How to receive data dragged from other applications (part 3 of 6)

Unserstanding data formats

Understanding Data Formats

In this section we will investigate various data formats used to transfer data from *IDataObject* instances. Data formats can vary in four different ways:

1. **The data format**
   Just like the clipboard a data object can store data in numerous different formats. In fact, data objects use the same way of describing data formats as that used by the clipboard. A format may be one of the built-in ones such as *CF_TEXT* or an application defined format such as "Rich Text Format".
2. **The storage medium used to transfer the data**
   The data can be delivered via various different mechanisms. These mechanisms are described by the *TYMED_** enumeration and are:
   TYMED_HGLOBAL
   The data is transferred via global memory accessed via a global memory handle of type *HGLOBAL*.
   TYMED_FILE
   The data is transferred via a disk file.
   TYMED_ISTREAM
   The data is transferred as a stream accessed through an *IStream* interface.
   TYMED_ISTORAGE
   The data is transferred in a Windows storage object accessed through an *IStorage* interface.
   TYMED_GDI
   The data is specified by a GDI component accessed via a *HBITMAP* handle.
   TYMED_MFPICT
   The data is a metafile accessed via a *HMETAFILEPICT* handle.
   TYMED_ENHMF
   The data is an enhanced metafile accessed via a *HENHMETAFILE* handle.
   The way we access the data depends on the storage medium. By far the most common storage medium is the global memory handle and we will concentrate mainly on that mechanism in this article.
3. **The "Aspect", or view of the data**
   This indicates how much detail should be provided when rendering the data. The possible aspects are: a full representation as an embedded object, a thumbnail view, an icon view or a print preview. The first of these is the norm and the only one considered in this article.
4. **The target device**
   The type of device the data is targetted at can also be specified. However, the usual case is that the data is device independent and this is the only case considered by this article.

Windows encapsulates all this information inside a *TFormatEtc* structure, defined in the `ActiveX` unit. *Listing 3* reproduces the definitions.

```
type
  tagFORMATETC = record
    cfFormat: TClipFormat;
    ptd: PDVTargetDevice;
    dwAspect: Longint;
    lindex: Longint;
    tymed: Longint;
  end;
  TFormatEtc = tagFORMATETC;
```

Listing 3

The fields are discussed briefly in *Table 3* below. See the Windows API help file for more detailed information.

TFormatEtc methods	
cfFormat	Tells us about the data format and is a clipboard format identifier.
ptd	Informs about the target device and is a structure of type *DVTARGETDEVICE*. The value of this field is **nil** if the data is device independent. We will always set this value to **nil** in this article.
dwAspect	The view aspect. In all the cases we will consider, this value will be *DVASPECT_CONTENT* meaning we should represent the data as an embedded object. Other possible values are *DVASPECT_THUMBNAIL*, *DVASPECT_ICON* and *DVASPECT_DOCPRINT*.
lindex	Tells us when the view aspect must be split across page boundaries. Often, and in all cases considered here, *lindex* is -1 indicating that all the data should be displayed.
tymed	Describes the media type of the storage medium. This is one of the *TYMED_** enumeration values discussed above.

Table 3

We will use *TFormatEtc* structures in one of two ways:

1. We will examine structures filled in by Windows to learn about a data object's format.
2. We will set up our own structures to request objects in a required format.

That concludes the discussion of data objects. Armed with this knowledge we can move on to the *next section* where we discuss how to interogate, and extract data from, data objects.