

http://msdn.microsoft.com/workshop/imedia/windowsmedia/ad.asp Go SEP OCT NOV 13 1998 1999 2000 About this capture

30 captures
13 Oct 1999 – 26 Oct 2008

msdn online
Web Workshop

All Products | Support | Search | microsoft.com Home

Microsoft

HOME | VOICES | LIBRARIES | COMMUNITY | DOWNLOADS | SITE GUIDE | SEARCH MSDN

 [show toc](#)  [index](#)

[Web Workshop](#) | [Streaming & Interactive Media](#) | [Windows Media Technologies](#)

Ad Insertion

December 14, 1998

- [Introduction](#)
- [Alphabet Soup Decoded](#)
- [Example One: Simple Ad Insertion](#)
- [Example Two: Simple ASX file with enhanced elements](#)
- [Example Three: Dynamic ASX file via ASP](#)
- [Summary](#)

Introduction

Streaming media is an example of compelling content on the Internet. **Adding advertisements using streaming media** is a great way to communicate commercial messages. In this article, we'll learn how to insert ads into a play list, and expand the idea to utilize dynamic ad personalization by using Active Server Page technology (ASP).

Alphabet Soup Decoded

There are many file formats and acronyms you must become familiar with for **Ad Insertion**:

- Microsoft's native file format for streaming media is known as Advanced Streaming Format ([ASF](#)). This low-overhead storage and transmission file format encapsulates multimedia data types (images, audio, and video) as well as embedded text (URLs, for example), and allows for the synchronization of these objects within a stream.
Individual ASF clips can be combined into *play lists* by creating [ASX files](#). These files use industry-standard Extensible Markup Language (XML) syntax. The ASX file is linked from the ASF file and basically describes each clip.
- Active Server Pages (ASP), are dynamic Web pages used in conjunction with Microsoft IIS (Internet Information Server). ASP is an environment in which you can combine HTML, scripts, and reusable ActiveX server components to create dynamic and powerful Web-based business solutions. ASP's enable server-side scripting for IIS with native support for both VBScript and Jscript.

Now, see the examples below. You are well on your way to adding ads.

[Back to top](#)

Example One: Simple Ad Insertion

In our first example, we will insert an ad in front of a single video clip. All visitors will see the same ad, and the same video clip.

```
<ASX Version="3">

  <ENTRY ClientSkip="no">
    <REF HREF="mms://127.0.0.1/AdInsertion/HollandAmerica200.asf"/>
  </ENTRY>

  <ENTRY>
    <REF HREF="mms://127.0.0.1/AdInsertion/NukeTest200.asf"/>
  </ENTRY>

</ASX>
```

The ASX elements are simple. This code defines the file as an ASX file, and tells Windows Media Player what version you're using.

```
<ASX Version="3">
```

Then we define two entries. Note that the first entry (the advertisement) has a special descriptor, `ClientSkip="no"`, that prevents the site visitor from skipping the ad. In order to enforce this behavior, the player disables three controls: [Next Clip](#), [Preview Mode](#), and the [slider bar](#).

After viewing the ad, the site visitor will immediately see the second entry with normal player behavior restored. How this works is the Windows Media Player starts buffering the second clip while the advertisement is still playing, which eliminates the traditional "black hole" between clips. This behavior is unique to Microsoft's media format, and is a feature customers appreciate.

Finally, the `</ASX>` tag marks the end of this file.

[↩ Back to top](#)

Example Two: Simple ASX File with Enhanced Elements

Example One showed the simplest advertisement scenario. Example Two will expand on this by adding [titles](#), [abstracts](#), and a ["Buy Now"](#) button. The Buy Now button can be used either to allow users to purchase content, or simply to send them to the advertiser's Web site. Examine the following code:

```
<ASX version="3">

<TITLE>Demo ASX File</Title>

<ENTRY ClientSkip="no">
  <REF HREF="mms://Raratonga/AdInsertion/HollandAmerica200.asf"/>
  <banner HREF="http://Raratonga/AdInsertion/images/MoreInfo.gif">
    <moreinfo HREF="http://www.hollandamerica.com"
  </BANNER>
  <TITLE>Holland America Travel</TITLE>
  <COPYRIGHT> 1998, Holland America </COPYRIGHT>
</ENTRY>

<ENTRY>
  <TITLE>Secret US Nuke Tests</TITLE>
```

```

<COPYRIGHT> Copyright MSNBC 6/11/98 </COPYRIGHT>
<REF HREF="mms://Raratonga/AdInsertion/NukeTest200.asf"/>
<ABSTRACT>
::Recently declassified footage shows that the United States conducted secret nuclear
weapons testing in the upper stratosphere just before the nuclear
weapons ban went into
effect. Tests were not always successful, with at least one rocket
exploding on the launch
pad.
<ABSTRACT>
<COPYRIGHT> Copyright 1998, MSNBC -- All Rights Reserved </COPYRIGHT>
lt;ENTRY>

</ASX>

```

Let's look at each of the new elements. The <TITLE> element describes a show title, which appears in the standalone player (see Figure 1 below). All of the elements described can also be used in conjunction with the embedded player, and rendered via JavaScript.


The <BANNER> tag points to a GIF for the button image which points to a URL. Notice how that target is used to point to a frame name, just like you would in a normal HTML page. And, like a normal HTML page, in-line JavaScript can be used in place of a URL.

The <ABSTRACT> tag describes Tool Tip text that appears when the user's mouse hovers over the clip title. <COPYRIGHT>, <TITLE>, and <AUTHOR> tags describe various display elements in the player also.

[↩ Back to top](#)

Example Three: Dynamic ASX Files via ASP

The most compelling advertisements are personalized to user preferences. Users interested in travel on cruise ships won't see backpacking ads, and vice-versa. This avoids annoying viewers, and at the same time commands premium rates from advertisers because they know only interested viewers will see their ad.

[Microsoft AdServer](#) , a component of Microsoft SiteServer, is one way to schedule these ads, but your **personalization can be as simple or as robust as you find necessary.**

Example Three will use a simple radio button selection that chooses one of two ads. This, in turn, will drive dynamic content and will focus on how ASP gets generated.

Example

There are three required steps to creating a dynamic play list:

1. **Ask the user for personalized information.** We use a simple form with two choices in our example below.
2. **Process the user preferences to create a personalized play list.** We use ASP to create the list and the Windows Media Player as an embedded object. The first step calls the page up containing the player reference. That page then calls a second ASP page up that returns the ASX file.
3. **Play the user's content.** Presto!

The user preference form (entitled *preferences.htm*) is shown below. The form method is set to **post** in order to make the URL cleaner for *PlayerPage.asp*.

<HTML>

<! Preferences.htm>

<BODY>

Please choose one of the following:

<FORM METHOD=POST ACTION=PlayerPage.asp>

<INPUT TYPE=RADIO NAME=AdPref VALUE=0> I like travel

<INPUT TYPE=RADIO NAME=AdPref VALUE=1> I like backpacking

<P>

<INPUT TYPE=SUBMIT>

</FORM>

</BODY>

</HTML>

PlayerPage.asp embeds the player, then passes user choices on to *BuildAsx.asp*.

<%

Option Explicit

```
` This function will process as many arguments as are passed in
` the Request.form object. In this example the function is overkill,
` but is listed here as cookbook code for your use
```

Function UrlArgs()

Dim item, sTemp

For each item in Request.Form

` Keep concatenating data

sTemp = item & "=" & Request.Form (item) & "&"

Next

If Len(sTemp) > 0 Then

` Trim the trailing &, if there is data in sTemp

sTemp = Left (sTemp, Len (sTemp) - 1)

End If

UrlArgs = sTemp

End Function

%>

<html>

<body>

<OBJECT

ID=MediaPlayer1

Name="MediaPlayer1"

CLASSID="CLSID:22D6F312-B0F6-11D0-94AB-0080C74C7E95"

```
codebase="http://activex.microsoft.com/activex/controls/mpplayer/en/nsmp
2inf.cab#Version=5, 1, 52, 701"
```

HEIGHT=412

WIDTH=326

>

<PARAM NAME="FILENAME"

VALUE="BuildAsx.asp?<%=UrlArgs() %>">

<PARAM NAME="AutoStart" VALUE="-1">

<PARAM NAME="ShowControls" VALUE="-1">

<PARAM NAME="ShowTracker" VALUE="-1">

<PARAM NAME="Autosize" VALUE="-1">

<PARAM NAME="ShowStatusBar" VALUE="-1">

<PARAM NAME="ShowDisplay" VALUE="-1">

<EMBED TYPE="application/x-mplayer2"

pluginspage="http://www.microsoft.com/windows/mediaplayer/download/default.asp"

```

        id=MediaPlayer1
        Name="MediaPlayer1"
        DisplaySize="4" //Fit To Size
        AutoSize="-1"
        BgColor="darkblue"
        ShowControls="-1"
        ShowTracker="-1"
        ShowDisplay="-1"
        ShowStatusBar="-1"
        VideoBorder3D="-1"
        WIDTH=320
        Height=313
        SRC="BuildAsx.asp?<%=UrlArgs() %>"
        autostart="-1"
        DESIGNTIMESP="5311"
    >
</EMBED>
</OBJECT>

</body>
</html>

```

[⬆ Back to top](#)

The real magic is performed in *BuildAsx.asp*. This page does not send the usual HTTP headers to the player. That is why the page works as an ASX file.

```

<%@ LANGUAGE="VBScript" %>
<%
-----
` Set the BASE_PATH Const below to point to your
` Windows Media Services server
-----
Option Explicit
Const BASE_PATH = "mms://media.MyDomain.com/AdInsertion/"

`Note that Request.ContentType is the "secret handshake"
`that allows us to create dynamic
`ASX files that do not contain HTTP header data.
Response.ContentType = "video/s-ms-asf" ` Set MIME type
Response.Expires = 0 `Force content to always refresh

Response.Write AsxHeader() `Create the opening ASX tag
Response.Write AdEntry() `Create an ad
Response.Write AsxEntry() `Create an XML entry for content
Response.Write AsxFooter() `Write the ASX footer
`And we're done!
-----
` Function AsxHeader
` This function returns an opening <ASX> tag
-----
Function AsxHeader ()
    Dim sTemp
    sTemp = "<ASX Version=""3"">" & vbNewLine & " " & vbNewLine
    sTemp = sTemp & "<TITLE> Demo Dynamic ASX File From Active " _
        & "Server Pages </Title>" & vbNewLine & " " & vbNewLine
    ASXHeader = sTemp `return the string
End Function
-----
` Function AdEntry
` This function returns an ad based on user request
-----
Function AdEntry()
    Dim sTemp
    sTemp = "<ENTRY>" & vbNewline

```

```

If cInt (0 & Request ("AdPref")) = 1 Then
    sTemp = sTemp & "          <REF HREF="" & BASE_PATH_
        & "BackpackingAd.asf"" />" & vbNewLine
Else
    sTemp = sTemp & "          <REF HREF="" & BASE_PATH_
        & "TravelAd.asf"" />" & vbNewLine
End If
sTemp = sTemp & "</entry>" & vbNewLine & vbNewLine
AdEntry = sTemp
End Function
\-----
\ Function AsxEntry
\ This function returns an XML entry for a specific story.
\ In real life this function will probably do a database lookup
\-----
Function AsxEntry ()
    Dim sTemp, sAbstract
    sTemp = "<ENTRY>" & vbNewLine
    sTemp = sTemp & "          <TITLE> Secret US Nuke Tests </TITLE>" & vbNewLine
    sTemp = sTemp & "          <REF HREF="" & BASE_PATH & "Nuketest200.asf""
/> & vbNewLine
    sTemp = sTemp & "          <ABSTRACT>" & vbNewLine
    sAbstract = "Recently declassified footage shows that the United
States conducted secret nuclear weapons testing in the upper
stratosphere just before the nuclear weapons ban went into effect.
Tests were not always successful, with at least one rocket exploding on
the launch pad."
    sTemp = sTemp & sAbstract & vbNewLine
    sTemp = sTemp & "          </ABSTRACT>" & vbNewLine
    sTemp = sTemp & "          <COPYRIGHT>Copyright 1998, MSNBC -- All Rights
Reserved </COPYRIGHT>" & vbNewLine
    sTemp = sTemp & "</ENTRY>" & vbNewLine & vbNewLine
    AsxEntry = sTemp
End Function
\-----
\ Function AsxFooter
\ This function returns a closing </ASX> tag
\-----
Function AsxFooter ()
    AsxFooter = "</ASX>" & vbNewLine & vbNewLine
End Function

%>

```

Summary

The examples above are great cookbook code samples to get you started. Either use a sample "as is," or build on them to create really dynamic Web pages.

Additional examples can be found at [Windows Media Technologies Technical Showcase](#). Click on *Code Samples* and use the "view source" feature to see how we did it. Also be certain that you download the [client SDK](#). Here, there are even more examples and ideas shown, including examples that work in both Netscape's and Microsoft's browsers.

[Back to top](#)

Did you find this material useful? Gripes? Compliments? Suggestions for other articles? [Write us!](#)

© [1999 Microsoft Corporation. All rights reserved. Terms of use.](#)