



## Topaz Systems, Inc. Info Path Demo How-To

Topaz Systems, Inc.  
650 Cochran Street, Unit 6  
Simi Valley, CA, 93065

www.topazsystems.com  
tech support: 805 520-8286  
support@topazsystems.com

©1995-2006, all rights reserved, US patent 6,307,955, and pending

**THIS GUIDE REFERS ONLY TO THE TopazSignImageDemo.xsn EXAMPLE**

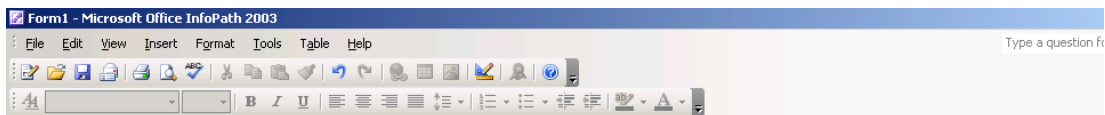
Welcome to the Topaz InfoPath Image Demo. This demonstration captures a signature and saves it to file as an image (.jpeg). This image is then imported into the InfoPath document. Because this demo will not create a biometric or an encrypted signature but instead creates a .jpeg image, the document is not legally binding. This is a demo only and should be used as a blueprint for creating your own signed InfoPath documents, and it is not intended necessarily as a “plugin” for InfoPath.

Begin by opening “TopazSignImageDemo.xsn” in Microsoft InfoPath. You will see this screen below.



Press the Sign button, and sign the writing surface of your Topaz tablet. Press Done to view your signature. Please note that no signing window will open upon pushing the Sign button, and no signature will appear until pushing the Done button. However, if you have an LCD pad, you should see the signature on the LCD as you are signing.

Your signature will appear on the InfoPath document after you have clicked “Done”.



#### SigPlus InfoPath Signature Image demo

To use:

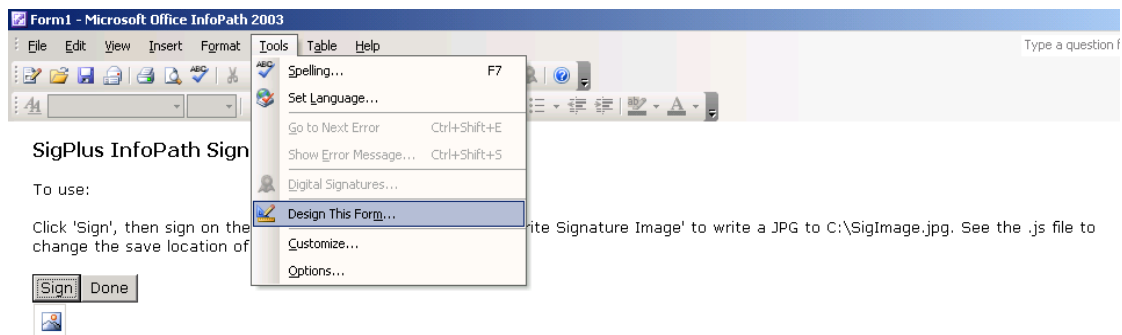
Click 'Sign', then sign on the signature pad. Once done, click 'Write Signature Image' to write a JPG to C:\SigImage.jpg. See the .js file to change the save location of the image file.

Sign Done

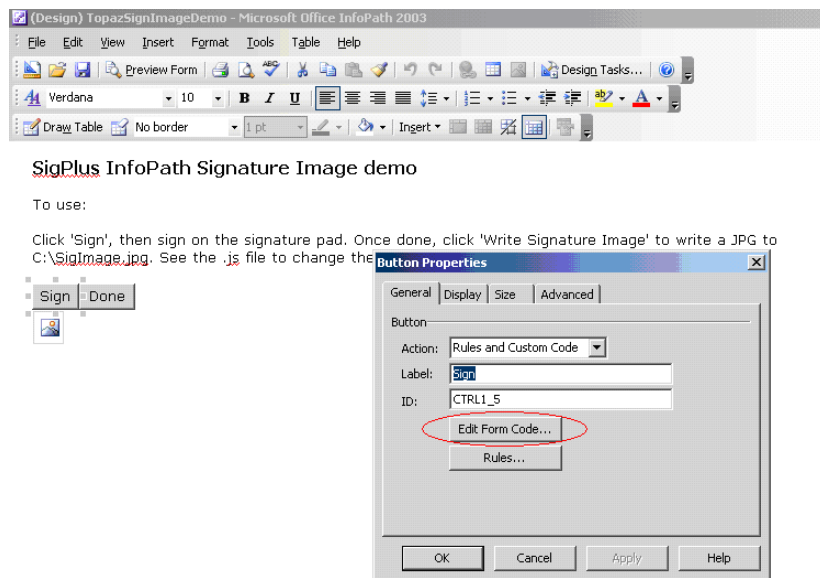
ED Smith

### Looking at the Code Behind the Demo

Next we will look at the JavaScript code behind this demo. To access the code, first go to (Tools>Design This Form).



In order to view the code behind each button, double click on the desired button. You will then see a button properties window as shown on the next page. Click on “Edit Form Code” in order to view the java script.



You will then see the code displayed. Shown below is what you will see.

```
// The following line is created by Microsoft Office InfoPath to define the prefixes
// for all the known namespaces in the main XML data file.
// Any modification to the form files made outside of InfoPath
// will not be automatically updated.
//<namespacesDefinition>
XDocument.DOM.setProperty("SelectionNamespaces", 'xmlns:my="http://schemas.microsoft.com/o:
var sigObj = new ActiveXObject("SIGPLUS.SigPlusCtrl.1");
//</namespacesDefinition>

//=====
// The following function handler is created by Microsoft Office InfoPath.
// Do not modify the name of the function, or the name and number of arguments.
//=====
function CTRL1_5::OnClick(eventObj)
{
    sigObj.InitSigPlus();
    sigObj.OpaqueMode=true;
    sigObj.TabletState=1;
}

//=====
// The following function handler is created by Microsoft Office InfoPath.
// Do not modify the name of the function, or the name and number of arguments.
//=====
function CTRL2_5::OnClick(eventObj)
{
    sigObj.TabletState=0;
    sigObj.ImageFileFormat=4;
    sigObj.ImageXSize=400;
    sigObj.ImageYSize=100;
    sigObj.ImagePenWidth=11;
    sigObj.JustifyMode=5;
    sigObj.WriteImageFile("C:\\SigImage.JPG");
    var objField1 = XDocument.DOM.selectSingleNode("//my:myFields/my:field1");
    objField1.text = "C:\\sigimage.jpg";
}
```

The next page will take you step by step through the code.

The InfoPath form contains an image object called my:field1, a Sign button called CTRL1\_5, and a Done button called CTRL2\_5.

In the <namespacedefinition> tag, we create a SigPlus object variable called sigobj using the activexobject() function call. In the onclick event for the Sign button (CTRL1\_5) we initialize the SigPlus object variable, make sure the opaque mode is true, and open the port so the signature can be captured.

(SEE BELOW)

```
function CTRL1_5::OnClick(eventObj)
{
    sigObj.InitSigPlus();
    sigObj.OpaqueMode=true;
    sigObj.TabletState=1;
}
```

The Onclick event for the Done button (CTRL2\_5) turns the pad off, sets the parameters to size the signature image, and writes the image to a file location ("C:\\SigImage.JPG"). It then creates a variable called objfield1 and sets it to the image object called my:field1. Finally we use the objfield1 variable to set the text property to the location of the .jpeg file, ("C:\\SigImage.JPG") which will automatically import the saved image into the my:field1 image object.

(SEE BELOW)

```
function CTRL2_5::OnClick(eventObj)
{
    sigObj.TabletState=0;
    sigObj.ImageFileFormat=4;
    sigObj.ImageXSize=400;
    sigObj.ImageYSize=100;
    sigObj.ImagePenWidth=11;
    sigObj.JustifyMode=5;
    sigObj.WriteImageFile("C:\\SigImage.JPG");
    var objField1 = XDocument.DOM.selectSingleNode("//my:myFields/my:field1");
    objField1.text = "C:\\sigimage.jpg";
}
```

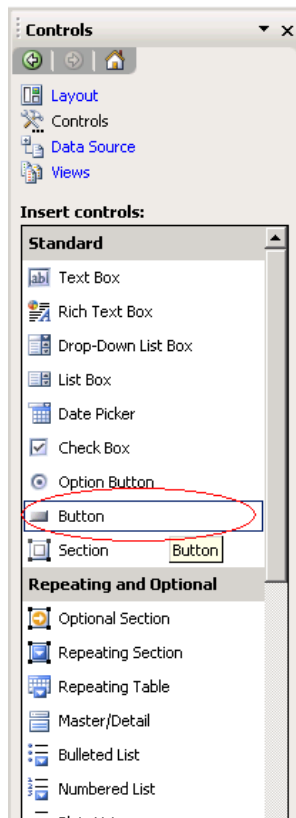
Now the signature image shows up in the InfoPath document. Using this demo the InfoPath developer can add as many image objects and signature images into the InfoPath document as they wish. Make sure to properly keep track of the image object names and saved .jpeg files so that your signatures show up in the locations you have specified.

## How to Programmatically Add Additional Signatures to your InfoPath Document

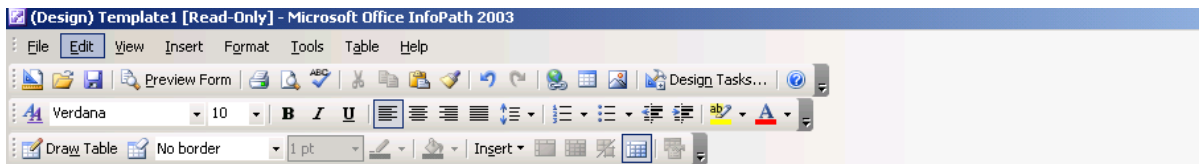
Start by placing the cursor on the line below the image object. Then choose (insert>more controls) from the menu at the top.



Of the controls that are now at your disposal, choose Button from the control list.



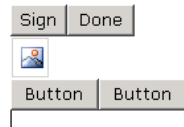
Insert the two new buttons. After inserting buttons place cursor below the first button as shown.



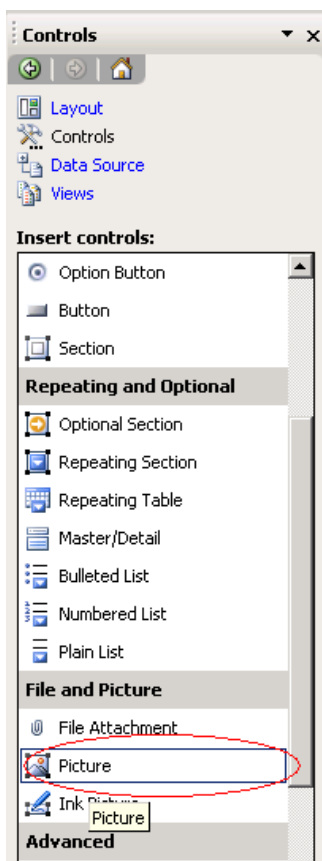
### SigPlus InfoPath Signature Image demo

To use:

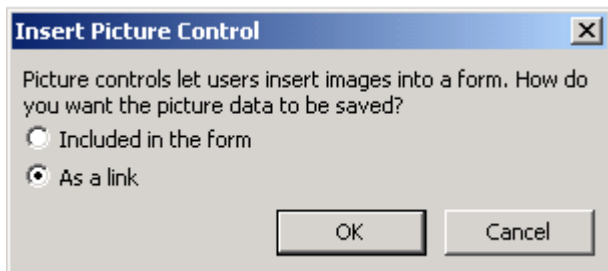
Click 'Sign', then sign on the signature pad. Once done, click 'Write Signature Image' to write a JPG to C:\SigImage.jpg. See the .js file to change the save location of the image file.



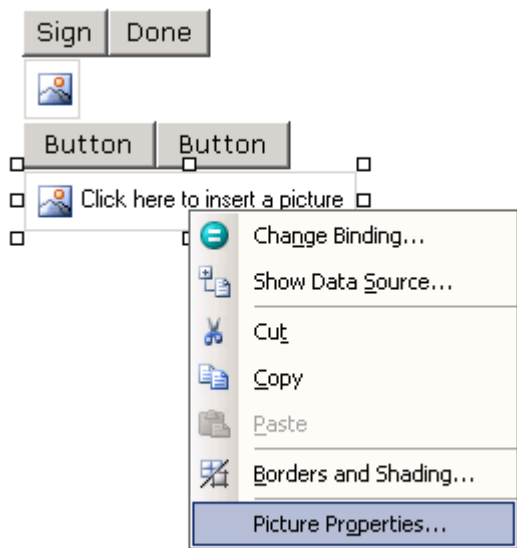
Choose Picture from the File and Picture section of the Controls window on the right.



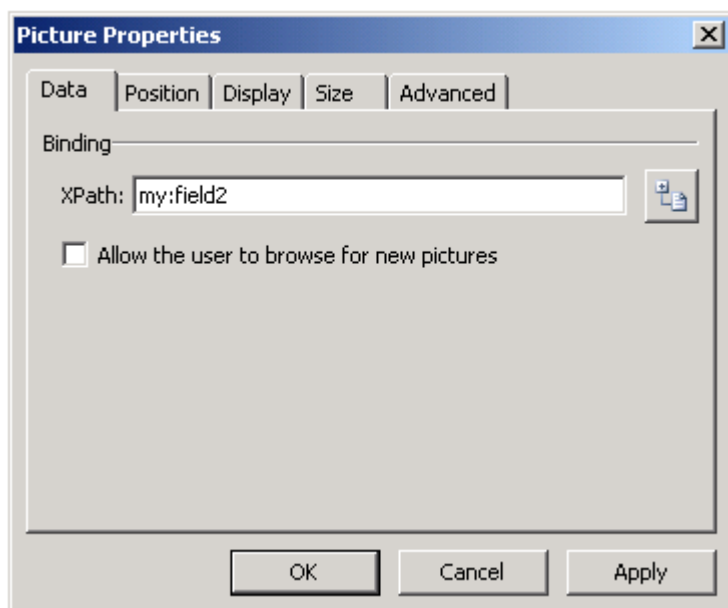
The Insert Picture Control window will now pop up, make sure to choose "As a link", then click "OK"



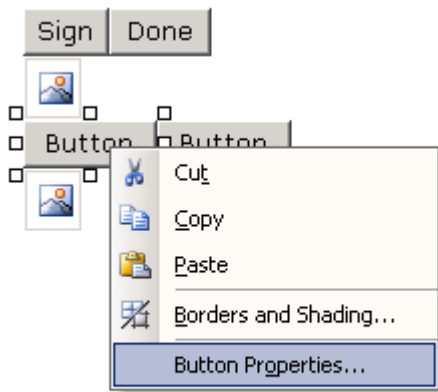
Right-click on your new Picture object, and choose the Picture Properties option.



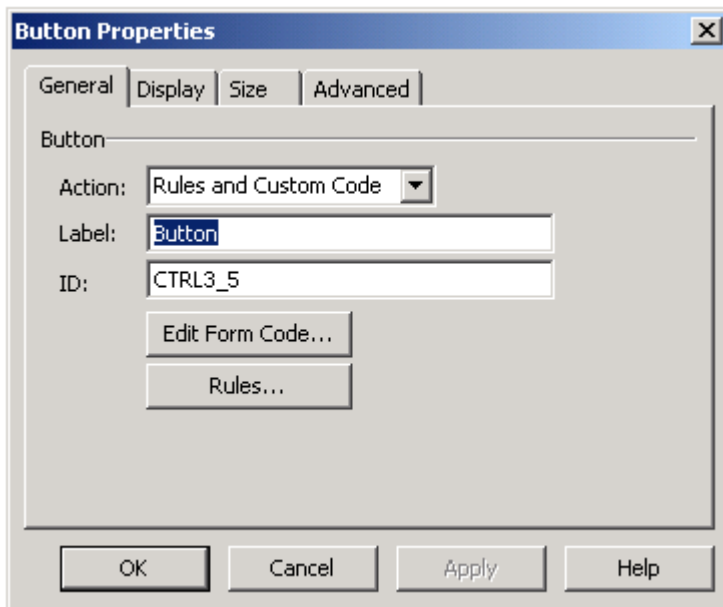
In the Picture Properties window, uncheck the "Allow the user to browse for new pictures" option, and also make sure the XPath is "my:field2". If not, change it to read my:field2. Next, click OK.



Next, right-click on your first new button, and click on the "Button Properties..." option.



Make sure in the Button Properties window that the ID is CTRL3\_5. If you wish to change the button name change the label property at this point. This button will serve to allow the second signature to be captured. For example you may call this button "Sign2".



Right-click also on the second new button, click on "Button Properties" there, and make sure that the ID for this button is CTRL4\_5. If you wish to change the button name change the label property at this point. This button will serve to bring the second signature image into the InfoPath document. You may want to call this button something like "Done2".

Double-click your first new button, and choose the "Edit Form Code" option in the Button Properties window. This will bring up the CTRL3\_5::OnClick event. Add the following code:

```
sigObj.InitSigPlus();  
sigObj.ClearTablet();  
sigObj.OpaqueMode=true;  
sigObj.TabletState=1;
```

(SHOWN ON NEXT PAGE)



```

//=====
// The following function handler is created by Microsoft Office InfoPath.
// Do not modify the name of the function, or the name and number of arguments.
//=====
function CTRL1_5::OnClick(eventObj)
{
    sigObj.InitSigPlus();
    sigObj.OpaqueMode=true;
    sigObj.TabletState=1;
}

```

Double click the second new button, and choose the "Edit Form Code" option in the Button Properties window. This will bring up the CTRL4\_5::OnClick event. Add the following code:

```

sigObj.TabletState=0;
sigObj.ImageFileFormat=4;
sigObj.ImageXSize=400;
sigObj.ImageYSize=100;
sigObj.ImagePenWidth=11;
sigObj.JustifyMode=5;
sigObj.WriteImageFile("C:\\SigImage2.JPG");
var objField1 = XDocument.DOM.selectSingleNode("//my:myFields/my:field2");
objField1.text = "C:\\sigimage2.jpg";

```

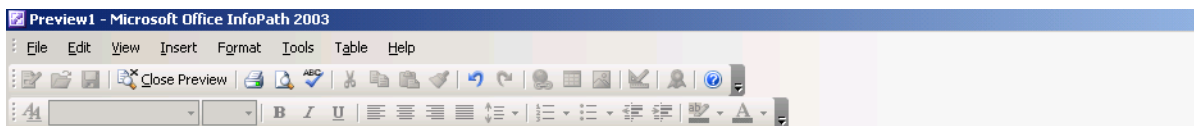
(SHOWN BELOW)

```

//=====
// The following function handler is created by Microsoft Office InfoPath.
// Do not modify the name of the function, or the name and number of arguments.
//=====
function CTRL4_5::OnClick(eventObj)
{
    // Write your code here
    sigObj.TabletState=0;
    sigObj.ImageFileFormat=4;
    sigObj.ImageXSize=400;
    sigObj.ImageYSize=100;
    sigObj.ImagePenWidth=11;
    sigObj.JustifyMode=5;
    sigObj.WriteImageFile("C:\\SigImage2.JPG");
    var objField1 = XDocument.DOM.selectSingleNode("//my:myFields/my:field2");
    objField1.text = "C:\\sigimage2.jpg";
}

```

As mentioned before, you can right-click on your buttons and change the text, of course, to "Sign2" and "Done2", or what you like. Essentially, the new buttons allow you to add a second signature to the InfoPath document. Shown on the next page is how the demo will appear with the second signature added. Now that you know how to add signatures to your InfoPath document, you can add as many as you like.



### SigPlus InfoPath Signature Image demo

To use:

Click 'Sign', then sign on the signature pad. Once done, click 'Write Signature Image' to write a JPG to C:\SigImage.jpg. See the .js file to of the image file.



For more information regarding products and support

Email: [support@topazsystems.com](mailto:support@topazsystems.com)

Phone: (805) 520-8286