

# How to receive data dragged from other applications (part 1 of 6)

---

## *Contents and introduction*

---

## Contents

---

- *Introduction*
- *About IDropTarget*
- *About IDataObject*
- *Understanding Data Formats*
- *Querying the Data Object*
  - *Checking for a specified data format*
  - *Enumerating all data formats*
- *Getting Data from the Data Object*
  - *Example 1: Text stored in global memory*
  - *Example 2: File list stored in global memory*
  - *Example 3: Data stored in a stream*
- *Implementing IDropTarget*
  - *Boilerplate code*
  - *Example 1: Listing data formats*
  - *Example 2: Displaying data from selected data formats*
- *Demo code*
- *Going Further*
- *Summary*
- *References*
- *Feedback*

## Introduction

---

In *article #11* we looked at how to catch files dragged and dropped on our application from Explorer. That's fine as far as it goes, but what about catching objects dragged and dropped from other applications. For example, how do we catch a text selection dragged from a text editor or word processor, or what about an HTML selection dragged from your web browser?

In order to receive such objects we have to make our application into a "drop target" that is recognised by Windows. We do this using COM (or OLE) drag and drop. As is usual with COM, we have to create an object that implements an interface – *IDropTarget* in this case – and then tell Windows about it. Windows then calls the methods of our implementation of *IDropTarget* to notify us of drag drop events.

If we accept a dropped object Windows makes the dropped object's data available via an object that supports the *IDataObject* interface. This object can provide the data in one or more data formats. Several different delivery mechanisms may be supported and objects can be targetted at different output devices. Source

applications may also advise how the data should be rendered. Our application needs to examine the available data formats and decide if it can accept any of them. If so we must also be able to extract the data from the data object.

In this article we will:

- ▶ Show how to register an application's window as a drop target.
- ▶ Interact with Windows to give visual feedback about whether we will accept a drop.
- ▶ Show how to examine all data formats supported by an object.
- ▶ Give examples of how to extract data from certain kinds of data object.

The first thing we need to do is to get to know the *IDropTarget* and *IDataObject* interfaces. We do this in the *next section*.

---

This article is copyright © Peter Johnson 2006



Licensed under a *Creative Commons License*.

---

*Copyright © Peter Johnson (DelphiDabbler) 2002-2020*