

The purpose and use of DispInterface

Some COM objects (not the Delphi ones) don't support VTable binding or in other words a normal interface. They only allow access via a Dispatch interface (as defined in *IDispatch*). This requires the code to do a name lookup to find the index (*DispID*) for a function and then call it passing the parameters as an array. Some languages like VB Script require this interface. If you assign this type of interface to a variant you can call the methods on the object. For example:

```
var
  v:variant;
begin
  v := CreateOleObject('Word.Application');
  v.Open('mydoc.doc');
end;
```

Internally Delphi looks up the index of the Open command and sends all the parameters. This is done at run time so `v.MakeMoney` will compile but will give an error at run time (as will any incorrect parameters).

The type library importer also creates a dispinterface for this type of object. This allows Delphi to check names and parameter types at compile time (and also gives you tool tips). It also looks up the DispID at compile time so it is a bit faster.

A normal interface (derived from *IUnknown*) gives direct access (no lookups, no parameter parsing) so it is generally the best option if it is available, which is pretty much every where these days.

An object which supports both *IUnknown* derived interfaces and the *IDispatch* mechanism is called a *Dual interface*.

Original resource:	The Delphi Pool
Author:	Marc Rohloff
Added:	2013-03-13
Last updated:	2013-03-13

Copyright © Peter Johnson (*DelphiDabbler*) 2002-2018