

# I-CUBE X

## How to control Flash with I-CubeX

### I-CubeX Reference

*I-CubeX: The ultimate MIDI controller!*

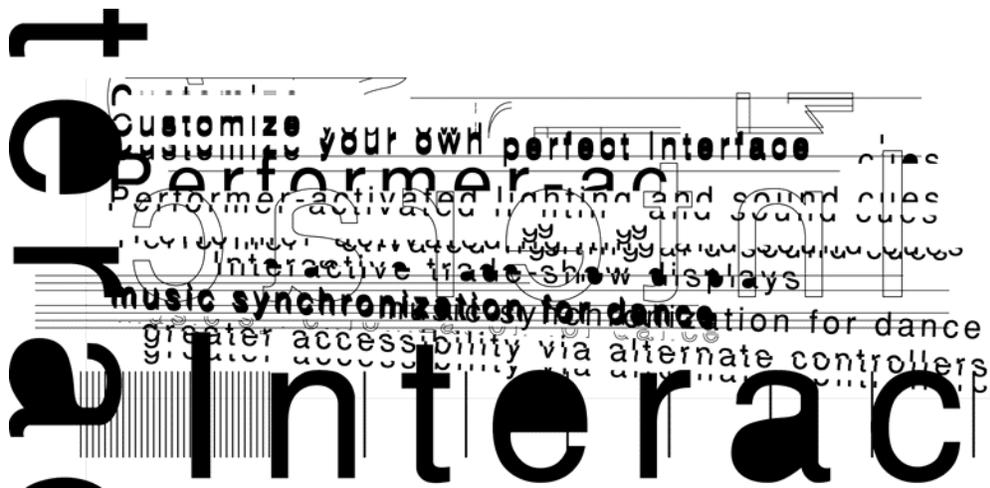
Infusion Systems Ltd.  
2033 Vendome avenue  
Montreal, QC  
Canada H4A 3M4

Tel: (514) 484-5850  
Fax: (514) 484-5852

Email: [info@infusionsystems.com](mailto:info@infusionsystems.com)  
<http://www.infusionsystems.com>



14 February 2008  
© Infusion Systems



## HOW TO CONTROL FLASH WITH I-CUBEX

### INTRODUCTION

This document explains some of the possibilities that exist when working with I-CubeX and the Adobe Flash Player.

The current Adobe Flash 9 Player reads files created in the Flash CS3 IDE and Flex Builder 3 authoring environments. Actionscript 3.0 now allows one to develop powerful multimedia applications for both the web and the desktop. When coupled with I-CubeX, the designer has the ability to control vector graphics and animations using high quality hardware components for such applications as live video mixing, interactive installations, and video games.

### INTERACTING WITH I-CUBEX

The current Flash file format does not currently have native support for common input protocols such as MIDI or Open Sound Control. For the first part of this tutorial we use a Flash-based VJ application called **visp** (<http://www.visp-vj.com>) written by Mike Creighton along with the **visp MIDI PROXY** (vMP). The second part involves I-CubeX and the Flash CS3 IDE.

We assume that you already have some understanding of both I-CubeX and Flash 9/Actionscript 3.0 authoring and some programming experience. We also assume that you already have the I-CubeX editor installed. If not, please refer to the I-CubeX editor tutorial.

### HARDWARE AND SOFTWARE USED IN THIS TUTORIAL

I-CubeX digitizer

I-CubeX sensors

I-CubeX editor (<http://icubex.com/download>)

Flash Player 9 or higher (<http://www.adobe.com/products/flashplayer/>)

Flash CS3 IDE (<http://www.adobe.com/products/flash/>)

Adobe Air Public Beta (<http://labs.adobe.com/downloads/air.html>)

visp and vMP (<http://www.visp-vj.com/download>)

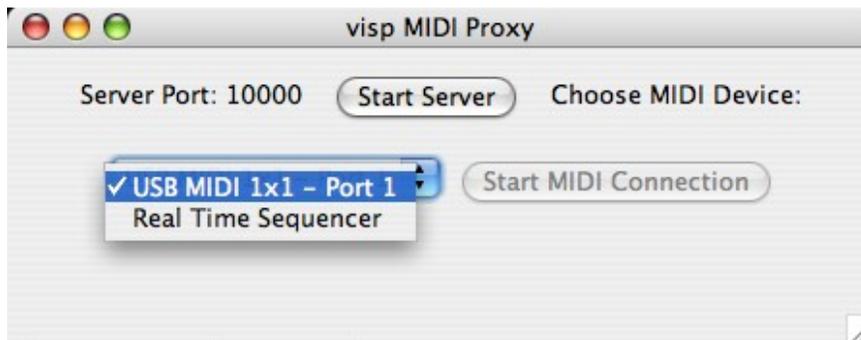
Java 1.5 or higher: <http://www.java.com/en/download/manual.jsp>.

## Part 1: I-CUBEX AND FLASH - visp AND THE visp MIDI PROXY

The first part of this tutorial centers on an existing complex Flash project that has MIDI built in, namely visp and the visp MIDI Proxy. To use the visp MIDI Proxy, Java 1.5 or higher is required. Java can be downloaded for all platforms at: <http://www.java.com/en/download/manual.jsp>.

Download the visp MIDI Proxy at <http://visp-vj.com/download/>. Note that if you are using Mac OS X 10.4.8 or higher you will have to download additional software. Refer to the download page in this case.

The first step is to plug in and activate your MIDI interface. When this is completed, start the vMP. You should see your MIDI interface in the drop down list. The entry for "Real Time Sequencer" refers to the sequencer in JAVA itself and is always present.



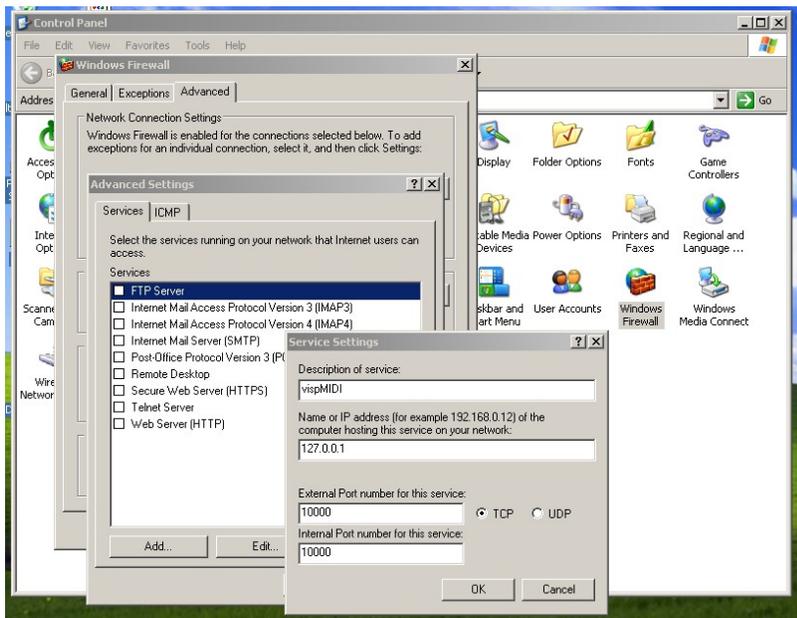
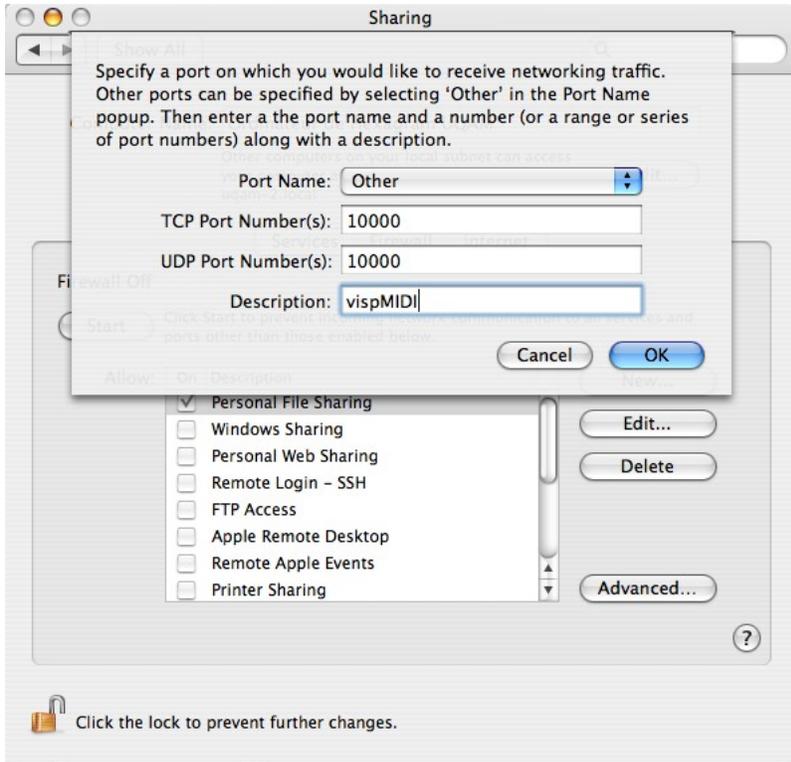
The "Server Port" refers to a value set in the vMP JAVA source code and can be modified.

Click on "Start Server". If the connection is made, click on "Start MIDI Connection". Your connection should now be established and the vMP popup should look like this (with your specific MIDI interface's name beside the port number):



If you have trouble connecting the server, it may be because your ports are blocked. On Mac OS X 10.3 – 4, you can fix this in "System Preferences > Sharing > Firewall" by

creating a "New" entry. In Mac OS X 10.5 and higher, this is in "System Preferences > Security". In Windows, this would be in "Control Panel > Windows Firewall > Advanced > Advanced Settings > Add".



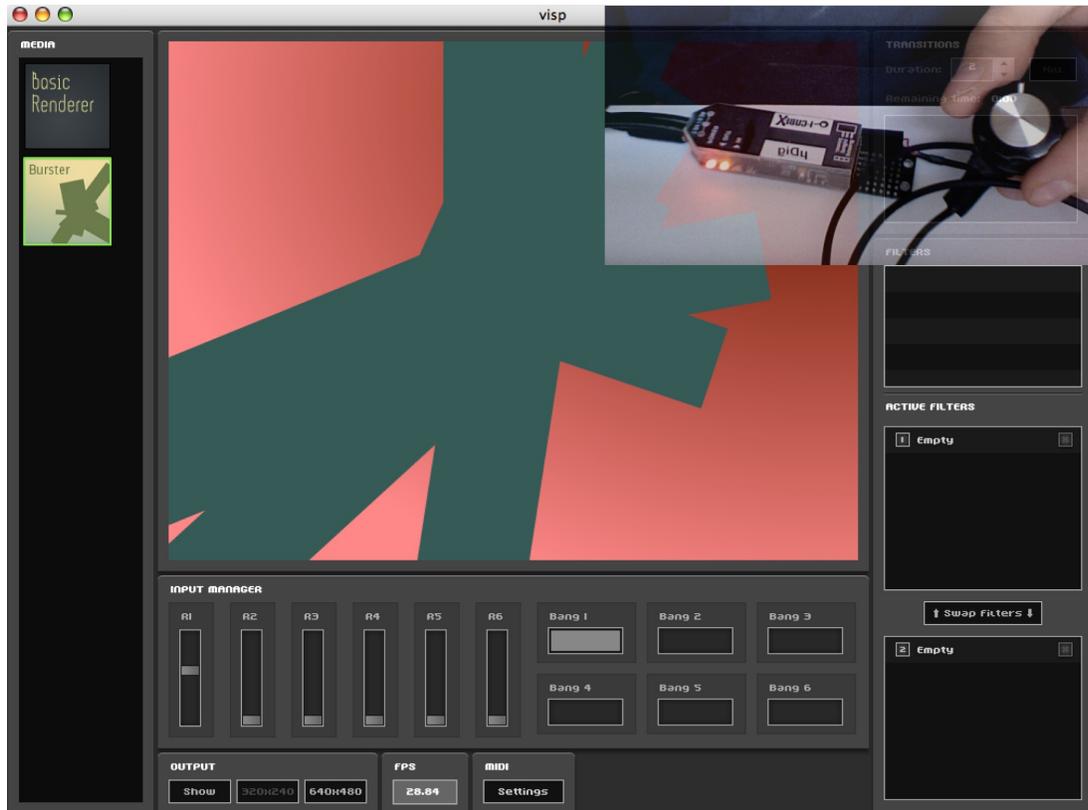
## TESTING THE vMP CONNECTION

Activate your I-CubeX digitizer and connect a turn sensor to the first input. Its output should be a Control Change message. When this is done, quit the editor and start the vMP.

Download the binary version of visp at <http://visp-vj.com/download/> (Also requires the Adobe AIR Public Beta 3 available here: <http://labs.adobe.com/downloads/air.html>). Start up the application and click on the "MIDI Settings" tab. You should see the following screen pop up:



visp uses a simple 'MIDI Learn system'. Click on the "Range 1" field. This field should now be highlighted by a red box. Turn the I-CubeX Turn sensor. If everything is working, you should see the number box in the "Range 1" field change values to the corresponding Control Change number assigned to the Turn sensor in the I-CubeX editor. Close the MIDI settings window. Now, when you use the Turn sensor, you should see the "R1" slider move correspondingly.



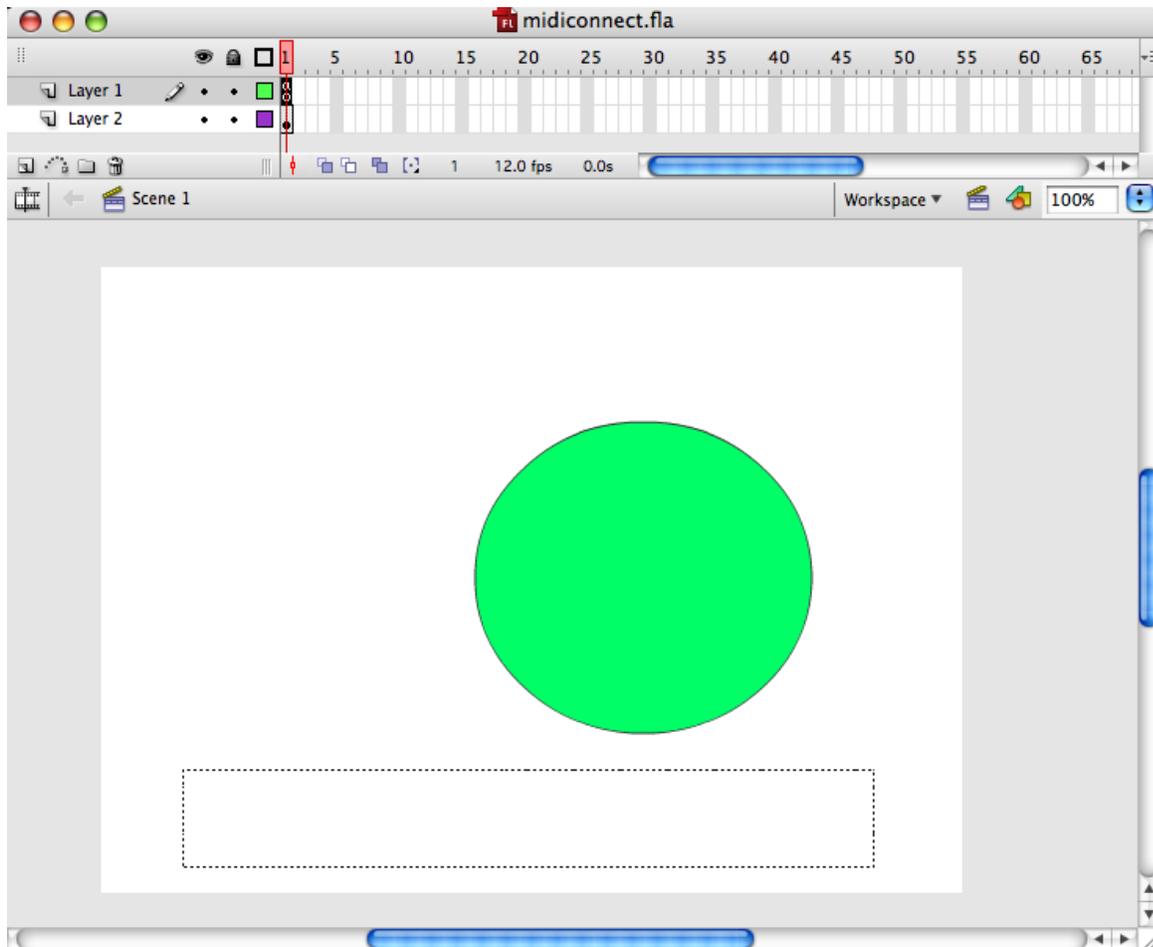
## Part 2: USING I-CUBEX WITHIN THE FLASH CS3 IDE

The visp project was chosen for this tutorial because it is a good example of the multimedia capabilities of the Flash Player. However, the entire project is built in Flex and Actionscript code. Many designers like to use the Flash CS3 IDE with its timelines and visual objects, and so for these users, we'd like to show you how to use the vMP from within the IDE. This portion of the tutorial requires the vMP and the Flash CS3 IDE.

Downloadable files included with this tutorial is a flash file, **midiconnect fla** and a precompiled version, namely **midiconnect.swf**. Along with the swf is **midiconnect.html**, an html document that will allow you to test in a web browser. **AC\_RunActiveContent.js** is required to be in the same folder as midiconnect.swf and midiconnect.html when running in a web browser.

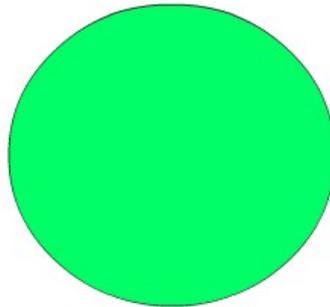
## MIDI INTO FLASH: midiconnect fla

This tutorial includes a Flash CS3 file for download called **midiconnect fla**. Before opening this file in the Flash IDE, make sure you have a turn sensor outputting Control Change messages and the vMP turned on, similar to the first part of this tutorial. Upon opening the file you should see this:



On the timeline, there are two layers, one that contains the Actionscript code, and the other that holds a beautiful green dot and a text field. Before getting into implementation details, let's test the Flash movie.

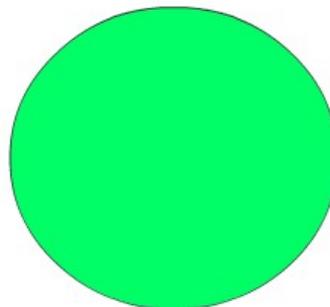
Click on "Control > Test Movie". You should see a Flash movie open up within the IDE. If the movie has indeed connected to the vMP, it should look like the following picture:



Socket Connected



Move the Turn sensor and you should see the corresponding MIDI message in the text field and the green dot change it's y position.



Controller # = 1 Data = 16



Publish this file by clicking on "File > Publish". In the same folder as midiconnect fla, you will see midiconnect.html, which can be used in a web browser, and midiconnect.swf which can be used in the Flash player outside of the IDE. You can now control your own Flash files using I-CubeX!

## IMPLEMENTATION OF midiconnect fla

The Actionscript code in midiconnect fla is heavily commented to help you extend its functionality. As a result, this document outlines only its basic functions.

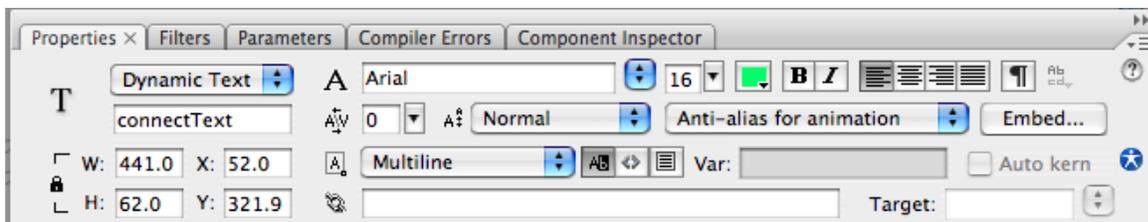
After the import statements comes this block of code:

```
12 //connect to the vMP
13 var xmlSocket : XMLSocket = new XMLSocket();
14 xmlSocket.addEventListener(Event.CONNECT, handleSocketConnected);
15 xmlSocket.addEventListener(DataEvent.DATA, handleSocketData);
16 xmlSocket.addEventListener(IOErrorEvent.IO_ERROR, handleSocketIOError);
17 xmlSocket.addEventListener(SecurityErrorEvent.SECURITY_ERROR, handleSocketSecurityError);
18 try{
19     xmlSocket.connect("127.0.0.1", 10000); //this IP and port setting must also be changed in vMP
20 } catch (e : SecurityError) {
21     //this maybe called if you haven't properly started the vMP or if the Flash movie has incorrect
22     connectText.text = "There was a security error when attempting to connect to the visp MIDI Proxy.";
23 }
```

This code initializes an XML socket connection to the vMP server over the network layer of the operating system. As a result of this code, the security settings for this Flash movie must be set to “Access network only” ( in “File > Publish Settings > Flash”). Failure to set this flag will result in a security error. If a connection is successfully established, the text field will be updated by the following code:

```
26 function handleSocketConnected(e : Event) : void {
27     connectText.text = "Socket Connected";
28 }
```

Notice that the text field is referenced by the variable `connectText`. This, like the the variable `circleMidi` which references the green dot, are set in each object’s respective “Properties” pane, as seen below:



The last code segment deals with incoming MIDI data from the I-CubeX. This is where you can control objects using the your I-CubeX hardware.

```
42 function handleSocketData(e : DataEvent) : void {
43     var data : XML = new XML(e.data);
44
45     switch(data.name().toString()){
46         case "c": //controller change
47             // receives a number (n) attribute
48             // receives a data (d) attribute
49             connectText.text = "Controller # = " + parseInt(data.@n) + " Data = " + parseInt(data.@d);
50             //move the green dot!
51             circleMidi.y = parseInt(data.@d) * 200/128 + 100; //magic numbers keep the green dot in view
52             break;
53
54         case "n1": //noteOn
55             // receives a pitch (p) attribute
56             // receives a velocity (v) attribute
57             connectText.text = "NoteOn # = " + parseInt(data.@p) + " Velocity = " + parseInt(data.@v);
58             break;
59
60         case "n0": //noteOff
61             // receives a pitch attribute
62             connectText.text = "NoteOff # = " + parseInt(data.@p);
63             break;
64     }
65 }
```

The main idea is that an incoming MIDI message is categorized according to type (i.e. control change, note on, or note off) via a `switch` construct. The corresponding data for each message is then expressed as an integer by using the `parseInt()` function. With this simple setup, one can use conditionals such as `if` statements to recognize different messages and therefore different sensors, and then control multiple objects within the Flash movie. The possibilities are limited only by your imagination!