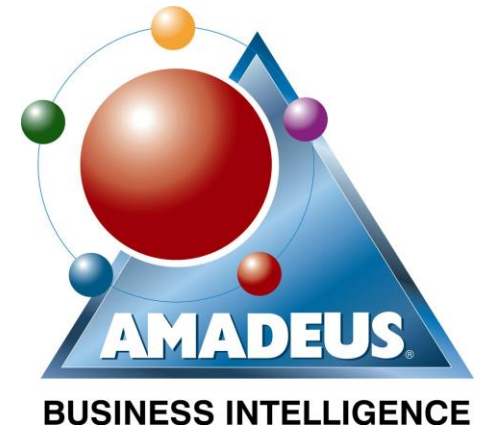




The Future's Bright The Future's .NET

David Shannon
Amadeus Software Limited
Views 2003



Topics to be Covered

- **Introduction**
- **What's Happening to SAS/AF?**
- **Technology, Technology Everywhere... but not a Perfect Answer in Sight**
- **Putting on Your Face: Web, Windows or Both?**
- **The Future's Bright**
- **The Future's .NET**
- **Conclusions**

Introduction

- **Application development with the SAS System has evolved:**
 - Originally used windows and frames contained in the SAS display manager environment
 - SAS/IntrNet brought the possibilities of client – server model
 - Integration Technologies gives the option of scalable development with either thin-client browser or a richer-client desktop application
 - Much has been written and discussed of the thin-client approach – this presentation concentrates on the desktop approach and to harness the power of the SAS System, whilst providing highly intuitive user interfaces

What's Happening to SAS/AF?

- **First released in around 1985**
 - At the time, virtually the only colour cross platform application development tool
 - Underwent several enhancement through the 1990's, notably for the release of SAS V8
- **Emphasis began to shift towards the end of V6**
 - Thin-client strategy adopted more
 - SAS/IntrNet module made available
- **Version 9 cometh...**
 - No new features added to SAS/AF
 - Is still supported by SAS Institute
 - Unofficially, Cary say it will continue to support it as long as there is demand...



What's Happening to SAS/AF?

- **Many applications exist in SAS/AF**
 - No need to immediately migrate applications
- **Long term strategy should at least consider options for migration**
 - **Concerns raised are:**
 - Investment made in SAS applications
 - Redundant skills sets
 - **Note that SAS Institute have developed SCL for V9**
 - **The getApplication() IOM method allows a call to SCL Classes through COM/DCOM etc.**
 - **This enables a strategy that distinguishes between skills sets**

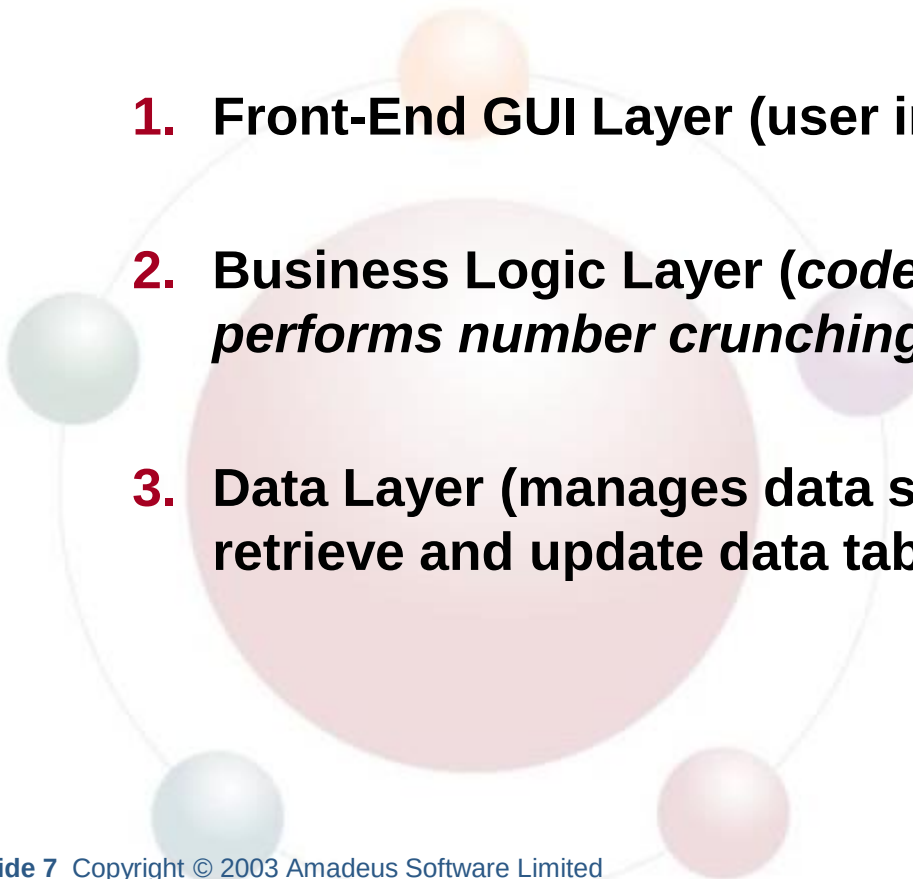


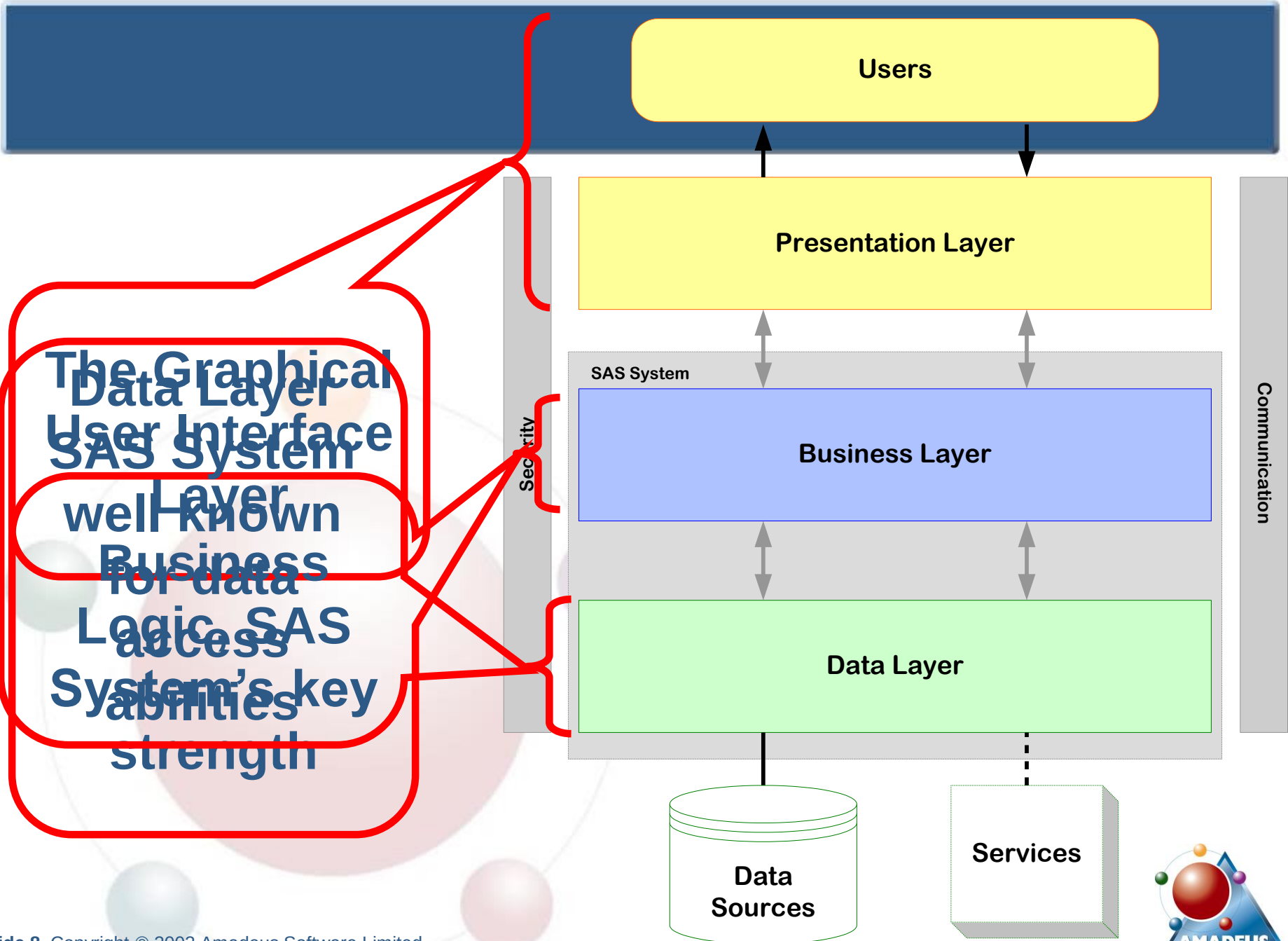
What's Happening to SAS/AF?

- **Why have SAS Institute taking this approach?**
 - Undoubtedly many resources have been devoted to SAS/AF
 - Emphasis shifted to web browser solutions in recent years
 - Growing demand for open and integrated solutions
 - Demand satisfied by introducing modules...
 - SAS/IntrNet for CGI web applications
 - AppDev Studio for developing SAS centric applications with a JSP focus
 - Integration Technologies opens SAS to standard communication protocols such as COM, DCOM, CORBA
 - The demand for open integrated software surely makes SAS/AF development infeasible?

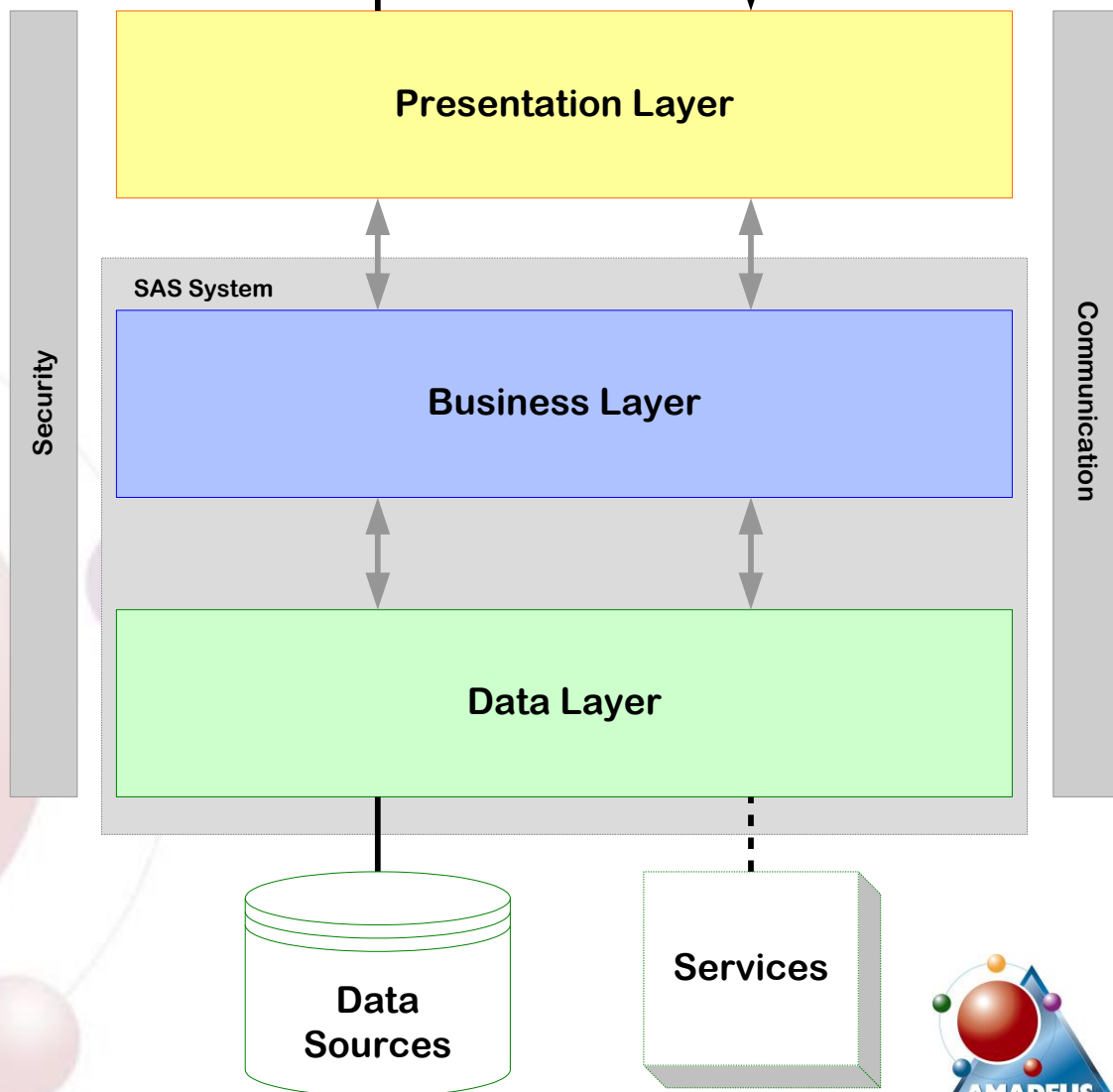
Technology, Technology Everywhere... but not a Perfect Answer in Sight

- **An ideal development strategy involves layers**
 - Logical layers distinguish between components in the application
- 1. **Front-End GUI Layer (user interface)**
- 2. **Business Logic Layer (*code which derives outcomes, performs number crunching etc.*)**
- 3. **Data Layer (manages data sources, performs the work to retrieve and update data tables)**

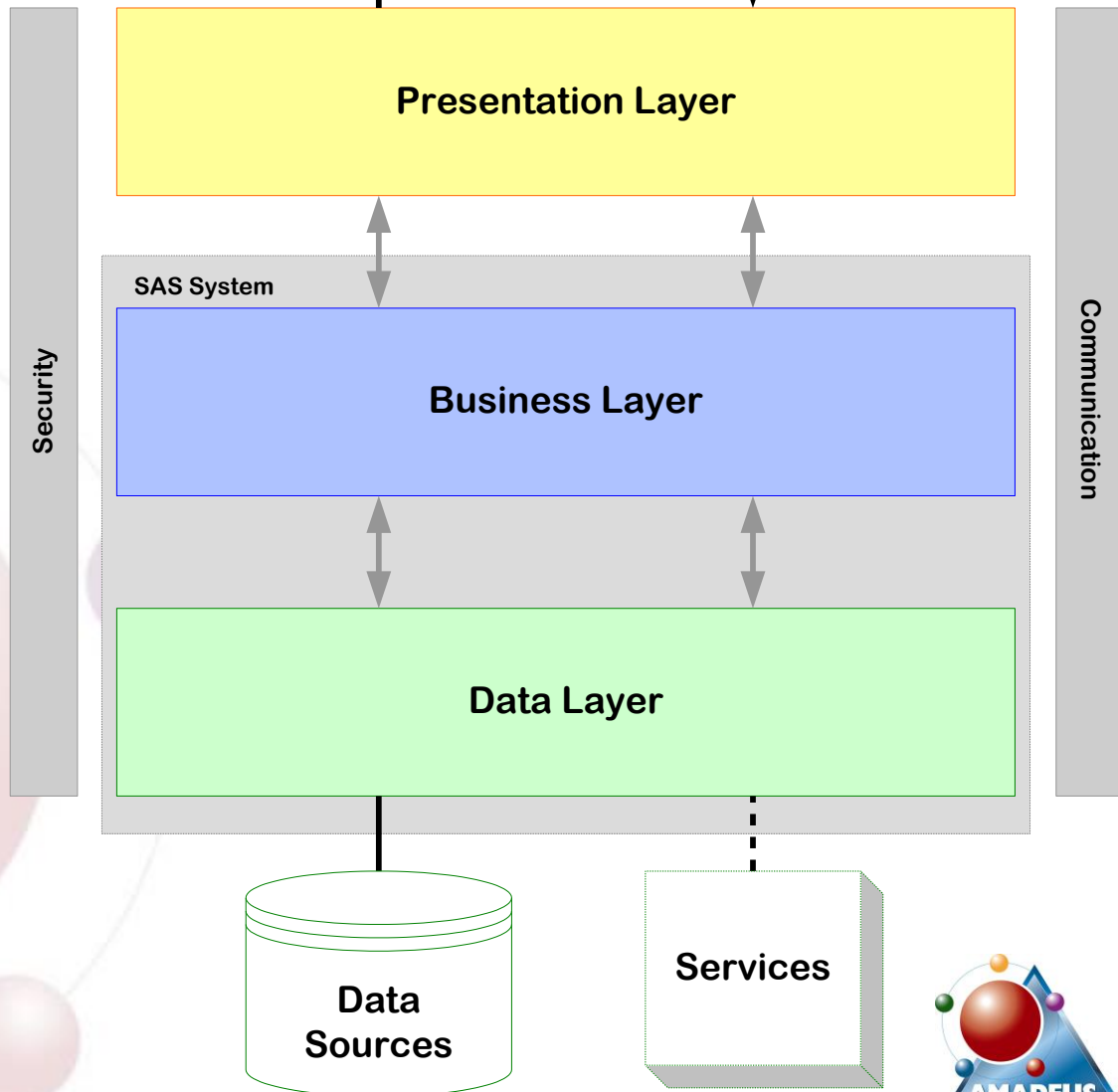




- Consider security & communication throughout
- The topic of services is worthy of a paper in its self (4pm - Hunt Room)

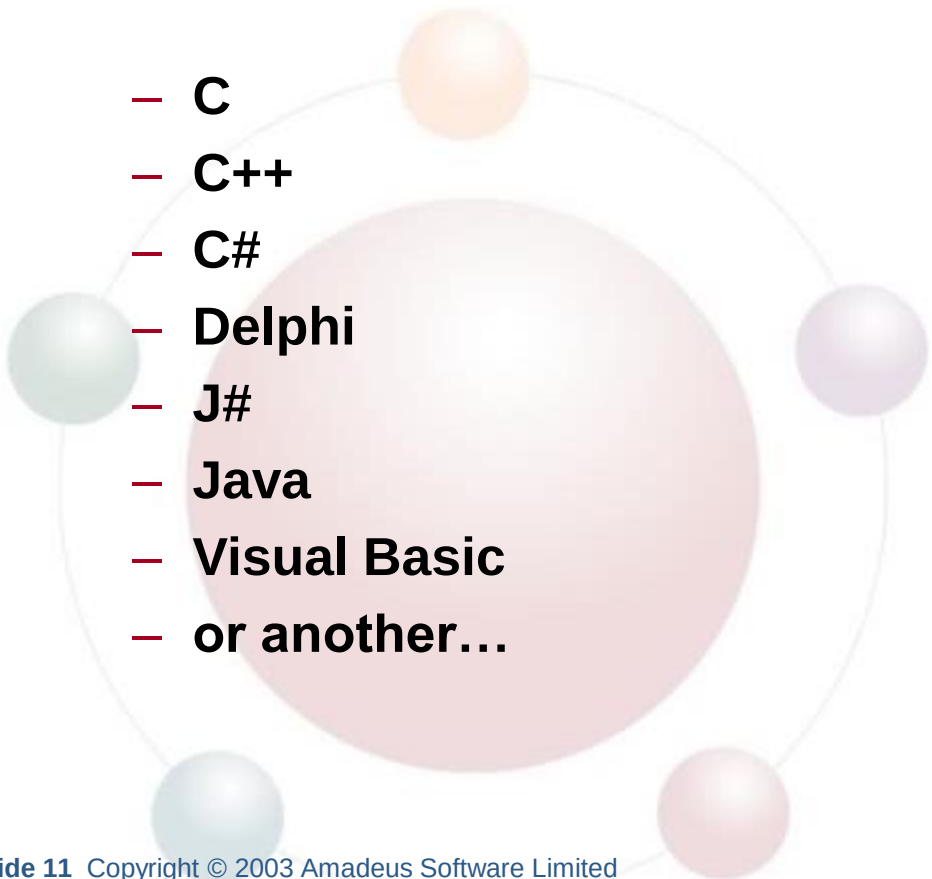


- The SAS System is clearly identified in our logical model
- Now let's concentrate on the presentation (or GUI) layer



Technology, Technology Everywhere... but not a Perfect Answer in Sight

- Which language should be used to build the user interface?

- 
- C
 - C++
 - C#
 - Delphi
 - J#
 - Java
 - Visual Basic
 - or another...

Technology, Technology Everywhere... but not a Perfect Answer in Sight

- Which language should be used to build the user interface?
 - **Potential for dilemma!**
 - **Consider factors such as:**
 - Companies IT strategy
 - Existing hardware infrastructure
 - Availability of skills sets for implementation and maintenance
 - **Critically don't overlook the users**
 - What is the primary function of the application?
 - Are users requiring remote access or widely dispersed?
 - What level of interactivity is involved?

Technology, Technology Everywhere... but not a Perfect Answer in Sight

- **So why Microsoft and why .NET?**
 - Over 80% of SAS users run SAS on a Windows operating system
 - This follows as, around 98% of PC's run on Microsoft Windows
 - SAS Institute have enabled interoperability, focussing on two ways:
 - Microsoft's COM
 - Java
 - **SAS developer community (April – May, 2003)**
 - Ten of the 11 SAS Institute samples are written in Microsoft languages
 - Only one is Java

Technology, Technology Everywhere... but not a Perfect Answer in Sight

- **So why Microsoft and why .NET?**

- **Cost:**

- Visual Studio .NET costs ~£1000
 - Gives four languages and developer edition operating systems and databases

- **Everything's gone .NET**

- .NET means Microsoft's latest generation of software, everything from Windows to Word
 - Future proof through the typical life span of an application
 - Runs on any Windows OS from Windows 98SE and newer
 - Built in to Windows XP (service pack 1)



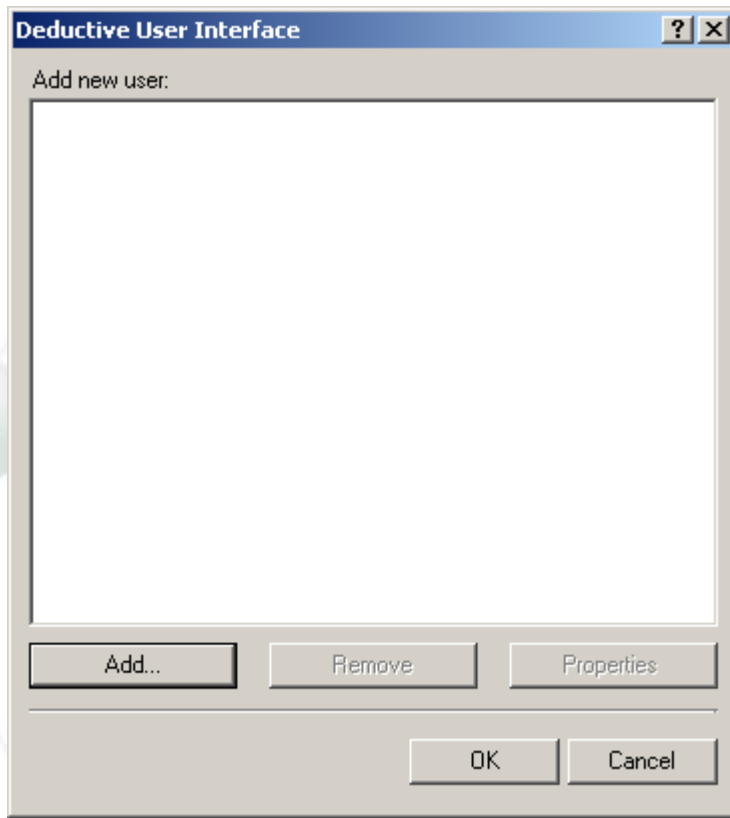
Technology, Technology Everywhere... but not a Perfect Answer in Sight

- **So why Microsoft and why .NET?**
 - **And not least, the applications end user!**
 - Support for seamless integration into other desktop software, e.g. Microsoft Excel
 - Consistent Windows look and feel
 - Most often user opinion deems whether software is successful or not
 - Studies show that, remarkably, the majority of opinions formed about software are on the physical presentation and appearance to the user
 - **System administrators demands can be satisfied**
 - More later...



Technology, Technology Everywhere... but not a Perfect Answer in Sight

- **So why Microsoft and why .NET?**
 - Intuitive interfaces are key to productivity and user satisfaction



A traditional deductive user interface

Technology, Technology Everywhere... but not a Perfect Answer in Sight

- **So why Microsoft and why .NET?**
 - Intuitive interfaces are key to productivity and user satisfaction



**An inductive
user
interface**



Putting on Your Face: Web, Windows or Both?

- There are several possible interfaces – each able to exploit the power of SAS

.NET Windows Forms

Windows Desktop Application

Offline Client (e.g. Laptop whilst disconnected from LAN)

Smart Client (e.g. PDA)

Putting on Your Face: Web, Windows or Both?

- There are several possible interfaces – each able to exploit the power of SAS

ASP .NET

Web Browser User Interface

Web Digital Dashboard

Thin Device

Putting on Your Face: Web, Windows or Both?

- **There are several possible interfaces – each able to exploit the power of SAS**

Microsoft Office & Other

Office Applications (e.g. Excel workbook data)

Outlook Messaging

SharePoint Server

Putting on Your Face: Web, Windows or Both?

- **Web has been the flavour of choice**
 - Demand has been to deploy distributable applications to end users through web browsers
 - Allows ease of deployment
 - Allows ease of maintenance
 - Leads to cost effective support
 - Common complaint: High level of latency
 - Common complaint: Desktop interactivity
 - Web deployment, in my opinion, is clearly appropriate in the right scenario... but by no means all

Putting on Your Face: Web, Windows or Both?

- **All aboard the merry-go-round**
 - Since the emergence of .NET there are clear signs of a shift towards the richer desktop application
 - Bourne from user frustration application architects are under pressure to deliver rich, intuitive environments for users to work in
 - This is quite possible with .NET – but better still this can be achieved with the maintenance benefits of a thin client solution



Putting on Your Face: Web, Windows or Both?

- All aboard the merry-go-round

“No Touch Deployment”

“No Touch Update”

- Terms associated with applications built in a .NET language such as Visual Basic or C#
- Means we use the web to deploy and upgrade rich-client desktop applications
- Allows the client software to upgrade itself without the user (or administrator) ever needing to interact with the software
- What are the possibilities?



Putting on Your Face: Web, Windows or Both?

	URL Launched App	Code Download	AppUpdater Component
No-Touch Update	Yes	Yes	Yes
No-Touch Deployment	Yes		
Dynamic Modules		Yes	
Offline			Yes
Full Trust			Yes

Putting on Your Face: Web, Windows or Both?

- **Combining the features of:**
 - A true Windows desktop application
 - Minimise administration by utilising no touch deployment and/or no touch updates
 - Harness the power of the SAS System...

Builds a winning solution for both user and administrator



The Future's Bright

- I'm keen not to miss one crucial point

The User

- **Traditional SAS applications have been cumbersome**
 - E.g. lots of mouse clicks required to achieve a single task
 - Ability to create the intuitive user interface with SAS/AF has been all but impossible



The Future's Bright

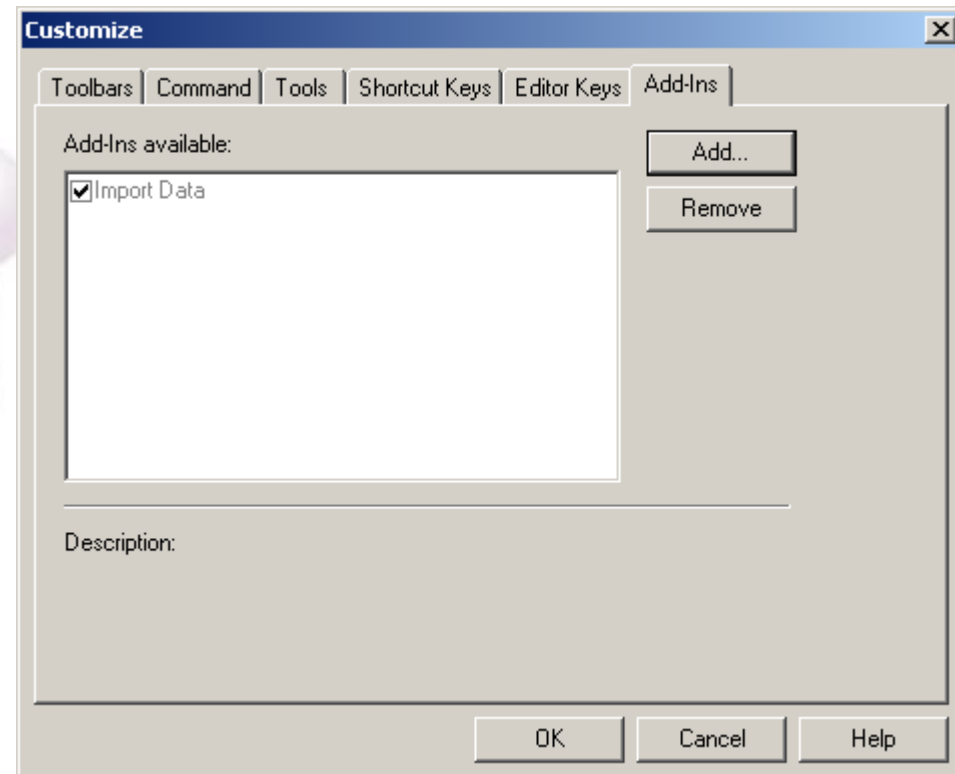
- **With .NET software...**
 - **Enable every day Windows environments for users to work in**
 - **Make use of the computing power of today's desktops**
 - **Consider the features in any Windows software – consider using those features in a bespoke .NET application**
 - **There are, of course, several possibilities for building applications that interact with SAS**

The Future's Bright

- **Enterprise Guide V2**

- The alternative front-end from SAS Institute to the SAS System for Windows
- Newest version supports add-ins.

**Build add-in with
Visual Studio, build
in to Enterprise
Guide V2**

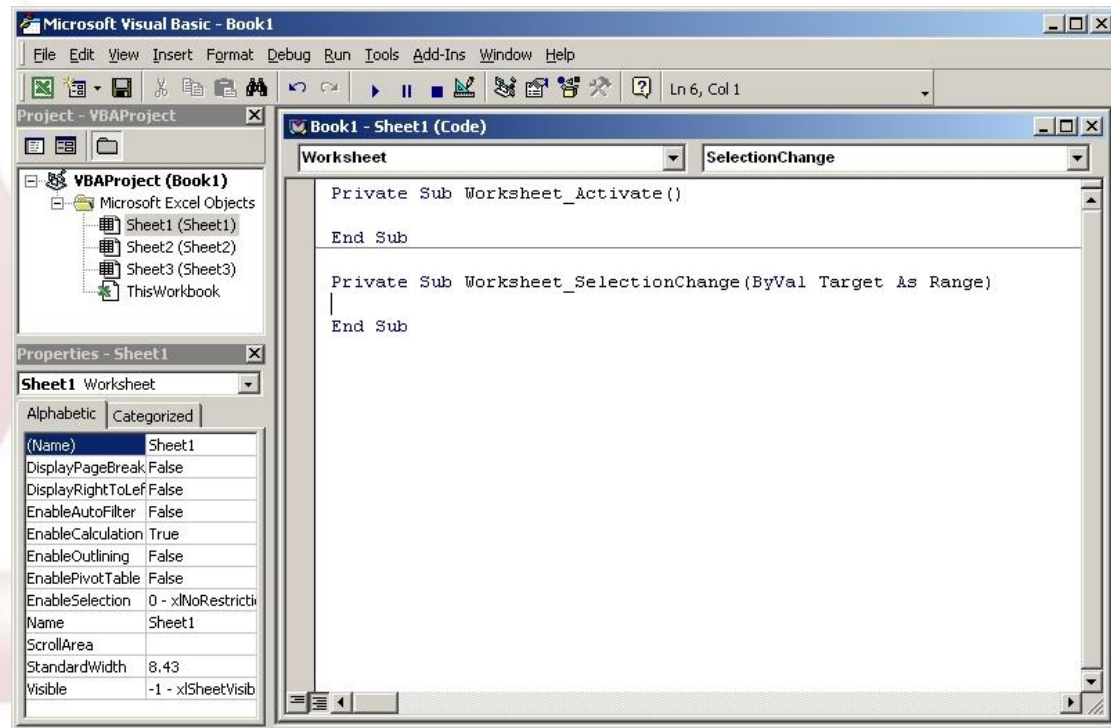


The Future's Bright

- **Container Applications**

- Application functionality resides wholly inside a container application
- E.g. Using VBA in Microsoft Excel to interact with SAS System's functionality

- Suitable for appropriate scenarios
- Does not create compiled code
- Not just Microsoft



The Future's Bright

- **Web Applications (ASP .NET)**

- The possibilities for interoperating with SAS via ASP .NET web applications are considerably furthered since classic ASP (3.0)
- Some SAS features are richer than SAS Institute's own AppDev Studio primarily orientated at JSP development
 - E.g. the ability to update SAS data in a grid through a web browser is not available in AppDev Studio until the next version
 - This functionality is possible with Visual Studio built ASP .NET pages for over a year
- Web applications are ideal with widely dispersed users or application of few steps from end-to-end with non-constant user interaction

The Future's Bright

- **Windows Forms**

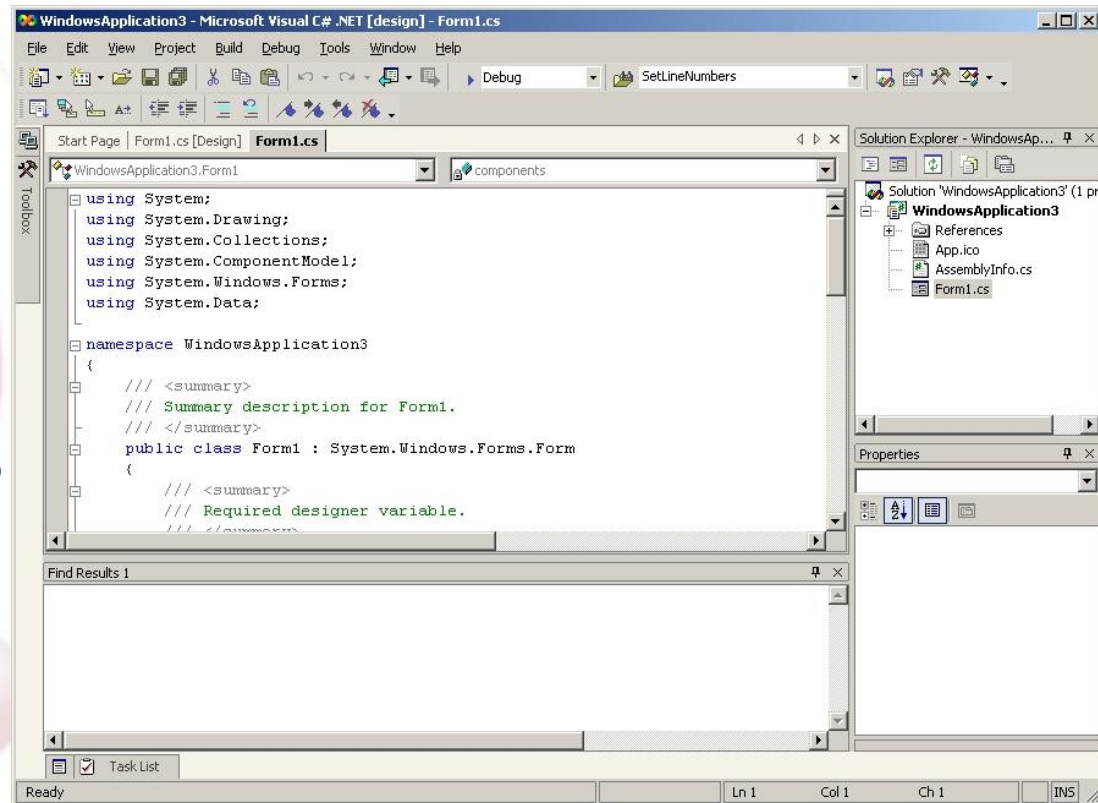
- Build bespoke applications
- Provide the most functionality, interactivity and stability
- Designs are rapidly advancing:
- E.g. Windows XP

- **Interact with Other Software**

- Office
- Databases

- **No Touch Updates**

- Build in ease of maintenance



The Future's .NET

- **Microsoft's .NET allows both**
 - Rich intuitive interfaces
 - Coupled with low overhead deployment
 - Ease of interoperation with the SAS System
 - Harness what the SAS System does best either,
 - Locally (same machine) or
 - Remotely (client – server)
- **Layered Application Design**
 - Often called *n*-tier (commonly 3-tier designs)
 - Distinguish between SAS and developer skills sets



Conclusions

- The possibilities for harnessing the power of the SAS System go beyond that of SAS/AF
- Consideration should be given to logical layers of functionality
- .NET applications do not need to be a compromise for users or administrators
- Finally the point is the user... a SAS application exists to fulfil a business function of its user, architects and designers shouldn't forget this

**So... the future is bright,
the future may well involve .NET**





Questions on The Future's Bright, The Future's .NET?

David Shannon

Senior Consultant

Amadeus Software Limited

Email: david.shannon@amadeus.co.uk

Web: www.amadeus.co.uk

