OneClickToQIF

© OneClickToQif, February 2016.

Licence: GNU GPL v3

Contents

Introduction

OneClickToQIF is a project consisting of a series of spreadsheet templates and macros, which are used to export data (such as bank accounts, mortgage ...) to QIF format, used by programs such as *GnuCash, Money* and *Quicken.*

With *OneClickToQIF* you can export, in just one click, **all the tabs in your spreadsheet**, both for simple and split transactions.

You can use the templates provided, or easily **adapt your own existing spreadsheet**, without any knowledge of macros.

Every tab is automatically exported from the spreadsheet to QIF files, both assets (bank accounts, cash ...) and liabilities (credit cards, mortgage ...).

You **do not need to indicate each time the columns containing data,** or the transactions to be exported: just label the columns the first time you use the template. Only new movements (those who have never been exported) will be exported to the QIF file.

License

OneClickToQIF is licensed GNU GPL v3: You can use the software free of charge study, share and modify.

https://github.com/OneClickToQif/OneClickToQif/blob/master/LICENSE

How to get help

Please first check the documentation on the project wiki:

https://github.com/OneClickToQif/OneClickToQif/wiki

If you do not find there what you need, you can send an email directly to the author:

oneclicktoqif@gmail.com

If you have requests or suggestions to improve the project, you can apply at:

https://github.com/OneClickToQif/OneClickToQif/issues

Typically only generic requests that improve the functionality of the project will be addressed. If you have specific needs, you can request at the email above.

About the project

Why did you create this project? Why should I use it?

I created **OneClickToQIF** for my own personal use.

I have used GnuCash for a long time. It is a double entry accounting program with many features:

https://www.gnucash.org/features.phtml

However, the GnuCash interface is not the best way to enter the details of transactions for daily use.

By contrast, spreadsheets provide a faster and simpler interface. It is much easier to enter data via MS Excel or Calc than via GnuCash interface or similar applications.

In addition, most banks and credit card issuers provide account statements in spreadsheet format. I usually download the transactions of my bank accounts and credit cards from the web of banks and financial institutions, in Excel format. I find it faster and more manageable to view these transactions in the spreadsheet. However, I want to have them synchronized in GnuCash.

Also for accounts with split transactions, I find it much easier to enter data via spreadsheets, than via GnuCash interface.

On the other hand, I do not want to give up all the benefits of GnuCash, especially to manage on one site all accounts for assets, liabilities, income and expenses, and generate reports from them.

With **OneClickToQIF** is very quick and easy to move data from spreadsheets to GnuCash or other bookkeeping application, using the former for insertion and some calculations, and the latter for all the functionality of a double-entry bookkeeping application.

Last version of the documentation

The last version of the documention is usually found in the English wiki at:

https://github.com/OneClickToQif/OneClickToQif/wiki

The last version in PDF format is found at:

https://github.com/OneClickToQif/OneClickToQif/tree/master/docs/english

How to join the project

The intention is ***OneClickToQIF*** to be a collaborative project where any person may contribute in the different ways:

- **Comments / Suggestions,** through:
  - Email: oneclicktoqif@gmail.com
  - URL: https://github.com/OneClickToQif/OneClickToQif/issues

    You can make comments that you think appropriate (please be positive) both on templates, macros, documentation, etc. This will help me to know how to improve the project.

    Please read before existing documentation and functionality requested in the previous URL.

- **Development**: If you want to make improvements to the templates or macros, or add new features, you can do it through the project:
  https://github.com/OneClickToQif/OneClickToQif

- **Other spreadsheet fomats:** the present project works for MS Excel. The next step is to adapt these templates / samples/ macros to ***LibreOffice / OpenOffice Calc***. We do not have a scheduled date for this. If you want to adapt them you are very welcome!

- **Adaptations / Translations**: if you speak another language and want to adapt a template or documentation to your language, please contact us. Also if you want to improve the present documentation: e.g. correct grammar or spelling, please let us know.

- **Questions:** we will try to answer your questions, see the "How to get Help" chapter This will also help us to know which aspects are the most complicated and likely to improve

- **Star the project** in GitHub

The project will only be updated if it creates interest, which we will see through any of the points above: comments, questions, contributions to the development, adaptations...

Alternatives to *OneClickToQIF*

GnuCash documentation includes several ways to export from Excel to QIF:

http://wiki.gnucash.org/wiki/FAQ#Q:_How_do_I_convert_from_CSV.2C_TSV.2C_XLS_.28Excel.29.2C_or_SXC_.28OpenOffice.org_Calc.29_to_a_QIF.3F

However I do not find practical neither of them, for two reasons:

1. Each time you export a spreadsheet, these alternative methods require entering various data and make several clicks.

   For example, the macros at "xl2qif.chez-alice.fr" need for each export to select the columns to be exported (and which ones not to be), and click on them in the proper order. This is error-prone and takes time to make that choice.

   Let's say you have to export 6 spreadsheet tabs (4 for bank accounts and 2 for credit cards) once a week. So each week, you would have to enter each of the tabs and repeat that process 6 times.

Other methods including pre-export to CSV format are even worse.

With *OneClickToQIF*, this process is done in a single click for the 6 tabs.

2. In these alternative methods, there is no way to export spreadsheets with split transactions.

   Therefore, I can not export my payroll spreadsheet to QIF format, or other spreadsheets containing split transactions.

   With *OneClickToQIF*, this process is also carried out in a single click

Furthermore, *OneClickToQIF* is Open Source and released under the GNU GPL license, so you can view the code, modify it if you consider it necessary, and have the support of many other contributors that will improve the functionality and solve potential issues.

Using the spreadsheet templates

Where do I start?

If you want to quickly test the functionality of **OneClickToQIF**, the easiest way is by checking the following example spreadsheet:

- Download "accounts.xlsx". Open it. In the "control" tab:
  - Click the "Select folder" button. Select the folder where the QIF files will be exported.
  - Click the "Export" button. The data files will be generated. The transactions in every data tabs will be marked as "Y" in the "Exported" columns.

The project has another spreadsheet (and another set of macros) for *split transactions*. If you do not know what "split transactions" mean, you might not need them. Such transactions are carry amounts to different (more than two) accounts, such as payrolls. Gnucash documentation explains this type of transactions:

http://www.gnucash.org/docs/v2.6/C/gnucash-guide/txns-registers-txntypes.html

If you need split transactions, follow these steps:

- Download "payrolls.xlsx".
  - Click the "Select folder" button. Select the folder where the QIF files will be exported.
  - Click the "Export" button. The data files will be generated. The transactions in the spreadsheet will be marked as "Y" in the "Exported" column.

Note: The payroll sample in this project is made from the point of view of accounting of an employee. The example in the GNUCash documentation is made from the point of view of a company accounts to pay salaries.

How to download bank account transactions and export them to QIF

1. Download the transactions in spreadsheet format, as you normally would
2. Paste them int the "Area for bank data " of the "checkin_accout"
3. Mark the rows for downloaded transactions to "N" in the "(Exported)" column
4. Go to the Control tab and click "Export"

What are the "auxiliary formulas" columns?

These columns are optional, and are used to convert some data that are not directly exportable as downloaded from the website of your bank or financial institution.

For instance:

- Sometimes when downloading dates, they are in "text" format. Auxiliary column "value" formula is used to interpret it as a date, so that it can be exported.
- Credit card transactions may be exporting positive value, when it should be negative, or vice versa. The "(Amount)" column can invert the value provided by the financial institution
- In the "(Memo)" column you can generate a comment from several downloaded columns, linking them to your own comment, and then export all as a single comment

How can I re-export a set of transactions?

You just manually mark them with 'N' in the "Exported" column.

Adapting your own spreadsheet to export data

If you already have a spreadsheet and you want to export it to QIF, you have two options:

a) Move the data to the sample spreadsheet templates provided, by copying and pasting in the appropriate columns.
b) Adapt your spreadsheet to use the project macros. This section details this option

You only need to do this once; then you can export your data to QIF with just one click from your spreadsheet each time you insert new transactions.

Note: You do not need any knowledge of macros to adapt your own spreadsheet. However, you need a minimum knowledge of Excel. If you need specific help adapting a spreadsheet, refer to the "How to get help" section.

First, you must determine if what you have is a spreadsheet with single or split transactions.

Simple vs. Split Transactions

If you do not know what split transactions are, you likely have simple transactions. Go to the section for simple transactions. If you want to be sure, continue reading.

Every transaction has at least two splits, but a transaction can have more than two splits.

A transaction with only two splits is called a simple transaction, since it only involves the current account and a single remote account. An extract from the bank account or credit card uses simple transactions, because it only involves the (bank or credit card) account and the remote account to which the money goes.

A transaction with three or more accounts is called split transaction. Such transactions are used for movements carrying amounts to different accounts, such as payrolls.

Gnucash documentation explanins these types of transactions here:

http://www.gnucash.org/docs/v2.6/C/gnucash-guide/txns-registers-txntypes.html

How to adapt a spreadsheet with simple transactions

Insert a new row under the header, which should be row # 2 (*). Tag this row of the spreadsheet with the following labels (without quotes):

- "*(Memo)*" column: el comentario del movimiento
- "*(Category)*" column: the category. In Gnucash this corresponds to the target account. Based on this value, the transaction will be imported into the appropriate account. In the sample templates this field is validated from a list.
- "*(Exported)*" column: this field indicates whether the transactions have previously been exported. Only transactions with "N" value will be exported. The export macro will automatically set the value to "Y".
- "*(Amount)*" column: transaction value
- "*(Date)*" column: date of the transaction

The next step is to import the macros:

1. Download these files:
    - Module_export.bas
    - Module_format.bas
    - Module_util.bas

    from the project at:

    https://github.com/OneClickToQif/OneClickToQif

2. Press "Alt+F11" to open Visual Basic editor.
3. Click right button on the VBAProject, and select "Import File".
4. Import the three files above

To begin to export you have two options:

a) Press "Alt+F8", then select "ExportAllToQIF" macro

b) Add a button to run the macro: At "Programmer" menu tab, "Insert" button, Form controls, click the button item to create a button in the spreadsheet. Then assign the "ExportAllToQIF" macro to the new button

(*)The position of the labels row can not be changed for the simple transactions spreadsheet. If you need it, make a request at the address in the "How to get help" section.

How to adapt a spreadsheet with split transactions

If you have a spreadsheet with split transactions:

1. Insert a new row. It can be at any position (any row number) on the top of the data.
2. In this new row:
    o   Label the first cell (column #1) as "**QIF**".
    o   Label the cell over the date column in the former row as "**Date**"
    o   Label the cell of the export column as "**Exported**". This column indicates which transactions have already been exported.
    o   Label the cell over the column with comments as "**Memo**"
    o   Label the cell over the columnn with the amount going to the actual account as "**Main**"
    o   Label the cells over other columns for amounts going to other accounts, with the name of those accounts. If you want to invert the amount when exporting (from positive to negative and viceversa), append "(neg)" to the label.

The next step is to import the macros:

1. Download these files:
    *   Module_export_split_transactions.bas
    *   Module_util_split_transactions.bas

    from the project at:

    https://github.com/OneClickToQif/OneClickToQif

2. Press "Alt+F11" to open Visual Basic editor.
3. Click right button on the VBAProject, and select "Import File".
4. Import the two files above

To begin to export, you have two options:

a) Press "Alt+F8", then select "ExportAllToQIF" macro

b) Add a button to run the macro: At "Programmer" menu tab, "Insert" button, Form controls, click the button item to create a button in the spreadsheet. Then assign the "ExportAllToQIF" macro to the new button

Note: The payroll sample in this project is made from the point of view of accounting of an employee. The example in the GNUCash documentation is made from the point of view of a company accounts to pay salaries.