

【A6】 Delphiテクニカルセッション

「Delphi+Flashで簡単アニメーション」

株式会社シリアルゲームズ
細川 淳

Adobe Flash とは

Adobe Flash の特徴

- デファクトスタンダードなアニメーション環境
 - インストールベースは98%以上だという
 - Adobe社調べ
http://www.adobe.com/products/player_census/flashplayer/
- ActionScript という制御言語を持つ
 - JavaScript ベースのスクリプト言語
- 米Macromedia 社から米Adobe社へ
 - 元々は FutureWave Software 社の製品

Adobe Flash の扱えるメディア

- 操作可能なメディア形式
 - ベクターグラフィクス
 - png, jpeg, psd などのラスターグラフィクス
 - mp3, wave などの音声
 - flv 形式のビデオ

Adobe Flash プレイヤーと作成環境

- 作成環境
 - 通常は Macromedia Flash 8 を使用する
 - swf はオープンフォーマット
 - フリーの作成環境も存在する
 - ParaFla や Suzuka など
- 再生環境(プレイヤー)
 - Adobe Flash Player 9 が最新



Macromedia Flash 8 Professional



Adobe Flash Player 9

※バージョンなどは共に 2007/06/05 現在

SWF フォーマット

SWF フォーマット (ヘッダ部)

Signature	3byte	FWS(無圧縮) / CWS (圧縮 ver 5 以降)
Version	1byte	SWF のバージョン
FileLength	4byte	ヘッダ部も含んだ全体の長さ
Data	※1	CWS の場合 ZLib 圧縮

※1

Dataのサイズは FileSize - HeaderSize として求める
FileLengthは無圧縮状態のサイズが入っているため使えない

SWF フォーマット (データ部)

※1	NBits	5bit	矩形の値1つを表すために必要なビット
	BoundsRect	NBits*4	ステージの矩形(単位は Twips)
	FrameRate	16bit	フレームレート(※2)
	FrameCount	16bit	全フレーム数
	TagCollection	※3	

※1

BoundsRect の次のアドレスは以下の式で求める
FrameRateAddress = (NBits * 4 + 5 + 7) shr 3

※2

下位8bitは小数点以下を現す
フレームレート=(FrameRate shr 8) + (FrameRate and \$ff) * 0.01;

※3

ここから先は命令タグと、そのパラメータブロックが続く

SWF Header 読み込みサンプル(擬似コード)

```

type
  TSWFHeader = packed record
    Signature: array [0.. 2] of Char;
    Version: Byte;
    FileLength: LongInt;
  end;

var
  FS: TFileStream; // SWFFile を読み込むファイルストリーム
  ZS: TCustomZStream; // ZLib 圧縮から読み出すストリーム
  MS: TMemoryStream; // データ部分を読み込むメモリストリーム
  SWFHeader: TSWFHeader; // SWF Header
  NBits: Integer; // NBits
  Index: Integer; // Data 部のビット単位のインデックス

begin
  FS := TFileStream.Create(SWFFileName, fmOpenRead); // SWFFileName には SWF ファイルのフルパス
  try
    FS.Read(SWFHeader, SizeOf(SWFHeader));

    with SWFHeader do begin
      // Signature
      if (Signature = 'FWS') then
        Compressed := False
      else if (Signature = 'CWS') then
        Compressed := True
      else
        raise EFlashFileError.Create(CErrInvalidFormat);
    end;
  end;

```

SWF Header 読み込みサンプル(擬似コード)

```

// Version
SWFVersion := Version;

// File Size
FileSize := FileLength;
if (Compressed) then
  FileSize := FS.Size;

// Read Data
if (Compressed) and (SWFVersion > 5) then begin
  ZS := TZDecompressionStream.Create(FS);
  try
    while (ReadBuff) do // ReadBuff は MS に解凍後のデータを読み込む関数
      ;
  finally
    ZS.Free;
  end;
else
  MS.CopyFrom(FS, FS.Size - FS.Position);
end;

```

SWF Header 読み込みサンプル(擬似コード)

```
// Data の解析
Index := 0; // GetBits, GetBoundsValue で随時インクリメントされるビット単位のインデックス

NBits := GetBits(MS.Memory, 5, Index); // GetBits は、メモリから指定ビットを読み出す関数
FrameIndex := ((NBits * 4 + 5 + 7) shr 3); // 5 は NBits のビット数

with BoundsRect do begin
  Left := GetBoundsValue; // GetBoundsValue は NBitsで表される bit を読んで返す関数
  Right := GetBoundsValue;
  Top := GetBoundsValue;
  Bottom := GetBoundsValue;
end;

// FrameRate, FrameCount
Move(PByteArray(MS.Memory)[FrameIndex + 0], FrameRate, 2);
Move(PByteArray(MS.Memory)[FrameIndex + 2], FrameCount, 2);

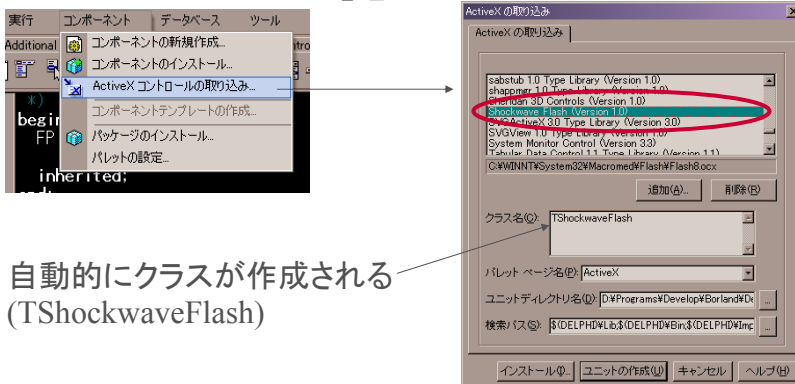
StrictFrameRate := (FrameRate shr 8) + (FrameRate and $ff) * 0.01;
FrameRate := FrameRate shr 8;
finally
  FS.Free;
end;
end;
```

Adobe Flash Player ActiveX

IShockwaveFlash について

Adobe Flash Player ActiveX の取り込み

- ActiveX の取り込みメニューから「ShockwaveFlash」を取り込む



IShockwaveFlash

- IShockwaveFlash インターフェース

- IID
 - D27CDB6E-AE6D-11CF-96B8-444553540000
- 主要なメソッド
 - Play 再生
 - Stop 停止
- 主要なプロパティ
 - Movie 再生する Flash Movie

主要メソッド・プロパティ紹介 Movie/Play/Stop

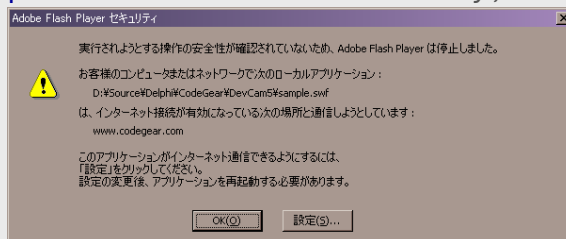
```
// 再生するムービーを指定する [none]
property Movie:WideStrng
    read Get_Movie write Set_Movie;

// ムービーを再生 [play()]
procedure Play; safecall;

// ムービーを停止 [stop()]
procedure Stop; safecall;
```

主要メソッド紹介 DisableLocalSecurity

```
// ローカルにある Movie がネットにアクセスするのを許可する
[none]
procedure DisableLocalSecurity; safecall;
```



DisableLocalSecurity を呼ばないと、このダイアログが現れる

主要メソッド紹介 Zoom/GotoFrame/CurrentFrame

```
// ムービーを拡大表示 [none]
// factor  相対的な拡大率を示す 100が最小(全表示) 1が最大
procedure Zoom(factor: SYSINT); safecall;

// 再生ヘッドを移動 [gotoAndStop]
// FrameNum  移動したいフレーム
procedure GotoFrame(FrameNum: Integer); safecall;

// 現在のフレームを返す [_currentframe]
// 0から始まるので注意
function CurrentFrame: Integer; safecall;
```

主要プロパティ紹介 AllowScriptAccess

```
// ドメイン間通信の制御 [none]
// always: スクリプトの実行が常に許可されます
// never: すべてのスクリプトの実行が禁止されます
// samedomain:
//   Flash アプリケーションが HTML ページと同じドメインに
//   置かれている場合にのみ、スクリプトの実行が許可されます
property AllowScriptAccess: wordString
  read Get_AllowScriptAccess write Set_AllowScriptAccess;
```

主要プロパティ紹介 TotalFrames/Loop/Scale

```
// 全フレーム [_totalframes]
property TotalFrames: Integer read Get_TotalFrames

// ループ再生 [none]
property Loop: WordBool read Get_Loop write Set_Loop;

// スケール方法 [none]
// ShowAll: 大きさにフィットさせて表示
// NoScale: 通常表示
property Scale: WideString read Get_Scale write Set_Scale;
```

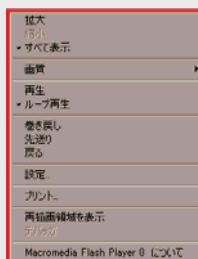
主要プロパティ紹介 Menu

```
// メニューの表示 [none]
property Menu: WordBool read Get_Menu write Set_Menu;
```

Menu = False



Menu = True



TFlashPlayerEx

TFlashPlayerEx クラスの作成

• 概要

- 自動生成された TShockwaveFlash を継承する
- Play, Stop が正しく働くように WM_TIMER をフック
- Movie プロパティへの代入を LoadFromFile でラップ
- Menu を"全く"表示させないことを可能にする

TFlashPlayerEx

```
TFlashPlayerEx = class(TShockwaveFlash)
private
  FShowMenu: Boolean;
  FPlaying: Boolean;
  FSWFFilename: String;
protected
  procedure WndProc(var VMsg: TMessage); override;
public
  constructor Create(VOwner: TComponent); override;
  procedure Play; reintroduce;
  procedure Stop; reintroduce;
  procedure SetBounds(VLeft, VTop, Vwidth, VHeight: Integer); override;
  procedure LoadFromFile(const VFileName: String);
published
  property ShowMenu: Boolean read FShowMenu write FShowMenu;
  property SWFFilename: String read FSWFFilename;
end;
```

TFlashPlayerEx

```
constructor TFlashPlayerEx.Create(VOwner: TComponent);
begin
  inherited;

  FShowMenu := True;
end;

procedure TFlashPlayerEx.LoadFromFile(const VFileName: String);
begin
  try
    if (FileExists(VFileName)) then begin
      FSWFFilename := VFileName;
      Movie := FSWFFilename;
    end;
  except
    end;
end;
```

TFlashPlayerEx

```
procedure TFlashPlayerEx.Play;  
begin  
    FPlaying := True;  
  
    inherited;  
end;  
  
procedure TFlashPlayerEx.Stop;  
begin  
    FPlaying := False;  
  
    inherited;  
end;
```

TFlashPlayerEx

```
procedure TFlashPlayerEx.SetBounds(vLeft, vTop, vWidth, vHeight: Integer);  
var  
    OO: IOleObject;  
    OIPS: IOleInPlaceObject;  
    tmpRect: TRect;  
begin  
    inherited;  
  
    DefaultInterface.QueryInterface(IOleObject, OO);  
  
    if (OO <> nil) then begin  
        OO.QueryInterface(IOleInPlaceObject, OIPS);  
  
        if (OIPS <> nil) then begin  
            tmpRect := BoundsRect;  
            OIPS.SetObjectRects(tmpRect, tmpRect);  
        end;  
    end;  
end;
```

TFlashPlayerEx

```

procedure TFlashPlayerEx.WndProc(var vMsg: TMessage);
var
  OK: Boolean;
begin
  OK := True;

  with vMsg do begin
    case Msg of
      WM_BUTTONDOWN: begin // 右クリックメニューの抑制
        OK := FShowMenu;
      end;

      WM_TIMER: begin // 再生を抑制
        OK := FPlaying;
      end;

      if (OK) then
        inherited
      else
        Result := 0;
    end;
  end;
end;

```

作成されたアプリケーション

サンプル ExternalInterfaceを呼ぶ

Menuの表示制御

FlashPlayerEx

再生を制御

メソッドのパラメータ
プロパティの値
各メソッド
プロパティを呼ぶ

ExternalInterface

ExternalInterface とは

- ヘルプより
 - ExternalInterface クラスは外部 API であり、ActionScript と Flash Player のコンテナとの間で直接通信できるようにするアプリケーションプログラミングインターフェイスです。コンテナの例としては、JavaScript が含まれる HTML ページや、Flash Player が埋め込まれたデスクトップアプリケーションなどがあります。

ExternalInterface の仕様

- Flash Player 8 以降が必要
- プロパティ
 - static available: Boolean
 - Player が外部インターフェースを備えているか

ExternalInterface の仕様

- メソッド
 - static addCallback(methodName: String, instance: Object, method: Function): Boolean;
 - 外部から呼ばれる swf 内の関数を登録
 - static call(methodName: String, [parameter: Object]): Object;
 - 外部の関数を呼び出す

ExternalInterface.addCallback の例

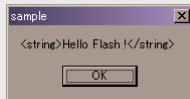
```
// ActionScript 側
import flash.external.*;

ExternalInterface.addCallback("Hello Flash", this, HelloFlashImpl);

function HelloFlashImpl(): String
{
    return "Hello Flash!";
}

// Delphi 側
ShowMessage(FFlashPlayerEx.CallFunction('<invoke name="Hello Flash" returnType="xml"></invoke>'));
```

// 実行結果



例で判るように、呼び出し・戻り値、それぞれ XML を利用する。

ExternalInterface.call の例

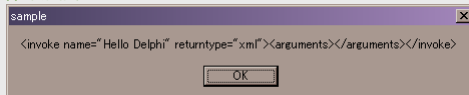
```
// ActionScript 側
import flash.external.*;

ExternalInterface.call("Hello Delphi");

// Delphi 側
procedure TForm1.FormCreate(Sender: TObject);
begin
    FFlashPlayerEx.OnFlashCall := FFlashPlayerExFlashCall;
end;

procedure TForm1.FlashPlayerExFlashCall(
    vSender: TObject;
    const vRequest: WideString);
begin
    ShowMessage(vRequest);
end;
```

// 実行結果



ExternalInterface のXML - 代表的なタグ

- Invoke タグ
 - ルートタグ
 - name 呼び出す関数名
 - returnType 戻り値を返す方法(XMLのみ)
- arguments タグ
 - 引数を示すタグ

ExternalInterface のXML-代表的な型

- string タグ 文字列を示す
 - `<string>文字列</string>`
- number タグ 数値を示す
 - `<number>123</number>`
- boolean 真偽値を示す
 - True の時 `<true/>`
 - False の時 `<false/>`
- array タグ 配列を示す
 - `<array><string>1</string><number>2</number></array>`
- undefined タグ undefined 値を示す
 - `<undefined/>`

ExternalInterface XML サンプル

```
// flash 側
ExternalInterface.addCallback("test", this, testImple);

function testImple(
    param1: String,
    param2: Number,
    param3: Array)
{
    // 処理
}

// 呼び出し XML
<invoke name="test" returntype="xml">
  <arguments>
    <string>abc</string>
    <number>123</number>
    <array>
      <number>456</number>
      <true/>
      <false/>
    </array>
  </arguments>
</invoke>
```