

```
// File: iTransast.cs
// Date created: 6/08/2009
// Original Author: Dave Peru
// Website: http://www.bphobbies.com
// Copyright (c) 2009 BP Hobbies LLC
// Version: 1.0
// Contact: dave@dperu.com
// Warning: This file contains sample code only, USE AT YOUR OWN RISK!
// Indemnify: By using this source code you agree to no indemnification.
// Note: Permission given to use this script in ANY kind of application if header
lines are left unchanged.
```

```
using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Net;
using System.Text;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.IO;
```

```
/// <summary>
/// Summary description for iTransact
/// </summary>
```

```
public class iTransact
{
```

```
    #region Class creation, destruction, resetting methods:
    #endregion
```

```
    public iTransact()
    {
    }
}
```

```
    #region Transaction Methods:
    #endregion
```

```
    static public string Sale(string strVendorId, string strVendorPassword, string
strHomePage, string strCCNum,
        string strCCMo, string strCCYr, string strCVV2Number, string strCost,
        string strBillingAddress, string strBillingZip,
        out string strStatus, out string strErrorCategory, out string
strErrorMessage, out string strAuthCode,
        out string strAVSCategory, out string strAVSResponse, out string
strCVV2Response, out string strXID)
```

```
    {
        strStatus = "";
        strErrorCategory = "";
        strErrorMessage = "";
        strAuthCode = "";
        strAVSCategory = "";
        strAVSResponse = "";
        strCVV2Response = "";
        strXID = "";
```

```
        StringBuilder sb = new StringBuilder();
        sb.Append("<?xml version='1.0'?>\r\n");
        sb.Append("\r\n");
        sb.Append("<SaleRequest>\r\n");
        sb.Append("    <CustomerData>\r\n");
```

```

sb.Append("    <Email>. </Email>\r\n");
sb.Append("    <BillingAddress>\r\n");
sb.Append("        <Address1>");
if (strBillingAddress == null || strBillingAddress.Length == 0)
    sb.Append(".");
else
    sb.Append(strBillingAddress);
sb.Append("</Address1>\r\n");
sb.Append("        <FirstName>. </FirstName>\r\n");
sb.Append("        <LastName>. </LastName>\r\n");
sb.Append("        <City>. </City>\r\n");
sb.Append("        <State>. </State>\r\n");
sb.Append("        <Zip>");
if (strBillingZip == null || strBillingZip.Length == 0)
    sb.Append(".");
else
    sb.Append(strBillingZip);
sb.Append("</Zip>\r\n");
sb.Append("        <Country>USA</Country>\r\n");
sb.Append("        <Phone>. </Phone>\r\n");
sb.Append("    </BillingAddress>\r\n");
sb.Append("    <AccountInfo>\r\n");
sb.Append("        <CardInfo>\r\n");
sb.Append("            <CCNum>");
sb.Append(strCCNum);
sb.Append("</CCNum>\r\n");
sb.Append("            <CCMo>");
sb.Append(strCCMo);
sb.Append("</CCMo>\r\n");
sb.Append("            <CCYr>");
sb.Append(strCCYr);
sb.Append("</CCYr>\r\n");
sb.Append("            <CVV2Number>");
sb.Append(strCVV2Number);
sb.Append("</CVV2Number>\r\n");
sb.Append("        </CardInfo>\r\n");
sb.Append("    </AccountInfo>\r\n");
sb.Append("    </CustomerData>\r\n");
sb.Append("    <TransactionData>\r\n");
sb.Append("        <VendorId>");
sb.Append(strVendorId);
sb.Append("</VendorId>\r\n");
sb.Append("        <VendorPassword>");
sb.Append(strVendorPassword);
sb.Append("</VendorPassword>\r\n");
sb.Append("        <HomePage>");
sb.Append(strHomePage);
sb.Append("</HomePage>\r\n");
sb.Append("        <EmailText>\r\n");
sb.Append("            <EmailTextItem>. </EmailTextItem>\r\n");
sb.Append("        </EmailText>\r\n");
sb.Append("        <OrderItems>\r\n");
sb.Append("            <Item>\r\n");
sb.Append("                <Description>Total </Description>\r\n");
sb.Append("                <Cost>");
string strAmount = String.Format("{0:C}",
decimal.Parse(strCost)).Replace("$", ""); // Handle "1.1" case.
sb.Append(strAmount);
sb.Append("</Cost>\r\n");
sb.Append("            <Qty>1</Qty>\r\n");
sb.Append("        </Item>\r\n");
sb.Append("    </OrderItems>\r\n");
sb.Append("    </TransactionData>\r\n");
sb.Append("</SaleRequest>\r\n");

```

```

        // Create a request using a URL that can receive a post.
        WebRequest request =
WebRequest.Create("https://secure.paymentclearing.com/cgi-bin/rc/xmltrans.cgi");
        // Set the Method property of the request to POST.
        request.Method = "POST";
        // Create POST data and convert it to a byte array.
        // string postData = "This is a test that posts this string to a Web
server. ";
        byte[] byteArray = Encoding.UTF8.GetBytes(sb.ToString());
        // Set the ContentType property of the WebRequest.
        request.ContentType = "Text/XML";
        // Set the ContentLength property of the WebRequest.
        request.ContentLength = byteArray.Length;
        // Get the request stream.
        Stream dataStream = request.GetRequestStream();
        // Write the data to the request stream.
        dataStream.Write(byteArray, 0, byteArray.Length);
        // Close the Stream object.
        dataStream.Close();
        // Get the response.
        WebResponse response = request.GetResponse();
        // Display the status.
        string s = ((HttpWebResponse)response).StatusDescription;
        // Get the stream containing content returned by the server.
        dataStream = response.GetResponseStream();
        // Open the stream using a StreamReader for easy access.
        StreamReader reader = new StreamReader(dataStream);
        // Read the content.
        string responseFromServer = reader.ReadToEnd();
        // Display the content.

        // Clean up the streams.
        reader.Close();
        dataStream.Close();
        response.Close();

        // Get values:
        //
        strStatus = getTagValueFromXML(responseFromServer, "<Status>", "</Status>");
        strErrorCategory = getTagValueFromXML(responseFromServer, "<ErrorCategory>",
"</ErrorCategory>");
        strErrorMessage = getTagValueFromXML(responseFromServer, "<ErrorMessage>",
"</ErrorMessage>");
        strAuthCode = getTagValueFromXML(responseFromServer, "<AuthCode>",
"</AuthCode>");
        strAVSCategory = getTagValueFromXML(responseFromServer, "<AVSCategory>",
"</AVSCategory>");
        strAVSResponse = getTagValueFromXML(responseFromServer, "<AVSResponse>",
"</AVSResponse>");
        strCVV2Response = getTagValueFromXML(responseFromServer, "<CVV2Response>",
"</CVV2Response>");
        strXID = getTagValueFromXML(responseFromServer, "<XID>", "</XID>");

        return "";
    }

    static public string Void(string strVendorId, string strVendorPassword, string
strHomePage, string strXID,
        out string strStatus, out string strErrorCategory, out string
strErrorMessage, out string strNewXID)
    {
        strStatus = "";
        strErrorCategory = "";
        strErrorMessage = "";
        strNewXID = "";
    }

```

```

StringBuilder sb = new StringBuilder();
sb.Append("<?xml version='1.0'?'>\r\n");
sb.Append("<GatewayInterface>\r\n");
sb.Append("  <VendorIdentification>\r\n");
sb.Append("    <VendorId>");
sb.Append(strVendorId);
sb.Append("</VendorId>\r\n");
sb.Append("    <VendorPassword>");
sb.Append(strVendorPassword);
sb.Append("</VendorPassword>\r\n");
sb.Append("    <HomePage>");
sb.Append(strHomePage);
sb.Append("</HomePage>\r\n");
sb.Append("  </VendorIdentification>\r\n");
sb.Append("  <VoidTransaction>\r\n");
sb.Append("    <OperationXI D>");
sb.Append(strXI D);
sb.Append("</OperationXI D>\r\n");
sb.Append("  </VoidTransaction>\r\n");
sb.Append("</GatewayInterface>\r\n");

// ' Note: "xml trans2. cgi " is used instead of "xml trans. cgi "
// '
// Create a request using a URL that can receive a post.
WebRequest request =
WebRequest.Create("https://secure.paymentclearing.com/cgi-bin/rc/xml trans2. cgi ");
// Set the Method property of the request to POST.
request.Method = "POST";
// Create POST data and convert it to a byte array.
// string postData = "This is a test that posts this string to a Web
server. ";
byte[] byteArray = Encoding.UTF8.GetBytes(sb.ToString());
// Set the ContentType property of the WebRequest.
request.ContentType = "Text/XML";
// Set the ContentLength property of the WebRequest.
request.ContentLength = byteArray.Length;
// Get the request stream.
Stream dataStream = request.GetRequestStream();
// Write the data to the request stream.
dataStream.Write(byteArray, 0, byteArray.Length);
// Close the Stream object.
dataStream.Close();
// Get the response.
WebResponse response = request.GetResponse();
// Display the status.
string s = ((HttpWebResponse)response).StatusDescription;
// Get the stream containing content returned by the server.
dataStream = response.GetResponseStream();
// Open the stream using a StreamReader for easy access.
StreamReader reader = new StreamReader(dataStream);
// Read the content.
string responseFromServer = reader.ReadToEnd();
// Display the content.

// Clean up the streams.
reader.Close();
dataStream.Close();
response.Close();

// Get values:
//
strStatus = getTagValueFromXML(responseFromServer, "<Status>", "</Status>");
strErrorCategory = getTagValueFromXML(responseFromServer, "<ErrorCategory>",
"</ErrorCategory>");

```

```

        strErrorMessage = getTagValueFromXML(responseFromServer, "<ErrorMessage>",
"</ErrorMessage>");
        strNewXID = getTagValueFromXML(responseFromServer, "<XID>", "</XID>");

        return "";
    }

    static public string Credit(string strVendorId, string strVendorPassword, string
strHomePage,
        string strAccountNumber, string strExpirationMonth, string
strExpirationYear, string strTotal,
        out string strStatus, out string strErrorCategory, out string
strErrorMessage, out string strXID)
    {
        strStatus = "";
        strErrorCategory = "";
        strErrorMessage = "";
        strXID = "";

        StringBuilder sb = new StringBuilder();
        sb.Append("<?xml version='1.0'?>\r\n");
        sb.Append("<GatewayInterface>\r\n");
        sb.Append("    <VendorIdentification>\r\n");
        sb.Append("        <VendorId>");
        sb.Append(strVendorId);
        sb.Append("</VendorId>\r\n");
        sb.Append("        <VendorPassword>");
        sb.Append(strVendorPassword);
        sb.Append("</VendorPassword>\r\n");
        sb.Append("        <HomePage>");
        sb.Append(strHomePage);
        sb.Append("</HomePage>\r\n");
        sb.Append("    </VendorIdentification>\r\n");
        sb.Append("    <CreditTransaction>\r\n");
        sb.Append("        <Total>");
        string strAmount = String.Format("{0:C}",
decim al . Parse(strTotal)). Replace("$", ""); // Handle "1.1" case.
        sb.Append(strAmount);
        sb.Append("</Total>\r\n");
        sb.Append("        <CustomerData>\r\n");
        sb.Append("            <Email>. </Email>\r\n");
        sb.Append("            <CustomerId>12345</CustomerId>\r\n");
        sb.Append("            <BillingAddress>\r\n");
        sb.Append("                <Address1>. </Address1>\r\n");
        sb.Append("                <FirstName>. </FirstName>\r\n");
        sb.Append("                <LastName>. </LastName>\r\n");
        sb.Append("                <City>. </City>\r\n");
        sb.Append("                <State>. </State>\r\n");
        sb.Append("                <Zip>. </Zip>\r\n");
        sb.Append("                <Country>USA</Country>\r\n");
        sb.Append("                <Phone>. </Phone>\r\n");
        sb.Append("            </BillingAddress>\r\n");
        sb.Append("            <ShippingAddress>\r\n");
        sb.Append("                <Address1>. </Address1>\r\n");
        sb.Append("                <FirstName>. </FirstName>\r\n");
        sb.Append("                <LastName>. </LastName>\r\n");
        sb.Append("                <City>. </City>\r\n");
        sb.Append("                <State>. </State>\r\n");
        sb.Append("                <Zip>. </Zip>\r\n");
        sb.Append("                <Country>USA</Country>\r\n");
        sb.Append("                <Phone>. </Phone>\r\n");
        sb.Append("            </ShippingAddress>\r\n");
        sb.Append("        </CustomerData>\r\n");
        sb.Append("        <AccountInfo>\r\n");
        sb.Append("            <CardAccount>\r\n");

```

```

sb.Append("<AccountNumber>");
sb.Append(strAccountNumber);
sb.Append("</AccountNumber>\r\n");
sb.Append("<Expi rati onMonth>");
sb.Append(strExpi rati onMonth);
sb.Append("</Expi rati onMonth>\r\n");
sb.Append("<Expi rati onYear>");
sb.Append(strExpi rati onYear);
sb.Append("</Expi rati onYear>\r\n");
sb.Append("</CardAccount>\r\n");
sb.Append("</AccountI nfo>\r\n");
sb.Append("</Credi tTransacti on>\r\n");
sb.Append("</GatewayI nterface>\r\n");

// ; Note: "xml trans2. cgi " is used i nstead of "xml trans. cgi "
// ;
// Create a request using a URL that can receive a post.
WebRequest request =
WebRequest.Create("https://secure.paymentcl eari ng.com/cgi -bi n/rc/xml trans2. cgi ");
// Set the Method property of the request to POST.
request.Method = "POST";
// Create POST data and convert it to a byte array.
// string postData = "This is a test that posts this string to a Web
server.";
byte[] byteArray = Encodi ng.UTF8.GetBytes(sb.ToStri ng());
// Set the ContentType property of the WebRequest.
request.ContentType = "Text/XML";
// Set the ContentLength property of the WebRequest.
request.ContentLength = byteArray.Length;
// Get the request stream.
Stream dataStream = request.GetRequestStream();
// Write the data to the request stream.
dataStream.Write(byteArray, 0, byteArray.Length);
// Cl ose the Stream obj ect.
dataStream.Cl ose();
// Get the response.
WebResponse response = request.GetResponse();
// Di spl ay the status.
string s = ((HttpWebResponse)response). StatusDescri pti on;
// Get the stream containing content returned by the server.
dataStream = response.GetResponseStream();
// Open the stream using a StreamReader for easy access.
StreamReader reader = new StreamReader(dataStream);
// Read the content.
string responseFromServer = reader.ReadToEnd();
// Di spl ay the content.

// Clean up the streams.
reader.Cl ose();
dataStream.Cl ose();
response.Cl ose();

// Get val ues:
//
strStatus = getTagVal ueFromXML(responseFromServer, "<Status>", "</Status>");
strErrorCategory = getTagVal ueFromXML(responseFromServer, "<ErrorCategory>",
"</ErrorCategory>");
strErrorMessage = getTagVal ueFromXML(responseFromServer, "<ErrorMessage>",
"</ErrorMessage>");
strXI D = getTagVal ueFromXML(responseFromServer, "<XI D>", "</XI D>");

return "";
}

```

#region Helper functions and methods:

#endregion

```
static public string getTagValueFromXML(string xml, string tagBegin, string tagEnd)
{
    int ix1 = xml.IndexOf(tagBegin);
    int ix2 = xml.IndexOf(tagEnd);
    if (ix1 == -1)
        return "";
    int ix3 = ix1 + tagBegin.Length;
    return xml.Substring(ix3, ix2 - ix3);
}
}
```