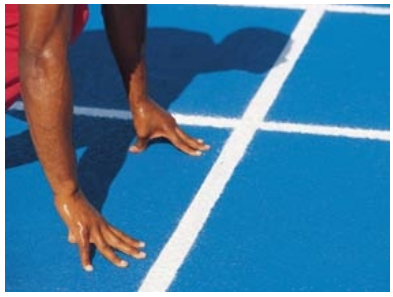
maXbox



QR Codes

1.1 QR with a lot of Things

Today we jump into a topic in programming called the Quick Response Code.
Using Object Pascal there are several ways you can generate QR codes, from a web service up to a component - to encode any text (URL, web store, phone number, short message). QR Codes store up to 4,296 alphanumeric characters of arbitrary text.
I'll show it in 3 ways:

- WinInet API
- Indy Sockets
- Direct Library

Wiki says. A QR code (abbreviated from Quick Response) is the trademark for a type of matrix bar-code (or two-dimensional bar-code) first designed for the automotive industry in Japan developed by Denso-Wave.
The point is that QR codes give users or customers instant access to the information they need on the move; whether its on your website, social media or another news channel a QR service is dedicated to making access to it quicker and easier than ever before.
We can put many information in a QR code; the code contains information such as a website address, telephone number, or a business card information.

There are two different types of QR code; "Static" and "Dynamic".

A static QR code holds all of the information you wish to pass within the code itself. The more information you include, the larger the QR code becomes.
Those codes are widely used in the web of things or embedded computing with the internet;
There are three topics in here. First technologies – simply put, this part is mainly for early adopters. It's about coding with QR code, developing toys, plugging in gadgets on the web (and we and many others actually did that!).
The second part is about new ideas, prototyping and new technologies that are in the lab. It's about research papers, and software philosophy, and about researchers worldwide. Third part is about end-users and products, and those products need a QR code or an RFID e.g.:

- Magazines and Gadgets
- 3D Printing Production
- Web Video Cam Pictures

1.2 QR Code Mode

Now we deal with QR code and put a real script first:

The point is that QR codes give users or customers instant access to the information they need on the move; whether its on your website, ask.com or another news channel a QR service is dedicated to making access to it quicker and easier than ever before.
The script you find at the examples:

```
393_QRCode3.TXT
393_QRCode2Direct_Detlef.TXT
```

But what's the matter with Web of Things? Simply that each thing, device or gadget will have an identity to communicate, could also be a RFID.
Statistics for who's scanning QR codes and with what device appear to be mixed,

The call is that simple:

```
GetQrCode3(250,250,'Q',QDATA, ExePath+AFILENAME);
OpenDoc(ExePath+AFILENAME);
```

This is so simple because google provides a service we use:

```
UrlGoogleQrCode=
          'http://chart.apis.google.com/chart?chs=%dx%d&cht=qr&chld=%s&chl=%s';
```

Using the Google Chart Tools / Image Charts (aka Chart API) you can easily generate QR codes, this kind of images are a special type of two-dimensional bar-codes. They are also known as hard-links or physical world hyper-links.
The API requires 4 simple fields be posted to it:

1. cht=qr this tells Google to create a QR code;
2. chld=M the error correction level of the QR code (see here for more info);
3. chs=wxh the width and height of the image to return (e.g. chs=250x250);
4. chl=text the URL encoded text to be inserted into the bar-code.

For example the Blaise Pascal URL we need the following:

```
Const

   URLGoogleQRCODE='http://chart.apis.google.com/chart?chs=%dx
%d&cht=qr&chld=%s&chl=%s';
   AFILENAME= 'mX3QRCode3.png';
   //QDATA= 'this is maXland on a stream dream firebox';
   QDATA= 'http://www.blaisepascal.eu/';

Type TQrImage_ErrCorrLevel=(L,M,Q,H);
```

There are 4 ECC Levels in QR Code (enumerator) as follows:

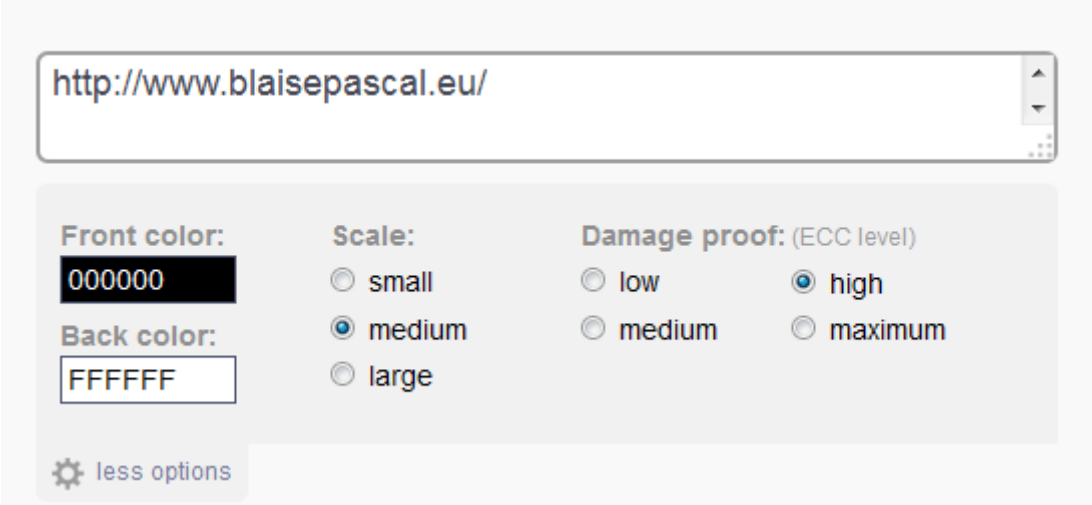
Level L - 7% of codewords can be restored
Level M - 15% of codewords can be restored
Level Q - 25% of codewords can be restored
Level H - 30% of codewords can be restored

So what does it mean? 7% of codewords can be restored just means that you can loose about 7% of the QR code and the built in redundancy will be able to restore the missing data. (i.e if you rip the QR code out of a paper advert and tear off a small bit of the QR code then you might still be able to scan it).
This also allows small images to be overlaid without affecting the scan.
A high ECC level adds more redundancy at the cost of using more space!

Whit that information you can type in maXbox:

```
//Internal mXLib
procedure getQRCodeDirect;
begin
  GetQrCode3(250,250,'Q',QDATA, ExePath+AFILENAME);
  OpenDoc(ExePath+AFILENAME);
end;
```

In a website those options ECC Level Q and resolution 250 * 250 stands for:



As I mentioned before static code, the advantage of dynamically coded is the update.
Static QR codes are created once and can not be changed. So if you were to create a QR code for your website and then change your website address, you would need to reprint all of your literature to keep it updated.

The maximum capacity for a standard QR code (according to Wikipedia) is 2,953 bytes or 23,624 bits. Each bit has two states, so the number of possible permutations is 2^23624 or about 3.4*10^7111.
More than the atoms in the universe (estimated to be 10x10x10... 81 times).

OK. don't fumble around, practice: http://www.qrstuff.com/

For now, data storage has a finite density and is limited by the size of the medium. The number of possible permutations will increase with the size or density of the QR code, so a (non-standard) infinitely large or infinitely dense QR code would have an infinite amount of permutations.
There is nothing magical about QR Codes. It is just machine speak and the storage dictates how much you can write. So how many different numbers can you write if you have a paper grid with 7089 squares ?

The 2D bar-code VCL components is a set of components designed for generating and printing bar-code symbols in your Delphi or C++ Builder applications. Use the components set like any other VCL components. J4L Components includes the QR-code implementation featuring: auto, byte, alpha, numeric and kanji encoding.

 Storing up to 4296 characters they are internationally standardised under ISO 18004, so a QR code is a QR code all over the world - they've been big in Japan forever, broke into Europe and the UK a few years back, and are now getting real traction in USA and China.

Think "print-based hypertext links" and you'll get the idea.

1.2.1 Behind the Code

As you already know the tool is split up into the toolbar across the top, the editor or code part in the centre and the output window at the bottom. Change that in the menu `/view` at our own style.

 Before this starter code will work you will need to download maXbox from the website. It can be down-loaded from http://www.softwareschule.ch/maxbox.htm (you'll find the download maxbox3.zip on the top left of the page). Once the download has finished, unzip the file, making sure that you preserve the folder structure as it is. If you double-click `maXbox3.exe` the box opens a default demo program. Test it with F9 / F2 or press **Compile** and you should hear a sound. So far so good now we'll open the examples:

`393_QRCode3.TXT`

If you can't find the two files try also direct as a file

http://www.softwareschule.ch/examples/393_QRCode3.TXT

Now let's take a look at the code of this project. Our first line is creating the object in line 53 we use the first methods to configure our Indy HTTP calling Port and IP with the help of `HTTPEncode`. The object makes a bind connection with the `Active` method by passing an encoded URL within a memory stream:

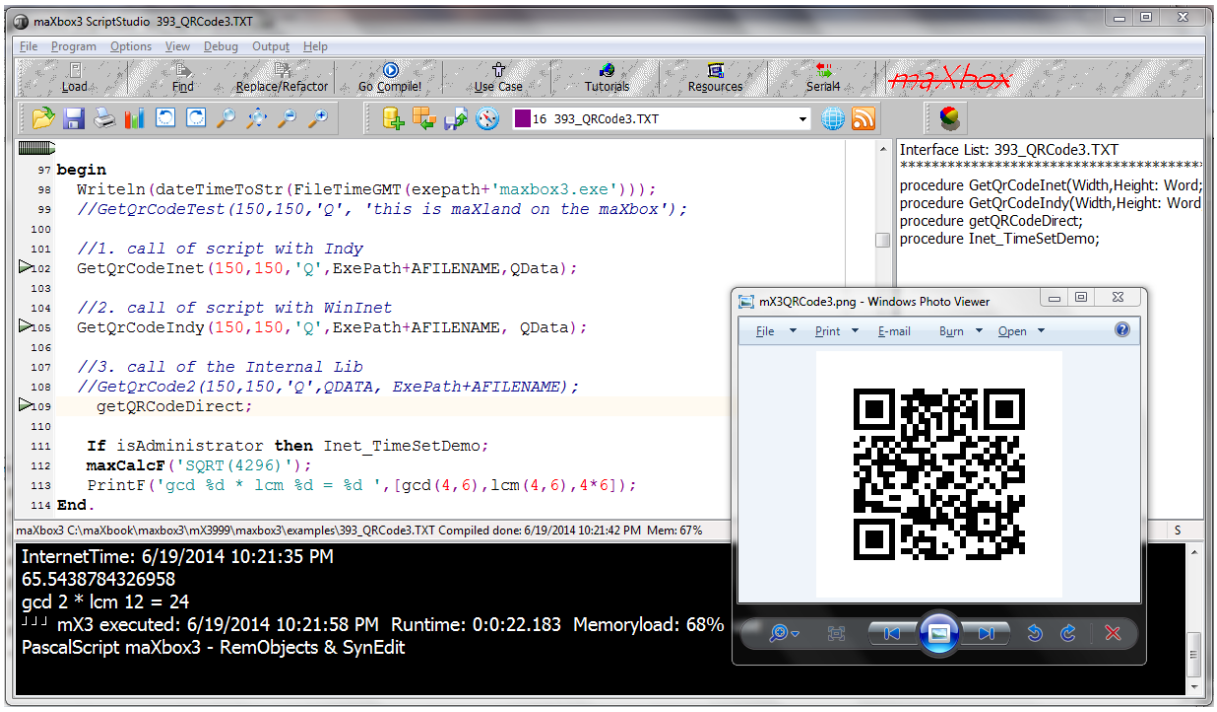
```
56 idHTTP.Get1(encodURL, pngStream);
```

Then I use the `TLinearBitmap` class to create a PNG format.
So the object `idHTTP` has some methods and properties like `Get` you can find in the `TIdCustomHTTP.pas` unit or `IdHTTPServer` library. A library is a collection of code or classes, which you can include in your program. Much more stuff or intricacies and details of the TCP/IP stack are hidden from the Indy programmer. A typical Indy client session looks like this:

```
with IndyClient do begin
  Host:= 'zip.pbe.com'; // Host call
  Port:= 6000; // Port to call the server on
  Connect; // get something to do
end;
```

Indy is different than other so called Winsock components you may be familiar with. Nearly all other components use non-blocking (asynchronous) calls and act asynchronously. They require you to respond to events, set up state machines, and often perform wait loops.

☝In facts there are 2 programming models used in TCP/IP applications. Non blocking means that the application will not be blocked when the app socket read/write data. This is efficient, because your app don't have to wait for connections. Unfortunately, it is complicated.



2: The Use Case of the QR App

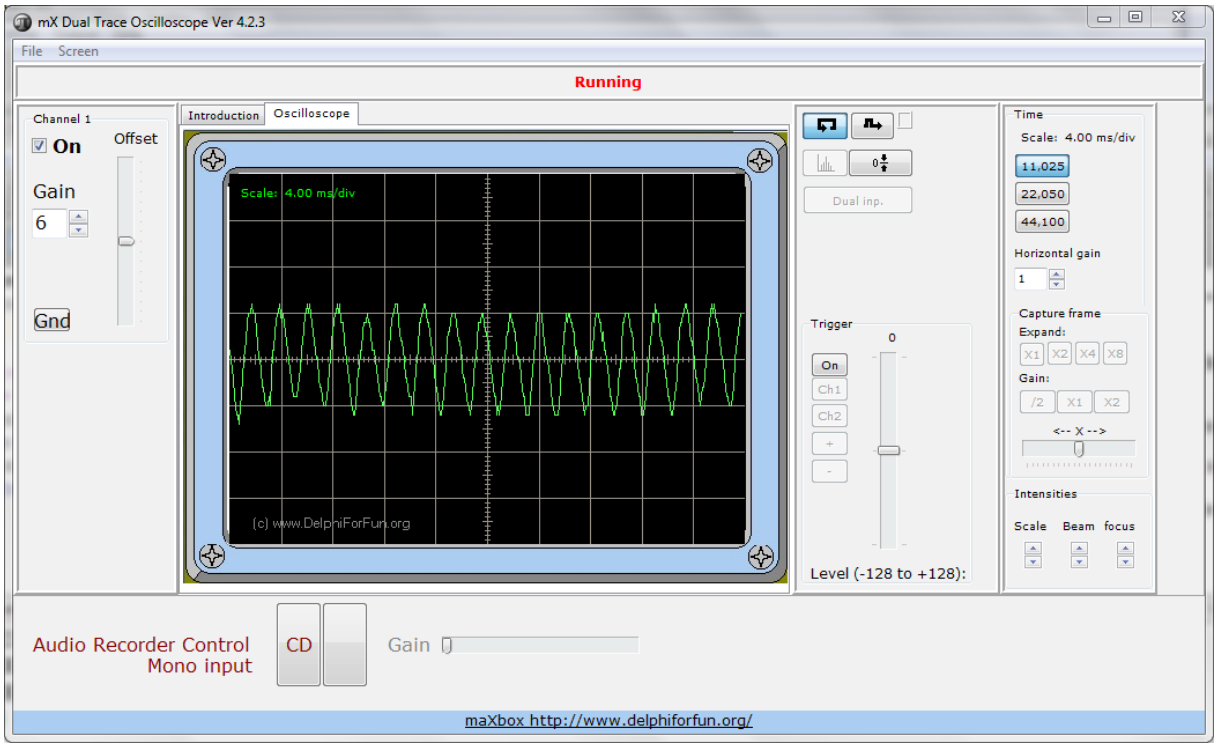
Let's get back to our Create in line 53. In line 10 you see the magic of the Google service link and File-name plus Data configuration of a `const` in line 11 and 12. For the geeks among us: The implementation of the QRCode Plugin was very simple thanks to a well documented API.

```
54 //Indy Socks

procedure GetQrCodeIndy(Width,Height: Word; C_Level,apath: string; const
Data: string);
var
  encodURL: string;
  idhttp: TIdHttp;// THTTPSend;
  pngStream: TMemoryStream;
```

```
begin
  encodURL:= Format(URLGoogleQRCODE,[Width,Height,C_Level,
                                HTTPEncode(Data)]);
  idHTTP:= TIdHTTP.Create(NIL)
  pngStream:= TMemoryStream.create;
  with TLinearBitmap.Create do try
    idHTTP.Get1(encodURL, pngStream)
    pngStream.Position:= 0;
    LoadFromStream2(pngStream,'PNG');
    //aImage.Picture:= NIL; //AssignTo(aimage.picture.bitmap);
    SaveToFile(apath);
    OpenDoc(apath);
  finally
    Dispose;
    Free;
    idHTTP.Free
    pngStream.Free;
  end;
end;
```

☞ Regular text will not be interpreted by most QR readers, so it will just be displayed as it is.
Have you ever wanted to see a visual representation of changes in sonic frequency or amplitude of a QR code? First up is a oscilloscope. This is the classic view of a wave flowing horizontally that shows how sound varies in 0 and 1 (black and white) of a QR code up to 4296 bytes. (Or how voltage varies in a wire). In maXbox you find the instrument by Options.



With this oscilloscope you can measure 0 or 1 of a QR code in /Options/..
This examples shows black as 220 Hz and white as 440Hz. This is because when you double octave, you are doubling cycles per second (e.g. A 440hz is one octave higher than A 220hz).

The following shows the magic behind the method `HttpGet()` or `HTTPServerGet()`:

```
procedure HTTPServerGet(aThr: TIdPeerThread; reqInf: TIdHTTPRequestInfo;
                                                 respInf: TIdHTTPResponseInfo);
```

One word concerning the thread: In the internal architecture there are 2 threads categories. First is a listener thread that “listens” and waits for a connection. So we don't have to worry about threads, the built in thread `TIdPeerThread` will be served by Indy through a parameter:

```
54 //WinInet

procedure GetQrCodeInet(Width,Height: Word; C_Level,apath: string; const
Data: string);
var
  encodURL: string;
  pngStream: TMemoryStream;
begin
  encodURL:= Format(URLGoogleQRCODE,[Width,Height, C_Level,
                          HTTPEncode(Data)]);
  pngStream:= TMemoryStream.create;
  HttpGet(encodURL, pngStream);    //WinInet
  with TLinearBitmap.Create do try
    pngStream.Position:= 0;
    LoadFromStream2(pngStream,'PNG');
    SaveToFile(apath);
    OpenDoc(apath);
  finally
    Dispose;
    Free;
    pngStream.Free;
  end;
end;
```

HTTP request messages contain many headers that describe information about the client, the target of the request, the way the request should be handled, and any content sent with the request. Each header is identified by a name, such as "Host" followed by a string value.
A last piece of code is one of my favour, get the right time!
Another way to check syntax before run is F2 or the **Syntax Check** in menu `Program`.

```
procedure Inet_TimeSetDemo; //run it in Admin mode to set sys time!

begin
  with TIdSNTP.Create(self) do
  try
    Host:= '0.debian.pool.ntp.org'
    writeln('InternetTime: '+datetimetoStr(datetime));
    if Synctime then
      writeln('Op System Time sync now!');
  finally
    Speak('Your SysTime is now sync with Inet Time '
           +TimeToStr(datetime))
    Free;
  end
end;
```

The Object `TIdSNTP` is a dynamically allocated block of memory whose structure is determined by its class type. Each object has a unique copy of every field defined in the class, but all instances of a class share the same methods. With the method `Get1` you can download files.

```
11 begin
12  myURL:= 'http://www.softwareschule.ch/download/maxbox_examples.zip';
13  zipStream:= TFileStream.Create('myexamples2.zip', fmCreate)
14  idHTTP:= TIdHTTP.Create(NIL)
15  try
16    idHTTP.Get1(myURL, zipStream)
```

Of course a lot of lines to get a file from the web; try it shorter with the function `wGet()`:

```
wGet('http://www.softwareschule.ch/download/maxbox_starter17.pdf','mytestpdf.pdf');
```

It downloads the entire file into memory if the data is compressed (Indy does not support streaming decompression for HTTP yet). Next we come closer to the main event of our web server, it's the event `onCommandGet` with the corresponding event handler method `@HTTPServerGet()` and one object of `TIdPeerThread`. You can use them as server to serve files of many kinds!
A full description of a socket connection includes the addresses of the sockets on both ends of the connection. You can describe the address of each socket endpoint by supplying both the IP address or host and the port number.

Many of the protocols that control activity on the Internet are defined in Request for Comment (RFC) documents that are created, updated, and maintained by the Internet Engineering Task Force (IETF), the protocol engineering and development arm of the Internet. There are several important RFCs that you will find useful when writing Internet applications:

- RFC822, "Standard for the format of ARPA Internet text messages," describes the structure and content of message headers.
- RFC1521, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for a format of Internet Message Bodies," describes method used to encapsulate and transport multipart and multiformat messages.
- RFC1945, "Hypertext Transfer Protocol—HTTP/1.0," describes a transfer mechanism used to distribute collaborative hypermedia documents.

In the last line we just start a browser to test our QR Code API work flow ☺

```
34 procedure letOpenBrowser;
35 // TS_ShellExecuteCmd = (seCmdOpen,seCmdPrint,seCmdExplore);
36 begin
37 //ShellAPI.ShellExecute(Handle,PChar('open'),'http://127.0.0.1:80/',Nil,Nil,0);
38   S_ShellExecute('http:'+IPADDR+':'+IntToStr(APORT)+'/','',seCmdOpen)
39 end;
```

1.3 QR Code Business

So how can you take advantage of this growing trend for your daily business? Keep in mind that much more than web addresses can be scanned. For example, you could:

1. Direct employers to your LinkedIn profile or your Vcard

2. Use QR code in a direct mail piece, or postcard to provide a discount
3. Take them to a page with detailed info that wouldn't easily fit in a print ad
4. Use them to deliver step-by-step instructional videos or a printable setup sheet
5. Let them register in an event such as a course or consultation
6. Use QR code to let customers send themselves a reminder via SMS
7. Link them to a special event video

Some notes about firewalls or proxy-servers when using this QR code service. It depends on your network infrastructure to get a file or not, maybe you can't download content cause of security reasons and it stops with Socket-Error # 10060 and a time out error.
Furthermore, it also depends on the firewall in use at both ends. If it's automatic and recognises data that needs a response automatically it will work. It needs an administrator to open ports etc. you're stuffed or configured. A firewall that only allows connections to the listening port will also prevent the remote debugger from working. But after all HTTP lets you create clients that can communicate with an application server that is protected by a firewall.

Feedback @
max@kleiner.com

Literature:

Kleiner et al., Patterns konkret, 2003, Software & Support
Links of maXbox, Web of Things and Indy:

http://www.softwareschule.ch/download/webofthings2013.pdf

http://www.softwareschule.ch/maxbox.htm
http://www.indyproject.org/Sockets/index.EN.aspx

http://sourceforge.net/projects/maxbox
http://sourceforge.net/apps/mediawiki/maxbox/
http://www.blaisepascal.eu/subscribers/vogelaar_elctronics_info_english.php