



Interfacing SAS with VB.Net



*Yves DeGuire
Maurice Djona*

*SAS Technology Centre
System Development Division
Statistics Canada*



Content

- Why interfacing SAS with VB.Net?
- Overview of various interface techniques
- Presentation of 2 basic techniques
- Presentation of 2 other advanced techniques





Why Interfacing VB.Net with SAS?

- SAS and VB.Net are the 2 most popular software packages for development at Statistics Canada.
- The 2 software packages complement each other:
 - SAS is excellent for large scale data processing and has a very rich function set
 - VB.Net helps you develop high quality graphical interfaces.
- SAS is becoming more and more open
 - Integration Technologies
 - SAS 9 Architecture





Short Overview of SAS

- A suite of integrated products and solutions for information delivery:
 - Data processing
 - Statistical Analysis
 - Reporting
 - etc.
- Available on multiple platforms
- Several proprietary programming languages
- Extremely powerful for batch processing
- The runtime environment is a SAS session





Short Overview of VB.Net

- Object-Oriented language
- Microsoft proprietary language for the Windows platform
- Very powerful for developing graphical user interfaces
- Visual Studio .Net is the development environment
- Typically, the .Net framework is the runtime environment





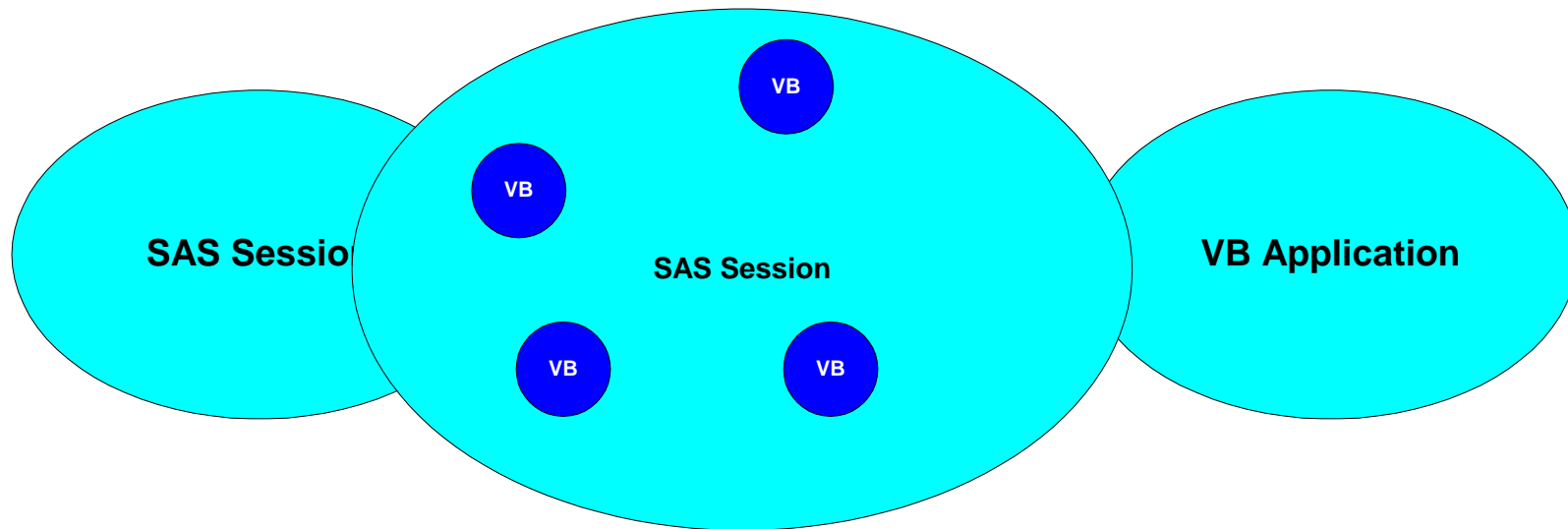
Various Possible Configurations

- Configurations based on the concept of client-server:
 - SAS Client / VB server without integration
 - SAS Client / VB server with integration
 - VB Client / SAS Server without integration
 - VB Client / SAS server with integration
- For each configuration, many interface techniques exist
- Client and server can operate on various network-linked machines



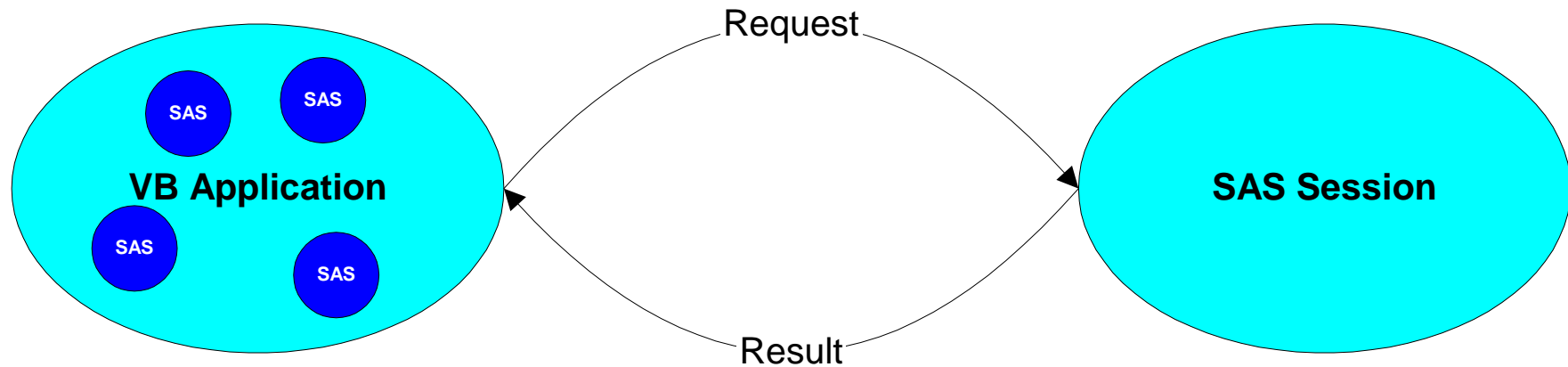


SAS Client / VB Server





VB Client / SAS Server





Interface Techniques WITHOUT Integration

Technique	SAS Client	SAS Server
Command Line	X	X
Pipe	X	X
Named Pipe	X	X
DDE	X	





Interface Techniques WITH Integration

Technique	SAS Client	SAS Server
DLL	X	
OLE Control	X	
OLE SAS		X
EG Task Addin		X
Integrated Object Model		X





Proc Deterministic

- One of the procedures available within the Banff system (the generalized edit and imputation system of Statistics Canada)
- This procedure provides valid values to fields to be imputed where only one possible imputed value will permit them to pass the set of edits.
- Example of a call:

```
proc DETERMINISTIC  
  data=determdata  
  instatus=determstat  
  out=outdata  
  outstatus=outstatus  
  edits="Q1 + Q2 + Q3 + Q4 - TOTAL = 0;"  
  id ident;  
run;
```





Command Line

- The statement to invoke SAS from the command prompt can be written in a VB application.
- Syntax to invoke SAS from the DOS command line:

```
C:\>C:\...\sas.exe -SYSIN C:\...\pgmname.sas -ICON -NOSPLASH  
-LOG C:\...\logname.log -PRINT C:\...\lstname.lst  
-SET envVar1 "value1" -SET envVar2 "value2"
```

- Equivalent Syntax in VB.Net:

Shell(CommandLine, style, wait, timeout)

Where CommandLine is the whole command shown above.

- The SAS job can access environment variables defined by `-SET` through the macro function `%SYSGET`.





Command Line – cont.

Demo





SAS OLE Automation

- SAS behaves as an “OLE automation server” by exposing methods and properties in VB.NET, that is the “OLE automation controller”.

- Elements of syntax in VB.NET:

MySas = CreateObject (“SAS.Application“)	‘Opens a SAS session
MySas. Submit (MySasCode	‘Submits the SAS code
MySas. ResultString	‘Text written within SAS
MySas. RC	‘Integer written within SAS
MySas. Quit	‘Ends the SAS session





SAS OLE Automation – cont.

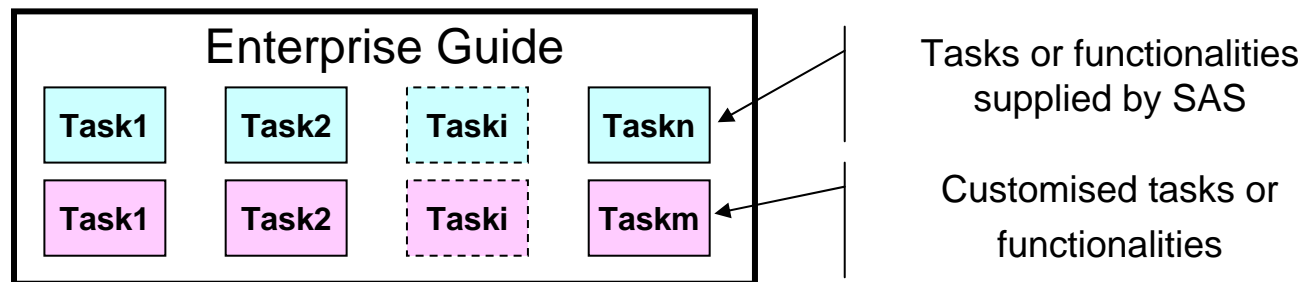
Demo





EG Task Addin

- **Enterprise Guide (EG):** Windows application to easily access the SAS functionalities; SAS code generator.
- **«Custom Add-in» for EG:** SAS provides a set of COM objects allowing you to develop, in .NET, custom tasks that can be integrated to EG
- **Main development steps in VB.NET for EG 3.0:**
 - Reference the components provided by SAS, such as SAS.EG.Addins.dll (in addition to required .NET components)
 - Mandatory implementation of the 3 interfaces [ISASTaskAddIn](#), [ISASTaskDescription](#) and [ISASTask](#) contained in SAS.EG.Addins
 - Compile the project to create a DLL
 - Record the DLL from the EG3 environment and use it...





EG Task Addin – cont.

Demo

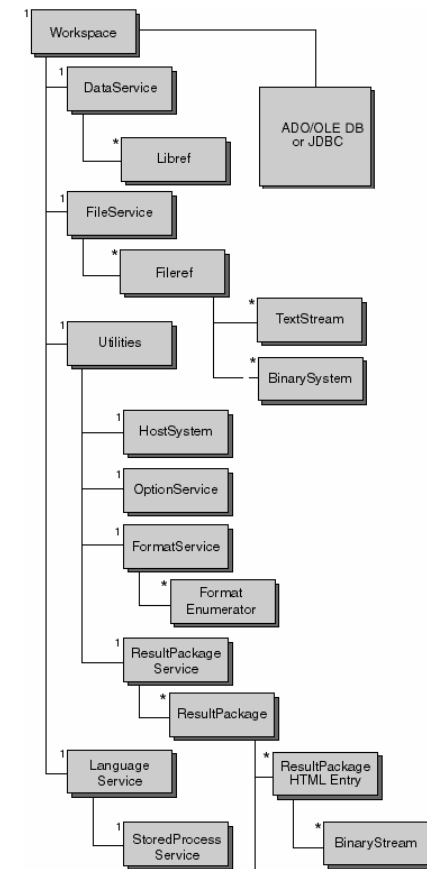




Integrated Object Model

Some definitions:

- **IOM:** Hierarchy of COM objects supplied by SAS to develop client applications, for example in VB.Net, in order to access SAS functionalities and data
- **IOM Server:** A server providing IOM services to client applications (e.g. create a SAS session on a Unix server for a Windows client application written in VB)
- **Main IOM interfaces:**
 - Workspace (SAS session)
 - LanguageService (submit, list/log, stored processes)
 - DataService (librefs, ADO, OLE-DB or JDBC)
 - FileService (filerefs, directories)
 - Utilities (result packages, formats, host info)
- **COM:** Windows client Vs IOM server on the same local machine
- **DCOM:** Windows Client Vs remote IOM server on Windows
- **IOM Bridge for COM:** Windows Client Vs remote IOM server on other platforms (Unix, OS/390)





Integrated Object Model – Cont.

Programming Elements in VB.Net:

- **Reference the relevant libraries:**
 - From SAS: SAS.dll, SASWorkspaceManager.dll (Via Interop to adapt COM to .NET)
 - From .NET: System.Data (to access SAS Data)

- **Some elements of syntax:**

Create a SAS session SAS on the server ServerDef, that can be on other OS system (ex. Unix):

```
Public obSAS As Workspace = obWM.Workspaces.CreateWorkspaceByServer(Name, _  
Visibility, ServerDef, LoginName, Password, xmlInfo)
```

Run the program NomProg.sas, and return results in a ResultPackage:

```
obSp.ExecuteWithResults(NomProg, paramList, obResultPackage)
```

Read a SAS Data set to fill a table in a .NET Dataset:

```
Dim obAdapter As New System.Data.OleDb.OleDbDataAdapter ("select * from  
Libname.tablename", "provider=sas.IOMProvider.9.1; data Source=_LOCAL_; SAS Workspace  
ID=obWS.UniqueIdentifier")
```

```
obAdapter.Fill(netDataset, tablename)
```

Read the SAS log:

```
obLanguageService.FlushLogLines(nbLinesRequested, carriageControls, lineTypes, logLines)
```





Integrated Object Model – Cont.

Demo





Comparative summary of the 4 methods: which one to choose?

	Command Line	SAS OLE Automation	IOM	EG Task Addin
Facilitates the use of complex SAS programs	Yes	Yes	Yes	Yes
SAS may run on	Windows	Windows	Windows, Unix, OS/390	Windows, Unix, OS/390
Distributed processing with thin client	Not recommended	Not recommended	Yes	Yes
Control of SAS by VB and SAS Feedback -> VB	Very limited	Limited	Important	Similar to EG
Access to results	Very limited	Limited	Diversified	Similar to EG
VB has access to SAS Data Sets	No	No	Yes	Yes
Functionalities of EG and possible Integration with MS Office	No	No	No	Yes
Development Effort	Very small	Small	Important	Important





Documentation

- “Command Line” and “SAS OLE Automation”:
SAS OnlineDoc
- EG Task Addin:
<http://support.sas.com/documentation/onlinedoc/guide/release30/addins/>
- Integrated Object Model:
 - <http://support.sas.com/rnd/itech/doc/dist-obj/winclnt/index.html> (Windows Client)
 - <http://support.sas.com/rnd/eai/oledb/index.htm> (Access to SAS Data)
 - **sas.chm**, **saswman.chm** et **sasoman.chm** files located at “C:\Program Files\SAS Institute\Shared Files\Integration Technologies”
- Visual Basic .NET:
 - Microsoft Developer Network (MSDN) Library, Microsoft Corporation 1987-2004





Questions / Comments

Yves DeGuire
SAS Technology Centre
System Development Division
Statistics Canada
613-951-1282
yves.deguire@statcan.ca

Maurice Djona
SAS Technology Centre
System Development Division
Statistics Canada
613-951-5331
maurice.djona@statcan.ca

