How to prevent a TWebBrowser from displaying a document's background

Recently I got asked how to prevent an HTML document's background from being displayed in a *TWebBrowser* control. On investigation I found that there are four likely places where a HTML or XHTML document sets the background are:

1. From an external style sheet imported via the `<link>` tag or via an *@import* statement in a `<style>` tag.
2. From CSS code embedded in a `<style>` tag.
3. From a *style* attribute in the `<body>` tag.
4. From various deprecated attributes of the `<body>` tag.

To remove the background we need to scan a loaded HTML document, find the attributes and objects that define the background and reset their values. The *styleSheets* collection of the web browser's document object gives access to both the external and embedded style sheets. So that deals with cases 1 and 2 above. To deal with case 3 we need to find the document's body tag and access its *style* property. This can be found via the body tag's *IHTMLElement* interface. Finally, for case 4 we need to find the relevant attributes of the body tag. They are exposed by the *background* and *bgColor* properties of *IHTMLBodyElement*.

Starting with cases 1 and 2, the following code does what we want:

```
procedure HandleStyleSheets(const Document: IDispatch);
var
  Doc: IHTMLDocument2;                      // document object
  StyleSheets: IHTMLStyleSheetsCollection;  // document's style sheets
  SheetIdx: Integer;                        // loops thru style sheets
  OVSheetIdx: OleVariant;                   // index of a style sheet
  StyleSheet: IHTMLStyleSheet;              // reference to a style sheet
  OVStyleSheet: OleVariant;                 // variant ref to style sheet
  RuleIdx: Integer;                         // loops thru style sheet rules
  Style: IHTMLRuleStyle;                    // ref to rule's style
begin
  // Get IHTMLDocument2 interface of document
  if not Supports(Document, IHTMLDocument2, Doc) then
    Exit;
  // Loop through all style sheets
  StyleSheets := Doc.styleSheets;
  for SheetIdx := 0 to Pred(StyleSheets.length) do
  begin
    OVSheetIdx := SheetIdx; // sheet index as variant required for next call
    // Get reference to style sheet (comes as variant which we convert to
    // interface reference)
    OVStyleSheet := StyleSheets.item(OVSheetIdx);
    if VarSupports(OVStyleSheet, IHTMLStyleSheet, StyleSheet) then
    begin
      // Loop through all rules within style a sheet
      for RuleIdx := 0 to Pred(StyleSheet.rules.length) do
      begin
        // Get style from a rule and reset required attributes.
        // Note: style is IHTMLRuleStyle, not IHTMLStyle, although many
        // attributes are shared between these interfaces
        Style := StyleSheet.rules.item(RuleIdx).style;
        Style.backgroundImage := '';  // removes any background image
        Style.backgroundColor := '';  // resets background colour to default
      end;
    end;
  end;
end;
```

The comments hopefully explain but briefly, we loop through all the style sheets, and all the rules within each style sheet. We use the style property associated with each rule to access and reset the required style properties.

The next routine deals with case 3, the style attribute of the body tag. We simply grab the *style* property of the body tag's *IHTMLElement* interface.

```
procedure HandleBodyStyleAttrs(const Document: IDispatch);
var
  Doc: IHTMLDocument2;    // document object
  BodyElem: IHTMLElement; // reference to body element
  Style: IHTMLStyle;      // reference to body element's style attribute
begin
  // Get document's IHTMLDocument2 interface
  if not Supports(Document, IHTMLDocument2, Doc) then
    Exit;
  // Get body tag's IHTMLElement interface
  if not Supports(Doc.body, IHTMLElement, BodyElem) then
    Exit;
  // Get style attribute of body element and reset required attributes
  Style := BodyElem.style;
  Style.backgroundImage := '';  // removes any background image
  Style.backgroundColor := '';  // resets background colour to default
end;
```

The last routine deals with case 4, the deprecated attributes of the body tag. This time instead of the *IHTMLElement* interface of the body tag we use its *IHTMLBodyElement* interface to access the relevant attributes.

```
procedure HandleBodyAttrs(const Document: IDispatch);
var
  Doc: IHTMLDocument2;        // document object
  BodyElem: IHTMLBodyElement; // reference to body element
begin
  // Get document's IHTMLDocument2 interface
  if not Supports(Document, IHTMLDocument2, Doc) then
    Exit;
  // Get body tag's IHTMLBodyElement interface
  if not Supports(Doc.body, IHTMLBodyElement, BodyElem) then
    Exit;
  // Reset required deprecated attributes of body tag
  BodyElem.background := '';  // removes any background image
  BodyElem.bgColor := '';     // resets background colour to default
end;
```

To use these routines you should load a new document into the web browser control, wait for the document to load then call each of the routines. Any background should hopefully be removed. Here's some example code that loads a HTML document named `Test.html` into a *TWebBrowser* named *WebBrowser1*:

```
begin
  WebBrowser1.Navigate('Test.html');
  while WebBrowser1.ReadyState <> READYSTATE_COMPLETE do
    Application.ProcessMessages;
  HandleStyleSheets(WebBrowser1.Document);
  HandleBodyStyleAttrs(WebBrowser1.Document);
  HandleBodyAttrs(WebBrowser1.Document);
end;
```

Going further

In addition to the background you may also want to reset the body text colour to the browser default. To do this set the *color* properties of *IHTMLRuleStyle* and *IHTMLStyle* and the *text* property of *IHTMLBodyElement* to the empty string.

You can also use these techniques to force a required background, colours, margins etc.

Author:	Peter Johnson
Contributor:	Peter Johnson
Added:	2007-10-29
Last updated:	2010-03-16