Application Note APLX-W-0402:

Interfacing to the World Wide Web using *APLX for Windows*

Introduction

This document describes how you can use APLX in conjunction with Microsoft's Internet Explorer and other Microsoft components to integrate World Wide Web access into your *APLX for Windows* applications.

The techniques described in this document use the ActiveX/OCX and OLE Automation features of *APLX for Windows*.

Displaying Web pages in your APLX windows

The first example we will look at allows you to display web pages in APLX windows. To do this, you need a Microsoft OCX control called the 'Microsoft Web Browser' control. It is very likely that you already have this installed on your Windows system, as it is part of Internet Explorer. Let us start by creating a window in APLX, and placing an instance of the Web Browser control inside the window:

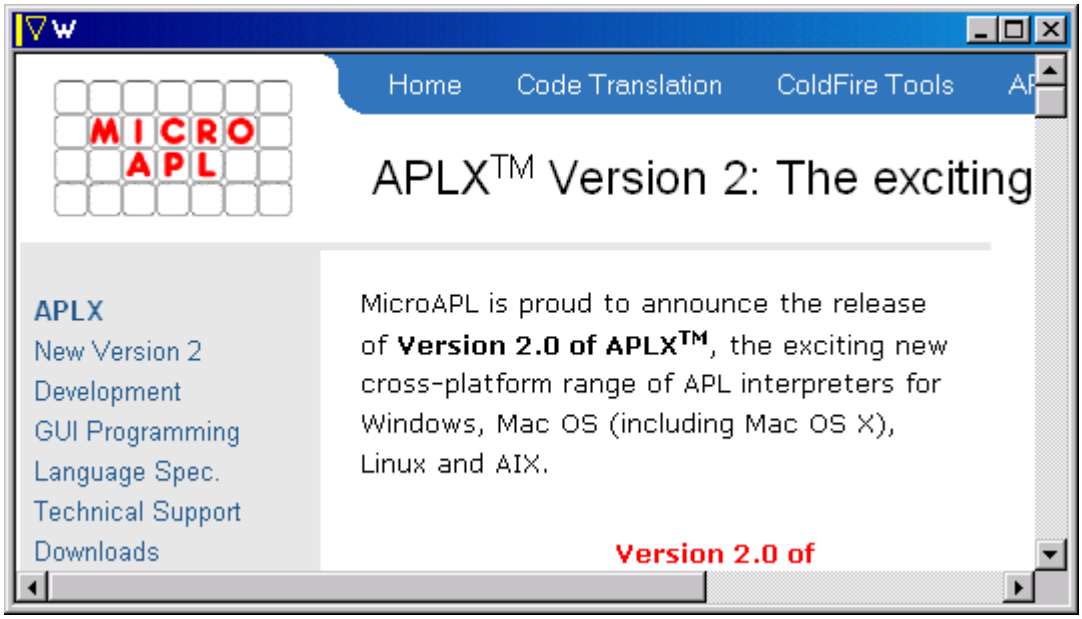
```
'W' ⎕WI 'New' 'Window'
'W.B' ⎕WI 'New' 'Microsoft Web Browser' ('align' ¯1)
```

This creates a blank window, and on it is a web browser (also initially blank). The `align` property is used to ensure that the browser fills the window if the user resizes it. If you get a `DOMAIN ERROR` on the second of these instructions, it might be because the Web Browser is not installed on your system.

Now let's display a web page inside the window. (We assume here that you are already connected to the internet). The Web Browser has a method called Navigate which will cause it to display a URL:

```
'W.B' ⎕WI 'Navigate' 'www.microapl.co.uk/apl' 0 0 0 0
```

(Don't worry about the additional arguments of four zeros). This causes the Web Browser control to access MicroAPL's web site, and display the requested page:



Note that, because we set the `align` property to `¯1`, the web browser control fills the window.

Of course, retrieving this page may take a little time, depending on the speed of your internet access and the load on MicroAPL's web server. The `busy` property tells you whether the web browser is busy loading the page:

```
      'W.B' ⎕WI 'busy'
0
```

This returns 1 if the control is busy, and 0 if it has finished loading. You can also arrange for a callback function to run when the load is complete:

```
      'W.B' ⎕WI 'onDocumentComplete' 'READY'
```

This will cause APLX to run the `READY` function when the page has successfully loaded and displayed. (As with all APLX callbacks, the actual callback is run when you execute `⎕WE`).

There are many other properties and methods of the Web Browser control (you can find the full documentation on Microsoft's web site http://www.microsoft.com). Here are a few of the most useful:

`LocationName` and `LocationURL` are read-only properties which return the page title and URL of the currently-displayed page:

```
      'W.B' ⎕WI 'LocationName'
APLX Version 2: The exciting cross-platform APL
      'W.B' ⎕WI 'LocationURL'
http://www.microapl.co.uk/apl/
```

The `silent` property is a boolean which determines whether the control can display dialog boxes. If set to 1, no dialogs will be displayed by the control.

The following methods take no arguments, and can be used to navigate through the history list or to standard locations:

`GoBack`	Navigates backward one item in the history list.
`GoForward`	Navigates forward one item in the history list.
`GoHome`	Navigates to the current home or start page.
`GoSearch`	Navigates to the current search page.

The `Refresh` method causes the current page to be reloaded.

Finally, there are several callbacks you can set in addition to `onDocumentComplete` described above. The callback `onNavigateError` is the most important of these; it fires when an error occurs during a page load.

Retrieving the HTML content of a page

The above example shows how to display web pages in your APLX application. Sometimes, however, you might want your APLX workspace to retrieve information from a web site without displaying the page. For example, you might want to retrieve the latest currency exchange rates from a web page provided by your bank.

One way of doing this is to pick up the raw HTML which the browser uses to display the page, and parse this to extract the information you want. You can do this using a Microsoft Scripting control:

```
      'HTTP' ⎕WI 'New' 'Microsoft.XMLHTTP'
```

This creates an object which encapsulates the HTTP protocol used by the World Wide Web.

```
      'HTTP' ⎕WI 'Properties'
 children class data events methods name opened progid properties self
tie xreadyState xresponseBody xresponseStream xresponseText
xresponseXML xstatus xstatusText

      'HTTP' ⎕WI 'Methods'
 Close Create Delete New Open Send Set Trigger Valueof Xabort
 XgetAllResponseHeaders XgetResponseHeader Xopen Xsend XsetRequestHeader
```

In this example, we'll extract MicroAPL's telephone number from the 'Contact Details' page of the MicroAPL website. To access the content of a page, we first need to open the URL:

```
      'HTTP' ⎕WI 'XOpen' 'GET' 'http://www.microapl.co.uk/contact.html' 0
```

This indicates that we want to open the named web page using the HTTP 'GET' protocol (Note: we have to use the full name `XOpen` here because of the name clash with the built-in APLX method `Open`).

Now we send the request:

```
      'HTTP' ⎕WI 'XSend'
```

(Again we have to use the full method name so as not to clash with the built-in `Send` method).

We can tell whether it is successful by reading back the status:

```
      'HTTP' ⎕WI 'xstatusText'
OK
```

It looks good. Let's retrieve the HTML of the page:

```
      PAGE ← 'HTTP' ⎕WI 'xresponseText'
      ⍴PAGE
4967
      60↑PAGE
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN
```

Now we've got the page contents, we can clean up by deleting the object we created:

```
      'HTTP' ⎕WI 'Delete'
```

Finally, we need to extract the information we want from the HTML text. This step depends on the page layout remaining substantially unchanged; the MicroAPL contact page has our telephone number preceded by the word 'Telephone' and followed by the HTML element `<BR>` (meaning line break). Using this knowledge of the page layout, we can extract the telephone number:

```
     (¯1+X⍳'<')↑X←(9+⎕SS PAGE 'Telephone')↓PAGE
(+44) 1825 768050
```

For more information

To learn more about the OCX/ActiveX interface of APLX for Windows, look at the chapter *OCX/ActiveX Controls and OLE Automation* in the APLX Help pages or the *APLX GUI Programming Manual*.

Full documentation on the Microsoft components described in this note can be found on the Microsoft web site or in the Visual Basic help files.