How-To Guide

WebSign 3 Simple Demo

Table of Contents

Overview

The Topaz WebSign 3 Simple Demo is an easy-to-use demo that illustrates how developers can integrate the robust signature capture capability offered by SigPlus into Active Server Pages and web based database technology. For proper functionality, SigPlus ActiveX will need to be installed on the client machine.

This document is meant for developers with a comprehensive understanding of HTML, VBscript, and Active Server Pages, not for the lay end user.

Download the WebSign3 files at: **www.topazsystems.com/Software/WebSign3.zip**

Then, unzip the files from the zipped directory 'aspsimpleexample'. There will be a number of files unzipped.
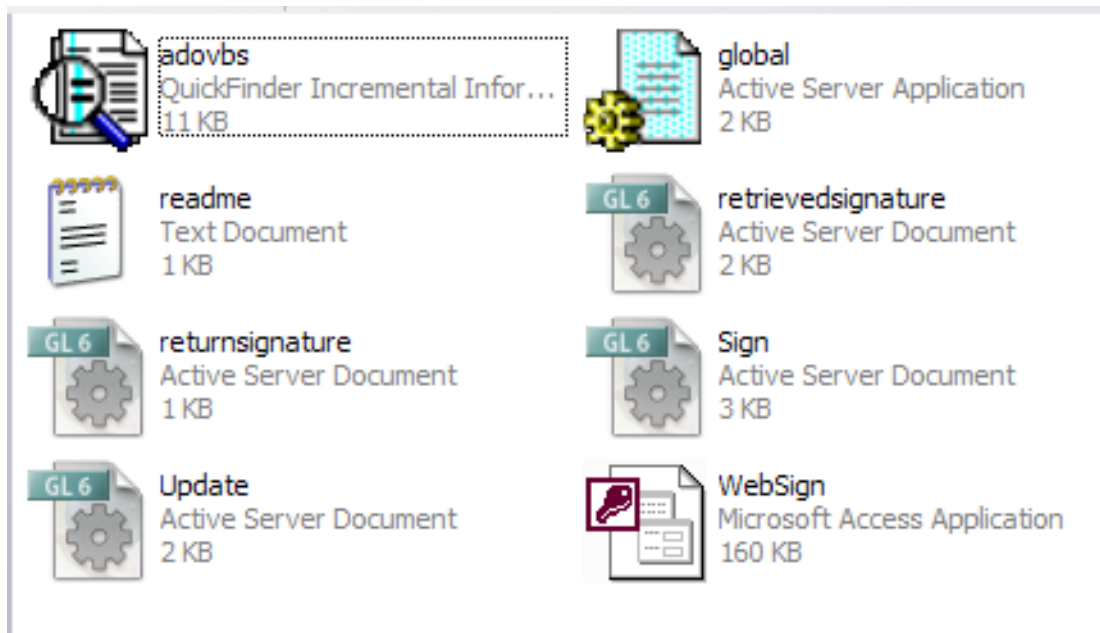
All of these files need to be uploaded into the server. The "adobvs.inc" file is a list of variables used with Microsoft Access that must be there for the database to work correctly. The "global.asa" file serves as a place where session variables (those variables that can hold info from the client to the server) are created and objects to be used are referenced. The "readme.txt" is simply the most basic directions on what ASPs to use and in what order for the demo to function correctly. The "WebSign.mdb" is the Access database that stores the signature data. The remaining files are Active Server Documents. These are the pages that the demo works through and will be discussed in detail below.

Sign.asp

The client-side "Sign" page is the first page that the user will start with. The code for this page is very simple comprised almost entirely of VBscript and HTML code. One important part of the code that differs from the simple construction of the DemoJavascript pages is that the Sign page uses the POST method to post the signature to the server in the form of an ASCII hex string, then runs Update.asp. This code can be seen below (if opened in a basic text editor such as Notepad):

```
//-->
</SCRIPT>
<FORM action="Update.asp" id=FORM1 method=post name=FORM1>
<P>        &nb
;         &nbsp
nbsp;         &
sp;         &nb
```

When the "Sign" page is loaded onto the server and run, it looks like this:

Type In Your Name: [   ]

Clear   Cancel   Sign   Save

To begin, the user simply needs to type in their name in the designated field and click " Sign". This will open the tablet and enable signature capture, as well as display the "Time Stamp" that will be included in the signature data throughout the WebSign3 demo process.

Once the "Time Stamp" appears, sign on the Topaz signature capture tablet.

Ea Smith

1/16/2015 3:41:26 PM

Type In Your Name: [EA Smith]

Clear   Cancel   Sign   Save

As the user is signing the tablet, the signature will appear in real time on the page. If the user wishes to end the signing instance, they click "Cancel". If they are not satisfied with their signature, they can click "Clear" and sign again. Once the signature is completed, it will look something like this:

Once the signature is complete, the user will click "Save". This will post the signature to the server and open the Update page.

Update.asp

The "Update" page runs server side once the user chooses to save and post their signature. The "Update" page essentially adds the new record to the specified table in the "WebSign" database that is located on the server. "Update.asp" differs from "Sign.asp" in that it contains the ASP code that is needed to perform the necessary server side functions (marked by the <% and %> tags). This code is responsible for adding the record, requesting the name, sig data, date and time from the server, and displaying it on the resulting client page. Additionally, "Update" returns an id number that the user will later use to retrieve their signature record. The ASP code can be seen below:

```
<%
Dim cnSignature
Dim rsSignature

Set cnSignature = CreateObject( "ADODB.Connection" )
Set rsSignature = CreateObject( "ADODB.Recordset" )

With cnSignature
    .Provider = "Microsoft.Jet.OLEDB.4.0"
    .ConnectionString = "Data Source =" & Server.MapPath ("/websignsimple/webSign.mdb")
    .Open
End With

rsSignature.CursorType = 1
rsSignature.LockType = 3

rsSignature.Open "SELECT * FROM Contracts", cnSignature, 1, 3

rsSignature.AddNew
rsSignature( "Name" ) = Request.Form("NameField")
rsSignature( "Signature" ) = Request.Form("SigField")
rsSignature( "DateTime" ) = Request.Form("DTStamp")
rsSignature.Update
Session("sigStr")=Request.Form("SigField")
Session("dateTimeS")=Request.Form("DTStamp")
%>
```
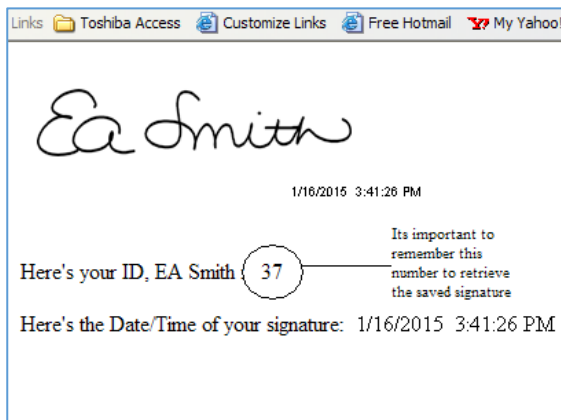
When loaded into the server and ran, the page will open with the signature, id number, date, time, and name. It should look something like this:

Once the "Update" page is opened, the signature data is saved in the server. Then, the user must remember their user ID, which will be needed to retrieve the saved signature.

Returnsignature.asp

The "Returnsignature" page is where users can go to retrieve their signature record. This page contains a simple form that will ask for the user's ID number and have a "Submit" button. The code for this page is very simple and once again, makes use of the POST command as with the "Sign" page.

```
<BODY>
<FORM action="retrievedsignature.asp" id=FORM1 method=post name=FORM1>
<FONT COLOR='#666699'><h2>Return Signature For Viewing</h2></Font><BR><B
<h4>Enter your existing ID here to return the stored signature: &nb
<INPUT id=IDField name=IDField></P>
<INPUT id=button1 name=DoneBtn type=button value=Submit language ="VBScr
style="BACKGROUND-COLOR: #666699; COLOR: white; FONT: bold 8pt verdana"
onmouseout="window.event.srcElement.style.background = '#666699'"
```

When loaded onto the server, this is how the "Returnsignature" page will appear:

Return Signature For Viewing

Enter your existing ID here to return the stored signature:

[   ]

Submit

Here, simply enter the ID number that was given by the "Update" page and click "Submit". The ID number is then posted to the "Retrievedsignature.asp" page for lookup in the database. If you specify an ID number that hasn't been assigned to a record by WebSign, then an error page will be generated.

```
38

ADODB.Field error '80020009'

Either BOF or EOF is True, or the current record has been deleted. Requested operation requires a current record.

/websignsimple/retrievedsignature.asp, line 47
```

If the record is valid and assigned to a record, then the "Retrievedsignature.asp" page will be loaded.

Retrievedsignature.asp

The "Retrievedsignature.asp" page receives the ID number and uses it to look up the corresponding record, retrieve the signature from the database, and send it back to the client for viewing. This page contains ASP code that is virtually the same as what can be found on the "Update" page (see above section on "Update.asp") that displays the signature information.

```
<%
Dim valID, strName, strDTStamp, sigStr, dateTimeS
valID = Request.Form("IDField")

Dim cnSignature
Dim rsSignature

Set cnSignature = CreateObject( "ADODB.Connection" )
Set rsSignature = CreateObject( "ADODB.Recordset" )

With cnSignature
    .Provider = "Microsoft.Jet.OLEDB.4.0"
    .ConnectionString = "Data Source =" & Server.MapPath ("/websignsimple/webSign.mdb")
    .Open
End With

rsSignature.CursorType = 1
rsSignature.LockType = 3

Response.Write(valID)
rsSignature.Open "SELECT * FROM Contracts WHERE ID=" & valID, cnSignature, 1, 3

Session("strName") = rsSignature( "Name" )
Session("sigStr") = rsSignature( "Signature" )
Session("dateTimeS") = rsSignature( "DateTime" )
%>
```

The major differences can be seen above, as the "Retrievedsignature.asp" page only retrieves certain fields form the database to display.

When the page is loaded it should look something like this:

The retrieved signature shows the signature, time stamp, and ID number. This is the retrieved signature from the previous examples.