



XSLT - Editing XML

[< Previous](#)[Next >](#)

Data stored in XML files can be edited from an Internet browser.

Open, Edit and Save XML

Now, we will show how to open, edit, and save an XML file that is stored on the server.

We will use XSL to transform the XML document into an HTML form. The values of the XML elements will be written to HTML input fields in an HTML form. The HTML form is editable. After editing the data, the data is going to be submitted back to the server and the XML file will be updated (we will show code for both PHP and ASP).

The XML File and the XSL File

First, take a look at the XML document ("tool.xml"):

```
<?xml version="1.0" encoding="UTF-8"?>
<tool>
  <field id="prodName">
    <value>HAMMER HG2606</value>
  </field>
  <field id="prodNo">
    <value>32456240</value>
  </field>
  <field id="price">
    <value>$30.00</value>
  </field>
</tool>
```

[View the XML file.](#)

Then, take a look at the following style sheet ("tool.xsl"):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
    <body>
```

```

<form method="post" action="edittool.asp">
<h2>Tool Information (edit):</h2>
<table border="0">
  <xsl:for-each select="tool/field">
    <tr>
      <td><xsl:value-of select="@id"/></td>
      <td>
        <input type="text">
          <xsl:attribute name="id">
            <xsl:value-of select="@id" />
          </xsl:attribute>
          <xsl:attribute name="name">
            <xsl:value-of select="@id" />
          </xsl:attribute>
          <xsl:attribute name="value">
            <xsl:value-of select="value" />
          </xsl:attribute>
        </input>
      </td>
    </tr>
  </xsl:for-each>
</table>
<br />
<input type="submit" id="btn_sub" name="btn_sub" value="Submit" />
<input type="reset" id="btn_res" name="btn_res" value="Reset" />
</form>
</body>
</html>
</xsl:template>

</xsl:stylesheet>

```

View the XSL file.

The XSL file above loops through the elements in the XML file and creates one input field for each XML "field" element. The value of the XML "field" element's "id" attribute is added to both the "id" and "name" attributes of each HTML input field. The value of each XML "value" element is added to the "value" attribute of each HTML input field. The result is an editable HTML form that contains the values from the XML file.

Then, we have a second style sheet: "tool_updated.xsl". This is the XSL file that will be used to display the updated XML data. This style sheet will not result in an editable HTML form, but a static HTML table:

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
    <h2>Updated Tool Information:</h2>
    <table border="1">
      <xsl:for-each select="tool/field">
        <tr>
          <td><xsl:value-of select="@id" /></td>
          <td><xsl:value-of select="value" /></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>

```

The PHP File

In the "tool.xml" file above, change the HTML form's action attribute to "edittool.php".

The "edittool.php" page contains two functions: The loadFile() function loads and transforms the XML file for display and the updateFile() function applies the changes to the XML file:

```
<?php
function loadFile($xml, $xsl)
{
    $xmlDoc = new DOMDocument();
    $xmlDoc->load($xml);

    $xslDoc = new DOMDocument();
    $xslDoc->load($xsl);

    $proc = new XSLTProcessor();
    $proc->importStyleSheet($xslDoc);
    echo $proc->transformToXML($xmlDoc);
}

function updateFile($xml)
{
    $xmlLoad = simplexml_load_file($xml);
    $postKeys = array_keys($_POST);

    foreach($xmlLoad->children() as $x)
    {
        foreach($_POST as $key=>$value)
        {
            if($key == $x->attributes())
            {
                $x->value = $value;
            }
        }
    }

    $xmlLoad->asXML($xml);
    loadFile($xml, "tool_updated.xsl");
}

if($_POST["btn_sub"] == "")
{
    loadFile("tool.xml", "tool.xsl");
}
else
{
    updateFile("tool.xml");
}
?>
```

Tip: If you don't know how to write PHP, please study our [PHP tutorial](#).

Note: We are doing the transformation and applying the changes to the XML file on the server. This is a cross-browser solution. The client will only get HTML back from the server - which will work in any browser.

The ASP File

The HTML form in the "tool.xml" file above has an action attribute with a value of "edittool.asp".

The "edittool.asp" page contains two functions: The loadFile() function loads and transforms the XML file for display and the updateFile() function applies the changes to the XML file:

```
<%
function loadFile(xmlfile,xslfile)
Dim xmlDoc,xslDoc
'Load XML and XSL file
set xmlDoc = Server.CreateObject("Microsoft.XMLDOM")
xmlDoc.async = false
xmlDoc.load(xmlfile)
set xslDoc = Server.CreateObject("Microsoft.XMLDOM")
xslDoc.async = false
xslDoc.load(xslfile)
'Transform file
Response.Write(xmlDoc.transformNode(xslDoc))
end function

function updateFile(xmlfile)
Dim xmlDoc,rootEl,f
Dim i
'Load XML file
set xmlDoc = Server.CreateObject("Microsoft.XMLDOM")
xmlDoc.async = false
xmlDoc.load(xmlfile)

'Set the rootEl variable equal to the root element
Set rootEl = xmlDoc.documentElement

'Loop through the form collection
for i = 1 To Request.Form.Count
'Eliminate button elements in the form
if instr(1,Request.Form.Key(i),"btn_")=0 then
'The selectSingleNode method queries the XML file for a single node
'that matches a query. This query requests the value element that is
'the child of a field element that has an id attribute which matches
'the current key value in the Form Collection. When there is a match -
'set the text property equal to the value of the current field in the
'Form Collection.
set f = rootEl.selectSingleNode("field[@id='" & _
Request.Form.Key(i) & "']/value")
f.Text = Request.Form(i)
end if
next

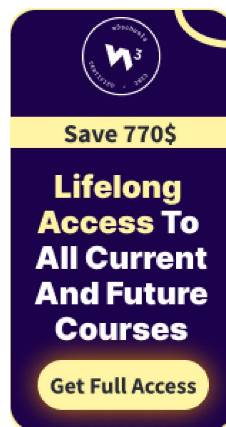
'Save the modified XML file
xmlDoc.save xmlfile

'Release all object references
set xmlDoc=nothing
set rootEl=nothing
set f=nothing

'Load the modified XML file with a style sheet that
'allows the client to see the edited information
loadFile xmlfile,server.MapPath("tool_updated.xsl")
end function

'If form is submitted, update the XML file and display result
' - if not, transform the XML file for editing
if Request.Form("btn_sub")="" then
loadFile server.MapPath("tool.xml"),server.MapPath("tool.xsl")
else
updateFile server.MapPath("tool.xml")
end if
end sub
```

```
end if
%>
```

[< Previous](#)[Log in to track progress](#)[Next >](#)

COLOR PICKER



[Spaces](#)[Upgrade](#)[Newsletter](#)[Get Certified](#)[Report Error](#)

Top Tutorials

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)
- [How To Tutorial](#)
- [SQL Tutorial](#)
- [Python Tutorial](#)
- [W3.CSS Tutorial](#)
- [Bootstrap Tutorial](#)
- [PHP Tutorial](#)
- [Java Tutorial](#)
- [C++ Tutorial](#)
- [jQuery Tutorial](#)

Top Examples

- [HTML Examples](#)
- [CSS Examples](#)
- [JavaScript Examples](#)
- [How To Examples](#)
- [SQL Examples](#)
- [Python Examples](#)
- [W3.CSS Examples](#)
- [Bootstrap Examples](#)
- [PHP Examples](#)
- [Java Examples](#)
- [XML Examples](#)
- [jQuery Examples](#)

Top References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)
- [SQL Reference](#)
- [Python Reference](#)
- [W3.CSS Reference](#)
- [Bootstrap Reference](#)
- [PHP Reference](#)
- [HTML Colors](#)
- [Java Reference](#)
- [Angular Reference](#)
- [jQuery Reference](#)

Get Certified

- [HTML Certificate](#)
- [CSS Certificate](#)
- [JavaScript Certificate](#)
- [Front End Certificate](#)
- [SQL Certificate](#)
- [Python Certificate](#)
- [PHP Certificate](#)
- [jQuery Certificate](#)
- [Java Certificate](#)
- [C++ Certificate](#)
- [C# Certificate](#)
- [XML Certificate](#)

[FORUM](#) | [ABOUT](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our [terms of use](#), [cookie and privacy policy](#).

Copyright 1999-2023 by Refsnes Data. All Rights Reserved.
W3Schools is Powered by W3.CSS.

