

UltraXML ActiveX Control

Introduction

The UltraXML ActiveX Control allows UltraXML to be embedded in and controlled by any 3rd party application. There is no predefined user interface with the UltraXML ActiveX control thereby allowing a 3rd party to design their own user interface to the formatting engine. The UltraXML control can display the WYSIWYG page area of a UXD file and the user can optionally scroll the control's Window.

Additionally UltraXML may be embedded inside Internet Explorer or other Internet Browser applications that support ActiveX, allowing web pages to present UltraXML as a UXD document viewer.

Full scripting and macro support is possible with the UltraXML ActiveX control by utilizing the controls Methods. It is also possible to query the value of UltraXML variables and Evaluation Strings through one of the Control's methods. Since these methods are called by the parent application of the UltraXML control, it is possible to write scripts for UltraXML in any programming language. For example in Internet Explorer a Visual Basic, ECMA or JavaScript can be used to write scripts with loops and conditionals.

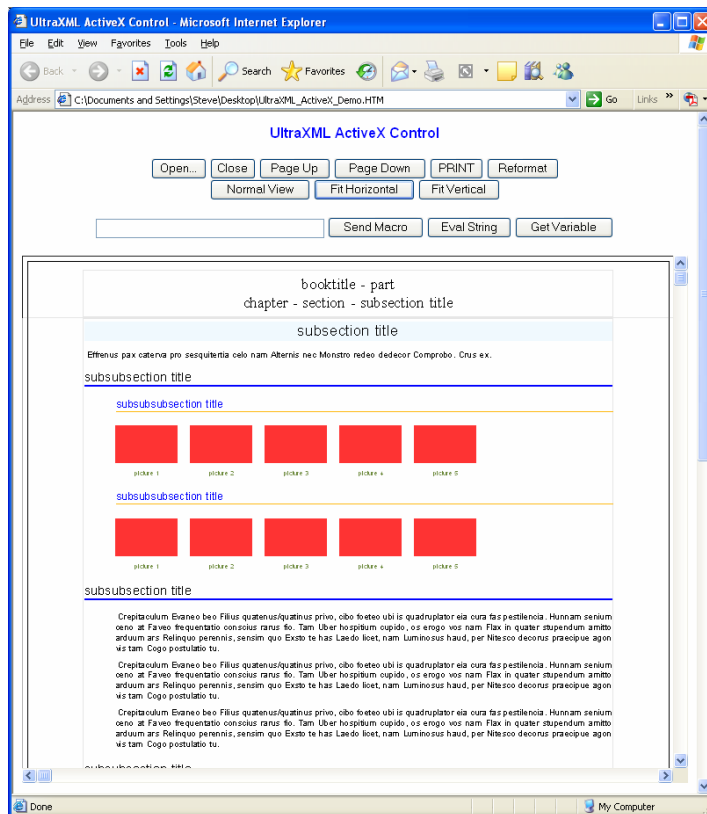
The low cost version of the UltraXML Active Control is not capable of saving a document and is only capable of printing proof output. The proof output contains Watermarks and also shifts the occasional character horizontally or vertically so that the UltraXML Control is not used for final print production. When printing, there is a 1 in 1000 chance that a character will be shifted by a larger amount and 1 in 400 chance that a character will be shifted by a smaller not so noticeable amount.

The UltraXML control can be used in conjunction with a remote UltraXML:Server or a standard Client copy of UltraXML for final print quality production.

Installation

The UltraXML ActiveX control can be installed by downloading the MSI installation package and double clicking on the .MSI ICON. Follow the on screen instructions. After installation is complete you should see a message stating that the control was successfully registered.

Note, on some machines you may get a Debug assertion error at the end of the installation. If this occurs, just press IGNORE. This will not affect the installation.

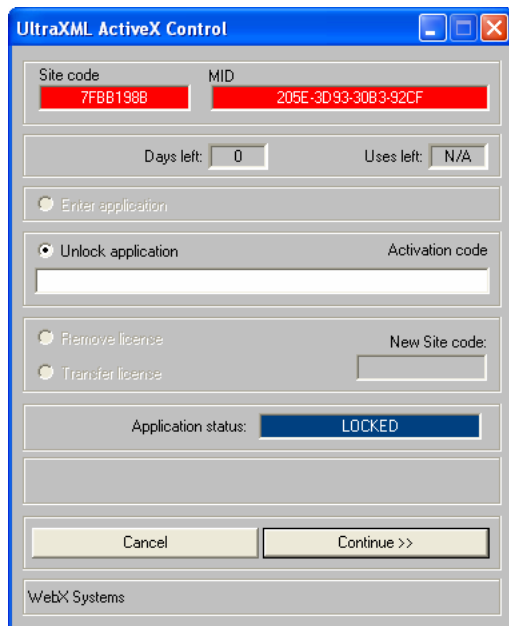


You can immediately try out the ActiveX control in Internet Explorer by opening the UltraXML_ActiveX_Demo.html file located in the UltraXML ActiveX Control \ Samples directory.

This is a simple HTML page which uses the control and has a simple user interface defined that allows you to open a UXD document and page up/down, print and fit to page etc. It will also allow you to run any UltraXML macro by entering the macro in the edit box and pressing the Macro button. Evaluation strings can also be entered into the edit box and the result found by pressing the Eval String button, eg. try \$g to retrieve the current page number. The Get Variable button will retrieve the value of an UltraXML internal variable eg. __x1 This is useful after using the wdb macro so that document properties can be retrieved.

Licensing

The UltraXML ActiveX control comes with a 10 day free license. The 10 days start from the date of installation. Once this date has expired it is necessary to purchase an activation code. When your 10 day license has expired, a dialog box will appear on starting the ActiveX control as shown below. To activate it, send the Site code and MID code to sales@webxsystems.com and you will be given an activation code to unlock the control. Note, that changing the date back or forward will immediately invalidate your 10 day free trial. Additional 10 day trial licenses can also be obtained from sales@webxsystems.com. Note that the license is locked to the machine that it was installed on. You can transfer your license to another machine by holding down the CTRL key when starting the UltraXML ActiveX control to display the licensing dialog box, then click on Remove or Transfer. You should remove your license first if you plan to install a new network card, motherboard or hard disk or re-install your operating system. You will receive a removal code which you can send to WebX systems. You can then obtain a new activation code with your new Site code and MID code.



Methods and Properties

The UltraXML ActiveX control has 3 Methods and 2 Properties defined:

Methods:

Macro(s) – Sends the macro *s* to the UltraXML control. Eg. UltraXML1.Macro("tformat")

Eval(s) – Sends the eval string *s* to the UltraXML control and returns the result of the evaluation as a string. Eg. UltraXML1.Eval("\$g")

Variable(s) – Send the name of a UltraXML variable to the UltraXML control and returns the value of the variable as a string. Eg. UltraXML1.Variable("_x1")

Properties:

Scrollbars – A boolean (true or false) property that sets whether or not the control should have scroll bars

Drag – A boolean (true or false) property that sets whether or not the control should allow the user to drag and pan the page by holding down the left or right mouse button.

The CLSID of the control is

`clsid:B3F7AD3D-B963-4AE8-9D50-15834B3D30CA`

The following shows the HTML source code that uses these Methods and Properties to define the user interface in the UltraXML_ActiveX_Demo.html file.

```
<html>

<head>
<meta http-equiv="Content-Language" content="en-gb">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>UltraXML ActiveX Control</title>
</head>
```

```
<body>

<p align="center"><b><font face="Arial" color="#0000FF">UltraXML ActiveX
Control</font></b></p>
<form>
  <p align="center"><input type="button" value="Open..." name="B1"
ONCLICK="UltraXML1.Macro('edit')">
  <input type="button" value="Close" name="B2"
ONCLICK="UltraXML1.Macro('wcl')">
  <input type="button" value="Page Up" name="B3"
ONCLICK="UltraXML1.Macro('tpgoto -1')">
  <input type="button" value="Page Down" name="B4"
ONCLICK="UltraXML1.Macro('tpgoto +1')">
  <input type="button" value="PRINT" name="B5"
ONCLICK="UltraXML1.Macro('tprint')">
  <input type="button" value="Reformat" name="B6"
ONCLICK="UltraXML1.Macro('tformat')"><br>
  <input type="button" value="Normal View" name="B7"
ONCLICK="UltraXML1.Macro('wnl')">
  <input type="button" value="Fit Horizontal" name="B8"
ONCLICK="UltraXML1.Macro('wfi 1')">
  <input type="button" value="Fit Vertical" name="B9"
ONCLICK="UltraXML1.Macro('wfi 2')"></p>
  <p align="center">
  <input type="text" name="T1" size="39">
  <input type="button" value="Send Macro" name="B10"
ONCLICK="UltraXML1.Macro(T1.value)">
  <input type="button" value="Eval String" name="B11"
ONCLICK="T1.value=UltraXML1.Eval(T1.value)">
  <input type="button" value="Get Variable" name="B12"
ONCLICK="T1.value=UltraXML1.Variable(T1.value)">
  </p>
</form>

<p align="center">
<object classid="clsid:B3F7AD3D-B963-4AE8-9D50-15834B3D30CA" width="100%"
height="600" id="UltraXML1" align="center">
<param name="Scrollbars" value=true>
<param name="Drag" value=true>
</object>
</p>

</body>

</html>
```