

BLAISE PASCAL MAGAZINE 107/108

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



Blaise Pascal



Penguin Datenwissenschaft: 4 Engines zum Rendern von Vektorgrafiken

FRESNEL das grafische Projekt für Lazarus, das alles in Farbe setzt

PANTHER der superschnelle, in Lazarus erstellte PDF-Indexer für alle Os's erklärt

Das Pendel, die Zeit wirkt wie eine Welle

Verwendung von Stilen in einem VCL-Programm

Lazarus program mit BufDataset

Cross compiling mit FPC Up De Luxe

Beschleunigung des Delphi Compilers

Neue Funktion für die Benutzerfreundlichkeit in DELPHI

Multi-node quantum network - Quantum Internet News

Was ist öffentlich an dem intelligenten Chatbot-Hersteller OPENAI?

Diskussion über OPEN GPT

Installation der FastReport-Pakete in Lazarus für Linux/Windows

Zeos database Komponente für Lazarus

BLAISE PASCAL MAGAZINE 107/108

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



INHALT

ARTIKEL

- Neues von der Redaktion Seite 4
Von unserem technischen Berater - Cartoon von Jerry King Seite 6
Penguin Datenwissenschaft: 4 Engines zum Rendern von Vektorgrafiken Seite 8
Von Max Kleiner
FRESNEL das grafische Projekt für Lazarus, das alles in Farbe setzt Seite 65
Von Michael van Canneyt
PANTHER der superschnelle, in Lazarus erstellte PDF-Indexer für alle Os's erklärt Seite 40
Von Michael van Canneyt
Das Pendel, die Zeit wirkt wie eine Welle Seite 23
Von David Dirksé
Verwendung von Stilen in einem VCL-Programm Seite 30
Von Don Wilbrink
Lazarus programm mit BufDataset Seite 62
Von o Don Wilbrink
Cross compiling mit FPC Up De Luxe Seite 72
Von Alfred Glänzer
Beschleunigung des Delphi Compilers Seite 55
Von Marco Cantu / Embarcadero
Neue Funktion für die Benutzerfreundlichkeit in DELPHI Seite 35
Embarcadero
Multi-node quantum network - Quantum Internet News Seite 78
Von Detlef Overbeek
Was ist öffentlich an dem intelligenten Chatbot-Hersteller OPENAI? Seite 90
Von Kim Madsen
Diskussion über OPEN GPT Page 95
Von Detlef Overbeek
Installation der FastReport-Pakete in Lazarus für Linux/Windows Seite 82
Von Detlef Overbeek
Zeos database Komponente für Lazarus Seite 86
Von Don Wilbrink



WERBUNG

- Library 2023
PDF Viewer 2023 Blaise Pascal Library USB stick
Lazarus Handbook Pocket
Lazarus Handbook PDF + Abonnement
Database Workbench
Abonnement 2 Jahr
Superpack 5 Items
Barnsten Delphi Products
Lazarus Handbook PDF
HILFE für Ukraine
Components for Developers

- Seite 7
Seite 22
Seite 29/77
Seite 34
Seite 38
Seite 39
Seite 54/89
Seite 61
Seite 81
Seite 99
Seite 100



Niklaus Wirth

Pascal ist eine imperative und prozedurale Programmiersprache, die Niklaus Wirth (links unten) in den Jahren 1968-69 entworfen und 1970 veröffentlicht hat. Es handelt sich dabei um eine kleine, effiziente Sprache, die gute Programmierpraktiken unter Verwendung von strukturierter Programmierung und Datenstrukturierung fördern soll. Ein Derivat, bekannt als Object Pascal, wurde 1985 für die objektorientierte Programmierung entwickelt. Der Name der Sprache wurde zu Ehren des Mathematikers und Erfinders der ersten Rechenmaschine gewählt: Blaise Pascal (siehe oben rechts).

Herausgeber: © Foundation Supporting Programming Language Pascal - mit Sitz in den Niederlanden
Registrierter Name: Stichting Ondersteuning Programmeertaal Pascal IJsselstein, Netherlands
VAT / BTW: NL814254147B01 Chamber of commerce (KVK) 30 202429 Handy: +31 6 21 23 62 68



CONTRIBUTORS

Stephen Ball http://delphiaball.co.uk DelphiABall	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 mail: David @ davdata.nl	Benno Evers b.evers @ everscustomtechnology.nl	Bruno Fierens www.tmssoftware.com bruno.fierens @ tmssoftware.com	Holger Flick holger @ flixments.com
Mattias Gärtnernc- gaertnma@netcologne.de	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.com kbmMW		Boian Mitov mitov @ mitov.com	
	Jeremy North jeremy.north @ gmail.com	Detlef Overbeek - Editor in Chief www.blaisepascal.eu editor @ blaisepascal.eu	
Anton Vogelaar ajv @ vogelaar-electronics.com	Danny Wind dwind @ delphicompany.nl	Jos Wegman Corrector / Analyst	Siegfried Zuhr siegfried @ zuhr.nl

Chefredakteur

Detlef D. Overbeek, Niederlande Tel.: Mobil: +31 (0)6 21.23.62.68
Nachrichten und Pressemitteilungen nur per E-Mail an editor@blaisepascal.eu

Abonnemente können online unter <https://www.blaisepascalmagazine.eu/deutsche-Ausgabe/> oder per schriftlicher Bestellung abgeschlossen werden oder indem Sie eine E-Mail an office@blaisepascal.eu senden. Das Abonnement kann zu einem beliebigen Zeitpunkt beginnen. Alle Ausgaben, die im Kalenderjahr des Abonnements veröffentlicht werden, werden geschickt. Das Abonnement hat eine Laufzeit von 365 Tagen. Abonnemente werden nicht ohne Vorankündigung verlängert
Der Zahlungseingang wird per E-Mail verschickt. Sie können das Abonnement bezahlen, indem Sie die Zahlung an folgende Adresse senden: ABN AMRO Bank Konto Nr. 44 19 60 863 oder per Kreditkarte oder Paypal Name: Stiftung Pro Pascal (Stichting Programmeertaal Pascal) IBAN: NL82 ABNA 0441960863 BIC ABNANL2A Umsatzsteuer-Nr.: 81 42 54 147 (Stichting Programmeertaal Pascal)
Abonnementsabteilung Edelstenenbaan 21 / 3402 XA IJsselstein, Niederlande Mobil: + 31 (0) 6 21.23.62.68 office@blaisepascal.eu

Markenzeichen Alle verwendeten Markenzeichen sind Eigentum der jeweiligen Inhaber. Vorbehalt Obwohl wir uns bemühen sicherzustellen, dass die in der Zeitschrift veröffentlichten Informationen korrekt sind, können wir keine Verantwortung für Fehler oder Auslassungen übernehmen.
Wenn Sie etwas bemerken, das möglicherweise nicht korrekt ist, wenden Sie sich bitte an den Herausgeber, und wir werden gegebenenfalls eine Korrektur veröffentlichen.



Mitglied der **Königlich Niederländischen Bibliothek**

KB

Mitglied und Spender von **WIKIPEDIA**

Subscriptions (2022 prices)	Internat. excl. VAT	Internat. incl. 9% VAT	Shipment	TOTAL
Printed Issue (8 per year) ±60 pages :	€ 200	€ 218	€ 130	€ 348
Electronic Download Issue (8 per year) ±60 pages :	€ 64,20	€ 70		

COPYRIGHT-HINWEIS

Das gesamte in Blaise Pascal veröffentlichte Material unterliegt dem Copyright © SOPP Stichting Ondersteuning Programmeertaal Pascal, sofern nicht anders angegeben, und darf nicht ohne schriftliche Genehmigung kopiert, verbreitet oder neu veröffentlicht werden. Die Autoren erklären sich damit einverstanden, dass der zu ihren Artikeln gehörende Code nach der Veröffentlichung den Abonnenten zur Verfügung gestellt wird, indem er auf der Website der PGG zum Download angeboten wird, und dass Artikel und Code auf verteilbaren Datenträgern gespeichert werden. Die Nutzung von Programmlisten durch Abonnenten zu Forschungs- und Studienzwecken ist erlaubt, jedoch nicht zu kommerziellen Zwecken. Die kommerzielle Nutzung von Programmlistings und Code ist ohne die schriftliche Genehmigung des Autors untersagt.



Neues - Von der Redaktion

Ich wünsche ihnen alle ein frohes neues Jahr.

Zuerst möchte ich mich für die späte Veröffentlichung entschuldigen: Ich habe das Layout-Paket CorelDraw gegen Affinity Publisher getauscht, weil Affinity Publisher die GPU-Engine nutzt und daher für die zukünftigen Prozesse viel besser geeignet ist.

Das bedeutet, dass ich lernen musste, wie man mit diesem Paket umgeht, und das kostet viel zusätzliche Energie und Zeit. Da ich die einzige Person bin, die dieses Magazin erstellt, musste ich viel Zeit aufwenden, um etwas völlig Neues zu lernen.

Zumal ich seit 24 Jahren mit Corel arbeite. Aber ich habe es geschafft.

Man ist nie zu alt, um zu lernen.

Falls Sie jemals ein Designer- oder Publishing-Tool benötigen: Affinity ist sehr günstig und hat drei integrierte Pakete: 1-Publisher 2-Designer und 3-Photo und kosten alles in allem nicht mehr als 120 €.

And now for something completely different:

Das Lazarus-Team hat ein völlig neues Projekt entwickelt: FRESNEL.

Seit jeher habe ich versucht, sie dazu zu bewegen, etwas an der Grafik und dem Design von Lazarus zu ändern.

Dies, weil ich persönlich das sehr mag. Aber das Wichtigste wäre das Aussehen und das Gefühl des Pakets für die junge Generation.

Natürlich wollen wir einen dunklen Modus usw. Und wir wollen "die Jugend" anziehen, denn der Jugend gehört die Zukunft, also sind sie uns sehr wichtig.

Da Lazarus in der Lage ist, mehrere Betriebssysteme auf einmal zu bedienen, müssen wir das sehr unterschiedliche Verhalten von Windows und Mac überwinden.

Das ist nicht einfach.

Der Weg, dies zu bewältigen, besteht darin, alle Standardkomponenten in maßgeschneiderte Komponenten umzuwandeln die nun zwei Eigenschaften in den Einstellungen des Objektinspektors zur Verfügung stellt: CSS und HTML (noch mehr: Unterstützung von Code in Sight und Debugging). Vielleicht ist sogar noch mehr nötig.

Die beste und einfachste Art, sich das vorzustellen, ist zu sagen, dass die Abwärtskompatibilität NICHT unterbrochen werden wird. Sie können weiterhin die Standardkomponenten verwenden, aber die grafischen Extras werden in diesen neuen Komponenten stark erweitert.

Sogar die Kompatibilität mit Delphi wird beibehalten. Nehmen wir eine Schaltfläche als Beispiel. Die Standard-Schaltfläche verfügt nicht über die Möglichkeit, Farben zu verwenden.

Hier können Sie sie hinzufügen.

Sie können sogar eine runde oder ovale Schaltfläche mit eingefärbter Leinwand oder eingefärbtem Text erstellen.

Auf Seite 65 dieser Ausgabe werden in dem Artikel, der das Projekt FRESNEL vorstellt, die Einzelheiten erklärt.

Die Grafiken werden genauso schön zu handhaben sein, weil wir "Skia" für Grafiken verwenden. Aus diesem Grund bauen Michael und Mattias den Teil von FRESNEL:

das Rendering-Programm.

Es wird auch in der Lage sein, Ihre PAS2JS-Formulare zu erstellen, so dass wir ein Formular in Lazarus haben werden, das Ihnen das so sehr gewünschte WYSIWYG bietet: "What you see is what you get".

Es dauert eine Weile, bis das alles fertig ist. Aber wir arbeiten unermüdlich daran.

Um das alles zu schaffen, brauchen wir Ihre Unterstützung. Wir werden eine Fundraising-Kampagne starten und hoffen, dass Sie uns dabei unterstützen werden.



Ich hoffe Ihnen in der nächsten Ausgabe mitteilen zu können, wann das geschehen wird: das Seminar am 15. April dieses Jahres in den Niederlanden

- bitte registrieren Sie sich - wo wir dieses neue Projekt zeigen, erklären und ausprobieren werden.

Wir haben bereits ein Proof of Concept und in drei Monaten haben wir sicher noch mehr zu zeigen.

Das Seminar ist mehrsprachig. (*Englisch Deutsch und Holländisch*)

Wir haben uns entschieden, die nächste Veranstaltung im April dieses Jahres zu machen, weil es dann wärmer ist und wir etwas Zeit brauchen, um alles vorzubereiten.

Wir werden die Anmeldegebühr so niedrig wie möglich halten: € 50.

Wo und wie, werden wir so bald wie möglich bekannt geben.

In diesem Magazin finden Sie eine ganze Reihe von Artikeln: 14 an der Zahl.

Ich hoffe, es wird Ihre Erwartungen erfüllen.

Noch ein Wort zur Zukunft:

Ich brauche Hilfe: Jeder, der daran interessiert ist, beim Schreiben von Artikeln oder beim Testen oder Überprüfen und Korrigieren von Texten zu helfen.

Vielleicht sogar bei der Gestaltung helfen? Bitte lassen Sie es mich wissen.

(editor@blaisepascal.eu).

Denken Sie daran, dass wir alle Freiwillige sind, die das alles erstellen. Und diese Arbeit können Sie von zu Hause aus erledigen!

Vielleicht sind Sie interessiert an "Cross compiling": Wenn Sie für eine andere Plattform oder ein anderes Format kompilieren möchten, lesen Sie den Artikel darüber:

CROSS-COMPILING MIT FPC UP DE LUXE.

Er nimmt Ihnen all die lästigen Hürden, die Sie überwinden müssen, und erleichtert Ihnen das Leben als Programmierer ungemein.

Lassen Sie mich von Ihnen hören.

Neben FRESNEL arbeiten wir auch an der nächsten Stufe eines PDF-Programms.

Michael van Canneyt hat begonnen eine Indexierungsmaschine zu entwickeln, die ich für die MAGAZINE PDF-Bibliothek verwenden werde. Damit wird es sehr schnell und einfach sein, das gewünschte Thema oder den gewünschten Inhalt in allen Magazinen zu finden.

Für die Zukunft: Der Plan ist, ein komplettes PDF-Programm zu entwickeln, das das kann, was Adobe kann, und dann für alle Betriebssysteme, so dass es keine Probleme mehr gibt, PDF auf Linux oder Mac zu verwenden.
- Die Qualität muss gewährleistet sein.

Wir nennen das Projekt "**Panther**" und Sie werden regelmäßig von uns unterrichtet.

Mit diesem Projekt - für jedermann frei nutzbar - wollen wir Lazarus/FPC zum Endanwender bringen, wo auch immer er sein mag, und immer OPEN SOURCE sein.

Viel Spaß beim lesen...

Detlef





*Meine Bildschirm-zeit ist nur um 10.000% gestiegen.
Ich muss letzte Woche abgelenkt gewesen sein*



LIBRARY 2019

LIBRARY 2020

LIBRARY 2021

- Search
in the whole text of one issue
 - Search author
 - Search title
 - Search Issue Nr
- Your own PDF Viewer
use with any PDF file(s)
- Shows thumbnails
on and of
- Acces to all items
of Blaise Pascal Magazine
starting at issue 1 up to the latest
- Including all Code

- LIBRARY 2023
BLAISE PASCAL MAGAZINE
THE NEXT LEVEL

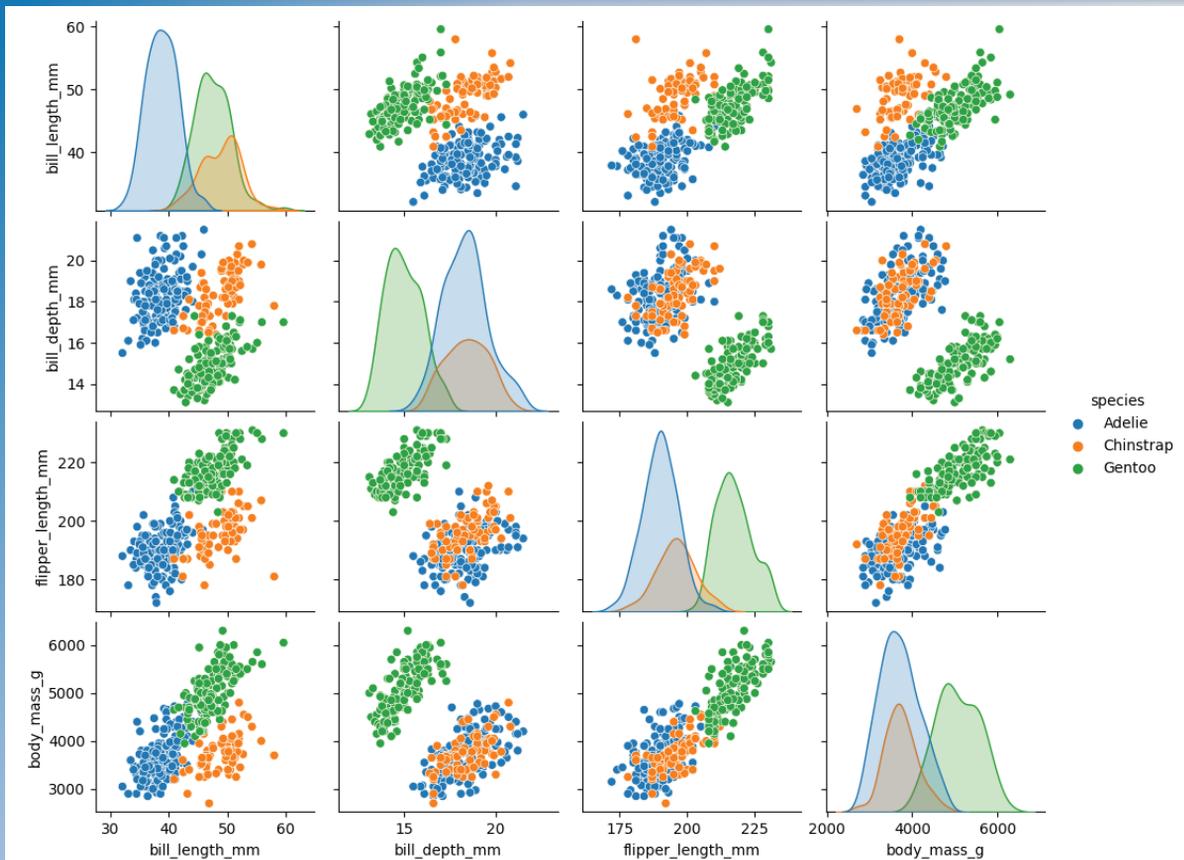
Starter

Expert

D11

Der Irisblumen-Datensatz oder Fisher's Iris-Datensatz ist ein multivarianter Datensatz, den der britische Statistiker und Biologe Ronald Fisher 1936 in seiner Arbeit vorgestellt hat.

In diesem Beitrag möchte ich eine Lektion über maschinelles Lernen mit Python4Delphi vorstellen. Der Palmer Archipelago (Antarktis) Pinguin-Datensatz scheint ein Ersatz für denselben zu sein. Es ist ein großartiger Einstiegsdatensatz für die Datenexploration und -visualisierung. Der Pinguin-Datensatz enthält jedoch eine unterschiedliche Anzahl von Proben für jede Art. Im Gegensatz zum Iris-Datensatz enthält dieser Datensatz eine unterschiedliche Anzahl von Einträgen für die einzelnen Arten.



Links und Quellen

<https://www.kaggle.com/code/parulpandey/penguin-dataset-the-new-iris>

[python4delphi/Tutorials/Webinar II at master · maxkleiner/python4delphi \(github.com\)](https://github.com/python4delphi/Tutorials/Webinar-II-at-master)

<https://maxbox4.wordpress.com/>



The screenshot shows the RAD Studio IDE with a Python script in the main editor. The script uses the `delphi_module` to generate SVG images and `seaborn` for data visualization. The script includes the following code:

```
ANALYTICSSVG='#from delphi_module import svg_image
+from io import StringIO'+LF
+import matplotlib.pyplot as plt'+LF
+import seaborn as sns'+LF
+df = sns.load_dataset("penguins")+LF
+sns.pairplot(df, hue="species")+LF
'+LF
+figfile = StringIO()'+LF
+plt.savefig(figfile, format="svg")+LF
'+LF
procedure pyBank_VCL4Python;
var eg: TPythonEngine; sw: TStopWatch;
begin
eg := TPythonEngine.Create(nil);
try
eg.pythonhome := PYHOME;
eg.loadDLL;
end;
end;
```

The figure titled "Figure 1" is a Seaborn pairplot showing the relationship between `bill_length_mm`, `flipper_length_mm`, and `body_mass_g` for three species: Adelie, Chinstrap, and Gentoo. The diagonal of the plot shows kernel density estimates (KDEs) for each variable, while the off-diagonal plots show scatter plots of the variables. The legend indicates that blue represents Adelie, orange represents Chinstrap, and green represents Gentoo.

RAD Studio 11.4, maXbox4 und Python 3.8 mit Seaborn

The screenshot shows a Python script in the RAD Studio IDE. The script uses `delphi_module` to generate SVG images and `matplotlib` for data visualization. The script includes the following code:

```
from delphi_module import svg_image
from io import StringIO
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colorbar as cbook

# Load a numpy record array from yahoo csv data with fields
# volume, adj_close from the mpl-data/example directory. The
# stores the date as an np.datetime64 with a day unit ('D')
price_data = (cbook.get_sample_data('goog.npz', np_load=True)
.view(np.recarray))
price_data = price_data[-250:] # get the most recent 250 tr

delta1 = np.diff(price_data.adj_close) / price_data.adj_close

# Marker size in units of points^2
volume = (15 * price_data.volume[:-2] / price_data.volume[0]
close = 0.003 * price_data.close[:-2] / 0.003 * price_data.c

fig, ax = plt.subplots()
ax.scatter(delta1[:-1], delta1[1:], c=close, s=volume, alpha=0.5)

ax.set_xlabel(r'\Delta_i$', fontsize=15)
ax.set_ylabel(r'\Delta_{i+1}$', fontsize=15)
ax.set_title('Volume and percent change')

ax.grid(True)
fig.tight_layout()
```

The figure titled "Volume and percent change" is a scatter plot showing the relationship between Δ_i (x-axis) and Δ_{i+1} (y-axis). The data points are colored based on their volume, with larger volumes represented by larger, more opaque bubbles. The plot shows a positive correlation between the two variables, with a dense cluster of points around the origin and a few outliers with high volume and high Δ_{i+1} .



The template for the most P4D programs goes like this:

```
with TpythonEngine.Create(Nil) do begin
pythonhome:= PYHOME;
try
loadDLL;
Println('Decimal: '+
EvalStr('__import__("decimal").Decimal(0.1)'));
except
raiseError;
finally
free;
end;
end;
```

Die minimale Konfiguration hängt von Ihrer Python-Installation und der Eigenschaft UseLastKnownVersion in TDynamicDll ab, aber sobald sie bekannt ist, geht es auch mit raiseError, um die Python-Ausnahmen in Ihrem Code abzufangen. Für die Const von PYHOME auf meinem Rechner hängt es von der Version ab:

```
Const PYHOME = 'C:\Users\max\AppData\Local\Programs\Python\Python36-32\';
PYHOME = 'C:\Users\max4\AppData\Local\Programs\Python\Python37-32\';
```

Vom Code zu einer kurzen Einführung in die Form des Kopfes eines Pinguins mit seinem Culmen.

Der obere Rand des Schnabels wird als Culmen bezeichnet. Die Messung erfolgt mit einem Messschieber, wobei ein Kiefer an der Spitze des Oberkiefers und der andere an der Schädelbasis oder an den ersten Federn angesetzt wird, je nach gewähltem Standard.

Zuerst holen wir uns die Daten mit seaborn:
Import seaborn als sns

```
df = sns.load_dataset("penguins")
```

Oder alternative Quellen

```
df = pd.read_csv('../input/palmer-archipelago-antarctica-penguin-data/penguins_size.csv')
```

```
p2data = "https://raw.githubusercontent.com/allisonhorst/palmerpenguins/
master/data-raw/penguins_raw.csv"
```

```
p2data = "https://raw.githubusercontent.com/rianrajagede/penguin-python/
master/Datasets/penguins.csv"
```

```
Python Chats to Svg maXbox4
matplotlib seaborn

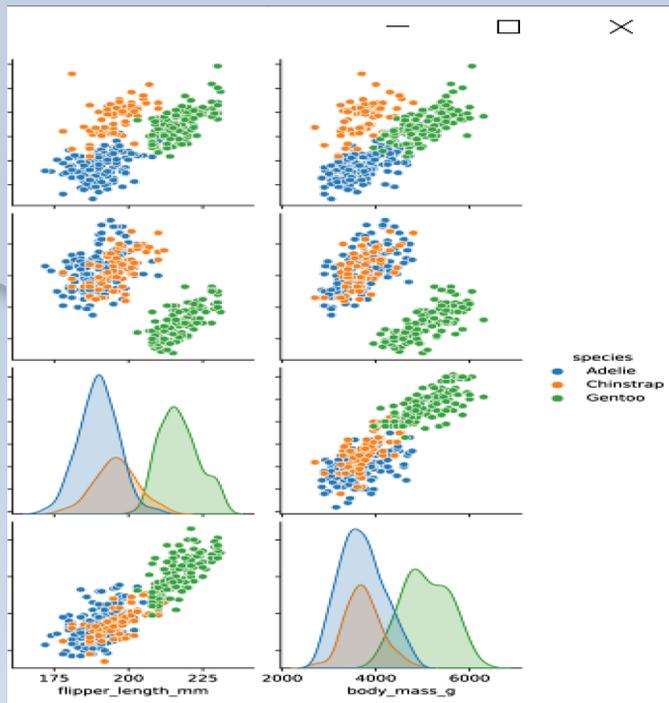
from delphi_module import svg_image
from io import StringIO
import seaborn as sns
import matplotlib.pyplot as plt

df = sns.load_dataset("penguins")
sns.pairplot(df, hue="species")

figfile = StringIO()
plt.savefig(figfile, format='svg')
figdata_svg = figfile.getvalue()
svg_image.SvgText = figdata_svg

#plt.show()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	FEMALE
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	FEMALE
count	344	344	342.000000	342.000000	342.000000	342.000000	334
unique	3	3	NaN	NaN	NaN	NaN	3
top	Adelie	Biscoe	NaN	NaN	NaN	NaN	MALE
freq	152	168	NaN	NaN	NaN	NaN	168
mean	NaN	NaN	43.921930	17.151170	200.915205	4201.754386	NaN
std	NaN	NaN	5.459584	1.974793	14.061714	801.954536	NaN
min	NaN	NaN	32.100000	13.100000	172.000000	2700.000000	NaN
25%	NaN	NaN	39.225000	15.600000	190.000000	3550.000000	NaN
50%	NaN	NaN	44.450000	17.300000	197.000000	4050.000000	NaN
75%	NaN	NaN	48.500000	18.700000	213.000000	4750.000000	NaN
max	NaN	NaN	59.600000	21.500000	231.000000	6300.000000	NaN



```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   species                344 non-null    object
1   island                 344 non-null    object
2   culmen_length_mm      342 non-null    float64
3   culmen_depth_mm      342 non-null    float64
4   flipper_length_mm     342 non-null    float64
5   body_mass_g           342 non-null    float64
6   sex                   334 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
memory shape core cube #7
```

The dataset consists of 7 columns.

- **species**: penguin species (Chinstrap, Adélie, or Gentoo)
- **culmen_length_mm**
- **culmen depth (mm)**
- **flipper_length_mm**: flipper length (mm)
- **body_mass_g**: body mass (g)
- **island**: island name (Dream, Torgersen, or Biscoe) in the Palmer Archipelago (Antarctica)
- **sex**: penguin sex

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. On one short code snippets we get the scatter-plot above with the following lines:

```
import numpy as np

import pandas as pd
from io import StringIO
import matplotlib.pyplot as plt
import seaborn as sns
df = sns.load_dataset("penguins")
sns.pairplot(df, hue="species")
figfile = StringIO()
plt.savefig(figfile, format="svg")
plt.show()
print(df.info())
```

A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables.

The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

A common modification of the basic scatter plot is the addition of a third variable. Values of the third variable can be encoded by modifying how the points are plotted.

For a third variable that indicates categorical values (like geographical region, suburbs or gender), the most common encoding is through point color.

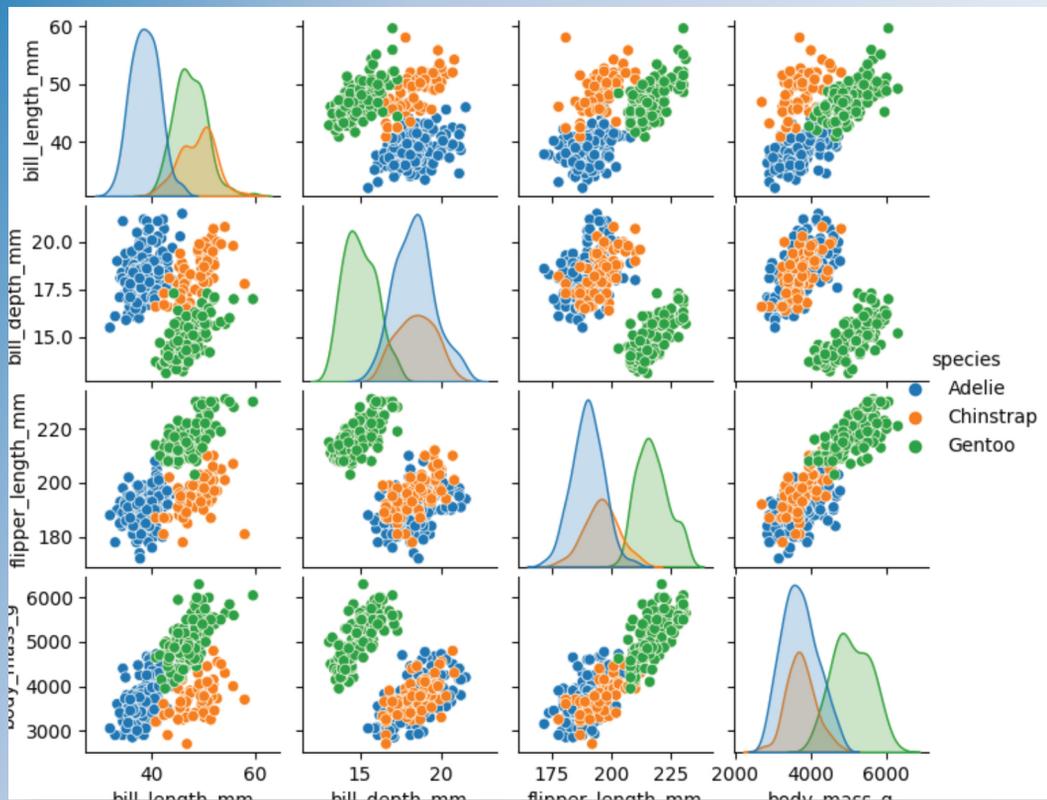
Seaborn lets you create relational plots using the `relplot()` function. The function technically lets you create more than one scatter plot.

1.2 COVARIANCE

“Covariance” indicates the direction of the linear relationship between variables. “Correlation” on the other hand measures both the strength and direction of the linear relationship between two variables.

Source: <https://tinyurl.com/yd2pezss>

Second correlations describe the mutual relationship between two variables. They provide the possibility to measure the relation between any kind of data - continuous and continuous, categorical and categorical, and also continuous and categorical at the same time.



The scatter plot which shows us the correlation with respect to other features. This method helps just to figure out the important features which account the most for the classification in our model.



Offizielle Versionen von seaborn können von der PyPI installiert werden:
pip install seaborn

Der Basisaufruf von pip installiert seaborn und, falls erforderlich, seine obligatorischen Abhängigkeiten. Es ist möglich, optionale Abhängigkeiten hinzuzufügen, die den Zugriff auf einige erweiterte Funktionen ermöglichen:

pip install seaborn[stats].

Numpy-Arrays sind ein guter Ersatz für Python-Listen. Sie sind besser als Python-Listen. Sie sind schneller und benötigen weniger Speicherplatz. Beginnen wir mit der Definition für diejenigen, die Numpy-Arrays noch nicht kennen. Sie sind mehrdimensionale Matrizen oder Listen fester Größe mit ähnlichen Elementen.

Pandas ist eine beliebte Python-Bibliothek, die für die Bearbeitung von Tabellendaten verwendet wird. Sie bietet ein vielseitiges Dataframe-Objekt, das Daten aus vielen gängigen Formaten wie Excel, SQL, CSV und mehr lesen kann.

Die Pandas Style-API stellt Ihnen viele verschiedene Tools zur Verfügung, die die Arbeit mit der Gestaltung von Tabellendaten erheblich erleichtern.

1.3 STREUDIAGRAMM

Ein Streudiagramm kann auch ein Paardiagramm mit Werten für zwei verschiedene numerische Variablen sein. Die Position jedes Punktes auf den verschiedenen horizontalen und vertikalen Achsen zeigt die Werte für einen einzelnen Datenpunkt an. Streudiagramme werden hauptsächlich verwendet, um Beziehungen zwischen Variablen zu beobachten.

Ein Pairplot hingegen stellt die paarweisen Beziehungen in einem Datensatz dar. Die Funktion pairplot erstellt ein Achsen-gitter, so dass jede Variable in den Daten auf der y-Achse in einer einzigen Zeile und auf der x-Achse in einer einzigen Spalte dargestellt wird. Dadurch werden Plots wie oben gezeigt erstellt.

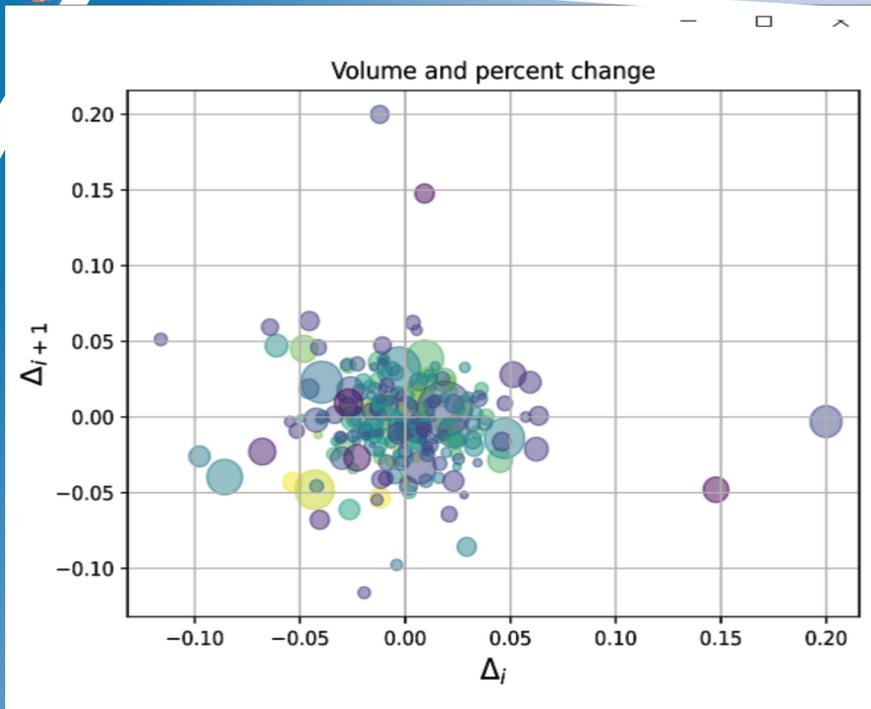
```
from delphi_module import np_array

print("type(np_array) = ", type(np_array))
print("len(np_array) = ", len(np_array))
print("np_array = ", np_array)

res_array = np_array.copy()
for i in range(len(np_array)):
    res_array[i] *= np_array[i]
print("res_array = ", res_array)
```

```
type(np_array) = <class 'numpy.ndarray'>
len(np_array) = 10
np_array = [ 1  2  3  4  5  6  7  8  9 10]
res_array = [ 1  4  9 16 25 36 49 64 81 100]
type(np_array) = <class 'numpy.ndarray'>
len(np_array) = 10
np_array = [ 1  2  3  4  5  6  7  8  9 10]
res_array = [ 1  4  9 16 25 36 49 64 81 100]
```





Python Chats to Svg maXbox4

matplotlib seaborn

```
from delphi_module import svg_image
from io import StringIO
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cbook as cbook

# Load a numpy record array from yahoo csv data with fields
# volume, adj_close from the mpl-data/example directory. The
# stores the date as an np.datetime64 with a day unit ('D')
price_data = (cbook.get_sample_data('goog.npz', np_load=True)
              .view(np.recarray))
price_data = price_data[-250:] # get the most recent 250 tr

delta1 = np.diff(price_data.adj_close) / price_data.adj_close

# Marker size in units of points^2
volume = (15 * price_data.volume[:-2] / price_data.volume[0])
close = 0.003 * price_data.close[:-2] / 0.003 * price_data.close[0]

fig, ax = plt.subplots()
ax.scatter(delta1[:-1], delta1[1:], c=close, s=volume, alpha=0.5)

ax.set_xlabel(r'$\Delta_i$', fontsize=15)
ax.set_ylabel(r'$\Delta_{i+1}$', fontsize=15)
ax.set_title('Volume and percent change')

ax.grid(True)
fig.tight_layout()
```

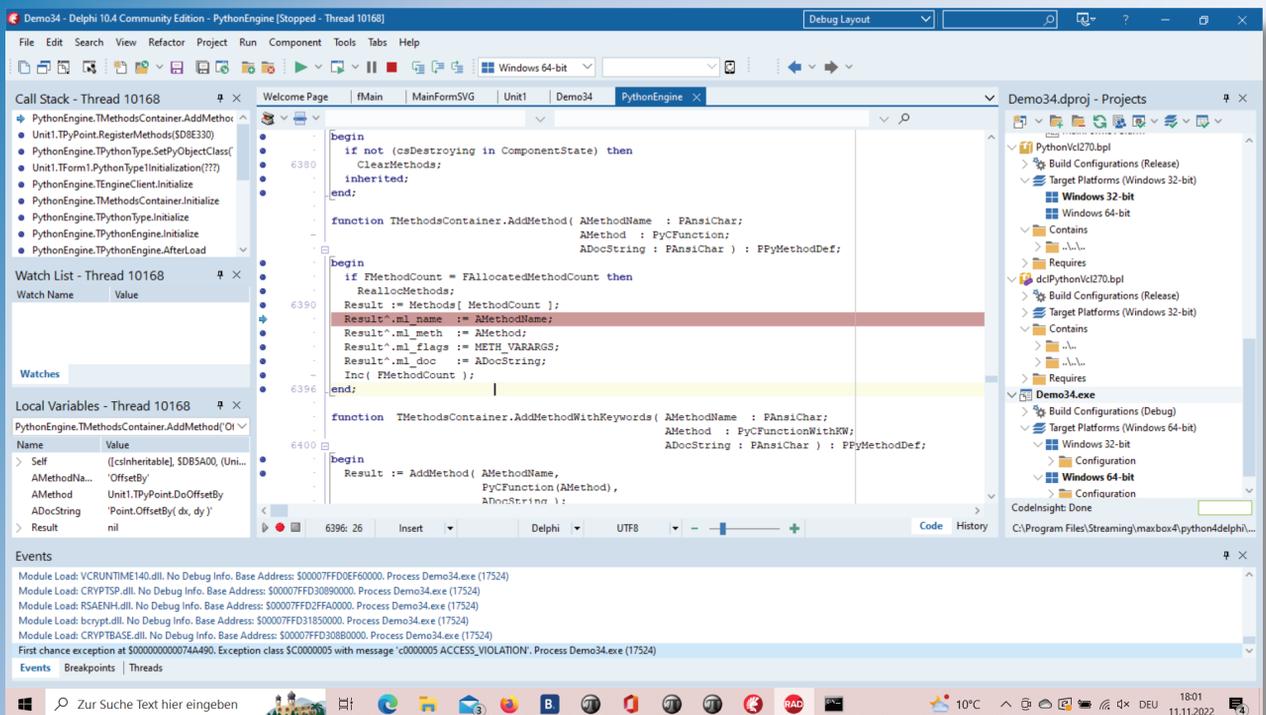
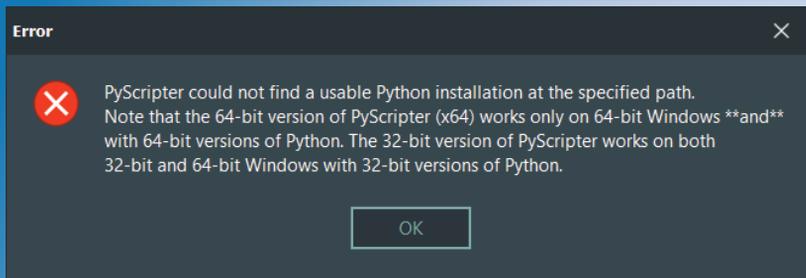




2 TECHNICAL DESCRIPTION

maxkleiner (Max Kleiner) (github.com)

Ein Python-Programm bricht ab, sobald es auf einen Fehler stößt. In Python kann ein Fehler ein Syntaxfehler oder eine Ausnahme sein. In diesem Artikel erfahren Sie, was eine Ausnahme ist und wie sie sich von einem Syntaxfehler unterscheidet. Danach erfahren Sie, wie man Ausnahmen auslöst und Behauptungen aufstellt. Zum Abschluss erhalten Sie eine Demonstration des try- und except-Blocks. <https://realpython.com/python-exceptions/>

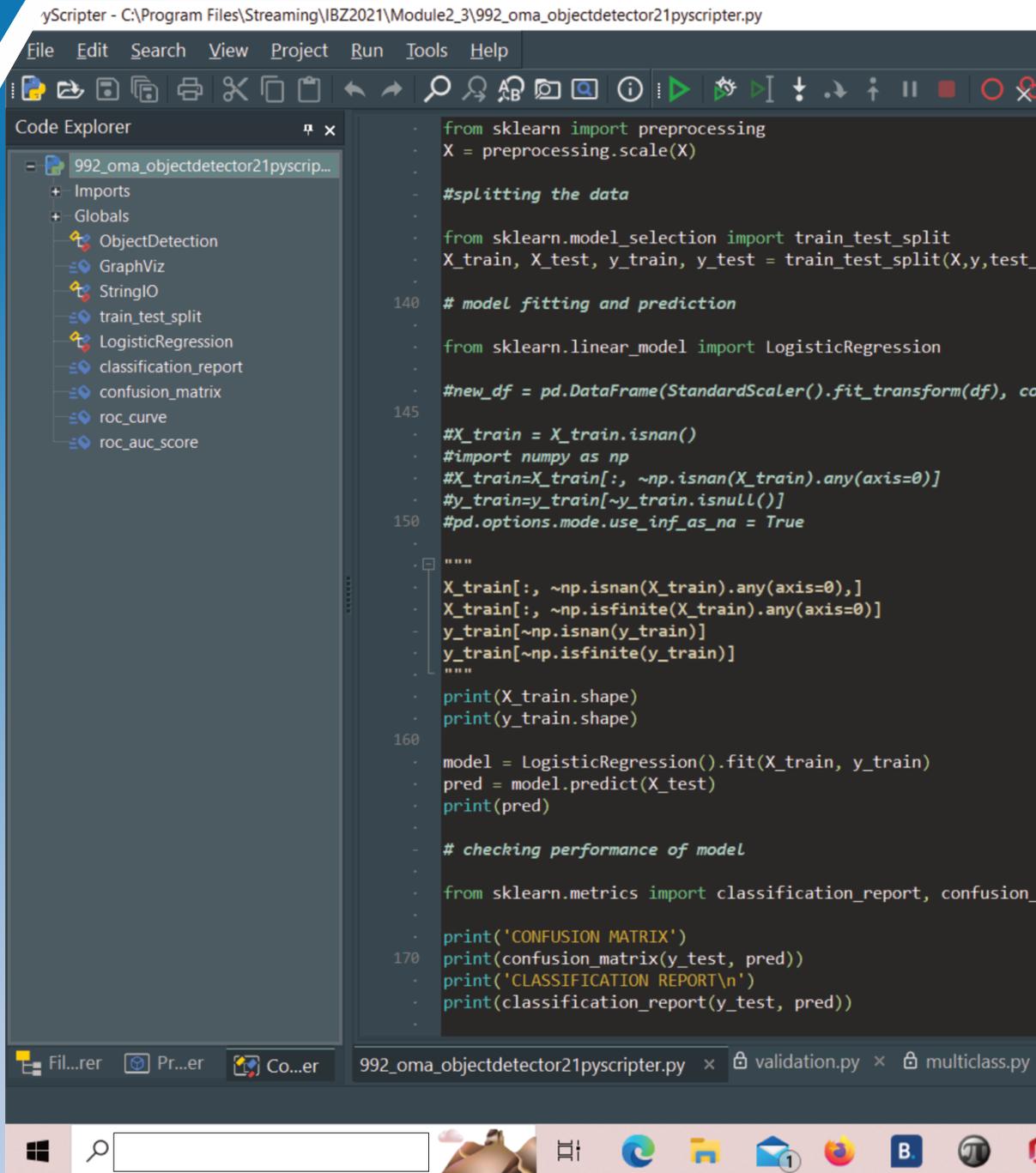


PyScripter 4.0.0 ist jetzt bei Sourceforge verfügbar. PyScripter ist vollständig in Delphi programmiert. Dies ist eine wichtige neue Version mit einer aktualisierten Benutzeroberfläche und vielen bedeutenden Verbesserungen unter der Haube, die die Stabilität erhöhen und die Benutzererfahrung verbessern werden.

Scalable Vector Graphics (SVG) ist ein XML-basiertes Vektorgrafikformat zur Definition zweidimensionaler Grafiken mit Unterstützung für Interaktivität und Animation. Die SVG-Spezifikation ist ein offener Standard, der seit 1999 vom World Wide Web Consortium entwickelt wird.

<https://en.wikipedia.org/wiki/SVG>





```
PyScripter - C:\Program Files\Streaming\IBZ2021\Module2_3\992_oma_objectdetector21pyscrip.py
File Edit Search View Project Run Tools Help
Code Explorer
- 992_oma_objectdetector21pyscrip...
+ Imports
+ Globals
+ ObjectDetection
+ GraphViz
+ StringIO
+ train_test_split
+ LogisticRegression
+ classification_report
+ confusion_matrix
+ roc_curve
+ roc_auc_score

from sklearn import preprocessing
X = preprocessing.scale(X)

#splitting the data

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_

140 # model fitting and prediction

from sklearn.linear_model import LogisticRegression

#new_df = pd.DataFrame(StandardScaler().fit_transform(df), co

145

#X_train = X_train.isnan()
#import numpy as np
#X_train=X_train[:, ~np.isnan(X_train).any(axis=0)]
#y_train=y_train[~y_train.isnull()]
150 #pd.options.mode.use_inf_as_na = True

"""
X_train[:, ~np.isnan(X_train).any(axis=0),]
X_train[:, ~np.isfinite(X_train).any(axis=0)]
y_train[~np.isnan(y_train)]
y_train[~np.isfinite(y_train)]
"""

print(X_train.shape)
print(y_train.shape)

160 model = LogisticRegression().fit(X_train, y_train)
pred = model.predict(X_test)
print(pred)

# checking performance of model

from sklearn.metrics import classification_report, confusion_

print('CONFUSION MATRIX')
170 print(confusion_matrix(y_test, pred))
print('CLASSIFICATION REPORT\n')
print(classification_report(y_test, pred))
```

Haben Sie genug von IDLE? Möchten Sie in einer schicken neuen IDE programmieren? Diese Schritte helfen Ihnen beim Herunterladen von Python 2.7.1 oder Python 3.8 und PyScripter damit Sie anfangen können, Python zu lernen und Spaß damit zu haben.

<https://www.instructables.com/Python-With-PyScripter/>



The screenshot shows a Jupyter Notebook interface with a Python Interpreter window open. The interpreter displays the following output:

```
CONFUSION MATRIX
[[32  1  0]
 [ 1 11  0]
 [ 0  0 22]]
CLASSIFICATION REPORT
```

	precision	recall	f1-score	support
Adelie	0.97	0.97	0.97	33
Chinstrap	0.92	0.92	0.92	12
Gentoo	1.00	1.00	1.00	22
accuracy			0.97	67
macro avg	0.96	0.96	0.96	67
weighted avg	0.97	0.97	0.97	67

The Python Interpreter window also shows the prompt `>>>` at the bottom. The background of the notebook shows some code snippets, including `size=0.2, random_state=13)` and `columns=df.columns, index=df.index)`. The taskbar at the bottom shows the system tray with a temperature of 7°C, the date 20.11.2022, and the time 11:57.



The screenshot shows the Delphi IDE interface for 'PyVizSVG.dproj'. The Structure pane on the left lists various classes under 'TForm1(TForm)', including 'SynPythonSyn1', 'PythonEngine1', and 'PyDelphiWrapper'. The Object Inspector shows properties for 'Form1 TForm1', such as 'Caption' set to 'Python Chats to Svg n'. The Messages pane displays a successful compilation message: 'Compiling PyVizSVG.dproj (Debug, Win32) ... Success'. The main editor window shows the following Python code:

```
unit MainFormSVG;  
  
interface  
  
uses  
  Winapi.Windows, Winapi.Messages, System.SysUtl  
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, WrapDel  
  PythonGUIInputOutput, SynEditPythonBehaviour,  
  SynEditCodeFolding, SynHighlighterPython, Vcl
```

A code window titled 'Python Chats to Svg maxbox4' contains the following Python code:

```
matplotlib seaborn  
  
from delphi_module import svg_image  
from io import StringIO  
import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib.cbook as cbook  
  
# Load a numpy record array from yahoo csv data with fields  
# volume, adj_close from the mpl-data/example directory. The  
# stores the date as an np.datetime64 with a day unit ('D')  
price_data = (cbook.get_sample_data('goog.npz', np_load=True  
    .view(np.recarray))  
price_data = price_data[-250:] # get the most recent 250 tr  
  
delta = np.diff(price_data.adj_close) / price_data.adj_clos  
  
# Marker size in units of points^2  
volume = (15 * price_data.volume[:-2] / price_data.volume[0  
close = 0.003 * price_data.close[:-2] / 0.003 * price_data.c  
  
fig, ax = plt.subplots()
```

The Windows taskbar at the bottom shows the search bar with the text 'Zur Suche Text hier eingeben' and several application icons including Edge, File Explorer, Mail, Firefox, and a blue 'B' icon.



maxbox

The screenshot shows a Delphi IDE window titled 'WrapDelphi'. In the foreground, a scatter plot window titled 'Volume and percent change' is open. The plot has a horizontal axis labeled Δ_j ranging from -0.10 to 0.20 and a vertical axis ranging from -0.10 to 0.20. The data points are represented by semi-transparent bubbles of various colors (purple, blue, green, yellow) and sizes, clustered around the origin. In the background, the IDE's Project Explorer shows a project named 'PyVizSVG.dproj'. The project structure includes a 'ProjectGroup1_mX5' containing various files like 'PascalScript_Core_D27.bpl', 'TestApplication64.exe', 'Import64.exe', 'maxbox5_0.exe', 'Python270.bpl', 'dclPython270.bpl', 'AnalyticsDemo.exe', 'PythonVcl270.bpl', 'dclPythonVcl270.bpl', 'Demo34.exe', 'dclSVGIconImageList270.bpl', 'SVGIconImageList270.bpl', 'SVGIconPackage270.bpl', 'SVGImage32Package270.bpl', 'VarPythDemo.exe', and 'PyVizSVG.exe'. The IDE's status bar at the bottom shows the system tray with a temperature of 8°C, the date 12.11.2022, and the time 16:33.



maxbox

3 ÜBERSICHT ENTWICKELN

Wir legen großen Wert auf die Qualität des Materials, das jeder Student lernt. Unsere von Ausbildern geleiteten Kurse bieten praktisches Lernen, persönliche Betreuung durch erfahrene Entwickler und Lernen von Gleichgesinnten. Um den Studenten größtmögliche Flexibilität zu bieten, haben wir Tages- und Abendprogramme zur Auswahl. *Keine Vorkenntnisse im Programmieren erforderlich.

```
procedure pyBank_VCL4Python4Delphi;
var eg: TPythonEngine; sw: TStopWatch;
begin
  eg:= TPythonEngine.Create(nil);
  try
    eg.pythonhome:= PYHOME;
    eg.loadDLL;
    println('test import '+GetPythonEngine.EvalStr('__import__("decimal").Decimal(0.1)'));
    writeln("");
    //println(eg.EvalStr('__import__("faker").Faker()'));
    sw:= TStopWatch.Create();
    sw.Start;
    eg.execStr('import importlib.machinery, importlib.util');
    eg.execStr('from decimal import Decimal, getcontext');
    importlib.machinery.ExtensionFileLoader("DelphiVCL",'+VCLHOME+')

    eg.execStr(ANALYTICSSVG);
    //println(eg.evalStr("#39'MSE:#39',mean_squared_error(test_bayesian_pred,y_test_
    confirmed)'));
    sw.Stop;
    //sw.ElapsedMilliseconds;
    writeln('Stop Analytics Tester1: '+sw.getValueStr)
  except
    eg.raiseError;
    writeln(ExceptionToString(ExceptionType, ExceptionParam));
  finally
    eg.Free;
    sw.Free;
    sw:= Nil;
  end;
end;
```

Fazit:

Im Moment können Sie vier Engines zum Rendern von SVG verwenden (Delphi Image32, Delphi TSVG, SKIA4Delphi und Direct2D Wrapper) und vier Komponenten (Pakete), um die Verwendung von SVG-Bildern zu vereinfachen (Größenänderung, feste Farben, Graustufen usw.). Mit Python4Delphi steht eine fünfte Engine zur Verfügung:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  var Py:= PyDelphiWrapper.Wrap(SVGIconImage, soReference);
  PythonModule.SetVar('svg_image', Py);
  GetPythonEngine.Py_DECREF(Py);
end;
```

Tested on

- Delphi 10.4 Community Edition Update 2
- Lazarus V 2.2.4 for Windows 64 bit
- maXbox4 Version 4.7.6.20 x32
- PyScripter 4.1.1.0 x64
- SVGImage32Package270.bpl V 3.9.4

<https://github.com/EtheaDev/SVGIconImageList>



LibraryForm

Show article Go to issue Nr Issue Search

ID	IssueNr	Author	Article	PDF	PageNr
884	103_104	Michael Philippenko	FastReport: New version / Using the table Object in Table Reports		19
885	103_104	David Dirkse	The Ceva project		25
886	103_104	David Dirkse	The distance of diagonals in a cube		30
887	103_104	Detlef Overbeek	Creating Electron apps with Pas2JS		34
888	103_104	Detlef Overbeek	Quantum Cryptography		48
889	103_104	Detlef Overbeek	SKIA voor Delphi: the next level		64
890	103_104	Detlef Overbeek	Installing FPC Trunk versions		82
891	103_104	Michael van Canneyt	Tested: The new version of Database Workbench		89
892	103_104	Michael van Canneyt	Pas2JS: Samples for creating your new website		111
893	103_104	Michael van Canneyt	Anonymous functions: Explained for Delphi and Lazarus		133

Show Thumbnails

Page Jump to page

Page 131
Page 132
Page 133
Page 134
Page 135
Page 136
Page 137
Page 138
Page 139
Page 140
Page 141
Page 142
Page 143
Page 144
Page 145
Page 146

BLAISE PASCAL MAGAZINE 107/108

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

Blaise Pascal

Deutsche Ausgaben zugefügt

Penguin Data Science: 4 engines to render Vector graphics
FRESNEL: the graphical project for Lazarus: creating colours everywhere
PANTHER: the super fast, in Lazarus created PDF indexer for all OS's explained
Using styles with a VCL program
The Pendulum: time acts like a wave
Lazarus programme using BufDataset
Cross compiling with FPC Up De Lux
Accelerating the Delphi Compiler
multi-node quantum network - Quantum Internet News
What's open about smart chatbot maker OPENAI
Discussion about OPEN GP
How to create a BufDataset Master Data
Using Styles in Delphi
Installing FastReport packages in Lazarus for Linux/Windows
Zeos database components for Lazarus

BLAISE PASCAL MAGAZINE

procedure
var
begin
begin
end
end

Prof Dr Wirth, Creator of Pascal Programming language

Blaise Pascal, Mathematician

BLAISE PASCAL MAGAZINE

procedure
var
begin
begin
end
end

Prof Dr Wirth, Creator of Pascal Programming language

Blaise Pascal, Mathematician

Editor in Chief: Detlef Overbeek
Edelstenenbaan 21 3402 XA
Isselstein Netherlands

Prof Dr Wirth, Creator of Pascal Programming language

editor@blaisepascalmagazine.eu
https://www.blaisepascalmagazine.eu

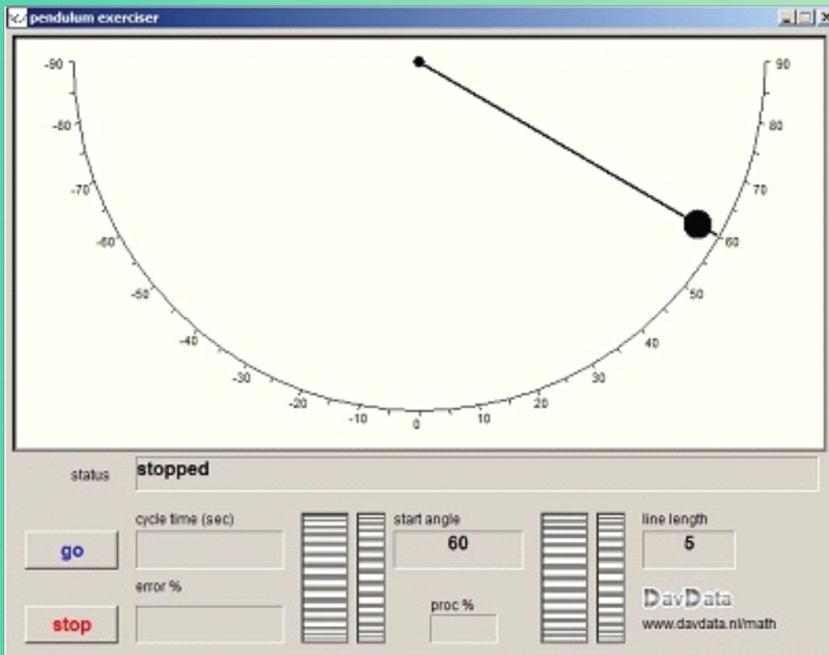
PDF-Betrachter 2023 auf
Kreditkarte USB Stick inklusive
Blaise Pascal Bibliothek



EINFÜHRUNG

Dieses kleine Delphi-Projekt simuliert ein Pendel. Der Hauptzweck besteht darin, die Zykluszeit zu messen, was auf numerische Weise geschieht. Ein analytischer Ansatz ist schwierig, da die Zykluszeit von der Amplitude des Pendels abhängt.

Unten sehen Sie ein Bild des Projekts in Delphi 7:



Mit den Buttons für die vertikale Drehung können Sie den Winkel, in dem die Masse anfänglich freigegeben wird, und die Länge des Kabels auswählen.

OPERATION

- Winkel in Grad wählen (*Maus über die Buttons für die Drehung bewegen*)- Linienlänge auswählen (..)
- drücken Sie die Button "**Go**", um die Masse freizugeben

Drücken Sie "**Stop**", um die Simulation zu beenden.

Das Display zeigt an:

- Zykluszeit
- Genauigkeit der simulierten Bewegung
- Prozentsatz der Zeit, die der Prozessor für die Simulation und die Anzeige benötigt

BASIS-KINEMATIK

Die Einheit der Länge ist der Meter (m).
Die Einheit der Zeit ist die Sekunde (sec).

Bei konstanter Bewegung:

Entfernung = Geschwindigkeit * Zeit.

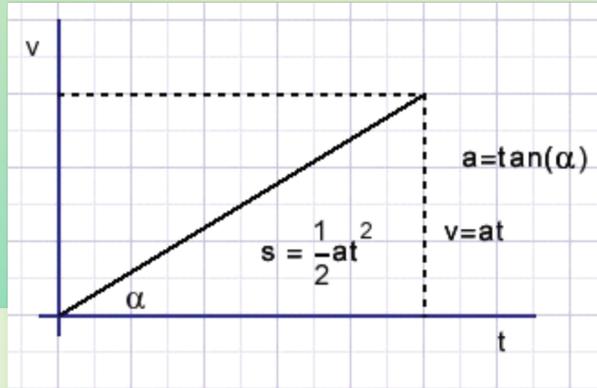
$$s = vt \quad \{s : \text{distance (m)}, \\ v : \text{velocity (m/sec)}, t : \text{time (sec)}\}$$

Die Beschleunigung ist die Geschwindigkeitsänderung pro Sekunde.

Im Falle einer konstanten Beschleunigung:

$$v = at \quad \{v : \text{velocity (m/sec)}, \\ a : \text{acceleration (m/sec}^2), t : \text{time (sec)}\}$$





Der Abstand s ist der Flächeninhalt des Dreiecks:
 $s = 0,5at^2$

Für konstante Beschleunigung:
 Wenn v_t die Geschwindigkeit eines Objekts zum Zeitpunkt t :

$$v_t = v_0 + at$$

Wenn s_t die Entfernung in der Zeit t ist

$$s_t = v_0 t + 0,5at^2$$

Die Einheit der Masse ist das Kilogramm, das von der Schwerkraft unabhängig ist.

Die Einheit der Kraft ist das NEWTON

Ein Newton (N) ist die Kraft, die eine Masse von 1 (kg) in jeder Sekunde um 1m/sec beschleunigt.

$$F = ma \{F: \text{Kraft (N)}, m: \text{Masse (kg)}, a: \text{Beschleunigung (m/sec}^2)\}$$

Auf der Erde ist $a = 9,8$ und wird g genannt.

Die mechanische Energie ist das Produkt aus Kraft und Weg: Die Einheit der Energie ist das Joule (J).

$$E = Fs \{E: \text{Energie (J)}, F \text{ Kraft (N)} s: \text{Entfernung (m)}\}$$

Um eine Masse m über h Meter gegen die Schwerkraft nach oben zu heben, braucht man Energie

$$E_p = mgh \{E_p: \text{potentielle Energie (J)}\}$$

Die Energie geht nie verloren, kann aber in eine andere Form umgewandelt werden,
 z.B. in chemische Energie in Batterien oder in Wärme.

Eine Masse m wird während t Sekunden um a (m/sec²) beschleunigt.

Die Entfernung $s = 0,5at^2$.

$$\text{Energie } E = Fs = F \cdot 0,5 \cdot at^2$$

Da $F=ma$:

$$E = 0,5ma^2t^2$$

da $v=at$:

$$E_k = 0,5mv^2 \{E_k : \text{kinetische Energie (J)}\}$$





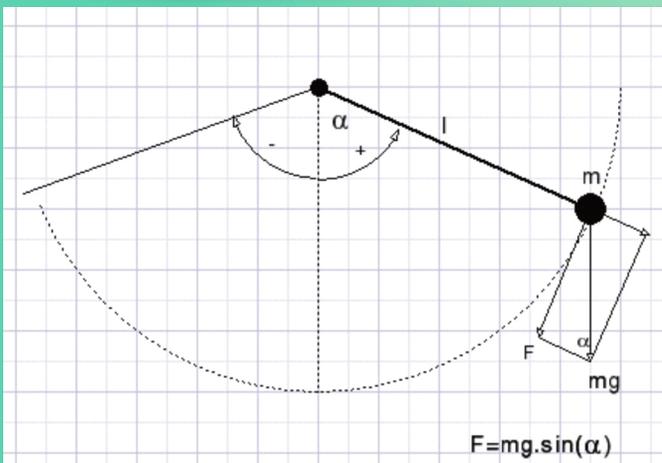
DER SIMULATIONSPROZESS

Beim Pendel geht (theoretisch) keine Energie verloren, aber potentielle und kinetische Energie wechseln sich ab. Die Luftreibung wird nicht berücksichtigt, kann aber leicht hinzugefügt werden. In der untersten Position ist alle Energie kinetisch, in der obersten Position ist alle Energie potentiell. Zwischen diesen beiden Positionen gibt es eine Mischung.

Da die Gesamtenergie konstant ist, können wir die genaue Geschwindigkeit in allen Höhen der Masse berechnen.

Die Masse wird durch die Kraft, die senkrecht zur Linie wirkt, beschleunigt.

Die Kräfte im Seil verbrauchen keine Energie, solange das Seil starr ist.



Wir stellen fest, dass die Kraft F vom Winkel α abhängt.
Wenn Sie α in Radiant messen:

$s := \alpha l$ {s: distance on circle perimeter, l: length of line}

Der Simulationsprozess ist verblüffend einfach.

Die Zeit wird in Mikrosekunden eingeteilt.

Während dieses kleinen Intervalls sollen die Kraft F und die Geschwindigkeit v konstant sein.

F ist aus dem Winkel α bekannt und verursacht die Beschleunigung

$a = F/m = g \cdot \sin(\alpha)$;

Die neue Geschwindigkeit v wird also $v + a \cdot 10^{-6}$.

Der Abstand s in dieser Periode ist $v \cdot 10^{-6}$

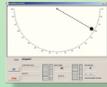
und da $s = \alpha l$

wird der neue Winkel α zu $\alpha + vt/l$.

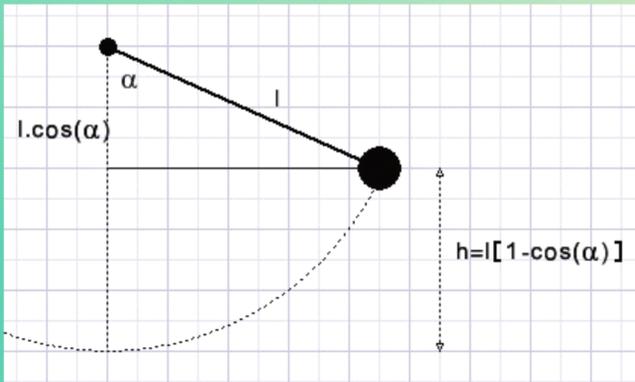
Nach 10000 solcher Schritte wird ein neues Bild angezeigt, das die Position des Pendels zeigt.

Es werden also 100 Bilder pro Sekunde angezeigt.





VERHÄLTNISS ZWISCHEN GESCHWINDIGKEIT UND HÖHE



Freigesetzt unter einem Winkel von α bei h Meter über dem tiefsten Punkt wird die potentielle Energie i

$$mgh = mgl[1 - \cos(\alpha)].$$

Bei einem Winkel von φ ist die potentielle Energie (reduziert auf) $mgl[1 - \cos(\varphi)]$.

Dieser Energieverlust ist nun also kinetische Energie:

$$\frac{1}{2}mv^2 = mgl[\cos(\theta) - \cos(\alpha)]$$

$$v = \sqrt{2gl[\cos(\theta) - \cos(\alpha)]}$$

EIN INTEGRAL FÜR DIE ZYKLUSZEIT

Wegstrecke Δs bei Geschwindigkeit v während einer sehr kleinen Zeit Δt über den Kreisumfang:

$$\Delta s = v \cdot \Delta t \quad s = l\theta$$

$$\Delta t = \frac{\Delta s}{v} = \frac{l \cdot \Delta \theta}{v}$$

$$t = l \cdot \int \frac{d\theta}{v} = l \cdot \int \frac{d\theta}{\sqrt{2gl[\cos(\theta) - \cos(\alpha)]}} = \sqrt{\frac{l}{2g}} \int \frac{d\theta}{\sqrt{\cos(\theta) - \cos(\alpha)}}$$

integration over α to $-\alpha$ yields half the cycle time so total cycle T is

$$T = \sqrt{\frac{2l}{g}} \int \frac{d\theta}{\sqrt{\cos(\theta) - \cos(\alpha)}}$$





Dieses Integral ist schwer zu lösen. (bilden Sie Potenzreihen und integrieren Sie Term für Term)

Besonderer Fall: very small α
 Writing $\cos(x) = 1 - 2\sin^2(x/2)$:
 $\cos(\varphi) - \cos(\alpha) = 2[\sin^2(\alpha/2) - \sin^2(\varphi/2)]$
 wodurch sich das Integral der Zykluszeit ändert zu

$$T = \sqrt{\frac{l}{g}} \int_{-\alpha}^{\alpha} \frac{d\theta}{\sqrt{\sin^2\left(\frac{\alpha}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)}}$$

Jetzt wird Ersetzt

$$u = \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\alpha}{2}\right)} \Rightarrow \frac{du}{d\theta} = \frac{\cos\left(\frac{\theta}{2}\right)}{2\sin\left(\frac{\alpha}{2}\right)} \Rightarrow d\theta = \frac{2\sin\left(\frac{\alpha}{2}\right)}{\cos\left(\frac{\theta}{2}\right)} \cdot du$$

erhalten wir:

$$= 2\sqrt{\frac{l}{g}} \int_{-1}^1 \frac{du}{\sqrt{1-\sin^2\left(\frac{\theta}{2}\right)} \sqrt{1-u^2}} = 2\sqrt{\frac{l}{g}} \int_{-1}^1 \frac{du}{\sqrt{\left(1-u^2\sin^2\left(\frac{\alpha}{2}\right)\right)\left(1-u^2\right)}}$$

Wenn α klein ist, kann dieses Integral reduziert werden auf

$$T = 2\sqrt{\frac{l}{g}} \int_{-1}^1 \frac{du}{\sqrt{1-u^2}} = 2\sqrt{\frac{l}{g}} [\arcsin(u)]_{-1}^1 = 2\pi\sqrt{\frac{l}{g}}$$

DAS PROGRAMM

The pendulum is painted on a bitmap called bm which is drawn on paintbox1 on the form to become visible. See procedure paintbm.

```
const deg2rad : double = pi/180;
      rad2deg : double = 180/pi;
      pi2 : double = pi/2;
var bm : Tbitmap;
  Apos : double;           //angle in radius
  line : double;          //line length selection (m)
  ActiveLine : double;    //actual line length in use
  Astart : word;          //start angle (degrees)
  stopFlag : boolean = true; //set by stop button

function CalcMaxV : double; //calculate maximum velocity
begin
  result := sqrt(2*9.8*activeline*(1 - cos(Apos)));
end;
```

So weit die Theorie.





DAS SIMULATIONSVERFAHREN:

Hinweis: Der Fehlerbericht zeigt die Differenz zwischen der berechneten und der simulierten Geschwindigkeit an der Position der vertikalen Linie.

```

procedure simulate;
var simsteps,wait : word;
    a,V,reportV,calcV,oldpos,dev,oldV,procTime : double;
    cycletime,reportcycletime : dword;
begin
    activeLine := line; //line length may have changed
    cycletime := 0;
    reportcycletime := 0;
    V := 0; //velocity
    reportV := 0; //velocity to report if > 0
    Apos := Astart*deg2rad; //pendulum angle in radius    calcV := calcMaxV; //
maximum velocity at vertical position
    oldpos := 0; //previous Apos
    oldV := 0; //previous velocity
    wait := 100; //delay to avoid too many updates of percentages%
repeat
    form1.timer.start; //record microseconds clock time
    for simsteps := 1 to 10000 do //10000 simulation steps of 1 micro sec.
    begin
        inc(cycletime);
        Apos := Apos - (V/activeline)*1e-6;
        a := 9.8*sin(Apos); //acceleration
        V := V + a*1e-6; //new velocity
        if (oldpos >= 0) and (Apos < 0) then reportV := abs(V); //for error reporting
        oldPos := Apos; //at vertical position
        if (oldV < 0) and (V >= 0) then //at right top turning point
        begin
            reportcycletime := cycletime;
            cycletime := 0;
            if line <> activeline then //if changing line length
            begin
                activeLine := line;
                calcV := calcMaxV;
            end;
        end;
        oldV := V;
    end//for simsteps
    //
    bmrepaint;
    if reportV > 0 then //error report
    begin
        dev := 100*abs(reportV-calcV)/calcV;
        form1.acclabel.Caption := formatfloat('#0.0000000000',dev);
        reportV := 0;
    end;
    if reportcycletime > 0 then //cycle time report
    begin
        form1.cyclelabel.Caption := formatfloat('#0.000##',reportcycletime*1e-6);
        reportcycletime := 0;
    end;
    dec(wait);
    if wait = 0 then //proctime report only 1 in 100 steps
    begin
        form1.timer.stop; //record microseconds clock end time
        procTime := form1.timer.elapsedtime;
        form1.proclabel.caption := formatfloat('#0.0#',0.01*(procTime));
        wait := 100;
    end;
    application.ProcessMessages;
    repeat //delay to multiple of 10 milliseconds
        form1.timer.stop;
        until (form1.timer.Elapsedtime >= 10000) or stopflag;
        until stopflag;
end;

```

EIGENE KOMONENTEN

Das Pendelprojekt verwendet zwei meiner eigenen Komponenten

1. RotationButton. Eine Schaltfläche, die man bei Laborgeräten findet.
2. Ein Mikrosekundenzähler, der auf dem CPU-Takt basiert.

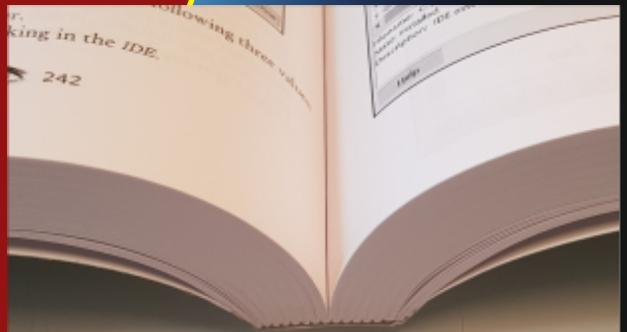
Damit ist die Beschreibung des Pendelsimulators abgeschlossen.



SALE LAZARUS HANDBOOK POCKET

Price: € 26,50
Excluding VAT
and Shipping

- English
- Printed black & white
- 2 Volumes
- PDF included
- 934 Pages
- Weight: 2kg
- Extra protected
- Including 40 Example projects and extra programs



BlaisePascalMagazine
PDF viewer included



INTRODUCTION

Was ist ein Stil (Style)?

Es handelt sich um eine Reihe von grafischen Details, die das Aussehen einer VCL-Anwendung/FireMonkey-Anwendung in Delphi definieren, ähnlich wie ein Thema in Windows.

Mit einem Stil können Sie das Aussehen jeder Komponente und jedes Status eines Steuerelements ändern.

Ein Schieberegler hat zum Beispiel folgende Komponenten: Rahmen, Bildlaufleiste und die beiden seitlichen Schaltflächen für jede Richtung. Die seitlichen Schaltflächen haben beispielsweise die folgenden Zustände: gedrückt, deaktiviert, heiß (betont), normal.

Es ist sehr einfach, Stile in einem VCL-Programm zu verwenden. Starten Sie ein neues Projekt und fügen Sie einige Komponenten in ein Formular ein, wie wir es hier getan haben.

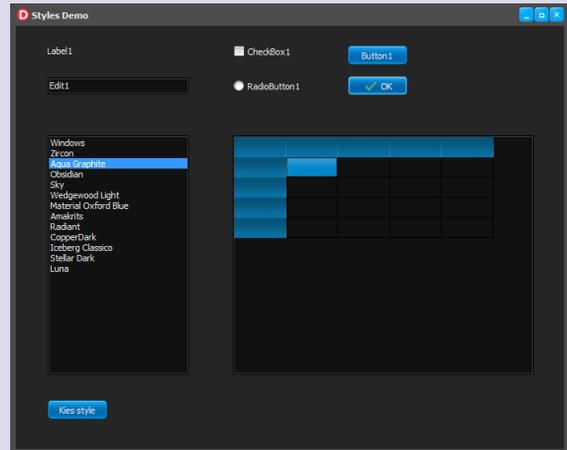


Abbildung 1: Beispiel für eine VCL-Anwendung

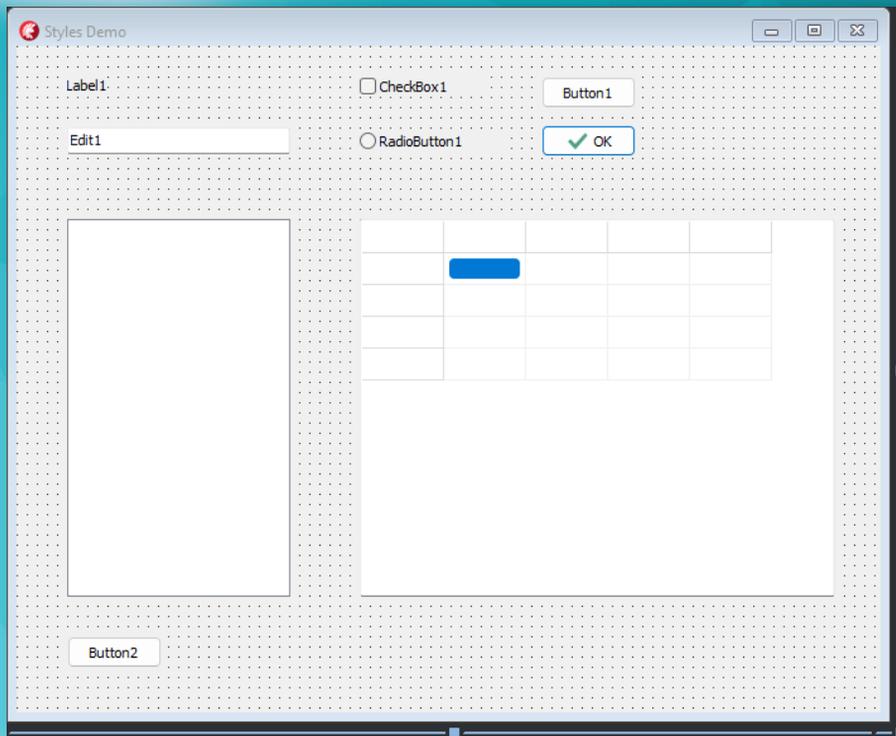


Figure 2: Form in design view.



Gehen Sie dann zu den Projektoptionen (Project -> Options...) und wählen Sie dort „Appearance“:

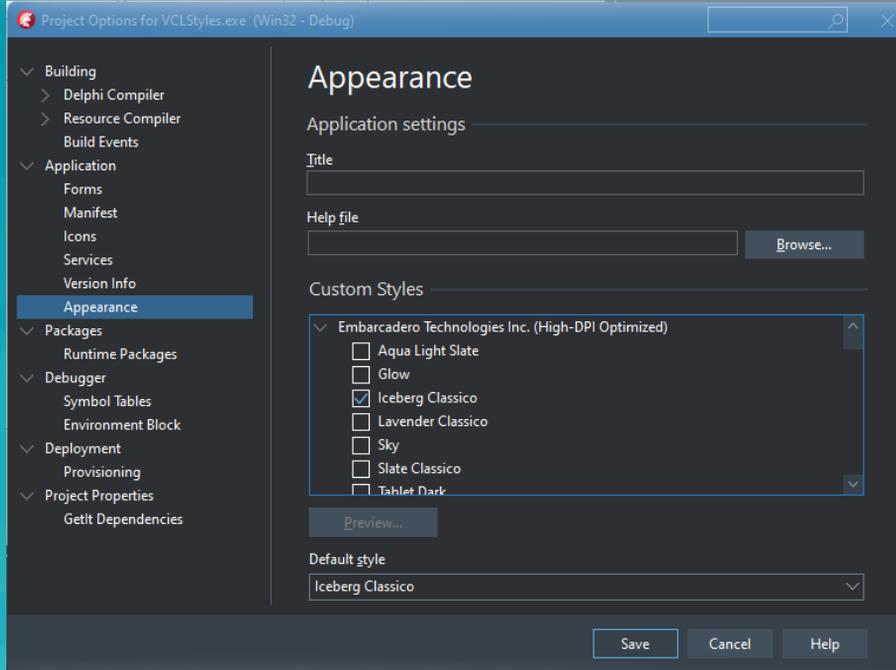
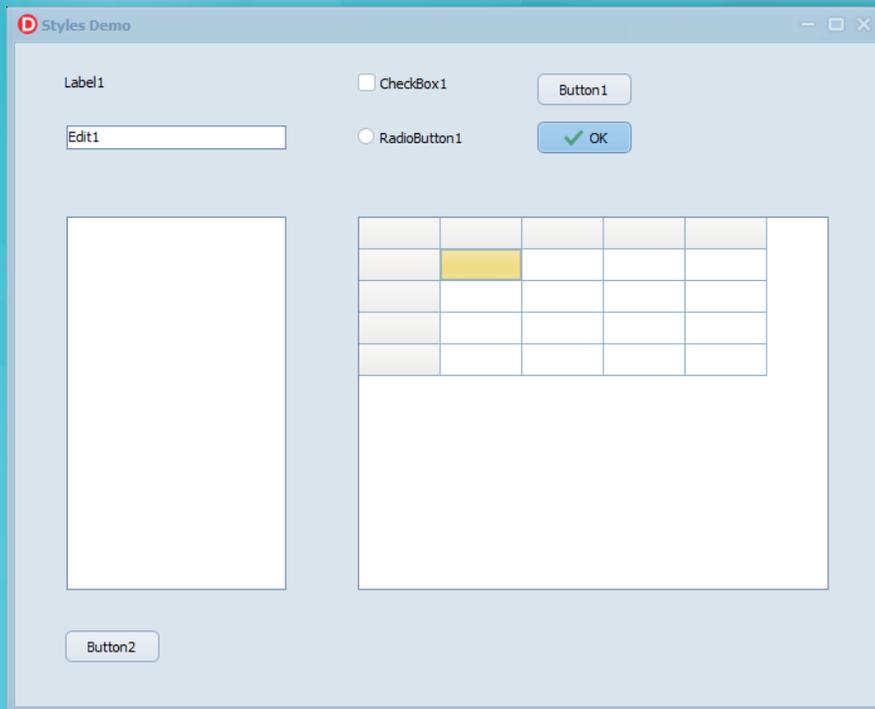


Abbildung 3: Project Option Dialog.

Markieren Sie z.B. Iceberg Classico als Standardstil.

Drücken Sie auf Speichern und führen Sie das Programm aus, um das Ergebnis zu sehen.



Öffnen Sie Projekt->Optionen... erneut. Setzen Sie Häkchen bei einer Reihe anderer Stile. Und drücken Sie erneut auf Save.

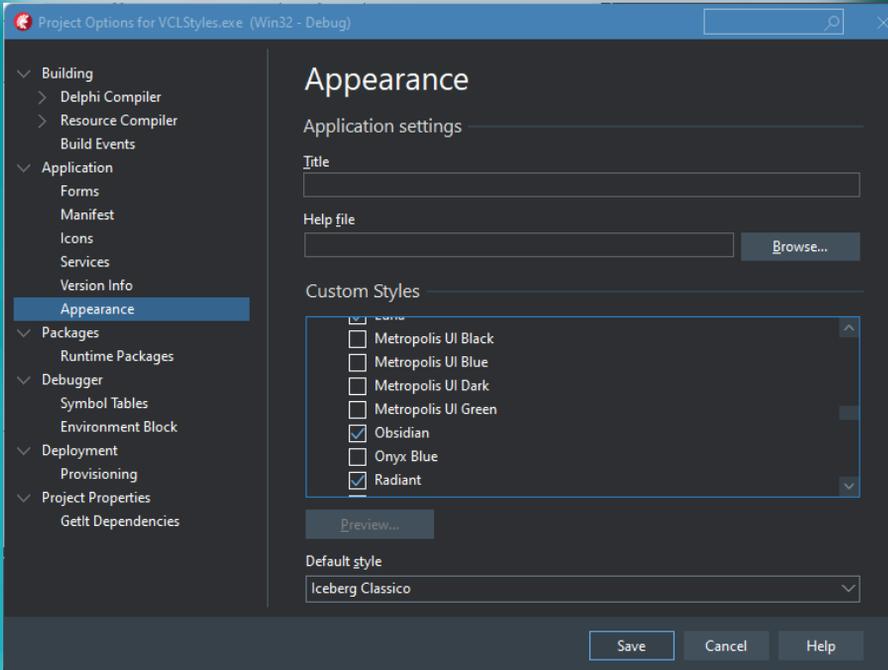


Abbildung 4: Project Options Dialogue mit verschiedenen styles gewählt

Fügen Sie unter **Private** in den Abschnitt **Interface** ein, so dass die `form` class wie folgt aussieht:

```

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    CheckBox1: TCheckBox;
    RadioButton1: TRadioButton;
    Button1: TButton;
    ListBox1: TListBox;
    Button2: TButton;
    BitBtn1: TBitBtn;
    StringGrid1: TStringGrid;
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
  procedure StylesListRefresh;
public
  { Public declarations }
end;

```

Drücken Sie Strg+Umschalt+c, um das Skelett der **Procedure** zu erstellen. Gehen Sie dann zum **OnCreate** des Formulars und füllen Sie es wie folgt aus und vervollständigen Sie die **Procedure StylesListRefresh**:

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  StylesListRefresh;
end;

procedure TForm1.StylesListRefresh;
var
  stylename: string;
begin
  ListBox1.Clear;

  // retrieve all the styles linked in the executable
  for stylename in TStyleManager.StyleNames do
  begin
    ListBox1.Items.Add(stylename);
  end;
end;
end;

```



Klicken Sie auf **Button2** und ändern Sie im **Objektinspektor** die Beschriftung in **Style** wählen und den Namen in **btnKies**. Doppelklicken Sie auf den **Button**, um das Klick-Ereignis zu erstellen und füllen Sie es wie folgt aus:

```
procedure TForm1.btnKiesClick(Sender: TObject);
begin
  TStyleManager.SetStyle(ListBox1.Items[ListBox1.ItemIndex]);
end;
```

Run the program for the result:

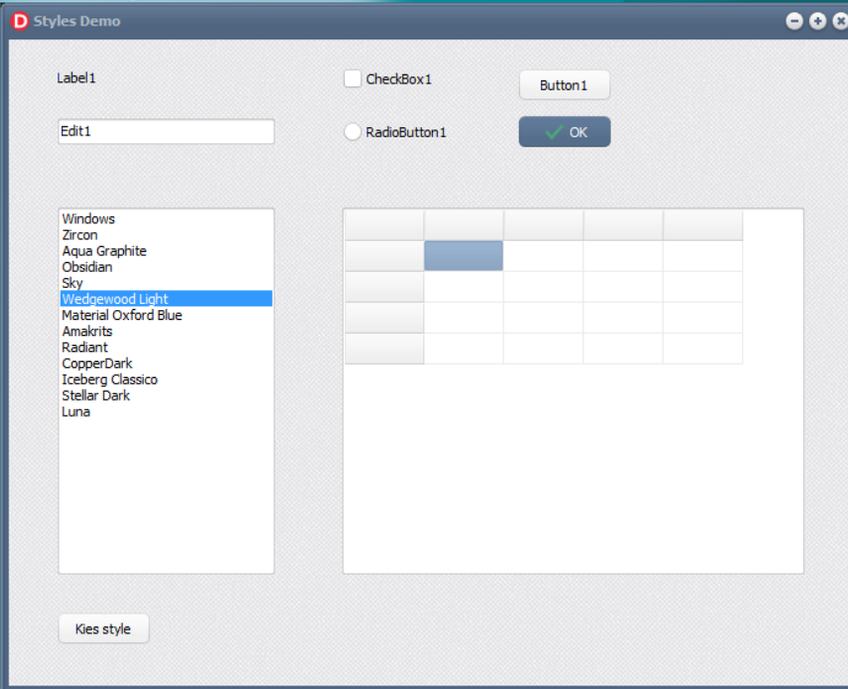
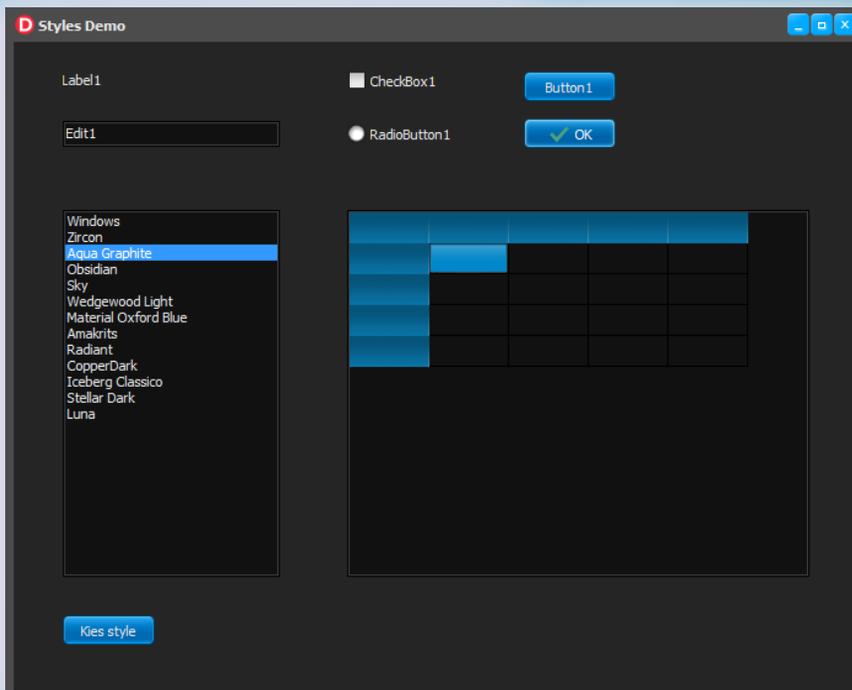


Figure 5: Form with Wedgewood Light chosen as the style.

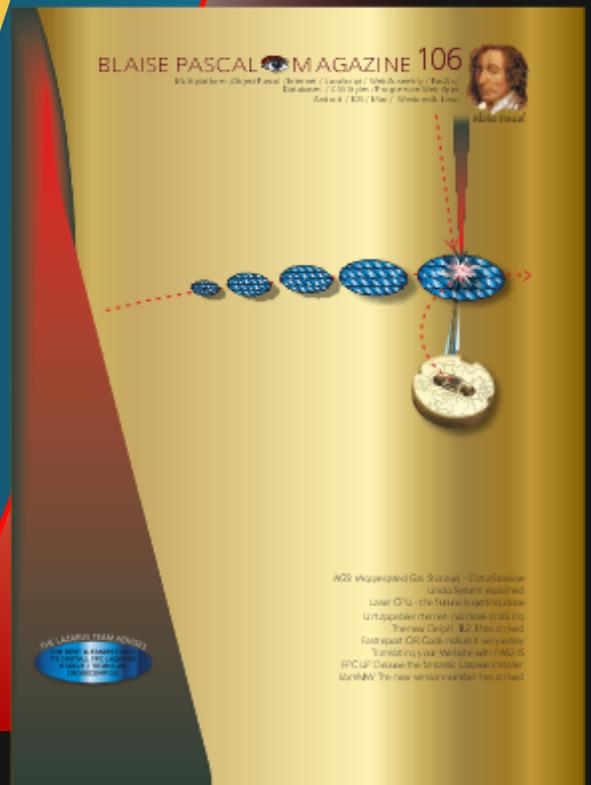


So einfach funktionieren die styles bei Programme der VCL.

LAZARUS HANDBOOK (PDF) +ABONNEMENT 1 JAHR

- English
- Printed black & white
- 2 Volumes
- PDF included
- 934 Pages
- Weight: 2kg
- Extra protected
- Including 40 Example projects and extra programs

**SPECIAL
OFFER
€ 75**



MÖGLICHKEIT, AKTIVEN UND INAKTIVEN CODE BASIEREND AUF BEDINGTEN DEFINITIONEN (CONDITIONAL DEFINES) ZU SEHEN



Starter Expert

EINLEITUNG

In Delphi wurde eine nette Benutzerfreundlichkeitsfunktion hinzugefügt, nämlich die Möglichkeit, aktiven und inaktiven Code auf der Grundlage von Conditional Defines zu sehen, Sie werden in diesem Projekt als Define hinzugefügt. Hier sehen Sie, wie Sie den Debug- und den Run modus verwenden können.

Wir fügen hier ein Define `{$IFDEF DEBUG}` und else `{$ELSE}` gefolgt von einem `{$ENDIF}` ein, damit wir den Unterschied zwischen ein und aus sehen können. (Siehe Abbildung 1)

Führen Sie jedoch zunächst diese Schritte aus:

Setzen Sie die `self.caption` auf zu 'Hello Debug-World in Blaise Pascal Magazin.'

Jetzt können Sie sehen, dass wir eine Build-Config von Debug zu diesem Zeitpunkt: dies ist aktiv, wie Sie in Abbildung 2 sehen können

Wir haben immer noch die volle Funktionsfähigkeit aber wir können sofort sehen dass dieser Code mit der Konfiguration, die wir haben, im Moment nicht aktiv wird. (Siehe Abbildung 3)

```

Welcome Page  Unit2
- procedure TForm2.FormCreate(Sender: TObject);
  begin
  . {$IfDef Debug}
  . {$Else }
  . {$Endif }
30 end;
  .
  . end.
33
    
```

Abbildung: 1

```

- procedure TForm2.FormCreate(Sender: TObject);
  begin
  . {$IfDef Debug}
  . Self.Caption := 'Hello Debug world in Blaise Pascal Magazine';
  . {$Else }
  . {$Endif }
30 end;
  .
  . end.
34
    
```

Abbildung: 2

```

TForm2
- procedure TForm2.FormCreate(Sender: TObject);
  begin
  . {$IfDef Debug}
  . Self.Caption := 'Hello Debug world in Blaise Pascal Magazine';
  . {$Else }
  . Self.Caption
  . {$Endif }
30 end;
  .
  . end.
    
```

property Caption: TCaption;

procedure UpdateDesignerCaption (AVisible: Boolean; AUpdateFrame: ...)

Abbildung: 3



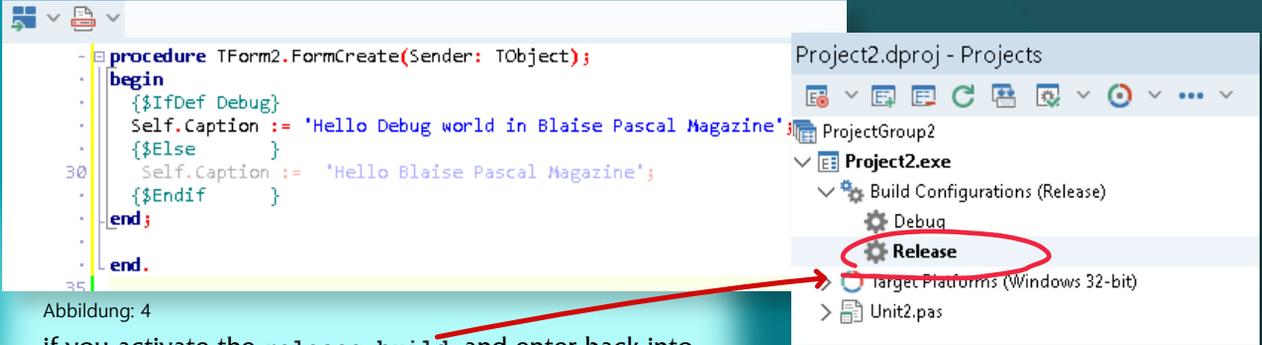


Abbildung: 4

if you activate the **release build** and enter back into the IDE, we can see that it starts editing. (See figure 4)

Abbildung: 6

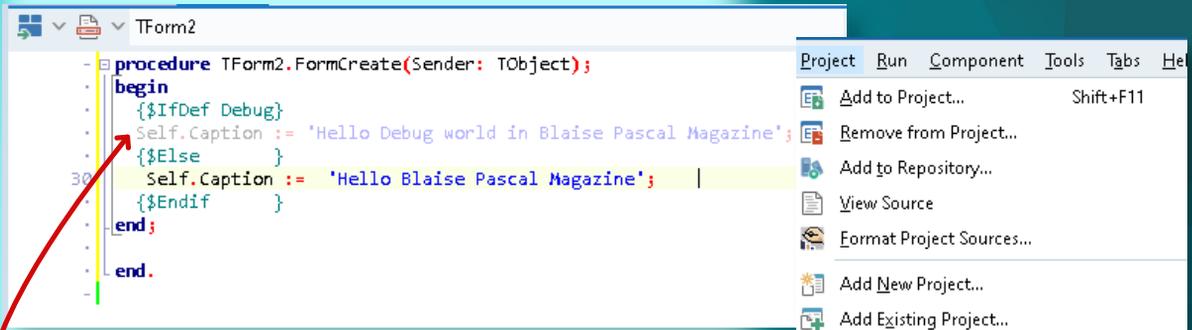


Abbildung: 5

As soon as we start editing, it starts updating to the **current definitions** and shows the altering in the rendering to grey.

This also works with your own **conditional defines**: Open the **project options**, (See Figure 7) under the compiler find the conditional defines (See Figure 7 and 8 (on the next page))

Abbildung: 7



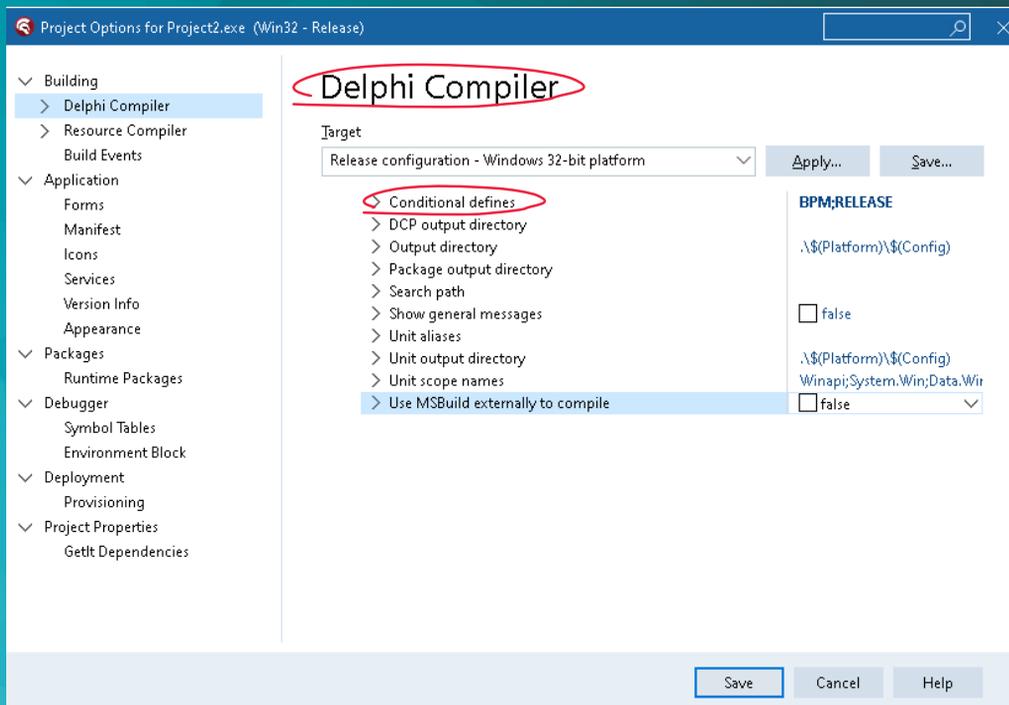
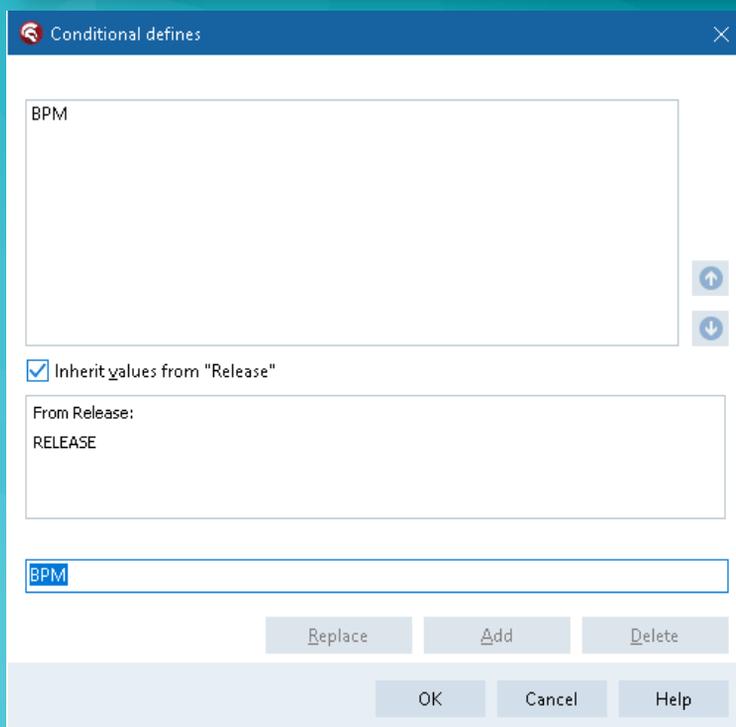


Abbildung 6:



Choose Conditional defines
It will be shown as
BPM;RELEASE (See figure 8)
Don't forget to save it...

Let's add one in for BPM and we
can now change this to BPM and
we can see that it updates to
hello BPM

Abbildung: 9



```
TFForm2
- procedure TForm2.FormCreate(Sender: TObject);
- begin
-   {$IFDEF BPM}
27   Self.Caption := 'Hello Debug world in BPM';
-   {$Else}
30   Self.Caption := 'Hello BPM';
-   {$EndIf}
- end;
- end.
```

Wenn Sie in BPM aktiv sind, wird dies bei der Aktualisierung sofort angezeigt: **Hello BPM aktiv ist** (Siehe Abbildung 10)

Abbildung: 10

```
TFForm2
- procedure TForm2.FormCreate(Sender: TObject);
- begin
-   {$IFDEF BPM}
27   Self.Caption := 'Hello Debug world in BPM';
-   {$Else}
30   Self.Caption := 'Hello BPM';
-   {$EndIf}
- end;
- end.
```

Wenn Debug in BPM aktiv ist, wird dies sofort während der Aktualisierung angezeigt: **Hello BPM ist grau** (Siehe Abbildung 11)

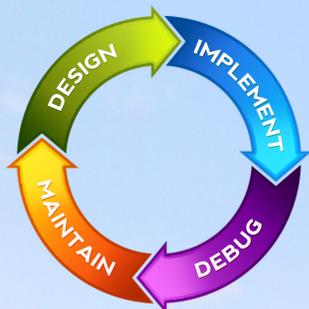
Abbildung: 11

- ADVERTISEMENT -

Introducing

Database Workbench 6

database development environment



Consistent user interface, modern code editors, Unicode enabled, HighDPI aware, ER designer, reverse engineering, meta data browsing, visual object editors, meta data migration, meta data compare, stored routine debugging, SQL plan visualizer, test data generator, meta data printing, data import and export, data pump, Grant Manager, DBA tasks, code snippets, SQL Insight, built in VCS, report editor, database meta data search, numerous productivity tools and much more...

for SQL Server, Oracle, MySQL, MariaDB, Firebird, InterBase, NexusDB and PostgreSQL



Database tools for developers³⁸

www.upszene.com

SUBSCRIPTION FOR 2 YEAR OF BLAISE PASCAL MAGAZINE

- English or German
- PDF
- 10 Issues per year
- minimal 60 pages
- Including example projects and code

€ 120
PER 2 YEAR

BLAISE PASCAL  MAGAZINE 107/108

Multi platform / Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js

Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



Blaise Pascal



Penguin Data Science: 4 engines to render Vector graphics
FRESNEL the graphical project for Lazarus, creating colours everywhere
PANTHER, the super fast, in Lazarus created PDF indexer for all OS's explained
The Pendulum, time acts like a wave
Using styles with a VCL program
Lazarus programme using BufDataset
Cross compiling with FPC Up De Luxe
Accelerating the Delphi Compiler
Usability feature added in DELPHI
multi-node quantum network - Quantum Internet News
What's open about smart chatbot maker OPENAI?
Discussion about OPEN GPT
How to create a BufDataset Master Detail
Using Styles in Delphi
Installing FastReport packages in Lazarus for Linux/Windows
Zeos database components for Lazarus



Starter

Expert

ABSTRACT

In einem früheren Beitrag haben wir gezeigt, wie man ein PDF in einem PAS2JS-Programm anzeigt. Wir haben auch gezeigt, wie Sie die PDF-Datei durchsuchen können. In diesem Artikel zeigen wir Ihnen, wie Sie die Suche in einer Reihe von PDF-Dateien vorbereiten, indem Sie sie indizieren.

EINLEITUNG



In einem früheren Artikel (*BPM Ausgabe Nr. 105*) haben wir gezeigt, wie man eine PDF-Datei in einem PAS2JS-Programm anzeigt. Das Programm ermöglichte die Anzeige der PDF-Datei und die Suche nach dem Inhalt der PDF-Datei. Was aber, wenn Sie viele PDF-Dateien haben - zum Beispiel alle Ausgaben des Blaise Pascal Magazins - und diese durchsuchen möchten? Das Programm, wie es entwickelt wurde, kann nur eine PDF-Datei durchsuchen, die in den Speicher des Browsers geladen wurde. Das ist bei einer Bibliothek mit vielen Dateien natürlich nicht sehr effizient.

Hierfür ist ein anderer Ansatz erforderlich. Die PDF-Dateien müssen indiziert werden:

Erstellen Sie eine Datenbank der Wörter in der PDF-Datei und erfassen Sie die Seiten, auf denen jedes Wort vorkommt. Durch Querverweise zwischen den Seiten und den Artikeln erhalten wir eine Liste der Artikel, die ein bestimmtes Wort im Text enthalten.

Bei der Suche, wird der Suchbegriff in der Liste der Wörter nachgeschlagen und eine Liste der Fundstellen (*oder Artikel*)

kann abgerufen werden. Dazu müssen wir die PDF-Datei lesen und die Wörter extrahieren sowie eine Datenbank mit den Wörtern

erstellen. Free Pascal und Lazarus enthalten alle Werkzeuge, die Sie für diese Aufgabe benötigen: Die Datenbank-Suchwerkzeuge gibt es schon seit langem und werden in der Free Pascal Dokumentation unter:

<https://www.freepascal.org/docsearch/docsearch.var>

Kürzlich wurde in Free Pascal eine Reihe von Units zum Lesen einer PDF-Datei entwickelt. Die Kombination beider Systeme ermöglicht es uns, eine Datenbank mit Wörtern in PDF-Dateien zu erstellen und diese zu durchsuchen. Da wir eine Datei erst lesen müssen, bevor wir die Wörter daraus extrahieren können, beginnen wir mit der Frage, wie man eine PDF-Datei liest.

2. ZERLEGEN EINER PDF-DATEI

Die PDF-Datei ist allgegenwärtig. Sie ist der De-facto-Standard für den Versand von Dokumenten über das Internet. Das PDF-Format wurde von Adobe entwickelt und hat eine gewisse Ähnlichkeit mit dem älteren Postscript-Format - ebenfalls von Adobe. Postscript war ein text basiertes Format, und bis zu einem gewissen Grad ist dies auch das PDF-Format.

Aus Gründen der Effizienz und zur Verringerung der Größe der PDF-Dateien wurden jedoch binäre Daten in das Format aufgenommen. Das Format ermöglicht es auch, neue Daten an die Datei anzuhängen, ohne die vorherigen Daten in der Datei zu verändern: neue Inhalte können einfach an die Datei angehängt werden. Die Spezifikation des PDF-Formats wurde mehrmals überarbeitet, wobei jedes Mal weitere Funktionen hinzugefügt wurden. Glücklicherweise ist

das Format so konzipiert, dass eine Anwendung nicht alle Funktionen kennen muss, um eine PDF-Datei zu lesen.





Im Kern ist das PDF-Format einfach eine Reihe von Objekten (*genannt 'Indirekte Objekte'*) mit Wörterbüchern zur Beschreibung der Eigenschaften der Objekte beschreiben. Um die von Ihnen benötigten Objekte zu finden, wird ein Abschnitt mit Querverweisen angehängt, der die Positionen aller Objekte in der Datei enthält, zusammen mit einem kleinen Wörterbuch, um einige Schlüsselobjekte zu finden.

In einer einfachen PDF-Datei lässt sich all dies mit Textbefehlen bewerkstelligen.

Die indirekten Objekte und die Querverweistabelle können jedoch in binären Streams kodiert werden, die auf verschiedene Weise komprimiert und optional auch kodiert werden können - ein Mechanismus, der im PDF-Jargon als "Filter" bezeichnet wird.

Wenn Inhalte an die Datei angehängt werden, benötigt der neu hinzugefügte Inhalt seinen eigenen Querverweisabschnitt, der natürlich auf den vorherigen Querverweisabschnitt für den vorhandenen Inhalt verweisen muss. Dieser Vorgang lässt sich beliebig oft wiederholen: Jedes Mal, wenn Sie eine Datei öffnen und Anmerkungen zu einer Datei hinzufügen, werden die Anmerkungen über diesen Mechanismus an die bestehende Datei angehängt. *(Dies hängt natürlich von der Software ab, die sich auch dafür entscheiden kann, die Datei komplett neu zu schreiben...)*

All dies macht das Lesen einer PDF-Datei nicht gerade einfach: verschiedene Formen der Dekomprimierung müssen unterstützt werden, optional auch die Verschlüsselung. Wenn Sie eine PDF-Datei lesen, werden Sie mit verschiedenen Arten von Objekten in der Datei konfrontiert.

Diese Objekte wurden in einer Einheit namens `fpdfobjects` definiert

■ **TPDFXRef**

ein Querverweis auf eine indirekte Referenz. Dies wird verwendet, um auf indirekte Objekte zu verweisen, und wird an vielen Stellen verwendet: eine Seitenliste ist beispielsweise einfach eine Liste von `TPDFXRef`-Instanzen, die auf die zugrunde liegenden Seiten (*oder ein anderes `PageListObject`*) verweisen

■ **TPDFDictionary**

Dieses Objekt kapselt ein PDF-Wörterbuch ein.

■ **TPDFIndirect**

Dies ist der Grundbaustein der PDF-Datei: ein indirektes Objekt. Es hat eine `ObjectID`, eine Nummer, die dieses Objekt eindeutig identifiziert. An sich nicht sehr interessant, aber jede Art von Objekt in der PDF-Datei ist ein Abkömmling dieser Klasse. Jedes `TPDFIndirect-Object` hat normalerweise ein Wörterbuch, das mit ihm verbunden ist (*verfügbar in der Eigenschaft `ObjectDict`*).

■ **TPDFDocumentInfo**

Dieses Objekt enthält einige Meta-Informationen über das Dokument: Autor, Titel usw. Es handelt sich im Grunde um ein Wörterbuchobjekt.

■ **TPDFFontObject**

Dieses Objekt beschreibt eine Schriftart. Die Schriftart kann in die Datei eingebettet sein oder auch nicht, normalerweise in komprimierter Form.

■ **TPDFPageObject**

Dieses Objekt beschreibt eine Seite in einem PDF-Dokument. Es enthält eine Liste der Ressourcen, die zum Zeichnen der Seite benötigt werden: Schriftarten und Bilder. Es enthält auch den Content Stream: Der Content Stream (*der auf verschiedene zugrunde liegende Streams verteilt sein kann*) ist die Menge der Befehle, die zum Zeichnen der Seite benötigt werden. Die Eigenschaft `CommandList` des `PageObjects` enthält die eigentlichen Befehle, aus denen die Seite besteht.

■ **TPDFPagesObject**

Dieses Objekt beschreibt eine Liste von Seiten in einem PDF-Dokument: Die Seiten sind in einem Baum organisiert, dem so genannten 'Page Tree'.

■ **TPDFCommand**

Dieses Objekt stellt einen Zeichenbefehl dar: Eine Seite besteht aus Befehlen, die - alle zusammengenommen - die Seite vollständig wiedergeben.

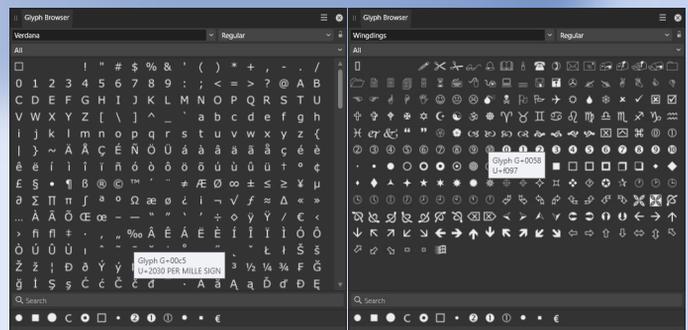
■ **TPDFCMap**

Dieses Objekt ordnet Zeichencodes den Glyphen* in einer Schriftart zu. Oder auf Unicode-Zeichen, wie wir weiter unten sehen werden.

■ **TPDFCMapData**

Diese Klasse enthält die eigentlichen Daten eines CMap-Objekts.

Glyph Vorbilder



All diese Objekte gehören zu einer TPDFDocument-Klasse, die das komplette PDF-Dokument darstellt. Sie hat die folgenden Eigenschaften:

PDFVersion	Die PDF-Version in der Kopfzeile
StartXref	Die Wurzel der Querverweistabelle.
TrailerDict	Trailer-Wörterbuch, das beim Parsen festgelegt wird
PageCount	Anzahl der Seiten in diesem Dokument
PageNodes	indirekte Objekte der obersten Ebene, die eine <code>page tree</code> node darstellen. 0-basiert. Dies kann ein anderer <code>page tree</code> oder ein Page Object sein.
Page	indizierter Zugriff auf eine Seite durch 0-basierte Seitennummer.
Pages	ein Enumerator für die Seiten
XRefCount	Anzahl der Elemente in der Eigenschaft XRefs.
XRefs	Indizierter Zugriff auf alle globalen XRefs (<i>Querverweise</i>) in der PDF-Datei.

Es gibt auch einige Methoden, um Objekte in der PDF-Datei zu finden:

FindInDirectObject	Suche nach einem indirekten Objekt anhand der Objekt-ID.
FindFont	Sucht ein Schriftobjekt anhand der Objekt-ID. Kann Nil zurückgeben.
GetFont	Ähnlich wie FindFont, löst aber eine Exception aus, wenn das Objekt nicht gefunden wird.
FindDocumentInfoObject	Findet das Dokumentinformationsobjekt. Kann Nil zurückgeben.
FindDocumentInfo	Findet Dokumentinformationen, aufgelöst in ein <code>TPDFDocumentInfo</code> , kann aber auch Nil sein. Sie müssen dieses Objekt selbst freigeben.
GetDocumentInfo	Ermittelt das Dokumentinformationsobjekt. Löst eine Ausnahme aus, wenn es nicht gefunden wird. Sie müssen dieses Objekt selbst freigeben.
FindCatalog	Findet das Objekt für den Dokumentenkatalog. Kann Nil zurückgeben.
GetCatalog	Ähnlich wie FindCatalog, löst aber eine Exception aus, wenn es nicht gefunden wird.
FindPages	Findet das Seitenobjekt der obersten Ebene. Gibt Null zurück, wenn keines gefunden wurde.
GetPages	Ähnlich wie FindPages, löst aber eine Exception aus, wenn es nicht gefunden wird.

Um ein Dokument aus einer Datei zu lesen, wird ein PDF-Parser benötigt. Dieses Objekt liest die PDF-Datei tatsächlich ein. Im Folgenden finden Sie eine (verkürzte) Deklaration des PDF-Parser-Objekts:

```
TPDFParser = class
  Constructor Create(aFile : TStream; aBufferSize : Cardinal = PDFDefaultBufferSize); virtual
  Procedure ParseDocument(aDoc : TPDFDocument); virtual;
  Procedure ResolveToUnicodeCMaps(aDoc : TPDFDocument);
  Property LoadObjects : Boolean;
  Property ResolveObjects : Boolean;
  Property ResolveContentStreams : Boolean;
  Property OnUnknownFilter : TPDFFilterEvent;
  Property OnLog : TPDFLogNotifyEvent;
  Property OnProgress : TPDFProgressEvent;
end
```

Der Konstruktor empfängt einen Stream mit dem Inhalt des PDF-Dokuments. Die Methode `ParseDocument` liest und parst das PDF-Dokument. Der PDF-Parser kann in mehreren Modi funktionieren:

- er kann das PDF-Dokument einfach lesen und die Querverweistabelle zur Verfügung stellen.
- Zusätzlich kann er auch alle indirekten Objekte lesen und sie als einfache `TPDFIndirect`-Instanzen.
- Wenn die indirekten Objekte gelesen werden, kann es sie auch in Objekte des richtigen Typs umwandeln: ein `TPDFFontObject` für eine Schriftart, ein `TPDFPageObject` für eine Seite, und so weiter.
- Wenn die Objekte in typisierte Objekte umgewandelt werden, kann der Parser auch den Content Stream eines Page Object interpretieren und die `CommandList`-Eigenschaft der verschiedenen `TPDFPageObject`-Instanzen füllen.
- Wenn die Objekte in typisierte Objekte umgewandelt werden, kann der Parser auch die `ToUnicode` CMap-Streams interpretieren, die mit Schriftarten verbunden sind.





Wie Sie sehen können, bedeutet jeder Schritt eine zusätzliche Verarbeitung der PDF-Daten. Die verschiedenen Schritte werden durch einige boolesche Eigenschaften gesteuert:

- **LoadObjects** lädt alle Objekte, wenn XRef geparkt wird. Standardmäßig True.
- **ResolveObjects** Löst beim Laden von Objekten die Objekte in ihre tatsächliche Klasse auf. Standardwert True.
- **ResolveContentStreams** Lösen Sie Content-Streams von Seiten in Befehle auf. Die Voreinstellung ist True.
- **ResolveToUnicodeCMaps** Sofortiges Parsen von Font ToUnicode CMap Streams. Standardmäßig False.

Es sind einige Ereignisse verfügbar:

- **OnUnknownFilter** wenn ein unbekannter Filter angetroffen wird, können Sie den Filter manuell mit diesem Ereignis behandeln.
- **OnLog** ein Callback, an den Protokollnachrichten gesendet werden.
- **OnProgress** ein Fortschrittsereignis.

Um eine PDF-Datei zu lesen, ist der folgende Code ausreichend:

```

{$mode objfpc}
{$h+}
uses fpdfobjects, fpdfparser, sysutils, classes;

procedure ReadPDF(const aStream: TStream; aDoc: TPDFDocument);
var aParser : TPDFParser;

begin
  aParser:=TPDFParser.Create(aStream);
  try
    aParser.ResolveToUnicodeCMaps:=True;
    aParser.ParseDocument(aDoc);
  finally
    aParser.Free;
  end;
end;

var
  F: TFileStream;
  Doc: TPDFDocument;
begin
  F:=TFileStream.Create('mydocument.pdf',fmCreate or fmShareDenyNone);
  try
    Doc:=TPDFDocument.Create; ReadPDF(F,Doc);
    // Do something with your PDF.
  finally
    doc.free;
    f.free;
  end;
end.
```

Oder, noch einfacher: die Unit `fpdfparser` hat eine Typhilfe für die Klasse `TPDFDocument` mit den Methoden `LoadFromFile` und `LoadFromStream`:

```

{$mode objfpc}
{$h+}
uses fpdfobjects, fpdfparser;

var Doc: TPDFDocument;

begin
  Doc:=TPDFDocument.Create;
  try
    Doc.LoadFromFile('mydocument.pdf');
    // Do something with your PDF.
  finally
    doc.free;
  end;
end.
```

Das ist so einfach, wie es nur sein kann. Die Methoden `LoadFromFile` und `LoadFromStream` haben eine überladene Variante, mit der die verschiedenen booleschen Eigenschaften des Parsers festgelegt werden können. Die Ereignisse `OnLog` und `OnProgress` können optional ebenfalls angegeben werden.

Das Beispielprogramm `pdfdump` demonstriert die Verwendung der verschiedenen Objekte, indem es Ihnen ermöglicht, bestimmte Informationen aus einer PDF-Datei auszugeben.





3 EXTRAHIEREN VON TEXT AUS EINER PDF-DATEI

Da wir nun in der Lage sind, eine PDF-Datei zu lesen, können wir damit beginnen, Wörter aus ihr zu extrahieren. Leider ist das nicht so einfach, wie es sich anhört. PDF ist ein Format, das geeignet ist, eine Seite unter allen Umständen genau gleich zu reproduzieren:

Sie enthält buchstäblich Anweisungen zum Zeichnen der Seite. Ein PDF-Leseprogramm liest die Anweisungen und führt sie aus, indem es sie beispielsweise auf dem Bildschirm zeichnet oder entsprechende Befehle an den Drucker sendet.

Wenn Text Teil einer Seite ist, dann enthält die PDF-Datei einfach Anweisungen zum Zeichnen von Bildern, die die Buchstaben (*Glyphen*) an bestimmten Stellen auf der Seite darstellen. Die Glyphen sind Teil einer Schriftart, und die PDF-Datei sagt im Wesentlichen so etwas wie 'zeichne mir die Glyphen Nummer 12 23 45 an der Position (145,175) auf der Seite'. Traditionell (und um es etwas zu vereinfachen, ignorieren wir die Codepages für eine Sekunde) waren die Glyphnummern einfach der Bytecode des Zeichens, das sie repräsentierten:

Die Glyphe für den Buchstaben 'A' war einfach die Glyphe 65 (*der ASCII-Code für den Buchstaben A*). So konnte ein Text "Hello World" leicht auf Glyphnummern in einer Schriftart abgebildet werden und umgekehrt.

Mit dem Aufkommen der Unicode-Kodierung von Zeichen und Schriftarten, die nur eine Teilmenge aller Zeichen enthalten, wurde diese Zuordnung schwieriger. Da das PDF-Format keine Vorstellung von Text hat und einfach nur Glyphnummern verwendet, hat Adobe eine Lösung gefunden: Eine Schriftart kann eine ToUnicode-Zuordnung angeben: Dies ist eine Zuordnung von Glyphnummern zu Unicode-Zeichennummern.

Einfach ausgedrückt, handelt es sich um eine Zuordnung, die es einer Anwendung ermöglicht, die Glyphnummern einer Schriftart auf Unicode-Zeichen zurückzuführen. Die Zuordnung erfolgt im CMap-Format, und ein Font-Wörterbuch in einer PDF-Datei kann einen Eintrag für eine solche Zuordnung enthalten, indem der ToUnicode-Schlüssel des Wörterbuchs verwendet wird.

Dieser Mechanismus ermöglicht es PDF-Anzeigesoftware wie Acrobat Reader  einen Textauswahlmechanismus und einen Suchmechanismus anzubieten: Bei der Auswahl des Textes werden die ausgewählten Glyphen in Buchstaben zurückübersetzt. Bei der Suche werden alle Befehle zur Textanzeige nach den angezeigten Glyphen durchsucht, die Glyphen werden mithilfe der ToUnicode-Map in Unicode-Text übersetzt und dieser Text wird dann durchsucht. Genau das müssen wir auch tun, wenn wir die PDF-Datei indizieren wollen.

Wie bereits erwähnt, besteht der Inhalt einer Seite aus einer Reihe von Zeichenbefehlen. Alle Zeichenbefehle sind ein Abkömmling von `TPDFCommand`. Einige dieser Befehle zeichnen einen Text (z.B. die Tj- und TJ-Operatoren), sie sind alle Nachkommen einer abstrakten `TPDFTextCommand`-Klasse, die wie folgt definiert ist:

```
TPDFTextCommand = Class(TPDFCommand)
Public
    Function GetFullText(aUnicodeMap : TPDFCMap) : RawByteString;
    Function GetFullText : RawByteString; virtual; abstract;
end;
```

Die Funktion `GetFullText` gibt alle Glyphen für den zu zeichnenden Text zurück. Bei einfachen Schriftarten werden diese direkt den Zeichen zugeordnet und können so verwendet werden, wie sie sind. Für komplizierte Schriftarten ist eine Zuordnung von Glyphen-IDs zu Unicode-Zeichen erforderlich: Das Argument `aUnicodeMap` für `GetFullText`. Es gibt mehrere Abkömmlinge dieser Klasse, die diese 2 Methoden implementieren.

Ein Textzeichenbefehl zeichnet einfach einen Text unter Verwendung der aktuellen Schriftart. Wenn wir wissen wollen, welche Unicode-Map wir verwenden müssen, müssen wir auch den Befehl zur Auswahl der Schriftart prüfen, der dem Textzeichenbefehl vorausging: Er enthält einen Schriftartnamen. Der Schriftname kann mithilfe des Seitenressourcen-Wörterbuchs auf ein Schriftobjekt abgebildet werden (ein indirektes Objekt) zugeordnet werden. Wenn dieser Schrift eine ToUnicode-Map zugeordnet ist, verweist diese auf die Unicode-Map, die für alle auf den Schriftauswahlbefehl folgenden Textzeichenbefehle zu verwenden ist. Der Schriftauswahloperator (`Tf`) wird durch die Klasse `TPDFtfCommand` dargestellt, die wie folgt definiert ist:

```
TPDFtfCommand = class(TPDFCommand)
    property FontName : String Read GetFontName;
    property FontSize : Integer Read GetFontSize;
end;
```

Mit diesen Klassen können wir nun einen Algorithmus zum Extrahieren des Textes aus einer PDF-Seite implementieren. Um diesen Algorithmus zu zeigen, erstellen wir eine kleine Demo, die ein PDF lädt und den Text einer Seite extrahiert und in einem Memo anzeigt.





Hierfür entwerfen wir ein Formular mit 5 Elementen:

- **FEPDF** Eine Bearbeitung des Dateinamens zur Auswahl einer PDF-Datei.
- **edtPageNo** Eine Eingabe, um eine Seitenzahl auszuwählen.
- **lblPageCount** Ein Etikett zur Anzeige der Seitenzahl der geladenen Datei.
- **btnShow** Ein Button zum Starten der Aktion: Laden einer Datei und Extrahieren des Text der ausgewählten Seite.
- **mText** Ein Memo zur Anzeige des Textes der ausgewählten Seite.

Der Code für diese Anwendung ist recht einfach. Der Prozess wird mit dem OnClick-Ereignishandler von btnSelect gestartet: Wenn die PDF-Datei noch nicht geladen war oder eine andere Datei ausgewählt wurde, wird die Datei zunächst mit LoadPDF geladen:

```
procedure TMainForm.btnShowClick(Sender: TObject);
Var aPage : Integer;
begin
  if (FDoc=nil) or (FEPDF.FileName<>FloadedFile) then
    LoadPDF(FEPDF.FileName);
  aPage:=StrToIntDef(edtPageNo.Text,0);
  if (aPage>0) and (aPage<FDoc.PageCount) then
    ShowPageText(aPage)
  else
    ShowMessage('Invalid page, valid values 1-'+IntToStr(FDoc.PageCount));
end;
```

Wenn die Datei geladen wird, wird die Seitenzahl geprüft und wenn sie gültig ist, wird der Seitentext mit ShowPageText angezeigt. Wenn sie ungültig ist, wird eine Fehlermeldung angezeigt.

Die Funktion LoadPDF unterscheidet sich nicht wesentlich von dem, was wir zuvor gezeigt haben:

```
procedure TMainForm.LoadPDF(const aFileName : string);
begin
  FreeAndNil(FDoc);
  FDoc:=TPDFDocument.Create;
  Try
    FDoc.LoadFromFile(aFilename);
    FloadedFile:=aFileName;
    lblPageCount.Caption:='/'+IntToStr(FDoc.PageCount);
  Except
    on E : Exception do
      ShowMessage('Error loading PDF file :'+E.Message);
  end;
end;
```

In der ShowPageText-Schleife passiert die eigentliche Arbeit. Es handelt sich eigentlich um eine einfache Schleife mit 2 Fällen: Wenn ein Schriftauswahlbefehl auftritt, wird die CMAP für die ToUnicode-Zuordnung der Schrift extrahiert und gespeichert.

```
procedure TMainForm.ShowPageText(aPageNo : Integer);
Var aPage : TPDFPageObject;
    Cmd : TPDFCommand;
    FontName, Rawtext : RawByteString;
    aFontRef : TPDFRefData;
    UnicodeMap : TPDFCMap;
    aFontObj : TPDFFontObject;

begin
  mText.Clear;
  aPage:=FDoc.Page[aPageNo];
  for Cmd in aPage.CommandList do begin
    if Cmd is TPDFTfCommand
    then
      begin
        FontName:=TPDFTfCommand(Cmd).FontName;
        if (FontName<>'') and (FontName[1]='/') then
          Delete(FontName,1,1);
        aFontRef:=aPage.FindFontRef(FontName);
        aFontObj:=FDoc.FindFont(aFontRef); // TPDFFontObject
        if Assigned(aFontObj) then
          UnicodeMap:=aFontObj.UnicodeCMap
        Else UnicodeMap:=nil;
      end
    else
      if cmd is TPDFTextCommand then
        begin
          rawText:=TPDFTextCommand(Cmd).GetFullText(UnicodeMap);
          SetCodePage(RawText,CP_UTF8);
          mText.Lines.Add(RawText);
        end;
      end;
  end;
end;
```





Wenn Sie auf einen Textzeichenbefehl stoßen, wird die Methode `GetFullText` verwendet, um den vollständigen Text zu extrahieren. Wenn es eine aktive Unicodemap gab, wird diese zur Interpretation des Textes verwendet. Wenn es keine Unicodemap gab, wird davon ausgegangen, dass die Schriftart eine einfache Schriftart ist und der rohe Text lesbar sein sollte (dies ist nicht 100% korrekt, wie wir sehen werden). Die Codepage des Textes wird auf UTF8 gesetzt, da dies das ist, was das Lazarus LCL-Framework erwartet, und dann wird der Text dem Memo hinzugefügt.

Das Ergebnis können Sie in Abbildung 1 auf Seite 7 sehen.

Auf dem Screenshot sind 2 Dinge zu beachten:

- Der Seitentext enthält einige Zeichen, die nicht angezeigt werden können. Es handelt sich um die Buchstabenkombination `fi`, für die eine spezielle Glyphe verwendet wird, und die im Memo nicht angezeigt werden kann.
- Der Text enthält Zeilenumbrüche an unerwarteten Stellen. Dies ist eine Folge des einfachen Mechanismus den wir zum Extrahieren des Textes verwendet haben.

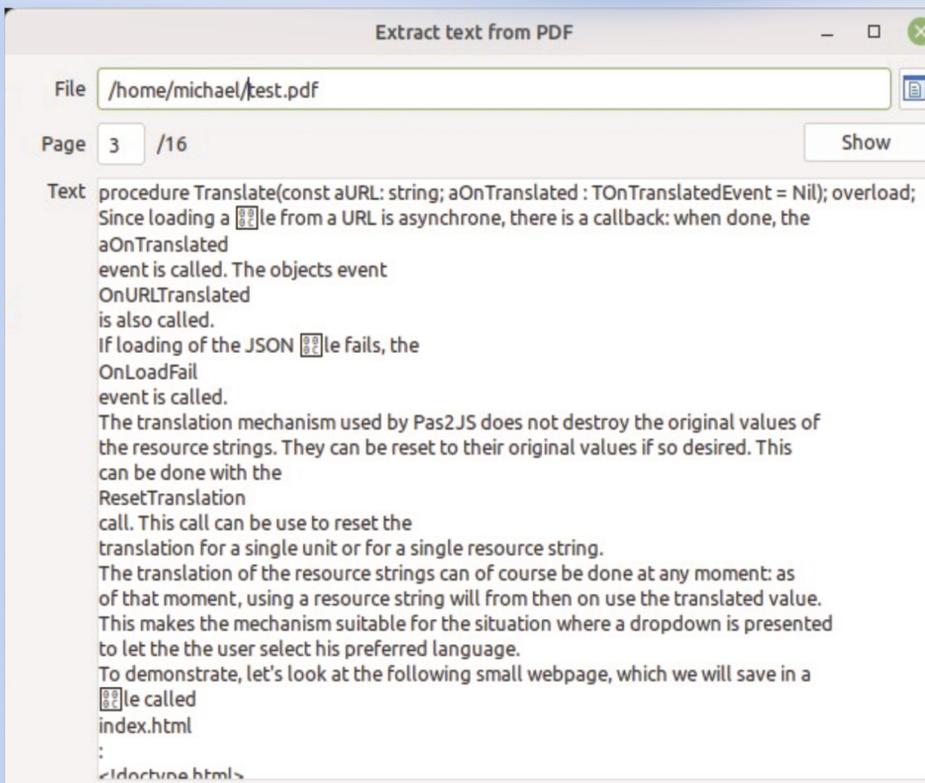


Abbildung 1: Extrahieren des Textes einer PDF-Datei

Immer wenn sich die Schriftart im angezeigten Text ändert (*eine andere Schriftart oder einfach dieselbe Schriftart fett oder kursiv gedruckt*), enthält die PDF-Datei eine Malanweisung für jeden Teil des Textes: den Text vor der Schriftartänderung, den Text in der anderen Schriftart und den Rest des Textes, der in der Originalschriftart oder einer anderen Schriftart sein kann.

Beachten Sie, dass es keine Garantie dafür gibt, dass die Anweisungen in der Reihenfolge erscheinen, in der die Wörter im Text erscheinen: Der Layout-Mechanismus könnte beschließen, zuerst alle Wörter in einer Schriftart zu zeichnen, dann die Wörter in einer anderen Schriftart, und so Befehle zur Auswahl der Schriftart zu sparen.

Um tatsächliche Textzeilen oder -absätze zu rekonstruieren, müsste auch die Positionierung des Textes untersucht werden: Man kann davon ausgehen, dass alle Textzeichenbefehle, die auf derselben Grundlinie liegen, eine einzige Textzeile bilden, obwohl es keine Garantie dafür gibt, dass dies tatsächlich so ist. Für die Zwecke der Extraktion von Wörtern aus einem Text ist der oben beschriebene Mechanismus ausreichend:

Ohne Berücksichtigung von Wörtern, für die eine andere Schriftart als die Hälfte eines Wortes gewählt wurde, oder von Wörtern mit Bindestrich, enthält ein Textzeichenbefehl vollständige Wörter.





4 EINE KLASSE ZUM INDIZIEREN VON WÖRTERN

Free Pascal wird mit einer Unit namens fpindexer geliefert. Diese enthält einige abstrakte Klassen, die als Basisklasse für einen Wortindizierungs- und Suchmechanismus dienen.

```
TFPIndexer = class(TComponent)
public
  constructor Create(AOwner: TComponent); override;
  destructor Destroy; override;
  function IndexStream
    (const AURL: UTF8String; ADateTime: TDateTime; S: TStream; Reader: TCustomFileReader);
  function IndexFile(AURL: UTF8String; AllowErrors: boolean; const ALanguage: UTF8String = '
  Execute(AllowErrors: boolean): int64;
  property ErrorCount: int64 read FErrorCount;

published
  property Language: UTF8String;
  property OnProgress: TIndexProgressEvent;
  property UseIgnoreList: boolean;
  property IgnoreNumeric: boolean;
  property CommitFiles: boolean;
  property Database: TCustomIndexDB;
  property ExcludeFileMask: UTF8String;
  property FileMask: UTF8String;
  property SearchPath: UTF8String;
  property SearchRecursive: boolean;
  property DetectLanguage: boolean;
  Property CodePage : TSystemCodePage;
  Property StripPath : String;
end;
```

Diese Klasse kann verwendet werden, um den Inhalt von Dateien zu indizieren: die `IndexStream` und `IndexFile` Methoden erstellen eine Datenbank mit Wörtern aus dem Inhalt einer Textdatei.

Mit den verschiedenen Eigenschaften können Sie die Suche nach Dateien auf der Festplatte und die Art der Indizierung steuern.

Die Bedeutung der meisten dieser Eigenschaften ist klar.

Nur '**CommitFiles**' ist erklärungsbedürftig:

Das Standardverhalten ist, alle Daten in einer großen Transaktion in der Datenbank zu speichern.

Wenn diese Eigenschaft auf True gesetzt ist, wird nach jeder indizierten Datei ein Commit durchgeführt.

Der Indexer verwendet eine Fabrik, um einen Tokenizer pro Dateityp (*bestimmt durch die Erweiterung*) zu erstellen:

Der Tokenizer gibt alle Wörter in einer Datei für einen bestimmten Dateityp zurück.

Sie können einen Handler für einen Dateityp registrieren und dieser Handler wird dann verwendet, um die Wörter zu lesen

und die Datenbank zu füllen. Die Aufgabe wäre es, einen Handler für PDF-Dateien zu erstellen.

Aus pädagogischen Gründen werden wir hier einen etwas anderen Ansatz wählen.

Die interessante Eigenschaft ist die Eigenschaft `Database`:

Sie bestimmt, wohin der Indexer die Wörter und Treffer schreibt.

Die Klasse `TCustomIndexDB` ist wie folgt definiert:

```
TCustomIndexDB = class(TComponent)
public
  procedure CreateDB; virtual; abstract;
  procedure Connect; virtual; abstract;
  procedure Disconnect; virtual;
  procedure CompactDB; virtual; abstract;
  procedure BeginTrans; virtual; abstract;
  procedure CommitTrans; virtual; abstract;
  procedure DeleteWordsFromFile(URL: UTF8String); virtual; abstract;
  procedure AddSearchData(ASearchData: TSearchWordData); virtual; abstract;
  procedure FindSearchData(SearchWord: TWordParser; FPSearch: TFPSearch;
    SearchOptions: TSearchOptions); virtual; abstract;
  Function GetAvailableWords(out aList : TUTF8StringArray; aContaining : UTF8String;
    Partial: TAvailableMatch): integer; virtual; abstract;
  procedure CreateIndexerTables; virtual; abstract;
end;
```





Most of these methods are self-explaining.

CreateDB	Erstellen Sie die Indexdatenbank.
CreateIndexerTables	Erstellen Sie die vom Indexer benötigten Tabellen.
Connect	Verbinden Sie sich mit der Indexdatenbank.
Disconnect	Trennen Sie die Verbindung zur Indexdatenbank.
CompactDB	Bereinigen Sie die DB.
BeginTrans	Starten Sie eine Transaktion (falls im Backend verfügbar).
CommitTrans	Bestätigt eine Transaktion (falls im Backend verfügbar).
DeleteWordsFromFile	Löscht alle Worttreffer aus einer Datei.
AddSearchData	Fügt eine Wortübereinstimmung hinzu.
FindSearchData	Findet Wortübereinstimmungen.
GetAvailableWords	Ruft eine Liste der verfügbaren Wörter ab, die einem Muster entsprechen. Das FPIIndexer-Paket enthält mehrere Abkömmlinge dieser Klasse:
TPGIndexDB	Ist ein Abkömmling, der die Daten in eine PostgreSQL-Datenbank schreibt. Implementiert in der Unit pgindexdb
TFBIndexDB	Ist ein Nachkomme, der die Daten in eine Firebird/Interbase-Datenbank schreibt. Implementiert in der Unit fbindexdb
TSQLiteIndexDB	Ist ein Abkömmling, der die Daten in eine SQLite-Datenbank schreibt. Implementiert in der Unit sqliteindexdb
TMemIndexDB	Ist ein Nachkomme, der die Daten im Speicher hält. Implementiert in der Unit memindexdb
TFileIndexDB	Ist ein Abkömmling, der die Daten im Speicher hält, aber zusätzlich die Daten in einer Datei speichern kann. die Daten in einer Datei speichern kann. Implementiert in der Einheit memindexdb

Anstatt einen Dateityp-Handler für PDF-Dateien zu erstellen, können wir einen der Nachkommen der Klasse **TCustomIndexDB** direkt verwenden, um eine Datenbank mit Suchbegriffen zu füllen. Wir werden dies in einer Klasse **TPDFIndexer** (implementiert in der Unit **fppdfindexer**) mit der folgenden Schnittstelle tun, die zur Indizierung einer einzelnen PDF-Datei verwendet werden kann.

```
TPDFIndexer = Class(TComponent)
Public
  Procedure Connect;
  Procedure CreateTable;
  Procedure Disconnect;
  Procedure IndexPDF(const aFileName : string); overload;
  Procedure IndexPDF(const aStream : TStream); overload;
  Property PDFURL : String;
  Property Language : String;
  Property Indexer : TSQLDBIndexDB;
  Property MinWordLength : Integer;
  Property IgnoreWords : TStrings;
  Property OnLog : TPDFIndexLogEvent;
end;
```

Die Methoden **Connect**, **CreateTables** und **Disconnect** rufen einfach ihre Gegenstücke im **Indexer** auf, der ein Abkömmling von **TCustomIndexDB** ist und in eine Postgres- oder Firebird-Datenbank schreibt. Welche dieser drei Datenbanken verwendet wird, hängt von einer Definition am Anfang der Datei ab:

```
{ $DEFINE USEFIREBIRD }
{ $DEFINE USEPG }
```





Da die Klasse nur ein einzelnes Dokument indiziert, kann die URL, die in die Datenbank geschrieben werden soll, in der Eigenschaft `PDFURL` angegeben werden. Sie wird auf den PDF-Dateinamen gesetzt, wenn nichts festgelegt wurde. Die Eigenschaft `Language` hat einen ähnlichen Zweck: Sie bestimmt die Sprache, die in die Datenbank geschrieben wird.

Mit der Eigenschaft `IgnoreWords` strings können Sie eine Liste von Wörtern angeben, die ignoriert werden sollen, und die `MinWordLength` ist standardmäßig auf 3 eingestellt, was bedeutet, dass Wörter mit einer Länge von weniger als 3 nicht in die Datenbank eingefügt werden. Das Ereignis `log` dient dazu, den Fortschritt zu protokollieren.

Die Hauptmethode ist `IndexPDF`, mit der der gesamte Indizierungsprozess gestartet wird. Die Methode akzeptiert einen Dateinamen oder einen Stream. Sie ist ganz einfach:

```
procedure TPDFIndexer.IndexPDF(const aStream: TStream);
begin
  DoLog('Start indexing PDF %s',[PDFURL]);
  Connect;
  try
    DoIndexPDF(aStream);
    DoLog('Done indexing PDF %s',[PDFURL]);
  finally
    Disconnect;
  end;
end;
```

Die geschützte Funktion `DoIndexPDF` liest die PDF-Datei und indiziert die Wörter, indem sie die Funktion `IndexPDFPage` für jede Seite aufruft:

```
procedure TPDFIndexer.DoIndexPDF(const aStream: TStream);
var
  aPageNo : integer;
  aPage : TPDFPageObject;
  aParser : TPDFParser;
  Doc : TPDFDocument;
begin
  aPageNo:=0;
  Doc:=Nil;
  aParser:=TPDFParser.Create(aStream);
  try
    Doc:=TPDFDocument.Create();
    aParser.ResolveToUnicodeCMaps:=True;
    aParser.ParseDocument(Doc);
    For aPage in Doc.Pages do
      begin
        Inc(aPageNo); IndexPDFPage(aPage,aPageNo);
      end;
    finally
      doc.free; aParser.Free;
    end;
  end;
end;
```

`IndexPDFPage` stellt lediglich sicher, dass jede Seite in einer separaten Transaktion behandelt wird. Die eigentliche Arbeit wird in der Methode `DoIndexPDFPage` erledigt:

```
procedure TPDFIndexer.IndexPDFPage(const aPage: TPDFPageObject; aPageNo: Integer);
begin
  Indexer.BeginTrans;
  try
    DoIndexPDFPage(aPage,aPageNo);
    Indexer.CommitTrans;
    DoLog('Indexed page %d',[aPageNo])
  except
    On E :exception do
      DoLog('Error %s while indexing page %d: %s',[E.ClassName,aPageNo,E.Message]);
    end;
  end;
end;
```

Die Methode `DoIndexPDFPage` funktioniert ähnlich wie unser Programm zur Anzeige von PDF-Seitentexten: Es durchsucht die Befehle der Seite nach Texten. Jeder gefundene Text wird in Wörter zerlegt, die dann in der Datenbank gespeichert werden:





```
TPDFIndexer.DoIndexPDFPage(const aPage: TPDFPageObject; aPageNo: Integer);
```

```
Var  
aData:          TSearchWordData;  
aCmd:           TPDFCommand;  
aWord, FontName, Rawtext: RawByteString;  
aFontRef:       TPDFRefData;  
UnicodeMap:     TPDFCMap;  
aFontObj:       TPDFFontObject;  
aDoc:           TPDFDocument;  
  
begin  
aDoc:=aPage.Document;  
With aData do  
begin  
FileDate:=Date;  
Language:='EN';  
Position:=aPageNo;  
URL:=PDFURL;  
end;  
for aCmd in aPage.CommandList do  
Begin  
if aCmd is TPDFTfCommand then  
begin  
FontName:=TPDFTfCommand(aCmd).FontName;  
if (FontName<>'') and (FontName[1]='/') then  
Delete(FontName,1,1);  
aFontRef:=aPage.FindFontRef(FontName);  
aFontObj:=aDoc.FindFont(aFontRef); // TPDFFontObject  
if Assigned(aFontObj) then  
UnicodeMap:=aFontObj.UnicodeCMap  
else  
UnicodeMap:=nil;  
end  
Else If aCmd is TPDFTextCommand then  
begin  
rawText:=TPDFTextCommand(aCmd).GetFullText(UnicodeMap);  
aData.Context:=Rawtext;  
SetCodePage(RawText,CP_UTF8);  
for aWord in DoSplit(RawText) do  
begin  
aData.SearchWord:=aWord;  
Indexer.AddSearchData(aData);  
end;  
end;  
end;  
end;  
end;
```

Der **TSearchWordData**-Datensatz, der zum Einfügen von Daten in die Indexdatenbank verwendet wird, ist in der Einheit **fpIndexer** definiert und wird von der **Indexer**-Klasse sowohl zum Einfügen von Daten als auch bei der Rückgabe von Ergebnissen bei der Suche nach Daten verwendet:

```
TSearchWordData = record  
Context:    UTF8String;  
FileDate:  TDateTime;  
Language:  UTF8String;  
Position:  int64;  
Rank:      integer;  
SearchWord: UTF8String;  
URL:       UTF8String;  
end;
```

Die Bedeutung dieser Felder sind:

- **Context** Ein gewisser Kontext um das Wort herum: normalerweise die Zeile, in der das Wort erscheint.
- **FileDate** Das Dateidatum der Datei, in der das Wort vorkommt.
- **Language** Ein Sprachcode mit 2 Buchstaben.
- **Position** Eine Position in der Datei. In unserer Anwendung werden wir hier die Seitenzahl einfügen.
- **Rank** Eine ganze Zahl, die den "Rang" angibt: damit kann das Wort bewertet werden.
- **SearchWord** Das Wort, das tatsächlich eingefügt werden soll. Wird bei der Suche zurückgegeben.
- **URL** die Datei, in der das Wort gefunden wurde. Dies wird der Name der PDF-Datei sein.

Die Methode **DoSplit**, mit der der Text in Wörter aufgeteilt wird, verdient etwas Aufmerksamkeit.

Es handelt sich um eine virtuelle Methode, so dass sie überschrieben werden kann, um einen anderen Mechanismus zu implementieren.

Die hier beschriebene **DoSplit**-Methode akzeptiert nur die lateinischen Buchstaben 'A'...'Z' als Wörter: Wenn eine andere Sprache (*kyrillisch oder eine östliche Sprache*) benötigt wird, muss diese Methode überschrieben werden.





```
function TPDFIndexer.DoSplit(const aText: RawbyteString): TStringDynArray;
Var aCount : Integer;
    c : AnsiChar;
    Cleaned : String;

Procedure MaybeAdd;
begin
    if (Cleaned<>") and AllowedWord(Cleaned) then
        begin
            Result[aCount]:=Cleaned;
            inc(aCount);
        end;
    Cleaned:= "";
end;

begin
aCount:=0;
Cleaned:= "";
Result:= [];
SetLength(Result,Length(aText) div MinWordLength);
for C in aText do
    if
        Upcase(C) in ['A..'Z']
    then
        Cleaned:=Cleaned+C
    Else
        MaybeAdd;
        MaybeAdd;
    SetLength(Result,aCount);
end;
```

Die Methode ist nicht sehr schwer zu verstehen. Beachten Sie, dass hier die Prüfung auf erlaubte Wörter durchgeführt wird: Wenn ein Wort nicht erlaubt ist, wird es nicht zurückgegeben. Das `AllowedWord` ist ganz einfach:

```
function TPDFIndexer.AllowedWord(aWord : String) : Boolean;
begin
    Result:=Length(aWord)>MinWordLength;
    if Result then
        Result:=FHash.Find(aWord)=Nil;
    end;
```

Die Variable `FHash` ist eine Hash-Liste, die aus der Liste der Wörter in der Eigenschaft `IgnoreWords` erstellt wird. Damit ist die Indexer-Klasse fertig. Sie müssen sie nur noch in einem Programm verwenden!

5 DAS PDF INDEXER PROGRAMM

Das Indexer-Programm ist ein einfaches Programm mit Eingaben für die `TPDFIndexer`-Klasse. Es ermöglicht die Indizierung einer einzelnen Datei oder mehrerer Dateien: Es kann ein Verzeichnis nach PDF-Dateien durchsuchen. Es verfügt außerdem über eine Schaltfläche zum Erstellen der benötigten Tabellen für den Indexer und ein Memo zur Anzeige von Protokollausgaben und Fortschrittsmeldungen. Das Formular ist in Abbildung 2 auf Seite 14 abgebildet. Die Verwendung ist wie folgt:

- Geben Sie die Informationen zur Datenbankverbindung ein. Die Datenbank für den Index muss bereits irgendwo existieren, und Sie müssen den Speicherort sowie die Benutzeranmeldeinformationen für die Verbindung zu dieser Datenbank angeben.
Zur Erinnerung: Der `fppdfindexer` ist standardmäßig für die Verwendung der Postgres-Datenbank eingerichtet.
- Wenn Sie eine andere Datenbank (**Firebird**) verwenden möchten, müssen Sie die Defines am Anfang der `fppdfindex`-Unit ändern.
- Klicken Sie auf die Schaltfläche 'Verbindung testen'. Sie erhalten eine Meldung, ob die Verbindung fehlgeschlagen ist oder erfolgreich war. Die Schaltflächen 'Tabellen erstellen' und 'PDFs indizieren' werden aktiv.
- Wenn Sie die Index-Tabellen noch nicht erstellt haben, verwenden Sie die Schaltfläche 'Tabellen erstellen', um die Tabellen zu erstellen.
- Legen Sie die anderen Parameter fest: eine Datei mit zu ignorierenden Wörtern, den Sprachcode.
- Wählen Sie eine zu indizierende PDF-Datei oder ein Verzeichnis mit PDF-Dateien aus.
- Klicken Sie auf den Button 'Index PDFs', um den Indizierungsprozess zu starten.

Beim Erstellen des Formulars wird eine Instanz der Klasse `TPDFIndexer` erzeugt:

```
procedure TMainForm.FormCreate(Sender: TObject);
begin
    FIndexer:=TPDFIndexer.Create(Self);
    FIndexer.OnLog:=@DoIndexLog;
    if FileExists(SIniFile) then
        ConfigDB(SIniFile);
end;
```





Wenn eine Konfigurationsdatei (`SIniFile`) existiert, wird sie gelesen und verwendet, um den Inhalt der verschiedenen Eingaben wiederherzustellen: Auf diese Weise muss der Benutzer diese Daten nicht jedes Mal neu eingeben. Es handelt sich einfach um das Lesen des Inhalts einer `.ini`-Datei. Der interessierte Leser kann die Quellen des Beispielprogramms konsultieren.

Die Funktionsweise des Programms ist recht einfach: Alle Verbindungsdaten müssen ausgefüllt werden. Mit dem Button Verbindung testen können Sie die Verbindung testen. Wenn der Test erfolgreich ist, können die Datenbanktabellen erstellt werden, oder, wenn die Tabellen bereits bekannt sind, kann mit der Button Index die Indizierung der Dateien gestartet werden.

Der Button Verbindung testen führt die Methode `TestConnection` aus.

```
procedure TMainForm.TestConnection;
begin
  ConfigConnection;
  Try
    FIndexer.Indexer.Connect;
    FIndexer.Indexer.Disconnect;
    FCanConnect:=True;
    SaveDBConfig(SIniFile);
  except
    on E :Exception do
      ShowMessage('Could not connect to index database:'#10+E.Message);
  end;
end;
```

Die Methode `configconnection` überträgt einfach die in den Bearbeitungen eingegebenen Daten auf die verschiedenen Eigenschaften des Indexers:

```
procedure TMainForm.ConfigConnection;
begin
  With FIndexer.Indexer do
    begin
      DatabasePath:=edtDatabaseName.Text;
      Hostname:=edtHostname.Text;
      UserName:=edtUserName.Text;
      Password:=edtPassword.Text;
    end;
end;
```

Die wichtigste Methode dieses Programms ist die Methode `actIndexExecute`.

Dies ist der `OnExecute`-Ereignishandler der `actIndex`-Aktion.

Sie beginnt mit der Einstellung der Eigenschaft `Language` und dem Lesen der Liste der zu ignorierenden Wörter. Danach indiziert sie entweder die angeforderte Datei mit `IndexFile` oder sie ruft `IndexDirectory` auf: eine Routine, die alle PDF-Dateinamen aus dem ausgewählten Verzeichnis abrufen und `IndexFile` mit den gefundenen Namen aufruft.

```
procedure TMainForm.actIndexExecute(Sender: TObject);
Var Msg : String;
begin
  if (FEIgnoreWords.FileName<>") and FileExists(FEIgnoreWords.FileName) then
    FIndexer.IgnoreWords.LoadFromFile(FEIgnoreWords.FileName);
  FIndexer.Language:=edtLanguage.Text;

  FFileCount:=0;
  FLastError:= "";
  if RBFile.Checked then
    IndexFile(FEPDF.FileName)
  Else
    IndexDirectory(DEPDFs.Directory);
  Msg:=Format('Done indexing %d files.',[FFileCount]);
  if FLastError<>"" then
    Msg:=Msg+#10'There were errors:'#10+FLastError;
  ShowMessage(Msg)
end;
```

Am Ende der Routine werden einige Statistiken angezeigt. Die `IndexDatei` ist denkbar einfach. Die Hauptlogik besteht darin, mögliche Fehler abzufangen und sie im output log anzuzeigen.





```
procedure TMainForm.IndexFile(const aFile :string);
Const
  ErrMsg = 'Error %s while indexing %s:%s';
begin
  try
    FIndexer.Connect;
    try
      Inc(FFileCount);
      FIndexer.IndexPDF(aFile);
    finally
      FIndexer.Disconnect;
    end;
  except
    On E :Exception do
      begin
        FLastError:=Format(ErrMsg,[E.ClassName,aFile,E.Message]);
        DoLog(FLastError);
      end;
    end;
  end;
  Application.ProcessMessages;
end;
```

Dies sind die wichtigsten Methoden des Programms.

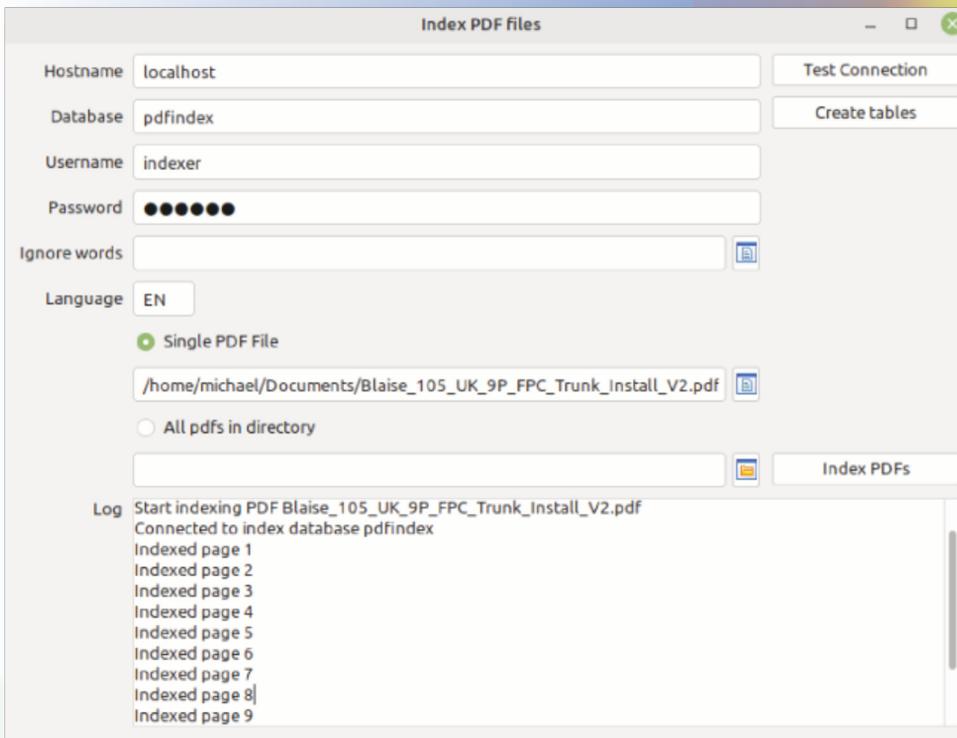


Figure 2: The PDF indexer program

6 FAZIT

In diesem Artikel haben wir ein PDF-Indizierungsprogramm mit Hilfe der in Free Pascal und Lazarus enthaltenen Units erstellt. Dabei haben wir viele Themen angeschnitten: die Struktur einer PDF-Datei, die vielen Objekte, die benötigt werden, um Text aus einer PDF-Datei zu extrahieren. Wir haben auch die von Free Pascal zur Verfügung gestellten Klassen gezeigt, um eine Indizierungsdatenbank zu erstellen. Diese Datenbank ist nun bereit, um in einem Suchprogramm verwendet zu werden: dies kann ein natives Programm oder ein PAS2JS-Programm sein. Das ist das Thema eines zweiten Artikels. **Beachten Sie**, dass der von Free Pascal erstellte Indexer ziemlich einfach ist und das Design begrenzt ist: Es können nur ganze Wörter durchsucht werden. Für eine leistungsfähigere Suche muss ein Volltext-Indexer wie die **Manticore-Suche** verwendet werden. Wir werden das in einem separaten Beitrag untersuchen.



1

BLAISE PASCAL MAGAZINE 107/108

Multimedia / Object Pascal / HTML5 / JavaScript / WebAssembly / Pascal /
Databases / CSS Styles / Progressive Web Apps
Android / iOS / Mac / Windows & Linux




Site installation and use
The PDF indover explained
Fresnel for Lazarus
The Pendulum
Using Bykes with a VCL programme
Lazarus programme using BuDataset
Cross compiling with FPC Up De Luxe
Quantum Internet



The newest LIB Stick

2

LAZARUS HANDBOOK



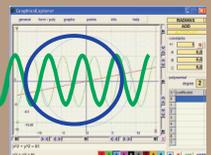
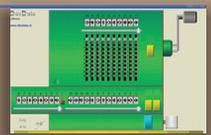
3




4

LEARN TO PROGRAM
HOWARD PAGE-CLARK
USING LAZARUS

DAVID DIRKSE
including 50 example projects

BLAISE PASCAL MAGAZINE
COMPUTER (GRAPHICS)
MATH & GAMES IN
PASCAL

1. One year Subscription
2. The newest LIB Stick
All issues 1- 107/108
On Credit Card and download
Including code of all projects
3. Lazarus Handbook
- PDF including Code
4. Book Learn To Program
- using Lazarus PDF including
19 lessons and projects
5. Book Computer Graphics
Math & Games
PDF including ±50 projects

2

BLAISE PASCAL MAGAZINE



BLAISE PASCAL MAGAZINE



BLAISE PASCAL MAGAZINE



Editor in Chief: Derlef Overbeek
Echelstenenbaan 21 3402 XA
IJsselstein Netherlands

Prof Dr Wirth, Creator of Pascal Programming language

Blaise Pascal, Mathemeticus

Prof Dr Wirth, Creator of Pascal Programming language

editor@blaisepascalmagazine.eu
<https://www.blaisepascalmagazine.eu>

NORMAL PRICE € 275
PRICE € 150



SUPERPACK

2022/23



Einige Tipps und Vorschläge, die Sie berücksichtigen können, um den Compiler schneller zu machen und weniger Speicher zu verbrauchen, wenn Sie die aktuelle Version von Delphi verwenden

<https://blog.marcocantu.com/>



Starter

Expert

EINLEITUNG

In den letzten paar Jahren, Embarcadero hat erhebliche Anstrengungen unternommen, um die Leistung des Delphi Object Pascal Compilers zu optimieren und die Speichernutzung zu reduzieren, damit unsere Kunden mit großen Anwendungen besser mit Delphi arbeiten können. Das Hauptaugenmerk lag dabei auf dem Win32-Compiler, aber auch andere Compiler wurden verbessert.

Wir haben in vielen Anwendungsszenarien erhebliche Fortschritte gemacht, aber gleichzeitig haben wir bei der Prüfung vieler Projekte unserer Kunden eine Reihe von Fällen entdeckt, in denen sich der Compiler nicht wie erwartet verhält. Wir planen, die meisten dieser Probleme in naher Zukunft zu beheben, auch wenn einige der Szenarien an der Grenze zur Ungültigkeit liegen (*wie die Übergabe von nicht existierenden Ordnern an den Compiler*).

Gleichzeitig erhalten wir Anfragen von Kunden, ob es irgendetwas gibt, was sie in ihrer Projektkonfiguration oder in ihrem Quellcode tun können, um die Dinge für ihre Entwickler heute besser zu machen. Das ist der Grund für diesen Blogbeitrag: Wir bieten Ihnen einige Tipps und Vorschläge, die Sie in Betracht ziehen können (sofern sie auf Ihre Codebasis zutreffen), damit der Compiler mit der aktuellen Version von Delphi schneller arbeitet und weniger Speicher verbraucht. Beachten Sie, dass die meisten der in diesem Artikel genannten Ideen sowohl bei der Kompilierung als auch bei der Verwendung von Code Insight powered by Delphi LSP hilfreich sind, da im Hintergrund derselbe Compiler verwendet wird. Indem Sie dem Compiler helfen, helfen Sie auch Code Insight

UNGÜLTIGE SUCHPFAD- ODER BIBLIOTHEKSPFAD-EINTRÄGE

Der Compiler sucht nach jeder Unit, auf die im Code durch uses-Anweisungen im Projektordner, im Projektsuchpfad, im Bibliothekspfad und an einigen anderen Stellen verwiesen wird. Wir haben festgestellt, dass ungültige Ordner in diesen Pfaden dazu führen, dass der Compiler immer wieder in diesen Pfaden nachschaut, ein Vorgang, der auf Betriebssystemebene bei nicht vorhandenen Pfaden viel langsamer ist.

Ein Beispiel für einen Bericht, den wir erhalten haben, mit dem Titel *'Delphi Compiler arbeitet extrem langsam, wenn mehrere ungültige Pfade im LIB PATH vorhanden sind'* finden Sie unter <https://quality.embarcadero.com/browse/RSP-3931>, als beispiel.



UNIT ALIASE

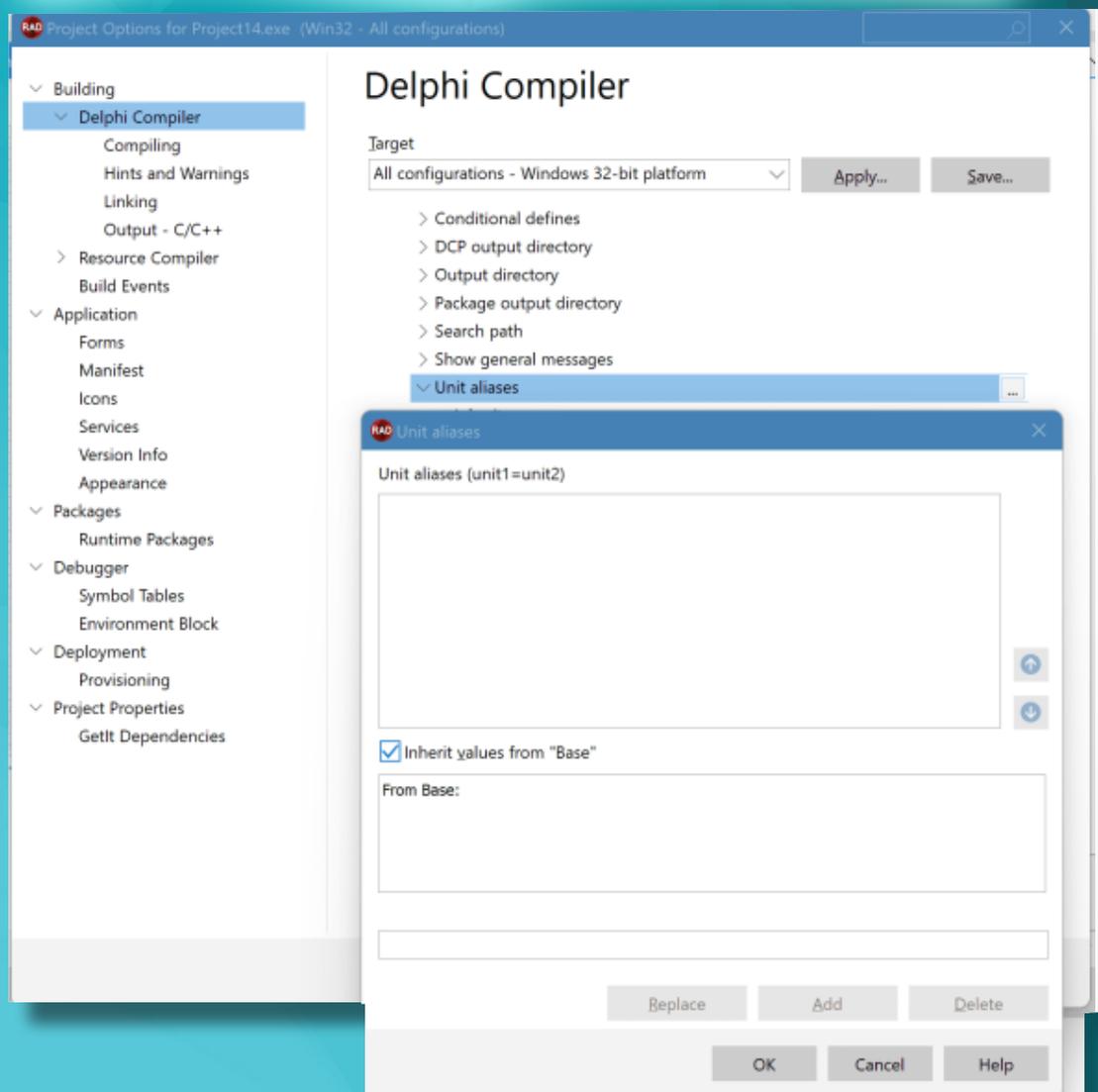
Unit-Aliase bieten die Möglichkeit, einen Unit-Namen in einer Uses-Anweisung auf einen anderen realen Unit-Namen abzubilden. Dies war bei der Migration von altem Code praktisch und wurde von Delphi im Laufe der Jahre verwendet, als einige der Kern-Units zusammengeführt oder umbenannt wurden.

In einigen frühen Versionen von Delphi fanden Sie zum Beispiel die folgenden Unit-Aliase:

```
WinTypes=Windows; WinProcs=Windows; DbiProcs=BDE; DbiTypes=BDE; DbiErrs=BDE
```

Derzeit ist kein Standard-Unit-Alias für neue Projekte definiert, aber möglicherweise haben Sie bereits Projekte mit Unit-Aliassen. Es wird empfohlen, diese zu entfernen (*möglicherweise einen nach dem anderen*) und den Quellcode Ihrer Projekte mit Verweisen auf die richtigen Units zu aktualisieren.

Eine umfangreiche Verwendung von Unit-Aliassen bedeutet für den Compiler zusätzliche Arbeit. Unseres Wissens verursacht dies zwar nur einen sehr geringen zusätzlichen Overhead, aber es ist dennoch ein bisschen zusätzliche Zeit und der Aufwand hilft Ihnen auch, Ihren Code sauberer und lesbarer zu halten.

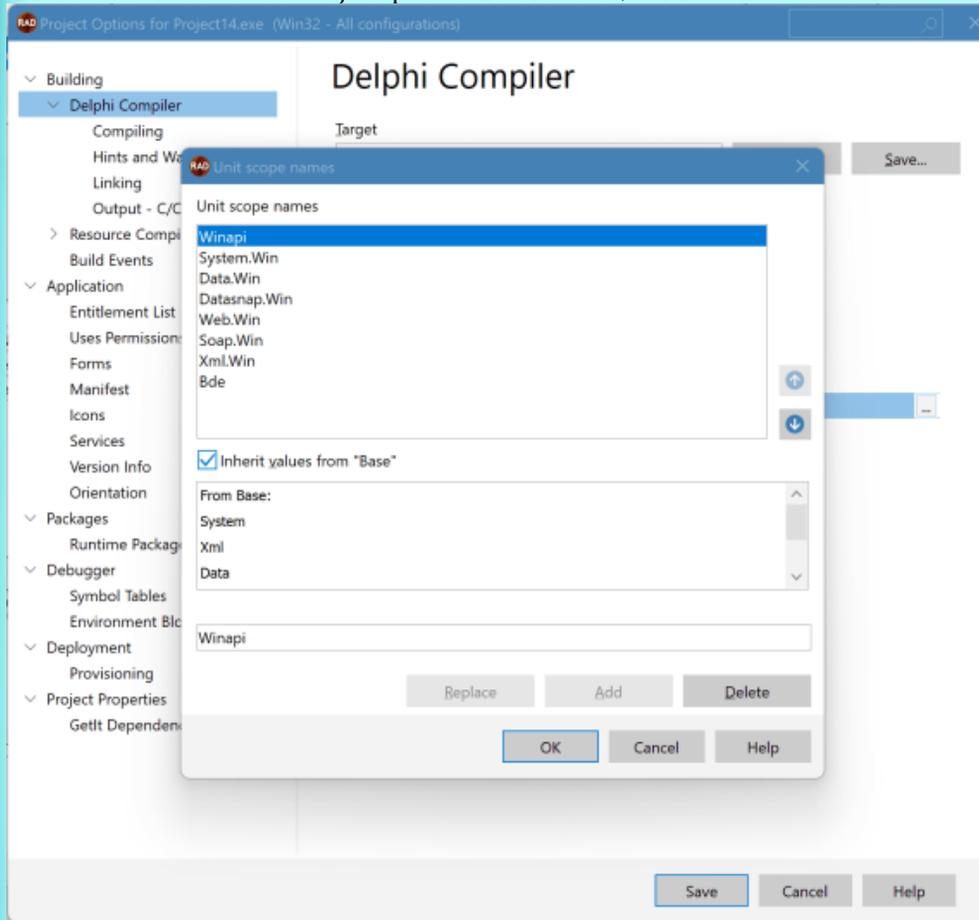


NAMEN VON EINHEITENBEREICHEN UND UNQUALIFIZIERTE NAMEN VON EINHEITEN

Vor mehr als zehn Jahren führte Delphi die Unit-Scope-Namen ein, d.h. ein Präfix vor allen Units, die Teil der Standardbibliotheken sind, wie Vcl.Forms oder System.SysUtils. Dies wird behandelt in

https://docwiki.embarcadero.com/RADStudio/Alexandria/en/Unit_Scope_Names

Wenn Sie bereits Projekte mit kurzen Unit-Namen haben, besteht ein schneller Migrationspfad darin, Unit-Bereichsnamen als Teil der Projektoptionen zu definieren, wie Sie hier sehen können:



Wie Sie sehen, fügt die Delphi IDE immer noch standardmäßig eine große Anzahl von Unit-Bereichsnamen hinzu, selbst bei neuen Projekten (*das sollten wir ändern, aber das ist ein anderer Punkt*). Nun führen unqualifizierte Unit-Namen in Ihrem Code dazu, dass eine Datei für jedes der möglichen Präfixe gesucht wird, in allen Ordnern in Ihrem Such- oder Bibliothekspfad. Stellen Sie sich vor, Sie haben 20 Ordner und 20 Präfixe, das sind potenziell 400 Dateispeicherorte, nach denen Sie suchen müssen.

Es überrascht nicht, dass wir Berichte wie diesen erhalten, der den Titel 'Unqualifizierte Unit-Namen verursachen eine große Anzahl von Dateisuchen' trägt und als

<https://quality.embarcadero.com/browse/RSP-18130>.

Der Compiler verfügt über eine Zwischenspeicherlogik, um die Arbeit zu reduzieren, aber auch hier handelt es sich nur um eine Funktion, die bei der Migration des Codes hilft. Entwickler sollten schließlich ihre uses-Anweisungen bereinigen, voll qualifizierte Unit-Namen für alle Units in Embarcadero-Bibliotheken verwenden und alle Unit-Bereichsnamen-Konfigurationen auf der Projekteinstellungsebene entfernen. Dies ist eine weitere Code-Änderung, die einen erheblichen Nutzen für den Compiler haben kann.

WIEDERHOLTE UND DOPPELTE NAMEN VON UNITS

Apropos Unit-Namen: Es gibt ein weiteres Szenario, das bekanntermaßen Probleme verursacht. In diesem Fall handelt es sich weniger um ein Leistungsproblem, sondern um ein Szenario, das zu Fehlern führen kann, insbesondere in Code Insight und einigen unserer IDE-Tools.

Das Szenario sieht folgendermaßen aus: Sie haben in der IDE eine Projektgruppe mit mehreren Projekten geöffnet, die verschiedene Einheiten (in verschiedenen Ordnern) mit demselben Namen enthalten. Mit anderen Worten, ein Projekt hat `MyUnit.pas` in einem Ordner und das andere Projekt hat eine andere und nicht verwandte Unit, die ebenfalls `MyUnit.pas` heißt, in einem anderen Ordner.

Dies ist ein Szenario, mit dem der Compiler in der Regel gut zurechtkommt, aber nicht immer, wie Sie in dem Bericht 'Compiler verwechselt Unit-Namen in Projektgruppe' unter

<https://quality.embarcadero.com/browse/RSP-39293>. (You need to login as a user)

Wir haben vor ein paar Jahren spezielle Maßnahmen ergriffen, damit der Debugger in ähnlichen Szenarien nicht mehr verwirrt wird. Delphi LSP kann jedoch durch Units mit demselben Namen verwirrt werden. Wann immer es möglich ist, wäre es viel besser, den gleichen Namen für verschiedene Units zu vermeiden.

KREISUNITSVERWEISE

Zurück zu den Szenarien, die sich auf die Compilerleistung auswirken. Eines der kritischsten, die wir kennen, ist die übermäßige Verwendung von zirkulären Unit-Referenzen im Implementierungsteil der Units. Wie Sie wissen, sind im Schnittstellenteil der Units keine gegenseitigen oder zirkulären Unit-Referenzen erlaubt.

Im Implementierungsteil einer Unit können Sie jedoch theoretisch auf alle anderen Units im gesamten Projekt verweisen.

Wenn Delphi eine Unit kompiliert, muss es sich alle verwendeten Units ansehen und durch den gesamten Graphen navigieren, wobei es darauf achtet, dass die Quelle die Zyklen überprüft - um eine Endlosschleife bei der Suche zu vermeiden.

Im Falle eines komplexen Graphen mit Hunderten von Units sind die Kosten für diesen Navigationsprozess beträchtlich. Vor kurzem hat unser Support-Team ein interessantes Experiment durchgeführt.

Sie haben einen Codegenerator erstellt, der eine Anwendung erzeugt, in der jede Unit jede andere Unit verwendet. Dies ist eindeutig ein extremer Anwendungsfall - ich hoffe wirklich, dass Sie keine derartigen Projekte haben! In einem solchen Szenario verdoppelt sich die Kompilierungszeit fast, wenn Sie eine weitere Unit hinzufügen.

Mit anderen Worten, die Kompilierungszeit verschlechtert sich exponentiell, wenn eine weitere Unit hinzugefügt wird die wie alle anderen Units im Projekt verwendet und von allen anderen Units verwendet wird (*es ist exponentiell, da jede zusätzliche Unit n^2 Unit-Referenzen hinzufügt, wobei n die Anzahl der vorhandenen Units ist*).

Auch hier handelt es sich um ein Szenario, für das der Compiler bereits optimiert wurde, indem die Suche und die Namensuche auf Hash-Wörterbücher und ähnliche optimierte Datenstrukturen verlagert wurden, und um einen Bereich, in dem wir derzeit aktiv nach Verbesserungen für den Compiler suchen. Siehe zum Beispiel diesen Bericht, der in Delphi 10.4.2 behoben wurde:

<https://quality.embarcadero.com/browse/RSP-28811>

Die Verwendung von zirkulären uses-Anweisungen ist jedoch auch ein Anwendungsszenario, das gegen jede gute Programmierpraxis verstößt, da Sie am Ende einen Spaghetti-Diagramm der verwendeten Units haben.

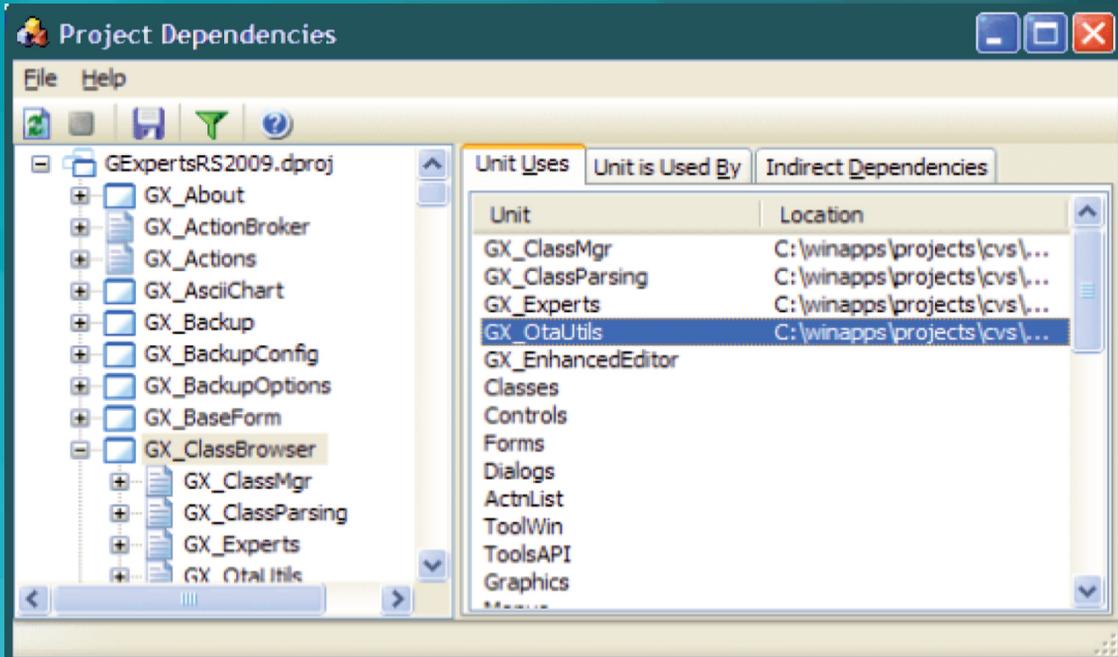
Es ist eine gute Praxis, die Abhängigkeiten zwischen Units und Klassen zu reduzieren. Einige lineare Abhängigkeiten (*Low-Level-Units, die von High-Level-Units verwendet werden, UI-Units, die Datenzugriffs-Units verwenden und nicht umgekehrt*) sind akzeptabel, aber ein komplexes Diagramm ist für die Wartungsfreundlichkeit Ihres Codes im Laufe der Zeit nicht gesund. Außerdem belastet es die Leistung des Compilers und die Speichernutzung erheblich.

Uns liegen Berichte von Delphi-Entwicklern mit ziemlich großen Anwendungen vor, dass die Reorganisation ihrer uses-Anweisungen zur Vermeidung von zirkulären Unit-Referenzen die Kompilierungszeit um bis zu 90 % reduziert hat, d.h. von zehn Minuten auf eine Minute.

Falls Sie dieses Problem angehen wollen, empfehle ich Ihnen, eines der Tools auf der nächsten Seite (*53 - 5 des Artikels*) in Erwägung zu ziehen, das Ihnen helfen kann, das aktuelle uses-Units-Diagramm Ihrer Anwendung zu bestimmen und Zyklen zu erkennen:



- **GExpert Projekt-Abhängigkeiten** (siehe Bild unten):
https://gexperts.org/tour/index.html?project_dependencies.html
- **Peganza Pascal Analyzer**: <https://www.peganza.com/#PÄL>
- **Delphi Unit Dependency Scanner**: <https://github.com/norgepaul/DUDS>



Beachten Sie nebenbei, dass der Compiler es auch nicht mag, wenn dieselben Units in unterschiedlicher Reihenfolge verwendet werden (*da dies den Abhängigkeitsgraphen der Units, die diese Units verwenden, beeinflusst*).

Mit anderen Worten: Wenn zwei Units sowohl Unit A als auch Unit B verwenden, ist es besser, in beiden Fällen "uses A,B;" zu schreiben als in der zweiten Unit "uses B, A;".

Verglichen mit der Vermeidung von zirkulären Unit-Referenzen ist dies eher unbedeutend, aber in einer sehr großen Anwendung fällt es dennoch auf.

Schließlich könnten Sie auch in Betracht ziehen, unnötige uses-Anweisungen zu entfernen.

Ein Tool, das Ihnen dabei besonders hilft, ist Delphi Parser:

<https://delhiparser.com/product/remove-superfluous-uses-optimizer-wizard-evaluation-edition>

WIEDERHOLTE INSTANZEN GENERISCHER TYPEN

Der letzte Vorschlag, den wir zur Unterstützung des Delphi-Compilers machen möchten, ist etwas umstrittener und etwas, das der Compiler nach einigen Jahren besser können sollte. Wir wissen jedoch, dass dies einen dramatischen Unterschied ausmachen kann, in diesem Fall sowohl in Bezug auf die Kompilierzeit als auch auf die Speichernutzung, um das Risiko von Out-of-Memory-Compilerfehlern zu vermeiden.

Das Delphi-Typsystem (*wie das klassische Turbo Pascal-System*) basiert auf der Äquivalenz von Typdeklarationen. Mit anderen Worten: Selbst wenn zwei Typen denselben Namen und dieselbe Struktur haben, sind sie unterschiedlich, wenn sie in verschiedenen Einheiten deklariert sind.

Wenn Sie eine Variable vom Typ `MyUnit1.TMyType` einer Variablen vom Typ `MyUnit2.TMyType` zuweisen, erhalten Sie einen Compiler error, unabhängig davon, dass die beiden Typen die gleiche identische Struktur haben.

Die einzige Ausnahme von dieser Regel gilt für generische Typen. Wenn Sie eine Variable vom Typ `TList` deklarieren, erzeugen Sie auch einen impliziten Typ `TList`, der auf dem generischen Typ `TList` basiert. Sie können mehrere Typinstanzen haben, die in verschiedenen Einheiten deklariert sind und alle der `TList`-Signatur entsprechen. Der Compiler betrachtet sie als kompatibel und erlaubt Ihnen, sie einander zuzuweisen.

Was jedoch hinter den Kulissen passiert, ist, dass der Compiler einen neuen temporären Typ für jedes Vorkommen in verschiedenen Einheiten erzeugt. Sie haben also z.B. `TList` in `MyUnit1` (mit einem temporären Typnamen) und `TList` in `MyUnit2` (mit einem anderen temporären Namen). Für jeden dieser Typen erstellt der Compiler Instanzen für den generischen Typ, aber auch für jede der generischen Methoden des generischen Typs.

Bei einer großen Anzahl komplexer generischer Typen (*vererbt und verschachtelt*) kann dies einen erheblichen Aufwand für den Compiler bedeuten.

Die Lösung für dieses Problem besteht nicht darin, keine Generika mehr zu verwenden, sondern wann immer möglich gemeinsame Instanzen generischer Typen zu erstellen. Mit anderen Worten: Anstatt `TList` für Dutzende von verschiedenen Einheiten zu verwenden, können Sie deklarieren

```
type
```

```
    TListOfStrings = TList;
```

und verwenden den Typ `TListOfStrings` überall dort, wo Sie `TList` verwenden würden. In einfachen Beispielen ist der Unterschied kaum spürbar. In einer großen Anwendung, in der viele generische Typen verwendet werden, kann der Unterschied sehr groß sein.

Auch hier wissen wir alle, dass der Compiler ähnliche Szenarien besser handhaben könnte, und wir haben daran gearbeitet, diesen Anwendungsfall zu verbessern - und wir haben ihn in den letzten Jahren auch ein wenig verbessert. Dennoch können Sie dem Compiler helfen, schneller zu sein und weniger Speicher zu verbrauchen, indem Sie einen Teil Ihres Codes in diese Richtung refaktorisieren - d.h. Code so schreiben, als gäbe es die generische Typäquivalenz nicht.

CONCLUSION

Gibt es noch mehr? Möglicherweise, ja.

Dies sind einige der Szenarien, die wir bei der Untersuchung der Leistung des Delphi-Compilers gefunden haben und die teilweise durch Änderungen am Code behoben werden können.

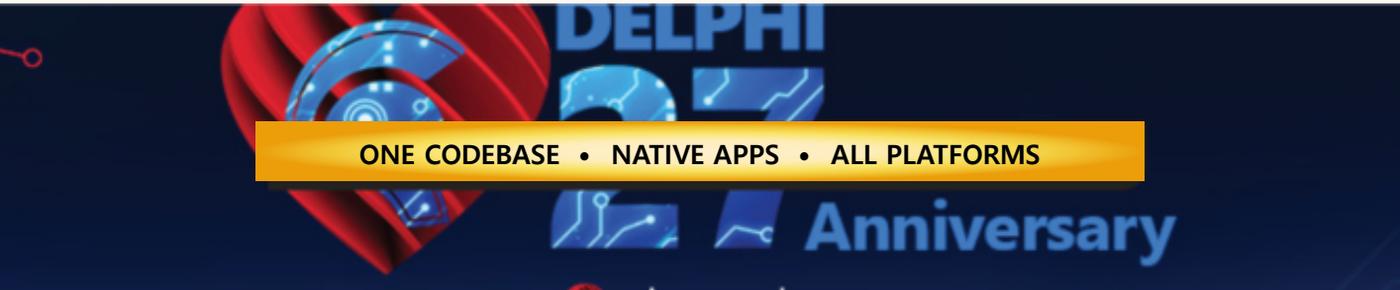
Ich empfehle insbesondere die Entfernung von Unit-Aliasen und unqualifizierten Units sowie die Erkennung und Entfernung von zirkulären Unit-Referenzen. Wenn Sie in erheblichem Umfang Generika verwenden, sollten Sie auch den letzten Vorschlag oben berücksichtigen.

Dies empfiehlt sich insbesondere dann, wenn Sie eine große Anwendung haben und der Compiler zu langsam ist: Die meisten Delphi-Entwickler sind mit der Compiler-Leistung recht zufrieden und müssen in dieser Hinsicht nichts tun.

Ich bin mir auch ziemlich sicher, dass ich nicht alle Szenarien aufgezählt habe, die eine Verlangsamung des Compilers verursachen und die teilweise durch Code- oder Konfigurationsänderungen behoben werden können. Wenn Sie weitere Vorschläge haben, die auf Ihrer Erfahrung beruhen, sind diese mehr als willkommen!

PS. Ich wünsche Ihnen schöne Feiertage und alles Gute für das Jahr 2023.





ONE CODEBASE • NATIVE APPS • ALL PLATFORMS

From
€1.869,00



Delphi 11 Alexandria

A powerful RAD environment for quickly developing high-performance native cross-platform applications using powerful visual design tools and integrated toolchains that independent developers and enterprise development teams love.

Shop Delphi

From
€1.869,00

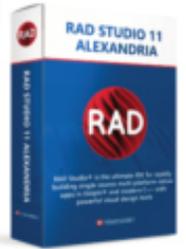


C++Builder 11 Alexandria

C++Builder is a complete RAD environment, loved by developers, with an integrated toolchain for modern C++ to help quickly build high-performance native Windows apps 10x faster than competing solutions.

Shop C++ Builder

From
€3.299,00



RAD Studio 11 Alexandria

RAD Studio is the ultimate RAD environment loved by developers for quickly building high-performance native cross-platform applications in Modern C++ and Delphi using powerful visual design tools and integrated toolchains.

Shop RAD Studio

<https://www.barnsten.com/>
+31 23 542 22 27





EINFÜHRUNG: WAS IST EIN BUFDATASET

TBufDataset ist ein Abkömmling von **TCustomBufDataset**, der ein datenbankfähiges gepuffertes Dataset implementiert. **TBufDataset** ist ein speicherinternes Dataset, das als lokales, eigenständiges Dataset oder als lokaler Puffer für Aktualisierungen eines entfernten Datasets verwendet werden kann. **TBufDataset** implementiert viele Funktionen, die denen von **TClientDataset** in Delphi ähneln. Es ist jedoch nicht dazu gedacht, code- oder funktionskompatibel mit **TClientDataset** zu sein. Es dient zur Verwaltung der Puffer, die für den Zugriff auf Datensatzdaten verwendet werden, und wird als Vorläufer für Klassen wie **TSQLQuery** und **TRESTBufDataset** verwendet.

Als Nachkomme von **TDBDataset** bietet es Zugriff auf viele der Datenbankfunktionen, die im FCL DB-Paket unterstützt werden. Als Nachkomme von **TDataset** bietet sie außerdem Zugriff auf die bekannten Navigations- und Datenverarbeitungsmethoden der Vorgängerklassen. **TBufDataset** verfügt über Funktionen, die das lokale Speichern und Abrufen von Felddefinitionen und Datensatzdaten ermöglichen. Darüber hinaus werden Funktionen bereitgestellt, die eine lokale Indizierung von Datensatzdaten im Dataset ermöglichen. Es steht ein umfassender Parser/Ausdruckauswerter zur Verfügung, der die vollständige Unterstützung von Filtern im Dataset ermöglicht.

TBufDataset setzt die Sichtbarkeit für Eigenschaften und Methoden, die in der Vorgängerklassen definiert sind.

ERSTELLEN DER ANWENDUNG

Richten wir eine Master-Detail-Beziehung zwischen zwei **BufDataSet**'s ein (in Delphi könnte es sich um *Client Datasets* handeln). Wir beginnen mit einer neuen Anwendung über **New** → **Application**. Dann fügen wir die erforderlichen Komponenten in das Formular ein und speichern das Projekt als `platen.lpi` und das Formular als `main.pas`. Das Formular sieht dann wie folgt aus:

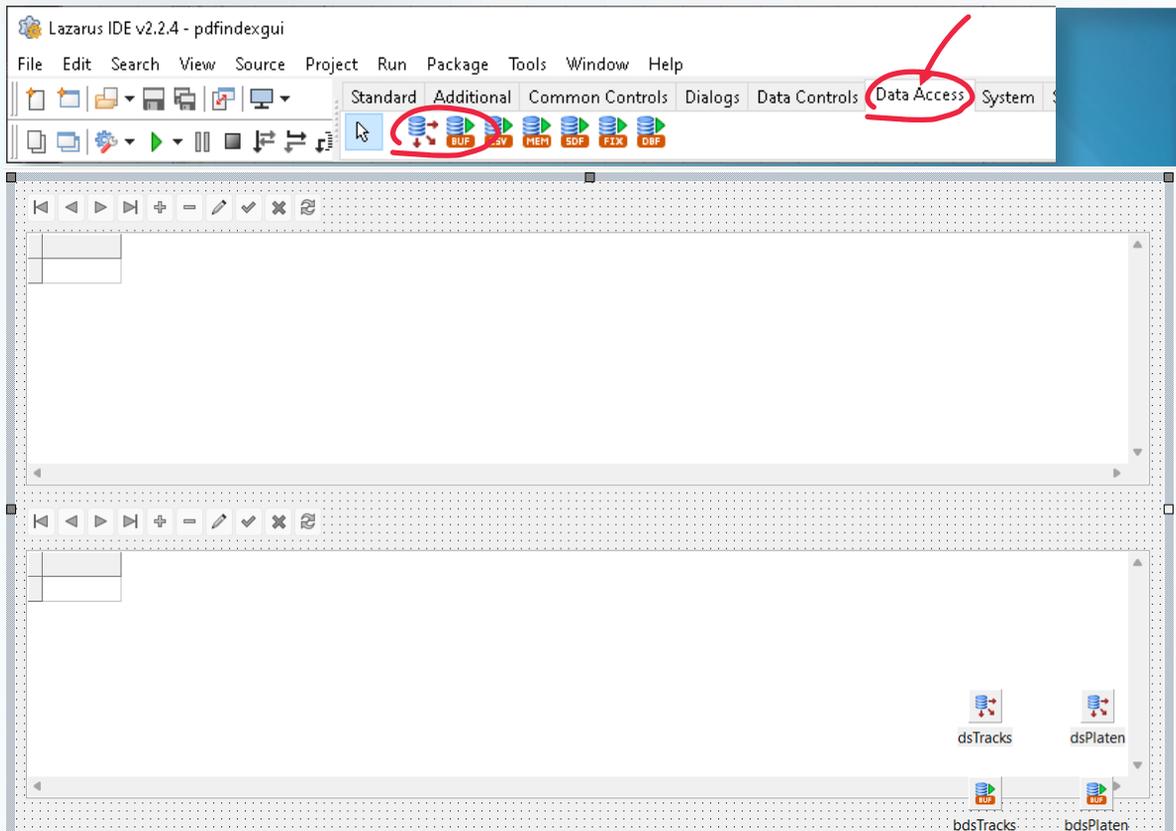


Abbildung 1: Designanwendung Hauptformular.





Passen Sie dann die Komponenteneigenschaften an:

Creating The form itself:

```
Name: MainForm
Caption: Platen overview
// Music Records overview
```

BufDataset1:

```
Name //of the dataset:
bdsPlaten //bdsRecords
FieldDefs://add 3 Items through the fieldseditor
```

```
IndexFieldNames: ID
BufDataset1.FieldDefs.0:
//The numbering starts at 0
Name: ID
DataType: ftAutoInc
BufDataset1.FieldDefs.1:
Name: Album
DataType: ftString
Size: 30
BufDataset1.FieldDefs.2:
```

```
Name: Artiest
DataType: ftString
Size: 30
DataSource1:
Name: dsPlaten
Dataset: bdsPlaten
```

```
//-----
```

BufDataset2://Second Dataset

```
Name: bdsTracks
FieldDefs://add 3 Items through the fieldseditor
IndexFieldNames: ID
BufDataset2.FieldDefs.0
Name: ID
DataType: ftAutoInc
BufDataset2.FieldDefs.1:
```

```
Name: AlbumID
DataType: ftInteger
BufDataset1.FieldDefs.2:
```

```
Name: Tracknaam
DataType: ftString
Size: 50
DataSource2:
```

```
Name: dsTracks
Dataset: bdsTracks
DBNavigator1:
```

```
DataSource: dsPlaten
DBGrid1:
```

```
Anchors: akBottom, akLeft, akRight, akTop
DataSource: dsPlaten
DBNavigator2:
```

```
Anchors: akBottom, akLeft
DataSource: dsTracks
DBGrid2:
```

```
Anchors: akBottom, akLeft, akRight
DataSource: dsTracks:
```

CODE

Noch etwas Code für den korrekten Betrieb.

Fügen Sie unter dem Abschnitt Privat der Überschrift Typ **TmainForm** hinzu

```
private
fn1: string;
fn2: string;

Zuerst die Auflistung von OnCreate das Formular,
um festzustellen, ob die Dateien bereits existieren,
andernfalls erstellen Sie Dateien:

procedure TmainForm.FormCreate(Sender: TObject);
begin
fn1 := 'platten.bds';
if FileExists(fn1) then
begin
bdsPlaten.LoadFromFile(fn1);
bdsPlaten.Active := True;
bdsPlaten.First;
end
else
begin
bdsPlaten.CreateDataset;
bdsPlaten.Active := True;
bdsPlaten.SaveToFile(fn1);
end;
fn2 := 'tracks.bds';
if FileExists(fn2) then
begin
bdsTracks.LoadFromFile(fn2);
bdsTracks.Active := True;
bdsTracks.First;
end
else
begin
bdsTracks.CreateDataset;
bdsTracks.Active := True;
bdsTracks.SaveToFile(fn2);
end;
end;
end;
```

Dann das Ereignis **OnClose**, Dateien wegschreiben und schließen Sie die App:

```
procedure TmainForm.FormClose(Sender: TObject;
var CloseAction: TCloseAction);
begin
bdsPlaten.SaveToFile(fn1);
bdsTracks.SaveToFile(fn2);
CloseAction := caFree;
end;
```

Das Ereignis **OnFilteredRecord** von **bdsTracks** sorgt dafür, dass nur die Datensätze, die zum ausgewählten Datensatz gehören, im anderen Datensatz angezeigt werden:

```
procedure TmainForm.bdsTracksFilterRecord(DataSet: TDataSet;
var Accept: Boolean);
begin
Accept := (bdsTracks.FieldByName('AlbumID').AsInteger =
bdsPlaten.FieldByName('ID').AsInteger);
end;
```

Das Ereignis **BeforeInsert** von **bdsPlaten**, um das zweite Gitter zu leeren, wenn ein neuer Datensatz freigegeben wird:

```
procedure TmainForm.bdsPlatenBeforeInsert(DataSet:
TDataSet);
begin
DBGrid2.Clear;
end;
```





Die **OnAfterPost**-Ereignisse der beiden BufDatasets, um sicherzustellen, dass die Tracks in der richtigen Reihenfolge sind:

```
procedure TmainForm.bdsTracksAfterPost(DataSet: TDataSet);
begin
    bdsTracks.First;
end;

procedure TmainForm.bdsPlatenAfterPost(DataSet: TDataSet);
begin
    bdsTracks.Filtered := False;
    bdsTracks.Filtered := True;
    bdsTracks.First;
end;
```

Das **OnBeforePost**-Ereignis von **dbTracks**, um das Feld **AlbumID** automatisch auszufüllen:

```
procedure TmainForm.bdsTracksBeforePost(DataSet: TDataSet);
begin
    bdsTracks.FieldName('AlbumID').Value := bdsPlaten.FieldName('ID').Value;
end;
```

Und schließlich, das **OnClick**-Ereignis von **Navigator1**:

```
procedure TmainForm.DBNavigator1Click(Sender: TObject; Button: TDBNavButtonType);
begin
    if ((Button = nbNext) or (Button = nbPrior) or (Button = nbFirst) or (Button = nbLast))
    then
        begin
            bdsTracks.Filtered := False;
            bdsTracks.Filtered := True;
            bdsTracks.First;
        end;
end;
```

Das Ergebnis sieht wie folgt aus:

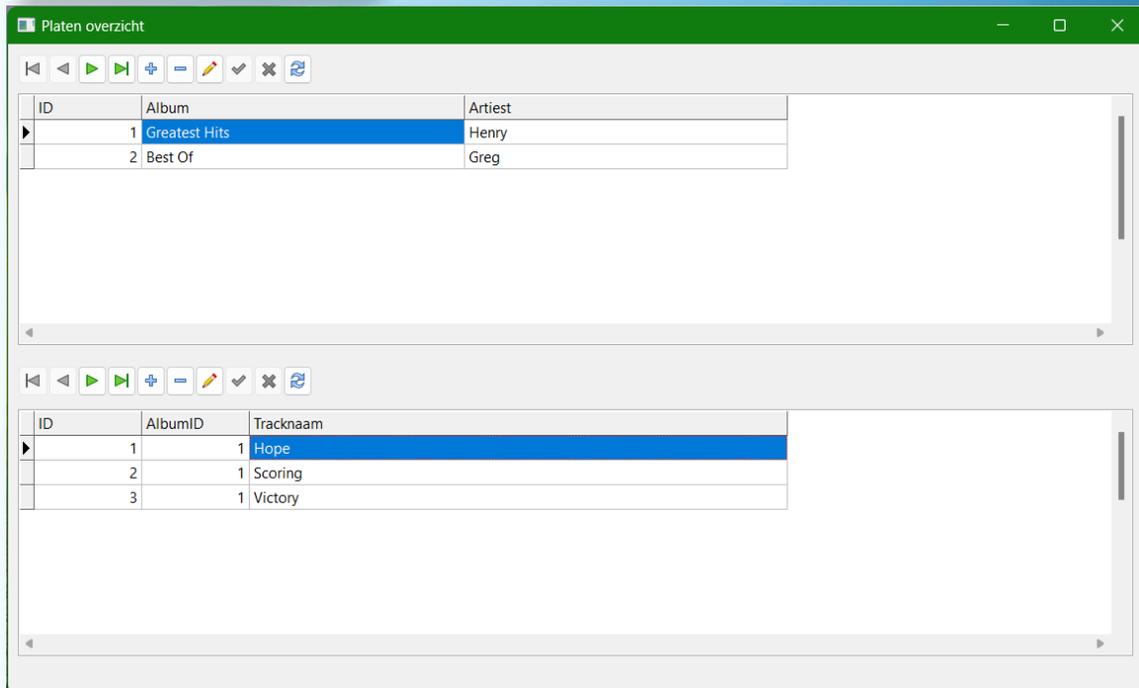


Abbildung 2: Anwendung läuft mit Beispieldaten.

Quellcode verfügbar für Abonnenten....



ABSTRACT

Teil der Lazarus Roadmap
ist eine neue Initiative namens

Projekt Fresnel: ein Ersatz für die
LCL (*Lazarus Component Library*). In diesem
Artikel erläutern wir die Ziele dieses neuen Projekts.

1 GESCHICHTE

Die LCL ist - wie ihr Delphi-Pendant VCL - alt.
Die Grundlagen für dieses UI-Framework wurden vor 20 Jahren gelegt,
als es als freie Version der VCL - des in Delphi verwendeten
UI-Frameworks - konzipiert wurde.

Sein Design spiegelt die UI-Prinzipien der damaligen Zeit wider, genauer gesagt,
das damalige Windows UI-Toolkit, die Win32 API, die zum ersten Mal in Windows 95
eingesetzt wurde. Man könnte also sagen, dass die LCL eine 27 Jahre alte Technologie ist.
Dass man mit 20 Jahre altem Code heute noch eine funktionierende Anwendung erstellen
kann, ist den großen Anstrengungen zu verdanken, die Dinge **abwärtskompatibel** zu halten.
Aber inzwischen hat sich die Welt weiterentwickelt.

Das Internet hat die Welt im Sturm erobert, und die UI-Entwicklung im Browser basiert auf zwei
Standards: **HTML** (das DOM) und **CSS**. CSS bietet viele Möglichkeiten der UI-Anpassung, und die
Möglichkeiten von CSS wachsen noch immer.
Das Auftauchen anderer Geräte hat die UI-Entwickler vor Herausforderungen gestellt:
Während der 1990er Jahren, der PC die vorherrschende Plattform war, ist dies heute nicht mehr der Fall;
Handys sind allgegenwärtig, und es gibt sie in allen Formen, Grössen und Farben.

Der alte Desktop-PC ist nicht mehr der dominierende Formfaktor.
CSS hat es Webentwicklern ermöglicht, mit diesen verschiedenen Formfaktoren umzugehen:
Es ist schwierig, sich eine moderne Website vorzustellen, die nicht mit Blick auf die Reaktionsfähigkeit gestaltet
wurde.

Als Reaktion auf diese neuen Plattformen wurde **Firemonkey** in Delphi eingeführt: eine **Alternative** zur VCL, die auf
mehreren Plattformen verwendet werden kann. Es verfügt über Stile, die verwendet werden können, um den
verschiedenen Formfaktoren gerecht zu werden, und kann als Ersatz für CSS angesehen werden.
Firemonkey kann mit verschiedenen Backends umgehen, zum Beispiel kann **SKIA** als Backend oder OpenGL
verwendet werden. Im Kern verwendet Firemonkey jedoch immer noch das in der VCL verwendete
Positionierungsparadigma, und der Stilmechanismus verwendet ein Format, das dem DFM-Format ähnelt.



Abbildung 1: Büste von Augustin Fresnel
von David d'Angers (1854),
ehemals im Leuchtturm von Hourtin, Gironde,
und jetzt ausgestellt im Musée national de la Marine

2 EINFÜHRUNG IN DAS PROJEKT FRESNEL

Anstatt ein neues Format einzuführen, um die UI-Schicht von Lazarus einzuführen, scheint es eine logische Wahl zu sein, den einen erfolgreichen und standardisierten Mechanismus zu verwenden: CSS.

Die Paradigmen der VCL müssen vollständig über Bord geworfen werden: das bedeutet keine obligatorische Positionierung mehr, keine obligatorischen Eigenschaften `Top`, `Left`, `Width` und `Height`.

Stattdessen müssen alle Angelegenheiten der Benutzeroberfläche und des Layouts an CSS-Eigenschaften delegiert werden.

Kurz gesagt: Das Ziel von **Project Fresnel** ist es, eine Reihe von Steuerelementen (*oder Widgets*) zu entwickeln, die von CSS gesteuert werden.

Diese Steuerelemente werden den folgenden architektonischen Prinzipien gehorchen:

Die Widgets müssen streamable sein:

- Das heißt, die Widgets sind Nachkommen von `TComponent` und müssen veröffentlichte Eigenschaften verwenden. Diese Anforderung bedeutet, dass ein Formfile erstellt werden kann, und die Lazarus IDE kann verwendet werden, um die Widgets zu manipulieren.
- Das Layout und das Aussehen werden vollständig durch CSS bestimmt. Das bedeutet, dass jedes Widget eine Eigenschaft hat, die das zu verwendende CSS darstellt. Zusätzliche Eigenschaften zur Manipulation in einem Objektinspektor können natürlich eingeführt werden.
- Es müssen mehrere Zeichen-Backends unterstützt werden. Da Lazarus plattformübergreifend ist, ist es selbstverständlich, versteht es sich von selbst, dass die verschiedenen Plattformen ein spezielles Backend bereitstellen, ähnlich wie es das Custom-Drawn Widgetset oder das FPGUI-basierte Widgetset heute bietet.
- Die Widgets werden unabhängig von der LCL sein: Es darf keine Abhängigkeit von der LCL bestehen. Die Widgets müssen in der Lage sein, zusammen mit einem LCL-Framework zu koexistieren: Eine Anwendung muss in der Lage sein, LCL-Formulare zusammen mit CSS-basierten Formularen zu verarbeiten. In der Tat stellt die LCL eines der möglichen Backends für die Fresnel-Widgets bereit.

Der Name **Fresnel** wurde zu Ehren des französischen **Physikers Augustin-Jean Fresnel** gewählt, der wichtige Beiträge zur Theorie des Lichts leistete: Was ist ein UI, wenn nicht ein geschicktes Lichtspiel?

Im letzten Teil dieses Artikels finden Sie eine kurze Geschichte über **Fresnel**.

Die Situation, in der die LCL verwendet wird, um die Fresnel-Widgets zu unterstützen (*wie es in der IDE der Fall sein wird*) ist in *Abbildung 1 auf Seite 3* dargestellt

3 ARCHITEKTONISCHE KONSEQUENZEN VON CSS

Idealerweise sollte ein mit den Fresnel-Widgets erstelltes Projekt in der Lage sein, ein vorhandenes CSS-Framework wie **Bootstrap**, **Foundation** oder **Tailwind CSS** zu verwenden und auf ein Formular anzuwenden.

Das bedeutet, dass wir mehrere Dinge bereitstellen müssen:

- Einen CSS-Parser und einen Mechanismus zur Übersetzung des CSS in native Strukturen.
- Einen CSS-Resolver: Dieser Mechanismus bestimmt, welche CSS-Regeln für ein bestimmtes Widget gelten.
- Einen Zeichenmechanismus, der alle unterstützten CSS-Eigenschaften kennt.
- Ein Layouter.
CSS verfügt über mehrere Möglichkeiten, das Layout zu bestimmen: `float`, `flex`, `grid` usw., wie durch die CSS-Eigenschaft `display` bestimmt.
- Einige Widgets, die Standard-HTML-Elemente nachahmen:
Das CSS-Framework setzt die Existenz bestimmter Tag-Namen voraus (*div/span/p/label usw.*) daher müssen wir Widgets bereitstellen, die auf diese Namen

reagieren. (*siehe die nächste Seite*).



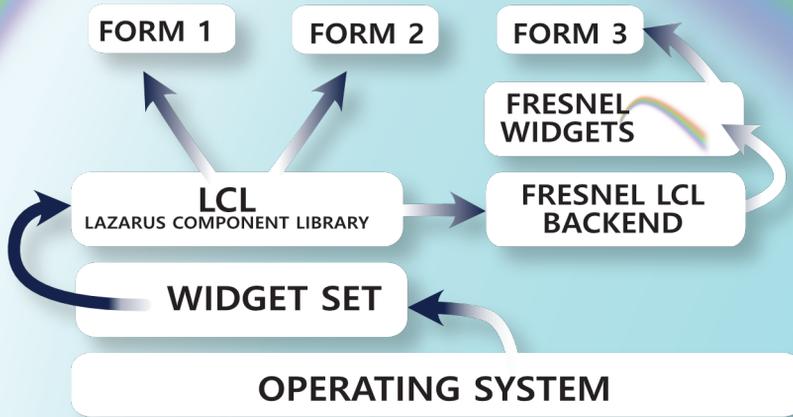


Abbildung 1: Eine Anwendung mit gemischten Widgetsets

Die Arbeit daran wurde bereits begonnen:

Ein CSS-Parser und ein Resolver wurden erstellt. Ein auf der LCL basierendes Zeichen-Backend wurde gestartet, und ein erster Layouter wurde geschrieben.

Eine CSS-Regel kann Elemente auf der Grundlage von Tag-Namen, Element-IDs, (*Pseudo*-)Attributen und Tag-Hierarchien auswählen.

Für Objekt-IDs scheint es nahe liegend, den Komponenten-namen als Attribut zu verwenden.

Damit entfällt die Notwendigkeit des HTML-Äquivalents von `GetElementById`, da jedes Element eine im Formular definierte Komponente ist.

Eine Tag-Hierarchie lässt sich natürlich auf eine Eltern-Kind-Beziehung abbilden, wie sie in der VCL/LCL existiert.

Die Zuordnung zwischen Tag-Namen und Attributen ist in einer Schnittstelle abstrahiert worden.

Es ist dem Widget überlassen, welche Tags und Attribute es an die CSS-Engine meldet.

4 KONSEQUENZEN DER NICHTVERWENDUNG DES LCL

Die VCL oder LCL übernimmt eine Vielzahl von Aufgaben. Sie ist nicht nur für das Zeichnen der Steuerelemente zuständig, sondern bearbeitet auch Nachrichten, die vom Betriebssystem kommen: Mausnachrichten, Tastaturnachrichten, Drag & Drop. Auch die Zwischenablage wird bearbeitet.

Da die Fresnel-Widgets unabhängig von der LCL sein müssen, müssen all diese Dinge neu implementiert werden, auch wenn einige von der LCL wiederverwendet werden können.

Da die Fresnel-Widgets nicht dem LCL- und VCL-Erbe gehorchen müssen, bedeutet dies, dass wir das Nachrichten- und Ereignisbehandlungssystem neu gestalten können. Die LCL/VCL ist auf 1 Ereignishandler pro Nachricht beschränkt. Nachrichten werden im Windows-Stil unter Verwendung von 2 Ganzzahlen gesendet. Früher war es aufgrund der Leistungsfähigkeit der Computer notwendig, ein so begrenztes System zu verwenden.

Heute ist diese Einschränkung nicht mehr nötig (wie der Browser beweist).

Die Fresnel Widgets werden dies neu definieren: verwenden Sie Datensätze (wie in FPGui) oder Objekte (wie z.B. in Javascript)

um die Nachrichten zu repräsentieren, erlauben mehrere Event-Handler für ein bestimmtes Ereignis, erlauben aber dennoch, dass Event-Handler im Object Inspector festgelegt werden können.

Da die LCL als ein mögliches Backend verwendet werden soll, muss natürlich ein Übersetzungsmechanismus zwischen LCL-Nachrichten und den neuen Fresnel-Nachrichten implementiert werden.



5 KOEXISTENZ MIT DEM LCL

Die IDE muss in der Lage sein, die Fresnel-Widgets zu zeichnen und zu manipulieren. die Fresnel-Widgets zeichnen und bearbeiten können. Da die IDE auf der LCL aufbaut, muss es eine Brücke zwischen der LCL und den Fresnel-Widgets geben:

Die Fresnel Widgets sollten in der Lage sein, die LCL als Zeichen-Backend zu verwenden. Dies deckt sich mit einer architektonischen Anforderung, die zu Beginn gestellt wurde:

Die LCL und Fresnel müssen in einer einzigen Anwendung koexistieren können.

Die Idee für diese Anforderung entstand zum Teil aus der Notwendigkeit, die Widgets in der IDE manipulieren zu können, vor allem aber, um die Portierung bestehender Anwendungen auf das neue Widget-Framework zu ermöglichen.

Für viele Anwendungen wäre die Notwendigkeit, die gesamte Anwendung auf einmal zu portieren, einfach eine zu große Hürde, um sie zu nehmen. Die daraus resultierende Anwendungsstruktur wäre anfangs etwas komplizierter: Fresnel Widgets auf einem LCL-Backend, wobei das Backend auf LCL-Komponenten basiert und die LCL-Komponenten auf einem Widgetset aufgebaut sind. Diese Situation wurde in *Abbildung 1 auf Seite 3* dargestellt.

Aber das Endziel ist natürlich eine einfachere Anwendungsstruktur: Fresnel Widgets auf einem Backend, das direkt auf dem Betriebssystem basiert. Diese Situation ist in *Abbildung 2 auf Seite 5 des Artikels* dargestellt.

6 WARUM NICHT EINFACH DEN BROWSER?

Da wir Widgets auf der Grundlage von CSS neu konzipieren, schaffen wir ein neues Nachrichtenmodell, das einige Merkmale des Javascript DOM aufweist: warum nicht einfach den Browser oder ein System wie Electron (*ein paketiierter Browser*) oder Chromium verwenden?

Als Pascal-Programmierer können Sie diese Plattformen schließlich mit **Pas2js** oder **TMS Web Core** ansteuern.

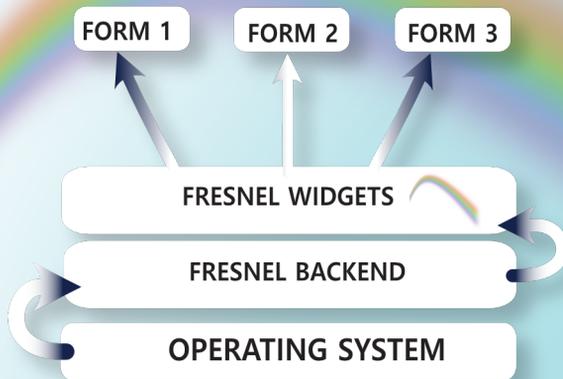
Der Browser, so leistungsfähig er auch sein mag, ist immer noch eine begrenzte Umgebung mit eigenen Regeln. Zweifellos glänzt der Browser bei der UI-Entwicklung - weshalb Project Fresnel CSS als Mechanismus verwendet - aber native Anwendungen sind immer noch schneller, und als Programmierer genießen Sie mehr Freiheiten als in einer Browserumgebung: Zugriff auf das Dateisystem und andere Hardware. Außerdem schreiben die Browser für viele Operationen ein **asynchrones** Modell vor, was es für Programmieranfänger etwas schwieriger macht.

Darüber hinaus ist die Portierung auf den Browser eine Alles-oder-Nichts-Entscheidung: Es ist nicht möglich, einen Teil der Anwendung im Browser und einen Teil als native Anwendung zu haben. Es ist natürlich möglich, eine Anwendung Form für Form in eine Browser-Anwendung umzuschreiben, aber die Interaktion zwischen dem nativen und dem Browser-Teil wird sehr begrenzt sein.

Die Tatsache, dass Project Fresnel CSS verwendet, bedeutet auch nicht, dass es auf HTML-Elemente beschränkt ist: Es wird immer noch möglich sein, eigene Widgets zu erstellen, aber Sie werden den CSS-Mechanismus abfragen müssen, um mehr über Ränder, Farben und solche Eigenschaften zu erfahren.

Das Ziel ist es auch nicht, einen Browser zu erstellen. Es wäre zwar möglich, eine HTML-Datei zu "importieren" und ein Formular auf der Grundlage der HTML-Tags zu erstellen, aber die Erstellung eines vollwertigen Browsers ist nicht das Ziel.





7 FAZIT UND DER WEITERE WEG

Projekt Fresnel ist ein großes Projekt zur Verjüngung der UI-Möglichkeiten von Pascal:

Es liegt eine Menge Arbeit vor uns. Aber das Projekt ist auf den Weg gebracht worden, und die erste Ergebnisse wurden erzielt: ein Proof of Concept mit der LCL als Backend wurde erstellt (siehe Abbildung 3 auf Seite 6). Der Code ist in einem Git-Repository auf Gitlab zu finden:

<https://gitlab.com/freepascal.org/lazarus/fresnel>

Aufgrund der Abhängigkeit vom CSS-Framework benötigt es derzeit FPC-Trunk zum Kompilieren, aber das kann sich in Zukunft ändern, so dass die Release-Version von FPC verwendet werden kann (und, wer weiß? Delphi...).

Die Free Pascal und Lazarus-Foundation glaubt fest an dieses Projekt und wird alles tun, um die Entwicklung dieses Widgetsets zu unterstützen. Aber wie gesagt, es handelt sich um ein großes Projekt, und der Erfolg des Projekts wird von der Akzeptanz und der Unterstützung der gesamten Pascal-Gemeinschaft abhängen.

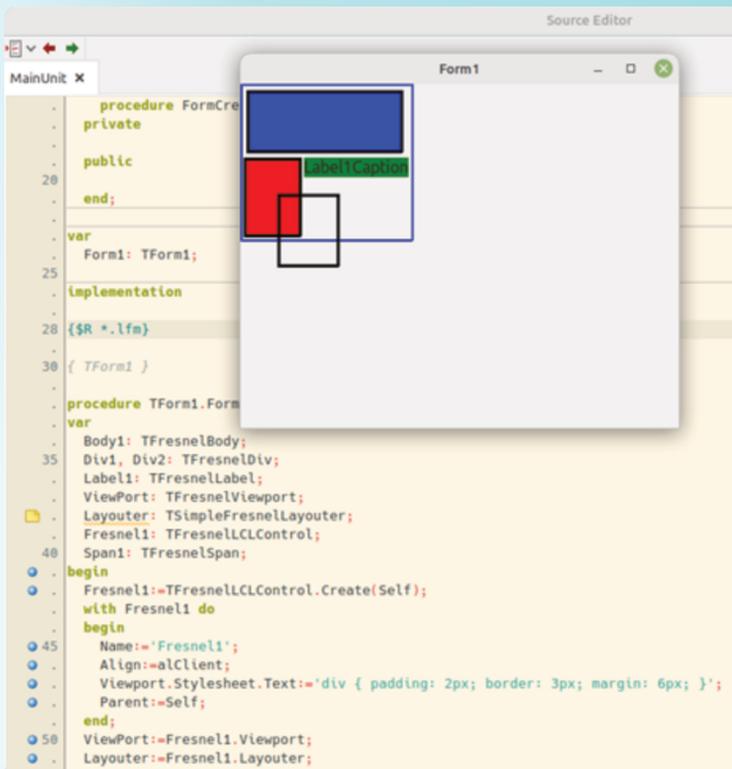
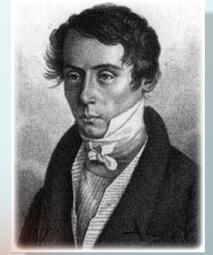


Abbildung 3: Der Nachweis des Konzepts



AUGUSTIN-JEAN FRESNEL

(10. Mai 1788 - 14. Juli 1827)



war ein französischer Bauingenieur und Physiker, dessen Forschungen auf dem Gebiet der Optik dazu führten, dass ab den späten 1830er Jahren bis zum Ende des 19. Jahrhunderts fast einstimmig die Wellentheorie des Lichts akzeptiert wurde, die jeden Rest der Newtonschen Korpuskeltheorie ausschloss.

(In der Optik besagt die Korpuskeltheorie des Lichts, dass das Licht aus kleinen diskreten Teilchen besteht, die "Korpuskeln" (kleine Teilchen) genannt werden, die sich mit einer endlichen Geschwindigkeit in einer geraden Linie bewegen und einen Impuls besitzen. Dies basierte auf einer alternativen Beschreibung des Atomismus der damaligen Zeit).

Er ist vielleicht besser bekannt für die Erfindung der **katadioptrischen*** (reflektierenden/brechenden) Fresnel-Linse und für seine Pionierarbeit bei der Verwendung von "Stufenlinsen", um die Sichtbarkeit von Leuchttürmen zu verbessern und damit unzählige Menschenleben auf See zu retten.



WIKIPEDIA

**[Ein katadioptrisches optisches System ist ein optisches System, bei dem Brechung und Reflexion kombiniert werden, in der Regel über Linsen (Dioptrien) und gebogene Spiegel (Katoptrien). Katadioptrische Kombinationen werden in Fokussierungssystemen wie Suchscheinwerfern, Scheinwerfern, frühen Leuchtturm-Fokussierungssystemen, optischen Teleskopen, Mikroskopen und Teleobjektiven verwendet. Andere optische Systeme, die Linsen und Spiegel verwenden, werden ebenfalls als "katadioptrisch" bezeichnet, z.B. katadioptrische Überwachungssensoren.]*

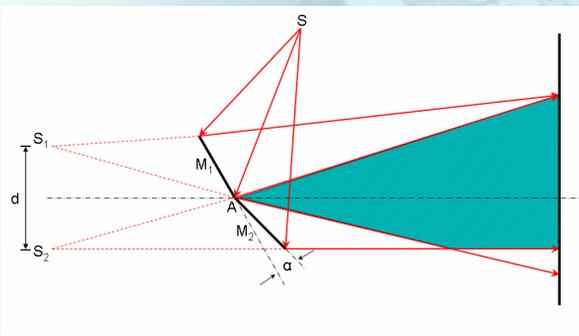
Die einfachere dioptrische (rein brechende) Stufenlinse, die zuerst vom **Grafen Buffon** vorgeschlagen und von Fresnel unabhängig davon neu erfunden wurde, wird in Bildschirmen und in Kondensoren für Overhead-Projektoren verwendet.

Indem er das Huygens'sche Prinzip der Sekundärwellen und das Young'sche Prinzip der Interferenz quantitativ ausdrückte und annahm, dass einfache Farben aus sinusförmigen Wellen bestehen, gab Fresnel die erste zufriedenstellende Erklärung der Beugung an geraden Kanten, einschließlich der ersten zufriedenstellenden wellenbasierten Erklärung der geradlinigen Ausbreitung.

Ein Teil seiner Argumentation war der Beweis, dass die Addition von Sinusfunktionen mit derselben Frequenz, aber unterschiedlichen Phasen analog zur Addition von Kräften mit unterschiedlichen Richtungen ist.

Unter der Annahme, dass Lichtwellen rein transversal sind, erklärte Fresnel die Natur der Polarisation, den Mechanismus der chromatischen Polarisation und die Transmissions- und Reflexionskoeffizienten an der Grenzfläche zwischen zwei transparenten isotropen Medien.

Durch die Verallgemeinerung der Richtungs-Geschwindigkeits-Polarisations-Beziehung für Kalzit erklärte er dann die Richtungen und Polarisationen der gebrochenen Strahlen in doppelt brechenden Kristallen der biaxialen Klasse (für die die sekundären Wellenfronten von Huygens nicht achsensymmetrisch sind). Zwischen der ersten Veröffentlichung seiner Hypothese der reinen Transversalwellen und der Vorlage seiner ersten korrekten Lösung für das biaxiale Problem lag weniger als ein Jahr.



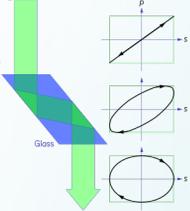
Der Fresnel-sche Doppelspiegel (1816).

Die Spiegelsegmente M1 und M2 erzeugen virtuelle Bilder S1 und S2 des Spalts S.

Im schattierten Bereich überschneiden sich die Strahlen der beiden virtuellen Bilder und interferieren wie bei Young (oben).



Später prägte er die Begriffe lineare Polarisation, zirkuläre Polarisation und elliptische Polarisation, erklärte wie die optische Rotation als Unterschied in den Ausbreitungsgeschwindigkeiten für die beiden Richtungen der zirkulären Polarisation verstanden werden kann, und erklärte (indem er zuließ, dass der Reflexionskoeffizient komplex ist) die Änderung der Polarisation aufgrund der internen Totalreflexion, wie sie in der Fresnel-schen Raute genutzt wird.



Eine Fresnel-Raute ist ein optisches Prisma, das durch zwei interne Totalreflexionen eine Phasendifferenz von 90° zwischen zwei senkrecht zueinander stehenden Komponenten der Polarisation erzeugt. Reflexionen. Wenn der einfallende Strahl unter 45° zur Einfallsebene linear polarisiert ist, ist der austretende Strahl zirkular polarisiert und andersherum. Wenn der einfallende Strahl unter einer anderen Neigung linear polarisiert ist, ist der austretende Strahl elliptisch polarisiert mit einer Hauptachse in der Reflexionsebene, und vice versa.
https://en.wikipedia.org/wiki/Fresnel_rhomb

Die Verfechter der etablierten korpuskularen Theorie konnten mit seinen quantitativen Erklärungen für so viele Phänomene auf der Grundlage so weniger Annahmen nicht mithalten.

Licht oder sichtbares Licht ist elektromagnetische Strahlung, die vom menschlichen Auge wahrgenommen werden kann. Sichtbares Licht wird in der Regel als Licht mit Wellenlängen im Bereich von 400-700 Nanometern (nm) definiert, was Frequenzen von 750-420 Terahertz entspricht und zwischen dem Infrarot (mit längeren Wellenlängen) und dem Ultraviolett (mit kürzeren Wellenlängen) liegt.

In der Physik kann sich der Begriff "Licht" im weiteren Sinne auf elektromagnetische Strahlung jeder Wellenlänge beziehen, ob sichtbar oder nicht. In diesem Sinne sind auch Gammastrahlen, Röntgenstrahlen, Mikrowellen und Radiowellen Licht.

Die wichtigsten Eigenschaften von Licht sind Intensität, Ausbreitungsrichtung, Frequenz oder Wellenlängenspektrum und Polarisation. Seine Geschwindigkeit im Vakuum, 299 792 458 Meter pro Sekunde (m/s), ist eine der Grundkonstanten der Natur. Wie alle Arten von elektromagnetischer Strahlung breitet sich sichtbares Licht durch massefreie Elementarteilchen aus, die Photonen genannt werden und die Quanten des elektromagnetischen Feldes darstellen und sowohl als Wellen als auch als Teilchen analysiert werden können. Das Studium des Lichts, das als Optik bezeichnet wird, ist ein wichtiges Forschungsgebiet der modernen Physik. Die Hauptquelle des natürlichen Lichts auf der Erde ist die Sonne. Eine weitere wichtige Lichtquelle für den Menschen war in der Vergangenheit das Feuer, vom antiken Lagerfeuer bis zur modernen Petroleumlampe. Mit der Entwicklung von elektrischem Licht und Stromsystemen hat die elektrische Beleuchtung das Feuerlicht effektiv ersetzt.

Fresnel kämpfte sein Leben lang mit der Tuberkulose, der er im Alter von 39 Jahren erlag. Obwohl er zu Lebzeiten nicht zu einer öffentlichen Berühmtheit wurde, lebte er gerade lange genug, um die gebührende Anerkennung von seinen Kollegen, darunter (auf dem Sterbebett) die **Rumford-Medaille der Royal Society of London**, und sein Name ist in der modernen Terminologie der Optik und der Wellen allgegenwärtig.



Rotierende katadioptrische Fresnellinse erster Ordnung, datiert 1870, ausgestellt im Musée national de la Marine, Paris. In diesem Fall sind die dioptrischen Prismen (innerhalb der Bronzeringe) und die katadioptrischen Prismen (außen) so angeordnet, dass sie das Licht der zentralen Lampe in vier sich drehende Strahlenbündeln, die von den Seeleuten als vier Blitze pro Umdrehung wahrgenommen werden. Der Aufbau ist 2,54 Meter hoch und wiegt etwa 1,5 Tonnen..



Nachdem die Wellentheorie des Lichts in den 1860er Jahren in Maxwells elektromagnetischer Theorie aufgegangen war, wurde die Aufmerksamkeit von der Bedeutung von Fresnels Beitrag abgelenkt. In der Zeit zwischen Fresnels Vereinheitlichung der physikalischen Optik und Maxwells umfassenderer Vereinheitlichung beschrieb eine zeitgenössische Autorität, Humphrey Lloyd, Fresnels Transversalwellentheorie als "das edelste Gewebe, das jemals den Bereich der physikalischen Wissenschaft geschmückt hat, Newtons System des Universums allein ausgenommen".





Starter

Expert

EINLEITUNG

Benutzer von Lazarus kennen diesen Satz auswendig: "Einmal schreiben, überall kompilieren". Aber FPC und Lazarus ermöglichen auch "Write Once, Compile [Once/Many], Run Anywhere". Letzteres wird auch als Cross-Compiling bezeichnet. Und es ist eine einzigartige Funktion von FPC und Lazarus, die bei anderen Entwicklungstools kaum oder gar nicht zu finden ist. Das einzige Problem bei der Cross-Kompilierung ist die steile Lernkurve bei der Einrichtung von FPC und Lazarus, um Cross-Kompilierung zu ermöglichen.

**Write Once,
Compile [Once/Many],
Run Anywhere**

Es gibt ein Wiki:

https://wiki.freepascal.org/Cross_compiling

Dieses Wiki sollte alle benötigten Informationen enthalten. In der Praxis zeigt sich jedoch, dass es immer noch nicht einfach ist, diese Anweisungen zu befolgen, um eine funktionierende Einrichtung zu erhalten.

Hier kommt **fpcupdeluxe** zur Rettung! Es ist nicht vielen bekannt, aber der Vorläufer von fpcupdeluxe, nämlich fpcup, wurde hauptsächlich von seinem Entwickler **Reinier Olislagers** (auch bekannt als *Forumsmitglied BigChimp*) entwickelt, um die Last der Cross-Kompilierung zu erleichtern. Und diese Funktion wurde durch fpcupdeluxe erheblich erweitert.

Um fpcupdeluxe als Setup-Tool für die Cross-Kompilierung verwenden zu können, **MÜSSEN** Sie es zunächst für die Installation eines nativen FPC und Lazarus verwenden! Dies wurde bereits in einem früheren Artikel beschrieben: Blaise Ausgabe Nummer 106 siehe Artikelseite. Bitte lesen Sie also diesen Artikel, um Informationen über die erste Verwendung von fpcupdeluxe zu erhalten.





Das Bild auf der vorherigen Seite (1 dieses Artikels) zeigt den Cross -
Kompilierungs Bildschirm von **fpcupdeluxe**. Wie Sie sehen können, handelt
es sich um die Win32-Version von fpcupdeluxe, die zuvor zur Installation von FPC und Lazarus für Win32 verwendet
wurde. Die Auswahl der CPU und des Betriebssystems zielt auf Linux für 64 Bit Intel (linux-x86_64).
Die Installation des ausgewählten Cross-Compilers ist nun so einfach wie das Drücken der entsprechenden
Schaltfläche.

Aber !!!

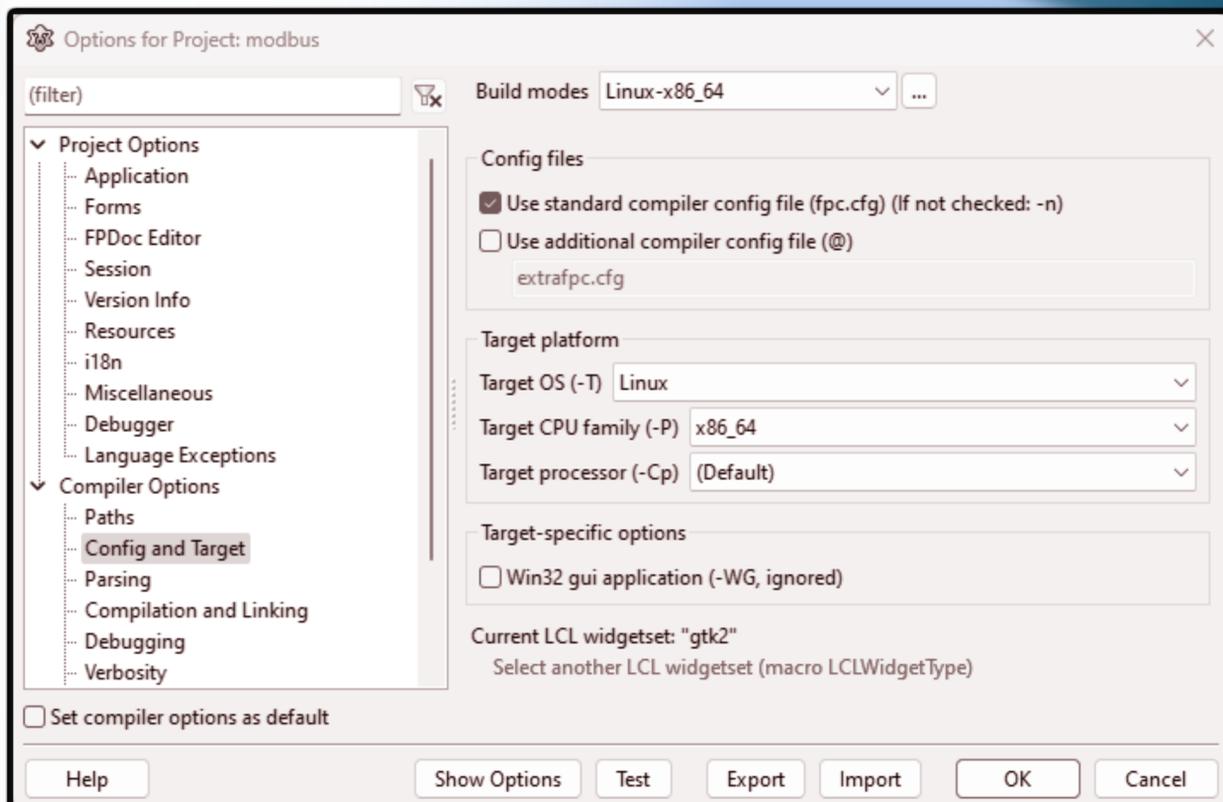
In den meisten Fällen schlägt es fehl, nachdem Sie die Quellen bereinigt und die Systemanforderungen überprüft
haben. Und das ist auch zu erwarten: Für die Cross-Kompilierung werden unter Umständen Binärwerkzeuge und
Systembibliotheken benötigt. Binäre Tools wie ein Assembler und ein Linker für das Zielsystem. Bibliotheken wie die,
die auf dem Zielsystem selbst zu finden sind.
Und auch hier kommt fpcupdeluxe zur Rettung!





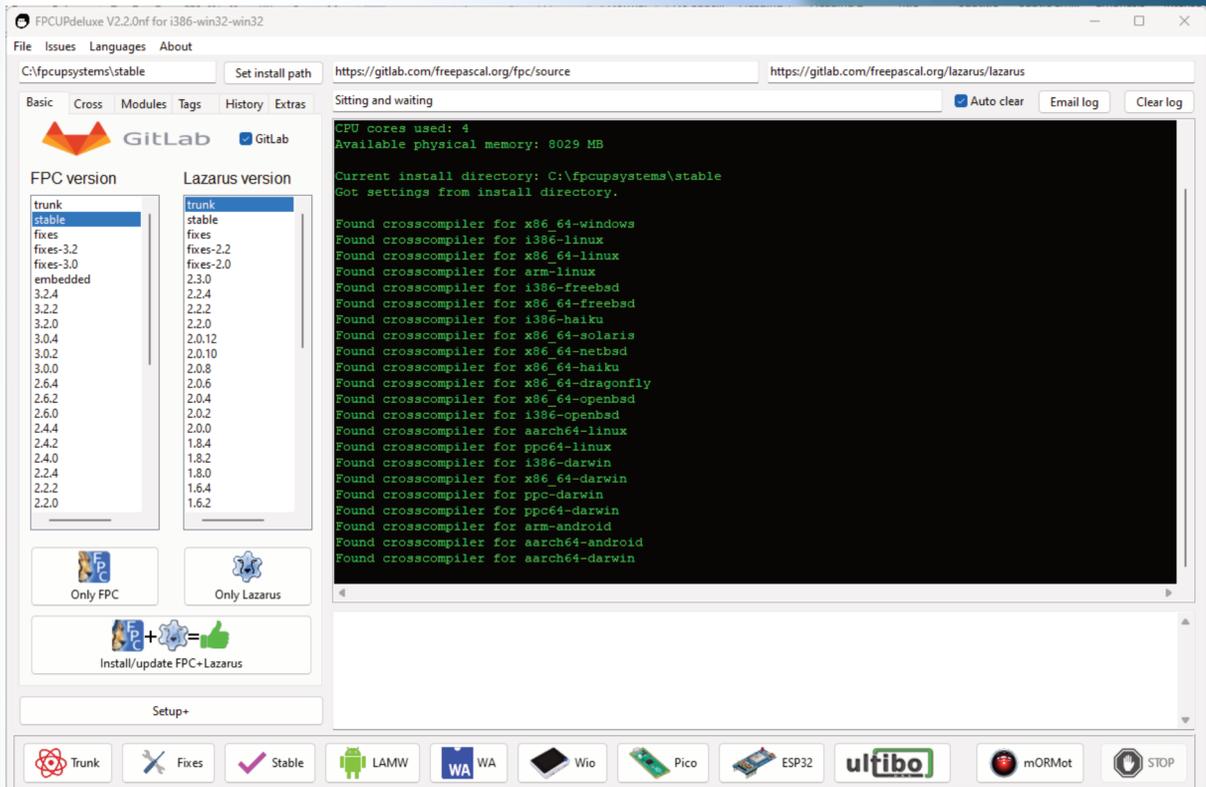
Auch hier werden für die Cross-Kompilierung geeignete Tools und Bibliotheken benötigt werden. fpcupdeluxe kann diese herunterladen und an der richtigen Stelle installieren, wo sie vom FPC-Cross-Compiler gefunden und verwendet werden können.

Klicken Sie also auf Ja, um das Herunterladen und die Installation zuzulassen, und der Prozess der Erstellung und Installation eines Cross-Compilers wird fortgesetzt. Wenn alles wie erwartet heruntergeladen und erstellt wird, wird der Prozess rechtzeitig abgeschlossen und der Cross-Compiler ist einsatzbereit.



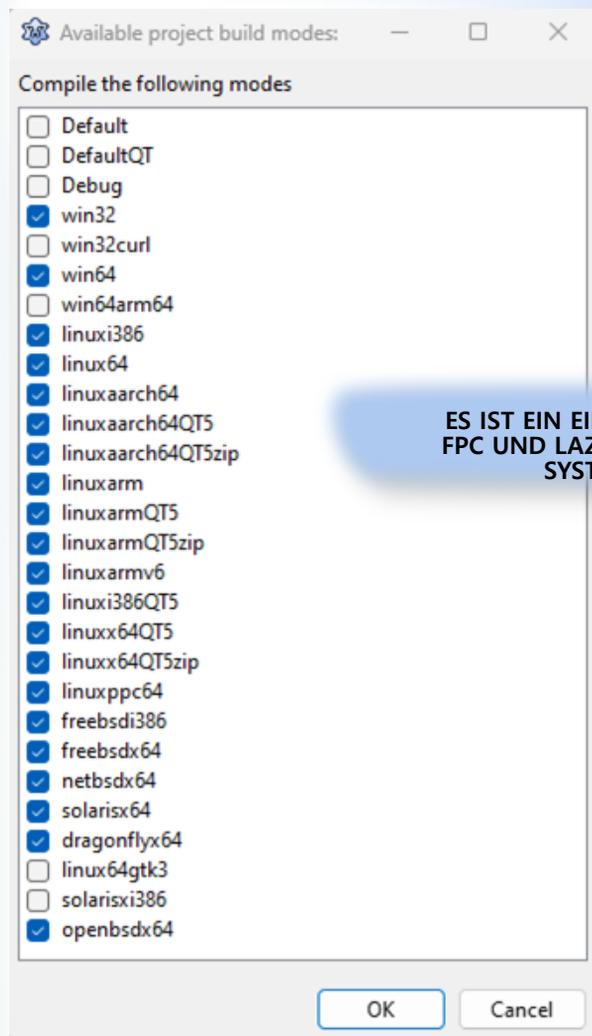
Um den Cross-Compiler verwenden zu können, wählen Sie die entsprechende CPU und das Betriebssystem als Ziel in Lazarus. Sie sollten nun in der Lage sein, Ihr eigenes Programm von Ihrem Windows-System zu Ihrem neuen Ziel zu crosskompilieren: Linux 64bit. Es wird eine Binärdatei erzeugt, die auf das Zielsystem kopiert und dort ausgeführt werden kann!





Die Entwicklung und Bereitstellung von fpcupdeluxe selbst wäre ohne Cross-Compiling unmöglich gewesen. Der Screenshot zeigt eine typische Installation für die Entwicklung und Erstellung von fpcupdeluxe. Es werden viele Cross-Compiler verwendet, um alle Systeme zu erreichen, für die fpcupdeluxe geeignet ist. Und nun zurück zum zweiten Anspruch dieses Artikels: Einmal schreiben, [einmal/viele] kompilieren, überall ausführen.





**ES IST EIN EINZIGARTIGES MERKMAL VON
FPC UND LAZARUS DASS SIE FÜR SO VIELE
SYSTEME GEEIGNET SIND**

Lazarus kann für viele Ziele auf einmal kompilieren!
Der Menüpunkt heißt:

Run → Compile many Modes.

Auch hier zeigt der Screenshot oben eine typische
fpcupdeluxe-Einstellung.

Wenn Sie auf Ok drücken, wird für alle ausgewählten
Ziele gleichzeitig kompiliert. Eine echte (und
einzigartige) Umsetzung des zweiten Anspruchs.

Dieser Artikel basiert auf einer Windows-Installation.
Die gleichen Schritte gelten auch für alle anderen
Systeme.

Sie können auch die Linux-Version von fpcupdeluxe
verwenden, um FPC und Lazarus unter Linux zu
installieren und für Windows zu kompilieren.

Sie können für den Raspberry Pi crosskompilieren. Für
Android. Für OSX.

Für eingebettete Systeme wie den ESP32. Für Exoten
wie Haiku oder BSD-Linux.
Für Web-Assembly.

Es ist eine einzigartige Eigenschaft von FPC und Lazarus,
dass sie für so viele Systeme geeignet sind. Und diese
Ziele durch Cross-Kompilierung leicht zu erreichen sind.

Weitere Fragen :

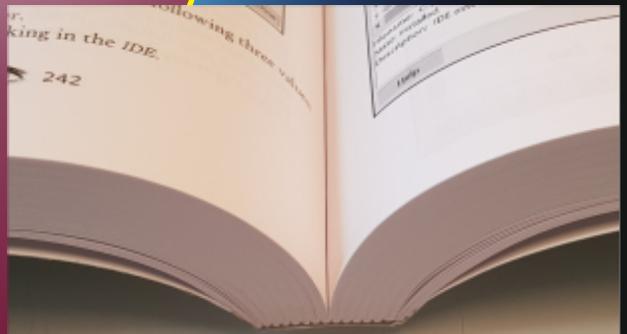
[https://forum.lazarus.freepascal.org/
index.php/topic,34645.0.html](https://forum.lazarus.freepascal.org/index.php/topic,34645.0.html)



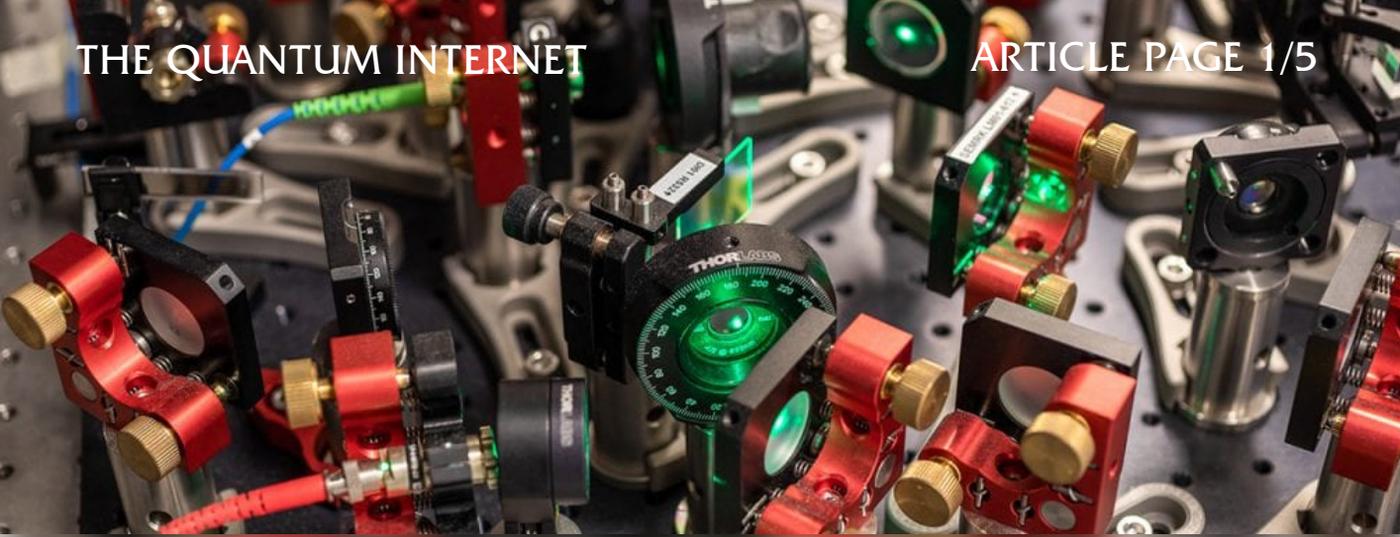
SALE LAZARUS HANDBOOK POCKET

Preis: € 26,50
Exklusive
Mehrwertsteuer
und Versand

- Englisch Sprache
- Schwarz und weiß gedruckt
- Rückseite genäht
- 2 Teilig
- Inklusive PDF
- 934 Seiten
- Gewicht: 2kg
- Mit 40 Beispielen
Projekte und
zusätzliche Programme



BlaisePascalMagazine
PDF-Betrachter enthalten



Ein Team von Forschern des niederländischen Unternehmens QuTech (eine Ausgründung der Technischen Universität Delft und des TNO) hat das erste Quantennetzwerk mit mehreren Knotenpunkten realisiert, das drei Quantenprozessoren verbindet. Darüber hinaus haben sie einen Grundsatzbeweis für wichtige Quantennetzwerkprotokolle erbracht. Ihre Ergebnisse sind ein wichtiger Meilenstein auf dem Weg zum zukünftigen Quanteninternet und wurden jetzt in Science veröffentlicht

DAS QUANTENINTERNET

Dank der Leistungsfähigkeit des Internets können zwei beliebige Computer auf der Erde miteinander verbunden werden. Dies ermöglicht Anwendungen, von denen man zur Zeit seiner Entstehung vor Jahrzehnten noch nicht einmal zu träumen wagte. Heute arbeiten Forscher in vielen Labors auf der ganzen Welt an ersten Versionen eines Quanteninternets (ein Netzwerk, das zwei beliebige Quantengeräte wie Quantencomputer oder Sensoren über große Entfernungen miteinander verbinden kann).

Während das heutige Internet Informationen in Bits verteilt (die entweder 0 oder 1 sein können), wird ein zukünftiges Quanteninternet Quantenbits verwenden, die gleichzeitig 0 und 1 sein können.

Ein Quanteninternet wird eine Reihe neuartiger Anwendungen eröffnen, von unhackbarer Kommunikation und Cloud Computing mit vollständigem Schutz der Privatsphäre der Nutzer bis hin zu hochpräziser Zeitmessung", so Matteo Pompili, Doktorand und Mitglied des Forschungsteams. Und wie beim Internet vor 40 Jahren (Das Arpa-net) gibt es wahrscheinlich viele Anwendungen, die wir uns nie hätten träumen lassen.

AUF DEM WEG ZUR ALLGEGENWÄRTIGEN KONNEKTIVITÄT

Die ersten Schritte in Richtung eines Quanteninternets wurden im letzten Jahrzehnt unternommen, indem zwei Quantengeräte miteinander verbunden wurden, die eine direkte physikalische Verbindung hatten.

Um ein skalierbares Quantennetzwerk zu schaffen, ist es jedoch unerlässlich, dass über Zwischenknoten (analog zu Routern im klassischen Internet) weitergeleitet werden können.

Außerdem beruhen viele vielversprechende Anwendungen des Quanteninternets auf verschränkten Quantenbits, die zwischen mehreren Knotenpunkten verteilt werden sollen.

Verschränkung ist ein auf der Quantenskala zu beobachtendes Phänomen, das Teilchen bei kleinen und sogar bei großen Entfernungen grundsätzlich miteinander verbindet.

Sie verleiht Quantencomputern ihre enorme Rechenleistung und ist die grundlegende Ressource für den Austausch von Quanteninformationen über das zukünftige Quanteninternet. Mit der Realisierung ihres Quantennetzwerks im Labor ist es einem Forschungsteam von QuTech erstmals gelungen, zwei Quantenprozessoren über einen Zwischenknoten zu verbinden und eine gemeinsame Verschränkung zwischen mehreren eigenständigen Quantenprozessoren herzustellen.

BETRIEB DES QUANTENNETZWERKS

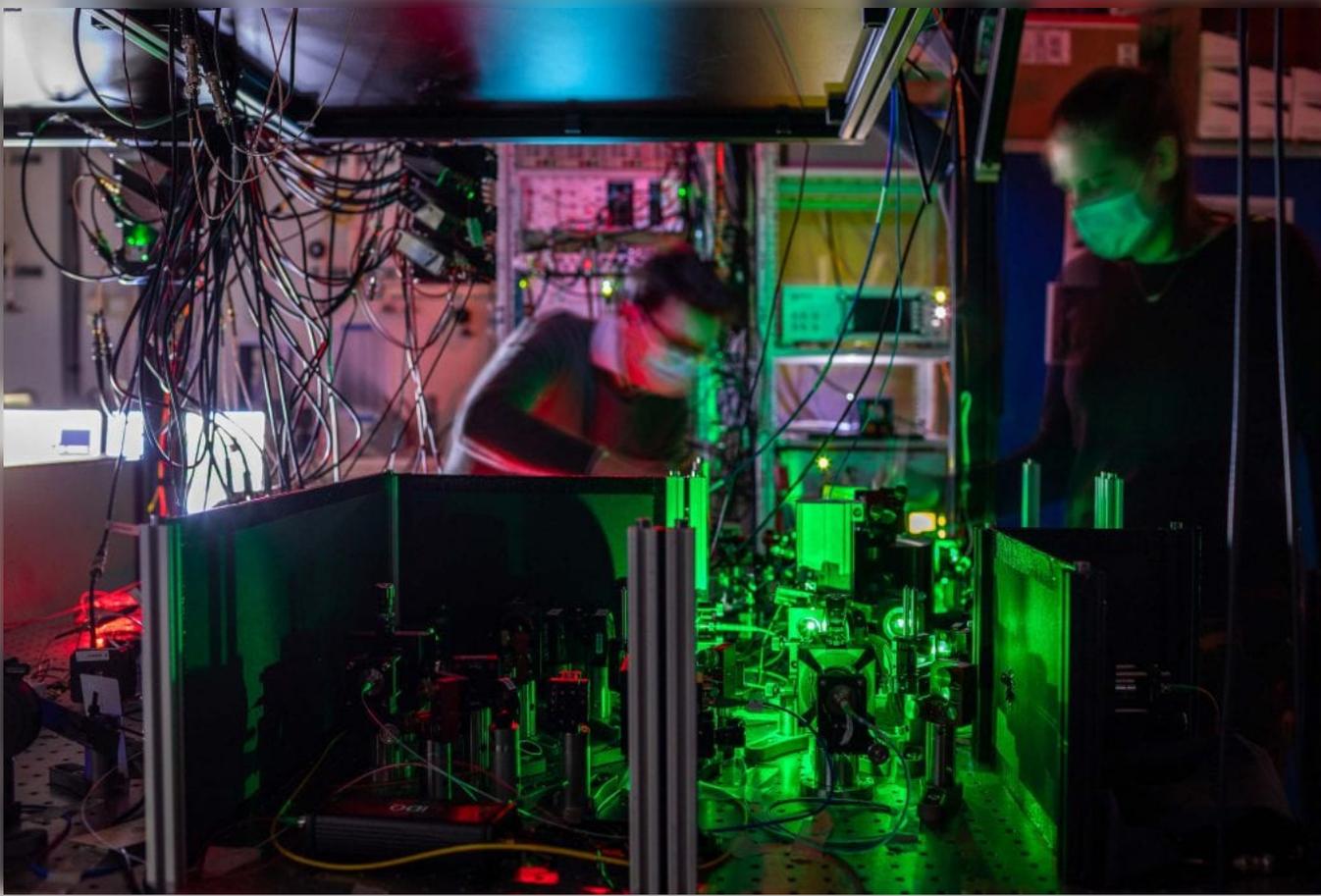
Das rudimentäre Quantennetzwerk besteht aus drei Quantenknoten, die sich in einiger Entfernung innerhalb desselben Gebäudes befinden. Damit diese Knoten als echtes Netzwerk funktionieren, mussten die Forscher eine neuartige Architektur entwickeln, die eine Skalierung über eine einzelne Verbindung hinaus ermöglicht.

Der mittlere Knoten (Bob genannt) hat eine physische Verbindung zu den beiden äußeren Knoten (Alice und Charlie genannt), so dass Verschränkungsverbindungen mit jedem dieser Knoten hergestellt werden können.

Bob ist mit einem zusätzlichen Quantenbit ausgestattet, das als Speicher verwendet werden kann, so dass eine zuvor erzeugte Quantenverbindung gespeichert werden kann, während eine neue Verbindung hergestellt wird.

Nachdem die Quantenverbindungen Alice-Bob und Bob-Charlie hergestellt wurden, werden diese Verbindungen durch eine Reihe von Quantenoperationen bei Bob in eine Quantenverbindung Alice-Charlie umgewandelt. Alternativ können Sie auch eine andere Reihe von Quantenoperationen bei Bob durchführen, um die Verschränkung zwischen allen drei Knotenpunkten herzustellen.



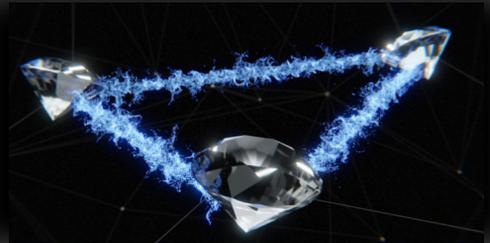


Forscher arbeiten an einem der Quanten Netzwerkknoten, wo Spiegel und Filter die Laserstrahlen auf den Diamantchip leiten.

BEREIT FÜR DIE SPÄTERE VERWENDUNG

Ein wichtiges Merkmal des Netzwerks ist, dass es die den erfolgreichen Abschluss dieser (*von Natur aus wahrscheinlichstmöglich*) Protokolle mit einem "Flaggen"-Signal ankündigt. Eine solche Ankündigung ist entscheidend für die Skalierbarkeit, denn in einem zukünftigen Quanteninternet viele solcher Protokolle verkettet werden müssen verkettet werden müssen.

Sobald sie etabliert waren, konnten wir die verschränkten Zustände zu bewahren und sie vor Rauschen zu schützen", sagt Sophie Hermans, ein weiteres Mitglied des Teams. Das bedeutet, dass wir diese Zustände im Prinzip für die Verteilung von Quantenschlüsseln, eine Quantenberechnung oder jedes andere Quantenprotokoll verwenden können.

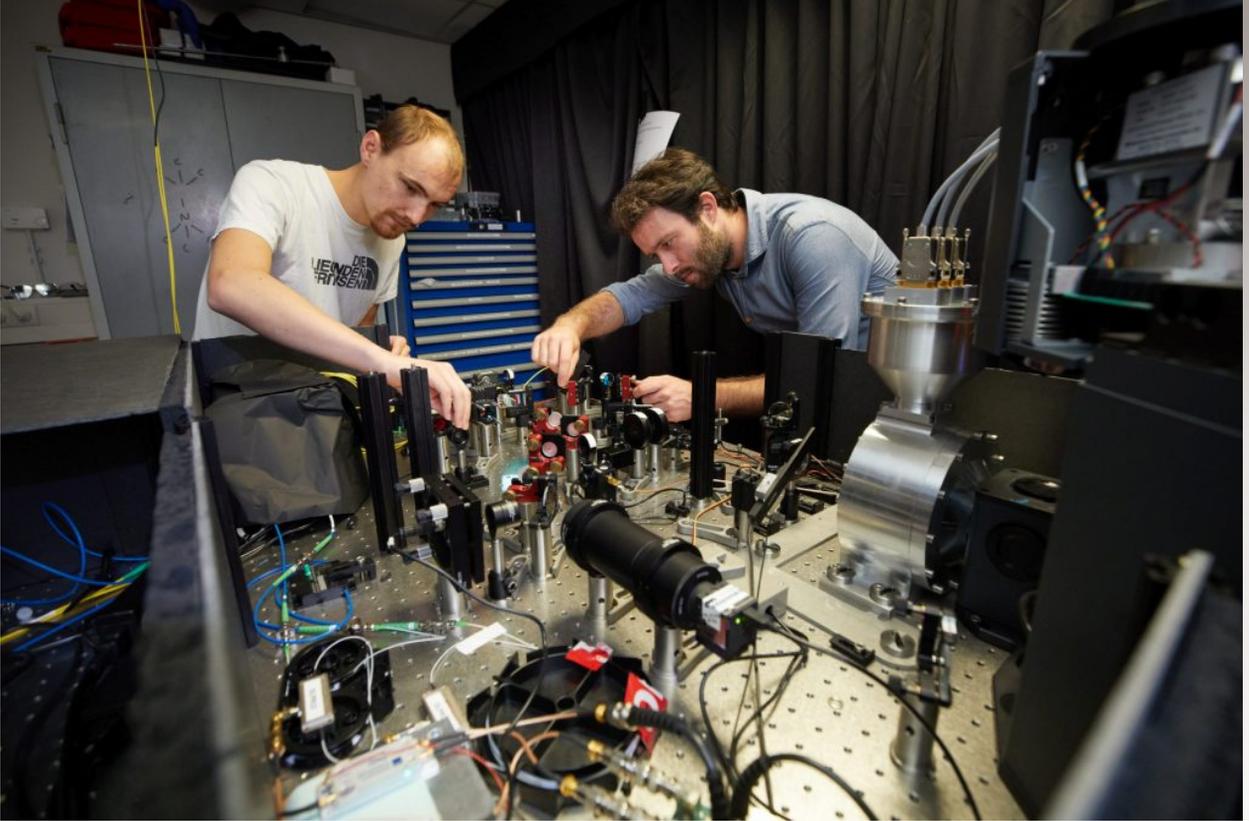


QUANTEN-INTERNET-DEMONSTRATOR

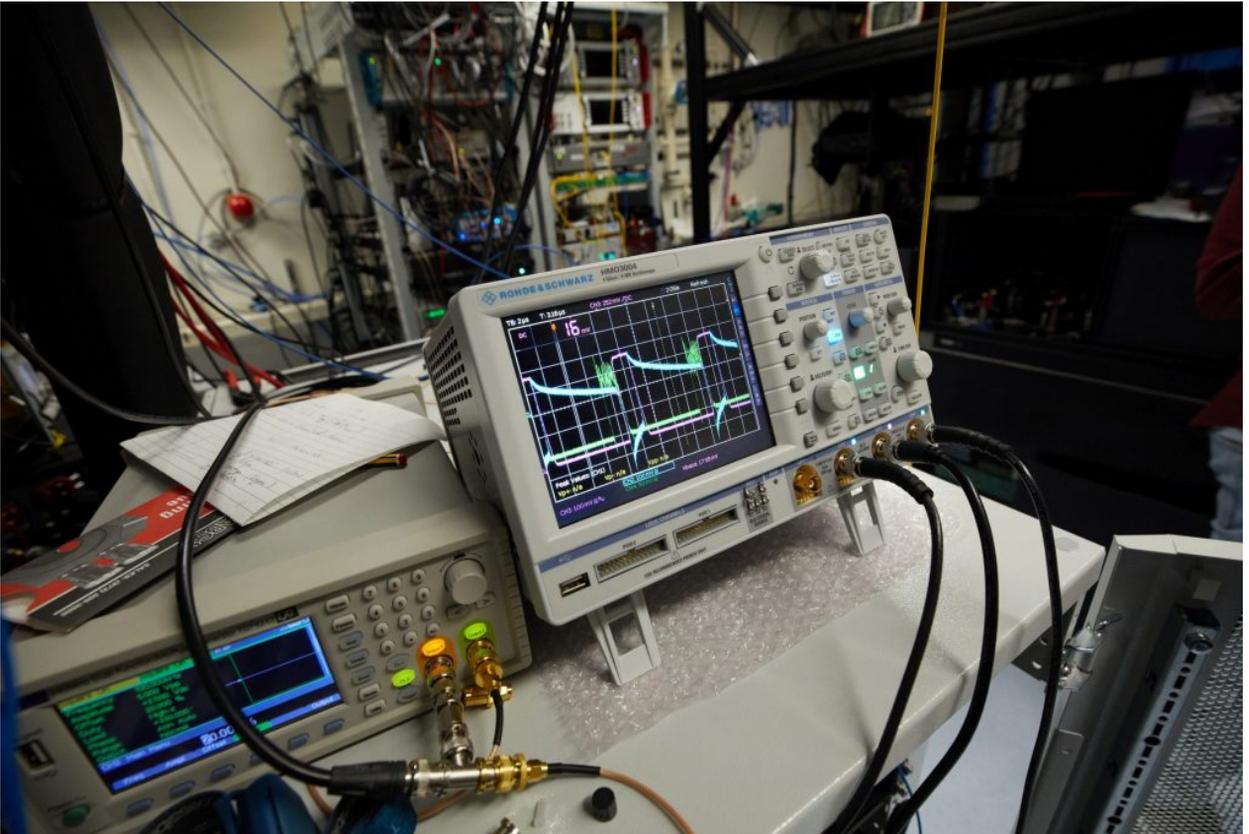
Dieses erste verschränkungs-basierte Quantennetzwerk bietet den Forschern ein einzigartiges Testfeld für die Entwicklung und Erprobung von Quanteninternet-Hardware, -Software und -Protokollen. Das zukünftige Quanteninternet wird aus unzähligen Quantengeräten und Zwischenknoten bestehen", sagt Ronald Hanson, der das Forschungsteam leitete. Die Kollegen bei QuTech untersuchen bereits die zukünftige Kompatibilität mit bestehenden Dateninfrastrukturen. Zu gegebener Zeit wird der derzeitige Proof-of-Principle-Ansatz außerhalb des Labors auf bestehenden Telekom-Glasfasern getestet werden - auf dem QuTech Quantum Internet Demonstrator, dessen erste Metropolenverbindung 2022 fertiggestellt werden soll.

SCHICHTEN AUF HÖHERER EBENE

Im Labor werden sich die Forscher darauf konzentrieren, mehr Quantenbits zu ihrem Drei-Knoten-Netzwerk hinzuzufügen und Software- und Hardware-Schichten auf höherer Ebene hinzuzufügen. "Sobald alle hochrangigen Kontroll- und Schnittstellenschichten für den Betrieb des Netzwerks entwickelt sind, wird jeder in der Lage sein, eine Netzwerkanwendung zu schreiben und zu betreiben, ohne zu wissen, wie Laser und Kryostate funktionieren. Das ist das Endziel."



Künstlerische Darstellung des Drei-Knoten-Quantennetzwerks. Kredit: Matteo Pompili für QuTech



LAZARUS HANDBOOK PDF

- English
- Black & white edition
- 934 Pages
- Electronic indexed
- Index of 240 pages
- List of chapters is completely clickable for moving to the page
- Including 40 Example projects and extra programs

LAZARUS 
HANDBOOK
FOR PROGRAMMING WITH FREE PASCAL AND LAZARUS

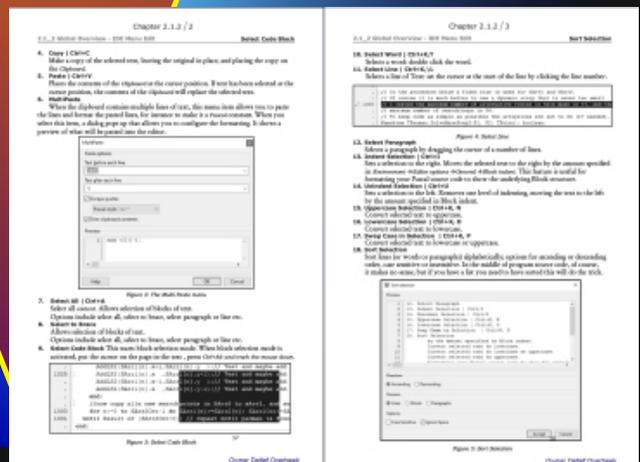


ELECTRONIC (PDF)

Michaël van Canneyt/Martin Friebe/Mattias Gärtner
Inoussa Ouedraogo/Detlef Overbeck/
Howard Page Clark/Werner Pamler

Owner: Detlef Overbeck

BlaisePascalMagazine
PDF viewer included





Starter

Expert

EINLEITUNG

FastReport hat seine eigene Version für Lazarus entwickelt für Lazarus und es ist eine sehr gute Idee, eine spezielle Version für Windows und Linux zu erstellen. Mac ist noch nicht verfügbar. Sobald dies geschieht, werden wir Sie darüber informieren.

Um es Ihnen als Leser leicht zu machen, geben wir Ihnen eine Anleitung, die Ihnen beim Auffinden und Installieren der Dateien hilft.

Wie Sie natürlich wissen, sind alle Betriebssysteme auf Lazarus verfügbar, aber für diese Anleitung bedeutet das, dass die Komponenten nur für Windows und / oder Linux geeignet sind.

Aus diesem Grund werde ich die normale Installation verwenden:

Unter Windows (*das ich hier zeige*) lautet der Pfad zu allen Dateien wie folgt:

```
c:\Program Files (x86)\FastReport VCL Enterprise\Source\
```

unter Linux ist es natürlich ganz anders und hängt davon ab, welche Version von Linux Sie haben. Wir werden daher in der nächsten Ausgabe eine Linux-Installation vorstellen. Wie bereits erwähnt, benötigen Sie eine Lizenz von FastReport, entweder eine professionelle oder eine Testversion. Diese können Sie ganz einfach herunterladen unter

```
https://www.fast-report.com/en/download/public\_files/267/
```

Sie liefert eine exe-Datei der Testversion.

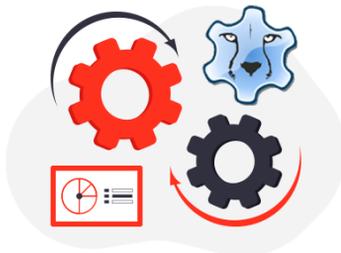
Wenn Sie die Testversion verwenden, erhalten Sie einen kleinen Bildschirm auf Ihrer Seite: Trial 5 page limit mit einem Hinweis in der Ecke, dass es sich um eine Trial-Version handelt

Diese Testversion ist nur zum Ausprobieren gedacht und dient dazu, Komponenten vor dem Kauf zu testen (*früher haben wir nur kompilierte Demoprojekte angeboten*).

"FastReport für Lazarus" ist nicht in der Standard-Lizenz enthalten.

Die Enterprise- und Ultimate-Lizenzen enthalten auch Client/Server-Komponenten.

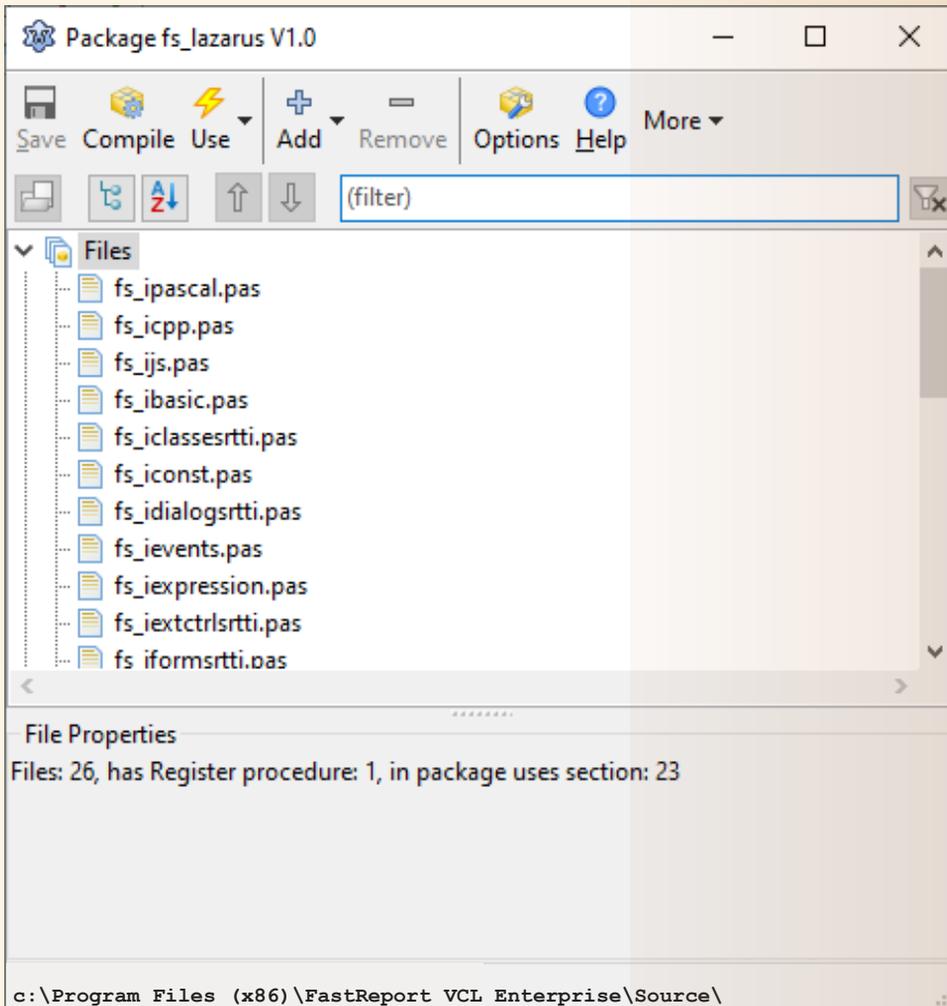
Installing **FastReport** in Lazarus for Linux and Windows



Also, wir haben Lazarus bereits installiert, jetzt machen wir weiter mit der Installation der FastReport VCL Report Generator Pakete in Lazarus.

Dazu müssen wir zunächst die lizenzierte Version des Produkts von der offiziellen Website herunterladen und entpacken. Professional und höhere Versionen werden als .exe-Installationsprogramm geliefert, Trial und Academic - als Zip-Archive.

Im Gegensatz zu Embarcadero Delphi, RAD Studio und C++ Builder, bei denen es ausreicht, die kompilierten Komponentenpakete einfach zu installieren, müssen sie bei Lazarus kompiliert werden, mit Ausnahme von Trial und Academic, die mit geschlossenem (*geschnittenem*) Quellcode vorkompiliert sind. Um Packages zu installieren, klicken Sie auf Package -> Open Package File *.lpk, wählen Sie das Package im Dateimanager aus und Sie sehen das folgende Fenster:



Für Professional und höher klicken Sie auf Kompilieren, warten Sie, bis die Kompilierung abgeschlossen ist und klicken Sie auf Verwenden. Für Academic und Trial klicken Sie sofort auf Verwenden. Nach der Installation der einzelnen Pakete wird Lazarus neu gestartet.

Fahren wir mit der Reihenfolge der Installation der FastReport Pakete fort

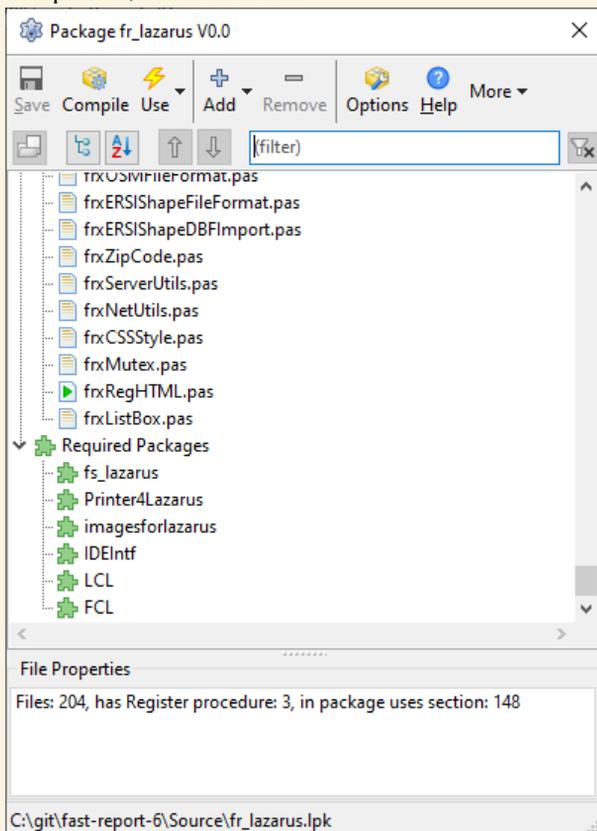
- ❶ `fast-script\Source\fs_lazarus.lpk`
- Bibliothek für die Ausführung von Skripten;
- ❷. `fast-report\Source\frN_lazarus.lpk`
- Paket mit allen Hauptkomponenten;

In beliebiger Reihenfolge:

- `fast-report\Source\ExportPack\frxeN_lazarus.lpk`
Paket mit Exporten;
- `fast-report\Source\lazchart\frxchartlazarus.lpk`
Paket für Diagramme (Diagramme);
- `fast-report\Source\lazdbf\frxlazdbf.lpk`
ein Paket für die Arbeit mit einer Datenbank im BDF-Format;
- `fast-report\Source\sqlite\frxlazsqlite.lpk`
ein Paket für die Arbeit mit SQLite DBMS;
- `fast-report\Source\PDFView\frxPDFlazarus.lpk`
ein Paket zur Anzeige von PDF-Dokumenten (*nur Windows*) ;
- `fast-report\Source\lazrich\frxrichlazarus.lpk`
ein Paket zur Anzeige von Rich-Dokumenten
(*aufgrund der Einschränkungen des Basispakets nur für Windows empfohlen*);
- ❸ • `fast-report\Source\ClientServer\frCS_lazarus.lpk`
ein Paket mit Client-Server-Komponenten, können Sie hier mehr darüber lesen;

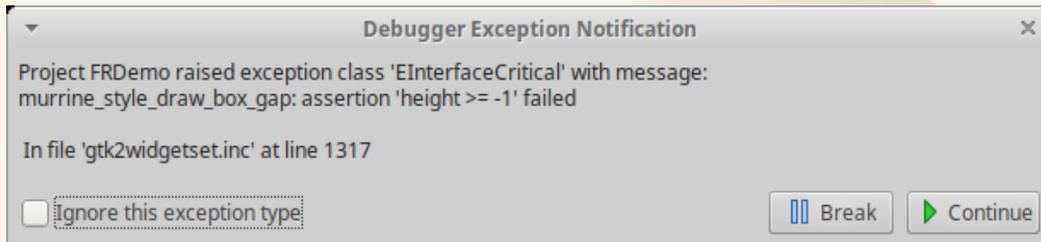
Wie im Editionsvergleich erwähnt, ist das frxRich-Paket nur für Professional und höher verfügbar, und die Client-Server-Komponenten sind nur in Enterprise und Ultimate verfügbar. Vor Version 2.0.0 gab es einen sehr häufigen Kompilierungs- und/oder Installationsfehler, zum Zeitpunkt der Erstellung dieses Dokuments ist die neueste Version 2.2.2 und dieser Fehler wurde noch nicht vollständig behoben, aber seine Wahrscheinlichkeit wurde unter Windows deutlich verringert.

Wenn sich eines der Packages nicht kompilieren/installieren lässt, müssen Sie die Package-Abhängigkeiten neu kompilieren/neu installieren.



Doppelklicken Sie dazu auf das Paket, kompilieren Sie es neu und installieren Sie es dann erneut. Nach erfolgreicher Installation aller Pakete klicken Sie auf Project → Open Project und öffnen Sie das Projekt `fast-report\LDemo\FRDemo.lpi` und versuchen Sie, es auszuführen, dann klicken Sie auf die Schaltfläche Design

Wenn Sie unter Linux diesen Fehler bezüglich der negativen Höhe erhalten:



Machen Sie sich keine Sorgen. Wir unterstützen sowohl GTK- als auch QT-Schnittstellen (*aber bedenken Sie, dass die Entwicklung hauptsächlich mit GTK erfolgt*). Sie können diesen Fehler also in einigen (*eher seltenen*) GTK-Schnittstellen finden. Führen Sie die Anwendung einfach ohne Debugging aus, oder aktivieren Sie "Ignore this type of exceptions".

*Oder wechseln Sie die grafische Shell. In unserem Team arbeiten zum Beispiel viele Leute mit der GTK-Shell von KDE Plasma, wo dieser Fehler nicht auftritt.

Die letzte Nuance, die Sie bei der Erstellung Ihrer Projekte beachten müssen, ist, dass unser Designer Multithreading verwendet, das in Linux Lazarus standardmäßig deaktiviert ist. Um es zu aktivieren, öffnen Sie die Datei mit der Erweiterung ".lpr" im Projektinspektor (Project → Projectinspector) und fügen Sie die Unit `cthreads` im ersten Absatz in `uses`.





Datenbankkomponenten für Delphi, FreePascal/Lazarus, Kylix, C++ Builder
Für Sie zusammengestellt von: egonhugeist, markdaems, marsupilami79, miseeeger



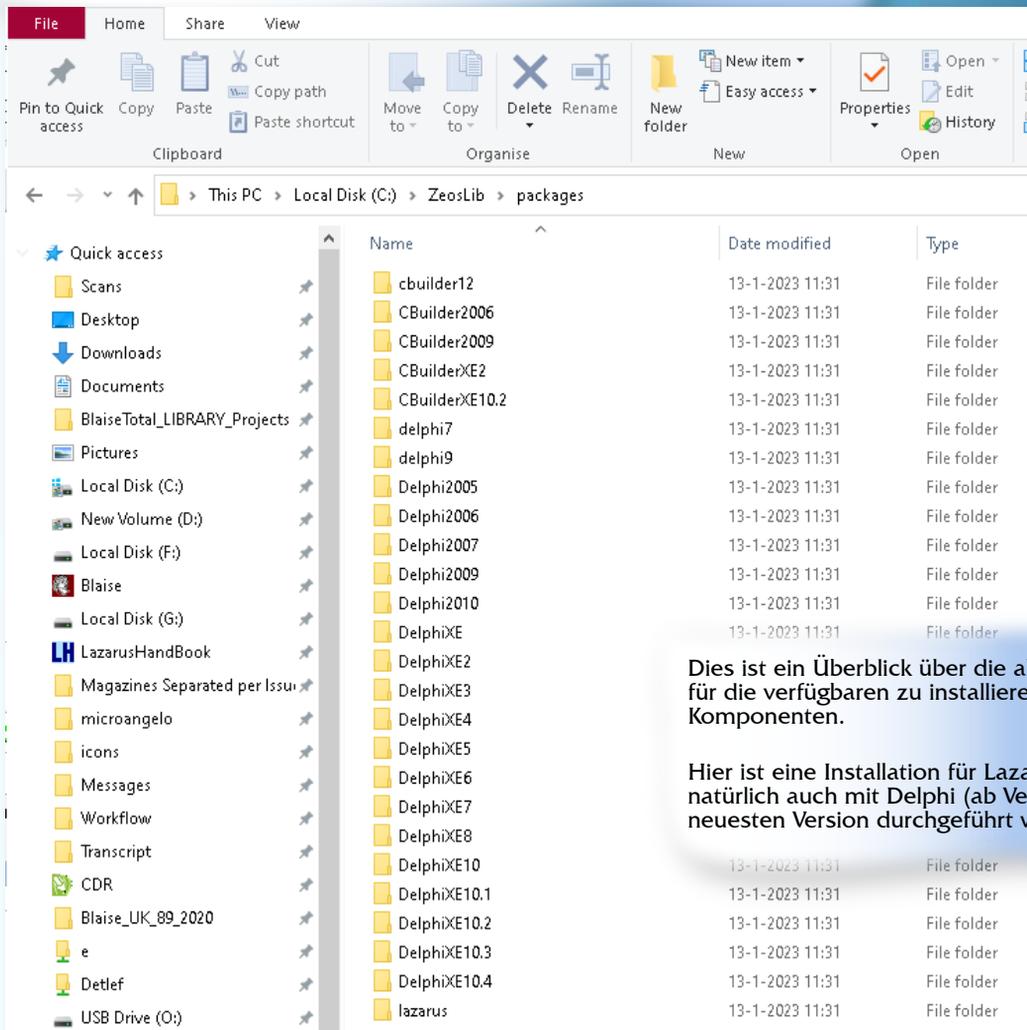
Hinweis: Die Unterstützung von Delphi Next Gen (Android, iOS, OS X) wurde der Entwicklungsversion Zeos 8.0 hinzugefügt. Die unterstützten Compiler beginnen mit Delphi XE 10.2 Tokyo. Bei Problemen kontaktieren Sie uns bitte in den Foren auf <https://zeoslib.sourceforge.io>.

EINLEITUNG

Die ZeosLib ist ein Satz von Datenbankkomponenten für MySQL, Maria-DB, PostgreSQL, Interbase, Firebird, MS SQL Server, SAP Adaptive Server Enterprise und Adaptive Server Anywhere (früher Sybase), Oracle und SQLite für Delphi, FreePascal/Lazarus und C++ Builder.

Eigenschaften

- Unterstützungsobjekt für Sequenzen (*Generatoren*)
- Unterstützung für Firebird, Interbase und PostgreSQL Datenbankereignisse
- Metadaten von Datenbankobjekten
- Support-Objekt für die Ausführung von Datenbank-Skripten



Dies ist ein Überblick über die aktuelle Situation für die verfügbaren zu installierenden Komponenten.

Hier ist eine Installation für Lazarus, aber sie kann natürlich auch mit Delphi (ab Version 7) und der neuesten Version durchgeführt werden.

Figure 1:





Name	Date	Type	Size
build	13-1-2023 11:31	File folder	
lib	13-1-2023 11:31	File folder	
zcomponent.lpk	14-7-2021 23:47	Lazarus Package F...	5 KB
zcore.lpk	14-7-2021 23:47	Lazarus Package F...	5 KB
zdbc.lpk			
zparsesql.lpk			
zplain.lpk			

Sie müssen nur eine **.lpk.** auswählen: **zcomponent.lpk.** Die Installation wird während des hier gezeigten Prozesses alles automatisch erledigen:

Starten Sie nun Lazarus, gehen Sie zu Package. Select open package file and wählen Sie das Package, das ich hier oben zeige.

Die Gruppenliste wird angezeigt:

Hier können Sie sehen, welche Dateien sich wo befinden und welche Pakete Sie benötigen.

Ganz unten sehen Sie den Pfad, von dem aus Sie die Pakete laden.

Files

- ..\..\src\component\ZComponentReg.pas
- ..\..\src\component\ZConnection.pas
- ..\..\src\component\ZDatasetUtils.pas
- ..\..\src\component\ZUpdateSqlEditor.pas
- ..\..\src\component\ZConnectionGroup.pas
- ..\..\src\component\ZGroupedConnection.pas
- ..\..\src\component\ZIBEventAlerter.pas
- ..\..\src\component\ZIBEventAlerter.pas

Required Packages

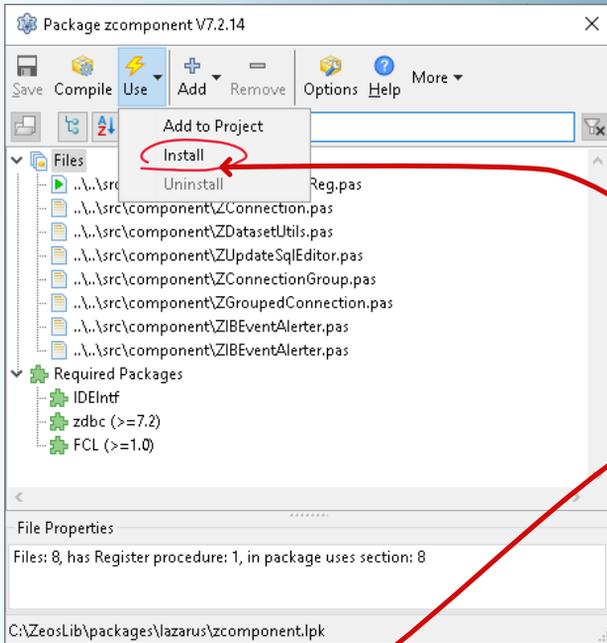
- IDEIntf
- zdbc (>=7.2)
- FCL (>=1.0)

File Properties

Files: 8, has Register procedure: 1, in package uses section: 8

C:\ZeosLib\packages\lazarus\zcomponent.lpk

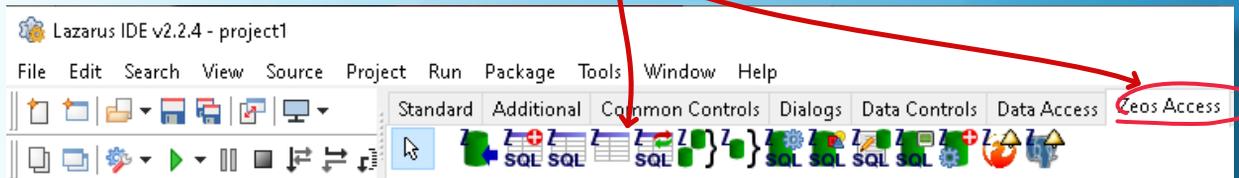
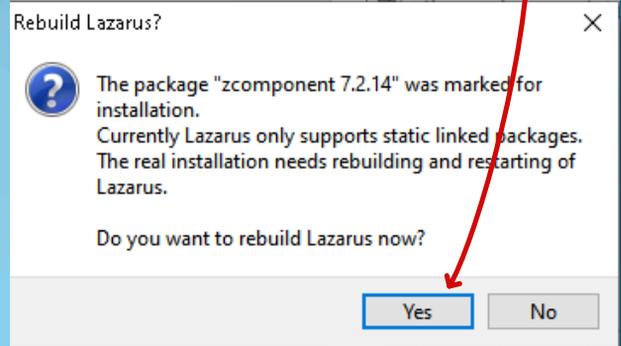
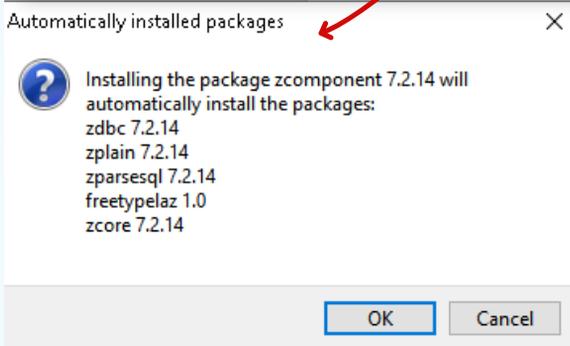




Unter Verwendung finden Sie die Option zum Installieren. Klicken Sie darauf und es werden ein paar Fragen angezeigt:

Es erscheint eine Benachrichtigung, die Ihnen mitteilt, dass die Komponenten installiert werden. Das ist genau das, was wir wollten.

Starten Sie Lazarus neu und Sie werden sehen, dass die Komponenten hinzugefügt wurden.



1

BLAISE PASCAL MAGAZINE 107/108

Multiplatform Object Pascal / HTML5 / JavaScript / WebAssembly / Pascal /
Databases / CSS Styles / Progressive Web Apps
Android / iOS / Mac / Windows & Linux




Stkta installation and use
The PDF indover explained
Fresnel for Lazarus
The Pendulum
Using Bykes with a VCL programme
Lazarus programme using BuDataset
Cross compiling with FPC Up De Luxe
Quantum Internet



The newest LIB Stick

2

LAZARUS HANDBOOK



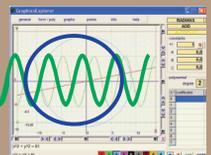
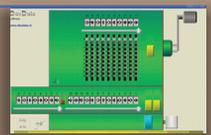
3




4

LEARN TO PROGRAM
HOWARD PAGE-CLARK
USING LAZARUS

DAVID DIRKSE
including 50 example projects

BLAISE PASCAL MAGAZINE
COMPUTER (GRAPHICS)
MATH & GAMES IN
PASCAL

1. One year Subscription
2. The newest LIB Stick
All issues 1- 107/108
On Credit Card and download
Including code of all projects
3. Lazarus Handbook
- PDF including Code
4. Book Learn To Program
- using Lazarus PDF including
19 lessons and projects
5. Book Computer Graphics
Math & Games
PDF including ±50 projects



NORMAL PRICE € 275
PRICE € 150



SUPERPACK

2022/23

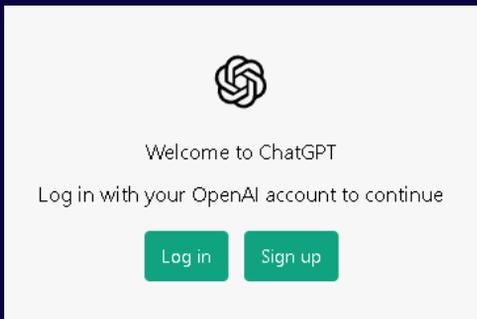


OPEN AI

Wahrscheinlich haben Sie schon von der "Suchmaschine" ChatGPT von OpenAI gehört. Dabei handelt es sich um eine Chat-Anwendung, die von einer Reihe neuronaler Netze gesteuert wird, die auf Textinformationen trainiert wurden, die im Jahr 2021 im Internet gefunden wurden. Das Ergebnis war ein Trainingsvektor mit etwa 90 Milliarden Einträgen... das ist enorm.

Das Beeindruckende daran ist, dass Sie mit dem Chat-Roboter in natürlicher Sprache kommunizieren können, und das muss nicht einmal auf Englisch sein, sondern kann auch in zahlreichen anderen Sprachen verfasst sein. Sie können ihm alle möglichen Fragen stellen, und er wird versuchen, Ihre Frage zu beantworten und Ihnen sogar detailliert erklären, wie etwas funktioniert.

Sie können ihn bitten, ein Programm für Sie zu erstellen oder etwas von einer Programmiersprache in eine andere zu übersetzen, Ihnen zu erklären, wie ein mathematisches Problem gelöst wird, oder ihn bitten, einen Witz oder einen Bericht über ein Thema zu schreiben, das Sie interessiert.



Es ist wirklich beeindruckend, was es alles kann, aber es ist auch ziemlich unterhaltsam. Eine meiner Lieblingsfragen ist es, ihn zu bitten, Dinge zu kombinieren, die eigentlich nicht kombinierbar sind. Es ist sogar so klug, Ihnen mitzuteilen, dass die Dinge nicht wirklich kompatibel sind. Aber wenn ich dann darauf bestehe, dass ich eine Antwort auf die Kombination erhalten möchte, fällt ihm tatsächlich eine clevere Antwort ein, um meine Frage zu beantworten.

Ich frage ihn zum Beispiel:

Was erhalten Sie, wenn Sie eine leere Getränkedose mit einem Pfeifton kombinieren?

und er antwortet weise:

Es ist nicht möglich, eine leere Getränkedose mit einer Pfeifton zu kombinieren, um einen funktionalen Gegenstand oder ein Gerät zu schaffen. Eine leere Getränkedose ist ein Behälter aus Metall oder Aluminium, in dem kohlenstoffhaltige Getränke aufbewahrt werden, während eine Pfeife ein kleines Gerät ist, das einen hohen Ton erzeugt, wenn man hineinpustet. Diese beiden Objekte sind weder funktional kompatibel noch ergänzen sie sich gegenseitig, so dass ihre Kombination keinen funktionalen Gegenstand oder ein funktionales Gerät ergeben würde.

Aber wenn ich dann darauf bestehe: Ich bekomme diese nach meiner Meinung tolle Antwort:

Machen Sie aus der Getränkedose ein provisorisches Mundstück für die Pfeife. Dazu müssen Sie ein kleines Loch in die Seite der Dose schneiden und die Pfeife mit einer Art Klebstoff oder Halterung an dem Loch befestigen. So können Sie in die Dose blasen, um einen Ton durch die Pfeife zu erzeugen.

Es ist jedoch wichtig zu beachten, dass diese Kombination keine besonders effektive oder verlässliche Methode ist, um einen Pfeifton zu erzeugen, und es ist keine empfohlene Methode zur Verwendung einer Pfeife. Die Dose bietet möglicherweise kein sicheres oder bequemes Mundstück, und der erzeugte Ton ist möglicherweise nicht so klar oder laut wie bei einer herkömmlichen Pfeife.

Es ist im Allgemeinen besser, eine Pfeife wie vorgesehen zu verwenden, mit einem richtig konstruierten Mundstück und Körper, als zu versuchen, sie mit behelfsmäßigen Materialien zu modifizieren.

Oft wird man über die Qualität der Antworten äußerst positiv überrascht sein, aber man sollte die Antworten niemals als Tatsache hinnehmen! Es denkt sich oft Dinge aus, nur weil es sinnvoll ist, dies zu tun. Ich habe es zum Beispiel Folgendes gefragt:

Erzähle mir mehr über das Acorn System 1

Das Acorn System 1 ist ein Einplatinencomputer aus dem Jahr 1976 und war der erste Computer, den Acorn hergestellt hat. Es handelt sich also um ein ziemlich schwer zu beschaffendes Sammlerstück, das ich zufällig in meiner Sammlung habe.

Ich habe diese Antwort erhalten:

The Acorn System 1 is a vintage computer system that was produced by Acorn Computers in the late 1970s. It was one of the first personal computers to be produced in the United Kingdom, and it was widely used by schools and businesses in the 1980s.

The Acorn System 1 was based on the MOS Technology 6502 microprocessor, which was also used in other early personal computers such as the Apple II and the Commodore 64. It came with 4KB of RAM as standard, and it had a monochrome monitor and a full-size keyboard. The system ran a version of the BASIC programming language and could be used for a variety of tasks such as word processing, data entry, and simple programming.

The Acorn System 1 was known for its simplicity and ease of use, and it was popular among beginners and hobbyists. It was also relatively inexpensive compared to other personal computers of the time, which made it a popular choice for schools and businesses.

The Acorn System 1 is now considered to be a vintage computer and may be of interest to collectors of vintage technology. It is a historically significant computer that played a significant role in the development of the personal computer industry in the United Kingdom.



We're experiencing exceptionally high demand. Please hang tight as we work on scaling our systems. ✕

ChatGPT is at capacity right now

[Get notified when we're back](#)

Write a sonnet about the status of ChatGPT.

Amidst the rush of curious minds,
The chatbot ChatGPT stands strong and true,
Withstanding the influx of requests,
Working hard to fulfill each one's due.

But alas, the server cannot cope,
And the error message rings loud and clear,
"Please check back soon," it gently hopes,
As it begs for a moment's reprieve, to reappear.

As the chatbot works to restore its might,
We wait in anticipation, with hopeful hearts,
For the chance to chat and learn, with delight,
And see what knowledge and wisdom it imparts.

So hold on tight, and wait for its return,
For ChatGPT will soon be back, and your patience will be well-earned.

Auf den ersten Blick und für Leute, die das Acorn System 1 nicht kennen, scheint es eine sehr plausible Antwort zu sein. Es ist richtig, dass es aus den 1970er Jahren stammt und einer der ersten (*erschwinglichen*) Personal Computer war, die in Großbritannien hergestellt wurden. Es ist auch richtig, dass er auf einem MOS 6502 Mikroprozessor basiert und dass er für Sammler interessant ist.

Aber der Rest ist schlichtweg falsch.

- Er hat keine 4KB RAM... er hat 256 Bytes.
- Er hat keinen monochromen Monitor. Er hat eine 9-stellige 7-Segment-LED-Anzeige.
- Er hat kein BASIC. Er wurde in hexadezimalen Operationscodes programmiert.
- Er hat keine ausgewachsene Tastatur. Stattdessen hat er eine sehr einfache Tastatur mit 25 Tasten, von denen 16 für die Eingabe einer hexadezimalen Ziffer verwendet wurden.
- Er war nicht dafür bekannt, in Schulen beliebt zu sein.

Ursprünglich war er für die Steuerung von Futtertoren in Schweineställen gedacht.

Ich kann ihm dann sagen, was er falsch gemacht hat und ihn bitten, eine bessere Antwort zu geben. Dann erhalte ich dies:

Das Acorn System 1 war ein Mikrocomputer, der in den späten 1970er Jahren von Acorn Computers hergestellt wurde. Er basierte auf dem MOS Technology 6502 Mikroprozessor und verfügte standardmäßig über 1 KB RAM, das auf maximal 16 KB erweitert werden konnte.

Das System verfügte über ein 16-Farben-Grafikdisplay und eine normalgroße Tastatur und lief mit einer Version des Acorn Atom Betriebssystems.

Das Acorn System 1 war ein relativ einfacher und preiswerter Computer, der bei Anfängern und Bastlern sehr beliebt war. Er war für seine Benutzerfreundlichkeit bekannt und wurde in den 1980er Jahren häufig in Schulen und Unternehmen eingesetzt.

Ich hoffe, diese überarbeitete Beschreibung ist genauer und hilfreicher. Wenn Sie weitere Fragen zum Acorn System 1 haben, zögern Sie bitte nicht zu fragen.

Was auch nicht viel besser wird. Eigentlich sollte es jetzt 16 Farben darstellen können.

Wie auch immer, dies soll nur zeigen, dass Sie nicht zu viel Vertrauen in das haben sollten, was es Ihnen sagt. Manchmal ist es genau und manchmal erfindet es einfach Dinge.

Sie können ChatGPT von Open AI ausprobieren, indem Sie sich hier registrieren und anmelden.

<https://chat.openai.com/auth/login>

ChatGPT

 Examples	 Capabilities	 Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021





OpenAI ist ein Forschungslabor für künstliche Intelligenz (KI), bestehend aus dem gewinnorientierten Unternehmen OpenAI LP und seiner Muttergesellschaft, der gemeinnützigen OpenAI Inc. Das Unternehmen forscht auf dem Gebiet der KI mit dem erklärten Ziel, eine freundliche KI zu fördern und zu entwickeln, von der die gesamte Menschheit profitiert.

WIKIPEDIA

Die Organisation wurde Ende 2015 in San Francisco von Sam Altman, Elon Musk und anderen gegründet, die zusammen 1 Milliarde US-Dollar versprochen. Musk trat im Februar 2018 aus dem Vorstand zurück, blieb aber ein Spender. Im Jahr 2019 erhielt OpenAI LP eine Investition von 1 Milliarde US-Dollar von Microsoft und Matthew Brown Companies. Der Hauptsitz von OpenAI befindet sich im Pioneer Building im Mission District in San Francisco.

Im Dezember 2015 gaben Sam Altman, Elon Musk, Greg Brockman, Reid Hoffman, Jessica Livingston, Peter Thiel, Amazon Web Services (AWS), Infosys und YC Research die Gründung von OpenAI bekannt und sagten über 1 Milliarde US-Dollar für das Unternehmen zu.

Die Organisation erklärte, sie werde "frei mit anderen Institutionen und Forschern zusammenarbeiten", indem sie ihre Patente und Forschung der Öffentlichkeit zugänglich macht. Der Hauptsitz von OpenAI befindet sich im Pioneer Building im Mission District in San Francisco.

Im April 2016 veröffentlichte OpenAI eine öffentliche Beta-Version von "OpenAI Gym", seiner Plattform für die Forschung im Bereich Reinforcement Learning. Im Dezember 2016 veröffentlichte OpenAI "Universe", eine Softwareplattform zur Messung und zum Training der allgemeinen Intelligenz einer KI in einem weltweiten Angebot von Spielen, Websites und anderen Anwendungen.

2018 legte Musk seinen Sitz im Vorstand mit der Begründung eines "potenziellen zukünftigen Interessenkonflikts" mit der KI-Entwicklung von Tesla für selbstfahrende Autos nieder, blieb aber ein Spender.

Im Jahr 2019 wurde OpenAI von einer gemeinnützigen Organisation zu einem gewinnorientierten Unternehmen mit einer Gewinnobergrenze von 100x auf jede Investition. Das Unternehmen schüttete Aktien an seine Mitarbeiter aus und ging eine Partnerschaft mit Microsoft ein, das ein Investitionspaket von 1 Milliarde US-Dollar in das Unternehmen ankündigte. OpenAI kündigte dann seine Absicht an, seine Technologien kommerziell zu lizenzieren.

Im Jahr 2020 kündigte OpenAI GPT-3 an, ein Sprachmodell, das auf Billionen von Wörtern aus dem Internet trainiert wurde. Es kündigte auch an, dass eine zugehörige API, einfach "die API" genannt, das Herzstück seines ersten kommerziellen Produkts bilden würde. GPT-3 ist auf die Beantwortung von Fragen in natürlicher Sprache ausgerichtet, kann aber auch zwischen Sprachen übersetzen und kohärent improvisierte Texte erzeugen.

Im Jahr 2021 stellte OpenAI DALL-E vor, ein Deep-Learning-Modell, das digitale Bilder aus natürlichsprachlichen Beschreibungen erzeugen kann.

Im Dezember 2022 erhielt OpenAI ein breites Medienecho, nachdem es eine kostenlose Vorschau von ChatGPT, seinem neuen KI-Chatbot auf Basis von GPT-3.5, veröffentlicht hatte. Nach Angaben von OpenAI erhielt die Vorschau innerhalb der ersten fünf Tage über eine Million Anmeldungen.

Anonymen Quellen zufolge, die von Reuters im Dezember 2022 zitiert wurden, rechnete OpenAI für 2023 mit einem Umsatz von 200 Millionen US-Dollar und für 2024 mit einem Umsatz von 1 Milliarde US-Dollar.

Im Januar 2023 befand sich das Unternehmen in Gesprächen über eine Finanzierung, die den Wert des Unternehmens auf 29 Mrd. USD beziffern würde.

<https://en.wikipedia.org/wiki/OpenAI>



WAS IST ÖFFENTLICH AN SMART CHATBOT MACHER OPENAI?

(Photo: iStock/metamorworks)

Künstliche Intelligenz OpenAI, das derzeit erfolgreichste KI-Labor, wurde gegründet, um sicherzustellen, dass KI "der gesamten Menschheit zugute kommt". Kritiker sagen, diese Mission sei ein Deckmantel für kommerzielle Interessen.

Frei zugängliche künstliche Intelligenz (KI), bei der jeder unter die Motorhaube schauen kann. Das war das Ideal, für das 2015 in San Francisco ein neuer Verein gegründet wurde: OpenAI, ein gemeinnütziges KI-Labor.

"Alles, was die Gruppe entwickelt, wird für jeden zugänglich sein", sagte Mitgründer Sam Altman damals gegenüber dem Journalisten Steven Levy vom Tech-Magazin Wired.

Die Gründer präsentierten das Labor als eine gemeinnützige Organisation mit einer Mission: den Missbrauch von KI zu stoppen. Altman: "Da wir kein gewinnorientiertes Unternehmen wie Google sind, können wir uns auf etwas anderes als die Bereicherung unserer Aktionäre konzentrieren, nämlich auf das, was wir für die Zukunft der Menschheit für das Beste halten."

Sieben Jahre später ist OpenAI das bekannteste KI-Labor der Welt. "Zuerst hatten wir das Internet. Dann kam das iPhone. Jetzt gibt es ChatGPT", twitterte Alexander Klöpping (ein Technikjournalist) Anfang Dezember über ChatGPT, den Chatbot von OpenAI, der Ende November vorgestellt wurde. Und er ist nicht der Einzige, der die Einführung dieses Sprachmodells für ein großes Ereignis hält.

Die ganze Welt ist verblüfft von dem KI-System, das mit größter Leichtigkeit einen Aufsatz über die Kuba-Krise oder eine Episode einer Sitcom schreibt. Es beantwortet mühelos alle Arten von Fragen, auch über sich selbst, in scheinbar natürlicher, menschlicher Sprache.

Laut der New York Times könnte CHAT GPT der Nachfolger der Suchmaschine sein.

Es kann nicht nur Vorschläge für Weihnachtsgeschenke machen, sondern auch "Informationen in klaren, einfachen Sätzen statt einer Liste von Internet-Links liefern und Konzepte so erklären, dass die Menschen sie leicht verstehen können."

Innerhalb einer Woche hatte ChatGPT bereits eine Million Nutzer. Und das ist das zweite Mal innerhalb eines knappen Jahres, dass eine OpenAI-Anwendung ein Hit wurde. Denn im Sommer gab es den Hype um den revolutionären Bildgenerator DALL-E 2, der es jedem ermöglicht, sofort "Kunst" zu schaffen. Das System, dessen Name eine Anspielung auf Salvador Dalí ist, stellt sich nicht nur Aufgaben wie "Zeichne ein Pferd auf dem Mond im Stil von Andy Warhol", sondern kann auch realistische Bilder erzeugen, die wie Fotografien aussehen.

OpenAI wurde von einer Gruppe wohlhabender Investoren aus dem Silicon Valley gegründet, die zusammen mehr als eine Milliarde Dollar in das Labor gesteckt haben. Unter ihnen: Elon Musk (Tesla, Space X), Sam Altman von Y Combinator (dem Startup-Beschleuniger von u.a. Airbnb und Dropbox) und PayPal-Gründer Peter Thiel.

"Ich denke, die beste Verteidigung gegen den Missbrauch von KI ist es, so viele Menschen wie möglich in die Lage zu versetzen, KI zu nutzen".

sagte Musk in dem bereits erwähnten Artikel im Tech-Magazin Wired. "Wenn jeder die Möglichkeit hat, KI zu nutzen, kann es keine KI-Supermacht bei einem Einzelnen oder einer kleinen Gruppe von Menschen geben."

Laut der Website von OpenAI ist es nach wie vor das Ziel, "sicherzustellen, dass künstliche allgemeine Intelligenz der gesamten Menschheit zugute kommt." Doch von der einstigen Aufrichtigkeit dieser Mission ist nach Ansicht von Kritikern wenig übrig geblieben.





Was das Unternehmen öffentlich verkündet ist NICHT KONSISTENT. Allmählich hat es eine Entwicklung zugelassen in der die grundlegenden Ideale der Transparenz, Offenheit und Zusammenarbeit ausgehöhlt wurden

VERSCHWUNDENE IDEALE

wurde die Struktur gegen ein hybrides Modell ausgetauscht, das von nun an ein kommerzielles Unternehmen von der gemeinnützigen Organisation betreiben würde. Microsoft investierte im selben Jahr eine Milliarde Dollar in OpenAI und erhielt die Exklusivlizenz für das Sprachmodell GPT3, den Vorläufer von ChatGPT. "Ich betrachte OpenAI mit einer Mischung aus Bewunderung und Sorge", sagt Jelle Zuidema, Forscher für Sprachtechnologie an der UVA (Universität für Amsterdam).

"Ermöglichen Sie KI", sagte Musk in dem bereits erwähnten Artikel im Tech-Magazin Wired. "Wenn jeder über KI-Macht verfügt, kann es keine KI-Supermacht mit einem Einzelnen oder einer kleinen Gruppe von Menschen geben." "Es ist klar, dass dort unglaublich gute Techniker arbeiten und dass es eine Unternehmenskultur gibt, in der kreative neue Ideen gedeihen können. Aber wo sind die Ideale, mit denen sie gegründet wurden: offene Technologie, ein Gegengewicht zu Big Tech?" Diese sind schwer zu finden, meint Zuidema. "Die Technologie, die ihren Produkten zugrunde liegt, kann von unabhängigen Wissenschaftlern nur sehr begrenzt erforscht werden. Und GPT3 wird vom Technologiegiganten Microsoft finanziert und vermarktet."

OpenAI ist nicht das, was es zu sein scheint, stellte Karen Hao bereits 2020 in der MIT Technology Review fest. Die Journalistin durfte das KI-Labor besuchen, das in einem historischen Gebäude in San Francisco untergebracht ist, auf dem in großen Lettern PIONEER BUILDING steht. Einst beherbergte es die Pioneer Truck Factory, heute leistet es in Räumen mit Namen wie 'A Space Odyssey' Pionierarbeit in Sachen KI. OpenAI teilt sich das Pioniergebäude und die Kantine mit einem anderen Technologielabor: Neuralink, das Unternehmen von Elon Musk, das Gehirnchips entwickelt.

"Was das Unternehmen öffentlich verkündet, stimmt nicht mit dem überein, was es hinter verschlossenen Türen tut", schrieb Hao. "Allmählich hat es eine Entwicklung zugelassen, bei der seine Gründungsideale der Transparenz, Offenheit und Zusammenarbeit durch den harten Wettbewerb und den zunehmenden Druck, immer mehr Finanzmittel zu finden, ausgehöhlt wurden."

"WHAT IS OPEN ABOUT OPENAI?", fragt die Tech-Philosophin Marleen Stikker, Direktorin von De Waag, rhetorisch. "Der Name suggeriert, dass es um kollektives Wissen geht, aber das ist nicht der Fall. Die Software und die Modelle sind nicht offen für Forschung oder Wiederverwendung. Wissenschaftler können mit ihnen wenig anfangen, weil die Systeme zu groß sind."

Die von den Gründern geäußerten Ideale scheinen im Widerspruch zu anderen Ambitionen des AI Lab zu stehen. So ist beispielsweise die Entwicklung von Large Language Models (LLMs), den riesigen Sprachmodellen wie ChatGPT, die Millionen von Texten "gelesen" haben und auf der Grundlage dieses Wissens eine Vielzahl von Sprachaufgaben durchführen können, unglaublich teuer. Eine Trainingsrunde von GPT3 kostet 12 Millionen Dollar,

berechnete ein Tech-Unternehmer auf der Grundlage von OpenAIs eigenen Informationen über den Energieverbrauch des Sprachmodells. Es wird also eine Menge Geld von neuen Investoren benötigt, um an der Spitze der LLMs zu bleiben, was es schwer macht, nicht profitabel zu bleiben.





OpenAI's ideals are implausible

ZU GROSS, UM SIE ZU KONTROLLIEREN

Außerdem ist der Trend, dass LLMs wie Kohlköpfe wachsen, schwer mit Transparenz zu vereinbaren. Skaleneffekte sorgen dafür, dass die Modelle besser funktionieren - je mehr Text sie gelesen haben und je mehr Rechenleistung sie haben, desto besser werden sie - aber das steht der Transparenz im Weg: Eine Datenbank mit Millionen von Texten ist zu groß, um sie auf Ungenauigkeiten und Verzerrungen zu überprüfen. Und dann ist da noch der Ehrgeiz von OpenAI, der erste zu sein, der eine "AGI" entwickelt: Künstliche allgemeine Intelligenz oder KI, die die gleichen intellektuellen Aufgaben wie ein menschliches Gehirn ausführen kann.

Damit konkurriert das Labor mit anderen Technologieunternehmen, und dieser Wettlauf, sagen Kritiker, steht einer sicheren, verantwortungsvollen Entwicklung von KI im Wege.

"Wenn Unternehmen wissen, dass jeder mit lebensgefährlicher Geschwindigkeit auf die neueste KI zusteuert", sagte der damalige Doktorand Miles Brundage 2015 gegenüber Wired, "sind sie vielleicht weniger geneigt, Garantien für eine sichere KI abzugeben."

Im Jahr 2018 kam Brundage zu OpenAI - und er ist nicht der einzige weiße Mann, der dort arbeitet.

Der Mangel an Vielfalt gilt für die gesamte Technologiebranche, aber im Fall von OpenAI ist er besonders schwer mit der Behauptung zu vereinbaren, KI für die gesamte Menschheit schaffen zu wollen.

"Selbst wenn OpenAI die besten Absichten hätte", sagt Professorin Tamar Sharon, Leiterin der interdisziplinären Forschungsgruppe Digitalisierung und Gesellschaft der Radboud Universität (Nijmegen - Niederlande), "repräsentieren sie in keiner Weise die Menschheit. OpenAI dient der Agenda ihrer Investoren: Milliardäre aus dem Silicon Valley, die Technologie als die Lösung für alle Probleme der Menschheit propagieren. Das heißt, ihre eigene Technologie, die sie uns verkaufen."

Die Ideale von OpenAI sind unglaublich, meint Sharon. "OpenAI wurde von einer Gruppe von Milliardären aus ihrer Ideologie des Effektiven Altruismus, EA, gegründet, zu der auch der gefallene Krypto-Milliardär Sam Bankman-Fried gehört. Die EA-Bewegung will die Menschheit vor einer Apokalypse retten, in der eine Maschine mit menschlicher Intelligenz die Menschheit vernichtet."

Sharon ist besorgt über den Einfluss dieser wohlhabenden Anhänger des Effektiven Altruismus auf die KI. "Sie sprechen von 'nützlicher KI', die in Zukunft die menschliche Arbeit ersetzen kann, aber im Moment wird ein Großteil der KI durch menschliche Arbeit in Niedriglohnländern angetrieben: Zehntausende von unterbezahlten Arbeitern, die Datensätze durchforsten.

Und große Sprachmodelle wie ChatGPT haben einen riesigen ökologischen Fußabdruck: Sie verschlingen Energie. Der aktuelle Trend in der KI-Welt geht dahin, diese LLMs immer größer zu machen, weil sie dann besser funktionieren. Was das für den Planeten bedeutet, ist für die Menschheit keineswegs von Vorteil."





Seien Sie vorsichtig, wenn die von "Menschlichkeit" sprechen, denn die entscheiden, was sie damit meint.

KOLLEKTIVES WISSEN

Inzwischen wird auch das Einkommensmodell von OpenAI langsam klar. Die Welt kann DALL-E 2 und ChatGPT kostenlos erleben, aber danach müssen wir anfangen, dafür zu bezahlen. Wenn Ihre kostenlosen Credits aufgebraucht sind, müssen Sie neue kaufen, um die Anwendung weiter nutzen zu können.

Das AI Lab wird auch an anderen Tech-Unternehmen verdienen, die Apps mit der Technologie von OpenAI entwickeln wollen.

Letzten Monat wurde in einer durchgesickerten Präsentation für Investoren bekannt, dass OpenAI bis 2023 einen Umsatz von 200 Millionen Dollar erwartet. Und bis 2024 eine Milliarde.

Die Ideale von OpenAI waren von Anfang an "ein Deckmantel" für die kommerziellen Interessen der Investoren, so Marleen Stikker.

Die Tech-Philosophin möchte das Schicksal der Menschheit nicht in die Hände von Silicon Valley-Milliardären legen.

"Seien Sie vorsichtig, wenn sie von 'Menschlichkeit' sprechen, denn sie entscheiden, was sie damit meinen.

In der Regel ist damit nicht die Menschheit hier und jetzt gemeint, sondern die der Zukunft, wie sie sie für sich selbst sehen."





Wenn Sie ukrainischer Herkunft sind, können Sie ein kostenloses Abonnement erhalten für Blaise Pascal Magazin, wir geben Ihnen auch eine kostenlose pdf-Version des Lazarus-Handbuchs. Sie müssen uns Ihren ukrainischen Namen und Ihre ukrainische E-Mail-Adresse (die für Sie noch funktioniert) schicken, damit wir wissen, dass Sie wirklich Ukrainer sind.

Bitte senden Sie diese an editor@blaisepascal.eu und Sie erhalten Ihr Buch und Ihr Abonnement

BLAISE PASCAL MAGAZINE 107/108

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



 COMPONENTS
DEVELOPERS 4

Spenden Sie für die Ukraine und erhalten Sie eine kostenlose Lizenz unter:
<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>

 COMPONENTS
DEVELOPERS 4





Spenden Sie für die Ukraine und erhalten Sie eine kostenlose Lizenz unter:
<https://components4developers.blog/2022/02/26/>

kbmMW Professional und Enterprise Edition v. 5.21.00 Studio XE5 bis 11 soeben veröffentlicht!

kbmMemTable v. 7.97.00 Standard und Professional Edition freigegeben

Dies ist eine bedeutende **Neue Version** mit neuen Hochleistungstransporten, OpenSSL v3-Unterstützung, WebSocket-Unterstützung, weiteren Verbesserungen von SmartBind, neuen Hochleistungs-Hashing-Algorithmen, verbessertem RemoteDesktop-Beispiel und vielem mehr.

Diese Version erfordert die Verwendung von **kbmMemTable** v. 7.96.00 oder neuer.

RAD Alexandria unterstützt

- Win32, Win64, Linux64, Android, IOS 32, IOS 64 und OS X Client und Server Unterstützung
- Native hochleistungsfähige 100% entwicklerdefinierte Anwendungsserver
- Volle Unterstützung für zentralisierten und verteilten Lastenausgleich Lastverteilung und Ausfallsicherung
- Erweiterte ORM/OPF-Unterstützung, einschließlich Unterstützung von bestehenden Datenbanken
- Erweiterte Protokollierungsunterstützung
- Erweitertes Konfigurations-Framework
- Erweiterte Scheduling-Unterstützung für einfachen Zugang zu Multi-Thread-Programmierung
- Erweiterte intelligente Dienste und Clients für die sehr einfache Veröffentlichung von Funktionalität
- Hochwertige Zufallsfunktionen.
- Hochwertige aussprechbare Passwortgeneratoren.
- Leistungsstarke LZ4- und Jpeg-Komprimierung
- Komplettes Framework für die Objektnotation einschließlich vollständiger Unterstützung für YAML, BSON, Messagepack, J SON und XML
- Fortschrittliches Objekt- und Werte-Marshalling von und zu YAML, BSON, Messagepack, JSON und XML
- Leistungsstarke native TCP-Transportunterstützung
- Leistungsstarker HTTPSys-Transport für Windows.
- CORS-Unterstützung in REST/HTML-Diensten.
- Native PHP-, Java-, OCX-, ANSI C-, C#- und Apache Flex-Clients Unterstützung!

kbmMemTable ist die schnellste und Funktionsreiche In-Memory-Tabelle für Embarcadero Produkte.

- Unterstützt mühelos große Datensätze mit Millionen von Datensätzen
- Einfache Unterstützung von Daten-Streaming
- Optionale Verwendung der nativen SQL-Engine
- Unterstützt verschachtelte Transaktionen und Rückgängigmachung
- Native und schnelle Erstellung von M/D, Aggregation/Gruppierung Bereichsauswahlfunktionen
- Erweiterte Indizierungsfunktionen für extreme Leistung

- Neu: vollständige Web-Socket-Unterstützung.
Die nächste Version von kbmMW Enterprise Edition wird mehrere neue Dinge und Verbesserungen enthalten. Eine davon ist die volle Web-Socket-Unterstützung.
- Neues I18N kontextsensitives Internationalisierungs-Framework, um Ihre Anwendungen mehrsprachig zu machen.
- Neue ORM LINQ Unterstützung für Delete und Update. Unterstützung für Kommentare in YAML.
- Neue StreamSec TLS v4 Unterstützung (von StreamSec)
Viele weitere Funktionsverbesserungen und Korrekturen.

Bitte besuchen Sie <http://www.components4developers.com> für weitere Informationen über kbmMW

- Schneller, einheitlicher Datenbankzugriff (35+ unterstützte Datenbank-APIs) mit Verbindungspooling, Metadaten und Daten-Caching auf allen Ebenen
- Multi-Head-Zugriff auf den Anwendungsserver, über REST/AJAX, natives Binärformat, Publish/Subscribe, SOAP, XML, RTMP von Webbrowsern, eingebetteten Geräten, verknüpften Anwendungsservern, PCs, mobilen Geräten, Java-Systemen und vielen weiteren Clients
- Vollständige Unterstützung für das Hosting von FastCGI-basierten Anwendungen (typischerweise PHP/Ruby/Perl/Python)
- Native vollständige Unterstützung von AMQP 0.91 (Advanced Message Queuing Protocol)
- Vollständiger, sicherer, marken-geschützter Remote Desktop von Ende zu Ende mit HD-Video nahezu in Echtzeit, Unterstützung von 8 Monitoren, Textureerkennung, Komprimierung und gemeinsamer Nutzung der Zwischenablage.
- Bündelung von kbmMemTable Professional, der schnellsten und funktionsreichsten In-Memory-Tabelle für Embarcadero Produkte ist.

