How to handle drag and drop in a TWebBrowser control (part 1 of 4)

Contents, introduction and overview

Contents

Introduction

While programming an application that uses a *TWebBrowser* component to display part of its interface I noticed that, although the application didn't use drag and drop, the *TWebBrowser* was indicating it would accept file drops. What is more, *TWebBrowser* would load and display text and HTML files, ruining my carefully designed output. Not only that but the control would also display a dialog box offering to download some types of files.

Obviously this kind of behaviour is not desirable, so I began to search for a means of inhibiting this default behaviour. My first goal was to find the circumstances under which *TWebBrowser* accepts drag-drop. Next was simply to stop *TWebBrowser* accepting file drops. Later, in another application I needed to go further and accept certain types of file drops, but wanted to control how this was handled. I then moved on to considering if I could handle text dragged from other applications and dropped on the browser control.

The remainder of this article presents my findings in the hope they will be useful to others who have been experiencing similar problems.

When *TWebBrowser* Accepts Drag-Drop

Any application that contains a *TWebBrowser* will not normally accept dragged files unless you intervene. However, if the application initialises OLE by calling *OleInitialize* then *TWebBrowser* **will** accept and try to display or download files dragged onto it!

Try it. Create a new GUI application and drop a *TWebBrowser* control on it then build and run the application. Try dragging a file from Explorer over the browser control. The "no entry" cursor will be displayed indicating that the file can't be dropped.

Now add *OnCreate* and *OnDestroy* event handlers to the main form as shown in *Listing 1*.

```
uses
  ActiveX;

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
begin
  OleInitialize(nil);
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
  OleUninitialize;
end;
```

Listing 1

Run the modified application and again drag a file from Explorer over the browser control. This time you will see a "shortcut" cursor. Drop the file and *TWebBrowser* will either display it or try to download it, depending on the file type. HTML and text files will be displayed as if you had navigated to them.

This little experiment illustrates the cirumstances when we need to take control of the web browser's handling of drag and drop. They are:

- Our application needs to initialise OLE.
- We either don't want, or want to take control of, *TWebBrowser*'s drag-drop handling.

If these circumstances apply to you, then read on. We'll start, in the *next section*, by looking at how to take charge of the browser control's drag and drop.
