Win32®, COM, And ActiveX™

Cameron Ferroni
Group Program Manager
Desktop And Business Systems Division
Microsoft Corporation

The Internet

- **Ubiquitous connectivity**
  - **No need to worry about modems, network protocols**
- **Applications don't have to be stand-alone any longer**
- **Web is the standard UI metaphor**

Activate Your Application

- **Connecting Win32® applications to the Internet**
- **Connecting OLE applications to the Internet**
- **Writing new applications for distributed computing and the Internet**

Demo – FTP Support

- Use wininet.dll to add FTP support - simple API
- Load and save images on a global server
- Application "publishes" data in native format

Along Comes The Web

- Put a UI on top of the Internet
- Browsers everywhere
- HTML documents as simple front ends
- PERL, CGI, ISAPI, on the back end
- Great for documents, people want more

Demo – Document Access
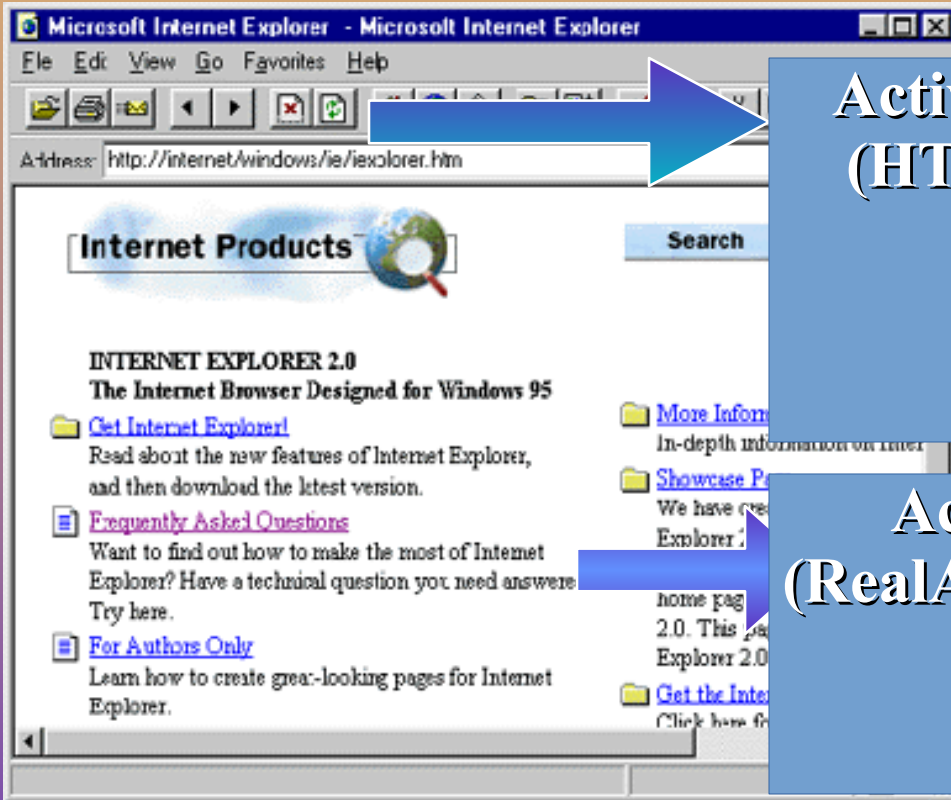
- Access FAQ from Help menu
- Dynamic Help information
- Publish information in one common format


```
ShellExecute(…, "open", "http://www.msn.com", …)
```

ActiveX Controls
Reusable, programmable objects

IUnknown

Active document (HTML viewer)

IOleContainer

IOleScriptSite

IUnknown

ActiveX controls (RealAudio, Shockwave)

IViewObject

IOleControl (optional)

Network services (URL moniker)

ActiveX Controls

- Embed controls easily into any container
- Add to a Visual C++® dialog box with zero code
- Use any ActiveX control, Internet- enabled or otherwise

Active Documents

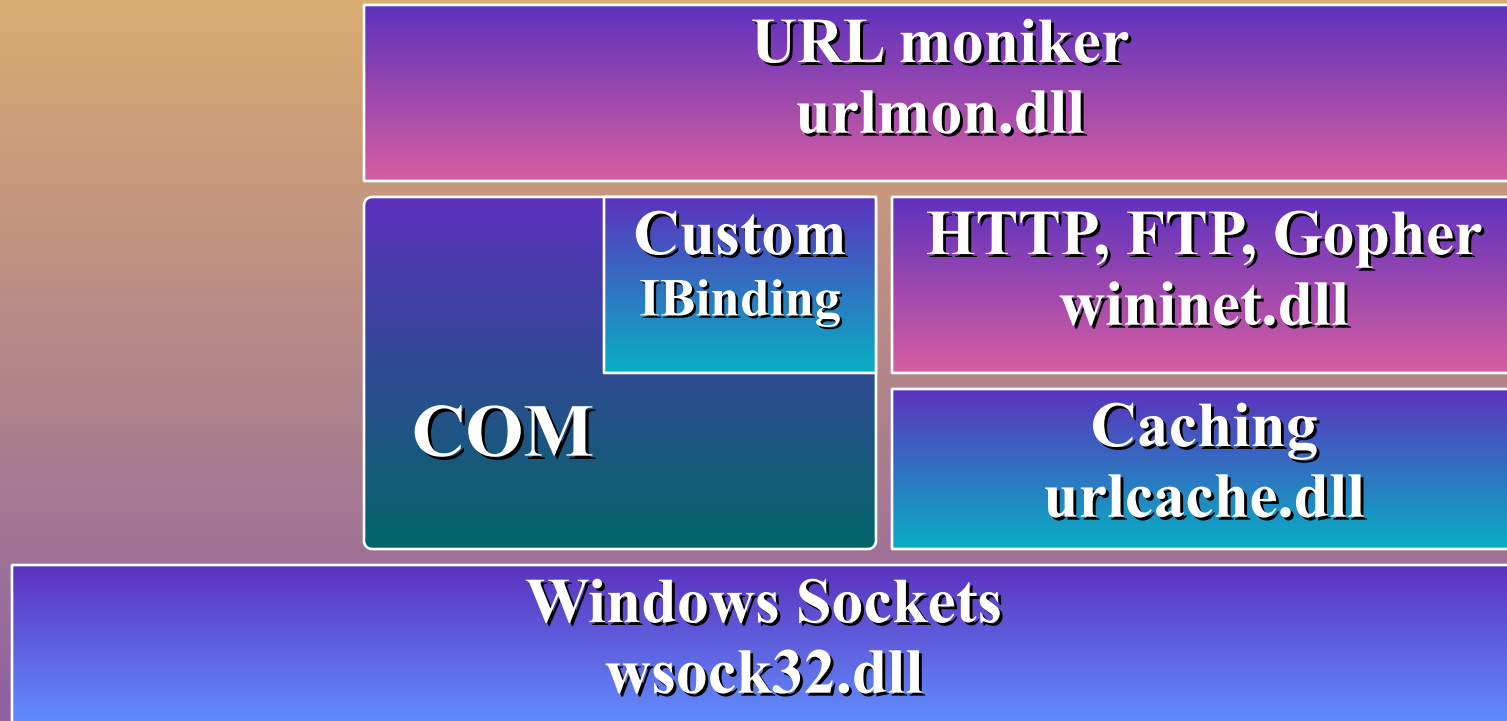**Container (Internet Explorer)**

E.g. (Visio, Word)

- IUnknown
- IOleObject
- IOleDocument
- IHLinkSite

Network services (URL moniker)

Demo – Document Object

- Existing application, existing file format
- Integrate seamlessly with the browser and the shell
- Easy navigation
- Add three interfaces if you are an InProc server

The Protocols – Today

URL moniker urlmon.dll	
Custom IBinding	**HTTP, FTP, Gopher** **wininet.dll**
COM	**Caching** **urlcache.dll**

Windows Sockets
wsock32.dll

- **HTTP is effectively one-way communication**
- **Use Windows® Sockets to stream the data**
- **Smart clients, smart servers**
  - **RealAudio, OnLive**

The Protocols – Tomorrow

	URL moniker urlmon.dll	
COM	Custom IBinding	HTTP, FTP, Gopher wininet.dll
		Caching urlcache.dll

Windows Sockets wsock32.dll

- Use COM interfaces directly from your application

Distributed COM
Leverage

- Existing skill set
- Existing tools
- Programmability
- Data marshaling
- Security
- Robustness

Existing Investment

- Any service you wrote to run out-of- process now works across the network
- Any existing 32-bit COM application just works with Distributed COM
- Any EXE generated with Visual Basic® 4.0 just works with Distributed COM
- Any MFC-generated OLE application just works with Distributed COM

Where Does ActiveX Fit In?

- Client can be either an active document, or an ActiveX Control
- For document objects:
    - A reference to the remote COM object can be returned as part of the document data
    - User can be asked for the name of the service to join
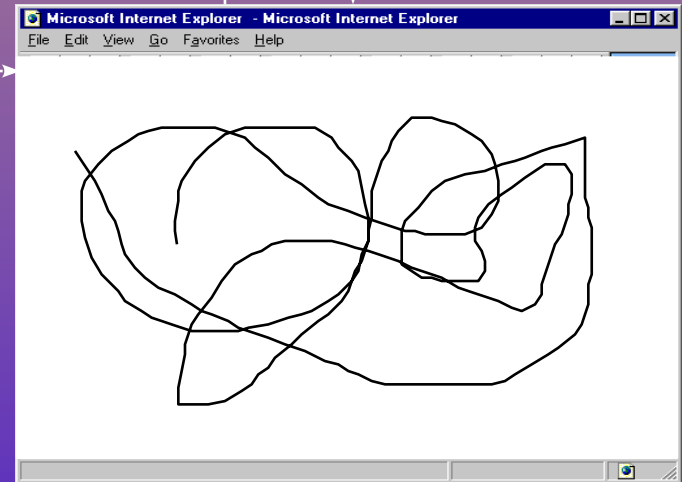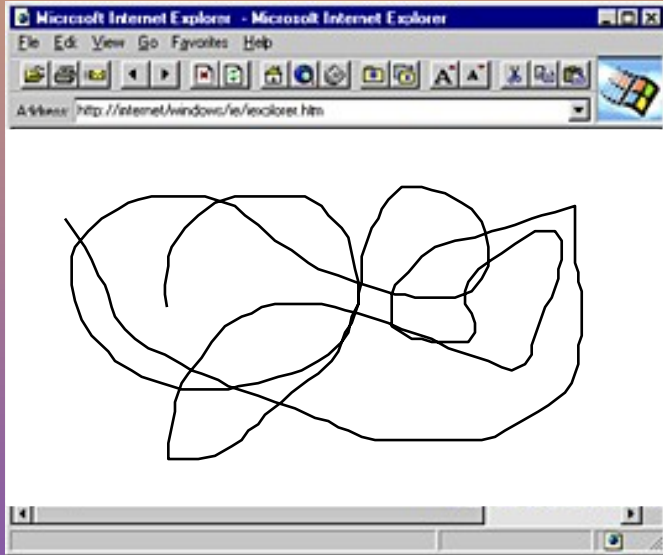- For ActiveX Controls:
    - ActiveX Control "binds" to URL moniker which returns a reference to the remote object

How It Works

Client

Server

ISAPI APP

HTTP "GET"

Document type+
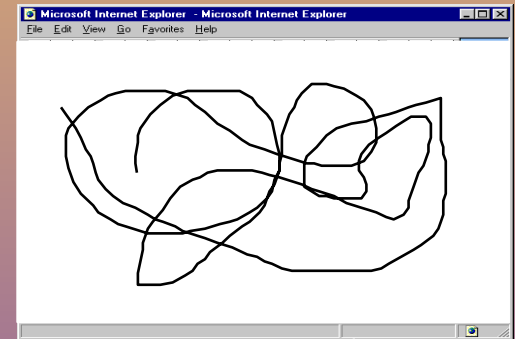
Interface pointer

Interface pointer
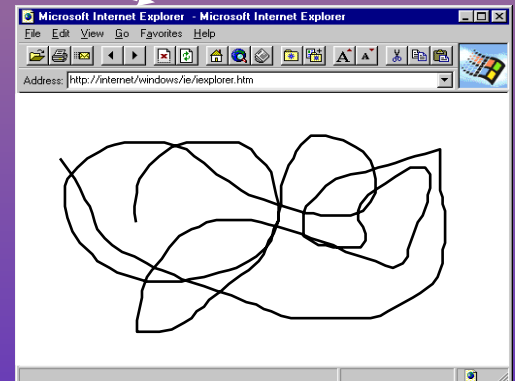
Invokes server object

DCOM

Demo - Scribble Object On The Server

Server

- In fact, the scribble control is on the server
- Exposes the same methods as the client
- When client draws, it calls the AddStroke method on the server object

DCOM

Client

And Beyond

- Now you can easily write truly high-performance, stable, custom active applications
- Groupware applications
- Multiple people working on the same document at the same time

Demo – Multiuser Scribble

- Use HTTP to browse to the document
- ISAPI application returns interface pointers
- Server keeps track of who else is participating

Server

DCOM
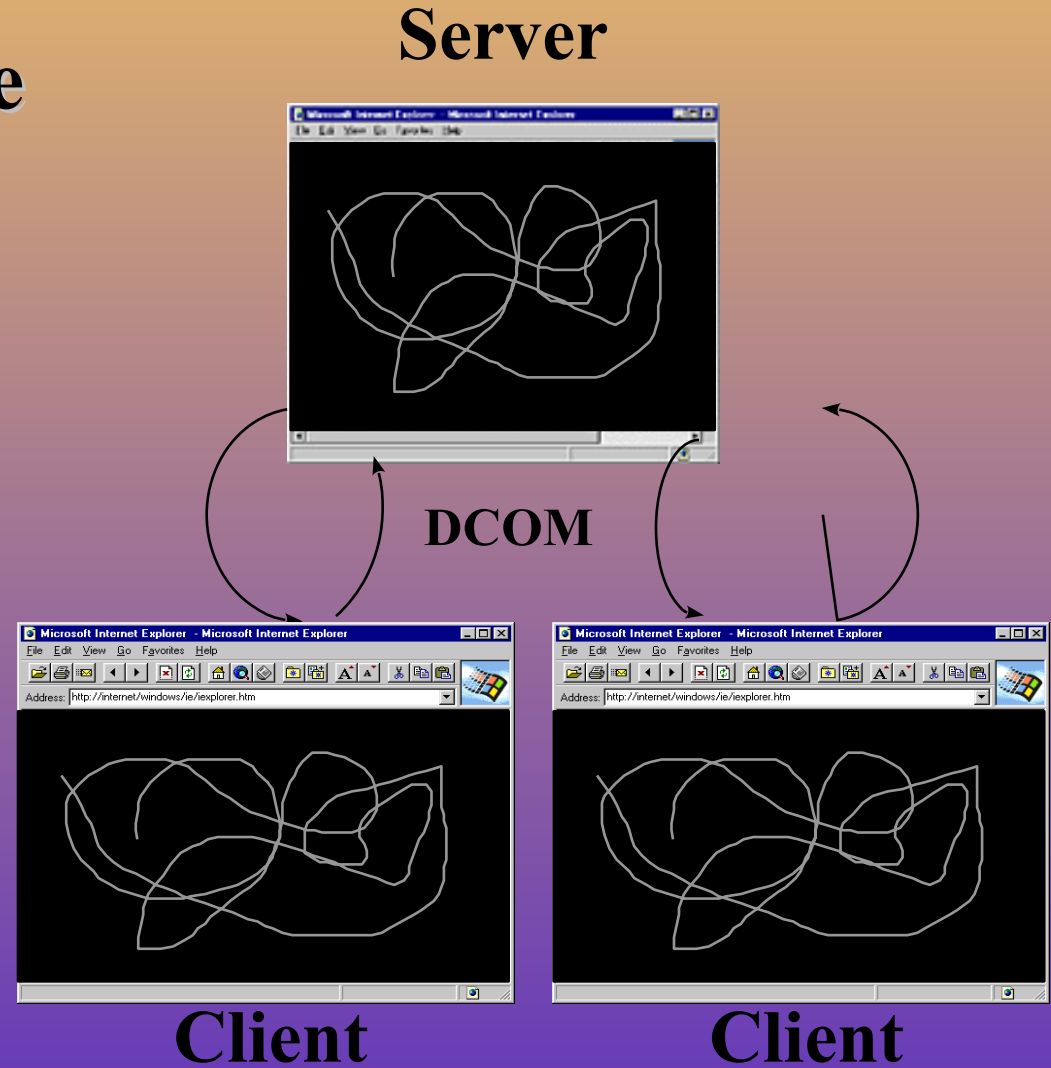
Client

Client

Making The Application Distributed

- **Design interfaces for the network**
  - **Internet: high-latency, low-bandwidth**
  - **Intranet: low-latency, high-bandwidth**
- **Choosing topology**
  - **Peer-to-peer**
  - **Client/server**
- **Server may handle keeping the clients in sync, resource locking, global operations, thread safety, etc.**
- **Client may handle local data, local operations**
- **As simple or as sophisticated as you need!**

Availability

- Distributed COM is on the PDC CD for Windows NT™ 4.0 Beta 1
- Available in Windows NT 4.0 Beta 2 - 4/96
- Beta available for Windows 95 by Q2 '96
- DCOM over HTTP Beta by Q2 '96

Take-Away

- Easy to enable your application
- By doing COM you are already there
- There is opportunity now in the document world
- There is room for even greater innovation in the application space