

1. Giới thiệu

Nếu bạn đã quen sử dụng JavaScript rồi thì bạn sẽ rất vui khi biết rằng JavaScript có thể liên lạc được với ActionScript. Tuy nhiên, cách này không hoạt động tốt đối với tất cả các loại trình duyệt.

Việc liên lạc này được xây dựng trên 2 công nghệ. Một là công nghệ LiveConnect được xây dựng trong những phiên bản trình duyệt Netscape trước phiên bản 6.0. Công nghệ thứ hai là ActiveX dùng để liên lạc giữa Flash và Internet Explorer. Nhưng nếu bạn đang thiết kế cho người dùng sử dụng trình duyệt của Windows thì cách này sẽ rất tốt.

Gửi thông điệp đến JavaScript

Gửi thông điệp từ ActionScript đến JavaScript thì chỉ cần viết code trong ActionScript nhưng nó cũng sẽ thay đổi nội dung trang HTML của bạn .

Nếu bạn tạo file Flash bằng cách chọn FSCommand trong Publish settings, bạn sẽ tạo được một file HTML đã được sửa chữa đầy đủ để nhận thông điệp. Công việc của bạn chỉ là thay thế những chỗ được đánh dấu Your code here bằng phần JavaScript của bạn.

Để hiểu rõ hơn cách nó làm việc như thế nào thì chúng ta hãy cùng nhau mở xê file html mà Flash đã tạo ra nhé

Đầu tiên, hãy chú ý đến thẻ OBJECT/EMBED, trong đó sẽ có một số phần để chấp nhận sự liên lạc. Tham số ID trong thẻ OBJECT sẽ giống với tham số NAME trong thẻ EMBED. Hai tham số này sẽ đặt tên cho movie của chúng ta trong trang web để JavaScript có thể gọi nó. Ngoài ra còn có một tham số khác trong thẻ EMBED, đó là tham số swLiveConnect để cho phép những phiên bản Netscape trước phiên bản 6.0 có thể liên lạc với Flash bằng công nghệ LiveConnect.

CODE

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/
[ic:cc]flash/swflash.cab#version=5,0,0,0"
ID=flashmovie WIDTH=120 HEIGHT=120>
<PARAM NAME=movie VALUE="17astojis.swf"> <PARAM NAME=quality
VALUE=high> <PARAM
NAME=bgcolor VALUE=#FFFFFF> <EMBED src="17astojis.swf" quality=high
bgcolor=#FFFFFF
WIDTH=120 HEIGHT=120
swLiveConnect=true NAME=flashmovie TYPE="application/x-shockwave-flash"
PLUGINSOURCE="http:
//www.macromedia.com/shockwave/download/
index.cgi?P1_Prod_Version=ShockwaveFlash"></
EMBED>
```

</OBJECT>

Trước đoạn code của thẻ OBJECT/EMBED sẽ có một đoạn script. Phần đầu tiên là một hàm JavaScript với tên của movie ID trong thẻ OBJECT và nối tiếp với

_DoFSCommand. Như ví dụ ở trên thì tên hàm của chúng ta sẽ là flashmovie_DoFSCommand.

Trong hàm này thì chúng ta sẽ truyền dữ liệu lại cho movie. Không may là Netscape và Internet Explorer nhìn nhận movie của chúng ta khác nhau. Internet Explorer nhìn movie với tên truyền vào tham số ID (flashmovie), còn Netscape nhìn movie với tên là document.flashmovie. Hãy xem đoạn code dưới đây xem nhé:

CODE

```
<script LANGUAGE=JavaScript>
function flashmovie_DoFSCommand(command, args) {
  if (navigator.appName.indexOf("Microsoft") != -1) {
    var flashmovieObj = flashmovie;
  } else {
    var flashmovieObj = document.flashmovie;
  }

  alert(command); alert(args);
}
```

Trong Internet Explorer, Flash không chỉ có thể liên lạc với JavaScript mà còn có thể liên lạc với VBScript. Đoạn code dưới đây được viết bằng JavaScript thay cho đoạn JavaScript trên:

CODE

```
if (navigator.appName && navigator.appName.indexOf("Microsoft") != -1 &&
navigator.userAgent.indexOf("Windows") != -1 &&
navigator.userAgent.indexOf("Windows 3.1") == -1) {
  document.write('<script LANGUAGE=VBScript> \n'); document.write('on
error resume next \n');
  document.write('Sub flashmovie_FSCCommand(ByVal command, ByVal args)\n');
  document.write('call flashmovie_DoFSCommand(command, args)\n');
  document.write('end sub\n'); document.write('</SCRIPT> \n');
}
```

</SCRIPT>

Vậy còn phần ActionScript phải làm sao đây. Xin trả lời là chỉ cần một dòng duy nhất mà thôi

CODE

```
fscommand ("alert", "This is alert 1.");
```

Nhận thông điệp từ JavaScript

Nhận thông điệp từ JavaScript thì có vẻ dễ hơn. Nhưng bạn phải nhớ rằng đã đặt tham số ID trong thẻ OBJECT và tham số NAME trong thẻ EMBED, nhớ đặt giống tên nhé. Và cũng nhớ đặt swLiveConnect=true trong thẻ EMBED.

Bây giờ thì đã sẵn sàng để truyền thông điệp từ JavaScript cho movie rồi. Hãy xem ví dụ này nhé, trong ví dụ này thì chúng ta sẽ sử dụng hàm gotoFrame để di chuyển movie sang frame thứ hai.

CODE

```
<FORM NAME="flashControlForm">  
<INPUT NAME="gotoFrame1" TYPE=Button VALUE="Frame 1"  
onClick="window.document.flashmovie.GotoFrame(1);"> </FORM>
```

Chắc chắn đến đây các bạn đang tự hỏi tại sao di chuyển sang frame thứ hai mà lại sử dụng gotoFrame(1), đúng không nào? Bởi vì hệ đếm của chúng ta bắt đầu từ số 0

(zero-based). Vì vậy, frame 1 sẽ là 0, frame 2 sẽ là 1, frame 3 sẽ là 2...

Có hơn 24 câu lệnh trong Flash movie. Tuy nhiên chúng ta không cần phải dành nhiều thời gian để tìm hiểu bởi vì như đã nói ở trên thì cách liên lạc này không phải hoạt động tốt đối với tất cả các loại trình duyệt.

Bạn có thể sử dụng hàm GetVariable và SetVariable để điều khiển những biến trong timeline của movie. Câu lệnh Zoom dùng để kéo dẫn movie. Hai hàm isPlaying và percentLoaded dùng để kiểm tra movie nào đang hoạt động. Play dùng để play một movie khi nó đang ngừng.

2. Luyện tập: Mở window mới

Có lẽ đây là một yêu cầu rất thường gặp đối với những người sử dụng Flash. Bạn có thể làm được điều này bằng cách sử dụng hàm `getURL` hoặc JavaScript.

Tuy nhiên, sử dụng JavaScript sẽ có rất nhiều đặc điểm mới và hay hơn, bạn có thể đặt lại những thuộc tính của window. Nào, cùng làm thử nhé:

- Tạo một movie mới.
- Đặt vào movie một button.
- Chèn đoạn code sau cho button vừa tạo

CODE

```
on (release) {  
    fscommand ("newwindow", "content.html");  
}
```

- Trong phần Publish Settings, chọn để publish ra file HTML cùng với Flash movie. Trong phần HTML, nhớ chọn vào mục Flash with FSCommand -

Publish movie của bạn.

- Mở file HTML mà Flash vừa tạo bằng một trình soạn thảo nào cũng được, như NotePad chẳng hạn. Hãy tìm phần để chèn JavaScript, và chèn đoạn code sau vào:

CODE

```
if (command == "newwindow") {  
    window.open(args,"","width=320,height=240,location=no,toolbar=no,  
    menubar=no");  
}
```

- Tạo thêm một file HTML đơn giản nữa và đặt tên là `content.html`

- Mở file HTML của bạn ra trong một trình duyệt và nhớ là trình duyệt đó phải hỗ trợ JavaScript đó nha. Khi nhấn vào button thì một cửa sổ mới sẽ xuất hiện, không có toolbar, kích thước 320x240.
- Vậy còn đối với những trình duyệt không hỗ trợ JavaScript thì sao? Đơn giản thôi, sử dụng  getURL . Điều quan trọng là cần phải cho Flash biết lúc nào sử dụng Flash, lúc nào sử dụng JavaScript.

Quay trở lại file HTML mà Flash đã tạo, chèn đoạn code JavaScript sau vào cuối đoạn code ta đã chèn vào lúc trước

CODE

```
function initComm() {
    window.document.newWindowMovie.SetVariable("jsCommOK","OK");
}
```

Đoạn code trên sẽ thử đặt giá trị cho biến jsCommOK là OK. Nếu trình duyệt có hỗ trợ JavaScript thì jsCommOK sẽ mang giá trị OK còn nếu không thì biến jsCommOK vẫn là undefined.

- Để chạy hàm initComm đầu tiên thì bạn sửa lại phần BODY của trang HTML như sau:

CODE

```
<BODY bgcolor="#FFFFFF" onLoad="initComm();">
```

Điều này có nghĩa là hàm initComm sẽ được gọi ngay sau khi trang này load xong. - Bây giờ hãy quay trở lại file Flash của bạn, sửa đoạn script của button thành:

CODE

```
on (release) { if (jsCommOK
== "OK") {
    fscommand ("newwindow", "content.html");
} else {
    getURL ("content.html", "_blank");
}
```

}

3. Luyện tập: Tạo một SlideShow được điều khiển bởi JavaScript

Trong phần này, chúng ta sẽ cùng nhau làm một file Flash không có ActionScript mà sẽ được điều khiển bằng JavaScript.

- Tạo một movie Flash mới với 3 frame hoặc nhiều hơn. Nội dung của mỗi frame thì không quan trọng, bạn muốn để gì trong đó cũng được nhưng một lời khuyên là mỗi frame nên khác nhau để chúng ta theo dõi sự thay đổi 😊 - Đặt câu lệnh stop() vào frame đầu tiên
- Publish ra file HTML
- Mở file HTML đó ra trong trình soạn thảo văn bản
- Truyền tham số ID trong thẻ OBJECT và NAME trong thẻ EMBED, cả hai sẽ mang giá trị là slideshow
- Nhớ đặt swLiveConnect=true trong thẻ EMBED - Tạo 2 button trong file HTML bằng đoạn code sau:

CODE

```
<FORM NAME="flashControlForm">
<INPUT NAME="next" TYPE=Button VALUE="Next" onClick="nextFrame();">
<INPUT NAME="prev" TYPE=Button VALUE="Previous"
onClick="prevFrame();"> </FORM>
```

- Mỗi button sẽ gọi một hàm. Cả hai hàm sẽ sử dụng TcurrentFame("/")biết đang ở frame thứ mấy, rồi sử dụng gotoFrame để di chuyển tới hoặc lui.

CODE

```

<script LANGUAGE="JavaScript"> function
nextFrame() {
    var frameNum = window.document.slideshow.TCurrentFrame("/");
    window.document.slideshow.GotoFrame(frameNum+1);
}
function prevFrame() {
    var frameNum = window.document.slideshow.TCurrentFrame("/");
    window.document.slideshow.GotoFrame(frameNum-1);
}
</SCRIPT>

```

Công việc tiếp theo bạn còn nhớ không: CHẠY THỬ đi nào

4. Gửi thông tin cho máy chủ

Cho đến tận bây giờ, những movie chúng ta làm được hầu hết là chỉ chạy một mình

(stand alone). Có nghĩa là nó chỉ chạy ở một máy khách (client-side) không có sự liên lạc với máy chủ (Server). Nên nhớ rằng Flash cũng có thể gửi trả thông tin cho Server giống như form của HTML. Vậy thì trong giờ thứ 18 này, chúng ta sẽ tìm hiểu về vấn đề này.

Trong phần này chúng ta sẽ học:

- Tìm hiểu về đối tượng LoadVars
- Tạo một chương trình server-side đơn giản
- Sử dụng Flash để gửi dữ liệu về cho Server

Đối tượng LoadVars

Đối tượng LoadVars bao gồm một tập hợp các câu lệnh và những biến đặc biệt để gửi dữ liệu cho Server giống như post form trong HTML. Chúng ta có thể tạo đối tượng giống như tạo những đối tượng khác.

Lấy dữ liệu

Dưới đây là một ví dụ. Thay vì sử dụng new Object() bằng new LoadVars. Sau đó thì đối tượng LoadVars mới sẽ được tạo.

CODE

```
myVars = new LoadVars();
```

Với đối tượng LoadVars, chúng ta có thể làm được 2 việc, đó là gọi và lấy dữ liệu. Để lấy dữ liệu, chúng ta sử dụng câu lệnh load. Cái chúng ta cần ở đây là một đường dẫn đến nơi chứa dữ liệu:

CODE

```
myVars.load("myURL.txt");
```

Ví dụ trên hoạt động giống như câu lệnh LoadVariables trong giờ thứ 10. Tuy nhiên, câu lệnh LoadVariables chỉ lấy và thay thế dữ liệu trong cùng một level với câu lệnh này mà thôi. Hãy tưởng tượng, chúng ta có một file chứa dữ liệu như sau:

CODE

```
name=George&ID=47
```

Gửi dữ liệu

Với đối tượng LoadVars, chúng ta cũng có thể gửi dữ liệu đi lên Server. Trước hết, chúng ta đưa dữ liệu vào đối tượng LoadVars, rồi sau đó sử dụng câu lệnh send để gửi dữ liệu đi.

CODE

```
myVars = new LoadVars(); myVars.name  
= "George"; myVars.ID = 47;  
myVars.send("serverprogram.cgi", "_self");
```

Đoạn code trên sẽ tạo một đối tượng LoadVars mới tên là myVars, rồi đưa hai thuộc tính vào đối tượng myVars. Sau đó, gửi dữ liệu đi lên Server, chính xác là đến chương trình CGI tên là echo.cgi.

Câu lệnh send cũng hoạt động tương tự như trong form của HTML. _self là Target, chúng ta có thể thay thế bằng các giá trị khác như đã được biết.

Nhưng còn một câu lệnh nữa là sendAndLoad. Câu lệnh này được ghép câu lệnh send và load. Có nghĩa là đối tượng LoadVars sẽ gửi dữ liệu lên Server, sau đó sẽ lấy giá trị trả về.

CODE

```
mySendVars = new LoadVars(); myLoadVars =  
new LoadVars(); mySendVars.name = "George";  
mySendVars.ID = 47;  
mySendVars.sendAndLoad("serverprogram.cgi", myLoadVars);
```


Trạng thái lấy dữ liệu

Hãy nhớ là câu lệnh send và load sẽ không lấy dữ liệu ngay tức thì. Có thể phải đợi trong một thời gian ngắn hoặc dài. Vì vậy, chúng ta sẽ có nhu cầu theo dõi trạng thái lấy dữ liệu. Chúng ta không thể sử dụng dữ liệu ngay lập tức sau câu lệnh load hoặc sendAndLoad, mà cần phải kiểm tra xem đã lấy dữ liệu xong chưa.

Cách đơn giản nhất là dùng một movie clip lặp đi lặp lại để kiểm tra. Cũng có thể sử dụng getBytesLoaded và getBytesTotal với những dữ liệu lớn. Dưới đây là ví dụ để kiểm tra việc lấy dữ liệu:

CODE

```
myLoadVars.onLoad = function(success) {  
  if (success) {  
    gotoAndStop("load done");  
  } else {  
    gotoAndStop("load problem");  
  }  
}
```

Như trong đoạn code trên thì đối số success sẽ nhận một trong hai giá trị true hoặc false để cho biết việc nhận dữ liệu đã xong hay chưa

Bài 7. Dùng XML với Flash

Càng ngày XML càng trở nên thông dụng trong các ứng dụng tin học, và ngay cả trong Flash, vậy trong phần này mình sẽ học các điều sau:

- XML căn bản
- Đối tượng XML trong Flash
- Phân tích và xử lý XML theo phương pháp đệ quy (recursive)

1. XML Căn bản

XML chỉ đơn giản là một văn bản để chứa dữ liệu. XML tương tự như HTML, cũng dùng các thẻ. Tuy nhiên, XML khác ở HTML là các thẻ HTML đã được ấn định trước cho các chức năng khác nhau, còn XML thì không.

Với XML, bạn có thể tự tạo cho mình các thẻ theo ý bạn để phù hợp cho mục đích riêng của bạn.

Có thể tạo một XML file với 1 trình biên tập văn bản (như Notepad, Textpad ...) đơn giản hay là các software chuyên để viết về XML (XMLSpy, Epic ...)

Với Flash MX, bạn có thể dễ dàng truy cập được dữ liệu trữ ở trong XML, và đối tượng XML của Flash sẽ tự động phân tích văn bản XML này.

Trong VNFX có rất nhiều bài viết về XML ([ví dụ như bài giới thiệu XML của Flash-](#)

[Lee](#)) các bạn có thể tham khảo thêm

1.1 Đối tượng XML

Đối tượng XML trong Flash gồm có nhiều hàm và đặc tính dùng để giúp bạn lấy và phân tích dữ kiện trong XML file một cách dễ dàng. Bước đầu tiên khi dùng đối tượng XML là tạo một phiên bản XML trước:

CODE

```
myXML = new XML()
```

1.2 Phân tích và xử lý văn bản thành XML

Sau khi tạo phiên bản XML trên, hiện giờ phiên bản này vẫn chưa có gì cả, tuy nhiên bạn có thể nhanh chóng tạo ra một tài liệu XML bằng cách dùng lệnh **parseXML**. Lệnh này sẽ nhập một chuỗi văn bản, và phân tích và xử lý nó thành 1 tài liệu XML.

```
CODE myXML = new
XML();
myXML.parseXML("<user><name>Gary</name><ID>47</ID></user>");
```

Hay bạn có thể viết trực tiếp như sau

```
CODE
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
```

Nếu mà chuỗi được nhập vào không thể tạo thành XML hoàn chỉnh thì bạn có thể dùng đặc tính **status** để kiểm tra.

```
CODE
myXML = new XML("<user><name>Gary</name><ID>47</user>");
trace(myXML.status);
```

Output window sẽ cho ra -9 khi chạy đoạn code trên. -9 có nghĩa là thiếu thẻ đóng (end tag), vì mình thiếu thẻ **</ID>**. Nếu output window cho ra **-10** thì bạn thiếu thẻ mở (start tag), và **0** có nghĩa là mọi việc hoàn chỉnh

1.3 Lấy dữ liệu từ đối tượng XML

Có nhiều hàm để làm việc này. Ví dụ, bạn có thể dùng **firstChild** để lấy nút (node) đầu tiên của đối tượng XML:

```
CODE
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.firstChild);
```

Output window sẽ cho ra `<user><name>Gary</name><ID>47</ID></user>`. Vì đây chính là nút đầu tiên của đối tượng XML. Nếu chúng ta đi sâu vào 1 lần nữa với **firstChild**

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.firstChild.firstChild);
```

Kỳ này output window sẽ cho ra `<name>Gary</name>`, vì phần tử **name** là nút đầu tiên của **user**

Ngoài cách trên ra, còn 1 cách nữa là dùng `childNodes`, là ma trận của của các nút. Cùng một ví dụ trên nhưng có thể dùng như sau với `childNodes`

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[0]);
```

Dưới nút **name**, chúng ta còn một nút nữa dưới nút **name**, và có thể dùng cách trên để lấy nút đó:

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[0].childNodes[0]);
```

và output window sẽ cho ra chữ **Gary**. Đây là nút văn bản (text node) của nút **name**. Nhìn thì tưởng đây là tận cùng rồi, không thể xuống sâu hơn được nữa, vì đây là nút cuối cùng, nhưng nếu bạn muốn lấy giá trị của nó như là chuỗi văn bản thì bạn có thể đi thêm 1 bước nữa như sau:

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[0].childNodes[0].nodeValue);
```

Nếu bạn muốn lấy cái **ID** ở trong nút thứ 2 thì thay đổi như sau:

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[1].childNodes[0].nodeValue);
```

* Để giải thích thêm về đoạn code trên, nếu bạn chỉ dùng `childNodes` thì kết quả sẽ là 1 đối tượng, còn nếu bạn dùng `nodeValue` thì sẽ ra chuỗi. Bạn có thể thử bằng đoạn code mình viết dưới đây

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
myVar1 = myXML.childNodes[0].childNodes[0].childNodes[0]; myVar2 =
myXML.childNodes[0].childNodes[0].childNodes[0].nodeValue;      trace
("myVar1: " +typeof(myVar1)); trace ("myVar2: " +typeof(myVar2));
```

và output window sẽ cho bạn thấy rõ, cái nào là đối tượng và cái nào là chuỗi.

Tự tạo XML từ "tay trắng"

đKhông biành dịch hết dịch câu này sao cho trên vậy. đúng nghĩa với "Creating XML from Scratch".

Các cách tạo XML ở trên đều từ 1 chuỗi văn bản, nhưng nếu như bạn không có 1 chuỗi văn bản nào trước thì sao? Bạn có thể dùng **createElement** để tạo ra các nút mới và dùng **createTextNode** để tạo ra các text node. Tuy nhiên 2 lệnh trên chỉ tạo ra nút chứ không thêm vào trong XML, vậy bạn phải thêm nó vào bằng lệnh **appendChild**.

Để minh họa, chúng ta sẽ tạo một XML tương tự như trên nhưng với các lệnh vừa đề cập.

CODE myXML = new

```
XML();
newElement = myXML.createElement("user");
myXML.appendChild(newElement); newElement =
myXML.createElement("name");
myXML.childNodes[0].appendChild(newElement); newText =
myXML.createTextNode("Gary");
myXML.childNodes[0].childNodes[0].appendChild(newText);
newElement = myXML.createElement("ID");
myXML.childNodes[0].appendChild(newElement); newText =
myXML.createTextNode("47");
myXML.childNodes[0].childNodes[1].appendChild(newText);
```

Nếu bạn muốn thay đổi giá trị của text node, thì bạn chỉ cần dùng `nodeValue`

CODE

```
myXML.childNodes[0].childNodes[1].childNodes[0].nodeValue = 53;
```

1.4 Thuộc tính

Thuộc tính (Attribute) gồm có từ khoá và giá trị của từ khoá đó, và dùng để định rõ một thành phần hơn. Ví dụ, đoạn XML dưới đây với attribute **type** làm rõ nghĩa của thành phần **name** hơn, ("alias" là bí danh)

CODE

```
<user>  
  <name type="alias">Gary</name>  
  <ID>47</ID> </user>
```

Nếu bạn muốn đưa đoạn code trên vào trong XML object thì phải đổi dấu " (ngoặc kép) thành dấu ' (ngoặc đơn)

CODE

```
trace(myXML.childNodes[0].childNodes[0].attributes.type);
```

Còn một cách nữa cũng cho ra kết quả như trên là dùng []

CODE

```
trace(myXML.childNodes[0].childNodes[0].attributes["type"]);
```

Và bạn có thể thay đổi giá trị của attribute, hay thêm attribute mới như sau:

CODE

```
myXML.childNodes[0].childNodes[0].attributes["type"] = "real";
```

Với câu trên, nếu attribute type chưa có thì nó sẽ được tạo

(trong sách các đoạn code trên tác giả viết lộn giữa type và alias, nếu bạn so sánh giữa sách và bài này thì sẽ thấy sự khác biệt, và có thể các source file cũng sẽ không chính xác, vậy các bạn cần kiểm tra lại)

Khác với node, attribute không thể truy cập bằng ma trận (array) nên bạn không thể dùng các lệnh như **length** hay dùng **[]** với index number. Nhưng bạn có thể dùng vòng lặp **for ... in** để truy cập từng attribute của node.

CODE

```
myXML = new XML("<user><name type='alias' validity='verified'>Gary</name><ID>47</user>");
for(attr in myXML.childNodes[0].childNodes[0].attributes) {
    trace(attr+": "+myXML.childNodes[0].childNodes[0].attributes[attr]); }
```

output window sẽ cho ra: **alias** và **verified**

2. Thêm vài AS về XML

Bạn cũng cần biết thêm một số điều về đối tượng XML. Nhất là điểm sau, bất cứ một phần nào của đối tượng XML cũng có thể là một đối tượng XML riêng biệt. Ví dụ bạn có thể lấy cái **node** đầu tiên của đối tượng XML và quy thành một biến

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</user>");
thisUser = myXML.childNodes[0];    thisUserName =
thisUser.childNodes[0];
thisUserNameText = thisUserName.childNodes[0].nodeValue; thisUserID
= thisUser.childNodes[1];
thisUserIDText = thisUserID.childNodes[0].nodeValue;
```

Bạn có thể biết được số **node** ở trong ở trong một **node** khác bằng thuộc tính **length** của **childNodes**. Ví dụ, **user** node có 2 node ở trong, vậy bạn có thể biết được bằng với đoạn code sau:

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</user>");
trace(myXML.childNodes[0].childNodes.length);
```

Ngoài việc có thể tìm được giá trị của một node văn bản bằng **nodeValue**, bạn có thể biết được tên của thành phần (hay thẻ) với **nodeName**. Ví dụ bạn có thể lấy được tên của thành phần (thẻ) đầu tiên của **user** là **name** như sau:

CODE

```
myXML = new XML("<user><name>Gary</name><ID>47</user>");  
trace(myXML.childNodes[0].childNodes[0].nodeName);
```

Bạn có thể biết được một node là thành phần của XML hay là một node văn bản (text node) bằng với thuộc tính **nodeType**. Nếu **nodeType** là 1 thì có là 1 thành phần của XML và có thể có nhiều thành phần ở trong node đó, nếu là 3 thì node đó chính là node văn bản.

Printing

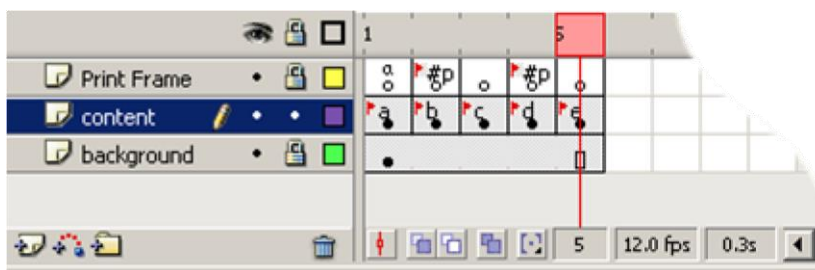
Không phức tạp như trình duyệt hoặc Server Communication, in ấn là một cách mà FLash truyền dữ liệu ra ngoài.

Khả năng in ấn của FLash là rất hữu ích vì nó cho phép chúng ta xây dựng những văn bản mà người sử dụng có thể in ra nội dung từ file flash của bạn. Đây thường là lựa chọn tốt hơn dựa vào hàm in ấn của trình duyệt. Trong phần này chúng ta sẽ :

- Học cách thiết lập movie của bạn cho việc in ấn
- Học cách sử dụng các lệnh in
- Tạo ra một biểu mẫu có thể in được

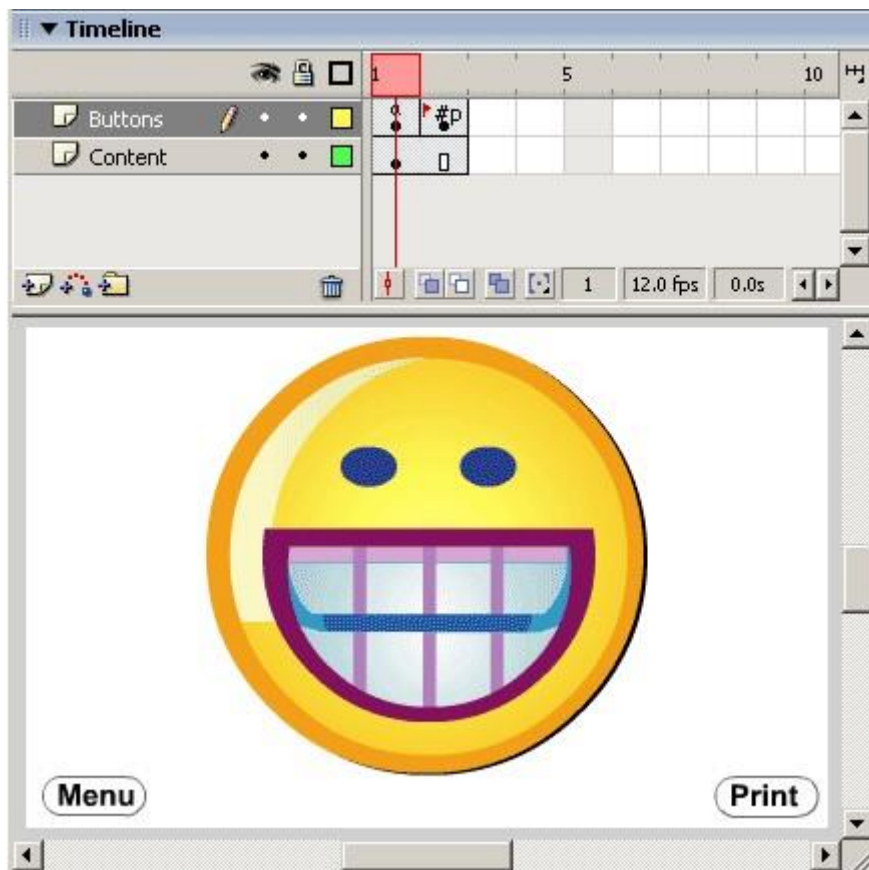
2 - Đặt tên cho frame

Bạn sẽ chỉ định những frame được in bằng ký hiệu "#p". Nếu ký hiệu đó không được đặt lên frame nào thì Flash sẽ in ra toàn bộ movie của bạn. Ngược lại, nếu bạn đặt từ hai ký hiệu trở lên trong movie của bạn thì flash sẽ in ra tất cả những frame đó. Hình 1 cho ta thấy timeline với hai frame đã được đặt ký hiệu in. Chú ý rằng, tất cả những frame có chứa ký hiệu in này sẽ được đặt ở một layer riêng. Theo cách đó, bạn có thể đặt được nhiều ký hiệu trên nhiều frame. Và nó cũng thuận tiện hơn khi bạn không muốn sử dụng ký hiệu "#p" như một label thật cho keyframe.



*Chú ý: Khi bạn đặt từ hai ký hiệu in trở lên, khi test movie, Flash sẽ xuất hiện lời cảnh báo : "WARNING : Duplicate layer.."

Đó là điều rủi ro có thể xảy ra, nhưng cũng sẽ không ảnh hưởng đến movie của bạn trừ phi bạn sử dụng label "#p" trong câu lệnh "gotoAndStop" hoặc những lệnh tương tự....



3- Lên kế hoạch in

Bạn cần phải suy nghĩ và lên kế hoạch cho movie của bạn khi muốn movie có khả năng in. Vì không thể chỉ in một frame hiện tại nên bạn

cần tạo ra những frame có khả năng in. Chú ý rằng mọi thứ tồn tại trên movie của bạn sẽ được in ra. Nó bao gồm cả nút Print (để khi nhấn vào sẽ thực hiện lệnh in), nếu trong movie có

tồn tại.

Vì vậy chúng ta cần phải có một frame chứa nút Print, và một frame khác tương tự nhưng không có nút Print cũng như các yếu tố không cần thiết khác. Frame thứ hai này sẽ được đặt nhãn là "#p"

Bạn có thể bố trí bằng cách sử dụng timeline chính. Ví dụ bạn có một menu và một nút Print trên hai layer riêng biệt, mà cần có

trong nội dung của biểu mẫu in. Những layer này nếu bạn không muốn in ra thì không cần kéo chúng sang frame có nhãn "#p" hai hình sau thể hiện điều này. Hình thứ nhất, là frame mà người dùng sẽ thấy xuất hiện trên trang web. Nó chứa nội dung, nút Print và menu

Ngược lại, trong hình hai là nội dung mà người duyệt web không thấy được, ít nhất là trên màn hình. Frame này được đặt tên là "#p"

bạn có thể thấy trên timeline. Layer Buttons không sử dụng keyframe giống như ở frame 1. Ngoài việc chặn nút Print không được in ra, có thể thêm vào frame này một số thông tin sẽ được in ra như địa chỉ, ...



Bây giờ bạn đã biết làm thế nào để chuẩn bị cho một biểu mẫu in, bây giờ cùng học AS cần thiết

II> Các câu lệnh

Có hai lệnh in chính được sử dụng. Chúng có một khác biệt rất nhỏ, nhưng về căn bản là hoạt động giống nhau.

1- Print

Lệnh thứ nhất là

QUOTE

Print

Lệnh này bao gồm hai tham số. Tham số thứ nhất là đối tượng để in. Thường là timeline chính, hoặc "_root". Bạn cũng có thể sử dụng

"this". Tuy nhiên, nếu bạn không muốn in những frame trong movie, bạn có thể sử dụng tham chiếu tới movie đó.

Tham số thứ hai là một trong ba tùy chọn sau :

QUOTE bframe

QUOTE bmovie
QUOTE bmax

Các tham số này giúp FLash có thể co giãn văn bản in theo ý của người sử dụng. Flash có thể co giãn văn bản in tới kích thước

của trang giấy mà không bóp méo văn bản. Ví dụ nếu frame được in có kích thước

550x400, thì chiều ngang sẽ được phóng to tới kích thước 550px chiều dọc có thể sẽ được scale theo tỷ lệ.

Khi sử dụng tùy chọn "bframe", từng frame sẽ tự scale để lấp đầy kích thước của trang đó. Nếu frame thứ nhất có nội dung với kích thước là 550x400

nhưng frame thứ hai chỉ chứa nội dung có kích thước 275x200, khi đó frame thứ hai sẽ tự động scale gấp đôi kích thước ban đầu. khi sử dụng tùy chọn "bmax", Flash sẽ kiểm tra toàn bộ các frame được in để xác định xem frame nào có kích thước lớn nhất. Các frame

còn lại sẽ scale dựa trên kích thước của frame lớn nhất, tính theo tỷ lệ. Điều này tạo nên các frame có kích thước tỷ lệ với nhau

Ví dụ, frame lớn nhất có kích thước 550x400, và nó lấp đầy trang in. Frame khác chỉ có kích thước 275x200, nó chỉ chiếm một nửa trang in.

Tùy chọn cuối cùng là "bmovie", trường hợp này bạn cần phải làm thêm một việc nhỏ nữa, đó là tạo ra frame mới có chứa một khung.

Khung này sẽ xác định kích thước lớn nhất được in ra đối với văn bản (bằng kích thước của khung). Bạn phải đặt tên frame này với ký hiệu

"#b". Và flash sẽ sử dụng kích thước của khung để scale toàn bộ các frame còn lại. Nếu có một phần văn bản này nằm ngoài khung in chúng sẽ không được in ra. Sau đây là ví dụ về lệnh "Print" :

QUOTE

```
on(release) { print(this,"bframe");  
}
```

Như bạn đã thấy, tùy chọn được coi như một chuỗi và được đặt trong dấu ""

2-PrintAsBitmap

ĐCầu liều khác biệnh này làm việť là lệnh Print sệť tươg tễ gự nhữi các ư lệnh Print đốoi tượg óđoi hai tham s hoặ và font chổ tươg t tóui máy in. Sau . đó, máy in sẽ xâý dặg lại các vector và nội dưng để in ra.

Ngược lại, PrintAsBitmap sẽ chuyển toàn bộ nội dung thành một ảnh bitmap lớn và

Điểm thuận lợi nhất của PrintAsBitmap là độ trong suốt của văn bản cũng được in ra. Nếu bạn có một đối tượng đồ họa bán trong suốt, chắc có ưu điểm chắn bạn sẽ cần đến chức năng in này, đồng thời nó cũng hoạt động tốt với nhiều loại máy in hơn.

Lệnh in chính là văn bản được in ra có tính thẩm mỹ cao, đường cong smooth hơn và chữ rõ ràng hơn khi được in ra. Nó cũng nhanh hơn khi in qua mạng.

Nguyên tắc chung nhất là sử dụng lệnh PrintAsBitmap khi bạn thật sự chắc chắn kết quả sẽ giống như nội dung hiển thị trên màn hình.

Sử dụng Print khi văn bản không cần độ chính xác cao hoặc khi bạn xây dựng movie để điều khiển môi trường như mạng nội bộ.

Chú ý: Nếu bạn sử dụng movie qua internet, lệnh in chỉ làm việc khi tất cả các frame trong movie được load xuống.

Thao tác : Tạo biểu mẫu in

Một điều mà người dùng ghét nhất trên một website là một biểu mẫu mà chúng ta phải in ra, điền vào, rồi gửi đi . Chúng ta đang sử dụng máy tính, đột nhiên lại phải đi kiếm cái bút trên bàn để viết ?

Vì vậy tại sao chúng ta không tạo ra biểu mẫu cho phép điền thông tin vào, sau đó được in ra với đầy đủ nội dung đã được nhập vào.

Chúng ta sẽ làm ví dụ đơn giản này, và sẽ thấy chúng thuận tiện hơn rất nhiều so với một biểu mẫu html yêu cầu người sử dụng in ra toàn bộ nội dung của site.

Bạn hoàn toàn có thể điều khiển những gì sẽ được in ra, vì vậy những thứ linh tinh trên trang web có thể được bỏ qua và những yếu tố mới như email address có thể được thêm vào.

1- Tạo một movie mới

2- Tạo ra các trường nhập dữ liệu (ví dụ tên, tuổi, địa chỉ, giới tính, email...) (xem hình) sau đó đặt tên layer này là CONTENT