

# FlashMidi Mac version

pre-alpha

## Introduction

I've been asked for a Mac version of FlashMidi ever since the PC version was released. But when the first requests arrived I haven't planned to release a Mac version for a simple reason : I've never had a Mac nor developed a program for the Mac platform.

But as this request became more often, I finally decided to think about it.

My first try was an Applet version. I didn't liked it because of :

- the limited control over the file system to save Midi files
- I guessed I'd have some problems with the Midi input.
- The more complex deployment of FlashMidi Flash movies.

But as this solution was the only one available for a PC developer to build a Mac application I gave a try.

This version worked, at least on PC, but as the Java applet and the Flash were communicating through JavaScript I ran into problems on InternetExplorer Mac and Safari.

A few days before I've attended a conference by Mike Chambers(macromedia) at the Centre Pompidou (Paris) where he explained Macromedia will release a set of scripts to make the communication between Flash and Javascript easier and cross platform. So I waited until the "Flash / JavaScript Integration Kit" was out. But when it was released I found some other problems which could be fixed and another one I didn't want to cope with : the use of FlashMidi for a developer would be incredibly hard : Create the Flash, configure the JavaScript, put the Applet jar in the right folder,...

So I did the only thing to be done : **I bought a Mac**. Not a brand new one but an old G4 Graphite on e-bay for 200€.

Since then I'm trying to re-create Flash Midi using Xcode. But as this is my first program it's a bit tortuous because I used, as much as possible, the technologies I already knew.

Which leads me to the next part :

## Help wanted

As you will see in the next part the current operation mode of FlashMidi isn't simple and can lead to a lot of error sources and big difficulties to track bugs.

My goal is to have a single [Mac application + a SWF], not the current [launcher + server/player + SWF wrapper + SWF] system.

If you think you can help you can leave a message in this forum :

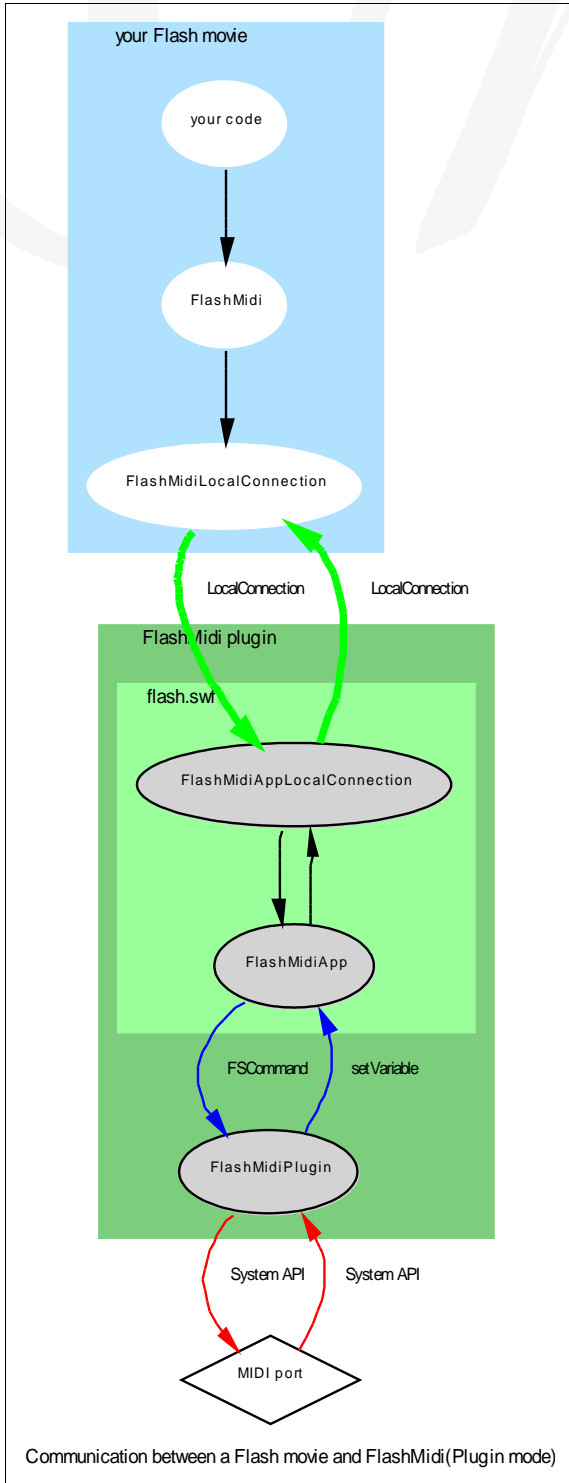
[https://sourceforge.net/forum/forum.php?forum\\_id=480671](https://sourceforge.net/forum/forum.php?forum_id=480671)

# How it works

## PC version

I will first explain how the PC version works :

The Flash movie starts FlashMidi using the System.Product.launch() function.



The main application is then launched.

This application is a Delphi program which embeds a Flash ActiveX.

This Flash movie communicate with the calling FlashMovie through LocalConnection. It forwards commands to the Delphi application through FSCCommands.

The Delphi application handles the commands and outputs the midi signal using system API.

When the Delphi application needs to send events to the calling Flash movie it sends a SetVariable command onto the embedded Flash movie which forwards it to the calling Flash movie.

When the Flash movie closes a timeout occurs and the application quits.

## Mac version

### Components of the Mac plugin

The Mac plugin is composed of the following components :

- A **launcher** : A RealBasic program which launched the **main application** and the **server**. *I chose Real basic because the application launched by Flash must be a plain executable(not a folder) and command line applications built in Xcode in C++ crashes the Flash plugin.*
- The **main application** : A Cocoa Objective-C program which loads the **HTML Wrapper** into a webview. *I chose Cocoa/Objective-C because Java application built in Xcode cannot use the Webview component.*
- The **Server** : A Java program which opens a XMLSocket server and handles the commands of the **appConnector**. This program sends the Midi commands using the *javax.sound.midi* package. *I chose Java because I already had the code from the Applet version.*
- The **HTML wrapper** : This is a simple HTML file loading the **appConnector**.
- The **appConnector** : A swf communicating with the client Flash movie in LocalConnection and the **server** in XMLSocket.

As you can see this is not very simple.

## Communication between the Flash movie and the plugin

The Flash movie starts the plugin with the System.Product.launch() function.

The application started is the **launcher** which is just a RealBasic application launching two other components : The **server** and the **Main application**. Once those two applications are running the **launcher** quits.

The **main application** is a Cocoa Objective-c application with a WebView containing the **HTMLWrapper** which contains the **appConnector** SWF.

The **server** is a Java Application built with Xcode which opens a XMLSocket server for the **appConnector** to communicate with it.

When the Flash movie executes a FlashMidi command :

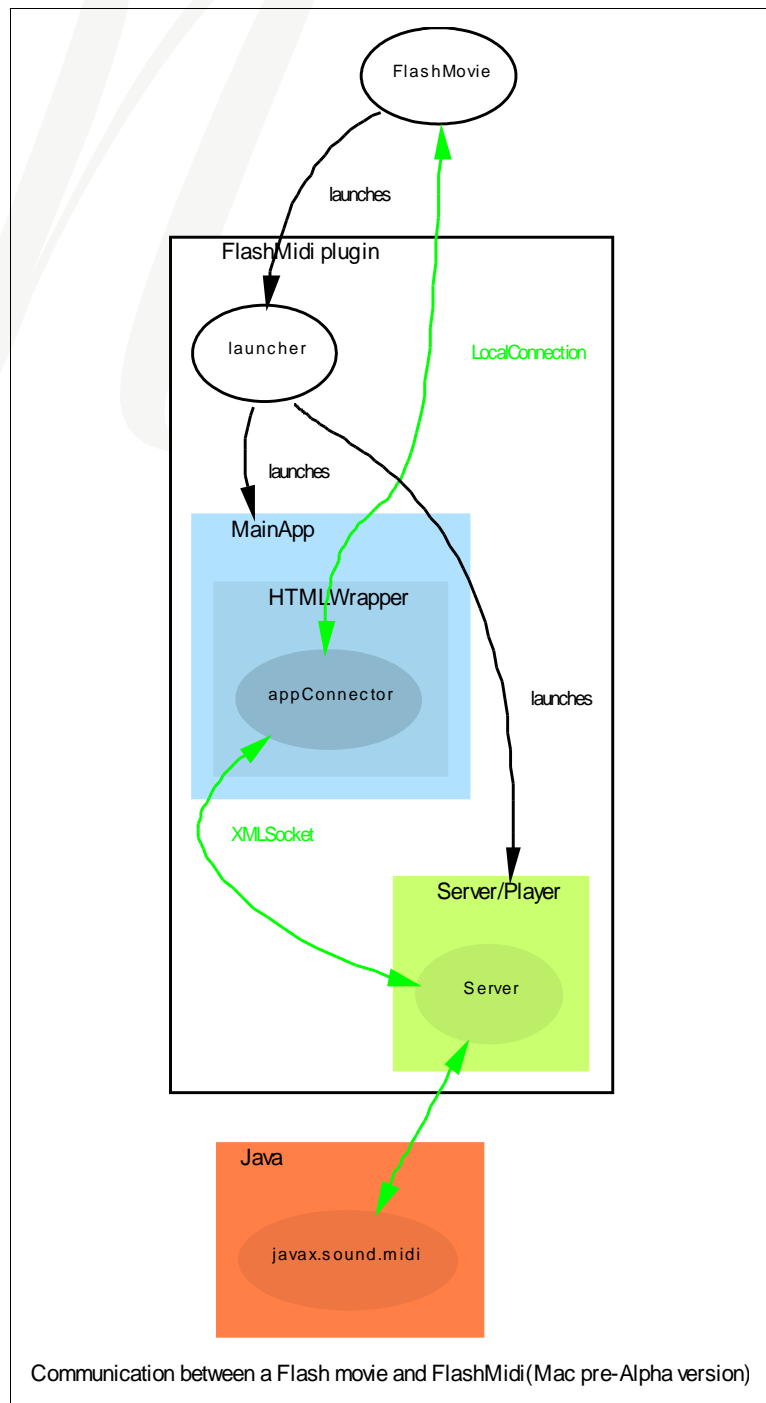
1. The command is sent to the **appConnector** through a LocalConnection.
2. The **appConnector** forwards the command to the **Server** through the XMLSocket.
3. The **server** handles the command.

When the server has to return an event:

1. It passes the event to the **app connector** through the XMLSocket
2. The **app connector** sends the event to the Flash movie through the LocalConnection.

When the Flash movie is closed :

1. A timeout occurs in the **app connector**, it performs a GetUrl.
2. The **HTMLWrapper** is unloaded
3. The **server** detects that the XMLSocket connection has been closed and quits.
4. The **Main application** detects the URL change and closes itself



## The perfect Mac Version

Here is what I think the FlashMidi Mac version should be :

### *The perfect application*

The installation searches the hard drive for the Netscape plugin for flash and stores its path somewhere.

FlashMidi should be a single executable written in Objective-C or C++ (for performance). When run it should instantiate the Netscape plugin and load the **appConnector** into it. The communication would use the FSCCommand /SetVariable mechanism described in the PC version.

### *The really good application*

FlashMidi should be a single executable written in Objective-C or C++ (for performance).

It should use a WebView component to load the **HTMLWrapper**.

It should open a XMLSocket server to communicate with the **appConnector**.

It should handle the Midi commands.

It's nearly the same mechanism than the current Mac version except the Java **server** is included into the **Main Application** and there is no need for the **launcher**.

If you think you can help and have enough time please contact me

Alexis ISAAC

[moi@alexisisaac.net](mailto:moi@alexisisaac.net)

or leave a message on the forum :

[https://sourceforge.net/forum/forum.php?forum\\_id=480671](https://sourceforge.net/forum/forum.php?forum_id=480671)