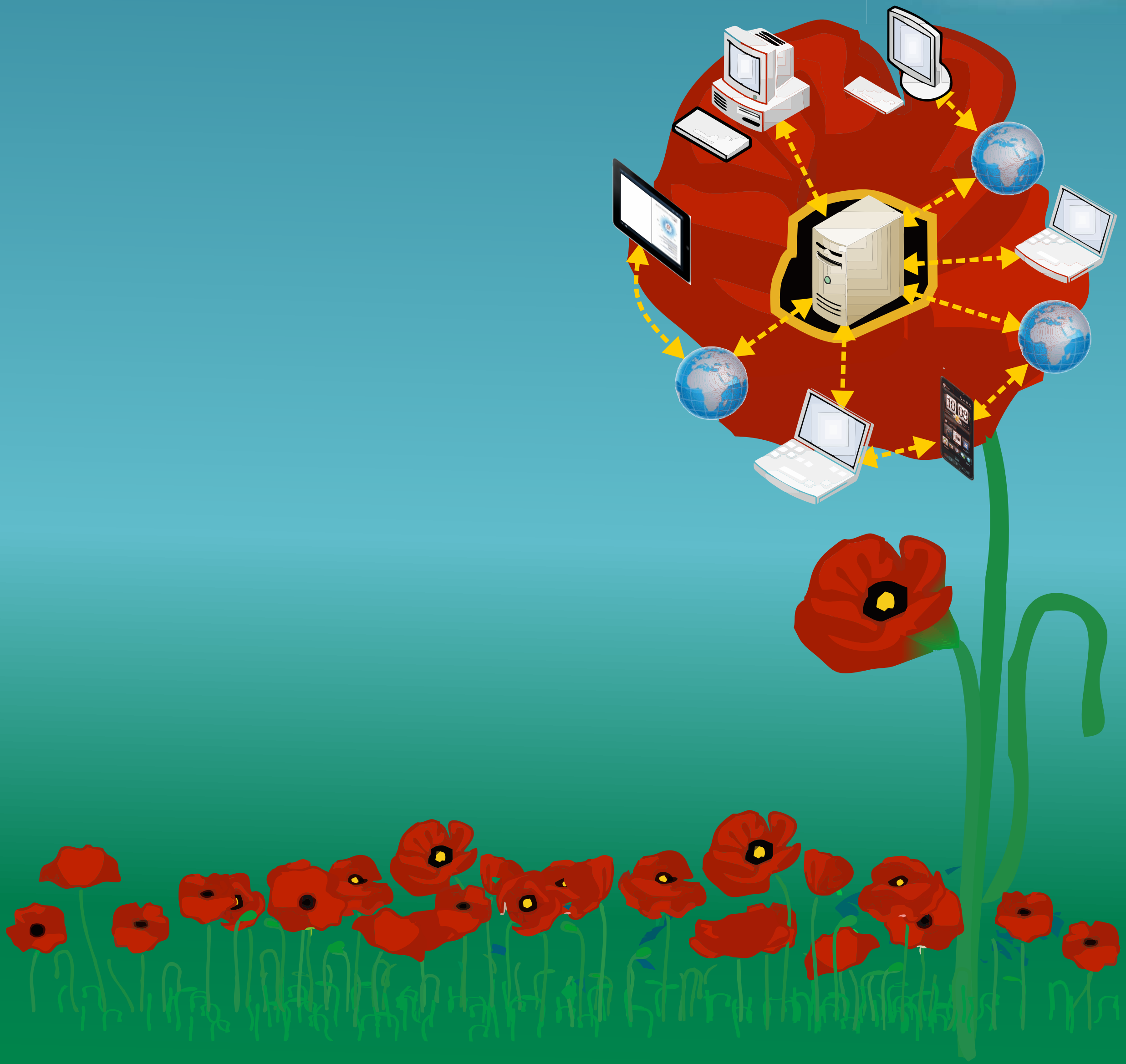
BLAISE PASCAL MAGAZINE 78/79

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

Blaise Pascal



Seven-Exit Windows 7 support runs out 2020
Installing TMSWEBCore for DELPHI
Win 10: Protecting your Privacy
Windows Defender: do you need another Anti Virus Program?
A powerful new Triumvirate for Web Development
Installing TMSWEBCore for Lazarus
Lazarus Object Pascal on RaspBerry
Microservices and Progressive Web Apps in Delphi
Puzzle: Crisper Cas - what is it and software for a JigSawPuzzle
REST easy with kbmMW #24 – XML, JSON or YAML to object conversion

BLAISE PASCAL MAGAZINE 78/79

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

Blaise Pascal

CONTENT

ARTICLES

ADVERTISERS

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

Pascal is an imperative and procedural programming language, which Niklaus Wirth designed 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

Colofon

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

Peter Bijlsma -Editor
peter @ blaisepascal.eu

Dmitry Boyarintsev
dmitry.living @ gmail.com

Michaël Van Canneyt,
michael @ freepascal.org

Marco Cantù
www.marcocantu.com
marco.cantu @ gmail.com

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

Benno Evers
b.evers
@ everscustomtechnology.nl

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

Holger Flick
holger@flixments.com

Primož Gabrijelčič
www.primoz @ gabrijelcic.org

Mattias Gärtner
nc-gaertnma@netcologne.de

Peter Johnson
http://delphidabbler.com
delphidabbler@gmail.com

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

John Kuiper
john_kuiper @ kpnmail.nl

Wagner R. Landgraf
wagner @ tmssoftware.com

Vsevolod Leonov
vsevolod.leonov@mail.ru

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

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

Kim Madsen
www.component4developers

Boian Mitov
mitov @ mitov.com

Jeremy North
jeremy.north @ gmail.com

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

Howard Page Clark
hdpc @ talktalk.net

Heiko Rompel
info@rompelsoft.de

Wim Van Ingen Schenau -Editor
wisone @ xs4all.nl

Peter van der Sman
sman @ prisman.nl

Rik Smit
rik @ blaisepascal.eu
www.romplesoft.de

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

B.J. Rao
contact@intricad.com

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

Anton Vogelaar
ajv @ vogelaar-electronics.com

Robert Welland
support@objectpascal.org

Siegfried Zuhr
siegfried @ zuhr.nl

Editor - in - chief
Detlef D. Overbeek, Netherlands Tel.: +31 (0)30 890.66.44 / Mobile: +31 (0)6 21.23.62.68
News and Press Releases email only to editor@blaisepascal.eu

Editors
Peter Bijlsma, W. (Wim) van Ingen Schenau, Rik Smit

Correctors
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	€ 250	€ 261,60	€ 85,00
Electronic Download Issue 60 pages	€ 60	€ 65,40	———
Printed Issue inside Holland (Netherlands) ±60 pages	———	€ 200	€ 40,00

Member and donator of WIKIPEDIA

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

Readers write

In this section we publish written opinions, articles, critics or or ideas of our readers.

Alexander Liberov.

Thanks for collecting this list. Maybe later you could publish what features are most often requested.

First thing first: All of our company Delphi development is done for Windows Desktop 32 and mostly 64 bit applications. I suspect that the vast majority of developers target Windows desktop.
Web, Android, IoT, Apple, etc.. Are nice toys, but Windows desktop is what pays us money.

Not necessarily in order of importance:

1. I went through 3 Delphi version upgrades (10.1 to 10.3 and before that we were - and occasionally still- using Delphi 5).
The process can be significantly improved.

1.1. Though Delphi has an option to transfer settings, **in reality it doesn't work.**
Many options like say source formatting do not transfer.

1.2. Every serious programmer uses a sizable list of 3rd party tools and components paid and free.
There should be a tool that will catalog all add-ons not just Getit but all.
Maybe it will require additional API to register add on tools.
There should be a way to keep web address, update download address, user name and password, maybe even original purchase documents, git or other version download location where applicable.
Maybe some version control tool to notify of updates and download these updates.
I understand Embarcadero does not want to reinvent wheels and compete with version control clients. There should be some interface to make it easier to download updates and keep local source code for remote repositories, like say JCL or Spring4D. There should be a tool significantly simplifying Delphi upgrades.

2. **I think Database handling is not yet well RAD-ified. I suggest the following improvements:**

2.1. There should be a repository of all data fields that keeps at least the following information:

2.1.1. Caption

2.1.2. Output and editing formats

2.1.3. Flags for Visible / Editable / Customizable

2.1.4. Event code snippets:
Data tables get linked to forms and components for example 1 table might be dropped/linked to multiple forms, grids, etc… These links should be saved for 2 way editing.
Say if I drop a table to a form I'd like to automatically create calculated fields with their calculations. The snippets should be as flexible as possible maybe in a setup section it would be possible to select any 2 components and what events get created when they are linked together.
We often have same 10 tables in similar grids on 100s of forms with the same or similar calculated fields and similar grid cell painting rules, GetText – SetText procedures, etc.
It would be nice to have to write these procedures once and then just checkbox if I need any of them in this particular place. If at some later point we need to add or delete a field to a table Delphi would automatically update all places with necessary snippets. Same with updates.

2.1.5. **Maybe the above repository snippets would work not only on data tables, but also on objects.**
Say if an object has a public field or a property with a [FIELD] attribute it also might be fitted with code snippets.

2.1.6. **This repository should be accessible through some automatable way with common DB admin tools and maybe Delphi code.**
Parts of this repository could be exported into the result project database to enable on-the-fly customization, translation, etc.
For example there might be an optional step after opening a table to load aptions, formats and other attributes from the database.

2.2. Poor-man or Fool's ORM: 2-way tool to create a unit that would create a component from a dataset with matching Interface (tblCustomer, TCustomer, ICustomer) with read/write (get/set) procedures, Load / Save to a database procedures.

3. **Tools to help separate business logic and data visualization.** For example

3.1. 2-way tool to scan all visual components on form and create a record or an object that can be passed from logic to visualization and back. This object would contain selectable properties, for example captions, colors, alignments, etc. This might be based on p. 2.1.4 above maybe. If say I add a TLabel or TEdit or TxyzLabel to a form Delphi automatically adds lines to procedures that collect form data and that refresh draw this form as well as a field/property in the record/object for each changeable property.

4. **Catalog of 3rd party tools and components.**
   We got stuck with Delphi 5 for a long time and now coming to a new marketplace.

Maybe your magazine could help better in this regard than Embarcadero.
Why not to maintain a catalog broken into subject sections. There should be some limited number of 'well-known' reviewers who would write reviews maybe even get paid for it by commercial vendors – sure with a disclosure.

If I'd not buy a book I'd never know about Spring4D and I stumbled only recently into MMX.
I understand that such a list might be easily overwhelmed by myriads of small vendors free and commercial.
To prevent this there will be a very limited reviewers board that will write reviews.
Tools that are not yet reviewed will reside in secondary lists. All tools would also have an individual message board so anybody also could write their opinion on it, but those boards would be secondary.

Maybe one more thing to add to the list is I often find myself programming data access in threads. It would be very beneficial to have a tool that would create all necessary code to create and initialize Database, Transaction and dataset components.

GExpert has a useful similar generic feature, but it doesn't treat dataset components correctly, so some additional manual work is still required.

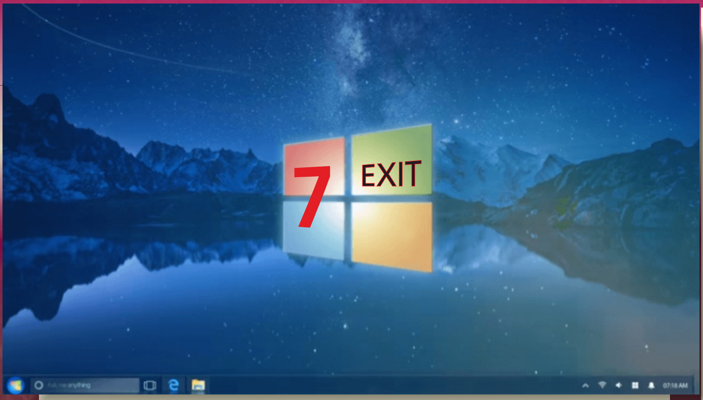
I think right now there isn't much value for developers between pro version and enterprise and architect. Besides none of Architect odd-ons over Enterprise support Interbase and Firebird. Adding more database – related RAD tools might be very beneficial.

This is what comes to my mind right now.

Sincerely,

Alexander Liberov.

By Detlef overbeek, translation Howard Page Clark



INTRODUCTION

Windows 7 is still widely used although official support for it ends early next year, which means you must prepare for its demise if you have not already done so. Microsoft's deadline is clear:
support for Windows 7 is ending on January 14, 2020.
There will be **no extension** to this date.

Is Microsoft trying to scare us?
What does "end of support" mean in practice, and what do current Windows 7 users need to be doing to be ready for this watershed?
This article attempts to answer these important questions, and offer you some solutions.

Many users are conservative, hating unnecessary change, hating the forced upgrade.
Because moving beyond Windows 7 means having to learn about a substantially changed operating system, and a possibly painful familiarisation process when presented with a changed user interface. There was a lot of resistance in 2014 when Microsoft ended support for **Windows XP**.

After all, why change something that is not broken, that works well and has become so familiar? Well... perhaps because of the latest developments.

At the time of the planned end to support for Windows XP, Windows 7 was already generally accepted as a worthy successor to XP, incompatibilities a demonstrable improvement over the system it replaced. Nevertheless, many users were unable to say goodbye to Windows XP. Some were unable to between old and new.
Others were just too attached to XP to give it up.

By contrast, the end of the support for Windows Vista was almost a non-event, since it was used by comparatively few, and had almost no fans who had become attached to it. Indeed it was widely regarded even by Microsoft junkies as a failed experiment.

There is certain to be widespread dissatisfaction in January 2020 when Microsoft ceases support for Windows 7, and it is likely that the resistance will be even greater than in 2014.

Windows 7's market share in 2020 will considerably exceed that of Windows XP in 2014.

Moreover, Windows 10 is by no means accepted by everyone. There are also quite a few Windows 7 users who have completely ignored or misunderstood the offer to get a Windows 10 upgrade for free.

These people will continue to use Windows 7 until they really can't do anything else. They may ignore all warnings and simply keep going with the old Windows 7 they know and love, which will continue to work. However, it will receive no further updates for any security issues which arise after January 2020.

What are your options? Either switch to Windows 10, or hope that your Windows 7 PC will not be attacked by viruses which appear in the wild post-January 2020? Let's examine what the end of support actually means and what its consequences are. Then we'll look at what the transition to a newer Windows version involves.

LAST EVER UPDATE

The end of support for Windows 7 has one main consequence: on January 14, 2020, you will get free security updates for Windows 7 for the last time, then never again.

Security holes discovered subsequently will no longer be closed. Discovering security flaws takes time. Approximately 1000 vulnerabilities in Windows 7 have been identified and included in the central database 'CVE Detail' since the first release of Windows 7. Of that number, 229 were not found until 2017, and 161 were not identified until 2018.

Among the 1,000 or so vulnerabilities, 269 are classified as "very dangerous security breaches", through which it is possible for malicious agents to retrieve and execute arbitrary code.
Of these 269 most dangerous vulnerabilities, 47 were not discovered until 2017 and 35 not until 2018.
This year, 2019, has seen a further 11 already identified.

Even worse is that a lot of code from Windows 7 is also included in its successors (Windows 8.1 and 10), since subsequent versions extend existing system code, rather than being written from scratch.
So insecurities discovered in the latest Windows version often indicates that a similar insecurity exists in earlier versions. Microsoft normally publishes more or less detailed information about newly discovered gaps once they have been closed. That gives attackers leads to find corresponding holes in Windows 7 quite quickly.

However, Microsoft will not release the source code of Windows 7 even after the support has been terminated, in order that no attacker can search for as-yet-undiscovered holes and back doors.

It is also true that the overall number of Windows 7 users will steadily decrease from January 2020 onwards, making attacks on Windows 7 less and less attractive for the Bad Guys.

On the other hand you can argue that it is precisely the less computer-savvy people who may continue to work on Windows 7 and that the more naive users are an easier target for attackers.
Extra security software cannot offer complete protection against future attacks: a virus scanner and firewall and the like can only limit the risk.
One of the characteristics of a security hole is that by its very nature you can exploit it to bypass security mechanisms.

DETAILS OF AN EXPLOIT

The English verb exploit means "to use something to one's own advantage". In computer jargon an exploit is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a system bug or vulnerability to cause unintended or unanticipated behaviour. Exploits affect computer software, hardware, or electronic devices such as satnavs that rely on software for their operation.

Malign exploits allow unauthorised people to gain control of a computer system. They may allow escalation of restricted privileges, or result in a denial-of-service (DoS or related DDoS) attack.

In Windows, the Server Message Block (SMB) is the transport protocol used for a wide variety of purposes such as file sharing, printer sharing, and access to remote Windows services.
The EternalBlue exploit used a critical leak in the SMB version 1 implementation of Windows XP and TCP port 445 to propagate itself. May 26, 2017

THE EFFECTS OF SUCCESSFUL ATTACKS

Attackers have already proved that there is a real risk from unsolved security holes. In the fairly recent past the EternalBlue XP exploit was used very successfully by WannaCry and Petya on unpatched PCs via the internet without any user interaction.

Large-scale damage ensued, not only to home users (who often had no backups) but also to many commercial and public service organisations. At some car manufacturers the production lines were halted. In Q-Park parking bays the parking machines no longer worked. The malware is said to spread like a worm.
In fact the spread is better described using the analogy of bacterial multiplication rather than a "worm" metaphor.
However you describe it, the attack was deadly and self-propagating. Once a system was infected, it actively started looking for new victims. All Windows systems on the same local network as the attacked computer became infected, unless they had been patched.

The consequences of missing updates can be disastrous: the crypto-trojan WannaCry (what an ironic name) exploited a security leak for infection.
The update dramas of Windows 7 are legendary, but after January 14, 2020, Windows 7 users will receive no more of them.
It is not yet clear how reliable the crypto infrastructure of Windows 7 will be when the support is terminated.

Windows' crypto infrastructure is the foundation for all kinds of processes in the operating system that have to do with encryption. The influence of the crypto API starts with the fact that Windows makes a secure connection with Microsoft's servers.

Applications such as Google Chrome, for example, trust Windows to validate certificates. When a leak is discovered in the crypto API and Microsoft does not close it (the question is not whether, but when that happens), then not only is your computer's operating system fundamentally compromised, but also all your personal data linked to online purchases.

AMBIGUITY

Because there is often a lack of clarity about what the "end of the support" means, we will clarify first what it does not mean.
After the end of support, Windows 7 will continue to run and be fully functional, at least as long as the installation is not stopped by malware or other attacks. Even if you were to succumb to an infection, you could reinstall Windows 7 again. Windows 7 will certainly not suddenly become a free product, so that anyone at all could use it after January 2020 without payment.

Microsoft retains all rights to the software, and nothing will change about the activation of the system if it needs to be reinstalled. After a reinstallation or certain changes in the hardware you still have to activate the software again. Microsoft will not disable the associated servers.

Some aspects of Microsoft support will not change after January 2020. Microsoft has not repaired non-critical bugs in Windows 7 since 2015, and that policy will continue.
You can also use all the help that Microsoft offers for 'self-study' after January 2020, such as the articles in the Knowledge Base at **`https://support.microsoft.com`**, and the forums at **`https://answers.Microsoft.com`**.

But that will not increase the security of your Windows 7 installation. Access to a maintained Knowledge Base doesn't really help with future security.

POSSIBLE POLICY EXCEPTIONS?

Although Microsoft has been saying for years that the January 14, 2020 end-of-support date is immutable, nevertheless two exceptions have been offered for extending Windows 7 support, although this will not benefit the average PC home user.

The first exception: if the update published on January 14 accidentally causes a new leak, it will be closed and a further update issued.
After that, the free updates finally end. The second exception applies to any licensee who is willing to pay for further support from Microsoft.

The conditions are that you must have a volume license agreement with Microsoft (agreements which are normally restricted to governments, companies, and institutions), and that you own an applicable version.
If you have a pre-installed Windows 7, or a DIY PC you fall outside the volume licensing condition.

Furthermore, this extended paid support is limited to a maximum of three years against payment. The exact costs are not known, but it is clear that they apply per device and will be high enough that customers will prefer to switch to Windows 10. In addition, the license fee will increase annually. Customers with a Software Assurance agreement are entitled to a discount.
Microsoft calls this paid support extension 'Extended Security Updates' (ESU).

The ESU is limited to the Windows 7 Professional and Enterprise editions. This means that neither the Home editions nor Windows 7 Ultimate users can receive extended support. Windows 7 Ultimate is completely identical to Enterprise in terms of functionality.

However, Ultimate was always intended for private users, and they don't get any support extension. If you think that the paid-for updates that Microsoft delivers after 2020 will somehow become available to everyone (or that Microsoft could be legally forced to make them available to all) you are mistaken.
All warranty and service obligations expired 10 years after the release of Windows 7 and you cannot sue the company on those grounds.
In any case, you would have to address your claim to the vendor, which for consumers is normally not Microsoft, but the retailer where you bought the PC with pre-installed Windows or the licence.

You will hardly be able to get updates in any unofficial ways either. This was tried before – and did not work. You could persuade Windows XP via a registry hack to download some updates, but they were intended for special banking and POS systems called 'Windows Embedded POSReady 2009' (POS stands for 'Point of Sale'). Those updates were not tested on desktop systems and were intended for a specific application area.

Even where the unofficial updates closed certain security holes, many others remained. In addition, applying untested updates could make your system unstable. If such hacks are shared for Windows 7, you are advised to stay far away from them, because they only offer a false sense of security.

END-OF-SUPPORT SIDE EFFECTS

When support for Windows 7 ends in 2020, support for the corresponding version of Internet Explorer also ends. The browser will only receive updates if it is running under a newer version of Windows, not under Windows 7.

Support for non-Microsoft hardware and software will also be affected, because many manufacturers will take the opportunity no longer to support Windows 7 and shift that to Microsoft.
Microsoft's end-of-support affects more than just the operating system and its browser.
Newer Office versions do not work on Windows 7 and older versions compatible with Windows 7 will likewise no longer receive security updates from Microsoft. Because infected Office files are often used for attacks, this can quickly become fatal.

Support for non-Microsoft hardware and software will also be affected,

Of course there are alternatives such as LibreOffice, which are certainly suitable for home use. But anyone who wants to use both Windows 7 and Microsoft Office after January 2020 has a problem. You may also have to look for alternatives in other areas.

Virus scanners are not a substitute for security updates, but many manufacturers may quickly stop supporting their products under Windows 7.
At least that was the case with Windows XP.
Although virus scanners from, for example, Norton (Symantec) and Avast still cover Windows XP SP3, for other companies such as Avira and Kaspersky XP is too old for them to bother with.

THE END IS NIGH!

The impending end of support for Windows 7 has prompted increasing numbers of non-Microsoft vendors to drop Windows 7 as a supported system.

For example, Adobe has already raised the system requirement to Windows 10 for real-time video editing in Premiere Pro and After Effects and for 3D compositions with Adobe Dimension.
CorelDraw Graphics Suite 2018 now has a minimum of Windows 10 as the required operating system. If you want to watch Netflix on Windows, you get a maximum of full HD with Windows 7. Windows 10 is required for 4K and HDR because the associated copy protection uses Play Ready 3.0 and Edge. DirectX 12 is also not available for Windows 7.

Even installing Windows 7 is becoming increasingly difficult. Not because Windows 7 prevents this itself, but because needed drivers are missing from the installation media (for example for USB 3.x).
Moreover, Win7 installs often lack support for Secure Boot, and can only be started by tweaking the UEFI mode.

It is a fact that many modern PCs no longer have an optical drive for the Windows installation DVD.
And even if the installation succeeds, problems may immediately arise.
Even if there are still Windows 7 drivers for modern hardware, they are increasingly outdated versions that manufacturers do not update to give access to all the latest functionality. For example, while graphic cards or integrated graphics show images correctly, they do not let you access their energy-saving functions.

The inability to apply security updates after January 2020 is, of course, a very important consideration in deciding whether or not to upgrade from Windows 7, because every closed off security breach is one less open door for attackers. But of course reliance on system updates should never be the only security measure you take against the ever-present threat from the Bad Guys who are to be found everywhere.

There is one last Note to be made: If you have created a dual boot system with Win7 and Win10 you will get into trouble in future. You can NOT remove Win7 without getting blue screens and data-loss. Your partition will not even start again.
So if you want to to setup a Windows 10 environment just use a new separate startup partition without win 7 on it. And: an SSD Disk maybe a very good new and fast setup...

Detlef

TMS WEB CORE FOR DELPHI
INSTALLATION GUIDE

INTRODUCTION
installation for **Delphi**

TMSWEBCore is a component framework designed for rapid application development of web clients using Object Pascal following the single-page architecture employed by other modern frameworks like Angular, vue.js, and React. Object Pascal, in turn, is a mathematically grounded and syntactically simple language with a rich heritage underpinning the World's first highly productive, object-oriented IDE (**Delphi/RAD Studio for WindowsOS**) and inspired several cross-platform compiler and dialect implementations (Free Pascal, Oxygene, Pascal Script, DWScript), as well as as well its current-day language siblings, C# and TypeScript.

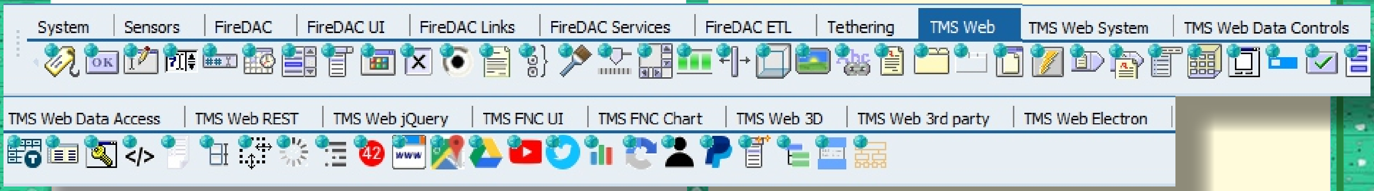


Figure 1,2: The TMS web Tab in Delphi (undocked view): the number of components that will come available is immense

In this first part of the article we have three main goals:

1. A very short list of how to install your **TMSWebcore** for Delphi and for Lazarus under Windows 10. This is nothing but a bulleted list with installation orders in pure text. This checklist is separately published in this magazine and you can download it is a text file. It will also become available as Install file by TMS.
   For each kind of installation there are two files: an installation file for Delphi and a separate one for Lazarus.
   This is the file for Delphi.

2. The same file created with images and extra annotations and images (The Delphi installation), which is at any time to be done first: so even if you install for Lazarus you first must install TMSWEBCcore.

3. An article that gives you insight in how the installation for Lazarus (2.X) can be done and answers some questions that might rise as soon as you install. It is a separate article on page 48 in this magazine

DOCUMENTATION
`c:\Users\Detlef1130\Documents\tmssoftware\TMS WEB Core RSXE12\Doc\`
You find here: TMSWEBCoreSetupGuide.pdf
TMSWEBCoreDevGuide.pdf

DEMOS
`c:\Users\Public\Documents\tmssoftware\TMS WEB Core Demos\`
(if you haven't changed the path).

PURCHASE TMSWEBCore
`https://www.tmssoftware.com/site/tmswebcoreintro.asp`

PREVIEW TMSFNCComponents:
`https://www.tmssoftware.com/site/tmsfncintro.asp`

One thing which is very important and we are working very hard on: creating a very easy to use Server for this, as well some very advanced ORM extras as XData can present us. That will be in the next issue. We also will create some instruction Videos and will create events where you can learn to work with this. So if you need only a very brief information: Here is the text-bullet list for Installation.

5 STEPS INSTALLATION LIST:

1. **Download the latest version of TMSWEBCore**

2. **Unzip The file in to a directory of your choice.**
   (it will give you a setup.exe)
   This directory is only needed for installation, after installing you can remove it.

3. **Start executing the `Setup.exe`.**
   After the first install which creates a Directory called `..\TMS WEB Core`

4. **After having installed this part another setup will start:**
   here you can make choices what you want to install: Delphi (it can vary depending on the number of Delphi's you want to install) It will show how many installations it could find.
   You can also install this for Lazarus, even if you do not have that installed. It then will create that directory and that can be used if you are at any time installing or planning to use it.
   (You do not need it for a Delphi installation). It creates this Directory `..\TMS WEB Core RSXE12` in case you selected Delphi RIO

5. **After installing this you need to check if your Installation is available:**
   During start you will see the TMS version on the Splash screen of Delphi during start. In Tools > Options you will find a section of TMS Web > Options. There you can find the version of the component you installed plus all other information.

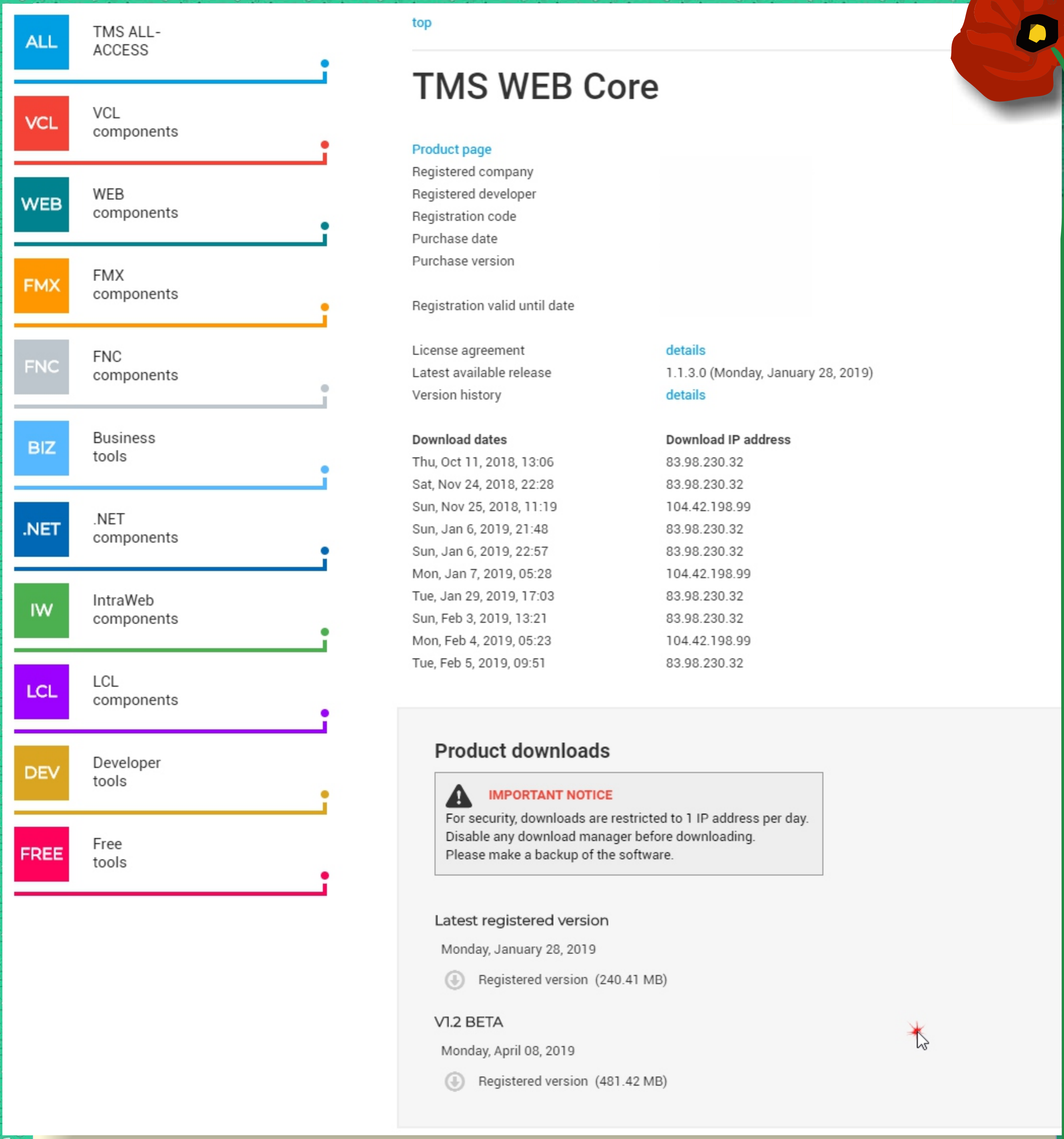
That's all folks!

The documentation and trial projects are installed here `c:\Users\Public\Documents\tmssoftware\`

(there is extra information at the end of this article)
For the list it is important to know that we might have overdone in simplicity: Don't get angry. We only want to make sure however experienced you are - you will have the least trouble to understand the installation and experience it as "easy".

GUIDED INSTALLATION FOR DELPHI: 5 illustrated steps

Heads up you must decide whether you are going to install for Delphi only or for both: Delphi and Lazarus. It makes a difference if you want to install them both. If you install Delphi only you will need only to check the installation for it as soon as it gets available and Run: TMSWEBCore XXXX.zip which contains the setup.exe Please note: If you want to install it for the second time or even more, as an update, you MUST clear everything from the installation: do not install it over the last version - it will not install properly. So uninstall it: make sure your Delphi installation is empty and has no components from earlier installations left.



ALL TMS ALL-ACCESS

VCL VCL components

WEB WEB components

FMX FMX components

FNC FNC components

BIZ Business tools

.NET .NET components

IW IntraWeb components

LCL LCL components

DEV Developer tools

FREE Free tools

top

TMS WEB Core

Product page

Registered company

Registered developer

Registration code

Purchase date

Purchase version

Registration valid until date

License agreement	details
Latest available release	1.1.3.0 (Monday, January 28, 2019)
Version history	details

Download dates	Download IP address
Thu, Oct 11, 2018, 13:06	83.98.230.32
Sat, Nov 24, 2018, 22:28	83.98.230.32
Sun, Nov 25, 2018, 11:19	104.42.198.99
Sun, Jan 6, 2019, 21:48	83.98.230.32
Sun, Jan 6, 2019, 22:57	83.98.230.32
Mon, Jan 7, 2019, 05:28	104.42.198.99
Tue, Jan 29, 2019, 17:03	83.98.230.32
Sun, Feb 3, 2019, 13:21	83.98.230.32
Mon, Feb 4, 2019, 05:23	104.42.198.99
Tue, Feb 5, 2019, 09:51	83.98.230.32

Product downloads

⚠ IMPORTANT NOTICE
For security, downloads are restricted to 1 IP address per day.
Disable any download manager before downloading.
Please make a backup of the software.

Latest registered version

Monday, January 28, 2019

Registered version (240.41 MB)

V1.2 BETA

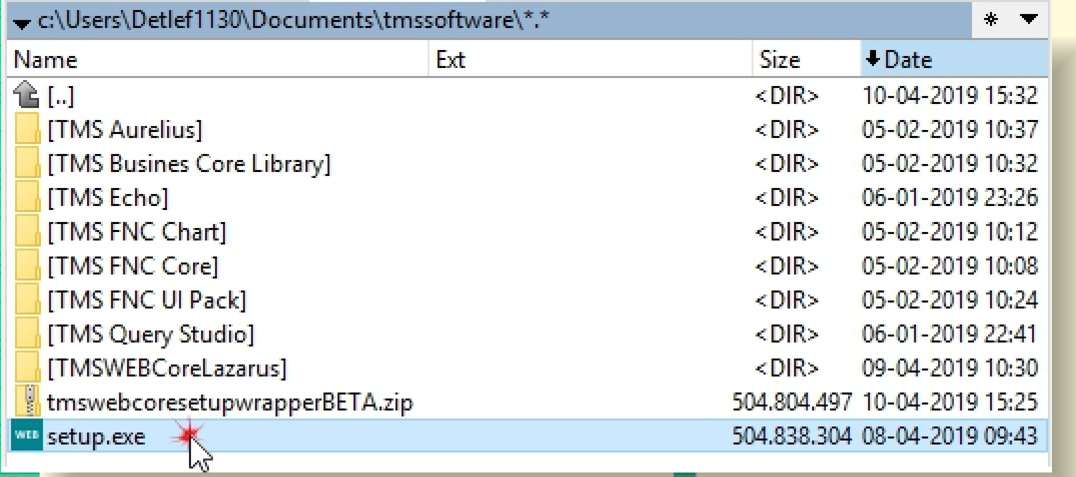
Monday, April 08, 2019

Registered version (481.42 MB)

Figure 3: You will have a very good overview of everything you install and what own, versions etc.
For obvious reasons we blanked some details

1. Download your File from TMS. (Exe)

See figure 3 on the last page (13)

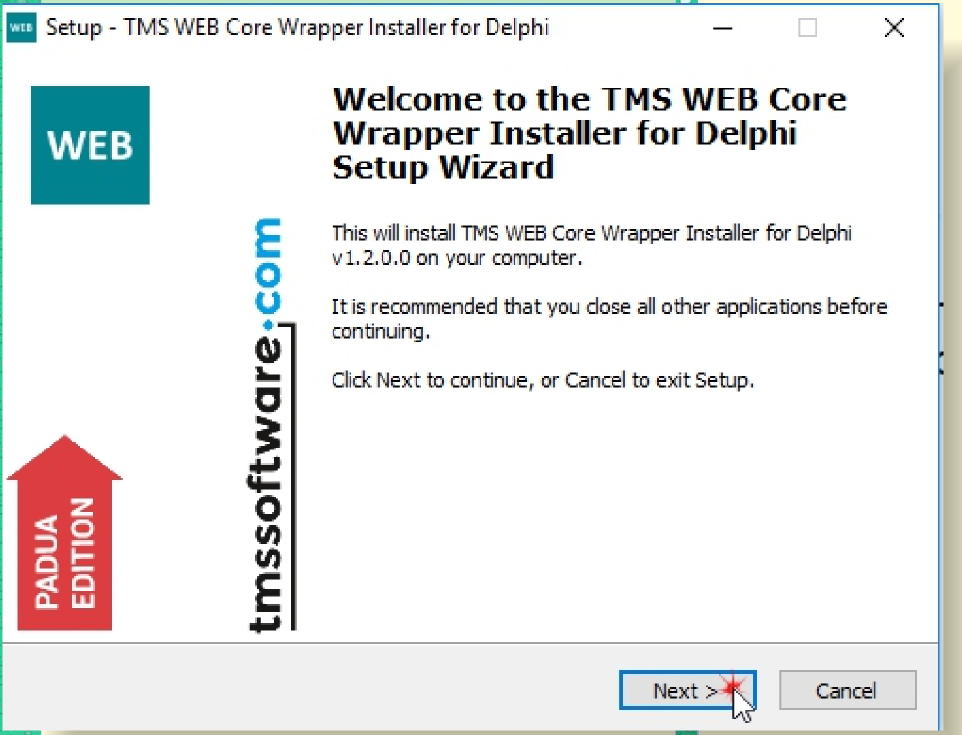


c:\Users\Detlef1130\Documents\tmssoftware\*.*

Name	Ext	Size	Date
[..]		<DIR>	10-04-2019 15:32
[TMS Aurelius]		<DIR>	05-02-2019 10:37
[TMS Busines Core Library]		<DIR>	05-02-2019 10:32
[TMS Echo]		<DIR>	06-01-2019 23:26
[TMS FNC Chart]		<DIR>	05-02-2019 10:12
[TMS FNC Core]		<DIR>	05-02-2019 10:08
[TMS FNC UI Pack]		<DIR>	05-02-2019 10:24
[TMS Query Studio]		<DIR>	06-01-2019 22:41
[TMSWEBCoreLazarus]		<DIR>	09-04-2019 10:30
tmswebcoresetupwrapperBETA.zip		504.804.497	10-04-2019 15:25
setup.exe		504.838.304	08-04-2019 09:43

Figure 4: First this package will be installed:

2. The first window will appear and all you need to do is click on OK



Setup - TMS WEB Core Wrapper Installer for Delphi

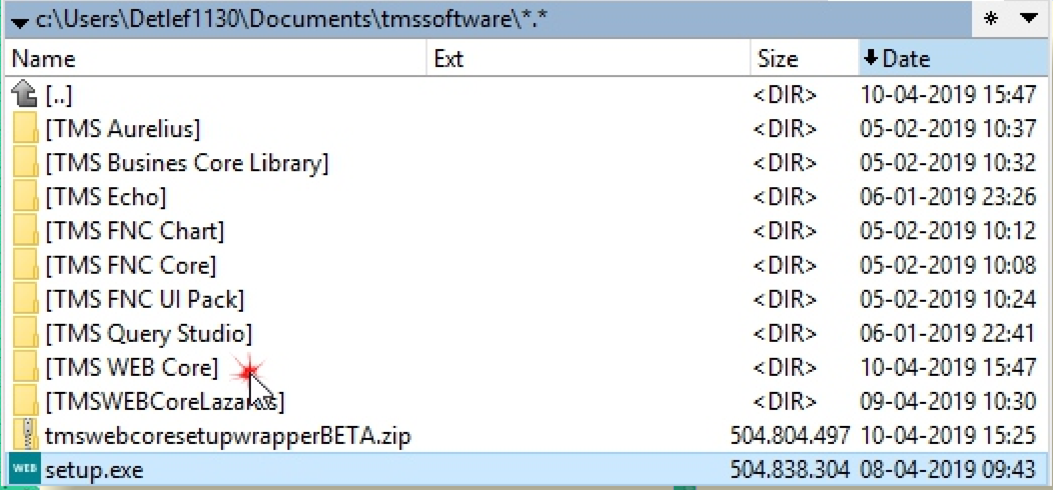
Welcome to the TMS WEB Core Wrapper Installer for Delphi Setup Wizard

This will install TMS WEB Core Wrapper Installer for Delphi v1.2.0.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

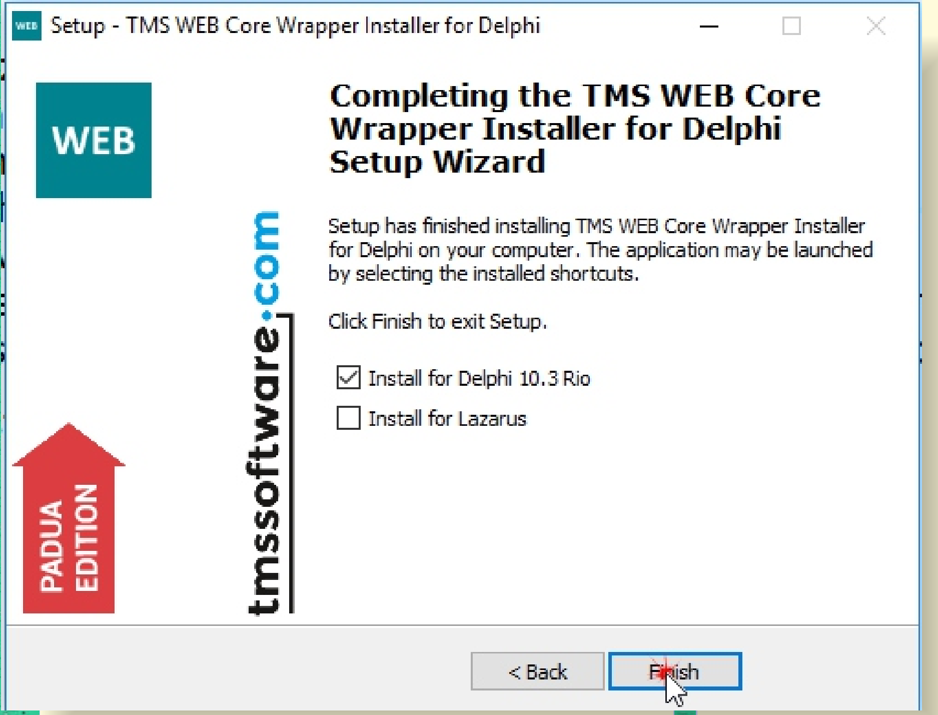
Figure 5: This is the first of the installations that need to be done.
It will finalize by installing TMSWebcore, as you can see in figure nr.6 on the next page(15)



c:\Users\Detlef1130\Documents\tmssoftware*.*

Name	Ext	Size	Date
[..]		<DIR>	10-04-2019 15:47
[TMS Aurelius]		<DIR>	05-02-2019 10:37
[TMS Busines Core Library]		<DIR>	05-02-2019 10:32
[TMS Echo]		<DIR>	06-01-2019 23:26
[TMS FNC Chart]		<DIR>	05-02-2019 10:12
[TMS FNC Core]		<DIR>	05-02-2019 10:08
[TMS FNC UI Pack]		<DIR>	05-02-2019 10:24
[TMS Query Studio]		<DIR>	06-01-2019 22:41
[TMS WEB Core]		<DIR>	10-04-2019 15:47
[TMSWEBCoreLazarus]		<DIR>	09-04-2019 10:30
tmswebcoresetupwrapperBETA.zip		504.804.497	10-04-2019 15:25
setup.exe		504.838.304	08-04-2019 09:43

Figure 6: this is the Directory that will be created by this first installation.It only prepares for the real installation that will be started after this process.



Completing the TMS WEB Core Wrapper Installer for Delphi Setup Wizard

Setup has finished installing TMS WEB Core Wrapper Installer for Delphi on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Install for Delphi 10.3 Rio

□ Install for Lazarus

Figure 7: Check the Delphi box(es) for your Delphi versions you want it to be installed for.

3. The installation choice for Delphi and or Lazarus will become available

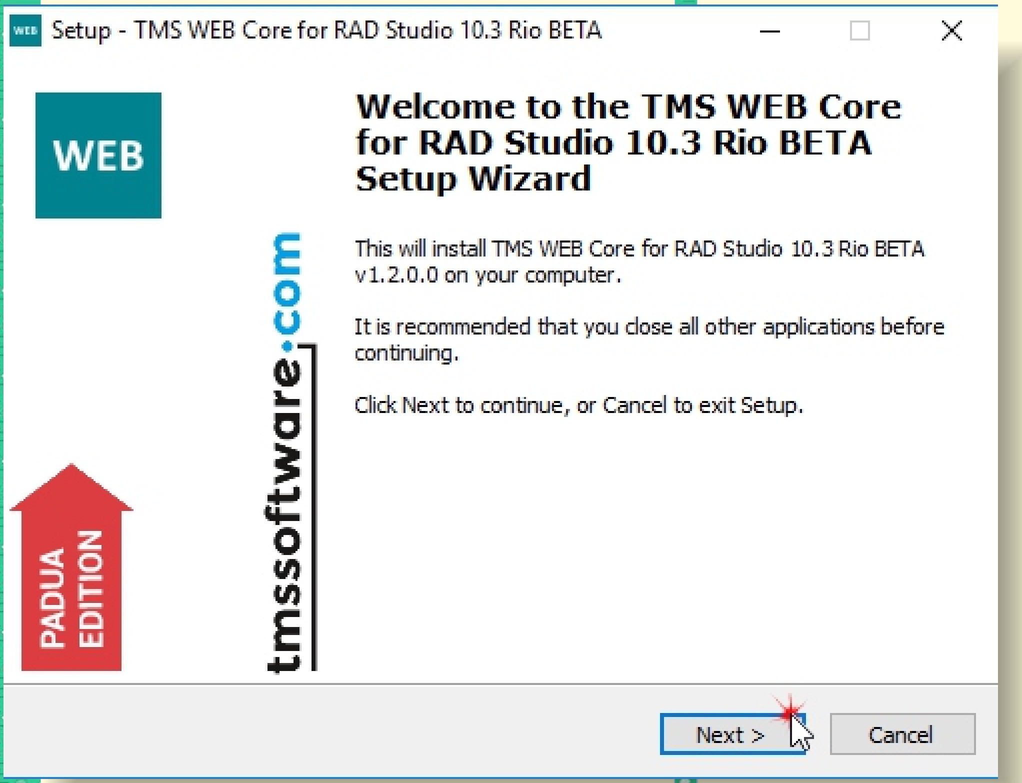
Check Delphi for only Delphi installation,
TMS WEB CoreRSXE12
(if it is for Delphi RIO)

INSTALLING LAZARUS & DELPHI
You can of course check both:
Lazarus and Delphi
and both will be installed.

TMSWEBCoreLazarus
(minimal version Lazarus 2.X)
and
TMS WEB Core RSXE12
(if it is for Delphi RIO)

In this case first Lazarus will be installed. After that Delphi will be installed.

If you have no Lazarus installed then you of course do not need to check that. It means no harm if you install it by accident both. There will only be an extra installation added.


Setup - TMS WEB Core for RAD Studio 10.3 Rio BETA

Welcome to the TMS WEB Core for RAD Studio 10.3 Rio BETA Setup Wizard

This will install TMS WEB Core for RAD Studio 10.3 Rio BETA v1.2.0.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 8: Each of the installations for a Delphi version will have a similar start screen

There is no check done if you have Lazarus installed and if your Lazarus program installed is Lazarus version 2.0 or above.

The Installation directory for TMSWecore for Lazarus looks like this:

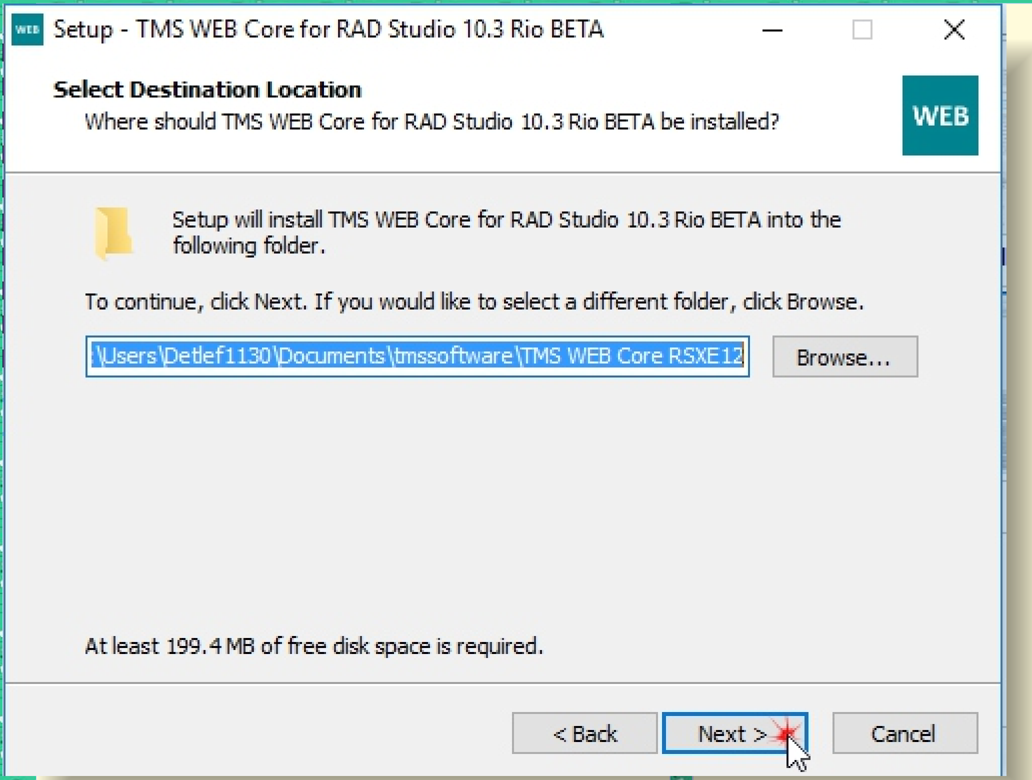
➔ ../TMSWEBCoreLazarus/
(minimal version Lazarus 2.X)

➔ or if you want only Lazarus installed you must under Windows also install for Delphi even if you have no Delphi.

➔ This is because your package will be registered then and only then it will work properly.

For further installation details of Lazarus please go to the Section of Installing Lazarus in this article

GUIDED INSTALLATION FOR DELPHI:



Setup - TMS WEB Core for RAD Studio 10.3 Rio BETA

Select Destination Location

Where should TMS WEB Core for RAD Studio 10.3 Rio BETA be installed?

Setup will install TMS WEB Core for RAD Studio 10.3 Rio BETA into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

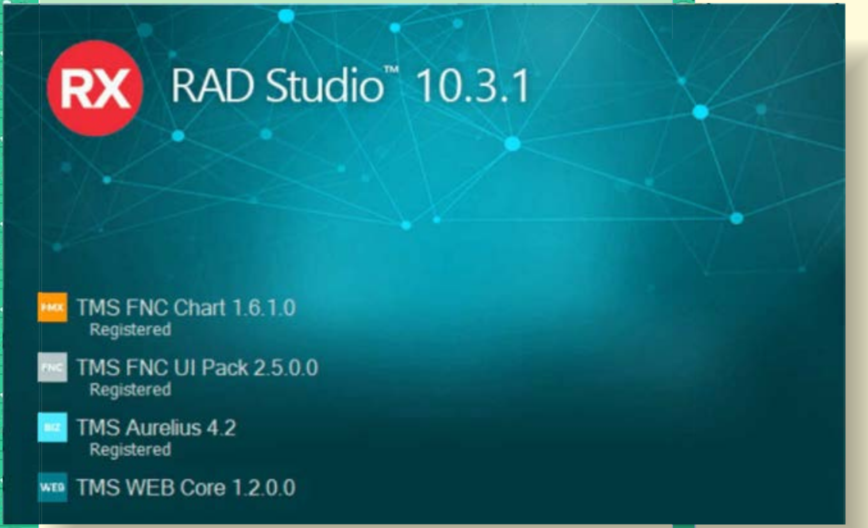
:\Users\Detlef1130\Documents\tmssoftware\TMS WEB Core RSXE12

At least 199.4 MB of free disk space is required.

Figure 9: You can give in any path you wish.

After installation you will find that WebCore is installed in to your chosen Delphi version. To convince your self that it's available do the following:

4. Start Delphi for checking all of its TMSWEB Core installing



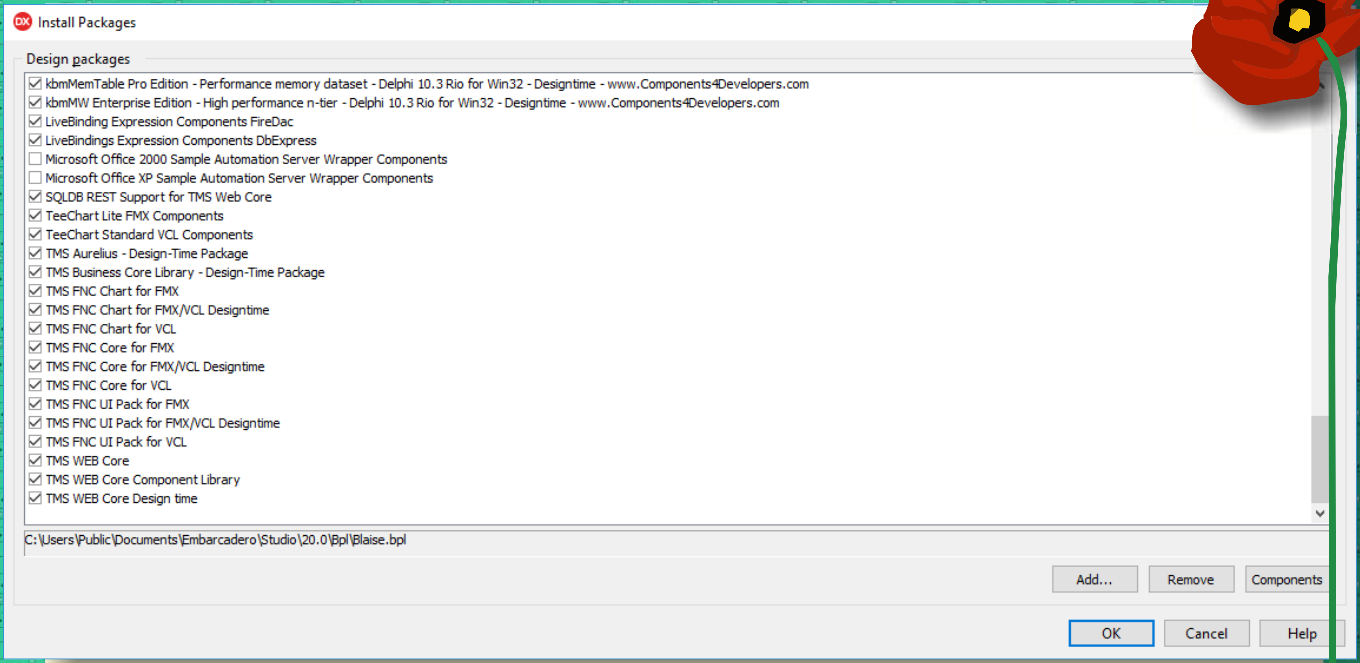
TMS FNC Chart 1.6.1.0
Registered
TMS FNC UI Pack 2.5.0.0
Registered
TMS Aurelius 4.2
Registered
TMS WEB Core 1.2.0.0

Figure 10: The RADStudio/Delphi Splash-screen which shows your component version loaded:

Start Delphi and during the start you will find a Notification that WebCore is available.

Now to make sure everithing works fine: **let's test the installation:**

GUIDED INSTALLATION FOR DELPHI:



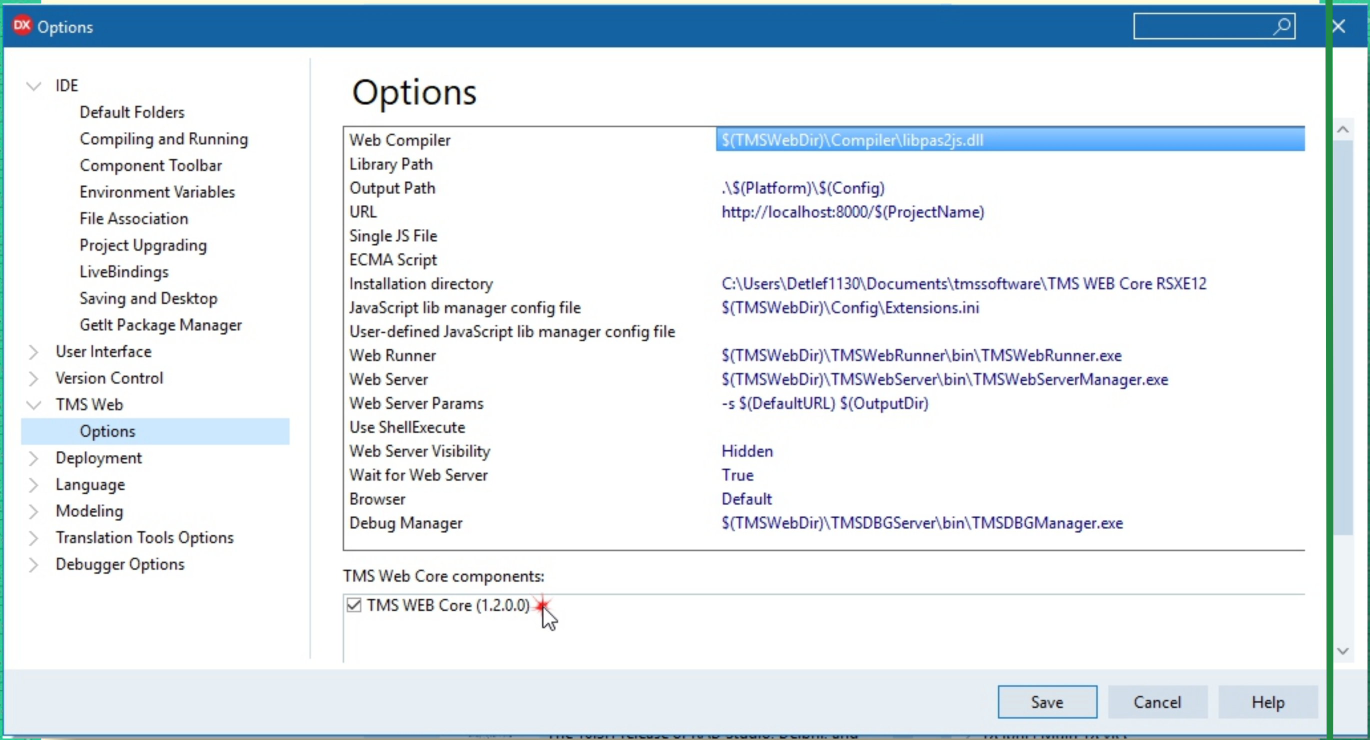
Install Packages

Design packages

- ☑ kbmMemTable Pro Edition - Performance memory dataset - Delphi 10.3 Rio for Win32 - Designtime - www.Components4Developers.com
- ☑ kbmMW Enterprise Edition - High performance n-tier - Delphi 10.3 Rio for Win32 - Designtime - www.Components4Developers.com
- ☑ LiveBinding Expression Components FireDac
- ☑ LiveBindings Expression Components DbExpress
- □ Microsoft Office 2000 Sample Automation Server Wrapper Components
- □ Microsoft Office XP Sample Automation Server Wrapper Components
- ☑ SQLDB REST Support for TMS Web Core
- ☑ TeeChart Lite FMX Components
- ☑ TeeChart Standard VCL Components
- ☑ TMS Aurelius - Design-Time Package
- ☑ TMS Business Core Library - Design-Time Package
- ☑ TMS FNC Chart for FMX
- ☑ TMS FNC Chart for FMX/VCL Designtime
- ☑ TMS FNC Chart for VCL
- ☑ TMS FNC Core for FMX
- ☑ TMS FNC Core for FMX/VCL Designtime
- ☑ TMS FNC Core for VCL
- ☑ TMS FNC UI Pack for FMX
- ☑ TMS FNC UI Pack for FMX/VCL Designtime
- ☑ TMS FNC UI Pack for VCL
- ☑ TMS WEB Core
- ☑ TMS WEB Core Component Library
- ☑ TMS WEB Core Design time

C:\Users\Public\Documents\Embarcadero\Studio\20.0\Bpl\Blaise.bpl

Figure 11. Go to Component → Install Components
and you will see an overview of all the components that are installed



Options

Web Compiler	$(TMSWebDir)\Compiler\libpas2js.dll
Library Path	
Output Path	.\$(Platform)\$(Config)
URL	http://localhost:8000/$(ProjectName)
Single JS File	
ECMA Script	
Installation directory	C:\Users\Detlef1130\Documents\tmssoftware\TMS WEB Core RSXE12
JavaScript lib manager config file	$(TMSWebDir)\Config\Extensions.ini
User-defined JavaScript lib manager config file	
Web Runner	$(TMSWebDir)\TMSWebRunner\bin\TMSWebRunner.exe
Web Server	$(TMSWebDir)\TMSWebServer\bin\TMSWebServerManager.exe
Web Server Params	-s $(DefaultURL) $(OutputDir)
Use ShellExecute	
Web Server Visibility	Hidden
Wait for Web Server	True
Browser	Default
Debug Manager	$(TMSWebDir)\TMSDBGServer\bin\TMSDBGManager.exe

TMS Web Core components:

☑ TMS WEB Core (1.2.0.0)

Figure 12: Go to Tools → Options | TMS Web → Options and you have all kinds of details.
The version number is also available. It seems everything is available.

5. TESTING with the first project

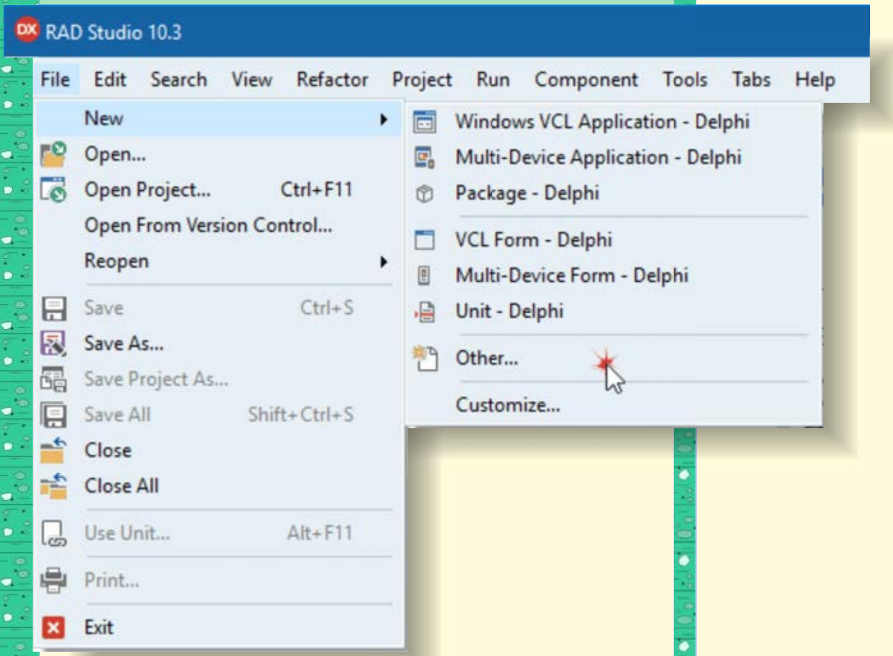


Figure 13: Click on Other to get the New Items overview.

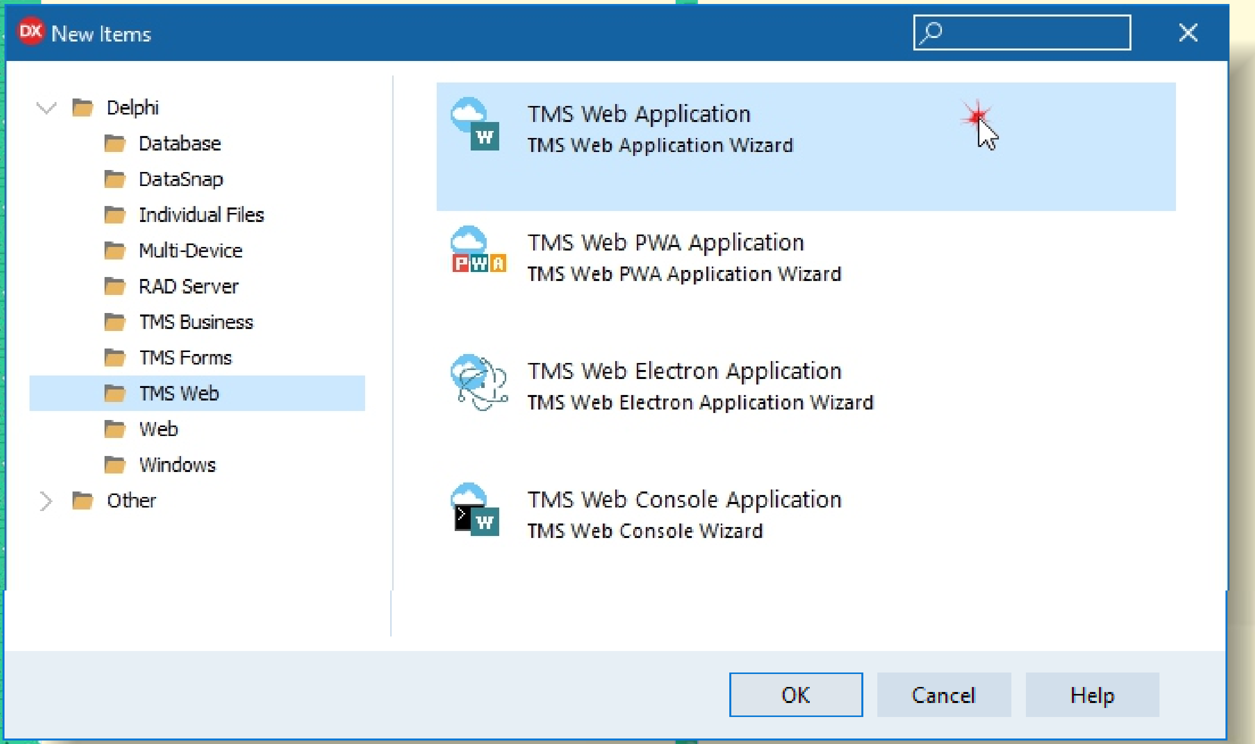


Figure 14: Chose for this time TMS Web Application.

We want to create a very simple app, just to test it all. At first sight a form will be shown. Remember this form is not a usual form.
It can only contain TMS Web components.

EXTRA INFORMATION: THE README.TXT

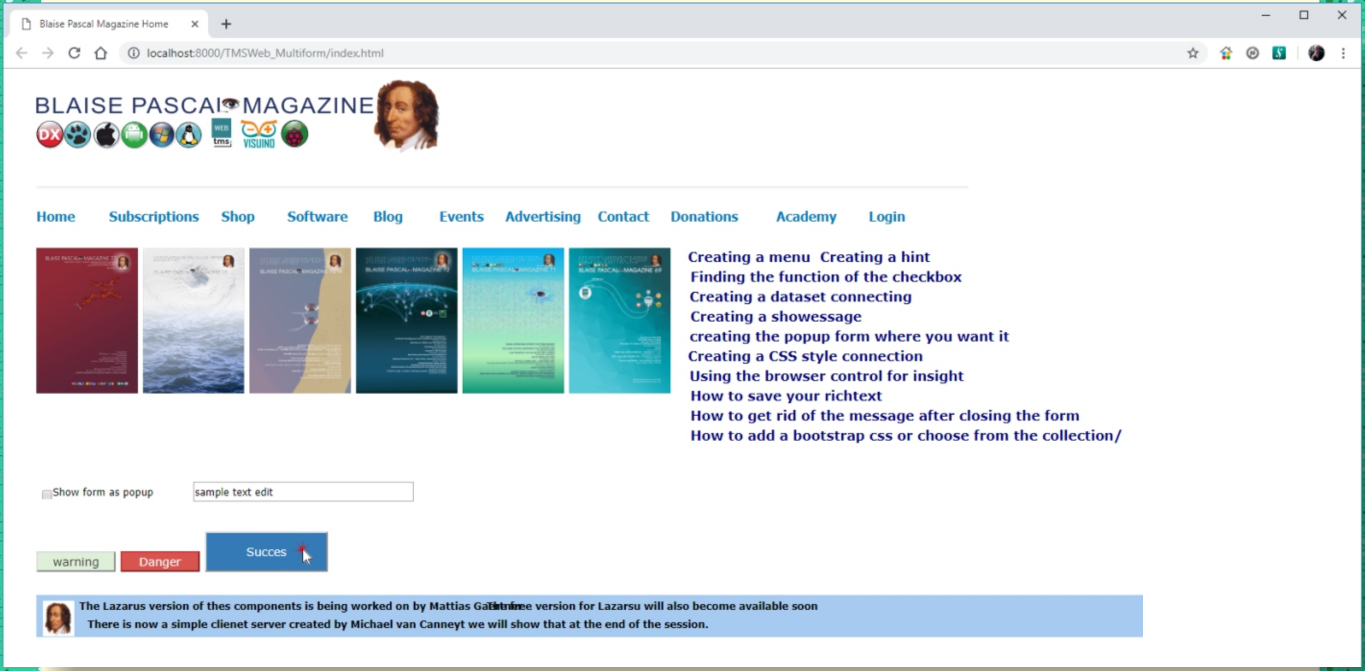
TMS WEB Core v1.2 Padua Edition

TMS WEB Core landing page:http://web.tmssoftware.com
TMS WEB Core product page: https://www.tmssoftware.com/site/tmswebcore.asp

TMS WEB Core provides out of the box a series of standard UI controls.
Additional controls & tools are available:
Web-enabled TMS FNC UI Controls

Via **TMS FNC UI Pack**, **TMS FNC Chart, TMS FNC Dashboard Pack** additional UI controls are available. To have access to the latest web-enabled **TMS FNC UI** controls in **TMS WEB Core**, it is important to download & install the latest release of **TMS FNC UI Pack, TMS FNC Dashboard Pack & TMS FNC Chart.**
When the latest version is installed, an extra tab will be available in the Delphi IDE tool palette with the FNC UI controls. It is also required to have this latest version installed to be able to run the demos with FNC controls.
The demos for FNC web-enabled controls are in the folder FNC under Demos: in this case installed here
`c:\Users\Public\Documents\tmssoftware\TMS WEB Core Demos\`
With **TMS XDATA V3.0** there is now also full support for consuming data from **XData REST API servers.**
The are several demos for **XData dataset** client usage from TMS WEB Core web client applications. We will elaborate on this in the next issue.
The **DEMOS** can be found under the subfolder **XData** under demos in the install folder.
TMS XData can be downloaded from
https://www.tmssoftware.com/site/xdata.asp

There is also the **TMSWEBCoreDevGuide.pdf** which is a highly valued information for developers. You can find all this in your download area.



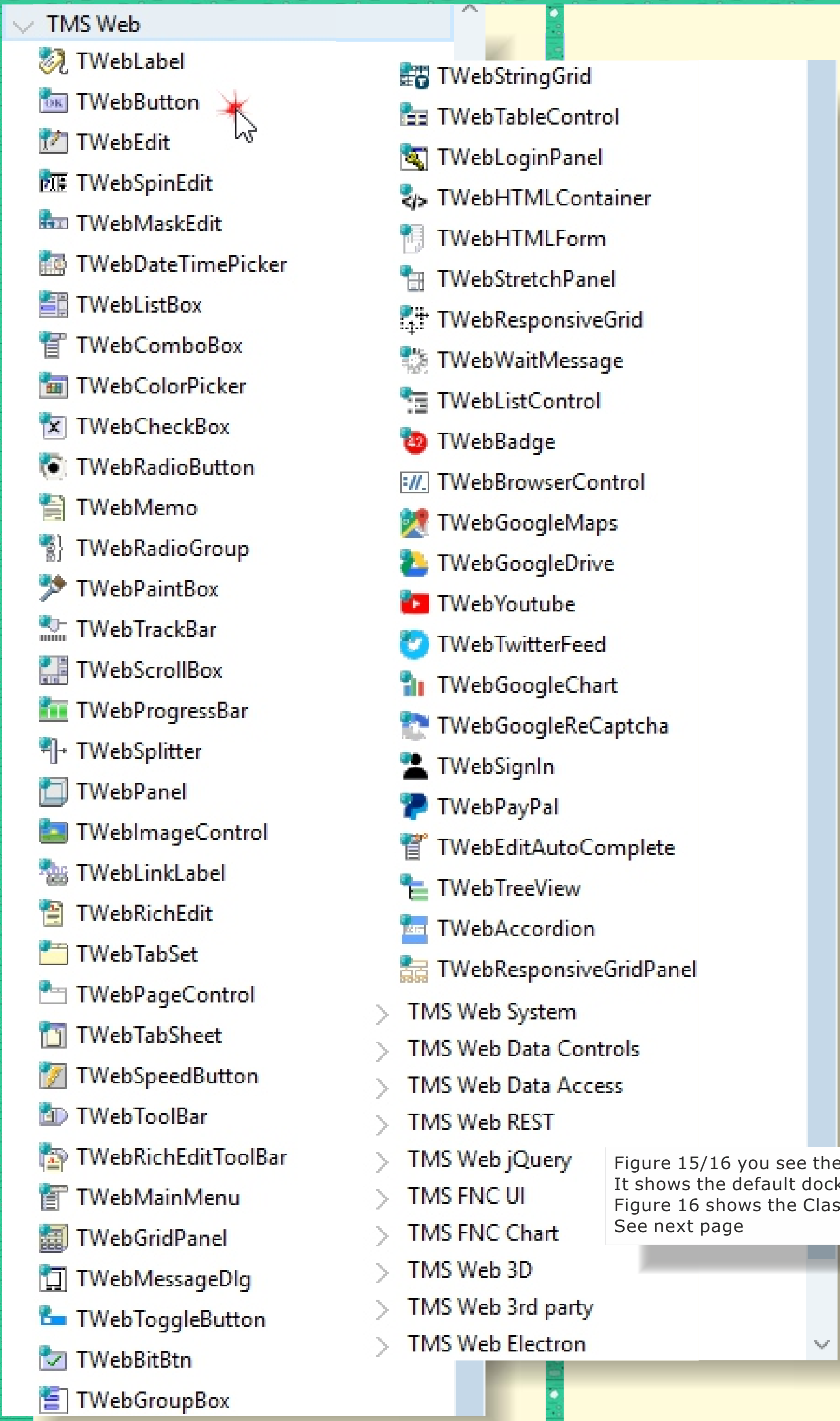
Creating a menu Creating a hint
Finding the function of the checkbox
Creating a dataset connecting
Creating a showessage
creating the popup form where you want it
Creating a CSS style connection
Using the browser control for insight
How to save your richtext
How to get rid of the message after closing the form
How to add a bootstrap css or choose from the collection/

□ Show form as popup   sample text edit

warning   Danger   Succes

The Lazarus version of thes components is being worked on by Mattias Ga[illegible]e free version for Lazarsu will also become available soon
There is now a simple clienet server created by Michael van Canneyt we will show that at the end of the session.

Figure 15: We are building our website once again and now with TMSWeb, here is a preview of the first trials we created already.



TMS Web

- TWebLabel
- TWebButton
- TWebEdit
- TWebSpinEdit
- TWebMaskEdit
- TWebDateTimePicker
- TWebListBox
- TWebComboBox
- TWebColorPicker
- TWebCheckBox
- TWebRadioButton
- TWebMemo
- TWebRadioGroup
- TWebPaintBox
- TWebTrackBar
- TWebScrollBox
- TWebProgressBar
- TWebSplitter
- TWebPanel
- TWebImageControl
- TWebLinkLabel
- TWebRichEdit
- TWebTabSet
- TWebPageControl
- TWebTabSheet
- TWebSpeedButton
- TWebToolBar
- TWebRichEditToolBar
- TWebMainMenu
- TWebGridPanel
- TWebMessageDlg
- TWebToggleButton
- TWebBitBtn
- TWebGroupBox
- TWebStringGrid
- TWebTableControl
- TWebLoginPanel
- TWebHTMLContainer
- TWebHTMLForm
- TWebStretchPanel
- TWebResponsiveGrid
- TWebWaitMessage
- TWebListControl
- TWebBadge
- TWebBrowserControl
- TWebGoogleMaps
- TWebGoogleDrive
- TWebYoutube
- TWebTwitterFeed
- TWebGoogleChart
- TWebGoogleReCaptcha
- TWebSignIn
- TWebPayPal
- TWebEditAutoComplete
- TWebTreeView
- TWebAccordion
- TWebResponsiveGridPanel

- TMS Web System
- TMS Web Data Controls
- TMS Web Data Access
- TMS Web REST
- TMS Web jQuery
- TMS FNC UI
- TMS FNC Chart
- TMS Web 3D
- TMS Web 3rd party
- TMS Web Electron

Figure 15/16 you see the loaded Components.
It shows the default docked view.
Figure 16 shows the Classical undocked view.
See next page

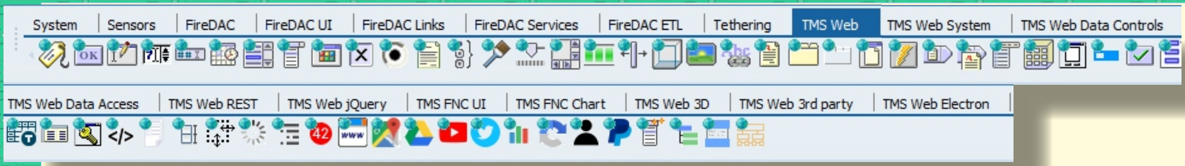


Figure 16:
Part of the components TMS has to offer for the web

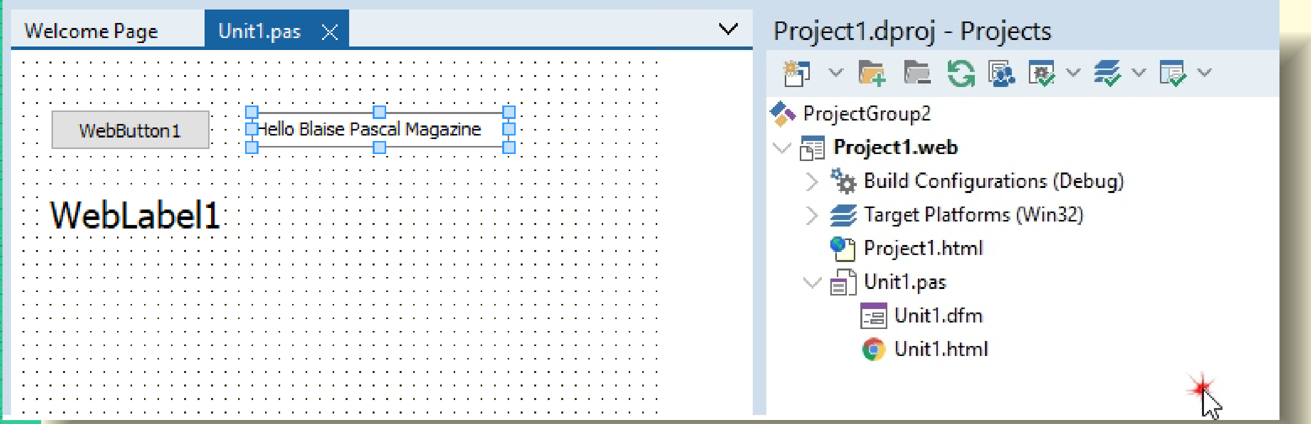


Figure 17 : this form shows the few components added for the project.

We created a simple project: Through a button add the text to the label.
Do realize you can't use normal components even the resemblance is overwhelming.

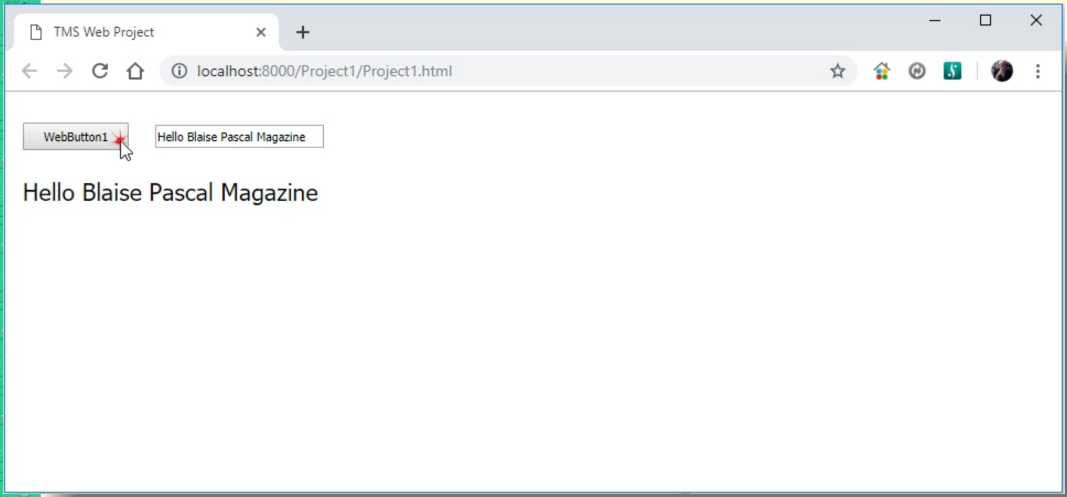

TMS Web Project
localhost:8000/Project1/Project1.html
WebButton1
Hello Blaise Pascal Magazine
Hello Blaise Pascal Magazine

Figure 18 : Google Chrome shows the result.

So the test succeeded!

On the nexpt pages you will find extra information provided by TMS

Supported Delphi IDEs
TMS WEB Core supports Delphi XE7, Delphi XE8, Delphi 10.0 Seattle, Delphi 10.1 Berlin, Delphi 10.2 Tokyo, Delphi 10.3 Rio

Packages
TMS WEB Core uses following packages in package group
TMSWEBCoreGroupDXExy.groupproj
with xy: 7 for Delphi XE7 8 for Delphi XE8 9 for Delphi 10.0 Seattle 10 for Delphi 10.1 Berlin 11 for Delphi 10.2 Tokyo
12 for Delphi 10.3 Rio

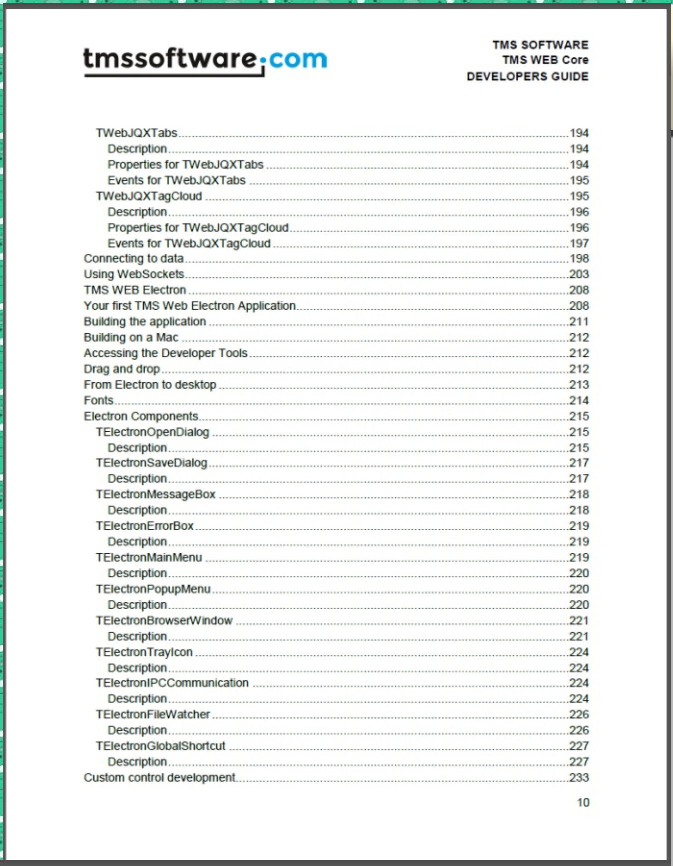
Core package:
TMSWEBCorePkgDXExy.dproj
Design-time component package:
TMSWEBCorePkgLibDXExy.dproj
Design-time IDE plugin package:
TMSWEBCorePkgDEDXExy.dproj

Registry keys
All settings are organized in the registry under key:

```
HKEY_CURRENT_USER\Software\tmssoftware\TMS WEB Core
```

The default registry key settings are:

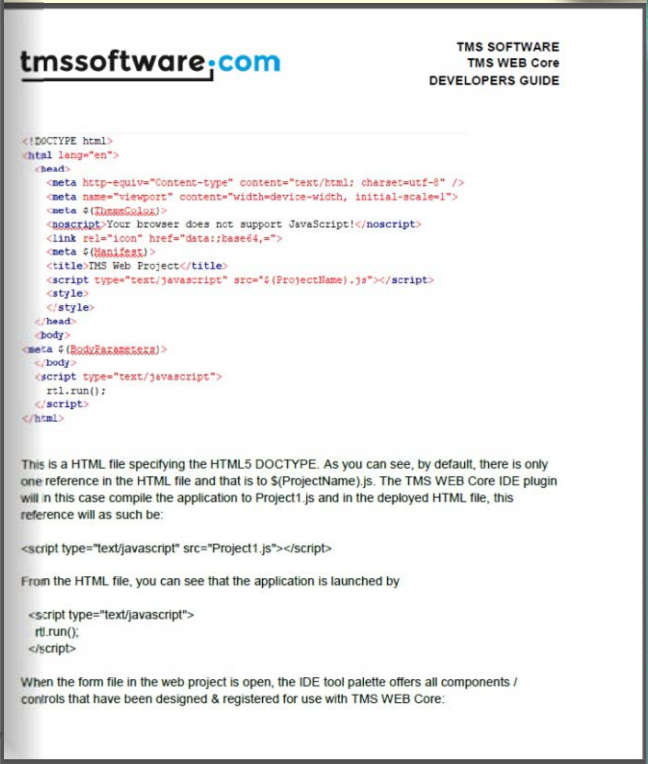
```
[HKEY_CURRENT_USER\Software\TMSSoftware\TMS WEB Core]
"Pas2JSCompiler"="$(TMSWebDir)\\Compiler\\libpas2js.dll"
"LibraryPaths"="$(TMSWebDir)\\Core Source;$(TMSWebDir)\\Core Source\\RTL"
"OutputPath"=".\\$(Platform)\\$(Config)"
"DefaultURL"="http://localhost:8000/$(ProjectName)"
"SingleJS"=dword:00000002
"WebServer"="$(TMSWebDir)\\TMSWebServer\\bin\\TMSWebServerManager.exe"
"WebServerParams"="-s $(DefaultURL) $(OutputDir)"
"DebugMode"=dword:00000001
"EcmaScript"=dword:00000001
"PPOutputPath"=".\\$(Platform)\\$(Config)"
"Browser"=dword:00000001
"WebServerShow"=dword:00000000
"WebServerWait"=dword:00000002
"WebServerVisibility"=dword:00000001
"StaticHTML"=dword:00000002
"License"=""
"DebugManager"="$(TMSWebDir)\\TMSDBGServer\\bin\\TMSDBGManager.exe"
"JSLibConfig"="$(TMSWebDir)\\Config\\Extensions.ini"
"JSLibConfigUser"=""
"WebRunner"="$(TMSWebDir)\\TMSWebRunner\\bin\\TMSWebRunner.exe"
```



tmssoftware.com

TMS SOFTWARE
TMS WEB Core
DEVELOPERS GUIDE

10



tmssoftware.com

TMS SOFTWARE
TMS WEB Core
DEVELOPERS GUIDE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta $(ThemeColor)>
    <noscript>Your browser does not support JavaScript!</noscript>
    <link rel="icon" href="data:;base64,=">
    <meta $(Manifest)>
    <title>TMS Web Project</title>
    <script type="text/javascript" src="$(ProjectName).js"></script>
    <style>
    </style>
  </head>
  <body>
<meta $(BodyParameters)>
  </body>
  <script type="text/javascript">
    rtl.run();
  </script>
</html>
```

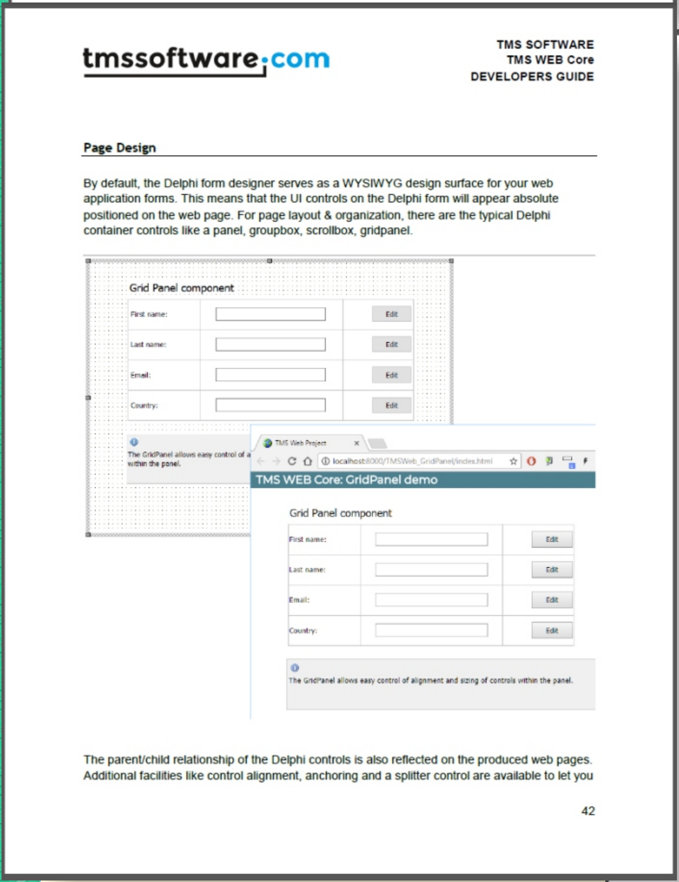
This is a HTML file specifying the HTML5 DOCTYPE. As you can see, by default, there is only one reference in the HTML file and that is to $(ProjectName).js. The TMS WEB Core IDE plugin will in this case compile the application to Project1.js and in the deployed HTML file, this reference will as such be:

```
<script type="text/javascript" src="Project1.js"></script>
```

From the HTML file, you can see that the application is launched by

```
 <script type="text/javascript">
  rtl.run();
 </script>
```

When the form file in the web project is open, the IDE tool palette offers all components / controls that have been designed & registered for use with TMS WEB Core:



tmssoftware.com

TMS SOFTWARE
TMS WEB Core
DEVELOPERS GUIDE

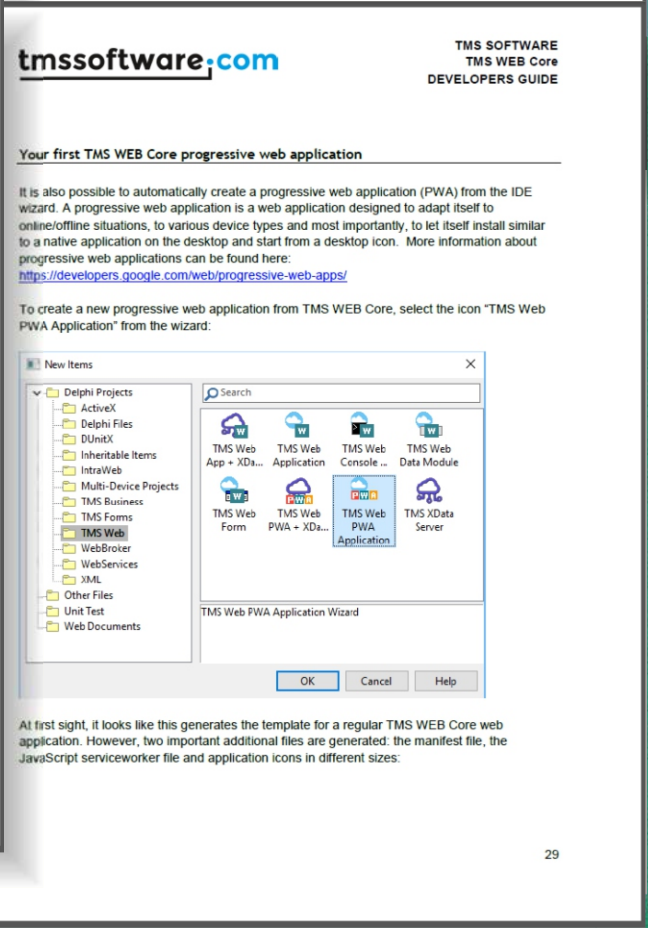
Page Design

By default, the Delphi form designer serves as a WYSIWYG design surface for your web application forms. This means that the UI controls on the Delphi form will appear absolute positioned on the web page. For page layout & organization, there are the typical Delphi container controls like a panel, groupbox, scrollbox, gridpanel.

Grid Panel component
First name:
Edit
Last name:
Edit
Email:
Edit
Country:
Edit
The GridPanel allows easy control of a
within the panel.
TMS Web Project
localhost:8000/TMSWeb_GridPanel/index.html
TMS WEB Core: GridPanel demo
Grid Panel component
First name:
Edit
Last name:
Edit
Email:
Edit
Country:
Edit
The GridPanel allows easy control of alignment and sizing of controls within the panel.

The parent/child relationship of the Delphi controls is also reflected on the produced web pages. Additional facilities like control alignment, anchoring and a splitter control are available to let you

42



tmssoftware.com

TMS SOFTWARE
TMS WEB Core
DEVELOPERS GUIDE

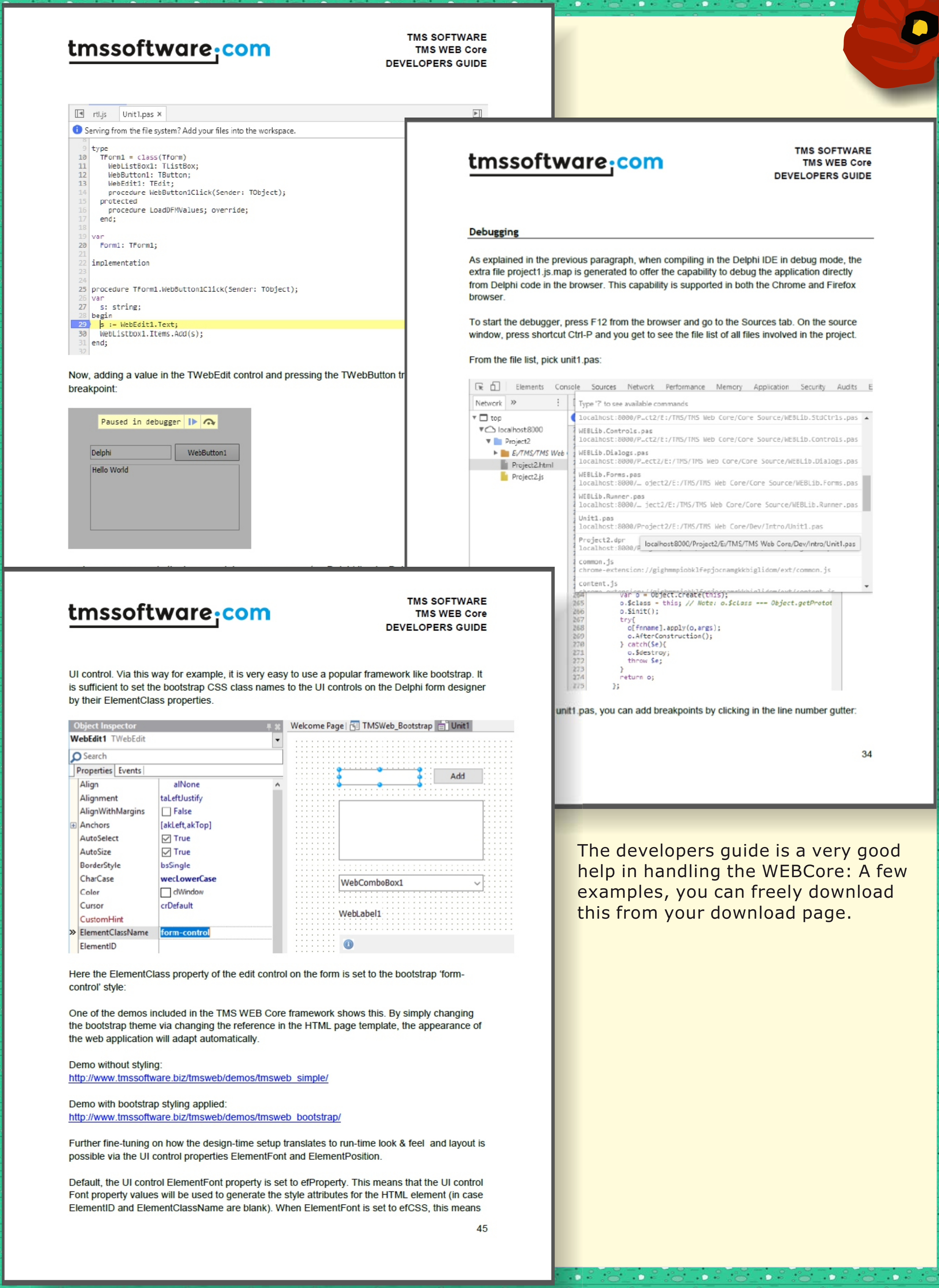
Your first TMS WEB Core progressive web application

It is also possible to automatically create a progressive web application (PWA) from the IDE wizard. A progressive web application is a web application designed to adapt itself to online/offline situations, to various device types and most importantly, to let itself install similar to a native application on the desktop and start from a desktop icon. More information about progressive web applications can be found here:
https://developers.google.com/web/progressive-web-apps/

To create a new progressive web application from TMS WEB Core, select the icon "TMS Web PWA Application" from the wizard:

At first sight, it looks like this generates the template for a regular TMS WEB Core web application. However, two important additional files are generated: the manifest file, the JavaScript serviceworker file and application icons in different sizes:

29



TMS SOFTWARE
TMS WEB Core
DEVELOPERS GUIDE

rtl.js
Unit1.pas ×
Serving from the file system? Add your files into the workspace.

```
type
  TForm1 = class(TForm)
    WebListBox1: TListBox;
    WebButton1: TButton;
    WebEdit1: TEdit;
    procedure WebButton1Click(Sender: TObject);
  protected
    procedure LoadDFMValues; override;
  end;

var
  Form1: TForm1;

implementation


procedure TForm1.WebButton1Click(Sender: TObject);
var
  s: string;
begin
  s := WebEdit1.Text;
  WebListbox1.Items.Add(s);
end;
```

Now, adding a value in the TWebEdit control and pressing the TWebButton tr
breakpoint:

TMS SOFTWARE
TMS WEB Core
DEVELOPERS GUIDE

Debugging

As explained in the previous paragraph, when compiling in the Delphi IDE in debug mode, the extra file project1.js.map is generated to offer the capability to debug the application directly from Delphi code in the browser. This capability is supported in both the Chrome and Firefox browser.

To start the debugger, press F12 from the browser and go to the Sources tab. On the source window, press shortcut Ctrl-P and you get to see the file list of all files involved in the project.

From the file list, pick unit1.pas:

```
localhost:8000/P…ct2/E:/TMS/TMS Web Core/Core Source/WEBLib.StdCtrls.pas
WEBLib.Controls.pas
localhost:8000/P…ct2/E:/TMS/TMS Web Core/Core Source/WEBLib.Controls.pas
WEBLib.Dialogs.pas
localhost:8000/P…ect2/E:/TMS/TMS Web Core/Core Source/WEBLib.Dialogs.pas
WEBLib.Forms.pas
localhost:8000/… oject2/E:/TMS/TMS Web Core/Core Source/WEBLib.Forms.pas
WEBLib.Runner.pas
localhost:8000/… ject2/E:/TMS/TMS Web Core/Core Source/WEBLib.Runner.pas
Unit1.pas
localhost:8000/Project2/E:/TMS/TMS Web Core/Dev/Intro/Unit1.pas
Project2.dpr
localhost:8000/P
common.js
chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/ext/common.js
content.js
```

localhost:8000/Project2/E:/TMS/TMS Web Core/Dev/Intro/Unit1.pas

```
        var o = Object.create(this);
        o.$class = this; // Note: o.$class === Object.getProtot
        o.$init();
        try{
          o[fnname].apply(o,args);
          o.AfterConstruction();
        } catch($e){
          o.$destroy;
          throw $e;
        }
        return o;
      };
```

unit1.pas, you can add breakpoints by clicking in the line number gutter:

34

TMS SOFTWARE
TMS WEB Core
DEVELOPERS GUIDE

UI control. Via this way for example, it is very easy to use a popular framework like bootstrap. It is sufficient to set the bootstrap CSS class names to the UI controls on the Delphi form designer by their ElementClass properties.

Align	alNone
Alignment	taLeftJustify
AlignWithMargins	□ False
Anchors	[akLeft,akTop]
AutoSelect	☑ True
AutoSize	☑ True
BorderStyle	bsSingle
CharCase	wecLowerCase
Color	□ clWindow
Cursor	crDefault
CustomHint	
ElementClassName	form-control
ElementID	

Here the ElementClass property of the edit control on the form is set to the bootstrap 'form-control' style:

One of the demos included in the TMS WEB Core framework shows this. By simply changing the bootstrap theme via changing the reference in the HTML page template, the appearance of the web application will adapt automatically.

Demo without styling:
http://www.tmssoftware.biz/tmsweb/demos/tmsweb_simple/

Demo with bootstrap styling applied:
http://www.tmssoftware.biz/tmsweb/demos/tmsweb_bootstrap/

Further fine-tuning on how the design-time setup translates to run-time look & feel and layout is possible via the UI control properties ElementFont and ElementPosition.

Default, the UI control ElementFont property is set to efProperty. This means that the UI control Font property values will be used to generate the style attributes for the HTML element (in case ElementID and ElementClassName are blank). When ElementFont is set to efCSS, this means

45

The developers guide is a very good help in handling the WEBCore: A few examples, you can freely download this from your download page.

This article is adapted from information at the Dutch website: https://www.ct.nl/

INTRODUCTION

Windows 10 regularly contacts Microsoft's data-processing servers. This may not be a problem for you. However, if you want to preserve your data privacy it is relatively straightforward to disable the regular default transmission of diagnostic data.

DIAGNOSTIC DATA

To prevent Windows from constantly sending diagnostic data back to Microsoft, it is sufficient to disable the DiagTrack service. This article explains what is behind this service, how to stop or disable it, and the purpose of this and other Windows telemetry functions. Telemetry (from the Greek tele = remote, and metron = measure) describes an automated communications process designed to monitor measurements and other data collected from remote clients.

The way Windows 10 regularly sends diagnostic data back to Microsoft is a frequent topic of online debate. Microsoft, as vendor and maintainer of a complex operating system, wants to ensure that your system remains stable and receives all needed updates. The more sceptical Windows users wonder whether the information Microsoft collects can be used to identify them and monitor their computer and browser use.
Microsoft offers adjustments in Windows default settings that allow you both to reduce the amount of data sent, and to view its contents. However, officially there is no way to disable telemetry.

This article discusses a number of ways in which you can filter or block communication with Microsoft's data-processing factory via a detour, based on a recent German study.

BSI

The German Federal Office for Information Security ("Bundesamt für Sicherheit in der Informationstechnik", usually abbreviated to BSI) is a high-level German federal government agency charged with managing government computer and communication security.
Its areas of expertise and responsibility include the security of computer applications, critical infrastructure protection, Internet security, cryptography, countering eavesdropping, certification of security products, and the accreditation of security test laboratories.

It is located in Bonn and employs over 600 people. Since February 2016 Arne Schönbohmits has been its president. He is a former business executive who took over the role from Michael Hange. BSI's predecessor was the cryptography department of Germany's foreign intelligence agency (BND). BSI continues to design cryptographic algorithms such as the Libelle cipher. It also initiated the development of the Gpg4win cryptographic suite.

The text that follows here is repeated on the next page. The BSI has investigated telemetry [illegible] Windows 10. Its findings have been made available in a study named SiSyPHuS Win10, whose aim was to assess the risks in using Windows 10, and to determine what conditions were needed for secure use of the operating system. In addition, the study includes helpful recommendations to users about how to maintain Windows 10 securely. The first part of this research has been made available online (see the link at the end of this article).

The BSI has investigated telemetry within Windows 10. Its findings have been made available in a study named SiSyPHuS Win10, whose aim was to assess the risks in using Windows 10, and to determine what conditions were needed for secure use of the operating system. In addition, the study includes helpful recommendations to users about how to maintain Windows 10 securely. The first part of this research has been made available online (see the link at the end of this article).

WINDOWS TELEMETRY SETTINGS

In Windows 10 you can locate options which influence the amount of data returned to Microsoft under 'Settings / Privacy / Diagnostic data and feedback'. You are given a choice between two levels of data content: basic and complete.
The second (complete) option is recommended by Microsoft, and is enabled by default. The ‘complete’ option includes the return of fairly sensitive information, such as a record of websites you have visited using Microsoft browser(s), and statistics about the applications that you use.
Choosing the ‘basic’ setting reduces the data flow, but does not stop it altogether. Microsoft claims that a certain minimum of information is necessary to provide your PC with updates. However, for reasons given later, we think that argument is a weak excuse for mining your data whether you like it or not.

In principle, Windows 10 can limit the transmission of diagnostic data even further than in the basic setting, though this applies only to the Enterprise and Education versions of Windows 10 together with the Server variants. For these operating system versions, administrators can set the level of detail to 'Security', provided you have modified your system policies to change them from the defaults.
The route you must walk in the Group Policy Editor is ‘Computer Configuration / Administrative Templates / Windows Components / Data Collection and Preview Builds. There you will see the 'Allow Telemetry' option. Setting that value to 'Security' ensures that the Windows Update service does not talk to Microsoft servers, among other things. Windows only checks this setting if you indicate in the right places in the system configuration that the PC will receive its updates via a different route – for example via a local WSUS server or via the SCCM (System Center Configuration Manager).

Since Windows 10 version 1803 it has been possible to view the diagnostic data Windows sends to Microsoft via 'Diagnostic data and feedback'. To do this, you have to flip the switch at the heading 'View diagnostic data' to 'On'. Thereafter, Windows copies and records up to 1 GB of the telemetry data it transmits. You can view this data via 'Open the diagnostic data viewer'. The first time you click on it, you will be redirected to the Microsoft Store where you can download the free 'Diagnostic Data Viewer' application. However, the tool does not give you a good overview of the data. In particular, if you have the settings on 'Full' you will soon get swamped, lost in an ocean of data.

WINDOWS TELEMETRY INTERNALS

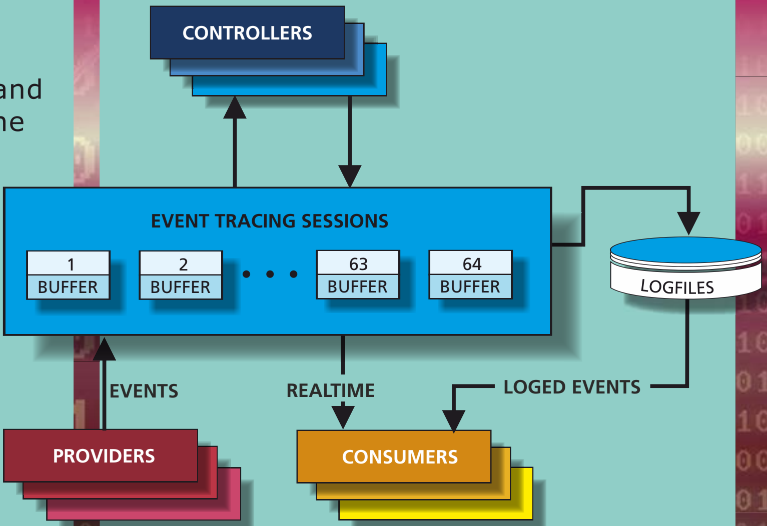
Windows telemetry relies on a built-in part of Windows named Event Tracing for Windows (ETW for short) for functions such as collecting, editing and forwarding diagnostic data about your computer and its use.
The BSI study’s authors have identified a central building block of Windows telemetry services, internally named DiagTrack, which is referred to as 'Connected User Experiences and Telemetry Service' in the options available in Windows settings. This service runs in a svchost.exe session, and its associated binary is diagtrack.dll located in the System32 folder.

EVENT TRACING FOR WINDOWS (ETW)

Trace events contain an event header and provider-defined data that describes the current state of an application or operation. You can use trace events to debug an application and perform capacity and performance analysis.
The ETW architecture has been designed with a Controller and Providers and Consumers (see the diagram below). ETW gives application programmers the ability to start and stop Windows event tracing sessions, and allow their applications to provide, instrument and consume trace events. The following information is about user-mode applications' use of ETW. For information about instrumenting device drivers that run in kernel mode, see “WPP Software Tracing and Adding Event Tracing to Kernel-Mode Drivers” in the Windows Driver Kit (WDK).

The ETW Consumer listens to two sessions: Diagtrack-Listener and Autologger-Diagtrack-Listener.
The Autologger session starts early in the boot process and writes its events in log files in the `%ProgramData% \ Microsoft \ Diagnosis \ ETLLogs` folder. If the DiagTrack service is started later during system start-up, the Autologger session is stopped, and DiagTrack starts its own session and captures trace events in real time. The service reads the auto-logged ETLLogs, edits them and then erases them.

The ETW infrastructure built in to Windows to record and process all kinds of events consists of four parts.
Providers are part of the operating system (they may be drivers or programs) which provide information in the form of events.
The controller activates and deactivates providers at the beginning and end of all ETW sessions. Each ETW session receives data from one or more provider, maintaining a buffer for interim event logging (event data can be stored in log files).



The illustration above depicts approximately how ETW has been designed.

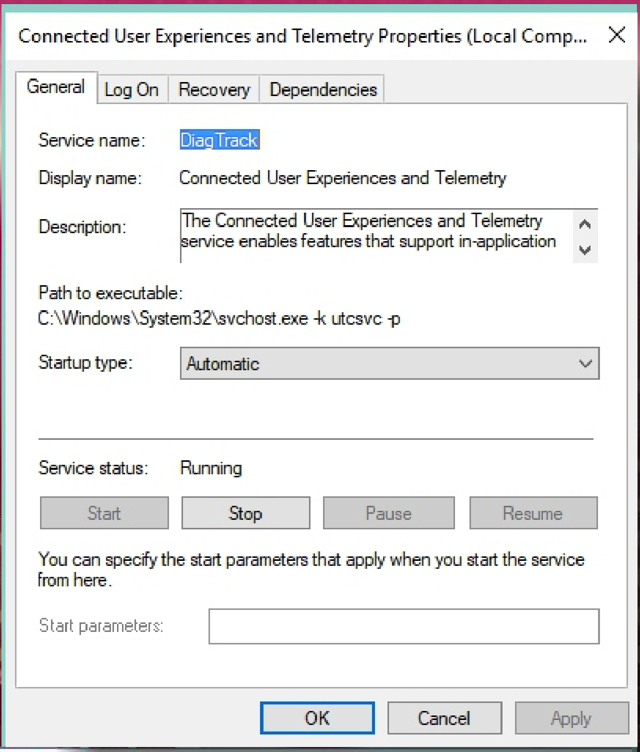
Consumers are programs that receive the data from one or more sessions in real time (or perhaps extract the data from log files) for further processing, perhaps to display the data or manipulate it in some other way.

The DiagTrack service is the operating system service designed to handle network traffic with Microsoft's servers, and so is the service that sends diagnostic data to Microsoft’s servers. The volume of transmitted data is determined by the Settings or Group Policy settings which indirectly control which ETW provider integrates the DiagTrack session. On their test system the BSI researchers counted 422 providers when the setting was ‘complete’, 410 providers when the setting was ‘basic’ and 4 providers when the setting was ‘Security’. However, those numbers are only a snapshot indication of typical differences between the settings.
The assignment of providers to telemetry settings is not fixed. In fact it depends on the contents of the `%ProgramData% \ Microsoft \ Diagnosis \ DownloadedSettings \ utc.app.json` file. This file is regularly replaced by Windows via automatic downloads. This means that the amount of data transmitted varies from PC to PC, even for PCs with identical operating systems and identical configurations.

NICE AND QUIET

To completely stop data traffic, you have to stop the DiagTrack service. The BSI experts' tests and my own experiments show that this is possible without your system becoming unstable, and updates continue to be installed correctly, whether that involves signature updates for Defender, normal patches, or major feature updates.

Once you switch off DiagTrack, the diagnostic data viewer no longer works as expected. If you later switch the service back on again, the viewer fails to show any data from the period when DiagTrack was inactive. We have not encountered any other side effects.



Connected User Experiences and Telemetry Properties (Local Comp...

Service name: DiagTrack

Display name: Connected User Experiences and Telemetry

Description: The Connected User Experiences and Telemetry service enables features that support in-application

Path to executable:
C:\Windows\System32\svchost.exe -k utcsvc -p

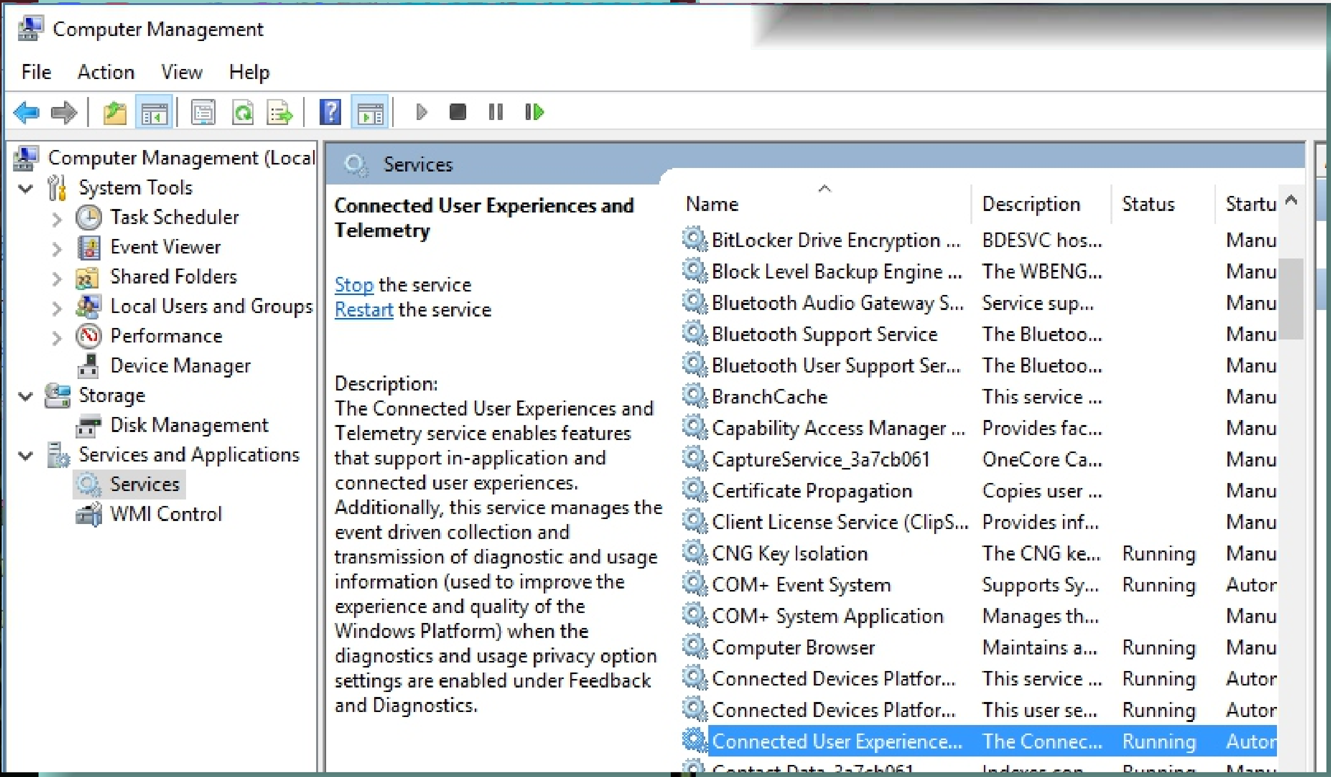
Startup type: Automatic

Service status: Running

You can specify the start parameters that apply when you start the service from here.

Start parameters:

STOPPING DIAGTRACK



Computer Management

Connected User Experiences and Telemetry

Stop the service
Restart the service

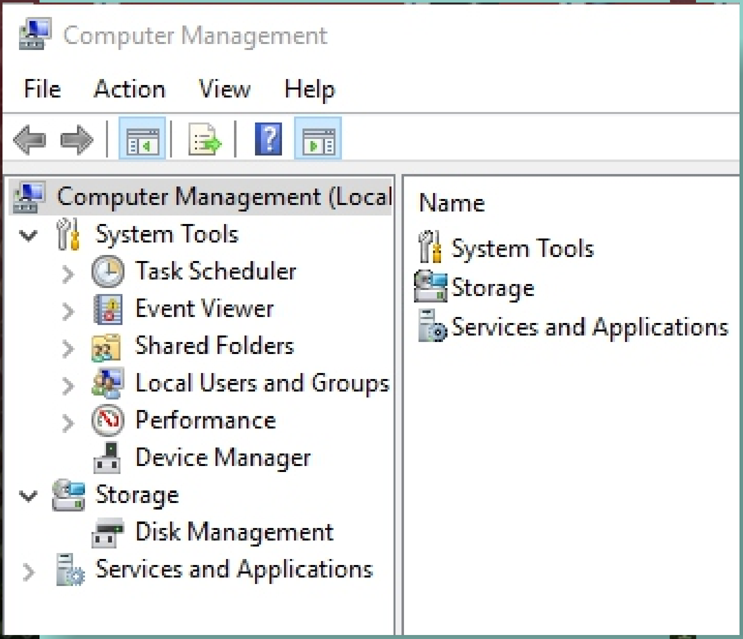
Description:
The Connected User Experiences and Telemetry service enables features that support in-application and connected user experiences. Additionally, this service manages the event driven collection and transmission of diagnostic and usage information (used to improve the experience and quality of the Windows Platform) when the diagnostics and usage privacy option settings are enabled under Feedback and Diagnostics.

Name	Description	Status	Startu
BitLocker Drive Encryption ...	BDESVC hos...		Manu
Block Level Backup Engine ...	The WBENG...		Manu
Bluetooth Audio Gateway S...	Service sup...		Manu
Bluetooth Support Service	The Bluetoo...		Manu
Bluetooth User Support Ser...	The Bluetoo...		Manu
BranchCache	This service ...		Manu
Capability Access Manager ...	Provides fac...		Manu
CaptureService_3a7cb061	OneCore Ca...		Manu
Certificate Propagation	Copies user ...		Manu
Client License Service (ClipS...	Provides inf...		Manu
CNG Key Isolation	The CNG ke...	Running	Manu
COM+ Event System	Supports Sy...	Running	Autor
COM+ System Application	Manages th...		Manu
Computer Browser	Maintains a...	Running	Manu
Connected Devices Platfor...	This service ...	Running	Autor
Connected Devices Platfor...	This user se...	Running	Autor
Connected User Experience...	The Connec...	Running	Autor
Contact Data_3a7cb061	Indexes con...	Running	Manu

You stop the DiagTrack service via Computer Management. The fastest way to get there is through the menu produced by the key combination Windows + X. Managing services is found under 'Service and applications / Services'.

Double-clicking 'Connected User Experience and Telemetry' opens the DiagTrack service's property page. To disable the service, click on 'Stop' and then choose 'Disabled' in the drop-down of Startup type.

The BSI experts also recommend that you prevent the start-up of ETW's 'Autologger-Diagtrack-Listener' session. You have to do this through editing a registry entry. Under the Registry's HKEY_LOCAL_MACHINE \ SYSTEM key, locate CurrentControlSet \ Control \ WMI \ Autologger \ Autologger-Diagtrack-Listener and give it the value 0. Our experience differs, and it has not been necessary to change a registry setting. We have not seen the file that that session creates on our test systems. Even if the file is created, only the DiagTrack service is able to make use of it, and with that service disabled it does nothing.



Neither disabling the DiagTrack service nor disabling the ETW session survived an upgrade of our test system from Windows 10 version 1803 to 1809. Clearly, following a function upgrade you have to make those privacy adjustments again.

ALTERNATIVES

In the BSI study you can find even more detailed advice on how to prevent transmission of telemetry data by adjusting network traffic settings. This includes, for example, making entries in the hosts file (% WinDir% \ System32 \ Drivers \ etc \ hosts) to redirect connections to the known servers of Microsoft's telemetry back-end to the invalid IP address 0.0.0.0.
You can achieve something similar by reconfiguring the firewall built into Windows (or your router's firewall). However, you would need to update those kinds of measures any time that Microsoft changes the names or IP addresses of their servers.

CONCLUSION

The bottom line is that if you want to prevent or restrict data transmission from your computer to Microsoft, there are known measures that are effective in at least two situations.

In environments where Enterprise versions of Windows 10 are running and which themselves have an infrastructure for distributing updates, converting the telemetry level to ‘Security’ is the option that you should consider. Administrators who do not find that measure sufficient can also turn off the Windows Update service and the cloud-based protection of Windows Defender.

In almost all other cases, switching off the DiagTrack service is a solution. This excludes a small group of users (such as Windows installations which Microsoft or another party itself maintains under contract). In these cases users have no rights to implement the changes outlined above.

BSI’s tests and our own experiments cannot rule out the possibility that in the longer term, unpleasant consequences may arise from tweaking the telemetry options. We don't have any experience of hot fixes that Microsoft sends only to PCs that are victims of exotic bug.
If your system must always be perfectly stable and you cannot take any risk, you would be better off protecting your privacy in other ways and leaving the DiagTrack service alone.

Microsoft's claim that telemetry data is necessary to be able to roll out updates does not appear to be true. Our Windows 10 installations with the DiagTrack service switched off have received the same updates as PCs that are otherwise identical except for the DiagTrack modifications.
It makes sense that Microsoft uses telemetry data to improve the stability of Windows. A little more transparency, and the certainty that information cannot be the telemetry data.

traced back to you as a user would certainly be appreciated by all users, and might encourage users to continue sending That would make juggling with all kinds of settings as described in this article completely unnecessary.

WINDOWS DEFENDER
DO YOU NEED ANOTHER ANTIVIRUS PROGRAM?

This article has been created with the help of several other suppliers Such As WikiPedia / Dutch CT Magazine / Microsoft / AV Comparatives. We thank them for their contribution

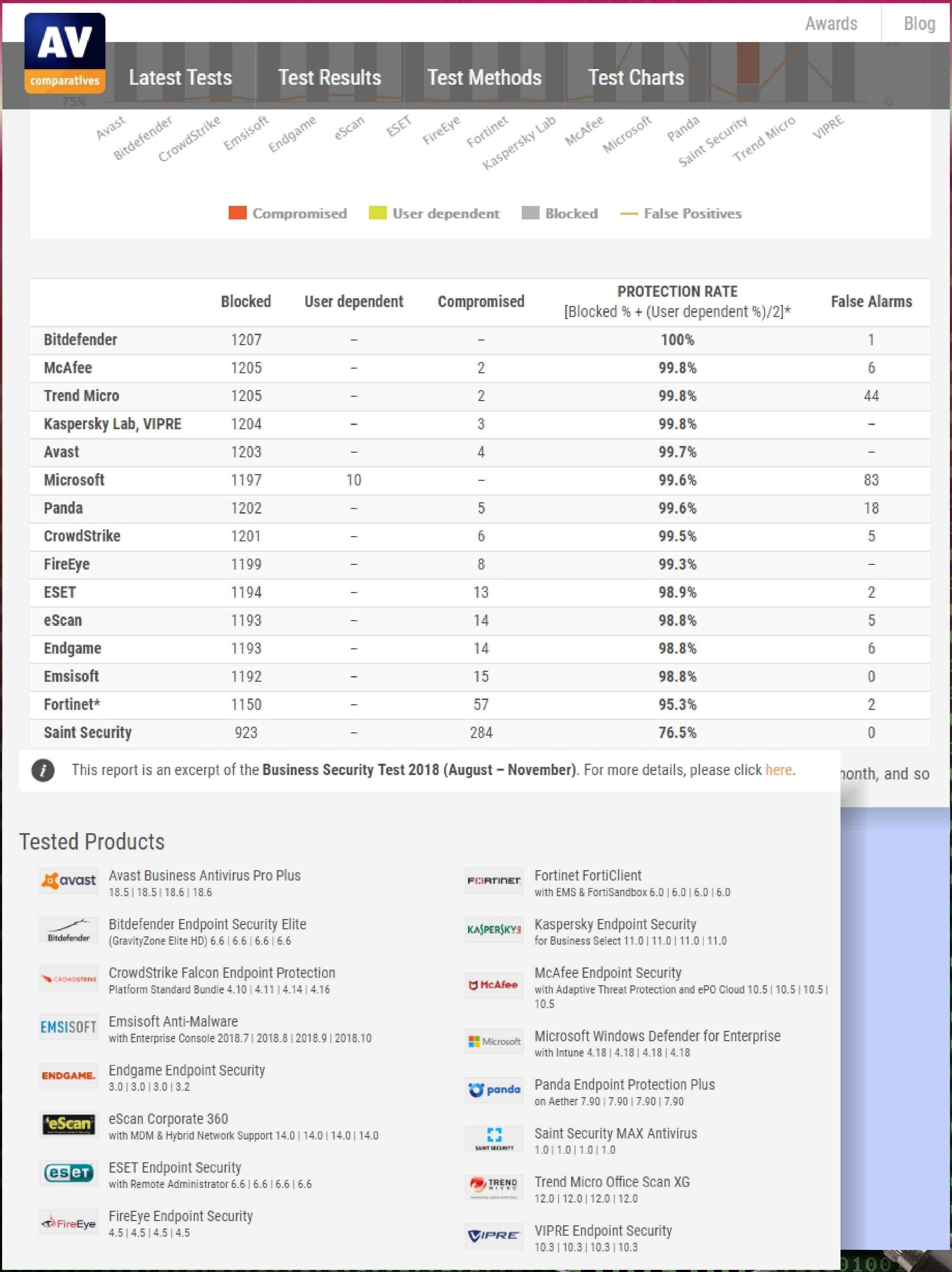
INTRODUCTION:
Windows users tend to have a natural resistance to proprietary antivirus programs, and for good reason: in the early days the protection they purported to offer was often very poor. Moreover, why would users pay for proprietary protection when there are numerous free antivirus programs for Windows, some good, some even excellent? After Windows' recent version upgrades, however, it is worth asking whether you need a separate Windows antivirus program at all. Because Windows Defender (which is installed by default with the operating system) has made huge catch-up strides since it first appeared, raising the question of whether installing a third-party antivirus program is still needed. This article examines the current Windows Defender (which since Windows 8 has been supplied free with the operating system). Could it be the most effective of all Windows antivirus programs?

In an internet age, an antivirus program is an absolute necessity on every Windows computer. Users like to pay for at least a sense of security. A glance at lists of the most popular online software sales quickly shows that antivirus functionality is a billion-dollar business, even though for many years free antivirus programs have been widely available.

The original fear that Microsoft's introduction of built-in antivirus functionality would lead to the demise of third-party antivirus vendors proved to be unfounded. Defender's virus recognition capabilities were too poor at the beginning for it to be a serious threat to established security programs. Microsoft Defender recognized only 60% of the malware it was fed in one 2015 virus scanner test, and further tests confirmed this poor showing; whereas the best performers in those tests typically blocked about 98% of the infection attempts. No software gave a 100% result. However the best performers were pretty good at what they were advertised to be capable of, bearing in mind that the tests were based on intentionally offered malware. Both existing antivirus producers and the cyber-criminals manufacturing the malware were scornful about Windows Defender.

The days of dismissing Windows Defender are long gone. Within six months of those early test comparisons Windows Defender had undergone extraordinary improvement. AV-Test and AV-Comparatives (two independent test institutes) had become specialists in exhaustive testing of antivirus programs.

The days of dismissing Windows Defender are long gone. Within six months of those early test comparisons Windows Defender had undergone extraordinary improvement. AV-Test and AV-Comparatives (two independent test institutes) had become specialists in exhaustive testing of antivirus programs.
An AV-Test assessment at the beginning of 2015 gave Windows Defender zero points. Six months later its protection factor had increased to three of the six possible points. Subsequently it has improved enormously. In 2018 the Windows antivirus program reached AV-Test's full score for the first time, and since then Defender has been at the top of all the listings. Defender also regularly shows up in AV-Comparatives test results. That is reason enough to take a good look at the situation in the wider antivirus market. In my research I also sought the views of various developer colleagues – people who don't like to have AV programs installed. However, Defender is Windows-specific and therefore very well integrated. They said sometimes Defender reports a false positive – but then you can do something constructive about that by passing on the result to Microsoft so they can take appropriate action.



	Blocked	User dependent	Compromised	PROTECTION RATE [Blocked % + (User dependent %)/2]*	False Alarms
Bitdefender	1207	–	–	100%	1
McAfee	1205	–	2	99.8%	6
Trend Micro	1205	–	2	99.8%	44
Kaspersky Lab, VIPRE	1204	–	3	99.8%	–
Avast	1203	–	4	99.7%	–
Microsoft	1197	10	–	99.6%	83
Panda	1202	–	5	99.6%	18
CrowdStrike	1201	–	6	99.5%	5
FireEye	1199	–	8	99.3%	–
ESET	1194	–	13	98.9%	2
eScan	1193	–	14	98.8%	5
Endgame	1193	–	14	98.8%	6
Emsisoft	1192	–	15	98.8%	0
Fortinet*	1150	–	57	95.3%	2
Saint Security	923	–	284	76.5%	0

ℹ This report is an excerpt of the **Business Security Test 2018 (August – November)**. For more details, please click here.

nonth, and so

Tested Products

Avast Business Antivirus Pro Plus
18.5 | 18.5 | 18.6 | 18.6

Bitdefender Endpoint Security Elite
(GravityZone Elite HD) 6.6 | 6.6 | 6.6 | 6.6

CrowdStrike Falcon Endpoint Protection
Platform Standard Bundle 4.10 | 4.11 | 4.14 | 4.16

Emsisoft Anti-Malware
with Enterprise Console 2018.7 | 2018.8 | 2018.9 | 2018.10

Endgame Endpoint Security
3.0 | 3.0 | 3.0 | 3.2

eScan Corporate 360
with MDM & Hybrid Network Support 14.0 | 14.0 | 14.0 | 14.0

ESET Endpoint Security
with Remote Administrator 6.6 | 6.6 | 6.6 | 6.6

FireEye Endpoint Security
4.5 | 4.5 | 4.5 | 4.5

Fortinet FortiClient
with EMS & FortiSandbox 6.0 | 6.0 | 6.0 | 6.0

Kaspersky Endpoint Security
for Business Select 11.0 | 11.0 | 11.0 | 11.0

McAfee Endpoint Security
with Adaptive Threat Protection and ePO Cloud 10.5 | 10.5 | 10.5 | 10.5

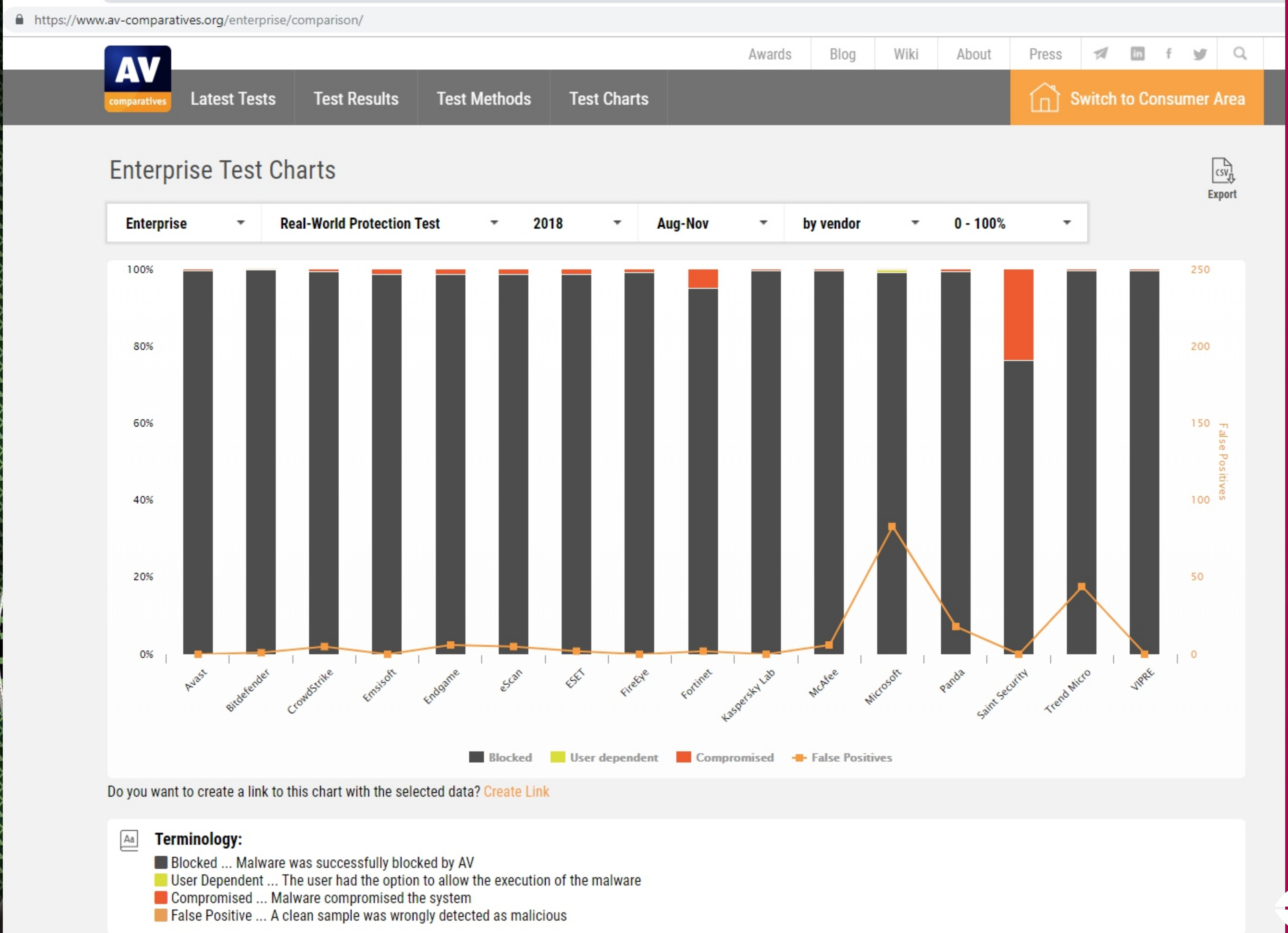
Microsoft Windows Defender for Enterprise
with Intune 4.18 | 4.18 | 4.18 | 4.18

Panda Endpoint Protection Plus
on Aether 7.90 | 7.90 | 7.90 | 7.90

Saint Security MAX Antivirus
1.0 | 1.0 | 1.0 | 1.0

Trend Micro Office Scan XG
12.0 | 12.0 | 12.0 | 12.0

VIPRE Endpoint Security
10.3 | 10.3 | 10.3 | 10.3



Enterprise Test Charts

Enterprise | Real-World Protection Test | 2018 | Aug-Nov | by vendor | 0 - 100%

Do you want to create a link to this chart with the selected data? Create Link

Terminology:

- ■ Blocked ... Malware was successfully blocked by AV
- 🟨 User Dependent ... The user had the option to allow the execution of the malware
- 🟥 Compromised ... Malware compromised the system
- 🟧 False Positive ... A clean sample was wrongly detected as malicious

https://www.av-comparatives.org/enterprise/

https://www.av-comparatives.org/comparison/?usertype=enterprise&chart_chart=chart2&chart_year=2018&chart_month=Aug-Nov&chart_sort=0&chart_zoom=0

THE PHOENIX RISING FROM THE ASHES

Defender's evolutionary advance is no coincidence. In its security blog Microsoft has stated that behind the scenes its security software has been completely changed, and that a large part of the improvement is due to artificial intelligence (AI) and machine learning (ML) (**https://www.microsoft.com/security/blog/**). On the basis of file properties such as the file's metadata, Defender tries to estimate how great a risk each scanned file poses. For a high risk file, virus protection performs an intensive analysis and might, for example, execute the file in a sandbox to discover the file's purpose more clearly.

*T*he 24th edition of the Microsoft Security Intelligence Report (SIR) is now available. And this year, I'm thrilled to share that not only can you download the PDF https://aka.ms/SIRv24Report, but you can also visit an online, interactive version https://www.microsoft.com/securityinsights that provides tools to filter and deep dive into the data.

This edition of the report is a reflection on last year's security events and includes an overview of the security landscape, lessons learned from the field, and recommended best practices.
I know you may find some of the trends, such as the increase in cryptocurrency mining and supply chain activity, worrisome. But I also hope you're encouraged to learn that the defensive techniques we've taken as a security community are paying off: there is good evidence that bad actors have been forced to change their tactics.

To create this report, the SIR team culled core insights and key trends out of a year's worth of data from multiple, diverse sources.
We analysed the 6.5 trillion security signals that go through the Microsoft cloud every day. We gathered insights from thousands of security researchers based around the world, and we learned lessons from real-world experiences, like the Ursnif campaign and the Dofoil coin-miner outbreak.

There is a lot going on, but the SIR team distilled the data down into four key trends:

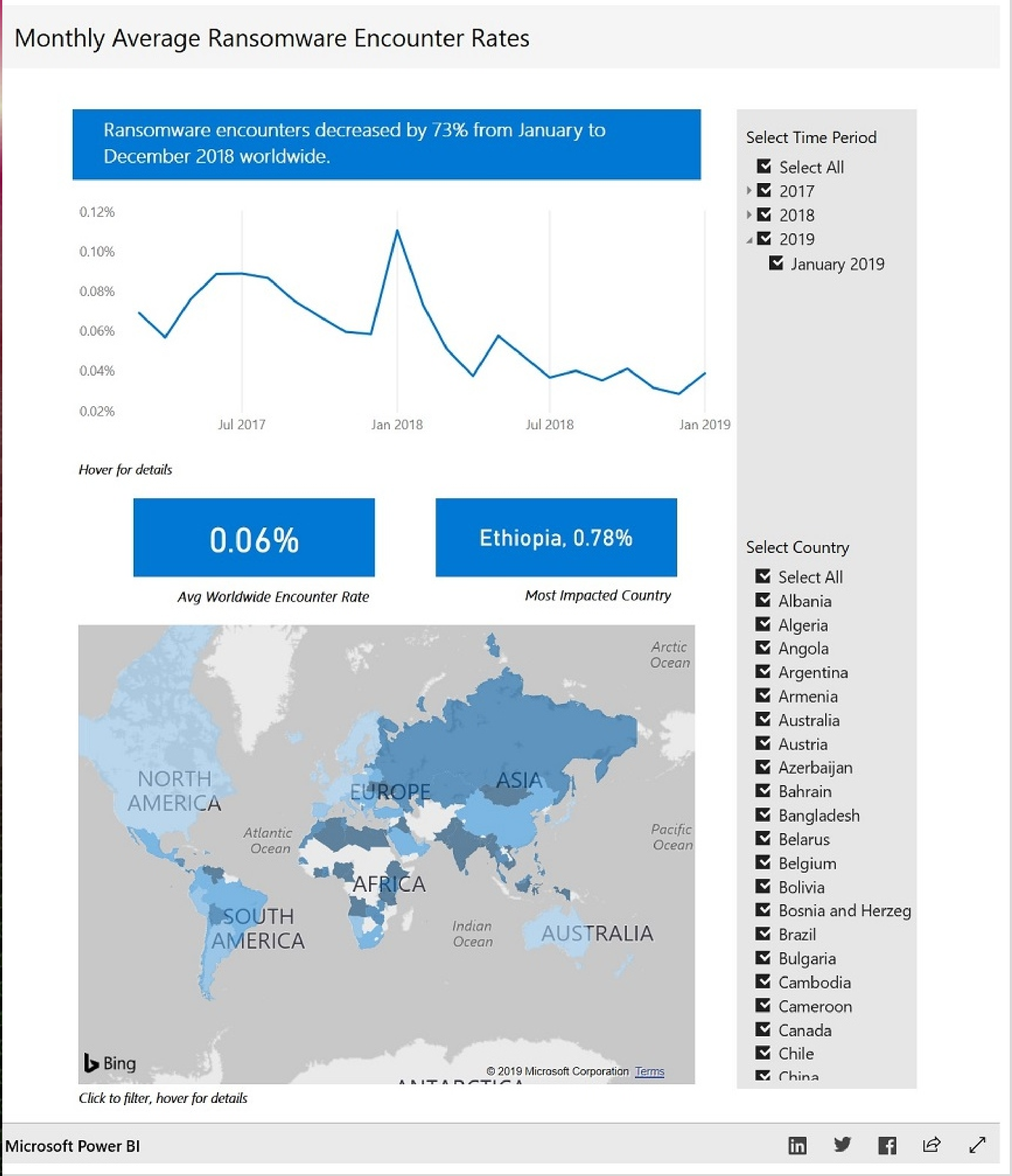
- Ransomware attacks are on the decline.
- Cryptocurrency mining is prevalent.
- Software supply chains are at risk.
- Phishing remains a preferred attack method.

RANSOMWARE ATTACKS ARE ON THE DECLINE

The decline of ransomware attacks shown in the 2018 data is a great example of how the security community is forcing bad actors to retreat. Just last year, we highlighted the large threat that ransomware played in the 2017 data, so this decline is notable. We believe that attackers have shifted from the highly visible phishing approach to stealthier methods because users have become smarter in their responses to attacks.

CRYPTOCURRENCY MINING IS PREVALENT

The decline in ransomware is good news. However, on the flip side we see continuing widespread cryptocurrency mining. This is one of the methods attackers have substituted for ransomware. Mining coins profitably requires an immense amount of computing power to perform complex calculations, so attackers install malware on users' computers to "steal" the necessary computing power. The SIR report provides a comprehensive overview of how cryptocurrency works and other factors driving this trend upwards.



Monthly Average Ransomware Encounter Rates

Ransomware encounters decreased by 73% from January to December 2018 worldwide.

Hover for details

0.06%

Avg Worldwide Encounter Rate

Ethiopia, 0.78%

Most Impacted Country

Click to filter, hover for details

Select Time Period

- ☑ Select All
- ☑ 2017
- ☑ 2018
- ☑ 2019
  - ☑ January 2019

Select Country

- ☑ Select All
- ☑ Albania
- ☑ Algeria
- ☑ Angola
- ☑ Argentina
- ☑ Armenia
- ☑ Australia
- ☑ Austria
- ☑ Azerbaijan
- ☑ Bahrain
- ☑ Bangladesh
- ☑ Belarus
- ☑ Belgium
- ☑ Bolivia
- ☑ Bosnia and Herzeg
- ☑ Brazil
- ☑ Bulgaria
- ☑ Cambodia
- ☑ Cameroon
- ☑ Canada
- ☑ Chile
- ☑ China

Microsoft Power BI

SOFTWARE SUPPLY CHAINS ARE AT RISK

An increase in attacks on software supply chains is a further trend Microsoft has been noted in recent years. One supply chain tactic attackers use is to incorporate a compromised component into a legitimate application or update package, which is then distributed to the users via the software. These attacks can be very difficult to detect because they take advantage of the trust that users have in their software vendors. The report includes several examples, including the Dofoil campaign, which illustrates how wide-reaching these types of attacks are and what we are doing to prevent and respond to them.

PHISHING REMAINS THE PREFERRED METHOD OF ATTACK

It’s unsurprising that phishing continues to be a popular method of attack, and we expect that to continue for the foreseeable future. The good news: much like with ransomware, bad actors have shifted tactics in response to the increasingly sophisticated tools and techniques deployed to protect users. We exposed many details of these new phishing methods that we hope you find useful in your efforts to defend against them.

LEARNING MORE

When I was a practitioner, I sought out reports like these https://www.microsoft.com/sir to help me better understand attacker techniques and plan my defences accordingly. I hope you find the insights, tips, and best practices that we’ve pulled together just as helpful. Download volume 24 of the Microsoft Security Intelligence Report **https://www.microsoft.com/sir** and then dig into the data specific to your region in the interactive website **https://www.microsoft.com/securityinsights.** The site is updated monthly, so you can keep up with emerging data and insights throughout the year.

The SIR serves to share some of the intelligence and insights that Microsoft generates as part of our broader security operations work, but it is not the whole story. Please also make sure to check out the latest announcements about Microsoft security innovations (https://blogs.microsoft.com/?p=52558214) aimed at helping defenders capitalize on updated security intelligence and protections, to help you stay ahead in the ever-changing cybersecurity landscape.

ARTIFICIAL INTELLIGENCE VERSUS REAL WORLD THREATS

AI allows previously unknown malware to be tested using the available calculation capacities as efficiently as possible. Microsoft has a detailed blog documenting the ways in which artificial intelligence (AI) and machine learning (ML) have contributed to stopping malware sources such as Emotet. Not that Microsoft invented the use of specially trained ML models in the hunt for viruses, since traditional antivirus producers have for a long time used ML to try to curb the flood of malware.

Emotet is a banking trojan which obtains financial information by injecting computer code into the networking stack of an infected Microsoft Windows computer, allowing sensitive data to be stolen via transmission. Emotet malware also inserts itself into software modules which are then able to steal address book data and perform denial of service attacks on other systems. It also functions as a downloader or dropper of other banking Trojans.
Emotet has evolved in its delivery, however the most prominent form has been inserting malicious documents or URL links inside the body of an email, sometimes disguised as an invoice or PDF attachment.
First reported in Germany, Austria, and Switzerland in 2014, Emotet was seen shortly thereafter in the United States where rather than entry via fake invoices it arrived via malicious JavaScript .JS files. When the malicious .JS files are executed, Emotet malware infects the current host.

Once Emotet infects a host, a malicious file that is part of the malware is able to intercept, log, and save outgoing network traffic via a web browser, allowing sensitive data about how to access the victim's bank account(s) being leaked.

EMOTET is a member of the Feodo family of trojan malware. When run in a virtual machine environment, Emotet changes its behaviour in ways that are intended to mislead malware investigators.
`https://www.microsoft.com/en-us/wdsi/threats`

MACHINE LEARNING (ML)
is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering, and computer vision, where it is not feasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

However, the AI is used differently depending on the producer. Sometimes simple AI models run locally on a client needing protection, with more complex operations performed in the producer's cloud. Logically, the different protection methods are used in sequence: from simple and fast on to more complex and slower. Only when a particular phase fails to give an unambiguous answer is the next slower phase started.

The proven recognition methods such as comparison with virus signatures, behavioural analysis and heuristics are also still used. Signature-based recognition has long been the benchmark for the protection that an antivirus program could offer. The program also has a collection of thousands of existing viruses that have already been mapped. The program's task is then to recognize as many of them as possible. The results nowadays no longer have much meaning. Inside the antivirus industry there is good cooperation, so it doesn't take long for a newly identified virus to appear in the virus databases of all major producers. 100% virus recognition is therefore the rule now, rather than the exception. However, recognising virus signatures loses its importance precisely because the chance of finding an already known virus file is high. This is because the cyber criminals, naturally, have access to all virus scanners. They manipulate their malware before its release until there are no antivirus programs (or only very few) that recognize the new malware.

In the simplest case, they use one of the numerous exe packers for this, which saves them having to change the actual malicious code. Thus every potential victim can receive an individual copy of the malware.

WATERSHED - THE REAL WORLD

Thus the so-called real-world test has become the most important indicator of the effectiveness of an antivirus program's protection. Real-world testing means that the antivirus program is confronted with real, recent threats, such as infected websites and e-mails with viruses. The testers check whether the system is infected and whether the antivirus software has successfully blocked the attack. It does not matter which security mechanism prevented the attack. The result of a real-world test is of course just a snapshot of capabilities measured against threats existing at the time of the snapshot.

Additionally, the test laboratories monitor the antivirus software's effect on the speed of the system, and how often false alarms are raised (false positives). These are the cases where the antivirus program classifies a harmless file or website as harmful.

All listed antivirus programs, including Windows Defender, have succeeded in passing the real-world tests of AV-Test and AV-Comparatives.
(see the section on page 32)
Depending on the period you are watching, sometimes one and sometimes another program is marginally better in the scoring. However, usually the infection rate after 200 to 300 attack attempts is no higher than 1%, rarely higher than that.

However, there are major differences between antivirus vendors in terms of the additional capabilities their product offers. Antivirus software producers try to make their products more appealing to potential customers by offering various kinds of extra functionality. Optional features might include a shielded online banking browser, a password manager, a file rescuer or a system tuner. These functions actually have nothing to do with virus protection. Most manufacturers also offer multiple versions of their security software: the more expensive, the more additional functions you get. In many cases you also receive technical support as a paying customer. This not only helps with questions about the antivirus program itself, but also gives practical help to assist in disinfecting an infected system.

FREE ANTIVIRUS

In addition to their paid-for program packages, many producers offer a completely free version of their virus hunter which contains the same antivirus engine as the paid version, but with most extra functions unrelated to virus control deactivated. The free program uses every opportunity to advertise the paid version, nagging the user to buy the extra features' functionality. If you are going to replace Defender with one of the free antivirus programs, you will have to accept those advertisements.
So you have quite a few options. You can stick with Windows Defender alone; you can use one of the paid antivirus programs, or you can install a free version with advertising.
Defender may be the right choice for many users. Not only because it is already installed, but also because it largely works unseen in the background, and because it is free.

A paid antivirus program is an option if you do not want to depend on Microsoft's protection, if you are dependent on ready access to technical support, or if you think you will benefit from the numerous extra functions. You should only use the free scanners from the antivirus producers if you do not find their intrusive advertising annoying. In all three cases you are actually well protected.

One argument against the use of Defender is the fact that it is so widespread, meaning that malware attackers have usually tried at least to get past the biggest obstacle. That is why there is clearly more malware targeting Windows than for all other operating systems combined.
The same goes for virus protection: a virus developer will first try to bypass Defender before he tries to get round less common antivirus programs.

The more popular Defender becomes, the stronger this effect will become.

The first malware to exploit a leak in an antivirus program on a large scale is also likely to affect users of Defender. Such attacks already exist in security researchers' laboratories.
For example, there are already discovered leaks in various well-known antivirus programs.
Because of these leaks, attackers can not only bypass virus protection, but even exploit the antivirus program itself to introduce infection.

SECURITY DEPENDS ON THE LEVEL OF TRUST

To ensure that an antivirus producer can offer optimum protection to their customers against current threats, they need data and more data. The data they need comes partly from their customers, which requires a certain relationship of trust.

To have to send more information to Microsoft than you already do may send shivers down your spine. But do you trust Bratislava, Moscow or Tokyo any more?

Some antivirus producers seek an advantage by using slogans such as 'IT security made in Germany' or something similar, and promise to take the minimum of data from your computer. To what extent you want the origin of an antivirus program to be taken into account is your choice. You must decide for yourself.

If your Windows computer becomes infected, you should no longer start the system, otherwise the virus may cause even more damage. In such a situation it is better to use one of the anti-virus producers' emergency systems.

A Linux-based live system is most often used. This boots directly from a DVD or a USB stick instead of Windows. An antivirus program installed in this way can then examine Windows from the ‘outside’ for possible malware. The virus signatures can be obtained on the spot from the internet, so that the antivirus program is working with the most current data. Scanning your system can take a long time, and if you want to play completely safe, you will use multiple live systems to have your system scanned with different antivirus engines. Because most scanners are high quality, the latter course of action is not usually required, though it is a more thorough belt-and-braces approach.
If infected files are found, and these files do not affect the operation of Windows, you can simply delete them and restart. If the infection affects Windows system files, then you might be able to replace them via a USB stick with a replacement file from another, non-infected Windows system.

A POWERFUL NEW TRIUMVIRATE
FOR WEB DEVELOPMENT

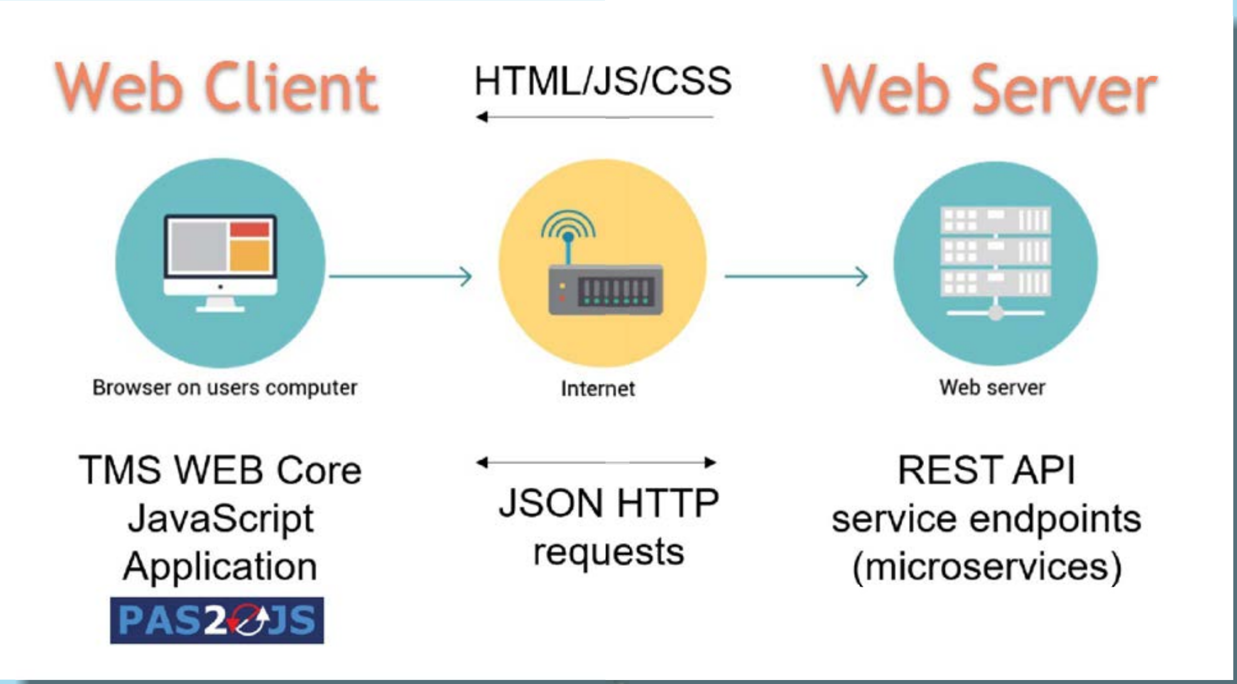
starter expert

Tripling Effort at Little Extra Cost

What are hundreds of hours of custom software engineering and development effort worth today? Unless it's highly specialized, uniquely targeted, and well-engineered, the answer is, "Not much." While that might sound like a glib retort, great software need not cost a fortune, particularly when you consider the plethora of inexpensive and/or free software frameworks and components that abound on the Internet today from both open-source and commercial venues.

In fact, as labours of love, some of the most excellent software tools and frameworks available today are free or nearly free to anyone wishing to expedite development efforts while honing developer skills.

In this regard, as this article will detail atop an existing IDE, two emerging technologies present a wonderful new and inexpensive opportunity for Object Pascal developers to retool and update their own abilities and services offerings. One is an ECMAScript(JavaScript)-to-ObjectPascal cross-compiler from the Free Pascal and Lazarus foundation (**https://foundation.freepascal.org/**) — the open-source community effort that brings us the Free Pascal Compiler as well as the Lazarus IDE (**https://www.lazarus-ide.org/**). The other, called "TMS WEB Core" from TMS Software, is both a Web component suite and framework that promises to provide the ultimate embodiment of encapsulated, simplified HTML5 programming for Object Pascal enthusiasts.



With HTML5 now mainstream in all modern Internet browsers and some form of Internet connectivity ubiquitous in so many of our modern smart appliances (often via browser technology), this confluence provides one of the greatest opportunities for software developers in this lifetime. To begin to see the implication and possibilities of this new common ground, one need only understand that HTML5 has equipped the pervasive hypertext markup language specification with many rich UI features (e.g., a canvas for drawing and vector graphics, video/multimedia support, local storage, et cetera) which will allow it to be the application container of choice around the world (so to speak, the world's first virtualized OS used by everyone, but controlled by no single commercial interest like the Internet itself).

Furthermore, key benefits to any developer targeting such cannot be overstated; these benefits include being a ubiquitous virtualized platform supporting all of today's most popular operating systems such as Windows, Android, macOS, iOS and Linux, as well as having a massive userbase which includes potentially all Internet-browsing citizens of Earth. As Bill once hoped would be the case for Windows as an explicitly commercial venture, such is no longer within reach for Microsoft (and neither should it be, nor need it be, for any single company or conglomerate — including Google, Apple, Facebook, Alibaba, Amazon — although the contributions from each to benefit such an open platform should be appreciated).

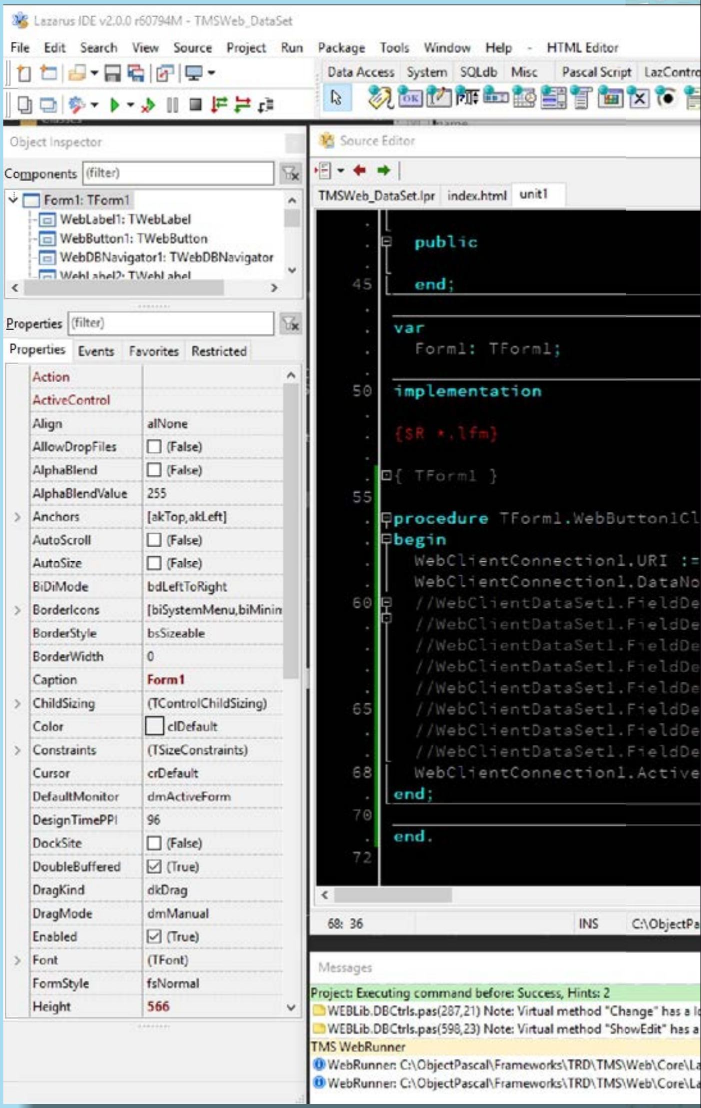
While, indeed, it is true that developing truly great software can be a time-consuming and, thus, an expensive venture, pricey development projects need not be the norm.

Firstly, in approaching such efforts, labour and financial burdens can be greatly lessened by avoiding bespoke creations in favour of well-regarded, framework-patterned, component-based solutions that employ open-standards along with a modicum of open-source content to minimize licensing and royalty costs.

Secondly, and perhaps equally important, all development efforts must be amortized over the greatest number of potential installations possible at the lowest cost of ownership sustainable within the given market (otherwise, someone else — either a new or existing competitor — eventually will steal your most profitable meal).

With these tenets in mind, this article presents a powerful triumvirate available today for Object Pascal developers wishing to partake affordably in this brave new world of HTML5-ready browsers and devices connecting to the Internet of Everything (IoE🙂). Hereunder, the three entities comprising this triumvirate are:

1. **Lazarus** — Delphi's free, open-source counterpart that avoids Delphi's commercial-use limitations and targets a bazillion platforms (depicted below),


Lazarus IDE v2.0.0 r60794M - TMSWeb_DataSet
File Edit Search View Source Project Run Package Tools Window Help - HTML Editor
Data Access System SQLdb Misc Pascal Script LazContro
Object Inspector
Components (filter)
Properties (filter)
Properties Events Favorites Restricted
Source Editor
TMSWeb_DataSet.lpr index.html unit1
68: 36 INS C:\ObjectPa
Messages

Form1: TForm1
WebLabel1: TWebLabel
WebButton1: TWebButton
WebDBNavigator1: TWebDBNavigator
WebLabel2: TWebLabel

Action	
ActiveControl	
Align	alNone
AllowDropFiles	□ (False)
AlphaBlend	□ (False)
AlphaBlendValue	255
Anchors	[akTop,akLeft]
AutoScroll	□ (False)
AutoSize	□ (False)
BiDiMode	bdLeftToRight
BorderIcons	[biSystemMenu,biMinin
BorderStyle	bsSizeable
BorderWidth	0
Caption	Form1
ChildSizing	(TControlChildSizing)
Color	□ clDefault
Constraints	(TSizeConstraints)
Cursor	crDefault
DefaultMonitor	dmActiveForm
DesignTimePPI	96
DockSite	□ (False)
DoubleBuffered	☑ (True)
DragKind	dkDrag
DragMode	dmManual
Enabled	☑ (True)
Font	(TFont)
FormStyle	fsNormal
Height	566

```
  public

  end;

var
  Form1: TForm1;

implementation

{$R *.lfm}

{ TForm1 }

procedure TForm1.WebButton1Cl
begin
  WebClientConnection1.URI :=
  WebClientConnection1.DataNo
  //WebClientDataSet1.FieldDe
  //WebClientDataSet1.FieldDe
  //WebClientDataSet1.FieldDe
  //WebClientDataSet1.FieldDe
  //WebClientDataSet1.FieldDe
  //WebClientDataSet1.FieldDe
  //WebClientDataSet1.FieldDe
  //WebClientDataSet1.FieldDe
  WebClientConnection1.Active
end;

end.
```

```
Project: Executing command before: Success, Hints: 2
WEBLib.DBCtrls.pas(287,21) Note: Virtual method "Change" has a lo
WEBLib.DBCtrls.pas(598,23) Note: Virtual method "ShowEdit" has a
TMS WebRunner
WebRunner: C:\ObjectPascal\Frameworks\TRD\TMS\Web\Core\La
WebRunner: C:\ObjectPascal\Frameworks\TRD\TMS\Web\Core\La
```

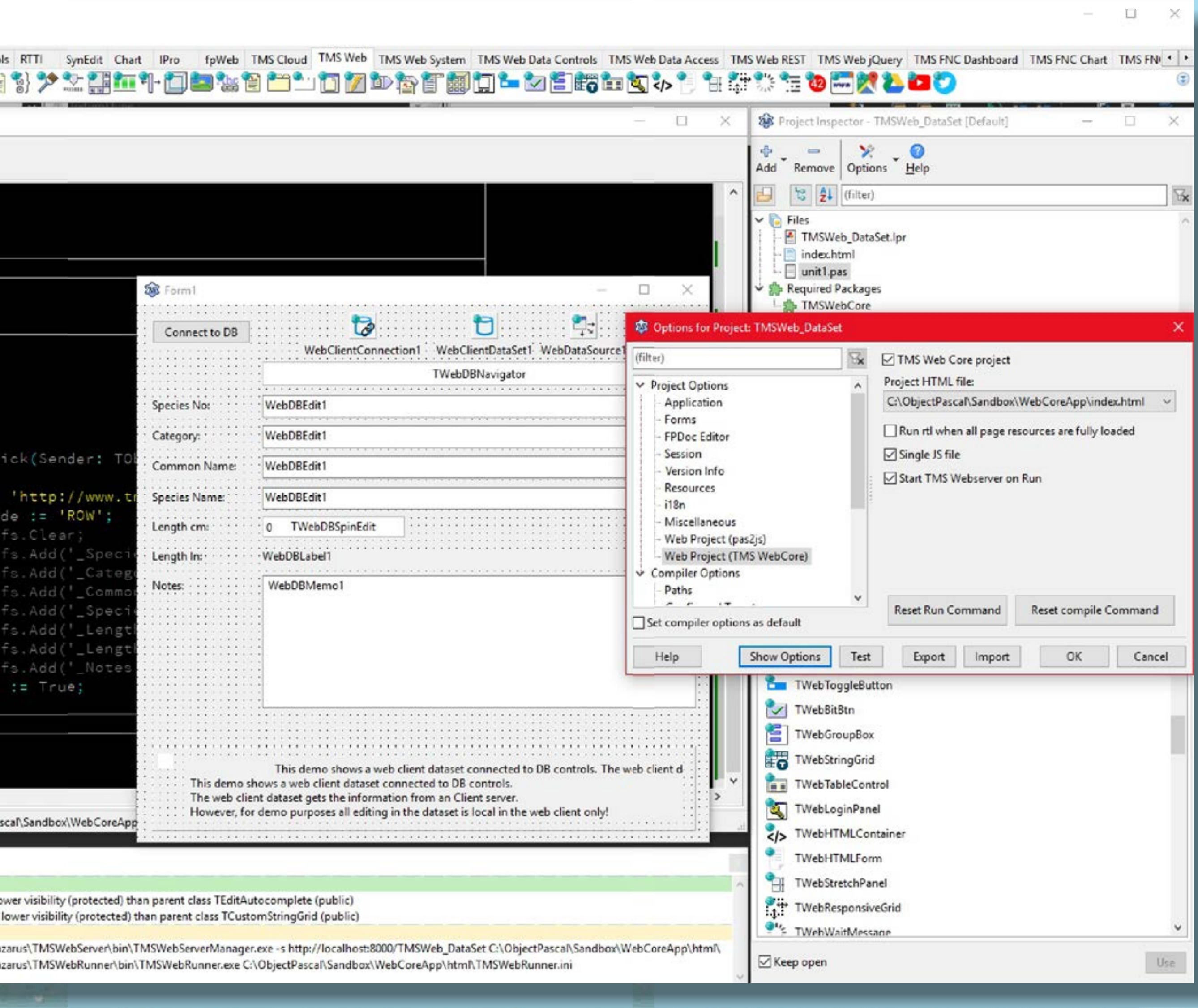
2. **Pas2JS** — an open-source Object Pascal to ECMAScript transpiler enabling Web programming with near Delphi 10 language support (including anonymous methods, advanced records, RTTI, and even attributes), and

3. **TMS WEB Core** — an inexpensive commercial Web component framework patterned upon Delphi's long-standing and highly-successful Visual Component Library.

In subsequent issues of Blaise Pascal Magazine and, very soon, at **ObjectPascal.org**, we will explore the configuration and use of Pas2JS as well as TMS WEB Core in a variety of applications. To kick things off in the next issue of this periodical, we will present FPC/Lazarus' new server-side REST Bridge components designed to ease exposing a datastore both RESTfully and securely on the Web. Thereafter, we will compare and contrast similar RESTful solutions from TMS Software based upon their Aurelius and XData component suites as well as similar offerings from other vendors.

FRAMEWORKS AND COMPONENTS New Isn't Necessarily Better

While good frameworks (not the bad ones that Gary Wisniewski discusses in an excellent post entitled "Welcome to Framework Hell" on his blog at **https://medium.com/@garywiz**) encapsulate patterns upon which to scaffold robust software solutions, components can take developers further by offering a means of



```
ick(Sender: TO
 'http://www.t
de := 'ROW';
fs.Clear;
fs.Add('_Speci
fs.Add('_Categ
fs.Add('_Commo
fs.Add('_Speci
fs.Add('_Lengt
fs.Add('_Lengt
fs.Add('_Notes
 := True;
```

Species No: WebDBEdit1
Category: WebDBEdit1
Common Name: WebDBEdit1
Species Name: WebDBEdit1
Length cm: 0 TWebDBSpinEdit
Length In: WebDBLabel1
Notes: WebDBMemo1

This demo shows a web client dataset connected to DB controls. The web client d
This demo shows a web client dataset connected to DB controls.
The web client dataset gets the information from an Client server.
However, for demo purposes all editing in the dataset is local in the web client only!

Options for Project: TMSWeb_DataSet

- Project Options
  - Application
  - Forms
  - FPDoc Editor
  - Session
  - Version Info
  - Resources
  - i18n
  - Miscellaneous
  - Web Project (pas2js)
  - Web Project (TMS WebCore)
- Compiler Options
  - Paths

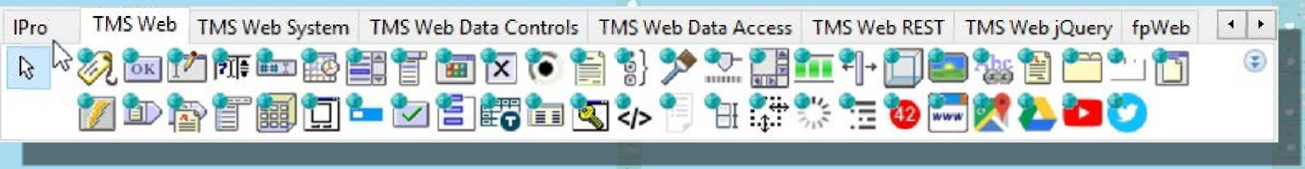
☑ TMS Web Core project
Project HTML file: C:\ObjectPascal\Sandbox\WebCoreApp\index.html
□ Run rtl when all page resources are fully loaded
☑ Single JS file
☑ Start TMS Webserver on Run
□ Set compiler options as default

```
scal\Sandbox\WebCoreApp
ower visibility (protected) than parent class TEditAutocomplete (public)
 lower visibility (protected) than parent class TCustomStringGrid (public)
zarus\TMSWebServer\bin\TMSWebServerManager.exe -s http://localhost:8000/TMSWeb_DataSet C:\ObjectPascal\Sandbox\WebCoreApp\html\
zarus\TMSWebRunner\bin\TMSWebRunner.exe C:\ObjectPascal\Sandbox\WebCoreApp\html\TMSWebRunner.ini
```

breaking up the complexity of a software application into manageable parts.

Great components, in turn, hide the complexity of their underlying implementation behind an interface.

A component framework, ultimately, combines intuitive software patterning of a framework with well-thought, consistent interface abstractions to provide interchangeable parts and sub-systems able to serve broad application use. Any decent framework or component set will serve these purposes without encumbering developer productivity with a complex API or impacting application performance due to structural inadequacies or coding inefficiencies.



It is a testament to Delphi's original Borland developers that, as tools and libraries go for developing on the Windows OS, the Visual Component Library has truly stood the test of time (24 years and counting) as a well-designed, easy-to-use object-oriented application framework and component set.

The Visual Component Library — or VCL as its known in Delphi and LCL as its known in Lazarus — offers a plethora of UI widgets, OS messaging and service encapsulations, and a bidirectional view-model data-binding mechanism to enable data changed in an application's UI to seamlessly update its underlying datastore (as permitted by business logic), and for changes that occur within the underlying datastore's contents (e.g., due to other users) to propagate appropriately to the application's UI.

For such data presentation and management purposes, the VCL and LCL both include excellent datastore abstractions provided via RESTful services interface (for both client and server implementation) as well as an intuitive data access layer consisting of TDataSet, TDataSource and TDataModule along with several connection mechanisms that support a wide variety of database servers and cloud service providers.

With the VCL's long-standing success for Windows development, and LCL for cross-platform development (as LCL works for many operating systems [unlike its Windows-only peer]), it is no wonder that TMS Software, an equally long-standing Belgium-based component maker for Windows, macOS, iOS, Android and Linux, chose the VCL/LCL as the basis for their new Web component framework, TMS WEB Core.

TMS WEB CORE
A Framework of Web Components

TMS WEB Core is a component framework designed for rapid application development of HTML (HTML5) clients using Object Pascal following the single-page architecture employed by other modern frameworks like Angular, React and vue.js. Object Pascal, in turn, is a mathematically grounded and syntactically simple language with a rich heritage underpinning the World's first highly productive, object-oriented IDE (Delphi/RAD Studio for Windows OS) and inspired several cross-platform compiler and dialect implementations (Free Pascal, Oxygene, Pascal Script, DWScript), as well as its current-day language siblings, C# and TypeScript.

If you've been a comfortable Delphi developer previously, you'll feel right at home working with TMS WEB Core for Lazarus in its very Delphi-like IDE pictured above. Developing in the Lazarus IDE using TMS WEB Core components, not only will you enjoy the wonderful Object Pascal language embodied there through the highly performant Lazarus code editor (which includes a detachable form designer encouraging clean separation of presentation and application logic), TMS WEB Core offers many drop-in web-based counterparts for common VCL/LCL controls (e.g., label, edit, button, listbox, grid, dataset, datasource, datamodule) as well as many more sophisticated UI components (particularly if you add in TMS' FNC suite) that can save you literally hundreds of hours of custom design effort. For more information about TMS's specific component offerings, visit **`tmssoftware.com/site/tmswebcore.asp`** and **`tmssoftware.com/site/blog.asp?post=476`**.

TMS WEB Core Packages

As a modern component framework for Web development, TMS WEB Core for Lazarus consists of:

❶ Visual and Non-Visual Web Component RTL (`TMSWEBCore.lpk`)

❷ Object Pascal to ECMAScript/JavaScript compiler and RTL (`pas2js.exe` and `pas2js_rtl.lpk`)

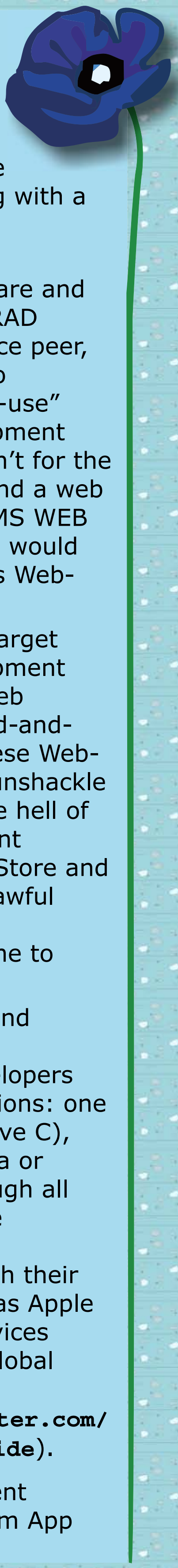
❸ Lazarus IDE Integration (`TMSWEBCorePkgDeLaz.lpk`)

❹ A single installation package (`TMSWEBCorePkgLibLaz.lpk`) which, upon installing alone, automatically integrates the three prior packages under Lazarus.

To see various use cases for TMS WEB Core and demonstration of its many visual controls as well as non-visual data access components, visit the TMS WEB Core demonstration page at **`tmssoftware.com/site/tmswebcoreintro.asp`**. For a sample project build with TMS WEB Core on Lazarus, including an overview of the "simple" coding process, see the post **"TMS Lab Day 4: TMS WEB Core from Lazarus"** accessible at **`tmssoftware.com/site/blog.asp?post=489`**.

What's Motivatiog this Triumvirate?

The delivery of native applications across platforms (from desktop PCs and MACs, to Linux workstations and servers, onto mobile devices, and into IoT nodes) requires a deep understanding of a variety of hardware abstractions (for both CPU and GPU architectures), server data communications standards (Ethernet|TCP/IP, BLE, USB, HTTP, et cetera) as well as the nuances of running atop several disparate operating systems (Windows, macOS, Linux, iOS, Android). Complicating matters, such cross-platform delivery mandates a cadre of platform-specific compilers along with a myriad of directives, switches, and deployment options.



This "tyranny of choice" in hardware and software platforms is what both RAD Studio's Delphi and its open-source peer, Lazarus, find themselves trying to rationalize today into an "easy-to-use" cross-platform integrated development (IDE). Try as they may, if it weren't for the open-source transpiler, Pas2JS, and a web component framework such as TMS WEB Core, the prospects for both IDEs would fade quickly in the daylight of this Web-centric world.

By offering the unique ability to target standards-compliant Web development (including building Progressive Web Applications) from within the tried-and-true Delphi and Lazarus IDEs, these Web-enabling newcomers promise to unshackle Object Pascal developers from the hell of compiler optioning and deployment configuration (including iOS App Store and Google Play) to transcend this godawful state of cross-platform application development we've unwittingly come to know.

For example, to support both iOS and Android until the recent advent of Progressive Web Applications, developers would create two separate applications: one built for iOS using Swift (or Objective C), and one built for Android using Java or Kotlin. They would then jump through all hoops mandated by their app-store gatekeepers, Apple and Google, to hopefully|eventually be blessed with their applications' release to humanity (as Apple iOS and Google Android mobile devices currently represent 97.7% of the global smartphone market according to GlobalStats: **`http://gs.statcounter.com/os-market-share/mobile/worldwide`**).

As Embarcadero noted in their recent whitepaper, "Smarter Cross-Platform App

Development: Creating Native Apps with a Single Codebase" (available here: **http://forms.embarcadero.com/VisualAppDevelopment**):

Because native apps are expensive and difficult to develop, and Web apps provide an inferior user experience, hybrid apps have gained popularity. Hybrid apps mimic native apps by running in a "WebView container." At startup, the app opens a browser window to a predefined address. Then the app, created with HTML5, CSS3, and JavaScript, is displayed in this WebView. Developers can use several frameworks to create hybrid apps, including Cordova (PhoneGap) and Appcelerator Titanium Mobile. The framework allows the app to access some system functions such as the camera or address book, which provides additional functionality over pure web apps.

With that said, Embarcadero missed a big one in this notable quotable: Progressive Web Applications. Discussed hereafter, this powerful new approach to cross-platform development is baked into TMS WEB Core and its Pas2js engine as the foremost proverbial ingredient to ensure developers can feast plentifully on new far-reaching design and implementation options.

THE FUTURE IS NOW: Progressive Web Apps

Starting back in 2014, in an effort not (to appear) to be evil (**https://en.wikipedia.org/wiki/Don%27t_be_evil**), Google began work on the then new open-source application container framework now referred to as a "Progressive Web App" (PWA). While a typical Web application is essentially a website adapted to mimic a slimmed-down desktop or mobile application, a PWA offers much more native app-like capabilities. For example, as listed hereafter, although a PWA can load like a regular Web page or Website, it can also offer the user functionality such as local installation, offline operation, push notifications, and access to device resources traditionally available only in native applications.

Key Features of a PWA

Progressive: Works for every user, regardless of browser choice

Responsive: Works for every device, regardless of size: desktop, mobile, tablet

Offline Enabled: Enhanced with the Service Worker API to work offline

App-like: Native feel with smooth 60/FPS animations + Single Page navigation

Fresh: Always up-to-date from the Service Worker API's update process

Safe & Secure: Served over HTTPS w/ options for HTTP 2

Discoverable: Identifiable as an "application" from the manifest

Re-engageable: Re-engage easily with the Push Notifications API

Installable: Bypass the app store and install directly from URL

Full-screen: Full Screen web app experience like native apps

Linkable: Share via URL, does not require complex installation.

In contrast to the relatively seamless, straightforward development experience afforded by TMS WEB Core in both Delphi and Lazarus, the convoluted cross-platform development experience presented by both same IDEs for purely native applications can be quite daunting. Under Delphi, you'll minimally need to master its Platform Assistant [PAServer] technology to connect the IDE to your deployment target, and you'll need to dive deeply into a litany of deployment settings as well. On Lazarus, cross-platform targeting is less constrained but also less well-defined and delineated; there you'll need to begin by building a custom IDE for each platform targeted for deployment.

Juxtaposed these native efforts, Web application development is far less complicated within these same IDEs. With

TMS WEB Core installed along side Pas2JS, you"ll enjoy a greatly simplified development and debugging experience targeting today's Internet browsers as your application container as their innate abstractions hide underlying differences of the hosting hardware platforms and operating systems. Concerning debugging, all leading browsers today offer a built-in debugger that works with the Web application's native language via source map files generated by the associated cross-compiler/transpiler (for more information on source maps, see: **github.com/ryanseddon/source-map/wiki/Source-maps:-languages,-tools-and-other-info**).

layout can be done at design-time using TWebForm (WEB Core's VCL/LCL TForm analogue) and the Delphi/Lazarus' object inspector to specify form and fit (size, position and styling) of any TMS Web components. Otherwise, to adhere to more traditional Web design practices (where Object Pascal's "Object Inspector" would be a strange and encumbering beast :), all HTML and CSS presentation work can be done using any of a plethora of standard layout tools available today (such as Adobe's Dreamweaver, Brackets, Bluefish, Atom, or Webflow) to produce Web content. In turn, all functional programming of this Web content (its customized behavioral plumbing) can be done in a manner familiar to seasoned Object Pascal developers using Lazarus or Delphi with TMS WEB Core installed. Now, with TMS WEB Core installation at hand, see the separate articles in this issue concerning TMS WEB Core installation for both Delphi and Lazaurs (pages 10 and 48).

RAD VERSUS CLASSIC WEB DESIGN

A (final) note concerning RAD versus classic Web design. While the Internet browsers themselves may vary somewhat in their adherence to and implementation of HTML5, CSS3 and ECMAScript/JavaScript standards (for which many "Polyfills" **https://remysharp.com/2010/10/08/what-is-a-polyfill** — such as Babel (**https://babeljs.io/**) and Modernizr (**https://modernizr.com/**) — abound to address browser deficiencies), most of those details can be ignored if choosing to use TMS WEB Core in its fullest of RAD experience.

To approach Web presentation design work using the RAD practices, most aspects of

About the Author

Robert Welland is a developer and technical writer for Learning Frameworks — a software consultancy based in North America with offices in Canada and the US. Robert can be reached at **support@objectpascal.org**.

GETTING INSTALLATION GOING

This section describes all steps needed to install TMS WEB Core manually from its TMS repository into the Lazarus IDE. For an introduction to TMS WEB Core, see the related articles in this issue by Robert Welland and Danny Wind or refer to TMS Software's online WEB Core Developers Guide, **webcore.html** available under: **`tmssoftware.biz/download/manuals/webcore`**

Before you begin, remove any prior version of TMS WEB Core for Lazarus and ensure the file path planned for this new installation contains NO SPACES. Although this limitation is corrected in its most recent release (2.0.2 as of 2019-04-16), Lazarus 2.0.0 does not support spaces in the path to its underlying Pascal-to-ECMAScript/JavaScript transpiler (**`pas2js.exe`**). If spaces are present, two warning messages are presented on IDE startup (and, thereafter, the IDE uses the transpiler path defined by its Free Pascal compiler/fpc settings instead).

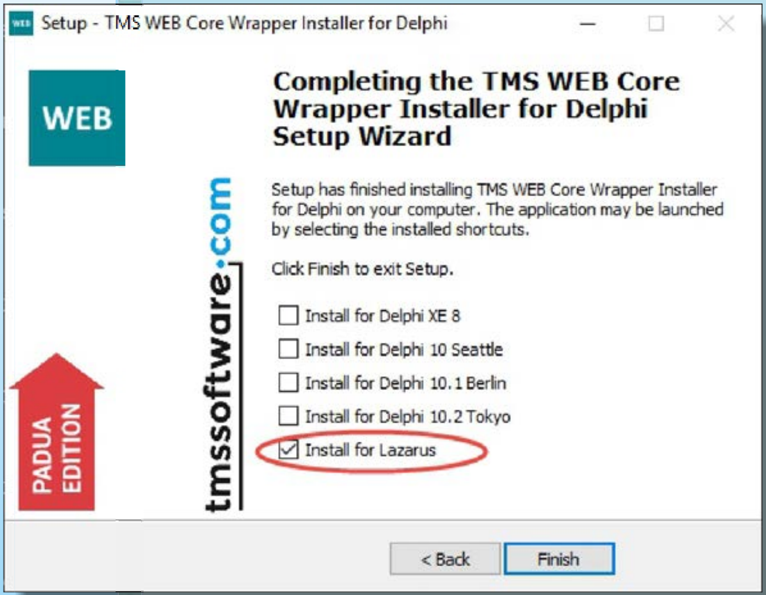
While latest TMS WEB Core for Lazarus consists of four **`*.lpk`** packages (2 design-time plus 2 run-time as listed above), only **`the TMSWEBCorePkgLibLaz.lpk`** package needs to be installed via the Lazarus IDE as the other 3 packages are automatically referenced and integrated when needed.

Since the current version of the Lazarus IDE (v2) statically links all design packages, installation of TMS WEB Core under Lazarus requires the IDE to rebuild itself to complete this integration. With there being no fully automated TMS WEB Core installer for Lazarus (yet), the current TMS WEB Core installer (common to both Delphi and Lazarus) will place all files needed for the manual installation under Lazarus (described below) in any folder of your choosing.

NOTE: To install Web Core on Lazarus, only the **`TMSWEBCorePkgLibLaz.lpk`** package needs to be installed in the IDE.

Before beginning manual installation under Lazarus, first complete TMS WEB Core repository installation via its Inno-based installer provided by TMS which provides the following option to install Lazarus:

Begin Installation


Setup - TMS WEB Core Wrapper Installer for Delphi

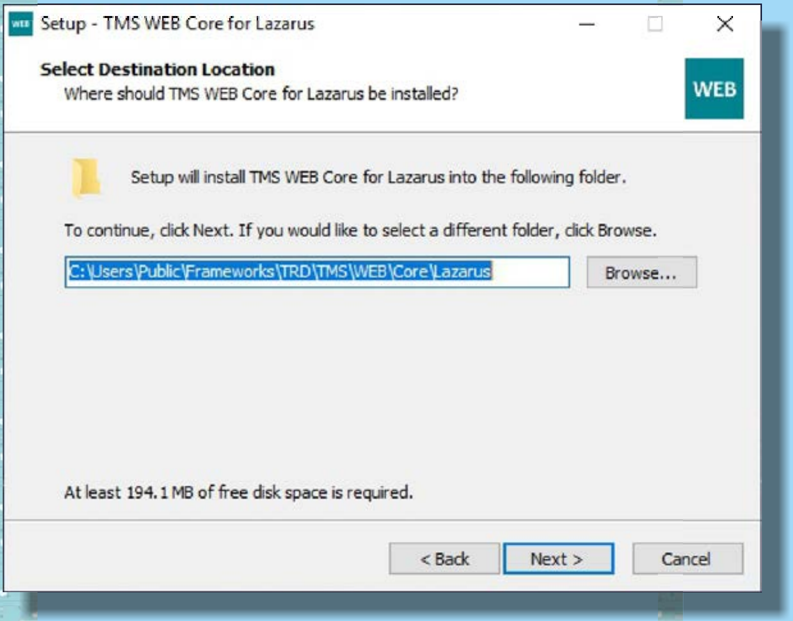
Completing the TMS WEB Core Wrapper Installer for Delphi Setup Wizard

Setup has finished installing TMS WEB Core Wrapper Installer for Delphi on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☐ Install for Delphi XE 8
☐ Install for Delphi 10 Seattle
☐ Install for Delphi 10.1 Berlin
☐ Install for Delphi 10.2 Tokyo
☑ Install for Lazarus

Select TMS WEB Core installation Path



Setup - TMS WEB Core for Lazarus

Select Destination Location

Where should TMS WEB Core for Lazarus be installed?

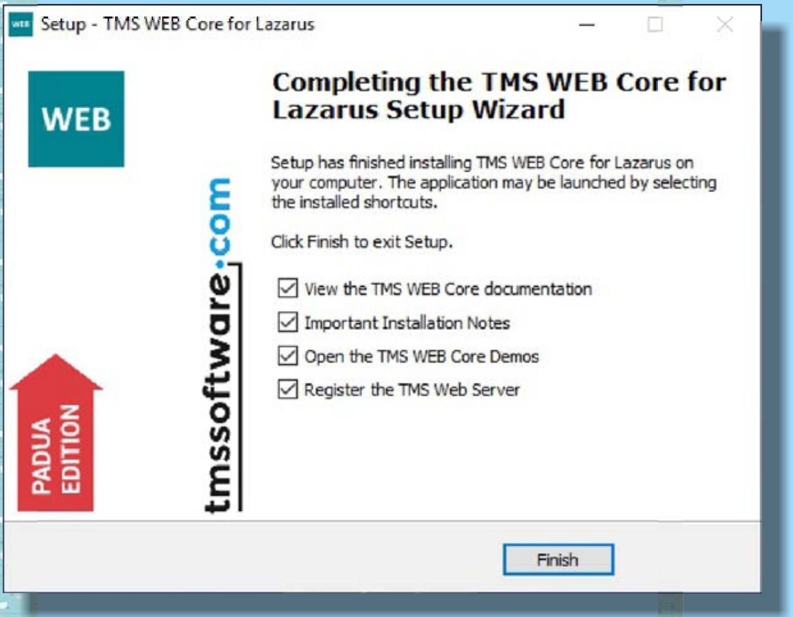
Setup will install TMS WEB Core for Lazarus into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Users\Public\Frameworks\TRD\TMS\WEB\Core\Lazarus

At least 194.1 MB of free disk space is required.

TMS Installation Progresses until Done



Setup - TMS WEB Core for Lazarus

Completing the TMS WEB Core for Lazarus Setup Wizard

Setup has finished installing TMS WEB Core for Lazarus on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ View the TMS WEB Core documentation

☑ Important Installation Notes

☑ Open the TMS WEB Core Demos

☑ Register the TMS Web Server

TMS WEB Core Packages Installed

❶ Visual and Non-Visual Web Component RTL (TMSWEBCore.lpk)

❷ Object Pascal to ECMAScript/JavaScript compiler and RTL (pas2js.exe and pas2js_rtl.lpk)

❸ Lazarus IDE Integration (TMSWEBCorePkgDeLaz.lpk)

❹ A single installation package (TMSWEBCorePkgLibLaz.lpk) which integrates the above under Lazarus.

IDE INTEGRTION UNDER LA ZARUS

To perform the manual installation process required to integrate WEB Core within the Lazarus IDE:

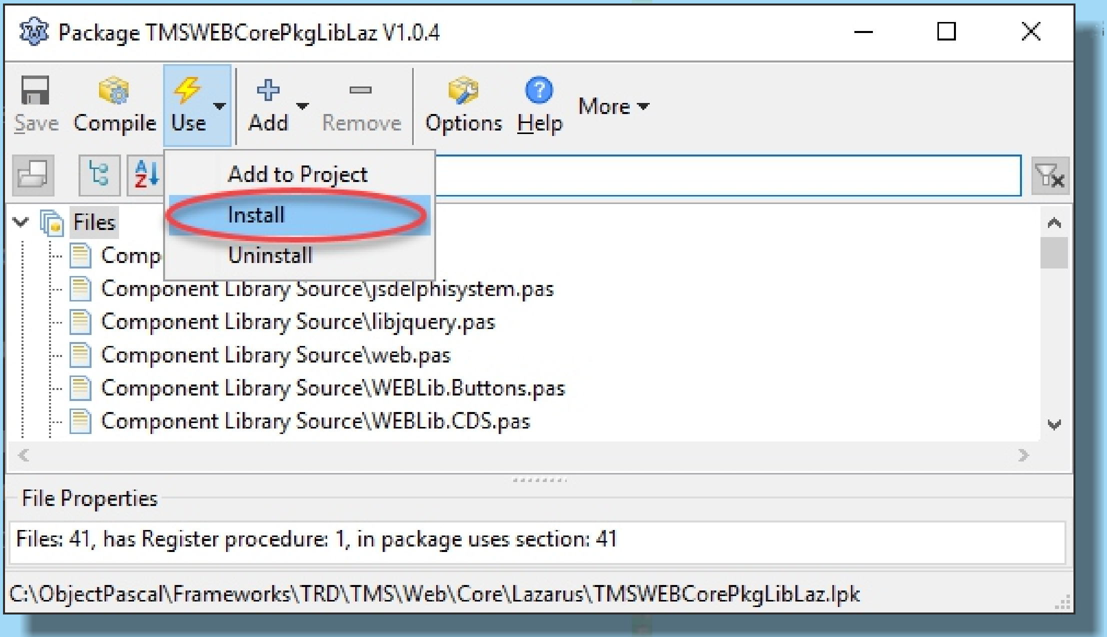
❶ Launch the Lazarus 2.0+ IDE

❷ Click on its menu item Package (ALT+C)

❸ Select the submenu item Open Package FIle (`*.lpk`) ...

❹ Navigate, via the resulting File Navigator, to the root directory of your TMS WEB Core for Lazarus repository (hereafter -- within the Lazarus IDE Options, et cetera, this installation directory is referred to as **`TMSWebDir`**)

❺ Select TMSWEBCorePkgLibLaz.lpk and,



Component Library Source\jsdelphisystem.pas
Component Library Source\libjquery.pas
Component Library Source\web.pas
Component Library Source\WEBLib.Buttons.pas
Component Library Source\WEBLib.CDS.pas

Files: 41, has Register procedure: 1, in package uses section: 41

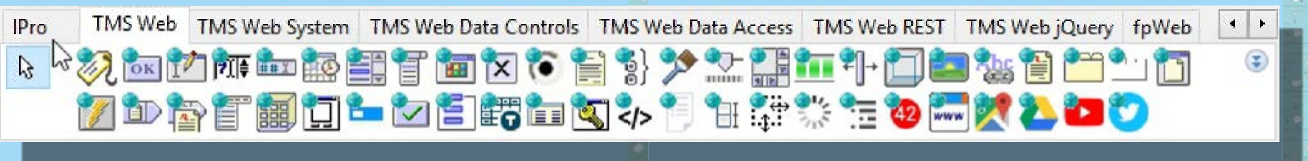
C:\ObjectPascal\Frameworks\TRD\TMS\Web\Core\Lazarus\TMSWEBCorePkgLibLaz.lpk

within the resulting Package Viewer, select Use | Install shown here:

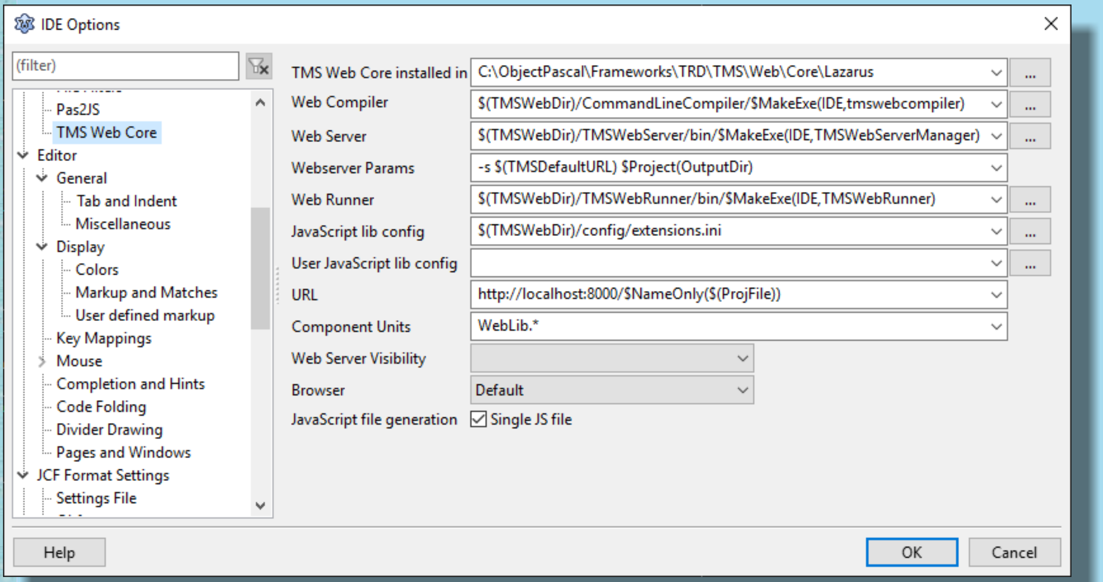
❻ Lazarus will proceed with compiling the package, then rebuilding itself and restarting itself to present you with a collection of 6 new tabs comprising all the controls available in TMS WEB Core Framework as shown here:

CONFIRMING INSTALLATION

With TMS WEB Core statically-linked in the newly rebuilt Lazarus IDE, the component palate should show 6 new tabs containing various TMS WEB Core components. Furthermore, as WEB Core's installation path (represented TMSWebDir) is computed from the location of **`TMSWEBCorePkgLibLaz.lpk`** as selected for installation (above), this location can be confirmed via the Lazarus menu item Tools menu > Options... (SHIFT+CTRL+O) > Environment >TMS WEB Core. Within the IDE Options dialogue (shown on the following) several configuration parameters and paths are available to fine tune TMS Web Core's installation. All such parameters and paths are itemized thereafter.



UNDERSTANDING IDE OPTIONS



IDE Options

(filter)

- Pas2JS
- TMS Web Core
- Editor
  - General
    - Tab and Indent
    - Miscellaneous
  - Display
    - Colors
    - Markup and Matches
    - User defined markup
  - Key Mappings
  - Mouse
  - Completion and Hints
  - Code Folding
  - Divider Drawing
  - Pages and Windows
- JCF Format Settings
  - Settings File

TMS Web Core installed in: C:\ObjectPascal\Frameworks\TRD\TMS\Web\Core\Lazarus

Web Compiler: $(TMSWebDir)/CommandLineCompiler/$MakeExe(IDE,tmswebcompiler)

Web Server: $(TMSWebDir)/TMSWebServer/bin/$MakeExe(IDE,TMSWebServerManager)

Webserver Params: -s $(TMSDefaultURL) $Project(OutputDir)

Web Runner: $(TMSWebDir)/TMSWebRunner/bin/$MakeExe(IDE,TMSWebRunner)

JavaScript lib config: $(TMSWebDir)/config/extensions.ini

User JavaScript lib config:

URL: http://localhost:8000/$NameOnly($(ProjFile))

Component Units: WebLib.*

Web Server Visibility:

Browser: Default

JavaScript file generation: ☑ Single JS file

Under IDE Options for Lazarus, upon selecting TMS WEB Core from the menu tree on the left, the following configuration parameters and paths are available to fine tune TMS Web Core's installation. Note that all default values are in bold and any custom path given is for reference only as the installation path (**TMSWebDir)** will vary with setup of TMS WEB Core for Lazarus.

- "TMS WEB Core installed in" (**defaults to TMS WEB Core's installation directory**)

  Defines the path wherein the TMS WEB Core for Lazarus installation was installed as discussed previously above. Relative to this path, the compiler searches for source files and resources needed to build each WEB Core project.

- Web Compiler **$(TMSWebDir)/CommandLineCompiler/$MakeExe(IDE,tmswebcompiler)**

  Defines the compiler binary used when setting up a new project (pas2js.exe). It will be searched in the PATH if no absolute path is set and corresponds to IDE macro Pas2JS.

- Web Server **$(TMSWebDir)/TMSWebServer/bin/$MakeExe(IDE,TMSWebServerManager)**

  Defines the webserver started when a project is run. By default, the TMS Web Server is used (based on TMS XData/Sparkle technology on Windows) and, otherwise, is the simpleserver application from the FPC project.

  Any web server can be specified here so long as it serves files from the directory in which it is started.

- Web Server Params **-s $(TMSDefaultURL) $Project(OutputDir)**

  Defines any extra command-line parameters required when launching the webserver. Additionally, it defines the project's output path where files from the web application (or any associated web content) will be served.

- Web Runner **$(TMSWebDir)/TMSWebRunner/bin/$MakeExe(IDE,TMSWebRunner)**

  Defines the browser/application used to launch the project's resulting web application (typically, a local browser defined below adjacent Browser). Alternatively, a Node.js executable may be specified here to launch a Node.js project. Under Windows, by default, TMSWEBRunner.exe is used; on all other operating systems, the shell script specified here is invoked.

JavaScript lib config **`$(TMSWebDir)/config/extensions.ini`**

Defines which JavaScript libraries and CSS files are added to each WEB Core project's main HTML file (specified separately under the Project Options.

- User JavaScript lib config

  Defines which JavaScript libraries are available to use within the WEB Core project (as CDN links/references loaded with the project's main HTML file)

- URL **`http://localhost:8000/$NameOnly($(ProjFile))`**

  Defines the URL used by the IDE to launch the web application embodied within each WEB Core project. If a web server other than the default defined by TMS is used, the URL can be modified here.

- Component Units **`WebLib.*`**

  Defines a namespace prefix to associate a framework containing visual and non-visual controls. By default, TMS WEB Core prefixes WebLib. to its controls while the related TMS FNC prefixes FNC. to its constituent components.

- Web Server Visibility **`HIDDEN (=blank)`**

  Defines whether the web server is hidden when launched or remains visible.

- Browser **`Default`**

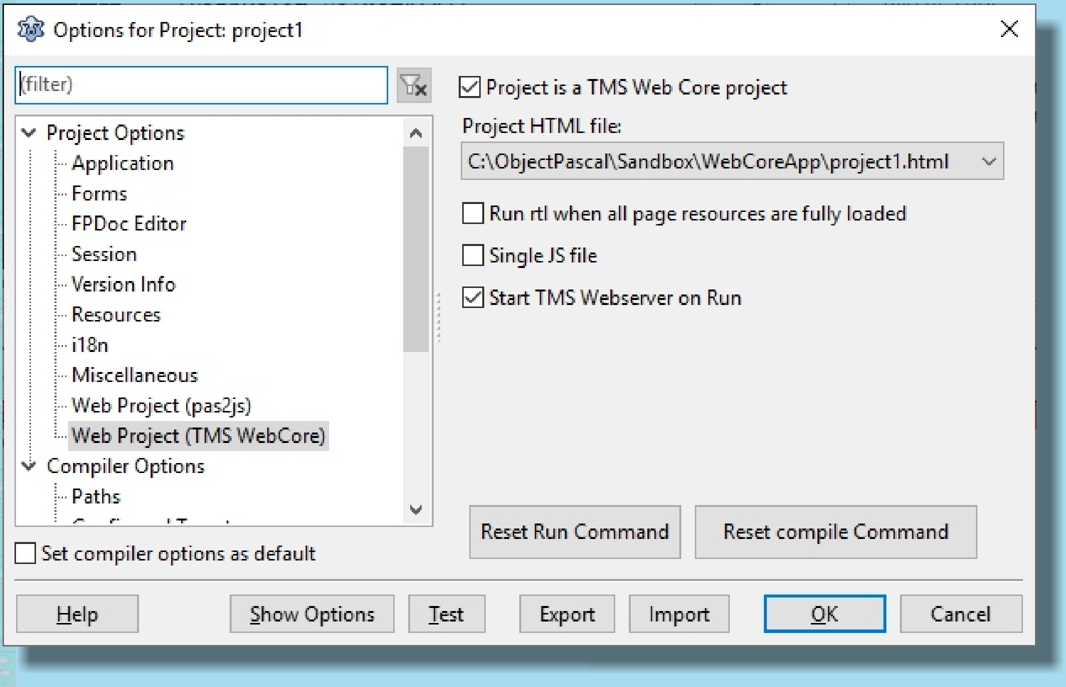
  Defines which HTML/Internet browser the IDE is to use when opening/running a project's Web application. When nothing is specified, this is the default operating system browser. It will be searched in the PATH if no absolute path is set and corresponds to IDE macro Pas2JSBrowser.

- JavaScript file generation **`Ckecked (=use single JS file)`**

  This checkbox defines whether a single ECMAScript/JavaScript file will be generated for the entire web application. If unchecked, each unit must have its own .js file manually specified in the project's main HTML file. By default, this box is checked so only one .js is generated for the project which is automatically specified in the project's main HTML file.

NOTE: The environment variable `TMSWebDir` defines this root location of the TMS WEB Core installation for Lazarus and this important environment variable is set in IDE Options by navigating to Tools | Options... | Environment | TMS WEB Core and entering the path adjacent the tagline, "TMS WEB Core installed in" (which really should read as TMSWebDir) shown above. This path defaults to TMS WEB Core's installation directory (for Lazarus). If overridden by the user in this IDE edit box provided, the user-defined value is persisted with the key WebInstallDir in the configuration file, TMSWebCoreOptions.xml, in the IDE's configuration directory.

UNDERSTANDING PROJECT OPTIONS



Options for Project: project1

(filter)

- Project Options
  - Application
  - Forms
  - FPDoc Editor
  - Session
  - Version Info
  - Resources
  - i18n
  - Miscellaneous
  - Web Project (pas2js)
  - Web Project (TMS WebCore)
- Compiler Options
  - Paths

☑ Project is a TMS Web Core project

Project HTML file: C:\ObjectPascal\Sandbox\WebCoreApp\project1.html

☐ Run rtl when all page resources are fully loaded

☐ Single JS file

☑ Start TMS Webserver on Run

☐ Set compiler options as default

Project Options are accessed via Project > Options... (SHIFT+CTRL+F11) > Web Project (TMS WEB Core). By default, the TMS WEB Core general settings defined under IDE Options are applied when a new project is created. Note here that all default values are in bold below and that any custom paths or values are given for reference purposes only as project settings may vary with use on TMS WEB Core for Lazarus. Once a project has been created or loaded under Lazarus (and be sure to save any newly created TMS Web Core projects immediately), the Lazarus IDE offers the following settings to finely tune each TMS WEB Core project:

- TMS WEB Core Project | checkbox: checked (=True)

  With this flag set, Lazarus performs some additional checks to ensure the correct component types are used (only components such as those from the TMS Web Core suite are supported presently for Web applications). Furthermore, when dropping controls on the designer form, if this parameter is checked|True, incorrect component classes are forbidden. Finally, setting this flag True ensures the tmswebrunner.ini is created and "normal" (*.exe|*.o) execution is intercepted to launch the webserver and TMSWebRunner.exe instead.

- Project HTML file: (`path to project location\<project mame>.html`)

  Optionally overrides IDE Options to define a different HTML file to launch the web application

- Run rtl when all page resources are fully loaded | `checkbox: unchecked (=False)`

  Optionally overrides IDE Options to operate RTL differently than previously defined

- Single JS file | `checkbox: unchecked (=False)`

  Optionally overrides IDE Options to render JS file(s) differently than previously defined

- Start TMS Webserver on Run | `checkbox: checked (=True)`

  Optionally overrides IDE Options to operate Web Server differently than previously defined

CREATING A TEST PROJECT

To confirm TMS WEB Core for Lazarus is properly installed, create sample WEB Core project from within the Lazarus IDE as follows:

❶ Select Project (ALT+P)

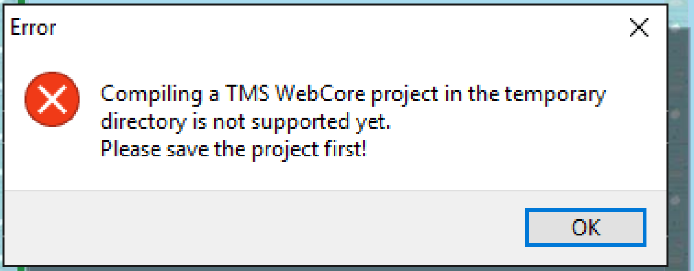
❷ Click New Project ... to open the "Create a new project" dialogue

❸ Under Project, select TMS WEB Core Application and click Okay

❹ On the subsequent form "TMS WEB Core Project Options" which opens, leave the items (Run Web application ..., Do not create form, unchecked) and simply click Okay

❺ (A new template project should then appear containing a blank FormX).

❻ Save the template project to desired location via Project (ALT+P) > Save Project OR Save Project As ...
NOTE: If you do not save the project before running it, you will see this error:



Compiling a TMS WebCore project in the temporary directory is not supported yet.
Please save the project first!

❼ Add a TMS WEB Core visual control from the TMS Web tab of Lazarus component palette (e.g. a TWebLabel and set its Caption := 'Hello World')

❽ Under IDE menu item Run (ALT+R), select Compile (CTRL+F9) and then Run (F9)

❾ Assuming you did not change the default IDE options (as listed hereafter), your sample project should run by launching a HTTP test server and opening an instance of your system's default Web browser to display your app.

❿ Note that beyond global IDE options tabulated above, a project's local options are accessible via IDE menu Project > Project Options... (SHIFT+CTRL+F11).

TROUBLESHOOTING ISSUES

Error: can't find unit "System"

This error is indicative of an improperly configured TMS WEB Core installation. If this error is observed upon compiling the test project created adjacent (or any other WEB Core effort), locate the ECMAScript/JavaScript Transpiler configuration file, pas2js.cfg, found in the Compiler directory under the installation root for TMS WEB Core. Therein, using just a standard text editor (such as Notepad++) to view the file's contents, confirm the following two paths are defined and are valid (make sure they exist on the installation system):

- -Fu path pointing to the source units for TMS WEB Core repository
- -Fu path pointing to the source units for the Pas2js open-source effort underpinning TMS WEB Core.

Can't Find Pre-compiled Units

This error message may be seen following a fresh install of the FPC or Lazarus IDE or Pas2js Transpiler or TMS WEB Core or any one of the four in sequence onto a system that already has/had a prior version of one or more of the these tools installed.

It is also seen when trying to build FPC, Lazarus, Pas2js and/or TMS WEB Core from source code. The error simply means that the pre-compiled units of FreePascal or the Pas2js RTL — for one reason or another — are simply not being found by the compiler. There are generally two causes for these errors:

- left over remnants from an prior installation
- environment problems (conflicting path declarations, duplicated environment variables with differing values, etc).

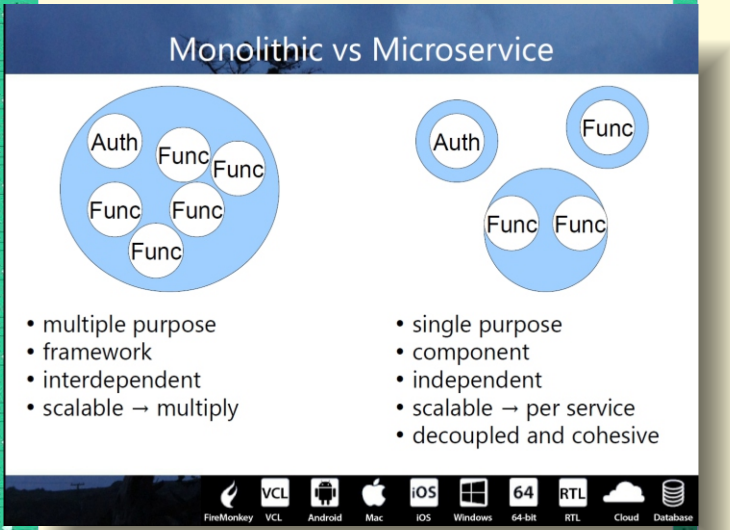
About the Author

Robert Welland is a software developer and technical writer for Learning Frameworks — a software consultancy based in North America with offices in Canada and the US. Robert can be reached at **support@objectpascal.org.**

BY DANNY WIND

MICROSERVICES

use a divide and conquer approach to monolithic and large (web)services. Just cut a large webservice into much smaller pieces that are decoupled, cohesive and easy to use. The benefit of this approach is simpler re-use, where you combine existing and new microservices for each project. Scalability is improved as well, just add more instances of the microservices that are heavily used.

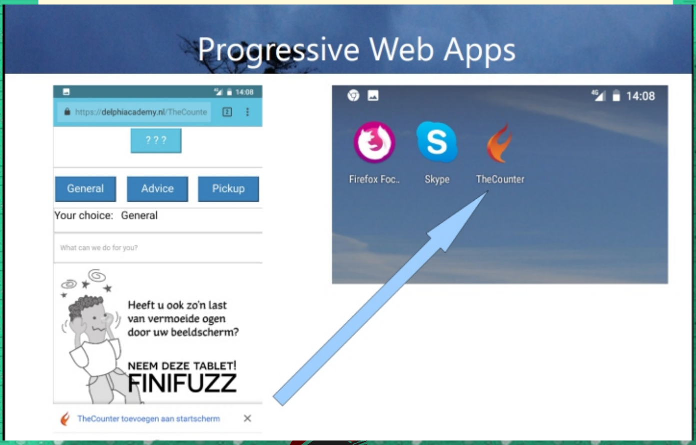


Monolithic vs Microservice

- multiple purpose
- framework
- interdependent
- scalable → multiply

- single purpose
- component
- independent
- scalable → per service
- decoupled and cohesive

Figure 1: Monolithic vs Microservice



Progressive Web Apps

Your choice: General

Heeft u ook zo'n last van vermoeide ogen door uw beeldscherm?

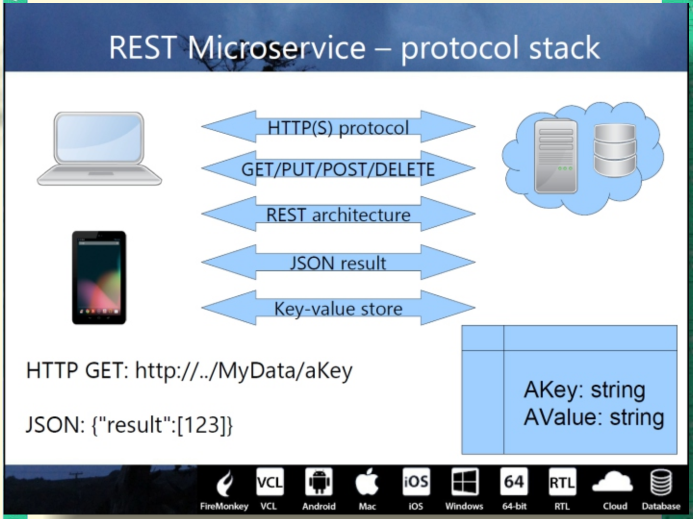
NEEM DEZE TABLET! FINIFUZZ

TheCounter toevoegen aan startscherm

Figure 2: Progressive Webb Apps - Install after first URL visit

PROGRESSIVE WEB APPS

allow the user to run a web page full screen as if it were an app. The PWA is installed, with icons, it looks and behaves as-if it were a native App. In reality its simply HTML5 with JavaScript, behaving as a very lightweight App.



REST Microservice – protocol stack

HTTP GET: http://../MyData/aKey

JSON: {"result":[123]}

AKey: string
AValue: string

Figure 3: Progressive Web App REST Microservice protocol stack

MICROSERVICES AND PROGRESSIVE WEB APPS

combined is where things get interesting. If you combine the back-end simplicity of microservices with a lightweight PWA you end up with an easy to deploy and easy to use architecture for small use cases. In this article you'll learn how to create both in a use case for a ticketing App for a Pharmacy. The code can very easily be re-used for many other small apps, ranging from in-shop offers of discounts to Apps for the Sports day at the local club.

WHAT DO YOU NEED?

The article uses Delphi 10 Community Edition to create the WebBroker microservice and TMS Web Core to transpile a Delphi App to a Web App. TMS Web Core uses the open source Pas2JS as the transpiler. With some manual work the same can be accomplished in Lazarus, but that is beyond scope of this article. As a side note, a version of Pas2JS with TMS WEB Core for Lazarus is on its way.

REST WEB MICROSERVICE OVER HTTP

The microservice we will create will be accessed over the web with HTTP(S) commands and is based on REST, REpresentational State Transfer. Each of the HTTP commands will be used to GET (Select), PUT (Insert), POST (Update) or DELETE (Delete) a value from the webservice. The translation from HTTP commands like PUT to an action such as Insert is a simplification, but for this simple use-case it more than covers what we need. Note that both GET and PUT are idempotent, which will lead to some issues

REST WEB MICROSERVICE OVER HTTP

The microservice we will create will be accessed over the web with HTTP(S) commands and is based on REST, REpresentational State Transfer.
Each of the HTTP commands will be used to GET (Select), PUT (Insert), POST (Update) or DELETE (Delete) a value from the webservice. The translation from HTTP commands like PUT to an action such as Insert is a simplification, but for this simple use-case it more than covers what we need.

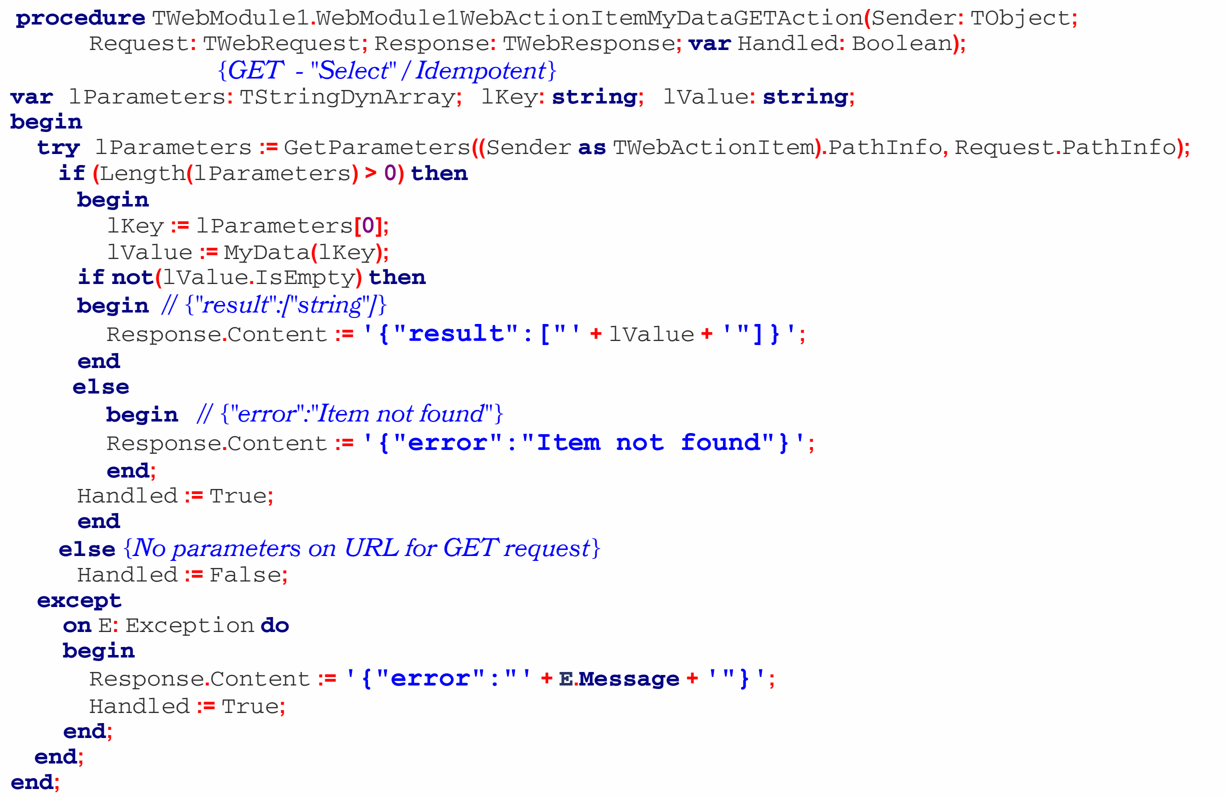
Note that both GET and PUT are idempotent, which will lead to some issues that we'll cover later on. If you'd like to delve into details on HTTP commands, please read the RFC document here
`https://tools.ietf.org/html/rfc2616`
scroll down to GET and the other HTTP methods in section 9.3, and try not to fall asleep.

Let's create the REST microservice

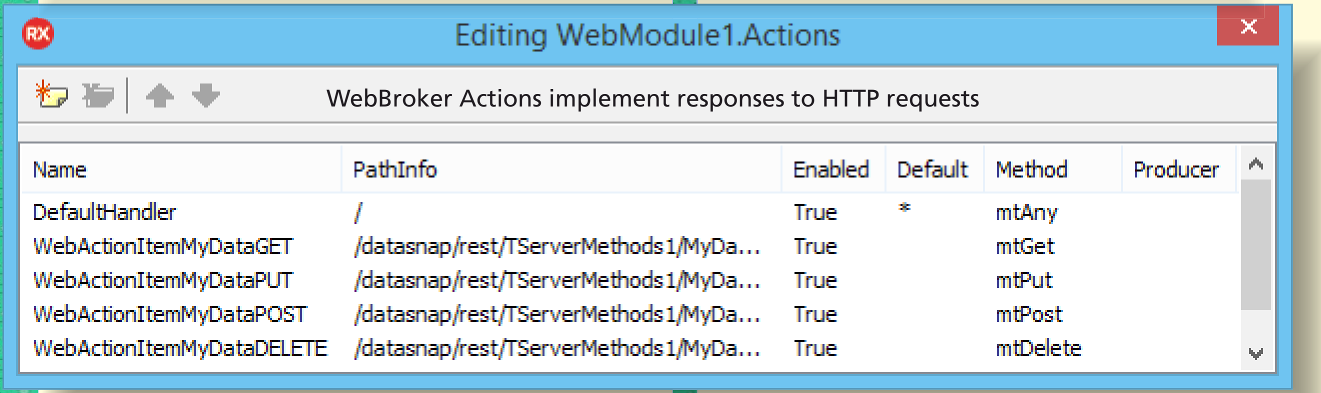
As with all things Delphi, creating a web server microservice is made easy with a wizard.
Just create a new WebBroker webservice with
File | New | Other, Delphi | Web | Web Server Application.

Use the Windows Stand-alone GUI VCL application for now. If at a later point you want to deploy it to IIS or Apache, just create an empty version for either and share the WebModuleUnit file with the one made in this Stand-alone GUI application.

Open the WebModuleUnit and open the Actions property editor for the WebModule. The Actions property implements responses to the HTTP requests that the microservice receives. Here we add four methods corresponding to HTTP GET, PUT, POST and DELETE.
The Delphi code for the GET method looks like this.



```
 procedure TWebModule1.WebModule1WebActionItemMyDataGETAction(Sender: TObject;
       Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
                    {GET - "Select" / Idempotent}
var lParameters: TStringDynArray;  lKey: string;  lValue: string;
begin
   try lParameters := GetParameters((Sender as TWebActionItem).PathInfo, Request.PathInfo);
     if (Length(lParameters) > 0) then
      begin
        lKey := lParameters[0];
        lValue := MyData(lKey);
      if not(lValue.IsEmpty) then
      begin // {"result":["string"]}
        Response.Content := '{"result":["' + lValue + '"]}';
      end
      else
        begin  // {"error":"Item not found"}
        Response.Content := '{"error":"Item not found"}';
        end;
      Handled := True;
      end
     else {No parameters on URL for GET request}
      Handled := False;
   except
     on E: Exception do
     begin
       Response.Content := '{"error":"' + E.Message + '"}';
       Handled := True;
     end;
   end;
end;
```



Editing WebModule1.Actions

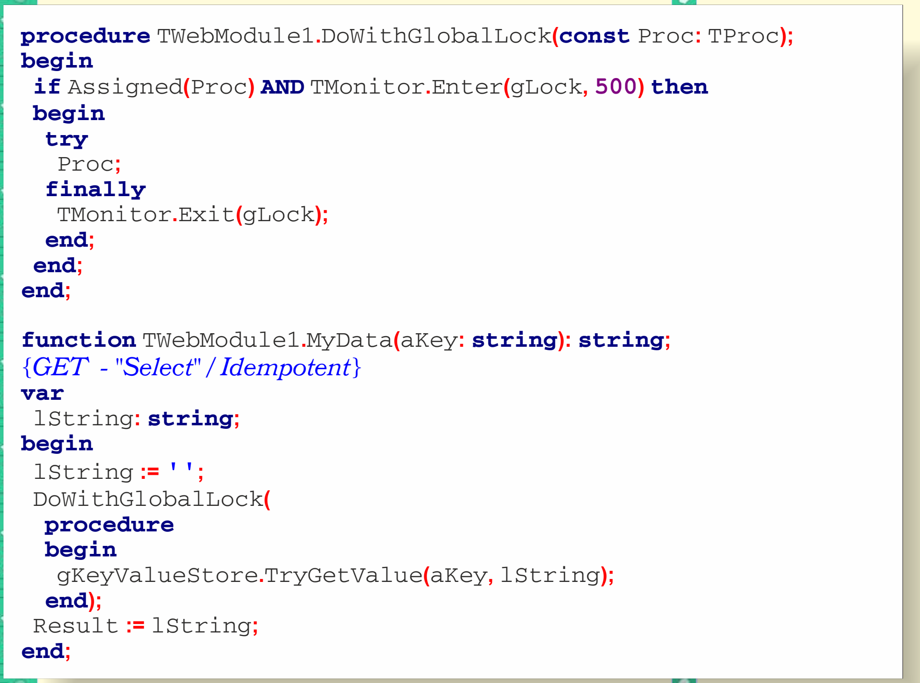
WebBroker Actions implement responses to HTTP requests

Name	PathInfo	Enabled	Default	Method	Producer
DefaultHandler	/	True	*	mtAny	
WebActionItemMyDataGET	/datasnap/rest/TServerMethods1/MyDa...	True		mtGet	
WebActionItemMyDataPUT	/datasnap/rest/TServerMethods1/MyDa...	True		mtPut	
WebActionItemMyDataPOST	/datasnap/rest/TServerMethods1/MyDa...	True		mtPost	
WebActionItemMyDataDELETE	/datasnap/rest/TServerMethods1/MyDa...	True		mtDelete	

What the GET request does is, it parses the HTTP parameters it receives from a HTTP request and returns a simplified formatted JSON response.
It uses a `TDictionary<string, string>` as a in-memory and transient Key-Value store.
At this point I'd suggest downloading the source files from the link at the end of this article and open the ProjectGroup in the Source folder in your Delphi 10 Community IDE.
Compile and run TheCounterServerForDelphiPro project, on the second tab start the server.

Now open this URL in your browser.
`http://localhost:8080/datasnap/rest/TServerMethods1/MyData/1`
The resulting response should look like this
`{"error":"Item not found"}`
Then try this URL which requests the default item 0 thats always added to the MyData TDictionary storage.`http://localhost:8080/datasnap/rest/TServerMethods1/MyData/0`The resulting response should look like this{"result":["Zero"]}
The JSON result strings formatting is based on the formatting thats also used in the DataSnap REST Messaging Protocol.

`http://docwiki.embarcadero.com/RADStudio/Rio/en/DataSnap_REST_Messaging_Protocol`
Although this session focuses on using Delphi 10 Community Edition, please note that the Delphi 10 Enterprise with DataSnap allows you to get the same result with just two lines of code, so that may be a better choice if you need to write more than a few REST EndPoints. Also the JSON result will be more complete, with support for escape characters when needed.
Multi-user and multi-threading
The WebBroker you've just created is multi-threading, for each user / request a separate thread is started to handle the request. The TDictionary used for storage is not thread-safe, so we need to add some protection around it.



```
procedure TWebModule1.DoWithGlobalLock(const Proc: TProc);
begin
 if Assigned(Proc) AND TMonitor.Enter(gLock, 500) then
 begin
  try
   Proc;
  finally
   TMonitor.Exit(gLock);
  end;
 end;
end;

function TWebModule1.MyData(aKey: string): string;
{GET  - "Select" / Idempotent}
var
 lString: string;
begin
 lString := '';
 DoWithGlobalLock(
  procedure
  begin
   gKeyValueStore.TryGetValue(aKey, lString);
  end);
 Result := lString;
end;
```

We use a **`TMonitor`** with a TimeOut to try to get a lock on the **`gKeyValueStore Dictionary.`** If it fails the Proc is not called, and an empty string is returned.

`TMonitor` has some advanced features to handle the TimeOut, one of them is a short SpinWait before it yields to other threads, making it more efficient than a simple Yield or Sleep. Because the dictionary is fast and efficient, this works out quite well.
This solution leads to serialization of calls into the web service, which limits its capability to handle many users. For our use case this is not relevant, it works just fine for the number of customers expected in a pharmacy. But if you need to scale up consider using a different approach.
Let's create the web app. First you'll need to install **TMS WEB Core** into your **Delphi 10**.

`https://www.tmssoftware.com/site/tmswebcoreintro.asp`
If you're not at all familiar with TMS WEB Core, you can take a look at the Web Core Intro video available on the above URL, or just follow the below steps, to get an idea of what it is.
We'll start our web app with **File | New | Other** and **Delphi | TMS Web | TMS Web Application**.
You'll get a new project that contains the regular Delphi files as well as a Project1.html.
Place a TWebButton on the Form. Then add a TWebHTTPRequest component.
In the OnClick event-handler of the Button code

```
WebHTTPRequest.Command := httpGET; {Select}
WebHTTPRequest.URL := 'http://localhost:8080/datasnap/rest/TServerMethods1/MyData/'
 + aKey;
WebHTTPRequest.Execute;
```

This is an asynchronous call, the result of which is returned at some point in the future and can be handled in the Response event-handler of the WebHTTPRequest.

```
procedure TDataModuleMain.WebHttpRequestResponse(Sender: TObject;  AResponse: string);
var
 lArray: TJSONStringDynArray;
begin
 {JSON is parsed and returned as an array of strings}
 JSONToStringArray(AResponse, lArray);
 if (Length(lArray) = 1) then
  WebButton.Caption := lArray[0];
end;
```

The **`JSONToStringArray`** is separately coded in a unit that's included in the download. The Pas2JS transpiler that is used by TMS WEB Core is a work in progress. The most frequently used RTL functions and procedures are available for transpilation, but not all, and some of the JSON functions need to be added manually. Add the supplied **`UnitJSONHelper.pas`** to the Project and continue.

With the REST microservice up and running, now run the web app. Try a value of '0' for the aKey (hardcode this or add an Edit for it) to get the 'Zero' result back from the REST microservice.

How does this work?

As soon as you run the web app the IDE plugin for **TMS WEB Core** will go through a couple of steps. The Delphi code is transpiled into **JavaScript** with the **Pas2JS** transpiler. Part of the RTL units used are added in JavaScript form as well. Also HTML Elements constructors matching the TWeb components you placed on the form will be added to the JavaScript code. The resulting files are placed in the **TMSWeb** output directory and then opened in the Web Browser.

The JavaScript code is run and the HTML Elements are constructed on the fly from code. So its all just HMTL and JavaScript files. This means that a transpiled Web App can be opened from the files on disk using a web browser.

Alternatively you could upload the **HTML** and **JavaScript** files to a webserver, like Apache or IIS. If you're going to host these files consider using **Nginx,** as its very efficient in serving static **HTML/JavaScript** files.

Look and feel in CSS

Because its HTML you style the look and feel using CSS. One of the easy ways of doing this is with a right-click on the Project in the Delphi IDE and enabling one of the bootstrap CSS libraries. Now set the ElementClassName property for each component you want to style, you could use 'btn-info' for the TWebButton component.

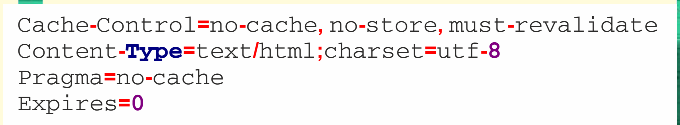
If you now run the web app the Button will have a blue bootstrap btn-info style. If you want you can take a look at the internals, just open the HTML file and notice that a <link rel="stylesheet" was added.

Idempotent HTTP Commands

According to the RFC specification the GET and PUT HTTP Commands are idempotent. The TL;DR summary boils down to the browser caching the response to URLs with GET and PUT commands. Any repeat of the same GET/PUT URL can be loaded from the browsers own cache, which saves a roundtrip to the REST microservice.

But in our setup we use GET requests to get the current value of aKey, the value of which can be updated or even deleted. So we need the GET request to actually be send to the microservice to get the latest value.

One solution to this issue is adding some headers to the GET and PUT HTTP request, which will tell the webbrowser we do not want it to cache these. This is what's used in our example source code. Just set the following in the Headers property of the TWebHTTPRequest



```
Cache-Control=no-cache, no-store, must-revalidate
Content-Type=text/html;charset=utf-8
Pragma=no-cache
Expires=0
```

Another solution would be to use the non-idempotent POST HTTP Command to get the new values.

Of course we need CORS (see the article in our previous issue 77 page 51.

because we need to make sure that the web browser won't block a request to a REST microservice that's on a different domain.

Supp

https://dannywind.nl/counter/TheCounterWeb.html

and this web page calls to a REST microservice hosted on this other domain in IIS

https://delphiacademy.nl/TheCounterISAPI.dll

if the microservice does not return a **Access Control Request Header** that allows calling into it from **dannywind.nl**, the web browser will **silently block the outgoing REST call.**
We need to add a **Cross Origin Resource Sharing (CORS)** header to the response of the REST microservice which lets the web browser know its OK to call into the webservice from the domain that the HTML and JavaScript files are hosted on.
In the WebModule of the microservice add a **BeforeDispatch** event-handler and code

```
procedure TWebModule1.WebModuleBeforeDispatch(Sender: TObject;
 Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
begin
 {Report back to the caller/browser that we allow
  Cross Origin Resource Sharing (CORS) for all calling domains}
 Response.SetCustomHeader('Access-Control-Allow-Origin','*');

 if Trim(Request.GetFieldByName(
   'Access-Control-Request-Headers')) <> '' then
 begin
  Response.SetCustomHeader(
   'Access-Control-Allow-Headers',
   Request.GetFieldByName('Access-Control-Request-Headers'));
  Handled := True;
 end;
end;
```

In the above code we're allowing any-and-all, '*', Origin domains.
For deployment you'd probably just add your own domains, which will prevent other domains from using your webservice, at least from compliant webbrowsers.

PROGRESSIVE WEB APPS

or PWA for short is the last piece of the puzzle.
Up until now we've been talking about a web app, although what we've made sofar is just a web page with HMTL and javaScript.
Why is that?

PWA' s will allow your web page to be installed as and App on the Home Screen, complete with icons, offline behaviour and full screen immersive mode that hides the browser.

The steps needed to transform a web page into a PWA are adding icons and a manifest that descibres the PWA and adding a service worker for offline support. In the included source code all these steps have been done for you, you can just edit the project to create your own PWA.

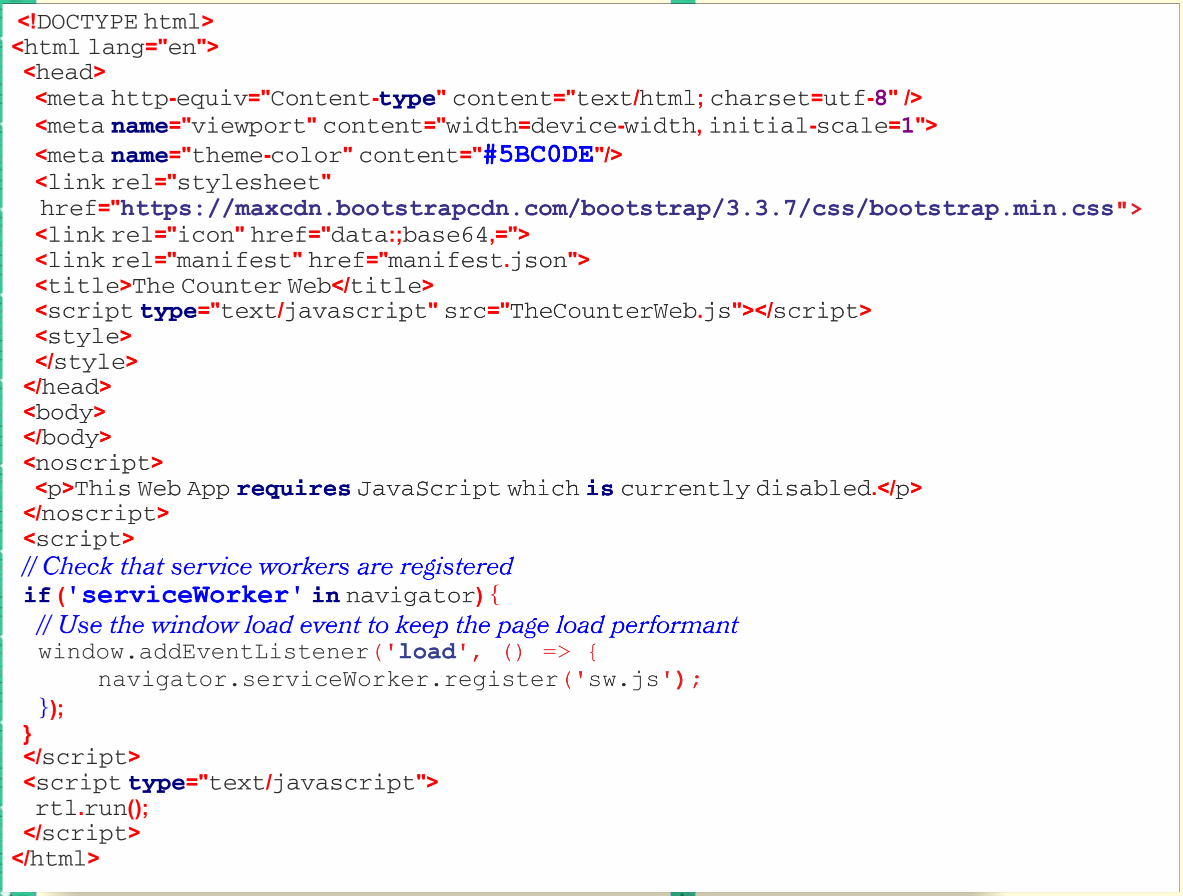
Add a manifest.json file with links to the icons for the PWA

```
{
 "name": "TheCounterWeb",
 "short_name": "TheCounter",
 "icons": [{
   "src": "FM_LauncherIcon_144x144.png",
     "sizes": "144x144",
     "type": "image/png"
   }, {
   "src": "FM_LauncherIcon_192x192.png",
     "sizes": "192x192",
     "type": "image/png"
   }, {
     "src":
     "FM_LauncherIcon_512x512.png",
     "sizes": "512x512",
     "type": "image/png"
   }],
"start_url": "TheCounterWeb.html",
"display": "standalone",
"orientation": "portrait",
"background_color": "#FFFFFF",
"theme_color": "#5BC0DE"
}
```

for the icons I've just used the default **`FM_LauncherIcon`** files and resized them to the required sizes as described in the PWA documents from Google.
Write a wrapper around the serviceworker framework supplied by Google

```
importScripts('https://storage.googleapis.com/workbox-cdn/releases/3.4.1/workbox-sw.js');

if (workbox) {
  console.log(`Yay! Workbox is loaded ??`);
  workbox.precaching.precacheAndRoute([
      { url: 'TheCounterWeb.js', revision: '1' },
      { url: 'TheCounterWeb.html', revision: '1' },
      { url: 'UnitDataMain.html', revision: '1' },
      { url: 'UnitFormMain.html', revision: '1' },
      { url: 'Finifuzz.png', revision: '1' },
 ]);

 workbox.routing.registerRoute(
  new RegExp('.*\.js'),
  workbox.strategies.networkFirst()
 );
 workbox.routing.registerRoute(
  // Cache CSS files
  /.*\.css/,
  // Use cache but update in the background ASAP
  workbox.strategies.staleWhileRevalidate({
   // Use a custom cache name
     cacheName: 'css-cache',
  })
 );
 workbox.routing.registerRoute(
  // Cache image files
  /.*\.(?:png|jpg|jpeg|svg|gif)/,
  // Use the cache if it's available
  workbox.strategies.cacheFirst({
   // Use a custom cache name
   cacheName: 'image-cache',
      plugins: [
        new workbox.expiration.Plugin({
     // Cache only 20 images
     maxEntries: 20,
     // Cache for a maximum of a week
     maxAgeSeco)
   ],
  })
 );
} else {
  console.log(`Boo! Workbox didn't load ??`);
}

console.log('Hello from sw.js');
```

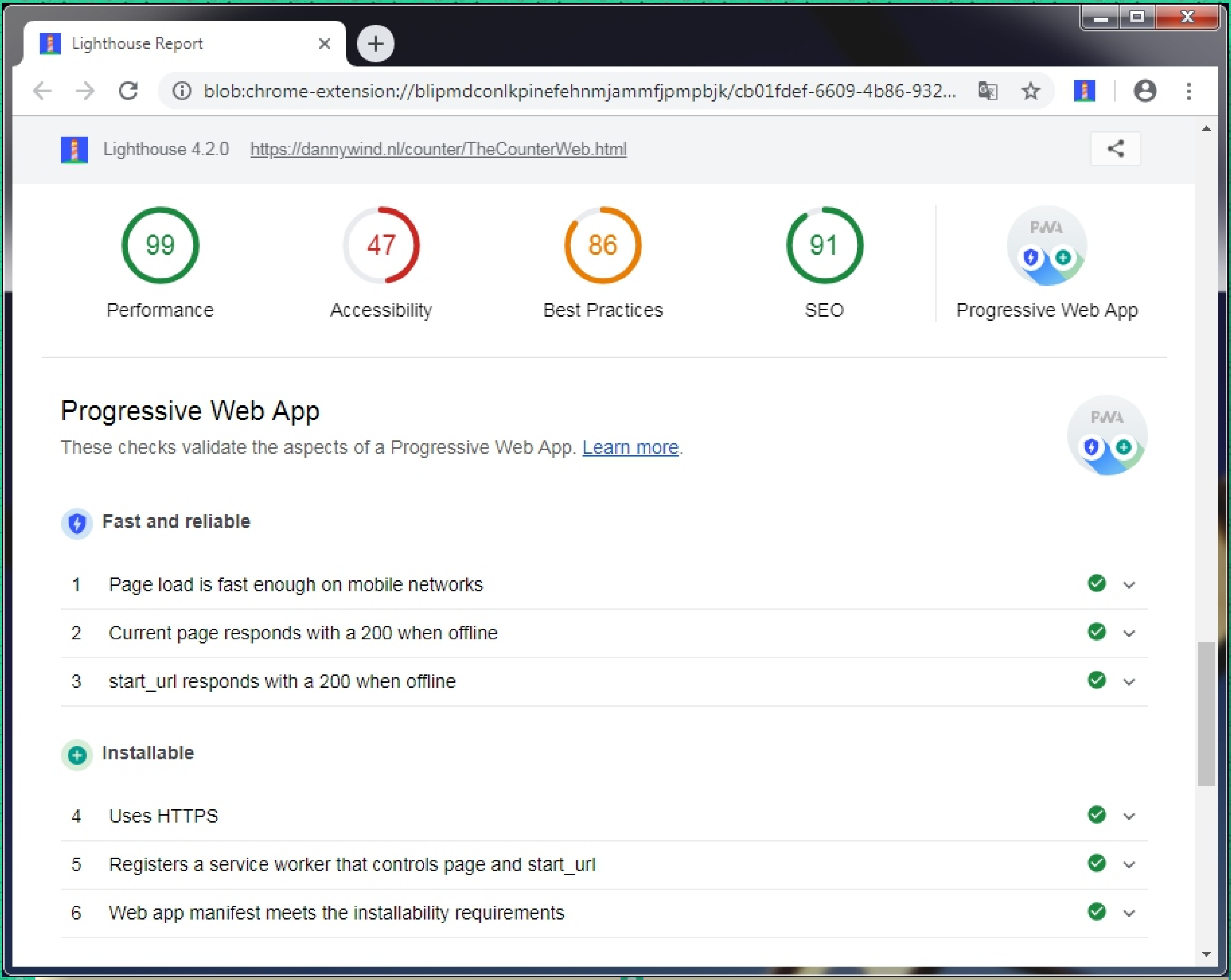
Here you add the files that are part of your web app, and need to be downloaded to the device if you want to run the web app offline.
Next you register routes that the service worker will use to refresh or reload these and other files.
For instance the CSS file is actually a link to the **Bootstrap CSS** file.
This is probably already cached by the browser, as a lot of online web pages use bootstrap.
But if you add your own CSS files, you'd still want them to be refreshed, so we use a **`staleWithRevalidate`** to make sure the page loads quickly from the local cache and then does a background re-load of the CSS file if needed.
Add some links to the mainifest and service worker in the HTML file.



```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="theme-color" content="#5BC0DE"/>
  <link rel="stylesheet"
   href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="icon" href="data:;base64,=">
  <link rel="manifest" href="manifest.json">
  <title>The Counter Web</title>
  <script type="text/javascript" src="TheCounterWeb.js"></script>
  <style>
  </style>
 </head>
 <body>
 </body>
 <noscript>
  <p>This Web App requires JavaScript which is currently disabled.</p>
 </noscript>
 <script>
 // Check that service workers are registered
 if ('serviceWorker' in navigator) {
  // Use the window load event to keep the page load performant
  window.addEventListener('load', () => {
       navigator.serviceWorker.register('sw.js');
  });
 }
 </script>
 <script type="text/javascript">
  rtl.run();
 </script>
</html>
```

Apart from the manifest.json and sw.js link there is also a **<noscript>** block that instructs the browser that this code really needs JavaScript to do its work, and which shows a text message if its not there.

There are a number of other requirements you need to fulfill to be PWA compliant, the main one being you need to use HTTPS.
Just use Let's Encrypt to get a SSL certificate for your domain and go from there.
The other requirements can be tested with the **Lighthouse plugin**. This plugin analyzes your PWA web page and determines if its good enough to be installed as a PWA when the user opens it in Chrome.


Lighthouse Report
blob:chrome-extension://blipmdconlkpinefehnmjammfjpmpbjk/cb01fdef-6609-4b86-932...
Lighthouse 4.2.0
https://dannywind.nl/counter/TheCounterWeb.html
99
Performance
47
Accessibility
86
Best Practices
91
SEO
PWA
Progressive Web App

Progressive Web App

These checks validate the aspects of a Progressive Web App. Learn more.

Fast and reliable

1 Page load is fast enough on mobile networks ✓

2 Current page responds with a 200 when offline ✓

3 start_url responds with a 200 when offline ✓

Installable

4 Uses HTTPS ✓

5 Registers a service worker that controls page and start_url ✓

6 Web app manifest meets the installability requirements ✓

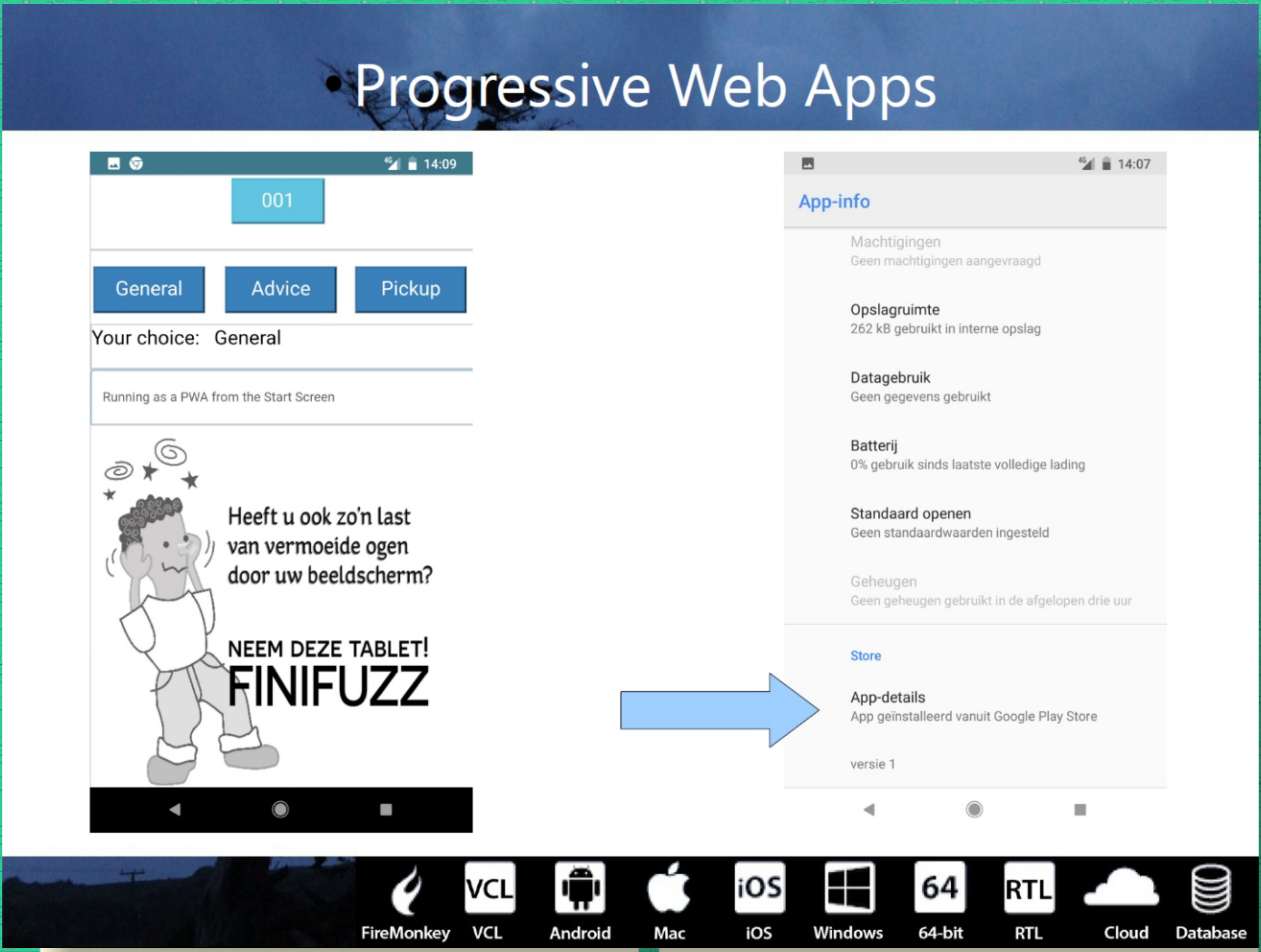
Figure 5: Running Lighthouse in Chrome to test PWA compliance

A little while ago **Bruno Fierens** from TMS asked me if I thought it would be a good idea to add these steps into a PWA wizard, and of course I said “Yes please!”.

In the most recent version of **TMS WEB Core** all these PWA steps have been integrated into a wizard, greatly simplifying the process.
Thank you Bruno!

For more background info on PWA please take a look at the Google Developers Doc
https://developers.google.com/web/progressive-web-apps/

After you've passed the PWA requirements, the end-user, when he visits your web-page will be asked to install it as a web app. After that the app will run **full screen**, and if you look at the App details it will even look like it' s installed from the App Store.
Also notice the size of the App, its just 262 Kb.



• Progressive Web Apps

Your choice: General

Running as a PWA from the Start Screen

App-info

Machtigingen
Geen machtigingen aangevraagd

Opslagruimte
262 kB gebruikt in interne opslag

Datagebruik
Geen gegevens gebruikt

Batterij
0% gebruik sinds laatste volledige lading

Standaard openen
Geen standaardwaarden ingesteld

Geheugen
Geen geheugen gebruikt in de afgelopen drie uur

Store

App-details
App geïnstalleerd vanuit Google Play Store

versie 1

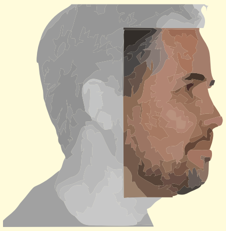
Figure 6: PWAs look and act like real apps

FINAL THOUGHTS

Small and scalable **REST microservices** are ideal for small companion apps, be it native or web based.

With PWA web apps you get the best of both, zero effort installation combined with native app look and feel.
Combine microservices and PWAs and you get easy to develop and deploy apps that can work stand-alone or be a complement to your existing desktop appplications.
Use them now to unlock all that potential of engaging the customers of your customers in the business process.
Also, its a lot of fun creating these PWA's.

Full download and video available at
https://dannywind.nl/delphi/coderage2018/



Danny Wind is a trainer, consultant and Delphi MVP. His favorite subjects are FireMonkey, User Interfaces and Parallel Programming. Meet Danny as a speaker on Developer Days, at SDN conferences, and other Delphi events in the Netherlands and Belgium.

OBJECT PASCAL ON RASPBERRY

How to install Free Pascal & Lazarus on a RaspBerry Pi 2/3 model B

By Rik Smit with technicical help from Don Wilbrink

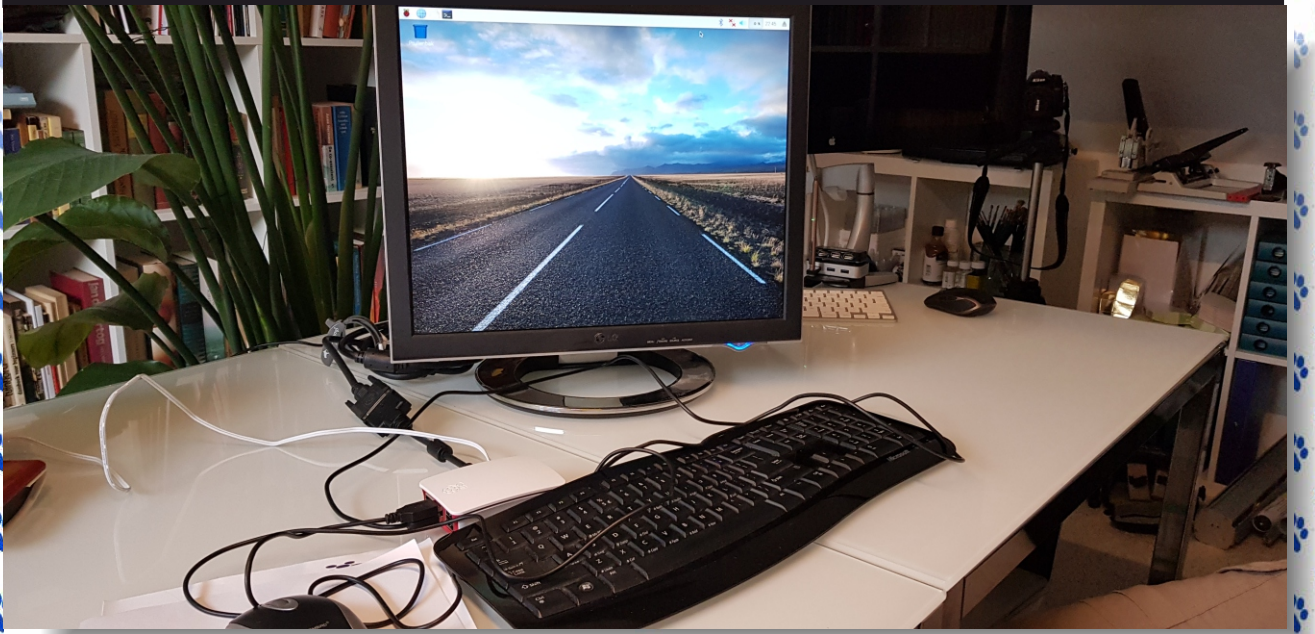


Figure 1: Raspberry Pi the test environment

INTRODUCTION

Naturally curious about the many applications possible with Object Pascal, this magazine's editor, Detlef Overbeek, commissioned **Rik Smit** to install and test the latest **Lazarus IDE version (v2.1.0) on a Raspberry Pi** single-board computer roughly the size of a credit card.

This article documents Rik's approach and findings so each reader can follow along to success in developing a wide range of applications running on this popular and inexpensive (yet quite powerful) mini computing platform.

Rik's efforts here using the latest Raspberry Pi expose both the reader as well as Object Pascal to a rich new world of IoT devices. Here, the **Internet of Things (IoT)** offers endless possibilities to retool and extend existing Object Pascal applications as well as to take on new projects. Using the **Pi's IoT** connectivity, as Rik documents herein, Object Pascal can control and coordinate many new and evolving sensor and automation technologies.

Employing the free Lazarus IDE atop the very inexpensive Pi single-board computer (costing less than €30), a pocket-sized solution can be had to do anything from operating your solar collectors to sending a text or email message the moment a solar flare occurs or your doorbell rings, and much more.

Working with open source software and hardware (like the Lazarus IDE and Raspberry Pi used herein), workforce capabilities can be strengthened and barriers to innovation lowered. Through such non-proprietary technologies, the future is made truly accessible to all people without need for excessive government mandates or commercial self-interest.

Note that this article was created with no prior experience working with the Raspberry Pi, nor any knowledge of Linux. As I like to develop with Pascal, I thought that this undertaking Detlef charged really shouldn't be all that difficult.

Fortunately open source brought us the Internet and I thank everyone who shares ideas and snippets of code thereon. I must also thank **Don Wilbrink** whose loads of patience and truly inexhaustible wealth of knowledge regarding Linux and the Raspberry Pi made this article possible.

This article is intended to be comprehensive; in particular, it targets readers who are buying a first Raspberry Pi and have little or no experience with Linux. For the more experienced among you, the complete script appears at the end of the article.

BEGIN

What's needed to get started?

- **Raspberry PI 2 or 3 Model B** (Version 1 is excluded because it has too many limitations)
- And a **suitable housing**, safety for everything. (risk of short circuits)
- A minimum of **16 GB micro SD card**, Class 10. (Note: 8 Gb is possible, but when installing Lazarus often gives disk space problems. The final installation will take slightly more than 6 GB disk space!)
- **HDMI (*)1 compatible monitor, or an older other (Digital / analogue)** and then use a HDMI to analogue / DVI (*2)adapter plug

END;

(*1) HDMI (High-Definition Multimedia Interface) is a proprietary audio/video interface for transmitting uncompressed video data and compressed or uncompressed digital audio data from an HDMI-compliant source device, such as a display controller, to a compatible computer
monitor, video projector, digital television, or digital audio device. HDMI is a digital replacement for analog video standards.

WIKIPEDIA

(*2) Digital Visual Interface (DVI) is a video display interface developed by the Digital Display Working Group (DDWG). The digital interface is used to connect a video source, such as a video display controller, to a display device, such as a computer monitor. It was developed with the intention of creating an industry standard for the transfer of digital video content.

- USB mouse, may be wireless
- USB keyboard, may also be wireless
- Power supply: 2.5A, 5.1V, micro USB B connection
- or possibly a battery charger, 4 AA batteries
  You can even attach a power supply - at least 2A - for a mobile. Of course you can hang on to it even more, but the nutrition must be powerful enough.
- Internet access: with PI 2 via cable, after all there is no onbaord wifi. With PI 3 B this is also possible via onboard WiFi. That appears to be fast enough.
- A clean OS system on the PI:
  In our case we use Raspbian

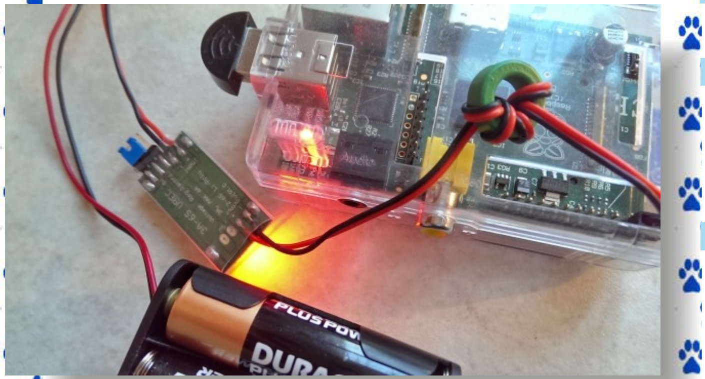


Figure 2: Raspberry Pi version 1 battery charged

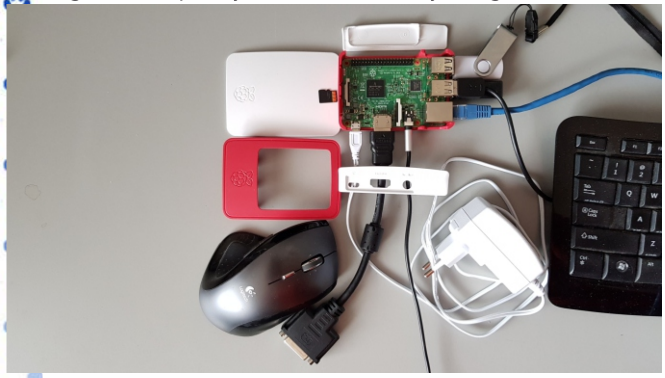


Figure 3: This is all we need

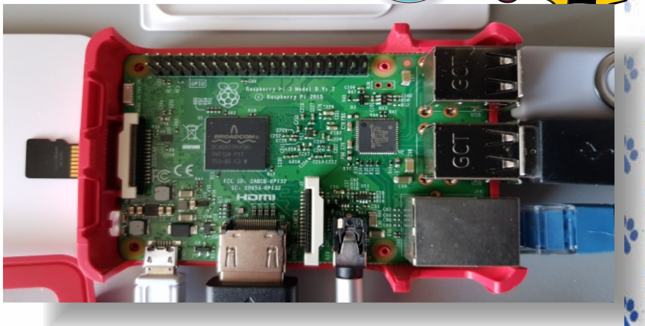


Figure 4 Raspberry Pi version 3 Model B V1.2 2015

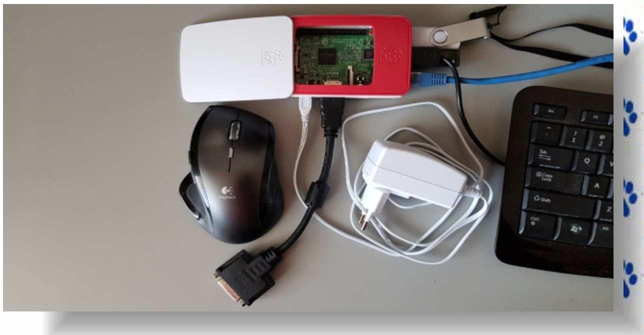


Figure 5: Raspberry Pi version 3 Model B V1.2 2015

STEP BY STEP PROCEDURE

1. Install Raspbian OS.
2. Get to know the Pi and find out what we all need to be set up - configuration work.
3. Retrieve a compiled (ready-made) FreePascalCompiler. See explanation **3, FREEPASCAL COMPILER** article 14/40 issue page 77
4. Retrieve files from the latest FPC version that can be set up with the SVN.

SVN, license for Apache Subversion, a software versioning and revision control system. See for this point the detailed explanation under number 4 on page 21. (4. Get the latest Free Pascal Compiler packages)

5. With the retrieved ready-made FPC version, the so-called SEED COMPILER, the new FPC compiler must be compiled itself. **(Bootstrapping)**

In general, bootstrapping usually refers to a self-starting process that is supposed to proceed without external input. In computer technology the term (usually shortened to booting) usually refers to the process of loading the basic software into the memory of a computer after power-on or general reset, especially the operating system which will then take care of loading other software as needed. The term appears to have originated in the early 19th-century United States (particularly in the phrase "pull oneself over a fence by one's bootstraps") to mean an absurdly impossible action, an adynaton.

6. Retrieve Lazarus files and required extra Lazarus packages.
7. Create Lazarus (LCL + IDE) using the self-compiled FPC compiler.
8. Post build steps. (resetting initialization steps) See 2-D

SET UP A CLEAN SYSTEM

To get the RaspBerry Pi to work, an operating system must be installed. You can download it via the Raspberry Pi download page:
`https://www.raspberrypi.org/downloads/`
There you can choose from NOOBS (easy installer for Raspbian) or two flavors (little or extensive) RaspBian. For this project, the **`RaspBian-Stretch.zip`** file from November 2018 with desktop and recommended software based on Debian Stretch has been retrieved. You can also use the latest **`NOOBS.zip`**. This includes even more options. The retrieved zip file contains an image that must be flashed to the SD card. Overwrite the SD card with new data.

WRITE THE IMG TO THE SD-CARD

A. → Unzip the 2018-11-13-raspbian-stretch.zip image file to your PC.

B. → You can then use an image writer (Win32DiskImager) - downloadable from **`https://sourceforge.net/projects/win32diskimager/`**, to flash the obtained image to the SD card.

C. → Next we need to select the Image File in Win32 Disk Imager and click the “Write” button and select YES. This process will take a couple of minutes, depending on the size of your IMG file and the speed of your computer, SD-Card reader and SD-Card. Then check the image with the control key and leave the imager if everything went well.

NOTE: Win32 Disk Imager does not seem to like IMG files stored on network shares, copy the IMG to your computer first if that’s the case.
NOTE: Make sure not to pull the SD-Card out of your computer without a proper Eject – some systems might have a write delay, and removing the card without following the proper eject procedure might corrupt the data on the card.

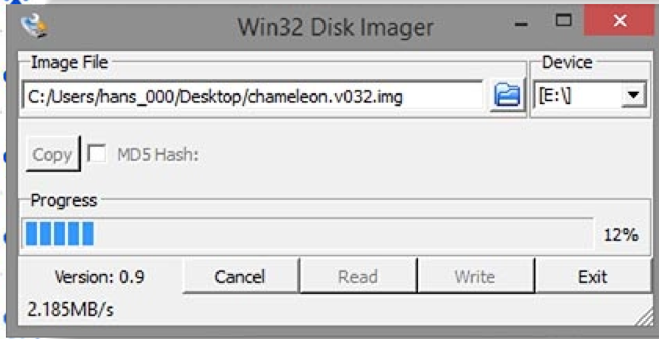

Win32 Disk Imager

Image File: C:/Users/hans_000/Desktop/chameleon.v032.img
Device: [E:\]
□ MD5 Hash:
Progress: 12%
Version: 0.9
2.185MB/s

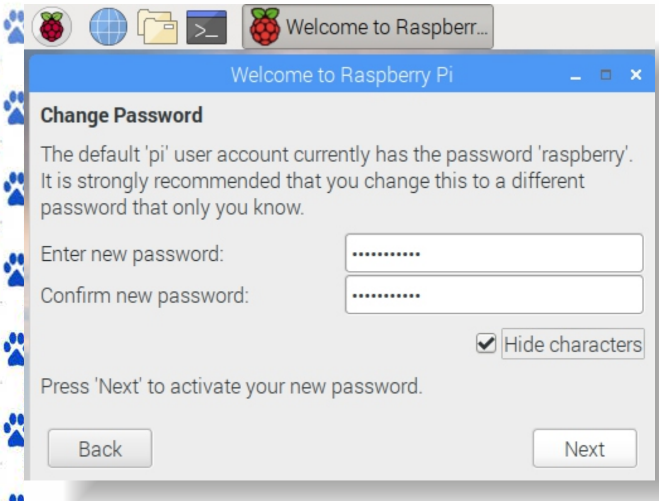
Figure 6: Writing an IMG file to an SD-Card

NOTE: Needless to say: just copying the img file to the SD card really doesn't work for an image file! It must be copied bit for bit.

BOOT AND LOGIN

1 → Place the prepared SD card in the SD slot on the bottom of the Raspberry Pi board, connect monitor (HDMI port), keyboard and mouse (USB) and if necessary connect an internet cable.

2 → Only then do you connect the power supply to the card. There is no on / off button on the RaspBerry Pi and it will now start up immediately for the first time. (Ergo: you can switch it off by calling a software shutdown and then - if nothing is blinking on the RaspBerry Pi board - removing the mains voltage).


Welcome to Raspberry Pi

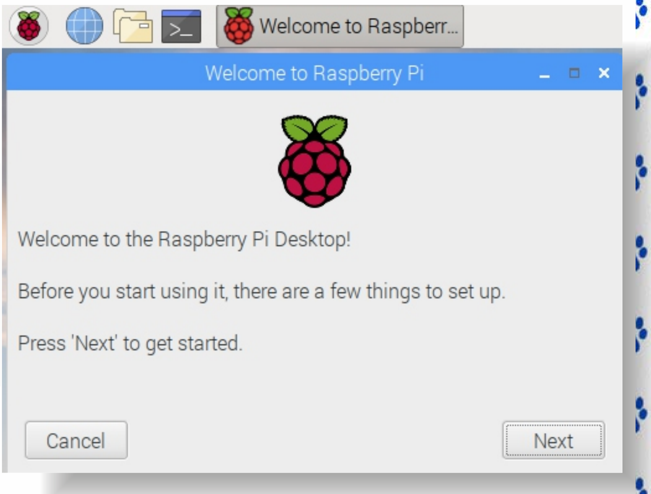
Change Password

The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know.

Enter new password: •••••••••••
Confirm new password: •••••••••••
☑ Hide characters

Press 'Next' to activate your new password.

Figure 7: Change password

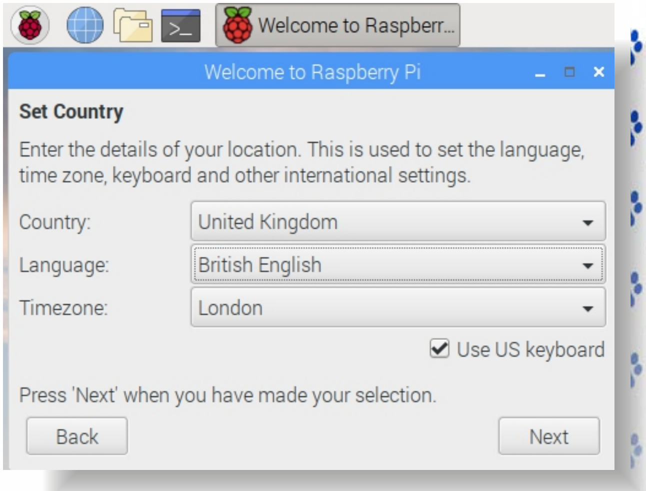

Welcome to Raspberry Pi

Welcome to the Raspberry Pi Desktop!

Before you start using it, there are a few things to set up.

Press 'Next' to get started.

Figure 8: Next


Welcome to Raspberry Pi

Set Country

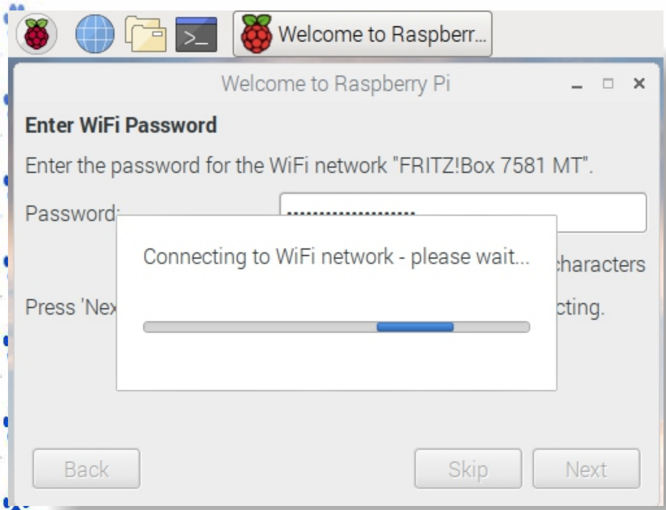
Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

Country: United Kingdom
Language: British English
Timezone: London
☑ Use US keyboard

Press 'Next' when you have made your selection.

Figure 9: Boot and Login

3 . → You must select a national language during the first start-up. In this window you can also immediately put a check mark if your keyboard has a USA layout. Then enter a new user and password (default is: user pi, password raspberry) and for the model RaspBerry Pi (3) with WiFi enter the password for the WiFi connection.



Enter WiFi Password

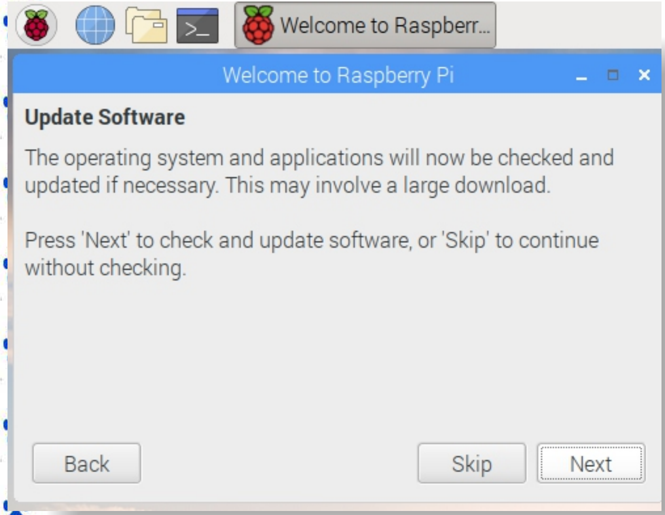
Enter the password for the WiFi network "FRITZ!Box 7581 MT".

Password: ....

Connecting to WiFi network - please wait...

Press 'Nex ... haracters ... cting.

Figure 10: Connecting

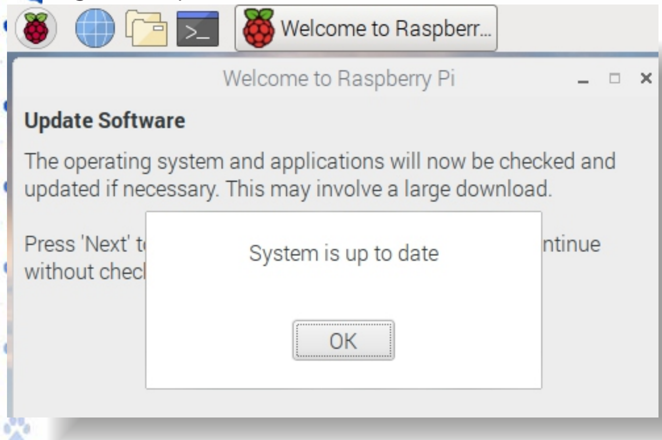


Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' to check and update software, or 'Skip' to continue without checking.

Figure 11: Update



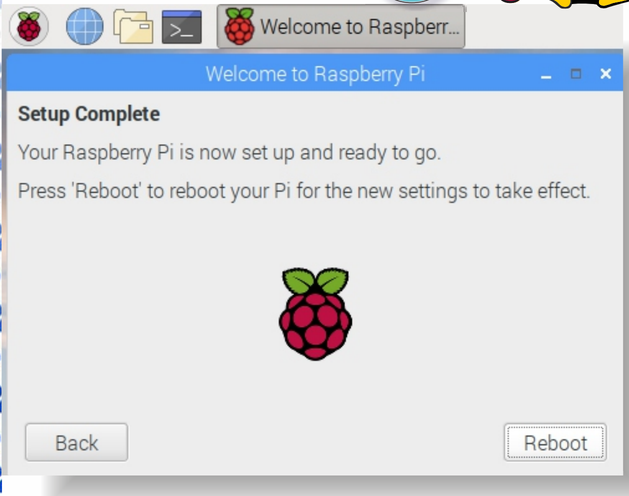
Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' t ... ntinue without checl

System is up to date

Figure 11: System is up to date



Setup Complete

Your Raspberry Pi is now set up and ready to go.

Press 'Reboot' to reboot your Pi for the new settings to take effect.

Figure 12: Complete

At this point, the most recent files are automatically retrieved and the operating system is installed on the RaspBerry Pi.

NOTE: That really takes time. So be patient and have a cup of coffee: it really won't be the last for today.

INFO: Once the OS has been installed you can still adjust these and other settings via the Raspberry Pi Configuration program.
This can be found via the main pulldown menu (***raspberry icon in the top left of the screen***), option preferences -> Raspberry Pi Configuration, see figure 15.

NOTE
Generally:

as **spaces are not supported under Linux or Raspbian without quotes,** the document shows paths without spaces.

Figure 13:

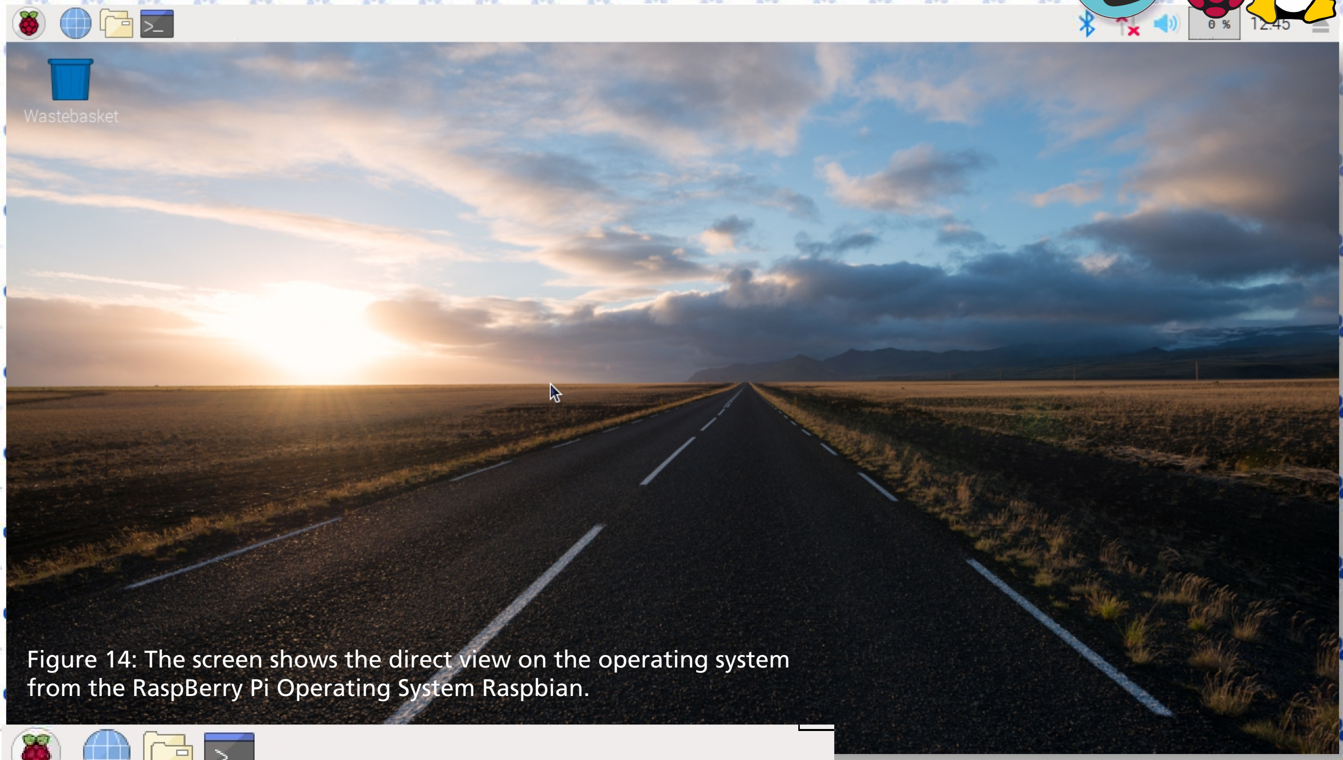


Figure 14: The screen shows the direct view on the operating system from the RaspBerry Pi Operating System Raspbian.

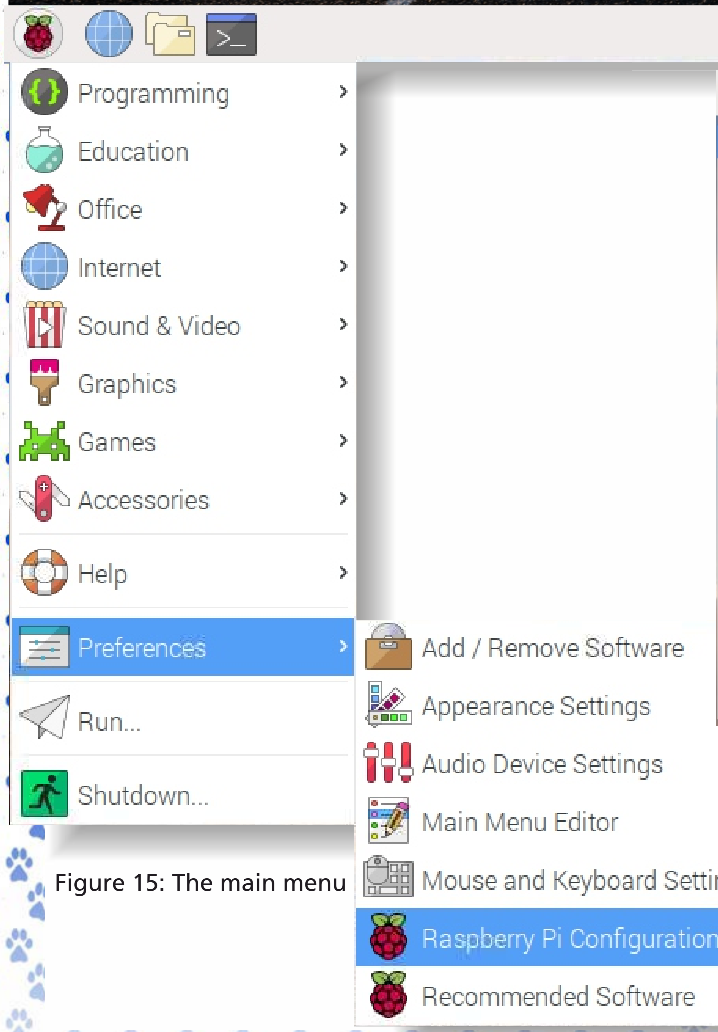
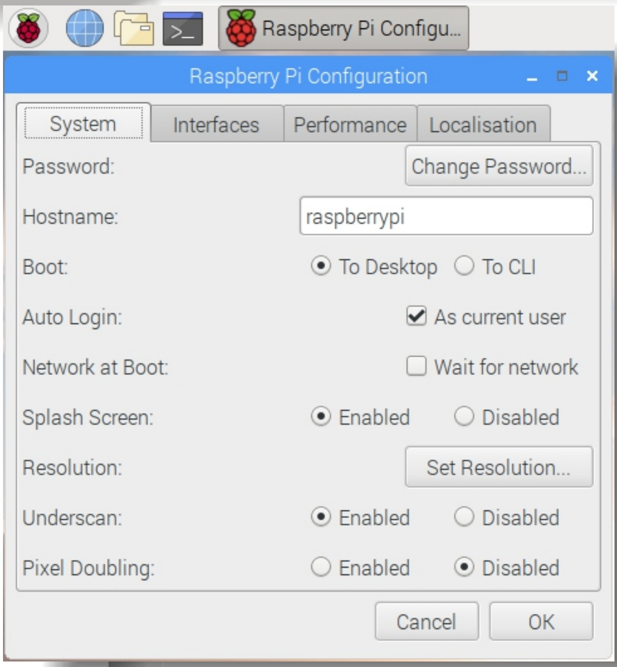


Figure 15: The main menu



Password:

Hostname: raspberrypi

Boot: ● To Desktop ○ To CLI

Auto Login: ☑ As current user

Network at Boot: □ Wait for network

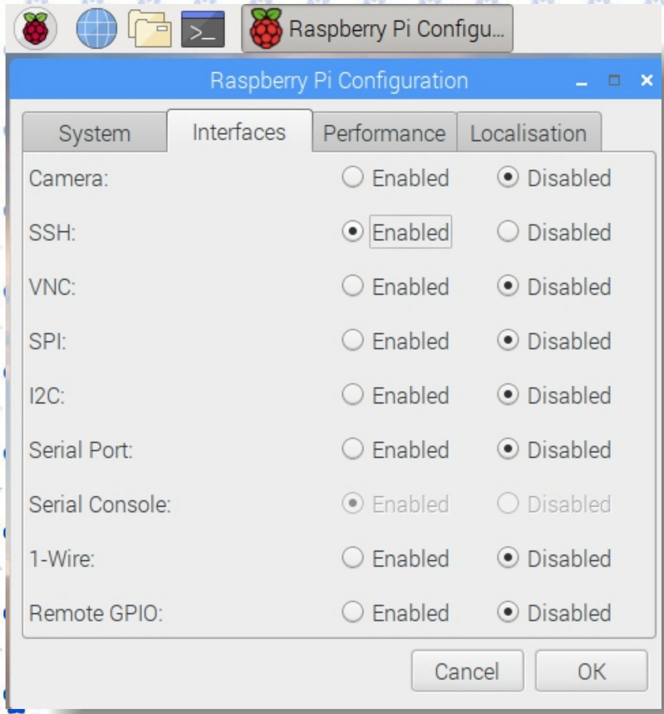
Splash Screen: ● Enabled ○ Disabled

Resolution:

Underscan: ● Enabled ○ Disabled

Pixel Doubling: ○ Enabled ● Disabled

Figure 16: System



Camera: ○ Enabled ● Disabled

SSH: ● Enabled ○ Disabled

VNC: ○ Enabled ● Disabled

SPI: ○ Enabled ● Disabled

I2C: ○ Enabled ● Disabled

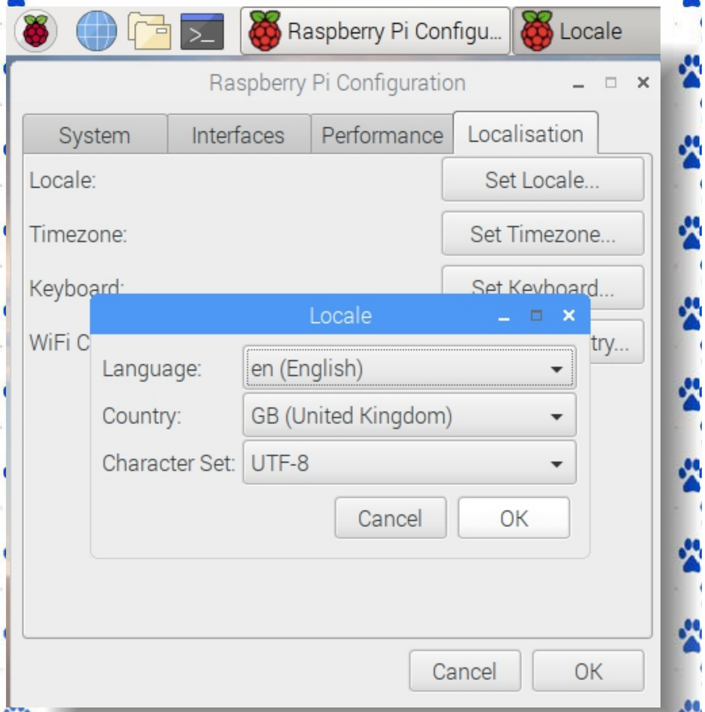
Serial Port: ○ Enabled ● Disabled

Serial Console: ● Enabled ○ Disabled

1-Wire: ○ Enabled ● Disabled

Remote GPIO: ○ Enabled ● Disabled

Figure 17: SSH Enabled



Language: en (English)

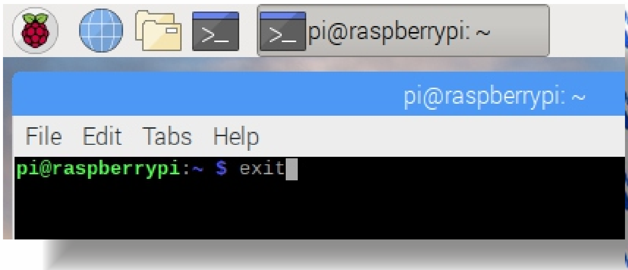
Country: GB (United Kingdom)

Character Set: UTF-8

Figure 18: Localizing UK

2. RASPBERRY PI - THE RASPBIAN OPERATING SYSTEM

Before we start working on the Free Pascal Compiler (FPC) and Lazarus we will explore Raspbian OS to get ourselves acquainted with it and we'll do that via a so-called terminal (a window into the mind's Pi:). You can open as many terminals as you want at the same time.



```
pi@raspberrypi:~ $ exit
```

Figure 19: The exit command in the terminal

How do you open a terminal? One of the ways to do that is to click on the main menu "terminal" icon at the top of the screen. You will come across another way later. You can close by clicking on the cross in the top right corner, or by typing EXIT in the terminal itself and closing with a press of the enter key:

```
exit <enter>
```

The terminal will become your working environment.

CONVENTIONS

NOTE: it is necessary to proceed according to the following agreements:
Enter the lines in this article with the `purple text` in the terminal window, followed by an enter key `<enter>`, paying particular attention to any space(s): `<space>`.
Of course, most assignments in the terminal can become shorter or even combined into one, but for the sake of clarity we each write them on their own line.
To get used to it, go and get some information and adjust all the issues of the OS RaspBian.

First request the version type (A)
and the ip address (B) of the RaspBerry Pi.
Then upgrade the system (C) to the latest files and expand the (D) SWAP memory.

A. RASPBERRY PI TYPE VERSION

To find out which RaspBerry Pi hardware you are using, you must use the cat … command. in an open terminal. NOTE: Remember that Linux is case sensitive: MyDir is really a different directory than mydir! Also pay attention to where the spaces <space> are. Sometimes for extra clarity in the command line where a space belongs, the word <space> is written and closed with <enter>, such as here: **`cat`** <space> **`/proc/cpuinfo`** <enter> so not **`cat / proc / cpuinfo`** <enter> We will also briefly discuss the used commands and any .extension options

`Info about cat`

`syntax: cat [OPTION…] [FILE…]`

function: it sends the specified file or files one after the other to the standard output.

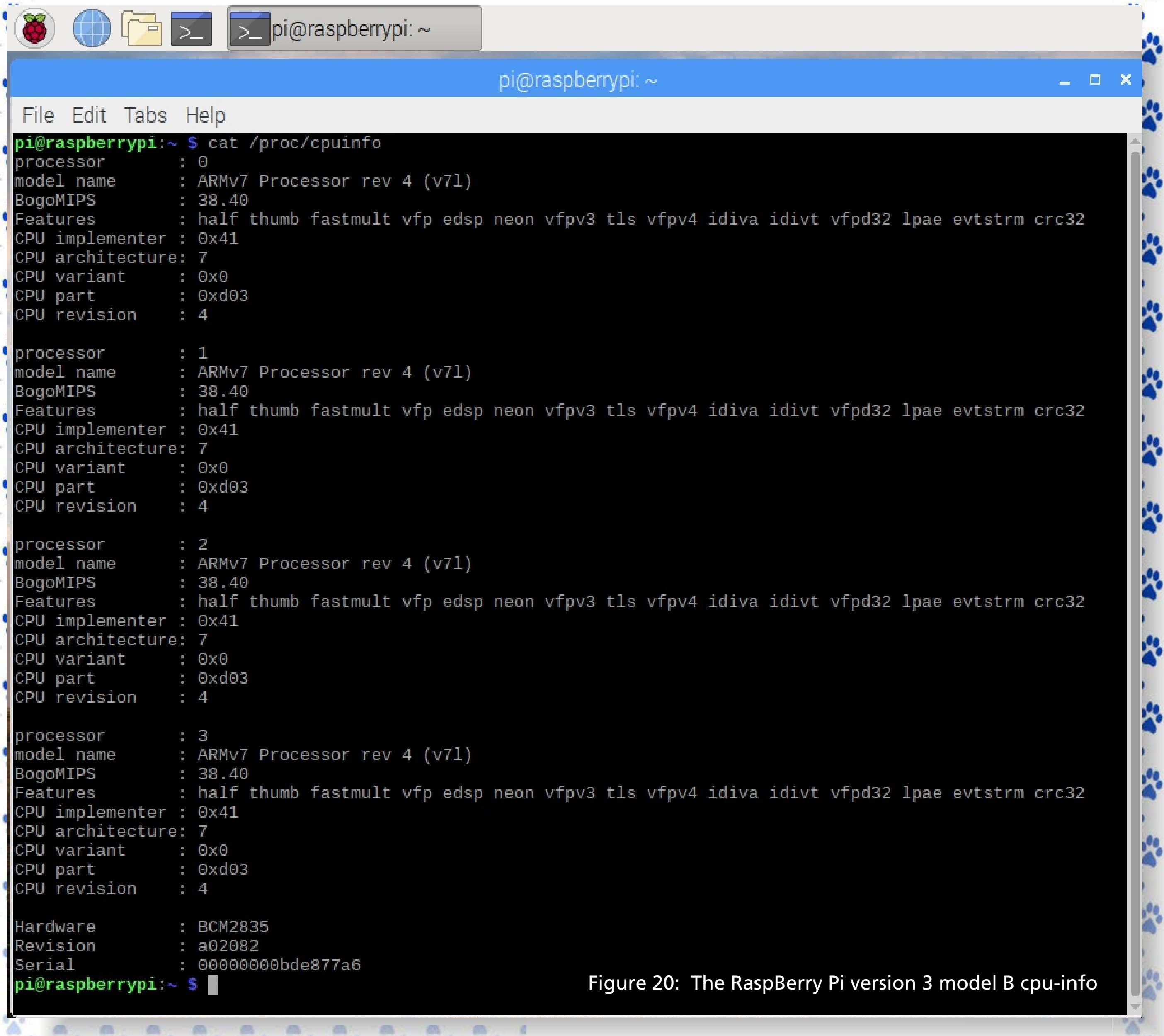
`Tip` *The --help after cat: cat <space> --help option provides a complete help overview.*

You can also often execute this option with other commands if you want more information about this.

`Info` *The first / (forward slash) in the specified path represents the root of the file system.*

The file system of Debian consists of a single tree structure with / as a start. Every user gets their own entry: for the pi user that is / home / pi.

The output of this command in the terminal shows you 4 blocks of text (one for each core processor).



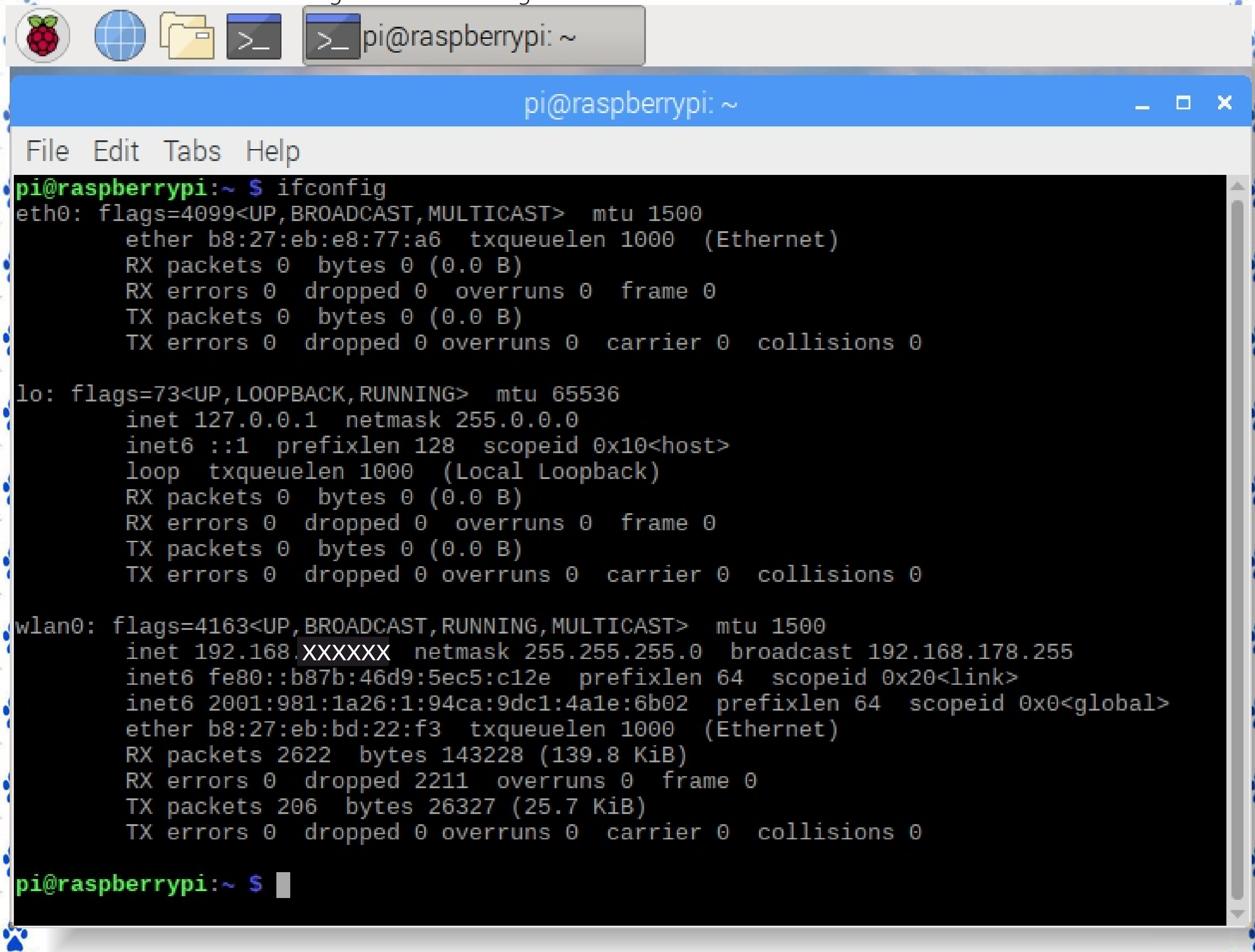
```
pi@raspberrypi:~ $ cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 4 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 1
model name      : ARMv7 Processor rev 4 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 2
model name      : ARMv7 Processor rev 4 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 3
model name      : ARMv7 Processor rev 4 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

Hardware        : BCM2835
Revision        : a02082
Serial          : 00000000bde877a6
pi@raspberrypi:~ $ 
```

Figure 20: The RaspBerry Pi version 3 model B cpu-info

B. THE PI'S IP ADDRESS

Another useful thing to know is your Pi's IP address. Enter the following in the terminal:
ifconfig <enter>

You will then see something like the following:



```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:e8:77:a6  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.XXXXXX  netmask 255.255.255.0  broadcast 192.168.178.255
        inet6 fe80::b87b:46d9:5ec5:c12e  prefixlen 64  scopeid 0x20<link>
        inet6 2001:981:1a26:1:94ca:9dc1:4a1e:6b02  prefixlen 64  scopeid 0x0<global>
        ether b8:27:eb:bd:22:f3  txqueuelen 1000  (Ethernet)
        RX packets 2622  bytes 143228 (139.8 KiB)
        RX errors 0  dropped 2211  overruns 0  frame 0
        TX packets 206  bytes 26327 (25.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $ 
```

Figure 21: ifconfig

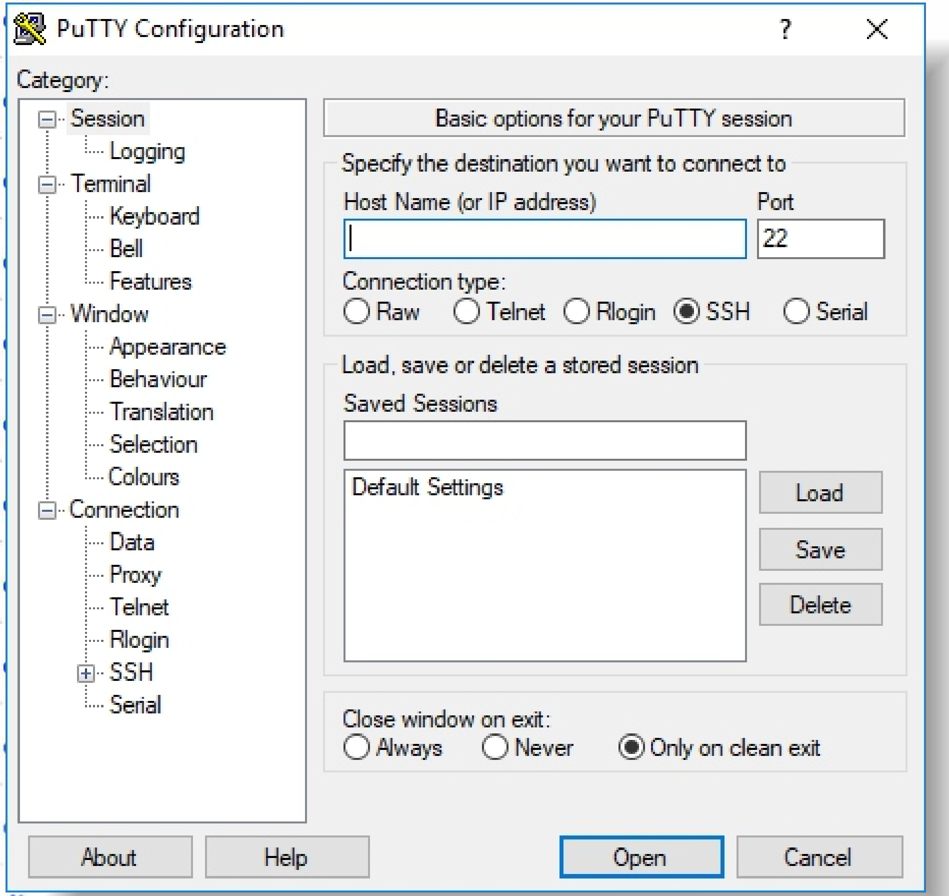
The IP address can be read from this. In my case: inet addr: **192.168.xxx.xxx.**
If your RaspBerry Pi is connected to a (network) cable, you will see your ethernet adapter behind eth0.

In addition to requesting the Ethernet / IP (*1) address, you can also enable the SSH (*2) connection on the RaspBerry Pi. You can do that again via the Raspberry Pi Configuration program.
You can use this to hang your RaspBerry Pi on your favorite PuTTY (*3) (**https://www.putty.org/**) or SSH client application on your PC to access the RaspBerry Pi from your computer.
Then an HDMI Monitor, USB Mouse and Keyboard are no longer needed to operate the RaspBerry Pi.
The advantage of this is that you can copy sequences of commands directly from the PC to the pi without having to tap them one by one on the pi. If you want to know more, read the following article:
"How to work with SSH connections (SSH Clients)":
https://www.tweaking4all.com/network-internet/ssh-clients/.

1* EtherNet/IP is an industrial network protocol that adapts the Common Industrial Protocol to standard Ethernet. EtherNet/IP is one of the leading industrial protocols in the United States and is widely used in a range of industries including factory, hybrid and process. The EtherNet/IP and CIP technologies are managed by ODVA, Inc., a global trade and standards development organization founded in 1995 with over 300 corporate members. EtherNet/IP uses both of the most widely deployed collections of Ethernet standards –the Internet Protocol suite and IEEE 802.3 – to define the features and functions for its transport, network, data link and physical layers. EtherNet/IP performs at level session and above (level 5, 6 and 7) of the OSI model. CIP uses its object-oriented design to provide EtherNet/IP with the services and device profiles needed for real-time control applications and to promote consistent implementation of automation functions across a diverse ecosystem of products. In addition, EtherNet/IP adapts key elements of Ethernet's standard capabilities and services to the CIP object model framework, such as the User Datagram Protocol (UDP), which EtherNet/IP uses to transport I/O messages.

2* Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.[1] Typical applications include remote command-line login and remote command execution, but any network servicecan be secured with SSH.

*3 PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers



Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

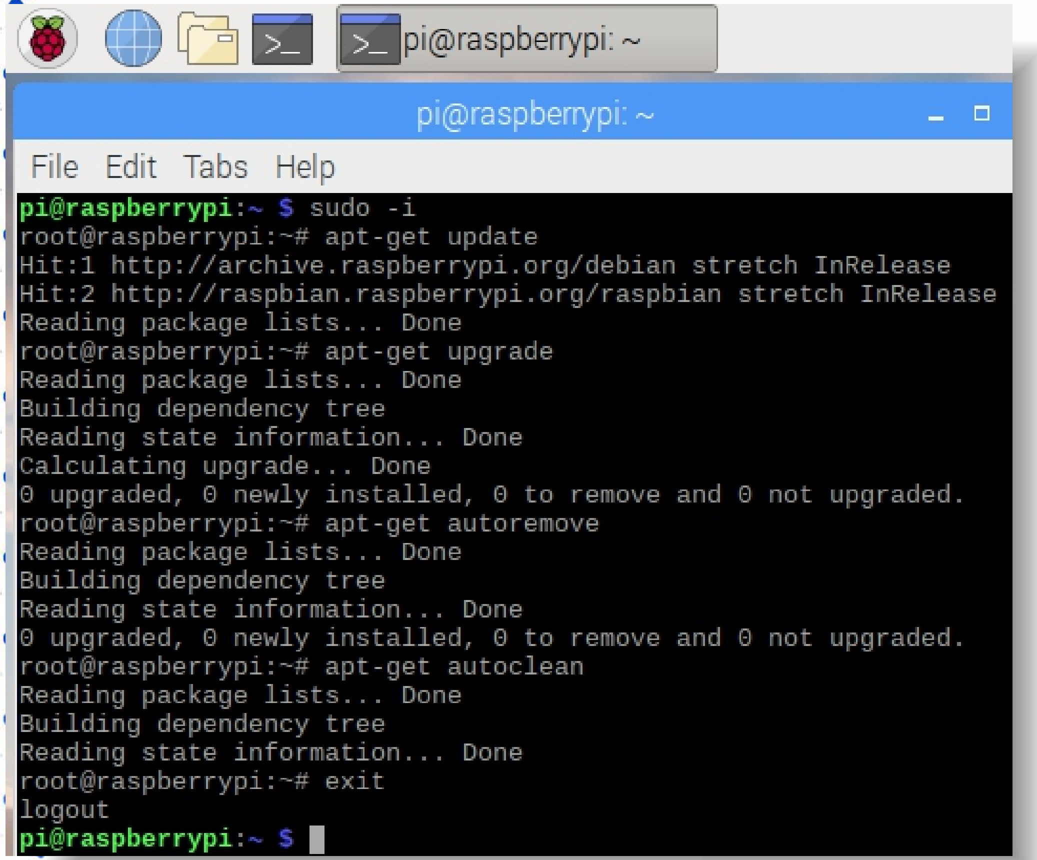
Figure 22: PuTTY

C. SYSTEM UPDATE

Before you continue, we will let the current system see if any updates can be retrieved. We must obtain the appropriate authorization for this. Under windows these are administrator rights.

On the next page is the sudo command.

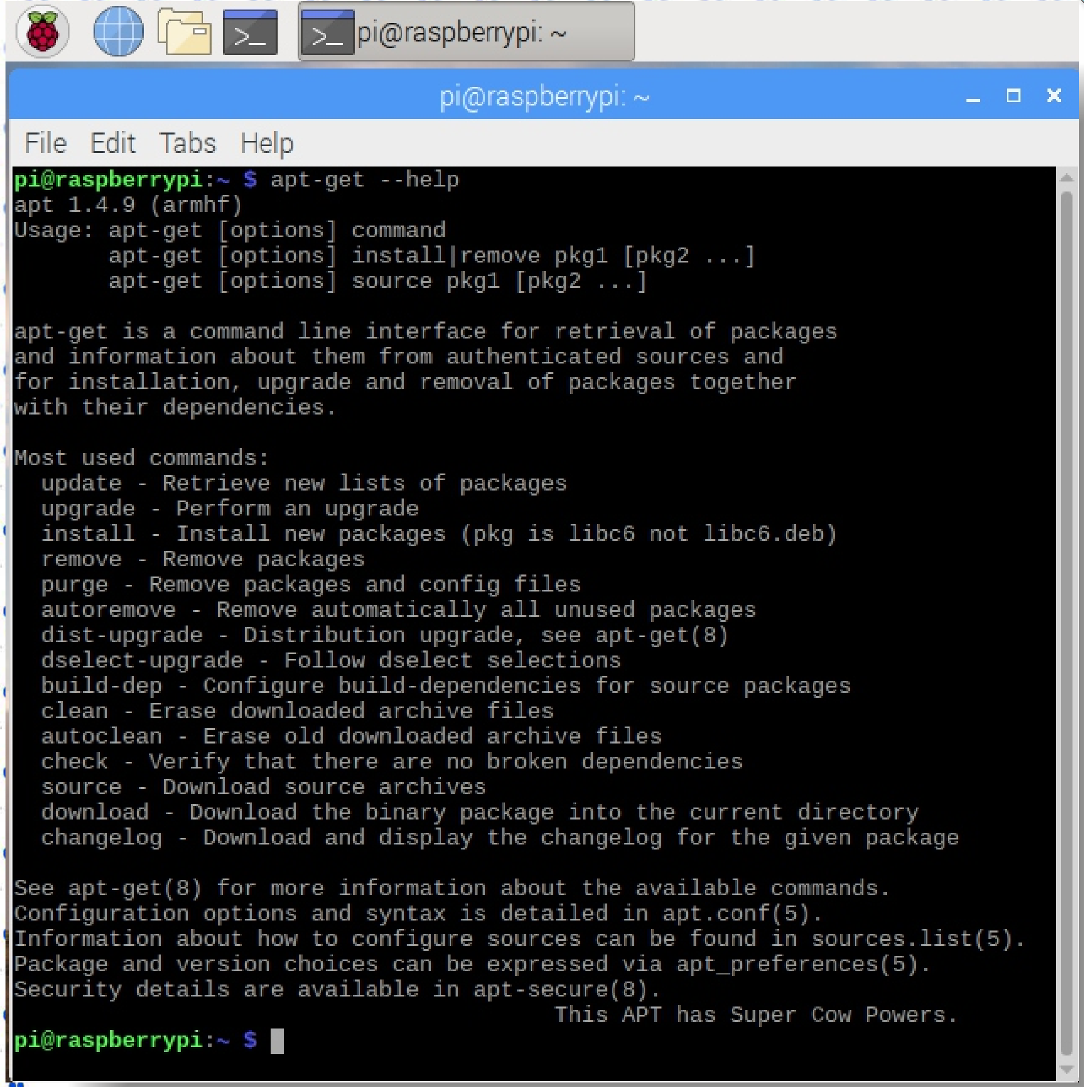
```
sudo <enter>
Info  sudo
      syntax: sudo [OPTION…] []
      function: root rights,execute a command as another user
sudo apt-get update <enter>          // retrieve new list of packages
sudo apt-get upgrade <enter>         // perform a upgrade
sudo apt-get autoremove <enter>      // remove automatically all unused packages
sudo apt-get autoclean <enter>       // erase old downloaded archive files
reboot

with OPTION -i [=login: run login shell as the target user]:
sudo -i <enter>
     apt-get update <enter>          // retrieve new list of packages
     apt-get upgrade <enter>         // perform a upgrade
     apt-get autoremove <enter>      // remove automatically all unused packages
     apt-get autoclean <enter>       // erase old downloaded archive files
exit                                 // exit from sudo
reboot
```



```
pi@raspberrypi:~ $ sudo -i
root@raspberrypi:~# apt-get update
Hit:1 http://archive.raspberrypi.org/debian stretch InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease
Reading package lists... Done
root@raspberrypi:~# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@raspberrypi:~# apt-get autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@raspberrypi:~# apt-get autoclean
Reading package lists... Done
Building dependency tree
Reading state information... Done
root@raspberrypi:~# exit
logout
pi@raspberrypi:~ $
```

Figure 23: sudo -i, apt-get update

```
Info  apt-get
      syntax:     apt-get [OPTION…] [COMMAND]
      functie:    package manager.
      Usage:      apt-get [options] command.
                  apt-get [options] install / remove pkg1 [pkg2 ...]
                  apt-get [options] source pkg1 [pkg2 ...]
```



```
pi@raspberrypi:~ $ apt-get --help
apt 1.4.9 (armhf)
Usage: apt-get [options] command
       apt-get [options] install|remove pkg1 [pkg2 ...]
       apt-get [options] source pkg1 [pkg2 ...]

apt-get is a command line interface for retrieval of packages
and information about them from authenticated sources and
for installation, upgrade and removal of packages together
with their dependencies.

Most used commands:
  update - Retrieve new lists of packages
  upgrade - Perform an upgrade
  install - Install new packages (pkg is libc6 not libc6.deb)
  remove - Remove packages
  purge - Remove packages and config files
  autoremove - Remove automatically all unused packages
  dist-upgrade - Distribution upgrade, see apt-get(8)
  dselect-upgrade - Follow dselect selections
  build-dep - Configure build-dependencies for source packages
  clean - Erase downloaded archive files
  autoclean - Erase old downloaded archive files
  check - Verify that there are no broken dependencies
  source - Download source archives
  download - Download the binary package into the current directory
  changelog - Download and display the changelog for the given package

See apt-get(8) for more information about the available commands.
Configuration options and syntax is detailed in apt.conf(5).
Information about how to configure sources can be found in sources.list(5).
Package and version choices can be expressed via apt_preferences(5).
Security details are available in apt-secure(8).
                                        This APT has Super Cow Powers.
pi@raspberrypi:~ $
```

Figure 24: apt-get --help

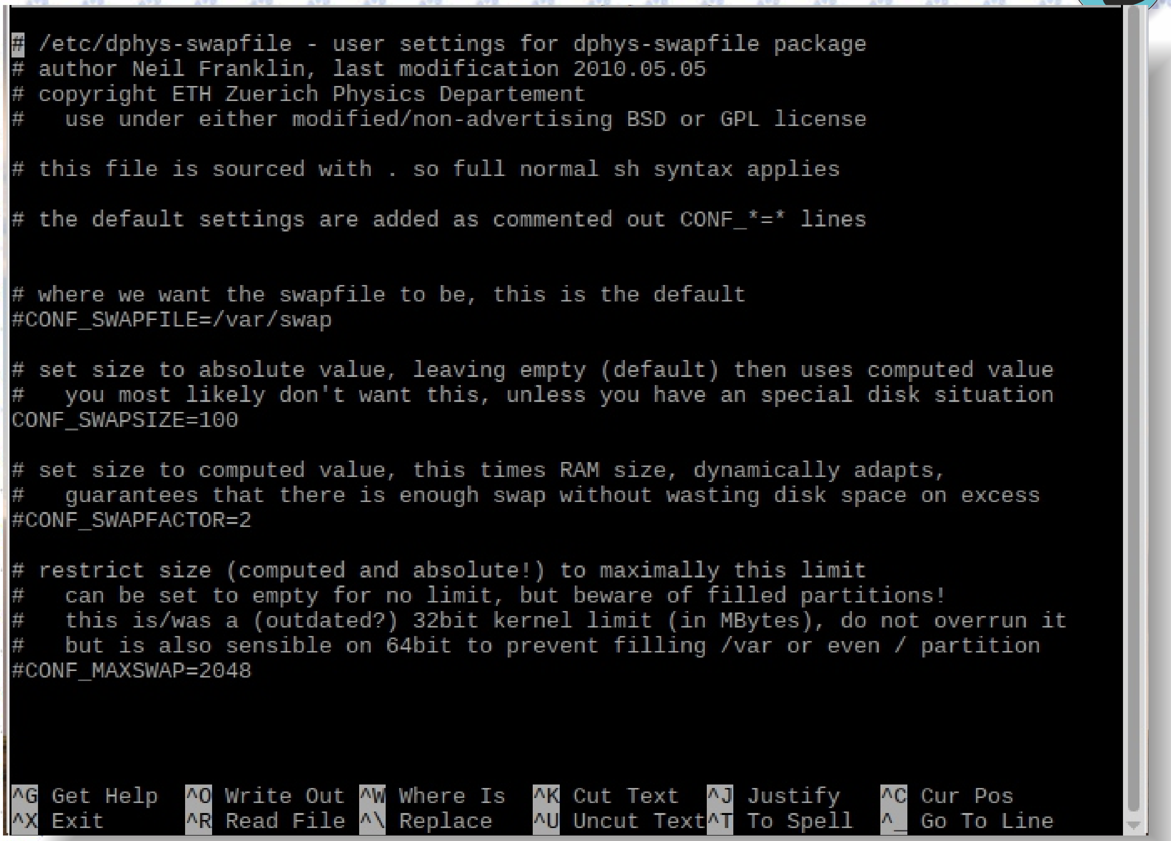
D. CONFIGURING RASPBIAN

The standard size of the swap file (*Central memory file*) is too small for what we want to do. So to prevent **"out of memory"** notifications when compiling, for example, Lazarus, we adjust this swap file using the supplied word processor **(NANO)** and as root user (sudo) in an opened terminal as follows:

```
sudo nano /etc/dphys-swapfile
```

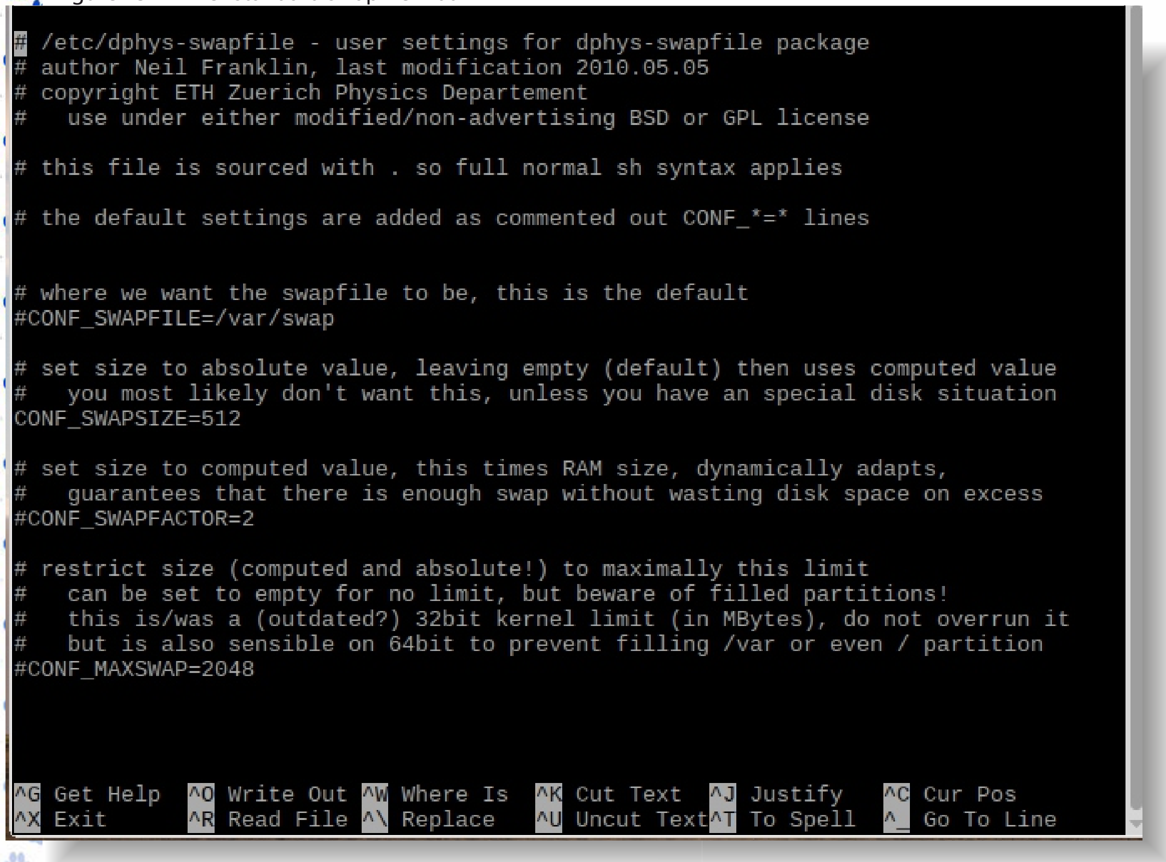
NOTE: Use the four directional arrow keys for navigation in the terminal.
If you have entered an order incorrectly and you are back at the prompt in the terminal, you can use the up arrow key to retrieve the previously entered commands in the terminal. Go with the left arrow to the place in the text where the error is, correct it and go back with the right arrow to the end of the line and then click the enter key. That can save a lot of unnecessary typing.
And then something: if you do nothing for a long time on the RaspBerry Pi then it goes to sleep: ergo screen turns black. Even when it performs a long-term listing. The process continues in the background. Moving the mouse briefly brings the screen back to life.

This gives a lot of rules in the NANO word processor. Search in that text for the line **`CONF_SWAPSIZE = 100`** and change the number 100 to 512.
Then press CNTRL + X to close NANO. Then choose Y / J when asked if the change you want should be saved and then press <enter>.



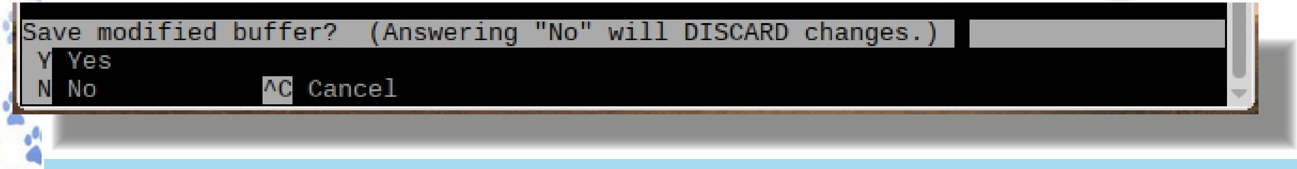
```
# /etc/dphys-swapfile - user settings for dphys-swapfile package
# author Neil Franklin, last modification 2010.05.05
# copyright ETH Zuerich Physics Departement
#   use under either modified/non-advertising BSD or GPL license

# this file is sourced with . so full normal sh syntax applies

# the default settings are added as commented out CONF_*=* lines


# where we want the swapfile to be, this is the default
#CONF_SWAPFILE=/var/swap

# set size to absolute value, leaving empty (default) then uses computed value
#   you most likely don't want this, unless you have an special disk situation
CONF_SWAPSIZE=100

# set size to computed value, this times RAM size, dynamically adapts,
#   guarantees that there is enough swap without wasting disk space on excess
#CONF_SWAPFACTOR=2

# restrict size (computed and absolute!) to maximally this limit
#   can be set to empty for no limit, but beware of filled partitions!
#   this is/was a (outdated?) 32bit kernel limit (in MBytes), do not overrun it
#   but is also sensible on 64bit to prevent filling /var or even / partition
#CONF_MAXSWAP=2048



^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Figure 25 NANO: standard swapfile=100



```
# /etc/dphys-swapfile - user settings for dphys-swapfile package
# author Neil Franklin, last modification 2010.05.05
# copyright ETH Zuerich Physics Departement
#   use under either modified/non-advertising BSD or GPL license

# this file is sourced with . so full normal sh syntax applies

# the default settings are added as commented out CONF_*=* lines


# where we want the swapfile to be, this is the default
#CONF_SWAPFILE=/var/swap

# set size to absolute value, leaving empty (default) then uses computed value
#   you most likely don't want this, unless you have an special disk situation
CONF_SWAPSIZE=512

# set size to computed value, this times RAM size, dynamically adapts,
#   guarantees that there is enough swap without wasting disk space on excess
#CONF_SWAPFACTOR=2

# restrict size (computed and absolute!) to maximally this limit
#   can be set to empty for no limit, but beware of filled partitions!
#   this is/was a (outdated?) 32bit kernel limit (in MBytes), do not overrun it
#   but is also sensible on 64bit to prevent filling /var or even / partition
#CONF_MAXSWAP=2048



^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

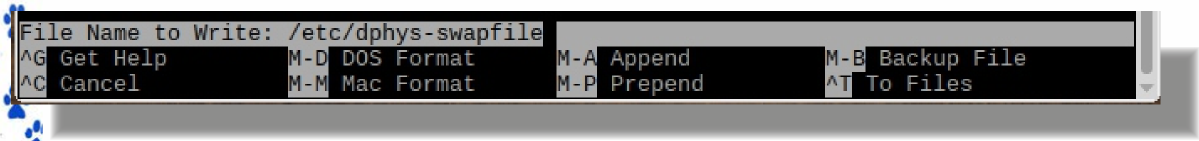
Figure 26: Choose <cntrl X>



```
Save modified buffer?  (Answering "No" will DISCARD changes.)
 Y Yes
 N No           ^C Cancel
```

Figure 27: NANO swapfile = 512 then <cntrl X>
then YES
then <enter>

choose YES



```
File Name to Write: /etc/dphys-swapfile
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```

Figure 28: Choose <enter>

Check it with:

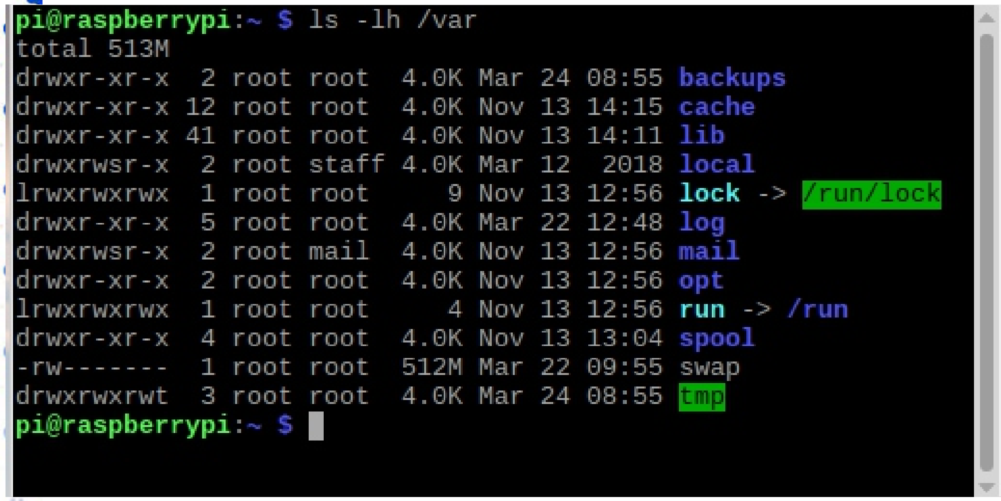
```
ls -lh /var
Info  ls   (list)
      syntax:  ls [OPTION]…  [FILE]…
               option -l  : use long listing format
               option -h  : human readable
      functie:
             list information about the files (the current directory by default).
```

Now reboot the PI by typing reboot or go to shutdown / reboot via the menu.

Hint: About error message: "**can't call the assembler, error -1**"
When recompiling the Lazarus IDE at a later time, you could run into this error.
In this case it indicates that we're running out of memory.
In the rare ocassion that this happens, up the swap value even more: **CONF_SWAPSIZE=1000**

But without a reboot we can also adjust the swap file. Type in

```
nano / etc / dphys-swapfile <enter>    // adjust the value 100 as done above
#  CONF_SWAPSIZE=100 # <-- change 100 in this line to 512
/etc/init.d/dphys-swapfile stop <enter>
/etc/init.d/dphys-swapfile start <enter>

ls -lh /var <enter>
```



```
pi@raspberrypi:~ $ ls -lh /var
total 513M
drwxr-xr-x  2 root root  4.0K Mar 24 08:55 backups
drwxr-xr-x 12 root root  4.0K Nov 13 14:15 cache
drwxr-xr-x 41 root root  4.0K Nov 13 14:11 lib
drwxrwsr-x  2 root staff 4.0K Mar 12  2018 local
lrwxrwxrwx  1 root root     9 Nov 13 12:56 lock -> /run/lock
drwxr-xr-x  5 root root  4.0K Mar 22 12:48 log
drwxrwsr-x  2 root mail  4.0K Nov 13 12:56 mail
drwxr-xr-x  2 root root  4.0K Nov 13 12:56 opt
lrwxrwxrwx  1 root root     4 Nov 13 12:56 run -> /run
drwxr-xr-x  4 root root  4.0K Nov 13 13:04 spool
-rw-------  1 root root  512M Mar 22 09:55 swap
drwxrwxrwt  3 root root  4.0K Mar 24 08:55 tmp
pi@raspberrypi:~ $
```

Figure 29: ls -lh /var: Show var's. Check swap is 512:

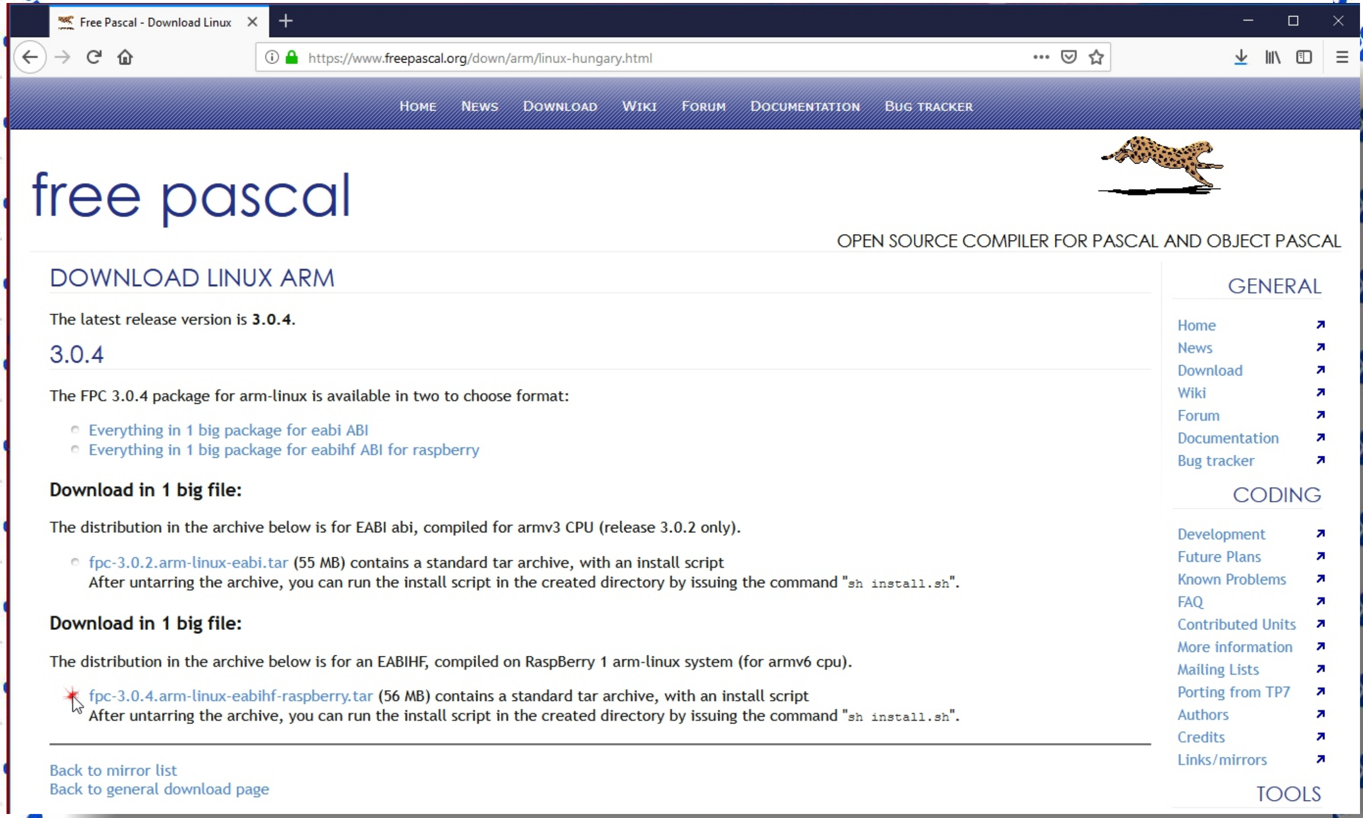
NOTE: See branch 8 at page 27 of this article (not the issue)
Remember to cut that back to the default after the builds!

3. THE FREE PASCAL COMPILER

Before we can install Lazarus, we will need Free Pascal Compiler (FPC) to be able to compile it.
But here comes the kicker… to get the latest FPC version we will need FPC to compile the latest FPC version. This is called Bootstrapping (according to the Wiki page). See chapter above.
So we need a compiled version of Free Pascal Compiler (fpc) to compile Lazarus.

We start by getting the latest compiled version of the Free Pascal Compiler (FPC) For the raspberry pi linux ARM V7 platform.
For bootstrapping we need an already compiled version of the FPC for the ARM processor. We will pick these up and place them in our own folder under / home / pi/ Downloads.

We obtain the required latest version of the fpc file from
`https://www.freepascal.org/down/arm/linux-hungary.html`
or you can download it from your personal download page.



DOWNLOAD LINUX ARM

The latest release version is **3.0.4**.

3.0.4

The FPC 3.0.4 package for arm-linux is available in two to choose format:

- Everything in 1 big package for eabi ABI
- Everything in 1 big package for eabihf ABI for raspberry

Download in 1 big file:

The distribution in the archive below is for EABI abi, compiled for armv3 CPU (release 3.0.2 only).

- fpc-3.0.2.arm-linux-eabi.tar (55 MB) contains a standard tar archive, with an install script
  After untarring the archive, you can run the install script in the created directory by issuing the command "`sh install.sh`".

Download in 1 big file:

The distribution in the archive below is for an EABIHF, compiled on RaspBerry 1 arm-linux system (for armv6 cpu).

- fpc-3.0.4.arm-linux-eabihf-raspberry.tar (56 MB) contains a standard tar archive, with an install script
  After untarring the archive, you can run the install script in the created directory by issuing the command "`sh install.sh`".

Back to mirror list
Back to general download page

Figure 30: Free Pascal: download linux arm fpc-3.0.4 raspberry pi release

Choose the version for the raspberry under FPC 3.0.4 package for arm linux.
In our case we have put the fpc-3.0.4.arm-linux-eabihf-raspberry.tar file on a USB stick.

Now you have two options with the file, to unpack it manually in a terminal or by means of utilities. In both cases we will extract this file under the folder **`/home/pi/Downloads/fpc-3.0.4.arm-linux.`**
The fpc-3.0.4.arm-linux folder is created automatically.

MANUALLY
Open a terminal.
You can find your USB stick under / media / pi / "name of the stick". in our case: 2AF0-8237
There you come with the following command:

```
cd<space>/media/pi/medianame <enter>

Info  cd
      syntax:     cd [OPTION -(l p e @)] [dir]
      functie:    change the current directory to DIR.The default dir
                  is the value of the HOME schell variabele
```

Next, under root rights (sudo), copy the file fpc-3.0.4.arm-linux-eabihf-raspberry.tar to the folder **home/pi/Downloads:**

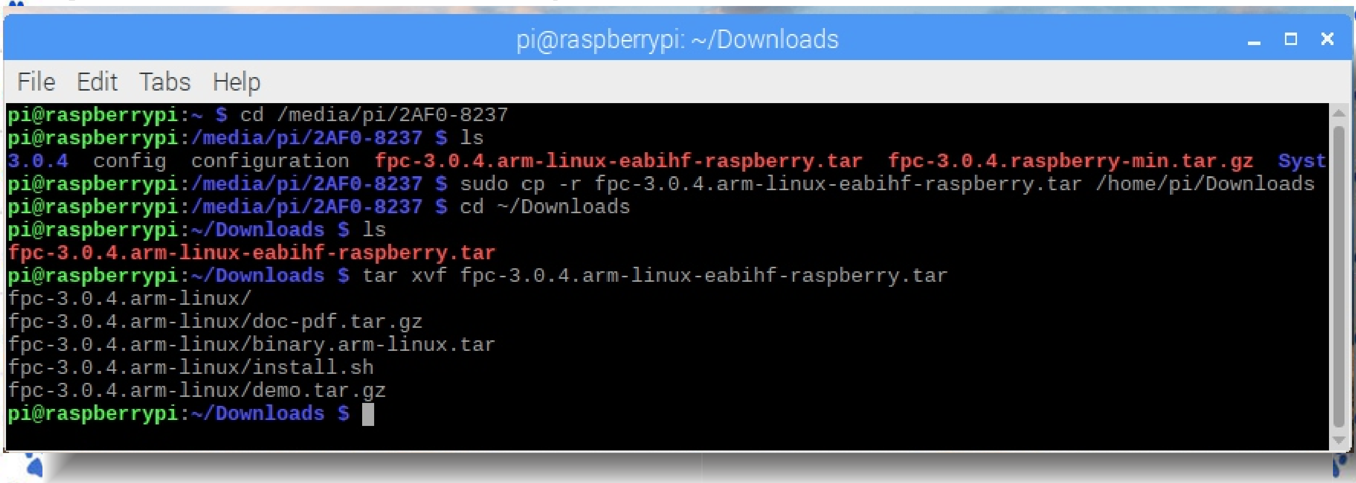
```
sudo<space>cp<space>-r<space>fpc-3.0.4.arm-linux-eabihf-raspberry.tar<space>
                                  /home/pi/Downloads  <enter>
Info  cp
      syntax:     cp [OPTION]… SOURCE… DEST/DIRECTORY
                  option -r, --recursive: copy directories recursifely
      functie:    copy SOURCE to DEST or multiple SOURCES to DIRECTORY
```

Go back to the Downloads folder with cd / home / pi / Downloads. This folder contains the fpc-3.0.4.arm-linux-eabihf-raspberry.tar File that you need to extract.
You can check that with the **ls** command.
You install a * .tar file with the tar command:

```
tar xvf fpc-3.0.4.arm-linux-eabihf-raspberry.tar <enter>

Info tar
     syntax: look up [OPTION -xvf] []
     function: install
```

You will see the result appear in the terminal (see figure 37 article page 18/40 issue page 81 for all copying and more work) :

```
fpc-3.0.4.arm-linux /   // a new fpc-3.0.4.arm-linux directory under the Downloads folder.
fpc-3.0.4.arm-linux / doc-pdf.tar.gz // an archive file in the new fpc-3.0.4.arm / linux folder.
fpc-3.0.4.arm-linux / binary.arm-linux.tar // the tar file that we are going to setup.
fpc-3.0.4.arm-linux / install.sh              // the installation file.
fpc-3.0.4.arm-linux / demo.tar.gz              // another archive file.
```



```
pi@raspberrypi:~ $ cd /media/pi/2AF0-8237
pi@raspberrypi:/media/pi/2AF0-8237 $ ls
3.0.4  config  configuration  fpc-3.0.4.arm-linux-eabihf-raspberry.tar  fpc-3.0.4.raspberry-min.tar.gz  Syst
pi@raspberrypi:/media/pi/2AF0-8237 $ sudo cp -r fpc-3.0.4.arm-linux-eabihf-raspberry.tar /home/pi/Downloads
pi@raspberrypi:/media/pi/2AF0-8237 $ cd ~/Downloads
pi@raspberrypi:~/Downloads $ ls
fpc-3.0.4.arm-linux-eabihf-raspberry.tar
pi@raspberrypi:~/Downloads $ tar xvf fpc-3.0.4.arm-linux-eabihf-raspberry.tar
fpc-3.0.4.arm-linux/
fpc-3.0.4.arm-linux/doc-pdf.tar.gz
fpc-3.0.4.arm-linux/binary.arm-linux.tar
fpc-3.0.4.arm-linux/install.sh
fpc-3.0.4.arm-linux/demo.tar.gz
pi@raspberrypi:~/Downloads $
```

Figure 31: Copy *.tar to Downloads folder and extract it to fpc-3.0.4.arm-linux

NOTE
Generally:

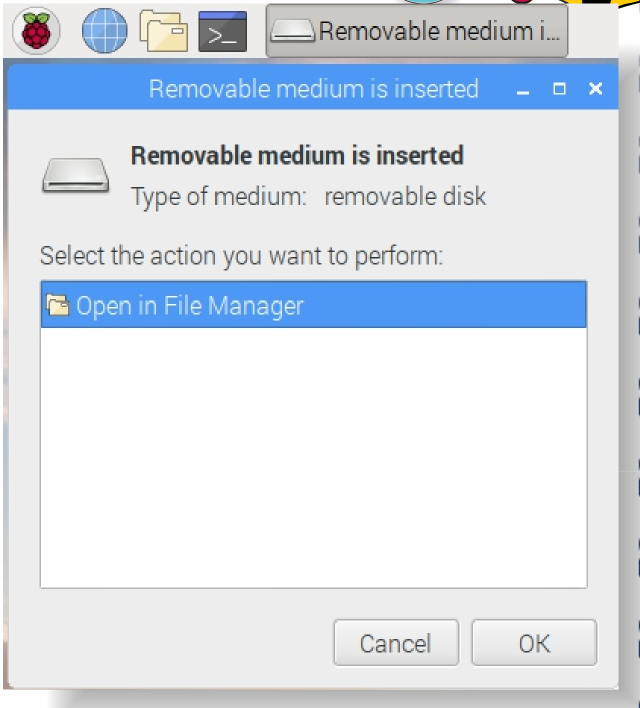
as **spaces are not supported under Linux or Raspbian without quotes,** the document shows paths without spaces.

```
Info .gz
     function: archive file as eg zip
     or 7zip?
     How do you extract a .gz file?
     Also with Xarchiver.
```

UTILITY

Put your stick in a USB input of the RaspBerry Pi. You will then see a window with the title "Removable medium has been inserted".

When asked in this window which action should be taken, choose "Open in file manager". You will now see the file management window with the name of the stick as its title, where you choose the fpc-3.0.4.arm-linux-eabihf-raspberry.tar file. Double-clicking on this file opens the Xarchiver file extractor.



Removable medium is inserted

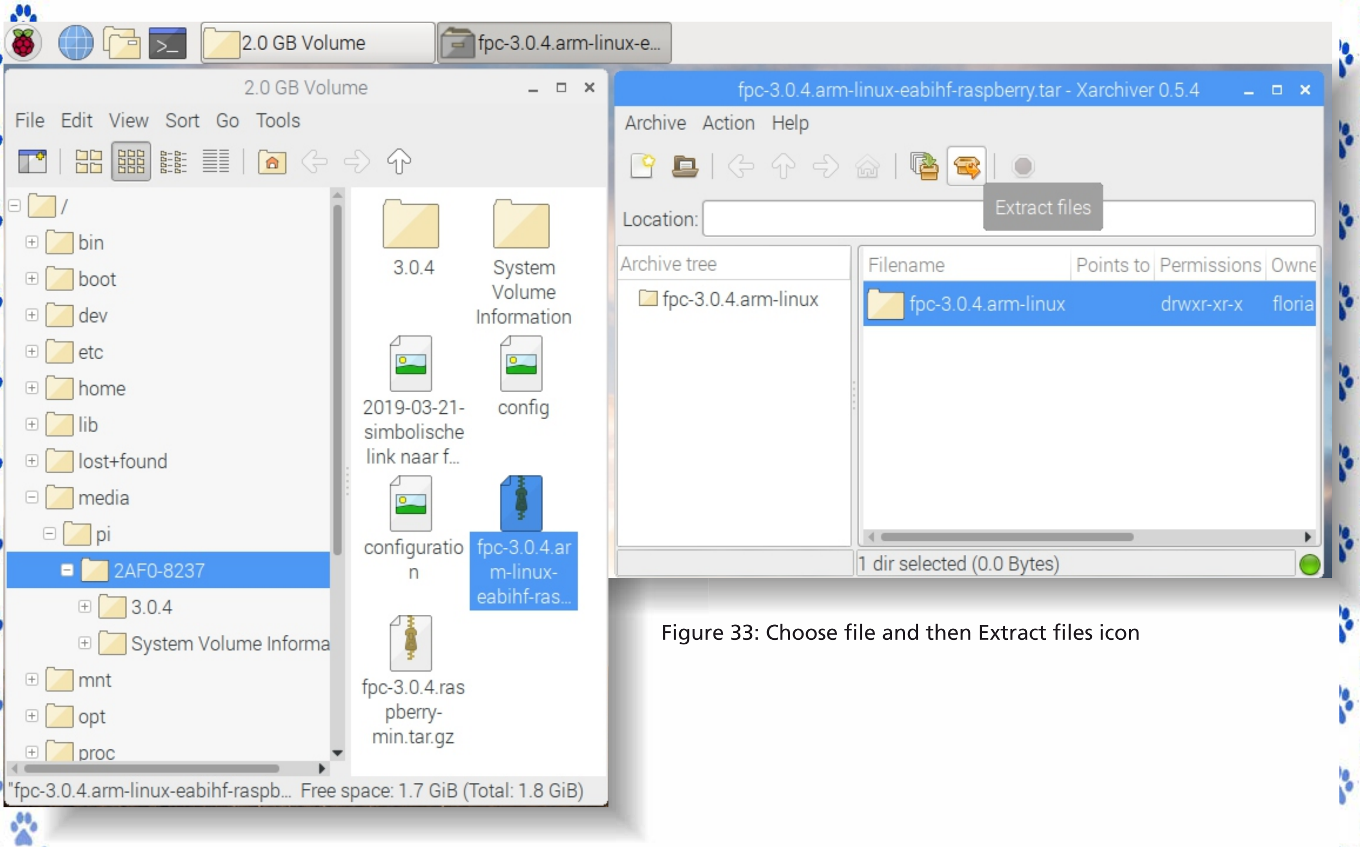
Removable medium is inserted

Type of medium: removable disk

Select the action you want to perform:

Open in File Manager

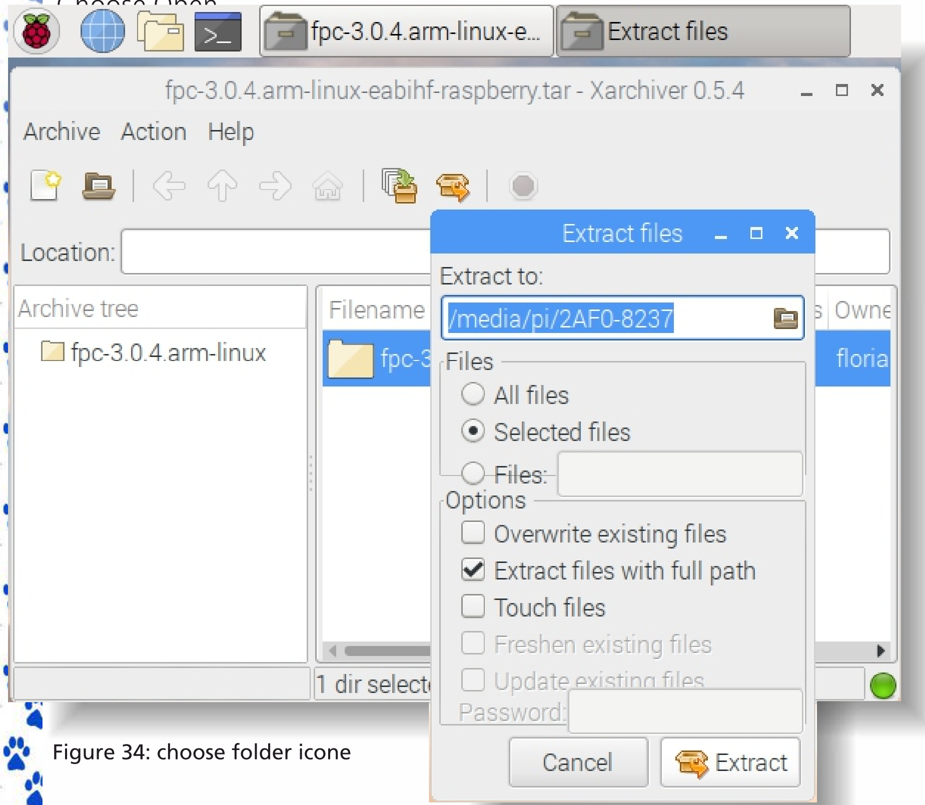
Figure 32: Removable medium



2.0 GB Volume
fpc-3.0.4.arm-linux-e...
File Edit View Sort Go Tools
/
bin
boot
dev
etc
home
lib
lost+found
media
pi
2AF0-8237
3.0.4
System Volume Informa
mnt
opt
proc
3.0.4
System Volume Information
2019-03-21-simbolische link naar f...
config
configuration
fpc-3.0.4.arm-linux-eabihf-ras...
fpc-3.0.4.raspberry-min.tar.gz
"fpc-3.0.4.arm-linux-eabihf-raspb... Free space: 1.7 GiB (Total: 1.8 GiB)
fpc-3.0.4.arm-linux-eabihf-raspberry.tar - Xarchiver 0.5.4
Archive Action Help
Extract files
Location:
Archive tree
fpc-3.0.4.arm-linux
Filename
Points to
Permissions
Owne
fpc-3.0.4.arm-linux
drwxr-xr-x
floria
1 dir selected (0.0 Bytes)

Figure 33: Choose file and then Extract files icon

On the left is the Archive tree structure from the tar file and on the right the file that is being extracted. Choose this. In the menu line above the Location field is an icon with the hint Extract Files. see Figure 33 on the previous page. Choose this option and in the following window "extract files" a number of questions are asked: under "Extract to:" instead of / media / medianame: 2AF0-8237: click on the folder icon on the right in the same line to open a new file manager and choose / home / pi / Downloads". Choose Open.



Extract files

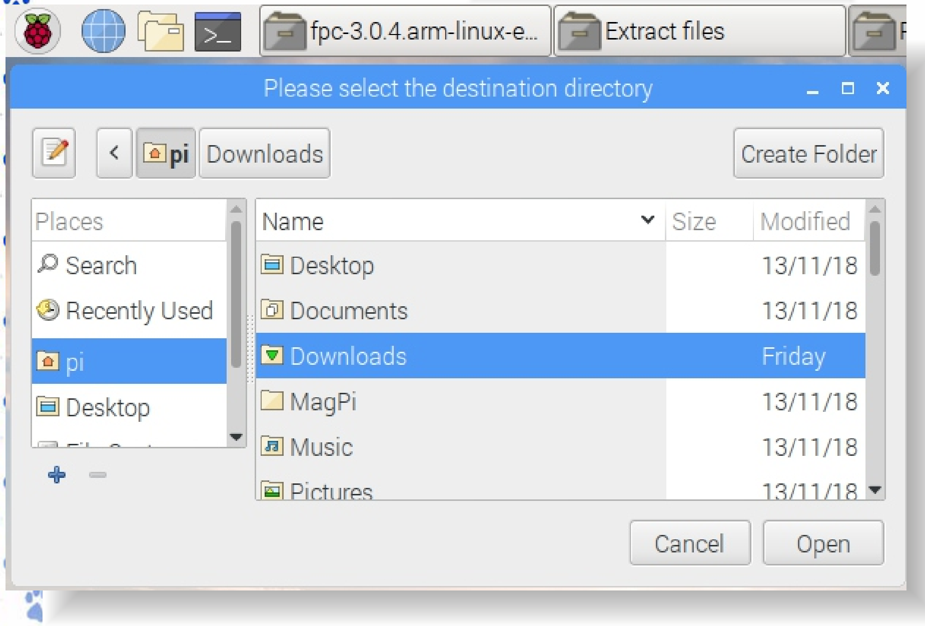
Extract to: /media/pi/2AF0-8237

Files
○ All files
● Selected files
○ Files:

Options
☐ Overwrite existing files
☑ Extract files with full path
☐ Touch files
☐ Freshen existing files
☐ Update existing files
Password:

Cancel | Extract

Figure 34: choose folder icone



Name	Size	Modified
Desktop		13/11/18
Documents		13/11/18
Downloads		Friday
MagPi		13/11/18
Music		13/11/18
Pictures		13/11/18

Figure 35: Change destination folder

Close all open windows and open a file manager. Look under
Downloads in the newly created folder fpc-3.0.4-arm-linux to see if the following files are there:

- binary.arm-linux.tar
- demo.tar.gz
- doc-pdf.tar.gz
- install.sh

Tip: If you are done with the USB stick and want to remove it, you cannot simply pull it out:
at the top right, next to the Bluetooht, wifi, ..., time icons, you will see a triangle with a line below it.
Clicking on this gives you the option to close your USB stick so that you can now safely remove it. See figure below.

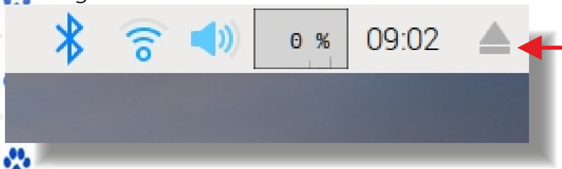


Figure 36: USB icon to remove Removable medium

INSTALL.SH

We arrived at the installation file install.sh in both ways. XXX.sh is a so-called Shellscript file. Similar to a .bat file under Windows. Go to the install.sh file:

```
cd ~/Downloads/fpc-3.0.4.arm-linux
Info cd ~         (~ this is called a tilde)
     [OPTION  ~ ]with this option you immediately go back to your own user root
     that is not the root / root) and hence the specified path is followed.
```

Sometimes you cannot execute it yet because it is not yet an executable file. You make it executable under root rights using the command below.

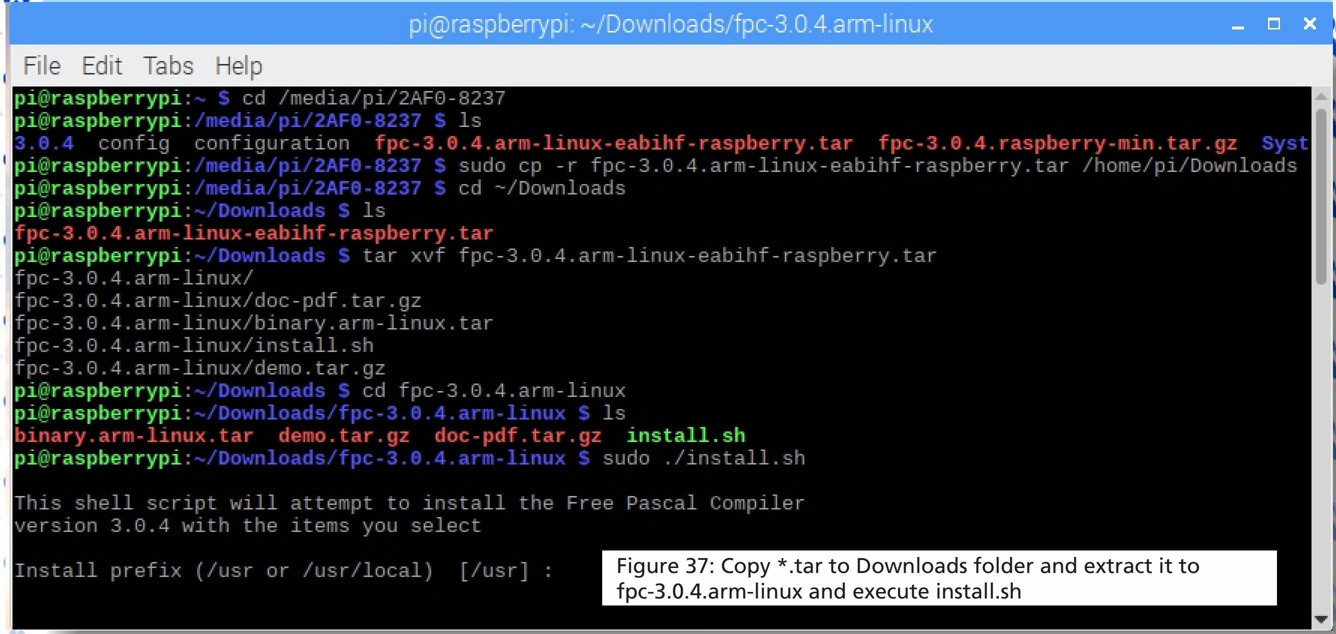
NOTE:

Pay attention to the color change of this file in the terminal before executing the command below.

```
sudo chmod +x install.sh <enter>
Info  chmod
      syntax: [OPTIE +x] []
```

chmod +x on a file (your script) only means, that you'll make it executable. Right click on your script and chose Properties -> Permissions -> Allow executing file as program, leaves you with the exact same result as the command in terminal.

```
sudo ./install.sh
```



```
pi@raspberrypi:~ $ cd /media/pi/2AF0-8237
pi@raspberrypi:/media/pi/2AF0-8237 $ ls
3.0.4  config  configuration  fpc-3.0.4.arm-linux-eabihf-raspberry.tar  fpc-3.0.4.raspberry-min.tar.gz  Syst
pi@raspberrypi:/media/pi/2AF0-8237 $ sudo cp -r fpc-3.0.4.arm-linux-eabihf-raspberry.tar /home/pi/Downloads
pi@raspberrypi:/media/pi/2AF0-8237 $ cd ~/Downloads
pi@raspberrypi:~/Downloads $ ls
fpc-3.0.4.arm-linux-eabihf-raspberry.tar
pi@raspberrypi:~/Downloads $ tar xvf fpc-3.0.4.arm-linux-eabihf-raspberry.tar
fpc-3.0.4.arm-linux/
fpc-3.0.4.arm-linux/doc-pdf.tar.gz
fpc-3.0.4.arm-linux/binary.arm-linux.tar
fpc-3.0.4.arm-linux/install.sh
fpc-3.0.4.arm-linux/demo.tar.gz
pi@raspberrypi:~/Downloads $ cd fpc-3.0.4.arm-linux
pi@raspberrypi:~/Downloads/fpc-3.0.4.arm-linux $ ls
binary.arm-linux.tar  demo.tar.gz  doc-pdf.tar.gz  install.sh
pi@raspberrypi:~/Downloads/fpc-3.0.4.arm-linux $ sudo ./install.sh

This shell script will attempt to install the Free Pascal Compiler
version 3.0.4 with the items you select

Install prefix (/usr or /usr/local)  [/usr] :
```

Figure 37: Copy *.tar to Downloads folder and extract it to fpc-3.0.4.arm-linux and execute install.sh

```
Info ./
    syntax: ./file_name
    function: stands for execution.
```

Immediately after the start of ./ the question appears what the **`PREFIX path`** should be: **`/usr`** or **`/usr/local`**. You must choose **`/usr/local`** and press <enter>.
During the installation you will be asked if you want the **`Textmode IDE (Y / n)?`**
want: **`choose Y`**, and **`Documents (Y / n)?`**
`choose n` and **`Install demos (Y / n)?`** `Again` **`choose n`**.

Now leaflets are created and things are installed in **`/usr/local.`**
If you now enter fpc <enter> in a terminal, you will get a whole story about the fpc compiler. In the beginning You can read that this is fpc version 3.0.4.

Do not install/start this seed compiler 3.0.4, we will use it only to compile fpc trunk.

For detailed information go to:
`https://developer.gnome.org/seed/stable/seed-building.html`

NOTE : To create a path just for your knowledge:
`export PPC_CONFIG_PATH=/home/pi/Downloads/fpc-3.0.4.arm-linux/bin` <ENTER>

and

`export PATH=$PPC_CONFIG_PATH:$PATH` <ENTER>

We are now going to create a folder source in which we will install things.

```
cd ~              -->//As Linux and Raspian bash is case-sensitive,
                     (otherwise error "bash:Cd: command not found" occurs):
mkdir source
Info  mkdir
      syntax   :   mkdir [OPTION -(m p v z)]… DIRECTORY
      function:   create the DIRECTORY(ies), if the do not already exist.
```

4. GET THE LATEST FREE PASCAL COMPILER PACKAGES

Now we are ready to retrieve all the files needed to create the latest version of the FPC using the just retrieved FPC-3.0.4 compiler. See the Bootstrapping story above. Since we do not know exactly which files we need, we will use a specially made program called SubVersion (a.k.a SVN).

SubVersion (SVN) see previous Blaises 75/76 page 16

Info: Connect to FPC Source Repository with SVN (SubVersion)
As an alternative to the daily zip files of the SVN sources, the SVN repository has been made accessible for everyone, with read-only access. This means that you can always have access to the latest source code. It is also a method which requires less bandwidth once you have done the first download (called a "checkout" or "co" in SVN lingo).

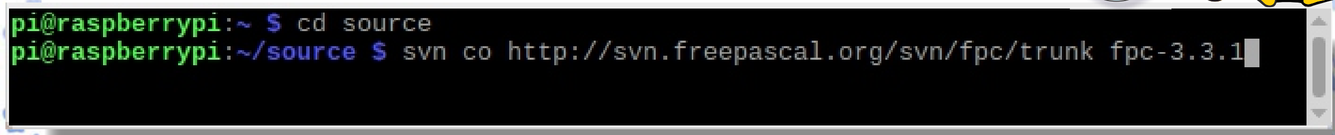
First we will get **"subversion"** (a.k.a SVN) so we can download the latest version from the Free Pascal SVN.
Open a terminal. (rightclick on the file and it will open for you with a long list and choose Chechkout) or:

```
sudo apt-get update
sudo apt-get install -y subversion    //download and install subversion with option: -y

info:
      Optie -y: without this option, you must manually press yes if prompted
      during execution. Now the script runs automatically.
```

Now we are going to retrieve the FPC 3.3.1 compiler files. Type the following two lines and wait patiently after each line, sometimes it can take a while:

```
cd ~/source
svn co http://svn.freepascal.org/svn/fpc/trunk fpc-3.3.1
Info: optie CO : stands for checkout.
```

A very large script is now being executed and stored in the folder source.



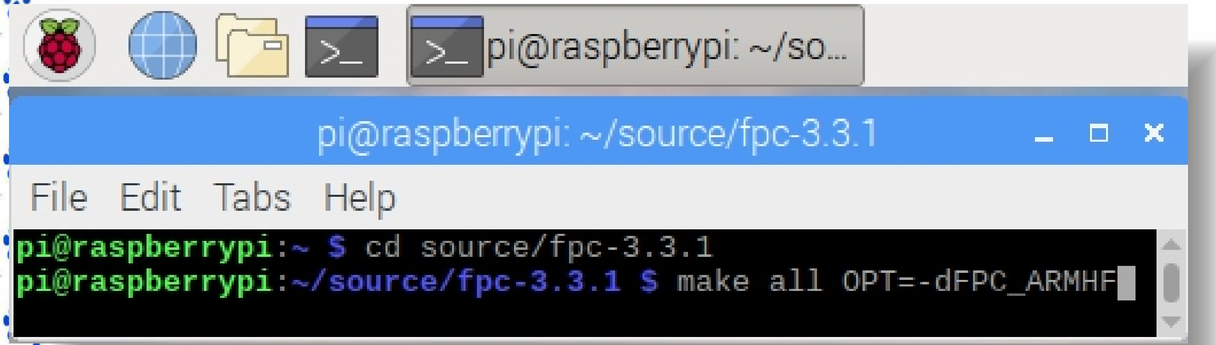
```
pi@raspberrypi:~ $ cd source
pi@raspberrypi:~/source $ svn co http://svn.freepascal.org/svn/fpc/trunk fpc-3.3.1
```

Figure 38: SVN freepascal trunk to new folder fpc-3.3.1

5. CREATE THE LATEST FREE PASCAL COMPILER

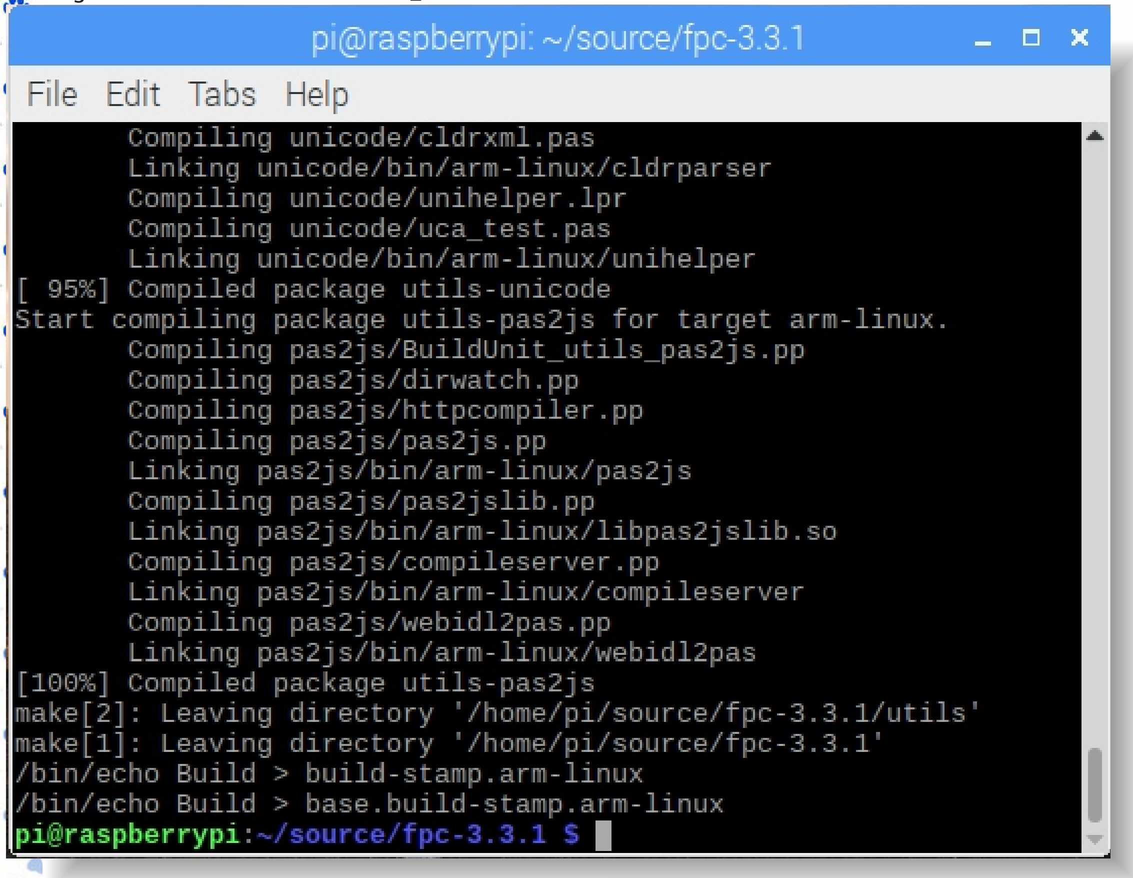
Now the last fpc version must be created from these files in / usr / local.
Enter the following and go get a cup of coffee, this will take a while:

```
cd ~/source/fpc-3.3.1
make all OPT=-dFPC_ARMHF
```



```
pi@raspberrypi:~ $ cd source/fpc-3.3.1
pi@raspberrypi:~/source/fpc-3.3.1 $ make all OPT=-dFPC_ARMHF
```

Figure 39: start make all OPT=-dFPC_ARMHF



```
       Compiling unicode/cldrxml.pas
       Linking unicode/bin/arm-linux/cldrparser
       Compiling unicode/unihelper.lpr
       Compiling unicode/uca_test.pas
       Linking unicode/bin/arm-linux/unihelper
[ 95%] Compiled package utils-unicode
Start compiling package utils-pas2js for target arm-linux.
       Compiling pas2js/BuildUnit_utils_pas2js.pp
       Compiling pas2js/dirwatch.pp
       Compiling pas2js/httpcompiler.pp
       Compiling pas2js/pas2js.pp
       Linking pas2js/bin/arm-linux/pas2js
       Compiling pas2js/pas2jslib.pp
       Linking pas2js/bin/arm-linux/libpas2jslib.so
       Compiling pas2js/compileserver.pp
       Linking pas2js/bin/arm-linux/compileserver
       Compiling pas2js/webidl2pas.pp
       Linking pas2js/bin/arm-linux/webidl2pas
[100%] Compiled package utils-pas2js
make[2]: Leaving directory '/home/pi/source/fpc-3.3.1/utils'
make[1]: Leaving directory '/home/pi/source/fpc-3.3.1'
/bin/echo Build > build-stamp.arm-linux
/bin/echo Build > base.build-stamp.arm-linux
pi@raspberrypi:~/source/fpc-3.3.1 $
```

Figure 40: finish make all OPT=-dFPC_ARMHF

```
sudo make install OPT=-dFPC_ARMF<SPACE>PREFIX=/usr/local
```

if you also want to save the install files then give the following command

```
sudo make install sourceinstall OPT=-dFPC_ARMHF PREFIX=/usr/local
Info make all
      syntax:
      function: creates the requested

Info make install
      syntax:
      function: installs the just created in the system: here in / usr / local
```

Now we want to know which fpc version we brought in. You do that by looking in the fpc folder:

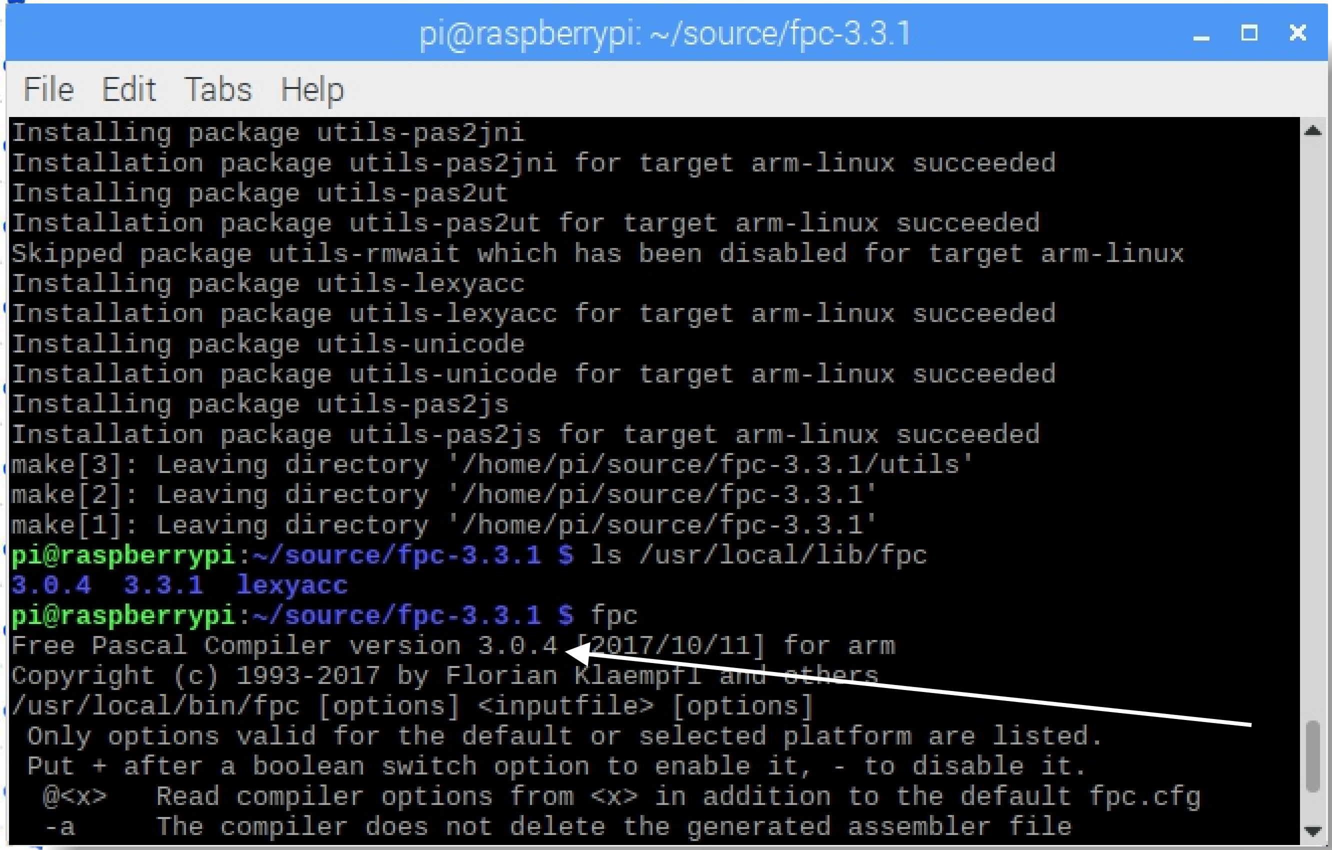
```
ls /usr/local/lib/fpc
```

You should now see the following folders there:

```
3.0.4    /    3.3.1    /    lexyacc
```

see fig 41

It is possible that the folder with inscription 3.3.1 has a higher number. But for now that is the latest version of the fpc. Let's check that by typing fpc in the terminal again. Hey, the link still refers to the fpc version 3.0.4.



```
Installing package utils-pas2jni
Installation package utils-pas2jni for target arm-linux succeeded
Installing package utils-pas2ut
Installation package utils-pas2ut for target arm-linux succeeded
Skipped package utils-rmwait which has been disabled for target arm-linux
Installing package utils-lexyacc
Installation package utils-lexyacc for target arm-linux succeeded
Installing package utils-unicode
Installation package utils-unicode for target arm-linux succeeded
Installing package utils-pas2js
Installation package utils-pas2js for target arm-linux succeeded
make[3]: Leaving directory '/home/pi/source/fpc-3.3.1/utils'
make[2]: Leaving directory '/home/pi/source/fpc-3.3.1'
make[1]: Leaving directory '/home/pi/source/fpc-3.3.1'
pi@raspberrypi:~/source/fpc-3.3.1 $ ls /usr/local/lib/fpc
3.0.4  3.3.1  lexyacc
pi@raspberrypi:~/source/fpc-3.3.1 $ fpc
Free Pascal Compiler version 3.0.4 [2017/10/11] for arm
Copyright (c) 1993-2017 by Florian Klaempfl and others
/usr/local/bin/fpc [options] <inputfile> [options]
 Only options valid for the default or selected platform are listed.
 Put + after a boolean switch option to enable it, - to disable it.
  @<x>    Read compiler options from <x> in addition to the default fpc.cfg
  -a      The compiler does not delete the generated assembler file
```

Figure 41: FPC points still to the ‘old’ 3.0.4 version

To solve this, we will use a so-called symbolic link.
To redirect the reference of the old FPC to the newer FPC, we will create a symbolize link to the ppcarm (which stands for Portable Pascal Compiler ARM) so that this refers to the newer FPC binary (ppcarm in folder 3.3.1).
Enter:

```
sudo rm -f /usr/local/bin/ppcarm
Info   rm
       syntax: rm [OPTION]… [FILE]
       Option  -f : force
               -r : recursive: remove directories and their contents recusrsively
               -d : remove empty derectories
       function: remove (unlink) the FILE(s).

sudo ln -sf /usr/local/lib/fpc/3.3.1/ppcarm<space>
        /usr/local/bin/ppcarm.
```

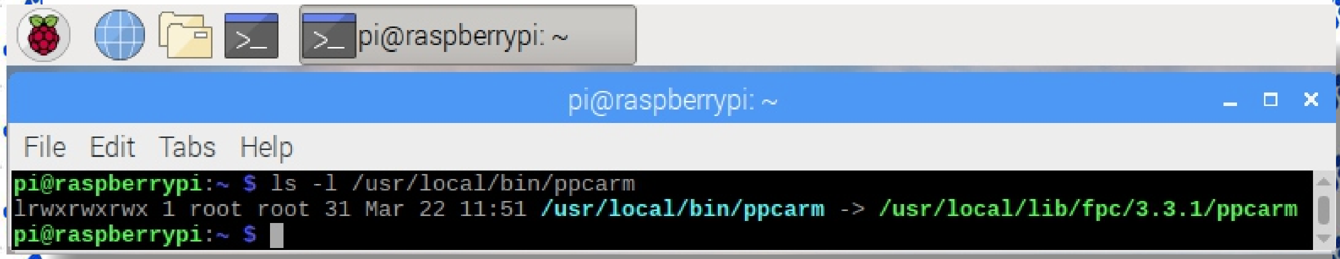
Note: For the sake of clarity, the two paths are placed below each other: type in the terminal one after the other, taking into account the space in between!

```
Info   ln
       syntax:     Option -sf: s = symbolic and f = force
       function:
             Makes a hard link, and with option -s it becomes a symbolic link.
             With option -f (force): delete existing destination files.
```

We check this symbolic link with: `ls -l /usr/local/bin/ppcarm.`

This is the result:

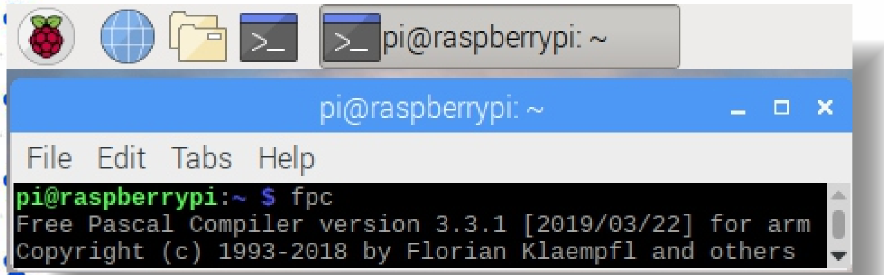
```
lrwxrwxrwx 1 root root 31 mrt 22 11:51 /usr/local/bin/ppcarm ->
/usr/local/lib/fpc/3.3.1/ppcarm
```



```
pi@raspberrypi:~ $ ls -l /usr/local/bin/ppcarm
lrwxrwxrwx 1 root root 31 Mar 22 11:51 /usr/local/bin/ppcarm -> /usr/local/lib/fpc/3.3.1/ppcarm
pi@raspberrypi:~ $ 
```

Figure 42:ppcarm logical link

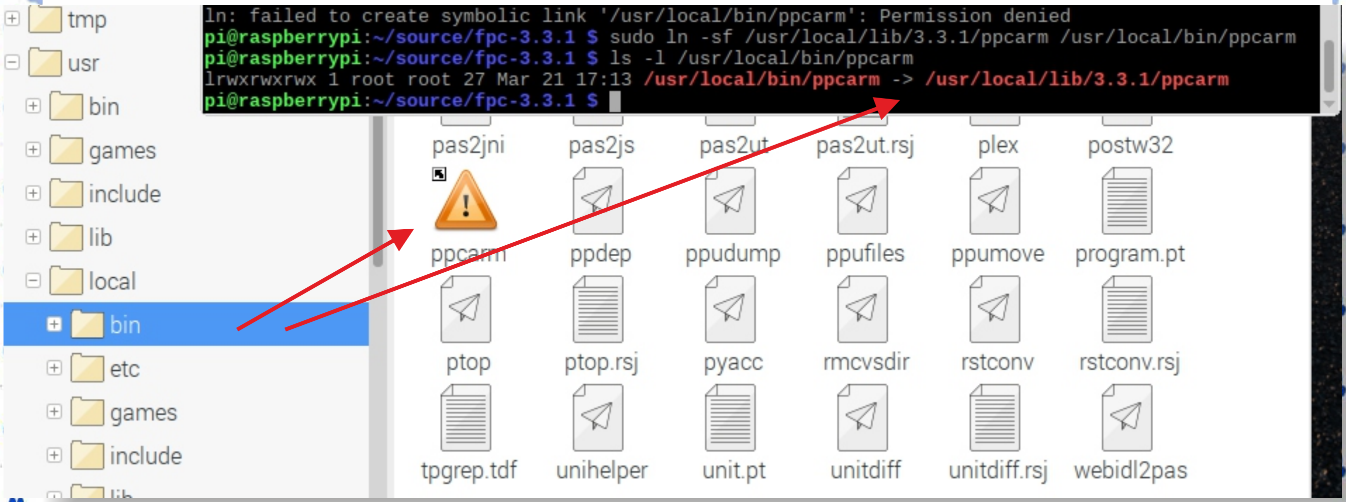
Now type fpc in a console and then you should see the new version number in the help content of the fpc.



```
pi@raspberrypi:~ $ fpc
Free Pascal Compiler version 3.3.1 [2019/03/22] for arm
Copyright (c) 1993-2018 by Florian Klaempfl and others
```

Figure 43: FPC points now to the right 3.3.1 version

You see that the symbolic link refers to the new fpc 3.3.1 ppcarm.
Note: But if there is something wrong in the link, you will see a colored warning triangle as a new link ppcarm. If the link is correct, the color is the same as the rest.



```
ln: failed to create symbolic link '/usr/local/bin/ppcarm': Permission denied
pi@raspberrypi:~/source/fpc-3.3.1 $ sudo ln -sf /usr/local/lib/3.3.1/ppcarm /usr/local/bin/ppcarm
pi@raspberrypi:~/source/fpc-3.3.1 $ ls -l /usr/local/bin/ppcarm
lrwxrwxrwx 1 root root 27 Mar 21 17:13 /usr/local/bin/ppcarm -> /usr/local/lib/3.3.1/ppcarm
pi@raspberrypi:~/source/fpc-3.3.1 $
```

Figure 44: path should be /usr/local/lib/fpc/ppcarm and not this /usr/local/lib/3.3.1/ppcarm

6. GET LAZARUS PACKAGES

1. libraries for Lazarus

We need a few extra packages to make lazarus. We will first collect this. We look at whether there are apt-get updates available and then get the packages one by one for clarity.

```
sudo apt-get update

sudo apt-get install -y libgtk2.0-dev
sudo apt-get install -y libcairo2-dev
sudo apt-get install -y libpango1.0-dev
sudo apt-get install -y libpangox-1.0-dev
sudo apt-get install -y libgdk-pixbuf2.0-dev
sudo apt-get install -y libx11-dev
sudo apt-get install -y gir1.2-coglpango-1.0
sudo apt-get install -y xorg-dev
```

Very often you have to enter the sudo command to get root rights. That can also be done differently.
For this you make a kind of super user also called Super User (su), therefore a special shell is opened for you with:

```
sudo su
Info   su
       syntax: su
       function: this will open the Super User (su) shell root @ raspberrypi:
       home / pi # in the terminal.
```

Now you can enter the following under the root rights:

```
apt-get update

apt-get install -y libgtk2.0-dev
apt-get install -y libcairo2-dev
apt-get install -y libpango1.0-dev
apt-get install -y libpangox-1.0-dev
apt-get install -y libgdk-pixbuf2.0-dev
apt-get install -y libx11-dev
apt-get install -y gir1.2-coglpango-1.0
apt-get install -y xorg-dev
```

To exit this special shell, tap:

```
exit
```

TIP: You could also get them all in a single apt-get install command by placing all packages behind the -y option, and space separated:

```
      sudo apt-get install -y libgtk2.0-dev libcairo2-dev libpango1.0-dev
```

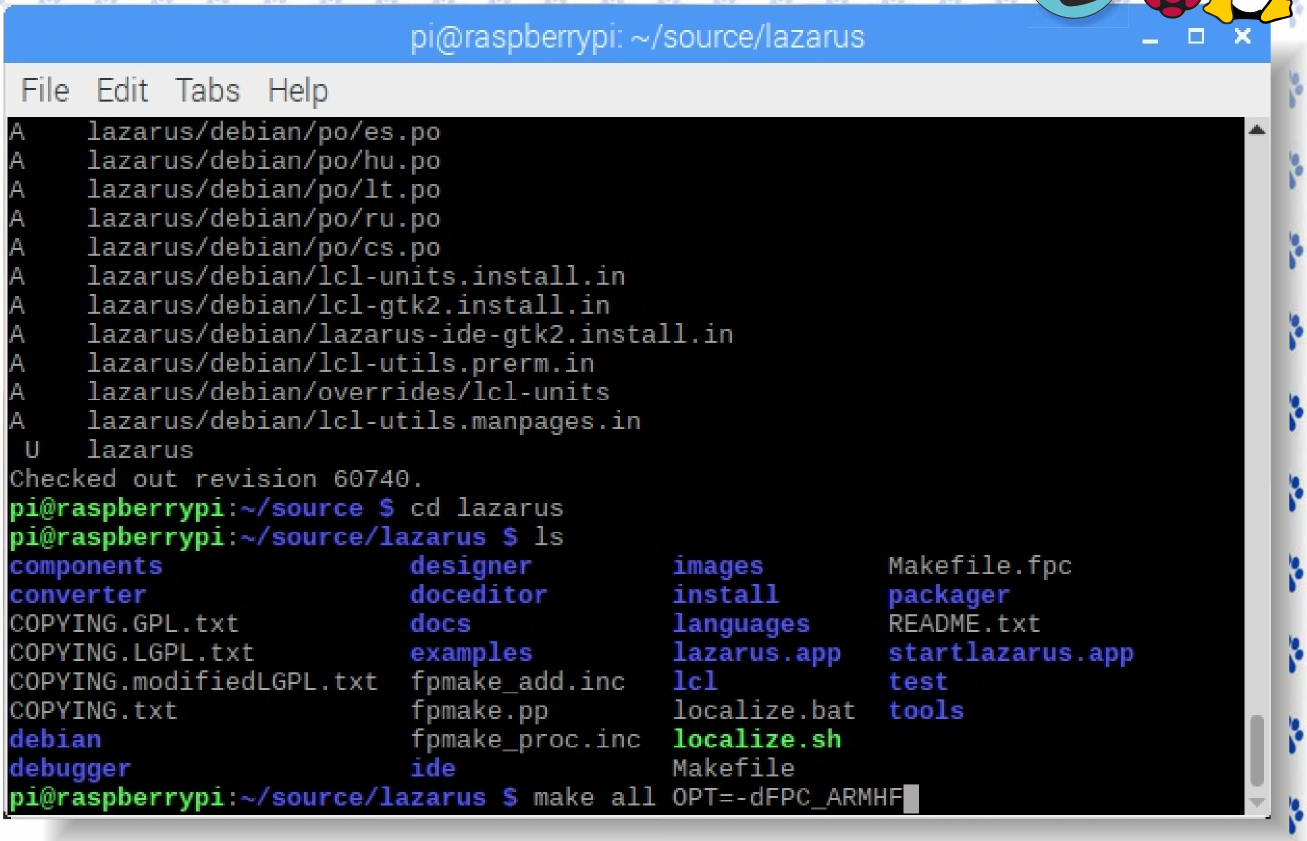
and so on.

7. INSTALL THE LAZARUS PASCAL IDE

We are going to save lazarus under the source folder, and see with the help of svn which version is available at **`http://svn.freepascal.org/svn/lazarus`**

```
cd ~/source

svn co http://svn.freepascal.org/svn/lazarus/trunk lazarus
```

To compile Lazarus (LCL + IDE) go to the dir where everything is unpacked.
Here it is / home / pi / source / lazarus:



```
A    lazarus/debian/po/es.po
A    lazarus/debian/po/hu.po
A    lazarus/debian/po/lt.po
A    lazarus/debian/po/ru.po
A    lazarus/debian/po/cs.po
A    lazarus/debian/lcl-units.install.in
A    lazarus/debian/lcl-gtk2.install.in
A    lazarus/debian/lazarus-ide-gtk2.install.in
A    lazarus/debian/lcl-utils.prerm.in
A    lazarus/debian/overrides/lcl-units
A    lazarus/debian/lcl-utils.manpages.in
 U   lazarus
Checked out revision 60740.
pi@raspberrypi:~/source $ cd lazarus
pi@raspberrypi:~/source/lazarus $ ls
components                 designer         images         Makefile.fpc
converter                  doceditor        install        packager
COPYING.GPL.txt            docs             languages      README.txt
COPYING.LGPL.txt           examples         lazarus.app    startlazarus.app
COPYING.modifiedLGPL.txt   fpmake_add.inc   lcl            test
COPYING.txt                fpmake.pp        localize.bat   tools
debian                     fpmake_proc.inc  localize.sh
debugger                   ide              Makefile
pi@raspberrypi:~/source/lazarus $ make all OPT=-dFPC_ARMHF
```

Figure 46: SVN Done, now make all….

```
cd ~/source/lazarus
```

and type:

```
make all OPT=-dFPC_ARMHF

sudo make install OPT=-dFPC_ARMHF PREFIX=/usr/local
```

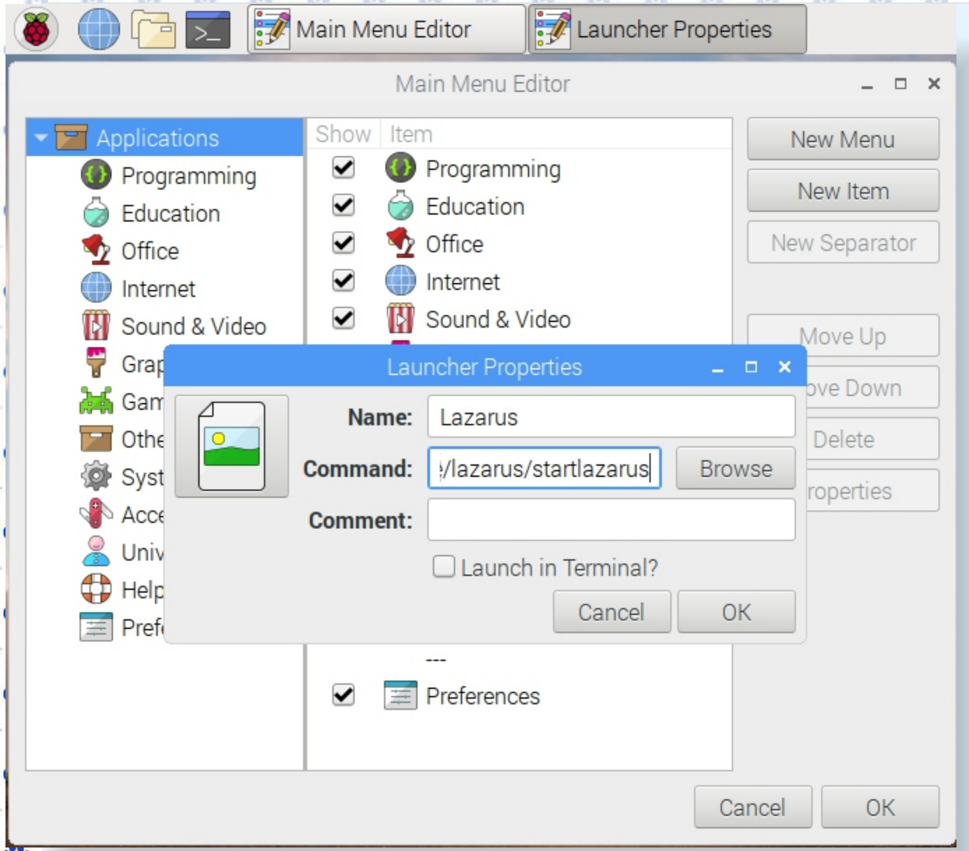
8. NOW CONTINUE WITH THE POST BUILD STEPS.

Set swapfile to default

```
sudo nano/etc/dphys-swapfile    //adjust the value 512 back to 100
#  CONF_SWAPSIZE=512 # <-- change 512 in this line to 100
/etc/init.d/dphys-swapfile stop
/etc/init.d/dphys-swapfile start

ls -lh/var // show the var's
```

there is a nano's shortcut to save the changes and close the document: <CTRL+O><ENTER><CTRL+X>

CONFIGURING AND TESTING LAZARUS what to do and/or try with the newly built Lazarus.
Lazarus should now be found under main menu `"raspberry" -> programming -> Lazarus.`
If not, you can arrange that yourself with the Main Menu Editor, give it the Name Lazarus and for the Command brew the computer to find the startlazarus executable (`/source/lazarus/startlazarus[link]`), the desktop icon is also in the folder.
Then under Programming you will find Lazarus



☑ Programming
☑ Education
☑ Office
☑ Internet
☑ Sound & Video
☑ Preferences

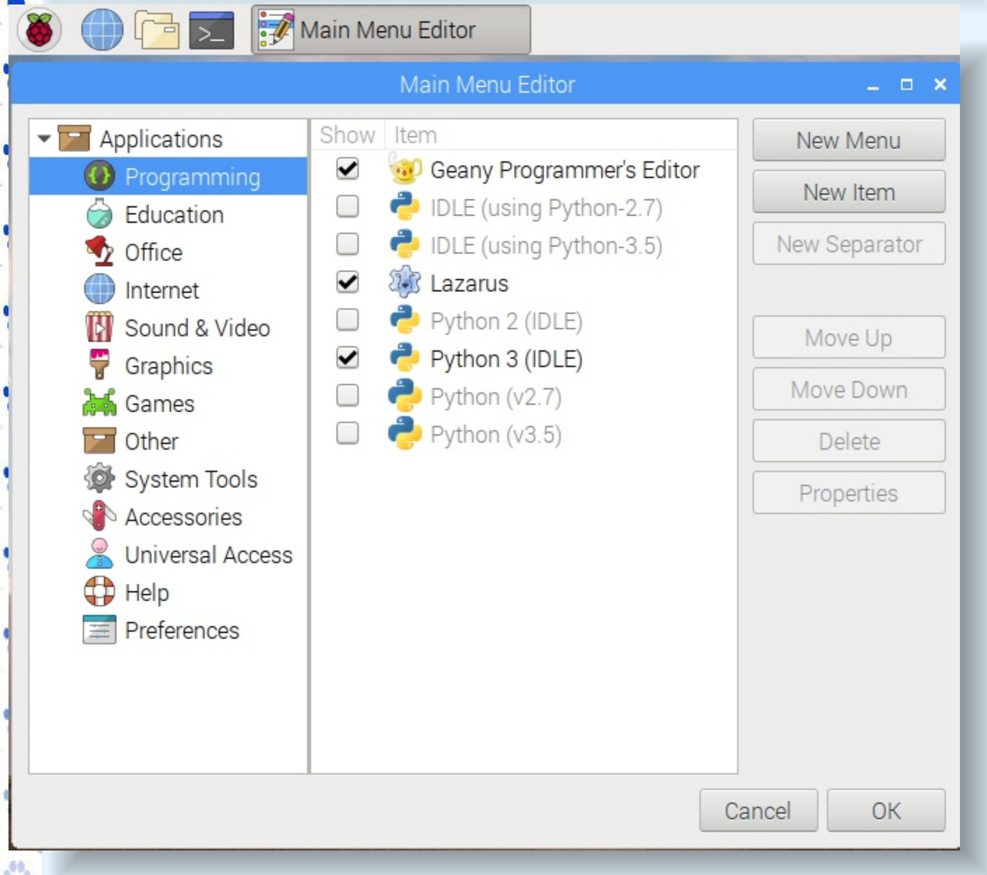
Name: Lazarus

Command: ı/lazarus/startlazarus

Comment:

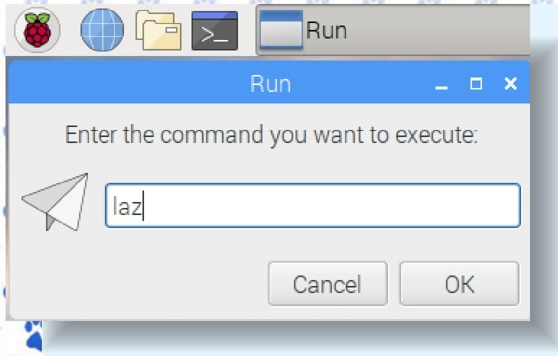
☐ Launch in Terminal?

Figure 47: apply lazarus path



☑ Geany Programmer's Editor
☐ IDLE (using Python-2.7)
☐ IDLE (using Python-3.5)
☑ Lazarus
☐ Python 2 (IDLE)
☑ Python 3 (IDLE)
☐ Python (v2.7)
☐ Python (v3.5)

Figure 48: You will find Lazarus under programming

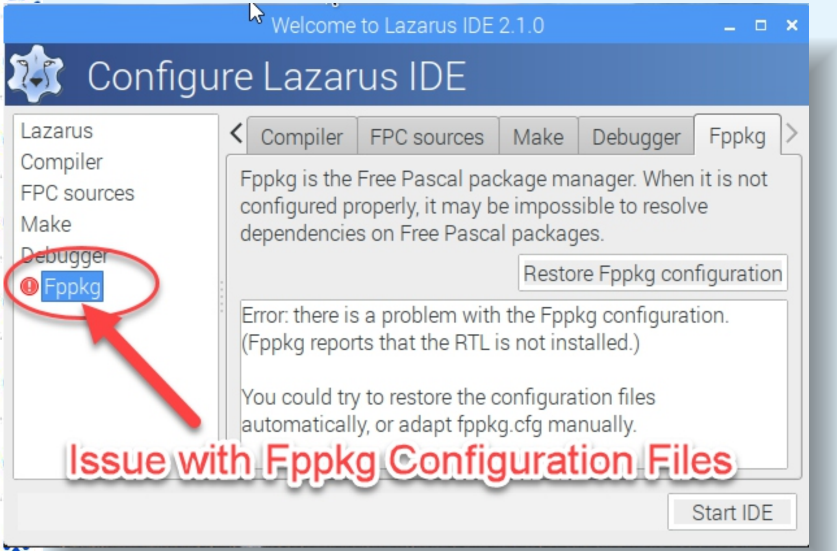


Enter the command you want to execute:

laz

Figure 49:execute lazarus

Figure 50: lazarus 2.1.0



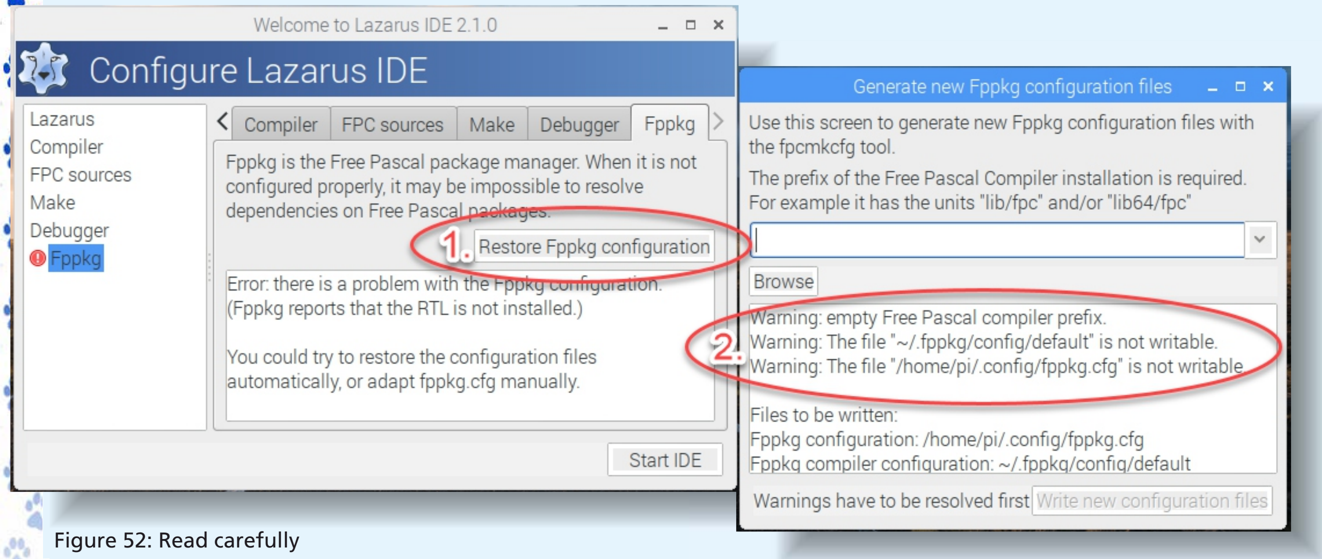
Fppkg is the Free Pascal package manager. When it is not configured properly, it may be impossible to resolve dependencies on Free Pascal packages.

Error: there is a problem with the Fppkg configuration. (Fppkg reports that the RTL is not installed.)

You could try to restore the configuration files automatically, or adapt fppkg.cfg manually.

Figure 51: Configuring ...wherein you see an error adjacent Fppkg, do the following:
If prompted to Configure Lazarus IDE via the above splash screen entitled `Configure Lazarus IDE` (Figure 51) AND there is an error indicated beside the (!) Fppkg selector (as highlighted above via the red arrow)"

1. Click on "Restore Fppkg configuration"to launch its associated "Generate new Fppkg configuration files" dialogue.


Welcome to Lazarus IDE 2.1.0
Configure Lazarus IDE
Lazarus
Compiler
FPC sources
Make
Debugger
Fppkg
Compiler
FPC sources
Make
Debugger
Fppkg
1.
Restore Fppkg configuration
Start IDE
Generate new Fppkg configuration files
Browse
2.
Write new configuration files

Fppkg is the Free Pascal package manager. When it is not configured properly, it may be impossible to resolve dependencies on Free Pascal packages.

Error: there is a problem with the Fppkg configuration. (Fppkg reports that the RTL is not installed.)

You could try to restore the configuration files automatically, or adapt fppkg.cfg manually.

Use this screen to generate new Fppkg configuration files with the fpcmkcfg tool.

The prefix of the Free Pascal Compiler installation is required. For example it has the units "lib/fpc" and/or "lib64/fpc"

```
Warning: empty Free Pascal compiler prefix.
Warning: The file "~/.fppkg/config/default" is not writable.
Warning: The file "/home/pi/.config/fppkg.cfg" is not writable.

Files to be written:
Fppkg configuration: /home/pi/.config/fppkg.cfg
Fppkg compiler configuration: ~/.fppkg/config/default
```

Warnings have to be resolved first

Figure 52: Read carefully

2. If either or both paths to which Lazarus needs to create or restore the configuration files associated with the Free Pascal RTL are shown with a warning stating **"not writable"**, then create each respective path manually as follows in 3. and 4. below.

3. Create an empty `fppkg.cfg` in `/home/pi/.config` using the Raspbian File Manager with hidden files made visible via `'Show Hidden'`
(also be sure to set the permissions associated with `fppkg.cfg` to `'Change Content' = Anyone`)

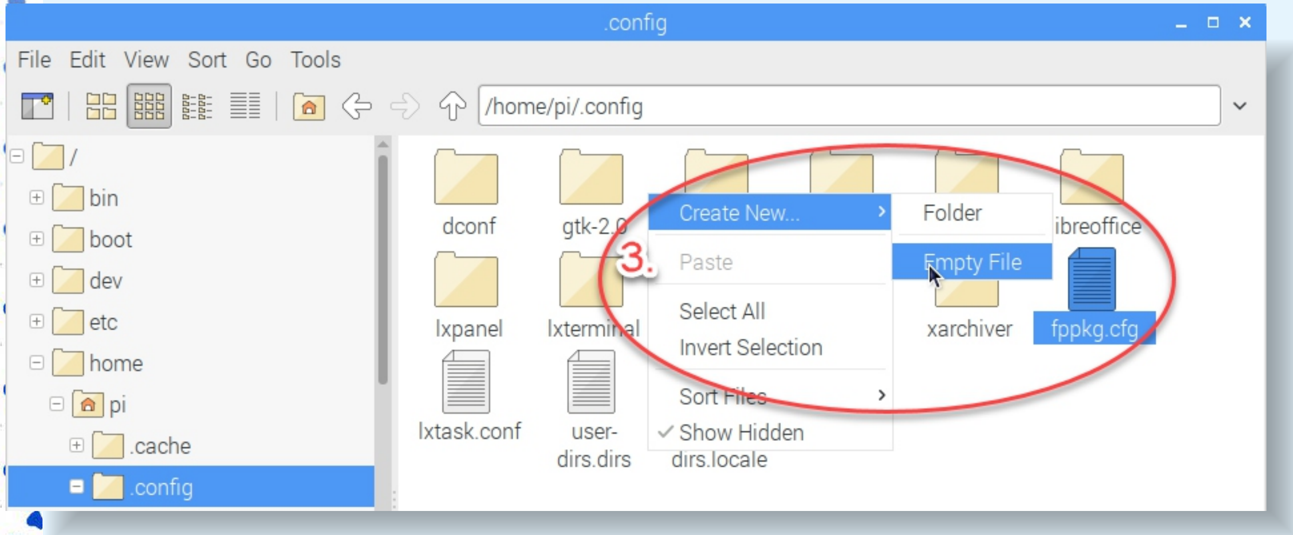
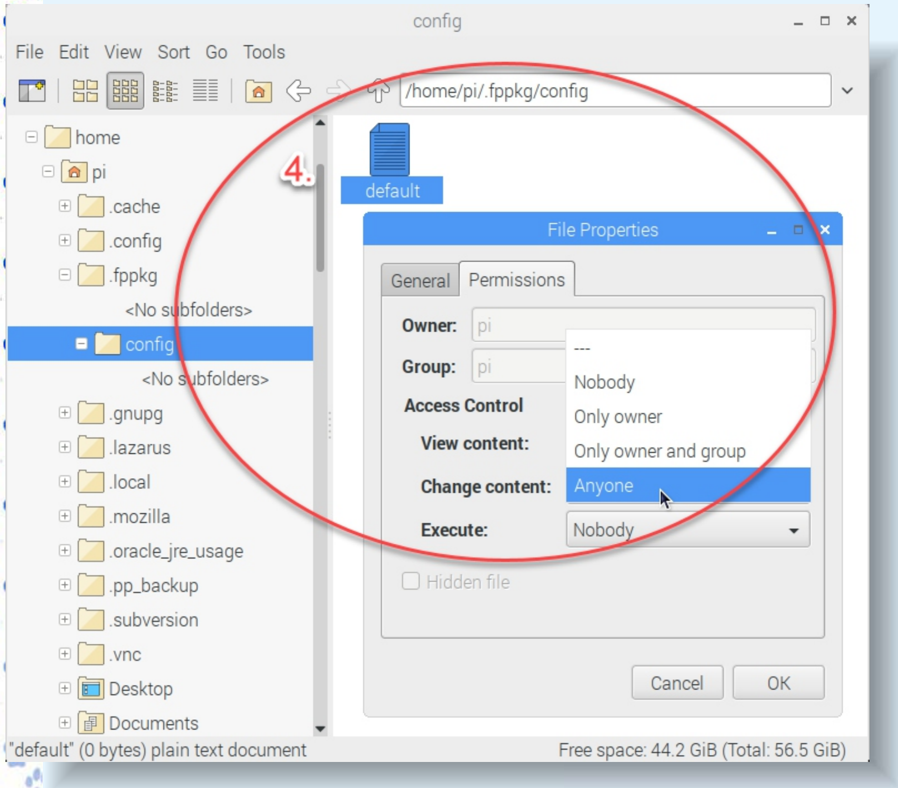


Figure 53: the files...

4. Create an empty `default` file under `/home/pi/.fppkg/config` using the Raspbian File Manager with hidden files made visible via `'Show Hidden'`.
Be sure to set the permissions associated with `default` to `'Change Content' = Anyone`)

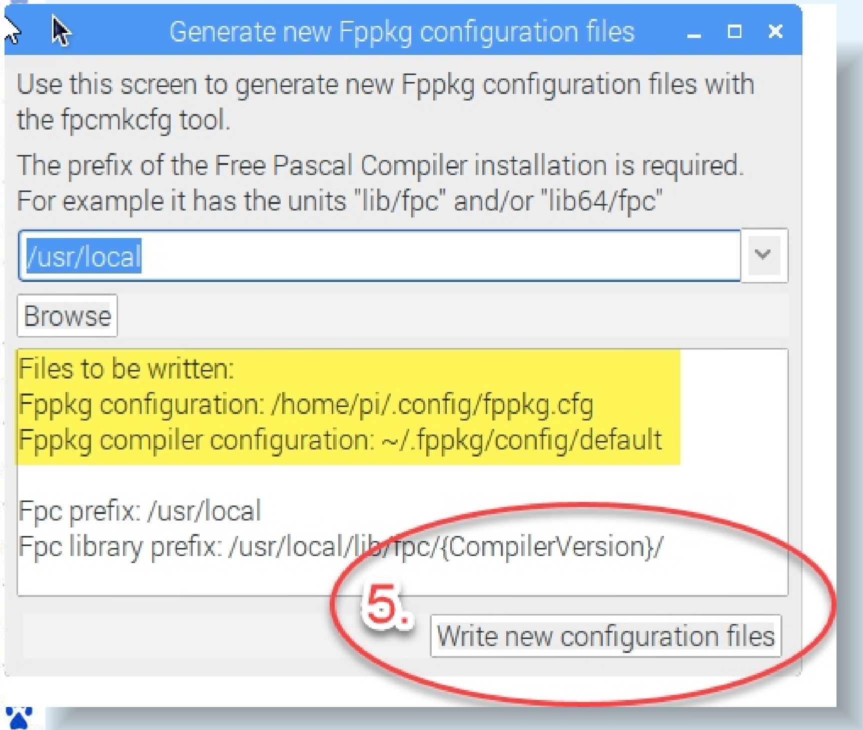


Owner: pi
Group: pi
Access Control
View content:
Change content: Anyone
Execute: Nobody
□ Hidden file

"default" (0 bytes) plain text document
Free space: 44.2 GiB (Total: 56.5 GiB)

Figure 54: the file propertys can be set

5. With both requisite files now writable, refresh the **"Generate new Fppkg configuration files"** by closing the dialogue and relaunching it by again clicking **"Restore Fppkg configuration"**. Both paths to the Free Pascal RTL configuration files should now appear ready to be `written/update`d as shown hereafter. If so, click: **`"Write new configuration files"`**.



Generate new Fppkg configuration files

Use this screen to generate new Fppkg configuration files with the fpcmkcfg tool.

The prefix of the Free Pascal Compiler installation is required. For example it has the units "lib/fpc" and/or "lib64/fpc"

/usr/local

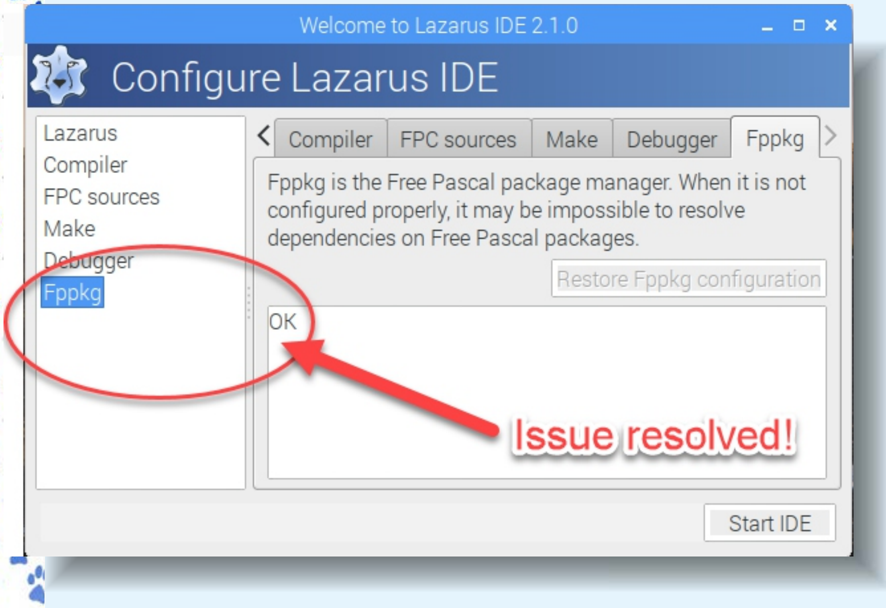
Browse

Files to be written:
Fppkg configuration: /home/pi/.config/fppkg.cfg
Fppkg compiler configuration: ~/.fppkg/config/default

Fpc prefix: /usr/local
Fpc library prefix: /usr/local/lib/fpc/{CompilerVersion}/

Figure 55:

6. If all goes well after clicking **"Write new configuration files"**, you should see the following to confirm all is ready to launch your new Lazarus installation on the Raspberry Pi.



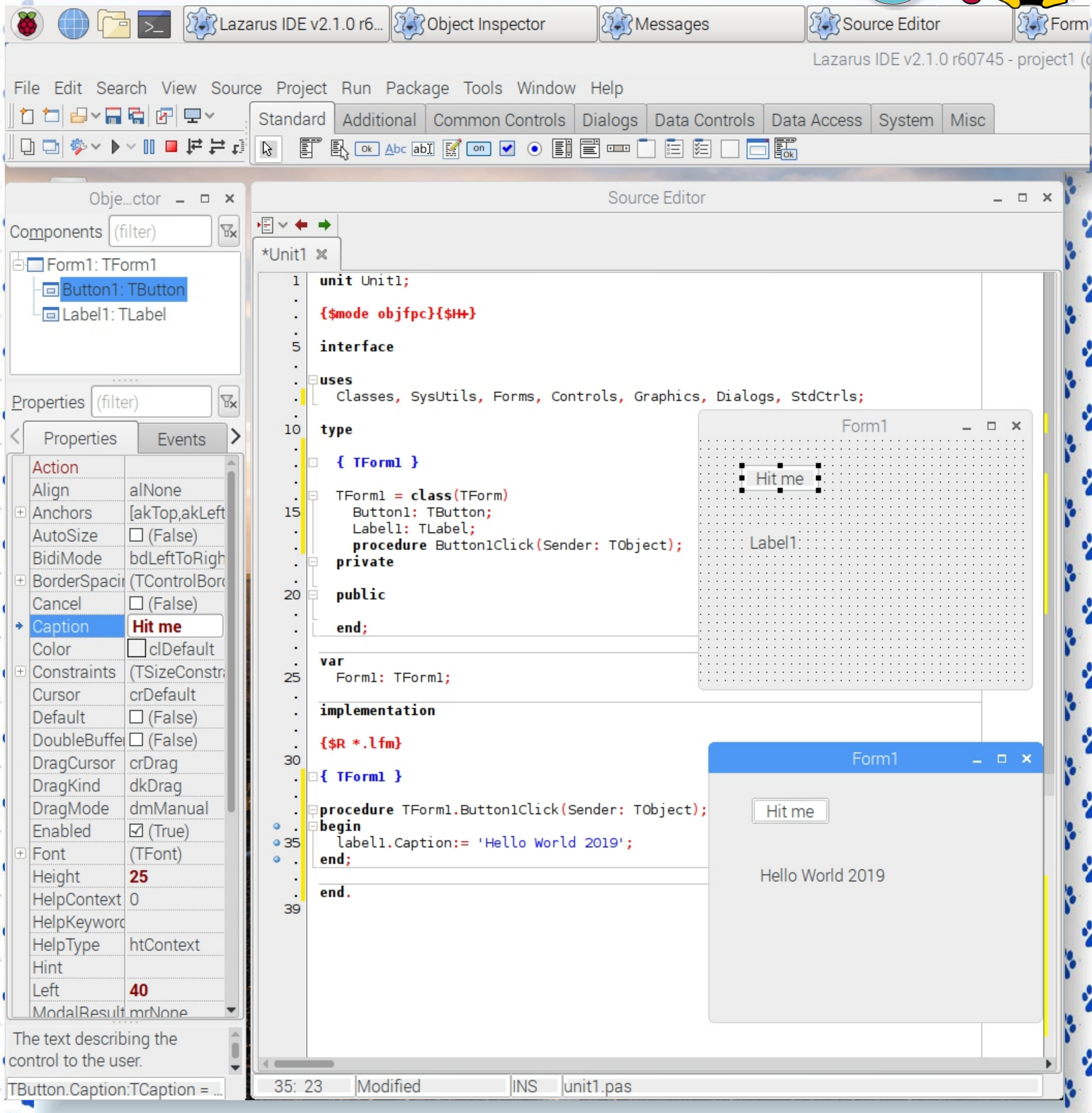
Welcome to Lazarus IDE 2.1.0

Configure Lazarus IDE

Fppkg is the Free Pascal package manager. When it is not configured properly, it may be impossible to resolve dependencies on Free Pascal packages.

OK

Figure 56:



```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

type

  { TForm1 }

  TForm1 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
  private

  public

  end;

var
  Form1: TForm1;

implementation

{$R *.lfm}

{ TForm1 }

procedure TForm1.Button1Click(Sender: TObject);
begin
  label1.Caption:= 'Hello World 2019';
end;

end.
```

Action	
Align	alNone
Anchors	[akTop,akLeft
AutoSize	□ (False)
BidiMode	bdLeftToRigh
BorderSpaci	(TControlBor
Cancel	□ (False)
Caption	Hit me
Color	□ clDefault
Constraints	(TSizeConstr
Cursor	crDefault
Default	□ (False)
DoubleBuffe	□ (False)
DragCursor	crDrag
DragKind	dkDrag
DragMode	dmManual
Enabled	☑ (True)
Font	(TFont)
Height	25
HelpContext	0
HelpKeyword	
HelpType	htContext
Hint	
Left	40
ModalResult	mrNone

The text describing the control to the user.

TButton.Caption:TCaption = ...

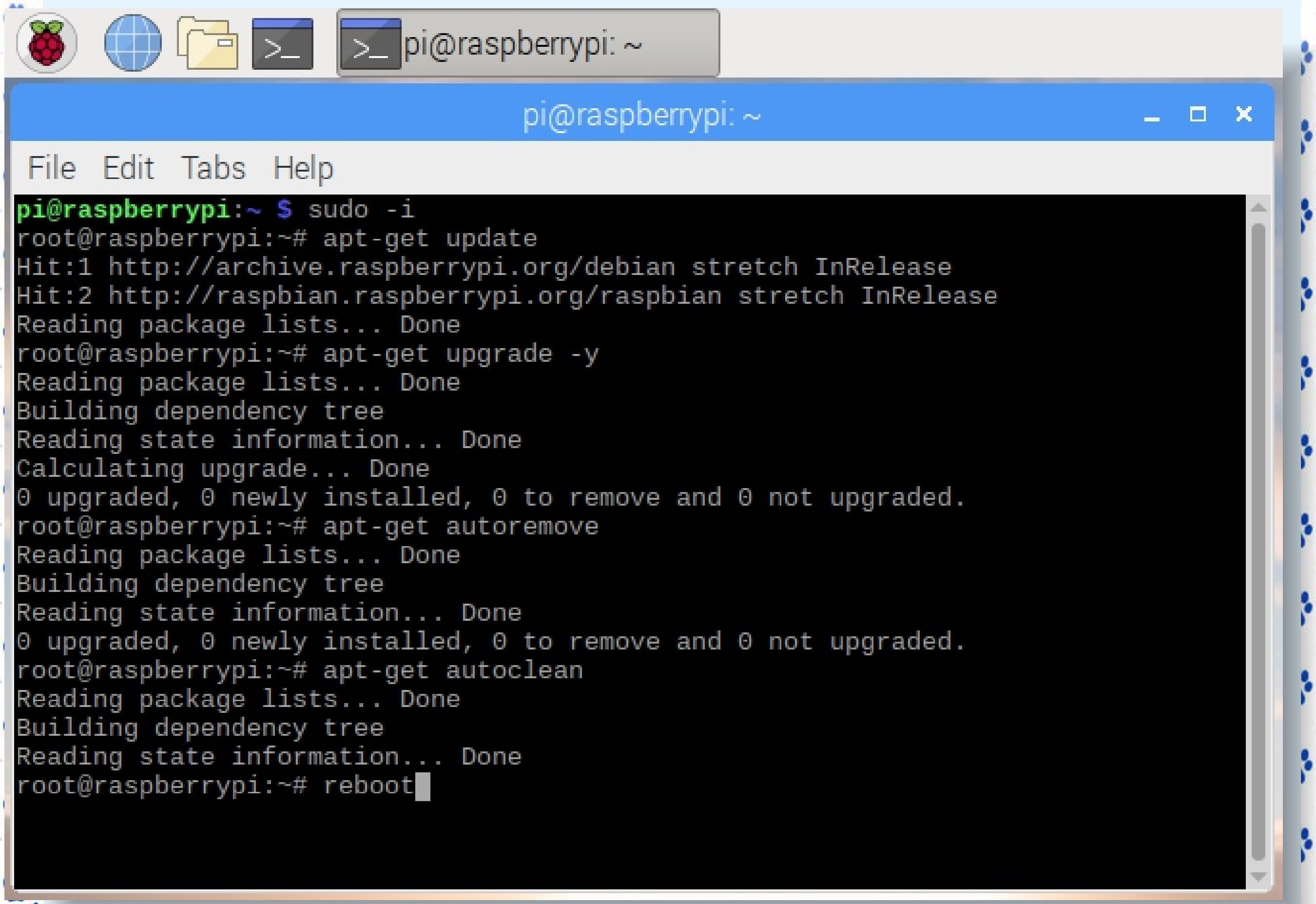
35: 23 Modified INS unit1.pas

Figure 57: Hello world program in Lazarus

Here after the script follows...

THE SCRIPT Scripted steps by Don Wilbrink.

Based on clean 2018-11-13-raspbian-stretch.img
Total install time: 35 minutes (from complete scratch)
by Don Wilbrink (2019-03-01)



```
pi@raspberrypi:~ $ sudo -i
root@raspberrypi:~# apt-get update
Hit:1 http://archive.raspberrypi.org/debian stretch InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease
Reading package lists... Done
root@raspberrypi:~# apt-get upgrade -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@raspberrypi:~# apt-get autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@raspberrypi:~# apt-get autoclean
Reading package lists... Done
Building dependency tree
Reading state information... Done
root@raspberrypi:~# reboot
```

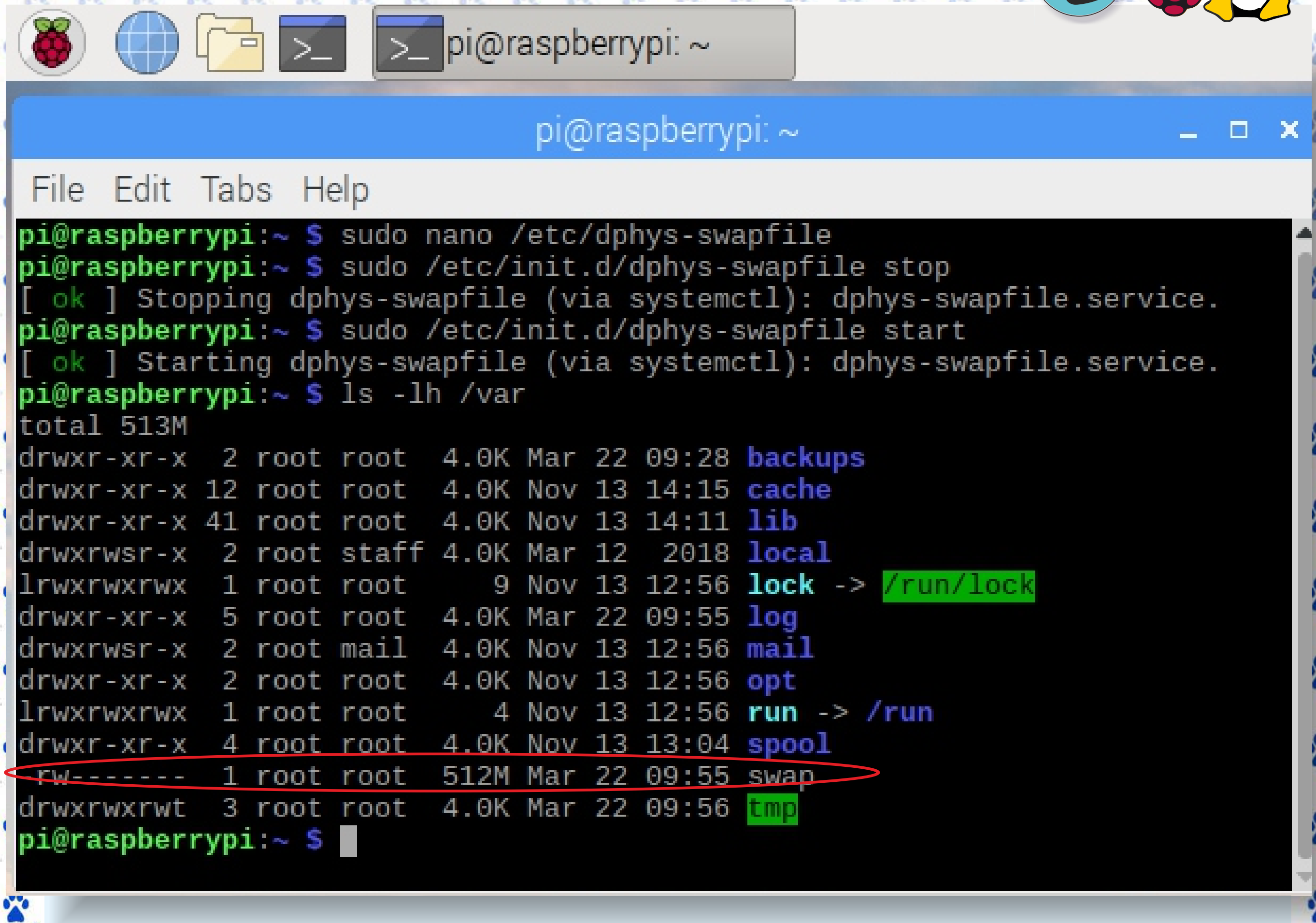
Figure 58 : Screenshot from a Raspberry Pi 3 Model B.

Start: **Update and Clean up**

```
sudo -i                   // -i, --login :run login shell as the target user
  apt-get update          // retrieve new list of packages
  apt-get upgrade -y      // perform a upgrade
  apt-get autoremove      // remove automatically all unused packages
  apt-get autoclean       // erase old downloaded archive files
reboot
```

We NEED to extend the swapfile (1,5 minute)

```
sudo nano /etc/dphys-swapfile
   // CONF_SWAPSIZE=100 # <-- change 100 in this line into 512
,  // you need to save the file using <CTRL-O><ENTER><CTRL-X>
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
  // Check
ls -lh /var
```

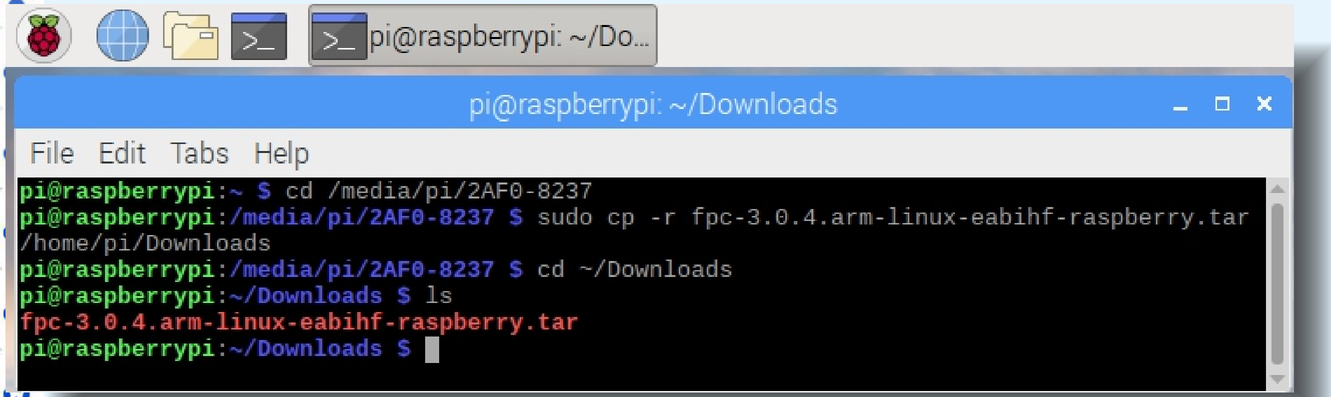


```
pi@raspberrypi:~ $ sudo nano /etc/dphys-swapfile
pi@raspberrypi:~ $ sudo /etc/init.d/dphys-swapfile stop
[ ok ] Stopping dphys-swapfile (via systemctl): dphys-swapfile.service.
pi@raspberrypi:~ $ sudo /etc/init.d/dphys-swapfile start
[ ok ] Starting dphys-swapfile (via systemctl): dphys-swapfile.service.
pi@raspberrypi:~ $ ls -lh /var
total 513M
drwxr-xr-x  2 root root  4.0K Mar 22 09:28 backups
drwxr-xr-x 12 root root  4.0K Nov 13 14:15 cache
drwxr-xr-x 41 root root  4.0K Nov 13 14:11 lib
drwxrwsr-x  2 root staff 4.0K Mar 12  2018 local
lrwxrwxrwx  1 root root     9 Nov 13 12:56 lock -> /run/lock
drwxr-xr-x  5 root root  4.0K Mar 22 09:55 log
drwxrwsr-x  2 root mail  4.0K Nov 13 12:56 mail
drwxr-xr-x  2 root root  4.0K Nov 13 12:56 opt
lrwxrwxrwx  1 root root     4 Nov 13 12:56 run -> /run
drwxr-xr-x  4 root root  4.0K Nov 13 13:04 spool
-rw-------  1 root root  512M Mar 22 09:55 swap
drwxrwxrwt  3 root root  4.0K Mar 22 09:56 tmp
pi@raspberrypi:~ $
```

Figure 59: swap changed finally changed to 512M

Copy fpc-3.0.4.arm.linux-eabhif-raspberry.tar from your USB stick
to the RaspBerry Pi. path: /home/pi/Downloads folder. (1,5 min)

```
cd /media/pi/Your-medianame
sudo cp -r fpc-3.0.4.arm-linux-eabihf-raspberry.tar<space>
           /home/pi/Downloads
cd ~/Downloads
// Check
ls
```

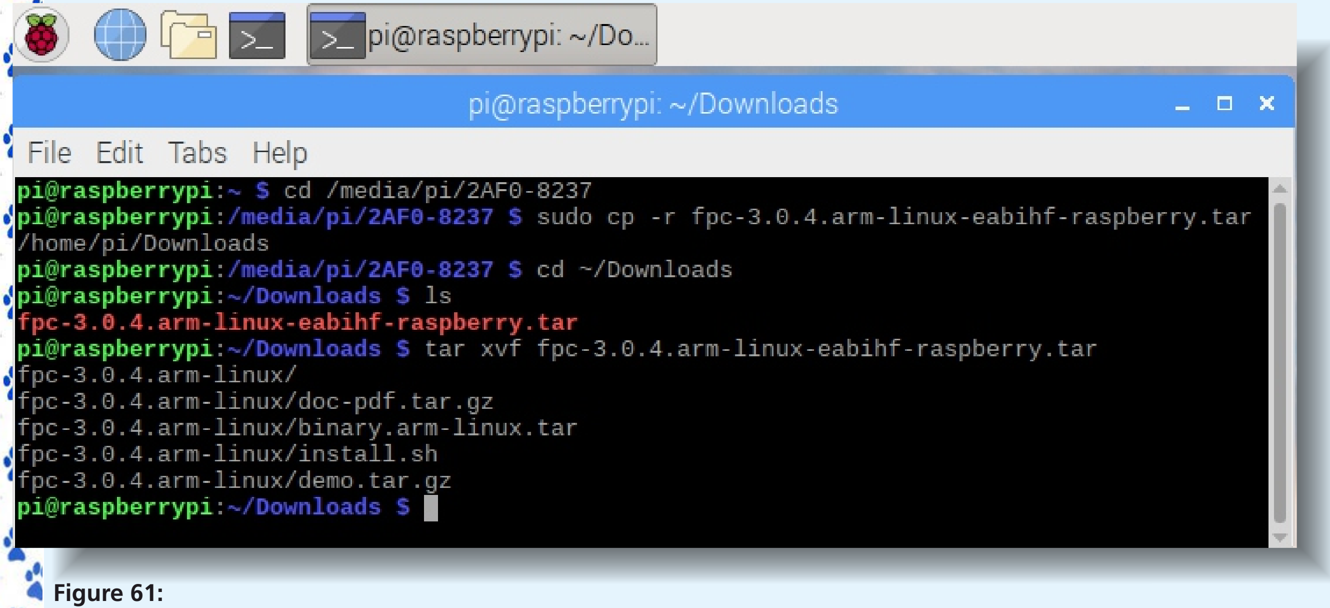


```
pi@raspberrypi:~ $ cd /media/pi/2AF0-8237
pi@raspberrypi:/media/pi/2AF0-8237 $ sudo cp -r fpc-3.0.4.arm-linux-eabihf-raspberry.tar
/home/pi/Downloads
pi@raspberrypi:/media/pi/2AF0-8237 $ cd ~/Downloads
pi@raspberrypi:~/Downloads $ ls
fpc-3.0.4.arm-linux-eabihf-raspberry.tar
pi@raspberrypi:~/Downloads $ 
```

Figure 60:

Here extract the fpc-3.0.4.arm-linux-eabihf-raspberry.tar file

```
tar xvf fpc-3.0.4.arm-linux-eabihf-raspberry.tar
```

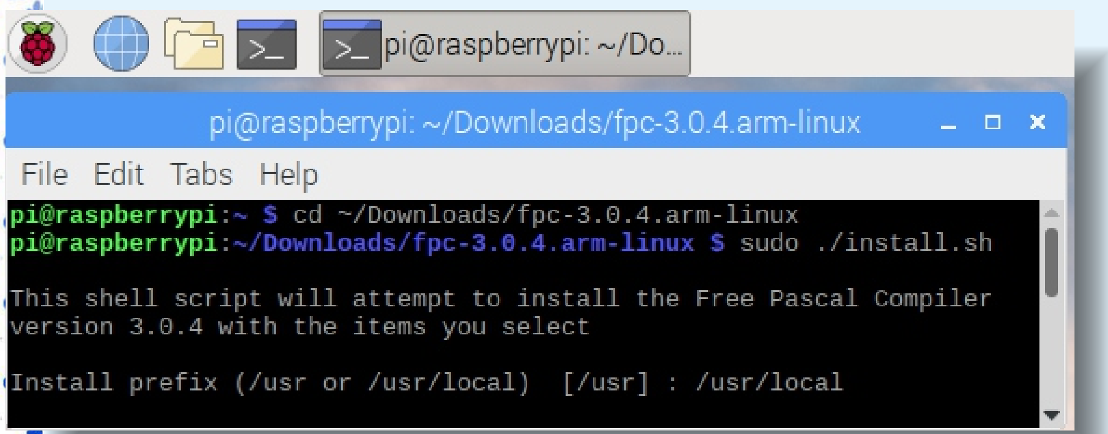


```
pi@raspberrypi:~ $ cd /media/pi/2AF0-8237
pi@raspberrypi:/media/pi/2AF0-8237 $ sudo cp -r fpc-3.0.4.arm-linux-eabihf-raspberry.tar
/home/pi/Downloads
pi@raspberrypi:/media/pi/2AF0-8237 $ cd ~/Downloads
pi@raspberrypi:~/Downloads $ ls
fpc-3.0.4.arm-linux-eabihf-raspberry.tar
pi@raspberrypi:~/Downloads $ tar xvf fpc-3.0.4.arm-linux-eabihf-raspberry.tar
fpc-3.0.4.arm-linux/
fpc-3.0.4.arm-linux/doc-pdf.tar.gz
fpc-3.0.4.arm-linux/binary.arm-linux.tar
fpc-3.0.4.arm-linux/install.sh
fpc-3.0.4.arm-linux/demo.tar.gz
pi@raspberrypi:~/Downloads $ 
```

Figure 61:

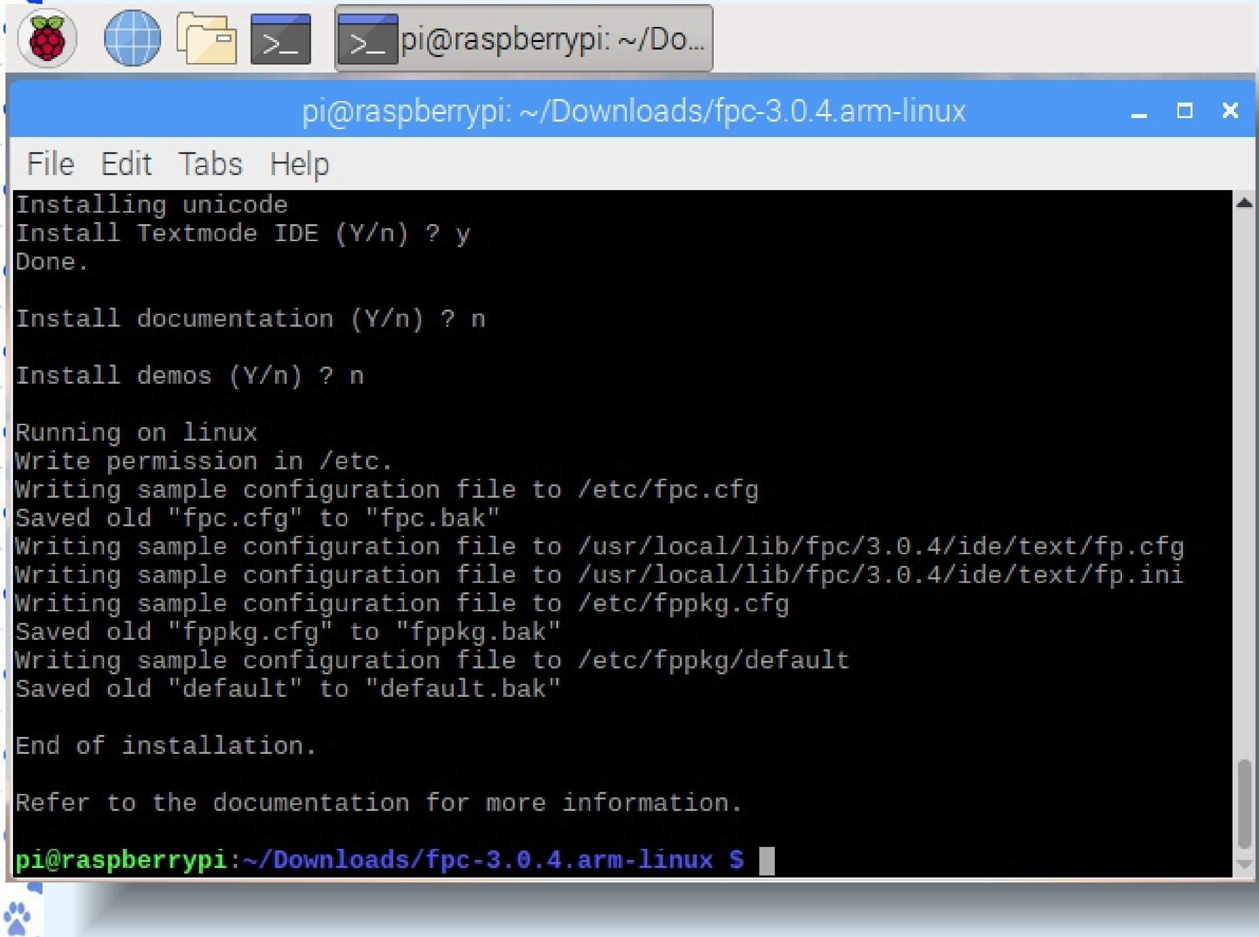
Install it. Choose installation in / usr / local: Choose y, n, n for the questions (1 min)

```
cd ~/Downloads/fpc-3.0.4.arm-linux
sudo ./install.sh
```



```
pi@raspberrypi:~ $ cd ~/Downloads/fpc-3.0.4.arm-linux
pi@raspberrypi:~/Downloads/fpc-3.0.4.arm-linux $ sudo ./install.sh

This shell script will attempt to install the Free Pascal Compiler
version 3.0.4 with the items you select

Install prefix (/usr or /usr/local)  [/usr] : /usr/local
```

Figure 62: Start

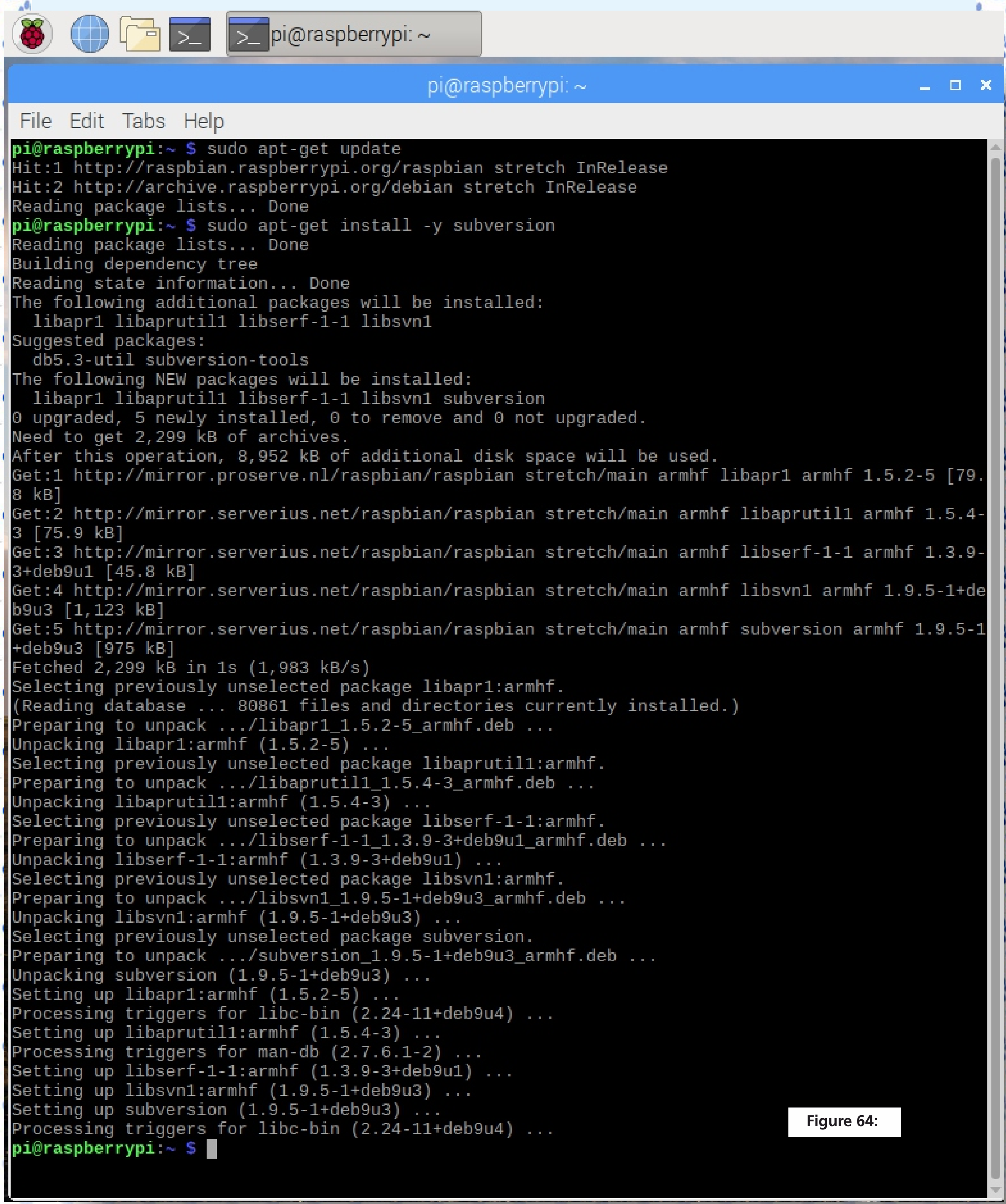


```
Installing unicode
Install Textmode IDE (Y/n) ? y
Done.

Install documentation (Y/n) ? n

Install demos (Y/n) ? n

Running on linux
Write permission in /etc.
Writing sample configuration file to /etc/fpc.cfg
Saved old "fpc.cfg" to "fpc.bak"
Writing sample configuration file to /usr/local/lib/fpc/3.0.4/ide/text/fp.cfg
Writing sample configuration file to /usr/local/lib/fpc/3.0.4/ide/text/fp.ini
Writing sample configuration file to /etc/fppkg.cfg
Saved old "fppkg.cfg" to "fppkg.bak"
Writing sample configuration file to /etc/fppkg/default
Saved old "default" to "default.bak"

End of installation.

Refer to the documentation for more information.

pi@raspberrypi:~/Downloads/fpc-3.0.4.arm-linux $ 
```

Figure 63: Finish

Install subversion (20 sec)

```
sudo apt-get update
sudo apt-get install -y subversion
```



```
pi@raspberrypi:~ $ sudo apt-get update
Hit:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease
Hit:2 http://archive.raspberrypi.org/debian stretch InRelease
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get install -y subversion
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapr1 libaprutil1 libserf-1-1 libsvn1
Suggested packages:
  db5.3-util subversion-tools
The following NEW packages will be installed:
  libapr1 libaprutil1 libserf-1-1 libsvn1 subversion
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,299 kB of archives.
After this operation, 8,952 kB of additional disk space will be used.
Get:1 http://mirror.proserve.nl/raspbian/raspbian stretch/main armhf libapr1 armhf 1.5.2-5 [79.
8 kB]
Get:2 http://mirror.serverius.net/raspbian/raspbian stretch/main armhf libaprutil1 armhf 1.5.4-
3 [75.9 kB]
Get:3 http://mirror.serverius.net/raspbian/raspbian stretch/main armhf libserf-1-1 armhf 1.3.9-
3+deb9u1 [45.8 kB]
Get:4 http://mirror.serverius.net/raspbian/raspbian stretch/main armhf libsvn1 armhf 1.9.5-1+de
b9u3 [1,123 kB]
Get:5 http://mirror.serverius.net/raspbian/raspbian stretch/main armhf subversion armhf 1.9.5-1
+deb9u3 [975 kB]
Fetched 2,299 kB in 1s (1,983 kB/s)
Selecting previously unselected package libapr1:armhf.
(Reading database ... 80861 files and directories currently installed.)
Preparing to unpack .../libapr1_1.5.2-5_armhf.deb ...
Unpacking libapr1:armhf (1.5.2-5) ...
Selecting previously unselected package libaprutil1:armhf.
Preparing to unpack .../libaprutil1_1.5.4-3_armhf.deb ...
Unpacking libaprutil1:armhf (1.5.4-3) ...
Selecting previously unselected package libserf-1-1:armhf.
Preparing to unpack .../libserf-1-1_1.3.9-3+deb9u1_armhf.deb ...
Unpacking libserf-1-1:armhf (1.3.9-3+deb9u1) ...
Selecting previously unselected package libsvn1:armhf.
Preparing to unpack .../libsvn1_1.9.5-1+deb9u3_armhf.deb ...
Unpacking libsvn1:armhf (1.9.5-1+deb9u3) ...
Selecting previously unselected package subversion.
Preparing to unpack .../subversion_1.9.5-1+deb9u3_armhf.deb ...
Unpacking subversion (1.9.5-1+deb9u3) ...
Setting up libapr1:armhf (1.5.2-5) ...
Processing triggers for libc-bin (2.24-11+deb9u4) ...
Setting up libaprutil1:armhf (1.5.4-3) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up libserf-1-1:armhf (1.3.9-3+deb9u1) ...
Setting up libsvn1:armhf (1.9.5-1+deb9u3) ...
Setting up subversion (1.9.5-1+deb9u3) ...
Processing triggers for libc-bin (2.24-11+deb9u4) ...
pi@raspberrypi:~ $
```

Figure 64:

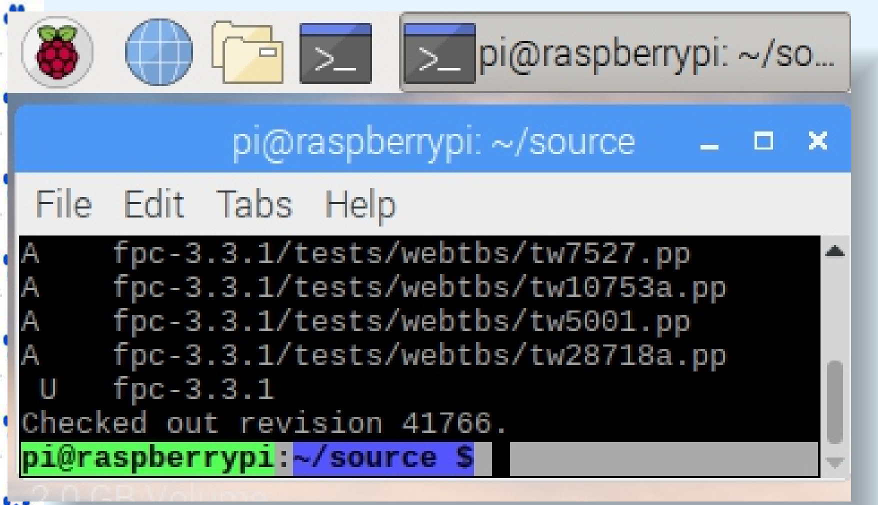
Make a source folder in root: PI

```
Cd ~
mkdir source
```

FREE PASCAL COMPILER (FPC) VERSION 3.3.1

Download fpc 3.3.1 source and place it in the new folder ~ / source / fpc-3.3.1: (2.5 min)

```
cd ~/source
svn co http://svn.freepascal.org/svn/fpc/trunk fpc-3.3.1
```

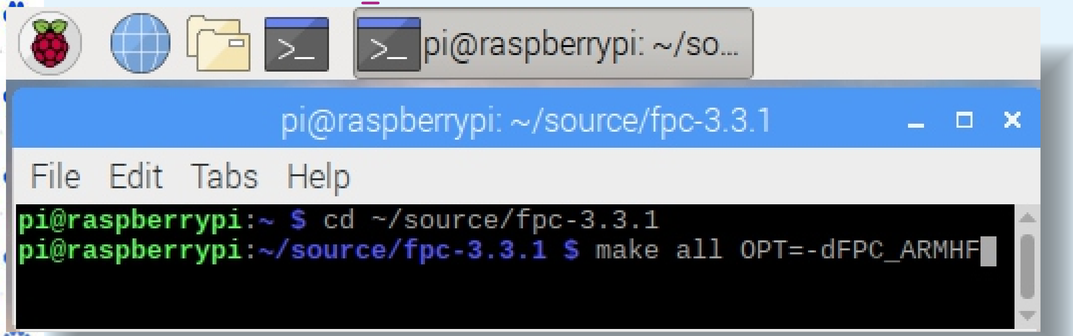


```
A     fpc-3.3.1/tests/webtbs/tw7527.pp
A     fpc-3.3.1/tests/webtbs/tw10753a.pp
A     fpc-3.3.1/tests/webtbs/tw5001.pp
A     fpc-3.3.1/tests/webtbs/tw28718a.pp
 U    fpc-3.3.1
Checked out revision 41766.
pi@raspberrypi:~/source $
```

Figure 65:

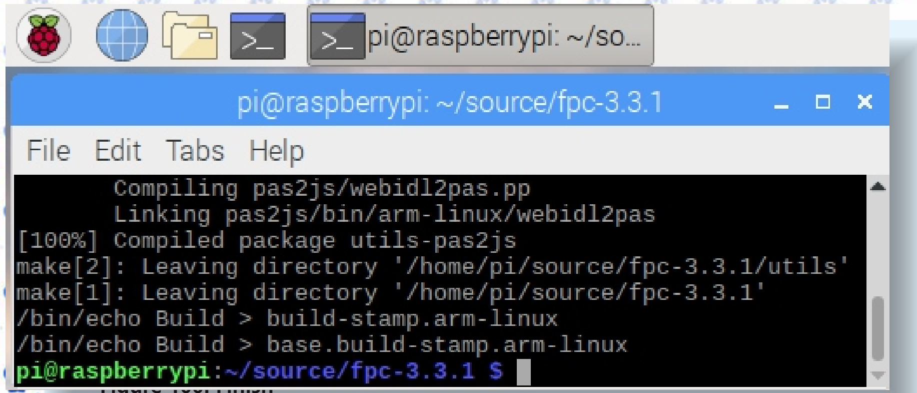
Compiling fpc 3.3.1 : (16 min)

```
cd ~/source/fpc-3.3.1
make all OPT=-dFPC_ARMHF     // (15 min)
```



```
pi@raspberrypi:~ $ cd ~/source/fpc-3.3.1
pi@raspberrypi:~/source/fpc-3.3.1 $ make all OPT=-dFPC_ARMHF
```

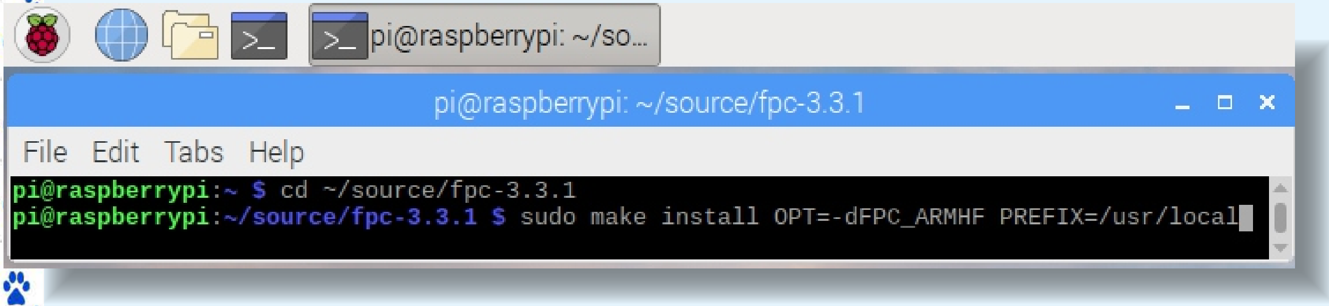
Figure 66: Start



```
        Compiling pas2js/webidl2pas.pp
        Linking pas2js/bin/arm-linux/webidl2pas
[100%] Compiled package utils-pas2js
make[2]: Leaving directory '/home/pi/source/fpc-3.3.1/utils'
make[1]: Leaving directory '/home/pi/source/fpc-3.3.1'
/bin/echo Build > build-stamp.arm-linux
/bin/echo Build > base.build-stamp.arm-linux
pi@raspberrypi:~/source/fpc-3.3.1 $
```

Figure 67: Finish

```
sudo make install OPT=-dFPC_ARMHF PREFIX=/usr/local //(1 min)
```

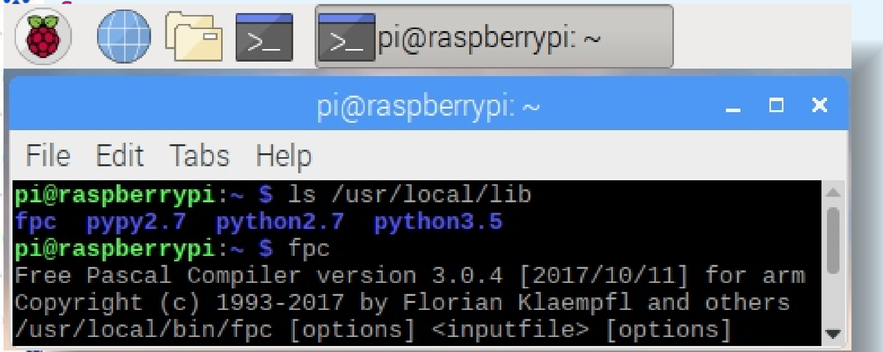


```
pi@raspberrypi:~ $ cd ~/source/fpc-3.3.1
pi@raspberrypi:~/source/fpc-3.3.1 $ sudo make install OPT=-dFPC_ARMHF PREFIX=/usr/local
```

Figure 68: Final

Check and test FPC version

```
ls /usr/local/lib/fpc
```

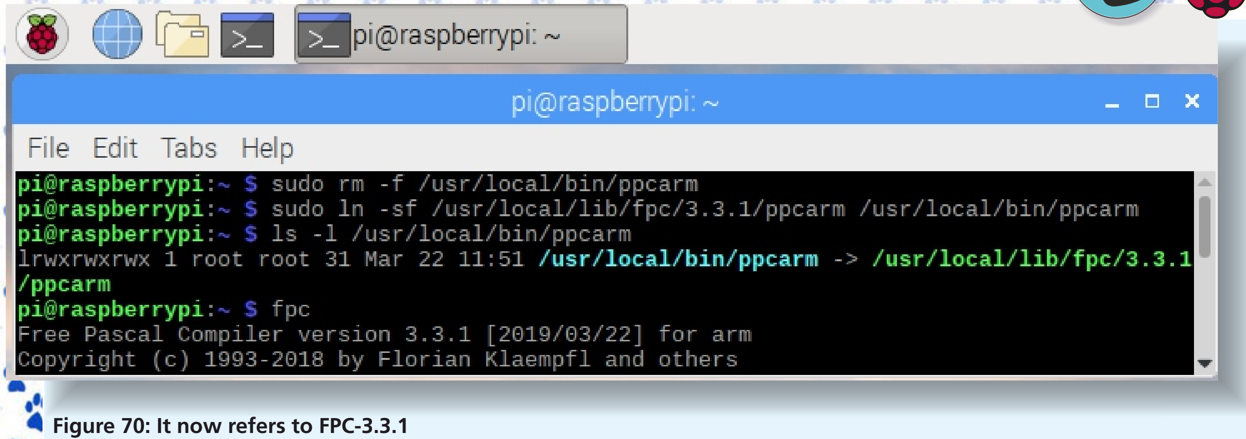


```
pi@raspberrypi:~ $ ls /usr/local/lib
fpc  pypy2.7  python2.7  python3.5
pi@raspberrypi:~ $ fpc
Free Pascal Compiler version 3.0.4 [2017/10/11] for arm
Copyright (c) 1993-2017 by Florian Klaempfl and others
/usr/local/bin/fpc [options] <inputfile> [options]
```

Figure 69: It still refers to fpc-3.0.4; see figure above

Create Symbolise link:

```
sudo rm -f /usr/local/bin/ppcarm
sudo ln -sf /usr/local/lib/fpc/3.3.1/ppcarm /usr/local/bin/ppcarm
// Check:
ls -l /usr/local/bin/ppcarm
```



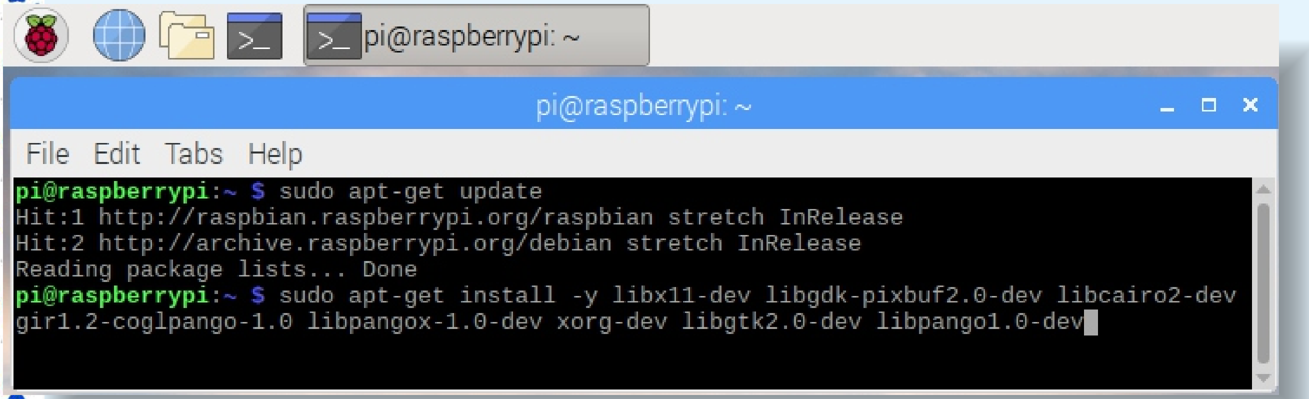
```
pi@raspberrypi:~ $ sudo rm -f /usr/local/bin/ppcarm
pi@raspberrypi:~ $ sudo ln -sf /usr/local/lib/fpc/3.3.1/ppcarm /usr/local/bin/ppcarm
pi@raspberrypi:~ $ ls -l /usr/local/bin/ppcarm
lrwxrwxrwx 1 root root 31 Mar 22 11:51 /usr/local/bin/ppcarm -> /usr/local/lib/fpc/3.3.1
/ppcarm
pi@raspberrypi:~ $ fpc
Free Pascal Compiler version 3.3.1 [2019/03/22] for arm
Copyright (c) 1993-2018 by Florian Klaempfl and others
```

Figure 70: It now refers to FPC-3.3.1

LAZARUS

Installing extra packages for lazarus: (1:50 min)

```
sudo apt-get update
sudo apt-get install -y libx11-dev libgdk-pixbuf2.0-dev
   libcairo2-dev gir1.2-coglpango-1.0 libpangox-1.0-dev
                 xorg-dev libgtk2.0-dev libpango1.0-dev
```

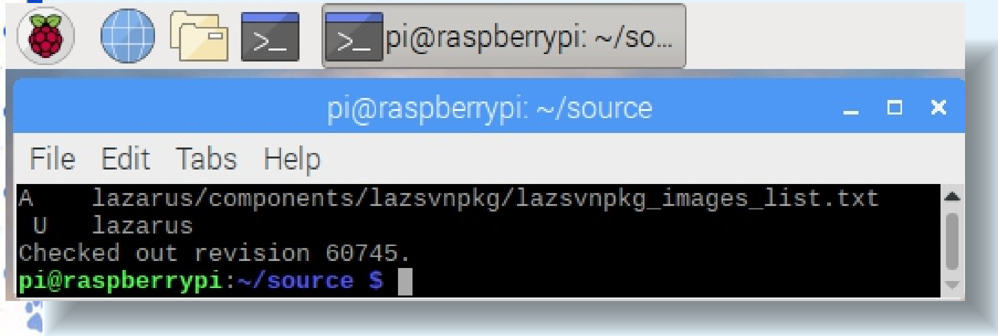


```
pi@raspberrypi:~ $ sudo apt-get update
Hit:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease
Hit:2 http://archive.raspberrypi.org/debian stretch InRelease
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get install -y libx11-dev libgdk-pixbuf2.0-dev libcairo2-dev
gir1.2-coglpango-1.0 libpangox-1.0-dev xorg-dev libgtk2.0-dev libpango1.0-dev
```

Figure 71: It now refers to FPC-3.3.1

Retrieve and place lazarus source in a new folder ~ / source / lazarus: (2 min)

```
cd ~/source
svn co http://svn.freepascal.org/svn/lazarus/trunk lazarus
```

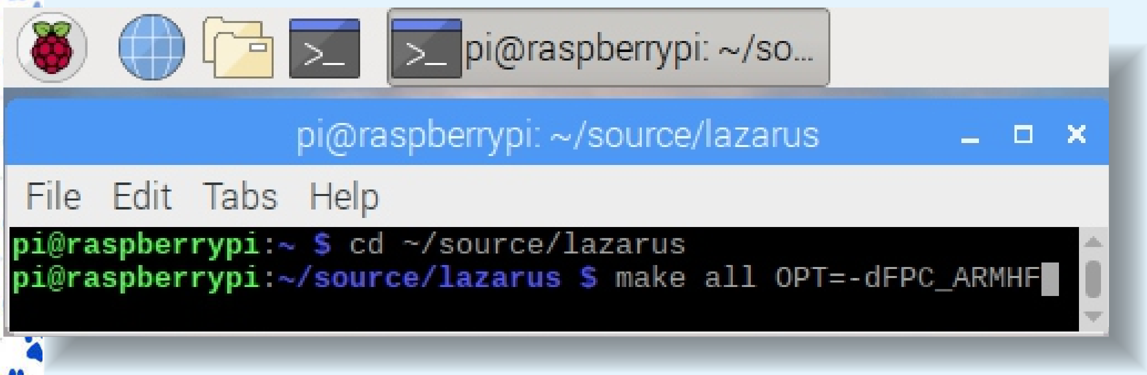


```
A    lazarus/components/lazsvnpkg/lazsvnpkg_images_list.txt
 U   lazarus
Checked out revision 60745.
pi@raspberrypi:~/source $
```

Figure 72: It now refers to FPC-3.3.1

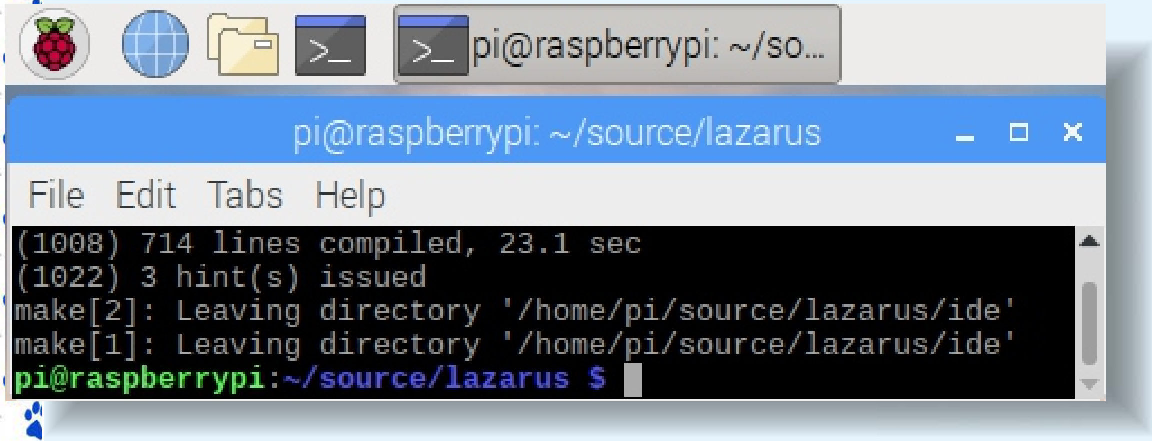
Lazarus compiling: (10 min)

```
cd ~/source/lazarus
make all OPT=-dFPC_ARMHF  (8:20 min)
```



```
pi@raspberrypi:~ $ cd ~/source/lazarus
pi@raspberrypi:~/source/lazarus $ make all OPT=-dFPC_ARMHF
```

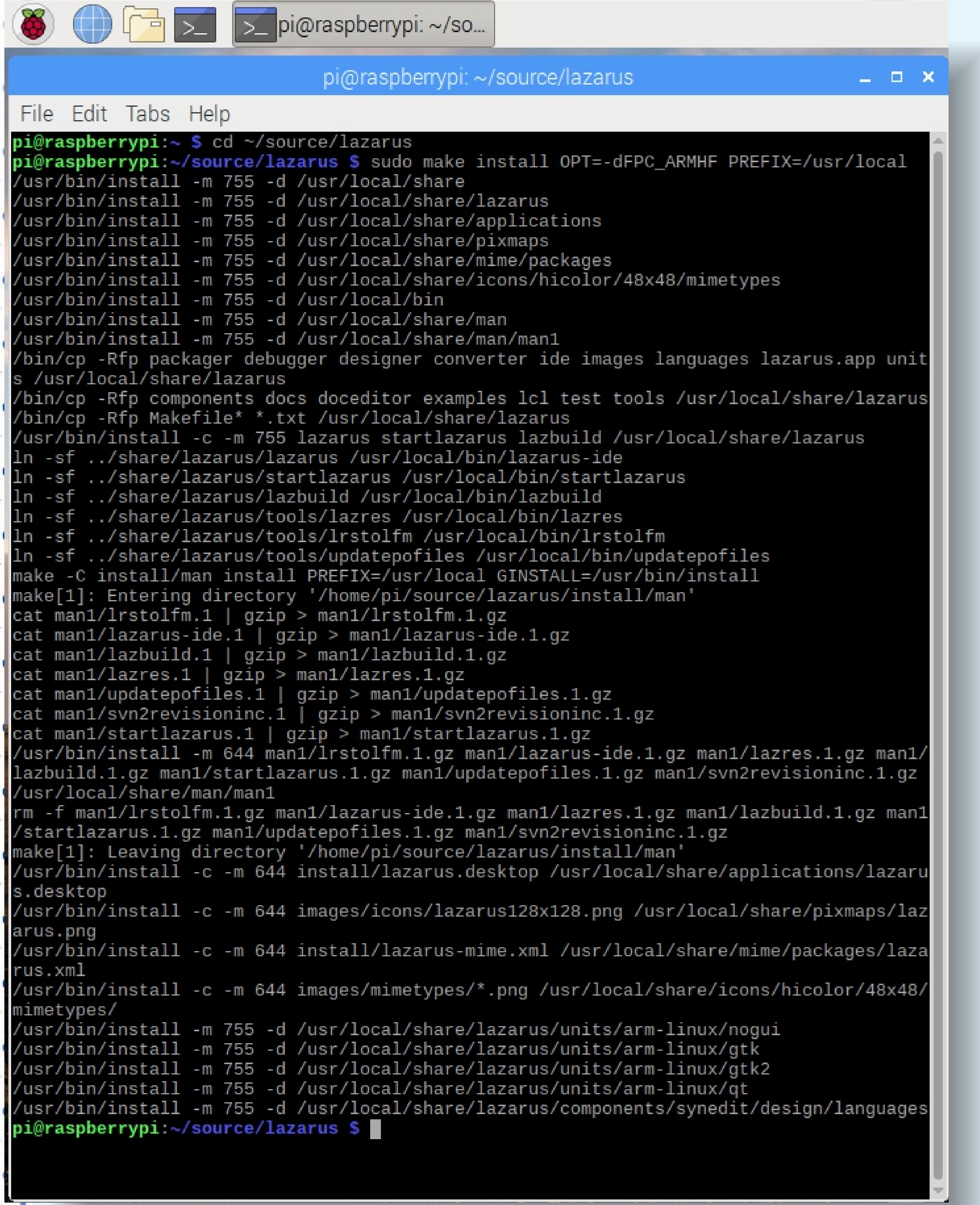
Figure 73: Start



```
(1008) 714 lines compiled, 23.1 sec
(1022) 3 hint(s) issued
make[2]: Leaving directory '/home/pi/source/lazarus/ide'
make[1]: Leaving directory '/home/pi/source/lazarus/ide'
pi@raspberrypi:~/source/lazarus $ 
```

Figure 74: End

```
sudo make install OPT=-dFPC_ARMHF PREFIX=/usr/local (2 min)
```



```
pi@raspberrypi:~ $ cd ~/source/lazarus
pi@raspberrypi:~/source/lazarus $ sudo make install OPT=-dFPC_ARMHF PREFIX=/usr/local
/usr/bin/install -m 755 -d /usr/local/share
/usr/bin/install -m 755 -d /usr/local/share/lazarus
/usr/bin/install -m 755 -d /usr/local/share/applications
/usr/bin/install -m 755 -d /usr/local/share/pixmaps
/usr/bin/install -m 755 -d /usr/local/share/mime/packages
/usr/bin/install -m 755 -d /usr/local/share/icons/hicolor/48x48/mimetypes
/usr/bin/install -m 755 -d /usr/local/bin
/usr/bin/install -m 755 -d /usr/local/share/man
/usr/bin/install -m 755 -d /usr/local/share/man/man1
/bin/cp -Rfp packager debugger designer converter ide images languages lazarus.app unit
s /usr/local/share/lazarus
/bin/cp -Rfp components docs doceditor examples lcl test tools /usr/local/share/lazarus
/bin/cp -Rfp Makefile* *.txt /usr/local/share/lazarus
/usr/bin/install -c -m 755 lazarus startlazarus lazbuild /usr/local/share/lazarus
ln -sf ../share/lazarus/lazarus /usr/local/bin/lazarus-ide
ln -sf ../share/lazarus/startlazarus /usr/local/bin/startlazarus
ln -sf ../share/lazarus/lazbuild /usr/local/bin/lazbuild
ln -sf ../share/lazarus/tools/lazres /usr/local/bin/lazres
ln -sf ../share/lazarus/tools/lrstolfm /usr/local/bin/lrstolfm
ln -sf ../share/lazarus/tools/updatepofiles /usr/local/bin/updatepofiles
make -C install/man install PREFIX=/usr/local GINSTALL=/usr/bin/install
make[1]: Entering directory '/home/pi/source/lazarus/install/man'
cat man1/lrstolfm.1 | gzip > man1/lrstolfm.1.gz
cat man1/lazarus-ide.1 | gzip > man1/lazarus-ide.1.gz
cat man1/lazbuild.1 | gzip > man1/lazbuild.1.gz
cat man1/lazres.1 | gzip > man1/lazres.1.gz
cat man1/updatepofiles.1 | gzip > man1/updatepofiles.1.gz
cat man1/svn2revisioninc.1 | gzip > man1/svn2revisioninc.1.gz
cat man1/startlazarus.1 | gzip > man1/startlazarus.1.gz
/usr/bin/install -m 644 man1/lrstolfm.1.gz man1/lazarus-ide.1.gz man1/lazres.1.gz man1/
lazbuild.1.gz man1/startlazarus.1.gz man1/updatepofiles.1.gz man1/svn2revisioninc.1.gz
/usr/local/share/man/man1
rm -f man1/lrstolfm.1.gz man1/lazarus-ide.1.gz man1/lazres.1.gz man1/lazbuild.1.gz man1
/startlazarus.1.gz man1/updatepofiles.1.gz man1/svn2revisioninc.1.gz
make[1]: Leaving directory '/home/pi/source/lazarus/install/man'
/usr/bin/install -c -m 644 install/lazarus.desktop /usr/local/share/applications/lazaru
s.desktop
/usr/bin/install -c -m 644 images/icons/lazarus128x128.png /usr/local/share/pixmaps/laz
arus.png
/usr/bin/install -c -m 644 install/lazarus-mime.xml /usr/local/share/mime/packages/laza
rus.xml
/usr/bin/install -c -m 644 images/mimetypes/*.png /usr/local/share/icons/hicolor/48x48/
mimetypes/
/usr/bin/install -m 755 -d /usr/local/share/lazarus/units/arm-linux/nogui
/usr/bin/install -m 755 -d /usr/local/share/lazarus/units/arm-linux/gtk
/usr/bin/install -m 755 -d /usr/local/share/lazarus/units/arm-linux/gtk2
/usr/bin/install -m 755 -d /usr/local/share/lazarus/units/arm-linux/qt
/usr/bin/install -m 755 -d /usr/local/share/lazarus/components/synedit/design/languages
pi@raspberrypi:~/source/lazarus $ 
```

Figure 75

Remember to cut the swap size back to the default after the builds!

```
nano /etc/dphys-swapfile
// CONF_SWAPSIZE=100  <-- change 512 in this line to 100
/etc/init.d/dphys-swapfile stop
/etc/init.d/dphys-swapfile start
```

FINISH

Agenda:
09:00-09:30
Welcome with coffee/tea/cake

09:30-10:00 Introduction and roadmap - by Barnsten

10:00-11:30 - RAD Server in Action - by Jon Aasenden
Jon Aasenden is Embarcadero's Technical Lead. Get an introduction about the perfect Back-end for Delphi apps. Learn how to use RAD Server on a private Windows Server or cloud host on Amazon, Rackspace or Azure Clouds. And how you can easily integrate REST cloud services from a variety of cloud, social and BAAS platforms such as Google, Amazon, Facebook, Kinvey, Pars and more.

11:30-11:45 - Break

11:45-12:45 - Modernizing your Delphi Language- by Bob Swart
De average Delphi developer has many years of experience with Delphi, and well acquainted with the average capabilities of the Delphi language. However, in the past decade, a number of extensions and enhancements have been added to the Delphi language. In this session, Bob will explain and demonstrate some of the more powerful features that may be known, but are not ofen used, like generics, anonymous methods, record and class helpers, threading as well as the new in-place variable declaration possibilities, all in order to modernize our use of the Delphi language.

12:45-13:30 - Lunch

13:30-14:30 - Using Delphi as a high-performant transaction-engine in a retail cloud product - by Jan Jacobs
Tilroy is a SaaS cloud product for retailers selling in-store and online. The point of sales and the webshop are using the same data ,transaction- and promotionengine. Although the majority of the system is build using Java and nodejs, we use the power of Delphi to process transactions and manage complex promotions.Learn how we used Delphi in combination with other technologies like nodejs and mongodb to process a large number of transactions in shops and webshops.

14:30-14:50 - Break

14:50-15:50 - Go to the web and beyond with Delphi and TMS WEB Core - by Bruno Fierens

TMS WEB Core is a framework that enables to create rich web client applications from Delphi with RAD methodology.The Delphi developed web client can use all power of HTML/CSS to design spectacular interfaces. This same technology can now also be applied to create cross-platform installable & offline-usable smartphone applications as well as cross-platform desktop applications for Windows, macOS and Linux. In this session, we offer an overview and hands-on examples of all new powerful capabilities.

16:00-17:00 - 2020 Vision for Delphi - by Danny Wind

Futureproof your Delphi applications, start now to take the lead in 2020.
In this session I'll demo several tecniques to achieve this, like parallel code with TTask, visualization of workflow, adding spotlight search, companion apps and other quick-wins and tips.

17:00 Q&A

The ticket price is € 69,95 incl. VAT, coffee/tea/refreshments and lunch.
Tickets are non-refundable but may be used by a colleage if you can't attend.

TICKETS:
https://www.eventbrite.nl/e/delphi-day-2019-tickets-58441488064#tickets

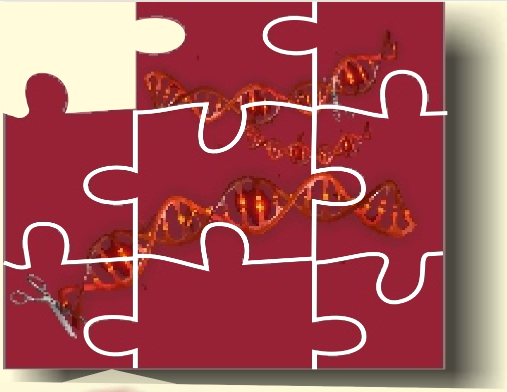
barnsten

- development tools
- consultancy
- components
- training
- hands-on workshops
- support

https://www.eventbrite.nl/e/delphi-day-2019-tickets-58441488064

CRISPER CAS - WHAT IS IT? HOW TO CREATE A (JIGSAW) PUXZZLE

BY PETER WOKKE AND DETLEF OVERBEEK



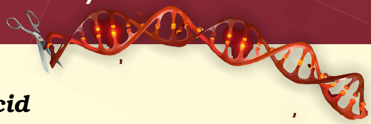
You often hear **CRISPR** pronounce as if it were a technique, just like the steam engine is an invention of man. In fact, it might be better to speak of a discovery.

CRISPR stands for: **CLUSTERED REGULARLY INTERSPACED PALINDROMIC REPEATS.**

You may immediately forget that term, the point is that in the genome of many microorganisms there are pieces of DNA that repeat themselves over and over, but whose function initially remained mysterious to scientists.
Those pieces of CRISPR were first found in bacteria in the 1980s. Later it appears to be a defense mechanism against viruses.
No superfluous luxury for bacteria: viruses kill billions and billions of bacteria every day.

In a far evolutionary past, bacteria developed a way to protect themselves against this: by storing bits of virus DNA in their own DNA. These CRISPRs are the memory of the immune system. This way they can recognize their attackers in the future - they always have a genetic "photo" of the enemy with them. This gives the bacteria a powerful weapon.

They can send assassins out with a picture of the enemy to keep up the metaphor.
In practice, these are proteins that, armed with a piece of **RNA,*1** cut all the **DNA*2** in their neighborhood that looks like the piece they have with them.
John van der Oost and his colleagues from **Wageningen University & Research** demonstrated this for the first time 10 years ago. It also appears possible to adjust the **CRISPR** in such a way that you can recognize and cut every DNA fragment very specifically.



***1**
Deoxyribonucleic acid
DNA is a molecule composed of two chains that coil around each other to form a double helix carrying the genetic instructions used in the growth, development, functioning, and reproduction of all known organisms and many viruses.
DNA and ribonucleic acid (RNA) are nucleic acids; alongside proteins, lipids and complex carbohydrates (polysaccharides), nucleic acids are one of the four major types of macromolecules that are essential for all known forms of life.

***2**
Ribonucleic acid (RNA) *is a polymeric molecule essential in various biological roles in coding, decoding, regulation and expression of genes.*
RNA and DNA are nucleic acids, and, along with lipids, proteins and carbohydrates, constitute the four major macromolecules essential for all known forms of life.
Like DNA, RNA is assembled as a chain of nucleotides, ***but unlike DNA it is more often found in nature as a single-strand folded onto itself, rather than a paired double-strand.***
Cellular organisms use messenger RNA (mRNA) to convey genetic information (using the nitrogenous bases of guanine, uracil, adenine, and cytosine, denoted by the letters G, U, A, and C) that directs synthesis of specific proteins.
Many viruses encode their genetic information using an RNA genome.

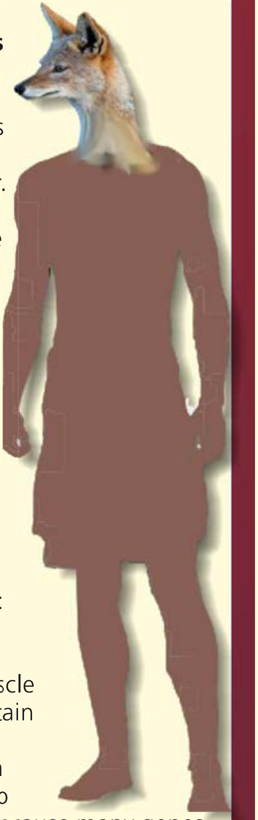
Some RNA molecules play an active role within cells by catalyzing biological reactions, controlling gene expression, or sensing and communicating responses to cellular signals. One of these active processes is protein synthesis, a universal function in which RNA molecules direct the assembly of proteins on ribosomes. This process uses transfer RNA (tRNA) molecules to deliver amino acids to the ribosome, where ribosomal RNA (rRNA) then links amino acids together to form proteins

This pioneering work in Wageningen helps scientists to use the bacterial defense system to make very precise changes to any DNA of their choice.
Not only in bacteria, but also in plants and animals - and therefore also in humans. You do the latter by sending the Cas proteins out with material that can recognize a gene in which a hereditary defect occurs. That protein cuts open your genome in the vicinity of the hereditary defect.
The best known of those molecular scissors is Cas9. That is why you often hear about CRISPR-Cas9. In this article we are talking about CRISPR-Cas from now on.
That is one half of the application. Because if the unwanted gene is cut open, then of course you want to repair that fracture - with a piece of DNA in which the error has been corrected. This is where the repair forces of your DNA come in handy. If you bring the right gene to the area that has been cut open, the repair forces do the rest of the work: .

they put it exactly in the right place.
This method therefore makes it possible to cut and paste into the DNA very precisely and efficiently. Jennifer Doudna is one of the scientists who first used this technique to adjust DNA in human cells

WHAT CAN YOU DO WITH CRISPR?

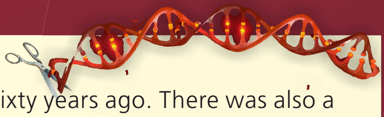
CRISPR gives us the opportunity to adjust the evolution of all life forms on earth.
From fighting hereditary diseases and malaria, to creating completely new animals.
Because we can look up which genes are defective in someone, we are able to repair them better and better.
According to reproductive biologist Sjoerd Repping, we can even remove hereditary diseases from human embryos in a short time.
He talks about the possibilities and the risks that still exist today.
For example, there is still the chance that you accidentally destroy another gene. Nevertheless, he expects that science will make great strides in the coming years.
In addition to repairing genes, you can also replace them with "better" genes. With this you could also adapt embryos to our own wish: the designer baby.
For example, do you want your child to have stronger bones or better muscle growth? Or that your child has a certain eye color? In theory it is possible.
At least for properties that requires a single gene. For example, it is still too complicated to adjust intelligence, because many genes play a role in this.



WHAT ARE THE OBJECTIONS TO ADAPTING DNA?

Not everyone is in favor of tinkering with DNA.
A common argument is that you "play for God" and that this should not be possible.
According to opponents, the biggest problem is that you cannot oversee the consequences.
You change real genes, but you have no idea if this has consequences.
For example, does CRISPR-Cas evoke immune responses in humans? We do not know. In most countries it is therefore (largely) prohibited to tinker with human DNA, with the exception of, for example, China.

Yet the law is being expanded in more and more countries. For example, since 2016 it has been legal in England to process embryos for scientific research.
It is not yet permitted to actually grow these processed embryos into babies, but the expansion of the law does pave the way for them.
In the Netherlands too, many parties, such as the Health Council, are now working hard to legalize genetic processing of embryos here too.



Something similar happened at IVF over sixty years ago. There was also a lot of criticism, because science would play for God.

However, the technology has since become widely accepted. It is therefore not inconceivable that adapting DNA will one day become legal and that we will tinker with the blueprint of human life
.

CRISPR SQUARED: WHAT ARE GENE DRIVES?

The genetic tool CRISPR-Cas makes it possible to tinker not only with individuals but with entire populations.
This is possible with the help of the so-called gene drives. They go a step further than normal genetic modification. Where the chance that an applied mutation is passed on is relatively small per generation, that with gene drives becomes many times greater.

With **CRISPR-Cas** it is even possible to bring extinct animals back to life. At least, to recreate them.
Scientists, under the leadership of geneticist **George Church**, are now focusing their attention on **bringing the mammoth to life**.
CRISPR-Cas makes it possible to adapt the cells of Asian elephants by adding genes from excavated mammoth carcasses from the arctic permafrost.

The purpose of this is not to make perfect copies of the gigantic furry creatures, but to give the elephants the characteristics of a mammoth that they are ultimately able to live in the icy climate of the tundra.

Church states in an interview with the **Canadian medium CBC** that we can bring this region back to its original state of more than twelve thousand years ago.

As a result, the current marshy permafrost could turn into a dry, cold steppe full of grass.
This climate change would reduce the number of dark plants that absorb heat, which, according to Church, will lead to a slowdown in the melting of the Arctic.

Perhaps in the future, technology can also serve its purpose for species that are still alive today.
After all,
one in four mammals,
one in eight bird species and
one in three amphibians are threatened with extinction.

You cannot use this technique to bring every extinct species to life.
To begin with, it is crucial that there is enough DNA left from an extinct species. In addition, there must be a closely related species into which the DNA can be introduced and this species must be able to reproduce.

Finally, a species may not have died out more than 800,000 years ago, because otherwise no DNA would remain. **Bringing back dinosaurs, for example, is therefore not possible.**

CRISPR-Cas is a technique that allows you to cut genes very precisely and replace them with others. A kind of copy-paste option for the DNA.

- **The technology is still in its infancy**, but seems to give people the opportunity to rewrite the building plan of all life forms on earth. You can not only treat hereditary diseases with it, but also adjust properties with it.

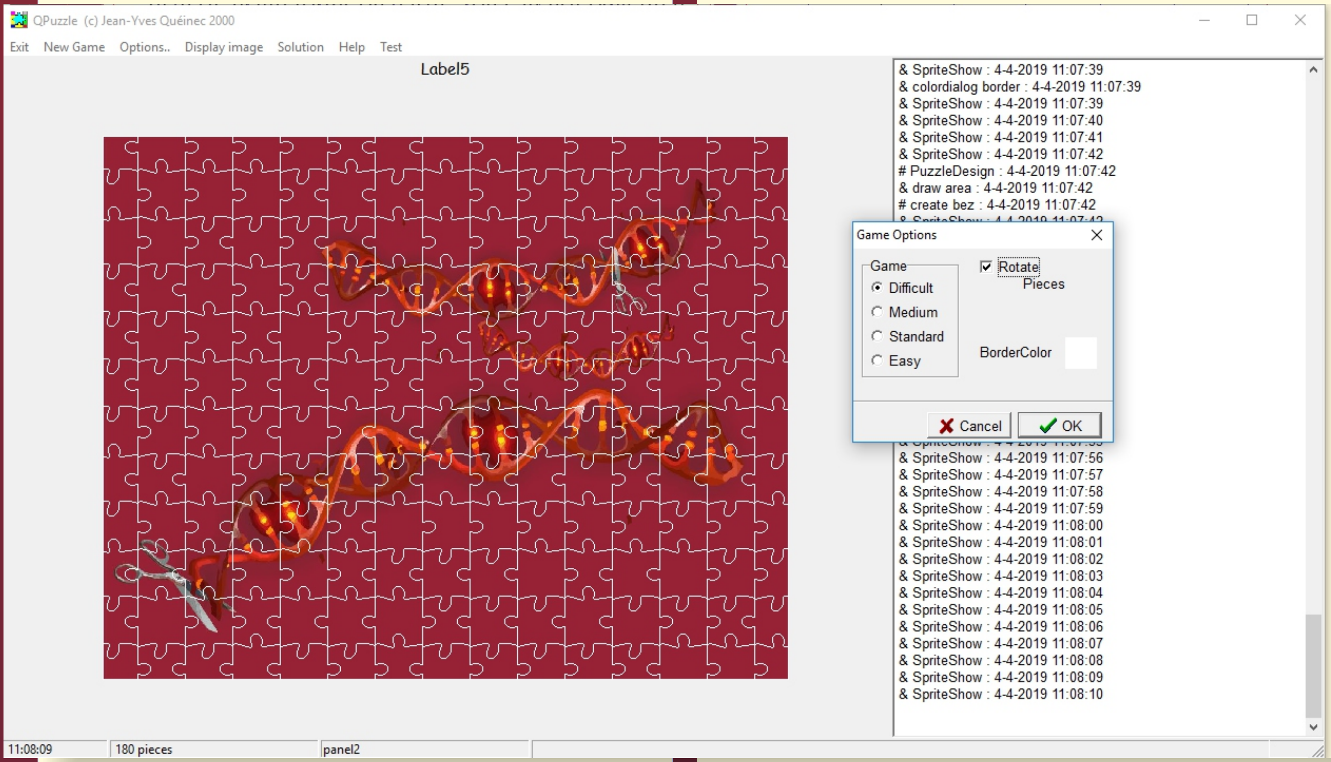
- **There is also resistance to CRISPR-Cas**.
  The impact can be huge, and the consequences are hard to oversee. Do we have to want to play for God?

- **The so-called gene drives are a sort of CRISPR-Cas squared.**
  With it you can not only change the characteristics of an individual, but even of an entire population.

- **Another scenario is to bring extinct animal species back to life.** For example with mammoth DNA found in Russian permafrost.

So because of this
PUZZLE in two ways (litterally as wel by cutting genes reminded me of something that I love personally and which is for programming very intersting.
#Not creating a database where you can drop in the changes as a Criper Cas-er, But something that has a relation with this puzzling theme: the JigSaw.
After searching the web and old code etc, I could not find anything that that offered not just a program but also the code. So I found finally something on a website that seemed to do what I want: Providing free code as wel as creating a JigSaw Puzzles in an interesting way, so that we could make the code available for everyone as well rebuild it to the modern standards.
It first was publshed by: **Jean-YvesQueinec.**

I asked **Peter Wokke** if he could have a look after the construction of the code and tidy it. It took some weeks but het finally succeeded in the first demo that does the real thing: it works under Rio-the latest version of Delphi today. I haven't yet tested it under older versions.
we hereby show some pictures of the working version.
You can add your own picture as long as it is Jpg.

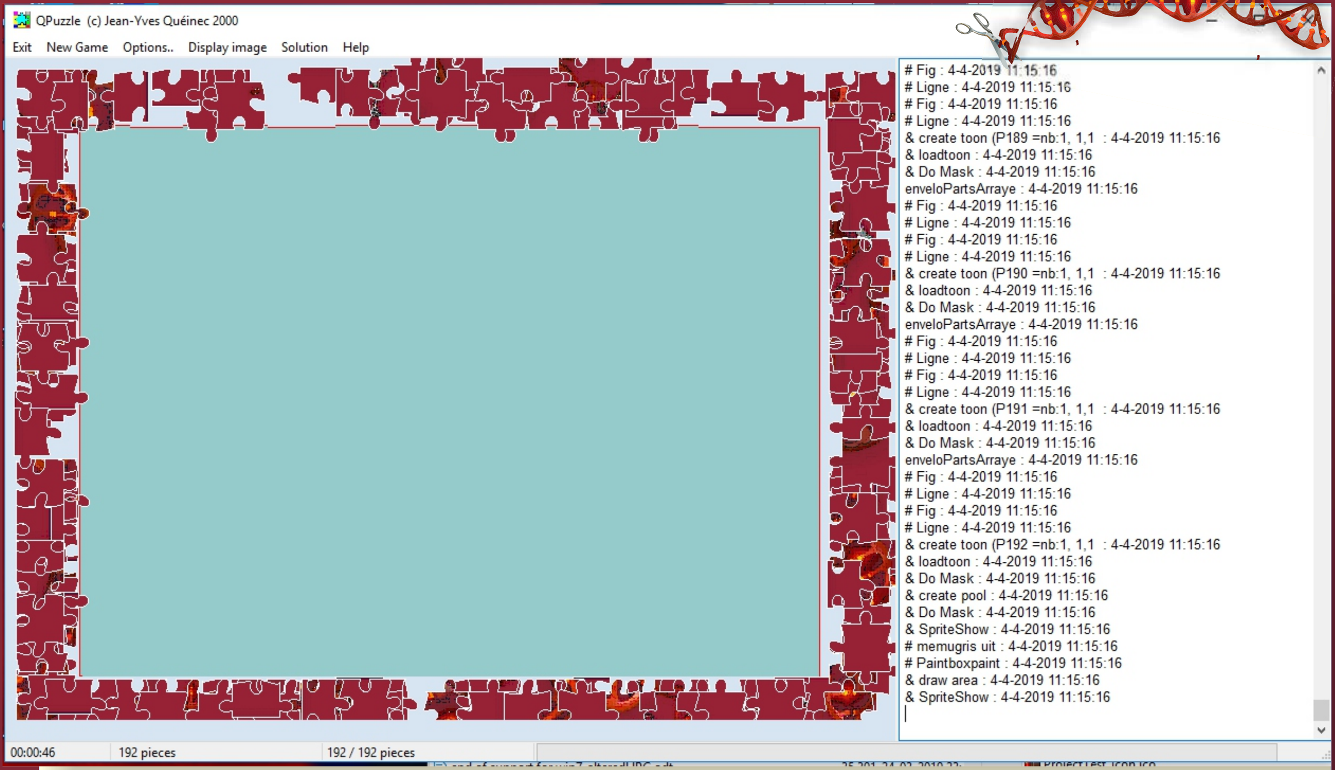


Game
● Difficult
○ Medium
○ Standard
○ Easy
☑ Rotate Pieces
BorderColor □

```
& SpriteShow : 4-4-2019 11:07:39
& colordialog border : 4-4-2019 11:07:39
& SpriteShow : 4-4-2019 11:07:39
& SpriteShow : 4-4-2019 11:07:40
& SpriteShow : 4-4-2019 11:07:41
& SpriteShow : 4-4-2019 11:07:42
# PuzzleDesign : 4-4-2019 11:07:42
& draw area : 4-4-2019 11:07:42
# create bez : 4-4-2019 11:07:42
& SpriteShow : 4-4-2019 11:07:42
& SpriteShow : 4-4-2019 11:07:55
& SpriteShow : 4-4-2019 11:07:56
& SpriteShow : 4-4-2019 11:07:57
& SpriteShow : 4-4-2019 11:07:58
& SpriteShow : 4-4-2019 11:07:59
& SpriteShow : 4-4-2019 11:08:00
& SpriteShow : 4-4-2019 11:08:01
& SpriteShow : 4-4-2019 11:08:02
& SpriteShow : 4-4-2019 11:08:03
& SpriteShow : 4-4-2019 11:08:04
& SpriteShow : 4-4-2019 11:08:05
& SpriteShow : 4-4-2019 11:08:06
& SpriteShow : 4-4-2019 11:08:07
& SpriteShow : 4-4-2019 11:08:08
& SpriteShow : 4-4-2019 11:08:09
& SpriteShow : 4-4-2019 11:08:10
```

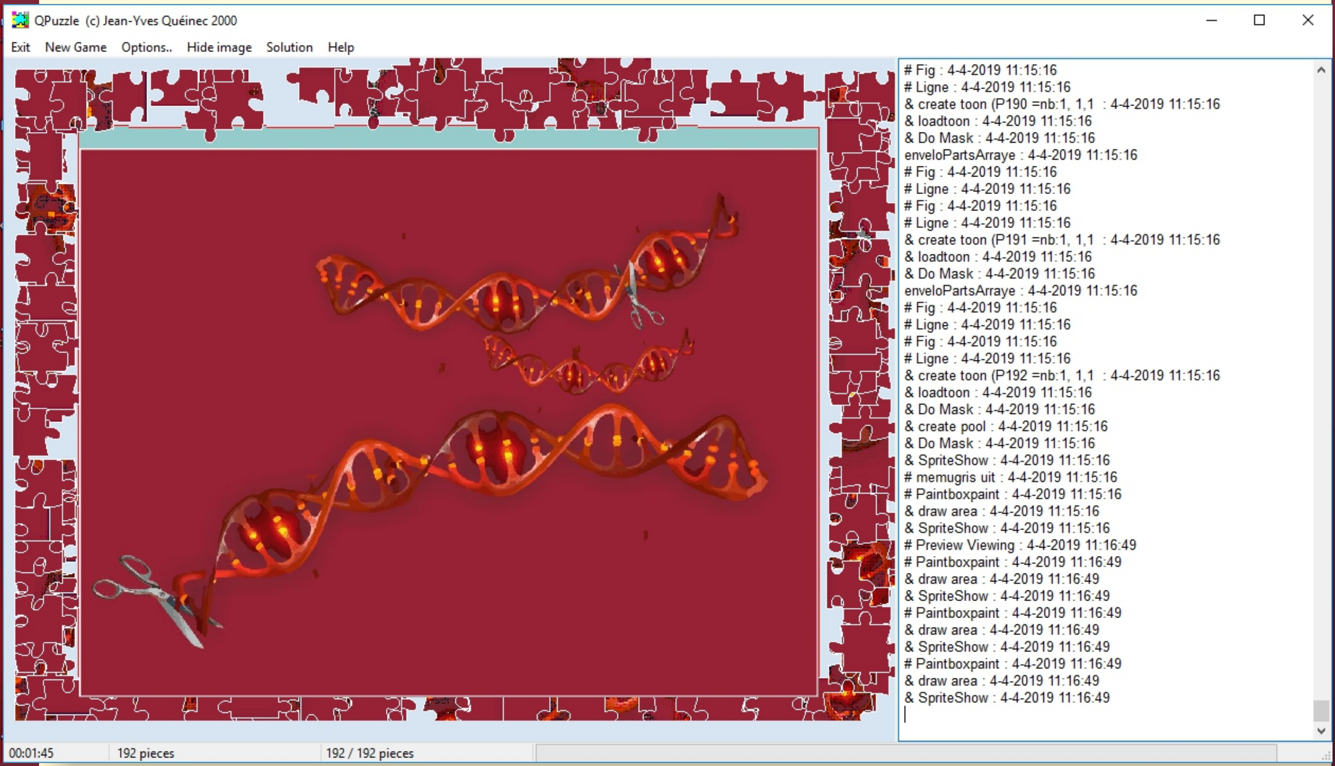
11:08:09 | 180 pieces | panel2

Figure 1: Overview of the Jigsaw-app
You can simply choose your level of difficulty which means more or less puzzle pieces.
You also can level the attraction-factor.



```
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
& create toon (P189 =nb:1, 1,1  : 4-4-2019 11:15:16
& loadtoon : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
enveloPartsArraye : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
& create toon (P190 =nb:1, 1,1  : 4-4-2019 11:15:16
& loadtoon : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
enveloPartsArraye : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
& create toon (P191 =nb:1, 1,1  : 4-4-2019 11:15:16
& loadtoon : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
enveloPartsArraye : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
& create toon (P192 =nb:1, 1,1  : 4-4-2019 11:15:16
& loadtoon : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
& create pool : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
& SpriteShow : 4-4-2019 11:15:16
# memugris uit : 4-4-2019 11:15:16
# Paintboxpaint : 4-4-2019 11:15:16
& draw area : 4-4-2019 11:15:16
& SpriteShow : 4-4-2019 11:15:16
```

Figure 2: Once you start the program you have a desktop with lots of pieces.



```
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
& create toon (P190 =nb:1, 1,1  : 4-4-2019 11:15:16
& loadtoon : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
enveloPartsArraye : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
& create toon (P191 =nb:1, 1,1  : 4-4-2019 11:15:16
& loadtoon : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
enveloPartsArraye : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
# Fig : 4-4-2019 11:15:16
# Ligne : 4-4-2019 11:15:16
& create toon (P192 =nb:1, 1,1  : 4-4-2019 11:15:16
& loadtoon : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
& create pool : 4-4-2019 11:15:16
& Do Mask : 4-4-2019 11:15:16
& SpriteShow : 4-4-2019 11:15:16
# memugris uit : 4-4-2019 11:15:16
# Paintboxpaint : 4-4-2019 11:15:16
& draw area : 4-4-2019 11:15:16
& SpriteShow : 4-4-2019 11:15:16
# Preview Viewing : 4-4-2019 11:16:49
# Paintboxpaint : 4-4-2019 11:16:49
& draw area : 4-4-2019 11:16:49
& SpriteShow : 4-4-2019 11:16:49
# Paintboxpaint : 4-4-2019 11:16:49
& draw area : 4-4-2019 11:16:49
& SpriteShow : 4-4-2019 11:16:49
# Paintboxpaint : 4-4-2019 11:16:49
& draw area : 4-4-2019 11:16:49
& SpriteShow : 4-4-2019 11:16:49
```

Figure 3: You can also now and then have a look at the original.

REST EASY WITH KBMMW #24 – XML, JSON OR YAML TO OBJECT CONVERSION

BY KIM BO MADSEN

starter expert

How often have you received an XML or JSON or for that matter a YAML file that you need to interpret?

I know I personally have received many from customers or customer partners who would like a new customer system to integrate nicely with them, transferring XML or JSON files.

Sometimes you may get an XSD (an XML Schema Definition file), which explains how the XML document is formatted, which is great, because then you can use kbmMW's XSD importer to interpret the XSD and generate Delphi class definitions making it easy to read and/or write the XML file according to the format given in the XSD. In fact, having the XSD, makes it easy to also read and write the data from and to JSON and other kbmMW supported object notation formats.

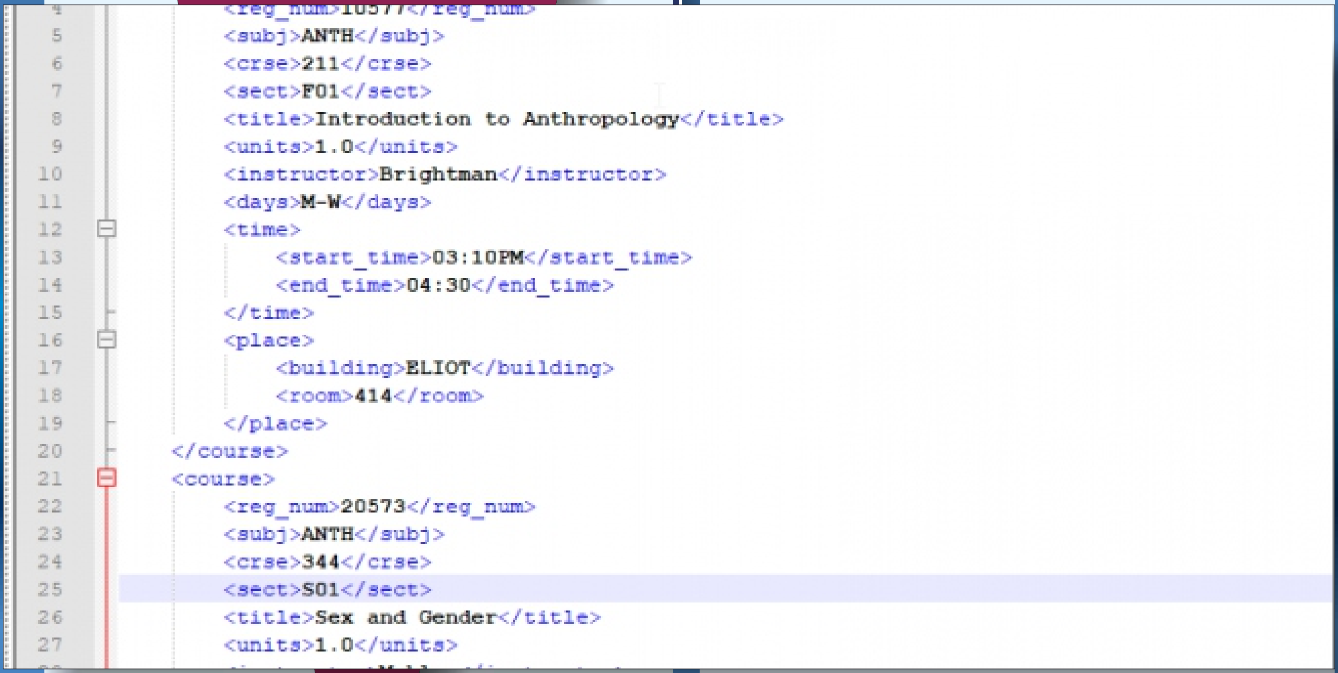
However I can't count the times I have received an example of the actual data file, rather than the XSD definition.

To interpret the file in such case, there are two options

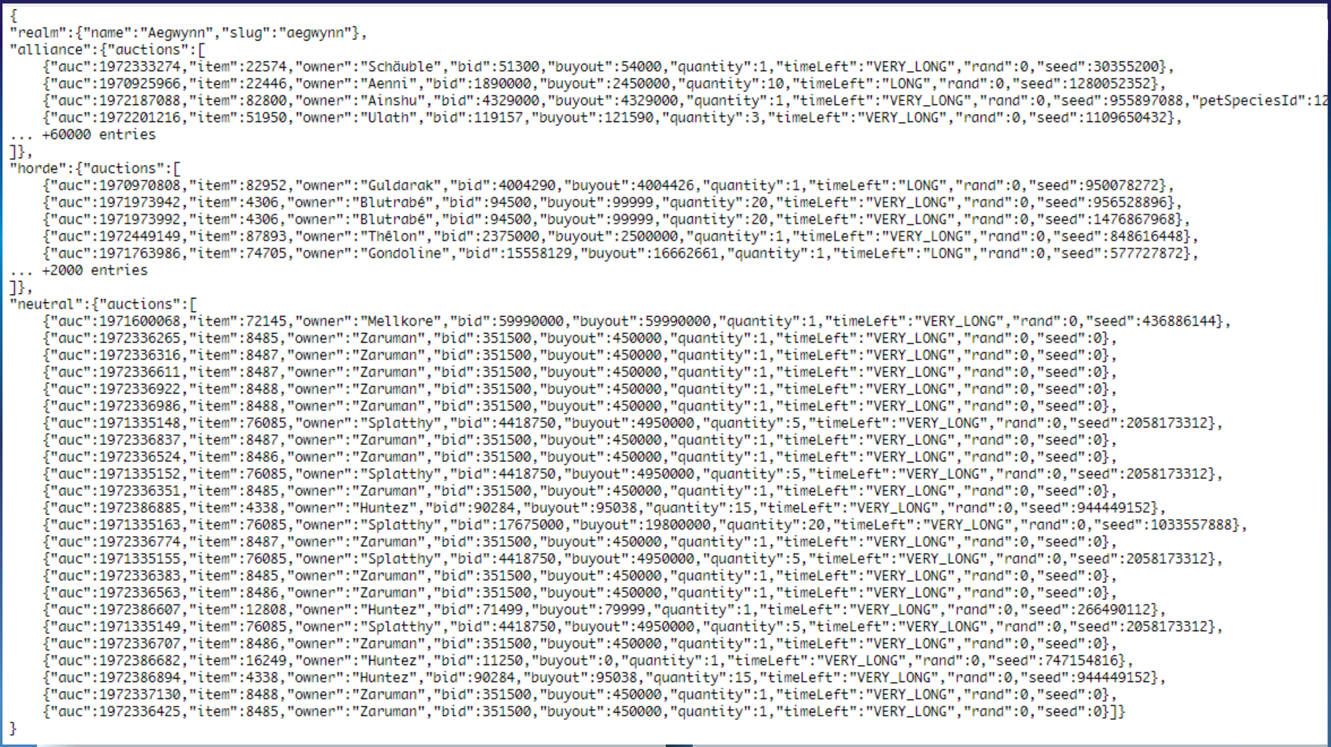
to use a generic XML or JSON or YAML reader to read the file, and manually extract required data from it
or to handcode Delphi class definitions with relevant kbmMW attributes, making it easy for kbmMW to both parse and generate such files
However the first option is not a very generic way to do things, and it does not help with writing a compatible file. The second option can be time consuming, because you need to analyse the file to determine its structure, to be able to generate the relevant object classes and lists, not to mention augmenting the classes and lists with relevant attributes.

Fortunately kbmMW comes to the rescue…again
That's the whole point of programming… to make life easier… so when I notice stuff that could make my life easier, kbmMW receives a new feature, hopefully making the (programming) life of kbmMW users easier too.

So next release of kbmMW contains built in support for parsing and analysing data files (XML, JSON and YAML for the time being), and generates relevant attribute augmented Delphi classes, making it easy to use kbmMW's marshalling framework to read and write such files.



```
        <reg_num>10577</reg_num>
        <subj>ANTH</subj>
        <crse>211</crse>
        <sect>F01</sect>
        <title>Introduction to Anthropology</title>
        <units>1.0</units>
        <instructor>Brightman</instructor>
        <days>M-W</days>
        <time>
            <start_time>03:10PM</start_time>
            <end_time>04:30</end_time>
        </time>
        <place>
            <building>ELIOT</building>
            <room>414</room>
        </place>
    </course>
    <course>
        <reg_num>20573</reg_num>
        <subj>ANTH</subj>
        <crse>344</crse>
        <sect>S01</sect>
        <title>Sex and Gender</title>
        <units>1.0</units>
```



```
{
"realm":{"name":"Aegwynn","slug":"aegwynn"},
"alliance":{"auctions":[
    {"auc":1972333274,"item":22574,"owner":"Schäuble","bid":51300,"buyout":54000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":30355200},
    {"auc":1970925966,"item":22446,"owner":"Aenni","bid":1890000,"buyout":2450000,"quantity":10,"timeLeft":"LONG","rand":0,"seed":1280052352},
    {"auc":1972187088,"item":82800,"owner":"Ainshu","bid":4329000,"buyout":4329000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":955897088,"petSpeciesId":12
    {"auc":1972201216,"item":51950,"owner":"Ulath","bid":119157,"buyout":121590,"quantity":3,"timeLeft":"VERY_LONG","rand":0,"seed":1109650432},
... +60000 entries
]},
"horde":{"auctions":[
    {"auc":1970970808,"item":82952,"owner":"Guldarak","bid":4004290,"buyout":4004426,"quantity":1,"timeLeft":"LONG","rand":0,"seed":950078272},
    {"auc":1971973942,"item":4306,"owner":"Blutrabé","bid":94500,"buyout":99999,"quantity":20,"timeLeft":"VERY_LONG","rand":0,"seed":956528896},
    {"auc":1971973992,"item":4306,"owner":"Blutrabé","bid":94500,"buyout":99999,"quantity":20,"timeLeft":"VERY_LONG","rand":0,"seed":1476867968},
    {"auc":1972449149,"item":87893,"owner":"Thêlon","bid":2375000,"buyout":2500000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":848616448},
    {"auc":1971763986,"item":74705,"owner":"Gondoline","bid":15558129,"buyout":16662661,"quantity":1,"timeLeft":"LONG","rand":0,"seed":577727872},
... +2000 entries
]},
"neutral":{"auctions":[
    {"auc":1971600068,"item":72145,"owner":"Mellkore","bid":59990000,"buyout":59990000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":436886144},
    {"auc":1972336265,"item":8485,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972336316,"item":8487,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972336611,"item":8487,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972336922,"item":8488,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972336986,"item":8488,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1971335148,"item":76085,"owner":"Splatthy","bid":4418750,"buyout":4950000,"quantity":5,"timeLeft":"VERY_LONG","rand":0,"seed":2058173312},
    {"auc":1972336837,"item":8487,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972336524,"item":8486,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1971335152,"item":76085,"owner":"Splatthy","bid":4418750,"buyout":4950000,"quantity":5,"timeLeft":"VERY_LONG","rand":0,"seed":2058173312},
    {"auc":1972336351,"item":8485,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972386885,"item":4338,"owner":"Huntez","bid":90284,"buyout":95038,"quantity":15,"timeLeft":"VERY_LONG","rand":0,"seed":944449152},
    {"auc":1971335163,"item":76085,"owner":"Splatthy","bid":17675000,"buyout":19800000,"quantity":20,"timeLeft":"VERY_LONG","rand":0,"seed":1033557888},
    {"auc":1972336774,"item":8487,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1971335155,"item":76085,"owner":"Splatthy","bid":4418750,"buyout":4950000,"quantity":5,"timeLeft":"VERY_LONG","rand":0,"seed":2058173312},
    {"auc":1972336383,"item":8485,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972336563,"item":8486,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972386607,"item":12808,"owner":"Huntez","bid":71499,"buyout":79999,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":266490112},
    {"auc":1971335149,"item":76085,"owner":"Splatthy","bid":4418750,"buyout":4950000,"quantity":5,"timeLeft":"VERY_LONG","rand":0,"seed":2058173312},
    {"auc":1972336707,"item":8486,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972386682,"item":16249,"owner":"Huntez","bid":11250,"buyout":0,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":747154816},
    {"auc":1972386894,"item":4338,"owner":"Huntez","bid":90284,"buyout":95038,"quantity":15,"timeLeft":"VERY_LONG","rand":0,"seed":944449152},
    {"auc":1972337130,"item":8488,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0},
    {"auc":1972336425,"item":8485,"owner":"Zaruman","bid":351500,"buyout":450000,"quantity":1,"timeLeft":"VERY_LONG","rand":0,"seed":0}]}
}
```

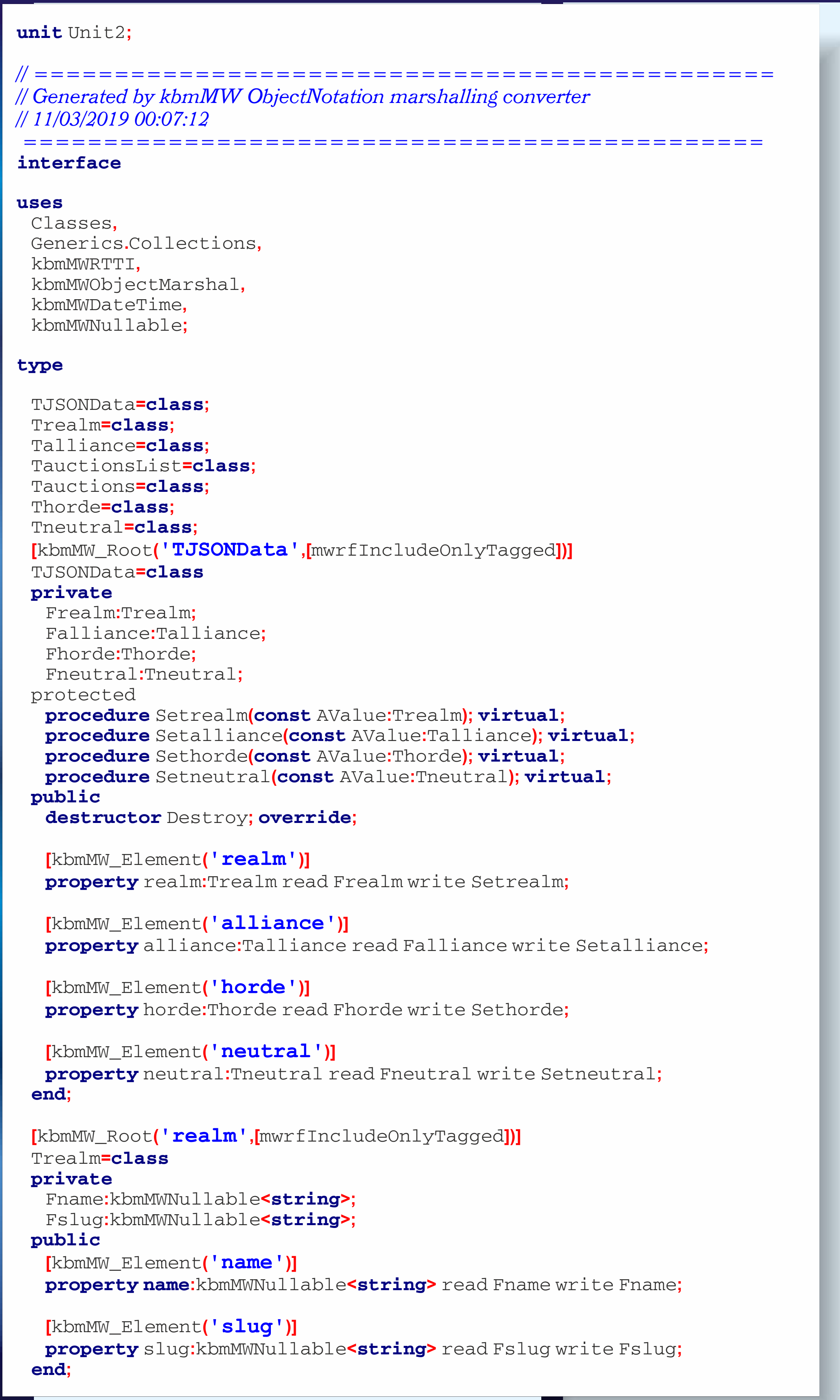
As you can see, it's fairly big, and relatively complex with multiple arrays of objects.

We make kbmMW analyse it and output the relevant Delphi based source code, that makes it easy to marshal/unmarshal such data:

```
var
 s:string;
...
 s:=TkbmMWJSONMarshal.GenerateDelphiClassFromUTF8File('auctions.json','Unit2','JSONData');
...
```

What happens is that we ask the JSON marshalling class to generate the source code for the classes needed for us, matching the contents of auctions.json, producing a unit called Unit2 and a main class called TJSONData. The reason for the main class, is that the main object in the json file is unnamed or anonymous, hence we need to provide some name for the generator to use in the Delphi code.

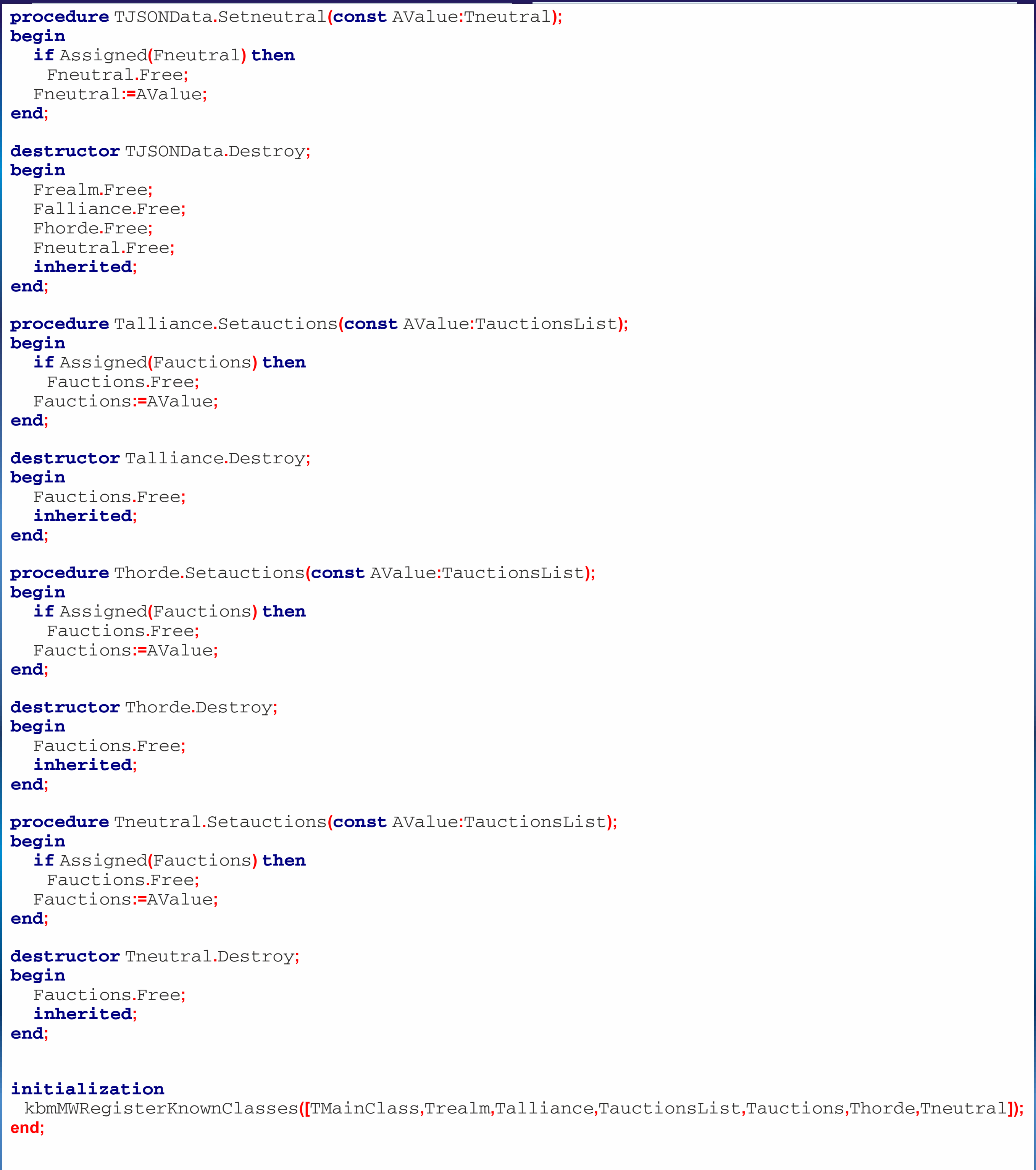
Simply write the contents of **s** to a file called Unit2.pas and add it to the projects where you need to read or write auctions.json files.



```
unit Unit2;

// =============================================
// Generated by kbmMW ObjectNotation marshalling converter
// 11/03/2019 00:07:12
 =============================================
interface

uses
 Classes,
 Generics.Collections,
 kbmMWRTTI,
 kbmMWObjectMarshal,
 kbmMWDateTime,
 kbmMWNullable;

type

 TJSONData=class;
 Trealm=class;
 Talliance=class;
 TauctionsList=class;
 Tauctions=class;
 Thorde=class;
 Tneutral=class;
 [kbmMW_Root('TJSONData',[mwrfIncludeOnlyTagged])]
 TJSONData=class
 private
  Frealm:Trealm;
  Falliance:Talliance;
  Fhorde:Thorde;
  Fneutral:Tneutral;
 protected
  procedure Setrealm(const AValue:Trealm); virtual;
  procedure Setalliance(const AValue:Talliance); virtual;
  procedure Sethorde(const AValue:Thorde); virtual;
  procedure Setneutral(const AValue:Tneutral); virtual;
 public
  destructor Destroy; override;

  [kbmMW_Element('realm')]
  property realm:Trealm read Frealm write Setrealm;

  [kbmMW_Element('alliance')]
  property alliance:Talliance read Falliance write Setalliance;

  [kbmMW_Element('horde')]
  property horde:Thorde read Fhorde write Sethorde;

  [kbmMW_Element('neutral')]
  property neutral:Tneutral read Fneutral write Setneutral;
 end;

 [kbmMW_Root('realm',[mwrfIncludeOnlyTagged])]
 Trealm=class
 private
  Fname:kbmMWNullable<string>;
  Fslug:kbmMWNullable<string>;
 public
  [kbmMW_Element('name')]
  property name:kbmMWNullable<string> read Fname write Fname;

  [kbmMW_Element('slug')]
  property slug:kbmMWNullable<string> read Fslug write Fslug;
 end;
```

```
[kbmMW_Root('alliance',[mwrfIncludeOnlyTagged])]
Talliance=class
private
  Fauctions:TauctionsList;
protected
  procedure Setauctions(const AValue:TauctionsList); virtual;
public
  destructor Destroy; override;

  [kbmMW_Element('auctions')]
  property auctions:TauctionsList read Fauctions write Setauctions;
end;

[kbmMW_Child('auctions',[mwcfFlatten])]
TauctionsList=class(TObjectList<Tauctions>);
[kbmMW_Root('auctions',[mwrfIncludeOnlyTagged])]
Tauctions=class
private
  FpetSpeciesId:kbmMWNullable<double>;
  FpetBreedId:kbmMWNullable<double>;
  FpetLevel:kbmMWNullable<double>;
  FpetQualityId:kbmMWNullable<double>;
  Fauc:kbmMWNullable<double>;
  Fitem:kbmMWNullable<double>;
  Fowner:kbmMWNullable<string>;
  Fbid:kbmMWNullable<double>;
  Fbuyout:kbmMWNullable<double>;
  Fquantity:kbmMWNullable<double>;
  FtimeLeft:kbmMWNullable<string>;
  Frand:kbmMWNullable<double>;
  Fseed:kbmMWNullable<double>;
public
  [kbmMW_Element('petSpeciesId')]
  property petSpeciesId:kbmMWNullable<double> read FpetSpeciesId write FpetSpeciesId;

  [kbmMW_Element('petBreedId')]
  property petBreedId:kbmMWNullable<double> read FpetBreedId write FpetBreedId;

  [kbmMW_Element('petLevel')]
  property petLevel:kbmMWNullable<double> read FpetLevel write FpetLevel;

  [kbmMW_Element('petQualityId')]
  property petQualityId:kbmMWNullable<double> read FpetQualityId write FpetQualityId;

  [kbmMW_Element('auc')]
  property auc:kbmMWNullable<double> read Fauc write Fauc;

  [kbmMW_Element('item')]
  property item:kbmMWNullable<double> read Fitem write Fitem;

  [kbmMW_Element('owner')]
  property owner:kbmMWNullable<string> read Fowner write Fowner;

  [kbmMW_Element('bid')]
  property bid:kbmMWNullable<double> read Fbid write Fbid;

  [kbmMW_Element('buyout')]
  property buyout:kbmMWNullable<double> read Fbuyout write Fbuyout;

  [kbmMW_Element('quantity')]
  property quantity:kbmMWNullable<double> read Fquantity write Fquantity;

  [kbmMW_Element('timeLeft')]
  property timeLeft:kbmMWNullable<string> read FtimeLeft write FtimeLeft;
```

```
    [kbmMW_Element('rand')]
    property rand:kbmMWNullable<double> read Frand write Frand;

    [kbmMW_Element('seed')]
    property seed:kbmMWNullable<double> read Fseed write Fseed;
  end;

  [kbmMW_Root('horde',[mwrfIncludeOnlyTagged])]
  Thorde=class
  private
    Fauctions:TauctionsList;
  protected
    procedure Setauctions(const AValue:TauctionsList); virtual;
  public
    destructor Destroy; override;

    [kbmMW_Element('auctions')]
    property auctions:TauctionsList read Fauctions write Setauctions;
  end;

  [kbmMW_Root('neutral',[mwrfIncludeOnlyTagged])]
  Tneutral=class
  private
    Fauctions:TauctionsList;
  protected
    procedure Setauctions(const AValue:TauctionsList); virtual;
  public
    destructor Destroy; override;

    [kbmMW_Element('auctions')]
    property auctions:TauctionsList read Fauctions write Setauctions;
  end;


implementation

procedure TJSONData.Setrealm(const AValue:Trealm);
begin
   if Assigned(Frealm) then
     Frealm.Free;
   Frealm:=AValue;
end;

procedure TJSONData.Setalliance(const AValue:Talliance);
begin
   if Assigned(Falliance) then
     Falliance.Free;
   Falliance:=AValue;
end;

procedure TJSONData.Sethorde(const AValue:Thorde);
begin
   if Assigned(Fhorde) then
     Fhorde.Free;
   Fhorde:=AValue;
end;

procedure TJSONData.Setneutral(const AValue:Tneutral);
begin
   if Assigned(Fneutral) then
     Fneutral.Free;
   Fneutral:=AValue;
end;
```



```
procedure TJSONData.Setneutral(const AValue:Tneutral);
begin
  if Assigned(Fneutral) then
   Fneutral.Free;
  Fneutral:=AValue;
end;

destructor TJSONData.Destroy;
begin
  Frealm.Free;
  Falliance.Free;
  Fhorde.Free;
  Fneutral.Free;
  inherited;
end;

procedure Talliance.Setauctions(const AValue:TauctionsList);
begin
  if Assigned(Fauctions) then
   Fauctions.Free;
  Fauctions:=AValue;
end;

destructor Talliance.Destroy;
begin
  Fauctions.Free;
  inherited;
end;

procedure Thorde.Setauctions(const AValue:TauctionsList);
begin
  if Assigned(Fauctions) then
   Fauctions.Free;
  Fauctions:=AValue;
end;

destructor Thorde.Destroy;
begin
  Fauctions.Free;
  inherited;
end;

procedure Tneutral.Setauctions(const AValue:TauctionsList);
begin
  if Assigned(Fauctions) then
   Fauctions.Free;
  Fauctions:=AValue;
end;

destructor Tneutral.Destroy;
begin
  Fauctions.Free;
  inherited;
end;


initialization
 kbmMWRegisterKnownClasses([TMainClass,Trealm,Talliance,TauctionsList,Tauctions,Thorde,Tneutral]);
end;
```

To actually read the file, create a program, add Unit2.pas to it, and write this code:

```
Procedure TForm1.Button1Click(Sender: TObject);
var
  m:TkbmMWCustomRTTIMarshal;
  o:TJSONData;
begin
   m:=TkbmMWJSONMarshal.Create;
   try
     TkbmMWJSONMarshal(m).AnonymousRoot:=true;
     o:=TkbmMWJSONMarshal(m).ValueFromUTF8File<TJSONData>('auctions.json');
     // Your complete parsed file is now contained in the object instance o.
   finally
     o.Free;
     m.Free;
   end;
end;
```

It is as simple as that.
Another example is the XML file courses.xml, also picked up from the internet somewhere.

```
<?xml version="1.0"?>
<root>
        <course>
                <reg_num>10577</reg_num>
                <subj>ANTH</subj>
                <crse>211</crse>
                <sect>F01</sect>
                <title>Introduction to Anthropology</title>
                <units>1.0</units>
                <instructor>Brightman</instructor>
                <days>M-W</days>
                <time>
                        <start_time>03:10PM</start_time>
                        <end_time>04:30</end_time>
                </time>
                <place>
                        <building>ELIOT</building>
                        <room>414</room>
                </place>
        </course>
        <course>
                <reg_num>20573</reg_num>
                <subj>ANTH</subj>
                <crse>344</crse>
                <sect>S01</sect>
                <title>Sex and Gender</title>
                <units>1.0</units>
                <instructor>Makley</instructor>
                <days>T-Th</days>
                <time>
                        <start_time>10:30AM</start_time>
                        <end_time>11:50</end_time>
                </time>
                <place>
                        <building>VOLLUM</building>
                        <room>120</room>
                </place>
        </course>
        <course>
                <reg_num>10624</reg_num>
                <subj>BIOL</subj>
                <crse>431</crse>
                <sect>F01</sect>
                <title>Field Biology of Amphibians</title>
                <units>0.5</units>
...
        </course>
...
</root>
```

To create Delphi source code to marshal and unmarshal that type of file

```
 var s:string;
...

s:=TkbmMWXMLMarshal.GenerateDelphiClassFromFile('courses.xml','Unit3','XMLData');
...
```

That will analyse courses.xml and produce a string containing Delphi source code for a unit called Unit3. We also, by convention, provide a main class name, but it will not be used since XML do not, like JSON, operate with an anonymous root datastructure.

The generated unit will look like this:

```
Unit Unit3;

// ======================================
// Generated by kbmMW ObjectNotation marshalling converter 10/03/2019 23:54:52
// ======================================

interface

uses
  Classes,
  Generics.Collections,
  kbmMWRTTI,
  kbmMWObjectMarshal,
  kbmMWDateTime,
  kbmMWNullable;

type

  Troot=class;
  TcourseList=class;
  Tcourse=class;
  Ttime=class;
  Tplace=class;
  [kbmMW_Root('root',[mwrfIncludeOnlyTagged])]
  Troot=class
  private
    Fcourse:TcourseList;
  protected
    procedure Setcourse(const AValue:TcourseList); virtual;
  public
    destructor Destroy; override;
   property course:TcourseList read Fcourse write Setcourse;
  end;
  [kbmMW_Child('course',[mwcfFlatten])]
  TcourseList=class(TObjectList<Tcourse>);
  [kbmMW_Root('course',[mwrfIncludeOnlyTagged])]
  Tcourse=class
  private
    Ftime:Ttime;
    Fplace:Tplace;
    Freg_num:kbmMWNullable<string>;
    Fsubj:kbmMWNullable<string>;
    Fcrse:kbmMWNullable<string>;
    Fsect:kbmMWNullable<string>;
    Ftitle:kbmMWNullable<string>;
    Funits:kbmMWNullable<string>;
    Finstructor:kbmMWNullable<string>;
    Fdays:kbmMWNullable<string>;
    Fxml_repository:kbmMWNullable<string>;
```

```
protected
  procedure Settime(const AValue:Ttime); virtual;
  procedure Setplace(const AValue:Tplace); virtual;
public
  destructor Destroy; override;

  [kbmMW_Element('time')]
  property time:Ttime read Ftime write Settime;

  [kbmMW_Element('place')]
  property place:Tplace read Fplace write Setplace;

  [kbmMW_Element('reg_num')]
  property reg_num:kbmMWNullable<string> read Freg_num write Freg_num;

  [kbmMW_Element('subj')]
  property subj:kbmMWNullable<string> read Fsubj write Fsubj;

  [kbmMW_Element('crse')]
  property crse:kbmMWNullable<string> read Fcrse write Fcrse;

  [kbmMW_Element('sect')]
  property sect:kbmMWNullable<string> read Fsect write Fsect;

  [kbmMW_Element('title')]
  property title:kbmMWNullable<string> read Ftitle write Ftitle;

  [kbmMW_Element('units')]
  property units:kbmMWNullable<string> read Funits write Funits;

  [kbmMW_Element('instructor')]
  property instructor:kbmMWNullable<string> read Finstructor write Finstructor;

  [kbmMW_Element('days')]
  property days:kbmMWNullable<string> read Fdays write Fdays;

  [kbmMW_Element('xml_repository')]
  property xml_repository:kbmMWNullable<string> read Fxml_repository write Fxml_repository;
end;

  [kbmMW_Root('time',[mwrfIncludeOnlyTagged])]
  Ttime=class
private
  Fstart_time:kbmMWNullable<string>;
  Fend_time:kbmMWNullable<string>;
private
  Fstart_time:kbmMWNullable<string>;
       Fend_time:kbmMWNullable<string>;
public
  [kbmMW_Element('start_time')]
       property start_time:kbmMWNullable<string> read Fstart_time write Fstart_time;

       [kbmMW_Element('end_time')]
       property end_time:kbmMWNullable<string> read Fend_time write Fend_time;
end;

[kbmMW_Root('place',[mwrfIncludeOnlyTagged])]
   Tplace=class
   private
  Fbuilding:kbmMWNullable<string>;
       Froom:kbmMWNullable<string>;
public
  [kbmMW_Element('building')]
       property building:kbmMWNullable<string> read Fbuilding write Fbuilding;

       [kbmMW_Element('room')]
       property room:kbmMWNullable<string> read Froom write Froom;
end;
```

```
implementation

procedure Troot.Setcourse(const AValue:TcourseList);
begin
  if Assigned(Fcourse) then  Fcourse.Free;
  Fcourse:=AValue;
end;

destructor Troot.Destroy;
begin
  Fcourse.Free;
  inherited;
end;

procedure Tcourse.Settime(const AValue:Ttime);
begin
  if Assigned(Ftime) then Ftime.Free;
  Ftime:=AValue;
end;

procedure Tcourse.Setplace(const AValue:Tplace);
begin
  if Assigned(Fplace) then  Fplace.Free;
  Fplace:=AValue;
end;

destructor Tcourse.Destroy;
begin
  Ftime.Free;
  Fplace.Free;
  inherited;
end;

initialization
 kbmMWRegisterKnownClasses([Troot,TcourseList,Tcourse,Ttime,Tplace]);

end.
```

And to read and write such XML data, this code can be used

```
procedure TForm1.Button2Click(Sender: TObject);
var
 m:TkbmMWCustomRTTIMarshal; o:Troot; s:string;
begin
  m:=TkbmMWXMLMarshal.Create;
  try
    o:=TkbmMWXMLMarshal(m).ValueFromFile<Troot>('courses.xml');
    // o now contains all courses.
    s:=TkbmMWXMLMarshal(m).ValueToString(o);
    // s now contains XML generated from o.
  finally
    o.Free;
    m.Free;
  end;
end;
```

If the kbmMW file analyser detects a type it does not know what to do with, it will auto generate one called **TUNKNOWN**, which will need to be manually handled by the developer, by replacing it with more relevant class definitions to be able to marshal and unmarshal that particular part of the data.

If you like what you see, please share the word. Let others know about the product!

Essentially help us to help you