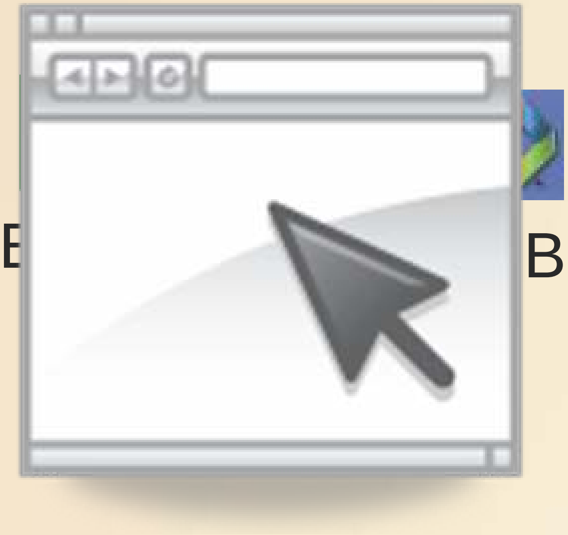
SAS® GLOBAL FORUM

SAS Integration Technologies, UNIX and Visual Basic .Net

Antoine Chevrette, Statistics Canada

What's it all about?

UNIX



E

Basic

The result is a system that is

- •Homogenous
- •Robust
- •Uses the best of all technologies

Agenda

- History
- Solution 1 (SAS/CONNECT)
- Solution 2 (SAS/SERVER)
- Browsing Large Files Technique
- Conclusion

Why did I decide to use all those technologies?

For years in the Survey of

Employment, Payrolls and Hours:

- Unix servers have been used to meet the needs of its surveys and clients
- Using SAS on a UNIX server is the best choice to speed up data manipulation
- SAS/AF has been used to develop the editing and GUI process management.

New challenges

- Received increasingly complex requests for data analysis
- Received increasingly complex requests for data editing GUIs.

Our tasks is to provide our clients with

- High-quality products
- Satisfy our client needs

Clients

What do our clients want (specifications) ?

Excel:

- Pivot Tables
- Filter
- Excel Graphics

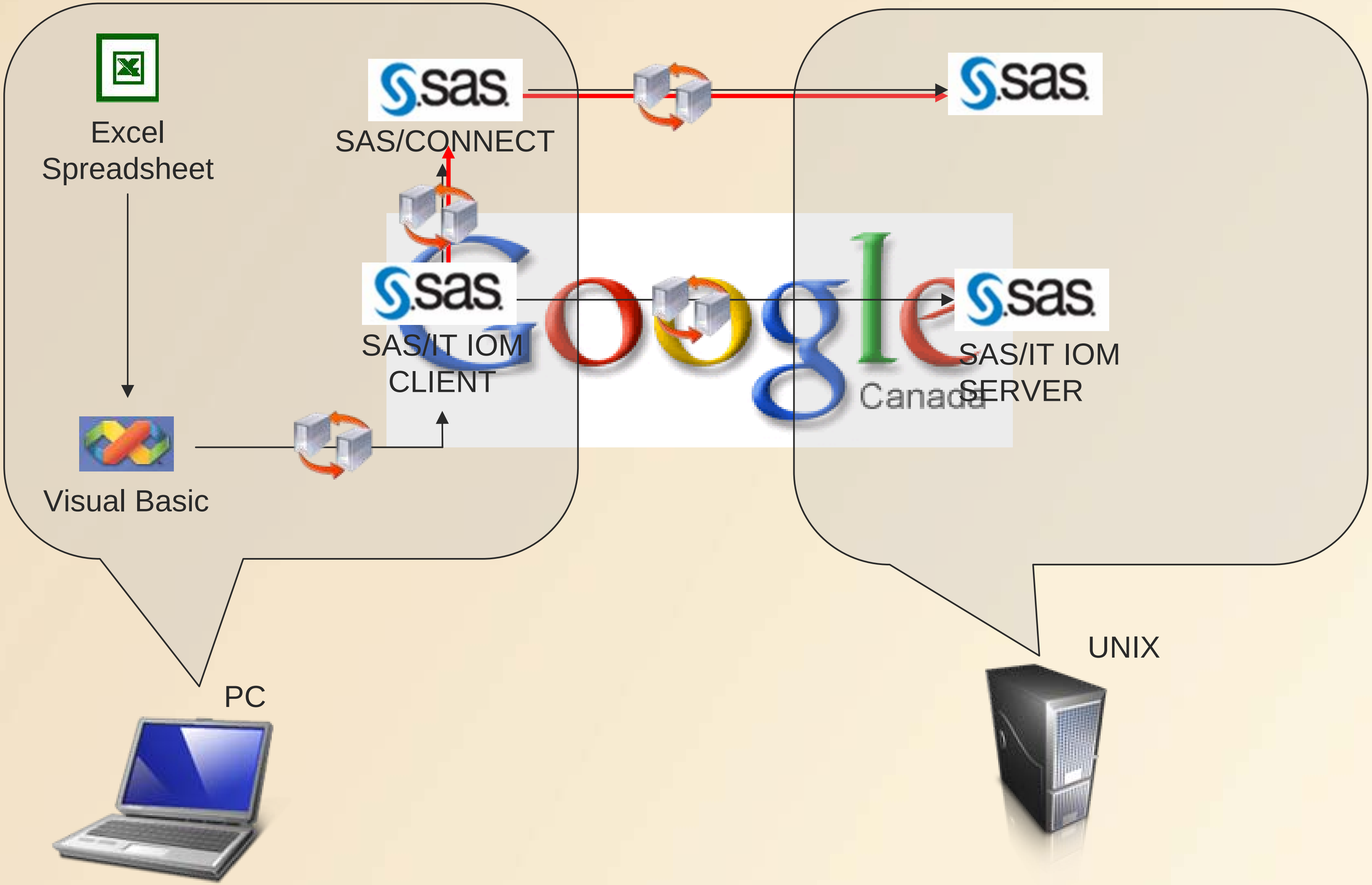
Try to redo excel features in SAS/AF on UNIX.

- Low resources (expertise not spread out)
- Time consuming
- Maintenance Cost
- Client Expectations

Why do we not give our Clients an Excel Spreadsheet,

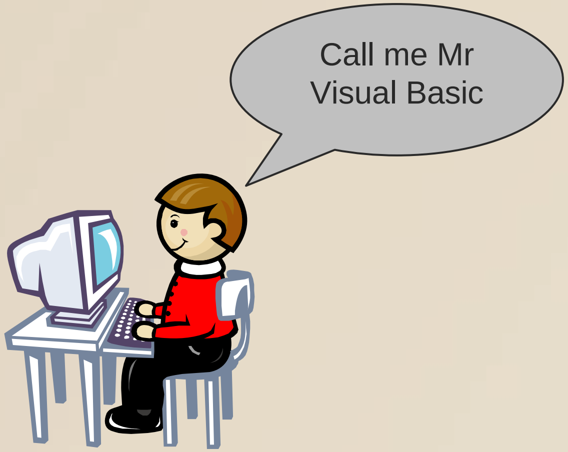
I am sure the technology is available!

Quest to find a solution!



General information for both Solutions

Visual Basic should be regarded as a normal SAS user (like a human).

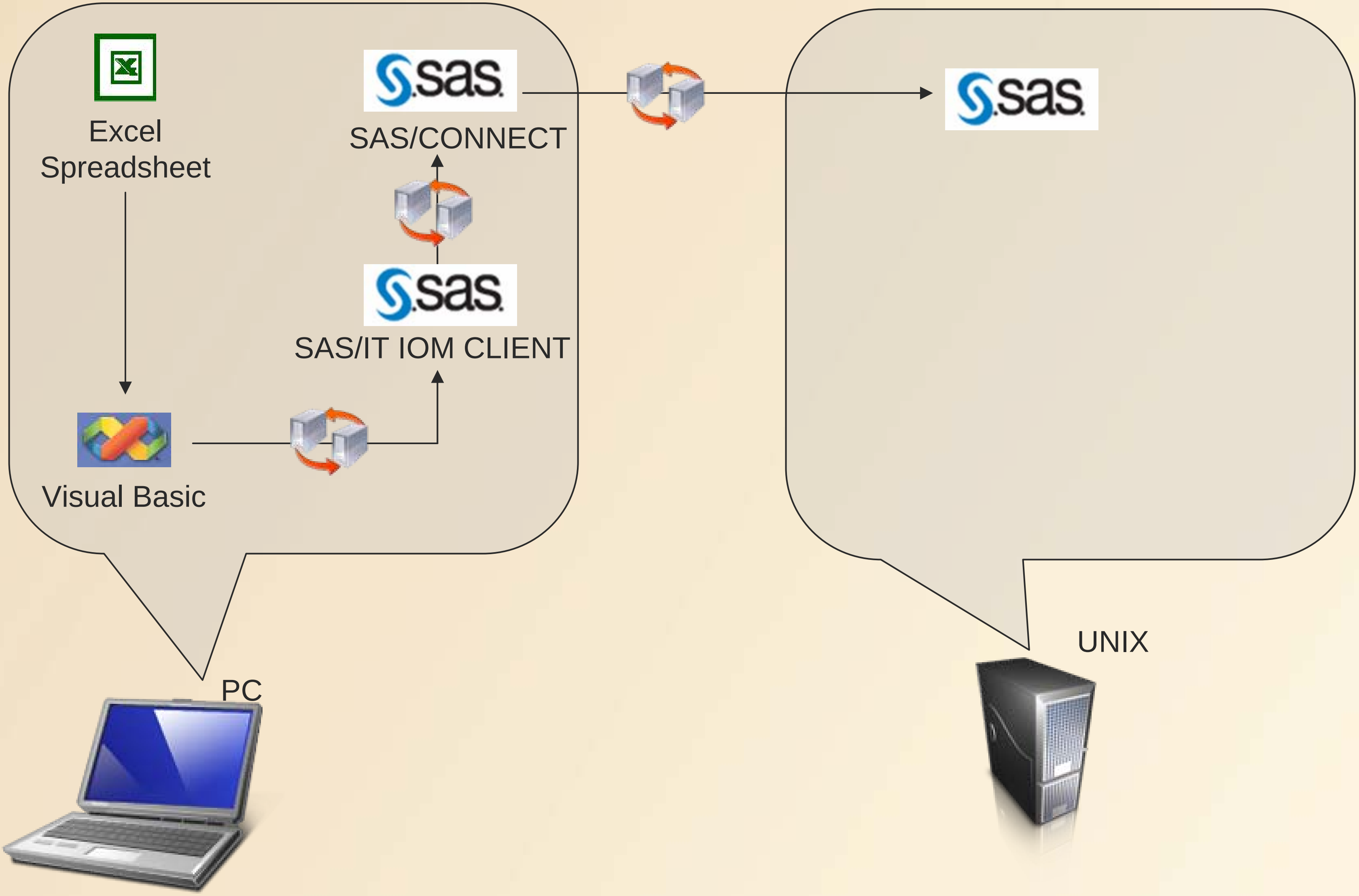


Visual Basic does not use a keyboard or the SAS interface.

SAS provides a set of TOOLS (IOM - Integration Object Model) that allow Visual basic to perform the same tasks as a human.

- open an SAS session
- submit code
- edit data
- read reports
- and so on

Let’s look at Solution 1



Excel Spreadsheet

SAS/CONNECT

SAS/IT IOM CLIENT

Visual Basic

PC

UNIX

Solution 1 (SAS/CONNECT)

What do you need?

- Microsoft Excel components.
- Visual Basic.net (Visual Studio)
- SAS/PC with SAS/CONNECT
- SAS IT/IOM client (inttech.exe).
- Unix Server with SAS.

In a VB project you have to reference
The following components :

- Microsoft Office Spreadsheet 11.0
- SAS: Integrated Object Model (IOM)
- SASIOMCommon
- SASWorkspaceManager

Connection Class provides the following functions:

- openWorkspace
- CloseWorkspace
- SubmitSascode
- RSubmitSascode
- Signon
- Sigoff
- openRS
- closeRS
- Log

Connection Script Class provides the following functions:

- setUsername
- setPassword
- makescript

Solution 1 – (SAS/CONNECT)

The following code allows sending of SAS code to UNIX:

```
Dim sasdata As String = "'/user/home/'"
Dim mySASconnection As New SasConnection
Dim myscript As New script

myscript.setUsername(username)
myscript.setPassword(password)
myscript.makeScript()

mySASconnection.openWorkspace()
mySASconnection.setservername("lsdsas")
mySASconnection.signon("libname mywork" & sasdata)

mySASconnection.RSubmitSASCode
        ("libname mywork" & sasdata & ";" + _
         " data mywork.company; " + _
         "     set sashelp.company(where=(level2='NEW YORK'));
              “+ _
         " run;")
```

Solution 1 - Excel part

The recordset that connects to a SAS dataset is defined as follows:

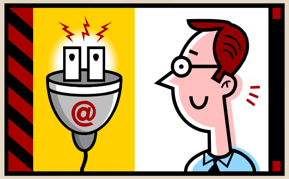
```
Dim rs As New ADODB.Recordset
rs = mySASconnection.openRs("mywork.company")
```

You now have to copy the information from your recordset to the excel spreadsheet.

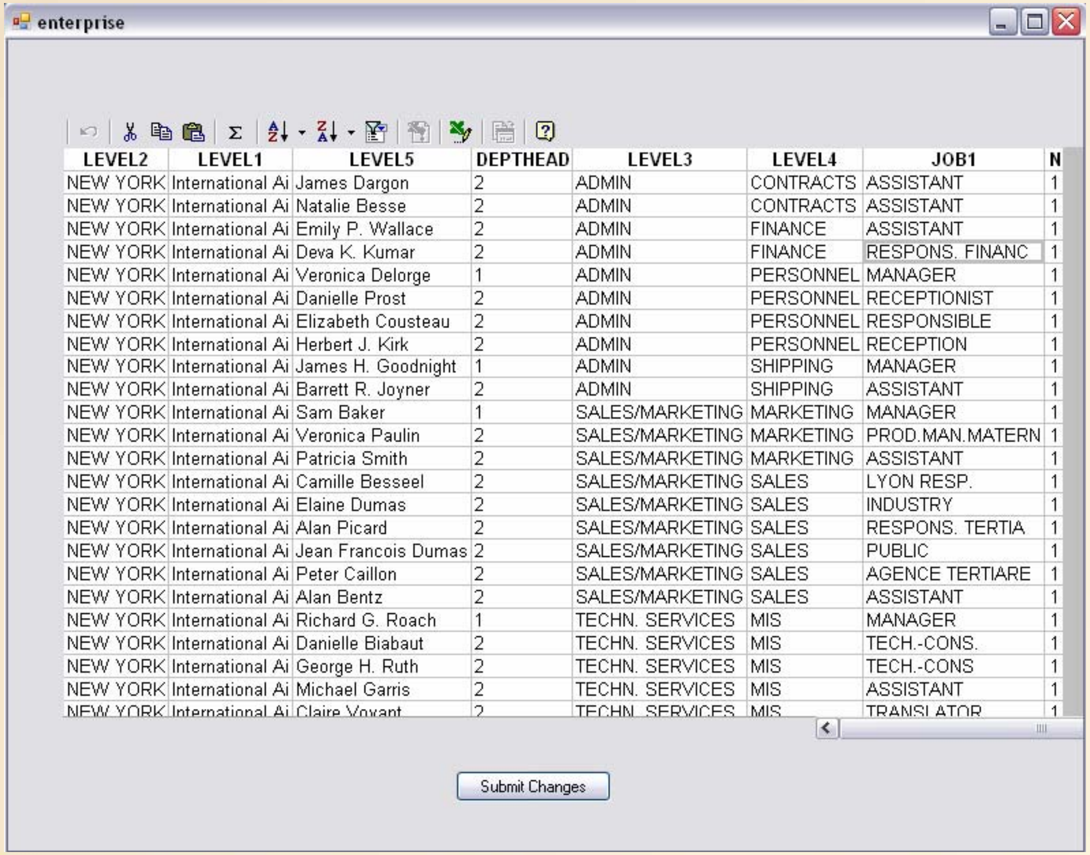
```
AxSpreadsheet1.ActiveWorkbook.ActiveSheet.Cells. _
CopyFromREcordset(rs)
```

Solution 1 (SAS/CONNECT)

It is now time to ‘unplug’ the connections :



```
mySASconnection.sigoff()
mySASconnection.closeRS()
mySASconnection.closeWorkspace()
```



LEVEL2	LEVEL1	LEVEL5	DEPTHEAD	LEVEL3	LEVEL4	JOB1	N
NEW YORK	International Ai	James Dargon	2	ADMIN	CONTRACTS	ASSISTANT	1
NEW YORK	International Ai	Natalie Besse	2	ADMIN	CONTRACTS	ASSISTANT	1
NEW YORK	International Ai	Emily P. Wallace	2	ADMIN	FINANCE	ASSISTANT	1
NEW YORK	International Ai	Deva K. Kumar	2	ADMIN	FINANCE	RESPONS. FINANC	1
NEW YORK	International Ai	Veronica Delorge	1	ADMIN	PERSONNEL	MANAGER	1
NEW YORK	International Ai	Danielle Prost	2	ADMIN	PERSONNEL	RECEPTIONIST	1
NEW YORK	International Ai	Elizabeth Cousteau	2	ADMIN	PERSONNEL	RESPONSIBLE	1
NEW YORK	International Ai	Herbert J. Kirk	2	ADMIN	PERSONNEL	RECEPTION	1
NEW YORK	International Ai	James H. Goodnight	1	ADMIN	SHIPPING	MANAGER	1
NEW YORK	International Ai	Barrett R. Joyner	2	ADMIN	SHIPPING	ASSISTANT	1
NEW YORK	International Ai	Sam Baker	1	SALES/MARKETING	MARKETING	MANAGER	1
NEW YORK	International Ai	Veronica Paulin	2	SALES/MARKETING	MARKETING	PROD.MAN.MATERN	1
NEW YORK	International Ai	Patricia Smith	2	SALES/MARKETING	MARKETING	ASSISTANT	1
NEW YORK	International Ai	Camille Besseel	2	SALES/MARKETING	SALES	LYON RESP.	1
NEW YORK	International Ai	Elaine Dumas	2	SALES/MARKETING	SALES	INDUSTRY	1
NEW YORK	International Ai	Alan Picard	2	SALES/MARKETING	SALES	RESPONS. TERTIA	1
NEW YORK	International Ai	Jean Francois Dumas	2	SALES/MARKETING	SALES	PUBLIC	1
NEW YORK	International Ai	Peter Caillon	2	SALES/MARKETING	SALES	AGENCE TERTIARE	1
NEW YORK	International Ai	Alan Bentz	2	SALES/MARKETING	SALES	ASSISTANT	1
NEW YORK	International Ai	Richard G. Roach	1	TECHN. SERVICES	MIS	MANAGER	1
NEW YORK	International Ai	Danielle Biabaut	2	TECHN. SERVICES	MIS	TECH.-CONS.	1
NEW YORK	International Ai	George H. Ruth	2	TECHN. SERVICES	MIS	TECH.-CONS	1
NEW YORK	International Ai	Michael Garris	2	TECHN. SERVICES	MIS	ASSISTANT	1
NEW YORK	International Ai	Claire Voyant	2	TECHN. SERVICES	MIS	TRANSLATOR	1

Solution 1 (Edit Unix Data)

Let’s say we edit a value from column Level5, we have to reopen a SAS connection and a RecordSet:

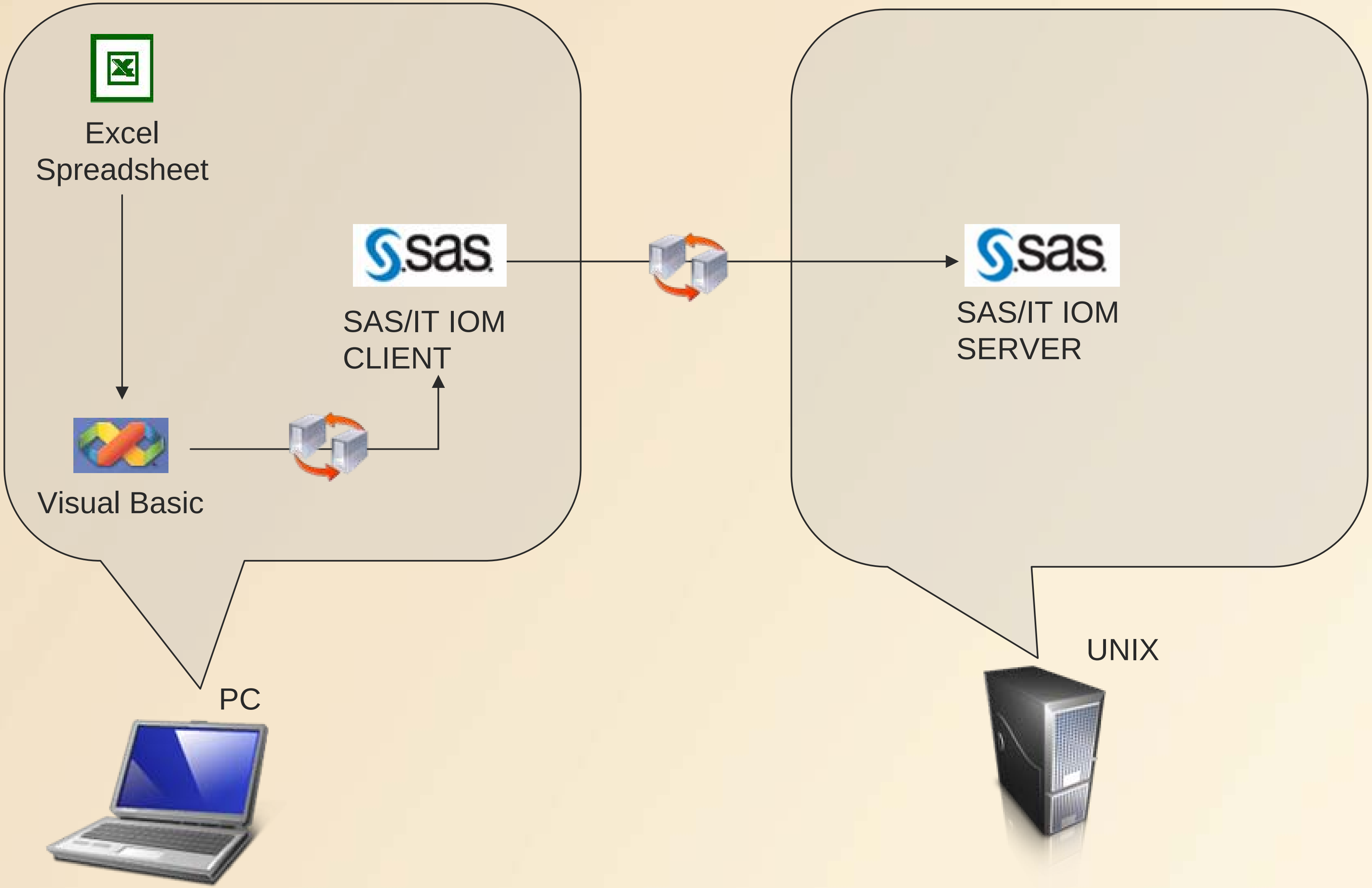
```
mySASconnection.openWorkspace()
mySASconnection.setservername("lsdsas")
mySASconnection.sigon("libname mywork" & sasdata)

Dim rs As New ADODB.Recordset
rs = mySASconnection.openRs("mywork.company")
```

Here is my update:

```
Dim i As Integer
rs.MoveFirst()
i = 2
While Not rs.EOF
        If rs("LEVEL5").Value <> AxSpreadsheet1.ActiveSheet.Cells(i, 3).value() Then
                rs("LEVEL5").Value = AxSpreadsheet1.ActiveSheet.Cells(i, 3).value()
                rs.Update()
        End If
        i = i + 1
        rs.MoveNext()
 End While
```

Let’s look at Solution 2



Solution 2 (SAS/SERVER)

What do you need?

- Microsoft Excel components.
- Visual Basic.net (Visual Studio)
- SAS IT/IOM client (inttech.exe).
- SAS IT/IOM Server
- Unix Server with SAS.

In a VB project you have to reference the following components :

- Microsoft Office Spreadsheet 11.0
- SAS:Integrated Object Model (IOM)
- SASIOMCommon
- SASWorkspaceManager

Connection Class provides the following functions:

- OpenWorkspace
- CloseWorkspace
- SubmitSascode
- openRS
- closeRS
- Log
- ServerName
- Port
- Login
- Password

Solution 2 (SAS/SERVER)

How do you open a connection?

```
Dim mySASconnection As New SasConnection

mySASconnection.ServerName = "lsdsas"
mySASconnection.port = 6066
mySASconnection.login = login.username.Text
mySASconnection.password = login.password.Text
mySASconnection.openWorkspace()
```

The procedures for submitting code and displaying and editing data are the same as in solution 1

X Command

Since you can run code on UNIX you can use the X command to run UNIX shell script from SAS.

```
mySasconnection.SubmitSASCode("X'/home/user/jobs.sh"
+ TextBox1.Text.ToString  + "';")
```

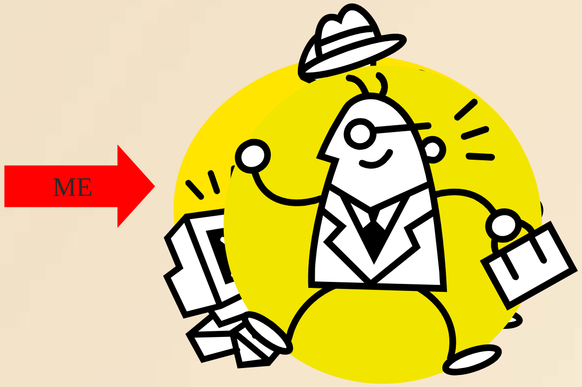
This allows you to **completely** manage a UNIX job from a Visual Studio environment without having to Install any additional software. (this works for both solutions)

Back to the client

After building a prototype using the preceding theory I showed it to the Client, here is his reaction:

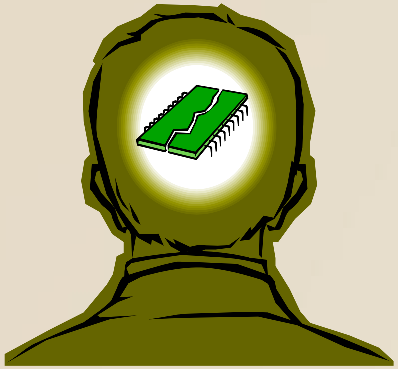
If I am not able to Browse a 70 million record dataset from UNIX in a vb form it's not worth doing it!

t! It's
ays
love it !



Problematic of Browsing Large Files

Performance will be seriously degraded if a file exceeds the PC’s memory capacity.

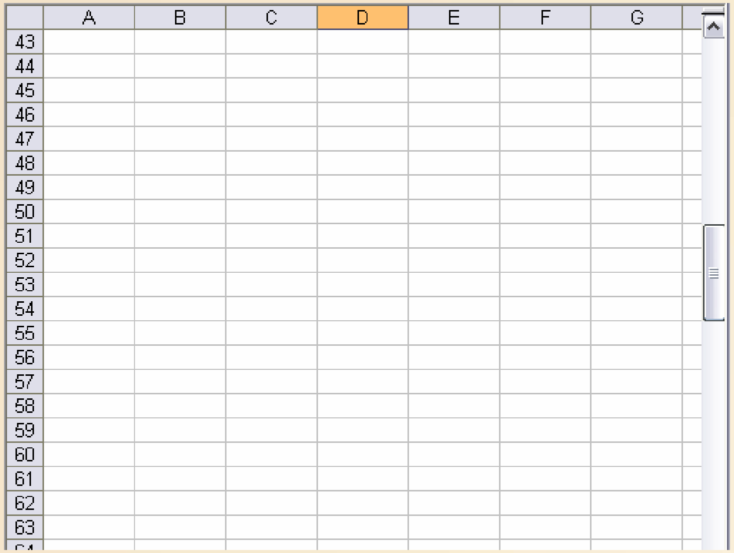


Google™ Canada

Solution of Browsing Large Files

This problem can be overcome using a virtual mode technique taken from the following article:

Implementing Virtual Mode with Just-In-Time Data Loading in the Windows Forms DataGridView Control. MSDN (http://msdn2.microsoft.com/en-us/library/ms171624.aspx)



Refreshed the data in a DataGrid each time the banner is accessed.
Example: 100 rows at a time

Solution of Browsing Large Files

The classes are supplied with the Microsoft article. But it has to be modified in order to be usable with SAS.

I included a modified version in my paper that work with SAS.

I even implemented it with Firebird database and it works great.

Conclusion

- There are various methods of combining VB.net and SAS,

  The final choice must be based on the organization’s needs.

- With solution 1 (SAS/CONNECT)

  - It costs less to develop an application.
  - More complex code,
  - loss of some functions associated with SAS Server (metadata)
  - the need for SAS/PC on the client computer.

These disadvantages are not enough to warrant the purchase of an SAS Server licence if the application to be built is straightforward.

If the application is at all complex, it would be better to buy SAS Server.

Conclusion

Whether SAS Server is used or not,

- integration of the SAS and VB.Net technologies allows for more rapid development of complex applications.
- An organization that already uses both technologies will certainly benefit by combining them.

SAS IT USER