

2007 Rexx Language Association Symposium

Forage With a Dinosaur: A beginner discovers how to use ooRexx and Excel Automation to solve a Real Problem

Les Koehler
03 May 2007

Table of Contents

[Abstract](#)

[The Problem](#)

[The Solution - Phase 1](#)

[The Solution - Phase 2](#)

[The Solution - Phase 3](#)

[The Solution - Phase 4](#)

[Follow Up](#)

[Conclusion](#)

Abstract

I will present the problem of making financial reports more usable by the board of directors of the Home Owners Association, where I am the treasurer.

I will then show how I discovered and used ooRexx and Excel Automation resources to solve the problem.

The presentation will include a demonstration of the programs that I developed, some of the insights I gained about using ooRexx, the value of Excel Automation and discussion of some of the code problems and solutions.

The Problem

After we changed management companies, we discovered that the software being used wasn't capable of selecting fonts for its financial reports.

With most of the directors being over age 60 and with diminishing eyesight (me most of all!), reading the reports was very difficult.

We investigated the management company trying to export the reports to Word and Excel, but the results were not satisfactory.

I finally got the idea of using "Print to File" so I could at least see the raw data. They tried that and it worked for the first report I was interested in. Cash12 Report.PRT

The Solution - Phase 1

Having attended several of Lee Peedin's presentations on Excel Automation and Rony Flatcher's tutorials and presentations on ooRexx, I hoped that I could finally put it to use and actually **learn** some of this new-fangled stuff!

A note to Lee on Friday (17Nov06) asking for a pointer to his Excel Automation web page and any advice he might have yielded a quick reply and a query for a sample of the problem.

I sent that off and he replied that it was an easy problem to solve ("Should just take a little bit of coding") but he had a busy weekend and would get to it as soon as he could.

Saturday morning (18Nov06) he sent me what I've renamed to for_les_prt.rex.bak which produced (position to row 49) G5401087.xls

and I replied (19Nov06) with

Wow!! That's terrific!! I had to twiddle with it just a little and since I don't yet grock ooRexx. It's a smidgen ugly... but it works. You'll readily see my changes... our code styles are very different. I like my code to be very dense since I use larger fonts than most folks and I want to see as much code on one screen as I can. Feel free to show me the ooRexx way!

for les prt.rex

I had fixed the problem of a trailing minus sign to make it a leading one, but I had used Classic Rexx.

Lee replied (20Nov06) by sending me what I now call for les prt2.rex.bak to show me the ooRexx way of doing it and added some code for controlling the margins.

As a bonus, he also reformatted the code to my style!

After some further testing, I decided that I didn't want boxes drawn around column one titles if there was no data to the right. That led to (21Nov06) for les prt2.rex

Finally (much later), I added:

- Month and year argument
- An HOADATE function as a ::Routine
- Change the month abbreviations to mixed case
- Print option
- Added KILLEXCEL from Lee's Excel_Automation901 as a ::Routine
- Added a TITLES ::Routine function to ensure that row titles contain both an account number and description.

which resulted in (12Dec06) Cash12ToExcel

Which is what I now use on a monthly basis.

The Solution - Phase 2

The next report to work on was the Balance Sheet. The management company figured out how to export that to Excel, but I wasn't satisfied with the layout Aug_06BalanceSheet.

What we needed was:

- Simple columns
- A Net Change column so we could see what had changed from the previous month.
- A "cross-check" to see if Assets and Liabilities match internally.

Obviously, this was going to be a LOT more complicated than simply modifying existing code! A note to Lee for pointers yielded:

An existing Excel workbook can be easily manipulated with ooRexx. Some basic examples are on my webpage <http://pragmaticlee.safedataisp.net> You'll see that they progress from creating a new workbook, to opening/reading/changing an existing workbook. If you have any specific questions, please send them my way.

With that wealth of information available, I downloaded everything that looked interesting and got to work.

Changing to simple columns was straightforward. A simple modification using Cash12ToExcel as a base did the trick.

Oops! The numbers were in whole dollars... the cents had disappeared.

Turns out that there are a lot of nitty-gritty details that you have to tell Excel. Lee provided some examples:

```
-- format a single cell to display 2 decimal places
xlobj~Range('A1')~Select
xlobj~Selection~NumberFormat = '#0.00'

-- format a range of cells to display 2 decimal places
xlobj~Range('B1:B3')~Select
xlobj~Selection~NumberFormat = '#0.00'

-- format an entire column to display 2 decimal places
xlobj~Columns('C:C')~Select
xlobj~Selection~NumberFormat = '#0.00'

-- format an entire column to display "currency" 2 decimal places
-- and commas as thousands marker
xlobj~Columns('D:D')~Select
xlobj~Selection~NumberFormat = '$#,##0.00'
```

With that information, I coded what I wanted and now had Oct06_BalanceSheet.xls

The next requirement to address was how to link cells between spreadsheets. The answers to this, and all the following Phases, lay in Excel_Automation012_014.txt and Excel_Automation17_18.txt.

And, of course, the accompanying code:

[RxXL Automation 012.rex](#)
[RxXL Automation 013.rex](#)
[RxXL Automation 014.rex](#)
[RxXL Automation 015.rex](#)

[RxXL Automation 016.rex](#)

[RxXL Automation 017.rex](#)

[RxXL Automation 018.rex](#)

First though, two things had to be decided:

- A naming convention for files
- A naming convention for cells

For files, I experimented with several conventions that used some form of one of the Rexx date formats. I finally settled on:

MonYY_ReportName

totally for my own readability.

Naming cells was easier. Excel sets some limitations, so I decided on:

_ActNum_Title

A small snippet of code solved the Excel limitations:

```
title=titles(col_a_data)
name='_'Translate(Space(title), '_____','()/& -')
```

You can see that [Titles.rex](#) simply accepts the ActNum and/or the Title and returns both.

With [Excel Automation012_014.txt](#) as guidance, and the examples shown in [RxXL Automation 012.rex](#)

this fragment essentially completed the Net Change task:

```
column = 2
data=Word(aline,col_a_words+1)
xlobj~Cells(row,column)~Value=data
xlobj~Cells(row,column)~name = name
If exists? & known?.name Then Do          /* Calc Net Change */
  column=4
  xlobj~Cells(row,column)~formula='=RC[-2]-`oldfile||name
End
```

Interestingly enough, even though I coded Relative formulas, because I didn't have the "R1C1 Reference Style" enabled under Tools->General, Excel automatically changed the formulas to use the Name of the cell referenced!

Adding some more code enabled me to create the formulas that would accumulate the Assets and Liabilities of the various Restoration and Replacement accounts and create a Cross Check.

So now (4Feb07) I had a spreadsheet that looked like [Nov06 BalanceSheet.xls](#) and [BalSheetToExcel.rex](#) was complete.

You might notice that neither of the previous programs is what you might think of as "pure ooRexx". That's one of the beauties of ooRexx! After 28 years of Classic Rexx, it's hard to think right-to-left. But with ooRexx you can mix both styles as you make the transition.

During this time period there were lots of other things going on, including portions of the programs below and correspondence about postings to the RexxLA list.

One of Rony Flatcher's postings prompted me to send him a private note about Set Theory that I was too embarrassed at the time to post to the list. His reply (2Jan07) shows his enthusiasm for his profession and why, I personally (and I'm sure all of his students and RexxLA members), think so highly of him. [Rony Note](#)

The Solution - Phase 3

The next report to improve was [Oct06 IncomeExpense.prt](#) but there was a marked difference between it and the usual [IncomeExpense.pdf](#) file.

The column titles were missing! Well, it's those are the kinds of things that make programming interesting. Since I wanted to insert new columns for %Variance for both Month-to-Date and Year-to-Date, it wasn't that big a deal.

A couple of things to note:

It's best to add columns from right to left. That way your mental pointer for the column name doesn't change. For example:

```
/* Insert columns */
insert_before = 'H'                                /* For YTD %var */
xlobj~Columns(insert_before':'insert_before)~Insert
insert_before = 'E'                                /* For MTD %var */
xlobj~Columns(insert_before':'insert_before)~Insert
```

When adding long formulas, do it **after** doing Autofit so it doesn't get confused about how wide to make the cells.

The result, [Nov06 IncomeExpense.xls](#) was satisfactory.

The program [IncomeExpenseToExcel.rex](#) was finished on 11Jan07, with some minor tweaks added after the next program was completed.

The Solution - Phase 4

The last program simply built on prior knowledge to add a Delta column to the Delinquency Report.

Since the data is confidential, here is an example of the result: [Mar07_Delinquency.xls](#) produced by [DelinquencyToExcel.rex](#)

Follow Up

In a final note to Lee (12Jan07), I said:

Lee,

Just thought you'd like to know that due to your Examples and answers to my questions, I've completed all four programs (except for the dialogs. See email of 24Dec).

Two programs were especially tricky, but your Examples (and some guesswork) solved the problems:

1-Inserting columns and data

2-Linking to data in a previous monthly report AND calculating the difference. This is the one where I had to guess about how to name a cell... I guessed right!

3-Examining a previous monthly report to determine WHICH names were known before attempting to calculate the difference in the current report.

Les

It took eight weeks (17Nov06 to 12Jan07) to go from concept to finished programs. Since I can only sit in front of the computer for about three hours (total) per day, and I never use it on Monday or Friday, subtracting Thanksgiving gives us $39 \times 3 = 117$ hours or 14.6 'work days'.

Not to bad for an ooRexx beginner!

There is a little tweaking yet to be done to get rid of the prompts when linking data between spreadsheets and I might decide to write a program to populate the [2008_Budget.xls](#) on a monthly basis, after I link the data between years.

Conclusion

This experience has proven the theory (with apologies to C.A.R Hoare):

Computer programming is an exact science. The problem having been precisely specified, you implement the appropriate algorithms to solve the problem. If the correct results aren't forthcoming, you change things a little bit and try again.

And, of course, it helps to have all the members of RexxLA willing to lend you their expertise!