

Internet Explorer integration in Notes 4.6

by Russ Lipton

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

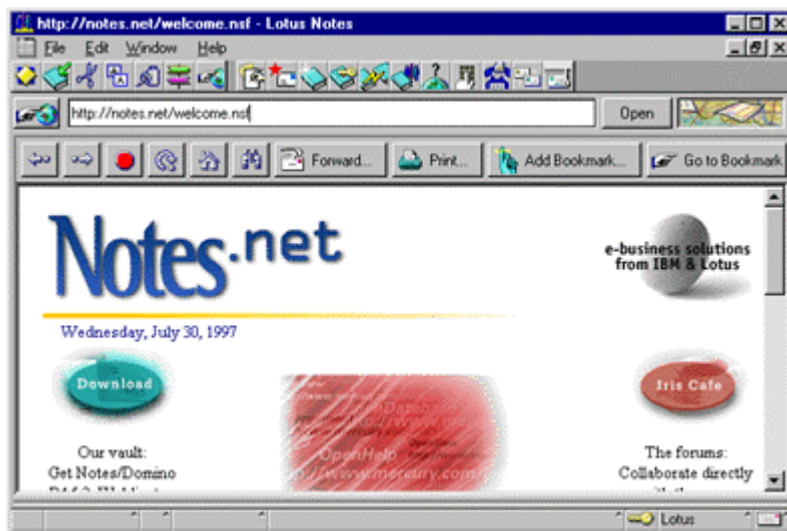
Overview

Earlier this year, Lotus made Microsoft's Internet Explorer (IE) ActiveX control available to Notes users. The IE control is essentially identical to the control used by Microsoft for their own product. Even Microsoft wraps its Internet Explorer user interface around that control.

By making the browser control available to Notes developers (and directly visible within the LotusScript IDE), we described in that article how you could embed the control within any Notes database. This is just as true today and the earlier article remains especially relevant to your custom application development.

Notes 4.6 now tightly integrates the same control into the Notes user interface itself, and includes several additional LotusScript enhancements that give you more flexibility when developing applications.

Although the pre-release 2 of Microsoft's IE 4.X control is, as of now, fairly compatible with its own IE 3.X control, Notes users may want to stick with IE 3.X. For late-breaking information on using the IE 4.X control, please see the Notes 4.6 release notes. At this time, Windows 95 or NT 4.0 platforms are required.



Taking a closer look at the integration

It is nearly impossible to predict the future of ActiveX controls in general, let alone within Notes. While emphasizing that his opinion is purely conjecture, developer Jay Rosenthal comments, "In the future, I see Web browsers going away. We probably won't need a separate Web browser to view and navigate the Web. The Web and all its content will simply become an extension of your desktop."

He adds, "Eventually, all applications will be automatically Web-enabled through component software and OS services. We see an evolution towards this, though from different directions, with both Microsoft's IE 4 and Netscape's Communicator."

Meanwhile, Notes 4.6 ties the control in more tightly with Notes by direct manipulation of the Windows system registry, clever mirroring of the IE cache and LotusScript enhancements.

Managing the system registry

The IE browser control stores its Internet settings in the system registry. Developers leveraged the similarity of the information stored in Notes location documents by updating the corresponding key in the registry whenever a location document changes within Notes.

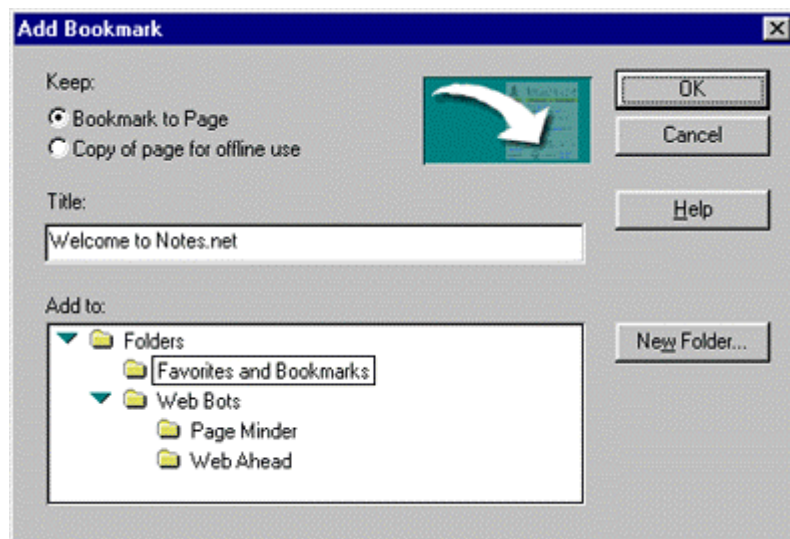
For example, suppose you have different network configurations for home and office. At home, you connect to the Internet through an Internet Service Provider (ISP); at work, you connect directly through a proxy or firewall. Once you create location documents for each environment and select the appropriate one for your physical location, you're done. Notes passes the changed values to the browser control.

Mirrored cache

It is very desirable to cache IE-rendered documents, so that users gain access to Notes features for organizing Web documents and browsing them offline. Unfortunately, IE does not provide a mechanism for replacing its cache with a Notes database.

Using published APIs, the Web Navigator runs in the background, mirroring pages into the database whenever an IE page is selected. In addition, when running agents, such as "Web Ahead," pages retrieved by the Web Navigator are also written directly into IE's cache. This implementation relies on notifications given to the Notes client by the IE browser control, both before and after an URL is retrieved.

Mirroring can be toggled on-and-off from the Internet Options form (for more information, see *Getting started with the Internet Explorer control* later in this article). When mirroring is deactivated, the "Add Bookmark" button allows you to either bookmark the page or save the entire page in your Notes database. Then, from the same dialog box, you can insert either one into an existing (or newly created) folder.



LotusScript enhancements

Notes also has some new enhancements to make it easier for you to customize applications with the browser control. These can be found and examined in the Personal Web Navigator template (perweb46.ntf) and include, among others:

- Setting the URL text displayed in the Navigation bar and the Window title
- Asynchronous invocation of the Web Retriever
- Starting or stopping the animation spinner
- Reading and writing system registry options

Setting the URL text displayed in the Navigation bar and the Window title

The Titlechange event, within the same form and Browser (Control) context, enables you to control the text that is displayed as an URL in the navigation bar and the title displayed for the window:

```
Sub Titlechange(Source As Webbrowser, Text As String)
    Dim NavBarURL As String
    ' Ask the browser object for the current URL
    NavBarURL = browserobject.LocationURL
    ' Set the Text in the Nav bar and the window title
    uidoc.NavBarSetText NavBarURL, Text
End Sub
```

The LocationURL property of the object is used to obtain the correct URL from the Browser Object Control.

Asynchronous invocation of the Web Retriever

This is available in the form of an additional argument to the getdocumentbyurl method in the NotesDatabase class. It is used extensively in the perweb46.ntf to do non-blocking retrievals of Web pages (the cache mirroring discussed above) so that they can be stored for offline use or searched with the Notes full-text index engine without making the user wait for the retrieval to complete.

This sample is from the NavigateComplete event of the Browser Control in the WebBrowserForm.

```
If docprofile.keeppage(0) = "0" Then
' Request the URL to be retrieved but don't wait for it to come in
Set nulldoc = db.getdocumentbyurl (url, True, False, Empty$, Empty$, Empty$,
Empty$,Empty$, True)
End If
```

Notice that the getdocumentbyurl call above does not return anything. It simply submits a retrieval request to Web Navigator and returns immediately. The developer cannot perform any other notesdocument methods with nulldoc such as putinfolder. If those actions must be performed, use the following statement instead:

```
Set webdoc = db.getdocumentbyurl (url, True, False, Empty$, Empty$,
Empty$, Empty$,Empty$)
```

Leaving off the last argument tells the Web Navigator to wait until the page has been retrieved and converted, and then return a valid document handle. You will see significant performance differences between the two.

Starting or stopping the animation spinner

To inspect (and if you choose, modify or apply) the animation spinner:

1. Open the *WebBrowserForm within Forms for the PerWeb template.
2. Select the Browser (Control) definition.
3. Select the "downloadbegin" event. We show the script below:

```
Sub Downloadbegin(Source As Webbrowser)
    ' Start Animation...
    uidoc.Navbarspinnerstart
End Sub
```

Navbarspinnerstart is a method defined on the Notes Uldocument class that starts the animation. A corresponding "Downloadcomplete" event and "Navbarspinnerstop" method stop the animation.

Reading and writing system registry options

The capability to read and write system registry options can be found in the Internet Functions script library. Although not new to 4.6, browsing it will help you better understand the implementation. Keep in mind that making changes to the Windows registry is a high-risk operation that can, in a worst case, make Windows unusable.

The following lines declare the Windows 32 API system calls that need to be made:

```
' This declares the routines used for retrieving key values  
' from the registry on a Windows 32 client.  
' NOTE: if you are running this app on a platform other than  
' Windows 32 Intel, then it will not work.
```

```
Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA" (Byval hKey&, Byval szSubKey$, Byval  
dwOptions&, Byval SAM&, hKey&) As Long
```

```
Declare Function RegCloseKey Lib "advapi32.dll" Alias "RegCloseKey" (Byval hKey&) As Long
```

```
Declare Function RegQueryValueEx Lib "advapi32.dll" Alias "RegQueryValueExA" (Byval hKey&, Byval szValue$, Byval  
dwReserved&, dwType&, Byval szValue$, lpdwBuf&) As Long
```

```
Declare Function RegSetValueEx Lib "advapi32.dll" Alias "RegSetValueExA" (Byval hKey&, Byval szValue$, Byval  
dwReserved&, Byval dwType&, Byval lpData$, Byval cbData&) As Long
```

Two additional functions are used to get and set the registry values `GetRegValue` and `SetRegValue`. Both functions take values of a subkey and the specific element within the subkey (such as a start page or home page within the `SOFTWARE\Microsoft\Internet Explorer\Main` subkey).

The `SetRegValue` takes an additional argument, which is the value that the new element should take.

Because the functions that you call in the Windows API are Windows 32-specific and must be declared in the declarations prior to usage, their presence will cause the Notes database to not work properly on other platforms.

Other Web Browser Control events

You can highlight the Web Browser Control within the programmer's pane in any form containing it (for instance the `*WebBrowserForm` of `perweb46.ntf`) to inspect its events. Some of these events include:

- `BeforeNavigate`
- `NavigateComplete`
- `StatusTextChanged` and `ProgressChange`

BeforeNavigate event

Code has been added here to open a new window, based on a link click event, which previews the page in the Notes preview pane.

```
Sub Beforenavigate(Source As Webbrowser, Url As String, Flags As Long, TargetframeName As String, Postdata As  
Variant, Headers As String, Cancel As Integer)
```

```
On Error 213 Resume Next
```

```

If uidoc.inpreviewpane And firstnavigatedone then
    Dim iedoc As New notesdocument(db)
    Dim newuidoc As NotesUIDocument

    Call iedoc.replaceitemvalue("Form", "WebBrowserForm")
    Call iedoc.replaceitemvalue("SaveOptions", "1")
    Call iedoc.replaceitemvalue("$ShowNavigationBar", "1")
    Call iedoc.replaceitemvalue("$NavigateToURL", Url)
    Call iedoc.save(True,False)
    Call w.viewrefresh
    Call w.editdocument(False, iedoc)
    Set newuidoc = w.currentdocument

    Set browserobject=newuidoc.GetObject("Browser")
    browserobject.Navigate Url, Flags, Targetframename, Postdata,Headers
    Cancel = True
End If

End Sub

```

You can also see where we access some of the Browser Object methods by using the NotesUIDocument method "GetObject" to obtain a handle to the object. Then, issue the Navigate method for the Browser Object to navigate to the URL in a new window.

NavigateComplete event

This is where we make the asynchronous call to retrieve the page into the Notes object store (depending on whether the user has enabled or disabled this in the Internet Options form).

StatusTextChange and ProgressChange event

These scripts print out informational progress messages to inform the user as the page is being read in, much as the user would expect to see in the standalone version of Internet Explorer.

How form formulas and scripts interact

The reason users experience such tight Notes and IE integration is due to the complex combination of form formulas and postopen scripts defined on the various forms and views within the perweb46.ntf.

A form formula is associated with a view, and allows a designer to display a different form based on an @function formula for the documents within that view. The form formulas in perweb46.ntf are driven by a new @function in 4.6 called @LocationGetInfo. LocationGetInfo accepts keywords as arguments. The keyword to obtain the current browser setting from the current location is [WebRetriever]. Here is the form formula:

```

@if(@LocationGetInfo([WebRetriever])=3;@If(form="Admin";"Admin";forward
="1";"HTMLForm";"WebBrowserForm");form)

```

This displays the "WebBrowserForm" for those users who select "Notes with Internet Explorer" while allowing everyone to see the Internet Options form (alias Admin). Users who choose "Notes" as their browser can still view the document in native Notes format.

The "HTMLform" is the main form for rendering Web pages with the Notes Navigator. Interactive Web forms that are generated on the Web are rendered in the Notes Navigator using the "Store Form in Document" attribute. As a result, all of their form layout data is stored with Notes; they don't have a form field and do not obey form formulas.

The following code defines the Postopen event of the "HTMLForm" to render it with the IE component browser:

```
If GetCurrentLocation = "3" Then
    source.close
    Print "DEBUG -- source should be closed"
    w.urlopen(url)
End If
```

GetCurrentLocation is a global routine found in the Internet Functions script library. It returns the numeric string equivalent of the browser setting in the current location. "3" represents Notes with Internet Explorer.

As well as getting the right form to display, the correct URL must be rendered. This is why the Postopen event on the WebBrowserform (which displays the Web Browser control) must navigate to the correct URL for the current page, as shown below:

```
If doc.hasitem("URL") And doc.URL(0) <> "" Then
    ' Print "Postopen Navigating to " + doc.URL(0)
    browserobject.navigate doc.url(0)
    If source.inpreviewpane Then
        firstnavigatedone = True
    End If
End If
```

Getting started with the Internet Explorer control

Now, we'll walk through the step for using the new Internet Explorer control.

Installing the control

Installing the control is quite straightforward:

1. Install IE 3.X. This can be downloaded from Microsoft's Web site (<http://www.microsoft.com/ie>) and used without charge.
2. If you have upgraded from Notes 4.5, replace the Personal Address Book design with the new pernames.ntf template.
3. Also, if you have upgraded from Notes 4.5, replace the design of your existing Personal Web Navigator database or enter a new Web Navigator database name in your location document under Advanced - Web Retriever Configuration. Enter your proxy or other site-specific settings here, if necessary.
4. Select "Notes with Internet Explorer" as the Internet browser selection in your location document. You can still select to run either Netscape Navigator or Internet Explorer as separate applications. Of course, you can also choose Notes as your navigator.

The Web browser control assumes the existence of a live direct Internet connection when the browser is launched from a workstation.

That is all there is to it. Whenever you load, navigate to a URL or click on an URL link, Notes will launch the IE control and navigate to that page. Visited pages can be automatically stored in your personal Web database. You can change this setting via the Action - Internet Options menu item when the personal Web database is open. Both the "Web Ahead" and "Page Minder" agents remain fully operative. Some additional help pages have been added to explain these features more fully.

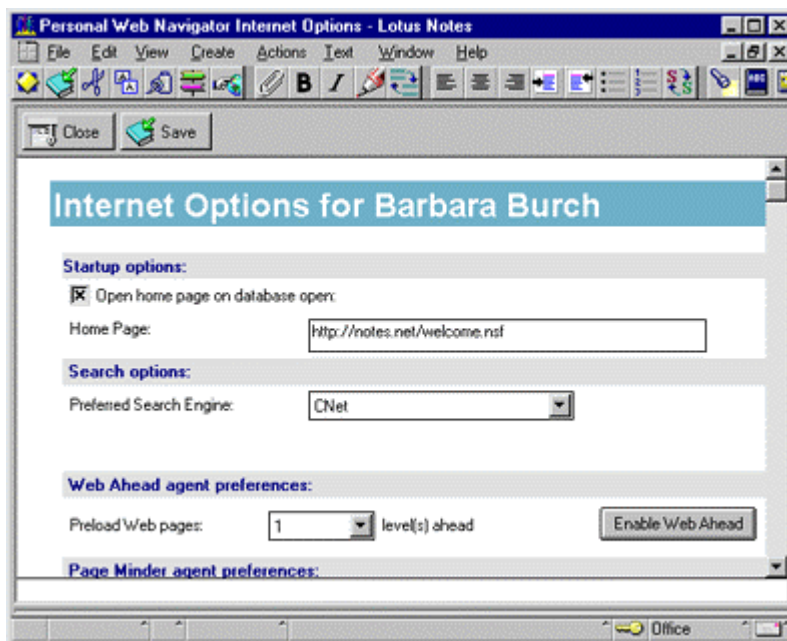
Forwarding Web pages

When you click the "Forward" button, you'll notice that there are now two ways to forward or mail a Web page to another user -- "Forward bookmark to page" or "Forward copy of page." The first method forwards only a bookmark to the Web page. This sends the addressee a Notes mail memo containing the URL as its single content field. Selecting that URL displays a properly rendered IE page.

Alternatively, you can forward the entire Web page, which is identical to how the Web Navigator forwarded Web pages in previous releases. Since it is the Notes Web Navigator version of the page that is forwarded, this method is limited by the capabilities of Web Navigator. For instance, Web pages containing frames will not forward correctly.

Changing IE configuration within Notes

Notes creates a default Internet Options document for you that can be displayed by choosing Actions - Internet Options.



Generally, any properties set within the Internet Options document only affect Web Navigator. However, there are two exceptions. The home page and search page settings within Internet Options are propagated to the proper keys in the system registry so that the IE web browser control can use them (see the next section for more details on changing IE properties).

You can also choose automatic or manual page storage for synchronizing your IE cache with your Notes cache. Selecting "Manually store pages for disconnected use" means that you must click Add Bookmark for each Web page that you want to store in your Notes database.

Changing IE settings within Notes

All browsers require a fairly common set of configuration options. Unfortunately, most browsers store that information in a non-standard fashion. IE stores its configuration information in the system registry, while Notes stores its information in the more flexible location documents of the Personal Address Book.

By allowing Notes to directly manage some IE settings, we have extended the flexibility of location documents to IE. When you make a change to a Notes location document, any settings common to both Notes and IE are inspected. If the location change requires that a change be made to an IE setting, the change is written to the system registry and IE is notified that a configuration change has occurred.

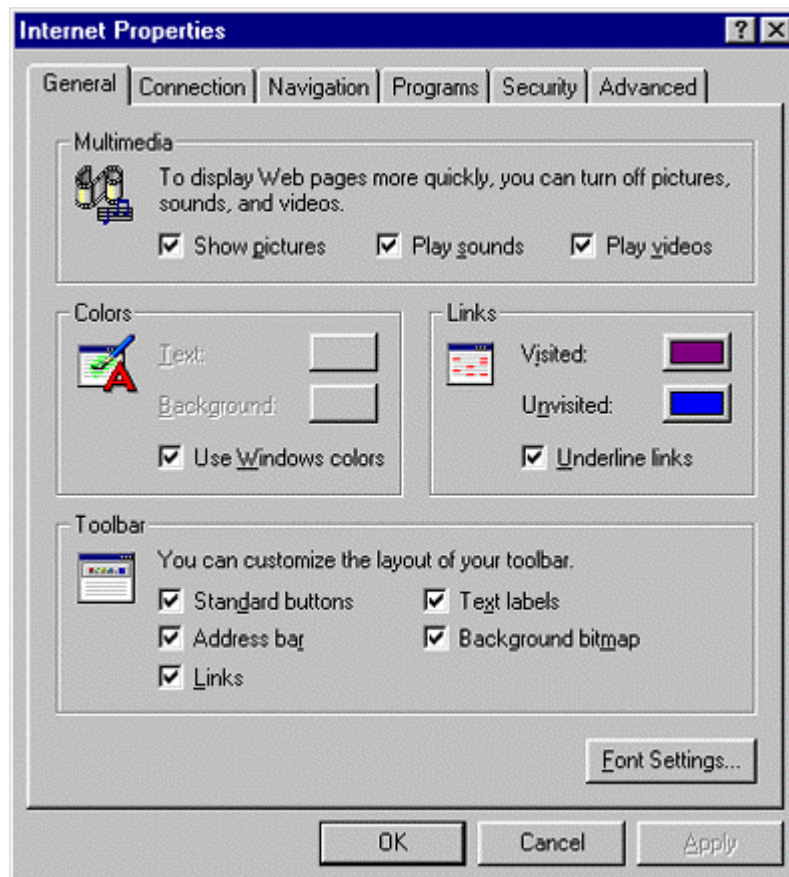
The following system registry keys may change as the result of a change to a Notes location record:

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\ProxyServer
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\ProxyEnable
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\ProxyOverride
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\RunJavaApplets
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\SyncMode

In most situations, you will not want or need to change these settings. Indeed, change them with care, since the Internet Properties menu selection controls the use of IE from Notes through a location document. If, for example, you change proxy settings directly within IE, it will not change your IE configuration, since proxy settings are uni-directionally propagated from the Notes location document to the system registry, not vice versa.

For convenience, we have also given you access to Internet Explorer's options. If you need to change any IE setting not also managed by Notes, you may do this without leaving the Notes environment:

1. Choose Actions - Internet Properties.
2. Browse normally through IE's own tabbed dialogs and make the desired change.



When you return from the IE dialog, your settings will take effect immediately.

Notes becomes your default display and mail client

With the 4.6 client, Notes can now serve both as your default Web browser and your default mail client.

This means that clicking on an HTML link within a Notes document launches the IE control within Notes. Likewise, clicking on a mailto link (of the form mailto:mailaddress) now launches Notes mail.

Neither addition requires any user interaction or intervention into Notes or IE settings. Programmatically, this is accomplished within the system registry. While the detail given here is not complete, registry keys are available to specify which application handles this request. Notes can now be specified as that application.

Considered together, this new release and the continuing work bind Internet Explorer's superior rendering engine with the integration and document management capabilities of Notes.

Copyright 1997 Iris Associates, Inc. All rights reserved.