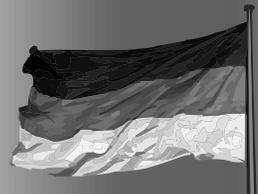
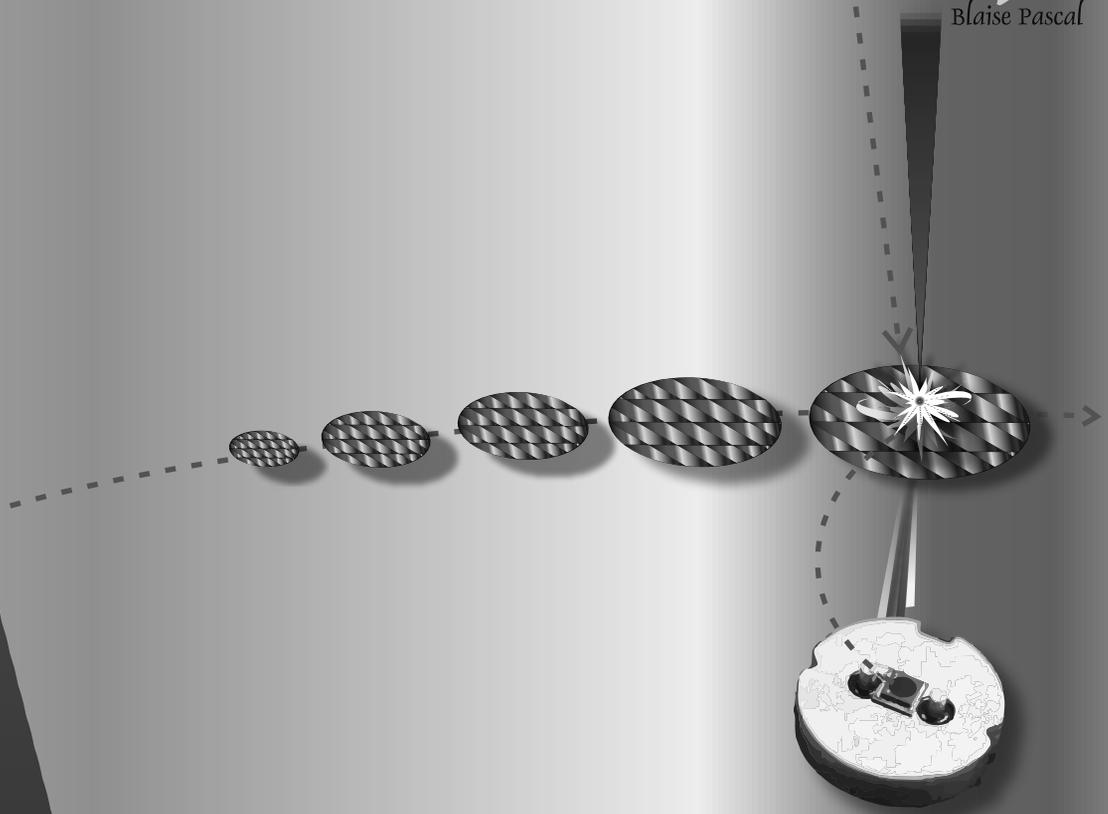


BLAISE PASCAL MAGAZINE 106

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



Blaise Pascal



AGSI (Aggregierter Gasspeicher) - Datenspeicherung

Undo System erklärt

Laser CPU - die Zukunft rückt näher

Untappable Internet - kein Knacken mehr

Das neue Delphi 11.2.1 ist da

Fastreport QR Code macht es sehr einfach

Übersetzen Sie Ihre Website mit PAS2JS

FPC UP DeLuxe der fantastische Lazarus Installer

kbmMW Die neue Versionsnummer 5.20.01 ist eingetroffen

DAS LAZARUS-TEAM RÄT

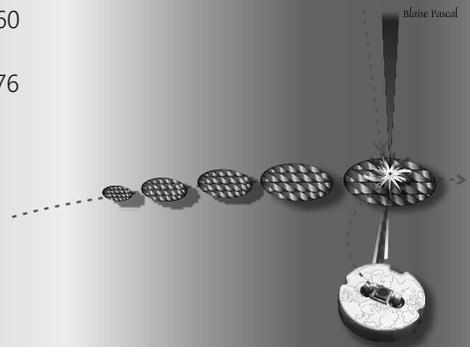
DER BESTE &
EINFACHSTE WEG
FPC LAZARUS ZU INSTALLIEREN
STABLE / TRUNK ODER
CROSSCOMPILE



INHALT

ARTIKEL:

Von Ihrem Herausgeber	Page 4
Karikaturen von Jerry King	
AGSI (Aggregierter Gasspeicher) - Datenspeicherung	Page 7
Von Max Kleiner	
Undo System erklärt	Page 26
Von David Dirkse	
Laser CPU - die Zukunft rückt näher	Page 70
Von Detlef Overbeek	
Untappable Internet - kein Knacken mehr	Page 22
Von Detlef Overbeek	
Das neue Delphi 11.2.1 ist da	Page 15
a universal way to install components and libraries into Delphi	
Von Danny Wind	
Fastreport QR Code macht es sehr einfach	Page 38
Von Michael Phillipenko	
Übersetzen Sie Ihre Website mit PAS2JS	Page 44
Von Michael van Canneyt	
FPC UP DeLuxe der fantastische Lazarus Installer	Page 60
Von Jos Wegman	
kbmMW Die neue Versionsnummer 5.20.01 ist eingetroffen	Page 76



INSERENTEN

Angebot für Abonnemente	Page 14
Laz Handbook	Page 6 / 13
Components4Developers (kbmMW)	Page 75 / 76
Database workbench	Page 36
FastReport	Page 37
Super Offer	Page 69
Course PAS2JS	Page 74



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	Marco Cantù www.marcocantu.com marco.cantu@gmail.com
David Dirkse www.davdata.nl E-mail: David@davdata.nl	Benno Evers b.evers@everscustomtechnology.nl	Bruno Fierens www.tmssoftware.com bruno.fierens@tmssoftware.com
Holger Flick holger@flicxments.com		
	Mattias Gärtner nc-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
Boian Mitov mitov@mitov.com	Jeremy North jeremy.north@gmail.com	Detlef Overbeek - Editor in Chief www.blaiseascal.eu editor@blaiseascal.eu
Howard Page Clark hdpc@talktalk.net	Heiko Rempel info@rompelsoft.de	Wim Van Ingen Schenau -Editor wisone@xs4all.nl
Rik Smit rik@blaiseascal.eu	Bob Swart www.eBob42.com Bob@eBob42.com	B.J. Rao contact@intricad.com
Daniele Teti www.danieleteti.it d.teti@bittime.it		Anton Vogelaar ajv@vogelaar-electronics.com
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@blaiseascal.eu

Trademarks All trademarks used are acknowledged as the property of their respective owners.

CMarkenzeichen 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.

Abonnements (2022 Preise)

	Internat. excl. VAT	Internat. incl. 9% VAT	+ Shipment	TOTAL
Gedruckte Exemplare ±60 Seiten	€ 200,00	€ 218	€ 130,00	€ 348,00
Elektronische Download-Ausgabe ±60 Seiten	€ 64,20	€ 70		€ 70

Mitglied der Königlich
Niederländischen Bibliothek

KB
KONINKLIJKE BIBLIOTHEEK



Abonnemente können online unter <https://www.blaiseascalmagazine.eu/deutsche-ausgabe/> oder per schriftlicher Bestellung abgeschlossen werden oder indem Sie eine E-Mail an office@blaiseascal.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: Stichting Pro Pascal (Stichting Programmeertaal Pascal)

IBAN: NL82 ABNA 0441960863 BIC ABNANL2A Umsatzsteuer-Nr.: 81 42 54 147 (Stichting Programmeertaal Pascal)

Abonnementabteilung Edelsteinenbaan 21 / 3402 XA Usselstein, Niederlande Mobil: + 31 (0) 6 21.23.62.68 office@blaiseascal.eu

Copyright notice

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 Programmlisten und Code ist ohne die schriftliche Genehmigung des Autors untersagt.



Von Ihrem Herausgeber

Zu unserer großen Freude haben wir auch eine deutschsprachige Version dieser Ausgabe erstellt. Wir machen das zum ersten Mal und haben noch viel zu lernen.

Daher bitten wir auch um Unterstützung durch Sprach- und Schreibkorrekturen.

Wir möchten Sie bitten, uns zu helfen.

Indem wir weitere Sprachen hinzufügen, machen wir die Sprache **Pascal** immer vertrauter und leichter verständlich.

Gerade in dieser Ausgabe haben wir einen Artikel über die einfachste Installation von Lazarus hinzugefügt: **FPC de Luxe**.

Es war noch nie so einfach

(*dank LongDirtyAnimAlf alias Alfred Glänzer*), es zu installieren und dieses Programm bietet bereits eine große Anzahl von Sprachen zur Bedienung an.

Sie können es ganz einfach mit fast einem einzigen Mausklick installieren, aber es eignet sich auch hervorragend für separate Anforderungen. Ab sofort ist die Erstellung einer Stammversion ein Kinderspiel...

Da wir gerade über **Lazarus** sprechen, kann ich schon einmal ankündigen, dass wir in der Dezember-Ausgabe einen Blick in die Zukunft werfen werden und Sie werden überrascht sein, was dann alles möglich sein wird...

Das **PAS2JS** zu einem sehr wichtigen Internet-Tool wird - so dass Sie Ihre eigene Website sehr einfach mit **Pascal** erstellen und auch verwalten können - haben wir geplant, einen Kurs dafür einzurichten. Mehr dazu in Kürze.

Dies wird dann durch ein Buch unterstützt, das alle Aspekte von **PAS2JS** beleuchtet, so dass es gut zum Kurs passt.

Der Kurs wird übrigens für maximal 12 Anmeldungen gleichzeitig vergeben.

Wenn sich mehr Kursteilnehmer anmelden, werden wir mehrere Termine reservieren.

Apropos buchen:

Das letzte Treffen hier in den Niederlanden war sehr erfolgreich und wir werden an diesem Ort bleiben.

Sie wird im März nächsten Jahres wieder stattfinden, da wir nun hoffen, keine Probleme mehr mit Krankheiten zu haben...

Delphi verfügt über eine neue Möglichkeit, neue Komponenten lokal hinzuzufügen und sie für Dritte verfügbar zu machen.

Danny Wind schreibt etwas darüber.

Ich habe noch ein paar weitere Dinge beschrieben, die die Zukunft angeht:

die Verwendung von Lasertechniken nicht nur für die Herstellung von Chips, sondern deren vollständige Ersetzung durch Chips, die Laser als Grundelement haben und somit vollständig mit lichtgesteuerten CPUs arbeiten.

Das allein würde bedeuten, dass wir den Energieverbrauch weltweit um mehr als 10% senken könnten.

Das ist eine weitere gute Nachricht.

In dieser Ausgabe geht es auch um die Sicherung des Internets mit Hilfe der Quantum-Technologie.

Natürlich würden Sie in der heutigen Zeit erwarten, dass wir uns auf klimafreundliche oder verbesserte Prozesse konzentrieren.

Nennen Sie uns Beispiele dafür, wie Sie das Programmieren nutzen können, um die Auswirkungen auf das Klima zu verringern oder zu beseitigen.

Senden Sie Ihre Idee an editor@blaisepascalmagazine.eu

Viele interessante Dinge zu lesen und zu tun, denke ich....





*Mein Telefon ist nass geworden,
also gieße ich einen trockenen Martini darüber,
um es zu trocknen.*



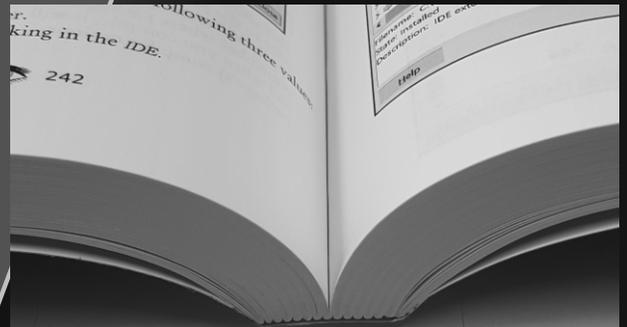
Lazarus Handbook Pocket

Preis: € 26,50
Exklusive
Mehrwertsteuer
und Versand

- Englisch
- Gedruckt in schwarz/weiß
- 2 Bände
- Inklusieve PDF
- 934 Seiten
- Gewicht: 2kg
- Extra geschützt
- Inklusive 40 Beispiele Projekte und Zusätzliche Programme



printed on FSC paper
<https://fsc.org/en/forest-management-certification>



<https://www.blaisepascalmagazine.eu/product-category/books/>



läuft unter Python3, Delphi, Jupyter-Notebook, Lazarus und maXbox4.

EINLEITUNG

Dieses Data-Science-Tutorial erklärt die so genannte **AGSI-Datenspeicherung** und seine Visualisierung der Zeitserie. **AGSI** steht für **Aggregated Gas Storage Inventory** und bietet Ihnen die Möglichkeit, immer auf dem Laufenden zu bleiben, wenn eine neue Serviceankündigung oder ein Update von einem unserer Datenanbieter auf der Website veröffentlicht wird. Die **Gas Infrastructure Europe (GIE)** stellt auch verwandte Daten wie die Storage Map und die Storage Investment Database unter <https://www.gie.eu/publications/maps/> zur Verfügung.

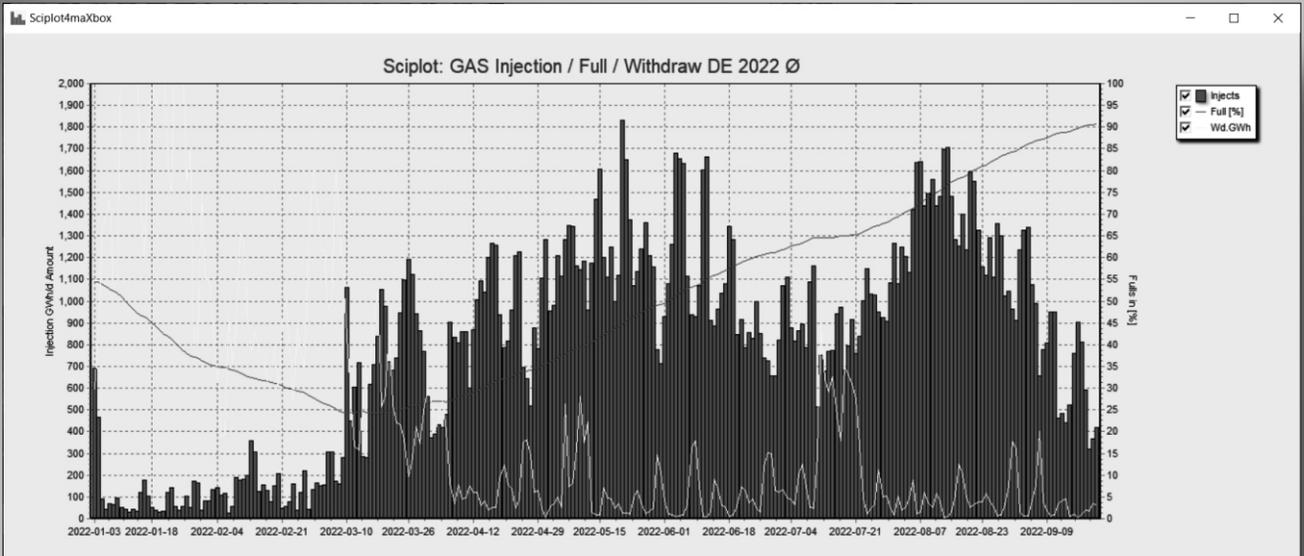


Abbildung 1: Das Ergebnis der Daten ist das Diagramm.

Wir verwenden `WinHttp.WinHttpRequest`, `JSONObjects` und die **TEECharts-Bibliothek** beim Laden und Testen des Plots. Außerdem wird ein **API-Schlüssel** benötigt, den Sie zunächst unter <https://agsi.gie.eu/account> finden.

Daten werden jeden Tag um 19:30 Uhr MEZ und ein zweites Mal um 23:00 Uhr aktualisiert. Bevor wir in den Code eintauchen, hier der Hauptteil des Skripts:

```
plotform:= getForm2(1400, 600, clsilver, 'Sciplot4maXbox');  
plotform.icon.loadfromresource(hinstance, 'ZHISTOGRAM');  
  
HttpResponse:=  
  getEnergyStreamJSON2(URL_AGSIAPI2, 'DE,2022-01-03,150', AGSI_APIKEY);  
JSON2Plot(plotform, letGenerateJSON2(HttpResponse));
```

starter

Der Hauptteil erzeugt ein Formular, ruft die **API** auf und zeichnet die Daten auf. **GIE** bietet einen **API-Dienst** (Application Programming Interface) auf seiner **AGSI** und **ALSI** Plattformen zur Veröffentlichung von Transparenz.

Über den API-Zugang können die Verbraucher die **AGSI-** und **ALSI-Website** umgehen und direkt und kontinuierlich auf die Daten zugreifen. Er ermöglicht das Extrahieren, Filtern und Aggregieren der Daten und das Erstellen beliebiger Teilmengen, ohne jeden Datensatz einzeln von der Website herunterladen zu müssen. Das **API-Exportformat** ist **JSON**.
Zum Beispiel eine Teilmenge von 150 Tagen:

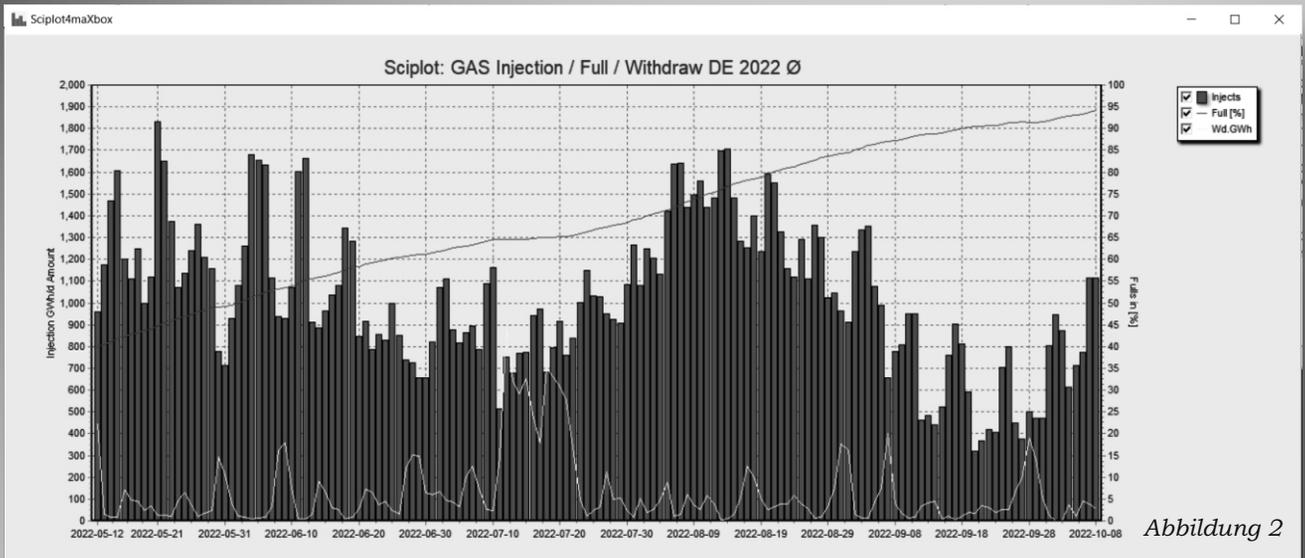


Abbildung 2

Die veröffentlichten Datensätze basieren auf der **EIC-Code-Zuordnungstabelle**, die **ACER** zur Verfügung gestellt wurde. Die Speicherdaten sind nach Unternehmen und Land aggregiert. Mit dem Anruf übergebe ich Land, Startdatum und Anzahl der Tage:

```
_getEnergyStreamJSON2(URL_AGSI_API2,'DE,2022-01-03,150',AGSI_APIKEY);
```

Alle verfügbaren Datensätze können auch im **Excel-**, **CSV-** und **JSON-Format** heruntergeladen werden. Die Daten in diesem Bericht zeigen eine aggregierte Ansicht - einzelne Datensätze nach Unternehmen und Lagerstätte sind ebenfalls zugänglich.

Beginnen wir also mit dem **API-Aufruf**:

```
_HttpResponse := _getEnergyStreamJSON2(URL_AGSI_API2,'DE,2022-01-03,150',AGSI_APIKEY);
```

Dieser Befehl und dieses Skript führen `WinHttp.WinHttpRequest` aus. Wenn Sie scheitern, erhalten Sie eine Reihe von Ausnahmen wie die folgende:

EXCEPTION: WinHttp.WinHttpRequest: Die für diesen Vorgang erforderlichen Daten sind noch nicht verfügbar, oder Sie haben einen gültigen Schlüssel vergessen:



```
AGSIPost: Failed at getting response: 403{
  "error": {
    "code": 403,
    "message": "The request is missing a valid API key.",
    "status": "PERMISSION_DENIED"
  }
}
```

Das Lustige ist die **JSON**-formatierte Ausnahme.

Seien Sie auch vorsichtig bei der Offenlegung Ihres Schlüssels, den ich von **Git** erhalte: GitGuardian hat den folgenden Google API-Schlüssel in Ihrem GitHub-Konto entdeckt.

Als nächstes folgt die Formatierung des get-Aufrufs mit einer gültigen **API-Schlüsselanforderung** in der Funktion **energyStream()**

```
function getEnergyStreamJSON2(AURL, feedstream, aApikey:string): string;
...
  encodURL:= Format(AURL,[HTTPEncode(asp[0]),(asp[1]),asp[2]]);
  writeln(encodurl) //debug
  hr:= httpRq.Open('GET', encodURL, false);
  httpRq.setRequestheader('user-agent',USERAGENTE);
  httpRq.setRequestheader('x-key',aAPIkey);
  ...
```

Und wo ist der fabelhafte Content-Type? Soweit ich verstanden habe, gibt es nur zwei Stellen in einer Web-Anfrage, an denen ein Content-Type festgelegt werden kann:

- ❶ Der Client legt einen Inhaltstyp für den Body fest, den er an den Server sendet (z.B. für get und post ist das Senden an den Server (z.B. für get und post) .
- ❷ Der Server legt einen Inhaltstyp für die Antwort fest.

Ein Absender, der eine Nachricht generiert, die einen Nutzdatenkörper enthält, muss ein Content-Type-Header-Feld in dieser Nachricht generieren, es sei denn, der beabsichtigte Medientyp der enthaltenen Darstellung ist dem Absender nicht bekannt; andernfalls können wir keine Antwort erhalten: 503503 - Dienst nicht verfügbar.

```
('headers={"Content-Type":"application/json"}')
httpRq.setRequestheader('Content-Type',application/json);
```

Es bedeutet, dass der HTTP-Header content-type nur für **PUT** gesetzt werden sollte und **POST** Anfragen. GET-Anfragen können "Accept"-Kopfzeilen haben, die angeben, welche Arten von Inhalten der Client versteht. Der Server kann dann anhand dieser Angaben entscheiden, welchen Inhaltstyp er zurücksenden soll. Als Option können Sie auch **TALWinInetHttpClient** verwenden. Es ist ein einfach zu bedienendes WinInet-basiertes Protokoll und unterstützt HTTPs. HTTP-Client-Komponente, mit der Sie beliebige Daten aus dem Web über das HTTP-Protokoll posten und abrufen können.

```
function TALHttpClient_Get(aUrl:AnsiString; feedstream, aApikey: string): string;
Var LHttpClient: TALWininetHttpClient; asp: TStringArray;
begin
  LHttpClient:= TALWininetHttpClient.create;
  asp:= splitStr(feedstream,',');
  LHttpClient.url:= Format(AURL,[HTTPEncode(asp[0]),(asp[1]),asp[2]]);
  LHttpClient.RequestMethod:= HTTPmt_Get; //HTTPrm_Post;
  LHttpClient.RequestHeader.UserAgent:=USERAGENTE;
  //LHttpClient.RequestHeader.CustomHeaders:=
  LHttpClient.RequestHeader.RawHeaderText:='x-key:'+aAPIkey;
  try
    result:= LHttpClient.Get1(LHttpClient.url); //overload;
  finally
    LHttpClient.Free;
  end;
end;
```

Alle fehlenden oder unvollständigen Daten werden auch auf **AGSI** angezeigt.
Als nächstes gehen wir zur Konvertierung unserer **JSON**-Antwort für die Darstellung mit **TJSONObject** über:

```
function letGenerateJSON2(HttpRqresponseText: string): TJSONArray;
var jo: TJSONObject;
begin
  jo:= TJSONObject.Create4(HttpRqresponseText);
  try
    //writeln(jo.getstring('data'));
    writeln(itoa(jo.getjsonarray('data').getjsonobject(0).length))
    writeln(itoa(jo.getjsonarray('data').length))
    result:= jo.getjsonarray('data');
    //write out to check
    for it:= 0 to result.length-1 do
      writeln(result.getjsonobject(it).getstring('gasDayStart')+''+
        result.getjsonobject(it).getstring('injection'));
    except
      writeln('EJson: '+ExceptionToString(exceptiontype, exceptionparam));
    end;
end;
```

Und dieses **JSON-Array**, das die obige Funktion zurückgibt, übergeben wir an den nächsten Plot:

```
procedure JSON2Plot(form1: TForm; jar: TJSONArray);
var chart1: TChart; cnt: integer; sumup,tmp2,tmp: double; gday: string;
begin
  form1.onclose:= @Form_CloseClick;
  chart1:= ChartInjector(form1);
  sumup:=0; tmp2:=0; tmp:=0;
  try
    for cnt:= 0 to jar.length-1 do
      begin
        //writeln(locate.getjsonobject(it).getstring('gasDayStart')+''+
        tmp:= jar.getjsonobject(jar.length-1-cnt).getdouble('injection');
        tmp2:= jar.getjsonobject(jar.length-1-cnt).getdouble('full');
        sumup:= sumup+tmp;
        gday:= jar.getjsonobject(jar.length-1-cnt).getstring('gasDayStart');
        chart1.Series[0].Addxy(cnt,tmp,gday,clgreen);
        chart1.Series[1].Addxy(cnt,tmp2,"",clred);
        chart1.Series[2].Addxy(cnt,jar.getjsonobject(jar.length-1-cnt).getdouble('withdrawal'),"",clyellow);
      end;
    except
      writeln('EPlot: '+ExceptionToString(exceptiontype, exceptionparam));
    end;
  PrintF('Landrange %d: Injection sum: %.2f',[jar.length-1,sumup]);
end;
```

Wie Sie sehen, haben wir 4 Serien zu zeichnen (einschließlich Zeitleiste):

- ① **Injection** (Injektion während des Gastages)
- ② **Vollständig** (Speicher / WGV (in%))
- ③ **Entnahme** (Entnahme während des Gastages (2-stellige Genauigkeit)).
- ④ **GasDayStart** (Der Beginn des gemeldeten Gastages)

Die Zeitreihe ist ein Ergebnis des Gastages und ein Trend ist verfügbar.

"Gastag" bezeichnet den Zeitraum von 5:00 Uhr bis 5:00 Uhr UTC des folgenden Tages bei Winterzeit und von 4:00 Uhr bis 4:00 Uhr UTC des folgenden Tages, wenn Sommerzeit gilt.

Der Gastag ist als UTC+1 für die MEZ oder UTC+2 in der Sommerzeit für die MESZ zu interpretieren (Definition: siehe CAM Network Code specific-cations).

Der API-Zugriff erfolgt über eine **REST**-ähnliche Schnittstelle (Representational State Transfer), die Datenbankressourcen in einem **JSON-Format** mit Content-Type im erwähnten Response Header bereitstellt.

Response Headers

```
X-Firefox-headers: h3
alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400
cache-control: no-cache, private
cf-cache-status: DYNAMIC
cf-ray: 7580b85f6dae0200-ZRH
content-encoding: br
content-type: application/json
date: Mon, 10 Oct 2022 16:26:53 GMT
server: cloudflare
x-robots-tag: noindex
```

Request Headers

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en;q=0.5
Alt-Used: agsi.gie.eu
Connection: keep-alive
Host: agsi.gie.eu
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Sec-GPC: 1
TE: trailers
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0
```

Abbildung 3.

Der Code der **Data Science Vision** enthält Anwendungsbeispiele und läuft unter **Python3, Delphi, Jupyter- Notebook, Lazarus und maXbox4**. **HINWEIS:** Der API-Service wird der Öffentlichkeit kostenlos zur Verfügung gestellt. Es werden nur die Daten zur Verfügung gestellt, die derzeit auf den Plattformen verfügbar sind.

Tipp: Um Daten direkt aus dem System zu extrahieren, können Sie in einem Browser auf einen der folgenden Links klicken (Web Verkehr gegenüber API-Verkehr):

AGSI+ <https://agsi.gie.eu/api?type=eu>

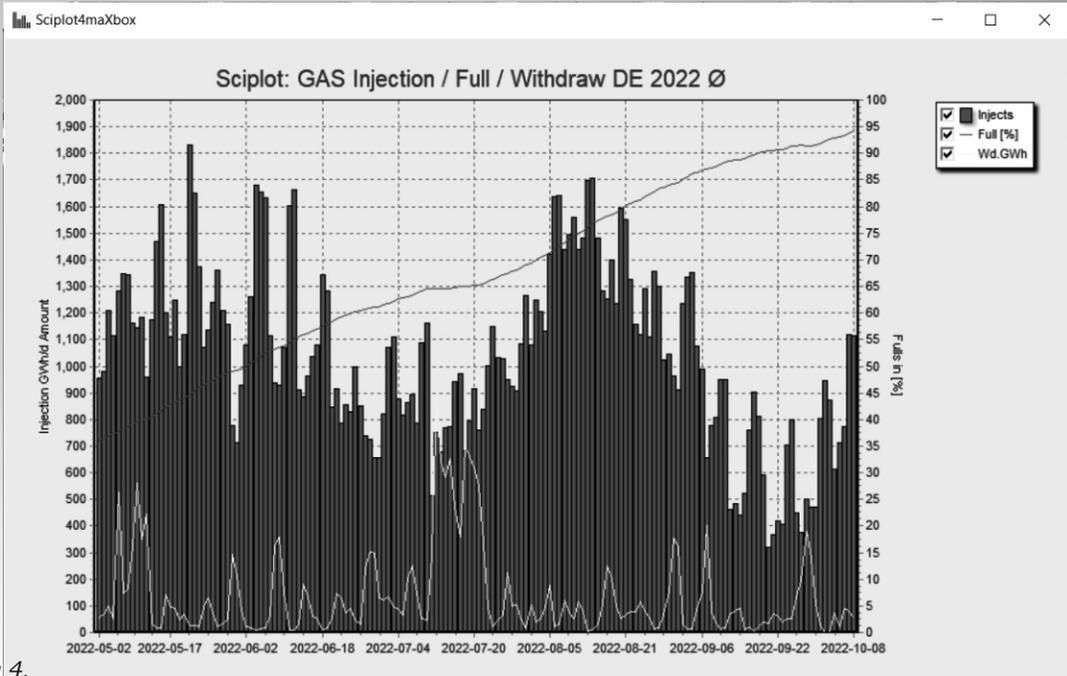


Abbildung 4.

Die Skripte und Bilder finden Sie unter: <https://github.com/maxkleiner/agsi-data>

REFERENZEN:

- <https://agsi.gie.eu/api>
- https://www.gie.eu/transparency-platform/GIE_API_documentation_v006.pdf
- https://svn.code.sf.net/p/alcinoe/code/demos/ALWinInetHTTPClient/_source/Unit1.pas
- <https://docwiki.embarcadero.com/Libraries/Sydney/en/System.Net.HttpClient.THTTPClient.Post>

Doc und Tool: <https://maxbox4.wordpress.com>

Script Ref: 1154_energy_api_agsi_plot14.txt

APPENDIX: zeigt eine WinAPIDownload-Klasse aus der maxbox4-Integration

{ *-----*)

```
TWinApiDownload = class(TObject)
private
    fEventWorkStart: TEventWorkStart;
    fEventWork: TEventWork;
    fEventWorkEnd: TEventWorkEnd;
    fEventError: TEventError;
    fURL: string;
    fUserAgent: string;
    fStop: Boolean;
    fActive: Boolean;
    fCachingEnabled: Boolean;
    fProgressUpdateInterval: Cardinal;
    function GetIsActive: Boolean;
public
    constructor Create;
    destructor Destroy; override;
    function CheckURL(aURL: string): Integer;
    function Download(Stream: TStream): Integer; overload;
    function Download(var res: string): Integer; overload;
    function ErrorCodeToMessageString(aErrorCode: Integer): string;
    procedure Stop;
    procedure Clear;
    property UserAgent: string read fUserAgent write fUserAgent;
    property URL: string read fURL write fURL;
    property DownloadActive: Boolean read GetIsActive;
    property CachingEnabled: Boolean read fCachingEnabled write fCachingEnabled;
    property UpdateInterval: Cardinal read fProgressUpdateInterval
        write fProgressUpdateInterval;
    property OnWorkStart: TEventWorkStart read fEventWorkStart
        write fEventWorkStart;
    property OnWork: TEventWork read fEventWork write fEventWork;
    property OnWorkEnd: TEventWorkEnd read fEventWorkEnd write fEventWorkEnd;
    property OnError: TEventError read fEventError write fEventError;
end;
```

The screenshot shows a script editor window with the following code:

```
in response body. }
434 // writeln('GPost: '+getPostGoogleAPI('http://www.ibm.ch', GoogleApiKey, ''));
435 // getURLScanAPI_Python('www.ibm.ch');
436
437 // writeln(GetEnergyStreamJSON('DE, 2022-03-01, 20', AGSI_APIKEY));
438 platform:= getForm2(1490, 600, clsilver, 'Sciplot4maxbox');
439 plotform.icon.loadfromresourceName(hinstance, 'ZHISTOGRAM');
440
441 HttpResponse:= getEnergyStreamJSON2(URL_AGSIAPI2, 'DE, 2022-01-03, 220', AGSI_APIKEY);
442 JSON2Plot(plotform, letGenerateJSON2(HttpResponse));
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
```

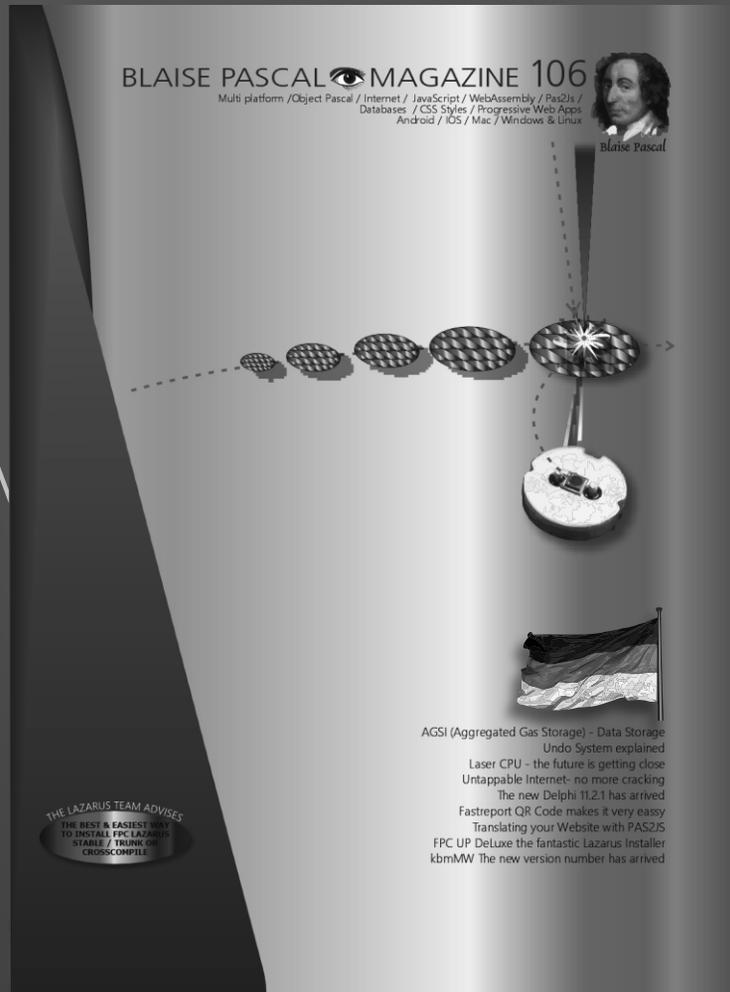
The chart displays 'Injeksi Gas (MWh)' on the left y-axis (0 to 2000) and 'Full (MWh)' on the right y-axis (0 to 100). The x-axis shows dates from 2022-03-06 to 2022-10-11. The legend indicates:

- Injeksi
- Full [M]
- Win GWh



BLAISE PASCAL MAGAZINE

ABONNEMENT DEUTSCHE AUSGABE



Preis: € 64,20
Exklusive Mehrwertsteuer

- Deutsch Sprachig
- PDF sowohl in Farbe wie schwarz/weiß
- 60 seiten oder mehr
- Inklusieve Code download
- Erscheint 10 mal pro Jahr

<https://www.blaisepascalmagazine.eu/deutsche-ausgabe/>

VON DANNY WIND

D11

starter

experte

KURZFASSUNG

Lokales GetIt Nur ab Delphi 11.2



Im Gegensatz zu seinem Namen ist das lokale **GetIt** nicht nur lokal. Es ist eine universelle Methode zur Installation von Komponenten und Bibliotheken in **Delphi** unter Verwendung einer Installationsdatei im **JSON-Textformat** und unterstützt sowohl lokale Dateien als auch Remote-Dateien. Bei der Verwendung lokaler Dateien kann es ohne Netzwerkverbindung ausgeführt werden und Sie können es für die Offline-Verteilung verwenden. Sie können die Elemente in der **JSON-Datei** auch auf Remote-Dateien von einer Website oder einem **GitHub-Repository** verweisen.

EINFÜHRUNG

Die einfachste lokale **JSON-GetIt-Datei** enthält nur ein paar Einträge. Einige beschreibende Informationen und einen Link zu einer Zip-Datei, einer lokalen Datei oder einem Download von einer HTTPS-Adresse. Anschließend wird die Zip-Datei kopiert oder heruntergeladen und entpackt und das Projekt kompiliert und in **Delphi** installiert. In dieser Einführung in das lokale **GetIt** beschreiben wir genau eine solche einfache lokale **JSON-GetIt-Datei**.

Das Schöne an dieser **JSON-Datei**-basierten Installation ist, dass sie es **Delphi**-Entwicklern ermöglicht, ihre Komponenten und Bibliotheken schnell und einfach mit anderen Entwicklern zu teilen.

Es basiert auf einer von Menschen lesbaren **JSON-Textdatei**, ohne dass externe Installationsprogramme oder Powershell-Skripte verwendet werden müssen.

Als **Delphi-Entwickler** müssen wir nur eine **JSON-Datei** zu unserem öffentlichen **GitHub-Repository** hinzufügen, was die Installation, Verwendung und Deinstallation unserer Komponenten sehr viel einfacher macht. Und weil die Datei für Menschen lesbar ist, kann jeder der Ihre Komponentenbibliothek installiert, in der Lage sein genau sehen, was das Installationsprogramm tut.

Kein versteckter Humbug.

Wie funktioniert das in der **Delphi IDE**?

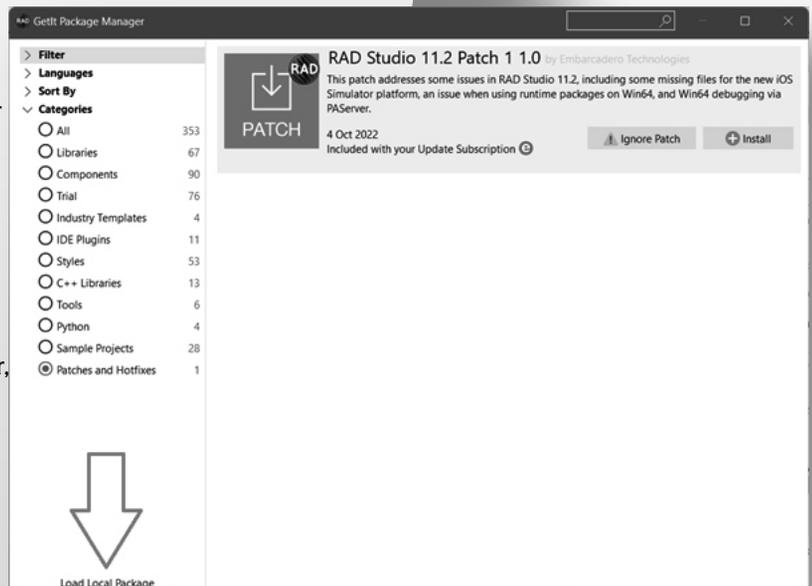
Sie laden eine **GetIt JSON-Datei** herunter, zum Beispiel aus einem **GitHub-Repository**, und speichern sie lokal.

Dann öffnen Sie den

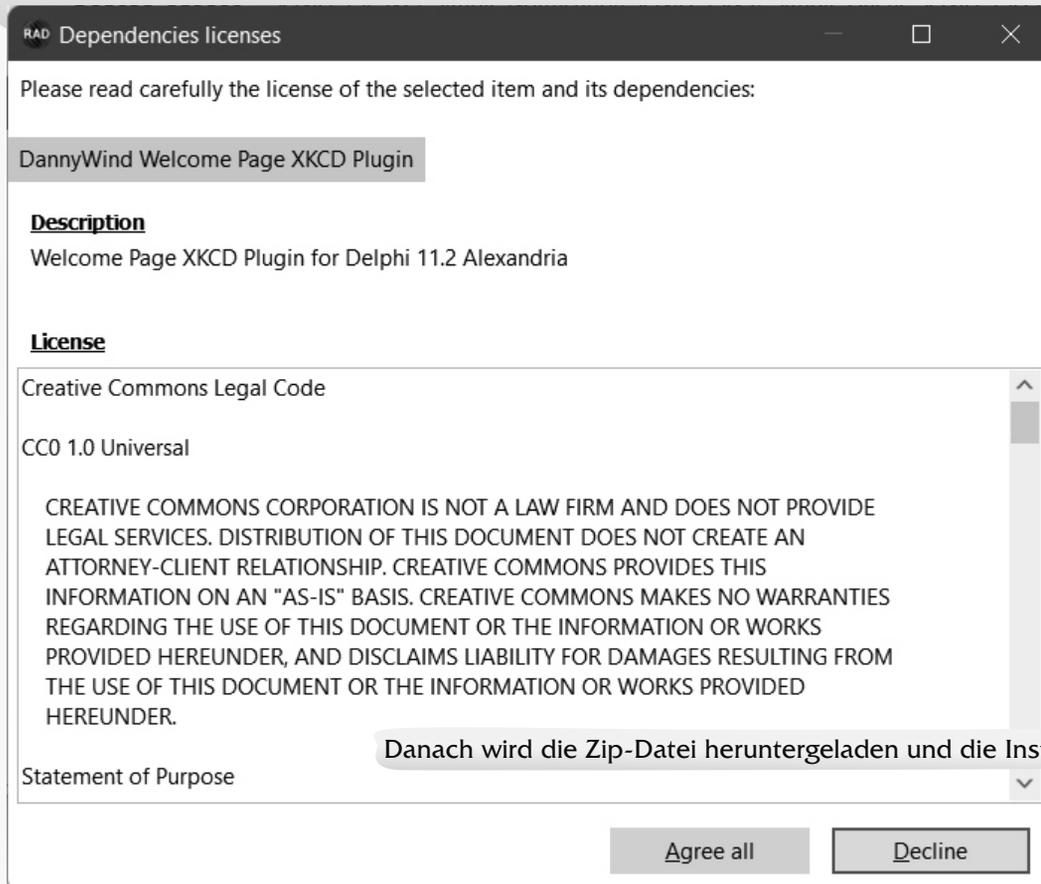
GetIt Package Manager

aus dem Menü **Tools** und klicken auf den kleinen Button unten links

"Lokales Paket laden".



Derzeit ist die Nutzung von Local GetIt an ein aktives Abonnement gebunden. Wenn Ihr Abonnement nicht aktiv ist, sehen Sie diese Schaltfläche möglicherweise nicht. Wählen Sie dann die JSON-Datei aus, und Local GetIt führt die entsprechenden Aktionen zur Installation durch. Einer der Standardschritte ist die Anzeige der Lizenzbedingungen.



Dependencies licenses

Please read carefully the license of the selected item and its dependencies:

DannyWind Welcome Page XKCD Plugin

Description
Welcome Page XKCD Plugin for Delphi 11.2 Alexandria

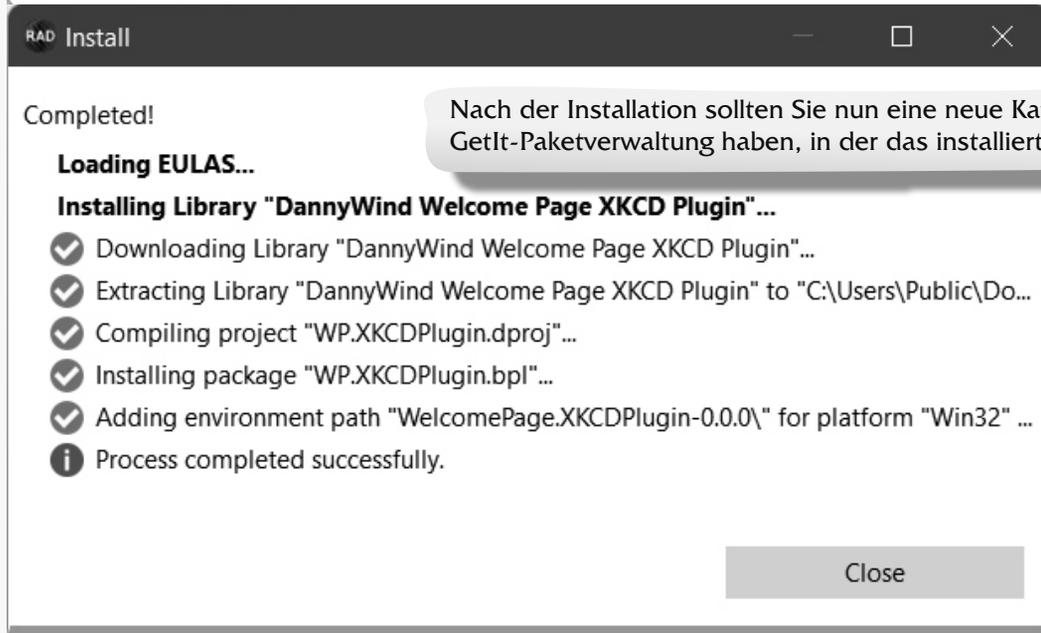
License
Creative Commons Legal Code
CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

Agree all Decline

Danach wird die Zip-Datei heruntergeladen und die Installation ausgeführt.



Install

Completed!

Loading EULAS...

Installing Library "DannyWind Welcome Page XKCD Plugin"...

- ✓ Downloading Library "DannyWind Welcome Page XKCD Plugin"...
- ✓ Extracting Library "DannyWind Welcome Page XKCD Plugin" to "C:\Users\Public\Do..."
- ✓ Compiling project "WP.XKCDPlugin.dproj"...
- ✓ Installing package "WP.XKCDPlugin.bpl"...
- ✓ Adding environment path "WelcomePage.XKCDPlugin-0.0.0\" for platform "Win32" ...
- i Process completed successfully.

Close

Nach der Installation sollten Sie nun eine neue Kategorie "Lokal" in der GetIt-Paketverwaltung haben, in der das installierte Paket sichtbar ist.





Das ist alles schön und gut, aber wie funktioniert das intern?
Werfen wir einen Blick auf den ersten Teil der **JSON GetIt-Datei**,
die für diese Installation verwendet wird.

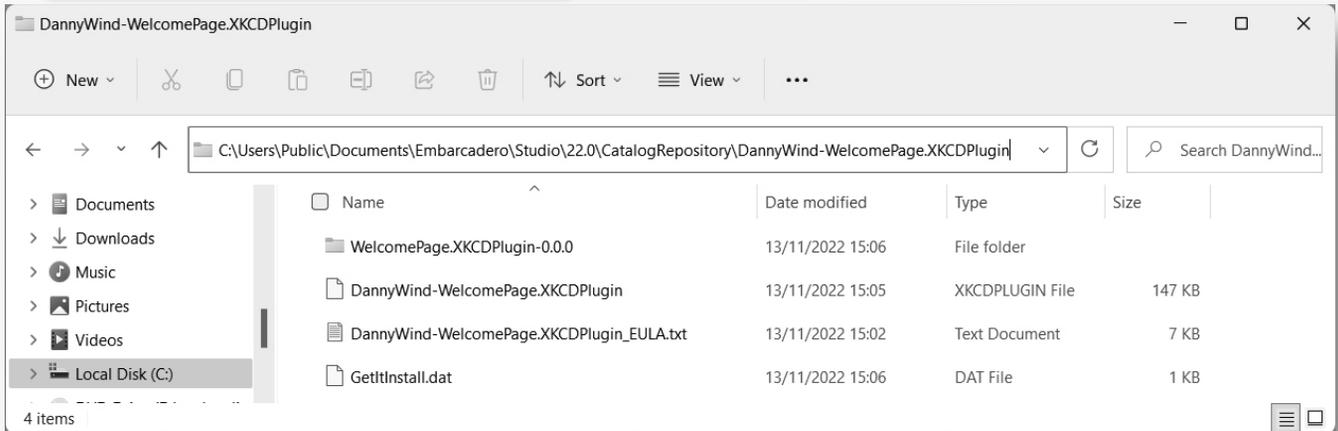
```
{
  "Id": "DannyWind-WelcomePage.XKCDPlugin",
  "Name": "DannyWind Welcome Page XKCD Plugin",
  "Version": "1",
  "Description": "Welcome Page XKCD Plugin for Delphi 11.2 Alexandria",
  "Vendor": "DannyWind",
  "VendorUrl": "https://dannywind.nl",
  "Image": "https://avatars.githubusercontent.com/u/78543577",
  "Tags": "DannyWind",
  "License":
    "https://github.com/DannyWindnl/WelcomePage.XKCDPlugin/raw/main/LICENSE",
  "Url":
    "https://github.com/DannyWindnl/WelcomePage.XKCDPlugin/raw/main/WelcomePage.XKCDPlugin-0.0.0.zip",
  "ProjectUrl": "https://github.com/DannyWindnl/WelcomePage.XKCDPlugin",
  "Modified": "2022-09-19 16:42:00",
  "LicenseName": "CC0-1.0 license",
  "TargetPath": "",
  "RequireElevation": "0",
  "AllUsers": "1",
  "Actions": [ [
  ]
  ]
}
```

Diese **JSON-Datei** installiert ein Welcome Page Plugin in **Delphi 11.2** aus einem **GitHub-Repository**. Um eine Umleitung zu vermeiden, verwenden wir die rohen Links zu den **GitHub-Dateien**. Die Lizenzdatei ist erforderlich. Bei der Installation zeigt das **GetIt**-Installationsprogramm sie an und fragt, ob der Benutzer damit einverstanden ist. Wenn ja, lädt es die Zip-Datei von der angegebenen Url herunter. Jetzt steht dort Url, aber Sie können hier auch eine lokale Datei angeben.

Verwendung einer lokalen Datei im gleichen Ordner wie die **JSON-Datei**:
"Url": "WelcomePage.XKCDPlugin-0.0.0.zip",



Anschließend wird diese Datei in das **CatalogRepository-Verzeichnis** entpackt



Der Inhalt der Zip-Datei wird in das Unterverzeichnis des Pakets mit dem Namen der **Zip-Datei WelcomePage.XKCDPlugin-0.0.0** entpackt. Aus dem obigen Format und den Bezeichnungen in der **JSON-Datei** gehen mehrere Dinge hervor. Wir müssen Schrägstriche vermeiden. Eine URL, die mit "https://" mit Escape-Zeichen beginnt, sieht in der **JSON-Datei** wie "https:\\\\" aus. Stellen Sie außerdem sicher, dass Sie die richtigen einfachen Anführungszeichen verwenden und sie nicht aus einem **Word-Dokument** kopieren und einfügen.

In diesem Beispiel habe ich **AllUsers** mit einem Wert von 1 verwendet, um in `$(BDSCatalogRepositoryAllUsers)` zu installieren, wie in der Abbildung zu sehen ist. Der empfohlene Standardwert ist 0, wodurch in `$(BDSCatalogRepository)` installiert wird und die Installation auf den Pfad des aktuellen Benutzers beschränkt wird.

Die **Id** ist erforderlich und muss eindeutig sein. **Id** wird auch verwendet, um die Reihenfolge zu bestimmen, in der Aktionen ausgeführt werden. Das Bild wird nach dem **GetIt** auf der Seite Installierte lokale Pakete angezeigt. Der Name wird für die Suche nach dem Paket in allen installierten **GetIt-Paketen** verwendet.

Da nur der erste Teil angezeigt wird, ist das Array **Actions** noch leer und wir müssen es mit Aktionen füllen, um das Plugin tatsächlich zu installieren. Unsere erste Aktion wird das Kompilieren des Projekts sein.

```
"Actions": [
{
  "Id": "1",
  "ActionId": "6",
  "Type": "2",
  "RequireElevation": "0",
  "Parameter": [
    {
      "Parameter": "WelcomePage.XKCDPlugin-0.0.0\\WP.XKCDPlugin.dproj"
    },
    {
      "Parameter": "Win32"
    },
    {
      "Parameter": "Release"
    }
  ],
  "ActionName": "CompileProject",
  "Description": "Compile WP.XKCDPlugin.dproj"
},
]
```



A couple of things to note here. The `ActionId` determines the `Action` that the Delphi IDE should perform. For `ActionId 6` the action is `CompileProject`. **NOTE** that the `ActionName` is completely optional and is just informative. However it is good to add this as it makes the **JSON** file more readable. **ActionId 6** has three parameters, the project file to compile, the list of platforms to compile (*Win32*) to and the configuration to use (*Release*).

After compiling we want to install the Package into the IDE.

```
{
  "Id": "4",
  "ActionId": "7",
  "Type": "3",
  "RequireElevation": "0",
  "Parameter": [
    {
      "Parameter": "WP.XKCDPlugin.bpl"
    }
  ],
  "ActionName": "InstallPackage",
  "Description": "Installs WP.XKCDPlugin.bpl package"
},
```

`ActionId 7` is `InstallPackage` and again the `ActionName` is purely descriptive.

NOTE that the `Type` here is 3 instead of 2.

The `Type` denotes the `Event` on which the `Action` needs to be performed.

For `Type 2` the `Event` is `after download`.

For `Type 3` the `Event` is `before install`.

As part of the install we also want to set the path for this package.

```
{
  "Id": "5",
  "ActionId": "1",
  "Type": "3",
  "RequireElevation": "0",
  "Parameter": [
    {
      "Parameter": "WelcomePage.XKCDPlugin-0.0.0\\"
    },
    {
      "Parameter": "cPasLibraryPath"
    },
    {
      "Parameter": "Delphi.Personality"
    },
    {
      "Parameter": "Win32"
    }
  ],
  "ActionName": "AddOptionPath",
  "Description": "Add library path for Delphi"
},
```



ActionId 1 is AddOptionPath,
where Parameter 1 is the Path,
Parameter 2 has different values for the **Delphi Pascal Library** (cPasLibraryPath),
the Delphi Browsing Path (cPasBrowsingPath) and some more.
The third Parameter originates from the ToolsAPI.pas file, which lists the personalities that can
be installed in the **Delphi IDE**.
The fourth parameter is optional and determines for which platform the path is added.
You can add an optional fifth parameter, a boolean, which when set to "False" only adds the
path to one of the compilers combined with the second parameter.
The default value is "true", which in most cases works out just fine.

At some point the package will be uninstalled.

```
{
  "Id": "14",
  "ActionId": "8",
  "Type": "5",
  "RequireElevation": "0",
  "Parameter": [
    {
      "Parameter": "WP.XKCDPlugin.bpl"
    }
  ],
  "ActionName": "UninstallPackage",
  "Description": "Uninstalls WP.XKCDPlugin.bpl package"
},
```

Where Type 5 is linked to Event "Before Uninstall". And the next step is to remove the added path.

```
{
  "Id": "15",
  "ActionId": "2",
  "Type": "5",
  "RequireElevation": "0",
  "Parameter": [
    {
      "Parameter": "WelcomePage.XKCDPlugin-0.0.0\\"
    },
    {
      "Parameter": "cPasLibraryPath"
    },
    {
      "Parameter": "Delphi.Personality"
    },
    {
      "Parameter": "Win32"
    }
  ],
  "ActionName": "RemoveOptionPath",
  "Description": "Remove library path for Delphi"
}
```



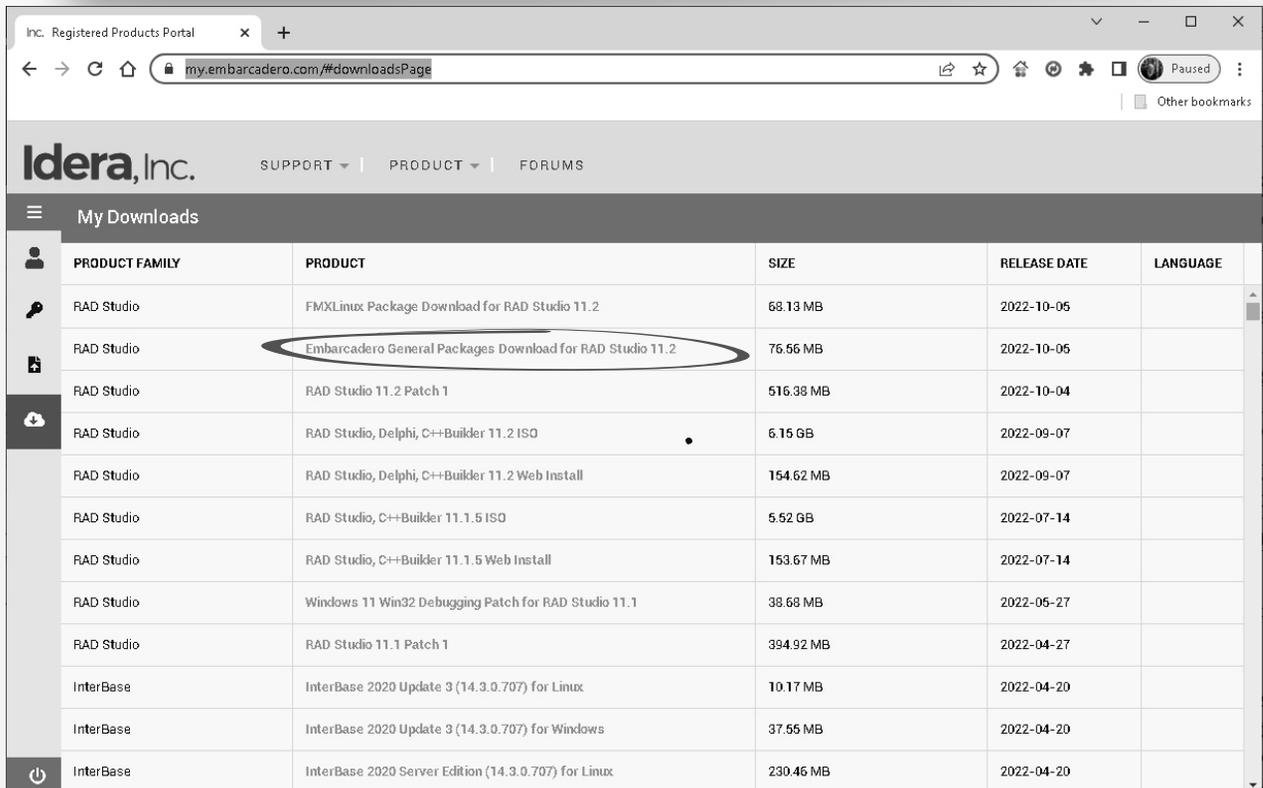
Auch mit dem Ereignis "Vor der Deinstallation" verknüpft.
Die sechs Ereignisse, die Sie verwenden können, sind in der Tabelle aufgeführt.

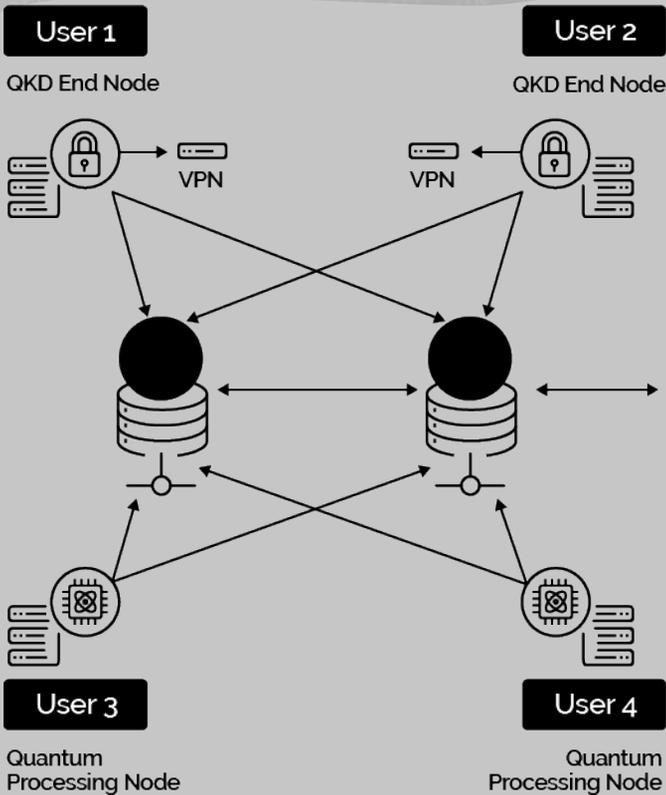
EventId "Type"

❶ Before Download	Wird ausgeführt, bevor das Paket-Zip heruntergeladen wird
❷ After Download	Wird ausgeführt, nachdem das Paket-Zip heruntergeladen und in das CatalogRepository extrahiert wurde
❸ Before Install	Wird vor der Installation des Pakets ausgeführt
❹ After Install	Wird nach der Installation des Pakets ausgeführt
❺ Before Uninstall	Wird ausgeführt, bevor das Paket aus dem CatalogRepository entfernt wird
❻ After Uninstall	Wird ausgeführt, nachdem das Paket und seine Dateien aus dem CatalogRepository

Embarcadero arbeitet an der Fertigstellung der Dokumentation für die lokale **GetIt-Unterstützung** und sie ist zum Zeitpunkt der Erstellung dieses Artikels noch nicht verfügbar. Sie können jedoch mehrere Beispiele für lokale **GetIt-JSON-Dateien** herunterladen über den Bereich Downloads unter <https://my.embarcadero.com/> wo Sie einige gebündelte lokale GetIt-Installationen finden, die für die Offline-Installation gedacht sind. Der "Embarcadero General Packages Download for RAD Studio 11.2" ist ein guter Ausgangspunkt und enthält mehrere **JSON-Beispiele** für die Installation von Paketen, die Sie als Grundlage für Ihre eigene lokale **GetIt JSON-Datei** verwenden können. (Siehe das Bild unten)

Das lokale **GetIt-Beispiel** für diesen Artikel finden Sie auf der **GitHub-Projektseite** unter <https://github.com/DannyWindnl/WelcomePage.XKCDPlugin>

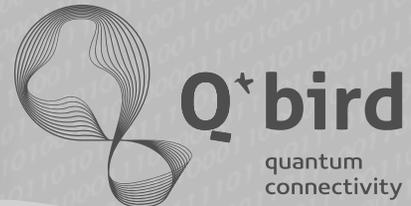




Endlich wurden die ersten Schritte zur Sicherung des Internets unternommen. Das ist vor allem für die Hafenbehörde von Rotterdam dringend notwendig. Denken Sie nur an die finanziellen und administrativen Gefahren. Außerdem könnte dies einen besseren Ausschluss der Unterwelt bedeuten. Die Akteure des Rotterdamer Hafens können sich beteiligen und von einem "ungenutzten" Multi-User-Quanten-Netzwerk für ihre kritischen Kommunikationssysteme profitieren.

Was bedeutet das für Sie?

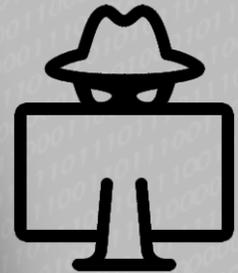
Eine ganze Reihe von Unternehmen und Firmen in verschiedenen Ländern experimentieren bereits mit Quantenkommunikation. Aber das niederländische Unternehmen "QuTech" mit seinem Spin-off-Unternehmen "Q*Bird" ist das erste, das eine neue Art von sicherem Quantennetzwerk einsetzt. Dieses kann mehrere Benutzer über einen zentralen Knotenpunkt auf skalierbare Weise verbinden (*was eine ständige Erweiterung ermöglicht*). Dies ermöglicht eine ungenutzte (*unzerbrechliche*) Internetverbindung zwischen mehreren Nutzern, die über das gesamte Hafengebiet verteilt sind. Der Hafen von Rotterdam ist ein wichtiger Teil der niederländischen Wirtschaft. Rotterdam schlägt jedes Jahr fast 15 Millionen Seecontainer um und ist damit einer der größten Häfen der Welt. Die Sicherung seiner Kommunikationssysteme verbessert die Sicherheit von zehntausenden von Seeschiffen jedes Jahr und den daraus resultierenden Wirtschaftsverkehr. Die Sicherung der Informationsübertragung über das Internet ist für jedes große Unternehmen entscheidend. Betrug in Informationskanälen hat erhebliche finanzielle Folgen und kann sogar lebensbedrohlich sein.



WIRTSCHAFTLICHE BEDEUTUNG

Der Hafen von Rotterdam ist ein wichtiges Industrie- und Logistikzentrum, in dem jährlich fast 15 Millionen Seecontainer umgeschlagen werden. Der Umschlag von trockenem und flüssigem Massengut (z.B. Öl) sowie von Stückgut beläuft sich auf insgesamt fast 500 Millionen Tonnen. Die gesamte wirtschaftliche Wertschöpfung des Hafens entspricht 8,2 % des niederländischen BIP (63 Mrd. €) und beschäftigt direkt und indirekt mehr als 500 000 Menschen.

Das Abhören von Kommunikationssystemen zwischen Parteien könnte zu erheblichen finanziellen Verlusten, Unterbrechungen kritischer Abläufe oder Sachschäden führen. Eine Quantenkommunikationsinfrastruktur trägt zu einer ungenutzten Verbindung bei und verbessert die Logistikketten, zu denen auch Rotterdam gehört, und damit eine lebenswichtige Infrastruktur auf dem europäischen Kontinent.



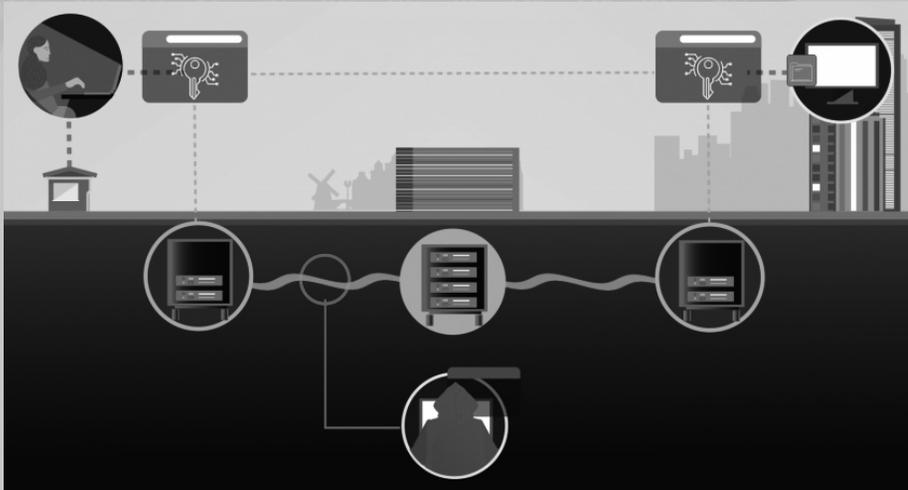


Abbildung 1: Diagramm des Quantenschlüsselverteilungssystems.
Schema erstellt mit einem Image: Simplot für QuTech.

Ein zentraler Knoten (C) verbindet die Benutzer A(links) und B(rechts). Wenn ein abhörender Hacker versucht, die geheimen Schlüssel zu erzwingen, sorgen die Gesetze der Quantenmechanik dafür, dass die Benutzer informiert werden, wenn die Schlüssel kompromittiert wurden. Ein neuer Satz von Schlüssel wird dann erstellt, um weitere Nachrichten sicher zu verschlüsseln

SPEZIELLE TECHNIK ZUR VERTEILUNG VON QUANTENSCHLÜSSELN ENTWICKELT VON Q*BIRD

Ein Team von Wissenschaftlern und Ingenieuren von QuTech - einer Zusammenarbeit zwischen der Technischen Universität Delft und TNO - hat eine alternative, bisher ungenutzte Methode zur Übertragung von verschlüsselten Informationen vorgestellt. Die Technologie hat ein derartiges kommerzielles Potenzial, dass sie beschlossen haben, sich von QuTech unter dem Namen "Q*Bird" abzuspalten.

Diese Technologie basiert auf einer speziellen Implementierung der Quantenschlüsselverteilung (QKD), die einen zentralen Knotenpunkt nutzt, um Benutzer zu verbinden, die sichere Kommunikation austauschen wollen. Das bedeutet, dass es möglich ist, kostengünstig ein Quanteninternet zu implementieren, das für mehrere Endnutzer skalierbar ist und größtenteils handelsübliche Geräte verwendet.“



(Die Quantenschlüsselverteilung (QKD))

WIKIPEDIA ist eine sichere Kommunikationsmethode, die ein kryptographisches Protokoll mit Komponenten der Quantenmechanik implementiert. Es ermöglicht zwei Parteien, einen gemeinsamen geheimen Zufallsschlüssel zu erzeugen, der nur ihnen bekannt ist und der dann zum Ver- und Entschlüsseln von Nachrichten verwendet werden kann.

Sie wird oft fälschlicherweise als Quantenkryptographie bezeichnet, da sie das bekannteste Beispiel für eine quantenkryptographische Aufgabe ist. .

Eine wichtige und einzigartige Eigenschaft der Quantenschlüsselverteilung ist die Fähigkeit der beiden kommunizierenden Benutzer, die Anwesenheit einer dritten Partei zu erkennen, die versucht, Kenntnis von dem Schlüssel zu erlangen.

Dies ergibt sich aus einem grundlegenden Aspekt der Quantenmechanik: Der Prozess der Messung eines Quantensystems stört das System im Allgemeinen. Ein Dritter, der versucht, den Schlüssel abzuhören, muss ihn auf irgendeine Art und Weise messen, was zu erkennbaren Anomalien führt.

Durch die Verwendung von Quantensuperposition oder Quantenverschränkung und die Übertragung von Informationen in Quantenzuständen kann ein Kommunikationssystem implementiert werden, das das Abhören erkennt. Wenn das Abhörniveau unter einem bestimmten Schwellenwert liegt, kann ein Schlüssel erzeugt werden, der garantiert sicher ist. ein Schlüssel erzeugt werden, der garantiert sicher ist (*d.h. der Abhörer hat keine Informationen über ihn*), andernfalls ist kein sicherer Schlüssel möglich und die Kommunikation wird abgebrochen.

Die Sicherheit der Verschlüsselung mit Quantenschlüsselverteilung beruht auf den Grundlagen der Quantenmechanik, im Gegensatz zur traditionellen Public-Key-Kryptographie, die sich auf die Rechenschwierigkeiten bestimmter mathematischer Funktionen stützt und keinen mathematischen Beweis für die tatsächliche Komplexität der Umkehrung der verwendeten Einwegfunktionen liefern kann. **QKD** hat eine nachweisbare Sicherheit, die auf der Informationstheorie und der Vorwärtsgeheimhaltung basiert.





Quantenschlüsselverteilung (QKD) - Fortsetzung



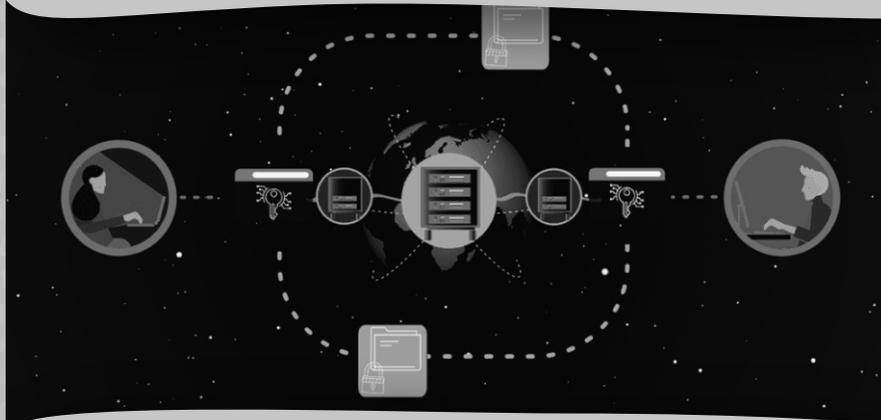
Der größte Nachteil der Quantenschlüsselverteilung ist, dass sie in der Regel von einem authentifizierten klassischen Kommunikationskanal abhängt. In der modernen Kryptographie bedeutet ein authentifizierter klassischer Kanal, dass man entweder bereits einen symmetrischen Schlüssel von ausreichender Länge oder öffentliche Schlüssel von ausreichendem Sicherheitsniveau ausgetauscht hat.

Wenn solche Informationen bereits verfügbar sind, kann man in der Praxis eine authentifizierte und ausreichend sichere Kommunikation erreichen, ohne QKD zu verwenden, z.B. durch die Verwendung des Galois/Counter-Modus des Advanced Encryption Standard.

QKD leistet also die Arbeit einer Stromchiffre zu einem Vielfachen der Kosten.

Die Quantenschlüsselverteilung wird nur zur Erzeugung und Verteilung eines Schlüssels verwendet, nicht zur Übertragung von Nachrichtendaten.

Dieser Schlüssel kann dann mit einem beliebigen Verschlüsselungsalgorithmus verwendet werden, um eine Nachricht zu verschlüsseln (und zu entschlüsseln), die dann über einen Standardkommunikationskanal übertragen werden kann. Der Algorithmus, der am häufigsten mit QKD in Verbindung gebracht wird, ist das One-Time-Pad, da es nachweislich sicher ist, wenn es mit einem geheimen, zufälligen Schlüssel verwendet wird. In der Praxis wird er oft auch mit Verschlüsselungsalgorithmen mit symmetrischem Schlüssel wie dem Advanced Encryption Standard Algorithmus verwendet.)



PRAKTISCHE UMSETZUNG

In einem ersten Schritt wird der zentrale Knotenpunkt für die Verteilung von Quantenschlüsseln bei der Hafenbehörde von Rotterdam untergebracht.

Auf der Grundlage dieser neuen Technologie werden Daten zwischen einer Reihe von Parteien im Hafen in einer abgeschirmten Umgebung ausgetauscht.

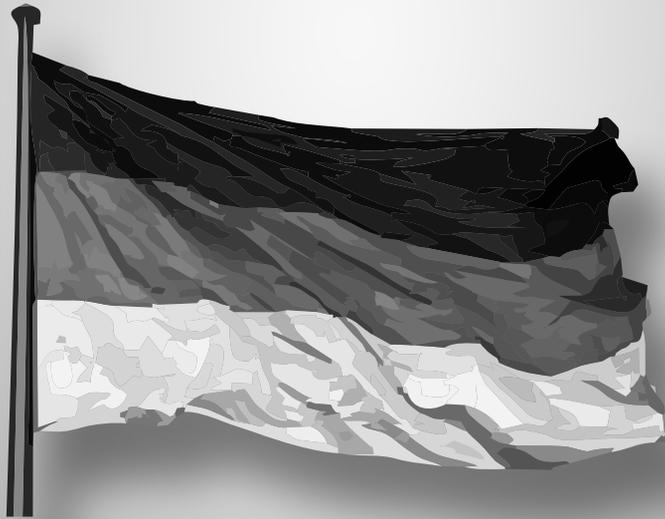
An diesem Test werden der **Hafenbetrieb Rotterdam**, **Portbase** und eine Reihe von nautischen Dienstleistern beteiligt sein. Ziel des Tests ist es, die technischen Möglichkeiten des Systems weiter zu validieren.

In Zukunft wird es mit einem solchen System möglich sein, allen Endnutzern eine sichere, nicht nachweisbare Verbindung zu bieten. Die Benutzer teilen sich Schlüssel, die mit Hilfe der Quantentechnologie (*und daher von Natur aus unknackbar und unangreifbar, da sich die Schlüssel durch Verschränkung in einem Stadium befinden, in dem sie ohne Änderung des Schlüssels nicht mehr zugänglich sind*), die sie dann zur Verschlüsselung von Nachrichten mit herkömmlicher Technologie verwenden werden.

Die Stärke dieses Systems liegt darin, dass es leicht auf viele weitere Benutzer ausgeweitet werden kann und dass die Kosten für die Erweiterung relativ gering sind. Nach der Fertigstellung können die beteiligten Parteien sicher sein, dass ihre Kommunikationslinie nicht manipuliert wurde. Aus technischer Sicht ermöglicht dieses Projekt eine enge Zusammenarbeit zwischen "**Single Quantum**" und "**Q*bird**": zwei niederländische High-Tech-Unternehmen mit komplementärem Know-how und globalen Ambitionen. Indem sie ihre Kräfte in diesem Projekt bündeln, werden sie eine starke soziale Wirkung haben.



Diese Ausgabe ist erhältlich
als deutsche Ausgabe und kann
kostenlos heruntergeladen werden



Wenn Sie eine kostenlose deutsche Ausgabe wünschen,
senden Sie Ihre Adresdaten und Ihre Emailadresse
an editor@blaisepascalmagazine.eu



Geschrieben in Delphi7,
 getestet in Delphi 11.2.1(Alexandria)
 und neu kompiliert in Lazarus

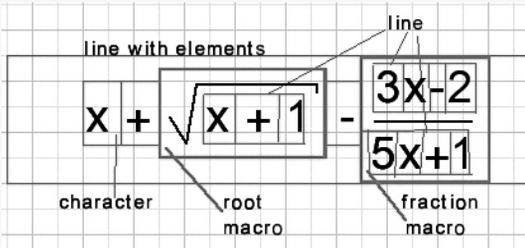
Einführung

Algebraische Texte benötigen zusätzliche Elemente, um Wurzeln, Brüche, Exponenten, Indizes, Vektoren, Matrizen und auch spezielle Symbole für Geometrie, Mengen und Logik zu implementieren. In diesem Artikel beschreibe ich die grundlegende Struktur eines solchen Editors. Nicht die Zeichenvorgänge, sondern die Bearbeitungsprozesse, die Folgendes umfassen Element einfügen, ersetzen, löschen und rückgängig machen. Der Editor ist **WYSIWYG** (What you see is what you get). Text wird auf die übliche Weise eingegeben. Spezielle Elemente können mit einem Mausklick aus Menüs ausgewählt werden. Es werden keine Skripte verwendet. Das Bild unten zeigt zwei Beispiele, die jeweils in wenigen Sekunden eingegeben wurden:

$$(a+b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$$

$$e = \lim_{x \rightarrow \infty} \left(1 + \frac{1}{x} \right)^x$$

Das ist die allgemeine Idee: Elemente werden zu einer Seite hinzugefügt, die das erste (*Nummer 1*) Element ist. Textelementtypen sind Seite, Rahmen, Tabelle, Zeichen, Zeile und Makro.
 Makroelemente enthalten eine Zeile (*oder Zeilen*) und grafische Elemente für Wurzeln, Klammern, Brüche ...
 Eine Zeile enthält Zeichen und Makros.



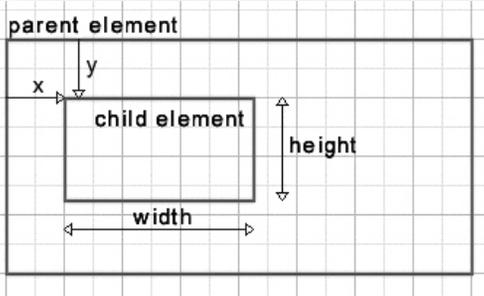
Die Textausrichtung ist vollständig automatisiert. Wenn Text in der Zeile einer Wurzel eingegeben wird, passt das Wurzelsymbol seine Größe an.
 Wenn Sie Text in der Nennerzeile eines Bruchs eingeben, wird die Position des Zählers so korrigiert, dass er zentriert bleibt.
 Klammern um eine Zeile werden vergrößert, wenn ein Bruchmakro hinzugefügt wird, das die Höhe der Zeile erhöht.



VERWURZELTE BÄUME

Wenn die Dimensionen des Elements A von der Größe der Elemente B und C abhängen, nennen wir A die Mutter von B und C. B und C sind die Kinder von A.

Für die Berechnung von Position und Größe benötigt ein Element diese Eigenschaften:



X: die linke Position relativ zum übergeordneten Element
Y: die obere Position im Verhältnis zum übergeordneten Element
Breite, Höhe: Größe des Elements.

Beachten Sie, dass eine Änderung der Position des übergeordneten Elements die x,y-Koordinaten der untergeordneten Elemente nicht verändert.

Der Grundgedanke ist folgender: Wenn die Größe eines untergeordneten Elements geändert wird, berechnet das übergeordnete Element die Position seiner untergeordneten Elemente sowie seine eigene Größe neu. Wenn sich die Größe des Elternteils ändert, berechnet der Elternteil....etc neu.

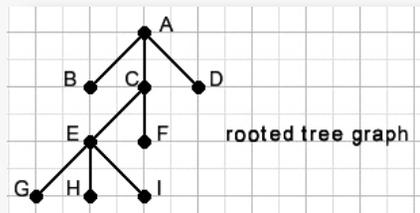
Elementtypspezifische Prozeduren sind:

- ◆ Berechnen seiner eigenen Breite und Höhe
- ◆ (Neu-)Berechnen der (x,y)-Position seiner Kinder
- ◆ Sich selbst auf einer Leinwand malen

Ein Element berechnet niemals seine eigene (x,y)-Position.

Bearbeitungsvorgänge (insert, delete, replace, undo) sind abstrakt und vom Elementtyp unabhängig. Das Diagramm, das die Abhängigkeiten der Elemente beschreibt, wird als verwurzelter Baum bezeichnet und ist ein spezieller Graph.

So wird ein verwurzelter Baum in der Mathematik dargestellt:



Die Punkte sind Elemente, die nach unten verlaufenden Linien verbinden die Kinder.

(In der Mathematik werden die Punkte als Scheitelpunkte bezeichnet, die Linien als Kanten)

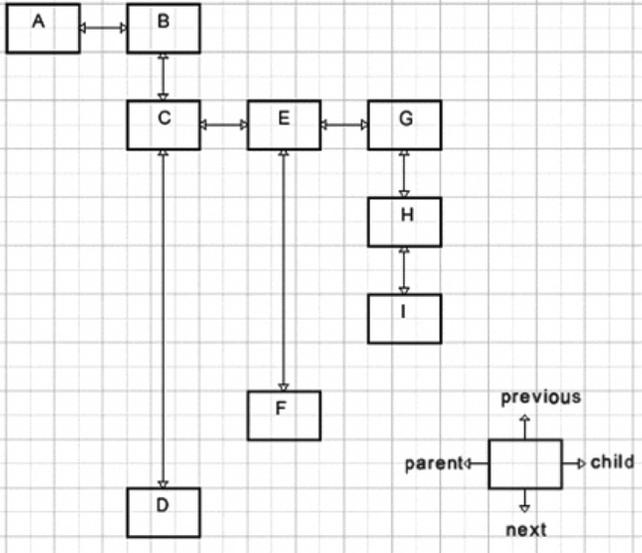
In Baumdiagrammen gibt es nur einen Pfad zwischen Punkten. Es gibt keine Schleifen.

In der obigen Abbildung sind B, C und D Kinder des Wurzelements A. C ist der Elternteil von E und F, G, H und I sind Kinder von E.

Hinweis: Dies ist ein australischer Baum, die Wurzel steht ganz oben und die Zweige wachsen nach unten.

Die obige Struktur ist auch aus Dateisystemen mit Ordnern bekannt, die Dateien und Ordner enthalten....

Für unseren Zweck ist die mathematische Darstellung nicht geeignet, das nächste Bild ist besser:



Um die Xtree-Prozeduren zu testen, wurde ein Übungsprogramm geschrieben.

```

const maxElement = 48; // maximum number of elements in exerciser
type TElementStatus = (esFree,esActive); // esFree: outside tree
TElement = record
  status: TElementStatus;
  mark : byte; // property 0,1,2,3 for background color
  posX,posY: byte; // position relative to parent
  parent: word;
  child : word;
  next : word;
  previous: word;
  Cstep: byte; // create step sequence number
  Dstep: byte; // delete (replaced) step number
end;
var element : array[1..maxelement] of TElement;
    
```

STATUS:

esFree : Element ist nicht aktiv, ist nicht Teil des Baums, keine Verbindung zu einem anderen Element.

esActive : Element wird verwendet, ist Teil des Baums mark ist ein Code für die Hintergrundfarbe des Elements. 0 ist weiß. Die Elemente werden auf einer Bitmap (Map) gezeichnet, die als Tabelle mit Spalten und Zeilen betrachtet wird.

posX ist die Spalte, **posY** ist die Zeile.

parent, child,next, previous :

Links zu anderen Elementen, um die Baumstruktur zu definieren.

Wenn keine Verknüpfung besteht, ist dieser Wert 0.

Die Bytes **Cstep**, **Dstep** ermöglichen Rückgängig-Operationen, siehe spätere Erklärung. Beachten Sie, dass das Element-Array eine bidirektional verknüpfte Liste enthält. Dies ist praktisch für die einfache Bewegung des Cursors.



OPERATIONEN BEARBEITEN

Insert (Einfügen)

placeA.....ein neues Element nach einem anderen Element einfügen

placeB.....ein neues Element vor ein anderes Element setzen

placeChildA....neues Element als letztes Kind platzieren

placeChildB....platziere neues Element als erstes Kind

Delete Löschen deaktiviert das Element einschließlich seiner Kinder.

Replace Ersetzen kombiniert einen Lösch- und einen Einfügevorgang, wodurch ein Element ersetzt wird. Die Kinder des gelöschten Elements werden mit dem ersetzenden Element verbunden.

Undo entfernt das zuletzt hinzugefügte Element oder reaktiviert ein gelöschtes Element.

DAS NEUE UNDO (RÜCKGÄNGIG)-SYSTEM

In früheren Versionen des Mathe-Editors war das Dokument das erste Element, Seiten waren Kinder des Dokuments. In späteren Versionen wurde das Dokument-Element abgeschafft und die Seite wurde zur Wurzel. Dies führte zu einem viel kleineren Element-Array, das jetzt nur noch die bearbeitete Seite enthält. Die Seitendaten werden auf dem Heap gespeichert und nur dann in das Element-Array geladen, wenn die Seite bearbeitet wird.

Die begrenzte Anzahl von Elementen ermöglicht dieses eingebettete Undo-System.

Für die Rückgängigmachung von Operationen müssen die Bearbeitungsaktionen pro Seite gespeichert werden. Zuvor wurden die Bearbeitungsdaten auf einen Stapel gelegt und für die Rückgängigmachung gepoppt. Wenn Sie zwischen den Seiten wechselten, wurden die Daten auf dem Stapel gelöscht. Das war unpraktisch. In diesem neuen Rückgängig-System werden gelöschte Elemente nicht mehr aus dem Baum herausgezogen, sondern nur noch in-aktiviert, wobei ihre Position im Baum erhalten bleibt.

Für die Undo-Prozesse mussten zwei Zähler eingeführt werden:

- **edStep** : Nummer der Bearbeitungsschrittfolge
- **edBase** : Nummer der Bearbeitungsbasis und eine Konstante
- **undoDepth** = 15, die Anzahl der Bearbeitungsschritte, die zurückgenommen werden können

Zu Beginn war $\text{edBase} = \text{edStep} = 0$. Es fand noch keine Bearbeitung statt. Vor dem Hinzufügen des ersten Elements: edBase und edStep sind beide auf 1 gesetzt.

Der $\text{Cstep} := \text{edStep}$ des neuen Elements zeigt an, dass dieses Element bei Schritt 1 erstellt wurde. Für jedes neue Zeichen wird edStep inkrementiert und sein Cstep -Wert auf edStep gesetzt. Nach dem Hinzufügen von 14 weiteren Zeichen ist edStep 15, edBase ist immer noch 1.

15 ist die maximale Anzahl von Schritten, die Sie sich merken können.

Bevor Sie also das nächste Element hinzufügen, wird edStep auf 16 erhöht, wodurch die maximale Anzahl (15) von Schritten überschritten wird. Jetzt wird edStep auf 1 zurückgesetzt und edBase wird auf 2 erhöht. Das hinzugefügte Element hat $\text{Cstep} := 1$, aber es gibt bereits ein Element mit $\text{Cstep} = 1$. Für dieses Element ist also $\text{Cstep} := 0$, womit es außerhalb des Bereichs für die Rückgängigmachung liegt. Der neue Rückgängig-Bereich ist 2, 3, ..., 15, 1.

Für jeden Bearbeitungsschritt muss also der gesamte Baum durchsucht werden. Das ist kein Problem, denn die maximale Anzahl der Elemente für eine einzelne Seite des Mathematik-Editors beträgt nur 7500.

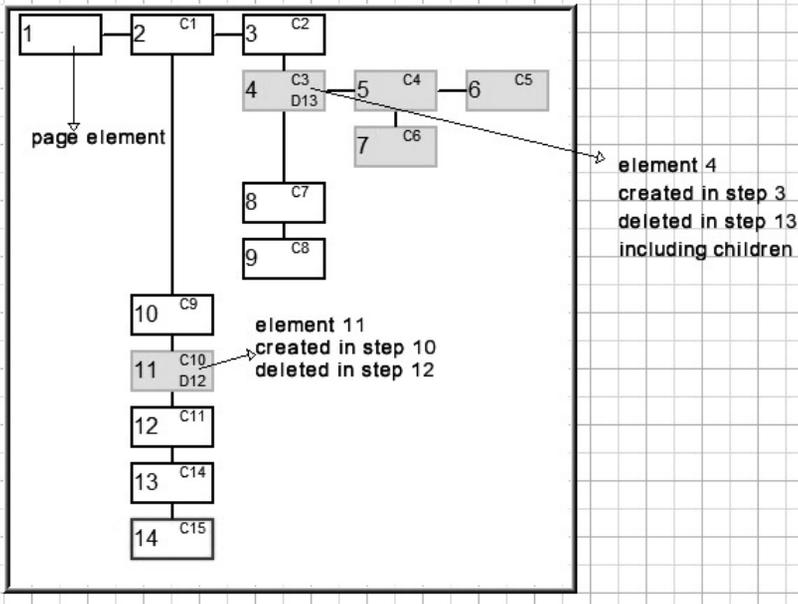
Wenn von diesem Punkt aus wiederholt Rückgängig gemacht wird, werden die Elemente mit $\text{Cstep} = 1, 15, 14, \dots, 3, 2$ aus dem Baum entfernt. Wenn $\text{edStep} = \text{edBase}$ nach dem Dekrementieren ist, sind keine Rückgängigmachungen mehr möglich und $\text{edBase} := 0$ und $\text{edStep} := 0$;

VERKNÜPFEN

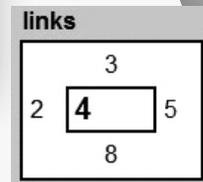
Wenn Sie eine Zeichentaste gedrückt halten, wird eine Zeile mit demselben Zeichen ausgegeben.
Wenn Sie ein Makroelement hinzufügen, werden auch Zeilenelemente hinzugefügt.
Es ist praktisch, diese Elemente mit einem einzigen Rückgängigmachen zu entfernen.
Deshalb gibt es ein `Linkflag`, das verhindert, dass der `edStep`-Zähler erhöht wird.
Die Prozedur `starttreelinking` setzt das `startflag`.
Der nächste Bearbeitungsvorgang setzt dann das `Linkflag`.

ANZEIGE DES BAUMÜBUNGSPROGRAMMS

Die folgende Abbildung zeigt eine typische Baumstruktur:

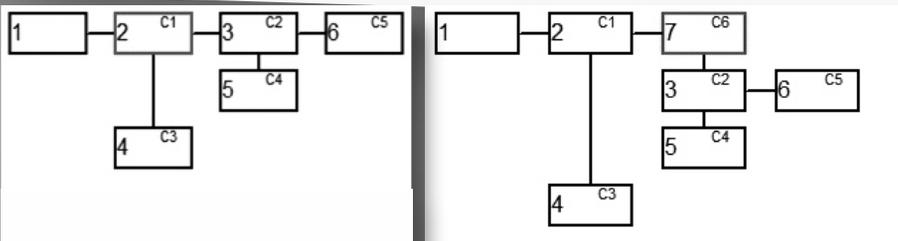


GELÖSCHTE ELEMENTE sind mit einem grauen Hintergrund versehen. In der obigen Abbildung ist Element 14 das aktive Element (*roter Rand*). Bearbeitungsvorgänge werden auf dieses Element angewendet. Das aktive Element wird durch einen Mausklick oder mit den Cursortasten ausgewählt. Die Eltern-Kind-Verknüpfungen und auch die Vorher-Nachher-Verknüpfungen zwischen den Elementen werden als eine einzige Linie dargestellt. Um die genauen Verknüpfungen zu sehen, bewegen Sie den Mauszeiger über das Element. Dann werden die Verknüpfungen in einer separaten `Paintbox` angezeigt:



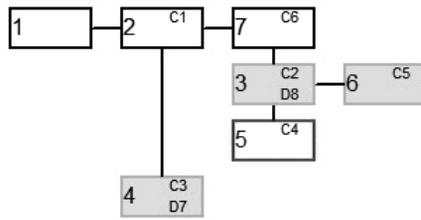
HINZUFÜGEN VON ELEMENTEN

Dazu müssen Sie ein freies Element auswählen, seine Eigenschaften festlegen, die Links setzen und die Prozedur `LinkIn()` auf, um das neue Element mit dem Baum zu verbinden. In der folgenden Abbildung wird das Element 7 als erstes Kind des aktiven Elements 2 hinzugefügt



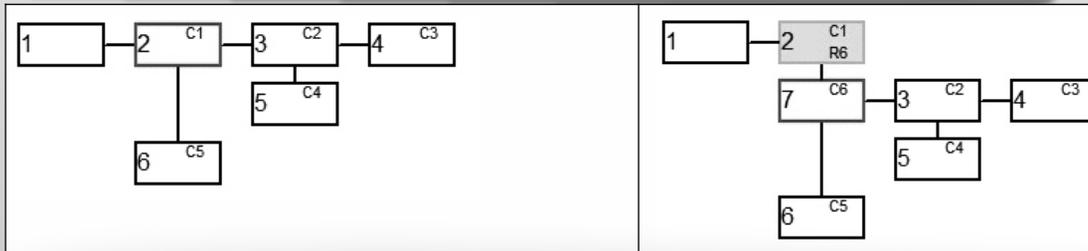
DELETING ELEMENTS

Siehe Bild. Die Elemente 4 und 3 werden anschließend gelöscht.
Hinweis: Element 6 hat Dstep=0, ist aber grau gefärbt, da sein Elternteil (3) gelöscht wird.



ERSETZEN (REPLACING) VON ELEMENTEN

In der Abbildung unten wird das Element 2 durch ein neues Element (7) ersetzt.



Das neue Element wird neben dem ersetzten Element platziert. Die Kinder von Element 2 werden in das neue Element 7 verschoben. Siehe Element 2: R6 bedeutet: ersetzt in Bearbeitungsschritt 6. Der übergeordnete Link der Kinder 3,5 muss auf 7 geändert werden.

Rückgängig (Undo) machen Wenn ein Element (sagen wir 20) in Schritt 10 gelöscht wird, wird sein Dstep-Wert auf 10 gesetzt. Wenn Edstep später auf 10 erhöht wird, liegt das Element 20 außerhalb des Bereichs für die Rückgängigmachung und wird zusammen mit seinen Kindern zerstört. Zerstören bedeutet, dass es aus dem Baum entfernt wird (durch die Prozedur Extract), seine Links auf 0 gesetzt werden und sein Status esFree wird, wodurch es als neues Element verfügbar wird.

Ersetzen von Elementen erkennen

Wenn Sie ein Element löschen, werden auch seine Kinder gelöscht. Bei einem ersetzten Element sind seine Kinder jedoch mit dem ersetzenden Element verbunden.

Wenn Sie ein ersetztes Element rückgängig machen, müssen die Kinder wieder angehängt werden.

Um die Erstellung und Löschung von der Ersetzung zu unterscheiden, wird Bit 7 der Cstep und Dstep Werte verwendet. Wenn dieses Bit gesetzt ist, bedeutet dies eine Ersetzung.

Da der edstep-Zähler vom Typ Byte ist, beträgt die maximal mögliche Anzahl der Rückgängigmachungen $2^7 - 1 = 127$.

Scannen des Baums

Für verschiedene Zwecke muss der Baum in geordneter Weise durchsucht werden. Die nächsten Suchvorgänge erfolgen in der Tiefe zuerst. Zuerst werden die Kinder ausgewählt, dann die nächsten, dann die vorherigen und zuletzt die Eltern. Der Baum wird von unten nach oben und dann von oben nach unten bis zur Wurzel durchlaufen.

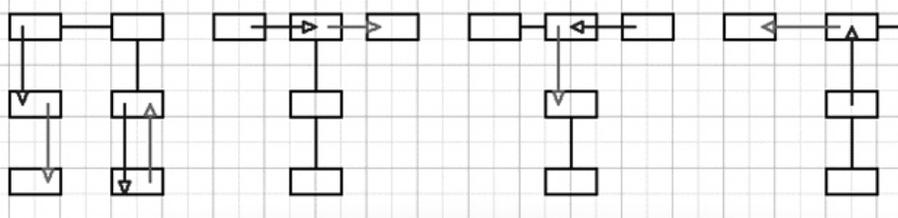


```
type TLink = (ltNone,ltParent,ltChild,ltNext,ltPrevious);
function ScanElement(var elmnt : word; lt : TLink) : TLink;
```

Es ist der Link zum Element `elmnt`. Die Funktion `scanElement` erstellt eine neue Verknüpfung zu einem anderen Element `elmnt` in der Baumstruktur.

Das nächste Bild zeigt das Scannen.

Die blaue Linie beginnt im vorherigen Element und zeigt auf das Element `elmnt`. Der grüne Pfeil zeigt auf das neue Element `elmnt` und das Ergebnis der Funktion ist der Link. Gelöschte Elemente werden übersprungen.



Eine ähnliche Funktion `fullscan(var elmnt : word; lt : TLink) : TLink;` Wählt auch die gelöschten Elemente im Baum aus.

SPEICHERN UND NEU LADEN

Um die Baumstruktur einschließlich der Rückgängig-Informationen zu speichern, genügt eine vorwärts verknüpfte Liste (*nur untergeordnete und nächste Links*).

Es ist nicht nötig, die vorherigen und übergeordneten Links zu speichern.

Nach dem Laden können die übergeordneten und vorherigen Verknüpfungen hinzugefügt werden.

KOMPRIMIEREN UND ERWEITERN

Ohne die Undo-Informationen zu speichern, können Bäume weiter komprimiert werden.

Wenn die Elemente B,C Kinder von A sind, schreiben wir den Baum als A(BC).

Wenn D,E,F Kinder von B sind : A(B(DEF)C) definiert den Baum.

Um die Baumstruktur zu speichern, müssen Codes für '(' und ')' hinzugefügt werden, aber die Verknüpfungen `parent`, `child`, `next`, `previous` können vernachlässigt werden.

Es besteht auch keine Notwendigkeit, die Elementnummern zu speichern. Beim Laden einer komprimierten Baumdatei werden neue Elementnummern zugewiesen, beginnend bei 1 für das Seitenelement.

DAS XTREE-Projekt

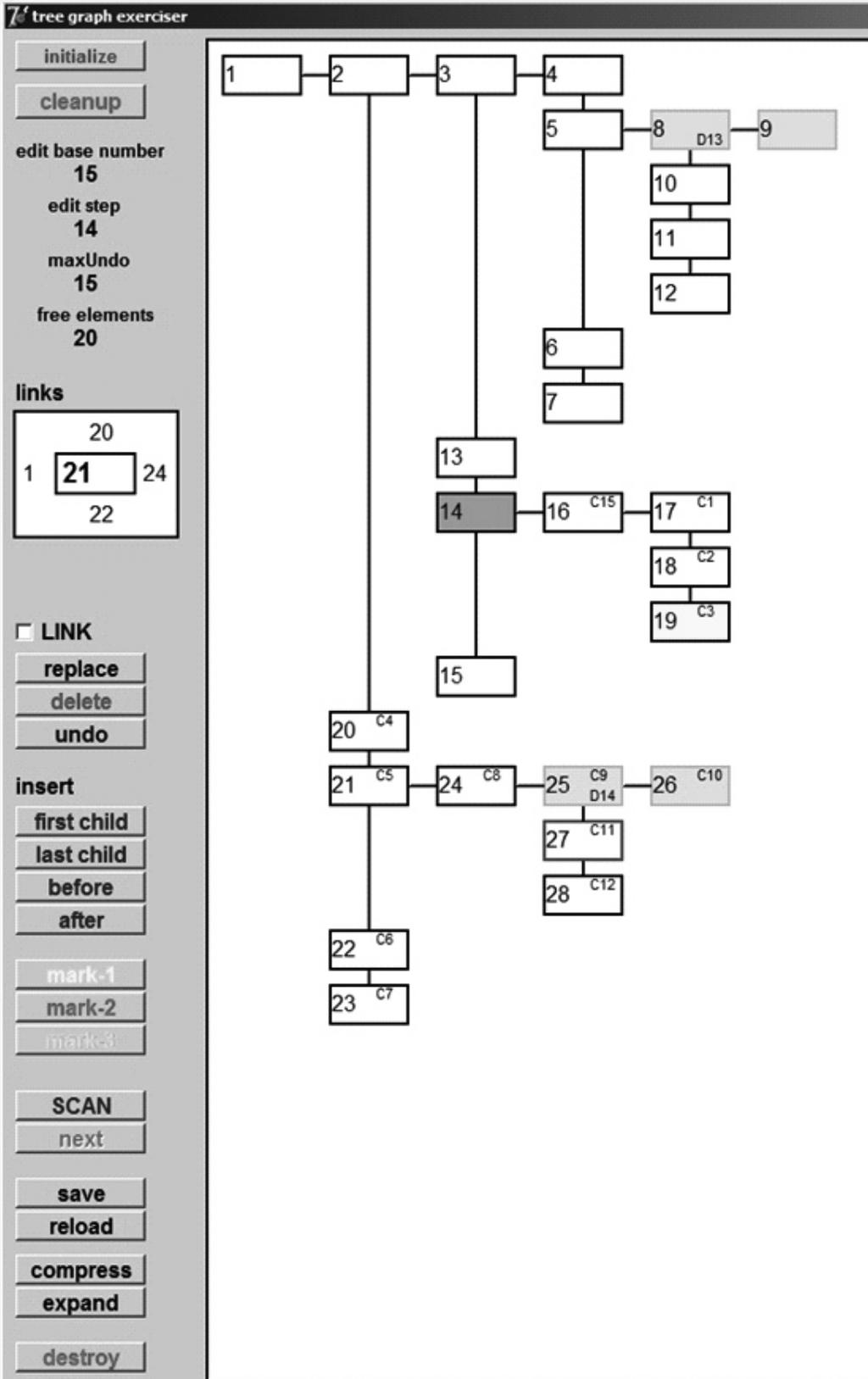
Dieses Projekt wurde erstellt, um die neuen Bearbeitungs- und Rückgängigmachungsverfahren zu testen, bevor sie in den mathematischen Editor implementiert werden.

Es gibt zwei Einheiten und ein Formular:

`Form1/Einheit1`: buttons, paintboxes, control, events, tree painting (BaumMalerei)

`xtree2_Unit`: Bearbeitungsprozeduren.





BUTTONS

Initialize	Zerstört den gesamten Baum. Setzt den Status von Element 1 auf esActive, dies ist die Wurzel.
cleanup	Zerstört gelöschte Elemente. Setzt Cstep=0 für alle Elemente. Alle Rückgängig-Informationen (Undo) werden gelöscht.
Link	Verknüpfen Sie die nächsten Bearbeitungsaktionen als eine Aktion, indem Sie denselben edStep-Wert verwenden.
Replace	ausgewähltes Element wird ersetzt
Delete	Ausgewähltes Element und seine Kinder deaktivieren.
Undo	zuletzt hinzugefügtes Element zerstören und / oder gelöschte Elemente wieder aktivieren.
Insert	neues Element einfügen hinzugefügt (nach/ vor dem ausgewählten Element oder als 1./letztes Kind)
mark(1,2,3)	Hintergrundfarbe auf/von ausgewähltem Element anwenden/entfernen
scan	Klicken Sie next um ein neues Element auszuwählen, Kinder zuerst.
Save/reload	Baum speichern/laden, einschließlich Rückgängig-Informationen
Compress/Expand	Komprimieren/Expandieren Sie den Baum ohne Rückgängig-Informationen, neue Elementnummern werden beim erneuten Laden zugewiesen.
Destroy	Löschen Entfernen Sie das ausgewählte Element aus dem Baum.

Beschreibung der **XTree2_unit** Einzelheiten entnehmen Sie bitte dem Quellcode.

PROZEDUREN ZUM NAVIGIEREN

```
function hasParent(el: word): boolean;
function hasChild(el: word): boolean;
function hasNext(el: word): boolean;
function hasPrevious(el: word): boolean;
```

Gibt true zurück, wenn das Element **el** ein Elternteil (**child, next, previous**) hat, das nicht gelöscht wird.

```
function getParent(var par: word; el: word): boolean;
function getChild(var chld: word; el: word): boolean;
function getNext(var nxt: word; el: word): boolean;
function getPrevious(var prev: word; el: word): boolean;
```

wenn das Element **el** ein übergeordnetes Element hat (**child, next, previous**) gibt **true** und die Elementnummer zurück.

```
function ScanElement(var elmnt: word; lt: TLink): TLink;
```

wenn das Element **el** ein übergeordnetes Element hat (**child, next, previous**) gibt **true** und die Elementnummer zurück.

```
function ScanElement(var elmnt: word; lt: TLink): TLink;
```

elmnt ist das übergebene Element, lt ist der Link (von **parent, child, next, previous**) zu **elmnt**. Ein nächstes Element im Baum wird in **elmnt** zurückgegeben, der neue Link ist das Ergebnis der Funktion.

```
function fullScan(var elmnt: word; lt: TLink): TLink;
```

Entspricht **ScanElement**, überspringt aber keine gelöschten Elemente. Diese Funktion wird verwendet, um vorherige und übergeordnete Links nach dem Neuladen hinzuzufügen und um den Baum zu durchsuchen, um Elemente außerhalb des Rückgängigmachungsbereichs zu finden.

BEARBEITUNG DER PROCEDUREN

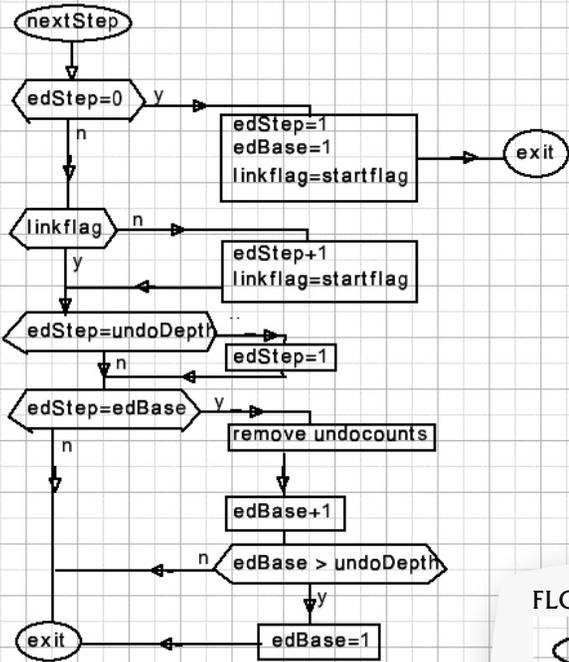
```
procedure PlaceA(del, sel: word);      place element sel after del
procedure PlaceB(del, sel: word);      place element sel before del
procedure PlaceChildA(del, sel: word); place element sel as last child of del
procedure PlaceChildB(del, sel: word); place element sel as 1st child of del
procedure replaceEL(del, sel: word);    replace element del by sel.
```

VERSTECKTE PROCEDUREN:

```
procedure Extract(el: word);           remove element el from tree, repair links
                                         note: element[1] (the root) cannot be extracted

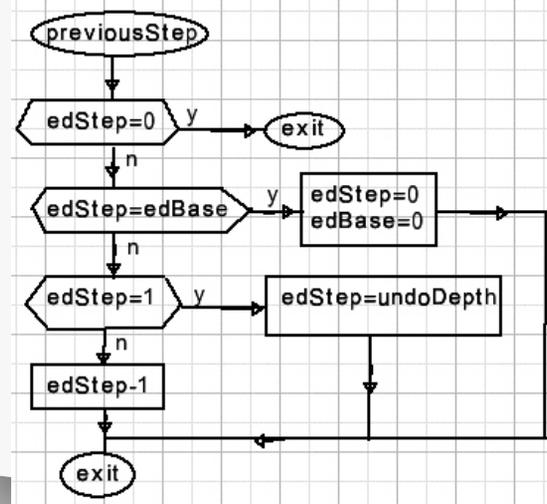
procedure LinkIn(el: word);           element[el] has correct links, set links to el from parent child next previous elements
procedure NextStep;                   increment the undoBase/undoStep counter before an insert or delete operation
                                         Search tree for elements having Cstep=Edstep or Dstep=edStep.
```





procedure previousStep;
dekrementieren Sie den undoBase/undoStep-Zähler nach der Rückgängigmachung.

FLOWCHART:



procedure removeUndoCounts;

Testen sie alle elemente:

If Cstep=edStep then set Cstep = 0,
If Dstep = edStep then destroy element

procedure setParentLinks(par : word);

platziert das Element außerhalb des Undo-Bereichs.
da sie außerhalb des Undo-Bereichs liegt.
Verknüpfung der Kinder zum Element par setzen.

LINKING

Procedure StartTreeLinking

Setzt das **startFlag**.

Beim nächsten Inkrement von edStep wird das **linkFlag** gesetzt,
was weitere **EdStep-Aktualisierungen** verhindert.

procedure stopTreeLinking

Löscht sowohl das startFlag als auch das linkFlag.

function TestELdeleted(el : word) : boolean;

Gibt true zurück, wenn ein Element oder ein Elternteil gelöscht wurde. Diese Funktion wird von der Treepainting-Prozedur aufgerufen..



ELEMENT ZERSTÖREN BENACHRICHTIGUNG

Elemente wie Frames, Tabellen, Matrizen, Koordinatensysteme und Bilder benötigen mehr Eigenschaften und das Element verweist auf eine Eigenschaftsliste. Wenn ein solches Element zerstört wird, müssen in einigen Fällen auch die Einträge in diesen Listen entfernt werden. Aus diesem Grund ruft die `xtree2_unit` vor jeder Zerstörungsoperation

procedure `destroyNotification(del : word);` um den Editor zu informieren, dass das Element `del` zerstört wird.

Unit1 description

Der Baum wird auf die Leinwand der Bitmap-Map gemalt, die später in `paintbox1` kopiert wird. Die Größe beträgt 1000*980 Pixel (H*V), aufgeteilt in 10 Spalten und 24 Zeilen mit 80*40 Pixeln. Auf der rechten Seite bleiben 2 Spalten übrig, in denen nicht verwendete Elemente in der Farbe Grau dargestellt werden. Die Eigenschaften `posX`, `posY` eines Elements geben die Spalte und die Zeile an.

procedure `getActiveXY(var x,y : smallInt; px,py : byte);` übersetzt die `x, y` Spalte/Zeile in die Pixelpositionen `px,py`.

Für Elemente im Leerlauf wird diese Berechnung durch **procedure** `getIdleXY(var x,y : smallInt; elnr : byte)` durchgeführt. Ein Element wird mit der **procedure** `paintElement(el : word)` gezeichnet, unter Verwendung der von **procedure** `paintmap` Eigenschaften `posX`, `posY`; die den gesamten Baum neu malt.

Mit der **procedure** `fullscan(. .)` wird der Baum durchlaufen und im Falle eines Child- oder Next-Links wird das **PFlag** gesetzt, was zur Berechnung von `posX`, `posY` und einem Aufruf von `paintElement` führt. Die höchste **Y-Position** wird in der Variablen `topY` gespeichert, um zu verhindern, dass sich Elemente überlappen.

procedure `repaintElement(elmnt : word);`

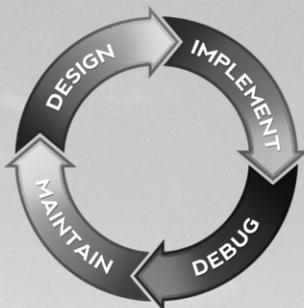
malt ein Element in der Karte und überträgt das geänderte Rechteck an `paintbox1` auf `Form1`. Dies wird verwendet, wenn ein neues Element ausgewählt oder die Markierungseigenschaft geändert wird. Damit ist die Beschreibung abgeschlossen.



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



FastReport VCL - reporting library for Delphi

Report generator FastReport VCL is a modern solution for integrating Business Intelligence in your software.

It's created for developers who want to use ready-made components for reporting.

FastReport VCL, with its simplicity of use, convenience, and small distribution size can provide high functionality and performance on almost any modern PC.

Features

- Reports outside of templates
- Flexibility and interactivity
- Easy integration with any
- Data Security
- Storage of documents
- Internationality

Exactly what heavy corporate reporting needs. FastReport is specifically optimized for speed and every day proves itself on heavy workloads in real businesses! If your clients want to get reports quickly - they just need FastReport!



starter

experte

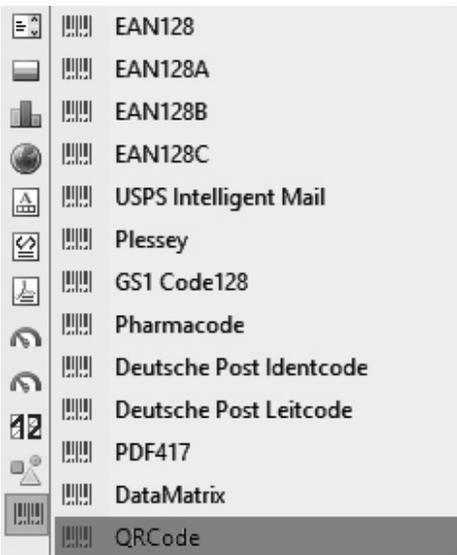
D11

In den letzten Jahren sind QR-Codes zu einem alltäglichen Bestandteil unseres Lebens geworden. Es handelt sich dabei um zweidimensionale Barcodes, die von einem digitalen Gerät leicht gelesen werden können und Informationen in Form einer Reihe von Pixeln speichern. Sie werden inzwischen häufig im Handel, in der Logistik und in der Produktion eingesetzt.

Im Gegensatz zu einem einfachen Strichcode wird ein QR-Code horizontal und vertikal gelesen. Daher können sie mehr Daten speichern. Die leichte Erkennbarkeit und Benutzerfreundlichkeit von QR-Codes hat ihre Popularität vorbestimmt.

Mit FastReport VCL können Sie QR-Codes ganz einfach in Ihren Berichten verwenden. Schauen wir uns diese Möglichkeit einmal genauer an.

Starten Sie den Berichtsgenerator-Designer. Wählen Sie die Komponente "QRCode" aus dem Popup-Menü des Werkzeugs "Barcode-Objekt".



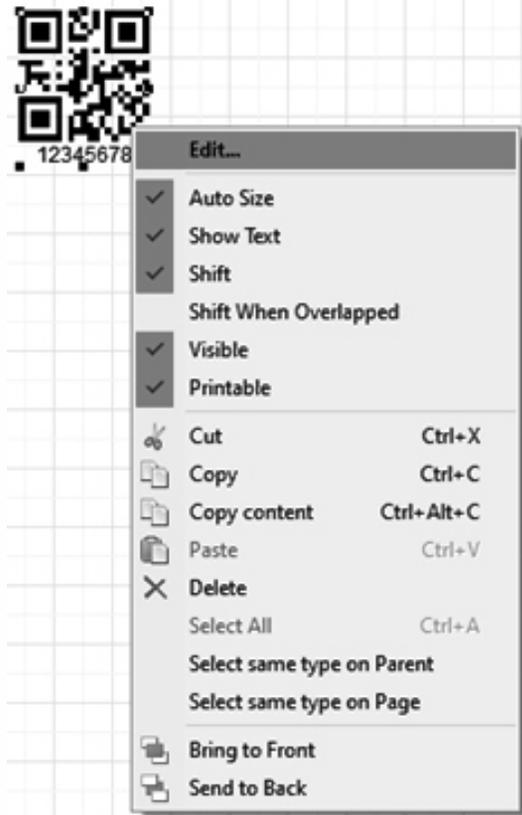
Wenn Sie einer Seite hinzugefügt werden, sieht der QRCode wie folgt aus:



Wählen Sie die Komponente "QRCode" aus



Doppelklicken Sie darauf, um den 2D-Barcode-Editor zu öffnen.
Sie können dies auch über das Kontextmenü tun,
indem Sie den Punkt "Edit" auswählen:



Barcode editing

So sieht der Barcode-Editor aus:



2D barcode editor

Schauen wir uns seine Hauptfunktionen im Detail an:

- ❶ Auf der Registerkarte "General" können Sie die Größe des Barcodes ändern, eine Signatur hinzufügen oder ihn drehen.
- ❷ Auf der Registerkarte "Other options":
 - Wenn CodePage <> 0 - wird der ECI-Modus verwendet;
 - Sie können die Textkodierung in der Eigenschaft Kodierung auswählen;
 - Die Eigenschaft ErrorLevels gewährleistet Redundanz für das korrekte Lesen von Daten mit einem teilweise beschädigten Codebild;
 - QuietZone erkennt, ob der QR-Code einen weißen Rahmen hat.
- ❸ Im Ausdruckeditor, in der Zeile "Code", können Sie:
 - Zugriff auf Felder der Datenquelle;
 - Verwenden Sie Systemvariablen;
 - Verschiedene Funktionen verwenden.

In der Objekteigenschaft "ExpressionPreset" können Sie die Voreinstellungen für die Generierung von Quittungen gemäß den Spezifikationen des SBER-Codes und des Schweizer Codes:



Auswahl der Voreinstellungen für den QR-Code

Sie können Ihre eigenen QR-Codes verschiedener Typen generieren. Dazu müssen Sie eine Zeichenkette eines bestimmten Formats angeben und diese in der Eigenschaft Text festlegen. Schauen wir uns diese Typen anhand von Beispielen für Barcodes und Daten näher an: URI ist ein Uniform Resource Identifier. Es handelt sich um eine Zeichenkette zur Identifizierung verschiedener Dateien, Dokumente, Bilder, E-Mails, Webdienste usw.:



<https://www.fast-report.com/en/>



<https://www.fastreport.ru/ru/>



- EmailAddress – E-mail address:



support@fast-report.com

- SMS –text message:



SMSTO:(71) 555-4444:Hello, Dolly! I'm fine!

- EmailMessage –email text:



MATMSG:TO:support@fast-report.com;
SUB:FastReport VCL question;B
ODY:Hello, I have a question about FastReport VCL.;

- Geolocation
– coordinates for the
real geographic location:



geo:-50.737563,-79.490016,120

- Call – telephone number:



tel:(71) 555-4444



- Wi -fi - Informationen zum Herstellen einer Wi-Fi-Verbindung:



WIFI:T:WPA;S:HoneyPot;P:youarewelcome;H:true;

Wir haben uns angesehen, wie Sie QR-Codes in FastReport VCL verwenden können. Diese Komponente bietet mehr Möglichkeiten für die Verwendung dieses Reportgenerators in modernen Arbeitsabläufen.

Lassen Sie uns a QR-Code aus Delphi/Lazarus-Code erstellen:

```

uses frxBarcode2D;
procedure TForm1.Button1Click(Sender: TObject);
var
    bcQR: TfrxBarcode2DView;
begin
    bcQR := TfrxBarcode2DView(frxReport1.FindObject('Barcode2D1'));
    { Set the barcode type }
    bcQR.BarType := bcCodeQR;

    { Depending on the type you want to use, you need to
      leave one line that assigns the text of a certain format }

    { If you want to set url type: }
    bcQR.Text := 'https://www.fast-report.com/en/';

    { If you want to set EmailAddress type: }
    bcQR.Text := 'support@fast-report.com';

    { If you want to set EmailMessage type: }
    bcQR.Text :=
        'MATMSG:TO:support@fast-report.com;SUB:FastReport.Net question;BODY:Hello, I have a question about FastReport VCL.';

    { If you want to set Geolocation type: }
    bcQR.Text := 'geo:-50.737563,-79.490016,120';

    { If you want to set SMS type: }
    bcQR.Text := 'SMSTO:(71) 555-4444:Hello, Dolly! I" m fine ! ';

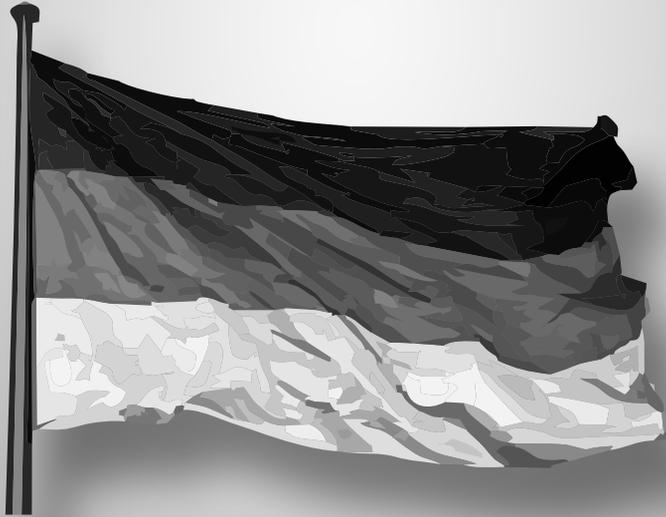
    { If you want to set Call type: }
    bcQR.Text := 'tel:(71) 555-4444';

    { If you want to set Wi-Fi type: }
    bcQR.Text := 'WIFI:T:WPA;S:HoneyPot;P:youarewelcome;H:true;';
    frxReport1.ShowReport();
end;

```



Wir suchen Hilfe bei der Korrektur von bereits
ins Deutsche übersetzte Artikel,
um auf idiomatische Ungenauigkeiten
und (Formatierungs-)Fehler
in den Artikel hinzuweisen.



Wenn Sie uns helfen möchten,
senden Sie bitte Ihre Reaktion an
admin@blaisepascalmagazine.eu

Von Michaël Van Canneyt



ABSTRAKT

Webanwendungen und Websites sind oft für die ganze Welt sichtbar, und dem Benutzer zu ermöglichen, Ihre Anwendung oder Website in seiner eigenen Sprache zu nutzen, ist eine deutliche Verbesserung der Benutzeroberfläche. Hier zeigen wir Ihnen, wie Sie Ihre **PAS2JS** Anwendung übersetzen können.

1 EINFÜHRUNG

Wenn Sie eine öffentliche Website oder eine öffentliche Webanwendung haben, ist diese für die ganze Welt sichtbar. Nutzer jeder Muttersprache können darauf zugreifen, und natürlich ist es für sie angenehmer, wenn sie Ihre Website in ihrer gewohnten Sprache nutzen können, anstatt in der **Lingua franca** des Internets: **Englisch**.

Bei der Übersetzung einer Webanwendung gibt es zwei Aspekte zu beachten, wenn Sie möchten, dass der Benutzer Ihre Website nutzen kann:

- 1 Die vom Code generierten Nachrichten.
 - 2 Die Texte, die in den **HTML-Code** eingefügt werden.
- Beide Arten der Übersetzung erfordern einen unterschiedlichen Ansatz.

2 NACHRICHTEN IN CODE ÜBERSETZEN

Delphi und **Free Pascal** unterstützen seit langem das Konzept der Resourcestrings: Dies sind benannte String-Konstanten im Code, die fast wie echte Konstanten funktionieren. Der Unterschied im Code besteht darin, dass sie sich in einem 'resourcestring'-Abschnitt im Code befinden und nicht in einem 'const'-Abschnitt.

Der Unterschied zur Laufzeit besteht darin, dass der Wert eines Resourcestrings: zur Laufzeit auf der Grundlage seines Namens abgerufen wird.

Der Mechanismus zum Abrufen des Wertes der Zeichenkette zur Laufzeit ist in **FPC** austauschbar. Sie können die Strings aus einer **DLL** (*der Standardmechanismus in Delphi*) oder einer `.po`-Datei (unter Unix üblich) beziehen oder einen eigenen Mechanismus entwickeln. Indem Sie die Quelle der Resourcestrings ändern, können Sie eine übersetzte Version der Strings bereitstellen.

In **PAS2JS** wird das Konzept der Resourcestrings beibehalten. Das generierte **Javascript** enthält die Standardversion der Resourcestrings, wie sie im Quellcode definiert sind.

Die Werte der Strings werden jedoch strukturiert aufbewahrt und mit einer speziellen Hilfsfunktion auf der Grundlage des Namens der Resourcestrings-Konstante abgerufen.

Es wird eine Methode bereitgestellt, um die Werte in den **Javascript-Quellen** zur Laufzeit zu setzen: Der **PAS2JS RTL** wird mit einer Einheit geliefert, mit der Sie die Werte der Resource Strings im Programm festlegen können. Genau wie der **FPC-Compiler** kann auch der **PAS2JS-COMPILER** eine **JSON-Datei** mit den Standard-Ressourcenstring-Definitionen erzeugen.

Es gibt eine Befehlszeilenoption, die die Erzeugung dieser Datei steuert:

- `-Jrnone` Es wird keine Datei erzeugt. Dies ist die Standardeinstellung.
- `-Jrunit` Der Compiler erzeugt eine JSON-Datei pro verwendeter Unit
- `-Jrprogram` Der Compiler erzeugt eine einzelne JSON-Datei mit allen im Programm verwendeten Ressourcen.

Die **JSON-Datei** hat die Erweiterung `json`.



Die mit `-Jrprogram` erzeugte Datei sieht wie folgt aus:

```
{
  "mystrings" : {
    "Header"      : „Übersetzung unter Verwendung von Resourcestrings“,
    "Paragraph"   : „Dieser Text wird übersetzt.“,
    "TranslateDirect" : „Für dieses Beispiel wird die direkte API verwendet.“,
    "TranslateJSON" : „Für dieses Beispiel wird die direkte API verwendet.“,
    "TranslateURL" : „Für dieses Beispiel wird eine URL verwendet.“,
    "Button"      : "Diese Seite übersetzen"
  },
  "SysUtils" : {
    "SAbortError"      : "Vorgang abgebrochen",
    "SApplicationException" : "Die Anwendung hat eine Ausnahme ausgelöst: ",
    "SErrUnknownExceptionType" : "Unbekannter Ausnahmetyp aufgetreten:"
  },
  "translate_basic" : {
    „BasicTitle" : "Übersetzung mit Resourcestrings - Basic API".
  }
}
```

Wie Sie sehen können, sind die Strings in einem **JSON-Objekt** pro Einheit gruppiert und es gibt einen speziellen Abschnitt, der für das Programm verwendet wird. Beim Kompilieren für eine einzelne Einheit würde die folgende Datei für die `mystrings` unit erzeugt werden:

```
{
  "mystrings" : {
    "Überschrift" : "Übersetzung unter Verwendung von Resourcestrings",
    "Absatz"      : "Dieser Text wird übersetzt.",
    "TranslateDirect" : "Für dieses Beispiel wird die direkte API verwendet.",
    "TranslateJSON" : "Für dieses Beispiel wird ein JSON-Objekt verwendet.",
    "TranslateURL" : "Für dieses Beispiel wird eine URL verwendet.",
    "Button"      : „Diese Seite übersetzen" "
  }
}
```

Diese Datei kann verwendet werden, um die Strings auf einfache Weise mit einem beliebigen externen Tool zu übersetzen. Das **JSON-Format** eignet sich sehr gut für die Bearbeitung mit verschiedenen Tools und das Zusammenführen ist einfach. Um die Strings tatsächlich zu übersetzen, enthält die **PAS2JS RTL** die Unit `rstranslate`, die zur Übersetzung der Strings zur Laufzeit verwendet werden kann. Die Unit enthält die folgenden Definitionen:

Type

```
{ TResourceTranslator }
TLoadFailEvent = Reference to Procedure (Sender : TObject; aCode : Integer; aError : String);
TONTranslatedEvent = Reference to Procedure (Sender : TObject; aURL : String);

TResourceTranslator = Class
  Class Function Instance : TResourceTranslator;
  Procedure Translate(Const aUnit,aString,aTranslation : String);
  Procedure Translate(Const aTranslations : TJSObject);
  procedure Translate(const aURL : string; aOnTranslated : TONTranslatedEvent = Nil);
  Procedure ResetTranslation(Const aUnit : String; const aString : String = "");
  Property OnLoadFail : TLoadFailEvent;
  Property OnURLTranslated : TONTranslatedEvent;
end;

Function ResourceTranslator : TResourceTranslator;

Procedure Translate(Const aURL : String; aOnTranslated : TONTranslatedEvent = Nil);
Procedure Translate(Const aUnit,aString,aTranslation : String);
Procedure Translate(Const aTranslations : TJSObject);

Procedure ResetTranslation(Const aUnit : String; Const aString : String = "");
```



Die Verwendung dieses Geräts ist recht einfach:

`TResourceTranslator` ist die Klasse, die die eigentliche Arbeit des Übersetzens der vom Compiler erzeugten Resource String-Strukturen übernimmt. Die Unit erstellt eine globale Instanz (*verfügbar in `TResourceTranslator.Instance`, mit der Abkürzungsfunktion `ResourceTranslator`*), so dass Sie keine eigene Instanz erstellen müssen.

Die Translate-Funktionen führen die eigentliche Übersetzung durch.

Die erste Form setzt den Wert einer einzelnen Ressourcenzeichenkette:

```
Procedure Translate(Const aUnit,aString,aTranslation : String); overload;
```

Die Parameter `aUnit` und `aString` identifizieren die Ressourcenzeichenkette, und `aTranslation` ist der Wert, der für die Ressourcenzeichenfolge festgelegt werden soll.

Das zweite Formular akzeptiert ein **JSON-Objekt** in der gleichen Form, wie es vom Compiler erzeugt wird:

```
Procedure Translate(Const aTranslations : TJSONObject); overload;
```

Es durchläuft alle Strings im **JSON-Objekt** und setzt die Übersetzung für jeden String. Das letzte Formular lädt eine **JSON-Datei** von einer URL, konvertiert die Datei in ein **JSON-Objekt** und ruft die vorherige Funktion auf.

```
procedure Translate(const aURL: string; aOnTranslated: TOnTranslatedEvent = Nil); overload;
```

Da das Laden einer Datei von einer **URL** asynchron abläuft, gibt es einen Callback:

Wenn dies geschehen ist, wird die `OnTranslated` aufgerufen. Das Objektereignis `OnURLTranslated` wird ebenfalls aufgerufen.

Wenn das Laden der **JSON-Datei** fehlschlägt, wird das Ereignis `OnLoadFail` aufgerufen.

Der von **PAS2JS** verwendete Übersetzungsmechanismus löscht nicht die ursprünglichen Werte der Ressourcenstrings. Sie können auf ihre ursprünglichen Werte zurückgesetzt werden, wenn Sie dies wünschen. Dies kann mit dem Aufruf `ResetTranslation` geschehen. Dieser Aufruf kann verwendet werden, um die Übersetzung für eine einzelne Einheit oder einen einzelnen Resource-String zurückzusetzen.

Die Übersetzung der Ressourcenketten kann natürlich jederzeit vorgenommen werden: Ab diesem Zeitpunkt wird bei der Verwendung einer Ressourcenkette der übersetzte Wert verwendet. Dadurch eignet sich der Mechanismus für den Fall, dass ein Dropdown-Menü angezeigt wird, aus dem der Benutzer seine bevorzugte Sprache auswählen kann.

Zur Veranschaulichung sehen wir uns die folgende kleine Webseite an, die wir in einer Datei namens `index.html` speichern werden:

```
<!doctype html>
<html lang="en">
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="bulma.min.css">
  <title id="translate-title"></title>
  <script src="translate.js"></script>
</head>

<body>
  <div class="section">
    <div class="box">
      <h2 class="title is-2" id="translate-header"></h2>
      <p class="block" id="translate-text"></p>
      <button id="btn-translate" class="button is-link"></button>
    </div>
  </div>
  <script>
    window.onload=rtl.run;
  </script>
</body>
</html>
```



Sie sehen, dass diese **HTML-Seite** keine Texte enthält. Die Texte werden von dem Programm in die Tags eingegeben.

Hier ist ein Teil des Programms. Es enthält eine Unit `mystrings`, die die meisten Resourcestrings enthält. Die Ressourcenkette `PageTitle` ist im Programm enthalten, um die verschiedenen Abschnitte zu zeigen, in denen sich die Ressourcenketten befinden.

Das Programm verfügt über einige Variablen, die den verschiedenen **HTML-Tags** mit einem ID-Attribut entsprechen. Die Variablen werden in der `Init-Routine` initialisiert:

```

program translation;
{$mode objfpc}
uses
    JS, Classes, SysUtils, Web, mystrings, rstranslate;

ResourceString
    PageTitle = Übersetzung mit Resourcestrings - Handbuch ;
Var
    aHeader : TJSHTMLElement;
    aPar : TJSHTMLElement;
    aButton : TJSHTMLButtonElement;
    IsDutch : Boolean;

procedure SetTexts;
begin
    aHeader.InnerHTML:=Header;
    aButton.InnerHTML:=Button;
    aPar.InnerHtml:=Paragraph+' '+TranslateDirect;
    Document.title:=PageTitle;
end;

Procedure Init;

    function GetEl(const aName: string): TJSHTMLElement;
    begin
        Result:=TJSHTMLElement(Document.getElementById(aName));
    end;

begin
    aHeader:=GetEl('translate-header');
    aPar:=GetEl('translate-text');
    aButton:=TJSHTMLButtonElement(GetEl('btn-translate'));
    aButton.onclick:=@DoTranslation;
end;

begin
    Init;
    SetTexts;
end.

```

Die eigentlichen Texte der HTML-Seite werden in der `SetTexts`-Routine festgelegt:

Wie Sie sehen können, befinden sich die Texte in konstanten (Ressourcen-)Strings, die in der Einheit `MyStrings` gespeichert sind.

Die resultierende Seite sieht wie in *Abbildung 1 auf Artikelseite 5* aus.

Der onclick-Handler des Buttons `aButton` (`DoTranslation`) wurde aus Gründen der Übersichtlichkeit im Code weggelassen. Wenn Sie auf die Schaltfläche klicken, wird die Seite vom Standard (Englisch) ins Deutsche und umgekehrt übersetzt.



Der Handler kann wie folgt implementiert werden:

```
function DoTranslation(aEvent: TJSMouseEvent): boolean;
begin
  IsDutch:=Not IsDutch;
  if IsDutch then
    begin
      Translate('Programm','PageTitle','Übersetzen mit Ressourcenstrings - direkte API');
      Translate('mystrings','Button','Diese Seite übersetzen');
      Translate('mystrings','Header','Übersetzung mit Ressourcenstrings');
      Translate('mystrings','Paragraph','Dieser Text wird übersetzt.');
```

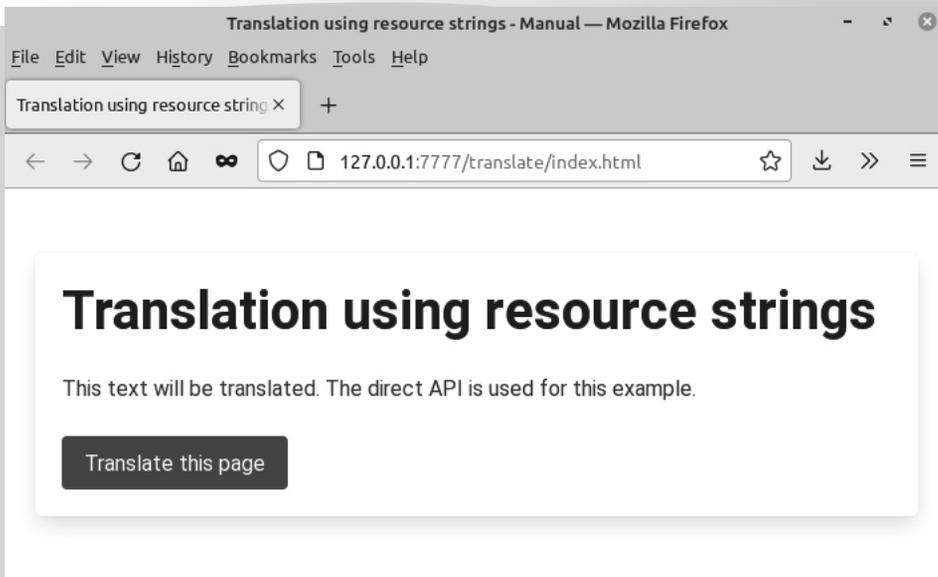


Abbildung 1: Die Seite in ihrer Standardsprache

Wie Sie sehen können, werden alle Strings manuell ins **Deutsche** übersetzt, indem der **Deutsche** Wert für den resourcestring direkt mit dem Aufruf `translate` gesetzt wird. Um wieder ins **Englische** zurückzukehren, wird die Routine `ResetTranslation` aufgerufen. Am Ende wird die `SetText`-Routine aufgerufen, die die Ressourcenketten einfach wieder anwendet. Die Seite sieht dann wie in *Abbildung 2 auf Artikelseite 6* aus.

Natürlich ist die obige Methode nicht empfehlenswert, um Ihr Programm zu übersetzen: Wenn Sie viele Sprachen unterstützen müssen und viele Strings zu übersetzen haben, würde die Menge an Code, die Sie dafür benötigen, sehr groß werden. Stattdessen ist es besser, die Übersetzungen in einem strukturierten Format (**JSON**) zu laden und eine kleine Routine zu haben, die die Strings übersetzt.

Die Routinen Unit `rstranslate` enthält eine Routine, die ein **JSON-Objekt** mit demselben Format wie die vom Compiler erstellte Datei akzeptiert und dieses zur Übersetzung aller Strings verwendet. Eine Möglichkeit, dies zu tun, besteht darin, die Übersetzung in die **HTML-Seite** einzubinden.



Dazu erstellen wir eine kleine **Javascript-Datei**, die das **JSON-Objekt** mit den Übersetzungen definiert:

```
var german = {
  "program" : {
    „URLTitle“:„Vertaling met resourcestrings - URL API“
  },
  "mystrings":{
    "Button": "Diese Seite übersetzen",
    "Header": "Übersetzung mit Resourcestrings",
    "Absatz": "Dieser Text wird gerade übersetzt.",
    "TranslateDirect": "Für dieses Beispiel wird die direkte API verwendet",
    "TranslateJSON": "Für dieses Beispiel wird ein JSON-Objekt verwendet."
    "TranslateURL": "Für dieses Beispiel wird eine URL verwendet."
  }
};
```



Abbildung 2: Die Seite auf Deutsch

Wir fügen dies in die HTML-Datei ein, indem wir die folgende Zeile hinzufügen:

```
<script src="Deutsch.js"></script>
```

und um auf die **Deutsch-** Variable in unserem Programm zugreifen zu können, definieren wir eine externe Variable in unserem **Pascal-Programm** wie folgt:

```
Var
  Deutsch: TJSObject; external name 'Deutsch';
```

Der Wert der deutschen Variablen ist dann das **JSON-Objekt** mit den Übersetzungen. Die **DoTranslate-Routine** wird nun einfach wie folgt:

```
function DoTranslation(aEvent: TJSMouseEvent): boolean;
begin
  Result:=True;
  IsDeutsch :=Not IsDeutsch;
  if IsDeutsch then
    Translate(Deutsch)
  else
    begin
      ResetTranslation('program','PageTitle');
      ResetTranslation('mystrings');
      SetTexts;
    end;
end;
```



Indem Sie Variablen für verschiedene Sprachen definieren, können Sie zwischen verschiedenen Sprachen umschalten. Natürlich werden beim Start des Programms alle Sprachen geladen.

Dies kann vermieden werden, indem das Skript-Tag dynamisch in den **HTML**-Code eingefügt wird, sobald die Sprache gewechselt werden muss: Der Browser lädt dann die neuen Übersetzungen. Wenn Sie allen Übersetzungen denselben Namen für die Variable geben, die das **JSON-Objekt** enthält, wird nur die zuletzt injizierte Sprache im Speicher des Browsers gehalten.

Ein Nachteil dieser Methode ist immer noch, dass Sie eine zusätzliche **Javascript**- Quelldatei benötigen. Dies erfordert die Umwandlung einer **JSON-Datei**, wie sie vom Compiler erzeugt wird, in eine **Javascript-Quelldatei**. Es wäre einfacher, wenn die **JSON-Datei** direkt verwendet werden könnte. Glücklicherweise ist dies möglich.

Der Aufruf `Translate` verfügt über eine überladene Version, der eine URL übergeben werden kann: Diese **URL** sollte auf eine **JSON-Datei** mit den Übersetzungen aller Strings verweisen, und zwar in dem Format, das der Compiler bereitstellt. Der Aufruf lädt diese Datei herunter und verwendet die Übersetzungen, um die in der **JSON-Datei** gefundenen Strings zu übersetzen. Es gibt einen `Callback`, der verwendet werden kann, um benachrichtigt zu werden, wenn die Übersetzung abgeschlossen ist (*das Herunterladen einer Datei erfolgt asynchron*), dieser `Callback` kann zur Aktualisierung der Anzeige verwendet werden.

Die dritte Version unseres Programms zeigt, wie dies genutzt werden kann. Das Anpassen unseres Programms besteht aus 3 Schritten:

- ❶ Entfernen Sie das Skript-Tag mit dem Verweis auf die Datei `deutsch.js`.
- ❷ Entfernen Sie die Definition der externen Variablen `Deutsch`.
- ❸ Ändern Sie die `DoTranslation`-Routine so, dass sie die **URL** verwendet. Die `DoTranslation`-Routine sieht nun wie folgt aus:

```
function DoTranslation(aEvent: TJSMouseEvent): boolean;

    Procedure DoTranslated(Sender: TObject; aURL: String);
    begin
        SetTexts;
    end;

begin
    Result:=True;
    IsDeutsch:=Not IsDeutsch;
    if IsDeutsch then
        Translate('deutsch.json',@DoTranslated)
    else
        begin
            ResetTranslation('program','PageTitle');
            ResetTranslation('mystrings');
            SetTexts;
        end;
end;
```

Beachten Sie, dass `SetTexts` beim Laden der **JSON-Datei** vom `OnTranslated`-Callback aus aufgerufen werden muss (*denn nur dann werden die Texte tatsächlich übersetzt*), aber beim Zurücksetzen der Übersetzungen direkt aufgerufen werden kann.



3 HTML-INHALT ÜBERSETZEN

In der Praxis werden Sie Ihre **HTML-Seite** wahrscheinlich nicht mit Texten aus Ressourcenstrings füllen. Ein Nachteil ist, dass jede Stelle, die übersetzt werden muss, ein **ID-Attribut** benötigt. Der zweite Nachteil ist, dass das **Javascript** geladen werden muss, bevor der Text auf Ihrer **HTML-Seite** erscheint. Wenn die Formatierung angewendet wird, wird der Text in verschiedene Teile zerschnitten, was unnatürlich ist.

Stattdessen wird Ihr **HTML-Code** höchstwahrscheinlich die Texte in der Standardsprache enthalten, wenn die Seite geladen wird.

Wie können wir also die Texte im **HTML** übersetzen? Es gibt viele Möglichkeiten.

Wir könnten Ressource-Strings verwenden und durch die Zuweisung einer **ID** an alle **HTML-Tags** alle Texte übersetzen. Dies hat viele der Nachteile der vorherigen Methode.

Wir können für jede Sprache eine **HTML-Seite** erstellen; dies ist ein langwieriger und mühsamer Prozess.

Hier stellen wir einen alternativen Ansatz vor, aus der in der `Rtl.HTMLTranslate Unit`.

Das erste Problem ist, dass wir wissen müssen, welche Texte wir übersetzen müssen.

Wir können einfach den **HTML-Text** scannen und alle Textfragmente extrahieren:

Das ist ganz einfach. Dann müssen wir die Übersetzung finden.

Mit einem **Hash-Mechanismus** können wir schnell nach Übersetzungen in einer Datei mit Übersetzungen suchen. Dies hat den Nachteil, dass der gesamte Kontext verloren geht und bei einzelnen Wörtern die Gefahr besteht, dass sie einfach falsch übersetzt werden.

Das englische Wort 'Save' kann zum Beispiel in mehrere Sprachen übersetzt werden ins Deutsche.

Welche Übersetzung zu verwenden ist, hängt vom Kontext ab und kann nicht allein anhand der Raute bestimmt werden. Ein besserer Ansatz ist es, allen zu verwendenden Texten eine eindeutige Kennung zu geben, so dass der Kontext des Wortes oder Satzes nachgeschlagen werden kann.

Aber vorzugsweise nicht die **ID**, um Konflikte zu vermeiden, wenn verschiedene **HTML-Fragmente** in einer einzigen Seite kombiniert werden.

Wir können dafür das Attribut `data` verwenden, das in der Javascript **DOM API** besonders behandelt wird.

In der Praxis bedeutet dies, dass alle Tags, die übersetzt werden müssen, wie folgt markiert werden müssen:

```
<h2 data-translate="Header">
  Translation using HTMLTranslator
</h2>
```

Das Attribut `data-translate` ist auf header Header gesetzt. Die Übersetzungsroutine wird sucht nach einem Text namens Header und ersetzt, wenn er gefunden wird, die Elemente `InnerText` oder `InnerHTML` mit dem übersetzten Text.

Manchmal muss man nicht den inneren Text übersetzen, sondern ein Attribut des **HTML Tags**.

Ein gutes Beispiel dafür ist das Attribut `placeholder` des Tags `Input` und `TextArea` Tags.

Die Einheit `Rtl.HTMLTranslate` verwendet hierfür eine Konvention:

Der Name des Attributs lautet Teil des Translate-Namens ist:

alles nach dem Bindestrich wird als `Attributname` betrachtet.

```
<textarea data-translate="memo-placeholder"
  placeholder="Type your remark here">
  Translation using HTMLTranslator
</textarea>
```



Wenn die Übersetzungsmaschine darauf stößt, sucht sie in den übersetzten Texten nach einer Übersetzung mit dem Namen 'memo-placeholder' und setzt das, was sie findet, in das Attribut 'placeholder'. Diese Konvention bedeutet natürlich, dass Sie keine Bindestriche in Ihren Übersetzungsnamen verwenden können.

Die TextArea kann möglicherweise zwei zu übersetzende Texte haben: das Platzhalter-Attribut und den eigentlichen Text im Textbereich. Je nachdem, wie HTML verwendet wird, müssen möglicherweise mehr Attribute übersetzt werden. Da wir nur ein einziges Datenattribut haben, wie können wir dem Rechnung tragen? Ganz einfach: der data-translate Mechanismus kann mehrere Namen verarbeiten, die durch ein Semikolon getrennt sind.

```
<textarea data-translate="memo;memo-placeholder"
placeholder="Type your remark here">
Translation using HTMLTranslator
</textarea>
```

Bei der Bearbeitung dieses Tags ersetzt die Übersetzungsmaschine den inneren Text durch den Inhalt des benannten Übersetzungsmemos und den Platzhalter durch den Inhalt von memo-placeholder.

Viele **Javascript-APIs** verwenden data- Tags, um ihr Rendering zu konfigurieren (z.B. verwenden *Grid-APIs dies häufig, um Spalten zu konfigurieren*), und der obige Mechanismus kann auch zur Übersetzung dieser Attribute verwendet werden. Die **API** der Einheit `Rtl.HTMLTranslate` ist recht einfach:

```
THTMLTagTranslator = class(TComponent)
Public
  procedure SetLanguageData(aData : TJSObject);
  Procedure TranslateHTMLTag(aEl : TJSHTMLElement; aScope : String = ""); overload;
  Procedure TranslateBelowHTMLTag(aEl : TJSHTMLElement; aScope : String = ""); overload;
  function GetMessageByName(const aScope, aName: string): String; overload;
  Function GetMessageByName(aRoot : TJSObject; const aScope, aName: string): String; overload;
  Function GetMessageByName(aScope : TJSObject; aName : string) : String; overload;
  Function GetMessageByName(aScope, aSeealsoScope : TJSObject; aName : string) : String; overload;
  Function HasLanguage(aLanguage : String) : Boolean;
  Function GetScope(const aScope : String) : TJSObject; overload;
  Function GetScope(aRoot : TJSObject; const aScope : String) : TJSObject; overload;
  Property CurrLanguageStrings : TJSObject Read FCurrLanguageStrings;

Published
  Property DataTagName : String Read GetDataTagName Write FDataTagName;
  Property DefaultScope : String Read FDefaultScopeName Write FDefaultScopeName;
  Property ContinueKey : String Read FContinueKey Write FContinueKey;
  Property Language : String Read FLanguage Write SetLanguage;
  Property LanguageSource : TLanguageSource Read FFileMode Write FFileMode;
  Property LanguageFileURL : String Read FLanguageFileURL Write FLanguageFileURL;
  Property LanguageVarName : String read FLanguageVarName write FLanguageVarName;
  Property TextMode : TTextMode Read FTextMode Write FTextMode;
  Property OnLanguageLoaded : TLanguageLoadedEvent;
  Property OnLanguageLoadError : TLanguageLoadErrorEvent;
end;
```



Die Klasse hat die folgenden öffentlichen/veröffentlichten **Eigenschaften**:

CurrLanguageStrings

die Daten für die aktuelle Sprache.

DataTagName

Name des Datenattributs. Sie können dies ändern, um ein anderes Datenattribut zu verwenden. Standard ist 'translate'.

DefaultScope

Standardbereich, der bei der Suche nach Übersetzungen verwendet wird. Zuerst wird der in TranslateHTMLTag übergebene Bereich geprüft, dann der Standardbereich, dann wird ContinueKey verwendet.

ContinueKey

ContinueKey-Name: Wenn diese Option gesetzt ist, gibt sie den Namen eines Bereichs an, in dem die Suche nach einem Übersetzungsbegriff fortgesetzt werden soll. Dies kann für die Vererbung von Formularen verwendet werden; ContinueKey kann gesetzt werden, um die Suche im geerbten Bereich fortzusetzen. Standardmäßig ist es leer.

language

Name der aktuellen Sprache (*wird kleingeschrieben*)

LanguageSource

Sprachquelle: Alle Sprachen in einer einzigen Datei (lsSingle), Sprachen in mehreren Dateien (lsMulti) oder eine Variable im globalen Javascript-Bereich (lsScoped) enthält alle Sprachen.

LanguageFileURL

Die URL der Datei, aus der die Sprachstrings geladen werden sollen. Bei LanguageSource=lsMulti sollte dies eine Formatvorlage mit dem Sprachcode sein (*verwenden Sie zum Beispiel: /lang-%s.json*).

LanguageVarName

wenn LanguageSource=lsScoped dann ist LanguageVarName der Name der globalen Window-Eigenschaft, die die Übersetzungen enthält.

TextMode

Textmode bestimmt, ob der Übersetzer InnerText (tmText) oder InnerHTML (tmHTML) verwendet, um den übersetzten Text festzulegen. **Beachten Sie**, dass die Verwendung von tmHTML zu einem Sicherheitsleck führen kann, da **Javascript** eingeschleust werden könnte.

OnLanguageLoaded

Ein Ereignis, das ausgelöst wird, wenn eine Sprache geladen wird (*unabhängig vom Mechanismus*).

OnLanguageLoadError

Ein Ereignis, das aufgerufen wird, wenn beim Laden einer Sprachdatei ein Fehler auftritt.

Die Klasse verfügt über die folgenden öffentlichen **Methoden**:

SetLanguageData

Setzen Sie das Sprachdatenobjekt direkt auf aData, anstatt eine **URL** zu verwenden. Die Daten werden gemäß der LanguageSource.

TranslateHTMLTag

Übersetzt einen einzelnen HTML-Tag aEl unter Verwendung des angegebenen Bereichs aScope. Wenn der Bereich nicht angegeben wird, wird der Stammbereich der aktuellen Sprache oder der Standardbereich verwendet.

TranslateBelowHTMLTag

Übersetzt das HTML-Tag aEl und alles darunter, unter Verwendung des angegebenen Bereichs aScope. Wenn der Bereich nicht angegeben wird, wird der Stammbereich der aktuellen Sprache verwendet.

GetMessageByName

Suche nach einer Nachricht mit dem Namen aName im angegebenen aScope, optional aSeeAlsoScope. Falls der Name des Bereichs angegeben ist, können Sie die Suche nach dem Bereich in aRoot starten. Die Nachricht ist leer, wenn keine Nachricht gefunden wurde.

HasLanguage

prüft, ob die angegebene Sprache verfügbar ist. Im Falle von lsMulti muss die aktuelle Sprache mit der übergebenen aLanguage übereinstimmen.

GetScope

findet das Scope-Objekt mit dem Namen aScope. Wenn aRoot angegeben ist, beginnt die Suche bei aRoot.



Beachten Sie, dass der Begriff "Scopes" verwendet wird. Dies ist für die Verwendung in SPA-Anwendungen gedacht. Jede "Seite" in der SPA kann bei der Suche nach Übersetzungen ihren eigenen Bereich verwenden:

Das bedeutet, dass es nicht notwendig ist, global eindeutige Namen für die Texte zu erfinden:

Die Namen müssen nur innerhalb eines Bereichs eindeutig sein.

Wie können wir also diese API nutzen, um unsere Anwendung zu übersetzen?

Zunächst einmal muss unser **HTML-Code** mit dem Attribut data-translate erweitert werden:

```

<!doctype html>
<html lang="en">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="bulma.min.css">
    <title data-translate="Title">Translation using resource strings - Manual</title>
    <script src="translate.js"></script>
  </head>

  <body>
    <div class="section">
      <div class="box">
        <h2 class="title is-2"
          data-translate="Header">
          Translation using HTMLTranslator
        </h2>
        <p class="block"
          data-translate="Paragraph">
          This text will be translated using data-translate.
        </p>
        <div class="field is-horizontal">
          <div class="field-label is-normal">
            <label class="label"
              data-translate="Name">
              Name
            </label>
          </div>
          <div class="field-body">
            <div class="field">
              <p class="control">
                <input class="input"
                  type="text"
                  placeholder="Type your name"
                  data-translate="Input-placeholder">
              </p>
            </div>
          </div>
        </div>
        <button
          id="btn-translate"
          class="button is-link"
          data-translate="Button">
          Translate this page
        </button>
      </div>
    </div>
    <script>
      window.onload=rtl.run;
    </script>
  </body>
</html>

```



Zweitens müssen wir eine Seite mit den Übersetzungen erstellen.

Beachten Sie, dass im Gegensatz zum Resource-String-Mechanismus alle Übersetzungen vorhanden sein müssen, auch die Originalsprache: Es gibt keine Möglichkeit, die Übersetzung zurückzusetzen. Die Datei sieht wie folgt aus:

```
{
  "nl" : {
    "html" : {
      "Title": "Vertaling met HTMLTranslator",
      "Button": "Vertaal deze pagina",
      "Header": "Vertaling met resourcestrings",
      "Paragraph": "Deze tekst wordt vertaald.",
      "Input-placeholder": "Typ uw naam",
      "Name": "Naam"
    }
  },
  "en" : {
    "html" : {
      "Title" : "Translation using HTMLTranslator",
      "Header" : "Translation using HTMLTranslator",
      "Button": "Translate this page",
      "Paragraph" : "This text will be translated using data-translate",
      "Input-placeholder": "Type your name",
      "Name": "Name"
    }
  }
}
```

Wie Sie sehen können, haben wir uns für den Bereich **"html"** entschieden. Wir speichern diese Datei als `lang.json`, neben unserer `index.html` Seite.

Zum Schluss passen wir den Code an. Das Programm beginnt mit der Initialisierung einer Instanz von `THTMLTagTranslator`. Es setzt den Standardbereich auf `'html'` und setzt die `LanguageSource` auf `lsMulti`, da wir mehrere Sprachen in unserer Datei haben.

Die `LanguageFileURL` wird auf `lang.json` gesetzt, den Namen der Datei, in der wir die Übersetzungen gespeichert haben.

eingeben, wird sie von demselben Ort wie die Seite `index.html` heruntergeladen.

```
uses
  JS, Classes, SysUtils, Web, Rtl.HTMLTranslate;
Var
  aButton: TJSHTMLButtonElement;
  aTranslator: THTMLTagTranslator;
  IsDutch: Boolean;
  // some code removed for clarity...

Procedure Init;
begin
  aTranslator:=THTMLTagTranslator.Create(nil);
  aTranslator.LanguageSource:=lsMulti;
  aTranslator.LanguageFileURL:='lang.json';
  aTranslator.OnLanguageLoaded:=@DoLangLoaded;
  aTranslator.DefaultScope:='html';
  aButton:=TJSHTMLButtonElement(Document.GetElementByID('btn-translate'));
  aButton.onclick:=@DoTranslation;
end;

begin
  Init;
end.
```



Wie Sie sehen können, ist der Code zum Übersetzen recht einfach. Sobald die Sprache eingestellt ist, wird DoLangLoaded ausgelöst, und der TranslateBelowHTMLTag kann für die eigentliche Übersetzung verwendet werden.

Beachten Sie, dass der Code dem TranslateBelowHTMLTag -Aufruf keinen zu verwendenden Bereich übergibt: der Standard Bereich auf **html** gesetzt, und das ist ausreichend. Schließlich ist die Funktion OnLanguageLoaded auf DoLangLoaded **gesetzt** ist, wo wir die eigentliche Übersetzung vornehmen:

```
procedure DoLangLoaded(Sender: TObject; aLanguage: String);
begin
  aTranslator.TranslateBelowHTMLTag(TJSHTMLTag(Document.body));
  aTranslator.TranslateBelowHTMLTag(TJSHTMLTag(Document.head));
end;
```

Wie Sie sehen können, rufen wir die Methode TranslateBelowHTMLTag zweimal auf:

- ◆ einmal für das Element body.
- ◆ einmal für das head-Element. Dieser Aufruf übersetzt das title-Tag und setzt damit den Seitentitel.

Beachten Sie, dass wir dem TranslateBelowHTMLTag-Aufruf keinen Bereichsnamen übergeben. Dies führt dazu, dass der Code den Standardbereich 'html' verwendet. Der onclick-Handler unserer Schaltfläche ist jetzt ganz einfach:

```
function DoTranslation(aEvent: TJSMouseEvent): boolean;
begin
  Result:=True;
  IsDeutsch:=Not IsDeutsch;

  if IsDeutsch
  then aTranslator.Language:='de'
  else aTranslator.Language:='en'
end;
```

Durch das Einstellen der Sprache wird die Übersetzung geladen, falls erforderlich, und wenn sie geladen ist, wird unser DoLangLoaded-Ereignishandler aufgerufen.



4 ERSTELLEN EINER DATEI MIT ÜBERSETZUNGEN

Es ist nicht notwendig, die **JSON-Datei** vollständig von Hand zu erstellen: Das **PAS2JS-Repository** enthält ein Tool, mit dem Sie alle Texte, die übersetzt werden müssen, aus den **HTML-Dateien** bezogen werden können. Das Tool heißt **extractlang**, Sie finden es unter Tools der **PAS2JS-Distribution**. Derzeit ist nur ein Befehlszeilentool verfügbar. Wenn Sie es ausführen durch die Befehlszeilenoption mit der Option **-h** wird eine Hilfeseite angezeigt:

Verwendung: `./extractlang [Optionen]`

Wenn Optionen eine oder mehrere der folgenden Optionen sind:

-h --help	Dieser Hilfetext
-c --clear	Löscht die JSON-Ausgabedatei (<i>standardmäßig wird die vorhandene Datei aktualisiert</i>).
-d --html-dir=DIR	Verzeichnis mit zu überprüfenden HTML-Dateien
-f --file-mode=MODE	Dateimodus festlegen: einer der beiden Modi einfach oder mehrfach
-o --output=FILE	Datei zum Schreiben von JSON-Übersetzungen (<i>kann je nach Dateimodus ein Suffix erhalten</i>)
-l --languages=LIST	Komma-getrennte Liste der zu erstellenden Sprachen
-m --minify	Ausgabe verkleinern
-n --name=NAM	Set name of data-tag to NAME (data-NAME)
-r --recurse	In Unterverzeichnissen des HTML-Verzeichnisses rekursieren
-s --single-scope=SCOPE	Alle Übersetzungsnamen in einen einzigen Bereich legen
-t --trash-values	Trash-Werte für andere Sprachen

Um zu funktionieren, benötigt das Tool mindestens zwei Optionen: **-h** und **-o**. Die **-h** Option gibt ein Verzeichnis an, in dem das Tool nach **HTML-Dateien** sucht. Die **-o** Option spezifiziert den Namen einer **JSON-Sprachdatei**, die erstellt oder aktualisiert werden soll.

Mit diesen Optionen durchsucht das Tool das Verzeichnis nach **HTML-Dateien** (*und Unterverzeichnissen, wenn -r angegeben ist*), sammelt alle **HTML-Tags** mit dem Attribut `data-translate`, und schreibt den inneren Text dieses Tags in eine **JSON-Datei**.

Standardmäßig wird nur eine englische Sprache erstellt.

Mit der Option **-l** können Sie aber auch den Text für weitere Sprachen erstellen: Für jede Sprache prüft das Tool, ob der genannte Text vorhanden ist, und wenn nicht, fügt es den Text hinzu.

Wenn wir das Programm im Verzeichnis unserer Beispielanwendung mit der folgenden Befehlszeile ausführen: `extractlang -d . -o lang.json -l 'en,nl' -s html`

Wir geben die Option **-s** an, weil dies der Bereich ist, den wir im Code verwendet haben. Wenn kein Bereich angegeben wird, wird für jeden verarbeitete **HTML-Bestand** ein eigener Bereich verwendet, wobei der Bereichsname der **HTML-Dateiname** ist, kleingeschrieben und ohne Erweiterung. Das Tool gibt Ihnen eine Diagnose aus:

```
Searching ./index.html for translatable terms, adding to scope : html
Found 6 translatable terms
Collected 1 message scopes
Copied 1 new scopes with 6 words, added 0 new words in existing scopes.
```



Wir erhalten dann die folgende Datei:

```
{
  „en“ : {
    „html“ : {
      „Title“ : „Translation using resource strings - Manual“,
      „Header“ : „Translation using HTMLTranslator“,
      „Paragraph“ : „This text will be translated using data-translate.“,
      „Name“ : „Name“,
      „Input-placeholder“ : „Type your name“,
      „Button“ : „Translate this page“
    }
  },
  „nl“ : {
    „html“ : {
      „Title“ : „Translation using resource strings - Manual“,
      „Header“ : „Translation using HTMLTranslator“,
      „Paragraph“ : „This text will be translated using data-translate.“,
      „Name“ : „Name“,
      „Input-placeholder“ : „Type your name“,
      „Button“ : „Translate this page“
    }
  }
}
```

Diese Datei ist bereit für die Übersetzung.

(Beachten Sie, dass die Ausgabe bei Verwendung der Option `minify` nicht für den Menschen lesbar und manuell schwer zu übersetzen ist)

Standardmäßig lädt das Tool eine bestehende Datei und fügt einfach neue Texte hinzu, so dass es sich für die Aktualisierung bestehender Übersetzungen eignet.

Als Hilfe beim Übersetzen können Sie die Option `-t` (*Papierkorb*) angeben.

In diesem Fall enthält die erste Sprache den Originaltext aus der **HTML-Datei**, und alle neuen Wörter in den anderen Sprachen enthalten chinesische Zeichen.

Dies kann als visuelle Kontrolle dienen, wenn Sie die Sprache in der eigentlichen **HTML-Seite** ändern: Alle nicht übersetzten Begriffe werden als chinesische Zeichen angezeigt, die (für nicht Chinesischen Menschen) hervorstechen.

5 SCHLUSSFOLGERUNG

PAS2JS bietet alle Werkzeuge, die Sie für die Erstellung einer lokalisierten Anwendung benötigen.

Die Mechanismen für die Übersetzung von Resourcedtrings und **HTML-Seiten** sind insofern ähnlich, als dass sie eine **JSON-Datei** verwenden:

ermöglicht es Ihnen, Übersetzungen von Resourcedtrings und **HTML-Tags** in einer einzigen **JSON-Datei** zu haben. Wie immer gibt es noch Raum für Verbesserungen, wie z.B. die Integration in die **Lazarus IDE**.

Ein **GUI-Programm** für das Extraktionstool und ein Tool zur Verwaltung von Übersetzungen.

Diese werden mit der Zeit erstellt. Aber alle grundlegenden Mechanismen sind vorhanden, so dass der ungeduldige Entwickler bereits loslegen kann.



Lazarus Handbook Pocket (UK) + Subscription (auch Deutsch) + PDF (auch Deutsch)

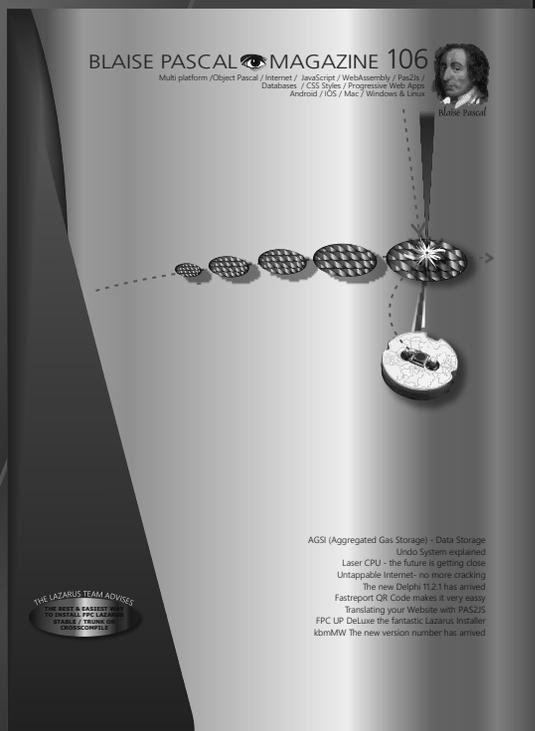
- English
- Printed black & white
- 2 Volumes
- Sewn +2 ribbons
- PDF included
- 934 Pages
- Weight: 2kg
- Shipment included
- Extra protected
- Including 40 Example projects and extra programs

**SPECIAL
OFFER**
€ 75



LAZARUS HANDBOOK
FOR PROGRAMMING WITH FREE PASCAL AND LAZARUS

PDF File
Including electronic Index



AGSI (Aggregated Gas Storage) - Data Storage
Undo System explained
Laser CPU - the future is getting close
Untappable Internet- no more cracking
The new Delphi 11.2.1 has arrived
Fastreport QR Code makes it very easy
Translating your Website with PAS2JS
FPC LP Decline the fantastic Lazarus installer
kbnMW The new version number has arrived

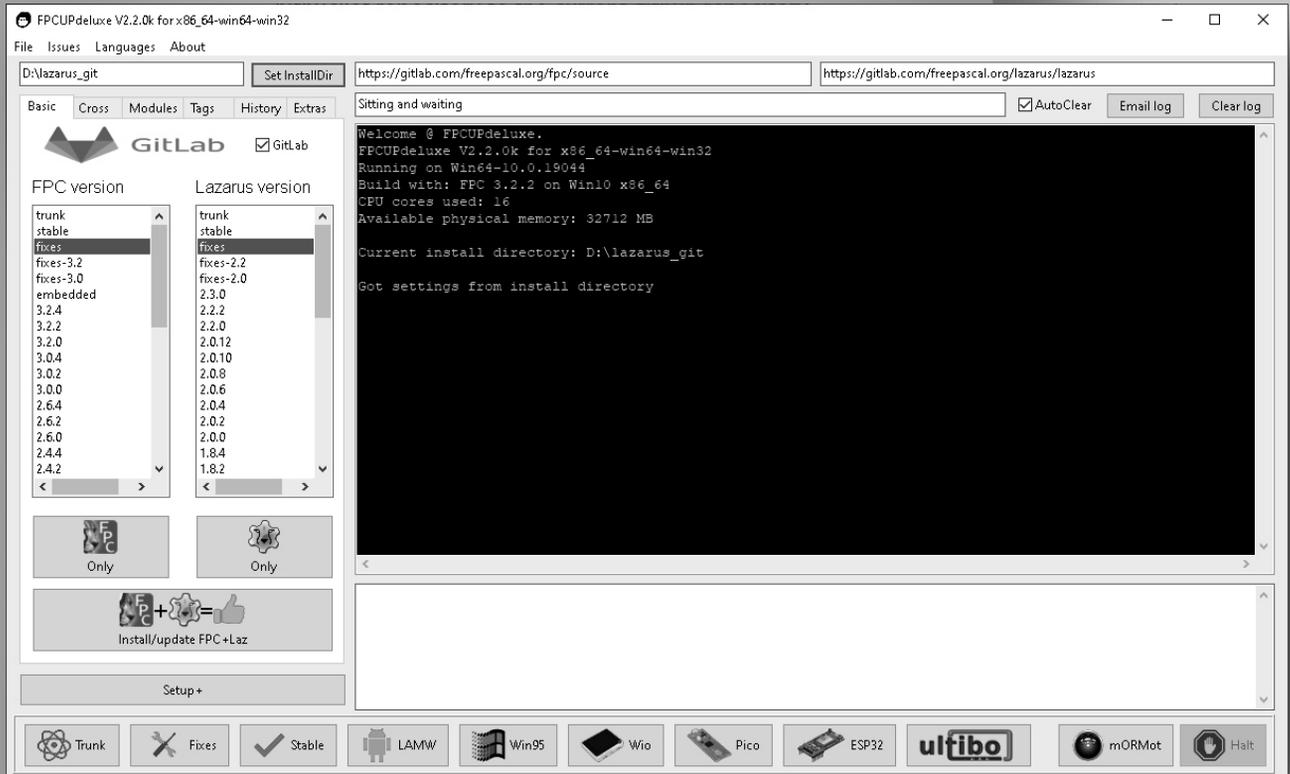
Installation von Freepascal und Lazarus mit FPCUPdeluxe auf Linux, Mac, Windows und anderen Betriebssystemen.



ABSTRACT Jos Wegman

Es gibt verschiedene Möglichkeiten, FPC und Lazarus zu installieren. In diesem Artikel werde ich eine Methode beschreiben, bei der ich **fpcupdeluxe** verwende. Mit dieser Methode haben Sie die Wahl zwischen verschiedenen Versionen von FPC und Lazarus wählen. Einschließlich Cross-Compilern und/oder Modulen.

Keywords: Fpcupdeluxe, FPC, Lazarus, installer, cross-compilers, modules.



1 INTRODUCTION

Das Programm ist kostenlos. Es gibt viele Möglichkeiten, **Lazarus** und / oder **FPC** zu installieren. Für die gängigen Plattformen gibt es Installationsprogramme für die stabilen Versionen auf der Website <https://www.lazarus-ide.org> im Download-Bereich.

In diesem Artikel möchte ich Ihnen einen anderen Ansatz vorstellen.

Die Anwendung **fpcupdeluxe**, gepflegt von der Person mit dem github-Namen "**LongDirtyAnimAlf**".

Das ursprüngliche Projekt wurde von Reinier Olieslagers ins Leben gerufen. Reinier ist leider am 4. Dezember 2014 verstorben.

Im August 2015 hat **LongDirtyAnimAlf** (alias **Alfred Glänzer**) das gesamte **Bit-Bucket-Repository** Reiniers in das aktuelle github Repository übertragen.

Von da an wurde das Programm stetig verbessert und erweitert. Hier zeige ich Ihnen den Installations Prozess zur Installation einer vollständigen Version von **FPC** und **Lazarus** aus den Quellen kompiliert unter Verwendung von **fpcupdeluxe**.

DAS LAZARUS-TEAM RÄT IHNEN
**DER BESTE
 UND EINFACHSTE WEG
 FPC LAZARUS ZU INSTALLIEREN
 STABLE / TRUNK ODER
 CROSSCOMPILE**

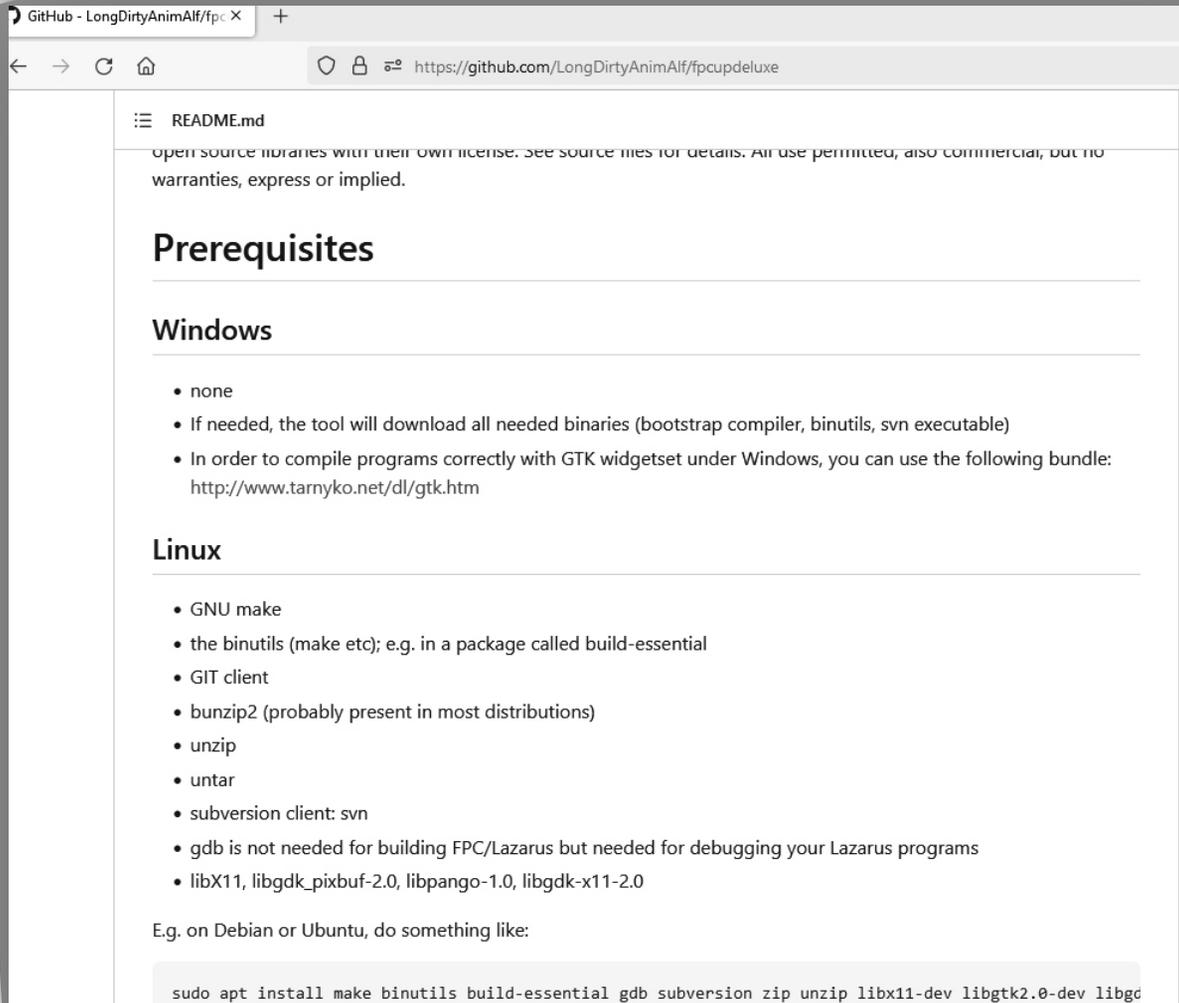


2 DOWNLOAD UND VORAUSSETZUNGEN

Im Repository finden Sie eine **README.ME**-Datei, die die notwendigen Informationen über die Voraussetzungen für die Installation enthält.

- ◆ Für **Windows** gibt es keine Voraussetzungen.
- ◆ Für **Linux** gibt es mehrere Pakete, die installiert werden müssen.
- ◆ Alle Dateien der Installation befinden sich im selben Verzeichnis, so dass Sie nur in diesem Installationsverzeichnis nach etwas suchen müssen.

Das heißt, alle Installationen des Programms sind unabhängig und Sie können beliebig viele davon erstellen und jeder Installation einen anderen Verzeichnisnamen geben .



GitHub - LongDirtyAnimAlf/fpc X +

← → ↻ 🏠 🔒 https://github.com/LongDirtyAnimAlf/fpcupdeluxe

☰ README.md

open source libraries with their own license. See source files for details. All use permitted, also commercial, but no warranties, express or implied.

Prerequisites

Windows

- none
- If needed, the tool will download all needed binaries (bootstrap compiler, binutils, svn executable)
- In order to compile programs correctly with GTK widgetset under Windows, you can use the following bundle:
<http://www.tarnyko.net/dl/gtk.htm>

Linux

- GNU make
- the binutils (make etc); e.g. in a package called build-essential
- GIT client
- bunzip2 (probably present in most distributions)
- unzip
- untar
- subversion client: svn
- gdb is not needed for building FPC/Lazarus but needed for debugging your Lazarus programs
- libX11, libgdk_pixbuf-2.0, libpango-1.0, libgdk-x11-2.0

E.g. on Debian or Ubuntu, do something like:

```
sudo apt install make binutils build-essential gdb subversion zip unzip libx11-dev libgtk2.0-dev libgc
```

Abbildung 1: Voraussetzungen

Diese Pakete sind in der Datei **README.ME** aufgeführt. Ich werde die Installation unter **Windows** durchführen. Sie benötigen also **KEINE** zusätzlichen Voraussetzungen. Das Programm **FPCUPDELUXE** können Sie von der Release-Seite des Repositorys herunterladen.

Links

<https://github.com/LongDirtyAnimAlf/fpcupdeluxe>
<https://forum.lazarus.freepascal.org/index.php/topic,26726.0.html>
<https://github.com/LongDirtyAnimAlf/fpcupdeluxe/releases>
<http://www.batterybutcher.com:8880/root/getinfohtml>



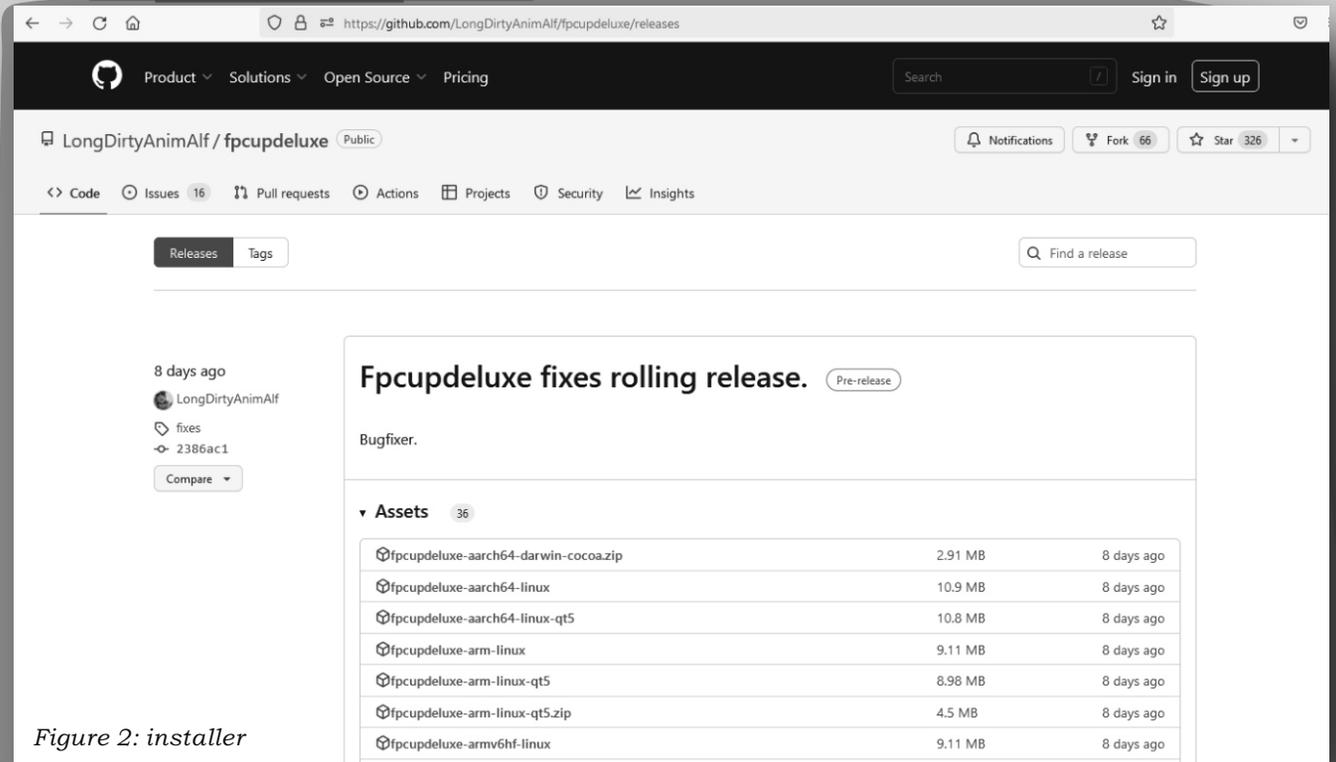


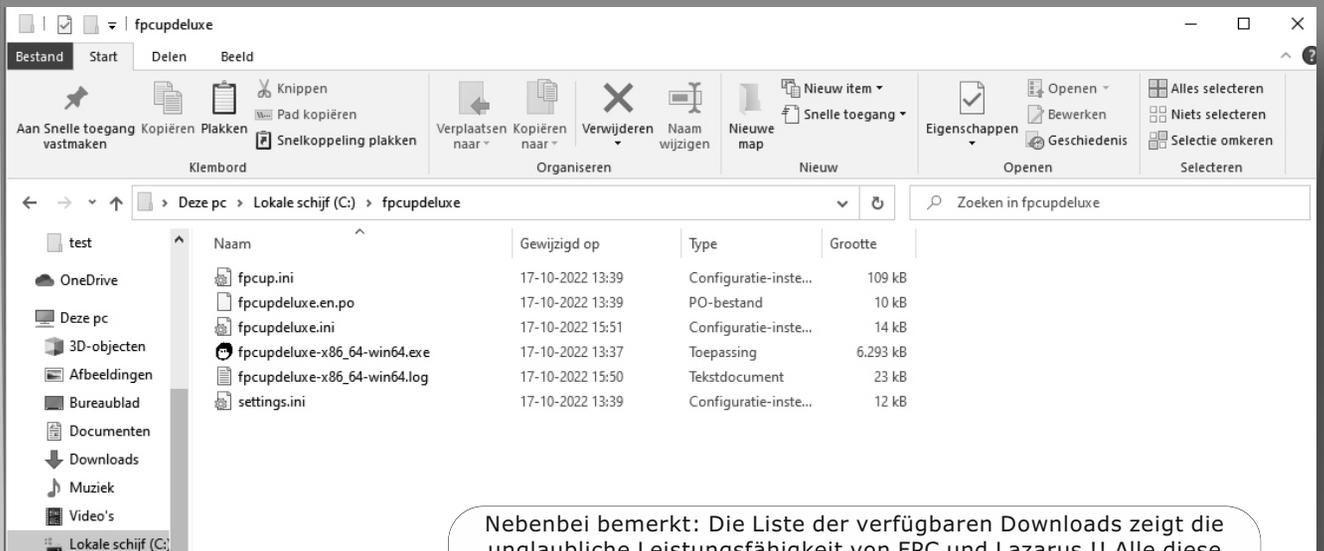
Figure 2: installer

Wählen Sie die für Ihre Plattform geeignete Anwendung. In meinem Fall ist dies die `fpcupdeluxe-x86_64-win64.exe`. Der Einfachheit halber habe ich ein Verzeichnis auf meinem System angelegt, in dem ich das heruntergeladene Programm abgelegt habe. Wenn **FPCUPDELUXE** zum ersten Mal gestartet wird, erstellt es mehrere Konfigurationsdateien, die von `fpcupdeluxe` benötigt werden. Die Konfigurationsdateien werden in demselben Verzeichnis wie **FPCUPDELUXE** gespeichert.

HINWEIS: Zu Ihrer Information: **FPC** selbst empfiehlt, die 32bit-Version unter Windows zu installieren. Der Grund dafür ist die fehlende erweiterte 80bit-Unterstützung unter win64, die die Installation von (32bit) Cross-Compilern, die diesen 80bit-Wert unterstützen, erschwert. Der Wechsel von win32 zu arm32 (RPI) funktioniert also ohne weiteres Zutun. Beim Wechsel von win64 zu arm32 erhalten Sie eine Warnung und müssen die Softfloat-Bibliothek aktivieren [-dFPC_SOFT_FPUX80]. **fpcupdeluxe** wird dies erkennen und eine Warnung ausgeben.

Links

<http://www.jhauser.us/arithmetic/SoftFloat-3/doc/SoftFloat.html>



Nebenbei bemerkt: Die Liste der verfügbaren Downloads zeigt die unglaubliche Leistungsfähigkeit von FPC und Lazarus !! Alle diese Versionen wurden auf einem Win11 PC erstellt. Alle lassen sich mit einem einzigen Klick erstellen !!! (viele in Lazarus kompilieren)

3 AUSWAHL DER INSTALLATIONSFUNKTIONEN

Starten Sie **FPCUPDELUXE**. Unter **LINUX** müssen die richtigen Berechtigungen gesetzt werden, bevor fpcupdeluxe gestartet werden kann. Unter **WINDOWS** doppelklicken Sie einfach auf das Symbol von **FPCUPDELUXE**.

Wenn Sie das Programm zum ersten Mal starten, wird ein Popup-Fenster angezeigt.

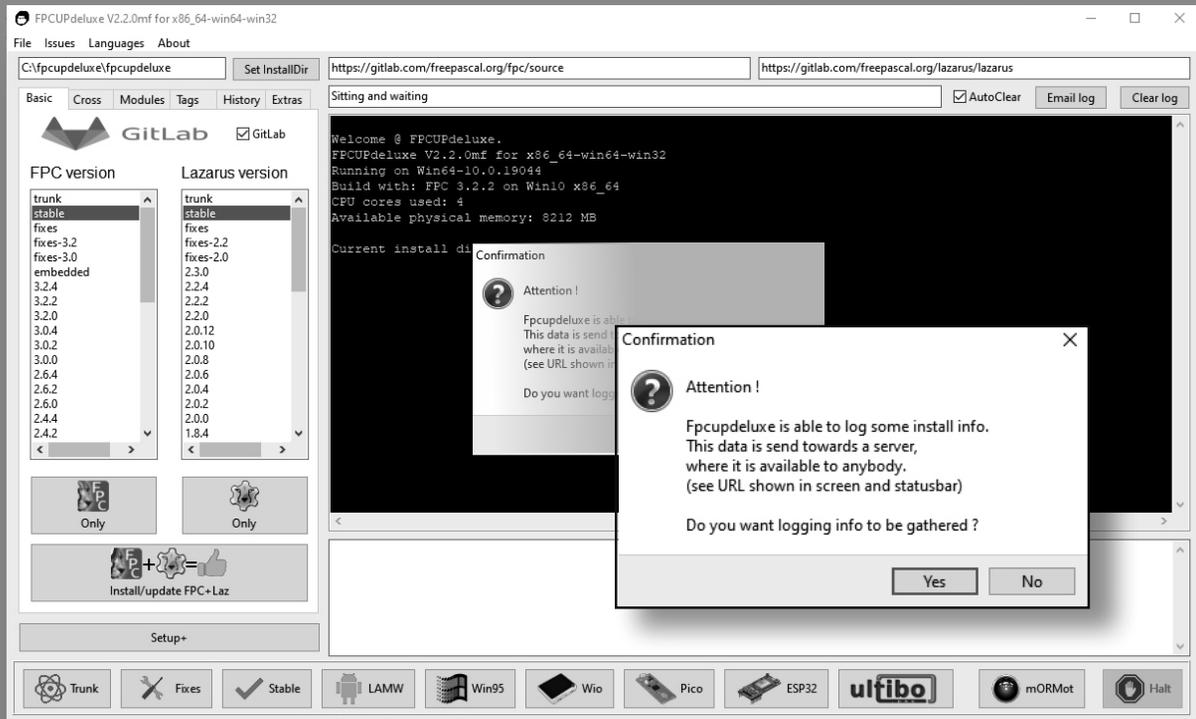


Abbildung 4: Erster Durchlauf

Sie können sich für eine der beiden Möglichkeiten entscheiden, es wird Ihre Installation nicht beeinträchtigen.

Wenn Sie sich für die Freigabe Ihrer Installationsdaten entscheiden, können Sie in der Datei `READ.ME` welche Daten wo gespeichert sind.

Bevor Sie **FPC** und **Lazarus** installieren können, müssen Sie ein paar Parameter einstellen.

Zunächst müssen Sie entscheiden, in welchem Verzeichnis Sie **FPC** und **Lazarus** installieren möchten. Sehen Sie sich das Bearbeitungsfeld in der oberen linken Ecke an.

Verwenden Sie die Schaltfläche "Set InstallDir", um das Verzeichnis Ihrer Wahl auszuwählen.

Der nächste Schritt ist die Auswahl der **FPC-Version**, in diesem Fall "stable".

Der letzte Parameter ist die **Lazarus-Version**, ebenfalls "stable".

Jetzt ist die Anwendung so eingestellt, dass sie die gewünschten Versionen von **FPC** und **Lazarus** installiert.

Um die Installation zu starten, klicken Sie auf den Button "FPC + Laz installieren/aktualisieren". Sie erhalten einen Bestätigungsdialog. (Siehe Abbildung 5 auf Seite 5 dieses Artikels).

Im Falle von einer Installation für Windows ist es ratsam, "Win32" zu wählen, da diese Version automatisch die Kompilierung für "Win64" enthält, so dass es nicht notwendig ist, diese ebenfalls zu installieren

3. AUSWAHL DER INSTALLATIONSFUNKTIONEN - FORTSETZUNG

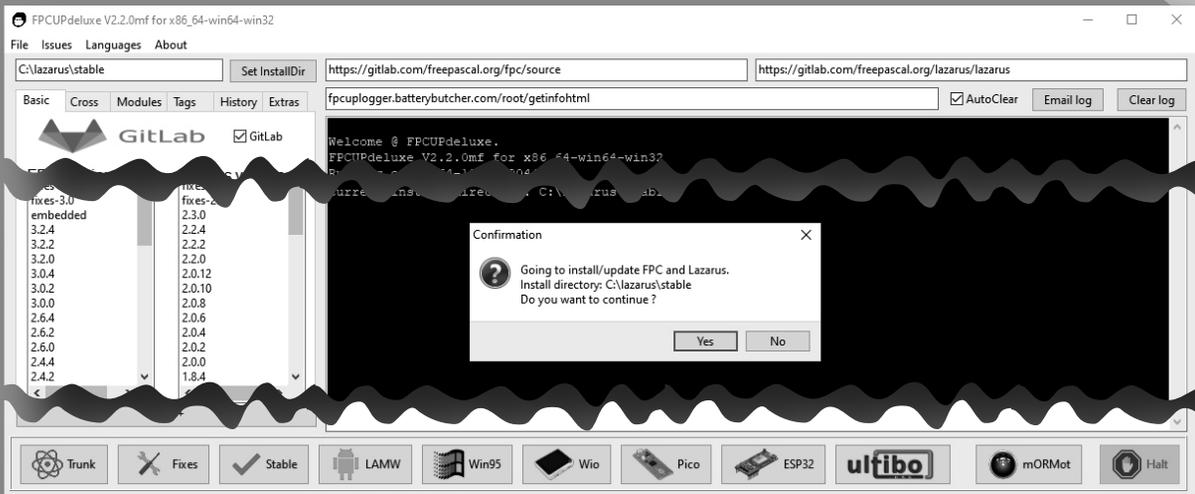


Abbildung 5: Bestätigung

Wenn Sie "Ja" drücken, können Sie sich zurücklehnen und entspannen. Warten Sie darauf, dass etwas Gutes passiert. Der Installationsvorgang wird durchgeführt. Fortschritts- und Aktionsmeldungen werden im Terminalfenster von **FPCUPDELUXE** angezeigt. Ein Beispiel für diese Meldungen finden Sie in Abbildung 6.

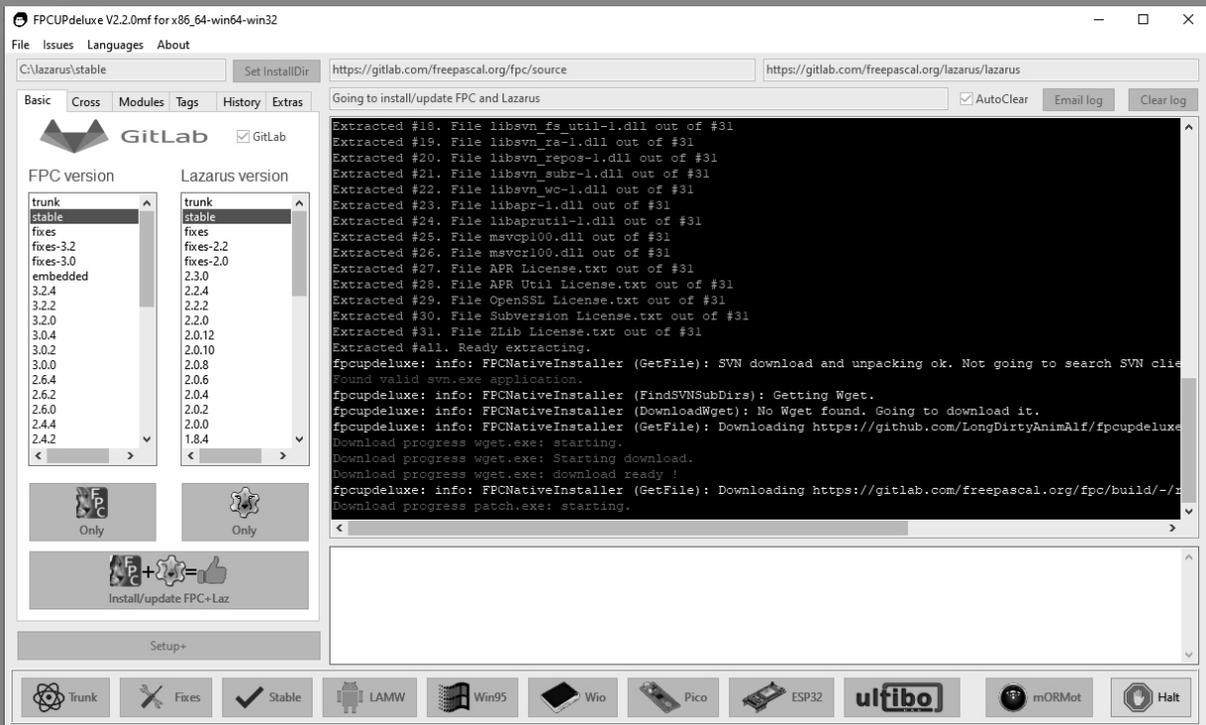


Abbildung 6: Kommentare

Wenn die Installation erfolgreich war, sehen Sie Bild 7. Siehe nächste Seite



3 AUSWAHL DER INSTALLATIONSFUNKTIONEN FORTSETZUNG

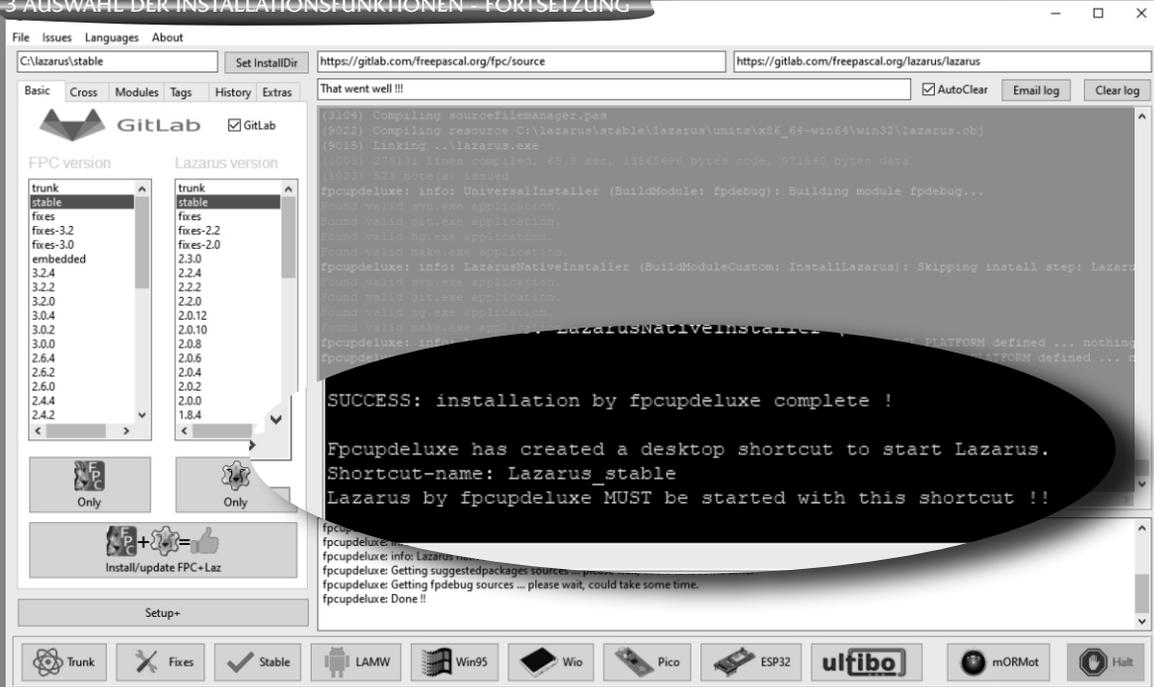


Abbildung 7: Erfolg

Bitte beachten Sie den letzten Satz im Terminal-Fenster.

FPCUPDELUXE hat ein Symbol auf dem Desktop erstellt, um **Lazarus** zu starten.

Dieses Symbol ist zwingend erforderlich, um **Lazarus** zu starten.

4 INSTALLATION DES/DER CROSS-COMPILER(S)

Jetzt ist **Lazarus** wie gewohnt einsatzbereit.

Jetzt werden wir einen Cross-Compiler installieren.

FPCUPDELUXE läuft noch.

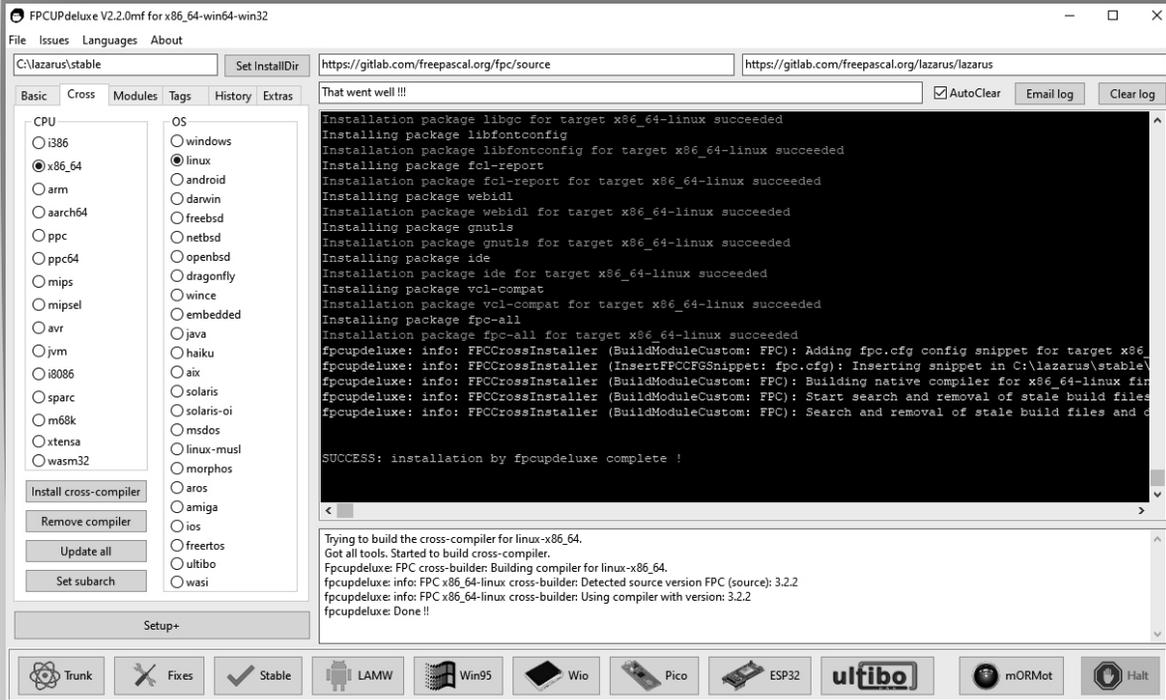


Abbildung 8: Cross-Compiler



4 INSTALLATION DES/DER CROSS-COMPILER(S) - FORTSETZUNG

Um einen Cross-Compiler zu installieren, müssen einige Parameter festgelegt werden. Wählen Sie zunächst die Registerkarte "cross". Auf dieser Registerkarte finden Sie die verschiedenen CPU- und OS-Möglichkeiten. Wählen Sie die Kombination des Cross-Compilers, die Sie installieren möchten. In diesem Beispiel ist die CPU = "x86_64" und das Betriebssystem = "Linux". Mit dieser Kombination wird der Cross-Compiler für 64-Bit-Linux-Anwendungen installiert.

HINWEIS: Bei einer ersten Cross-Installation wird ein Fehler aufgrund fehlender Tools gemeldet. Erlauben Sie **FPCUPDELUXE**, diese Tools herunterzuladen und fahren Sie fort. Drücken Sie den Button „Cross-Compiler“ installieren und lassen Sie **FPCUPDELUXE** die Arbeit machen. Nach der Installation des Cross-Compilers wird eine Erfolgsmeldung im Terminalfenster angezeigt.

5 INSTALLATION DER MODULE

Fpcupdeluxe ist in der Lage, mehrere Komponentenpakete zu installieren. Die so genannten Module. Wählen Sie zunächst die Registerkarte "Module".

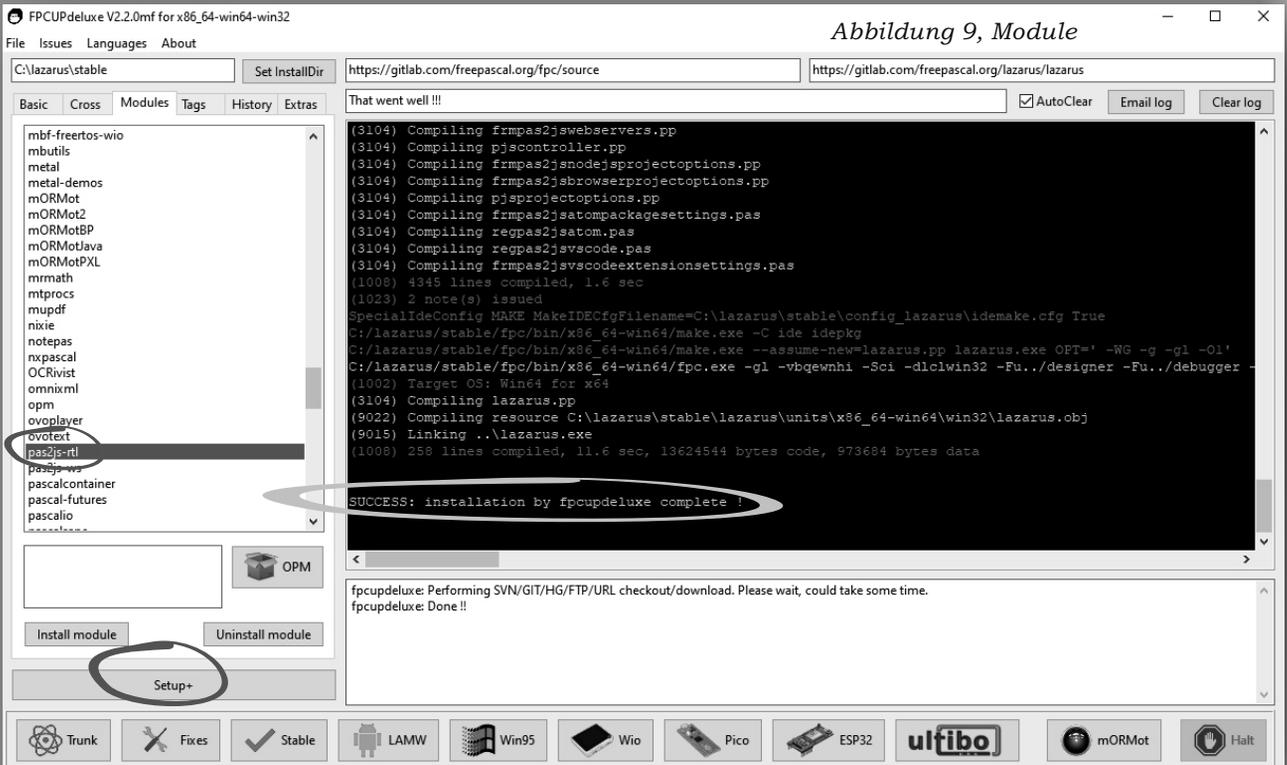


Abbildung 9, Module

Um zum Beispiel den **pas2js** Compiler, das Projekt und die **rtl**-Dateien zu installieren, können Sie das Modul "pas2js-rtl" auswählen, auf den Button "Modul installieren" klicken und der **PAS2JS** Compiler wird installiert.

Dieser Vorgang kann für jedes Modul in der Listbox wiederholt werden. Nach erfolgreicher Installation eines Moduls erhalten Sie eine Erfolgsmeldung im Terminalfenster.

6 SCHLIEßEN SIE DIE INSTALLATION AB UND BEGINNEN SIE MIT DER ARBEIT.

Zu diesem Zeitpunkt ist die vollständige Installation von **FPC**, **Lazarus**, Cross-Compiler und Modulen abgeschlossen. **FPCUPDELUXE** kann beendet werden. Wie bereits erwähnt, erstellt **FPCUPDELUXE** ein Symbol auf dem Desktop, um **Lazarus** zu starten. Sie müssen diese Verknüpfung verwenden, um **Lazarus** zu starten. Sie können in den Eigenschaften des Programmstarters nachsehen, welche Befehlszeilenparameter angegeben werden müssen, um **Lazarus** ordnungsgemäß zu starten. Während der Verwendung von **FPCUPDELUXE** habe ich nicht viele Optionen des Programms geändert. Auf dem Hauptbildschirm links unten gibt es einen Button „Setup+“. Wenn Sie den Button drücken, wird die Abbildung auf der nächsten Seite angezeigt.



Abbildung 10: Das Startersymbol



Advanced settings

Proxy settings

HTTP proxy URL: HTTP proxy port:

HTTP proxy username: HTTP proxy password:

Miscellaneous settings

- Get FPC/Laz repositories.
- Get package repositories.
- Include LCL with cross-compiler.
- FPC/Laz rebuild only.
- Use system FPC for Lazarus.s
- Include Help.
- Split FPC source and bins.
- Split Lazarus source and bins.
- Use wget/libcurl as downloader.
- Use jobs for GNU make.
- Be extra verbose.
- Auto-switch repo URL.
- Send location and install info.
- Only use fpcup bootstrappers.
- Use local repo-client.
- Check for fpcupdeluxe updates.
- Enable software emulation of 80 bit floats.
- Allow patching of sources by online patches.
- Re-apply local changes when updating.
- Add context for FPC and Lazarus files.
- Always ask for confirmation.

Options Override

FPC options Debug

Laz. options Debug

Branch and revision

FPC branch FPC hash/tag

Laz. branch Laz. hash/tag

Source patching

Add FPC patch Add Laz. patch

Rem. FPC patch Rem. Laz. patch

Pre and post install scripts

FPC pre FPC post

Lazarus pre Lazarus post

CPU/OS Subarch

Select CPU Select OS

Search options

fpcup full auto custom

Cross Build Options Override (i.e. -CfSoft)

Compiler Override

ARM target

none armel armeb armhf

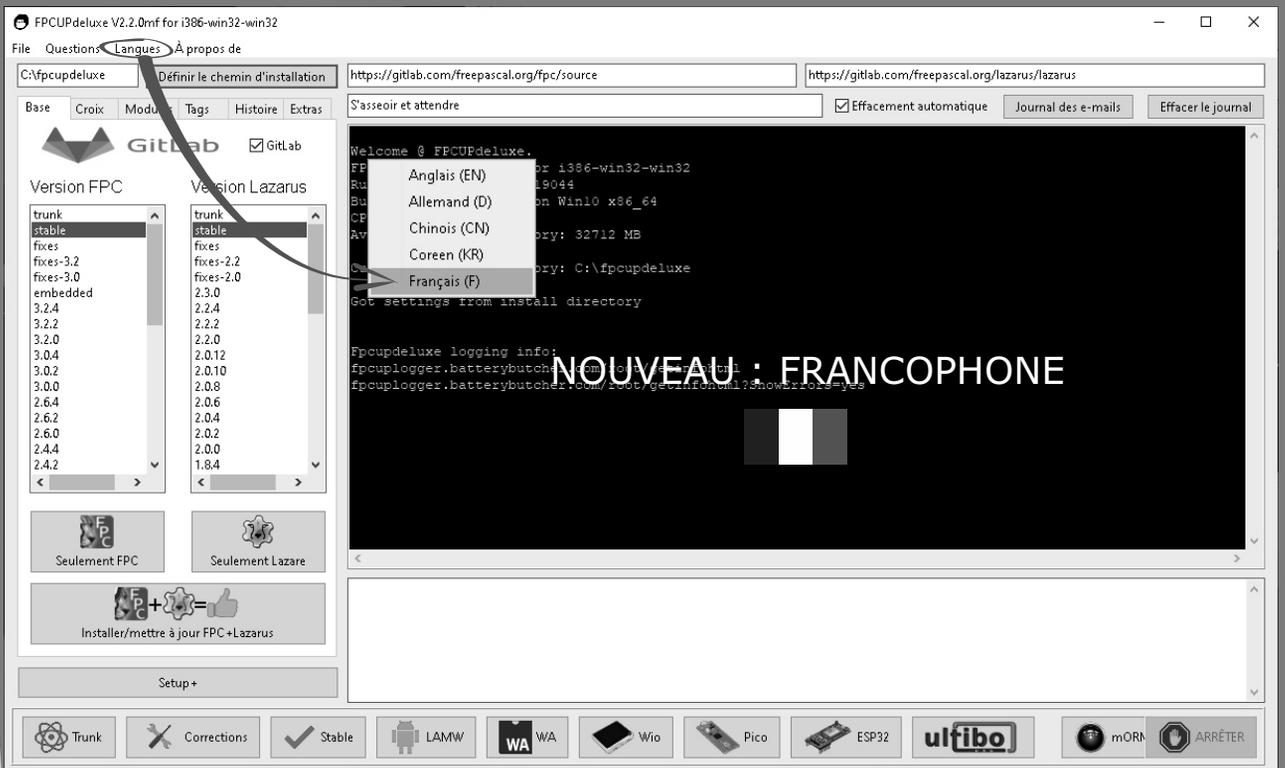
Abbildung 11: Extra Setup

Dieser Bildschirm bietet die Möglichkeit, das Verhalten von fpcupdeluxe im Detail anzupassen.

7 SCHLUSSFOLGERUNG

In diesem Artikel habe ich Ihnen gezeigt, wie Sie **FPC**, **Lazarus**, **Cross-Compiler** und **Module** mit Hilfe von **FPCUPDELUXE** installieren können. Der Hauptvorteil dieser Methode besteht für mich darin, dass es möglich ist, mehrere Versionen von **FPC** und **Lazarus** zu installieren, ohne dass sich die einzelnen Versionen gegenseitig beeinflussen. Das macht es einfach, einen aktuellen Trunk-Compiler zu installieren und diesen zum Kompilieren meines Quellcodes ohne große Probleme zu verwenden. Jede installierte Version wird in einem eigenen Verzeichnis installiert, sogar die Konfiguration und die Module befinden sich im Installationsverzeichnis. Wenn ich eine Version entfernen muss, genügt das Löschen eines Verzeichnisses, um dies zu tun. Keine Dateien, die von einer bestimmten Version von **FPC**, **Lazarus** benötigt werden, werden außerhalb des Installationsverzeichnisses gespeichert. Verwenden Sie das entsprechende Thema im **Lazarus-Forum** für zusätzliche Informationen und Fragen.

SPRACHEN





starter

expert

ZUSAMMENFASSUNG

Laser nutzen Gase, Feststoffe oder Flüssigkeiten als Quelle optischer Verstärkung und erzeugen kohärente Laserstrahlen. Diese bestehen aus Photonen - Teilchen, die die kleinste diskrete Menge eines Quants elektromagnetischer Strahlung darstellen. Kohlendioxid (CO₂)-Laser verwenden Gas zur optischen Verstärkung, während photonische Laser, die energieeffizienter und kompakter sind, Halbleiterkristalle zur direkten Verstärkung der Strahlen verwenden.

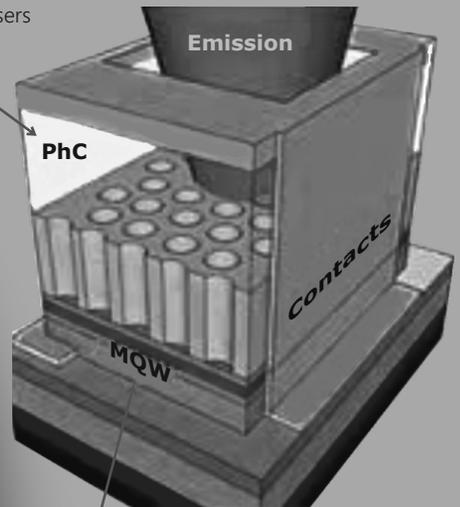
EINFÜHRUNG

Heute wird die Laserfertigung weitgehend von CO₂-Lasern und Faserlasern dominiert - selbst für Routineaufgaben wie das Herstellen winziger Löcher in Leiterplatten für Smartphones. Man geht davon aus, dass photonische Laser, die Strom direkt nutzen, das Potenzial haben, die Energiekosten der Fertigung zu senken. Eine Halbierung der Kosten und Emissionen beim Laserschneiden sollte in Zukunft möglich sein, und vielleicht lassen sich die Kosten aufgrund ihrer kompakten Größe sogar noch weiter senken. Bevor diese Kosteneinsparungen jedoch realisiert werden können, sind noch erhebliche technische Hürden zu überwinden. Die Erfindung der **PCSELS (Photonic-Crystal Surface-Emitting Lasers)** an der japanischen **Universität Kyoto** im Jahr 1999 verbesserte die Qualität der Photonenstrahlen im Labor. Um sie jedoch effektiv in der Produktion einzusetzen, muss ein Kompromiss zwischen Leistung und Strahlqualität gefunden werden. Dies ist eines der Hauptziele des Programms "**Photonik und Quantentechnologie für die Gesellschaft 5.0**".

(Photonik ist eine Technologie, die den Schwerpunkt auf auf das Erkennen, Erzeugen, Transportieren und Verarbeiten von Licht. Die Technologie ist ähnlich wie die Elektronik. Es gibt nur einen großen Unterschied: Anstelle von Elektronen werden Photonen zur Übertragung von Informationen verwendet. Dies ist sowohl energieeffizienter als auch schneller. Photonik wird zum Beispiel in Displays, Beleuchtungen, Lasern, Solarzellen, Sensoren und Glasfasernetzen eingesetzt.)

Obwohl die photonischen Laser, die hergestellt werden, die mit der Produktion verbundenen Probleme noch nicht vollständig gelöst haben, verwendete die Gruppe Spiegel und formte eine **PCSEL-Mikrostruktur** um, um den Lichtverlust im Jahr 2021 zu reduzieren. Mit diesem Aufbau haben sie die erste erfolgreiche Bearbeitung einer Metalloberfläche mit einem Single-Chip **PCSEL** durchgeführt.

Photonic crystal lasers



Multiple Quantum Well Schematic

Der Multi-Quantum-Well (MQW)-Solver ist ein 1D-Physik-basierter Solver zur Berechnung der optischen und elektronischen Eigenschaften von Multi-Quantum-Well-Stapeln.

Der Solver liefert die Koeffizienten für Verstärkung und spontane Emission, sowie das elektronische Banddiagramm, die Bandstruktur, und Wellenfunktionen. Die Ergebnisse des MQW-Solvers werden häufig als Eingaben für das TWLM-Element in INTERCONNECT verwendet, um kantenemittierende Halbleiterlaser oder Elektroabsorptionsmodulatoren zu modellieren. Das MQW-Solver-Simulationsobjekt in der Finite-Elemente-Entwurfsumgebung (FE IDE) bietet eine grafische Benutzeroberfläche (GUI) für die Ausführung der MQW-Simulationen. Eine Reihe von skriptbasierten Funktionen (buildmqwmaterial, mqwgain, mqwindex) kann ebenfalls zur Ausführung des MQW-Solvers verwendet werden.



Der nächste Schritt besteht darin, die Laser zu vergrößern und Verpackungs- und Kühlsysteme zu entwickeln. In der Zwischenzeit hat das Programm eine Reihe von Technologien hervorgebracht, die bereits im Einsatz sind oder demnächst eingesetzt werden. **PSCEL-Sensoren** werden bereits im nächsten Jahr in kommerziellen Geräten zum Einsatz kommen.

2 LIGHT WAVES

Viele Lasertechnologien, die das größte Segment des Lasermarktes ausmachen, kamen bereits vor 20-30 Jahren auf den kommerziellen Markt. **Japan** ist in einer guten Position, um auf dieser Welle zu reiten, sagte **Šarunas Vaškelis**, CEO von **Direct Machining Control**, einem litauischen Entwickler von **Lasersoftware** mit Sitz in **Vilnius**.

Obwohl **Japan** seit seiner Blütezeit in den 1990er Jahren bei der Produktion von Lasertechnologie hinter den Konkurrenten in den USA und Deutschland zurückgeblieben ist, bleibt es eine wichtige Kraft in den verwandten Branchen **Optik** und **Photonik**.

So ist es beispielsweise weltweit führend in der **Photonik**, die Technologien wie **Laserdioden** und **LEDs**, **Glasfaseroptik**, **optische Sensoren**, **Displays** und **Solarzellen** umfasst. Werfen Sie einen Blick auf das Land, in dem die Hersteller ihren Hauptsitz haben, und Sie werden sehen, dass Japan im Jahr 2015 mit etwa 30% des Weltmarktes an der Spitze stand, eine Anmerkung von **Vaškelis**, der einen Bericht über die Zusammenarbeit erstellt hat.

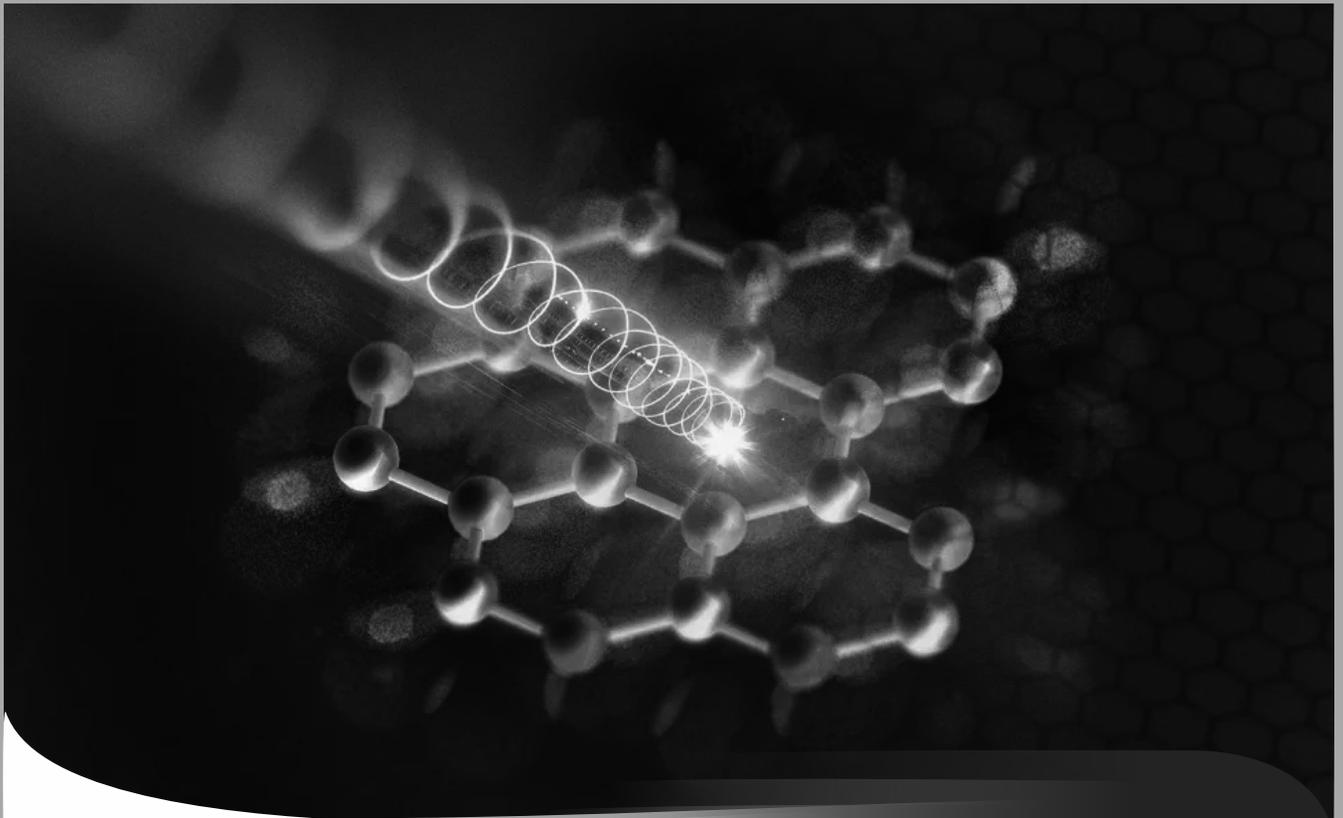
Die jüngsten Weltereignisse beschleunigen auch die Nachfrage nach neuen Technologien für effiziente, sichere und flexible intelligente Fabriken. So hat beispielsweise der Krieg in der Ukraine - einer wichtigen globalen Öl- und Gasquelle - die Aufmerksamkeit auf die Möglichkeit von Unterbrechungen der Energieversorgung gelenkt, und die COVID-19-Pandemie hat den Wunsch nach einem Wechsel zu ferngesteuerten Anlagen verstärkt.

Einzigartig kleine, effiziente **PCSELS** für **Light Detection and Ranging (LiDAR)** bieten neue Möglichkeiten für ferngesteuerte **Fabrikroboter** und Elektrofahrzeuge. **KI** für die Materialhandhabung und photonische Kommunikationssysteme erreichen ebenfalls realistische Anwendungsfälle und könnten flexiblere Systeme und eine noch nie dagewesene Datensicherheit bieten für ein hochdigitalisiertes Produktionsparadigma bieten.

Laut **John Lincoln**, **Geschäftsführer der UK Photonics Leadership Group**, wird auch die Nachhaltigkeit eine Rolle bei der Einführung spielen. Es gibt einen globalen Zielkonflikt zwischen den langfristigen Umweltplänen der Regierungen und den viel kurzfristigeren Kosten für die Unternehmen. Es ist zu erwarten, dass sich die Notwendigkeit, diese beiden Aspekte in Einklang zu bringen, ziemlich plötzlich ändert.



(Image credit: Stephen Alvey, Michigan Engineering)



Stetig fortschreitende Fortschritte in der Elektronik und Informatik haben Möglichkeiten geschaffen, Ziele zu erreichen, die früher undenkbar schienen: autonome Maschinen zu bauen, komplexe **Deep-Learning-Probleme** zu lösen und sofort über alle Orte des Planeten zu kommunizieren.

Bedauerlicherweise sind wir bei all den Fortschritten noch nicht an dem Punkt, an dem wir sein sollten. Unsere Systeme, die sich immer noch auf diese altmodischen elektronischen Prozessoren verlassen, sind in einer sehr unangenehmen Realität verankert:

Die schiere Physik der Elektronen begrenzt ihre Bandbreite und zwingt sie, enorme Wärme zu produzieren, was bedeutet, dass sie Unmengen an Energie verbrauchen. Das Internet verbraucht heute etwa 17% der weltweiten Energieressourcen.

Da die Nachfrage nach schneller und energiesparender Künstlicher Intelligenz (KI) steigt, suchen Forscher nach Möglichkeiten, über Elektronen hinaus in die Welt der Photonen vorzustoßen. Sie ersetzen elektronische Prozessoren durch photonische Designs, die Laser und andere Lichtkomponenten einbeziehen. Einige Beobachter sind zwar skeptisch, dass die Technologie die analoge Datenverarbeitung verändern kann, bauen Forscher im optischen Bereich jetzt Systeme, die erhebliche Vorteile bei der **KI** und beim **Deep Learning** zeigen.



THE DELPHI COMPANY

-est 1998-

OS X Android iOS Windows



Four Platforms
One develop environment
One Expertise

Vier platforms
Eén ontwikkelomgeving
Eén expertise

DELPHI

www.delphicompany.nl
info@delphicompany.nl

INTERESSIERT DARAN
ZU LERNEN WIE MAN
PAS2JS BENUTZT
UND PASCAL FÜR
IHRE INTERNET
ANWENDUNG?

PA 2 JS



send your reservation to
editor@blaisepascalmagazine.eu



DONATE FOR UKRAINE AND GET A FREE LICENSE AT:

<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>
(Just click)



Wenn Sie ukrainischer Herkunft sind, können Sie ein kostenloses Abonnement für das Blaise Pascal Magazin erhalten,

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.

Senden Sie es bitte an editor@blaisepascal.eu und Sie erhalten Ihr Buch und Ihr Abonnement kostenlos.

BLAISE PASCAL  MAGAZINE



Blaise Pascal



DONATE FOR UKRAINE AND GET A FREE LICENSE AT:
<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>
(Just click)

KBMMW PROFESSIONAL AND ENTERPRISE EDITION V. 5.20.01 JUST NOW RELEASED!

This is a significant new release with new high performance transports, **OpenSSL v3 support**, WebSocket support, further improvements to SmartBind, new high performance hashing algorithms, improved RemoteDesktop sample and much more.

This release requires the use of **kbmMemTable v. 7.96.00** or newer.

- **RAD Studio XE5 to 11 Alexandria supported**
- Win32, Win64, Linux64, Android, IOS 32, IOS 64 and OSX client and server support
- Native high performance 100% developer defined application server
- Full support for centralized and distributed load balancing and failover
- Advanced ORM/OPF support including support of existing databases
- Advanced logging support
- Advanced configuration framework
- Advanced scheduling support for easy access to multithread programming
- Advanced smart service and clients for very easy publication of functionality
- High quality random functions.
- High quality pronounceable password generators.
- High performance LZ4 and Jpeg compression
- Complete object notation framework including full support for YAML, BSON, Messagepack, JSON and XML
- Advanced object and value marshalling to and from YAML, BSON, Messagepack, JSON and XML
- High performance native TCP transport support
- High performance HTTPSys transport for Windows.
- CORS support in REST/HTML services.
- Native PHP, Java, OCX, ANSI C, C#, Apache Flex client support!
- **NEW: FULL WEBSOCKET SUPPORT.**
The next release of kbmMW Enterprise Edition will include several new things and improvements. One of them is full WebSocket support.
- New I18N context sensitive internationalization framework to make your applications multilingual.
- New ORM LINQ support for Delete and Update.
- Comments support in YAML.
- New StreamSec TLS v4 support (by StreamSec)
- Many other feature improvements and fixes.

kbmMemTable is the fastest and most feature rich in memory table for Embarcadero products.

- **Easily supports large datasets with millions of records**
- **Easy data streaming support**
- **Optional to use native SQL engine**
- **Supports nested transactions and undo**
- **Native and fast build in M/D, aggregation/grouping, range selection features**
- **Advanced indexing features for extreme performance**
- High speed, unified database access (35+ supported database APIs) with connection pooling, metadata and data caching on all tiers
- Multi head access to the application server, via REST/AJAX, native binary, Publish/Subscribe, SOAP, XML, RTMP from web browsers, embedded devices, linked application servers, PCs, mobile devices, Java systems and many more clients
- Complete support for hosting FastCGI based applications (PHP/Ruby/Perl/Python typically)
- Native complete AMQP 0.91 support (Advanced Message Queuing Protocol)
- Complete end 2 end secure brandable Remote Desktop with near realtime HD video, 8 monitor support, texture detection, compression and clipboard sharing.
- Bundling kbmMemTable Professional which is the fastest and most feature rich in memory table for Embarcadero products.

 **COMPONENTS
DEVELOPERS 4**

