eMKT Browserless Coding For C# .Net and Excel

Browserless Basic Instructions and Sample Code

7/23/2013
Table of Contents
Using Excel .................................................................................................................................................................................... 3
  Configuring Excel for sending XML to eMKT ............................................................................................................................ 3
  Sandbox instructions for developing eMKT XML ...................................................................................................................... 5
  Sample Excel code for the Sandbox environment: ................................................................................................................... 6
  Sample Excel code for the Production environment: ............................................................................................................. 7
  Executing the Excel code .......................................................................................................................................................... 8
Using C# ........................................................................................................................................................................................ 8
  Configuring C# for eMKT .......................................................................................................................................................... 8
  C# eMKT Class ........................................................................................................................................................................... 9
  C# Sample code for the Sandbox that utilizes the class ......................................................................................................... 12
  C# Sample code for Production that utilizes the class ........................................................................................................... 12
Using Excel

Configuring Excel for sending XML to eMKT

#1 Open Excel

#2 Configure Excel to contain the Developer tab.
   A) Press the Excel File button

   B) Press the Excel Options link

Press Here
C) Check the “Show Developer tab in Ribbon” option

#3 Once the Developer tab is on the toolbar then select the Developer tab and then select the Visual Basic button.

#4 Next, select the Tools from the menu and then choose References.
#5 Scroll down the pop-up list and locate the “Microsoft WinHTTP Services, version 5.x” and check the box and press OK.

![Available References:](image)

**NOTE:**
If you are unable to locate the WinHTTP service then you probably have to install/register the winhttp.dll that is associated with this service. In order to install/register the winhttp.dll, follow these 2 steps. However, you will need System Admin rights of your computer to perform these steps:
- Go to the 'Start' menu, type 'Run' in the 'Start Search' box and press 'Enter.' Type 'Regsvr32 C:\Windows\System32\Winhttp.dll' in the 'Run' window and press 'Enter' to register the DLL. Click 'Yes' in the pop-up to confirm the registration of a new DLL.
- Reboot your PC.

**Sandbox instructions for developing eMKT XML**

#1 From the Developer Tab, press the Insert button and then add an ActiveX button to your spreadsheet as shown:

![Developer Tab](image)

#2 Click on the button and it should create an empty CommandButton1_Click
procedure for you.

#3 Add the contents of the procedure shown below to your procedure.

#4 Modify the code to contain the appropriate user name and password.

#5 Modify the code to contain your eMKT XML and the correct date.

**Sample Excel code for the Sandbox environment:**

**Note:** The XML at the top of the Excel routines are hardcoded queries for testing purposes. You might want to obtain the data from a spreadsheet cell or from a text file or by some other means.

```
Private Sub CommandButton1_Click()
    Dim oWinhttp As WinHttpRequest
    Dim sXML As String
    Set oWinhttp = New WinHttpRequest
    Dim result As Variant

    'Hardcoded XML to obtain Market Results. You could obtain this info 'or part of the info like the date from a spreadsheet cell or a flat 'file. The double quotes are to get the string itself to be able to 'contain a quote.
    sXML = "<?xml version="1.0"?>" & _
    "<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">" & _
    "<Header/>" & _
    "<Body>" & _
    "<QueryRequest xmlns="http://emkt.pjm.com/emkt/xml">" & _
    "<QueryMarketPrices type="Public" day="2013-07-01">" & _
    "<All/>" & _
    "</QueryMarketPrices>" & _
    "</QueryRequest>" & _
    "</Body>" & _
    "</Envelope>"

    With oWinhttp
        'Only Uncomment this next line if you need to go through a
        'Firewall
        'Call .SetProxy(2, "myproxy.mycompany.com:80","*.YOURCO.com")
        'Call .SetTimeouts(300000, 300000, 300000, 300000)
        'Sandbox
        Call .Open("POST", "https://emkttrn.pjm.com/emkt/xml/query")
        Call .SetCredentials("user", "password", 0)
        Call .SetRequestHeader("POST", "/emkt/xml/query HTTP/1.1")
        'Sandbox
        Call .SetRequestHeader("HOST", "emkttrn.pjm.com")
    End With
```

Call .SetRequestHeader("Content-Type", "text/xml")
Call .SetRequestHeader("Content-Length", Len(sXML))
Call .SetRequestHeader("SOAPAction", "/emkt/xml/query")
Call .Send(sXML)
'Wait for the response asynchronously.
Call .WaitForResponse
'These next 2 lines display the response text in a popup window.
'You could also store to a cell or to a flat file
result = .ResponseText
MsgBox result
Call .Abort
End With
End Sub

**Sample Excel code for the Production environment:**

Private Sub CommandButton1_Click()
    Dim oWinhttp As WinHttpRequest
    Dim sXML As String
    Set oWinhttp = New WinHttpRequest
    Dim result As Variant

    'Hardcoded XML to obtain Market Results. You could obtain this info
    'or part of the info like the date from a spreadsheet cell or a flat
    'file. The double quotes are to get the string itself to be able to
    'contain a quote.
    sXML = "<?xml version="1.0"?>" & _
        "<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">" & _
        "<Header/>" & _
        "<Body>" & _
        "<QueryRequest xmlns="http://emkt.pjm.com/emkt/xml">" & _
        "<QueryMarketPrices type="Public" day="2013-07-01">" & _
        "<All/>" & _
        "</QueryMarketPrices>" & _
        "</QueryRequest>" & _
        "</Body>" & _
        "</Envelope>"

    With oWinhttp
        'Only Uncomment this next line if you need to go through a
        'Firewall
        'Call .SetProxy(2, "myproxy.mycompany.com:80","*.YOURCO.com")
        'Call .SetTimeouts(300000, 300000, 300000, 300000)
        'production
        Call .Open("POST", "https://emkt.pjm.com/emkt/xml/query")
        Call .SetCredentials("user", "password", 0)
        Call .SetRequestHeader("POST", "/emkt/xml/query HTTP/1.1")
        'production
    End With
End Sub
Call .SetRequestHeader("HOST", "emkt.pjm.com")
Call .SetRequestHeader("Content-Type", "text/xml")
Call .SetRequestHeader("Content-Length", Len(sXML))
Call .SetRequestHeader("SOAPAction", "/emkt/xml/query")
Call .Send(sXML)
'Wait for the response asynchronously.
Call .WaitForResponse
'These next 2 lines display the response text in a popup window.
'You could also store to a cell or to a flat file
result = .ResponseText
MsgBox result
Call .Abort
End With
End Sub

Executing the Excel code

#1 From within the Developer tab, Press the Design Mode button to exit out of design mode.

#2 Press the button that you had added to the spreadsheet.

Your code should then execute the code associated with that button. It will connect to eMKT and it will submit the XML.

Using C#

Configuring C# for eMKT

A C# class was developed for the purposes of connecting to eMKT and being able to send and receive XML.

#1 Open Visual Studio. Note: Code was developed using Visual Studio 10 version 4.0

#2 Copy the code below into a text file and save the file under a name and ensure the file has the .cs extension.

#3 Create a project, and add the class file created above to your project.

#4 Invoke the various methods of the class to accomplish the sending and
receiving of XML to eMKT. (Described later in the document)

**C# eMKT Class**

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Runtime.InteropServices;
using System.Net;
using System.Windows.Forms;
using System.Threading;

namespace eMktBrowserless
{
    public class eMktTransfer : IDisposable
    {
        public enum XMLTypes { Query, Submit };
        private XMLTypes xmltype = XMLTypes.Query;
        private string soapaction = "/emkt/xml/query";
        private bool disposed = false;
        public void Dispose()
        {
            Dispose(true);
            GC.SuppressFinalize(this);
        }
        protected virtual void Dispose(bool disposing)
        {
            if (!disposed)
            {
                if (disposing)
                {
                    // dispose-only, i.e. non-finalizable logic
                    Connection = null;
                    GC.Collect();
                }
                // shared cleanup logic
                disposed = true;
            }
        }
        ~eMktTransfer()
        {
            Dispose(false);
        }
        public eMktTransfer(string userStr, string passwordStr)
        {
            user = userStr;
            password = passwordStr;
        }
        public XMLTypes XMLType
        {
            set
```
{ 
    xmltype = value;
    if (xmltype == XMLTypes.Query)
        soapaction = "/emkt/xml/query";
    else
        soapaction = "/emkt/xml/submit";
}

private StringBuilder xmlinputfilecontents = new StringBuilder(""); 
private StringBuilder respondedata = new StringBuilder("" ); 
private StringBuilder xmloutputfilecontents = new StringBuilder("" ); 

public string ResponseData 
{
    get { return respondedata.ToString(); } 
}

private string password = ""; 
private string user = ""; 
public string Password
{
    get { return password; } 
    set { password = value; } 
}

public string User
{
    get { return user; } 
    set { user = value; } 
}

private HttpWebRequest Connection = null; 
private string url;

public string WebURL 
{
    get { return url; } 
    set 
    {
        url = value;
        if (Connection == null)
            
            Connection = (HttpWebRequest)WebRequest.Create(url);
    }
}

public string xmlInputFileContents
{
    set
    {
        xmlinputfilecontents.Append(value);
    }
    get
    {
        return xmlinputfilecontents.ToString();
    }
}

public string xmlOutputFileContents
{

```csharp
void uploadFile(string fileSpec)
{
    using (StreamReader streamReader = new StreamReader(fileSpec))
    {
        string responseData = streamReader.ReadToEnd(); // Read the entire content of the file
        // Process the data

        // Close the file stream to release resources
        streamReader.Close();
        fileSpec.Close();
    }
}
```
try {
    if (!SetupConnectionData()) return;

    Stream postStream = Connection.GetRequestStream();
    string sXML = uploadString;
    sXML = Encoding.ASCII.GetBytes(sXML);
    postStream.Write(byteArray, 0, byteArray.Length);
    postStream.Close();

    // Assign the response object of 'WebRequest' to a 'WebResponse' variable.
    GetResponseData();
}
catch (Exception e)
{
    MessageBox.Show("General Exception" + e.Message, "General Exception",
    MessageBoxButtons.OK);
}

C# Sample code for the Sandbox that utilizes the class

using (eMktTransfer emkt = new eMktTransfer("username", "password"))
{
    emkt.XMLType = eMktTransfer.XMLTypes.Query;
    emkt.xmlInputFileContents = emkt.xmlFileReader(@"c:\Personal\sample.xml");
    // Send up the XML input file contents to eMKT
    emkt.Upload(emkt.xmlInputFileContents);
    // This is the response back from eMKT
    string testResult = emkt.ResponseData;
}

C# Sample code for Production that utilizes the class

using (eMktTransfer emkt = new eMktTransfer("username", "password"))
{
    emkt.XMLType = eMktTransfer.XMLTypes.Query;
    emkt.xmlInputFileContents = emkt.xmlFileReader(@"c:\Personal\sample.xml");
    // Send up the XML input file contents to eMKT
    emkt.Upload(emkt.xmlInputFileContents);
    // This is the response back from eMKT
    string testResult = emkt.ResponseData;
}