

Integrating the Web browser control into your databases

by Jay Rosenthal

[Editor's note: This article resides in "Notes Today", the technical Webzine located on the <http://www.notes.net> Web site produced by Iris Associates, the developers of Domino/Notes.]

Introduction

In the no-holds-barred war over browsers, Lotus Notes stands above the fray. Lotus is committed to the ongoing development of the InterNotes Web Navigator and will be adding such features as frames and nested tables in the future. But we are equally committed to giving you the fullest possible access to your browser of choice.

Notes 4.5 enables selection of Microsoft's Internet Explorer, Netscape's Navigator or another browser as your default viewer.

Microsoft's Active X technology makes it possible for us to take this support still further for Internet Explorer. Because the Explorer has been packaged as an ActiveX Web Browser Control, we can integrate this control within the Notes environment through a custom template downloadable with this article and give you that same control for use in any Notes database.

Effectively, you can now embed the full rendering capabilities of the Explorer browser (hereafter "IE") within your Notes applications, with complete visibility into its properties and methods.

ActiveX and the Web Browser Control

In many respects, ActiveX controls represent a marketing, rather than a technical direction for Microsoft. These controls repackage OLE/OCX controls for use in networked, Internet-capable environments.

While burdening applications with greater overhead and offering less native security than Java applets, they give developers immediate access to mature, reusable components developed within standard Microsoft-supported languages such as C++ and Visual Basic.

Microsoft provides both an Internet Explorer object as well as a Web browser control to developers. The former exposes the IE user interface, while the browser control exposes all internal properties and methods except the user interface.

We use the Web browser control in the sample Lotus Notes template so that you can take advantage of collaborative Notes features unavailable to ordinary IE users.

Architecturally, Microsoft's own Internet Explorer product is fundamentally nothing other than the Internet Explorer object. It wraps the user interface around the Web browser control. This means that Notes developers and users now have access at all times to the most current functional version of Internet Explorer ... running transparently within Notes and under the Notes user interface.

Lotus Notes and the Web Browser Control

Arguably, the Notes environment provides more support for developers working with the Web Browser Control than Microsoft's own tools. The Notes design interface supports loadable classes, giving the Control (and all ActiveX components) first-class status. As a corollary, properties and methods are fully exposed and available to developers for scripting.

There are a few Microsoft-driven system requirements that must be observed. Notes clients must be running under either Windows95 or NT 4.0. (NT 3.51 does not support the Web Browser Control.)

Also, note that both Microsoft environments, as shipped, provide Internet Explorer 2.X. You can download a free version of the most current Explorer product from Microsoft's site (<http://www.microsoft.com/ie/>). A 3.X version of Explorer must be installed before the template can be used.

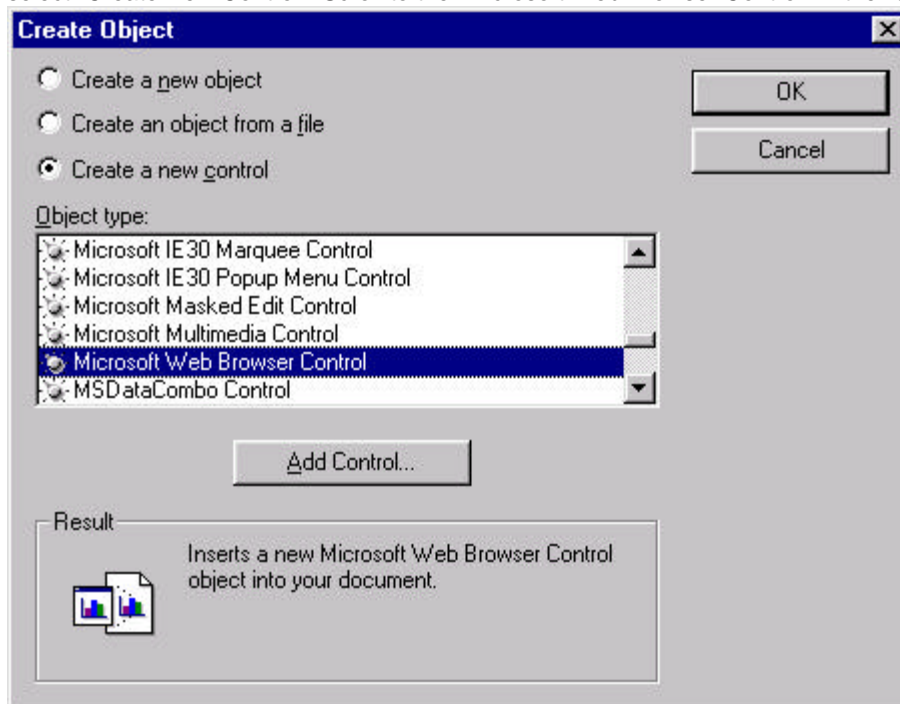
Using the Sample Template (755 Kb - available in Notes Today on <http://www.notes.net>)

1. Download the sample template and open it. Notes will load it into your local workspace with the title, "ActiveX Web Browsing". This template can be considered a launcher for the Web Browser Control.
2. When you open the template for the first time, it opens the Browser Control to the Web site currently designated as the start page within your system version of Internet Explorer.

Defining Your Own IE-Capable Databases

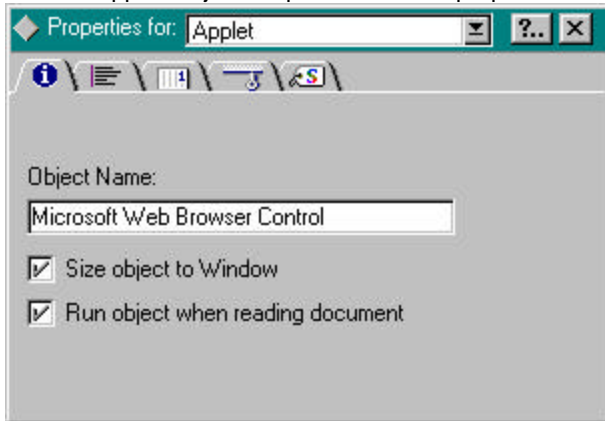
To integrate the Web Browser Control into any notes database, follow these steps:

1. Open any Notes database.
2. Choose Create-Design-Form.
3. Choose Properties-Form and define a name for the form (for instance, "IE Browser").
4. Position the cursor in the form design window and choose Create-Object. From the displayed dialog, select "Create New Control". Scroll to the "Microsoft Web Browser Control" in the list and choose OK.

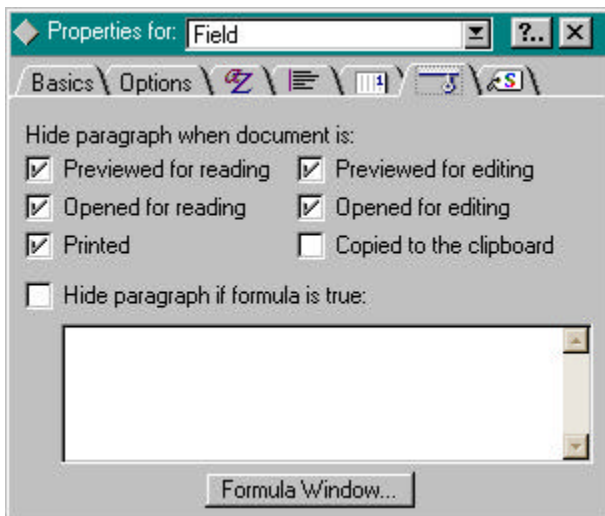
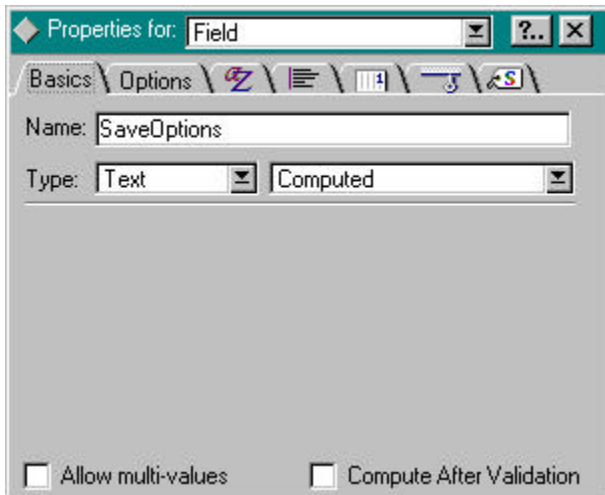


An Applet menu will be displayed dynamically on the menu bar.

5. Choose Applet-Object Properties. Set the properties as shown below:



6. Choose Create-Field. This enables you to define a "SaveOptions" field to eliminate forced user prompting upon exit from the form. Set the value of the field to 0.



7. Create the following PostOpen script event for the form:
Sub Postopen(Source As Notesuidocument)

```

Dim w As New notesuiworkspace
Dim uidoc As notesuidocument
Dim doc As notesdocument
Dim browserobject As Variant
Set uidoc=w.currentdocument
Set browserobject=uidoc.GetObject("Microsoft Web Browser Control")
browserobject.GoHome
End Sub

```

8. Save the form.

"IE Browser" will now be available as a menu item under Create. Choose Create-IE Browser. The Browser control will navigate to the currently defined Internet Explorer start page and display it within the form. Press ESC or choose File-Close to close the form.

You can conveniently define action buttons for navigation as well. To create a button that enables users to navigate to a specified URL:

1. Choose Create-Action. Enter "OpenURL" for the button title and choose an icon from the supplied list.
2. From the design pane, select the Script radio button. Enter the following script for the Click event:

```

Sub Click(Source As Button)
Dim w As New notesuiworkspace
Dim uidoc As notesuidocument
Dim doc As notesdocument
Dim browserobject As Variant
Set uidoc=w.currentdocument
Set browserobject=uidoc.GetObject("Microsoft Web Browser Control")
Dim URL As String
URL=Inputbox$("URL:", "Navigate to URL")
If URL <> "" Then
browserobject.Navigate(URL)
End Sub

```

Essentially, a user-specified string is assigned as input through a dialog. The string is passed as a parameter to the Navigate method of the browser object and is retrieved for display. Of course, in a real application, you might want to add some error-checking.

To create a button that navigates backwards through a history list of retrieved documents:

1. Choose Create-Action. Enter "Back" for the button title and choose an icon from the supplied list.
2. From the design pane, select the Script radio button. Enter the following script for the Click event:

```

Sub Click(Source As Button)
Dim w As New notesuiworkspace
Dim uidoc As notesuidocument
Dim doc As notesdocument
Dim browserobject As Variant
Set uidoc=w.currentdocument
Set browserobject=uidoc.GetObject("Microsoft Web Browser Control")
browserobject.GoBack
End Sub

```

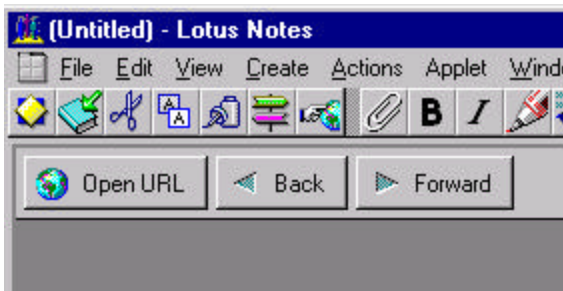
To create a button that navigates forward through a history list of retrieved documents:

1. Choose Create-Action. Enter "Forward" for the button title and choose an icon from the supplied list.

2. From the design pane, select the Script radio button. Enter the following script for the Click event:

```
Sub Click(Source As Button)
  Dim w As New notesuiworkspace
  Dim uidoc As notesuidocument
  Dim doc As notesdocument
  Dim browserobject As Variant
  Set uidoc=w.currentdocument
  Set browserobject=uidoc.GetObject("Microsoft Web Browser Control")
  browserobject.GoForward
End Sub
```

When you have completed these scripts, save the form. Again, select Create-IE Browser from the menu bar. Three action buttons, "OpenURL", "Back" and "Forward" will appear on an action bar within the form and execute under the control of the Web Browser Control.



Note that the only statement that differs between the "Back" and "Forward" scripts is "browserobject.GoBack" and "browserobject.GoForward". These specify the execution of two different methods assigned to the same object.

Where Next?

As a Notes developer, integration of the Web Browser Control gives you complete visibility into all accessible IE properties and methods. We recommend that you download and study the Control's SDK, which is readily available from the Microsoft Web site (<http://www.microsoft.com/intdev>). Other useful documents include a FAQ and short white paper on using the control.

We expect to integrate the Web Browser Control more tightly into the Notes environment in an upcoming release. However, this template gives you the tools you need today to build exciting new types of Notes applications that are fully Web-aware and that take advantage of the latest features of the Internet Explorer product. Notes applications can now execute with "IE Inside".

ABOUT THE AUTHOR

Jay Rosenthal has been a developer at Iris since December of 1995. He began his work at Iris in the InterNotes group developing the InterNotes Web Navigator and is now focusing his efforts on Component Browser integration. Before joining Iris, Jay worked at Axent Technologies where he developed security software for Windows-based systems. In his off-hours, Jay climbs mountains and enjoys other outdoor activities.

Copyright 1997 Iris Associates, Inc. All rights reserved.