

# I-CUBE X

## How to use I-CubeX with ... Director v8.5

*I-CubeX: The ultimate MIDI controller!*

Infusions Systems Ltd.  
2033 Avenue de Vendome  
Montreal, Qc  
Canada H4A 3M4

Toll free (USA/Canada): 1 877 4 ICUBEX (1 877 442 8239)

Tel: +1 (514) 484 5850

Fax: +1 (514) 484 5852

Email: [info@infusionsystems.com](mailto:info@infusionsystems.com)

<http://www.infusionsystems.com>



12 February 2004  
© 1997-2004 Infusion Systems

Interac  
Customize your own perfect interface  
Performer-activated lighting and sound cues  
Interactive trade show displays  
music synchronization for dance  
greater accessibility via alternate controllers

# How to use Director v8.5 with I-CubeX

---

## INTRODUCTION

This document explains the various possibilities that exists to the end-user when working with the I-CubeX and Macromedia Director software.

## PRODUCT

Macromedia Director is an authoring tool for creating interactive multimedia.

## INTERACTING WITH I-CUBEX

The graphic designer can create a multimedia application which will react to certain MIDI incoming and/or outgoing data by using a an Macromedia Xtra plugin for MIDI. Several Xtra plugin products are available namely: Xmidi for Macintosh OS 8.6, OS 9.x and OS X (<http://www.troikatronix.com>), MIDlio X-Tra for Macintosh OS 8.6, OS 9.x and PC Windows 95/88/ME/NT/2000/XP (<http://www.audiomulch.com/midiio>) and SequenceXtra for Macintosh OS 8.6, OS 9.x, OS X and PC Windows 95/88/ME/NT/2000/XP (<http://www.sourceforge.nu/index2.htm>).

Before you start building a specific multimedia application, you must have some idea of what you want to do. But before you do so, we provide here a simple step by step example of what can be done with Director and the I-CubeX digitizer just to get you started.

We assume that you already have some understanding of both products.

## SEQUENCEXTRA v1.9.3 FOR WINDOWS

### STEP 1

Before you start Macromedia Director, make sure that you have installed properly the SequenceXtra according to their instructions.

### STEP 2

Start Macromedia Director. By default, the application creates an empty "Stage", "Score" and "Cast" windows. For the purpose of the demonstration, we will import two files in the Cast window : one avi and one bitmap. We let you choose two of your choice. Here, we use the files named small.bmp and Drums.avi. To import these files, just right click in the Cast window and follow as indicated (see figure 1 and 2).

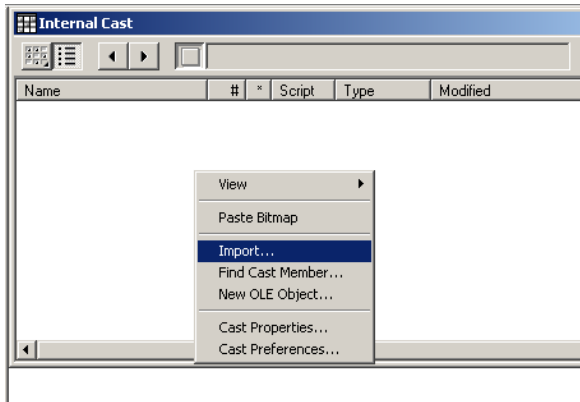


figure 1

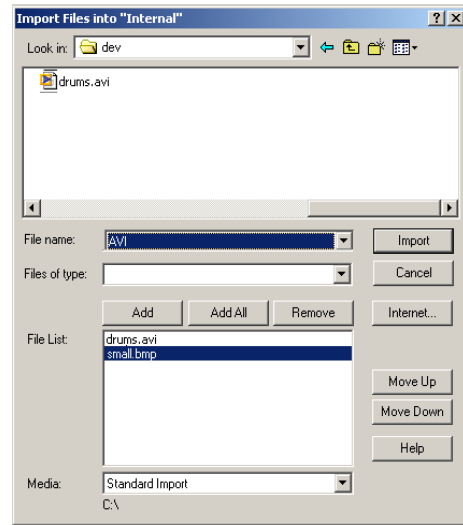


figure 2

### STEP 3

Drag the Drums Cast member into the Score window as shown in figure 3. Also, resize your window for the avi in the Stage window as shown.

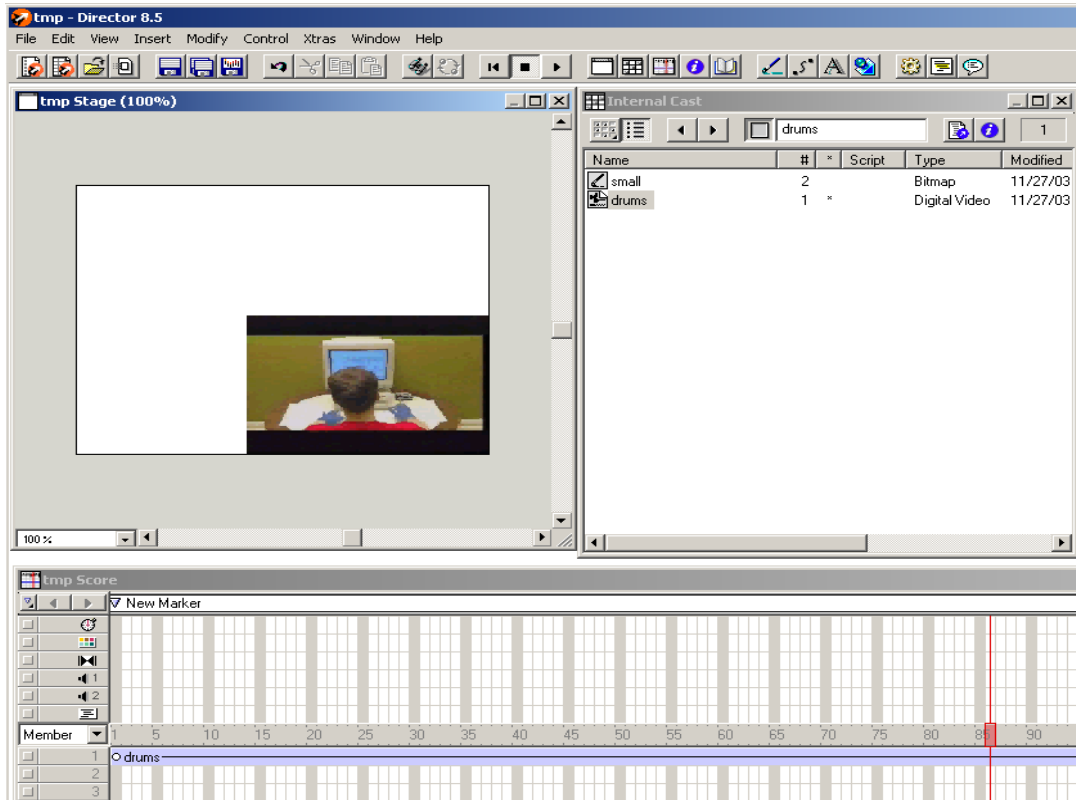


figure 3

#### STEP 4

Drag the small Cast member into the Score window as shown in figure 4. Also, reposition the bitmap in the Stage window as shown.

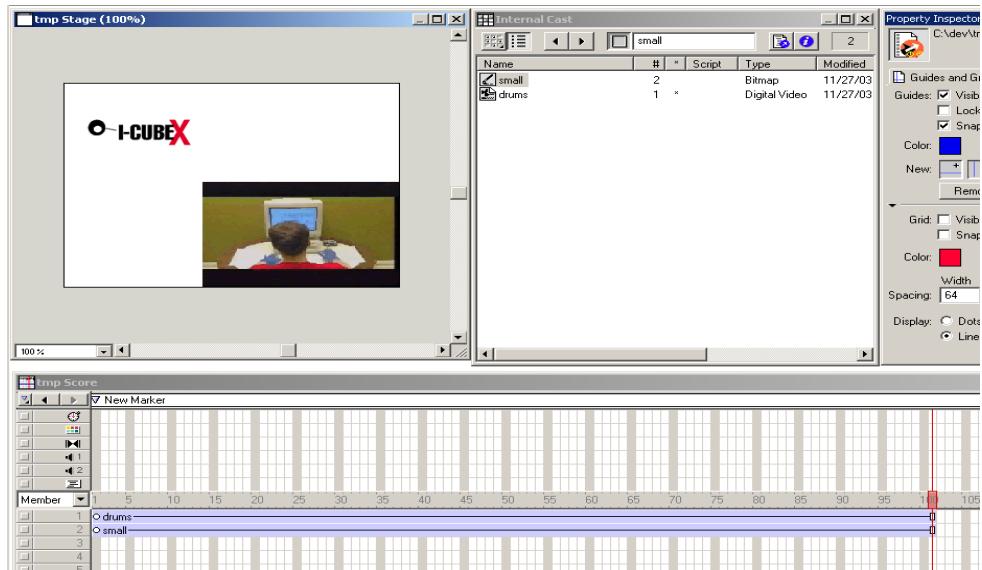


figure 4

#### STEP 5

The next step is more trickier. You must be somewhat knowledgeable of the programming language environment of Director called Lingo. For the purpose of the demonstration, we will use a behavior. A behavior is prewritten Lingo script that you use to provide interactivity and add interesting effects to your movie. With that behaviour, we will then modify its code so to add some interactivity with incoming MIDI data.

So to add a behaviour, we must first open the library window. Selet the Window menu and choose Library Palette (see figure 5). You should the library window as shown in figure 6.

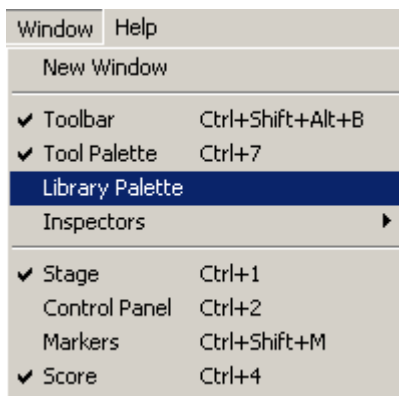


figure 5

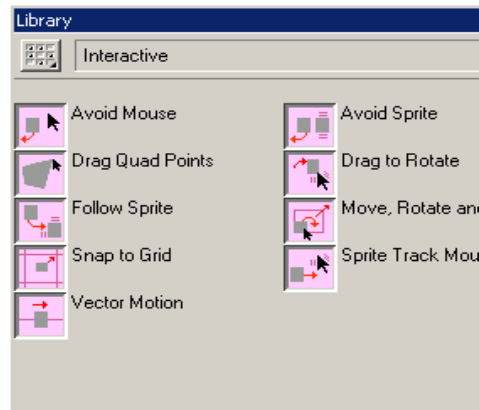


figure 6

## STEP 6

The library has quite a variety of behaviours. They are grouped as indicated on figure 7; just click on the bottom right corner of the little square found on the top left part of the window. Select the group Animation / Interactive (see figure 8).

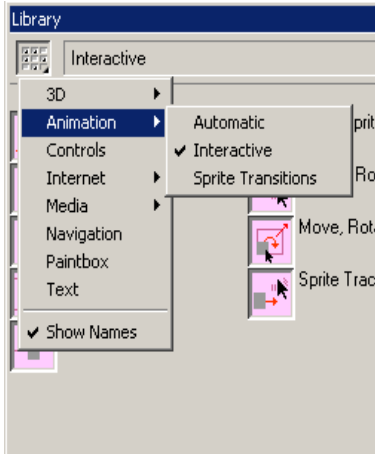


figure 7

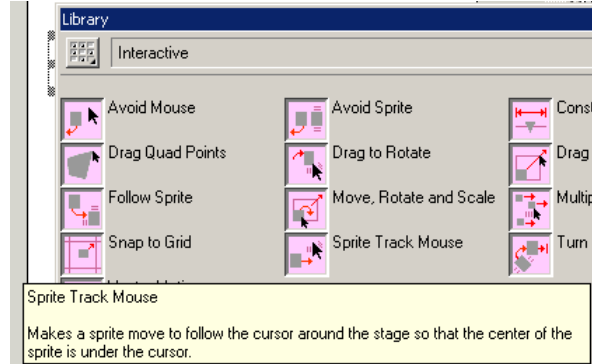


figure 8

## STEP 7

Then select the Sprite Track Mouse with the mouse and drag it into the Score window on top of the small track. A window should popup as shown in figure 9. Click OK. Close your Library palette window.

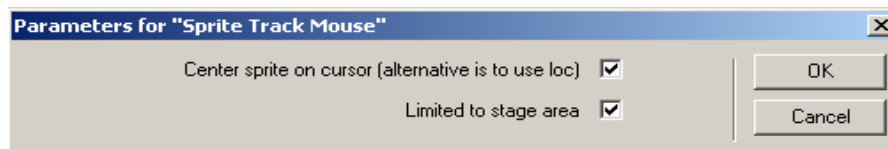


figure 9

## STEP 8

Double click on the Sprite Track Mouse item found in the Internal Cast Window as shown on figure 10. The window Behavior Inspector should appear as in figure 11.

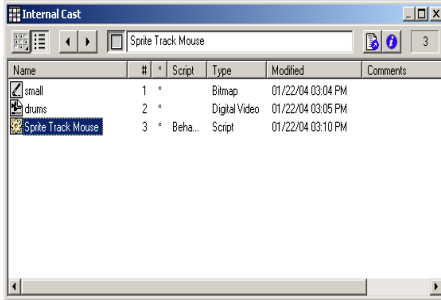


figure 9

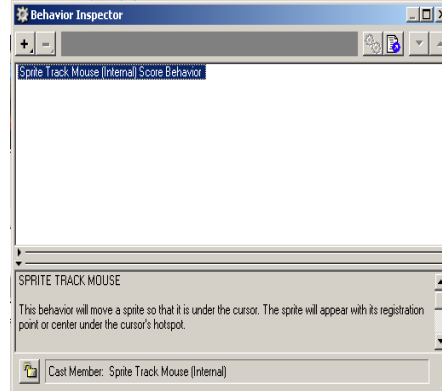


figure 10

## STEP 9

Right click on the Sprite Track Mouse (Internal) Score Behaviour found inside the Behavior Inspector window. A pop-up menu should appear as shown in figure 11. Select the Script option in the menu.

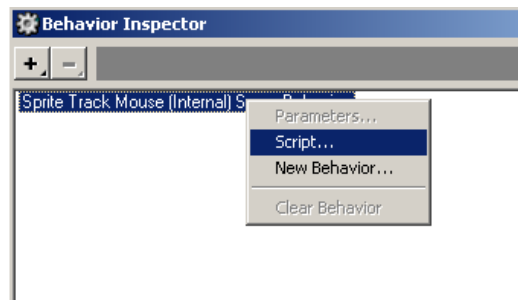


figure 11

A window called: Behavior Script 3: Sprite Track Mouse should be shown as a result of selecting Script as in figure 12.

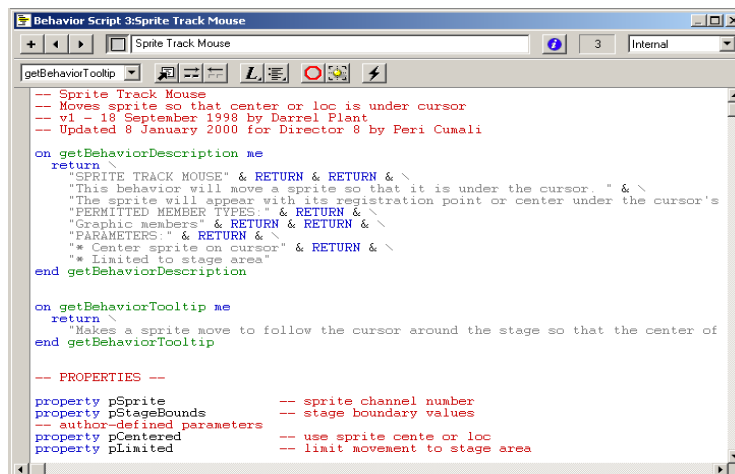


figure 12

## STEP 10

Essentially, what we have done is opening a prewritten Lingo script that Director provides to the end user. You can actually edit that Lingo program or Lingo script to do specific operation for yourself. This is what we will do now. Modify the Lingo script so to control a moving sprite behavior. The sprite will move according to the mouse cursor as the original behavior did but we will add an extra behavior to it. We will rotate the bitmap on itself according to a MIDI incoming data NOTE ON message value.

Take a look at the script (We assume that you have a minimal knowledge of Lingo here). These two methods or functions are in the script :

- 1- [on mInitialize me](#)
- 2- [on mUpdate me](#)

We will modify both of them for the purpose of the demonstration. Goto on mInitialize me:

```
...
on mInitialize me
-- called from beginsprite ...
...
```

Then insert the following code (the one marked in blue):

```
...
on mInitialize me
-- called from beginsprite
-- sets initial values for all conditions
pSprite = me.spriteNum
pStageBounds = point (the stage.rect.width, the stage.rect.height)

-- ICubeX code... we use the first MIDI device
sxSetOutput(0)
-- end of ICubeX code

end mInitialize
...
```

Then add to **on mUpdate me** the blue code as indicated:

```
...
on mUpdate me
-- called from prepereFrame handler
-- moves sprite to position under cursor MIDI incoming data.

-- I-CubeX code added here
--
global isEvent
global theEvent

isEvent = sxGetMIDIIn()
if listP(isEvent) then
theEvent = string(isEvent)
-- On NOTE ON message we do ...
if (isEvent[2] >= 144) AND (isEvent[2] <= 159) then
-- NOTE ON will change the position of the sprite
```

```

        sprite (pSprite).rotation = isEvent[4] * 2.85
    end if
end if
-- End of the ICubeX code

if pCentered then
    vRect = mCenteredRect (the mouseLoc, sprite (pSprite).rect)
else
    vRect = mCenteredRect (the mouseLoc + \
        mCenterOffset (), sprite (pSprite).rect)
end if
if pLimited then vRect = mStageLimit (vRect)
sprite (pSprite).rect = vRect

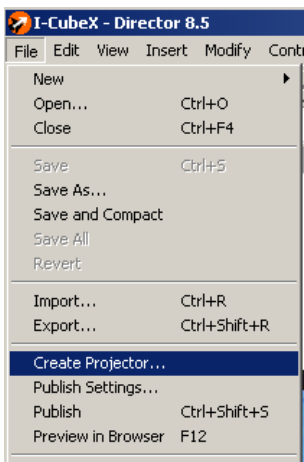
end mUpdate
...

```

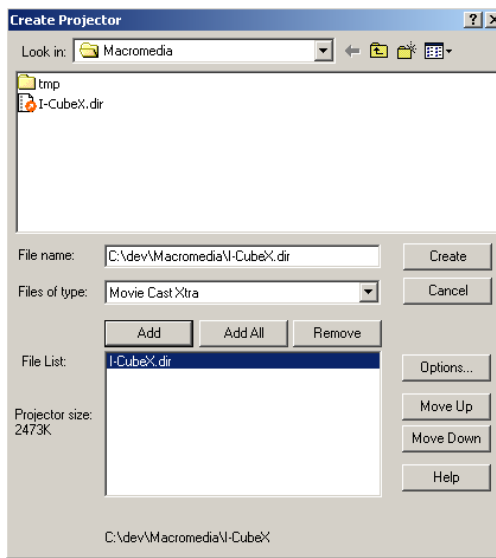
This change essentially modify a sprite behavior. It will rotate according to the NOTE ON message value.

### STEP 11

Save your project from the File / Save menu option, we used the name I-CubeX for the project name. Then from your File menu select create Create Projector... (figure 13). Select I-CubeX.dir file and click on the Add button. Then click on Create button. This will create an executable called Projector.exe.



**figure 13**



**figure 14**



## STEP 12

Start I-CubeX editor and set the Midi settings; Make sure that you digitizer is connected to the MIDI interface (see figure 10 and 11).

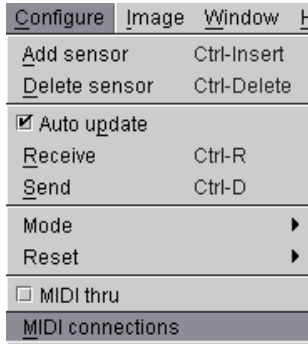


figure 15

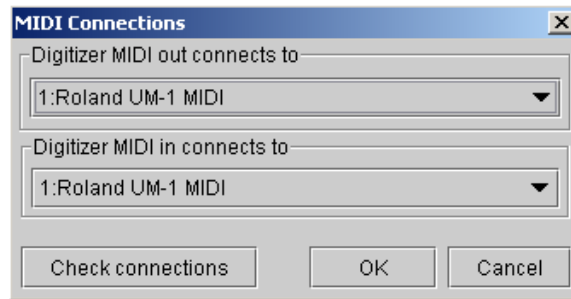


figure 16


## STEP 13

From the File menu of the I-CubeX editor, select New so that the digitizer is properly resetted.

## STEP 14

From the Configure menu, select Add sensor so that you create a virtual sensor.

## STEP 15

Inside the sensor window, click on the ON button (see figure  ). Then select Note On in the Midi Map section and Channel 1.

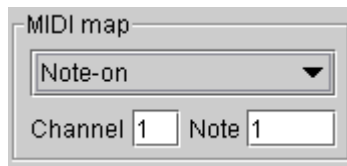


figure 17

## STEP 16

Quit the I-CubeX editor application. Connect a sensor such as the Turn Sensor to input 1 of the digitizer. Start the Projector.exe application. Move your mouse cursor then play around with the sensor. You should see the bitmap move then rotate on itself !