

# **A GUIDE TO USING THE XETEX XSIGNER ACTIVEX CONTROL**

Version 1.3

May 8<sup>th</sup>, 2001

Daniel J. Sanders (Xetex, Inc.)  
dsanders@xetex.com

© 1999-2001 Xetex, Inc. All Rights Reserved.  
Unpublished rights reserved under the copyright laws of the United States.

Xetex, Inc. ("Xetex") makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Xetex reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation to notify any person of such revision or changes.

## Table Of Contents

<b>ACTIVEX CONTROL OVERVIEW</b> .....	<b>3</b>
IN GENERAL .....	3
<b>XETEX XSIGNER™ OVERVIEW</b> .....	<b>3</b>
<b>SUPPORTED PLATFORMS</b> .....	<b>3</b>
OPERATING SYSTEMS .....	3
BROWSERS .....	3
<b>XSIGNER FUNCTIONALITY</b> .....	<b>4</b>
XSIGNER PROPERTIES .....	4
XSIGNER ACTIVATION: VBSCRIPT & HTML .....	4
<b>SAMPLE ACTIVATION SCRIPT</b> .....	<b>5</b>
<b>XSIGNER USER INTERFACE</b> .....	<b>6</b>
XSIGNER PROCESS FLOW .....	6
Authenticode prompt .....	6
Certificate Selection .....	7
MS Crypto Signing .....	8
XSigner output .....	8

## ACTIVEX CONTROL OVERVIEW

### In General

ActiveX controls are components (or objects) you can insert into a web page or other application to deliver a program to the end-user, for execution on the end-user's computer. ActiveX refers to a loosely defined set of COM-based technologies. For the purposes of this application, ActiveX controls are like Windows applications that run inside a container. ActiveX controls may be written in any language but the control must be compiled (prior to deployment) into native machine language by a compiler that supports COM in order to communicate with the ActiveX Windows framework properly. ActiveX controls are typically written in Visual Basic, C or C++.

## XETEX XSIGNER™ OVERVIEW

XSigner is an ActiveX Control written in C++. It contains code that uses the Microsoft Crypto Application Programming Interface (MS CAPI) to sign text strings and to output the signed result in the format of a base64-encoded string. The control also provides a user interface in the form of dialog boxes that allows the user to:

- view the contents of the MSIE certificate store
- choose a certificate
- view the text to be signed
- execute a signing operation

## SUPPORTED PLATFORMS

### Operating Systems

XSigner uses the Microsoft Crypto API. MS CAPI is included as a built-in standard feature of Windows 98, Windows NT & Windows 2000. It is not included in Windows 95 but is available as an add-on, automatically installed when downloading & installing Internet Explorer 4 or later.

XSigner has been tested on the following platforms:

- Windows NT 4.0
- Windows 2000
- Windows 98
- Windows 95 with IE 4.0 or higher
- Windows Millennium

### Browsers

XSigner also makes use of Microsoft Authenticode technology. Authenticode allows a user downloading an ActiveX control (or any other piece of software) to examine a certificate presented by the software provider, in order to prevent the risk of accepting software from unrecognized vendors. The certificate, issued by Microsoft and Thawte/Verisign, certifies the binding between the software and its signer thereby providing assurance that the software was written and is being provided by an authenticated software vendor. Upon viewing the certificate the user may accept or decline the download. Authenticode is built into Internet Explorer 4.0 and later. Thus, older versions of Internet Explorer do not support ActiveX controls.

XSigner has been tested on the platforms noted in the preceding section using the following browsers:

- Microsoft Internet Explorer 4.0
- Microsoft Internet Explorer 5.0
- Microsoft Internet Explorer 5.5

## XSIGNER FUNCTIONALITY

XSigner is activated as follows: the client browser is directed to an HTML page containing a VBScript (Visual Basic Script). The VBScript sets the 'StringToSign' variable on the control, calls the 'Sign' method on the control and receives the output via another of the control's variables: 'SignedString'. The output is then posted back to the server via an HTML form.

## XSIGNER PROPERTIES

The table below details the properties and variables necessary to invoke and activate XSigner. The ActiveX control's relevant variables and properties are as follows:

Variable Name	Property/ Variable Type	Variable Purpose	Variable Value (before signing)	Variable Value (after signing)
ID	HTML Object tag attribute	Identify the control to be used by VBScript	"XSigner"	Unchanged
Class ID	HTML Object tag attribute	Identify the type of control to be loaded.	"CLSID:63849D57-928A-492B-A4E0-8E76F70F3E54"	Unchanged
Codebase	HTML Object tag attribute	Declare the server-side location of the code to be downloaded.	"XSigner.cab "	Unchanged
StringToSign	ActiveX input variable string.	Assign this variable with the string to be signed before calling the Sign method.	Value assigned by VBScript or HTML.	Unchanged
Sign	ActiveX method	Activates the ActiveX control's core functionality – digital signing.	Not Applicable (method)	
SignedString	ActiveX output variable string.	The ActiveX control places the value of the signed output into this variable for retrieval by the VBScript	null	<ul style="list-style-type: none"> <li>• A base64 encoded string containing the output of the signing operation in PKCS7 format.</li> <li>• "Empty" if the cert store was empty</li> <li>• "User Cancellation" if the user cancelled</li> </ul>

## XSIGNER ACTIVATION: VBSCRIPT & HTML

1. The ActiveX control is loaded by the browser upon seeing the following html tag:

```
<OBJECT ID="XSigner" codebase="XSigner.cab" CLASSID="CLSID:63849D57-928A-492B-A4E0-8E76F70F3E54"></OBJECT>
```

2. Once the control is loaded, the VBScript activates by setting the input variable:  
**XSigner.StringToSign="Do you wish to sign this string?"**
  3. The VBScript then calls the Signing method as follows:  
**XSigner.Sign**
  4. Upon completion of the signing operation, the VBScript execution will retrieve the output from the signing operation with the following:  
**theForm.signed.value=XSigner.SignedString**
- (where "theForm" represents the html form containing an input of type "Hidden" named "signed").

### SAMPLE ACTIVATION SCRIPT

The following script is an example of how to activate the XSigner with a VBScript in an html page. **Note that the script handles three potential outcomes:** (1) a successful signing operation (2) user cancellation (3) an empty certificate store. If there is another problem, the Windows error handling will take over and kill the signing process.

```
<HTML>
<HEAD>
<TITLE>Xetex ActiveX Signing Control</TITLE>
</HEAD>
<BODY>
<OBJECT ID="XSigner" codebase="XSigner.cab" CLASSID="CLSID:63849D57-928A-492B-
A4E0-8E76F70F3E54"></OBJECT>

<FORM name=signingForm action= http://server1.xetex.com:8080/servlet/
method=post>
<INPUT type=hidden value="Whatever" name=text>
<INPUT type=hidden name=signed>

<SCRIPT language=VBScript>
dim theForm
set theForm = Document.forms("signingForm")
XSigner.StringToSign="This is the string to be signed"
XSigner.Sign
theForm.signed.value = XSigner.SignedString

if (theForm.signed.value = "Empty") Then
Document.write("You have no certificates in the browser cert store")

elseif (theForm.signed.value = "User Cancellation") Then
Document.write("<b>An Error Occurred</b>")
else
Document.write("<b>The signing operation completed successfully. The output
was:</b><br>")
wholeString=theForm.signed.value
Document.write(wholeString)
End If
</SCRIPT>
</FORM>
</BODY>
</HTML>
```

## XSIGNER USER INTERFACE

The XSigner uses MFC (Microsoft Foundation Class) to create windows for a user dialog and HTML to format the pages hosting XSigner.

The steps are as follows:

1. User reaches the start page and sees Authenticode Prompt (figures 1a & 1b)
2. Upon Acceptance by user, page continues loading and the VBScript activates XSigner. The certificate selection window (figure 2) is displayed. User chooses a certificate and presses OK.
3. User sees the MS CAPI dialog window (Figure 3) and authorizes the signing operation by pressing OK.
4. Page finishes loading (with or without output displayed).



**Figure 1a:** XSigner start page opening view.

### XSigner Process Flow

#### **Authenticode prompt**

The user's browser is directed to the starting point for digital signing, e.g. an HTML page containing a VBScript and a contract to be signed. The VBScript immediately activates the XSigner and the user sees the Authenticode prompt shown in figures 1a and 1b.

(See next page for a detail of the Authenticode prompt).

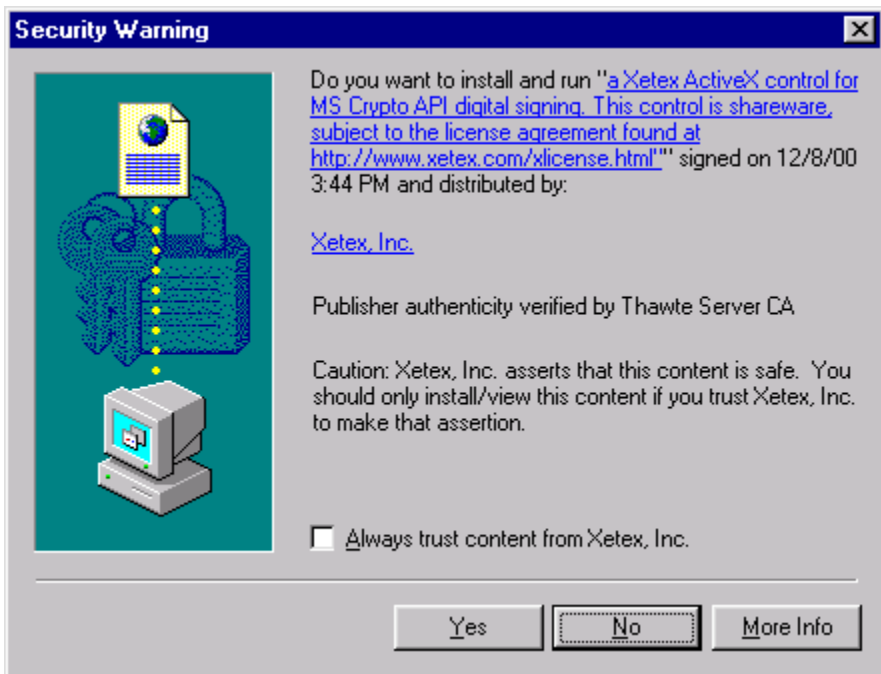


Figure 1.b: The Authenticode prompt.

This dialog (Figure 1b) provides the user with the following assurances regarding the piece of software that they are about to download:

Xetex Inc. created this software.

The content has not been altered since Xetex Inc signed it. If the code had been altered, post-signing, the Authenticode dialog would warn of an inconsistency.

Xetex Inc. asserts that the content is safe for the user's computer.

### Certificate Selection

After accepting the Authenticode prompt, the user sees a dialog box from which he is requested to read the text to be signed, choose a certificate, and press OK. All the user's current certificates are presented for viewing. If the user has a number of certificates, and wishes to sign with a specific certificate, he must choose that certificate from the pull-down window.



Figure 2: the certificate selection dialog.

## MS Crypto Signing

After choosing the correct certificate and pressing OK, XSigner executes the MS CAPI signing operation, which encrypts the text string with the private Key associated with the user certificate chosen. The security level listed on this dialog window will be the same one chosen by the user when the certificate was originally downloaded. Figure 3 shows the dialog window.

We recommend that the user set the security level to HIGH when downloading the certificate. The disadvantage of this is that it requires that the user choose and remember a password for the certificate. The advantage is, of course, increased security, as other users of the computer cannot use the certificate unless they have the password.



Figure 3: the MS CAPI signing dialog window.

## XSigner output

You can use VBScript to retrieve the output from XSigner and format it in any way you like. Our demo simply takes the output (XSigner.SignedString) and writes it out to the HTML page:

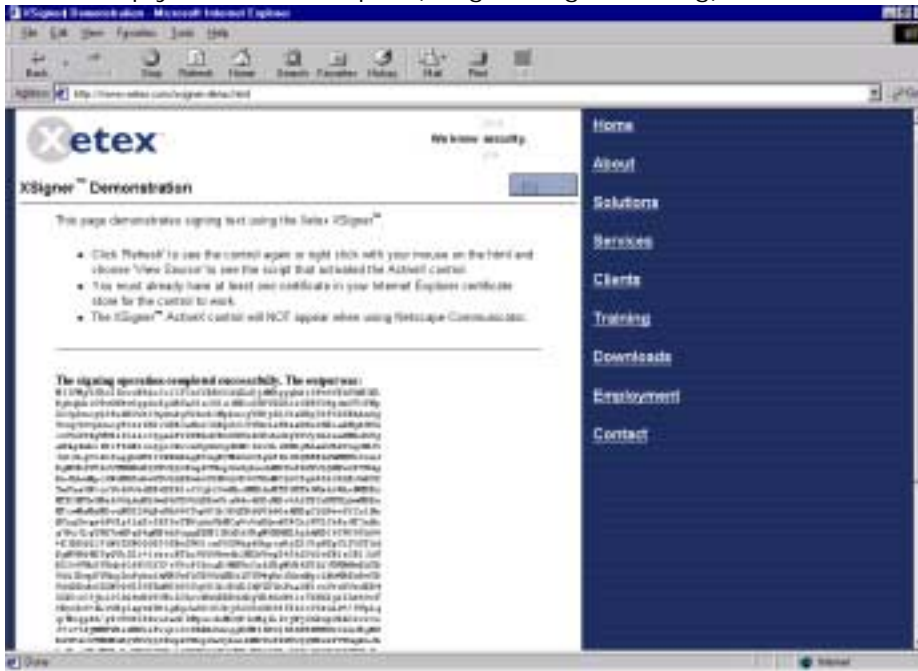


Figure 4: XSigner output written to html page.