

## 吉原 泰介

株式会社ミガロ.

RAD事業部 技術支援課 顧客サポート

# Delphi/400でFlash動画の実装 —YouTubeプレイヤーの作成

Flashの機能を、ActiveXを利用して  
Delphi/400アプリケーションへ組み込む方法を紹介する。

- FlashとActiveXコントロール
- ActiveXコントロールの取り込み
- YouTubeプレイヤーの実装
- まとめ



### 略歴

1978年3月26日生まれ  
2001年 龍谷大学法学部卒  
2005年07月株式会社ミガロ入社  
2005年07月システム事業部配属  
2007年04月RAD事業部配属

### 現在の仕事内容

Delphi/400とJC/400の製品試験、および月100件に及ぶ問い合わせやサポート、セミナー講師などを担当している。

## 1. FlashとActiveXコントロール

近年のWeb上では、画像や動画、音楽がコンテンツとして組み込まれることが一般的になっている。【図1】

こういった画像、動画、音楽などのコンテンツをWebで扱うための技術として、広く使われるようになったのが「Flash」である。私達は普段Webブラウザで何気なくWebページを閲覧する際に、動画などを表示/再生するのに、実はこのFlashを利用している。実際、ネットワークに接続しているほとんどのPCには、Flashを利用するための環境が導入されている。

### ActiveXコントロール

このFlashはどのような形で私達のPCに提供されているかというと、実は「ActiveXコントロール」という技術を利用している。

ActiveXはMicrosoft社が提供するインターネット関連技術の総称である

が、最も代表的なのが、Webブラウザ上で動的なWebページを実現するActiveXコントロールである。ActiveXコントロールをActiveXと呼ぶ場合も多い。

ActiveXコントロールはネットワークを通じてWebサーバーからダウンロードされ、Webブラウザに機能を追加する形で使用される。つまり、Webブラウザ上で本来提供されていない機能を連携して使用しているわけである。

そして、このActiveXコントロールでFlashを利用することによって、昔はWebで扱うことができなかった動画や音楽を、Webブラウザ上で見たり聞いたりすることができるようになった。他にも身近なものとして、Windowsアップデート機能の更新チェックやインストールなども、ActiveXコントロールによってWebブラウザから提供されている。

ActiveXコントロールの機能をもう少し分かりやすく説明していく。

例えば、Delphi/400のアプリケーションでExcelの出力を扱う場合、COM(コンポーネント・オブジェクト・モデル)コントロールという技術が使用されている。COMコントロールは、Delphi/400のアプリケーションの中で、Excelという別アプリケーションの機能を利用するための連携を提供している。

では、ActiveXコントロールとCOMコントロールと何が違うかというと、実はほとんど同じものだと考えてよい。違いとしては、ActiveXコントロールは「Webで利用する技術」という点だけである。

### Delphi/400からFlashを扱う

以上のように考えてくると、ActiveXコントロールを利用して、Delphi/400からFlashを扱えようだというイメージができるのではないだろうか。

本稿では、インターネット動画共有サービスで有名なYouTubeの動画再生アプリケーションを題材として、Delphi/400からFlashを利用する方法

図1



図2



図3



図4

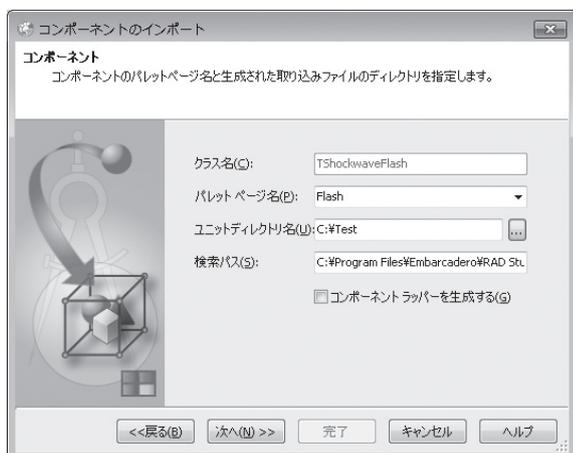


図5



を紹介したい。

## YouTube

<http://www.youtube.com/>

次章では、最初のステップとして、Flash を扱う機能を Delphi/400 に取り込む方法から説明していく。

## 2. ActiveXコントロールの取り込み

Delphi/400 で Flash 動画を扱うためには、Flash を再生するためのインターフェース情報が必要となる。このインターフェース情報は「TLB (タイプライブラリ)」と呼ばれる。

Delphi/400 には、ActiveX コントロールからインターフェースを抽出して、Delphi/400 のプログラムソースを自動で作成してくれる機能が提供されている。この機能を利用する。

### TLBの自動作成

開発画面のメニューから [コンポーネント | コンポーネントのインポート] を実行すると、ウィザード画面が表示される。【図 2】

ここで「ActiveX コントロールの取り込み」を選択し、「次へ」を実行する。

次の画面では、その PC にインストールされている ActiveX コントロールが一覧で表示されるので、Flash を再生するための「Shockwave Flash」という ActiveX コントロールを選択して「次へ」を実行する。【図 3】

次の画面では、この ActiveX コントロールの取り込み先を指定する。「パレットページ名」は任意の名前、「ユニットディレクトリ名」には自動生成されるプログラムソースの保存先ディレクトリを設定して、「次へ」を実行する。【図 4】

最後の画面では「ユニットの作成」を選択して「完了」を実行する。【図 5】

完了すると、保存先に指定したディレクトリに、ShockwaveFlashObjects\_TLB.pas というソースファイルが自動で作成されている。【図 6】

これは、ShockwaveFlash の TLB (タイプライブラリ) 情報のプログラムソースである。このインターフェース情報の

ソースは 1000 ステップ以上のプログラムからなるが、これらをコーディングすることなくウィザードで自動作成できるので、TLB の自動作成再生機能は非常に便利な機能といえる。

もちろんこの ShockwaveFlash に限らず、ActiveX で提供される機能は同様に Delphi/400 に取り込むことが可能である。

なお、ソースを見ると分かる通り、ソースの前半にはコメントで警告が書かれている。これは自動生成で作成されたプログラムソースなので、独自にソースを変更しても、次に自動生成した時には上書きで消えてしまうという注意書きである。

### Flash再生のためのクラス

ここで自動生成されたプログラムソースの詳細を完全に理解するのは難しいかもしれないが、中身を見てみると、ソースの①にあるように TShockwaveFlash というクラスが定義されている。【図 7-①】

またソースの②にあるように、このクラスには、Play や Stop といった Flash 再生を操作するためのメソッドも用意されていることが分かる。【図 7-②】

つまり、インターフェースだけでなく Flash を再生するためのクラスも自動で作成されているので、このクラスを利用するだけで、Flash を Delphi/400 から操作することができるのである。

次章では、実際にこのクラスを使用して、YouTube の動画再生プレイヤーを作成する。

## 3. YouTubeプレイヤーの装束

実際に、YouTube の動画を再生するアプリケーションを作成してみよう。

(1) はじめに新規プロジェクトを作成し、先に ActiveX コントロールから作成した自動生成ユニットの ShockwaveFlashObjects\_TLB.pas を組み込む。

これは [プロジェクト | プロジェクトに追加] から保存している ShockwaveFlashObjects\_TLB.pas を指定する。【図 8】

(2) これでプロジェクトに組み込まれたので、実際に Flash を使いたいユニッ

トで、interface 部の uses 節に ShockwaveFlashObjects\_TLB を追記する。【ソース 1】

(3) ShockwaveFlashObjects\_TLB で定義されたクラスを利用するために、変数を定義する。【ソース 2】

(4) 次に、アプリケーション画面を作成する。

画面には、YouTube で再生する動画の ID を入力する TEdit (①) と、再生用の TBitbtn (②) と停止の TBitbtn (③)、そして Flash を再生するための TPanel (④) を配置する。【図 9】

(5) Flash の再生部分には TPanel を使用している。

ShockwaveFlashObjects\_TLB で用意されているのはクラスであって、コンポーネントではないからである。今回は画面で描画する領域として、TPanel を利用する仕組みにしている。

この TPanel に Flash のクラスの描画を割り当てるために、画面起動時にフォームの Create イベントで、ソースに示すような処理を行う。【ソース 3】

(6) 画面終了時にはこれを破棄するために、フォームの Destroy イベントで、ソースに示すような処理を行う。【ソース 4】

(7) ここから、Flash を再生 / 停止するボタンの処理をプログラミングすることになる。

その前に、自動生成された TShockwaveFlash のクラスで使用できる、主な機能を記載しておく。

【TShockwaveFlashクラスで用意される主な機能】

- ・ Playメソッド: Flash Movieを再生するメソッド
- ・ Stopメソッド: Flash Movieを停止するメソッド
- ・ Movieプロパティ: 再生する Flash Movie を指定するプロパティ(※)

※ Flash のファイルをここで指定する。URL でも指定できる。今回の YouTube 動画の場合は、URL を指定する。

図6

```

Unit3 ShockwaveFlashObjects_TLB Project3
unit ShockwaveFlashObjects_TLB;

// ***** //
// 警告
// -----
// このファイルで宣言されている型はタイプライブラリから読み取られたデータから
// 生成されています。このタイプライブラリが明示的あるいは（このタイプライブラ
// リを参照しているほかのタイプライブラリ経由で）間接的に再インポートされた
// リ、タイプライブラリの編集中にタイプライブラリエディタの「更新」コマ
// ンドを実行した場合、このファイルの内容はすべて再生成され、手動で加えた変更
// はすべて失われます。
// ***** //

// $Rev: 17244 $
// 下に説明されたタイプ ライブラリから 2010/08/21 17:07:36 に生成されたファイル。

// ***** //
// タイプ ライブラリ: C:\Windows\system32\Macromed\Flash\FFlash10e.ocx (1)
// LIBID: {D27CDB6B-AE6D-11CF-96B8-444553540000}
// LGID: 0
// ヘルプファイル:
// ヘルプ文字列: Shockwave Flash
// 依存関係リスト:
// (1) v2.0 stdole. (C:\Windows\system32\stdole2.tlb)
// エラー:
// ヒント: IShockwaveFlash.TGotoLabel のパラメータ 'label' の名前が 'label_' に変更さ
// ヒント: IShockwaveFlash.TSetProperty のパラメータ 'property' の名前が 'property_'
// ヒント: IShockwaveFlash.TGetProperty のパラメータ 'property' の名前が 'property_'
// ヒント: IShockwaveFlash.TCallLabel のパラメータ 'label' の名前が 'label_' に変更さ
// ヒント: IShockwaveFlash.TSetPropertyNum のパラメータ 'property' の名前が 'propert
// ヒント: IShockwaveFlash.TGetPropertyNum のパラメータ 'property' の名前が 'propert
// ヒント: IShockwaveFlash.TGetPropertyAsNumber のパラメータ 'property' の名前が 'pr
// "IShockwaveFlash" のパレット用ビットマップの作成時エラー: コントロール ビットマ
// "FlashProp" のパレット用ビットマップの作成時エラー: サーバー C:\Windows\system3
// "FlashObjectInterface" のパレット用ビットマップの作成時エラー: サーバー C:\Wind
// ***** //
// ***** //
// メモ:

```

図7

```

Unit3 ShockwaveFlashObjects_TLB Project3
TShockwaveFlash = class(TOLEControl)
private
  FOnReadyStateChange: TShockwaveFlashOnReadyStateChange;
  FOnProgress: TShockwaveFlashOnProgress;
  FOnFSCommand: TShockwaveFlashFSCommand;
  FOnFlashCall: TShockwaveFlashFlashCall;
  FIntf: IShockwaveFlash;
  function GetControlInterface: IShockwaveFlash;
protected
  procedure CreateControl;
  procedure InitControlData; override;
  function Get_InlineData: IUnknown;
  procedure Set_InlineData(const ppIUnknown: IUnknown);
public
  procedure SetZoomRect(left: Integer; top: Integer; right: Integer; bottom: Integer);
  procedure Zoom(factor: SYSINT);
  procedure Pan(x: Integer; y: Integer; mode: SYSINT);
  procedure Play;
  procedure Stop;
  procedure Back;
  procedure Forward;
  procedure Rewind;
  procedure StopPlay;
  procedure GotoFrame(FrameNum: Integer);
  function CurrentFrame: Integer;
  function IsPlaying: WordBool;
  function PercentLoaded: Integer;
  function FrameLoaded(FrameNum: Integer): WordBool;
  function FlashVersion: Integer;
  procedure LoadMovie(layer: SYSINT; const url: WideString);
  procedure TGotoFrame(const target: WideString; FrameNum: Integer);
  procedure TGotoLabel(const target: WideString; const label_: WideString);
  function TCurrentFrame(const target: WideString): Integer;
  function TCurrentLabel(const target: WideString): WideString;
  procedure TPlay(const target: WideString);
  procedure TStopPlay(const target: WideString);
  procedure SetVariable(const name: WideString; const value: WideString);
  function GetVariable(const name: WideString): WideString;
  procedure TSetProperty(const target: WideString; property_: SYSINT; const value:
  function TGetProperty(const target: WideString; property_: SYSINT): WideString;

```

(8) 再生ボタンが押された処理では、Movie プロパティに YouTube の URL とあわせて画面で入力された動画の ID をセットして、Play メソッドで Flash の再生を行う。【ソース 5】

#### YouTube の URL

<http://www.youtube.com/v/>

指定する ID は、YouTube で公開されている動画に設定されている ID である。例えば、次の URL は Google ストリートビューの YouTube 動画の URL であり、「v/」以降の文字列が ID である。

[http://www.youtube.com/v/XD\\_4WSwkWEU](http://www.youtube.com/v/XD_4WSwkWEU)

なお、YouTube 動画上を右クリックしてポップアップされるメニューから「Copy embed html」を選択し、エディタ等に貼り付けるなどして、ID を含む情報を確認することもできる。

(9) 最後に停止ボタンを押された時の処理では、単純に Stop メソッドを実行するだけである。【ソース 6】

(10) これだけで、簡単な YouTube プレイヤーの完成である。早速実行して Flash の再生を試してみよう。【図 10】

なお(9)に関連して、ShockwaveFlash 自体に「Stop メソッドを実行しても停止しない場合がある」という動作問題がある。

Stop メソッドの動作を確実に解決する場合には、本レポートの最後に補足掲載している、ソースのクラス修正を加えることで、Delphi/400 側で対応することができるので参考にしてほしい。【ソース 7】 【ソース 8】

## 4. まとめ

この YouTube 動画再生アプリケーションの開発は非常にシンプルで、簡単に Flash の再生操作機能を実現することができた。

もちろん Flash の再生プログラム自体を自身ですべて開発しようと思えば、

複雑で高度なプログラムが必要だが、Delphi/400 の ActiveX コントロールの自動取り込み機能によって、それらの開発はほとんど必要としない。

本稿のアプリケーションは YouTube の動画を再生する機能だけであるが、Movie プロパティに Flash ファイルを設定すれば、通常の Flash も再生することができる。つまり、Web ページ同様、Delphi/400 のアプリケーションも、画面に Flash 動画などを組み込んだ見た目のよい画面を作ることが可能なのである。【図 11】

また、最近では Flash を利用したプログラムで、「Flex」と呼ばれるプログラム (ActionScript) を組み込んだ Flash もある。これらと連携したアプリケーションの開発も面白いだろう。【図 12】

この Flash の連携だけでも、Delphi/400 アプリケーションには、さまざまな開発の可能性が含まれているのが分かる。他の ActiveX コントロールの利用や今後の開発において、ヒントや参考にさせていただければ幸いである。

■

図8

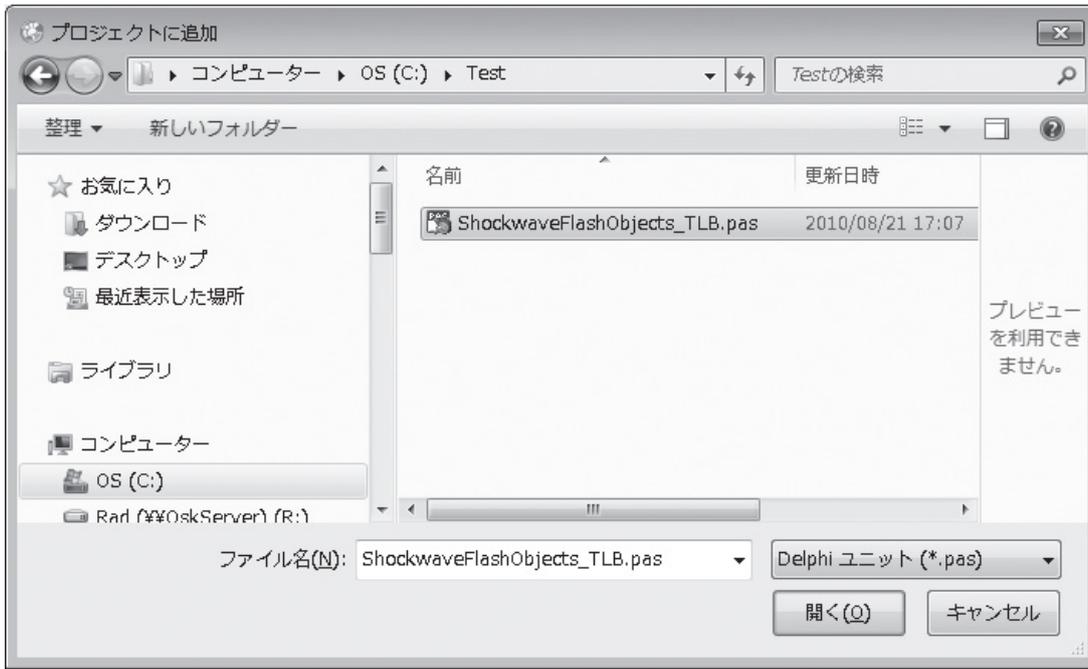


図9



図10



図11



図12



### ソース1 Uses節

```
unit Unit1;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, ExtCtrls, Buttons, ShockwaveFlashObjects_TLB;
```

### ソース2 宣言部

```
private  
  { Private 宣言 }  
  //表示用Flash  
  fFlashplayer : TShockwaveFlash;
```

### ソース3 初期処理(表示Flash生成)

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  //Flashをクラスから生成  
  fFlashplayer := TShockwaveFlash.Create(Self);  
  //最大化  
  fFlashplayer.Align := alClient;  
  //表示用FlashplayerのPanelの上に設定 (親子関係)  
  fFlashplayer.Parent := Panel1;  
end;
```

### ソース4 終了処理(表示Flash破棄)

```
procedure TForm1.FormDestroy(Sender: TObject);  
begin  
  //表示用Flashplayerを破棄  
  fFlashplayer.Free;  
end;
```

### ソース5 再生ボタンクリック

```
procedure TForm1.btnPlayClick(Sender: TObject);  
begin  
  //入力のVIDを指定してMovieに設定  
  fFlashplayer.Movie := 'http://www.youtube.com/v/' + Edit1.text;  
  //Flashを実行  
  fFlashplayer.Play;  
end;
```

#### ソース6 停止ボタンクリック

```
procedure TForm1.btnStopClick(Sender: TObject);
begin
  //Flashを停止
  fFlashplayer.Stop;
end;
```

#### ソース7 修正クラス宣言部

```
//宣言部
type
  //Stop、Playメソッドの動作を補正するためのクラス
  TFlash = class(TShockwaveFlash)
  private
    FPlaying : Boolean;
  protected
    procedure WndProc(var vMsg: TMessage); override;
  public
    procedure Play; reintroduce;
    procedure Stop; reintroduce;
    property Playing: Boolean read FPlaying;
  end;
```

#### ソース8 修正クラス実装部

```
//実装部
{ TFlashPlayer }
//Playメソッド
procedure TFlash.Play;
begin
  FPlaying := True;

  inherited play;
end;

//Stopメソッド
procedure TFlash.Stop;
begin
  FPlaying := False;

  inherited stop;
end;

//メッセージ procedure TFlash.WndProc(var vMsg: TMessage);
begin
  case vMsg.Msg of
    WM_TIMER:
      if (FPlaying) then
        begin
          inherited;
        end
      else
        begin
          vMsg.Result := 0;
        end;
      else
        inherited;
      end;
end;
```

