Finding HTML elements by ID in a TWebBrowser document

It's often useful to be able to find HTML elements using the elements' ID attribute, because it is unique in a HTML document.

JavaScript provides a built-in method to do that – *document.getElementById*. Unfortunately there isn't a similar method available from `MSHTML` for use with HTML documents loaded into a *TWebBrowser*. So we need to provide our own. Here's the code:

```
uses
  MSHTML, SysUtils, Variants;

function GetElementById(const Doc: IDispatch; const Id: string): IDispatch;
var
  Document: IHTMLDocument2;     // IHTMLDocument2 interface of Doc
  Body: IHTMLElement2;          // document body element
  Tags: IHTMLElementCollection; // all tags in document body
  Tag: IHTMLElement;            // a tag in document body
  I: Integer;                   // loops thru tags in document body
begin
  Result := nil;
  // Check for valid document: require IHTMLDocument2 interface to it
  if not Supports(Doc, IHTMLDocument2, Document) then
    raise Exception.Create('Invalid HTML document');
  // Check for valid body element: require IHTMLElement2 interface to it
  if not Supports(Document.body, IHTMLElement2, Body) then
    raise Exception.Create('Can''t find <body> element');
  // Get all tags in body element ('*' => any tag name)
  Tags := Body.getElementsByTagName('*');
  // Scan through all tags in body
  for I := 0 to Pred(Tags.length) do
  begin
    // Get reference to a tag
    Tag := Tags.item(I, EmptyParam) as IHTMLElement;
    // Check tag's id and return it if id matches
    if AnsiSameText(Tag.id, Id) then
    begin
      Result := Tag;
      Break;
    end;
  end;
end;
```

The routine takes a reference to the *IDispatch* interface of the web browser control's *Document* object. It returns either the *IDisptach* interface of the tag with the required Id or **nil** if no tag with the required id exists. The return value can be cast to the required interface of the tag, for e.g. *IHTMLElement* or *IHTMLElement2*. The comments in the code should explain what's happening.

Example

Let's try this out. First create a small HTML document containing at least one element that has an *id* attribute. Here's an example that we'll name `Test.html` and store in the same directory as the test application:

```
<?xml version="1.0"?>
<!DOCTYPE html
        PUBLIC "//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>Tip#36 Test</title>
  </head>
  <body>
      <p>Paragraph with no id</p>
    <p id="myid">Paragraph with id = myid</p>
  </body>
</html>
```

Now create a new Delphi application and drop a *TWebBrowser* and a *TButton* on the form. Add the following *OnShow* event handler to the form to load the test document into the browser:

```
procedure TForm1.FormShow(Sender: TObject);
begin
  WebBrowser1.Navigate(
    'file:///' + ExtractFilePath(ParamStr(0)) + 'test.html'
  );
end;
```

Next, add an *OnClick* event handler for the button:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  Elem: IHTMLElement;
begin
  Elem := GetElementById(WebBrowser1.Document, 'myid') as IHTMLElement;
  if Assigned(Elem) then
    ShowMessage(
      'Tag name = <' + Elem.tagName + '>'#10 +
      'Tag id = ' + Elem.id + #10 +
      'Tag innerHTML = "' + Elem.innerHTML + '"'
    );
end;
```

This exercises our *GetElementById* routine by finding the paragraph element with *id* = "myid" and displays some information about the element in a message box.

One use for this code is to change the HTML displayed by a block level tag. To illustrate this drop another button on the form and give it the following *OnClick* event handler:

```
procedure TForm1.Button2Click(Sender: TObject);
var
  Elem: IHTMLElement;
begin
  Elem := GetElementById(WebBrowser1.Document, 'myid') as IHTMLElement;
  if Assigned(Elem) then
    Elem.innerHTML := 'My new text';
end;
```

Run the application the click *Button1* and note the inner HTML displayed in the message box. Now click *Button2* and watch the text of the second paragraph change. If you click *Button1* you'll see the inner HTML has changed.

There's lots more you can do with *GetElementById* to make your documents more dynamic. For example you can change the image displayed by an image element. Experiment!

Author:	Peter Johnson
Contributor:	Peter Johnson
Added:	2007-10-23
Last updated:	2007-10-29