



MonsterPay
Sample Code for Synchro

Copyright and Trademark

© 2011 MonsterPay NV. All Rights Reserved. MonsterPay and the MonsterPay logo are registered trademarks of MonsterPay NV. Designated trademarks and brands are the property of their respective owners.

Notice of Liability

The information in this guide is distributed on an “as is” basis. All information provided in this document is provided with good will. The authors and publishers of this manual are not responsible for loss, or purported loss due to any contents of this publication.

Summary of Revisions

Version	Date	Changed By	Changes Made
1.0.0	01 Sep 2007	D Liu - Setcom	Original document created
1.0.1	12 Jan 2009	D Liu - Setcom	Added Revision history and table of content
2.0.0	02 Nov 2009	D Liu - Setcom	Formatted for MonsterPay
2.0.1	18 Oct 2010	D Liu - Setcom	Corrected error on page 30 and 35

Table of Content

SUMMARY OF REVISIONS	3
TABLE OF CONTENT	4
INTRODUCTION	5
PRE-REQUISITE.....	5
SYNCHRO	5
ASP.NET SAMPLE CODE FOR SYNCHRO WEB SERVICE	6
Creating the Web Reference	6
Visual Basic Sample Code.....	7
C# Sample Code.....	15
CLASSIC ASP VBSCRIPT SAMPLE CODE FOR SYNCHRO HTTP SERVICE.....	24
PHP SAMPLE CODE FOR SYNCHRO HTTP SERVICE.....	28
COLDFUSION SAMPLE CODE FOR SYNCHRO WEB SERVICE	32
ADDITIONAL INFORMATION	36

Introduction

The following are Visual Basic ASP.Net sample codes to reference and consume the MonsterPay Synchro Web and HTTP Services and load the XML Data.

Please refer to MonsterPay Implementation Guide for detailed specification.

Pre-Requisite

Knowledge of ASP.Net, Visual Basic.Net, VBScript or PHP is required.

Synchro

Synchro is used to verify and retrieve transaction details. After receiving the Auto-Redirect or Auto-Post variables, your system needs to pass these variables back to MonsterPay for verification. Once verified, MonsterPay will send the transaction details, including buyer, financial, shipping and billing information. Only partial buyer's credit card or bank account details will be passed back.

We offer the following options to verify the Auto-Redirect and Auto-Post variables:

MonsterPay Synchro Web Services - Using XML and MonsterPay's Synchro Web Service, the merchant can verify the Auto-Redirect and Auto-Post variables and retrieve the transactions details in XML format.

MonsterPay Synchro HTTP Services - Using a backend HTTP Form Post and MonsterPay's Synchro HTTP Services, the merchant can verify the Auto-Redirect and Auto-Post variables, and retrieve the transactions details.

Because both the Auto-Redirect and Auto-Post use the same variables and message request format, variable verification and data retrieval can be done using any of the above two methods. The only difference between these methods is in the way the communication between you and MonsterPay occurs.

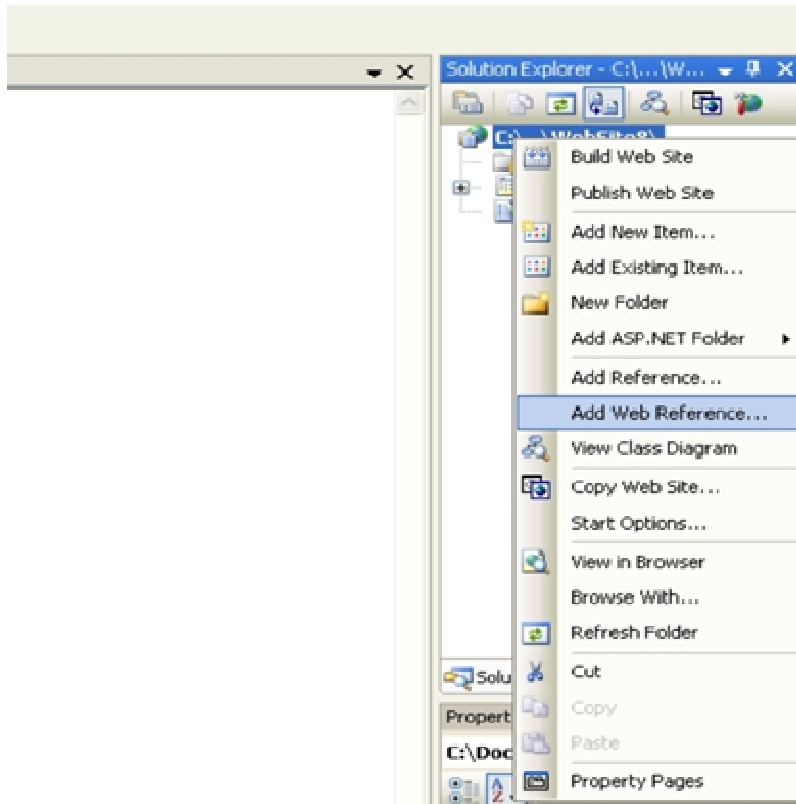
Both methods will utilise MonsterPay's industry standard 128-bit SSL security certificates for message encryption. Always ensure that all communication requests to MonsterPay are sent to a URL starting with <https://www.monsterpay.com>

ASP.Net Sample Code for Synchro Web Service

The following are Visual Basic ASP.Net sample codes to reference and consume the MonsterPay Synchro Web Services and load the XML Data.

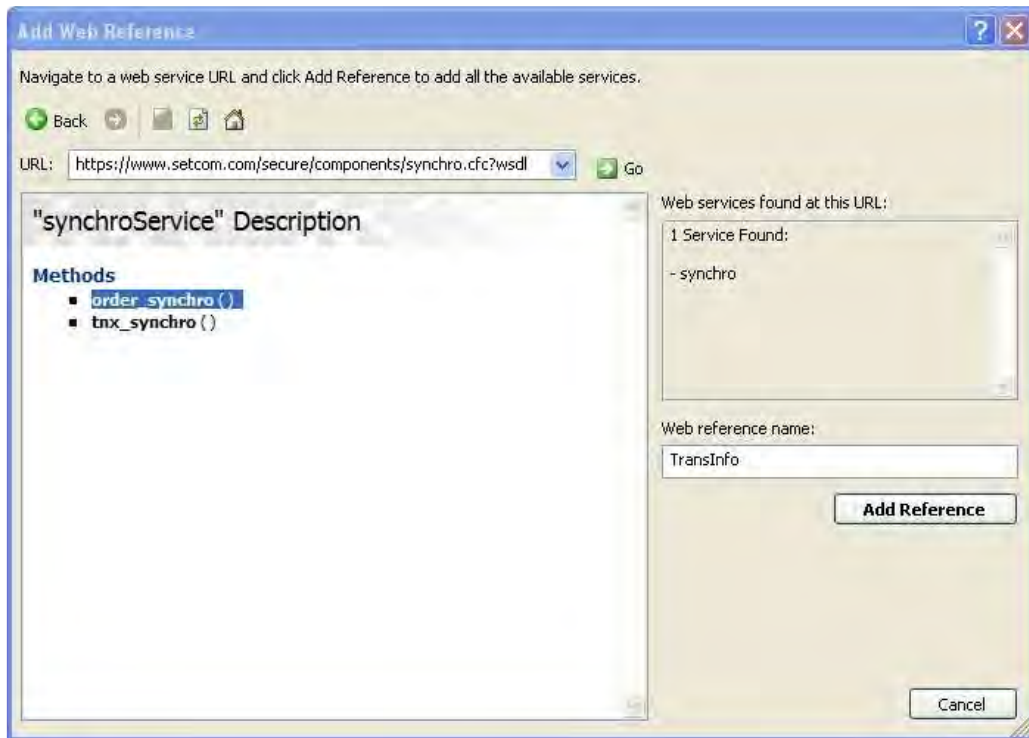
Creating the Web Reference

1. Using Visual Studio 2005, right click on your Solution and select 'Add Web Reference'



2. Enter the MonsterPay Synchro Web Services URL:
<https://www.monsterpay.com/secure/components/synchro.cfc?wsdl>.

Be sure to include the ?wsdl variable at the end of the URL. Click go, and then rename the web reference name as you see fit. Invoke the order_synchro method shown in the list on the left.



3. The web reference is now displayed in your solution explorer.

Visual Basic Sample Code

The following code can be used in the Redirect to Page or Backend Update Page on the merchant's web site to reference and consume the MonsterPay Synchro HTTP Services and load the XML Data.

When a web form is added in Visual Studio 2005 it creates two files: one with an aspx file extension and one with an aspx.vb extension.

The aspx file has the user interface while the aspx.vb file holds the code that does all the work.

The source code below can be copied and pasted into an aspx.vb file. Make a procedure call to the DoSynchro() sub in the Page_Load sub.

```
Partial Class _Default
    Inherits System.Web.UI.Page
    Structure MyItems
        Dim lidSKU As String
        Dim lidDescription As String
        Dim lidPrice As String
        Dim lidQuantity As String
        Dim lidWeight As String
        Dim lidHeight As String
        Dim lidWidth As String
    End Structure
End Class
```

Sample Code for MonsterPay Synchro

```
Dim lidLength As String
Dim lidColor As String
Dim lidSize As String
Dim lidHandling As String
Dim lidRequire_shipping As String
Dim lidOption1Name As String
Dim lidOption1Value As String
Dim lidOption2Name As String
Dim lidOption2Value As String
Dim lidIsVoucher As String
```

End Structure

```
Private Sub DoSynchro()
    Dim MerchantID, UserName, Password, newMyXml As String
    Dim errorMsg As String = ""
    ' Set the following values to your own details or that of
    ' one of the testsellers if you are still testing.
    MerchantID = "1234567892"
    UserName = "testseller2@monsterpay.com"
    Password = "testseller"

    ' Check the URL or Post data
    If Request("tnxid") = "" Then
        errorMsg = "tnxid not found<br>"
    End If
    If Request("checksum") = "" Then
        errorMsg = "checksum not found<br>"
    End If
    If Request("parity") = "" Then
        errorMsg = "parity not found<br>"
    End If
    If Not errorMsg = "" Then
        Response.Write(errorMsg)
        Response.Flush()
        Response.End()
    End If
    Dim tnxID As String = Request("tnxid")
    Dim checksum As String = Request("checkSum")
    Dim parity As String = Request("parity")

    Dim callWebService As New TransInfo.synchroService

    Dim xDoc As System.Xml.XmlDocument = New System.Xml.XmlDocument

    ' TransactionInfo(0), (1), (2) are the transaction id ,
    ' checksum, and parity respectively

    Try
        newMyXml = callWebService.order_synchro(MerchantID, UserName,
        Password, tnxID, checksum, parity)

        ' Load the returned XML string into an XML Document
        xDoc.LoadXml(newMyXml)

    Catch ex As Exception
        Response.Write(ex.Message)
    End Try
End Sub
```

Sample Code for MonsterPay Synchro

```
End Try

If Not xDoc Is Nothing Then
    GetTransactionDetails(xDoc)

Else
    Response.Write(newMyXml)
End If

End Sub

Private Sub GetTransactionDetails(ByRef xmlDoc As
System.Xml.XmlDocument)
    ' Use this sub to get the information you would like to display or
    use.
    ' Note that all numerical values have already been fixed (divided
    by 100)
    ' I have only assigned the xml output to string variables. It is
    your job to
    ' use what you want.

    ' Transaction Outcome Details
    Dim status As String =
xmlDoc.SelectSingleNode("./order_synchro/outcome").InnerText()
    Dim errorCode As String =
xmlDoc.SelectSingleNode("./order_synchro/outcome/error_code").InnerText()
    Dim errorDescription As String =
xmlDoc.SelectSingleNode("./order_synchro/outcome/error_desc").InnerText()
    Dim errorSolution As String =
xmlDoc.SelectSingleNode("./order_synchro/outcome/error_solution").InnerText
()

    Dim orderID As String =
xmlDoc.SelectSingleNode("./order_synchro/outcome/order/id").InnerText()
    Dim fundsAvailable As String =
xmlDoc.SelectSingleNode("./order_synchro/outcome/order/funds_available").In
nerText()

    Dim alertText As String =
xmlDoc.SelectSingleNode("./order_synchro/outcome/order/alerts/alert/text").
InnerText()

    ' Buyer Details
    Dim buyerUsername As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/username").InnerText()

    ' Buyer shipping details
    Dim shippingAddressTitle As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/title").Inn
erText()
    Dim shippingAddressFirstname As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/firstname")
.InnerText()
```

Sample Code for MonsterPay Synchro

```
        Dim shippingAddressLastname As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/lastname").
InnerText()
        Dim shippingAddressEmailAddress As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/email_adre
ss").InnerText()
        Dim shippingAddressStreet1 As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/street1").I
nnerText()
        Dim shippingAddressStreet2 As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/street2").I
nnerText()
        Dim shippingAddressCity As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/city").Inne
rText()
        Dim shippingAddressState As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/state").Inn
erText()
        Dim shippingAddressZip As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/zip").Inner
Text()
        Dim shippingAddressCountry As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/country").I
nnerText()
        Dim shippingAddressContactNumber As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/contact_num
ber").InnerText()

' Buyer billing details
        Dim billingAddressTitle As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/title").Inne
rText()
        Dim billingAddressFirstname As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/firstname").
InnerText()
        Dim billingAddressLastname As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/lastname").I
nnerText()
        Dim billingAddressEmailAddress As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/email_adres
s").InnerText()
        Dim billingAdresstreet1 As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/street1").In
nerText()
        Dim billingAdresstreet2 As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/street2").In
nerText()

        Dim billingAddressCity As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/city").Inner
Text()
        Dim billingAdresstate As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/state").Inne
rText()
        Dim billingAddressZip As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/zip").InnerT
ext()
        Dim billingAddressCountry As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/country").In
nerText()
```

Sample Code for MonsterPay Synchro

```
    Dim billingAddressContactNumber As String =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/contact_number").InnerText()

    ' payment instrument details
    Dim paymentInstrumentType As String =
xmlDoc.SelectSingleNode("./order_synchro/payment_instrument/type").InnerText()

    ' financial details (amounts etc.)
    Dim financialTotal As String =
(CDbl(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_total").InnerText) / 100).ToString("F2")
    Dim financialSubTotal As String =
(CDbl(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_sub_total").InnerText) / 100).ToString("F2")
    Dim financialShipping As String =
(CDbl(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_shipping").InnerText) / 100).ToString("F2")
    Dim financialTax As String =
(CDbl(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_tax").InnerText) / 100).ToString("F2")
    Dim financialInsurance As String =
(CDbl(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_insurance").InnerText) / 100).ToString("F2")
    Dim financialCurrency As String =
xmlDoc.SelectSingleNode("./order_synchro/financial/currency").InnerText()
    Dim financialPromo As String =
xmlDoc.SelectSingleNode("./order_synchro/financial/amount_promo").InnerText()

    ' additional information
    Dim additionalInfoShippingOption As String =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/shipping_option").InnerText()
    Dim additionalInfoTaxOption As String =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/tax_option").InnerText()
    Dim additionalInfoInsuranceOption As String =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/insurance_option").InnerText()
    Dim additionalInfoPromoCode As String =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/promo_code").InnerText()
    Dim additionalInfoPromoDesc As String =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/promo_short_description").InnerText()
    Dim additionalInfoComment As String =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/comment").InnerText()

    ' line item details
    Dim ItemsCount As Integer =
xmlDoc.SelectNodes("./order_synchro/line_items/lid").Count
    Dim Counter As Integer

    Dim ItemStructure(ItemsCount) As MyItems

    For Counter = 1 To ItemsCount
```

Sample Code for MonsterPay Synchro

```
        ItemStructure(Counter).lidSKU =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/sku").InnerText()
        Dim mystuf As String =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[1]/sku").InnerText(
)
        Dim mystuf2 As String =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[2]/sku").InnerText(
)
        ItemStructure(Counter).lidDescription = "hello"
        ItemStructure(Counter).lidPrice =
(CDbl(xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/price[1]").InnerText) / 100).ToString("F2")
        ItemStructure(Counter).lidQuantity =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/quantity").InnerText()
        ItemStructure(Counter).lidWeight =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/weight").InnerText()
        ItemStructure(Counter).lidHeight =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/height").InnerText()
        ItemStructure(Counter).lidWidth =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/width").InnerText()
        ItemStructure(Counter).lidLength =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/length").InnerText()
        ItemStructure(Counter).lidColor =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/color").InnerText()
        ItemStructure(Counter).lidSize =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/size").InnerText()
        ItemStructure(Counter).lidHandling =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/handling").InnerText()
        ItemStructure(Counter).lidRequire_shipping =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/require_shipping").InnerText()
        ItemStructure(Counter).lidOption1Name =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/option1name").InnerText()
        ItemStructure(Counter).lidOption1Value =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/option1value").InnerText()
        ItemStructure(Counter).lidOption2Name =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/option2name").InnerText()
        ItemStructure(Counter).lidOption2Value =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/option1value").InnerText()
        ItemStructure(Counter).lidIsVoucher =
xmlDoc.SelectSingleNode("./order_synchro/line_items/lid[" & Counter &
"/is_voucher").InnerText()
        Next Counter
```

' custom fields

'Note that these can have more than one field, in which case you would need to do a loop like with line items

Sample Code for MonsterPay Synchro

```
'This will only be if the seller entered custom fields
Dim customFields As String =
xmlDoc.SelectSingleNode("./order_synchro/custom_fields").InnerText()
If customFields <> "" Then
    Dim customName As String =
xmlDoc.SelectSingleNode("./order_synchro/custom_fields/custom_field/name").
InnerText()
    Dim customValue As String =
xmlDoc.SelectSingleNode("./order_synchro/custom_fields/custom_field/value")
.InnerText()
Else
    customFields = ""
End If

'status
Response.Write("Status : " + status + "<br>")

'transaction information
Response.Write("orderId = " & orderId & "<br>")
Response.Write("fundsAvailable = " & fundsAvailable & "<br>")

'outcome
Response.Write("errorCode = " & errorCode & "<br>")
Response.Write("errorDescription = " & errorDescription & "<br>")
Response.Write("errorSolution = " & errorSolution & "<br>")

>alerts

Response.Write("alertText = " & alertText & "<br>")

'Buyer(Details)
Response.Write("buyerUsername = " & buyerUsername & "<br>")

' Buyer shipping details
Response.Write("shippingAddressTitle = " & shippingAddressTitle &
"<br>")
    Response.Write("shippingAddressFirstname = " &
shippingAddressFirstname & "<br>")
    Response.Write("shippingAddressLastname = " &
shippingAddressLastname & "<br>")
    Response.Write("shippingAddressEmailAddress = " &
shippingAddressEmailAddress & "<br>")
    Response.Write("shippingAddressStreet1 = " & shippingAddressStreet1
& "<br>")
    Response.Write("shippingAddressStreet2 = " & shippingAddressStreet2
& "<br>")
    Response.Write("shippingAddressCity = " & shippingAddressCity &
"<br>")
    Response.Write("shippingAddressState = " & shippingAddressState &
"<br>")
    Response.Write("shippingAddressZip = " & shippingAddressZip &
"<br>")
    Response.Write("shippingAddressCountry = " & shippingAddressCountry
& "<br>")
    Response.Write("shippingAddressContactNumber = " &
shippingAddressContactNumber & "<br>")

'' Buyer billing details
Response.Write("billingAddressTitle = " & billingAddressTitle &
"<br>")
```

Sample Code for MonsterPay Synchro

```
        Response.Write("billingAddressFirstname = " &
billingAddressFirstname & "<br>")
        Response.Write("billingAddressLastname = " & billingAddressLastname
& "<br>")
        Response.Write("billingAddressEmailAddress = " &
billingAddressEmailAddress & "<br>")
        Response.Write("billingAddressstreet1 = " & billingAddressstreet1 &
"<br>")
        Response.Write("billingAddressstreet2 = " & billingAddressstreet2 &
"<br>")
        Response.Write("billingAddressCity = " & billingAddressCity &
"<br>")
        Response.Write("billingAddressstate = " & billingAddressstate &
"<br>")
        Response.Write("billingAddressZip = " & billingAddressZip & "<br>")
        Response.Write("billingAddressCountry = " & billingAddressCountry &
"<br>")
        Response.Write("billingAddressContactNumber = " &
billingAddressContactNumber & "<br>")

        ' payment instrument details
        Response.Write("paymentInstrumentType = " & paymentInstrumentType &
"<br>")

        ' financial details (amounts etc.)
        Response.Write("financialTotal = " & financialTotal & "<br>")
        Response.Write("financialSubTotal = " & financialSubTotal & "<br>")
        Response.Write("financialShipping = " & financialShipping & "<br>")
        Response.Write("financialTax = " & financialTax & "<br>")
        Response.Write("financialInsurance = " & financialInsurance &
"<br>")
        Response.Write("financialCurrency = " & financialCurrency & "<br>")
        Response.Write("financialPromo = " & financialPromo & "<br>")

        ' additional information
        Response.Write("additionalInfoShippingOption = " &
additionalInfoShippingOption & "<br>")
        Response.Write("additionalInfoTaxOption = " &
additionalInfoTaxOption & "<br>")
        Response.Write("additionalInfoInsuranceOption = " &
additionalInfoInsuranceOption & "<br>")
        Response.Write("additionalInfoPromoCode = " &
additionalInfoPromoCode & "<br>")
        Response.Write("additionalInfoPromoDesc = " &
additionalInfoTaxOption & "<br>")
        Response.Write("additionalInfoComment = " &
additionalInfoInsuranceOption & "<br>")

        'amount of line items
        Response.Write("ItemsCount = " & ItemsCount & "<br>")

        ' line item details
        Dim outPutCounter As Integer
        For outPutCounter = 1 To ItemsCount
            Response.Write("<br><br><br>" & "Item no :" & outPutCounter &
"<br>")
            Response.Write("ItemStructure.lidSKU = " &
ItemStructure(outPutCounter).lidSKU & "<br>")
```

Sample Code for MonsterPay Synchro

```
        Response.Write("lidPrice = " &
ItemStructure(outPutCounter).lidPrice & "<br>")
        Response.Write("lidQuantity = " &
ItemStructure(outPutCounter).lidQuantity & "<br>")
        Response.Write("lidWeight = " &
ItemStructure(outPutCounter).lidWeight & "<br>")
        Response.Write("lidHeight = " &
ItemStructure(outPutCounter).lidHeight & "<br>")
        Response.Write("lidWidth = " &
ItemStructure(outPutCounter).lidWidth & "<br>")
        Response.Write("lidLength = " &
ItemStructure(outPutCounter).lidLength & "<br>")
        Response.Write("lidColor = " &
ItemStructure(outPutCounter).lidColor & "<br>")
        Response.Write("lidSize = " &
ItemStructure(outPutCounter).lidSize & "<br>")
        Response.Write("lidHandling = " &
ItemStructure(outPutCounter).lidHandling & "<br>")
        Response.Write("lidRequire_shipping = " &
ItemStructure(outPutCounter).lidRequire_shipping & "<br>")
        Response.Write("lidOption1Name = " &
ItemStructure(outPutCounter).lidOption1Name & "<br>")
        Response.Write("lidOption1Value = " &
ItemStructure(outPutCounter).lidOption1Value & "<br>")
        Response.Write("lidOption2Name = " &
ItemStructure(outPutCounter).lidOption2Name & "<br>")
        Response.Write("lidOption2Value = " &
ItemStructure(outPutCounter).lidOption2Value & "<br>")
        Response.Write("lidIsVoucher = " &
ItemStructure(outPutCounter).lidIsVoucher & "<br><br><br>")

        Next outPutCounter
        ' custom fields
        Response.Write("Custom Field = " & customFields & "<br><br><br>")

        If Not Session("customName") Is Nothing Then
            Response.Write("Custom Value = " & Session("customName") &
"<br><br><br>")
        End If

    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        Call DoSynchro()
    End Sub
End Class
```

C# Sample Code

The following code can be used in the Redirect to Page or Backend Update Page on the merchant's web site to reference and consume the [MonsterPay Synchro HTTP Services](#) and load the XML Data.

When a web form is added in Visual Studio 2005 it creates two files: one with an aspx file extension and one with an aspx.cs extension.

The aspx file has the user interface while the aspx.cs file holds the code that does all the work.

The source code below can be copied and pasted into an aspx.cs file:

```
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Web.Services;
using TransInfo;
using System.Xml;
using System.Text;
using System;

public partial class _Default : System.Web.UI.Page
{
    string errorMsg = "";

    protected void Page_Load(object sender, EventArgs e)
    {
        string myXML = "";
        string txnid = Request["txnid"];
        string checksum = Request["checksum"];
        double parity = Convert.ToDouble(Request["parity"]);

        //Set the following values to your own details or that of one of the
        //testers if you are still testing.

        string Identifier = "1234567892";
        string Username = "testseller2@monsterpay.com";
        string Pwd = "testseller";

        // string url =
        "https://www.monsterpay.com/secure/components/synchro.cfc?wsdl";

        if (validate())
        {
            synchroService ws = new synchroService();

            System.Xml.XmlDocument xmlDoc = new System.Xml.XmlDocument();

            try
            {
                myXML = ws.order_synchro(Identifier, Username, Pwd, txnid,
checksum, parity);
                xmlDoc.LoadXml(myXML);
            }
            catch (Exception ex)
            {
                errorMsg = ex.ToString();
                Response.Write(errorMsg);
            }

            if (!(xmlDoc == null))
            {
                GetTransactionDetails(xmlDoc);
            }
        }
    }
}
```

Sample Code for MonsterPay Synchro

```
    }
    else
    {
        errorMsg = "xDoc is null: " + myXML;
        Response.Write(errorMsg);
    }

    // variables must be resubmitted back to MONSTERPAY either via
    // 1) MonsterPay Web Services
    // 2) MonsterPay Synchro HTTP Services
}

private void GetTransactionDetails(XmlDocument xmlDoc)
{
    // build email string....

    StringBuilder mymail = new StringBuilder();

    //order status
    String status =
xmlDoc.SelectSingleNode("./order_synchro/outcome/status").InnerText;

    // order information
    String orderID =
xmlDoc.SelectSingleNode("./order_synchro/outcome/order/id").InnerText;
    String orderFundsAvailable =
xmlDoc.SelectSingleNode("./order_synchro/outcome/order/funds_available").In
nerText;

    //outcome
    String errorCode =
xmlDoc.SelectSingleNode("./order_synchro/outcome/error_code").InnerText;
    String errorDescription =
xmlDoc.SelectSingleNode("./order_synchro/outcome/error_desc").InnerText;
    String errorSolution =
xmlDoc.SelectSingleNode("./order_synchro/outcome/error_solution").InnerText
;

    //alerts
    String alert =
xmlDoc.SelectSingleNode("./order_synchro/outcome/order/alerts/alert").Inner
Text;

    String alertText =
xmlDoc.SelectSingleNode("./order_synchro/outcome/order/alerts/alert/text").
InnerText;

    //seller details
    String sellerId =
xmlDoc.SelectSingleNode("./order_synchro/seller/identifier").InnerText;
    String sellerUserName =
xmlDoc.SelectSingleNode("./order_synchro/seller/username").InnerText;
    String sellerReference =
xmlDoc.SelectSingleNode("./order_synchro/seller/reference").InnerText;

    // buyer information
    String buyerUsername =
xmlDoc.SelectSingleNode("./order_synchro/buyer/username").InnerText;
```

Sample Code for MonsterPay Synchro

```
        // shipping information
        String shippingAddressTitle =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/title").Inn
erText;
        String shippingAddressFirstname =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/firstname")
.InnerText;
        String shippingAddressLastname =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/lastname").
InnerText;
        String shippingAddressEmailAddress =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/email_addre
ss").InnerText;

        // shipping address information
        String shippingAddressStreet1 =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/street1").I
nnerText;
        String shippingAddressStreet2 =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/street2").I
nnerText;
        String shippingAddressCity =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/city").Inne
rText;
        String shippingAddressState =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/state").Inn
erText;
        String shippingAddressZip =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/zip").Inner
Text;
        String shippingAddressCountry =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/country").I
nnerText;
        String shippingAddressContactNumber =
xmlDoc.SelectSingleNode("./order_synchro/buyer/shipping_address/contact_num
ber").InnerText;

        // billing information
        String billingAddressTitle =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/title").Inne
rText;
        String billingAddressFirstname =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/firstname").
InnerText;
        String billingAddressLastname =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/lastname").I
nnerText;
        String billingAddressEmailAddress =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/email_addres
s").InnerText;

        //Buyer billing details
        String billingAddressStreet1 =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/street1").In
nerText;
        String billingAddressStreet2 =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/street2").In
nerText;
```

Sample Code for MonsterPay Synchro

```
        String billingAddressCity =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/city").InnerText;
        String billingAddressState =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/state").InnerText;
        String billingAddressZip =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/zip").InnerText;
        String billingAddressCountry =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/country").InnerText;
        String billingAddressContactNumber =
xmlDoc.SelectSingleNode("./order_synchro/buyer/billing_address/contact_number").InnerText;

        //payment instrument details
        String paymentInstrumentType =
xmlDoc.SelectSingleNode("./order_synchro/payment_instrument/type").InnerText;

        // financial details (amounts etc.)
        String financialTotal =
Convert.ToString(Convert.ToDouble(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_total").InnerText) / 100);
        String financialSubTotal =
Convert.ToString(Convert.ToDouble(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_sub_total").InnerText) / 100);
        String financialShipping =
Convert.ToString(Convert.ToDouble(xmlDoc.SelectSingleNode("./order_synchro/financial//amount_shipping").InnerText) / 100);
        String financialTax =
Convert.ToString(Convert.ToDouble(xmlDoc.SelectSingleNode("./order_synchro/financial/amount_tax").InnerText) / 100);
        String financialInsurance =
Convert.ToString(Convert.ToDouble(xmlDoc.SelectSingleNode("./order_synchro/financial//amount_insurance").InnerText) / 100);
        String financialCurrency =
xmlDoc.SelectSingleNode("./order_synchro/financial/currency").InnerText;
        String financialPromo =
xmlDoc.SelectSingleNode("./order_synchro/financial/amount_promo").InnerText;
;

        // additional information
        String additionalInfoShippingOption =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/shipping_option").InnerText;
        String additionalInfoTaxOption =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/tax_option").InnerText;
        String additionalInfoInsuranceOption =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/insurance_option").InnerText;
        String additionalInfoPromoCode =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/promo_code").InnerText;
        String additionalInfoPromoDescription =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/promo_short_description").InnerText;
```

Sample Code for MonsterPay Synchro

```
String additionalInfoComment =
xmlDoc.SelectSingleNode("./order_synchro/additional_information/comment").InnerText;

// amount of line items
int lineitems =
Convert.ToInt32(xmlDoc.SelectSingleNode("./order_synchro/line_items").Attributes.GetNamedItem("count").Value);

//integers for loops
int i = 0;
int iii = 0;

//arrays for line items
string[] lidSKU = new string[20];
string[] lidDescription = new string[20];
decimal[] lidPrice = new decimal[20];
string[] lidQuantity = new string[20];
string[] lidWeight = new string[20];
string[] lidheight = new string[20];
string[] lidwidth = new string[20];
string[] lidlength = new string[20];
string[] lidcolor = new string[20];
string[] lidSize = new string[20];
string[] lidhandling = new string[20];
string[] lidrequire_shipping = new string[20];
string[] lidis_voucher = new string[20];
string[] lidVoucher_email = new string[20];
string[] lidVoucher_sms = new string[20];
string[] lidOption1name = new string[20];
string[] lidOption1value = new string[20];
string[] lidOption2name = new string[20];
string[] lidOption2value = new string[20];

XmlNodeList nodelist;
//XmlNode node for items ;
nodelist = xmlDoc.SelectNodes("./order_synchro/line_items/lid");

//loop through items
foreach (XmlNode mynode in nodelist)
{
    //assign the values of each line item;
    lidSKU[i] = mynode.ChildNodes.Item(0).InnerText;
    lidDescription[i] = mynode.ChildNodes.Item(1).InnerText;
    lidPrice[i] =
Convert.ToDecimal((mynode.ChildNodes.Item(2).InnerText));
    lidPrice[i] = lidPrice[i] / 100;
    lidQuantity[i] = mynode.ChildNodes.Item(3).InnerText;
    lidWeight[i] = mynode.ChildNodes.Item(4).InnerText;
    lidheight[i] = mynode.ChildNodes.Item(5).InnerText;
    lidwidth[i] = mynode.ChildNodes.Item(6).InnerText;
    lidlength[i] = mynode.ChildNodes.Item(7).InnerText;
    lidcolor[i] = mynode.ChildNodes.Item(8).InnerText;
    lidSize[i] = mynode.ChildNodes.Item(9).InnerText;
    lidhandling[i] = mynode.ChildNodes.Item(10).InnerText;
    lidrequire_shipping[i] = mynode.ChildNodes.Item(11).InnerText;
    lidis_voucher[i] = mynode.ChildNodes.Item(12).InnerText;
    lidVoucher_email[i] = mynode.ChildNodes.Item(13).InnerText;
    lidVoucher_sms[i] = mynode.ChildNodes.Item(14).InnerText;
    lidOption1name[i] = mynode.ChildNodes.Item(15).InnerText;
```

Sample Code for MonsterPay Synchro

```
        lidOption1value[i] = mynode.ChildNodes.Item(16).InnerText;
        lidOption2name[i] = mynode.ChildNodes.Item(17).InnerText;
        lidOption2value[i] = mynode.ChildNodes.Item(18).InnerText;

        i = +1;
    }

    //XmlNode list for custom fields
    XmlNodeList customfields;

    //Strings for custom fields
    string[] customFieldname = new string[20];
    string[] customFieldvalue = new string[20];

    //get all custom fields
    customfields =
xmlDoc.SelectNodes("./order_synchro/custom_fields/customfield");
    foreach (XmlNode mynode2 in customfields)
    {
        customFieldname[iii] = mynode2.ChildNodes.Item(0).InnerText;
        customFieldvalue[iii] = mynode2.ChildNodes.Item(1).InnerText;
        iii = +1;
    }

    //This code will show the details on the web page
    //status
    Response.Write("Status : " + status + "<br>");

    //transaction information
    Response.Write("Transaction ID : " + orderID + "<br>");
    Response.Write("Funds Available : " + orderFundsAvailable +
"<br>");

    //outcome
    Response.Write("Error Code : " + errorCode + "<br>");
    Response.Write("Error Description : " + errorDescription + "<br>");
    Response.Write("Error Solution : " + errorSolution + "<br>");

    //alerts
    Response.Write("Alert :" + alert + "<br>");
    Response.Write("Alert Text :" + alertText + "<br>");

    //seller details
    Response.Write("Seller Identifier :" + sellerId + "<br>");
    Response.Write("Seller Username :" + sellerUserName + "<br>");
    Response.Write("Seller Reference :" + sellerReference + "<br>");

    //buyer information
    Response.Write("Buyer Username :" + buyerUsername + "<br>");

    //shipping address information
    Response.Write("<br> + \"Shipping Adress Information\" + "<br>");
    Response.Write("Title :" + shippingAddressTitle + "<br>");
    Response.Write("Name :" + shippingAddressFirstname + "<br>");
    Response.Write("Surname :" + shippingAddressLastname + "<br>");
    Response.Write("Email :" + shippingAddressEmailAddress + "<br>");

    //shipping address
    Response.Write("Street :" + shippingAddressStreet1 + "<br>");
```

Sample Code for MonsterPay Synchro

```
Response.Write("City :" + shippingAddressCity + "<br>");
Response.Write("State :" + shippingAddressState + "<br>");
Response.Write("Zip :" + shippingAddressZip + "<br>");
Response.Write("Country :" + shippingAddressCountry + "<br>");
Response.Write("Contact Number :" + shippingAddressContactNumber +
"<br>");

//billing address information
Response.Write("<br>" + "Billing Adress Information" + "<br>");
Response.Write("Title :" + billingAddressTitle + "<br>");
Response.Write("Name :" + billingAddressFirstname + "<br>");
Response.Write("Surname :" + billingAddressLastname + "<br>");
Response.Write("Email :" + billingAddressEmailAddress + "<br>");

//shipping address
Response.Write("Street :" + billingAddressStreet1 + "<br>");
Response.Write("City :" + billingAddressCity + "<br>");
Response.Write("State :" + billingAddressState + "<br>");
Response.Write("Zip :" + billingAddressZip + "<br>");
Response.Write("Country :" + billingAddressCountry + "<br>");
Response.Write("Contact Number :" + billingAddressContactNumber +
"<br>");

//Payment Instrument information
Response.Write("Payment Type :" + paymentInstrumentType + "<br>");

//Financial information
Response.Write("Total :" + financialTotal + "<br>");
Response.Write("Sub Total :" + financialSubTotal + "<br>");
Response.Write("Shipping :" + financialShipping + "<br>");
Response.Write("Tax :" + financialTax + "<br>");
Response.Write("Insurance :" + financialInsurance + "<br>");
Response.Write("Currency :" + financialCurrency + "<br>");
Response.Write("Promo :" + financialPromo + "<br>");

//Additional Information
Response.Write("Shipping Option :" + additionalInfoShippingOption +
"<br>");
Response.Write("Tax Option :" + additionalInfoTaxOption + "<br>");
Response.Write("Insurance Option :" + additionalInfoInsuranceOption
+ "<br>");
Response.Write("Promo Code :" + additionalInfoPromoCode + "<br>");
Response.Write("Promo Description :" +
additionalInfoPromoDescription + "<br>");
Response.Write("Comment :" + additionalInfoCommment + "<br>");

//Product(s) Info
for (int ii = 0; ii < lineitems; ii++)
{
    Response.Write("lidSKU :" + lidSKU[ii] + "<br>");
    Response.Write("lidDescription :" + lidDescription[ii] +
"<br>");

    Response.Write("lidPrice :" + lidPrice[ii] + "<br>");
    Response.Write("lidQuantity :" + lidQuantity[ii] + "<br>");
    Response.Write("lidWeight :" + lidWeight[ii] + "<br>");
    Response.Write("lidheigth :" + lidheigth[ii] + "<br>");
    Response.Write("lidwidth :" + lidwidth[ii] + "<br>");
    Response.Write("lidlength :" + lidlength[ii] + "<br>");
    Response.Write("lidcolor :" + lidcolor[ii] + "<br>");
    Response.Write("lidSize :" + lidSize[ii] + "<br>");
```

Sample Code for MonsterPay Synchro

```
        Response.Write("lidhandling : " + lidhandling[ii] + "<br>");
        Response.Write("lidrequire_shipping : " +
lidrequire_shipping[ii] + "<br>");
        Response.Write("lidis_voucher : " + lidis_voucher[ii] +
"<br>");
        Response.Write("lidVoucher_email : " + lidVoucher_email[ii] +
"<br>");
        Response.Write("lidVoucher_sms : " + lidVoucher_sms[ii] +
"<br>");
        Response.Write("lidOption1name : " + lidOption1name[ii] +
"<br>");
        Response.Write("lidOption1Value : " + lidOption1value[ii] +
"<br>");
        Response.Write("lidOption2name : " + lidOption2name[ii] +
"<br>");
        Response.Write("lidOption2value : " + lidOption2value[ii] +
"<br>");
    }

    //Custom fields info
    for (int iiii = 0; iiii < iii; iiii++)
    {
        Response.Write("Custom Field Name : " + customFieldname[iiii] +
"<br>");
        Response.Write("Custom Field Value : " + customFieldvalue[iiii]
+ "<br>");
    }
}

private bool validate()
{
    if (Request["tnxid"] == null)
    {
        ErrorMsg = "No tnxid value found!";
        Response.Write(ErrorMsg);
        return false;
    }

    if (Request["checksum"] == null)
    {
        ErrorMsg = "No checksum value found!";
        Response.Write(ErrorMsg);
        return false;
    }

    if (Request["parity"] == null)
    {
        ErrorMsg = "No parity value found!";
        Response.Write(ErrorMsg);
        return false;
    }
    return true;
}
}
```

Classic ASP VBScript Sample Code for Synchro HTTP Service

The following sample code can be used in the Redirect to Page or Backend Update Page on the merchant's web site:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%
    ' The following ASP page is used to handle the Auto-Redirect/Post
    ' after a user has completed payment via MonsterPay. It also deals with
    ' the MonsterPay HTTP Synchro Service.
    ' This page should be the target of the Auto-Redirect and Auto-Synchro
    ' facilities that can be set within MonsterPay's admin page.
    ' As discussed in the MonsterPay Implementation guide
    '(http://www.monsterpay.com/www/downloads/MonsterPay_Sample_Code_for_Synchro.
pdf)
    ' when payment is completed, MonsterPay will post three values to your
server:
    '     txnid
    '     checksum
    '     parity
    ' These three values plus some other info as set out in the implementation
    ' guide has to be sent back to the MonsterPay Synchro Service. The code below
serves
    ' as an example of how this is done in ASP using VBScript.

    ' First collect and define all the information that we will be posting to the
    ' HTTP Synchro Service
    txnID = Request("txnid")
    checksum = Request("checksum")
    parity = Request("parity")

    ' Replace the merchID, username and password values with your own
    ' details when you wish to implement this code.
    ' FOR TESTING LEAVE AS IS
    merchID = "1234567892"
    username = "testseller2@monsterpay.com"
    password = "testseller"

    ' validate that the txnid, checksum and parity values was returned
    ' and is present, return an error if the values are missing.
    errorMsg = ""
    if txnID = "" then
        errorMsg = errorMsg&"txnid not found<br>"
    end if
    if checksum = "" then
        errorMsg = errorMsg&"checksum not found<br>"
    end if
    if parity = "" then
        errorMsg = errorMsg&"parity not found<br>"
    end if
    if not errorMsg = "" then
        response.Write(errorMsg)
        response.Flush()
        response.End()
    end if

    ' generate a single string containing all the information that
    ' needs to be posted to MonsterPay
    infoToSend = infoToSend & "&identifier=" & merchID
    infoToSend = infoToSend & "&usrname=" & username
    infoToSend = infoToSend & "&pwd=" & password
    infoToSend = infoToSend & "&txnid=" & txnID
    infoToSend = infoToSend & "&checksum=" & checksum
    infoToSend = infoToSend & "&parity=" & parity
    infoToSend = infoToSend & "&Method=order_synchro"
```

Sample Code for MonsterPay Synchro

```
' To post the information to the MonsterPay Synchro Service you will need a
component called MSXML.
' This component id freely available at:
' http://www.microsoft.com/downloads/details.aspx?FamilyID=3144b72b-b4f2-
46da-b4b6-c5d7485f2b42&DisplayLang=en

' Here we create an instance of the component to post with.
' NOTE that the URL we are posting to contains the complete URL encoded
information set that is needed
' by the Synchro Service.
Set xmlObj = Server.CreateObject("Msxml2.ServerXMLHTTP")
xmlObj.Open "POST",
"https://www.monsterpay.com/secure/components/synchro.cfc?wsdl" & infoToSend, FALSE

' Now send the post.
xmlObj.Send

' Get the returned information from the Synchor Service and remove HTML
encoding.
' The HTMLDecode replaces the &gt; encoding to normal angled bracket < etc
xmlString = HTMLDecode(xmlObj.responseText)

' As we have completed the post, we can now discard xmlObj
Set xmlObj = nothing

' The string that was returned by the MonsterPay Synchoro is in xml format,
but there is some info that
' we don't need. the following statement strips out the relevent xml markup.
xmlString = Mid(xmlString, inStr(1, xmlString, "<order_synchro>"), (inStr(1,
xmlString, "</order_synchro>")- inStr(1, xmlString, "<order_synchro>") + 16))

' To view the xml in Internet Explorer uncomment the following code block:
' Response.Write("<?xml version=""1.0"" encoding=""UTF-8""?>" & xmlString)
' Response.Flush()
' Response.End()

' Now that we have the returnd xml from the synchro service all we need to do
is extract the relevant
' information to display and store. To do this we need to somehow parse the
xml. This is achieved via the
' same Microsoft Component we used earlier.

' The following code block creates a xml document object from the xml string
that was returned by the
' MonsterPay Synchro Service and the extracts the information that is
displayed to the client.
set xmlDocObj = Server.CreateObject("Microsoft.XMLDOM")
xmlDocObj.async = false
xmlDocObj.resolveExternals = false
xmlDocObj.loadXML(xmlString)
xmlDocObj.setProperty "SelectionLanguage", "XPath"

'retrieve outcome of the transaction
status = xmlDocObj.selectSingleNode("//order_synchro/outcome/status").text

' validate error code in order to display declined details
err_code =
xmlDocObj.selectSingleNode("//order_synchro/outcome/error_code").text
if err_code <> "0" then

' retrieve error details
err_desc =
xmlDocObj.selectSingleNode("//order_synchro/outcome/error_desc").text
err_solution =
xmlDocObj.selectSingleNode("//order_synchro/outcome/error_solution").text
' display error details to buyer
response.Write("<br><br><b>Your payment is " & status & "</b><br>" & _
```

Sample Code for MonsterPay Synchro

```

        "<b>Error Code:</b>" & err_code & "<br>" & _
        "<b>Error Description:</b>" & err_desc & "<br>" & _
        "<b>Error Solution:</b>" & err_solution & "<br>" & _
        "<b>Transacton ID:</b>" & tn timer
    else
        ref =
xmlDocObj.selectSingleNode("//order_synchro/seller/reference").text
        method =
xmlDocObj.selectSingleNode("//order_synchro/payment_instrument/type").text
        amount =
cDbl(xmlDocObj.selectSingleNode("//order_synchro/financial/amount_total").text)/100
        sku =
xmlDocObj.selectSingleNode("//order_synchro/line_items/lid/sku").text

        ' retrieve transaction details
        tn timer
xmlDocObj.selectSingleNode("//order_synchro/outcome/order/id").text
        funds_avail_date =
xmlDocObj.selectSingleNode("//order_synchro/outcome/order/funds_available").text
        ' order not declined
        tn timer = ""
        ' retrieve transaction specific details for the buyer
        Set oRoot =
xmlDocObj.selectSingleNode("//order_synchro/outcome/order/alerts/alert")
        for each x in oRoot.childNodes
            Select Case x.baseName
                case "text"
                    tn timer = x.text
                case else
            end select
        next

        ' validate the payment type in order to display the correct
information
        response.Write("<br><br><b>Your payment is " & status &
"</b><br><br>")

        ' validate priority of tn timer
if tn timer <> "" then
    response.Write("<br><br>" & tn timer & "<br>")
end if

        response.Write("<b>Order Details:</b><br><b>Ref:</b>" & ref & "<br>"
& _
        "<b>Product Code:</b>" & sku & "<br>" & _
        "<b>Payment Method:</b>" & method & "<br>" & _
        "<b>Total:</b>" & amount & "<br>" & _
        "<b>Funds Available:</b>" & funds_avail_date)

        comment =
xmlDocObj.selectSingleNode("//order_synchro/additional_information/comment").text
        response.Write("<br><b>comment:</b>" & comment & "<br>")

    end if

    ' We're done with the xml parser object, so discard it
set xmlDocObj = nothing

    ' This function serves to remove html encoding from any string passed to it.
Private Function HTMLDecode(ByVal encodedstring)
    Dim tmp, i
    tmp = encodedstring
    tmp = Replace( tmp, "&quot;", chr(34) )
    tmp = Replace( tmp, "&lt;", chr(60) )
    tmp = Replace( tmp, "&gt;", chr(62) )

```

Sample Code for MonsterPay Synchro

```
tmp = Replace( tmp, "&#38;" , chr(38) )
tmp = Replace( tmp, "&#32;" , chr(32) )
For i = 1 to 255
    tmp = Replace( tmp, "&#" & i & ";" , chr( i ) )
Next
HTMLDecode = tmp
End Function
```

%>

PHP Sample Code for Synchro HTTP Service

The following are sample codes that can be used in the Redirect to Page or Backend Update Page on the merchant's web site:

```
<?php
/*
REQUIREMENTS:
PHP5 with the following libraries installed and enabled:
1. libcurl v7.9.0 or later. libcurl is an implementation
of CURL, which allows you to send XML over HTTPS using
server-to-server HTTP POST requests. Learn more about
libcurl at http://us2.php.net/curl/.
2. WDDX Functions. In order to use WDDX, you will need
to install the expat library (which comes with
Apache 1.3.7 or higher).
Learn more about WDDX at http://www.openwddx.org/
and http://www.php.net/manual/en/ref.wddx.php
3. SimpleXML
*/
/*
Before reading through this example, make sure that you have
read the MonsterPay Implementation Guide available from
http://www.monsterpay.com/www/downloads/MonsterPay\_Implementation\_Guide.pdf
This code is given as is and is intended to serve as an example.
The code demonstrates how to post back the txnid, checksum and parity
values to the MonsterPay HTTP Synchro service.
It also shows how to prepare the information that is returned from
the Synchro service for displaying to the client (very basic).
*/
if (!function_exists('wddx_deserialize')) {
// Clone implementation of wddx_deserialize
function wddx_deserialize($xmlpacket) {
if ($xmlpacket instanceof SimpleXMLElement) {
if (!empty($xmlpacket->struct)) {
$struct = array();
foreach ($xmlpacket->xpath("struct/var") as
$var) {
if (!empty($var["name"])) {
$key = (string) $var["name"];
$struct[$key] = wddx_deserialize($var);
}
}
return $struct;
} elseif (!empty($xmlpacket->array)) {
$array = array();
foreach ($xmlpacket->xpath("array/*") as $var) {
array_push($array, wddx_deserialize($var));
}
return $array;
} else if (!empty($xmlpacket->string)) {
return (string) $xmlpacket->string;
} else if (!empty($xmlpacket->number)) {
return (int) $xmlpacket->number;
} else {
if (is_numeric((string) $xmlpacket)) {
return (int) $xmlpacket;
} else {
return (string) $xmlpacket;
}
}
} else {
$xml = simplexml_load_string($xmlpacket);
$datanode = $xml->xpath("/wddxPacket[@version='1.0']/data");

```

Sample Code for MonsterPay Synchro

```
return wddx_deserialize($datanode[0]);
}
}
}
/*
Declare the variables that will hold the information that needs to be
sent to the HTTP Synchro Service.
The first three (tnxid, checksum and parity) are sent from MonsterPay via
URL encoding.
Replace the $merchID, $username and $password values with your own
details when you wish to implement this code. FOR TESTING LEAVE AS IS
*/
$tnxid = $_GET["tnxid"];
$checksum = $_GET["checksum"];
$parity = $_GET["parity"];
$errMsg = "";
if(empty($tnxid)) {
$errMsg .= "tnxid not found<br>";
}
if(empty($checksum)) {
$errMsg .= "checksum not found<br>";
}
if(empty($parity)) {
$errMsg .= "parity not found<br>";
}
if($errMsg != "") {
die($errMsg);
}
// Web page receives Synchro Auto-Redirect variables from MonsterPay.
// The following details should be used for testing
// Must use your own account details when going live.
$merchID = "1234567891";
$username = "testseller1@setcom.com";
$password = "testseller";
/*
Now we open a communication line to the HTTP Synchro Service.
*/
/* Creates variable ($monsterpay_string) - Synchro Auto-Redirect
variables + Identifier & Username & Password.*/
$monsterpay_string = 'method=' . 'order_synchro' .
'&identifier=' . $merchID .
'&username=' . $username .
'&pwd=' . $password .
'&tnxid=' . $tnxid .
'&checksum=' . $checksum .
'&parity=' . $parity;
// send $monsterpay_string to MonsterPay by utilizing CURL
$monsterpay_url = "https://www.monsterpay.com/secure/components/synchro.cfc?wsdl";
// MonsterPay Synchro url
$ch = curl_init(); // initialize curl handle
curl_setopt($ch, CURLOPT_URL, $monsterpay_url); // set $setcom_url to post to
MonsterPay
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1); // return into a variable
curl_setopt($ch, CURLOPT_POST, 1); // set POST method
curl_setopt($ch, CURLOPT_POSTFIELDS, $monsterpay_string); // set Post variable
$monsterpay_result = curl_exec($ch); // Perform the POST and get the data returned
by MonsterPay.
if (curl_errno($ch)) {
$curlError = curl_error($ch); // If CURL returns an error, stores it in a variable.
echo "error" . $curlError . "<br>";
} else $curlError = '';
curl_close($ch);
if (empty($curlError)) {
/*
```

Sample Code for MonsterPay Synchro

Filter data returned from the MonsterPay Synchro Service. What we are after is the WDDX Packet so we filter the string to get just the wddx packet and then de-serialize it so that we are left with a string of xml data. See the MonsterPay Implementation Guide for the structure of the xml data.

In displaying the details from the xml we make use of a user defined function to get the correct currency symbol. The GetCurrencySymbol function

is very straight forward. It takes the currency code from the xml and returns

the correct currency symbol. If it can't find the correct symbol it returns

the given country code.

```
*/
$monsterpay_wddx = trim($monsterpay_result);
$monsterpay_xml = wddx_deserialize($monsterpay_wddx);
$order_synchro = simplexml_load_string($monsterpay_xml);
//tnx details
$tnx_status = $order_synchro->outcome->status;
$tnx_id = $order_synchro->outcome->order->id;
$funds_avail = $order_synchro->outcome->order->funds_available;
//error details
$error_code = $order_synchro->outcome->error_code;
$error_desc = $order_synchro->outcome->error_desc;
$error_solution = $order_synchro->outcome->error_solution;
//seller details
$seller_ref = $order_synchro->seller->reference;
$seller_email = $order_synchro->seller->username;
//buyer details
$buyer_ref = $order_synchro->buyer->reference;
$buyer_uname = $order_synchro->buyer->username;
$buyer_title = $order_synchro->buyer->billing_address->title;
$buyer_fname = $order_synchro->buyer->billing_address->firstname;
$buyer_lname = $order_synchro->buyer->billing_address->lastname;
$buyer_email = $order_synchro->buyer->billing_address->email_address;
$buyer_street1 = $order_synchro->buyer->billing_address->street1;
$buyer_street2 = $order_synchro->buyer->billing_address->street2;
$buyer_city = $order_synchro->buyer->billing_address->city;
$buyer_state = $order_synchro->buyer->billing_address->state;
$buyer_zip = $order_synchro->buyer->billing_address->zip;
$buyer_country = $order_synchro->buyer->billing_address->country;
$buyer_cnumber = $order_synchro->buyer->billing_address->contact_number;
//payment details
$pmt_type = $order_synchro->payment_instrument->type;
//financial details
$tnx_amount = $order_synchro->financial->amount_total;
$currency = $order_synchro->financial->currency;
//Reformat amount values
function getCurrency($amt, $cur) {
if (strlen($cur) > 0) {
switch (strtoupper($cur)) {
case 'ZAR' :
$cur_sym = 'R';
break;
case 'GBP' :
$cur_sym = '£';
break;
case 'USD' :
$cur_sym = '$';
break;
case 'EUR' :
$cur_sym = '€';
break;
default:
$cur_sym = '';
}
}
}
```

Sample Code for MonsterPay Synchro

```
} else $cur_sym = '';
if ($amt < 0) { $amt=abs($amt);
$new_amt = '-' . $cur_sym . number_format(($amt /
100),2,'.','');
} else $new_amt = $cur_sym . number_format(($amt /
100),2,'.','');
return $new_amt;
}
/*validation of transaction outcome in order to display the correct
information to the buyer */
if ($tnx_status != 'Complete') //transaction is unsuccessful
{
//display error information returned by MonsterPay
echo "<strong>Your order has been Declined.</strong><br><br>";
echo "Sorry " . $buyer_title . " " . $buyer_fname . " " .
$buyer_surname . " your payment to " . $seller_email . " was
<strong>unsuccessful</strong>.<br><br>";
echo "Why was my order declined?<br>";
echo "Error Code: " . $error_code . "<br>";
echo "Error Description: " . $error_desc . "<br>";
echo "Error Solution: " . $error_solution . "<br>";
} //end txn declined
else //transaction is successful
{
//format txn amount to include txn currency
$tnx_amount = getCurrency($tnx_amount,$currency);
//output on screen for buyer
echo "Your transaction is <strong>" . $tnx_status . "</strong><br><br>";
/*validate the transaction type in order to display the correct information to the
buyer*/
if ($pmt_type != "Credit Card") //non credit card funded payment
{
/*loop through all the transaction notifications and display the details to the
buyer*/
foreach ($order_synchro->outcome->order->alerts->alert as $alerts)
{
echo "Transaction Notification:
<br><br><fieldset>" . $alerts->text . "<br></fieldset><br>";
}
}
echo "<strong>Transaction ID</strong>: " . $tnx_id . "<br>";
echo "<strong>Transaction Amount</strong>: " . $tnx_amount .
"<br>";
echo "<strong>Funds Available</strong>: " . $funds_avail .
"<br>";
echo "<strong>Reference</strong>: " . $seller_ref . "<br>";
} //end txn successful
} // END - [if (empty($sCurlError))]
?>
```

Coldfusion Sample Code for Synchro Web Service

The following are sample codes that can be used in the Redirect to Page or Backend Update Page on the merchant's web site:

```
<!---
    The following Coldfusion page is used to handle the Auto-Redirect/Post
    after the buyer has completed the payment via MonsterPay. It also deals
with
    the MonsterPay HTTP Synchro Service.
    The Auto-Redirect/Post URL in the Seller's MonsterPay account should
    point to this page.
    As discussed in the MonsterPay Implementation guide
    (http://www.MonsterPay.com/www/downloads/MonsterPay_Sample_Code_for_Synchro.pdf)
    when payment is completed, MonsterPay will post three values to your
server:
        tnxid
        checksum
        parity
    These three values plus some other info as set out in the implementation
    guide has to be sent back to the MonsterPay Synchro Service. The code below
serves
    as an example of how this is done in Coldfusion.

    First collect and define all the information that we will be posting to the
    HTTP Synchro Service
    --->

<!---To send the variables to MonsterPay for validation you will need to invoke the
MonsterPay Syncho Web Service--->
<CFINVOKE WEBSERVICE="https://www.monsterpay.com/secure/components/synchro.cfc?wsdl"
METHOD="order_synchro" RETURNVARIABLE="synchro_res">
<!---
    Replace the merchID, username and password values with your own
    details when you wish to implement this code.
    FOR TESTING LEAVE AS IS
    --->
    <CFINVOKEARGUMENT NAME="identifier" VALUE="1234567892">
    <CFINVOKEARGUMENT NAME="usrname" VALUE="testseller2@monsterpay.com">
    <CFINVOKEARGUMENT NAME="pwd" VALUE="testseller">
<!---end test account details--->
<!---below are the three variables that was posted to the Auto-Redirect/Post URL
from MonsterPay--->
    <CFINVOKEARGUMENT NAME="tnxid" VALUE="#trim(tnxid)#">
    <CFINVOKEARGUMENT NAME="checksum" VALUE="#trim(checksum)#">
    <CFINVOKEARGUMENT NAME="parity" VALUE="#trim(parity)#">
</CFINVOKE>

<!--- parse XML response returned from MonsterPay--->
<CFOBJECT ACTION="CREATE" CLASS="MSXML2.DOMDocument.4.0" NAME="oXml" TYPE="COM">
<CFSET oXml.async=false>
<CFSET oXml.resolveExternals=false>
<CFSET oXml.validateOnParse=true>
<CFSET oXml.loadXML(synchro_res)>

<cfset notification = "">
<!---script retrieves important transaction information that must be displayed to
the buyer--->
<cfscript>
    myxmldoc = XmlParse(synchro_res);
    selectedElements = XmlSearch(myxmldoc,
"/order_synchro/outcome/order/alerts/alert/text");
    for (i = 1; i LTE ArrayLen(selectedElements); i = i + 1)
```

Sample Code for MonsterPay Synchro

```
{
    notification = selectedElements[i].XmlText;
}
</cfscript>

<!---validate for XML parsing error--->
<CFIF oXml.parseError.errorCode NEQ 0>
    <div>
        <TABLE BORDER=0 WIDTH="439" CELLPADDING=0 CELLSPACING=0>
            <TR>
                <TD BGCOLOR=#D2E4FC VALIGN=MIDDLE ALIGN=LEFT STYLE="font-size:16px"><B>Auto-
Synchro Demo: An error occurred</B></font> </TD>
            </TR>
            <TR>
                <TD VALIGN=TOP> Unable to continue due to the response string that could not
be parsed. The error details are attached below: <BR>
                Error code: <CFOUTPUT>#oXml.parseError.errorCode#</CFOUTPUT> <BR>
                Description: <CFOUTPUT>#oXml.parseError.reason#</CFOUTPUT> </TD>
            </TR>
        </TABLE>
    </div>

<CFELSE>
    <!--- retrieve info from response --->
    <CFSET outcome = oXml.selectSingleNode('/order_synchro/outcome/status')>
    <CFSET err_code = oXml.selectSingleNode('/order_synchro/outcome/error_code')>
    <CFSET err_desc =
oXml.selectSingleNode('/order_synchro/outcome/error_description')>
    <CFSET err_solution =
oXml.selectSingleNode('/order_synchro/outcome/error_solution')>

    <CFIF NOT isDefined("outcome")>
        <div>
            <TABLE BORDER=0 WIDTH="439" CELLPADDING=0 CELLSPACING=0>
                <TR>
                    <TD BGCOLOR=#D2E4FC VALIGN=MIDDLE ALIGN=LEFT STYLE="font-
size:16px"><B>Auto-Synchro Demo: An error occurred</B></font> </TD>
                </TR>
                <TR>
                    <TD VALIGN=TOP> Unable to continue due to the unknown transaction outcome.
Please contact <A HREF="mailto:support@monsterpay.com">support@monsterpay.com</A>
for more assistance. </TD>
                </TR>
            </TABLE>
        </div>

        <CFELSEIF trim(outcome.text) EQ "Complete" OR trim(outcome.text) EQ "Pending">
            <!---display required transaction information if the outcome of the transaction
is either Pending or Complete--->
            <cfset funds_avail =
oXml.selectSingleNode('/order_synchro/outcome/order/funds_available').text>
            <cfset txn_type =
oXml.selectSingleNode('/order_synchro/payment_instrument/type').text>

            <div>
                <TABLE BORDER=0 WIDTH="439" CELLPADDING=0 CELLSPACING=0>
                    <TR>
                        <TD BGCOLOR=#D2E4FC VALIGN=MIDDLE ALIGN=LEFT STYLE="font-
size:16px"><B>Auto-Synchro Demo: Your Transaction is
<cfoutput>#trim(outcome.text)#</cfoutput></B></font> </TD>
                    </TR>
                    <TR>
                        <TD VALIGN=TOP> Your transaction is
<cfoutput>#trim(outcome.text)#</cfoutput>. The transaction details are attached
below:
                        <TABLE BORDER="0" CELLPADDING="2" CELLSPACING="0" WIDTH="100%">
                            <TR VALIGN="TOP">
```

Sample Code for MonsterPay Synchro

```

        <TD WIDTH="30%" ALIGN="RIGHT">Transaction Notification: </TD>
        <TD WIDTH="80%"><CFOUTPUT>#notification#</CFOUTPUT></TD>
    </TR>
    <TR VALIGN="TOP">
        <TD WIDTH="30%" ALIGN="RIGHT">Transaction ID: </TD>
        <TD WIDTH="80%"><CFOUTPUT>#tnxid#</CFOUTPUT></TD>
    </TR>

    <cfif tnx_type EQ "eCheque">
        <TR VALIGN="TOP">
            <TD WIDTH="30%" ALIGN="RIGHT">Funds Available: </TD>
            <TD WIDTH="80%"><CFOUTPUT>#funds_avail#</CFOUTPUT></TD>
        </TR>
    </cfif>

    <CFSET amount =
oXml.selectSingleNode('/order_synchro/financial/amount_total')>
    <CFSET currency =
oXml.selectSingleNode('/order_synchro/financial/currency')>

    <TR VALIGN="TOP">
        <TD ALIGN="RIGHT">Amount: </TD>
        <TD><CFOUTPUT>
            <CFIF isDefined("currency")>
                #currency.text#
            </CFIF>

            <CFIF isDefined("amount")>
                <CFIF isNumeric(amount.text)>
                    #decimalFormat(amount.text/100)#
                <CFELSE>
                    Unavailable
                </CFIF>
            <CFELSE>
                Unavailable.
            </CFIF>
        </CFOUTPUT></TD>
    </TR>
    <CFSET ref = oXml.selectSingleNode('/order_synchro/seller/reference')>
    <TR VALIGN="TOP">
        <TD ALIGN="RIGHT">Reference: </TD>
        <TD><CFOUTPUT>
            <CFIF isDefined("ref")>
                #ref.text#
            </CFIF>
        </CFOUTPUT></TD>
    </TR>
</TABLE></TD>
</TR>
</TABLE>
</div>

<CFELSE>
    <!--the transaction was declined or there was an error with the payment-->
    <div>
        <TABLE BORDER=0 WIDTH="439" CELLPADDING=0 CELLSPACING=0>
            <TR>
                <TD BGCOLOR=#D2E4FC VALIGN=MIDDLE ALIGN=LEFT STYLE="font-size:16px"><B>Auto-Synchro Demo: Your Transaction was Unsuccessful</B></font> </TD>
            </TR>
            <TR>
                <TD VALIGN=TOP> Your transaction was unsuccessful. The error details are attached below: <BR>
                    <!--display error information to the buyer-->
                    Error Code:
                    <CFIF isDefined("err_code")>
                        <CFOUTPUT>#err_code.text#</CFOUTPUT>

```

Sample Code for MonsterPay Synchro

```
        <CFELSE>
            Unavailable.
        </CFIF>
    <BR>

    Description:
    <CFIF isDefined("err_desc")>
        <CFOUTPUT>#err_desc.text#</CFOUTPUT>
    <CFELSE>
        Unavailable.
    </CFIF>
    <BR>

    Solution:
    <CFIF isDefined("err_solution")>
        <CFOUTPUT>#err_solution.text#</CFOUTPUT>
    <CFELSE>
        Unavailable.
    </CFIF>
</TD>
</TR>
</TABLE>
</div>
</CFIF>
</CFIF>
```

Additional Information

For additional information, please contact MonsterPay's Sales Department.

Tel:	United Kingdom	+44 (20) 3051 6320
	United States of America	+1 (408) 850 6530
	South Africa	+27 (83) 913 0000

Fax:	United Kingdom	+44 (20) 7681 3303
	United States of America	+1 (408) 351 8057
	South Africa	(086) 615 1486

Email sales@monsterpay.com

Web www.monsterpay.com