

Introduction

ASIO Control is an ActiveX control allowing access to multi-channel audio devices from applications like Matlab, Visual Studio, etc. With ASIO control, your application can record and/or playback multiple channels of audio in synchrony either to/from memory or hard disk.

Installation

After installing the control, it must be registered before use. To do this, open a command window, change to the directory containing the file ASIOControl.ocx and type

```
regsvr32 ASIOControl.ocx
```

On Vista this needs to be done in administrator mode. If successful, a small dialog box confirming registration will appear.

Use in Matlab

The simplest way to use the control in Matlab is in conjunction with the .m file, ASIOControl.m. Copy this file into your Matlab path and type:

```
hASIO = ASIOControl;
```

You should see a figure created as shown on the right. This figure is the holder for the control and should be left open for as long as you want to use ASIO. It will also log the activity of the control. The handle returned by ASIOControl will be needed to edit properties and make recordings later.

Communication with the control and ASIO devices is through the control's properties and its methods. To view the properties of the control, type:

```
get(hASIO)
```

This should give a result something like:

```
ASIODeviceName: ''
ASIODevices: {4x1 cell}
  numInputs: 0
  numOutputs: 0
  sampleRate: 0
  inputChannels: {0x1 cell}
  outputChannels: {0x1 cell}
channelsInputting: NaN
channelsOutputting: NaN
```

The first two properties are the important ones to begin with. ASIODeviceName is the name of the ASIO device that is currently selected for use. ASIODevices is a cell array listing all valid names (4 in my case). If this list is empty, you don't have any ASIO device drivers installed. You will need to get one for your sound card. If it didn't come with a driver, the ASIO4ALL driver is worth a try.



To see the list of devices, type:

```
hASIO.ASIODevices
```

which will give a result depending on the devices installed like:

```
ans =  
  
    'Adobe Default Windows Sound 1.5'  
    'ASIO DirectX Full Duplex Driver'  
    'ASIO Multimedia Driver'  
    'Creative ASIO'
```

To use a device, simply set ASIODeviceName either directly as a string or by referencing a member of ASIODevices. For example, on my machine, the following two commands achieve the same end:

```
hASIO.ASIODeviceName = 'Creative ASIO';  
or  
hASIO.ASIODeviceName = hASIO.ASIODevices{4};
```

An even more compact way of selecting a device is to specify the name when loading the control, e.g.:

```
hASIO = ASIOControl('Creative ASIO');
```

will load an ASIOControl and select the Creative soundcard for use. Having selected a device, the properties of the control will look something more like:

```
ASIODeviceName: 'Creative ASIO'  
ASIODevices: {4x1 cell}  
    numInputs: 22  
    numOutputs: 18  
    sampleRate: 48000  
    inputChannels: {22x1 cell}  
    outputChannels: {18x1 cell}  
channelsInputting: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
channelsOutputting: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

The full list of properties and their meanings is:

ASIODeviceName	(read/write) Currently selected ASIO Device
ASIODevices	(read only) List of all installed devices
numInputs	(read only) Number of input channels
numOutputs	(read only) Number of output channels
sampleRate	(read/write) Sample rate [Hz]
inputChannels	(read only) List of the names of all the input channels
outputChannels	(read only) List of the names of all the output channels
channelsInputting	(read/write) Vector indicating which channels to use (1 = used, 0 = not used)
channelsOutputting	(read/write) Vector indicating which channels to use (1 = used, 0 = not used)

These properties can all be accessed and, in some cases modified, programatically or using a GUI property editor. For example, to load the control, select a device and set the sample rate:

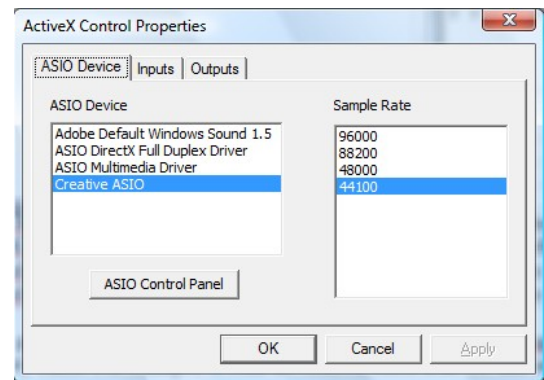
```
hASIO = ASIOControl('Creative ASIO');  
hASIO.ASIODeviceName = 'Creative ASIO';  
hASIO.sampleRate = 44100;
```

Alternatively, typing:

```
hASIO.propedit
```

will bring up the property editor window. The first page of the property editor shows the list of devices, highlighting the currently selected one, and also the list of available sample rates. The 'ASIO Control Panel' button accessed the control panel for the current device which can also be done by typing:

```
hASIO.controlPanel
```



The other two pages of the property editor show the available input and output channels and highlights those selected for recording or playback appropriately (as set by the `channelsInputting` and `channelsOutputting` properties).

To perform tasks (e.g. recording and playing sounds) the control's methods are used. The full list of methods is:

<code>controlPanel</code>	Opens the ASIO device control panel
<code>playFile(filename)</code>	Plays a multi-channel .wav file
<code>recordToArray(nSamples)</code>	Records nSamples of audio and returns an array of samples
<code>recordToArray(playArray)</code>	Plays an array of samples and records simultaneously
<code>recordToFile(nSamples, recordFile)</code>	Records nSamples of audio and stores in a .wav file
<code>recordToFile(playFile, recordFile)</code>	Plays a .wav file and records another simultaneously

An example of the use of some of the methods to record and playback sounds is shown below:

```
% ASIO Control example script
% Load control and select Creative driver
hASIO = ASIOControl('Creative ASIO');
% Set sample rate
hASIO.sampleRate = 48000;
% Select input channels 1, 2, 17 and 18
hASIO.channelsInputting = [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1];
% Record 1/2 second (24000 samples) from the four channels to an array
x = hASIO.recordToArray(24000);
% x will now be a four column 24000 row array of samples
% Now set up two output channels (1 and 2). Note if the vector passed to
% channelsOutputting is shorter than the number of channels, the control
% fills zeros automatically for the rest
hASIO.channelsOutputting = [1 1];
% Record another 1/2 second this time outputting the sound recorded
% previously on input channels 17 and 18. The length of recording is now
% determined by the number of columns in the input array
y = hASIO.recordToArray(x(:, 3:4));
```

Use in other applications

Being an ActiveX control, the ASIO control can be used in other applications. For example, in Visual Basic, ASIO recording can be easily implemented by adding the ASIO control to a form.