Calculate IBAN numbers from a Belgian bank account number

By monokid

My previous post was about SEPA. I went on a quest to figure out, how exactly I could convert an old, national bank account number, into an IBAN number. Not an easy task, I tell you.

You see, google dropkicks you in the face with a bazillion search resultsfor online converter tools, offered by financial institutions, but there's very little clear documentation that tells you (in layman terms), how to convert it, manually.

Until I came across this PDF:

http://www.sepabelgium.be/files/Standard-IBAN-BIC-BBAN-v3-NL.pdf(Dutch)

I want to know, how to convert an old number, the old-fashioned way. Nitty gritty, step-by-step, calculations. For clarity's sake, I'll run you through a conversion. *Do note, this is about BELGIAN bank account numbers, only, I'm afraid.*

A little heads-up: I'll be talking about a visual or an electronic format. When I first heard about this, I had no idea what was going on, but it's very simple to explain: The electronic format is a format which is the easiest to process for a logical algorithm, ie.: your computer. These formats are just syntaces which have no extra characters like dashes, underscores, spaces, etc. A visual format is a format you'll most likely encounter on an invoice or a bank statement. It's a number which is formated in a way, so it is easier to read and comprehend by us imperfect human beings. These DO have dashes, underscores, spaces.

Eg.:

Electronic format: 510007547061
Visual format: 510-0075470-6 1

What do IBAN numbers look like?

Universally:
[ISO COUNTRY CODE] [2 CHECK DIGITS ##] [BANK ACCOUNT NUMBER UP TO 30 digits]

For Belgium:
BE [2 CHECK DIGITS ##] [BANK ACCOUNT NUMBER 12 DIGITS]

Eg.: BE 62 510 0075470 61 (visual format)

What is needed?

Technically speaking, which parameters are needed?

- Old bank account number (visual or electronic format)
- ISO country code

That's it! If you look at the IBAN format for Belgian bank account numbers, all we need to figure out, is the check digit. This is what we're going to calculate, through an algorithm.

Algorithm

Basically, there's two approaches. One approach performs the conversion through the "classical" method. The second is a neat and simple workaround, which is faster and easier.

Classical approach:

1. Create a dummy IBAN number, by appending the country code and the check digits. Since the check digits are currently unknown, just use two zero's. Append these in front of the old bank account number. Remove all characters other than letters and numbers

2. Move the first four digits (country code and two check digits) to the back of the account number.

3. Convert the country code letters, into numbers, through the Roman alphabet system (<>Roman numberals!!!), where A=10, B=11, C=12. For "BE", "B" = 11 and "E" = 14. You should now have a string of just numbers.

4. Perform a Modulus 97 on the string of numbers. The MOD97 should return the remainder of the derivation.

5. Subtract 98 with the remainder of the MOD97 operation. This is the IBAN check digit. If the check returns an single digit, insert a leading "0".

6. Reassamble the IBAN number:
[BE] [CALCULATED CHECK DIGITS FROM STEP 5] [###-########-##]

OH SHI-!

Try this in Excel. Go on. I dare you. That's right. If you try to convert the number, using the modulus97 operation, Excel will crap out and return a #NUM! error. I'll save you the googling, Excel cannot handle modulus operation on large numbers, like bank account numbers. Apparently, Microsoft has made note of this issue, but from what I can tell, they aren't going to do anything about it.

FFFFFFUUUUUUUUUUUU

Have no fear! Lazy developer is here! There's an easier and quicker method to calculate the check digit:

Quick 'n dirty:

1. Retrieve the old bank acount number check digits (last two digits).

2. Convert the ISO country code "BE" to numbers, using the Roman alphabet system – which is 11 and 14 for "BE"

3. Create a dummy string of numbers like this: [Check digits] [Check digits] [Converted country code numbers] [two zero's]

4. Perform the operation:
98 – modulus97(dummy string)

5. The result of the previous operation is the check digit for IBAN. Reassemble the IBAN bank account number:
[BE] [CALCULATED CHECK DIGITS FROM STEP 5] [###-########-##]

Why is this easier? The dummy string of numbers that needs to be calculated through the modulus 97 is much smaller and easier to handle through code. Also, for Belgian bank account numbers, the converted country code will always be 1114. From a developers point of view, the raw pseudo code is about as follows:

1. Drop all non-alphanumerical characters from a bank account number
2. Retrieve the last two digits (old bank account number check digit)
3. 98-(mod97(####111400) where ## stands for the check digits, which need to be appended twice.
4. create IBAN number by appending country code, the calculated check digits from step 3 and the the old bank account number, in that order

To clarify with an example, using the quick 'n dirty method:

Old bank account number:
510-0075470-61

1. Remove all non-alphanumerical characters:
510007547061

2. Retrieve last two digits:
61

3. Create "dummy" by using the check digits, twice and converting BE into numbers and appending two zero's:
61 61 11 14 00

4. 98 – mod97(dummy) operation:
98-(mod97(6161111400) = 62

5. Assemble the IBAN number by appending BE + check digit + bank account number:
BE 62 510 0075470 61

Needless to say, the quick method is only applicable on Belgian bank account numbers!

Resources and tools

Like I said earlier in this post, there's an entire barrage of tools and calculators, converters and whatnot, available on the net, for free.

I'll provide links to a few which are most interesting for business consultants and developers:

- ISABEL (offline) conversion tool:

ISABEL is a (great) payment platform that helps enterprises manage their transactions through multiple banks. They created a free conversion tool, which can convert old formats to the new format one-by-one, or in batch. You'll need to feed it an exported file, from your software, specify a few parameters and it'll process it back into an .XML or .TXT file. Pretty nifty tool. Ignore the update error when you start the tool.

- International online IBAN converter:

This website lets you input the country code, bank account number and requires a quick CAPTCHA. It will find the BIC code and IBAN bank account number for you.

- IBAN/BIC conversion and check webservices

Probably the most interesting tool for developers, by IBANIC.BE. These are web services which can be addressed from customized software environments. The web service needs to be invoked with a single or multiple parameters. It can return a whole lot of useful things:

-
    - retrieve bank name from old bank account number
    - return BIC code from old bank account number
    - return IBAN from old bank account number
    - Validation check of old bank account number
    - etc.

The advantage of using this webservice, is that no calculations needs to be done by the software and the returned data is always up to date. Whereas, should you want to perform the calculation client-side, you'll need to perform the operation on the client and you'll need to keep a table of all BIC codes up to date – locally (while these codes can change on a daily basis).

The downside, is that your client software is dependent on the reliability of the uptime of this webservice.