

FOLDER NAME : Bobmu2

PRESENTATION: API Strategy - Final

DATE CREATED: 6/2/95



MS6 5006840
CONFIDENTIAL

Microsoft API Strategy

September 19

MS6 5006841
CONFIDENTIAL

Situation

- Windows application development is the standard for ISV, SP, and corporate developers
 - Key Win32 APIs are wrapped by VBRUN, MFC, Delphi,...
- New, non-MS platform is emerging
 - Driven by Internet distribution
 - Java based
 - Cross-platform
 - Network centric
 - Windows agnostic

Implication

- Computers are now sufficiently powerful for Java middleware to succeed for form and content-centric applications
 - Driven by Internet distribution.
- MS has lost significant control over innovation for the new Internet platform
 - Sun's Java.* classes are well established

Microsoft Goals

- Provide compelling reasons for Windows development
- Establish leadership in the new Internet platform, distributed computing environment and programming model
 - Match the competition with Java and cross-platform (OS) support
 - Support rapid, maintainable development of distributed applications
 - language neutral, seamless in Java

Microsoft Goals

- Re-establish the foundation for innovation
 - Regain control of the platform definition and innovation
 - End the least common denominator phenomenon
 - Leverage our distributed services, interoperate with others
- Win by: Creating a compelling developer message
 - HTML leadership
 - Being the best offering on Windows; making Windows the best for the Internet
 - Support for all mainstream languages (not just Java)
 - Great Microsoft tools, great support from systems for Microsoft Tools

Objectives

- ① Developer adoption of Windows-specific interfaces
 - Optimize for Windows
- ② Developer adoption of MS-specific cross-platform libraries
 - Stick with Microsoft, stay in the program
- ③ Developer adoption of our tools and frameworks
 - Allows us to steer the community on the best path

Application Categories

- Conventional, "thick" Windows client apps
 - Office
 - Not at short-term risk

■ Client/Server forms apps

- Corporate, vertical
- Moving to Web-based technology "Weblications"

■ Web-based content apps, multimedia

- CD-ROM today, "Weblication" tomorrow
- Encarta, Amazon.com

■ Collaborative apps

- Realtime: Netmeeting, chat
- Workflow: Notes, Exchange
- Moving to Web-based technology over next 2 years

■ Games

- Quake, Monster Truck Madness
- Serious ones not at risk right now

Category Messages

■ Client/Server forms apps

- Transition simple apps to Web technology
- More complex apps will require better tools
- Based on: Trident, Denali/Viper, High level language frameworks

■ Web-based content apps, multimedia

- Leverage HTML features of Trident
- Leverage Windows-specific class libraries based on COM Windows, including Trident
- For Cross-platform support, use our tools and class libraries subset

Strategy for Content & Client/Server Developers

More on Server  Leverage IE share to focus
Less on Client  on MS extended HTML

Client development  Netscape ONE, Java.* only
 MS Class libraries

- Why?
- Great support for IE on Win32
 - Great services, tools, size, speed
 - Great server support
 - Reasonable cross-platform support
 - IE, Mac, Unix
 - Reasonable Navigator support

Where we are today

- MS has taken the lead in HTML
 - New Navigator 4 object model extensions indicate their HTML support is read-only
 - Trident can incorporate Netscape extensions, go beyond
- Java environment deeply entrenched
 - Sun's Java.* classes, bytecodes well established
 - Netscape ONE could fragment the market
- Windows API inaccessible from Java
 - MS has extended Java in IE 3 with COM, AWT extensions, DirectX, but we need to do more

Where we are today

- ActiveX controls are platform and browser dependent
 - ActiveX controls on Windows are tied to User
 - All controls call Win32 APIs
 - Support in Navigator is weak
 - Today's Mac and Windows embedding/control interfaces are not compatible
 - New ActiveX controls for Mac use Mac-specific interfaces

Implications:

- Cross-platform ActiveX Control support requires a higher-level abstraction and new code
- Support in Navigator for ActiveX controls must be fixed

Approach

- Tools drive business developer adoption
 - Compelling tools drive adoption of MS strategy
 - Tools determine the libraries, frameworks used in solutions
 - Tools are the hook to developer information programs
- Define services which are compelling to developers and are only available on Windows
 - "Your competitors will take advantage of this stuff..."
- Where developers demand cross-platform capabilities, supply a superior solution so that they are using Microsoft interfaces and tools
 - Above all, keep them in the program, keep them using our tools

Competition - Netscape ONE

- HTML
- Javascript
- Plug-ins
- Java Applets
- Live Connect (including IIOP)
- Internet Foundation Classes (IFC)
- Java Runtime Environment
- Browser protocols: (HTTP, LDAP, IIOP, SMTP, ...)

Competition - Netscape ONE

- Liveconnect is object model
 - Netscape equivalent to JavaBeans
 - Supports IOP for distributed computing
- Internet Foundation Classes
 - UI (Basic controls, multifont text, windowing animaption, drag/drop, single threaded concurrency, object persistence, localization)
 - Messaging: SMTP, POP3, IMAP, NNTP
 - Security: SSL, X.509v3 certs, signatures, extensibility
 - Distributed Services: IOP exposure

Competition - Javasoft

- Java.* is Java's "studio" class libs for the '90s:
 - AWT, Net, Lang, IO, JDBC
 - AWT
 - Simple windows, frames, widgets
 - Simple events
 - Layouts
 - Net - classes for networking
 - TCP/IP sockets, URLs (http, ftp, gopher, wais, ...)
 - IO - file i/o
 - sequential and random access
- Borland's Latte and Symmantec's Café are high quality RAD tools targeted at Java.*

Competition - Javasoftware New Classes

- Java Beans Embedding/Control support
 - Ties Beans to AWT
 - Clipboard, Keyboard accelerators, Drag/drop, "Uniform Data Transfer"
- Media - Framework:
 - 2D: (Adobe), 3D: (Intel, SGI)
 - Animation, Audio, Collaboration, Video, Telephony
- Security - Authentication, code signing, encryption
- Commerce (Wallet)
- Network Management: (SunSoft)
- Late: Q2 '97?

Competition - Java Beans

- **Component model for Java Platform**
 - Higher level than Java classes
 - Leverages java. and VM facilities (likely to be folded back into java.)
 - major overlap with OLE (interop "for free")
 - Additional tool-centric featureset
- **Endorsed by system and tool vendors - Borland, IBM, Netscape, Sun**
 - Sun designed infrastructure (serialization, etc.)
 - Component features designed and engineered by Borland
- **Current plans are dependent on AWT**
 - Implication: Container and all services must be AWT based

Java Beans Facilities

- Interface publication and discovery: “Reflection”
- Event handling
- Persistence
- RAD infrastructure: Inspectors and editors
- Layout: Embedding (still TBD)
- Distributed location and activation: RMI and Corba

Competitive Comparison

Category	ActiveX/COM	JavaBeans	NextStep/ObjC
Interface Publishing & Accessibility	<ul style="list-style-type: none"> Hyperbbs Query Interface 	<ul style="list-style-type: none"> Reflection Introspection APIs 	<ul style="list-style-type: none"> Java Reflection API Introspection (JDK1.1) Java Compiler
Event/Interface/Component	<ul style="list-style-type: none"> ActiveX Events Model 	<ul style="list-style-type: none"> Beans events model 	<ul style="list-style-type: none"> Persistence Model-View-Controller (MVC) Model-View-Controller (MVC) Model-View-Controller (MVC) Model-View-Controller (MVC)
Persistence	<ul style="list-style-type: none"> Persist 	<ul style="list-style-type: none"> Beans serialization Specific control 	<ul style="list-style-type: none"> Persistence Model-View-Controller (MVC) Model-View-Controller (MVC) Model-View-Controller (MVC) Model-View-Controller (MVC)
Layout	<ul style="list-style-type: none"> ActiveX Control/DocObject Layout Specs 	<ul style="list-style-type: none"> Beans (AWT based) Embedded layout Spec 	<ul style="list-style-type: none"> Model-View-Controller (MVC) Model-View-Controller (MVC) Model-View-Controller (MVC) Model-View-Controller (MVC)
Security	<ul style="list-style-type: none"> SSL/ARC4/NTL/SSIP 	<ul style="list-style-type: none"> Unclear 	<ul style="list-style-type: none"> SSL
Distributed app services	<ul style="list-style-type: none"> DCOM/RPC 	<ul style="list-style-type: none"> RMI CORBA 	<ul style="list-style-type: none"> IIOP (part of CORBA)
Database connectivity	<ul style="list-style-type: none"> ADO, OLE-DB 	<ul style="list-style-type: none"> JDBC 	<ul style="list-style-type: none"> JDBC

DCOM/IOP: Competition

- **Sun: Java RMI (Remote Method Invocation)**
 - Highly criticized as awkward and non-intuitive
 - In beta with free DOOR ORB (non-IOP)
 - Integration with Sun's JOE/NEO product line key
 - Expect IOP and built into Java 1.1 (Fall release)
- **Netscape: bundling Visigenic's VisiBroker**
 - Technology acquired through merger with PostModern
 - More transparent, simpler than RMI, very similar to Java/(D)COM integration
- **OMG: A spec with vendor implementations**

DCOM/IIOP: Technical Overview

■ DCOM

- *Security*: pluggable RPC security via SSPI. NT security primarily. no SSL/SET SSPI plan-of-record.
- *Transports*: most network transports, including TCP. final HTTP-tunneling plan incomplete.
- *Performance*: ~15% over raw TCP sockets. Excellent scalability.
- *Tools*: well integrated with most languages. supports arbitrarily complex types. works with GUI applications.

■ IIOP

- *Security*: none OMG-defined. expect NS to use SSL & SET, certificates for authorization (access-checking)
- *Transports*: TCP-only. no HTTP-tunneling.
- *Performance*: varies wildly. NS and Sun under investigation.
- *Tools*: weak for C/C++, type-limitations are constraining. works poorly with GUI. Java is unfortunately a great fit. Sun's JavaIDL and NS's VisiBroker SDK's are at least as good as our Java/COM integration tools (JavaTLB).

Competitive Summary

- Sunsoft's core classes have developer mindshare
 - Will be significantly enhanced in V2
- Netscape's Internet Foundation Classes will be driven/limited by future success of Navigator
 - Cross-browser claim yet to be tested
 - Primarily consumer driven, not likely to have major impact in corporations
- MS ability to lead Java developers is largely driven/limited by IE share
- DCOM/IIOP will be driven by time to market

Recommendations

- Follow similar strategy to Browser: Embrace and Extend - particularly with Java and Java.*
- Focus on synergy between IE and Tools
 - HTML + Script + Active controls as target platform
 - Host IE features as a plugin in Navigator
- Make integration between Java and ActiveX as seamless as possible
- Adopt JavaBeans as Java language integration for OLE/COM

Recommendations (cont'd)

- Build out existing language-specific frameworks (VBRUN, MFC, Java.*)
 - RAD depends on language / framework integration
 - Make language-specific part as small as possible
 - Make most services language independent
 - Fix existing problems: Versioning, typelibs, ...
 - Create system COM interfaces that are optimized for support of frameworks

Recommendations (cont'd)

- Windows as language-independent COM
 - All new system services as language-independent, well-structured COM objects and interfaces
 - Restructure the most leveraged Windows services as well-structured COM objects
- Focus is on simplifying the Webication developers job on Windows
- Platforms and tools work together on this - must be great for language framework integration
- Make a subset of these services available cross-platform
 - "Mechanically" think Windows APIs to COM to allow Java programmers access to Windows APIs

Recommendations (Con't)

- Make DCOM Adoption a High Priority
 - Establish Openness with OpenGroup
 - Ship DCOM in IE4
 - Tools adoption in VB, VC, VJ
 - Add COM/DCOM support to Netscape Navigator
 - Provide a MS COM/CORBA? Or leave to 3rd parties?

Alternatives & Scenarios

- **Alternative Strategy: Build MS Java classes to compete with Java.***
 - Java.* has momentum, we'll never catch up
 - Java.* is the "studio" of Java, we'll look like we don't get it
- **Best Case Scenario: Fragmented Class definitions**
 - Netscape/Sun fragment Java Classes, object model
 - Even a fragmented Java will be successful, but will target Sun-defined least common denominator initially
 - Leaves MS time, opportunity to build Windows-based momentum around our definitions, tools
- **Worst Case Scenario: Sun/Netscape/Borland team up on Java classes and Java tools**

Top Priorities for COM Windows

- Our value add is making Networked Windows Weblications easy
 - Java.net only defines URL, Sockets interfaces
 - We can (must) take leadership on advanced services
 - Sunsoft is focused on "wrapping" existing definitions
- All work must be tuned for language integration, especially Java
- Build in our fundamental distribution technologies
 - Load balancing, name transparency (DCOM vs. RMI)
 - Transactions

Top Priorities for COM Windows

- Storage/Directory
 - Folders/Namespace/DFS
 - Directory service management and access - OLE DS
 - Property access
- Database query and access (vs. JDBC)
 - ADO, OLE DB
- Security (vs. JECF)
 - Authentication, Authorization, Wallet, Encryption, Signing, Safety
- Simple Messaging

Single Windows Model

- We must have a single *Windows* story for client and server components/extensions
 - Current market situation mandates Netscape support
 - Apache should be guided to use our model on NT
 - More than APIs, it's also the complete model and environment
- Requires:
 - Plug-in for Navigator to run MS HTML and COM services - utilizes DCOM
 - Denali and Viper extensions work under NT Netscape Server (NSAPI), NT Apache server - utilizes DCOM
 - Common controls as Java classes for cross-browser/platform
 - Strategy for making UNIX Apache, Netscape migration easy

Windows Advantages

- *Single platform target* in a time of fragmentation (client and server)
 - need to regain this
- Client market share
 - ISVs need to, will optimize for Windows
 - New things appear on windows first
- Windows has the complete version of...
 - Full IE
 - Full Trident
 - Full set of Windows Interfaces (available to Java)
 - Full set of client-side database APIs
- Microsoft is focused on Windows
 - Performance will be best on windows
 - New things appear on windows first
- Multimedia/graphics is best on Windows
 - Great DirectX
 - Fastest, best support of new devices
- Focus of Distributed services innovation on both client and server
- Active Server platform
 - by far the coolest server programming environment

Windows Innovation

- In the past, represented as ongoing flood of new APIs
 - This is continuing
 - It is becoming a liability -- too much complexity
- Focus **MUST** be on simplicity as we move to COM interfaces
 - Easy to figure out
 - Easy to access new services
 - Easy to exploit the network from Windows
 - Language / Framework integration
- Security blanket
 - We're here for the long run
 - We protect your investments

Cross-platform Support

- Goal: Create a safe Active Control binary which can run in on Win32, Mac, Unix
- Requires:
 - Container platform: Internet Explorer, Navigator
 - Middleware API: Java.* + X-Platform COM Subset
 - Trident subset (No DirectX)
 - Drawing, controls
 - Database subset (No local storage)
 - Optionally Safe: Signature tells IE to sandbox control
 - VM
 - Single Microsoft VM
 - Extend Windows Java VM to support VB
 - Use Collusa technology to support C++, SQL, ...

Cross-platform Support

- Approach:
 - Design infrastructure for cross-platform Java and VB
 - Once design complete, decide whether it is worth supporting cross-platform VB or just using Java
 - Focus C++ products on high-performance Windows development

Language / Framework Integration

- Languages already promote a programming model
- Tool presents a user model for the developer
- Approach needs to be tool centric
 - Tools provide direction to developer - this needs to be consistent with our systems strategy
- Developers want straightforward access to system services
- By providing quality system services which meet developer needs, the language specific framework can be made thinner

Tools Evolution

- Finish Tools V5 releases
- Target IE 5 (or 6) at application development
- Focus on combining Internet Studio, Java, VB into RAD Internet development environment
 - HTML + Script + Active controls as target platform
 - Trident forms
 - Make IE the IDE
 - Client-side Active controls, server-side Denali, Viper objects
 - Written in Java and VB
 - Active scripting on Client and Server
 - Javascript and VB Script
- Enhance VC++ for current customers

Java Approach

- Use our Windows VM to own Java.* and Beans
- Build on Java.* , Beans as the Java framework
 - Extend these to run best on Windows
- Expose Win32 functionality in Java.* /Beans
 - Provide MS APIs where possible
 - Change Java.* /Beans to support MS objectives - ActiveX
 - Use Sun's approach where necessary
- Provide MS alternatives for other Sun services
 - Net, RMI, Media, commerce, security

MFC Evolution

- Minimal evolution of MFC 4.2
 - New system features as static link libraries
 - New common controls - Datetime picker, Coolbar, ...
- Future MFC
 - Implement new functionality as COM classes
 - Address Versioning
 - Investigate restructuring of MFC to reduce fragility
 - Stay compatible with MFC 4.2!

ATL

- Make ATL work very well in the MFC environment
 - Common marketing message
- ATL 2.0 for hard-core ActiveX controls creation
 - Keep ATL small and fast
 - Remain focused on components, not apps

VBRUN Evolution

- Leverage common infrastructure with Java
- Slimmed down & Versionable
 - Strip out Ruby
 - Package as standalone Active Designer that we deprecate over time
 - Strip out ExCode engine
 - VB should target the Windows VM
 - Move some VB objects into a core system runtime where they can also be shared by other tools.
 - E.g., Printer, Clipboard, Screen
 - We have already done this for Font and Picture

TypeLib Requirements

- Extremely compact, great performance
- Sufficient for all core languages
 - Type support for VB, Java, VB Script, JavaScript, SQL, and C/C++ including debugging
- Enable multiple providers
 - Dynamic creation
 - Language independent interfaces
- Easily queryable and updatable
 - Object description database
 - Generic operators for data manipulation
- Extensible

TypeLibs currently achieve none of these!

Typelib Recommendation

- Build a new core type system
 - Leverage many concepts from Java Class File
 - Support Typelib requirements
- Provide “Beans” view for Java
 - Concepts could be leveraged for other languages
- Provide “Typelib” view for downlevel OLE

Trident Forms

- Trident needs to become a core part of the programming model
 - HTML part of HTML+Scripting+Active Controls
 - Time to market is critical
 - Programming Models:
 - HTML Active Document
 - VB/Java Code attached to to Trident object model
 - HTML Active Controls
 - Form becomes a control runtime
 - HTML Active Dialogs
 - Forms used as dialog manager
 - Trident and AWT objects are independent

Drawing Approach

- Build language-independent COM interfaces to GDI and DirectX
 - Thin layer on top of GDI
 - Make subset cross-platform
 - Focus on ISV requirements
 - Fit these into AWT framework
- Why will ISVs use this?
 - Brings richer GDI/DirectX specifics to Java
 - Provides these services to other VB, C++,...
- We've already extended "Graphics" in AWT with "GraphicsX COM interfaces"
 - Beziers, font extensions, Intl., etc.

Database

- ADO is language independent API
 - Small OLE DB mapping under development
- Client side support to ODBC and OLE DB data
- 3 Tier through Active Data
- JDBC
 - Widely adopted, equivalent to RDO 1
 - Forward-only cursor, no bookmarks, no batch update, no snapshot semantics, no local indexing
 - Extend JDBC with ADO semantics

Developer Message

- HTML + Script + Active controls as target platform
 - Our RAD tools target this platform
- Target Language-dependent frameworks plus MS COM Windows Classes to take full advantage of Windows
 - Cross-platform ISVs target subset for LCD
 - MS will make your components work under Netscape client and server; Apache NT Server
- Use VIPER/DCOM for distributed apps under the Active Server environment
- Use C++ for platform-dependent, high performance components

How does Microsoft Win?

- Redefine the platform
 - HTML + Script + Active controls
- Embrace, extend, lead where momentum is with Sunsoft
- Do a better job than Net ONE for developers
- Developers use our tools
 - Approach, mindshare
- Requires leadership in Browser marketshare
 - Developers target MS API extensions
 - Focus on Windows

Summary of Necessary MS Actions

- Focus future IE and tools on “Weblications”
 - Define the new platform
 - Build RAD tools
- Java and Java.*
 - Extend Java.* classes with MS features, services
 - Provide MS alternatives for other Sun services
 - Adopt JavaBeans as Java language integration for OLE/COM
- Single Microsoft VM targeted by Java and VB
 - Use Collusa technology to support C++, SQL, ...

Summary of Necessary MS Actions

- DBSD
 - Expose Win32 functionality in Java.* /Beans
 - Focus DBSD API efforts on language-independent COM objects
 - Top priority is distributed services
 - Build language-independent COM interfaces to GDI and DirectX
 - Restructure the most leveraged Windows services as well-structured COM objects
 - Beat IOP with DCOM

Summary of Necessary MS Actions

- Cross Platform
 - Work with DonBrad to provide subset cross-platform
 - Get serious about ActiveX support under Navigator
 - Get IIS extensions to work under NT Netscape Server
- Communicate the message:
 - Focus each team on the new platform: HTML + Scripting + Active controls + Windows