

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

Demo code, going further, summary and feedback

Demo Code

A demo program to accompany this article is available for download.

The demo includes the complete source code of four demo programs:

1. A simple application containing nothing but a *TWebBrowser* control to illustrate how, unless OLE is initialised, the control does not accept drag and drop.
2. An extension of the first demo with OLE initialised to show that the *TWebBrowser* control now accepts drag and drop.
3. The complete source code of *Case Study 1* that shows how to disable drag and drop in a *TWebBrowser* control when OLE is initialised.
4. Source code for *Case Study 2* that shows how to intervene to customise *TWebBrowser*'s drag-drop handling.

The code was developed using Delphi 7 Professional. It will need some minor changes to compile with earlier versions, but should compile on later Win32 personalities of the compiler without change, although this has not been tested.

This source code is merely a proof of concept and is intended only to illustrate this article. It is not designed for use in its current form in finished applications. The code is provided on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. The source code is released under the same *Creative Commons License* as this article. If you agree to all this then please download the code using the following link.

[Download the demo code](#)

Going Further

My *GUI for the PasH Pascal Highlighter* program uses a *TWebBrowser* control and overrides drag and drop using similar methods to those described here. If you are interested in seeing how this works please feel free to download and examine the program's source code.

Summary

In this article we discovered that the default drag and drop behaviour of a *TWebBrowser* control varies depending on whether OLE has been initialised.

We examined general techniques that can be used to override the web browser control's drag drop handling by implementing the *IDropTarget*, *IDocHostUIHandler* and *IOleSlientSite* interfaces.

Next we looked at how to inhibit *TWebBrowser* from handling drag and drop at all.

Finally we moved on to learn how to customise the control's handling of drag and drop in applications that need to support drag and drop but do not want the control's default behaviour.

References

The Windows API help file supplied with Delphi provided documentation for *IDropTarget*.

Two of my earlier articles provide much of the background information relied on in this article. They are:

- ▶ "*How to customise the TWebBrowser user interface*" – provides detailed information on hosting and modifying the behaviour of *TWebBrowser* controls.
- ▶ "*How to receive data dragged from other applications*" – explains how to handle OLE drag and drop.

Feedback

I hope you enjoyed this article and found it useful. If you have any observations, comments or have found any errors please *contact me*.

This article is copyright © Peter Johnson 2007-2013



Licensed under a *Creative Commons License*.

Copyright © Peter Johnson (DelphiDabbler) 2002-2020