

10-29-2014

## Setting up a SAM in Excel using VBA

Amir Borges Ferreira Neto  
West Virginia University, amneto@mix.wvu.edu

Follow this and additional works at: [https://researchrepository.wvu.edu/rri\\_tech\\_docs](https://researchrepository.wvu.edu/rri_tech_docs)



Part of the [Regional Economics Commons](#)

---

### Digital Commons Citation

Ferreira Neto, Amir Borges, "Setting up a SAM in Excel using VBA" (2014). *Regional Research Institute Technical Documents*. 9.

[https://researchrepository.wvu.edu/rri\\_tech\\_docs/9](https://researchrepository.wvu.edu/rri_tech_docs/9)

This Article is brought to you for free and open access by the Regional Research Institute at The Research Repository @ WVU. It has been accepted for inclusion in Regional Research Institute Technical Documents by an authorized administrator of The Research Repository @ WVU. For more information, please contact [researchrepository@mail.wvu.edu](mailto:researchrepository@mail.wvu.edu).

# Regional Research Institute West Virginia University

Technical Document Series



## Setting up a SAM in Excel using VBA

AMIR BORGES FERREIRA NETO

RRI TechDoc 2014-05

Date submitted: 10/29/2014

Key words/Codes: CGE, SAM, Spreadsheet, Excel, VBA; C68,  
R13

# Setting up a SAM in Excel using VBA\*

## **Abstract**

This technical document describes how to set up a full Social Accounting Matrix from its own partition in a spreadsheet file. This is done using VBA code.

## **Problem Context**

A Social Accounting Matrix, SAM, (summarized as in Figure 1) is a powerful tool to perform economic analysis. There are several ways to build a SAM, but starting from scratch can be a costly and time-consuming process. Therefore many analysts rely on commercial sources for SAM data provided by a third-party. The downside on this is that such data can be closely tied to specific software, which can make it difficult for subsequent editing or modifications. Regional Research Institute Resource Document 2014-03 discusses some aspects of one solution to this problem, and how to work on a multi-software environment (IMPLAN, GAMS and Excel). This document augments the toolkit for this multi-software environment by describing and providing a method for setting up the SAM on a spreadsheet after successfully obtaining each of its partitions from GAMS (grey sub-matrices on Figure 1).

## **Dealing with Excel**

After exporting each SAMs partition from GAMS to Excel spreadsheet, the SAM will reside in separate sheets in the file. Follow the steps below to collect the partitions into a single sheet in the conventional SAM format:

---

\*Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant No. 1235684 and USDA NIFA Award 2012-67009-19660.

Figure 1: Social Accounting Matrix

	Industries	Commodities	Factors	Institutions	Trade	Total
Industries		Make Matrix			Commodity Exports	Industry Output
Commodities	Use Matrix			Domestic Commodity Consumption		Domestic Commodity Output
Factors	Value Added					Value Added
Institutions		Domestic Commodity Sales	Factor Disbursements	Inter-Institutional Transfers	Commodity & Factor Services Exports	Institutional Receipts, Commodities, Factor
Trade	Commodity Imports		Factor Services Imports	Commodity Imports & Remittances	Trans-shipments	Services Imports, Remittances
Total	Industry Output	Domestic Commodity Output	Factor Outlay	Institutional Outlay	Commodity Factor Services Exports	

Source: Weisskoff (2000).

1. Create a new sheet (lets call it SAM);
2. Enable the Developer ribbon and Macros in your Excel file;
  - (a) Right-click in a ribbon -> Customize the Ribbon -> select Developer and press ok.
3. In the Developer Ribbon select Visual Basic (or press Alt+F11);
4. Double click in the SAM sheet and paste the following code.
  - (a) After pasting the code, click in the play button in the ribbon to run the code. If the partitions are as assumed, the code will read properly and will set-up the SAM in the SAM sheet;
  - (b) The SAM sheet must be empty for the code to work correctly. Otherwise, there will be issues in the set-up SAM. If the first run has problems, clear the content generated so that the destination for the generated SAM is empty.

Notes:

- it is assumed that each partition begins in the A1 cell of each sheet, in other words, A1 is blank, and beginning in A2 there is the head of each row of the partition; and the first row is the transpose of column A;

- the name of each partition follows the one on this code, and the name of each spreadsheet sheet follows the name of the partition;

The code is built to a SAM of a maximum of 50 commodities and activities, 15 factors, 20 institutions and 4 trade regions. If needed, this can be modified by expanding the number of components to be read in the part of the code called sectors. For example, the SAM has 60 activities/-commodities, change: Worksheets(“USE”).Range(“A1:A50”).copy to Worksheets(“USE”).Range(A1:A60).copy.

## Supporting Algorithm(s)/Code

```
'=====
'==== Code to Set up a SAM after exporting from GAMS =====
'== This code was developed by Amir B. Ferreira Neto ==
'== It reads the excel file exported from GAMS and set-up ==
'== the Social Accounting Matrix in the right position ==
'=====

'=====
'====
'====          Social Accounting Matrix          ====
'====
'====          A      C      F      INST      T      '
'====          1      2      3      4      5      '
'==== A      1      MAKE      '
'==== C      2      USE      IUSE      CEXPRT      '
'==== F      3      FD      FEXPRT      '
'==== INST  4      IMAKE      FS      TRNSFR      IEXPRT      '
'==== T(FT) 5      CIMPRT      FIMPRT      IIMPRT      TRNSHP      '
'=====

'=====
'====          SECTORS          =====
'====
'==== First we set up the border of the SAM
'==== by copying the boarder of partitions
'==== The "With" routine: first calculate the number
'==== (a variable) of the right row in which we should
'==== paste the next part, and then paste the boarder
'==== in the right place the partitions as well.
'=====
```

```
Sub FullSAM()
Worksheets("USE").Range("A1:A50").copy
Worksheets("SAM").Range("A1:A50").PasteSpecial

Worksheets("USE").Range("A2:A50").copy
With Sheets("SAM")
    com = .Range("A" & Rows.Count).End(xlUp).Row + 1
    .Range("A" & com).PasteSpecial
End With

Worksheets("FD").Range("A2:A15").copy
With Sheets("SAM")
    fac = .Range("A" & Rows.Count).End(xlUp).Row + 1
    .Range("A" & fac).PasteSpecial
End With

Worksheets("IMAKE").Range("A2:A21").copy
With Sheets("SAM")
    inst = .Range("A" & Rows.Count).End(xlUp).Row + 1
    .Range("A" & inst).PasteSpecial
End With

Worksheets("CIMPRT").Range("A2:A5").copy
With Sheets("SAM")
    trade = .Range("A" & Rows.Count).End(xlUp).Row + 1
    .Range("A" & trade).PasteSpecial
End With

com1 = com - 1
fac1 = fac - 1
inst1 = inst - 1
trade1 = trade - 1
trade2 = Range("A" & Rows.Count).End(xlUp).Row

Worksheets("SAM").Range("A2", Range("A2").End(xlDown)).copy
Worksheets("SAM").Range("B1").PasteSpecial Transpose:=True

'=====  
'===== ACTIVITIES =====  
'=====  
'===== Use =====  
Worksheets("USE").Activate  
Worksheets("USE").Range("B2").Select  
Worksheets("USE"). -
```

```

Range(Selection, Selection.End(xlDown)).Select
Worksheets("USE"). -
Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM").Range("B" & com).PasteSpecial

'=====FD===== '
Worksheets("FD").Activate
Worksheets("FD").Range("B2").Select
Worksheets("FD"). -
Range(Selection, Selection.End(xlDown)).Select
Worksheets("FD"). -
Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM").Range("B" & fac).PasteSpecial

'=====COMMODITIES===== '
'=====Make===== '
Worksheets("MAKE").Activate
Worksheets("MAKE").Range("B2").Select
Worksheets("MAKE"). -
Range(Selection, Selection.End(xlDown)).Select
Worksheets("MAKE"). -
Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
Range(Cells(2, com), Cells(com1, fac1)).PasteSpecial

'=====I-Make===== '
Worksheets("IMAKE").Activate
Worksheets("IMAKE").Range("B2").Select
Worksheets("IMAKE"). -
Range(Selection, Selection.End(xlDown)).Select
Worksheets("IMAKE"). -
Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
Range(Cells(inst, com), Cells(trad1, fac1)).PasteSpecial

```

```
'=====                                ====='  
Worksheets("CIMPRT").Activate  
Worksheets("CIMPRT").Range("B2").Select  
Worksheets("CIMPRT"). -  
    Range(Selection, Selection.End(xlDown)).Select  
Worksheets("CIMPRT"). -  
    Range(Selection, Selection.End(xlToRight)).copy  
  
Worksheets("SAM").Activate  
Worksheets("SAM"). -  
    Range(Cells(trade, com), Cells(trade2, fac1)).PasteSpecial  
  
'=====                                ====='  
'=====                                ====='  
'=====                                ====='  
'=====                                ====='  
Worksheets("FS").Activate  
Worksheets("FS").Range("B2").Select  
Worksheets("FS"). -  
    Range(Selection, Selection.End(xlDown)).Select  
Worksheets("FS"). -  
    Range(Selection, Selection.End(xlToRight)).copy  
  
Worksheets("SAM").Activate  
Worksheets("SAM"). -  
    Range(Cells(inst, fac), Cells(trade1, inst1)).PasteSpecial  
  
'=====                                ====='  
Worksheets("FIMPRT").Activate  
Worksheets("FIMPRT").Range("B2").Select  
Worksheets("FIMPRT"). -  
    Range(Selection, Selection.End(xlDown)).Select  
Worksheets("FIMPRT"). -  
    Range(Selection, Selection.End(xlToRight)).copy  
  
Worksheets("SAM").Activate  
Worksheets("SAM"). -  
    Range(Cells(trade, fac), Cells(trade2, inst1)).PasteSpecial  
  
'=====                                ====='  
'=====                                ====='  
'=====                                ====='  
'=====                                ====='  
                                I-Use
```



```

Worksheets("IUSE").Activate
Worksheets("IUSE").Range("B2").Select
Worksheets("IUSE"). -
    Range(Selection, Selection.End(xlDown)).Select
Worksheets("IUSE"). -
    Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
    Range(Cells(com, inst), Cells(fac1, trade1)).PasteSpecial

'==== TRNSFER ====='
Worksheets("TRNSFER").Activate
Worksheets("TRNSFER").Range("B2").Select
Worksheets("TRNSFER"). -
    Range(Selection, Selection.End(xlDown)).Select
Worksheets("TRNSFER"). -
    Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
    Range(Cells(inst, inst), Cells(trade1, trade1)).PasteSpecial

'==== IIMPRT ====='
Worksheets("IIMPRT").Activate
Worksheets("IIMPRT").Range("B2").Select
Worksheets("IIMPRT"). -
    Range(Selection, Selection.End(xlDown)).Select
Worksheets("IIMPRT"). -
    Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
    Range(Cells(trade, inst), Cells(trade2, trade1)).PasteSpecial

'=====
'===== TRADE ====='
'=====
'==== CEXPRT ====='
Worksheets("CEXPRT").Activate
Worksheets("CEXPRT").Range("B2").Select
Worksheets("CEXPRT"). -
    Range(Selection, Selection.End(xlDown)).Select
Worksheets("CEXPRT"). -
    Range(Selection, Selection.End(xlToRight)).copy

```

```
Worksheets("SAM").Activate
Worksheets("SAM"). -
    Range(Cells(com, trade), Cells(fac1, trade2)).PasteSpecial

'=====FEXPRT====='
Worksheets("FEXPRT").Activate
Worksheets("FEXPRT").Range("B2").Select
Worksheets("FEXPRT"). -
    Range(Selection, Selection.End(xlDown)).Select
Worksheets("FEXPRT"). -
    Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
    Range(Cells(fac, trade), Cells(inst1, trade2)).PasteSpecial

'=====IEXPRT====='
Worksheets("IEXPRT").Activate
Worksheets("IEXPRT").Range("B2").Select
Worksheets("IEXPRT"). -
    Range(Selection, Selection.End(xlDown)).Select
Worksheets("IEXPRT"). -
    Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
    Range(Cells(inst, trade), Cells(trade1, trade2)).PasteSpecial

'=====TRNSHP====='
Worksheets("TRNSHP").Activate
Worksheets("TRNSHP").Range("B2").Select
Worksheets("TRNSHP") -
    .Range(Selection, Selection.End(xlDown)).Select
Worksheets("TRNSHP"). -
    Range(Selection, Selection.End(xlToRight)).copy

Worksheets("SAM").Activate
Worksheets("SAM"). -
    Range(Cells(trade, trade), Cells(trade2, trade2)).PasteSpecial

End Sub
'====='
```

## References

Weisskoff,R. (2000) Missing Pieces in Ecosystem Restoration: The Case of the Florida Everglades. *Economic Systems Research*, 12(3):271-304.