



Point Managers have the ability to deliver History and Event data to a web service using HTTP and XML. This Application Note describes a simple example of a web service that can receive data from Point Managers via HTTP/XML using Microsoft Internet Information Services (using asp scripting in Visual Basic) and storing the data in an SQL table.

The example files can be downloaded from <http://www.pointsix.com/download/XMLServiceExample.zip>

The following is a link that describes how to use ASP with Internet Information Services:
<http://support.microsoft.com/default.aspx?scid=kb;en-us;297943>

ASP Script

```
<%  
    Response.Expires = -1000  
  
    set conn=CreateObject("ADODB.Connection")  
    conn.open = "Driver={SQL Server};Server=(local);Database=P6History;Uid=p6x;Pwd=sample"  
  
    set objXML = Server.CreateObject("Microsoft.XMLDOM")  
    objXML.ValidateOnParse = True  
  
    objXML.Load(Request)  
    If objXML.ParseError.errorCode <> 0 Then  
        Response.Write("<ErrorList><Error>" & objXML.parseError.reason & "At line: " & objXML.parseError.line &  
        "</Error></ErrorList>")  
        conn.close  
    Else  
        Set objRootElement = objXML.documentElement  
    End If  
  
    dRecID = objXML.documentElement.getAttribute("ID")  
    dRecName = objXML.documentElement.getAttribute("Name")  
    dRecNoSensor = objXML.documentElement.getAttribute("NoSensors")  
  
    Set objNode = objXML.documentElement.firstChild  
  
    For Each xmlPNode In objRootElement.childNodes  
        For Each xmlNode In xmlPNode.childNodes  
            Select Case xmlNode.nodeName  
                case "TimeStamp"  
                    tStamp = xmlNode.text  
                case "HIndex"  
                    hIndex = xmlNode.text  
                case "PointID"  
                    pID = xmlNode.text  
                case "SensorID"  
                    sID = xmlNode.text  
                case "SensorName"  
                    sName = xmlNode.text  
                case "SensorType"  
                    sType = xmlNode.text  
                case "PointIndex"  
                    pIndex =xmlNode.text  
                case "PointType"  
                    pType = xmlNode.text  
                case "Value"  
                    val = xmlNode.text  
                case "Units"  
                    unit = xmlNode.text  
                case "Status"  
                    stat = xmlNode.text  
            End Select  
        Next  
  
        sql = "INSERT INTO RawHistory  
(ManID,ManName,RecTime,Hindex,PointID,SensorID,SensorName,SensorType,PointIndex,PointType,IOValue,Units,Status)"  
        sql = sql & " VALUES ('" & dRecID & "','" & dRecName & "','" & tStamp & "','" & hIndex & "','" & pId &  
        "','" & sID & "','" & sName & "','" & sType & "','"  
        sql = sql & pIndex & "','" & pType & "','" & val & "','" & unit & "','" & stat & "')"  
  
        //Response.Write(sql)
```

```

conn.Execute(sql & ";" )
Next

xmlString = "<?xml version=""1.0""?>" & vbCrLf
xmlString = xmlString & "<ErrorList>" & vbCrLf
xmlString = xmlString & " <Success>Download Complete</Success>" & vbCrLf
xmlString = xmlString & "</ErrorList>"

Response.Write(xmlString)

conn.Close
Set objNode = Nothing
set objXML = Nothing
set conn = Nothing

%>

```

The script first opens a connection to the SQL database called P6History using an ODBC database driver. The script starts the XMLDOM component and then loads the XML data. The XMLDOM component then parses the XML data from the Point Manager into individual components. The script iterates through the parsed components and then inserts the data into an SQL table called RawHistory. If the data was successfully parsed and then posted to the database, the script then sends an acknowledgement via HTTP to the Point Manager. If there was an error, the script sends the error response via HTTP to the Point Manager.

SQL Table Description

The database name is "P6History". The table is called "RawHistory".

Field Name	Type	Size	Description
ManID	char	17	Point Manager Ethernet Address
ManName	char	19	Point Manager Name
RecTime	datetime		Time stamp of the history record
Hindex	tinyint		Used to distinguish between multiple records with the same time stamp
PointID	char	20	"SensorID_PointIndex"
SensorID	char	16	Sensor serial number
SensorName	char	20	Sensor name
SensorType	tinyint		Sensor type
PointIndex	tinyint		The I/O position in the sensor
PointType	tinyint		The I/O points data type
IOValue	char	20	Value of the I/O point. Set to "na" when Status = 2.
Units	char	8	Engineering units of the value
Status	tinyint		0 – online; 2 - offline

Point Manager Setup

The following is list commands used to setup the Point Manager.

Command	Comment
XTS1	Set time format
XDHS1,66.147.93.113,/P6/P6Hist.asp,80	XML Delivery Destination
XPHS1,010105000030,00000100,2,15	XML Delivery Attempt Interval
HOP	Offline indicator is logged when no updates are received during the History Interval
HIS010105000000,00000100	Set the History Interval

The sensors were installed using the Point Manager's AutoAdd function. Make sure you have the Point Manager TCP/IP settings set correctly. You will need to have a Name Server IP address.