QuestradeReconcile Python Script

QuestradeReconcile © 2019 kerouac01850

Table of Contents

1 Introduction

The QuestradeReconcile macro is python code that uses the Questrade application programming interface (API) to fetch account, position, balance, equity, 30 day activity, and dividend data into a LibreOffice spreadsheet file. This script is meant to be run infrequently to help provide a dashboard view when, for example, re-balancing a portfolio. It is not meant to track real time market conditions and will raise an error when invoked too frequently.

The Questrade platform requires that client applications like the QuestradeReconcile script help protect the integrity of the service from abuse. The service provides a clear expectation of the level of service that the API commits to fulfill. See details here: https://www.questrade.com/api/documentation/rate-limiting.

Index of Tables

Table of Figures

1.1 Reconcile a Spreadsheet with Questrade

Open the Sample spreadsheet and press the *[Reconcile]* button. Inspect the newly updated values in the Accounts, Balances, Positions, Equities, Activities, and Dividends sheets.
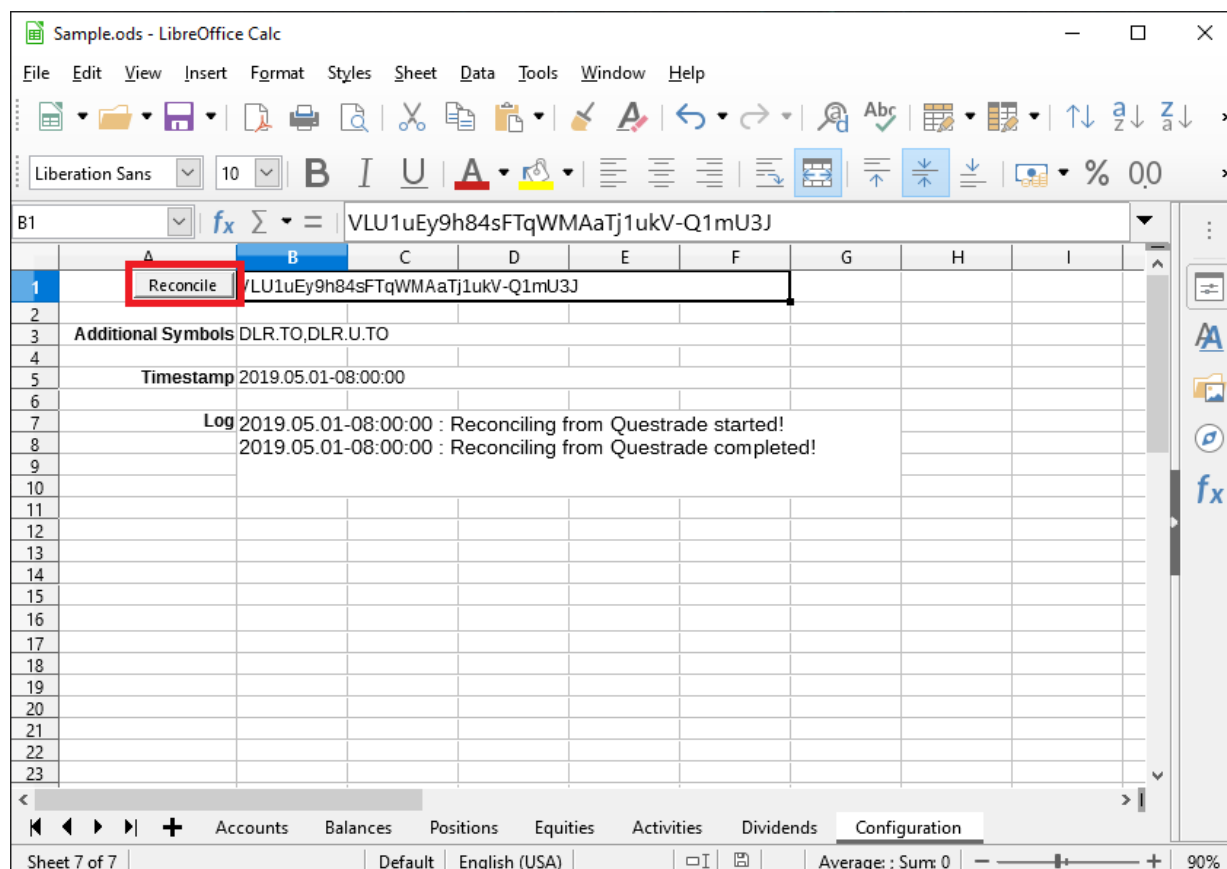
Figure 1: Run the QuestradeReconcile Python Macro

Subsequent sections describe the column definitions.

1.2 Spreadsheet Fields and Formatting

This legend lists the field type, 's' for example for strings, and the corresponding spreadsheet formatting applied to that type of value. These field types are used internally by the python script and are only included here to help document the corresponding columns.

Table 1: Spreadsheet Formatting Codes

Description	Field Type	Calc Format
String	s	@
Date	d	MMM D, YYYY
Boolean	b	BOOLEAN
General	n	General
Number with 1 Decimal Place	n0	#,##0;[RED]-#,##0
Number with 2 Decimal Places	n1	#,##0.0;[RED]-#,##0.0
Number with 3 Decimal Places	n2	#,##0.00;[RED]-#,##0.00
Number with 4 Decimal Places	n3	#,##0.000;[RED]-#,##0.000
Number with 5 Decimal Places	n4	#,##0.0000;[RED]-#,##0.0000
Number with 6 Decimal Places	n5	#,##0.00000;[RED]-#,##0.00000

The following sections list the supported data fields and their formatting for each sheet.

1.2.1 Accounts Sheet

https://www.questrade.com/api/documentation/rest-operations/account-calls/accounts

Table 2: Accounts Sheet Column Definitions

Column Name	Field Type	Description
number	n	Eight-digit account number (e.g., "26598145").
type	s	Type of the account (e.g., "Cash", "Margin"). See **Account Type** section for all allowed values.
clientAccountType	s	Type of client holding the account (e.g., "Individual"). See **Client Account Type** for full list of values.
status	s	Status of the account (e.g., Active). See **Account Status** for full list of values.
isPrimary	b	Whether this is a primary account for the holder.
isBilling	b	Whether this account is one that gets billed for various expenses such as inactivity fees, market data, etc.

1.2.1.1 Account Type

Table 3: Account Type Column Values

type	Description
Cash	Cash account.
Margin	Margin account.
TFSA	Tax Free Savings Account.
RRSP	Registered Retirement Savings Plan.
SRRSP	Spousal RRSP.
LRRSP	Locked-In RRSP.
LIRA	Locked-In Retirement Account.
LIF	Life Income Fund.
RIF	Retirement Income Fund.
SRIF	Spousal RIF.
LRIF	Locked-In RIF.
RRIF	Registered RIF.
PRIF	Prescribed RIF.
RESP	Individual Registered Education Savings Plan.
FRESP	Family RESP.

1.2.1.2 Client Account Type

The following table specifies all supported account client types.

Table 4: Client Account Type Column Values

clientAccountType	Description
Individual	Account held by an individual.
Joint	Account held jointly by several individuals (e.g., spouses).
Informal Trust	Non-individual account held by an informal trust.
Corporation	Non-individual account held by a corporation.
Investment Club	Non-individual account held by an investment club.
Formal Trust	Non-individual account held by a formal trust.
Partnership	Non-individual account held by a partnership.
Sole Proprietorship	Non-individual account held by a sole proprietorship.
Family	Account held by a family.
Joint and Informal Trust	Non-individual account held by a joint and informal trust.
Institution	Non-individual account held by an institution.

1.2.1.3 Account Status

The following table specifies all supported account status values.

Table 5: Account Status Column Values

status
Active
Suspended (Closed)
Suspended (View Only)
Liquidate Only
Closed

1.2.2 Balances Sheet

https://www.questrade.com/api/documentation/rest-operations/account-calls/accounts-id-balances

Table 6: Balances Sheet Column Definitions

Column Name	Field Type	Description
balanceType	s	The type of balance is one of **perCurrencyBalances**, or **combinedBalances**.
account	n	Eight-digit account number (e.g., "26598145").
accountType	s	Type of the account (e.g., "Cash", "Margin"). See **Client Account Type** section for all values.
currency	s	Currency of the balance figure (e.g., "USD" or "CAD").
cash	n2	Balance amount.
marketValue	n2	Market value of all securities in the account in a given currency.
totalEquity	n2	Equity as a difference between cash and marketValue properties.
buyingPower	n2	Buying power for that particular currency side of the account.
maintenanceExcess	n2	Maintenance excess for that particular side of the account.
isRealTime	b	Whether real-time data was used to calculate the above values.

1.2.3 Positions Sheet

https://www.questrade.com/api/documentation/rest-operations/account-calls/accounts-id-positions

Table 7: Positions Sheet Column Definitions

Column Name	Field Type	Description
account	n	Eight-digit account number (e.g., "26598145").
accountType	s	Type of the account (e.g., "Cash", "Margin"). See **User Account Type** section for all allowed values.
currency	s	Currency of the equity (e.g., "USD" or "CAD").
symbol	s	Equity symbol.
symbolId	n	Internal symbol identifier.
openQuantity	n3	Position quantity remaining open.
currentPrice	n2	Current price of the position symbol.
currentMarketValue	n2	Market value of the position (quantity x price).
averageEntryPrice	n3	Average price paid for all executions constituting the position.
totalCost	n2	Total cost of the position.
openPnl	n2	Unrealized profit/loss on this position.
dayPnl	n2	Unrealized profit/loss on this position based on open current prices.
closedQuantity	n3	Portion of the position that was closed today.
closedPnl	n2	Realized profit/loss on this position.
isUnderReorg	b	Designates whether a symbol is currently undergoing a reorg.
isRealTime	b	Designates whether real-time quote was used to compute PnL.

1.2.4 Equities Sheet

https://www.questrade.com/api/documentation/rest-operations/market-calls/symbols-id

Table 8: Equities Sheet Column Definitions

Column Name	Field Type	Description
account	n	Eight-digit account number (e.g., "26598145").
accountType	s	Type of the account (e.g., "Cash", "Margin"). See **User Account Type** section for all allowed values.
currency	s	Currency of the equity (e.g., "USD" or "CAD").
symbol	s	Equity symbol namer.
symbolId	n	Internal symbol identifier.
description	s	Symbol description (e.g., "Microsoft Corp.").
listingExchange	s	Primary listing exchange. See **Listing Exchange** section for the list of all listing exchanges.
securityType	s	Security type (e.g., "Stock"). See **Security Type** section for all allowed values.
prevDayClosePrice	n2	Closing trade price from the previous trading day.
yield	n4	Dividend yield (dividend / prevDayClosePrice).
pe	n4	Trailing 12-month price to earnings ratio.
eps	n4	Trailing 12-month earnings per share.
outstandingShares	n0	Total number of shares outstanding.
marketCap	n0	Market capitalization (outstandingShares * prevDayClosePrice).
averageVol20Days	n0	Average trading volume over trailing 20 days.
averageVol3Months	n0	Average trading volume over trailing 3 months.
dividend	n4	Dividend amount per share.
dividendDate	d	Dividend declaration date.
exDate	d	Dividend ex-date.
lowPrice52	n2	52-week high price.
highPrice52	n2	52-week high price.
tradeUnit	b	Smallest number of units that can be traded.

1.2.4.1 Listing Exchange

The following table specifies all supported listing exchanges.

Table 9: Listing Exchange Column Values

listingExchange	Description
TSX	Toronto Stock Exchange.
TSXV	Toronto Venture Exchange.
CNSX	Canadian National Stock Exchange.
MX	Montreal Exchange.
NASDAQ	NASDAQ.
NYSE	New York Stock Exchange.
NYSEAM	NYSE AMERICAN
ARCA	NYSE ARCA.
OPRA	Option Reporting Authority.
PinkSheets	Pink Sheets.
OTCBB	OTC Bulletin Board.

1.2.4.2 Security Type

The following table specifies all supported security types.

Table 10: Security Type Column Values

securityType	Description
Stock	Common and preferred equities, ETFs, ETNs, units, ADRs, etc.
Option	Equity and index options.
Bond	Debentures, notes, bonds, both corporate and government.
Right	Equity or bond rights and warrants.
Gold	Physical gold (coins, wafers, bars).
MutualFund	Canadian or US mutual funds.
Index	Stock indices (e.g., Dow Jones).

1.2.5 Activities Sheet

Account activities, including cash transactions, dividends, trades, etc.

https://www.questrade.com/api/documentation/rest-operations/account-calls/accounts-id-activities

Table 11: Activities Sheet Column Definitions

Column Name	Field Type	Description
account	n	Eight-digit account number (e.g., "26598145").
accountType	s	Type of the account (e.g., "Cash", "Margin"). See **User Account Type** section for all allowed values.
currency	s	Currency of the equity (e.g., "USD" or "CAD").
transactionDate	d	Transaction date.
symbol	s	Equity symbol name.
symbolId	n	Internal symbol identifier.
type	s	Activity Type
action	s	No documentation from Questrade.
quantity	n3	The quantity
price	n4	The price
grossAmount	n2	Gross amount
commission	n2	The commission
netAmount	n2	Net Amount
tradeDate	d	Trade date
settlementDate	d	Settlement date
description	s	No documentation from Questrade.

1.2.6 Dividends Sheet

!!!!! CURRENTLY ONLY TSX STOCKS ARE SUPORTED !!!!!

Here you'll find dividend information and dividend paying stocks with complete payout histories, increase announcements, growth, yield and much more.

https://dividendhistory.org/

Table 12: Dividends Sheet Column Definitions

Column Name	Field Type	Description
account	n	Eight-digit account number (e.g., "26598145").
accountType	s	Type of the account (e.g., "Cash", "Margin"). See **User Account Type** section for all allowed values.
symbol	s	Equity symbol name.
symbolId	n	Internal symbol identifier.
currency	s	Currency of the equity (e.g., "USD" or "CAD").
dividend	d	Dividend declaration date.
payout	d	Dividend ex date.
amount	n6	Dividend amount per share.
note	s	** indicates unconfirmed/estimated

2 Installation and Configuration

The spreadsheet file must have seven sheets with the following names: Accounts, Positions, Balances, Equities, Activity, Dividends and Configuration. Except for the Configuration sheet any existing data on these sheets will be cleared when the QuestradeReconcile python macro is run. Data from the Questrade on-line platform will be used to update these sheets. The sheets can be moved within the file: sheet order does not affect anything. Any other sheets in the file are ignored.

The Configuration sheet must have the following cells defined for the QuestradeReconcile macro to function correctly. In the Sample spreadsheet file the adjacent cells to the left are used as labels. The labels are not required by the macro. Add these cells to your custom spreadsheet.

- **Cell $Configuration.B1**: Questrade authentication *token_value* text string. See 2.4 Questrade API Activation for more details.

- **Cell $Configuration.B3:** A text cell with a comma separated list of equity symbols. The equities on the positions sheet is sourced directly from the Questrade API. Equity values for these symbols are appended to the Equities sheet in addition to those already referenced by the Positions sheet.

- **Cell $Configuration.B5**: A date cell indicating the last time the macro was run. It is updated automatically by the QuestradeReconcile script.

- **Cell $Configuration.B7**: A text cell for logging macro status. Usually helps to merge surrounding cells to form a text block using B7 as the root cell. This cell is overwritten with each invocation of the macro and records any errors encountered when running the script.
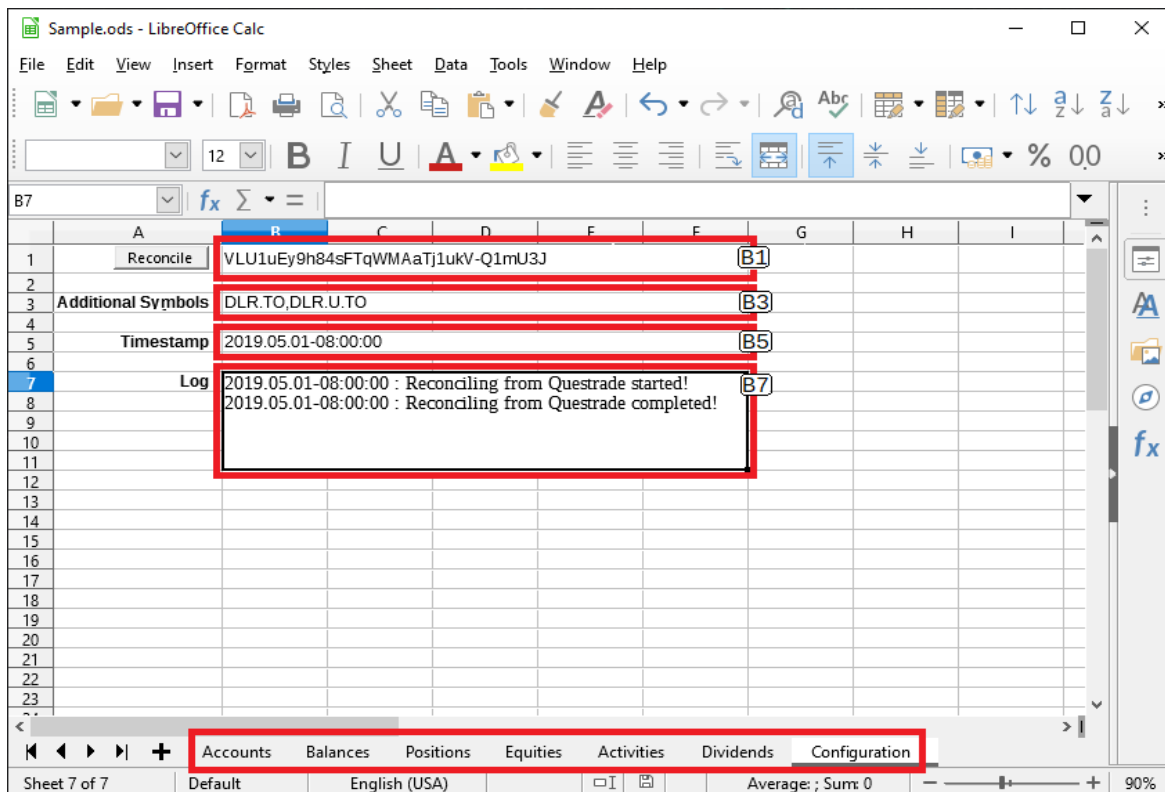
Figure 2: QuestradeReconcile Configuration Properties

The configuration items can be changed to different pages or cell locations by modifying the python code. See **config_token**, **config_equities**, **config_timestamp**, and **config_log** at the top of the QuestradeReconcile.py macro file. The file can be modified by using a Windows text editor such as Notepad or Write.

```
config_token = '$Configuration.B1'
config_equities = '$Configuration.B3'
config_timestamp = '$Configuration.B5'
config_log = '$Configuration.B7'
```

The following sections describe the installation steps necessary to run the QuestradeReconcile python macro from within the Sample or any other LibreOffice Calc spreadsheet.

Before proceeding be hyper aware of two special instructions and warnings:

1. After generating a refresh token in 2.4 Questrade API Activation DON'T CLOSE THE POP-UP WINDOW before saving the token into the spreadsheet Figure 1: Run the QuestradeReconcile Python Macro.

2. The underlying questrade-api-1.0.3 library has incomplete or undocumented error handling that may require a token reset as described in 2.4.3 Significant Bug in questrade-api-1.0.3 Library. If you get an authentication error try the described workaround.

2.1 LibreOffice Installation

Go to https://www.libreoffice.org/download/download/ to download and find installation instructions for the LibreOffice suite.

Figure 3: LibreOffice Download at www.libreoffice.org

When LibreOffice installation is complete verify:

- Directory ***C:\Program\Files\LibreOffice\program*** exists and the ***python.exe*** file can be found there.
- Launch LibreOffice Calc using whatever method you like.

2.2 LibreOffice Macro Security Configuration

LibreOffice macros must be enabled to run the QuestradeReconcile python script. Open the Calc Sample spreadsheet and navigate to the *[Tools → Options]* dialog or press *[Alt+F12]*.
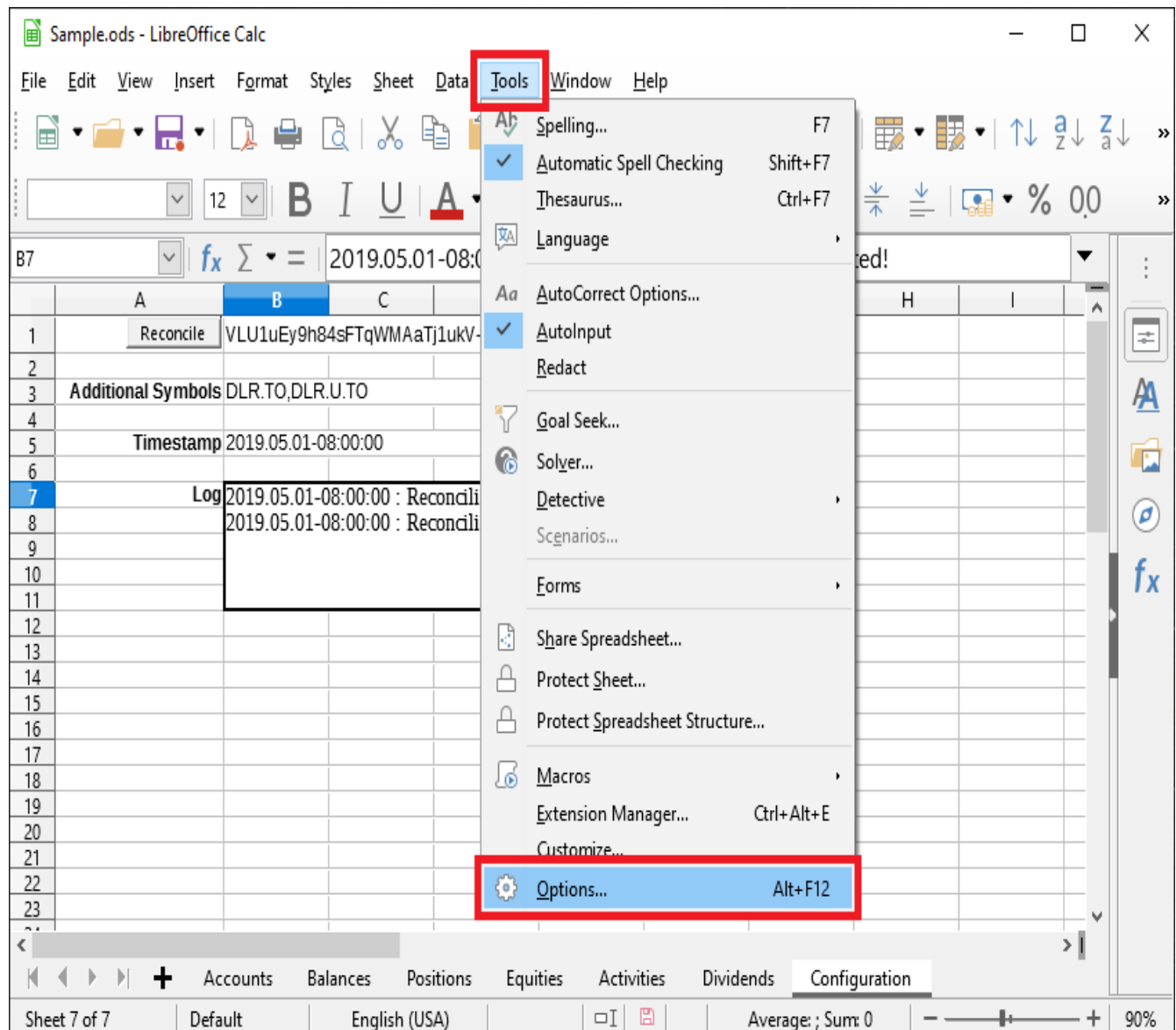
Figure 4: LibreOffice Tools Options

Set security to **Medium** to require confirmation before the QuestradeReconcile python macro will run.
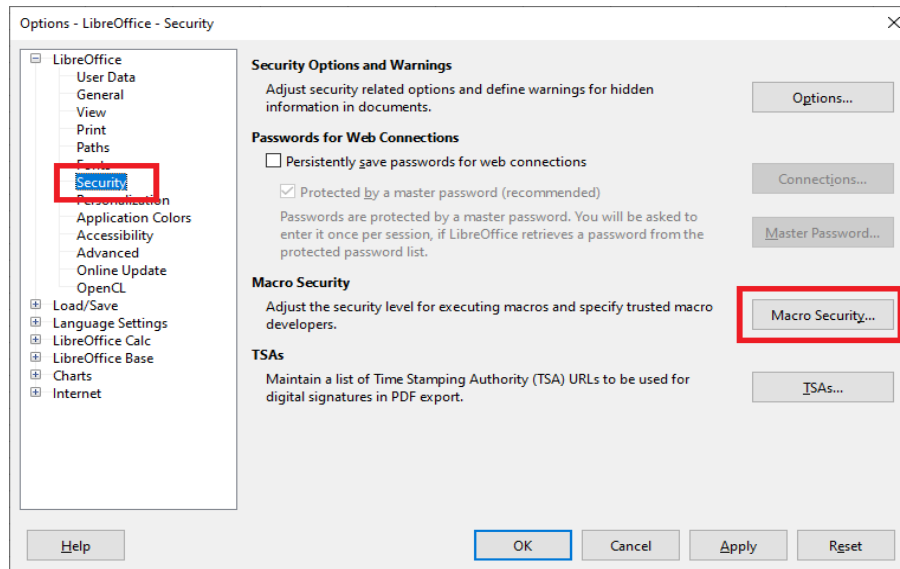
Figure 5: LibreOffice Macro Security
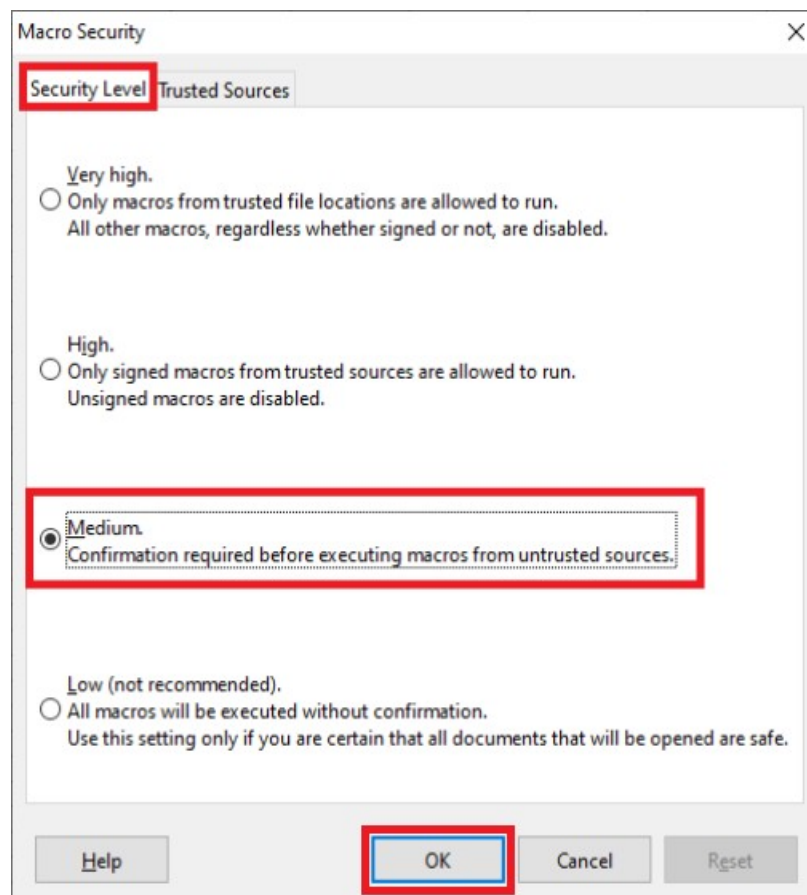
Figure 6: LibreOffice Medium Security Level

When security settings have been updated then proceed to verify that python macros are enabled by choosing the *[Tools → Macros → Run Macro]* menu option from within LibreOffice Calc, then run **HelloWorldPython** from the **HelloWorld** Library.
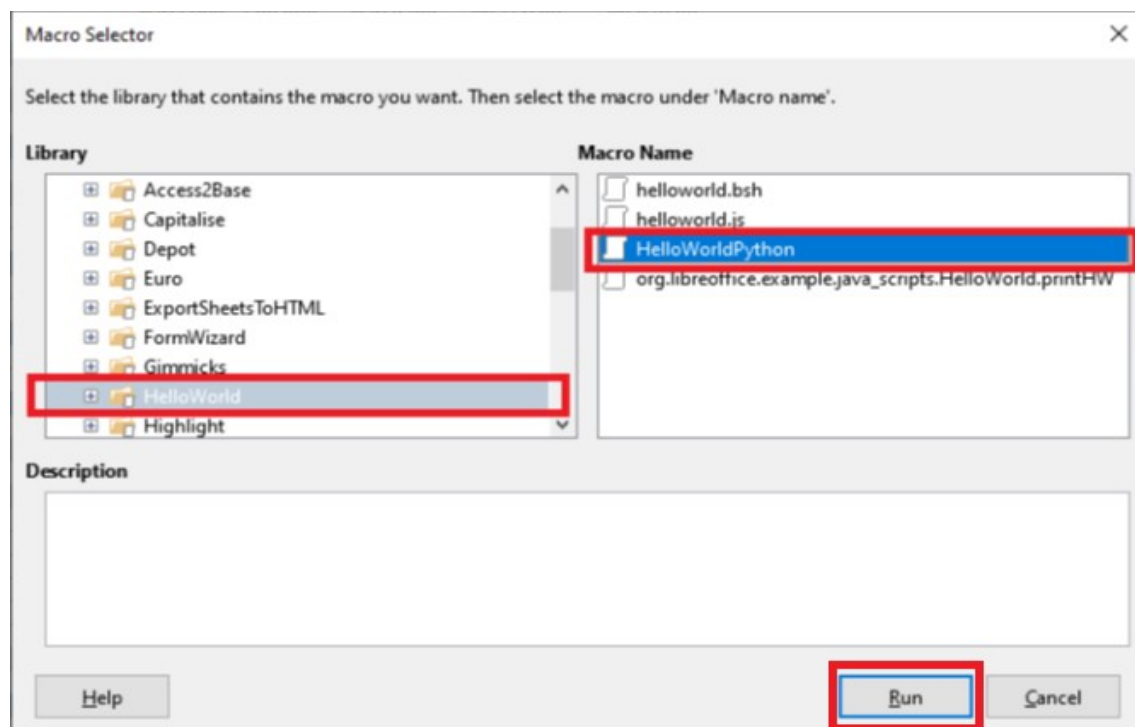
Figure 7: LibreOffice Run HelloWorldPython

The **HelloWorldPython** macro will open a LibreOffice Writer document showing "Hello World (in Python)" when successful.
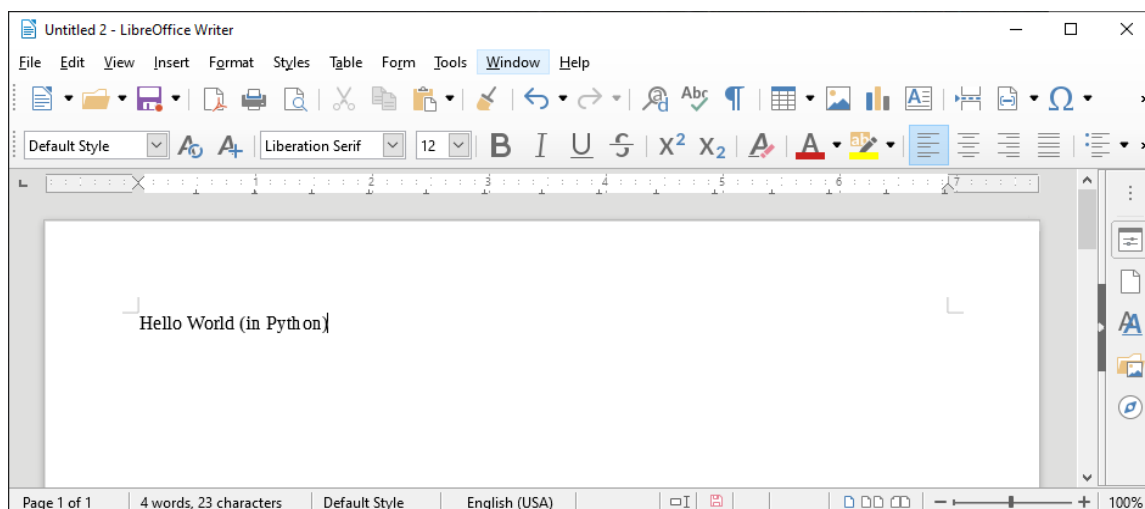
Figure 8: LibreOffice HelloWorld (in Python)

2.3   Python questrade-api-1.0.3 Library Installation

Python questrade-api-1.0.3 library must be installed into the LibreOffice python site packages. First download the ZIP file from https://github.com/tgardiner/questrade_api.

Click on *[Clone or Download]* button then select *[Download ZIP]* to download *questrade_api‑master.zip* into your download directory.
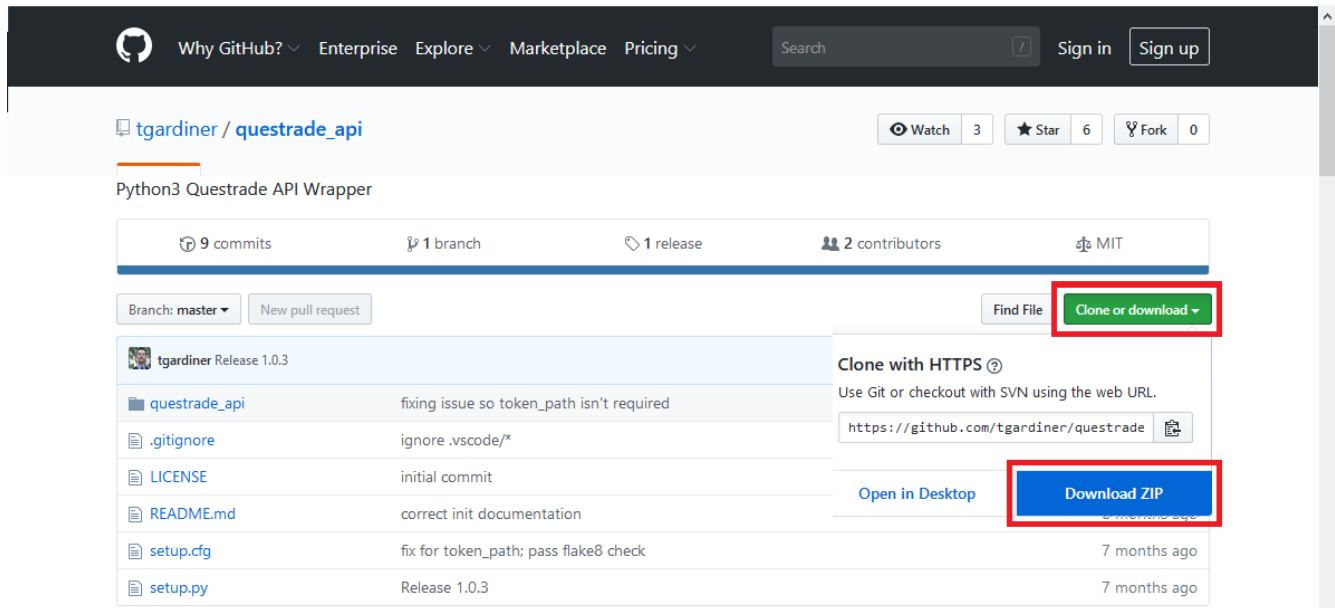
Figure 9: questrade-api-1.0.3 Download ZIP

From the download directory select the *questrade_api‑master.zip* file and then use the right-click Windows context menu to extract all files from the ZIP archive into the download directory.
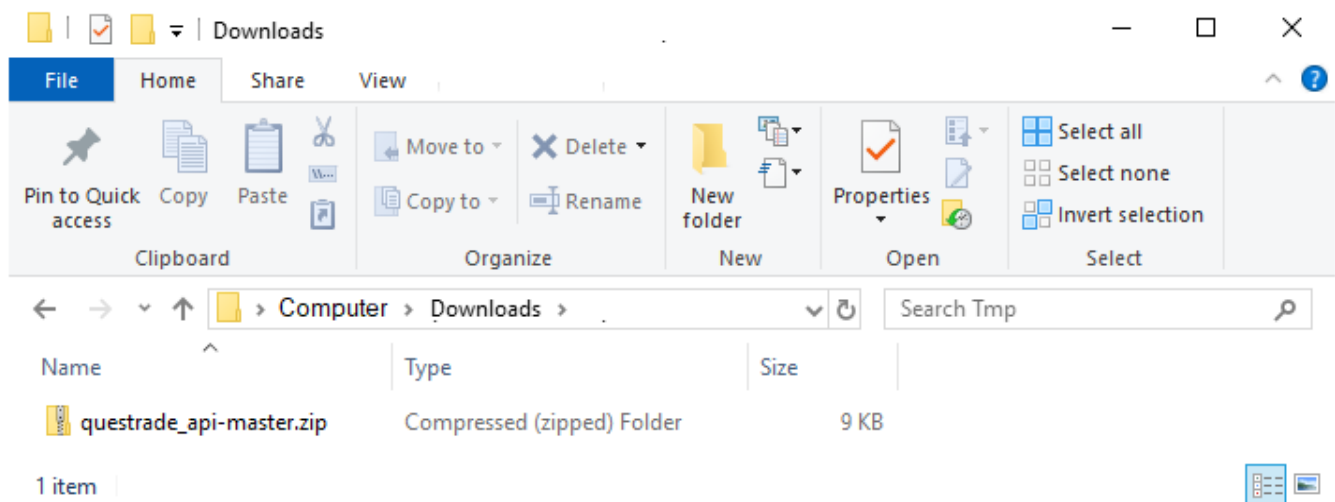
Figure 10: questrade-api.10.3 Windows Download Directory

Copy the recently extracted *questrade_api* sub-directory from its *questrade_api-master* *parent directory to **C:\Program Files\LibreOffice\program\python-core-3.5.7\lib\site-packages**.*

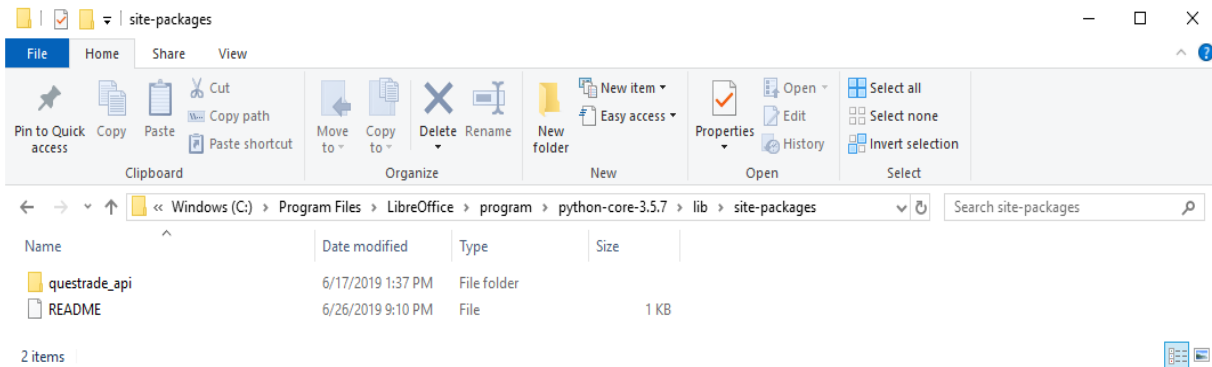*The updated site-packages directory should now look like this:*

Figure 11: LibreOffice Windows Site Packages Directory

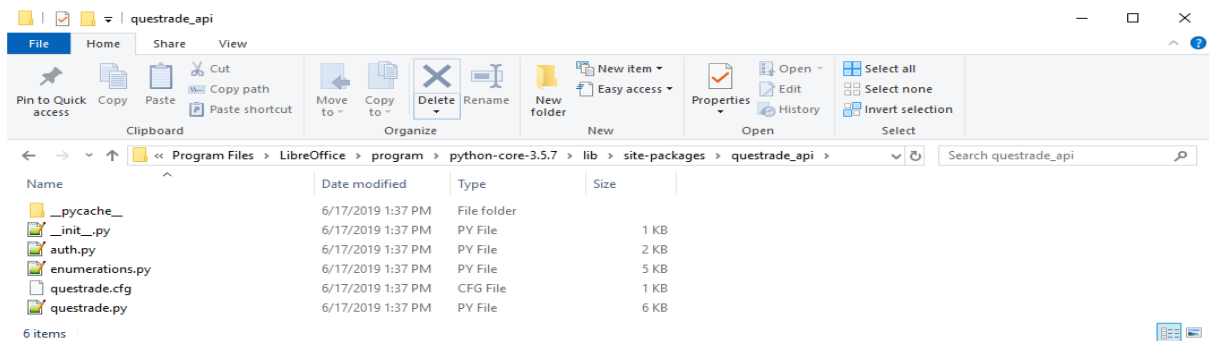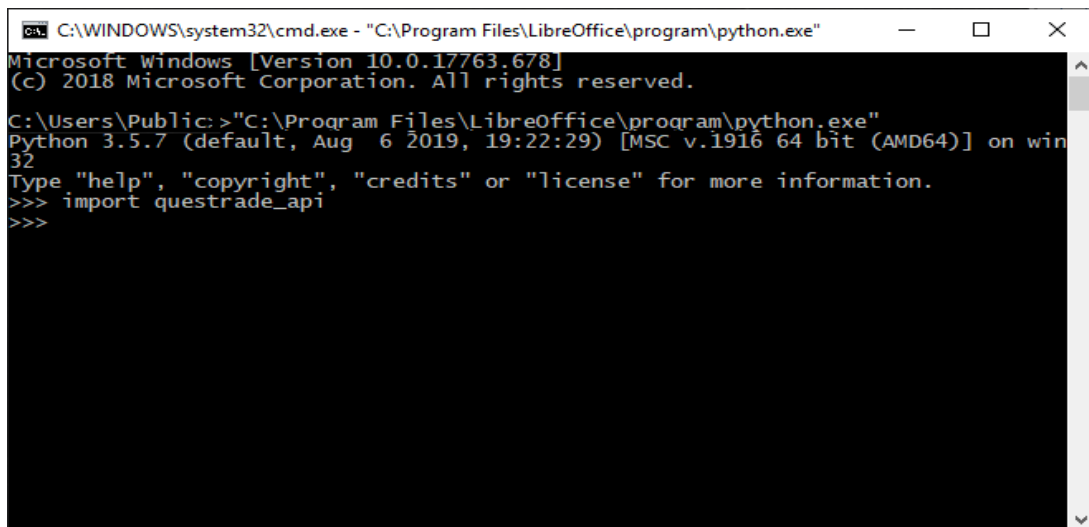The newly copied *questrade_api* directory has the following contents.

Figure 12: questrade-api.1.0.3 Windows Extracted ZIP Directory

When the files have been copied, proceed to verify that the **questrade_api** module is available to the python interpreter:

- *[Windows-R] at the Windows desktop* followed by typing *cmd [Enter]* into the Run dialog box.
- **"C:\Program Files\LibreOffice\program\python.exe"** into the Windows command shell.
- **import questrade_api [Enter]** at the >>> python prompt within the Windows command shell.

No error messages means the Questrade API was successfully loaded into the Python interpreter.

Figure 13: Python import questrade_api

2.4   Questrade API Activation

The Questrade platform implements security measures as defined by the OAuth 2.0 protocol. To allow QuestradeReconcile to be authorized by the Questrade platform a refresh token is needed. This token is obtained by registering the QuestradeReconcile application and then by generating a time limited authorization token within the Questrade web portal.

2.4.1 Create Questrade Token

First, log into Questrade web portal and navigate to settings from the *[Client Name]* pull-down menu.
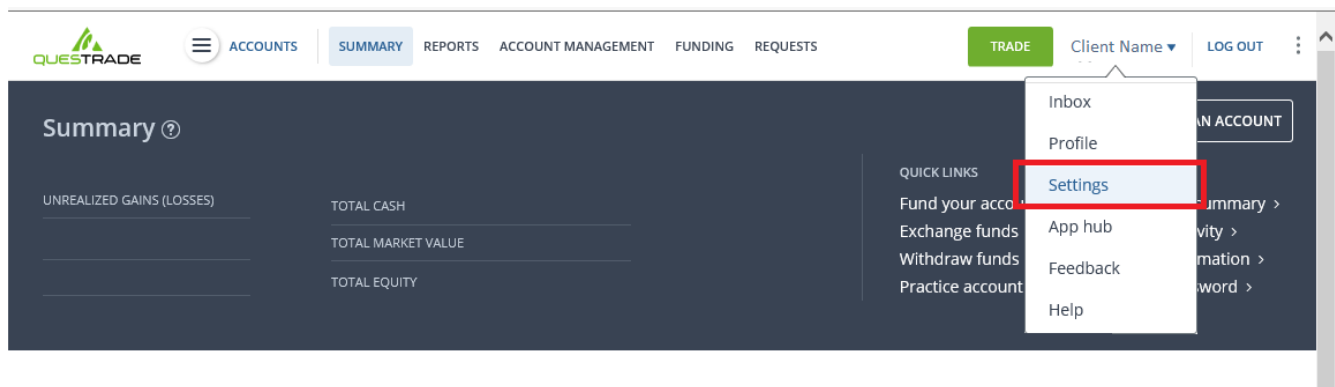

Figure 14: Questrade Launch Settings Menu

Read the Questrade API terms and conditions and agree that you wish to enable authenticated connections from QuestradeReconcile. Rejecting the terms and conditions means the macro will not be permitted to access the Questrade API and the QuestradeReconcile macro will fail.
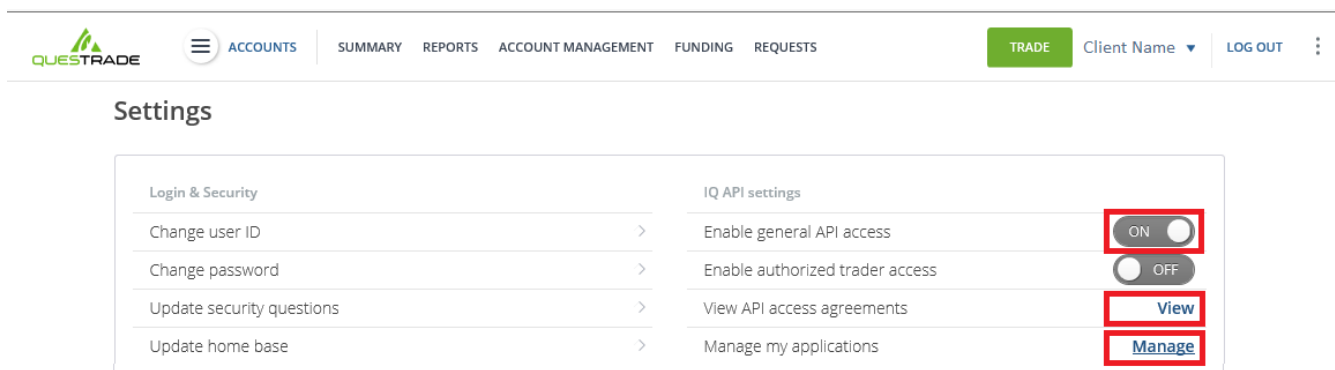

Figure 15: Questrade Settings Dialog

Register QuestradeReconcile as a personal application.

Figure 16: QuestradeReconcile Security Dialog

Generate a refresh token. DON'T CLOSE THE POP-UP WINDOW. Read the next step before proceeding.

Figure 17: Questrade Security Token Dialog

Copy the token from the Questrade pop-up window into the Sample spreadsheet or within your own customized spreadsheet then close the Questrade pop-up window.

The only guaranteed method of copying the token is to select it as a text block in the browser, then copy using the right-click context menu, then paste into the spreadsheet cell. Don't close the pop-up window until you've verified the copied text is identical.

The pop-up window is the only permanent record of the token so closing the window without first copying means the token is lost forever and a new one must be generated.
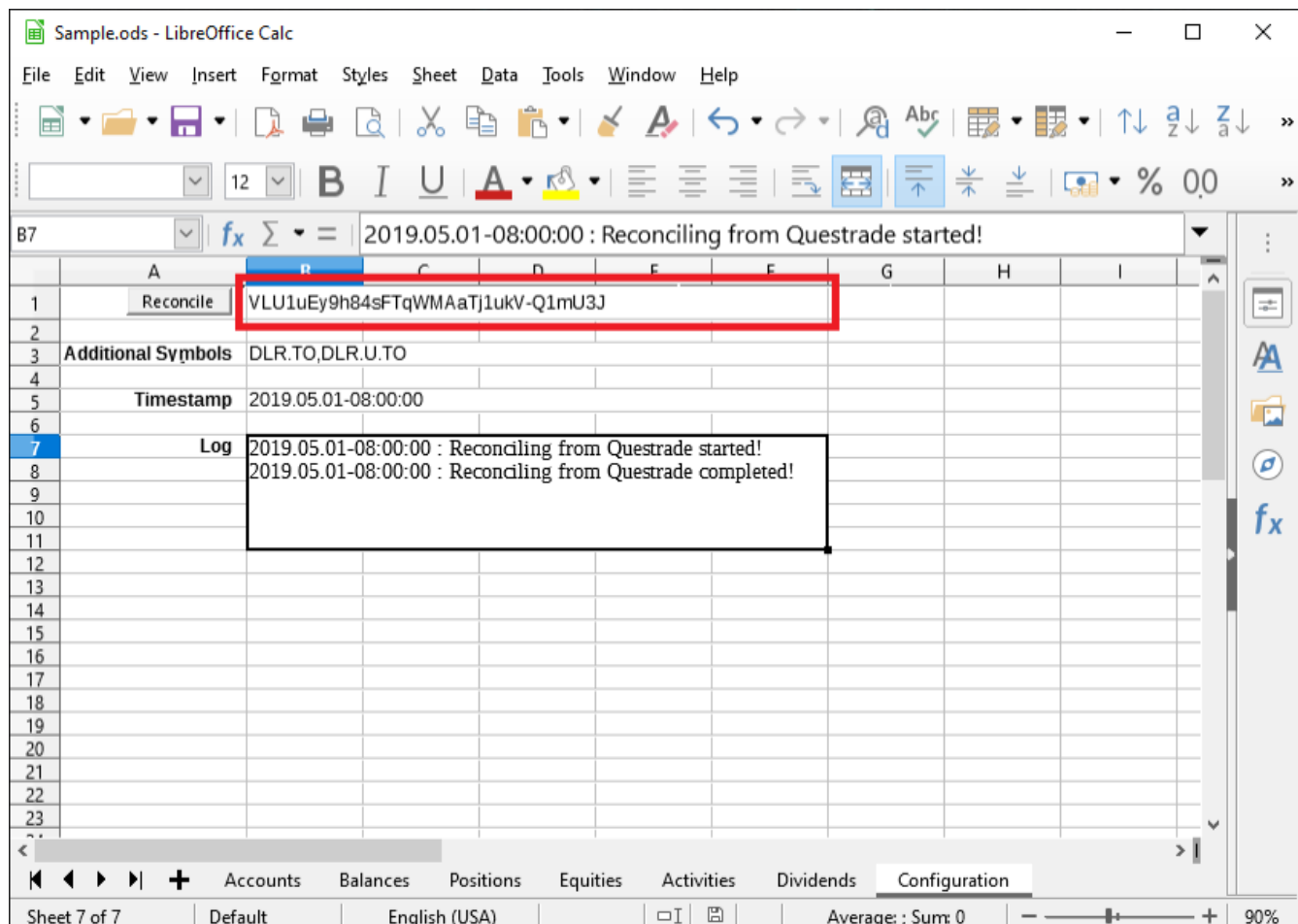

Figure 18: LibreOffice QuestradeReconcile Security Token Configuration

To revoke authorization at any time click Delete within the Questrade Personal apps menu shown above.

2.4.2 Verify Questrade Token

Let's verify the new token. Launch a command window and type the following commands including quotes as shown below.

NOTE: You have to paste the refresh token into the command line where refresh_token is assigned a value. So replace **insert your token here** with the token generated by the Questrade pop-up window. Keep the single quotes but replace everything between them.

- *[Windows-R] at the Windows desktop* followed by typing ***cmd [Enter]*** into the Run dialog box.
- **"C:\Program Files\LibreOffice\program\python.exe"**
  into the Windows command shell.
- **from questrade_api import Questrade [Enter]**
  at the >>> python prompt within the Windows command shell.
- **q = Questrade(refresh_token='insert your token here') [Enter]**
  at the >>> python prompt within the Windows command shell.
  Don't attempt this a second time when successful. See section 2.4.3 Significant Bug in questrade-api-1.0.3 Library.
- **q.time [Enter]**
  at the >>> python prompt within the Windows command shell.

No error messages means the Questrade API was successfully loaded into the Python interpreter, the API connected to the Questrade platform, and the server responded to a query for the server's time value.
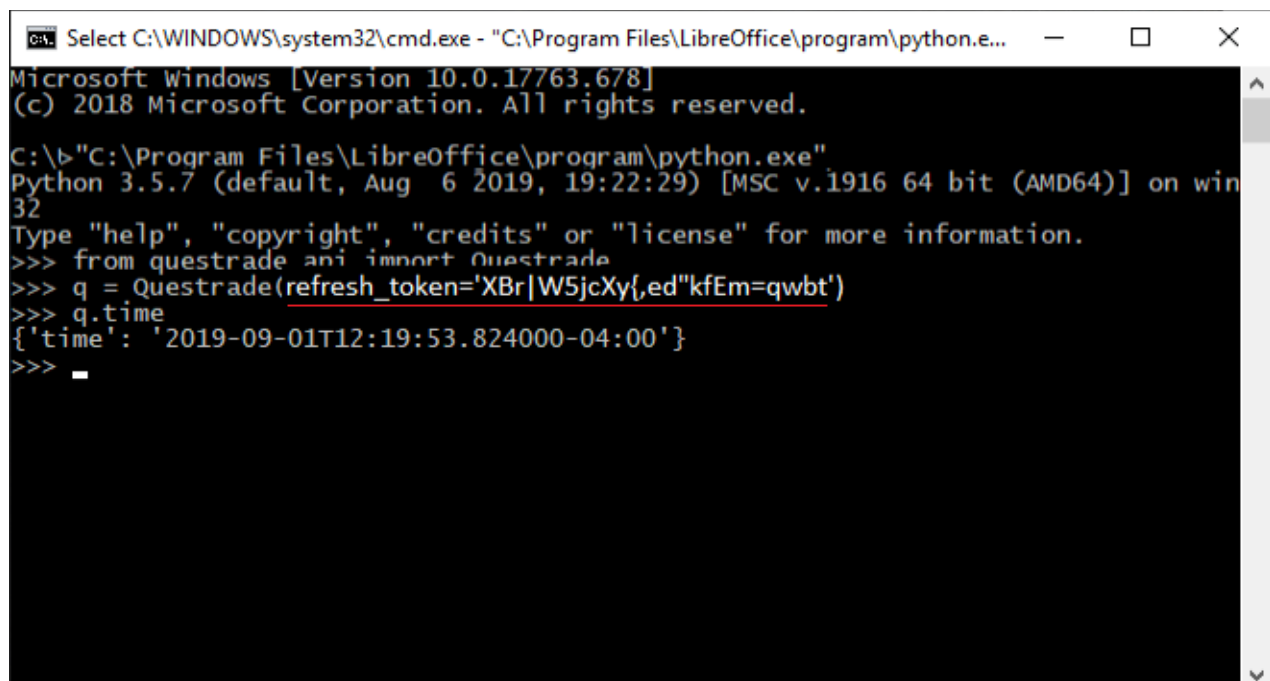

Figure 19: Windows Python Security Token Validation

2.4.3 Significant Bug in questrade-api-1.0.3 Library

The python questrade-api-1.0.3 Library keeps state in a file called **.questrade.json**, with a leading period, in your Windows user directory. In some cases it's necessary to:

1) delete this file and,

2) generate a new token using the Questrade portal as described above.

Connections initiated from the questrade-api-1.0.3 library will seemingly fail to connect for some unspecified reason and the above steps will resolve the problem.

The questrade-api-1.0.3 library documentation mentions "future initiations will not require a refresh token" after the API has been initialized the first time, but this can't be correct since tokens have a life cycle and therefore expire.

Supplying a refresh_token may actually be forbidden until it has expired. This is a software defect and an area of future research but the workaround mentioned above will fix the issue.

> **Important:** A token will be created at `~/.questrade.json` and used for future API calls
>
> - If the token is valid future initiations will not require a refresh token

```
from questrade_api import Questrade
q = Questrade()
```