **Windows** reboot is as easy as/ typing in **Ubuntu** in a **PowerShell** window

🔕 delphi@Windows11_Betz - 🛛 X + 🗸	×						
Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.							
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows							
PS C:\Users\Delphi> Ubuntu Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)							
<pre>* Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://ubuntu.com/advantage</pre>							
System information as of Wed Oct 27 12:48:57 CEST 2021							
System load: 0.08 Processes: 8 Usage of /: 0.8% of 250.98GB Users logged in: 0 Memory usage: 17% IPv4 address for eth0: 172.18.98.49 Swap usage: 0%							
0 updates can be applied immediately.							
The list of available updates is more than a meek old. To check for new updates run: sudo apt update							
This message is shown once a day. To disable it please create the /home/delphi/.hushlogin file. delphi@Windowsll_Beta:~\$							

13. The settings for the new **Ubuntu** machine are stored in the **WSL** config file, should you need to change any defaults.

WSL Configuration https://docs.microsoft.com/en-us/windows/wsl/wsl-config

14. Our next major step is to install libraries needed for **Delphi** on this machine. We need to install header files for compilation, libraries for linking and running our application and **PAServer** for deployment and download of the **Linux SDK**

- 15. To be able to compile to Linux the Delphi compiler requires access to header and library files. The build-essential package on Ubuntu contains links to several packages one of which is the libc6-dev. That is a GNU C library with development libraries, with .so shared object files and header files. The build-essential package contains the Linux SDK that Delphi needs
- 16. Install the build-essential package by typing this command in the **Ubuntu** command prompt: sudo apt install build-essential



PAGE 8 / 20

17. We also need to install System.zlib for PAServer

sudo apt install zlib1g-dev

18. There are additional packages you may need depending on what functionality you use in your application

sudo apt install joe wget p7zip-full curl
openssh-server libcurl4-gnutls-dev libncurses5

- 19. Having installed all these packages and libraries in the Linux machine we now need to make the Linux SDK of libraries and files available to Delphi. Delphi uses PAServer to download the Linux SDK and to deploy and debug applications on Linux
- We need to install PAServer on the Linux machine.
 With WSL2 copying the PAServer is pretty easy, as the filesystem of Linux is accessible from the Windows Explorer.
 Notice Tux at the left bottom of the File Explorer

PAServer				- 0	×
⊕ New - 🐰 🚺 🛅	E) 🖄 🛍 🏷 Sort -	≣ View			
← → → ↑ 🚞 ≪ Local Disk (C:) > Program Files (x86) > Embarcadero	> Studio > 22.0 > PAServer	~ C	🔎 Search PASen	ver
> 🔀 Pictures	Name ^	Date modified	Туре	Size	
> 🚺 Videos	🕑 🗋 LinuxPAServer22.0.tar.gz	31/08/2021 05:49	GZ File	5.897 KB	
> 🏪 Local Disk (C:)	munger.patch	20/05/2021 11:12	PATCH File	8 KB	
> 💿 DVD Drive (D:) radstudio_11	PAServer22.0.pkg	31/08/2021 05:56	PKG File	79.936 KB	
> 😑 Shared Folders (\\vmware-host) (책: setup_paserver	31/08/2021 07:36	Application	36.422 KB	
>					
> 🐲 Network					
> 🗴 Linux					
4 items 1 item selected 5,75 MB					

21. Copy the LinuxPAServer22.0.tar.gz file over to Linux into the home/username folder, which in our case is home/delphi



PAGE 9 / 20

늘 delphi		7		- 0	×
🕀 New - 👗 🚺 ն	E) 🖄 🗒 🏷 Sort -	\equiv View \sim			
← → → ↑ ¹ → Linux → Ubs	untu > home > delphi >		v C A	Search delphi	
v A linux	Name ^	Date modified	Туре	Size	
V Duntu	andscape .landscape	27/10/2021 14:44	File folder		
host	.bash_logout	27/10/2021 14:44	BASH_LOGOUT File	1 KB	- 1
	.bashrc	27/10/2021 14:44	BASHRC File	4 KB	- 1
) dev	.motd_shown	27/10/2021 14:44	MOTD_SHOWN File	0 KB	- 1
> etc	.profile	27/10/2021 14:44	PROFILE File	1 KB	- 1
✓ home	.sudo_as_admin_successful	27/10/2021 14:44	SUDO_AS_ADMIN	0 KB	- 1
V delphi	LinuxPAServer22.0.tar.gz	31/08/2021 05:49	GZ File	5.897 KB	_ '
7 items 1 item selected 5,75 MB					

22. From the **Linux** command-line we need to unpack the gz (GNU zip) file with the command tar -xf LinuxPAServer22.0.tar.gz

🧿 delphi@Wir	nd	ows11_Bet	a: ~					– o ×	
delphi@Windows11 Beta:~\$ tar -xf LinuxPAServer22.0.tar.gz									
delphi@Windo	ОW	s11_Bet	ta:~\$ ls	; -alF				<u> </u>	
total 5928									
drwxr-xr-x	4	delphi	delphi	4096	Oct	27	15:30	./	1
drwxr-xr-x	3	root	root	4096	0ct	27	14:44	/	
-rw-rr	1	delphi	delphi	220	0ct	27	14:44	.bash_logout	
-rw-rr	1	delphi	delphi	3771	0ct	27	14:44	.bashrc	
drwxr-xr-x	2	delphi	delphi	4096	0ct	27	14:44	.landscape/	
-rw-rr	1	delphi	delphi	0	0ct	27	14:44	.motd_shown	
-rw-rr	1	delphi	delphi	807	0ct	27	14:44	.profile	
-rw-rr	1	delphi	delphi	0	0ct	27	14:44	.sudo_as_admin_successful	
-rw-rr	1	root	root	6038507	Aug	31	05:49	LinuxPAServer22.0.tar.gz	
drwxr-xr-x	2	delphi	delphi	4096	Aug	31	05:47	PAServer-22.0/	
delphi@Windo	DW	s11_Bet	:a:~\$						

23. Change directory to PAServer-22.0 and run PAServer

cd PAServer-22.0 ./paserver

24. In **PAServer** get the current IP address of the Linux Machine with the i command





delphi@Windows11_Beta: ∼/PAServer-22.0	/_	×
delphi@Windows11_Beta: <mark>~/PAServer-22.0\$./paserver</mark> Platform Assistant Server Version 13.0.11.9 Copyright (c) 2009-2021 Embarcadero Technologies, Inc.		1
Connection Profile password <press enter="" for="" no="" password="">:</press>		
Starting Platform Assistant Server on port 64211		
Type ? for available commands ≻i 172.17.194.143 ≻		

25. Go back to **Delphi** and use that IP to connect the **Delphi** Connection Profile Manager to the **Linux** Machine. The **Connection Profile Manager** can be found under Tools | Options and then under → Deployment

S Options				× هر
 IDE Default Folders Component Toolbar Environment Variables File Association Project Upgrading LiveBindings Saving and Desktop Getit Package Manager User Interface Language Version Control Deployment Connection Profile Manager Provisioning SDK Manager Translation Tools Modeling Debugger 	Connectio Profiles Linux 64-bit WSL2 Ubuntu	n Profile Mar Properties Platfor <u>m</u> : Host name: Port <u>n</u> umber: Pass <u>w</u> ord:	Linux 64-bit 172.17.194.143 64211 Enter password Iest Conr	import
	Eggin Zob	y Ingriditie E	Serve Evolution	Importa
		Sa	ave Cancel	Help

- 26. Test if the connection works with the button.
- Note that WSL2 maps network connections of your Linux machine to localhost as well, so you may also use localhost instead of the direct IP address. However this sometimes fails if there are port number or other conflicts with the Windows machine
- 28. Next go to the **SDK** manager and download the **Linux SDK** from the **Linux** machine



S Options	× هِــــــــــــــــــــــــــــــــــــ
 IDE Default Folders Compiling and Running Component Toolbar Environment Variables File Association Project Upgrading LiveBindings Saving and Desktop Gett Package Manager User Interface Language Version Control Deployment Connection Profile Manager Provisioning SDK Manager Translation Tools Modeling Debugger 	SDK versions Properties Add a New SDK × Select a platform: * Unux 64-bit * Select a profile to connect: * WSL2 Ubuntu, (Host: '172.17.194.143', Port Number: '64211') * Select an SDK version: * Ubuntu 20.04.3 LTS * Make the selected SDK active * Make the selected SDK active * Make the selected SDK active *
	Save Cancel Help

- 29. After that we will have completed the second major preparation step, install libraries needed for **Delphi** on this machine
- 30. Next we need to install **Apache** on the **Linux** machine
- 31. If you already have an **IIS** http server running on your **Windows** machine the localhost mapping of **WSL2** back to the **Windows** machine may fail as both **IIS** and **Apache** will use the same port 80. In that case either disable / stop **IIS** or install **Apache** on a different network port

Ubuntu manual for Apache installation https://ubuntu.com/server/docs/web-servers-apache

- 32. The actual install is pretty easy, we just need to run this command sudo apt install apache2
- 33. Then we can start **Apache** sudo service apache2 start
- 34. And test if it works by opening a web browser on the **Windows** side and navigate to either localhost or the IP address of our **Linux** machine









×

....



If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

35. If at any time you reboot **Ubuntu** the **apache2** service will need to be started again as well

The default installation of **Apache** under **Ubuntu**, as with most modern Linux systems, uses the event **MPM**, which is based on the worker **MPM**. This uses multiple child processes with many threads each, where each thread handles one connection at a time.

In our previous articles on web services we created a simple web service that we will use again in our Apache web module. The web service has an in-memory Key Value store. We will create a new **Apache Web Server Application** and merge that with the previously created web service.

36. Create a new **Apache Web Server Application** in **Delphi** with File | New | Other and choose Web - Web Server Application



Blaise Pascal Magazine 99/100 2021









	on	\times /		
Apache Module Options	Anselse medule			
Provide data to create the	Apache module			
	ler name			
		•		
	Apache version:			
WebBroker	Apache version 2.4 V			
	webserviceapache_module			
	Apache unit:			
	Web.HTTPD24Impl			
1700001				
TOPOOLO LO LO MACO - M				
3 of 3 << <u>B</u> ack	Next >> Finish Cancel	Help .		
40 Save a mod_ name 41 Now	all the files to a temporary direc webserviceapache which is a re . Keep these file names, do not either use your own source code e code and combine the mod we	tory. NOTE that everse of the we change them e from the previ	the project is observiceapac	called he_module lownload the
sourc web s	ervice server application. Copy	ebserviceapac only the project	che with the pr t files	eviously created
sourc web s	ervice server application. Copy	ebserviceapac only the project	t files	eviously created
sourc web s Projects ⊕ New ~ 🔏 🔲	ervice server application. Copy	ebserviceapac only the project	t files	eviously created
sourc web s Projects \bigcirc New \sim \swarrow \bigcirc \leftarrow \rightarrow \sim \uparrow \bigcirc Use	ervice server application. Copy	ebserviceapac only the project View udio > Projects	t files	earch Projects
sourc web s Projects \bigcirc New \sim \checkmark \bigcirc \leftarrow \rightarrow \sim \uparrow \bigcirc Use \checkmark Quick access	ervice server application. Copy	ebserviceapac only the project view ~ ···· adio > Projects Date modified	che with the pr t files ∽ C	earch Projects
Sourc web s Projects ↔ New ~ & □ ← → ~ ↑ □ « Use ~ Quick access DelphillenWindows11 →	ervice server application. Copy	ebserviceapac only the project view ~ ··· edio > Projects Date modified 27/10/2021 17:56	che with the pr t files ✓ C ♀ Sa Type JSON File	earch Projects
Sourc web s Projects Projects New ~ ☆ □ ← → ~ ↑ □ « Use ~ ↓ Quick access □ Delphi11enWindows11 #	ervice server application. Copy	ebserviceapac only the project view ~ ···· edio > Projects Date modified 27/10/2021 17:56 27/10/2021 17:56	che with the pr t files ✓ C ♀ Si Type JSON File Delphi Project File	earch Projects Size 8 KB 2 KB
Sourc web s Projects ↔ New ~ 🔏 🚺 ← → ~ ↑ 🚔 « Use ~ ♀ Quick access Delphi11enWindows11 # Source #	ervice server application. Copy	ebserviceapac only the project view ~ ··· dio > Projects Date modified 27/10/2021 17:56 27/10/2021 17:56	t files ✓ C ♀ Su Type JSON File Delphi Project File	earch Projects Size 8 KB 2 KB 59 KB
Sourc web s Projects New ~ ☆ □ ← → ~ ↑ □ « Use ~ ↑ Quick access Delphi11enWindows11 # Source # Desktop #	ervice server application. Copy	ebserviceapac only the project wide > Projects Date modified 27/10/2021 17:56 27/10/2021 17:56 27/10/2021 17:56	 che with the print files → C	earch Projects Size 8 KB 2 KB 59 KB 1 KB
Sourc → Projects → New → ☆ □ ← → → ↑ ■ « Use ↓ Quick access Delphi11enWindows11 # Source # Desktop # ↓ Downloads	ervice server application. Copy	ebserviceapac only the project image: state	 the with the print files C 2 Solution Type JSON File Delphi Project File LOCAL File IDENTCACHE File 	earch Projects Size 8 KB 2 KB 59 KB 1 KB 1 KB 1 KB
Sourc web s Projects New ~ ☆ □ ← → ~ ↑ □ « Use ~ ★ Quick access Delphi11enWindows11 # Source # Desktop # Downloads # Downloads #	ervice server application. Copy	ebserviceapac only the project Image: state	 the with the print of the print of	earch Projects Size 8 KB 2 KB 59 KB 1 KB 1 KB 1 KB 1 KB
Source Projects Projects New ~ 36 □ C Use C → ~ ↑ □ « Use C Use C DelphillenWindows11 # Downloads D Desktop Documents Pictures	ervice server application. Copy	view ~ wio > Projects Date modified 27/10/2021 17:56 27/10/2021 17:56 27/10/2021 17:56 27/10/2021 17:56 27/10/2021 17:56 27/10/2021 17:56 27/10/2021 17:56	 with the print files G 2 Si Type JSON File Delphi Project File LOCAL File IDENTCACHE File Delphi Form Delphi Source File 	earch Projects Size 8 KB 2 KB 59 KB 1 KB 1 KB 1 KB 1 KB 1 KB



42 and place them in the **WebServiceServer** subdirectory. Do not copy the WebModuleUnit1 files, as our new **Apache** server will use the previously created **WebModuleUnit1**

Contraction Contraction Contraction				- 0	×
🗢 New - 🏑 🚺 🗍	〕 E] 12 ⊡ 1√ sor	t ~ 🔳 View ~ …			
\leftarrow \rightarrow \checkmark \uparrow 🔁 \ll Blaise_V	WebServices_2021 > Source > WebServic	ePart5 > WebServiceServer >	~ C	, Search WebS	ervic
	Name	Date modified	Туре	Size	
V Y Quick access	E FormUnit1	17/02/2021 15:00	Delphi Form	2 KB	
DelphillenWindows11 #	WebModuleUnit1	27/10/2021 18:13	Delphi Form	2 KB	- 1
Source 🖈	C mod_webserviceapache	27/10/2021 17:56	Delphi Project File	2 KB	
🛄 Desktop 🛛 🖈	Contemporary and services and the services of	27/10/2021 19:50	Delphi Project File	59 KB	
🚽 Downloads 🛛 🖈	C WebServiceISAPI	26/06/2021 17:11	Delphi Project File	1 KB	
🔛 Documents 🛛 🖈	S WebServiceISAPI	27/10/2021 17:35	Delphi Project File	58 KB	
Pictures 🛷		23/06/2021 19:17	Delphi Project File	1 KB	
26 items 2 items selected 59,7 KB	-				

- 43 Open the **ProjectGroup**, one folder up, containing the other web service client and server projects and add the new mod_webserviceapache project to the group
- 44. Enable the **Linux** target for the **Apache** module, set to release and build the project



45 The resulting .so Shared Object library file is called libmod_webserviceapache.so, a **library** that contains a module with a web service for **apache**.

This file can now be copied over to the **Linux** machine, however the apache modules folder we need to place it in requires sudo.

So we first copy it to our home folder and then copy it over to the **apache2** modules folder



PAGE 16 / 20







52 Should Apache give an error when restarting, disable the new module with a2dismod and check if all the names are spelled correctly. Keep in mind that **Linux** is case-sensitive

> #If Apache does not start correctly sudo a2dismod libwebserviceapache sudo service apache2 restart

53 List of the names

Project name in Delphi	<pre>mod_webserviceapache.dpr</pre>
Module name for the web service	webserviceapache module
Compiled shared object library	libmod_webserviceapache.so
that contains our module with the	_
web service for apache	
Chosen name for the module through	libwebserviceapache
the conf and load files in Apache	
Handler entry in the shared object	libmod_webserviceapache-handler
library, a concatenation of .so	
library name + -handler	





54. If all went well you should now see the web service when you navigate to the URI from your Windows machine. Remember that localhost is mapped to the Linux machine. http://localhost/webserviceapache × Meb Server Application + × (C http://localhost/webserviceapache ₹ Ð ... Web Server Application 55 To test its in-memory storage, run the WebServiceClient to store an additional value in the Key Value store. The **WebServiceClient** project is included in the source files for this article. 56 Compile and run the **WebServiceClient** and use the following apache **URI** to request the default added value for key 0 http://localhost/webserviceapache/KeyValue/0 🕽 Then change the **URI** to this to add a new value **Neo** with the **PUT** button 57. http://localhost/webserviceapache/KeyValue/1 Web Service Client × http://localhost/webserviceapache/KeyValue/1 GET POST PUT DELETE Neo {"result":["OK"]} 58 .Then request the value in a web browser http://localhost/webserviceapache/KeyValue/1





				<u></u>	
	🕒 localhost/webserviceapache/Key 🗙 🕂		_ /	0	×
÷	$ ightarrow$ G \oplus http://localhost/webserviceapache/KeyValue/1	≲≡	Ē	٢	
{"resi	ult":["Neo\r\n"]}				

The /r/n at the end of **Neo** are the **JSON** representation of a Carriage Return and Line Feed, because the **MemoBody** had an empty line after **Neo** when I put the value.

All looks well and is up and running and works, but we're not entirely ready yet. When we started creating this key value web service in an earlier article we defined a use case with a limited number of users. This limitation was necessary because we created the Key Value store as a simple in-memory global variable that needed to be locked for each incoming connection. This leads to serialization of requests from connections, as each request has to wait until earlier requests are finished before it can acquire the Key Value store resource.

In the previous article for **IIS** we left out the information that this works fine on a default **IIS**, because by default **IIS** is configured for one worker process. That means our module will only be loaded in-memory once. If you start tuning **IIS** performance on large servers and for special use cases, you could eventually end up creating a Web Garden with multiple worker processes. In that case the in-memory key value store would become fragmented, as each process, under each worker, can only access its own global memory.

It's like starting multiple applications, each application can only use its own memory space. In that case we'd need a different approach. But since **IIS** usually runs best with one worker process and the in-memory **Key Value** store as it is is really only suited for a limited number of users due to the serialization issue, it's just fine as is.

For **Apache** however, having multiple processes is part of its default configuration. With the event **MPM**, which is default in **Ubuntu**, when the workload increases **Apache** will increase threads for the current child process. Each connection gets its own thread. Eventually however, when the thread limit for the child process is reached, it will also create an additional child process to serve incoming requests.

We can configure this behaviour by changing the **MPM** configuration. Please take a look at the settings for StartServers, ThreadsPerChild (and ThreadLimit) as well as MaxRequestWorkers in combination with MaxSpareThreads.

How the worker (and event) MPM work: https://httpd.apache.org/docs/trunk/mod/worker.html

What about the lifetime of apache modules? The default setting for the lifetime of a child process, i.e. a running apache module is infinite. The specific setting for this is called MaxConnectionsPerChild, and its default value is 0. Per the apache manual; "If MaxConnectionsPerChild is 0, then the process will never expire", which means the primary child process, our web service module, will keep on running so long as apache is running.



Additional MPM information on each setting: https://httpd.apache.org/docs/2.4/mod/mpm_common.html

As a starting point you could use the following configuration in the mpm_event.conf file in /etc/apache2/mods-enabled.

StartServers	1
MinSpareThreads	25
MaxSpareThreads	150
ThreadLimit	150
ThreadsPerChild	150
MaxRequestWorkers	150
MaxConnectionsPerChild	0

Some additional tips; after you restart your machine the **Ubuntu** machine will be shut down. You can restart it from a **PowerShell** with: **Ubuntu** and then start **Apache** from the **Ubuntu** command-line with

sudo service apache2 start

Also, if you need to go back and reset your **Ubuntu Linux** machine to its default you can run these two **PowerShell** commands. Note that this will remove the existing **Ubuntu** machine and install a fresh new Ubuntu.

wsl --unregister Ubuntu wsl --install -d Ubuntu

CONCLUSION

This concludes our series on web services. In these five articles we created a key value store in a web service, a Windows client and a **JavaScript** client. We also deployed our web service **ISAPI** on Windows and looked at how **CORS** influences our deployment and now we deployed it to a module on Apache inside **WSL2**. The deployment steps for **Apache** work just as well on a cloud hosted **VPS** with **Ubuntu**.

With the base code of this web service and these articles you can now start to create and deploy your own web services.

Happy coding!

Blaise Pascal Magazine 99/100 2021







FAST REPORTS offers 50% discount on "TEAM" licenses for all products

Suitable for 2-4 developers. Included build server license.

Write to support@fast-report.com
with subject "special offer on
TEAM licence November 2021"


BUILDING AND (AUTOMATICALLY) TESTING INTRAWEB APPLICATIONS WITH DELPHI

By Bob Swart

INTRODUCTION

In this article, I will show how we can build and test IntraWeb Applications with Delphi. And by testing I do not simply mean "manual testing", but also ATOZED

- and more importantly – automatic testing. I'll demonstate two automatic testing techniques for IntraWeb Applications: Unit Testing and actual Browser Testing. Both of which can be run automatically, like a testing robot.

INTRAWEB

IntraWeb is a development Framework made by AtoZedSoftware, and already available to install with Delphi itself. For the demo, you can use the bundled or evaluation edition of IntraWeb and play along, or just use the competed example that is included with this article.

In order to start an IntraWeb application with Delphi, do File \rightarrow New \rightarrow Other, and go to the **IntraWeb** node in the project types tree.

🔇 New Items	<u>م</u>	\times
✓ ■ Delphi ■ ActiveX ■ DataSnap	IntraWeb Application Wizard	
 DUnitX IntraWeb Multi-Device RAD Server 	New DUnit Test Project Creates a new IntraWeb DUnit test project	
 TMS Forms Web Windows Other 		
Template Libraries 🔻	OK Cancel Help	





If you do not see an IntraWeb entry, you may not have IntraWeb installed on your machnie. Go to https://www.atozed.com/intraweb/download/ and download your free evaluation version.

We'll start with the IntraWeb Application itself, so in the first screenshot, double-click on the IntraWeb Application Wizard which will give us a dialog to select the Application Type, and specify the project Options as well as the Project Name and Base Directory.

	Welcome to the IntraWeb Application Wiz	and Select the time of
•	application you want to create and any ad	ditional options
	Application Type	Options
U B	 StandAlone Application (Indy) 	Use JCL Stack Trace
		<u>Pool Data Connections</u>
	StandAlone Application (Http.sys)	Ulse ScaleMM2 Mem Manager
	O ISAPI Extension	Use FastMM4 Mem. Manager
ק	O IW Library	
Ę	Project <u>N</u> ame	
	Blaise100Web	
•	Base Directory	
	ers\bob\OneDrive\Documenten\Embarca	adero\Studio\Projects\ Browse

For the example project of this article, I've selected a StandAlone Application (Indy) which is the easiest to use for now. The default options are fine as well, and as Project Name I've specified Blaise100Web.

After you click on the OK button, a new project is created, consisting of a number of units.



BUILDING AND TESTING INTRAWEB APPLICATIONS



Blaise100Web.dproj - Projects

🕫 ~ 🗊 🗉 C 🖶 🐼 ~ 🧿 ~ … 🕚

🔚 Blaise

∨ 🖪 Blaise100Web.exe

- > hereit and a state of the state of the
- > O Target Platforms (Windows 32-bit)
- > 🖹 ServerController.pas
- > 🖹 Unit1.pas
- > 🖹 UserSessionUnit.pas

The **ServerController** can be used to configure the **Web Application** itself, similar to the **VCL** Application object, affecting all client connections and sessions. The **UserSessionUnit** can be used to store any user session specific information, not available for any other (user) session. And finally, the Unit1 contains the **IntraWeb** Form itself.

Using the **IntraWeb Components**, we can design a visual web application similar to designing a regular **VCL** application. For the example project, I created a simple calculator, using a **TIWEdit** control, a **TIWLabel**, and 6 **TIWButton** controls. The editbox is named **IWEditX**, the label **IWLabelAnswer** and the six buttons resp. **IWButtonAdd**, **IWButtonSubtract**, **IWButtonMultiply**, **IWButtonDivide**, **IWButtonEquals**, and **IWButtonClear**.

0	Blaise100Web UserSessionUnit	Unit1 •
+ * _ / _ = Clear	0	0
	+ - * /	= Clear
	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·

The application should work as follows: you enter a value in the editbox, like 14, then an operator, like multiply, followed by another value in the editbox, like 3, and then click on the equals button to see the answer inside the editbox (*and the label*).

In order to remember the last operator that we clicked upon, before the calculation is done, I've defined a custom type **TOperator**, as follows:



BUILDING AND TESTING INTRAWEB APPLICATIONS



type

TOperator = (opEquals, opAdd, opSubtract, opMultiply, opDivide);

The clear button should clear the entire calculation, so it's **OK** to refresh the entire screen. The implementation of the OnClick event is as follows:

```
procedure TIWForm1.IWButtonClearClick(Sender: TObject);
begin
IWLabelAnswer.Caption := '0';
IWEditX.Text := '0';
```

ActiveOperator := opEquals;
end:

Intermediate results will be stored inside the Label,

so if you enter 14 and click on the multiply button, the value of 14 will be placed inside the label (and also retained inside the editbox). The operator "opMultiply" will need to be remembered, ready to be used when we've entered the next value and press on the equals button or another operator.

This means that the **CalculateResult** button will use the ActiveOperator to compute the new result, applying itself between the Label and the Editbox, placing the result back in the label. To keep things simple, I've only considered integer values, so we'll be using the StrToIntDef function, converting the contents of the Label and EditBox from a string to an integer (with a default value of 0).

```
procedure TIWForm1.CalculateResult(NewOperator: TOperator);
var
X,Y: Integer;
begin
X := StrToIntDef(IWEditX.Text,0);
Y := StrToIntDef(IWLabelAnswer.Caption,0);
 try
 case ActiveOperator of
    opAdd: Y := Y + X;
  opSubtract: Y := Y - X;
  opMultiply: Y := Y * X;
   opDivide: Y := Y div X;
  else Y := X;
 end:
 IWLabelAnswer.Caption := Y.ToString
 except
 on E: Exception do
  IWLabelAnswer.Caption := E.Message
 end;
ActiveOperator := NewOperator;
end:
```

This method will be called by all operator buttons, in order to make the calculator engine actually run.

Most **IntraWeb** controls can send full postback events, as well as async events, which are often called **Ajax** events. The latter are more powerful, since they do not result in a complete refresh of the browser window in the which the application runs. Which is what I'm looking for when building this simple calculator, so instead of writing event handlers for the OnClick event, we should write event handler code for the OnAsyncClick event.



PAGE 5 / 12

BUILDING AND TESTING INTRAWEB APPLICATIONS

The buttons with the well-known browser icons can be used to start a specific browser to display the **IntraWeb** calculator project. As an example, clicking on the Edge button will launch the Edge browser with the 127.0.0.1:8888 URL, showing the calculator just as we've designed it inside **Delphi**.

	127.0.0.1:8	888	× +		-	0	×
\leftarrow	\rightarrow G	(i) 127.0.0.1:88	88	ĩo (Not syncing	9	
_							
3	}		14				
	+ -	* / =	Clea	r			

We can now manually enter 14, click on the Multiply button (which will place 14 in the label), and then enter 3 in the edibox. Ready to click on the Equals button to see the answer.

This may be a small, but useful web application that can be a part of a bigger web application, and ideal to demonstrate techniques to *(automatically)* test it.

UNIT TESTING INTRAWEB APPLICATIONS

The very first screenshot in this article shows the two IntraWeb project wizards, one of which is the New **DUnit Test Project** option. The new project includes a unit called IWTestCase 1.pas with the following definition code:

type T = class(TTestCase) private protected procedure SetUp; override; procedure TearDown; override; published {\$IFDEF CLR}[Test]{\$ENDIF} procedure Test; end;

Derived from TTestCase, we should implement the SetUp and TearDown to create the **IntraWeb** Form, and inside Test add code to actually test the functionality of the calculator buttons.

The actual testing code is suggested in the comments already:



BUILDING AND TESTING INTRAWEB APPLICATIONS



procedure T.Test;	
begin	
// with NewSession do try // with MainForm as TIWForm1 do begin	
// add your test code here	
// end; // finally // Free; // end; end;	
We need to add the IntraWeb unit with our calculator form to the uses clause of the test project, as well as the ServerController and UserSessionUnit . Removing the comments from the sample T.Test method, leaving the actual Test code empty, will however not result in a working application. The Dunit project will give us an error "IW application not property initialized".	
🕐 DUnit: An Xtreme testing framework — 🗆 🗙	
<u>F</u> ile Test <u>T</u> ree <u>O</u> ptions Actio <u>n</u> s	
Test Hierarchur	
Progress:	
Score: 0%	
Tests Run Failures Errors Overrides Test T Total Time 1 1 0 1 0 0:00:00 0:00:00.007	
Test Name Failure Type Message Location	
Test Exception IW application not properly initialized. \$00724	

I have been unable to get this to

work. I welcome any feedback by the readers of this magazine, and if I can get it to work, I will post an update on my website (*see the end of this artice for my contact details*).

A discussion about this issue can also be found in the **AtoZed** Forums at https://www.atozed.com/forums/thread-973-post-2598.html





DELPHI DUNITX PROJECT FOR INTRAWEB Although the IntraWeb DUnit project doesn't seem to work, we can still create a regular Delphi DUnitX project and add the IntraWeb unit as a regular testcase. The **DUnitX** project can be found in the **DUnitX** category of the new project dialog. New Items ρ 🗸 🛅 Delphi DUnitX Project ActiveX Create New DUnitX Test Project 🛅 DataSnap 📁 DUnitX DUnitX Unit IntraWeb Create New DUnitX Test Unit Multi-Device RAD Server TMS Forms 🖿 Web Windows > 🖿 Other **Template Libraries** ок Cancel Ŧ Help

For the new **DUnitX** project, we can start by defining a number of options.



We get a similar test class, I've named it **TBlaise100WebTest**, with **Setup, TearDown** and several Test methods.

In the Setup, we need to create our IntraWeb Form, and in the **TearDown**, we need to free it. We can subsequently use the **TIWForm1** instance inside the Test method.



BUILDING AND TESTING INTRAWEB APPLICATIONS





FIWForm1.IWButtonEqualsAsyncClick(nil, nil);

System.Assert(FIWForm1.IWLabelAnswer.Caption = '42',

'Answer should be 42'); end:

The **DUnitX** project is shows the output on the console, and in this case without errors confirms that the answer is indeed 42.

C:\Users\bob\OneDrive\Documenten\Embarcadero\Studio\Projects\Blaise100Web\DUnitX\Win32\Debug\Blaise100Test.exe DUnitX - (c) 2015-2018 Vincent Parrett & Contributors License - http://www.apache.org/licenses/LICENSE-2.0 Fixture : Unit2 Fixture : Unit2.TBlaise100WebTest Test : Unit2.TBlaise100WebTest.Test1 Running Setup for : Test1 Test : Unit2.TBlaise100WebTest.Test2.TestA Test : Unit2.TBlaise100WebTest.Test2.TestB Running Fixture Teardown Method : Destroy one testing. Tests Found : 3 Tests Ignored : 0 Tests Passed : 3 Tests Leaked : 0 Tests Failed : 0 sts Errored : 0 one.. press <Enter> key to quit._

Blaise Pascal Magazine 99/100 2021





However, unlike the **IntraWeb Unit Testing Framework** (*which is appears to be broken at this time*), the **DUnitX** is not really able to offer the **IntraWeb** experience. Although we can click on buttons by called the OnAsyncClick or OnClick events, we cannot make use of the **UserSession** or **Data Module Pooling**, because there are no user sessions in the **DUnitX** test environment. This would be the additional value and main reason to use the **IntraWeb** specific **Unit Testing Framework**.

BROWSER TESTING OF INTRAWEB APPLICATIONS

Another approach to automatic testing of **IntraWeb Applications** is using an embedded browser in a **Delphi** application, sending commands to the IntraWeb application through the browser. **Delphi** used to contain a **TWebBrowser** component, based on **Internet Explorer**, but we now have the ability to use the Edge based embedded browser.

In order to install the Edge browser support in **Delphi 11**, we need to use the **GetIt Package Manager**, and search for **Edge**. The **EdgeView 2 SDK 1.0.664.37** can be installed on your machine, and includes the **32-bit** and **64-bit DLLs** that are required for successful deployment of your **Delphi** application that uses this **Edge** browser control.

Setlt Package Manager						×
 Filter All Installed Subscription Only Updates Sort By Name Vendor 			EdgeView2 SDK 1.0.664.37 by Microsoft The Microsoft Edge WebView2 control enables you to host web content in your Winds 25 Jan 2021 Microsoft.Web.WebView2 License FMX Style - WedgewoodLight 1.0 by Embarcadero Technologies FMX Multi-Device Style - Wedgewood Light.	ows app	Install	
○ Date∨ Categories		Button1	1 May 2020 Embarcadero Software License			
All Libraries Components Internet Of Things Trial	264 47 63 1 58	Forms Deedbark Edit Raturs	VCL Style - WedgewoodLight 1.0 by Embarcadero Technologies VCL Windows Style - Wedgewood Light. Updated for HighDPI.			
 Industry Templates IDE Plugins Styles C++ Libraries Tools 	4 3 43 6 4		Embarcadero Sortware License			
 New Sample Projects Promoted Patches and Hotfixes 	8 22 4 1					

The **EdgeBrowser** control replaces the old TWebBrowser control, and is easy to use. I've placed it on a VCL Form, just below a TPanel with a TComboBox with item http://127.0.0.1:8888 for the **URL** of our **IntraWeb** application, and a TButton with "GO"



BUILDING AND TESTING INTRAWEB APPLICATIONS



🔇 WebTest	—	×
http://127.0.0.1:8888/	GO	
1		

The implementation of the OnClick event for the GO button is as follows:

procedure TFormWebTest.Button1Click(Sender: TObject);
begin
EdgeBrowser1.Navigate(cbURL.Text);
end;

And this will show our IntraWeb application with the calculator:

🕖 WebTest	- 0	×
http://127.0.0.1:8888/	~ G0	
0	0	
+ - * / =	Clear	

In order to automatically test the calculator, we should place values inside the **TIWEdit** and click on **TIWButtons**. This can be done by sending **JavaScript** commands to the **EdgeBrowser** control.

For example, to perform the **Async** click events of a button we can execute the following method, passing the **TEdgeBrowser** and exact name of the **TIWButton** control:

```
procedure TFormWebTest.IWButtonClick(EdgeBrowser: TEdgeBrowser;
const Name: String);
begin
EdgeBrowser.ExecuteScript('$("#' + Name + "').trigger("click");');
end;
```



BUILDING AND TESTING INTRAWEB APPLICATIONS

The JavaScript trigger('click') method will ensure we execute the click of the TIWButton.

Adding a value to the **TIWEdit** control requires a bit more work. Initially I tried the following approach:

```
procedure TFormWebTest.IWEditValue(EdgeBrowser: TEdgeBrowser;
const Name, Value: String);
begin
EdgeBrowser.ExecuteScript(Name + '.value = ''' + Value + ''';');
end:
```

end:

However, although the value appears inside the control in the browser, the value is not updated "inside" the **IntraWeb** application. So calling the **IWEditValue** method for **TIWEditX** with value '42" will still result in the **IntraWeb Application** thinking there is no value (*or no change*) in the control. We have to explicitly tell the **IntraWeb** Application that the contents of the control has changed, by triggering the change method, which results in the following code for the **IWEditValue** method:

procedure TFormWebTest.IWEditValue(EdgeBrowser: TEdgeBrowser; const Name, Value: String); begin EdgeBrowser.ExecuteScript(Name + '.value = ''' + Value + ''';'); EdgeBrowser.ExecuteScript('\$(''#' + Name + ''').trigger(''change'');');

With these two primitives, we can build an automatic scripting scenario, starting in the EdgeBrowser1NavigationCompleted event, which indicates that the Edge Browser has completed the navigation, and is ready for more.

I found that the first time we load the **IntraWeb** application, we get two **NavigationComplete** event triggers, so we better skip the first one, and respond to the second one. As you will see in the example IntraWeb Web Test project, I'll (ab)use the Tag property of the Form (initially with the value 0) for this purpose:

Also, in order to watch the progress of the test, I add a Sleep (500) after each command that I send to the browser. For more complex applications involving database access, we usually wait 30 seconds between command to ensure we can proceed to the next step.

```
procedure TFormWebTest.EdgeBrowser1NavigationCompleted(
 Sender: TCustomEdgeBrowser; IsSuccess: Boolean;
WebErrorStatus: TOleEnum);
begin
 case Tag of
 0: Tag := Tag + 1;
 1: begin
    Sleep(500);
    IWEditValue(EdgeBrowser1,'IWEDITX','42');
    Tag := Tag + 1;
    Sleep(500);
    IWButtonCLick(EdgeBrowser1,'IWBUTTONDIVIDE');
    Tag := Tag + 1;
    Sleep(500);
    IWEditValue(EdgeBrowser1,'IWEDITX','3');
    Tag := Tag + 1;
    IWButtonCLick(EdgeBrowser1,'IWBUTTONEQUALS');
    Tag := Tag + 1;
   end;
 end:
end;
```



BUILDING AND TESTING INTRAWEB APPLICATIONS



For this example, we start by adding '42' to the contents of the IWEDITX control, then click on the IWBUTTONDIVIDE, followed by placing '3' in the IWEDITX control. A final click on the IWBUTTONEQUALS should result in the value 14 being displayed in the TIWLabelAnswer.

Using this technique, I can test simple web forms, with multiple different values and operator buttons. For more complex **IntraWeb** applications, I have added primitives to select an option (choice) in a **TIWComboBox** or **TIWListBox**, and toggle a **TIWCheckBox** or **TIWRadioButton**. These help in more complex automatic test scenarios that cannot be covered by the **DUnitX** project because it lacks the support for **IntraWeb** UserSessions.

SUMMARY

IntraWeb can be used to create visual browser based applications using **Delphi** programming techniques and components. In this short article, I could not even begin to cover the more advanced feature like User Sessions, Content Handlers and Database connectivity, but these allow you to develop powerful and scalable web applications.

Testing such applications takes time and effort, so automatic testing techniques are useful and helpful. Unfortunately, the **IntraWeb** Unit Testing Framework does not seem to work at this time, and the regular **DUnixX** framework included with Delphi does not truly support **IntraWeb** test clients. Although we can test simple events and code on an **IntraWeb** Form, it's not possible to test user session behavior for example, which makes it difficult to test real world **IntraWeb** applications.

Using **Browser Based Testing**, we can send control the contents of elements in the browser. I this paper I've limited myself to **TIWEdit** controls and **TIWButtons**, but it's not difficult to extend this to other **IntraWeb** controls, for both synchronous and asynchronous events (*as I demonstrated with the asynchronous update of the* **TIWEdit** *control*). Although the sample demo "script" in this article is small, we've used this testing technique to process complex testing scenarios (*where the end result is not displayed on* a **TIWLabe1**, *but actually written back to the database, so after the testing script, we could simply perform some* **SQL** *to check if the output was correct*).

Using automatic testing means you can build a set of testing scripts or scenarios to verify that certain features of the web application continue to work. Not just simple calculations, but also screen interactions. Running these tests at night for daily tests, as well as after each test and deployment check-in ensures that we do not encounter regressions or bugs that used to be fixed (*as long as we remember to write a test for said scenario*).

If you want to learn more about the power of **IntraWeb Web development with Delphi**, I can recommend my **Delphi XE IntraWeb XI Development** courseware manual from http://www.drbob42.com/courseware/ to get started (even if you already have some experience with **IntraWeb**).

Although the title mentions **Delphi XE** and **IntraWeb XI**, the book is still applicable today with **IntraWeb** 14 and 15. You can of course also hire me personally for a training or consultancy event, helping or designing any Delphi application, including but not limited to **IntraWeb** development.

Thank you for reading this article, and this 100th issue of **Blaise Pascal Magazine**. It's been an honor, and I hope we'll meet again in real life soon.

Bob Swart (Bob@eBob42.com) Bob Swart Training & Consultancy (eBob42) http://www.drbob42.com linkedin.com/in/drbob42



THE LIB (STICK) APP CREATING A LIBRARY PDF VIEWER

CREATING A LIBRARY PDF VIEWER WORKING WITH A DATABASE BY DETLEF OVERBEEK

BLAISE PASCAL MAGAZINE

PAGE 1 / 11

starter

expert

ABSTRACT

Because of the publishing of our hundreds issue we have fulfilled a plan which was since long one of my active plans. Since the introduction of the **LIBstick** (A collection of items on a Credit card-like USB Stick) we knew the stick might become overcrowded and so we had to find a new solution in the future. The **Adobe PDF** containing all issue in one was becoming to large and might even become unstable.

And the Future is now: the new **LIBstick** has turned into a small program. We will call it the **LIB** APP

INTRODUCTION

The LibStick as a program which is containing a PDF viewer from and made available by **TMS**, a grid from **TMS** and a small DataBase – a ClientDataSet – which has the ability of using **SQL**, preferred much over the standard ClientDataSet. I know only one component provider that has this extra:

Components4Developers.

The LIBstick must meet a number of requirements: The program itself,

- A DATABASE that can handle all the articles and the Library of all Items published by us and the code of all the articles that has code.
- 2 A SEARCH MECHANISM for finding articles or terms and authors.
- SHOW THE RESULT IMMEDIATELY, which is a problem because the LIBstick contains about 5800 pages or more than 600 articles, and that's included in PDF files that have to show the correct page and illustrations.
- **4** THE PROGRAM SHOULD BE SMALL and use as little code as possible.

Because I always try to show interesting pieces of code, I will show them in this article as well.

If you would like to have extra features in this **LIBstick** please let me know: editor@blaisepascalmagazine.eu The **LIBstick** contains of course all code and issues.





3	LIBraryFo	m			- 0	×
Sh	how articl	e Go to issue Nr Issue Search	< <first <="" next="" prior=""> Last>> Add Delete</first>	Edit Save	Cancel Refresh	CSV
ID	IssueNr	Author	Article	PDF	PageNr	^
5	1	Peter Bijlsma	Representing graphics for math functions			6
6	5 1	Detlef Overbeek	Client Dataset Toolkit			8
7	7 1	Henk Schreij	Website name checking			9
8	3 1	Herman Peeren	Database normalisation			9
9	91	Bob Swart	Delphi 2007 and VCL Component Building			13
10	1	Detlef Overbeek	Wallpaper			16
11	1	Henk Schreij	Coding for two or more types			17
12	2 1	Tim Opsteeg	Recipe for creating a cookbook			21
13	8 1	Rik Smit	Exploring Rave Reports by			25
14	1	David Dirkse	Cover story: Drawing on a changing background			33
			the second se			×





THE LIB APP PAGE 2 / 11

Search

🧟 LIBraryForm

61

1

ID IssueNr Author 5 1

Show article Go to issue Nr Issue

Peter Bijlsma

Detlef Overbeek

CREATING A LIBRARY PDF V

BLAISE PASCAL	MAGAZINE
---------------	----------

16 17

21

25 33

/IE	WER INTEGRATED WITH A DATABASE				b
			_	F	
	< <first <="" next="" prior=""> Last >> Add Delete</first>	Edit Save	Cancel Ref	Hat DoWeth, Creator of Pascal Programming language	Raise Patcal, Mathemati
	Article	PDF	PageNr	^	
	Representing graphics for math functions			6	
	Client Dataset Toolkit			8	
	Website name checking			9	
	Database normalisation			9	
	Delphi 2007 and VCL Component Building			13	

		and an		
14	1	David Dirkse	Cover story: Drawing on a changing background	
13	1	Rik Smit	Exploring Rave Reports by	
12	1	Tim Opsteeg	Recipe for creating a cookbook	
11	1	Henk Schreij	Coding for two or more types	
10	1	Detlef Overbeek	Wallpaper	
9	1	Bob Swart	Delphi 2007 and VCL Component Building	
8	1	Herman Peeren	Database normalisation	
7	1	Henk Schreij	Website name checking	

□ « » ④ ⊖ ℃ 🖶 Page Jump to page



If you start the program the only window it has will appear, containing the grid and the pdf viewer. The grid contains the number of the issues, the author and article as well the page where the article starts on.



THE LIB APP PAGE 3 / 11

🕵 LIBraryForm

CREATING A LIBRARY PDF VIEWER INTEGRATED WITH A DATABASE

BLA	ISE PA	SCAL N	/IAGAZINE
-----	--------	--------	-----------

- 🖅 🔊

						1200
S	how articl	e Go to issue Nr Issue Search	< <first <="" next="" prior=""> Last >> Add Delete</first>	Edit Save	Cancel Refr	acal Programming language
ID	IssueNr	Author	Article	PDF	PageNr	^
	5 1	Peter Bijlsma	Representing graphics for math functions			6
	61	Detlef Overbeek	Client Dataset Toolkit			8
	7 1	Henk Schreij	Website name checking			9
	8 1	Herman Peeren	Database normalisation			9
	9 1	Bob Swart	Delphi 2007 and VCL Component Building			13
1	0 1	Detlef Overbeek	Wallpaper			16
1	1 1	Henk Schreij	Coding for two or more types			17
1	2 1	Tim Opsteeg	Recipe for creating a cookbook			21
1	3 1	Rik Smit	Exploring Rave Reports by			25
1	4 1	David Dirkse	Cover story: Drawing on a changing background			33
	-					

BLAISE PASCALOMAGAZINI



- This series of buttons are quite simple to understand, you can open any PDF on your hard disk you want to have a look at. So the program allows you to act as your own PDF viewer for all purposes.
- If the Item selected has more than on page you can browse through the page back and forwards
- **3** Increase or decrease the Pdf Image
- A Rotate left or right
- 9 Print the PDF
- **6** Jump to page: in the edit field near to it you enter the page number you want to look at.



A special item is to drag and drop a pdf over the form and it will open the PDF.

>



THE LIB APP PAGE 4 / 11

CREATING A LIBRARY PDF VIEWER INTEGRATED WITH A DATABASE

erenne 🧕	100
For Country Country of Pascal Programming Language	Rate Parta, Mathematican

T

🧟 LIBrary	Form								end Prof De Wirth, Creator of Pa			
Show ar	ticle G	to issue Nr	31 Search	<<	<pre>sFirst < Prior Next > Last >></pre>	Add Delete	Edit Save	Cancel F	Refresh	CSV		
ID Issue	Vr Author			Article			PDF	PageNr		^		
368 31_32	Jim Dut	f		Book Review: Coding in	Delphi Coding Delphi Author: Nick Hodg	les				4		
369 31_32	Alexan	ler Alexeev		Designing an API: comr	non mistakes					8		
370 31_32	Michaë	Van Canneyt		Newest Leap developm	ents			-		21		
371 31_32	B.J Rao			3D Printing						26		
372 31_32	Michaë	Van Cannevt		Kinect ?!						33		
373 31_32	Primoz	G 🎇 LIBraryFo	ərm								- 0	×
374 31_32	Detlef (Show artic	le Go to issue Nr 31	Search	< <first <="" prior<="" th=""><th>Next > Last >></th><th>Add De</th><th>lete Edi</th><th>t Save</th><th>Cancel F</th><th>lefresh C</th><th>sv</th></first>	Next > Last >>	Add De	lete Edi	t Save	Cancel F	lefresh C	sv
375 31_32	David E	ir ID IssueNr	Author		Article			PI	DF	PageNr		^
3/0 31_32	Detlef	368 31 32	Jim Duff		Book Review: Coding in Delphi Coding D	elphi Author: Nick H	Hodges			, agern		4
378 31 32	Alexan	369 31_32	Alexander Alexeev		Designing an API: common mistakes	•						8
379 31 32	Max Kl	370 31_32	Michaël Van Canneyt		Newest Leap developments						;	21
380 31 32	Detlef (371 31_32	B.J Rao		3D Printing						:	26
381 31_32	David D	ir 372 31_32	Michaël Van Canneyt		Kinect ?!							33
382 31_32	Jeroen	373 31_32	Primoz Gabrijelicic		Smart Mobile Studio 2.0							41
383 31_32	Fikret	374 31_32	Detlef Overbeek		Interview with David I: plans about updati	ing buying etc						48
		75 21 32	David Dirkse		A simple superscript text editor							50
$\Box \ll$	X 🕀	376 31_32	Detler Overbeek	Divider so v	ou can enlarge the y	views of t	he grid	or the	e imao	ie.		52
		270 21 22	Bruno Florens	Sivicici, SO y	ou can charge the		ine grie	or th	c intag	,c		57 65
1423	Page 92	379 31 32	Max Kleiner		The maXhoy Pure Code Page				_			77
	, ogene	380 31 32	Detlef Overbeek		Interview with Ray Konopka							93
		381 31_32	David Dirkse		Programming Bitmap Rotation							98
	Page 93	382 31_32	Jeroen Pluimers		Introduction to Model, View and View Mo	odel (MVVM) and th	e Caliburn Micro	for			1	02
1		383 31_32	Fikret Hasovic		kbmFMX for XE5 (android)						1	13
		384 33	Max Kleiner		maXbox Starter 5 Start with DLL A Library	for All.						5
	Page 94	85 33	Rik Smit		Suite Rays components							14
<u></u>		316 33	Max Kleiner		Interview with Pofessor Dr. Niklaus Wirth							35
		387 33	Jeroen Pluimers		Steps in MindScape AppView Step-By-Ste	p						26
1.1	Page 95	388 83	Detlef Overbeek		New: Delphi XE 6							43
HOT IN		389 3	David Dirkse		Programming Iruth Table reduction							46
		390 33	Laramy North		Undating DotTapper							59
	Page 96	392 33	Fikret Hasovitch		Six new ready to use components!							65
		393 34	Cary Jensen		Ode To the Code							4
	Dage 07	394 34	Detlef Overbeek		Animation Lab for Android							5
	i uge si	395 34	Boian Mitov		Multiple Inheritance /Parallel Processing a	and GP GPU						8
		396 34	Detlef Overbeek		Interview with Boian Mitov: All about und	lerstanding the RTTI	/ Attributes / Fu	nctional				7
	Page 98	397 34	RikSmit		Raize Components Version 6PART II- Cont	inuation from issue	33					18
		398 34	MaxKleiner		QR (Quick Response)Codes							24
		399 34	David Dirkse		Image Compression							29
	Page 99	400 34	Jeremy North		Review : Documentation Insight by Deviet	t Software						30
		401 54	Gary Jernen Michael von Cannest		Unit testing - test driven development							41
· Charletter		403 34	Kim Madse		Serving an off the shelf Content Manager	ment System						53
<u> </u>	Page 100	404 35	Jim Duff		BOOK Review Delphi Cookbook Author: D	Daniele Teti						4
		405 35	Detlef Overbeek		Boian Mitov an interview							7
	D 101	406 35	Christan Budde		Smart Mobile Studio: Where to use it and	with what? Comme	nt					16
	Page for	407 35	Christan Budde		Smart Mobile Studio: Building simple nati	ive applications for '	Windows 7					17
		408 35	Christan Budde		Smart Mobile Studio: node.js on a Raspbe	erry Pi						19
	Page 102	409 35	Howard Page Clark		Debugging Techniques in Lazarus							23
		410.05	Man Minimum		and crowing							40
		[] «	» ④ Θ ℌ C 届	Page Jump to pag	e							
	Page 103		~~~001									
		0.059	^ ^									Ê
		30	Page 92		-							
<	0 104				Ray Konopka Interview with the famous writer	of Developing Cus	tom Delphi Con	ponents				
			Page 93		RAIZE							
		- <u></u>				IQUAL Control of the second	ur passion	Incon	9			
					Delphi components		ESITE	INSF	EX			
			Page 94		Develop		MAST	EK				
					DELP		ratchPad					
					Compo	nents RAIZ						
			Page 95 🗸			COMP	ONENTS					
		<	>			As you can However, I	imagine, most of my v continued to use Turk	o Pascal at hor	in⊂. ne and got			-



THE LIB APP PAGE 5 / 11

CREATING A LIBRARY PDF VIEWER INTEGRATED WITH A DATABASE







BLAISE PASCAL MAGAZINE

THE LIB APP PAGE 6 / 11

S LIBraryForm

ş- 🚴 🌄

-					Prof DcWirth, Creator of Pascal Programming language	Rain P
Sh	ow articl	e Go to issue Nr Issue Search	<pre><<first <="" next="" prior=""> Last >> Add Delete</first></pre>	Edit Save	Cance	
ID.	IssueŴr	Autor	Article	PDF	PageNr	^
854	97	Detlef Overbeek	TMS FNC components for Lazarus: RichEditor		4	9
855	97	Detlef Overbeck & Mattias Gaertner	New components for Lazarus		6	5
856	98 <	Max Kleiner	P thon4Delphi		1	3
857	98	Mattias Gaertner	How to alter your Debugger from GNU to FPC in LAZARUS		2	1
858	98	Detlef Overbeek & Ben Jones	Creating a radom password generator in LAZARUS		21	3
859	98	Michael van Canneyt	GIT continued: Contributing		50	5
860	98	Fast Reports	Fast Reports: Uploading objects to FastReport using the http and https protocols		6	5
861	98	Detlef Overbeek	Recover Icons on your Windows desktop		7:	5
862	98	Embarcadero	RAD Studio 11		70	5
863	99					
						- Y

First select the author or article then click on "show article"

	LIBraryFo	rm				-	>	<
S	how articl	le Go to issue Nr Issue Search	< <first <="" next="" prior=""> Last >> Add Delete</first>	Edit Save	Cancel	Refresh	CSV	
ID	IssueNr	Author	Article	PDF	PageNr			^
	5 1	Peter Bijlsma	Representing graphics for math functions				6	
	51	Detlef Overbeek	Client Dataset Toolkit				8	
	7 1	Henk Schreij	Website name checking				9	
	3 1	Herman Peeren	Database normalisation				9	
	91	Bob Swart	Delphi 2007 and VCL Component Building				13	
1	0 1	Detlef Overbeek	Wallpaper				16	
1	1 1	Henk Schreij	Coding for two or more types				17	
1	2 1	Tim Opsteeg	Recipe for creating a cookbook				21	
1	3 1	Rik Smit	Exploring Rave Reports by				25	
14	4 1	David Dirkse	Cover story: Drawing on a changing background				33	
								v

Enter in the Issue field the number of the issue you want to see, then click the "Go to issue Nr" button In the search field you can give in any text. It will search automatically on author or article.

	LIBraryFo	rm					×
SI	ow articl	e Go to issue Nr Issue Search	<-First < Prior Next > Last >> Add Delete	Edit Save	Cancel Refresh	CS	v
ID	IssueNr	Author	Article	PDF	PageNr		^
5	1	Peter Bijlsma	Representing graphics for math functions				6
e	1	Detlef Overbeek	Client Dataset Toolkit			1	8
7	1	Henk Schreij	Website name checking			1	9
8	1	Herman Peeren	Database normalisation				9
9	1	Bob Swart	Delphi 2007 and VCL Component Building			1	3
10	1	Detlef Overbeek	Wallpaper			1	6
11	1	Henk Schreij	Coding for two or more types			1	7
12	1	Tim Opsteeg	Recipe for creating a cookbook			2	1
13	1	Rik Smit	Exploring Rave Reports by			2	5
14	1	David Dirkse	Cover story: Drawing on a changing background			3	3
		•				-	· ·

These buttons are meant to browse through the grid.

LIBraryFo	rm			- 0	×
iow artic	e Go to issue Nr Issue Search	< <first <="" next="" prior=""> Last >> Add Delete</first>	Edit Save	Cancel Refresh	CSV
IssueNr	Author	Article	PDF	Pagelvr	^
1	Peter Bijlsma	Representing graphics for math functions			6
1	Detlef Overbeek	Client Dataset Toolkit			8
1	Henk Schreij	Website name checking			9
1	Herman Peeren	Database normalisation			9
1	Bob Swart	Delphi 2007 and VCL Component Building			13
1	Detlef Overbeek	Wallpaper			16
1	Henk Schreij	Coding for two or more types			17
1	Tim Opsteeg	Recipe for creating a cookbook			21
1	Rik Smit	Exploring Rave Reports by			25
1	David Dirkse	Cover story: Drawing on a changing background			33
	LIBraryFo is articl issueNr i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i 1 i i i i	LIBrayForm Cost of issue Nr Issue Search IssueNr Author Search IssueNr Author Search I Peter Bijlsma Search I Detlef Overbeek Search I Henk Schreij Search I Herman Peeren Search I Detlef Overbeek Search I Henk Schreij Search I Henk Schreij Search I Tim Opsteeg Search I Rik Smit Search I David Dirkse Search	LIBrayForm tow article Go to issue Nr Issue Search < <first< th=""> < Prior Next > Last >> Add Delete IssueNr Author Article I Peter Bijlsma Representing graphics for math functions I Detlef Overbeek Client Dataset Toolkit I Hern Schreij Website name checking I Herman Peeren Database normalisation I Bob Swart Delphi 2007 and VCL Component Building I Detlef Overbeek Wallpaper I Henk Schreij Coding for two or more types I Tim Opsteeg Recipe for creating a cookbook I Rik Smit Exploring Rave Reports by I David Dirkse Cover story: Drawing on a changing background</first<>	LIBrayForm tow article Go to issue Nr Issue Search < < < first < Prior Next > Last >> Add Delete Edit Save IssueNr Author Article PDF PDF <td>LlBrayform C of to issue Nr Issue Search C first < Prior Next > Last >> Add Delete Edit Save Cancel Refresh IssueNr Author Article PDF Pagenr I Peter Bijlsma Representing graphics for math functions PDF Pagenr I Detlef Overbeek Client Dataset Toolkit Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernsan Peeren Database normalisation Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Bob Swart Delphi 2007 and VCL Component Building Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Bether Overbeek Wallpaper Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernk Schreij Coding for two or more types Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernk Schreij Coding for two or more types Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernk Schreij Coding for two or more types Image: Client Dataset Toolkit Image: Client Dataset Client Dataset Toolkit I Tim Opsteeg Recipe for creating a cookbook Image: Client Dataset Dioling Dataset Dioling Dataset Dioling Dataset Dioling Dataset Dioling Bave Reports Dioling</td>	LlBrayform C of to issue Nr Issue Search C first < Prior Next > Last >> Add Delete Edit Save Cancel Refresh IssueNr Author Article PDF Pagenr I Peter Bijlsma Representing graphics for math functions PDF Pagenr I Detlef Overbeek Client Dataset Toolkit Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernsan Peeren Database normalisation Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Bob Swart Delphi 2007 and VCL Component Building Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Bether Overbeek Wallpaper Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernk Schreij Coding for two or more types Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernk Schreij Coding for two or more types Image: Client Dataset Toolkit Image: Client Dataset Toolkit I Hernk Schreij Coding for two or more types Image: Client Dataset Toolkit Image: Client Dataset Client Dataset Toolkit I Tim Opsteeg Recipe for creating a cookbook Image: Client Dataset Dioling Dataset Dioling Dataset Dioling Dataset Dioling Dataset Dioling Bave Reports Dioling

These functions are not applicable in the LIBstick because they are meant for making corrections. So in the running app you will not see them...



THE LIB APP PAGE 7 / 11

CREATING A LIBRARY PDF VIEWER INTEGRATED WITH A DATABASE

BLAISE PASCAL MAGAZINE

E......

🕵 LIBraryFor	LiBraryForm				
Show article	e Go to issue Nr Issue Mattias	< <first <="" next="" prior=""> Last >> Add Delete</first>	Edit Save	Post DcWirth, Gwasor of Pascal Programming Language	Raio Patra, Mathematica
ID IssueNr	Author	Article	PDF	PageNr	^
854 97	Detlef Overbeek	TMS FNC components for Lazarus: RichEditor			49
855 97	Detlef Overbeek & Mattias Gaertner	New components for Lazarus			60
856 98	Max Kleiner	Python4Delphi			8
857 98	Mattias Gaertner	How to alter your Debugger from GNU to FPC in LAZARUS			21
858 98	Detlef Overbeek & Ben Jones	Creating a radom password generator in LAZARUS			28
859 98	Michael van Canneyt	GIT continued: Contributing			50
860 98	Fast Reports	Fast Reports: Uploading objects to FastReport using the http and https protocols			66
861 98	Detlef Overbeek	Recover Icons on your Windows desktop			75
862 98	Embarcadero	RAD Studio 11			76
863 99					

If you type in the Name of the author the articles written by him will become listed. Note that the grid is as large as the number of his article.

							L.,
ID	sueNr	Author	Article	PDF	PageNr		^
F	77	Mattias Gaertner	PAS2JS		5	8	
/12	81_82	Mattias Vara Page	Speeding up the boot time of a Toro appliance		8	9	
742	85	Detlef Overbeek / Mattias Gartner	Creating athread to handle The Activity Indicator		3	2	
786	90	Mattias Gaertner	The new Big Sur for the Mac		6	2	
799	91	Mattias Gärtner	Pas2JS install for Lazarus: How to use it		5	4	
812	92	Mattias Gaertner	Web Service Part 1		6	10	
732	94_95	Detlef Overbeek / Mattias Gärtner	Latest Version of Free TMS Webcore for macOS/Linux/Windows		6	6	
84.	96	Detlef Overbeek & Mattias Gärtner	The Chamelon Game for the Web, written in Pascal,		6	2	
855	97	Detlef Overbeek & Mattias Gaertner	New components for Lazarus		6	i0	
857	98	Mattias Gaertner	How to alter your Debugger from GNU to FPC in LAZARUS		2	1	
							*

Here you see all of the mentioned articles: you can either select one and move to that one or select the issue and select a page.

ILAISI	IBraryFo	rm				~
Sh	ow articl	e Go to issue Nr Issue Quantum	< <first <="" next="" prior=""> Last >> Add Delete</first>	Edit Save	Cancel Refresh	CSV
ID	lssueNr	Author	Anticle	PDF	PageNr	
464	42	Detlef Overbeek	Quantum Computing			5
583	62	Detlef Overbeek	Quantum computing			9
589	63	Detlef Overbeek	Quantum computing Basic insights			10
591	63	Detlef Overbeek	Quantum computing Part 2. Building a quantum computer			10
615	66	Detlef Overbeek	Majorana, the new solution for QuantumBits?			6
638	71	Detlef Overbeek	Optical computing /lightning fast the new and fastest computing engine besides			45
653	73_74	Professor Stephanie Wehner	Quantum Internet: Professor Stephanie Wehner			30
805	92	Detlef Overbeek	Quantum Computing			16
843	96	Detlef Overbeek	Quantum Internet already active.			27

Write a word or a sentence of the subject you are interested in and then press enter The result will be a list of articles with your word of interest.

TRABE	LIBraryFo	rm		N				-	
Sh	iow articl	e Go to issue Nr Issue	Quantum	< <first <="" next="" prior=""> Last >></first>	Add Delete	Edit	Save Cancel	Refresh	CSV
ID	IssueNr	Author			- \	PDF	PageN	lr	
464	42	Detlef Overbeek	Quantum Con	nputing					5
583	62	Detlef Overbeek	Quantum com	nputing					9
589	63	Detlef Overbeek	Quantum com	nputing Basic insights					10
591	63	Detlef Overbeek	Quantum com	puting Part 2. Building a quantum computer					10
615	66	Detlef Overbeek	Majorana, the	new solution for QuantumBits?					б
638	71	Detlef Overbeek	Optical compu	iting /lightning fast the new and fastest comput	ting engine besides				45
653	73_74	Professor Stephanie Wehner	Quantum Inter	rnet: Professor Stephanie Wehner					30
805	92	Detlef Overbeek	Quantum Con	nputing					16
843	96	Detlef Overbeek	Quantum	met already active,					27

Click on the line and select "Show article" to let this issue and article pop up.



THE LIB APP PAGE 8 / 11

CREATING A LIBRARY PDF VIEWER INTEGRATED WITH A DATABASE

BLAISE PASCAL MAGAZINE	

🖀 LIBraryForm – 🗆 🗙							
Show article Go to issue Nr Issue Search		< <first <="" next="" prior=""> Last >></first>	Add Delete Edit Save	Cancel Refresh CSV			
ID 1 IssueNr ₂ Author	3 Article		4 PDF	6 PageNr 7			
	1770						
DataSource1 khmMemTable1 khmMemSQL1	BIN khmBinaryStreamFormat1						
(<u>~~~~~~</u> (<u>~~~~~~~~</u>				E			
				Quick config[2.5.1.8] Add all DB fields			
				Remove all DB fields Remove all columns			
1 		Π					
	Jump to page	Q					
Selectable/Draggable area (only active at designtime)							
ImageList1							
	• • • •						
•	•	•	-	•			
This is the basic form							

- 1 DataSource standard on the Data Acces group of the palette in Delphi
- 2 kbmMemTable
- 3 kbmMemSQLTable
- 4 kbmBinaryStreamFormat

You can get these kbmMemTable components for free if you download the kbmMW Community Edition v. 5.16.00 for Delphi 10.4.2 Sydney:

https://portal.components4developers.com.

the other components:

- 5. PDFViewer and
- 6. DBAdvGrid can be downloaded from TMS Software as a Trial TMS VCL UI Pack and the **TMSFNCWXPDF Pack**

https://www.tmssoftware.com/site/tmsfncuipack.asp#product-downloads https://www.tmssoftware.com/site/tmsvcluipack.asp#product-downloads

The Colour Button can be downloaded from you subscription page: https://www.blaisepascalmagazine.eu/your-downloads/



procedure TLIBraryForm.FormCreate(Sender: TObject);

var
Cols: TDBGridColumns;
begin

kbmMemTable1.Open; kbmMemTable1.Active;

kbmMemTable1.LoadFromFile(ExtractFilePath(Application.exename)+'LibraryBPM_Binary.detlef');

ReportMemoryLeaksOnShutdown:=true;

kbmMemSQL1.Parser.FormatSettings.DateSeparator:='/'; kbmMemSQL1.Parser.FormatSettings.ShortDateFormat:='yyyy/mm/dd'; kbmMemSQL1.Tables.Add('articles',kbmMemTable1);

kbmMemSQL1.ExecSQL(SQLStd);

PDFViewer1.NextPage;
end;

```
procedure TLIBraryForm.EditSearchKeyDown(Sender: TObject; var Key: Word;
 Shift: TShiftState);
var
 SearchText, Query: string;
 i: integer;
begin
 if Key=VK RETURN then
 begin
  SearchText:=Trim(EditSearch.Text);
  for i:=length(SearchText) downto 1 do
  begin
   if SearchText[i] in ["",""] then
    SearchText[i]:='_;
  end:
  if SearchText=" then
   Query:=SQLStd
  else
   Query:=SQLStd+' where Author like "%'+SearchText+'%" or Article like "%'+SearchText+'%";
  kbmMemSQL1.ExecSQL(Query);
 end;
end;
```

BLAISE PASCAL MAGAZINE



THE LIB APP PAGE 10 / 11 THE CODE OVERVIEW



procedure TLIBraryForm.ClrBtnIssueNrClick(Sender: TObject); var IssueNo, Row, MovedRows, ArticleCnt, d: Integer; Field: TField; FindIssueNr, CurIssueNr: string; ds: TDataSet; begin IssueNo:=StrToIntDef(EditIssueNr.Text,-1); if IssueNo<1 then begin MessageDlg('Invalid issue number',TMsgDlgType.mtError,[mbOk],0); exit; end; FindIssueNr:=IntToStr(IssueNo); ds:=kbmMemSOL1; ds.DisableControls; try ds.First; Row:=0; ArticleCnt:=0; while not ds.Eof do begin Field:=ds.FieldByName('IssueNr'); CurIssueNr:=Field.AsString; if IsIssueNr(CurIssueNr,FindIssueNr) then begin inc(ArticleCnt); //Field:=ds.FieldByName('Author'); //s:=s+', '+IntToStr(ArticleCnt)+':'+Field.AsString; end else begin if ArticleCnt>0 then break; end; inc(Row); ds.Next; end; finally ds.EnableControls; end; if ArticleCnt>0 then begin MovedRows:=ds.MoveBy(100); ds.MoveBy(-MovedRows-ArticleCnt); //ShowMessage('AAA1 '+IntToStr(ArticleCnt)+' '+s); d:=(ArticleCnt+2)*DBAdvGrid1.DefaultRowHeight+PanelButtonsDBGrid.Height+5; PanelTopDBGrid.Height:=d; OpenSelectedPDF;

end; end;

Blaise Pascal Magazine 99/100 2021

THE LIB APP PAGE 11 / 11

THE CODE OVERVIEW

procedure TLIBraryForm.OpenSelectedPDF; var Field: Tfield; ID, PageNo: Integer; Issue, PDFFilename: string; Row: integer; ds: TDataSet; begin if DBAdvGrid1.SelectedRowCount=0 then exit;

ds:=kbmMemSQL1; Field:=ds.Fields[0]; if Field=nil then exit;

ID:=Field.AsInteger; Field:=ds.Fields[1]; Issue:=Field.AsString; Field:=ds.Fields[6]; PageNo:=Field.AsInteger;

if PageNo<1 then begin
MessageDlg('Missing page information',TMsgDlgType.mtWarning,[mbOk],0); exit;
end;</pre>

// open pdf
PDFFilename:=GetIssuePDFFilename(Issue);
if not FileExists(PDFFilename) then
begin MessageDlg('PDF not found:'#13 +PDFFilename,TMsgDlgType.mtError,[mbOk],0); exit;
end;

LoadPDFFile(PDFFilename,PageNo);
end;

procedure TLIBraryForm.ClrBtnIssueNrClick(Sender: TObject);

var

IssueNo, Row, MovedRows, ArticleCnt, d: Integer; Field: TField;

FindIssueNr, CurIssueNr: string; ds: TDataSet;

begin

end;

IssueNo:=StrToIntDef(EditIssueNr.Text,-1);
if IssueNo<1 then begin MessageDlg('Invalid issue number',TMsgDlgType.mtError,[mbOk],0); exit;
end;</pre>

FindIssueNr:=IntToStr(IssueNo);

```
ds:=kbmMemSQL1;
```

```
ds.DisableControls;
try
 ds.First;
 Row:=0;
 ArticleCnt:=0;
 while not ds.Eof do begin
  Field:=ds.FieldByName('IssueNr');
   CurIssueNr:=Field.AsString;
  if IsIssueNr(CurIssueNr,FindIssueNr) then
    begin inc(ArticleCnt);
  endelse begin if ArticleCnt>0 then break;
  end;
  inc(Row);
  ds.Next;
 end;
finally ds.EnableControls;
end;
if ArticleCnt>0 then
begin
 MovedRows:=ds.MoveBy(100);
 ds.MoveBy(-MovedRows-ArticleCnt);
  d:=(ArticleCnt+2)*DBAdvGrid1.DefaultRowHeight+PanelButtonsDBGrid.Height+5;
 PanelTopDBGrid.Height:=d;
 OpenSelectedPDF;
end;
```









50% Discount on initial licenses for the Enterprise Edition

valid until 30 November 2021 Cupon code is: BPEKON21





SCALING TO A VERY LARGE AI CAUSES UNPRECEDENTED INSIGHTS

OpenAl GPT-3

When the second-to-last version of the artificial intelligence **GPT-3** was released last year, there was an impression that it could master human language quite well and generate fluent texts on command.

This article was

partually rewritten after readings in the New Scientist. For those who would like the original please visit that website.

PAGE 1/6

https://www.newscientist.com/issue/3355/

While the world was still asleep, critics pointed to the architecture's many flaws and simplistic structure. Just mindless machine, they thought. But there are reasons to believe that AIs like **GPT-3** could very soon develop language skills, reasoning, and other features of what we consider intelligence to be at the human level.

GPT-3's success is due to one thing: it was bigger than any AI of its kind, which roughly means it had many more artificial neurons.

Nobody expected that this increase in scale would make such a difference. But as **AIs** grow in size, they're proving to be the equal of humans in all sorts of tasks, but also demonstrating the ability to take on challenges they've never had before.



Generative Pre-trained Transformer 3(GPT-3) is an autoregressive language model that uses deep learning to produce human-like text. It is the third-generation language prediction model in the **GPT-n** series

WIKIPEDIA (and the successor to GPT-2) created by OpenAI, a San Francisco-based artificial intelligence research laboratory. PT-3's full version has a capacity of 175 billion machine learning parameters. GPT-3, which was introduced in May 2020, and was in beta testing as of July 2020, part of a trend in natural language processing NLP) systems of pre-trained language representations.

The quality of the text generated by **GPT-3** is so high that it can be difficult to determine whether or not it was written by a human, which has both benefits and risks. Thirty-one **OpenAI** researchers and engineers presented the original May 28, 2020 paper introducing **GPT-3**. In their paper, they warned of **GPT-3's** potential dangers and called for research to mitigate risk. **David Chalmers**, an Australian philosopher, described **GPT-3** as "one of the most interesting and important AI systems ever produced". Microsoft announced on September 22, 2020 that it had licensed "exclusive" use of **GPT-3**; others can still use the public API to receive output, but only Microsoft has access to **GPT-3's** underlying model.)



SCALING TO A VERY LARGE AI CAUSES UNPRECEDENTED INSIGHTS



As a result, some are beginning to think that the inexorable push for larger scale will lead to AIs with capabilities comparable to those of humans. By significantly scaling up current methods, after a period of 20 years of computer improvements, it seems likely that human-level language literacy and semantics may be achievable.

Neanin ma n t i c s Meanir Senterce Dr. Nimer Abusalim

Senantics mann tics

(Semantics (from Ancient Greek: semantikós, "significant" is the study of meaning, reference, or truth. The term can be used to refer to subfields of several distinct disciplines, including philosophy, linguistics and computer science.)

> That would be incredible if true. Few experts thought that machine intelligence would advance as a mere exercise in semantics. Of course, many still doubt that that will actually happen. Time will tell. For a long time it was unclear whether computers could do such a thing,

Unlike ordinary software, researchers do not give instructions to neural networks. Instead, they are designed to be trained in a task until they learn to do it properly.

Given a large array of animal images, for example with a human annotation for each image, such as "dog" or "cat," a neural net can be trained to predict the correct tag for an image it has not seen before. Every time it's mislabelled, there's a systematic way to tell it so that, if there are enough examples, the network improves itself by recognizing the animals.



PAGE 3/6

SCALING TO A VERY LARGE AI CAUSES UNPRECEDENTED INSIGHTS



Assistant Professor of Linguistics, Data Science & Computer Science Co-PI, ML² Group & CILVR Lab New York University



PhD 2016, Stanford NLP Group & Stanford Linguistics

https://cims.nyu.edu/~sbowman/

But these

neural networks, also called "models," are not limited to identifying cats and dogs. In 1990, then at the University of California, San Diego, Jeffrey Elman devised a way to train a neural network to process language. He discovered that he could delete a word from a sentence and train a network to predict the missing word.

Elman's model couldn't do much more than tell the difference between nouns and verbs. What made it so interesting was that it didn't require meticulous human notes. He could create training data by simply removing random words.

Eventually, researchers realized that it was easy to retrain a model to address more specific problems. These include language translation: answering questions and sentiment analysis, where models measure whether, for example, a human movie review is positive or negative.

By the time Bowman completed his PhD in 2016, language models had mastered many of the more routine tasks.

No one claimed that these models had anything remotely resembling intelligence: Analyzing sentiment can be as simple as choosing words like "great" or "I loved it" from a review.

But language models also got better at more difficult tasks, almost as fast as you could imagine them.

"Language AIs show that scaling alone can create surprising new skills"

The trick was to train

models on more and more data - and to handle huge chunks of text from the Internet and other sources, the models had to be bigger. The field of AI was also building neural networks in new ways, creating new arrangements of neurons with different connections.

In 2017, Google researchers created a neural architecture called the "transformer" that proved to be particularly scalable. In pursuit of everimproving performance, researchers have upgraded transformer-based models from hundreds of millions of parameters, each roughly analogous to a connection between neurons, to hundreds of billions — in just a few years.



SCALING TO A VERY LARGE AI CAUSES UNPRECEDENTED INSIGHTS



ARGUMENTATION

This way of thinking seems to be correct: It paid of. The scaled inverter model has shown things that exceeded all expectations. By the end of 2020, an inverter-derived architecture called BERT had overcome some very difficult challenges.

BERT is an open source machine

learning framework for Natural Language Processing NLP.
 BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. The BERT framework was pretrained using text from Wikipedia and can be fine-tuned with question and answer datasets.
 BERT, which stands for Bidirectional Encoder Representations from

Transformers, is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. (In **NLP**, this process is called attention.

One of these

was general reading comprehension.

Another ability tested involved common sense reasoning.

The models were asked to analyze sentences such as "The trunk does not fit in the trunk of a car because it is too large" and to determine whether "it" refers to the trunk or the trunk. The correct answer is, of course, the suitcase. Solving this task requires a certain depth. The models solved it on a human level, meaning they literally performed as well as humans.

In recent years, progress has been dazzlingly rapid. And while architectural innovations like the transformer have been significant, most of these advances can be attributed to scale. The very clear trend is that most of the tests we can think of will be solved once you add more scale to it.

Nowhere is this relationship between scale and smartness more apparent than in the case of GPT-3, which arrived in May 2020. With 175 billion parameters, GPT-3 was just an upscaled version of GPT-2 released in February 2019 with 1.5 billion parameters.

Still, it showed GPT-2 a huge leap in its language skills, from struggling to write coherent paragraphs to producing 2,000-word essays that could pass for human level.

Admittedly, it is still inconvenient to discover large language models. If you ask GPT-3 how many eyes a foot has, it could tell you two. And there are still plenty of capabilities that models like GPT-3 don't have, such as understanding cause and effect: for example, figuring out which of the phrases "it started raining" and "the driver turned on the windshield wiper" should come first.





Still, an analysis of the progress already made suggests that such shortcomings will not be insurmountable. Indeed, in 2020, OpenAI researchers discovered that economies of scale are predictable. They follow a clear law: For every increase in the size of a GPT-style model, it can predict a missing word a little better, which translates into improved performance on all sorts of language tasks. This tendency has been demonstrated for models whose neural networks range from the size of a roundworm's brain to that of a rabbit.

In addition, new possibilities can appear out of nowhere. For example, scaled-down versions of GPT-3 showed little numeracy - unsurprisingly, considering they're only trained to predict the next word.

But math skills somehow appeared in the full version. "Only scale can unlock surprising new possibilities".

A recent workshop predicted that at the rate at which models have improved on various language tasks, all such tasks could be solved when models reach the investment level of the Large Hadron Collider, the multinational physics experiment near Geneva in Switzerland - \$ 10 billion to \$100 billion, a large but far from unmanageable amount.

Solving every conceivable language task would not necessarily mean that a model was intelligent. Language behavior is only a subset of what people do. Nevertheless, recreating these skills in a machine would be a huge task as it seems to be a giant step towards achieving an artificial general intelligence. An AI that can do everything humans can do, including self-improvement.

People who develop these models have gone to great lengths to show that scaling up solves these problems.



SCALING TO A VERY LARGE AI CAUSES UNPRECEDENTED INSIGHTS

PAGE 6/6

Of course,

not everyone is convinced of the

power of scaling up. To some extent, models can be

thought of as memory machines, he says. As they get bigger, they remember more. But for a model to remember everything, or even as much as Google Search, for example, it would have to be unimaginably large. "That's probably way outside the limits of what we can train at the moment," says Raffel.

"We need better ways to assess AIs and how they relate to humans"

The counter-argument is that if scaled-up models can indeed reason human-like, they don't need to remember everything — after all, humans don't. At this point, there is no doubt that human-level reasoning goes beyond current language AIs. The question is, will scaling up make this happen?

For others, there are now reasons to disagree. It has often been argued that when models succeed in developing a process of reasoning, they can only do so because they have memorized patterns from countless examples.

The ability to learn new things from scratch is one of the many signals that models can reason the way humans do.

It will take time to see how far scaling artificial intelligence can take. Many people believe in the need for different approaches to make further progress.

But whichever way we get to artificial general intelligence, if it is indeed a realistic goal, what's already clear from scaling language models is that we need more sophisticated ways to assess the intelligence of AIs — and how they behave. relate to ours.

We've only just started developing the tools needed to see if what language models do really resemble human capabilities. But recent work has already yielded some intriguing results.

The scaling itself is now changing. Researchers recently figured out how to come up with models that can train themselves on images and videos, as well as on words. This allows them to learn from much more data of a much richer nature, more like humans do. Soon Google will show results of a model with a trillion parameters, the largest ever. Who knows what new revelations that will reveal?



This article was partually rewritten after readings in the New Scientist. For those who would like the original please visit that website.

https://www.newscientist.com/issue/3355/



kbmMW Community Edition v. 5.16.00 released for Delphi 10.4.2 Sydney!!

Components4Developers has decided to let

the world have the opportunity to play with the hundreds of thousands of lines of code, containing thousands of functions, procedures, methods, classes and types, covering the vast number of features that makes kbmMW the leader of n-tier development environments.

We have now released our **second Community Edition** which contains almost all features from kbmMW Enterprise Edition, but limited in some areas, partly due to technical limitations, partly due to a few artificial limitations. It further contains kbmMemTable Community Edition which is a direct match of kbmMemTable Standard Edition.



Community Edition is FREE to use as long as the applications it is used in provides a direct or indirect revenue not reaching US\$5000 and the company distributing/producing the applications have a yearly turnover of less than US\$50000.

Read the license.txt file for all details.

Community Edition can only be used for producing 32 bit Windows applications, but you have access to SmartServices, REST, WIB, kbmMemTable, SQL, ORM, SmartEvent, SmartBinding, Scheduler, Logging, Configuration, Ciphers, transports, XML, JSON, YAML, BSON, MessagePack, i18n and the countless of other features you have read about here on the blog!

Community Edition can thus also be used as a 60 day trial for kbmMW Enterprise Edition. Community Edition does not contain source code, and will only work together with the version of Delphi it was released for.

Current release match latest Delphi Sydney 10.4.2, including Community Edition.

If you want go further, then purchase kbmMW Enterprise Edition which really is the best value for money and while being the oldest existing, mostly backwards compatible, n-tier framework for Delphi, it is also the only one bringing you to the bleeding edge of current technology.

Sign up at https://portal.components4developers.com and download immediately.

_--

kbmMW is the premiere **n-tier product** for Delphi, C++Builder and FPC on Win32, Win64, Linux, Java, PHP, Android, IOS, .Net, embedded devices, websites, mainframes and more. Please visit http://www.components4developers.com for more information about kbmMW.

Components4Developers is a company established in 1999 with the purpose of providing high quality development tools for developers and enterprises. The primary focus is on SOA, EAI and systems integration via our flagship product kbmMW.

kbmMW is currently used as the backbone in hundreds of central systems, in hospitals, courts, private, industries, offshore industry, finance, telecom, governments, schools, laboratories, rentals, culture institutions, FDA approved medical devices, military and more.





50% Discount on initial licenses for **kbmMW Enterprise Edition**

valid until 30 November 2021 See page 168 inside the magazine

KBMMW PROFESSIONAL AND ENTERPRISE EDITION V. 5.18.00 RELEASED!

RAD Studio XE5 to 11 Alexandria supported

- Win32, Win64, Linux64, Android, IOS 32, IOS 64 and OSX client and server support
- Native high performance 100% developer defined application server
- Full support for centralized and distributed load balancing and failover
- Advanced ORM/OPF support including support of existing databases
- Advanced logging support
- Advanced configuration framework
- Advanced scheduling support for easy access to multithread programming
- Advanced smart service and clients for very easy publication of functionality
- High quality random functions.
- High quality pronouncable password generators.
- High performance LZ4 and Jpeg compression
- Complete object notation framework including full
- support for YAML, BSON, Messagepack, JSON and XML Advanced object and value marshalling to and from
- YAML, BSON, Messagepack, JSON and XML High performance native TCP transport support
- High performance HTTPSys transport for Windows.
- CORS support in REST/HTML services. Native PHP, Java, OCX, ANSI C, C#, Apache Flex client support!

kbmMemTable is the fastest and most feature rich in memory table for Embarcadero products.

- Easily supports large datasets with millions of records
- Easy data streaming support
- Optional to use native SQL engine Supports nested transactions and undo

COMPONENTS

DEVELOPERS

- Native and fast build in M/D, aggregation/grouping, range selection features
- Advanced indexing features for extreme performance

- New I18N context sensitive internationalization framework to • make your applications multilingual. New ORM LINQ support for Delete and Update. Comments support in YAML. New StreamSec TLS v4 support (by StreamSec) Many other feature improvements and fixes.

Please visit

http://www.components4developers.com for more information about kbmMW

- High speed, unified database access (35+ supported database APIs) with connection pooling, metadata and data caching on all tiers
- Multi head access to the application server, via REST/AJAX, native binary, Publish/Subscribe, SOAP, XML, RTMP from web browsers, embedded devices, linked application servers, PCs, mobile devices, Java systems and many more clients
- Complete support for hosting FastCGI based applications (PHP/Ruby/Perl/Python typically) Native complete AMQP 0.91 support (Advanced Message
- Queuing Protocol)
- Complete end 2 end secure brandable Remote Desktop with near realtime HD video, 8 monitor support, texture detection, compression and clipboard sharing.
- Bundling kbmMemTable Professional which is the fastest and most feature rich in memory table for Embarcadero products.

