



Macromedia Flash and VB: A Potent Combination

Mindfire Solutions

www.mindfiresolutions.com

March 11, 2003

Abstract

This article focuses on an often-neglected but all-important aspect of any software application – the user interface. For years, desktop applications have had boring gray rectangular windows, buttons etc. Animation, multi-media, visual effects and custom elements were possible, but required large amount of custom programming. Even making a non-rectangular button would need good C++ skills. The web and its use of Flash change the equation. Web-based applications look far more attractive and user-friendly than comparable desktop application, but are easy to develop with Flash. In this article, we present ideas for using Flash inside VB applications, thus injecting new life into desktop software products.

Copyright

© 2002 Mindfire Solutions. All rights reserved.



Contents

Abstract	1
Contents	2
Specifications	2
Introduction	3
Starting with Flash (for the VB programmer).....	3
Flash programming: ActionScript.....	3
Flash file formats	3
FSCommand in ActionScript.....	4
Flash and VB.....	4
Flash.ocx (SWFlash.ocx) ActiveX component.....	4
Flash playing inside VB programs.....	4
FSCommand event.....	5
Set/GetVariable functions.....	6
Our sample application	6
Creating the simple Flash file	6
Programming VB-Flash interaction.....	7
Advanced Topics	7
Division of responsibilities	7
Data exchange between Flash and VB.....	8
Use of XML	8
Passing notifications from VB to Flash	8
VB/Flash issues in a real-life project.....	9
VB crashed while adding the Flash component on some machines	9
Passing values from one Flash frame to another	10
Error in Flash for scripts, which execute for long	10
Passing key events to Flash from VB form containing other controls	11
Conclusion	12

Specifications

- Windows, Visual Basic 6.0, Macromedia Flash MX



Introduction

In today's world, where it is not just about time, memory and space, LOOKS do matter. While browsing through websites, you might have seen and appreciated interesting interfaces with visual and multi-media effects. On the contrary, users of desktop applications developed in Visual Basic/VC++ have to deal with monotonous, gray interface.

This article might change how you feel about your desktop applications in the times to come. It explains how you can actually use the user-interface capabilities of Flash with the programming and database power of Visual Basic, to make Windows applications that look "slick".

Starting with Flash (for the VB programmer)

Designing screens using Flash is fun and easy because of the large number of inbuilt, easy to use components it contains - for example textbox, list box, scrollbar, buttons etc. After installing Macromedia Flash MX (not the free Flash player) you can use it to create screens and "movies" with different Flash components.

Flash programming: ActionScript

Flash MX isn't just about designing; rather you can make it intelligent by using ActionScript, which Flash exposes to the developers. With the use of ActionScript it is even possible to extend the existing Flash components to suit your needs.

Using ActionScript you can also make animations. For example in our application we were using a menu that slides upwards when a button was clicked and went off sliding downwards when the button was clicked again.

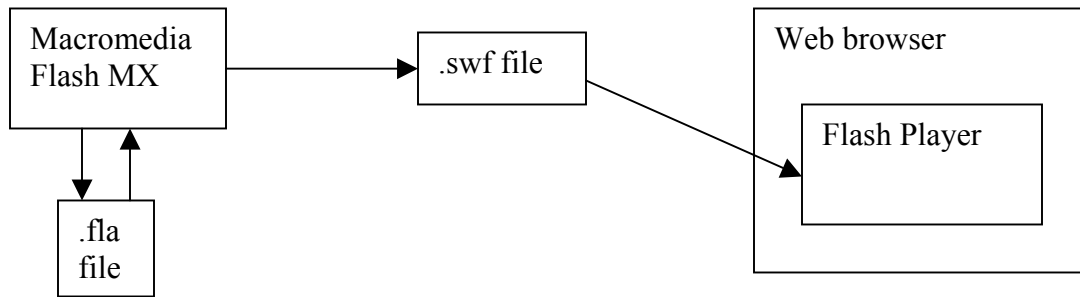
The detailed documentation needed to learn ActionScript and how to create screens, comes along with the installation of Flash MX.

Remember that ActionScript is a full-featured programming language, and if you are a VB programmer you can pick up ActionScript quite rapidly.

Flash file formats

Using Flash MX, you can create a .fla file that contains your graphics and Action Script. Then you can convert this .fla to .swf file using the Export Movie command under the File option in menu.

The Flash files that you see on web sites (which play inside your browser) are .swf files. .swf files are capable of being played out by the Flash Player as depicted in the following diagram.



FSCCommand in ActionScript

To be able to communicate with any application outside the Flash environment, ActionScript exposes a method, FSCCommand (command, arguments). This method takes two parameters, first one is the command you wish to execute and second the arguments that you can pass if needed.

FSCCommand can be called from anywhere in flash, on a button click, during execution of the flash, or anywhere you want to call some outside code. For demonstration purpose our sample application calls it on the Button click.

Flash and VB

Flash.ocx (SWFlash.ocx) ActiveX component

To view the Flash .swf files in VB you have to add the Flash component to the project. To add a component, click on the Project and then on Components. Look for the name “Shockwave Flash” in the list and include it in the project. The file name against this component points to Flash.ocx (or SWFlash.ocx).

This component is installed on your system when you install Macromedia Flash MX or the Flash Player. It is a full-featured ActiveX component with interfaces that enable its usage within VB applications, using Microsoft’s COM/ActiveX technologies.

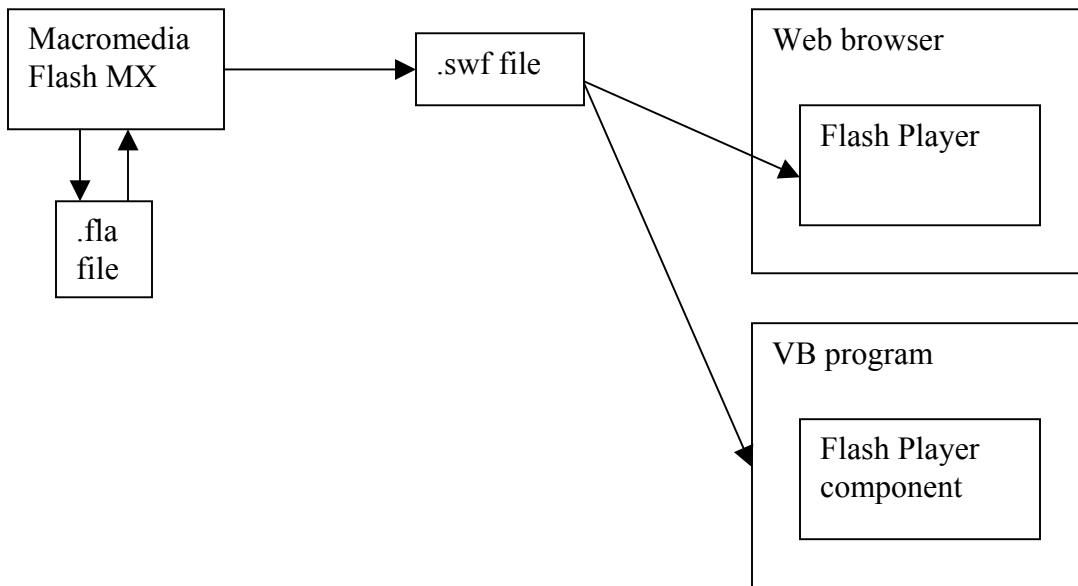
This component has the ability to play Flash .swf files and is the core of Flash integration with Visual Basic programs.

Flash playing inside VB programs

The fundamental point to remember is: Flash integration in VB simply means the ability to play Flash files inside your VB programs using the Flash component. These .swf files are exactly the same as any Flash file played in your browser and nothing special. The same content can be reused on the Web and inside your desktop VB programs.



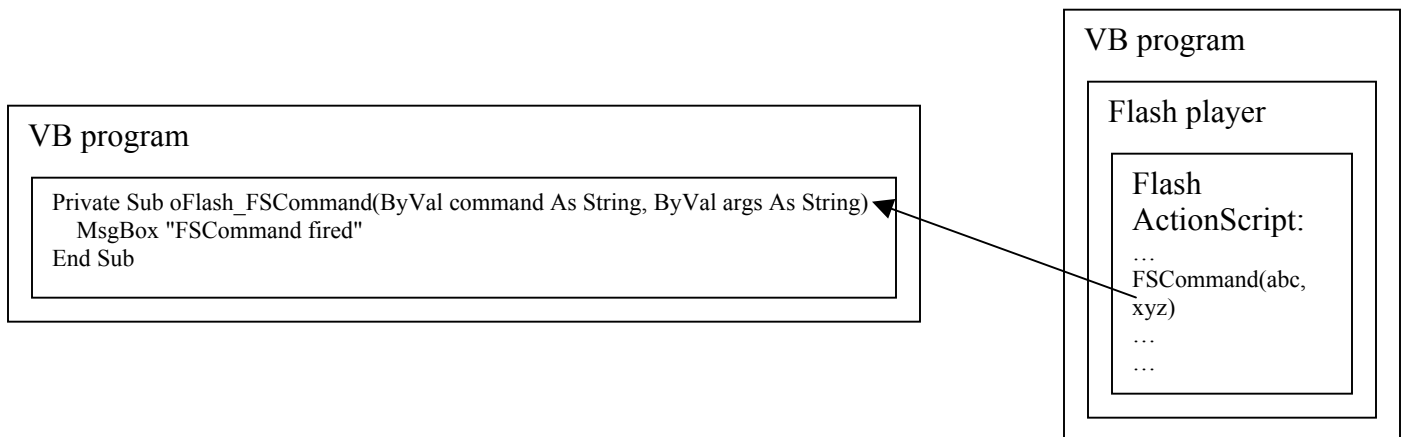
Flash and VB: A potent combination



FSCCommand event

The Flash component has several functions and events, which you can find by using VB's Object Inspector. The most important of these is the FSCCommand event. As you may guess, this is closely related to the FSCCommand ActionScript function.

When you call FSCCommand in your ActionScript and if the Flash file is being played inside the Flash component, it will generate a FSCCommand event in your VB program. If we write code inside the FSCCommand in VB, this code will then execute.





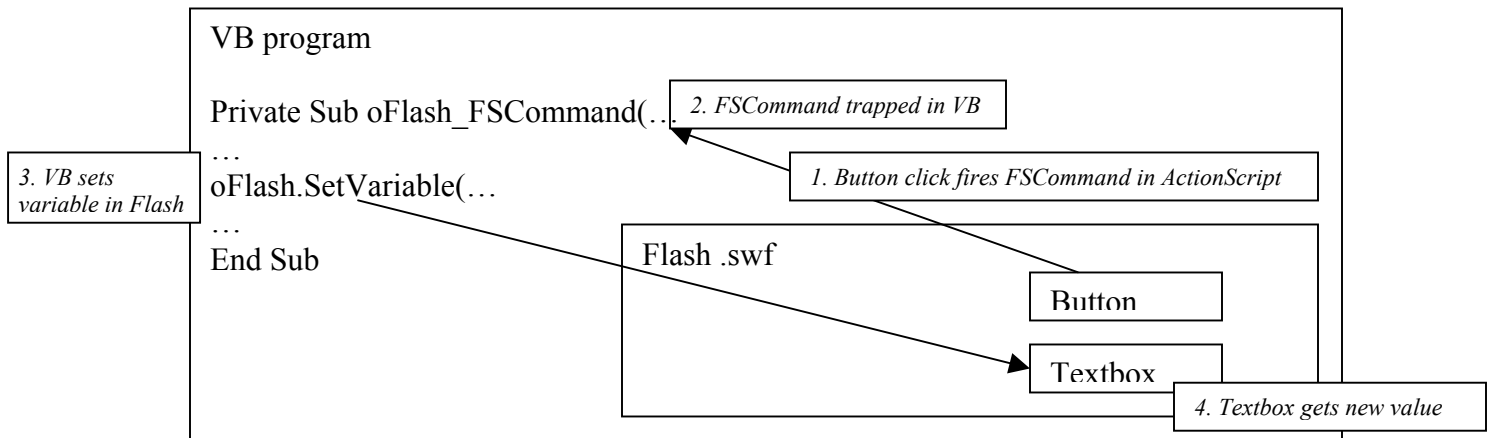
Set/GetVariable functions

These functions are used when communication needs to flow in the opposite direction i.e. from your VB program to the Flash player.

A very useful technique is to loop inside the Flash program on a variable, and then to set the variable from VB to indicate something.

Our sample application

To illustrate the concepts described in this article, we have made a simple sample application. When you click a button in Flash, it will notify your VB program which will then set a Flash textbox's text from the VB program. It looks like this:



Creating the simple Flash file

Open Flash MX application. Create a new file and add a textbox from Tools window to the frame. The basic structure in Flash is called a Frame. Right click on the textbox and choose Properties. This will open a properties window at the bottom of Flash screen. Select Dynamic Text in the Text tool option. Set the Instance name as sampleField. Set the Var as sampleFieldVar. Then enter any text into the textbox. This value we will change from VB using FSCCommand – but more about that later.

Now to add a button, on the right side you might see a Flash UI Components window. In case it's not there, click on Window in menu and click on Components. This brings up the UI components. Drag and drop a Push Button onto the frame. Right click on the button and click on Actions. This opens a window where you can type. In case you are not able to type into that window find an option within this actions window to be set to Expert Mode instead of a Normal Mode. Now type in the following lines there:

```

on(press) {
fsccommand("setvalue", "Sample Value");
}
  
```



Now save this file as sample fla. Then to convert it to an swf file, go to File and click on Export Movie. Save the file as sample.swf. That's it, you are ready to use this flash file in your VB application.

Programming VB-Flash interaction

Insert a ShockwaveFlash component onto a VB form. Name the Flash component as swfFlashScreen. Now to load the Flash (.swf) file into this component use the following function:

Call swfFlashScreen.LoadMovie(0, filename)

The filename is the full path of the swf file. The first parameter signifies the level where Flash is to be loaded; we used 0 for root level.

The FSCommand event is invoked automatically when FSCommand is called from Flash file. So you have the command and arguments in VB (generated from Flash), and you can perform the required actions on them.

Now, that we are able to pass command/values from Flash to VB, what about the other way round, i.e. how to pass information from VB to Flash? I will explain how we did it by passing a value from VB to Flash.

In Flash file, we made a textbox named sampleField, with var as sampleFieldVar. We can set the value of the textbox from VB using this Var field name. Use the following method to do it:

Call swfFlashScreen.SetVariable("sampleFieldVar", value)

The first parameter is the textfield Var name, and the second is the value to be assigned. This call works for the currently loaded Flash screen. Now that value is in the flash textbox, you can use it the way you like.

Advanced Topics

Division of responsibilities

When making “cool” user interfaces in VB using Flash, we have to remember that we are dealing with two development environments and two programming languages. Therefore, it becomes important to have a clear division of responsibility.

Though we will not delve into it here, good and anyway-should-be-followed programming practices like object-oriented design, modularity, responsibility division, abstraction and layered programming become especially crucial in achieving results with two languages.



Data exchange between Flash and VB

Since strings are very large in VB6 (upto 2Gb) and strings can be passed back and forth between VB and Flash using FSCCommand and SetVariable, all data interaction between VB and Flash can be achieved easily.

However, some design decisions need to be made:

- Get records in VB and pass one by one, or pass as one string and parse in Flash?
- How much data do you hold in Flash? If you hold too little, you'll have to ask VB frequently for updated data. If you hold too much, you're not being nice!

Most design decisions refer to granularity, and it is important to decide on these before embarking on programming.

Another way of data-exchange would be through XML, as I explain below.

Use of XML

XML files can be used as data source for Flash. Flash exposes an XML object that can be used to load the XML file using their relative/absolute path. All the methods to retrieve the attributes/node values are exposed by the XML object.

Following snippet shows how to load an XML file kept in the same folder as the swf file.

```
treeXML = new XML();           // variable treeXML declared and initialized
treeXML.onLoad = buildTree;    // buildTree function will be called on the load of      xml file
treeXML.load("XMLFile.xml");  // load xml file named XMLFile.xml into the object treeXML.
function buildTree(success) {
    // if success is true then xml loaded properly do your thing here after the xml is loaded
}
```

treeXML is of type XML as declared in Flash. It has its own methods and properties, which can be looked into from Flash help.

For programs where XML is a predominant feature, it would be most convenient to use XML files for data-interchange. Especially complex user-interfaces which are data-driven can be easily driven through the XML parse mechanism.

Passing notifications from VB to Flash

Invoking VB from Flash is easy by using the fscommand ActionScript call. To notify Flash from VB, we have no direct mechanism.

Suppose Flash calls VB to do something using the FSCCommand, and proceeds after VB has done some processing in the FSCCommand. In this case, VB needs to tell Flash when processing is over. Since VB can't call Flash, we will use the data-passing mechanism (SetVariable) to simulate function calls.



Flash and VB: A potent combination

We achieve this by having an ActionScript variable and waiting on it. Flash loops on this variable waiting for VB to set the variable, and proceeds execution after VB has put in a value there.

For example, put a textbox on the Flash frame. The initial value of a textbox is 'undefined'. We made use of this property of textbox to wait on the timer. Then we called the VB fscommand to carry out our operation there. When VB returns, it sets that variable we are waiting on. Then we clear the interval and proceed in Flash. Until this variable is set, Flash keeps on looping in that function.

Example:

Create a textbox with Var name: VBCalling

```
// this function in flash makes a VB call and waits until the variable is set to some value from VB
function CallVBFunction() {
    fscommand("Test", "Test"); // function call to VB
    keepWaitingForResponse(); // call the flash function
}

// this function doesn't go ahead until the VBCalling is set from VB
function keepWaitingForResponse() {
    intervalID = setInterval(
        function () { // this function is called ever 100 ms until the intervalID is cleared
            if (VBCalling != undefined) {
                VariableReturned(VBCalling);
            }
        }, 100);
}

// this function is called only after the variable is set from VB
function VariableReturned(value) {
    clearInterval(intervalID); // clear the interval and stop the looping
    // you can use the value if you want to and do your thing here
}
```

VB/Flash issues in a real-life project

VB crashed while adding the Flash component on some machines

We faced a problem that on some machines when we added the Flash component onto a VB form, VB crashed giving a memory reference error. After trying various sorts of things and not been able to find a proper solution, we thought of downloading the latest Flash version. Bingo!!! It worked.

If you have the default Flash MX as shipped by Dreamweaver, it will not work in VB. You need to update it to the first service release. The updated Flash.ocx comes along with the latest download of Flash viewer.



Passing values from one Flash frame to another

Sometimes we needed to use the values set in one frame of Flash in another frame. We thought of using the textbox itself to pass this shared information. We defined a textbox of same name and of 0 height/width/fontsize, in all the frames which needed to share some value. We can set the textbox in one frame and use it in others if it is defined there too. The values can be passed this way.

Error in Flash for scripts, which execute for long

This error creeps in when some script is running in Flash for long time and is taking up the system resources (error: “a script in this movie is causing Flash player to run slowly”).

In our project we load an XML file to create a tree structure to display various subjects/lessons/etc as a menu in Flash. There are various functions called recursively, hence they were in a loop for a longer time than Flash can handle. Due to this we were getting this error message from Flash indicating that the script is causing instability. We had to somehow remove this message.

To solve this problem, we put some delay in the functions which were called recursively. This would free the system resources for some time and prevent Flash from popping up the warning message.

```
childLoaded = true;           // state of this variable is checked inside function

// in this function, the recursive calls are made
function foo() {

// based on previous conditions, setBranches() is called recursively from within
// foo(), so we put a delaying algorithm around this function call, which gives time
// to flash to free resources and not produce the warning message.

// setBranches(); // don't use this

intervalID = setInterval( function () {
if (childLoaded) { // we check for this variable, if true then we call setBranches(), otherwise we delay for 50
milliseconds and then this variable gets set to true again.
setBranches();
}
}, 50);
childLoaded = true;
}

// recursive function
function setBranches() {
childLoaded = false; //set variable to false, so that next time this function is being called, it gets a delay of 50
ms
clearInterval(intervalID); // clear the interval
// continue doing your thing here
}
```



Passing key events to Flash from VB form containing other controls

In some cases we had a VB form, which was displayed on top of the Flash screen (which was also loaded in another VB form). We had to pass the key events to Flash from the superceding VB form. This could be done using the KeyDown/KeyUp/etc functions exposed by VB form.

The problem we faced was that, our VB form contained other components like webbrowser and Flash component, which capture all the key events and don't allow them to be passed further.

So we used a workaround for this. We kept a textbox in that VB form and set its bounds such that it wasn't visible on our screen. A textbox has the key handling events, where we can handle them in our way. Now we just had to set the focus on the textbox every time the webbrowser/Flash component was in focus.

This was done using the GotFocus event in these components. So for both these components, whenever this event was called, we forced the SetFocus of TextBox hence all the key events were passed via textbox to our form. From there we passed them to Flash using the SetVariable() method talked about previously. We have to set the KeyPreview for the Form as true in the form load to capture the key events.

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
DoKeyUp (KeyCode)      ' function which passes the key code to the flash using the set variable
End Sub
```

```
Private Sub Form_Load()
Me.KeyPreview = True   ' set KeyPreview True, so that form can capture all the key events
End Sub
```

```
' whenever the webbrowser gets focus, this event is called
Private Sub wb_GotFocus()
    Me.Text1.SetFocus      ' we set the focus forcibly to text field 'Text1'
End Sub
```

For Flash to capture these keys, we use the same algorithm to wait on a variable, which can be set to the value of keycode and captured in flash. This was an infinite loop without clearing interval.

```
// define a textfield with Var name as KeyCode

// this function always keeps waiting to capture the keycode from VB
function keepWaitingForKeyResponse() {

    intervalID = setInterval(function () {
        if (KeyCode != undefined) {           // keycode is set from VB
            KeyPressed(KeyCode);             // function call when key press is passed to Flash
            KeyCode= undefined;              // we set the variable to undefined again to wait again
        }
    }, 100);
}

// function which handles key events
```



```
function KeyPressed(key) {  
    // do your thing here  
}
```

Conclusion

In one of my first jobs, we spent days investigating how to make buttons of different shapes for a consumer application. When we did implement it, it was lines and lines (or was it miles and miles) of C++ code. If I had read this article, I would have put an amoeba-shaped button into my VB form in 2 seconds. Why I would want amoeba-shaped buttons is another discussion altogether ☺

While this article has talked about programming aspects, the market-savvy amongst you must already be envisioning the possibilities opened up by a marriage of VB and Flash. Flash does one thing and it does it very well: User interfaces. VB does one thing and does it very well: programmer productivity. If you merge these, you have a potent combination.

Flash user interfaces improve your product perception, they keep the user hooked, they set you apart from the competition and they keep your market alive. A sense of momentum is important for every software application's user base, and rejuvenating the user interface can generate more enthusiasm than may be immediately apparent.

What are you waiting for?

Mindfire Solutions is an offshore software services company in India. Mindfire possesses expertise in multiple platforms, and has built a strong track record of delivery. Mindfire passionately believes in the power of integrating a Flash UI in desktop programs and its many advantages for software product companies.

If you want to explore the potential of this idea, please drop us an email at info@mindfiresolutions.com. We will be glad to help you.

To know more about Mindfire Solutions, please visit us on www.mindfiresolutions.com
