Processing Bank Data for Reporting and Budgeting

By James K Beard

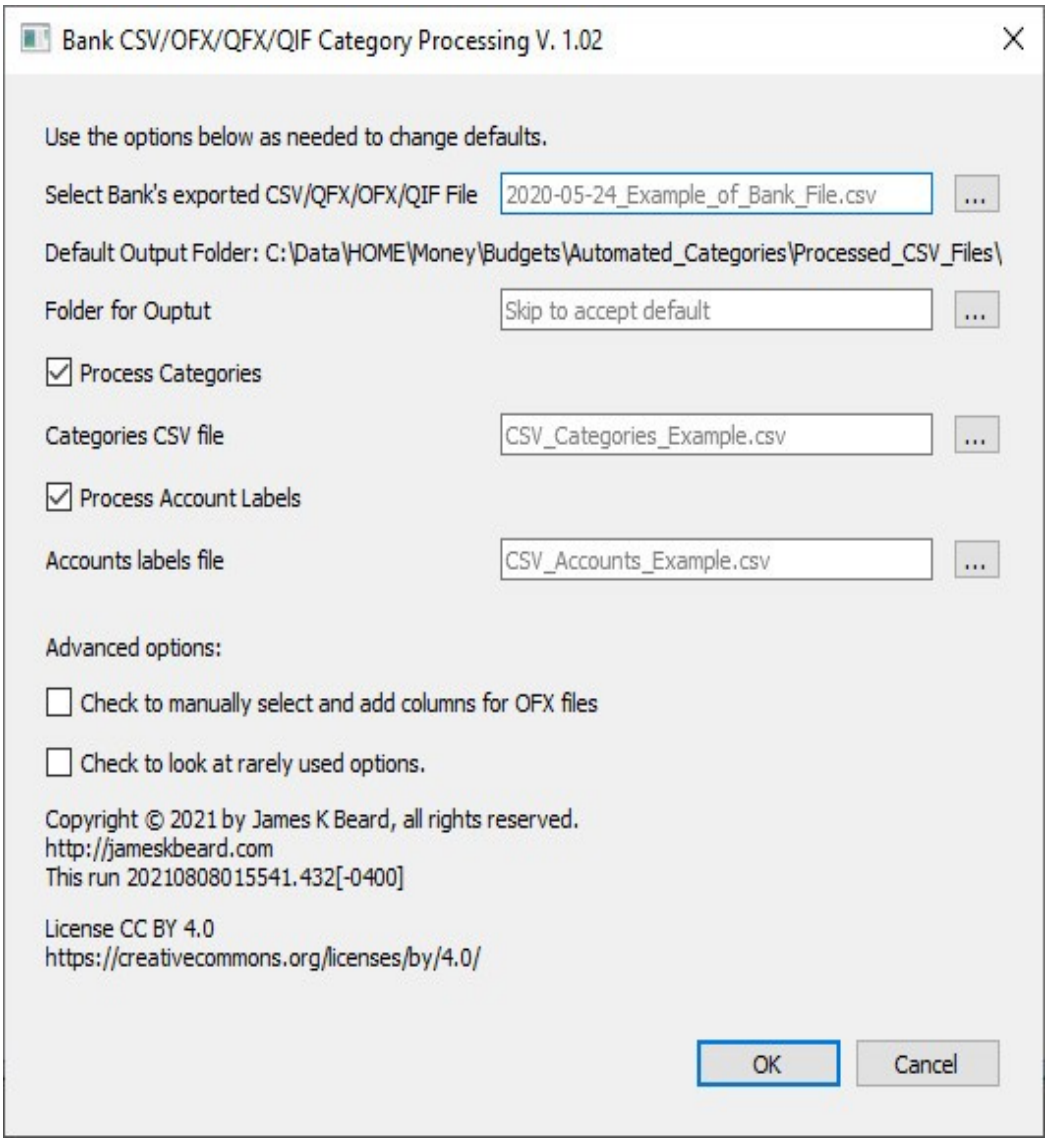

Table of Contents

Table of Figures

Index of Tables

Executive Summary

This document accompanies a Windows computer program that processes bank downloads of transaction data files to produce extra information about transaction categories and subcategories. The program produces comma-separated value files that are loaded by the user into spreadsheets such as Microsoft Excel, Corel Quattro Pro, LibreOffice Calc, Lotus 1-2-3, and others. A pivot table, which is a utility included with almost all currently available spreadsheet programs, then produces short transaction summaries that are easily used in understanding expenses, cash flows, and budgets.

The simplest use of this program is to produce CSV files for spreadsheets from Quicken or OFX files. This program can perform that function immediately after downloading.

Not all bank transactions can be assigned categories by the program because the information provided by the bank sometimes does not covey sufficient information to do so. Category and subcategory information can be added to these transactions by the user to the data in the empty category fields in the spreadsheet produced by this program. Information from the user's known history of such transactions, or information obtained manually from the bank by looking at check images fleshes out the categories and subcategories, so that the information summarized by the pivot table is complete.

Introduction

What this program does for you

This program is a utility that reads transaction downloads from banks in all the most common formats, including CSV or Excel, Quicken, Microsoft Money, etc. From bank information that this program reads form these files, this program produces comma-separated value files, storing the results in a file with a name beginning with the current date and three letters, then the input file name, yyyy-mm-dd_JKB_[the input file name].csv, in a separate location of your choosing. All the fields in the output file are in formats recognized by all spreadsheet programs as data, or that can be processed by spreadsheets. This puts the user in complete control of how to process and summarize bank transactions using spreadsheets.

A valuable and powerful spreadsheet utility, the pivot table, can be used on this program's output to produce summaries of transaction data. To simplify this process, this program can produce extra columns in the output spreadsheet for transaction categories and subcategories. If your bank data includes transactions from more than one account and account identification information is supplied, this program can produce a column with plain text account identification of your choice, such as "Ozzie's Checking" and "Harriet's Checking."

If you are not fluent with spreadsheets but you want the flexibility of this program, please refer to sections Loading the Output Files into a Spreadsheet and Making the Categories Work for You below.

Explicit step-by-step instructions, with screen shots as illustrations, take you through all the steps that you need to use the program outputs and produce simple reports.

Pivot table expertise is not common, so this document provides a walk-through on how to produce a transaction summary from processed bank data using a simple pivot table. The method shown here is just three steps.

Should You Use This Program?

This program is not a substitute for paid financial programs such as Quicken, Microsoft Money, Personal Capital, or even free programs like GnuCash. The commercial packages do not require that the user operate or understand accounting, spreadsheets, or financial reporting, and instant help is available by phone or computer chat. GnuCash and other free or shareware utilities provide similar capability without included phone or online live support unless contracted separately. This program does automate notably time-consuming tasks such as providing usable spreadsheet from raw bank data, and assigning categories and subcategories when bank information is available to support that. Most importantly, it allows the user full control of the process. If you need to be able to ask people to talk you through processing bank data, you need a paid package. If you fell constrained or restricted by financial software and are proficient with spreadsheets or are willing to learn, this program is for you.

How to Use This Program

In the following paragraphs we will walk through downloading and installing this program, and using it for the first time.

Installing This Program

This program comes as files collected and organized in a ZIP archive file. It’s simplest when you create a new folder and extract all the files from the ZIP archive into this folder. The files are listed in Table 1 below. All of the files must be present for the program to run. Two of them are in subfolders; these subfolders are in the ZIP archive and must be downloaded with CSV_Processing.exe and in the same folder as that file, and contain the small files indicated in Table 1 so that the program can find them when it starts up.

The last three files, an information text file, this manual as a PDF file, and a file produced when the program is first run, CSV_Processing.ini, do not need to remain in the same folder as the program. The file CSV_Processing.ini is the “memory” of the program and is updated each time it is run; this allows the program to “remember” what folders and files were used when the program was last run.

Table 1: Files for This Program

Name of File	Function
CSV_Processing.exe	Main program; double-click in this file to run
Qt5Core.dll	Used by program for user interface
Qt5GUI.dll	Used by program for graphical user interface (GUI)
Qt5PrintSupport.dll	Connects Qt version 5 to Windows printing
Qt5Widgets.dll	Provides the various devices used in the GUI
./platforms/qwindows.dll	Using Qt version 5 capabilities in Windows
./printsupport/windowsprintsupport.dll	Printing the Qt widgets on Windows
OFX_Fields.csv	List of plain text titles for CSV output files when Quicken/OFX input is used
VC_Redistributables.txt	Information; see the section When Microsoft Visual C 2013 Application Support is Missing
CSV_Processing_Manual_03.pdf	This short Users' Manual
CSV_Processing.ini	Not in installation; appears after program is run

Copying the Files

The simplest way to install this program is to make a new, empty directory and copy the ZIP archive file into this folder and use the built-in EXTRACT ALL of your ZIP archive. If you simply use the Windows built-in recognition of ZIP archives as folders instead of WinZIP, BitZipper, PKZip or similar utility, simply drag-and-drop the files and folders in the same configuration that you see in the archive.

The file OFX_Fields.csv is a table of column headings that are used in lieu of OFX field names in your CSV output file when you use Quicken or OFX files for input. The program reads this file when it starts. If you edit this file and the program gets an input I/O error, replace it with the one from your ZIP archive and try again.

This file, CSV_Processing_Manual_03.pdf, may be obtained separately or be included in the ZIP archive. It can be moved or deleted without affecting the program. The text file VC_Redistributables.txt file has served its purpose once the program is running and may be deleted.

The file CSV_Processing.ini is not included with installation. The program generates this file the first time it is run. You should leave CSV_Processing.ini in place because it is the memory of the last file and folders used, and simplifies usage of the program. The usual Windows installation processes, copying files into the system areas and modifying the registry, are not done by copying this program to your computer. Rather than using the Windows registry to store and retrieve its "memory," this program saves its data in CSV_Processing.ini.

This program installation does not use the Windows registry or change file associations. So, once the files are copied and the program will start up, installation is complete. Uninstallation is accomplished simply and completely by deleting the files in Table 1.

When Microsoft Visual C 2013 Application Support is Missing

Most Windows applications are compiled with Microsoft development tools. The compiler used for this program, Absoft Pro Fortran 2021, is no exception. Compiled applications use redistributable libraries that are integrated into Windows during installation of a commercial application. If you double-click on CSV_Processing.exe and see a pop-up stating that MSVCP120.dll or similarly named DLL is missing, then you have not yet installed an application compiled using Microsoft Visual C 2013. You can install the Microsoft Visual C 2013 Redistributable files as needed by this program that you get from the Microsoft page

https://www.microsoft.com/en-ph/download/details.aspx?id=40784

An updated version for very high resolution computers such as those used for graphics arts in some studios is available through a download link on

https://support.microsoft.com/en-us/topic/update-for-visual-c-2013-redistributable-package-d8ccd6a5-4e26-c290-517b-8da6cfdf4f10

or directly from this Microsoft page:

https://aka.ms/highdpimfc2013x64enu

Once installed, this package is maintained by Windows Update and never needs attention again. NEVER download this DLL or any other Windows system file from a non-Microsoft website.

This program is compiled as a 64-bit application; future versions may be released as a 32-bit program. I recommend that you install both 32-bit and 64-bit versions of the Microsoft Visual C Redistributable.

Data Files that You Provide to Enable Capabilities

This program can be used simply to reformat data provided by the bank. Optionally, this program can add a column to your spreadsheet, “Account,” that identifies the account for each transaction in simple text. More importantly, this program can use the data provided by the bank to assign categories and subcategories for use in summarizing data for reports and analysis, or for budgeting. Small data tables must be provided by the user for each of these two optional capabilities. You may need to find and update these tables from time to time, particularly the Categories table. I recommend that you add a folder where you copied the program and put only the two tables in that folder. A good place for that is a subfolder of the folder where you placed this program.

The Categories Table

The program uses bank data to assign categories and subcategories. You provide a table that identifies bank data by using a snippet that you see in the bank data fields that describe the payee or other information about the transaction. This data consists of a keyword or set of keywords, which we call snippets, that identifies a transaction, with the category and subcategory that correspond to transactions

identified by these keywords. Each line of your table enables this program to produce one category and subcategory combination.

The Accounts Table

This program has the capability to identify the account for each transaction from data provided by some banks. Some banks allow downloading of multiple accounts at once, and this capability allows labeling each transaction with the account name. A small set of data, with each line consisting of an identifier such as the last four digits of the account number, with the account name, are used to implement this by the program.

Locations of Bank Downloads and Processed Files

The simplest way to keep aware of what files are downloaded from the bank, which files are produced by this program, and the program files themselves, is to keep separate folders for each group of files. Before you begin working with the program, make a folder for the bank download files, and move the bank files that you intend to process to that folder. Then, make a separate folder for processed files and leave it empty.

If you simply use the default Windows “Downloads” folder, you can leave the files there. When you select the user input file, use the list of system folders to the left of the directory listing to select “Downloads.”

Using The Program

First startup

When CSV_Processing is first started up, you will see a pop-up menu. An example is shown as Figure 1. In addition to the menu, you will see a program menu and a contained text window. The text window contains routine messages written as the program executes, and constitutes a run log for a given session with the program. You can save this log as a text file at any time.

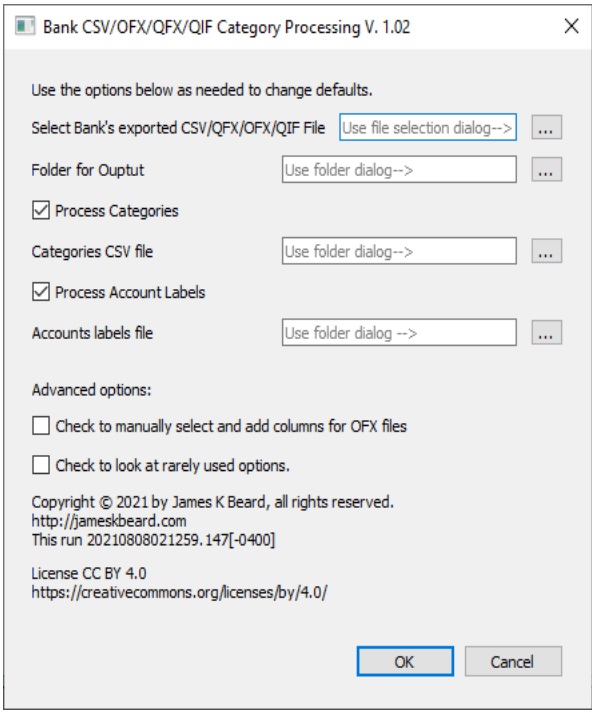

Figure 1: Opening Dialog on First Run

Note that there is a prompt in each of the text entry fields. The text entry fields for files and folders all prompt you to click on the icon with an ellipsis to the right of the window to use a File Manager window for the process. Of course, you can type in a file name if you prefer.

For the program to operate, you must enter an input file and an output location. Use the ellipsis icon to the right of the first field and navigate to the folder where you have collected the bank download files, and select the one that you want to process. Click on the ellipsis icon to the right of the second field and navigate to the empty folder that you made for processed files.

If you have set up tables for categories and subcategories, use the third field to find and select that file. If you have not done that, or do not wish to use the program to generate categories and subcategories, uncheck the box marked “Process Categories.”

If you have set up tables for account identification, use the third field to find and select that file. Or, if you do not wish to use the program to identify accounts, uncheck the box marked “Process Account Labels.”

The remaining fields are self-explanatory and add generality. The defaults are defined to be OK for most use of this program. Inspect them and change any that you find are important for the program to deal properly with your bank’s downloaded files. Then, click OK.

The program will generate a file, CSV_Processing.ini, in the same folder as CSV_Processing.exe, that saves the files and locations that you have entered. On subsequent runs, these locations and file names will be the defaults. The dialog after the first run will be similar to the example shown as Figure 2. The graphic on the front page is a larger version of Figure 2.

The following sections describe the options of the opening dialogue of Figure 1 and Figure 2.

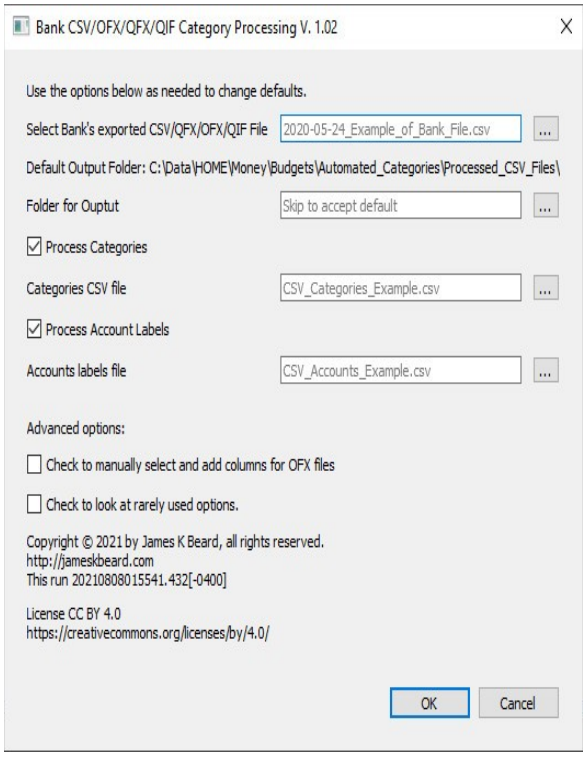

Figure 2: Opening Dialog After First Run

Select Bank's Downloaded File

Click the ellipsis icon to the right of the text entry field to open a file manager window. Navigate as you would with any navigation window to select the bank’s downloaded file to process and select it, then click OK. The bank’s downloaded file may be in any of the formats in Bank Formats that This Program can Read below. The program remembers this setting in CSV_Processing.ini.

Select the Folder for Output

Click the ellipsis icon to the right of the text entry field to open a file manager window. Navigate as you would with any navigation window to select the folder that you created for the outputs of this program and click OK. The program remembers this setting in CSV_Processing.ini.

The Categories Table

Leave this box checked if you want to process categories. Unchecking this box will omit adding columns to the output file for categories and subcategories. This box will be checked when you start the program.

If the Process Categories box is checked, the program will need a table of snippets of transaction fields that identify each set of category and subcategories. Perform this entry by clicking the ellipsis icon to the right of the text entry window and navigate to the categories table, select it, and click OK. The program remembers this setting in CSV_Processing.ini. See Eliminating Duplicate Category Assignments below for details on how to set up this table and make it work for long lists of transactions.

The Account Identification Table

Leave this box checked if you want to process account labels. When your bank allows you to download data on multiple accounts as one file, you will need this capability. If your bank data does not supply account numbers or other identification, you can uncheck this box and the program will not add an Account ID column to the output file. This box will be checked when you start the program.

If the Process Account Labels box is checked the program will need a table of identifiers of different accounts. Click the ellipsis icon to the right of the text entry box, navigate to the CSV file where you have stored the Accounts table, and click OK. The program remembers this setting in CSV_Processing.ini.

Advanced Options

The following options were put in with the first program versions but have not proven to be necessary with the bank files encountered to date, or have not seen any use beyond testing. They have been moved to a second dialog, which is only displayed if you check the “Advanced options” checkbox. If you use any of them, please let me know so that I don’t delete them in future versions.

Tab as Field Delimiter?

Some programs have output files that use tabs as field delimiters, in place of commas or in addition to commas. I have not seen a bank file that does this, but the program has the capability to recognize tabs as field delimiters for cases where you use files from a source other than a bank.

Other Field Delimiter Characters?

Rarely, a program will use a field delimiter other than a comma or a tab character. If you encounter a file that does this, you can add the characters here. Do NOT use the semicolon character, “;” because the semicolon is used by standard spreadsheet functions as a delimiter between arguments.

Semicolons are used by this program for its output; we give these examples:

- We use the spreadsheet function DATE(yyyy,mm,dd) to translate between whatever the bank file provides and a standard spreadsheet date, for example.
- Also, when the program encounters a comma in a Payee or Memo field, it replaces it with a semicolon so that the field will be loaded by your spreadsheet as a single field.

When a bank uses a semicolon to avoid breaking up a field, adding the semicolon as a delimiter will cause this program to break up the field anyway. Some banks use semicolons as delimiters in the Payee, Memo, or other fields. If a bank uses the DATE function or other standard spreadsheet function that uses semicolons as argument delimiters, adding them here as delimiters will cause this program to break up the spreadsheet function into multiple columns.

Prefix output filename with today's date?

A good way to make sure that folder listing of files puts the current output at the head of the list is to prefix the file name with today's date. That is done automatically by this program, with the added delimiter "JKB_" to avoid concatenating two dates, when you prefix your file names with the current date (as I do). This box is checked when you start the program.

Embed Long Numbers w/o Decimal as Text?

Sometimes numerical fields longer than ten or so digits, such as credit card numbers and some account numbers, are interpreted by spreadsheets as numbers and are converted into floating point numbers with a maximum of 14 or 15 digits. If such a field has more digits, the trailing digits are lost. This capability imports bank data into the standard spreadsheet "interpret this as text" function '=T("<long string of numbers>")' so that the output field is imported into your spreadsheet without destroying data. This box is checked when you start the program.

Case Sensitive Snippet Checking?

Snippets of text seen in bank payee or memo fields are used to recognize them and assign Categories by this program. To simplify usage, this text comparison ignores the difference between upper case and lower case. If you want to use case-sensitive snippets of payee or memo fields in your Categories table, check this box. This box is unchecked when you start the program.

Bank Formats that This Program can Read

This program can read all formats available from most banks for their checking and savings accounts. See Table 2 for a list of the formats supported by the program.

The output of this program is in comma-separated value format, which is loaded directly by all spreadsheet programs.

Table 2: File Formats Accepted by CSV Processing Program

File extent	Format	Remarks
CSV	Comma-separated value	Always available to , but not always simple to use in your spreadsheet program
QFX	Quicken for Windows, Mac	Intuit, Inc. format, uses a header of a few lines followed by an OFX file
OFX	Open Financial Exchange	An XML format very commonly used by banks and other financial institutions
QIF	Quicken Interchange Format	Used by Quicken until 2003, still used by Microsoft Money

Although there is a difference between files for Quicken for Windows and for Mac, and Quicken programs for one platform will not read those written for the other, this program reads either format. The simplest format is QIF. The most comprehensive is OFX, which is also used by QFX after a Quicken-specific header of a few lines of text. OFX is Open Financial Exchange, an extensible markup language (XML) that is formulated for the financial community and is used by nearly all banks and financial institutions in storing and exchanging data of many types, including transactions. The OFX standards are published online at

https://www.ofx.net/

One might think that asking the bank for CSV files would be the simplest way to go for this program, but this is not the case at all. There is no published standard for bank-provided CSV files, and the only requirements are that they be loaded by spreadsheet programs, then display and print properly. To make this happen, when some fields important to us in identifying categories and subcategories contain commas, banks use *ad hoc* methods such as enclosing all fields in quotes, deliberately nonstandard date formats that spreadsheets recognize as plain text but do not recognize as dates, etc. I have found that processing CSV files from banks requires a complete interpretation subprogram for each bank. Three subprograms are included in CSV_Processing and CSV from your bank may or may not be fully supported by the program. If you try the CSV program and you have a problem with it, please report the problem, including the URL of the bank web site where you downloaded the CSV ffile, and get another format from the bank.

The preferred file format is OFX. If your bank does not offer OFX data, look for Quicken (QFX) data instead. This program will simply echo the Quicken-specific text lines at the beginning of the program to program log, ignoring them otherwise, and read the OFX in the Quicken file for you. If your bank offers QIF, you might try that format.

As a note for the geeks among us, this program treats OFX data generally: a first reading of the file takes milliseconds to generate a database of all the structures in the file, and the fields in those structures. This data, compiled every time this program is run on a file with OFX data, is used in a

second pass by this program to generate information that is used in making up the output transaction data. Thus, this program is not dependent on the standard as published, or how the bank uses OFX to make up its transaction summaries.

This program supports data encountered in bank downloads for customer transactions, but does not support all possible OFX functions. For example, OFX supports images, but this program does not.

Using the Output

One the program runs and the ambiguities in category matches are resolved, the text window, which contains the program log, will be similar to Figure 3. The log begins with all the program inputs, the full pathnames for the Categories and Accounts CSV tables and the number of rows in each, the headings to be used in the output CSV file, the headings processed for a match with categories, the augmented list of headings in CSV format, and the final message “Normal exit.”

The categories in the table used in the figure is real, and was arrived at in producing a table that worked for over 1000 transactions from multiple types of accounts.

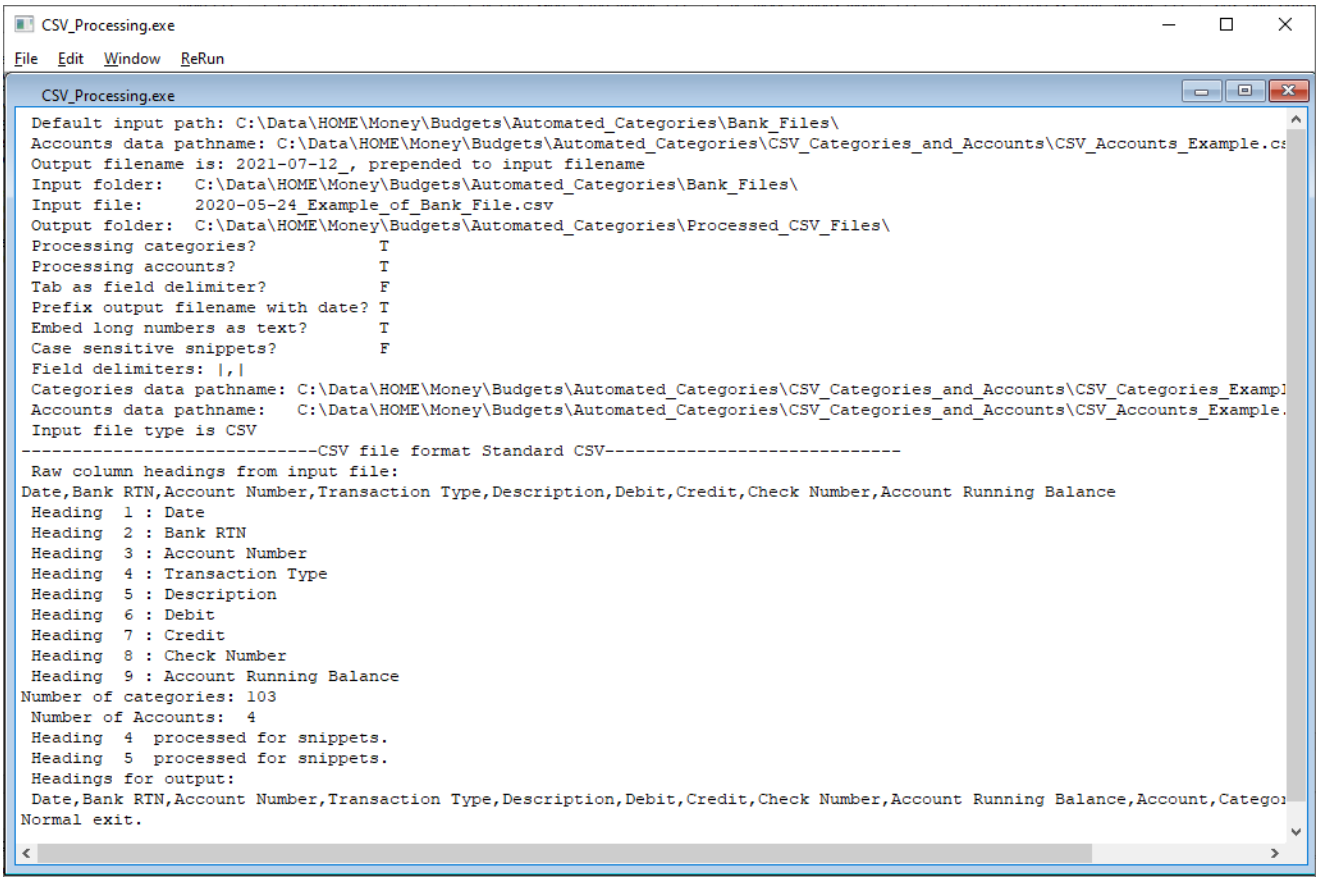

Figure 3: Program Log for a Normal Run

Loading the Output Files into a Spreadsheet

To use the output, navigate to the folder where the program stores the output files and double-click on the file that you want to look at. It will be of the form yyyy-mm-dd_JKB_<input file name>.csv, the current date and a delimiter “JKB_” between the current date and the input file name. Your default spreadsheet will load the file.

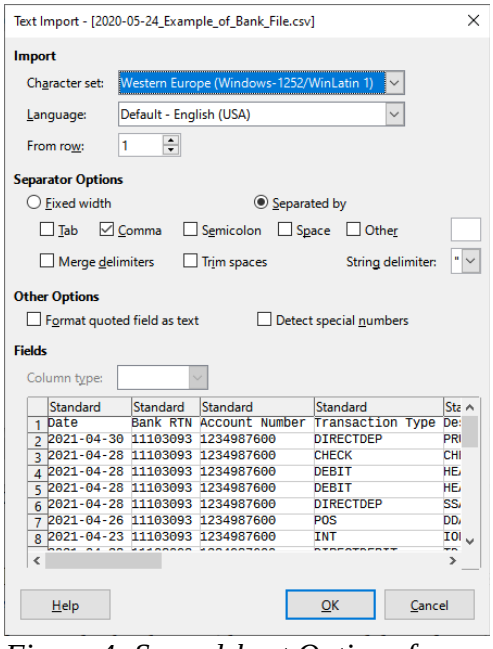

Figure 4: Spreadsheet Options for Importing CSV Files

Be sure and look at the spreadsheet options before loading any CSV file, particularly for the first time. See Figure 4 for the option window for LibreOffice Calc; others will be similar. Note the top two fields asking what language to use, select a one-byte character set and English as the language. The character set “Western Europe (Windows-1252/WinLatin 1)” with the language set as “English (USA)” will import plain text CSV files as legible spreadsheets, and, when saved, will not convert them into two-byte characters that this program does not read properly. Find the Separator Options and make sure that

Semicolon is unchecked, because semicolons are used as delimiters within fields by this program and some bank transaction files that this program processes; if you load a CSV file with this option checked, it will replace semicolons with commas and treat them as column field delimiters, and when you save the file again, the semicolons will have become commas, and thus field delimiters. You may recover by loading the file in Notepad or other text editor and manually changing those "bad" commas back to semicolons.

The date is in an international standard format common to all modern spreadsheets, ISO/IEC 26300:2-2015, which is the number of days from an arbitrary starting date, apparently January 1, 1900. The program translates whatever format the bank provides to a spreadsheet function "=DATE(yyyy;mm;dd)" which produces that number in column A, which you will see when you first load the file into a spreadsheet. The first thing you will do is to convert Column A into dates. You do this by clicking on the column header, the letter A, which highlights the column. Then, you open Format → Cells, select the Numbers tab on the settings window, select Date as the format, and click OK. See Figure 5 for what you will see while doing this.

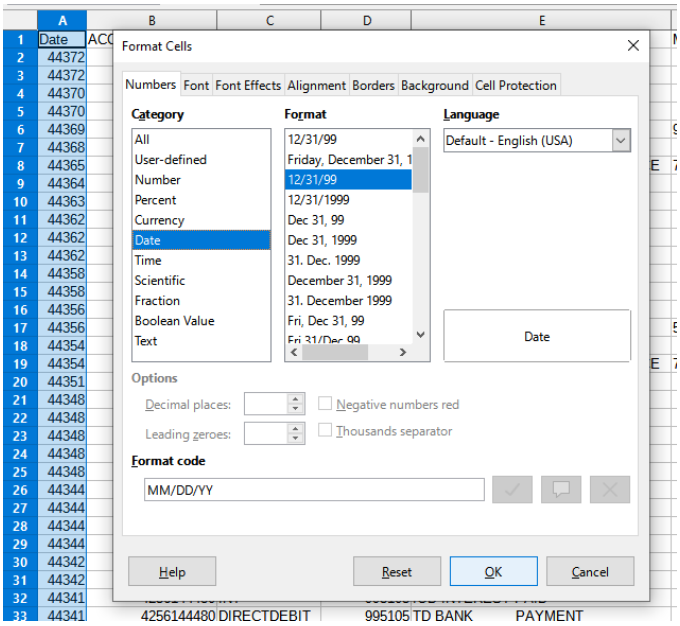

Figure 5: Converting the Date Display in Column A

The date format will be wider than the numbers produced by the DATE function, so the column will seem to be all "###" after you do this. To widen column A just enough to show the dates properly, put the cursor on the line between the A and B columns and double-click.

The amounts of the transaction will be decimal numbers. You will want them to be displayed as currency. To do this, select the column or columns, and use File → Cells, then, on the Numbers tab, select "Currency" as the format, and click OK.

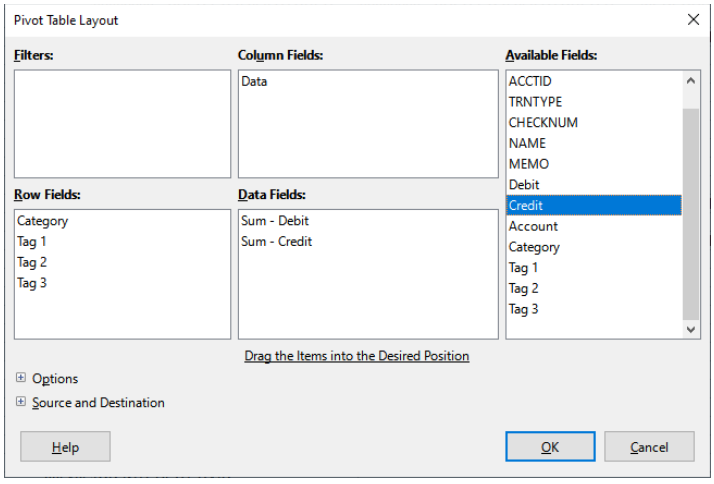

Figure 6: Simple Pivot Table Setup

At this point you have a spreadsheet showing your processed bank data.

Producing a Pivot Table of the Data for Reports

The simplest way to make a report from the spreadsheet data is to use a pivot table. Once the spreadsheet is open, and dates and currency are formatted properly, put the cursor on any column heading and choose Insert → Pivot Table. A window will open asking you to select a source; the default should be "Current selection" which is fine, so select OK. This will bring up the pivot table configuration window. See Figure 6 for the pivot table setup window. On the right, there is a column that will list all the columns in your spreadsheet file. Use the mouse to drag Category, Tag 2, Tag 2, and Tag 3 to the Row Fields window. The column Fields window will have Data in it by default. Drag Debit and Credit (or Amount, or what you use for the currency field or fields) to the Data Fields window. You can ignore the Filters and Column Fields windows. Click OK. The spreadsheet will produce another sheet that contains the pivot table. An example is shown as Figure 7. Note that there is a large amount in the bottom row, where the Category field is empty. The following Section addresses how to fill in Categories for transactions for which the bank does not provide information to support assigning categories.

	A	B	C	D	E
1				Data	
2	Category	Tag 1	Tag 2	Sum - Debit	Sum - Credit
3	Amazon	Digital	(empty)	$22.39	
4		(empty)	(empty)	$234.96	
5	Cars	Gas	(empty)	$50.91	
6		Sirius XM	(empty)	$22.00	
7	Computer	Software	(empty)	$138.60	
8	Entertainmen	Amazon	PPV	$56.13	
9		Movies	Tickets	$20.62	
10	Financial	Credit Card	(empty)	$123.45	
11		Income	Prudential		$11.11
12			Social Security		$65.44
13		Interest	(empty)		$0.07
14	House	AT&T	(empty)	$22.56	
15		COMCAST	WWW	$128.95	
16		Food	(empty)	$130.78	
17		Hardware	(empty)	$77.80	
18		Magazines	Readerâ€™s	$17.98	
19		Maintenance	(empty)	$95.94	
20		News	WSJ	$134.97	
21		Security	(empty)	$65.01	
22		Supplies	(empty)	$65.37	
23		Utilities	Gas	$55.55	
24			Municipal	$2.35	
25			PSEG	$100.00	
26		(empty)	(empty)	$87.22	
27	Medical	Health Insura	AmeriHealth	$222.22	
28		Insurance	Drugs	$47.53	
29	(empty)	(empty)	(empty)	$788.94	$1,000.00
30	**Total Result**			**$2,712.23**	**$1,076.62**

Figure 7: Pivot Table for Categories

Financial reporting often takes the form of four columns: annual budget, budget for period such as month or quarter or year-to-date, actual for that period, and over/under budget. Also, credits, deposits or revenue is presented first, with debits, expenses, or disbursements presented as a separate set of columns. Here, you can select the rows of your output CSV file for all dates in a given reporting period and produce actual figures for any period for which you have bank transaction data. The program always separates them so you can produce separate tables which produce outputs with pivot table results like those shown in Figure 7 with only the credits or debits shown. Copy or reference the pivot table results to form your report pages.

	A	B	C
1		Data	
2	Account	Sum - Debit	Sum - Credit
3	Credit Card	$1,035.99	
4	Harriet Checking	$941.63	$0.02
5	Ozzie Checking	$734.61	$1,076.59
6	Savings		$0.01
7	**Total Result**	**$2,712.23**	**$1,076.62**

Figure 8: Pivot Table for Accounts

You can make other pivot tables to show different summaries. See Figure 8 for a pivot table for just Accounts in the Row Fields box.

Making the Categories Work for You

The subsection immediately preceding mentions that bank transaction data may not be sufficient to assign categories. This normally happens with deposits, transfers, and with any transaction where details of the payee or transaction are not supplied to the bank. We will begin by showing how to flesh out blank category and subcategory fields in your spreadsheet.

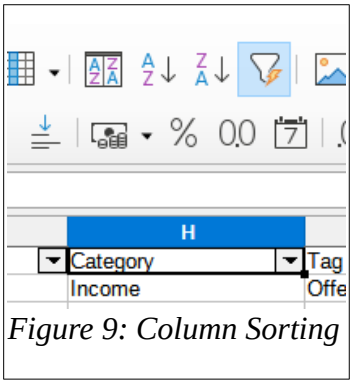

Figure 9: Column Sorting

Setting Up the Column Selection

You can easily set up the spreadsheet to display only rows of interest for a specific task, such as fleshing out blank Category fields. Here are a few examples that you will find useful.

The most important is to set up the columns for selecting Category fields to display. This is done by placing the cursor in one of the headings, then clicking the small funnel icon in the spreadsheet toolbar. See Figure 9 for the funnel toolbar, optionally with the tiny lightning symbol at the bottom, which is highlighted in the figure. The down-arrow icon appears on each heading, as shown in the figure. Clicking on the down-arrow icon produces a drop-down menu of all the items in that column, beginning with “blank” for columns with no entry. See Figure 10 for an example of the column selection drop-down menu. To select all rows having a given entry, uncheck All and check the box for that entry.

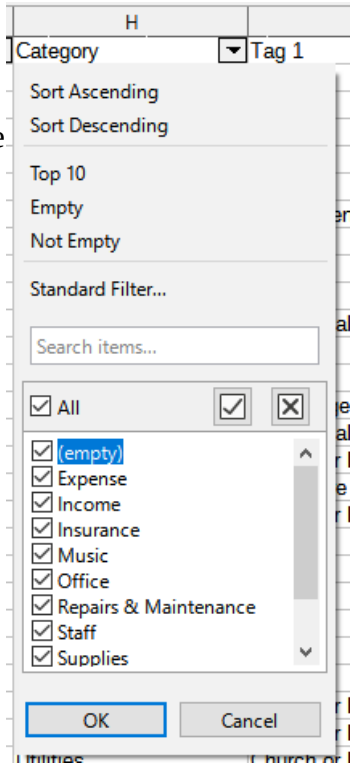

Figure 10: Column Selection

Making the Headings and Dates Always Visible

Under View → Freeze Cells, select Freeze First Row to make the headings visible as you scroll down the spreadsheet past the first page of rows. Select Freeze First Column to make the dates visible as you scroll right to view the Categorizes and subcategories.

Filling In blank Category Fields

What we need here, to fill out the Category for transactions where the bank has insufficient data to assign a category, is to look at all rows that have blank entries. So, we uncheck All and check “empty” to display only those rows.

At this point, we will have all rows displayed which so not have categories assigned. You then add categories and subcategories manually. Information on checks can be obtained from most banks by displaying Activity or Transactions, then clicking through a check number or transaction amount to display the check. Other information must be obtained from those who made transactions for which the bank has no digital data.

Eliminating Duplicate Category Assignments

The process described in this paragraph is the same job that you must do with Quicken outputs every month: set up the program to recognize the Categories and tags that you desire, working from the bank's information as provided. This program has additional capability so that when you have done it once, occasional touch-up is needed but the only transactions that you must address are those for which the bank provides no information, such as cash deposits.

The most detailed part of using the program for assigning categories is dealing with multiple category matches for a single transaction. For many situations this will not occur. As you deal with larger numbers of types of transactions and larger numbers of transactions, this will eventually happen. This program provides a simple, powerful set of tools to control assignment of Categories that will support successful unique and appropriate assignment of categories for these situations.

The program works by using snippets of text found in either of two fields in the bank's data. These two fields are usually Name or Payee and Memo or Information. Your first run after defining these snippets in the Categories CSV file may inadvertently produce a match with more than category/subcategory rows.

```
Multiple categories in row 49, saved headings: AMZN DIGITAL*3M9NG73Z3 88888-802-3080 WA
Cat row 2, Snippets |AMAZON;|AMZN;|AMAZON, Category Amazon, Tags Digital,,
Cat row 3, Snippets |AMAZON;|AMZN;|AMAZON;|LIVE, Category Amazon, Tags Live,,
Cat row 4, Snippets |AMAZON;|AMZN;|AMAZON;|DIGITAL, Category Amazon, Tags ,,
Multiple categories in row 54, saved headings: Amazon Prime*062GK4OA3   Amzn.com/billWA
Cat row 2, Snippets |AMAZON;|AMZN;|AMAZON, Category Amazon, Tags Digital,,
Cat row 3, Snippets |AMAZON;|AMZN;|AMAZON;|LIVE, Category Amazon, Tags Live,,
Cat row 4, Snippets |AMAZON;|AMZN;|AMAZON;|DIGITAL, Category Amazon, Tags ,,
Multiple categories in row 56, saved headings: AMAZON.COM*EI5ON9L93 AMZNAMZN.COM/BILLWA
Cat row 2, Snippets |AMAZON;|AMZN;|AMAZON, Category Amazon, Tags Digital,,
Cat row 3, Snippets |AMAZON;|AMZN;|AMAZON;|LIVE, Category Amazon, Tags Live,,
Cat row 4, Snippets |AMAZON;|AMZN;|AMAZON;|DIGITAL, Category Amazon, Tags ,,
Multiple categories in row 59, saved headings: AMAZON.COM*3K7Q006Y3 AMZNAMZN.COM/BILLWA
Cat row 2, Snippets |AMAZON;|AMZN;|AMAZON, Category Amazon, Tags Digital,,
Cat row 3, Snippets |AMAZON;|AMZN;|AMAZON;|LIVE, Category Amazon, Tags Live,,
Cat row 4, Snippets |AMAZON;|AMZN;|AMAZON;|DIGITAL, Category Amazon, Tags ,,
Multiple categories in row 61, saved headings: Amazon.com*3H5RI5UP3     Amzn.com/billWA
Cat row 2, Snippets |AMAZON;|AMZN;|AMAZON, Category Amazon, Tags Digital,,
Cat row 3, Snippets |AMAZON;|AMZN;|AMAZON;|LIVE, Category Amazon, Tags Live,,
Cat row 4, Snippets |AMAZON;|AMZN;|AMAZON;|DIGITAL, Category Amazon, Tags ,,
Normal exit.
```

Figure 11: Program Log When Multiple Categories are Matched

Whenever the program finds more than one match, the text window will have an output that gives the spreadsheet row of the bank data, then the spreadsheet row of each category match. An example of this output is shown in Figure 11.

This program provides a very simple and powerful way for you to resolve a match with multiple category/subcategory matches: you can use “must” snippets, and “not” snippets. The ones we start with are called “or” snippets. The way the program uses these is summarized in Table 3.

Table 3: The Three Types of Snippets

Type	Starting Character	Function
OR	NONE, or vertical bar “|”	Any occurrence in the field of the bank file allows a match
MUST	Plus sign “+”	When used, OR snippets are ignored, and each and every MUST snippet must be present for a match
NOT	Backslash “\”	If found, there will not be a match regardless of OR or MUST snippet matches

Note that the program output for multiple matches adds the vertical bar “|” when there is no starting character. You may use the vertical bar in your category snippet if you like.

Category snippets may include any character, even blanks, except the semicolon “;”. The semicolon is used as a delimiter between the snippets.

Given the text in the two fields, and the snippets and categories, you add MUST or NOT snippets to the fields to ensure that only the correct one is matched with the correct bank transaction.

You use the three types of snippets to resolve multiple category matches by

- Adding a NOT snippet that matches the wrong transaction but not the correct transaction, or
- Adding MUST snippets or making OR snippets into MUST snippets by adding a plus sign “+” as the first character to force a match with the correct transaction but not with the incorrect transaction.

There is another point that is important to use when improving your snippet lists: this program usually evaluates two fields of bank data. If a snippet match is found for *either one* then this program will declare a match. So, examine both fields in the program log when fixing duplicate associations.

Fix only one or two duplicate matches at a time, then re-run the program. This allows you to test your new snippet list, and you will find that a fix often eliminates more than one duplication.

Note that Figure 11 shows that simply adding OR snippets for LIVE and DIGITAL does not prevent a match for all three categories for any transaction where AMAZON or AMZN matches text in a description field. Using MUST and NOT snippets as shown in Table 4 resolves the ambiguities. The OR snippets, disabled by use of MUST snippets in two rows, were left alone.

Table 4: First Three Category Rows That Resolve Ambiguities

Category Snippets	Category	Tag 1
AMAZON;AMZN;+Digital;\Live;\Mktp;\Prime;\BILLWA	Amazon	Digital
AMAZON;AMZN;+Live;\Digital;\Mktp;\Prime;\BILLWA	Amazon	Live
AMAZON;AMZN;\Digital;\Live	Amazon	

Resolving Problems

The Program Does Not Start

If you downloaded and extracted the program from the ZIP archive, run any 64-bit version of Windows, and double-clicking on the program does not bring up the opening dialog shown in Figure 1 on page 8, you probably do not have the Qt DLL files in the proper place. Review the section Installing This Program beginning on page 5, and make sure that all the files in Table 1 on page 6 are in place, and try again.

If the program still does not start, please send me the URL for where you got the ZIP file, information on your computer (Start → Windows Administrative Tools → System Information, then File → Export As → Text file). If you are running something else, such as WINE under Linux or OS/2, ask the source of the environment that you are using for support. The email address jkbeard@jameskbeard.com will get to me

CSV Files Scrambled in the Spreadsheet?

If you load a CSV file gets scrambled in your spreadsheet program, do not save the file. Try these things first:

- Exit the spreadsheet program without saving the file, then double-click the file or open it from the spreadsheet program, and look at the CSV options screen of Figure 4 on page 8. Make sure that the Character Set is a single-byte (English, Western Europe, Windows code page 1252, ISO 8859-1) and the language is set as English (US) or equivalent.
- Make sure that the radio button for Separated By is checked, the box for Comma is checked, and the button for Semicolon is NOT checked.
- For more detail, review the instructions for loading CSV files in the section Loading the Output Files into a Spreadsheet beginning on page 14.

If semicolons are interpreted as field delimiters and you have saved the file, run the program again if it is the program output file. If you have scrambled a Categories table, you can load it in Notepad by opening Notepad and dragging the file icon onto the Notepad window, or by right-clicking the file and selecting “Open with” and navigating to Notepad. There, you can change the improper commas back

to semicolons. If the file is large and there were many semicolons, it may be quicker to change all the commas to semicolons and then to change errant semicolons back to commas.

If you have saved a file in two-byte character format, the program will interpret every other character as a NUL, which is ASCII zero. Then, none of the text comparisons in the program will work. Try loading it into the spreadsheet and using Save As to get it back into single-byte text format. This can happen without the user noticing and be difficult to reverse. If this turns out to be a common problem, I will modify the program to recognize this and discard NUL characters. This never occurs in files downloaded from banks to US locations; if you have scrambled a bank download file, download it again before you spend a lot of time working with a scrambled version. In general, you should never overstore a downloaded file.

Mystery Category Matches?

The use of snippets with different logical effects on matching fields in bank transaction information is powerful and simple, but may be difficult to follow at first. A good step-by-step process to recover from attempting to control category matches with long strings of snippets with unpredictable results is

- Copy the existing strings of snippets to a backup text file and delete them from the Categories table, and start over with one or two simple keywords.
- Run the program again with your simplified Categories table. With your Categories table also loaded in your spreadsheet program, review the text window information for each duplicated category assignment; review the section Eliminating Duplicate Category Assignments that begins on page 18. Then, use a little thought to select a NOT snippet for wrongly associated categories for just one case. The MUST snippets capability can help with difficult cases.
- Don’t forget to use <Ctrl>S to save the Categories table after each change so the program will see your changes.
- Run the program and reexamine the text window to observe the effects on that one case after each change in the Categories table. If you have more than two MUST or NOT snippets, review what you have and see if you can simplify your snippets.
- Once you have resolved multiple assignments for one case, you can use what you have learned to address the remaining cases in the text output file. Continue to address just one multiple assignment at a time.

Need Help with Spreadsheets?

If you need help in getting started with spreadsheets or if you want to learn more than the few step-by-step operations in this document, there are many options. Most of them involve Microsoft Excel

because of the huge market share, and learning Excel will work for you when you use other spreadsheets.

There are many excellent tutorials for spreadsheets and for pivot tables available online and in bookstores, usually priced very moderately.

Your local community college or municipality will offer free or inexpensive in-person or Zoom live classes in spreadsheet usage.

License, Bug Reports, and New Versions

Creative Commons License Attribution 4.0 International (CC BY 4.0)

This program, auxiliary files included in the ZIP archive, and this User Manual, are copyrighted and licensed to protect you and me from patent trolls and others who would claim someone else's work as their property, copyright it themselves, and prevent its free use by you and me. Originally written for personal use, I found this program to be a valuable time-saver in church work. I hope you find it useful for home, non-profit organization, or small business budgeting and financial reporting. If you have a novel use of this program, let me know so that I consider your needs when updating the program.

A legal disclaimer is required to protect people who allow others to use their software from trolls who look for ways to extract money from anyone.

THIS PROGRAM AND MANUAL ARE PROVIDED "AS IS" WITH NO GUARANTEE OF SUITABILITY FOR ANY PURPOSE. BY DOWNLOADING THIS PROGRAM, YOU AGREE THAT YOUR USE OF THIS PROGRAM IS ENTIRELY AND COMPLETELY AT YOUR OWN RISK. IN USING THIS PROGRAM, YOU AGREE TO HOLD NO ONE RESPONSIBLE FOR ANY RESULTS OR CONSEQUENCES OF YOUR USE OF THIS PROGRAM.

Error Reporting

Continued improvement of this program depends largely on finding and fixing bugs and errors. I'm just one user, and exercise this program with files from just a few banks. If a score or more users work with about ten banks, we are likely to find nearly all the bugs and errors. Please report the bugs to jkbeard@jameskbeard.com or jameskbeard@comcast.net. In the bug report, please save the text window file and attach it, and include enough information so that I can reproduce the error. If you

include a text file that contains confidential information, feel free to edit it to eliminate private information. To reproduce the error, I will need:

1. The run log, which you obtain by saving the text window before exiting the program. The last lines in it will tell me what error occurred, and what software procedure encountered the error.
2. The input files, which include the bank file, the Categories file, and the Accounts file. Feel free to edit the files to eliminate confidential data, such as $1,420.65 becomes $123.45, and "TD Bank" becomes "West Bank of the Mississippi." Note that for debugging, I just need one single row or XML block, the one that caused the error. Try deleting all other rows and if you can produce the error with just the heading row and one data row, with the numbers edited, send that.
3. The error message, or whatever happened when the problem became apparent. Please be explicit and complete here.

If you encountered an error and exited the program without saving the text window, run the program again and, when you get the error, save the text window before you exit. If the program crashed and the text window closed by itself, simply tell me that.

Please understand that whatever can be done about a bug or error depends on being able to reproduce the problem, and also on the amount of time available to spend on the program. Remember that this program is free and comes without guarantees or obligation for fixes and upgrades.

New Versions

Absoft Pro Fortran is the compiler used for developing this program, and it uses a proprietary graphical user interface (GUI) environment, which they call Absoft Windowed Environment (AWE). The AWE environment provides some access to Qt capabilities which this program uses to provide a graphical user interface (GUI). Absoft makes compilers for Windows, Linux, and Mac. I have the compilers for Windows and Linux but not Mac. This is an expensive compiler and I have no personal use for writing Mac applications, so I have no plans to produce a version for Mac. I will produce a version for Linux if people request it.

I compiled the current program as a 64-bit Windows application, which will not run on 32-bit Windows. Since 32-bit Windows has become rare, this should not be a problem for anyone. But, there is nothing that prevents compilation of this program as a 32-bit application. If anyone needs a 32-bit version, please let me know.

The program produces CSV files that load in spreadsheets to produce columns of fixed format, with minor variations for different input file formats. If you input CSV format, the column headings used by the bank in their export are used without change. This program accepts all common file formats that are available by download from banks; see Table 2 on page 6 for a list. The QIF format defines eleven

headings which are all used in the output, plus Account ID, Category, Tag 1, Tag 2, and Tag 3. The OFX formats define 25 or more headings, of which this program provides six: Date, Account ID, Transaction Type, Check Number, Payee, and Memo, and adds Bank ID, Category, and Tags. I can add user control of the headers by a series of menus, with some possible usefulness for the OFX formats. For the current release, simplicity of use prevailed over flexibility. If you use QFX or OFX files and want to go through a list of 25 to 50 possible column headings and check which ones you want to see in the output and in what order, let me know, and I will consider it.

Epilogue

This work uses a commercial Fortran 2018 compiler from Absoft. A proprietary API provided as part of the compiler, Absoft Windowed Environment (AWE), supports the dialog and text window used for the graphical user interface (GUI) that is so important for making this program accessible to all. The GUI widgets are provided through Qt libraries. The Qt libraries are compiled and included with the application to avoid "dependency hell" problems with Qt installations or compatibility with different versions of Qt installed on different computers. The word processor and generator of the PDF file for this User Manual is LibreOffice Writer. The spreadsheet used in the screen shots is LibreOffice Calc.

Versions

Table 5: Versions and Updates

Date	Version	Remarks
July 13, 2021	1	Initial Release
July 24, 2021	1.01	User interface simplification
August 8, 2021	1.02	Allows Quicken, OFX files to not have line breaks
August 23, 2021	1.03	Tolerates QFX/OFX files that don't use CHECKNUM