How to receive data dragged from other applications (part 6 of 6)

Demo program, going further, summary, references and feedback

Demo Code

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

The demo includes the complete source to the two examples presented in the *previous section*.

The code was developed using Delphi 7 Professional, but may compile with earlier versions and later Win32 personalities of the compiler, 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

While the code presented in this article works fine, it does rather clutter up the code of the main form. A more tidy, but slightly more complex, solution is to implement *IDropTarget* in a separate class and to call back to the main program to determine whether a dragged object can be dropped and how to configure the drag cursor. It is also possible to wrap up the code that interogates and reads data objects into a separate class.

My *GUI for the PasH Pascal Highlighter* program uses OLE drag-drop handling and isolates the *IDropTarget* implementation in its own class. If you are interested in this approach please feel free to download and examine the program's source code.

Summary

This article has provided an introduction to working with OLE Drag and Drop and has shown how to implement *IDropTarget* and how interogate and extract some kinds of data from a data object via its *IDataObject* interface. The article also showed how to register a window to receive OLE drag-drop notifications by associating an *IDropTarget* implementation with the window.

In addition, a demo providing source code of the two examples was also made available.

Finally some suggestions were made about to how to improve the code by isolating the *IDropTarget* code in its own class.

References

The Windows API help file provided with Delphi proved invaluable in writing this article, as did an in depth article published, I believe, on UNDO. Unfortunately I haven't been able to find a recent working link to this article.

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*.