How to call Delphi code from scripts running in a TWebBrowser (part 3 of 6)

Registering the external object with TWebBrowser

Registering the external object with TWebBrowser

Having implemented the COM object that extends the *external* object how do we tell the web browser about it?

The answer is by creating a container object that hosts the web browser control and implements the *IDocHostUIHandler* interface. Any web browser control container object must also implement *IOleClientSite*. *IDocHostUIHandler* has a *GetExternal* method. In our implementation of this method we will pass a reference to our custom *external* object to the browser.

Further reading

For detailed information about creating a container for *TWebBrowser* and implementing *IOleClientSite* and *IDocHostUIHandler* please see "*How to customise the TWebBrowser user interface*".

There are numerous other methods of *IDocHostUIHandler* that we have to implement, but we can get away with stubbing them out. In a previous article (see the "Further reading" box) I discussed *IDocHostUIHandler* in detail and presented a do nothing implementation – *TNulWBContainer*. I won't repeat that presentation here, so please check out the earlier article if you need to review how this is done.

Doing all the hard work in *TNulWBContainer* means that our implementation of *IDocHostUIHandler* and *IOleClientSite*, which we will call *TExternalContainer*, can be quite simple if we descend it from *TNulWBContainer*. *Listing 4* has the declaration of the class, while *Listing 5* shows its implementation.

```
type
  TExternalContainer = class(TNulWBContainer,
    IDocHostUIHandler, IOleClientSite)
  private
    fExternalObj: IDispatch;  // external object implementation
  protected
    { Re-implemented IDocHostUIHandler method }
    function GetExternal(out ppDispatch: IDispatch): HResult; stdcall;
  public
    constructor Create(const HostedBrowser: TWebBrowser);
  end;
```

Listing 4

```
constructor TExternalContainer.Create(
  const HostedBrowser: TWebBrowser);
begin
  inherited Create(HostedBrowser);
  fExternalObj := TMyExternal.Create;
end;

function TExternalContainer.GetExternal(
  out ppDispatch: IDispatch): HResult;
begin
  ppDispatch := fExternalObj;
  Result := S_OK; // indicates we've provided script
end;
```

Listing 5

Notice that we create an instance of our *external* object extension in the constructor and store it in a field of type *IDispatch*. We then implement *GetExternal* to pass back a reference to the *external* object in *ppDispatch* and return *S_OK* to indicate we have provided the object.

We pass a reference to the web browser control we are hosting to the constructor. This reference is simply passed on to the inherited constructor where it is recorded and notified that our object is its container. See the implementation of *TNulWBContainer.Create* for details of how this is done.

We have now completed the code necessary to register the *external* object with the web browser. In the *next section* we look at how to call into the *external* object from JavaScript.
