

小松学史の携帯でFlash講座 Flash Lite

2005年9月16日
講師：小松学史

Flash Liteとは？

- Flash Liteとは携帯電話上で表示される、Flashの形式です。

Flashモバイルコンテンツ対応端末の現状

- 2003年5月にFlash Lite1.0が発表され、NTT Docomoの携帯電話、505i、506i、FOMA900iシリーズに専用のPlayerが搭載されています。
- 2004年7月にはFlash Lite1.1がauの携帯電話、最新のWINシリーズにPlayerが搭載されています。
- 2005年1月からは、Vodafone対応の機種も発売されました。

http://www.nttdocomo.co.jp/p_s/imode/spec/flash.html

http://www.au.kddi.com/ezfactory/tec/spec/new_win/ezkishu.html

Flashモバイルコンテンツ対応端末の現状

- Docomo
 - 1.0対応(505、506、FOMA900) 約30機種
 - 1.1対応(FOMA901、FOMA700)約20機種
- au
 - 1.1対応(CDMA1X、CDMA1X WIN) 約25機種
- Vodephone
 - 1.1対応 約10機種

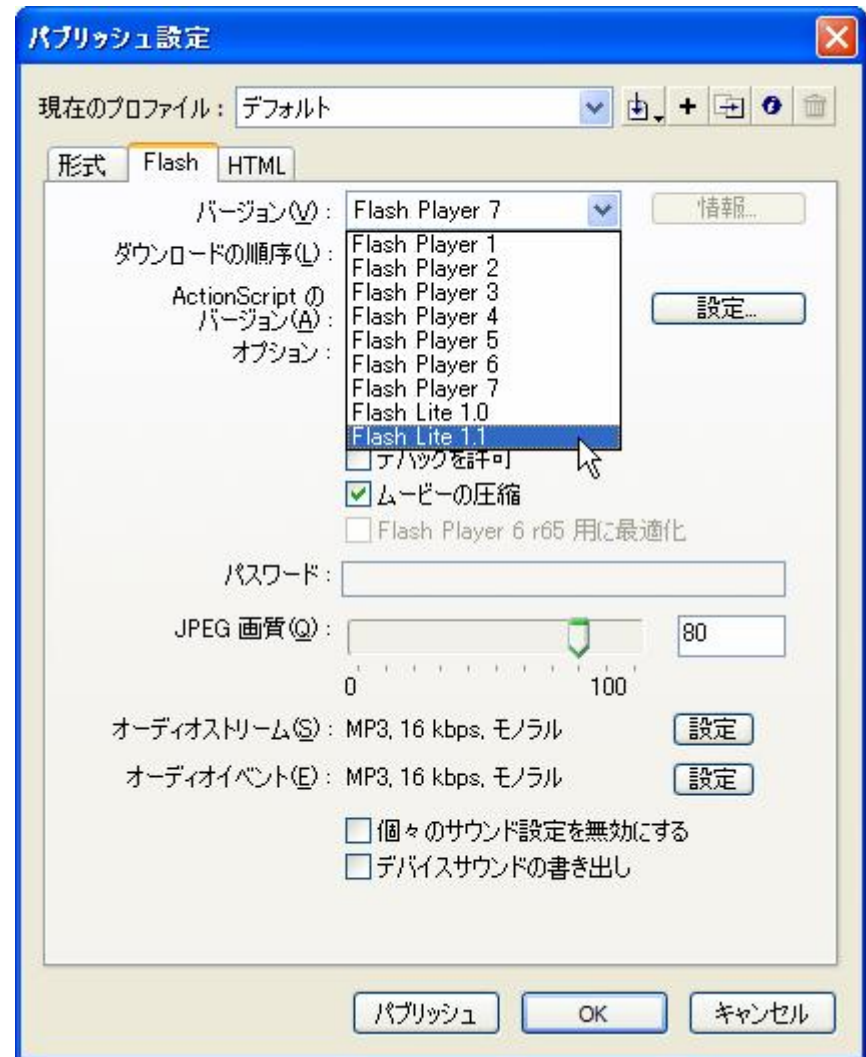
Flash Lite 1.1 ファイルを書き出すために

- Macromedia Flash MX Professional 2004 を、7.2にアップデートすることで、携帯端末用Flash Lite 1.1 のコンテンツを作成することができるようになります。
- Flash Lite1.1から使用できる技術でなければ、FlashMXなどでも開発可能です。
- アニメーションと簡単なアクションを使用するくらいでしたら、Flash 4形式に書き出すことで、Flash対応携帯電話にて確認することができます。

ファイル作成時の設定 (バージョン)

- バージョンの設定
 - パブリッシュ設定で「Flash Lite1.1」に設定します

※Flash Lite1.1からの技術を使用していなければ、Flash Lite1.0環境においても閲覧可能です

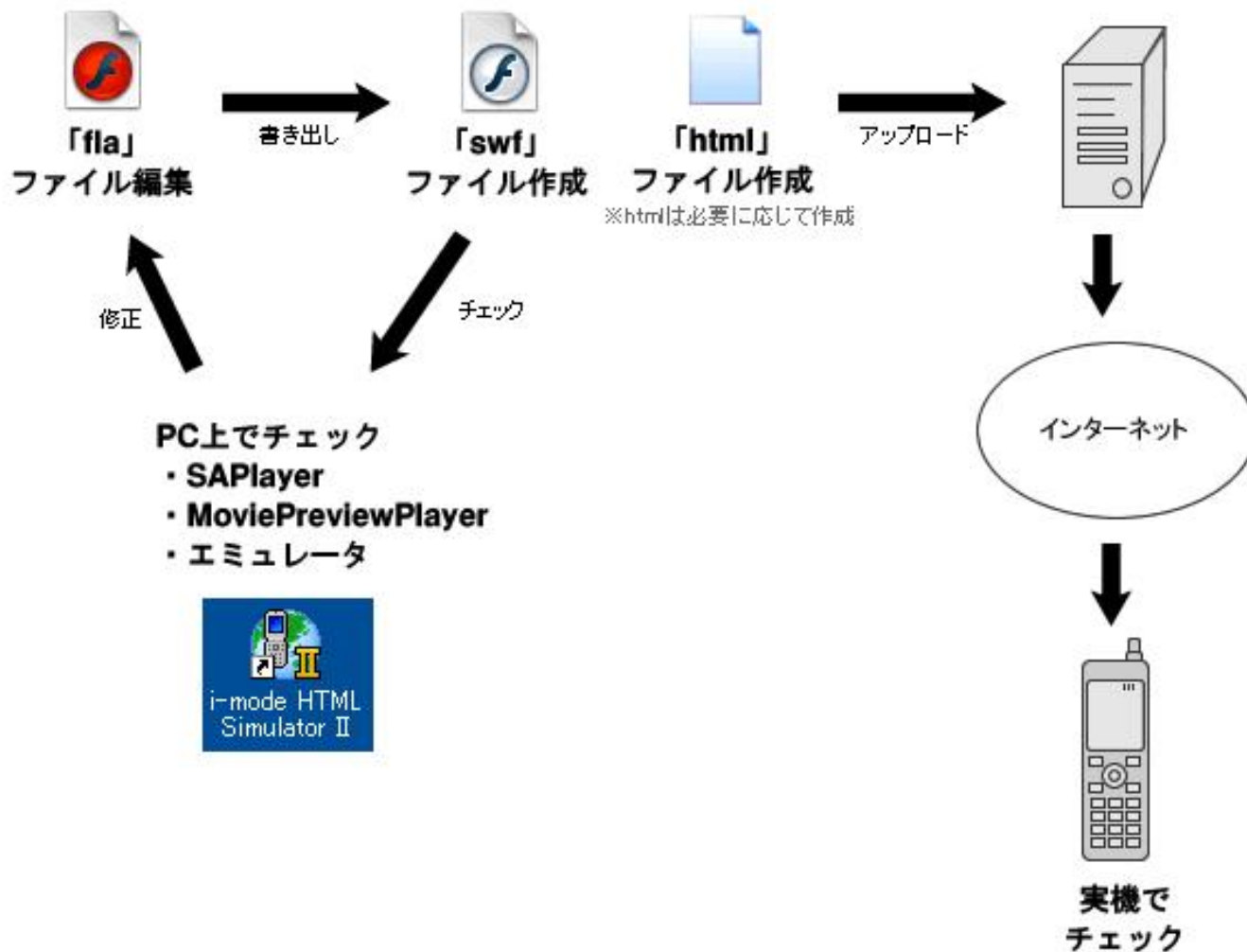


ファイル作成時の設定(サイズ、fps)

- サイズの設定
 - ドキュメントプロパティでサイズを設定します
 - 最適なサイズは、用途や実機に合わせて設定します
- フレームレートの設定
 - 携帯電話の処理速度を考慮した上で、fpsを設定します
 - Flash Lite1.0→5～8程度
 - Flash Lite1.1→10～15程度



制作の流れ



ファイルを作成する上で

- 制作上の基本情報については、Macromediaから配布されている、CDK内のPDFを閲覧しましょう。

モバイル&デバイスデベロッパーセンター

http://www.macromedia.com/jp/devnet/devices/development_kits.html

※キャリア毎にCDKがあります (Vodephone除く)

現在のFlashと、Flash Lite1.0およびFlash Lite1.1の違い

- アニメーションについて
- ボタンについて
- サウンドについて
- htmlについて
- テキストについて
- アクションについて
- Flash Liteの制限事項(データサイズ、メモリ)
- FScommand2とプラットフォーム変数

アニメーションについて

- タイムラインを使用したアニメーションは、(Flash4以降から使用できるものは)正常に動作します。
- 例外としてMovieClip内のアニメーションは、マスクとして動作しません。(Flash Player6以降から導入されたため)
- また、携帯電話自体のマシンスペックが低いため、処理が多い視覚効果は避けましょう。(アルファ値を変更するアニメーションなど)

ボタンについて

- 多くの携帯電話にはポインティングデバイスがついていませんので、ユーザーは携帯電話の上下キーもしくは、ダイヤルキーでアクセスします。
- 上下キーを使用した場合、ボタンがフォーカスの対象になります。フォーカスされたボタンシンボルはオーバースタイルの絵に変わります。(決定キーを押しても、ダウンスタイルの絵は表示されません)

ボタンについて(2)

- ダイヤルキーでボタンにアクセスするには、ActionScriptで設定をします。

例)

```
on (press, keyPress "1") {  
    gotoAndStop(10);  
}
```

※携帯場合「keyPress”キーコード”」で設定します。キーコードは1～9、*、#が使用することができます。

ボタンについて(3)

- 通常の携帯電話はポインティングデバイスがありませんので、マウスに関するActionScriptの多くがサポートされていません。

例) `releaseOutside`、`dragOver`、`dragOut`、`startDrag`、`stopDrag`

サウンドについて

- Flash Lite1.0ではMfi形式、SMF形式に対応しています。
- Flash Lite1.1では、Mfi形式、SMF形式、SMAF形式が対応され、また、サウンドの波形をサンプリングした形式もサポートされました。

※Flash Lite1.1対応実機のほとんどがMP3をサポートされていなかったりします…。

サウンドについて(2)

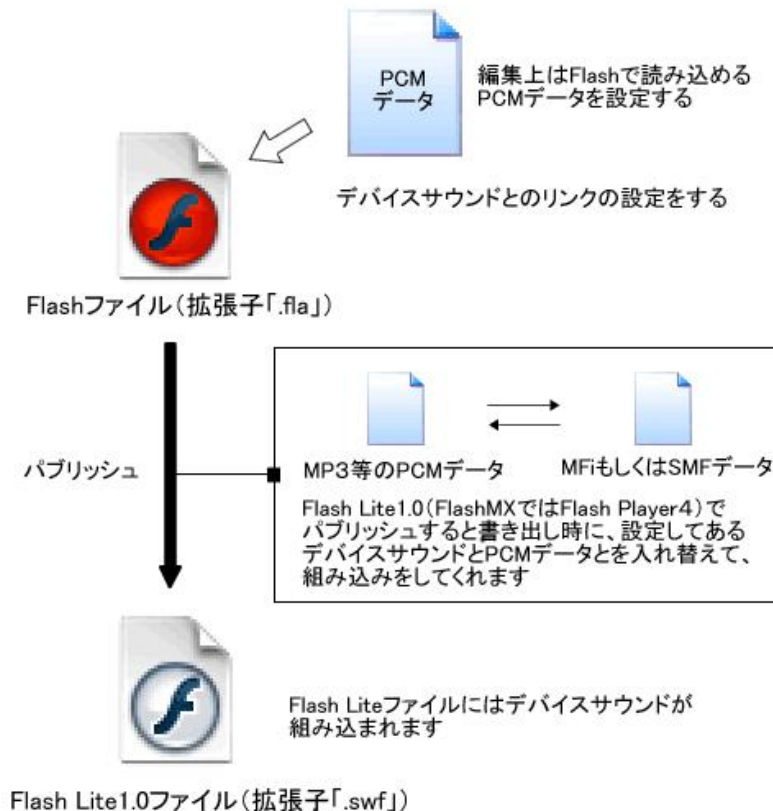
- Flash MX 2004では、MfiやSMAFサウンドを読み込むことができません。
- サウンドを設定するときには、MP3などのFlashで使用できるサウンドで仮にタイムライン上に設定します。

サウンドについて(3)

- 設定したサウンドのプロパティ(ライブラリ)で使用したい「デバイスサウンド」を指定します。
- デバイスサウンドを指定された、サウンドは書き出し時に、デバイスサウンドと置換されます。

サウンドについて(4)

• Flash Lite1.0、および1.1でのデバイスサウンドの変換図



サウンドについて(5)

サウンド対応状況

| | ver | Mfi | SMF | SMAF | MP3 |
|-------------------|-----|-----|-----|------|-----|
| 505i、506i | 1.0 | ○ | × | × | × |
| FOMA900i | 1.0 | ○ | ○ | × | × |
| FOMA901i、FOMA700i | 1.1 | ○ | ○ | × | × |
| au | 1.1 | × | × | ○ | × |
| Vodephone | 1.1 | × | × | ○ | × |

htmlについて

- 携帯のブラウザで表示することのできるFlashには2つの再生方式があります。
 - インライン再生方式
 - インタラクティブ再生方式
- Flash MX 2004で書き出されるHTMLの記述とは異なりますの注意が必要です。

htmlについて(2)

- インライン再生方式

- アニメーションGIFのように、html内にswfを表示する方法です
- 上下キーはhtmlのスクロールに予約されます。
- ボタンやサウンドなどが使用できません。

htmlについて(3)

- インタラクティブ再生方式
 - Swfのみを表示する再生方式です
 - ブラウザのスクロールが表示されません。
 - ボタンやサウンドの制御が可能です。
 - 基本的にはSwfに直接アクセスします。

htmlについて(4)

記述例)

```
<object data="swfファイル名" type="application/x-shockwave-  
flash" width="200" height="100">  
<param name="bgcolor" value="#99CCCC">  
<param name="loop" value="on">  
<param name="quality" value="high">  
</object>
```

※インライン再生方式のHTML

※auなどは、XHTML Basic 1.0 に準拠した記述にします

htmlについて(5)

記述例2)

```

```

※auではimgタグでも設定することが可能です

テキストについて

- 入力テキストフィールドについて
 - Flash Lite1.0では、入力テキストがサポートされていません。
 - Flash Lite1.1では、入力テキストがサポートされています。
- Flash Lite1.1での入力テキストフィールドは、上下キーを使用した際にフォーカスの対象になり、フォーカスしている状態で決定キーを押すことで、携帯電話のテキストのエディタが起動します。

テキストについて(2)

- デバイスフォントについて
 - フォントを埋め込みにすると、データ容量が大きくなりがちです。
 - 読んでもらうための文章などはデバイスフォントにするほうが良いでしょう。
- デバイスフォントを使用する際の注意！
 - 携帯電話の実機には、フォントの種類が1～2種類くらいしか入っていません。
 - 用意されているデバイスフォントのサイズは、携帯電話によって異なります。

テキストについて(3)

- Flash Lite1.0対応機種

→ほとんどが24pxしかない(20pxのものもあります)

例) 12pxで指定した、デバイスフォントは24pxのサイズになり
テキストが半分切れます

- Flash Lite1.1対応機種

→キャリア、メーカー毎に、種類、数などが異なります。

例) Docomoは12,16,20,24のうち3種類

例) CDMA1X は16~36のうち、3~4種

例) CDMA1X WIN は12~36のうち、3~4種

例) Vodephone のある機種においては24pxしかない

アクションについて

- Flash4ベースということ
- 基本的な構文について
- 演算子について
- 外部ファイル
- スクロール
- Flash5のアクションで使用できるもの
- プラットフォーム変数
- FScommand

Flash4ベースということ

- Flash Lite1.0および1.1で、現在のFlashのActionScriptと大きく異なるのは、Flash4をベースにしていることです。
- しかし、Flash4の頃にサポートされているActionScriptでもサポートされていないものもあります。また、Flash5以降からの技術もいくつかサポートされています。

基本的な構文について

- Flash4のころは、現在と異なる構文ですが、Flash MX 2004では、書き出し時にFlash4の構文に対応したものに交換してくれます。
- しかし、現在の記述と異なるところもたくさんあります。
 - プロパティはドットシンタックスで記述OK
 - メソッド(関数)はドットシンタックスでの記述NG
 - `_root`、`_parent`はOK。 `this`はNG

演算子について

- 数式演算子と連結演算子について
 - 「+」という演算子は、現在のFlashでは数式演算子と連結演算子の両方の役割を担いますが、Flash Lite1.0および1.1では、数式演算子のみの役割しかできません。

「+」演算子についての例

例1)

```
a = 1;
```

```
b = 2;
```

```
c = a + b;
```

```
trace(c);
```

Flash Player5以降 →出力結果は「3」

Flash Lite1.0、1.1 →出力結果は「3」

「+」演算子についての例2

例2)

```
a = "1";
```

```
b = 2;
```

```
c = a + b;
```

```
trace(c);
```

Flash Player5以降 →出力結果は「12」

Flash Lite1.0、1.1 →出力結果は「3」

「+」演算子についての例2

例3)

```
a = "Flash";
```

```
b = "Lite";
```

```
c = a + b;
```

```
trace(c);
```

Flash Player5以降 →出力結果は「FlashLite」

Flash Lite1.0、1.1 →出力結果は「0」

Flash Lite1.0および1.1での連結演算子

- Flash Lite1.0および1.1での連結演算子は「add」を使用します。

例4)

```
a = "Flash";
```

```
b = "Lite";
```

```
c = a add b;
```

```
trace(c);
```

Flash Player5以降 →出力結果は「FlashLite」

Flash Lite1.0、1.1 →出力結果は「 FlashLite 」

比較演算子について

- 比較演算子について

- 「==」という演算子は、現在のFlashでは数、および文字列の比較(等価)する役割を担いますが、Flash Lite 1.0および1.1では、数のみの比較しかできません。

比較演算子についての例

例1)

```
a = "Macromedia"  
if(a == "Microsoft"){  
  trace("aはマイクロソフトである");  
}else{  
  trace("aはマイクロソフトではない");  
}
```

Flash Player5以降 →出力結果は「aはマイクロソフトではない」

Flash Lite1.0、1.1 →出力結果は「aはマイクロソフトである」

Flash Lite1.0、1.1において、比較演算子「==」は変数aを数字とみなし、「0」として考えてしまう。

比較演算子についての例2

例2)

```
a = "Macromedia"  
If(a eq "Microsoft"){  
trace("aはマイクロソフトである");  
}else{  
trace("aはマイクロソフトではない");  
}
```

Flash Player5以降 →出力結果は「aはマイクロソフトではない」

Flash Lite1.0、1.1 →出力結果は「 aはマイクロソフトではない」

文字列の等価比較には「eq」を使用します。非等価には「ne」を使用します。また、数式の非等価は現在「!=」を使用しますが、Flash Lite1.0および1.1では「<>」を使用します。

外部ファイル

- 外部ファイル読み込みの制限について
 - loadMovie、loadMovieNum
 - loadVariables、loadVariablesNum
- Flash Lite 1.0では、外部のファイルを読み込むコマンドはサポートされていません。
- Flash Lite 1.1では、条件付でサポートされています。

外部ファイル読み込みの制限

- ボタンアクション、キーアクセスを行ったときのみ実行されます。(待ち受けや、インライン再生では不可)
- 1回のアクセスについて、1ファイルのみです。
 - loadMovieを行ったら、同じイベント内では、その後のloadMovieおよびloadVariablesが実行されません。

キャリア別外部ファイル読み込みの制限について

- Docomo

- ベースのファイルを含む、読み込んだ全てのファイル容量トータルが100KBになった時点で、外部ファイルの読み込み不可

- au

- 一度に読み込める外部ファイルの容量が100KB (CDMA1Xは48KB) 以内であれば、何度でも読み込むことが可能。

外部ファイル、メモリ、キャッシュについて

- 外部ファイルはキャッシュにたまりません。
- unloadMovieなどを使用しても、メモリが開放されません。

loadMovieの記述について

- 記述例)

- ◎ loadMovie (“data/loadclip.swf”, ”Target_mc”);
- × loadMovie (“data/loadclip.swf”, Target_mc);
- × loadMovie (“data/loadclip.swf”, _root.Target_mc);
- ◎ loadMovie (“data/loadclip.swf”, ”/Target_mc”);
- × _root.Target_mc.loadMovie (“data/loadclip.swf”);

※ドットシンタックスで記述現在の書き方は、Flash Lite1.1では使用することができません。

※パスの指定はスラッシュシンタックスで行う必要があります。

テキストスクロール

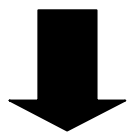
- scrollプロパティ、maxscrollプロパティについて
- Flash Lite1.0では、スクロールの制御を行うコマンドはサポートされていません。
- Flash Lite1.1では、スクロールの制御を行うコマンドはサポートされています。
- 現在のテキストインスタンスのプロパティと記述は似ていますが、Flash Lite1.0および1.1のでの記述はテキストフィールドに設定した「変数」のプロパティとして扱います。

Flash5のアクションで使用できるもの

- for文
 - CDKではサポートの記述はありませんが、使用することが可能です。
- switch文
- Mathオブジェクト
 - Mathオブジェクトは定数のみ使用できます。
 - (Math.sin(x)のように)変数を使用した場合は近似値を割り当ててくれますがエミュレート値は誤差が大きいです。
- 演算子
 - ++ -- += -= %= *= /= || ! &&
- ブランケットアクセス
 - 例) `_root["mcdata"]._x = 100;`

プラットフォームの機能と変数

- Flash Lite、デバイス、ホストアプリケーション、またはFlash Playerが特定の機能に対応しているかどうかを示すための変数が使用できるようになりました。



- `_capMFi`
 - 変数 `_capMFi` を使用できるかどうかを示します。
- `$version`
 - 変数 `$version` には、Flash Lite のバージョン番号が格納されます。
- その他

FSCommand2()

- FSCommand2() とは？
- FSCommand2()関数は、Flash Lite 1.1 でサポートされている新しいActionScript 関数です。
- この関数により Flash Lite コンテンツと Flash Player、ホストアプリケーション、および Flash Player のホストとなるデバイスとの間の通信を行うことができます。

FSCommand()・FSCommand2()でできること

- URL のエンコーディング
- 入力テキストフィールドの制御
- アニメーションのレンダリング画質を設定
- 使用可能ワークメモリのサイズをKB単位で返す
- ワークメモリの合計サイズをKB単位で返す

FSCommand()・FSCommand2()でできること

- 現在の時刻と日付の取得
- 端末の情報データの取得
 - 音量情報
 - 電源情報
 - ネットワーク情報
 - バッテリー残容量の情報
 - ID 情報
 - プラットフォーム情報
- 端末のバイブレータ機能の作動と停止
- 端末の言語設定の取得
- 端末のバッテリー残容量の情報の取得
- その他

FSCommand2記述例

```
status = FSCommand2( "GetDateMonth" );
```

キーフレームに記述すれば、この場合変数statusに現在の月(1 ~ 12)が設定されます。

記述方法は、どのFSCommand2も、ほぼ同じで引数の設定などが、それぞれのコマンドで異なります。詳細はCDKについているオーサリングガイドを参照してください。CDKについているサンプルも参考になるでしょう。

FSCommand()・FSCommand2()でできること

- auの着FlashもFSCommand2()を使用したものになります。

```
status = FSCommand2 ("GetCallerInfo","CallerName","CallerNumber")
```

- 変数status ステータスの戻り値が入ります
- 変数CallerNumber かかってきた電話番号が入ります
- 変数CallerName かかってきた電話番号で登録されている人の名前が返ります

- CDKには記述されていませので、詳細はEzfactoryで
- <http://www.au.kddi.com/ezfactory/mm/flash04-01.html#chakuflash>

Flash Liteの制限事項(データサイズ、メモリ)

- データ容量の上限

- Flash Lite1.0では20KB(FOMA900iは100KB)
- Flash Lite1.1準拠のものでは100KB
- auのCDMA1Xのシリーズだけは、ブラウザ上で展開できるデータサイズが48KBになっています。(待ち受けなどの容量は100KBまでOK)

Flash Liteの制限事項(データサイズ、メモリ)

- ワークメモリの上限はメーカーや出荷時期によってことなります。
- ワークメモリの上限
 - 505iシリーズ 200~300KB
 - 505isシリーズ 506iシリーズ 300~400KB
 - FOMA900iシリーズ 1MB
 - FOMA901iシリーズ、FOMA700シリーズ 2MB
 - au端末 1MB~3MB程度

容量の上限を超えてしまうと・・・

- データ容量の場合
 - 読み込みが完了できません
- ワークメモリの場合
 - 1フレームあたりのパスの量が多くて越える場合は、表示すらされません
 - 画像を使いすぎてしまった場合は、画像のエリアが赤い矩形の表示になります。

ローテクとうまく付き合う、ノウハウ

- フレームの利用
- キーイベントとボタンの取り扱い
- ドットシンタックスとスラッシュシンタックス
- tellTarget
- call
- phpとming

フレームの利用

- 継続して何かActionScriptを実行したいとき、現在のFlashでは、setIntervalやonClipEvent(enterFrame)などを使用したりします。
- しかし、setInterval もonClipEventもFlash5以降のものなので、Flash Lite1.0および1.1では使用することができません。

onClipEvent(enterFrame)の代替案

- onClipEvent(enterFrame)はとても便利です。毎フレーム何かアクションを処理したい場合とかに使ったりします。しかし、前述のとおりonClipEventは使えません。ではどのように設置すれば同じようなことができるのでしょうか？

onClipEvent(enterFrame)の代替案

- enterFrameとは、つまるところ「フレームに入ったら」という意味ですから、フレームレートが12fpsの時には、理論上1秒間に12回実行すると考えられます。
- ムービークリップ(以下MC)内のタイムラインを使用すれば同じようなことができます。
- MCのタイムラインは最後のフレーム行くと自動的に、最初のフレームに戻りますので、これを利用して、MCのキーフレームに実行するアクションを入れます。
- MC内のフレームを、2フレーム以上作成し、フレームのループをするように仕組みを作成し、キーフレームに設定したアクションを何度も実行させるようにすれば良いですね。

キーイベントとボタンの取り扱い

- ボタンのフォーカスする順番は、上から下へ、同じ高さの座標ならば、左から右へフォーカスしていきます。
- 例えば、環状にならんだボタンがあったとしても、環状そってボタンがフォーカスすることはありません。
- Flash4のころには、ボタンのフォーカスする順番を制御するActionScriptがありませんでしたので、仕方がない話です。
- ボタンのフォーカスする順番は制御できないのですが、ボタンのフォーカスする見た目上の順番を定義する方法があります。
- 通常携帯電話にはポインティングデバイスがありませんので、ボタンのヒット領域というのはそれほど重要ではありません。

キーイベントとボタンの取り扱い

- 通常携帯電話にはポインティングデバイスがありませんので、ボタンのヒット領域というのはそれほど重要ではありません。
- しかし、フォーカスの対象は実はこのヒット領域だったりします。
- ヒット領域だけを上から順番に並べることができれば、見た目上の順番を制御することが可能です。

※欠点としては、ポインティングデバイスを使用した場合には、見た目のボタン上で選択できないことがあります。

※フォーカスの領域が見えてしまうので、スクリプトで表示を見えなくしたほうがよいでしょう

例) `_focusrect = false;`

ドットシンタックスとスラッシュシンタックス

- 現在のFlashはドットシンタックスが記述しますが、Flash4の時代はターゲットパスは全てスラッシュシンタックスでした。
- 現在のドットシンタックスの書き方で、Flash MX 2004が変換してくれますが、コマンドによっては、変換がうまくいかないものがあります。

ドットシンタックス

- オブジェクトの構成を「ドット」で区切って、記述する構文です。

例)

```
_root.counter = 1;  
//メインタイムラインにある変数の設定
```

```
dog_mc.counter = 1;  
//同じ階層にある「dog_mc」というムービークリップ内の変数の設定
```

```
_parent.couter = 1;  
//1階層上にある変数の設定
```

などと、書けます。
これも、ActionScript1.0と同じですね。

※thisは使用できません。

ドットシンタックスが使えないのは？

- ドットシンタックスは、プロパティおよび変数の設定で使用することができますが、メソッドについてはドットシンタックスで指定できません。「tellTarget」を使用します。
- 「tellTarget」やその他のメソッドで使用する、ターゲットパスはスラッシュシンタックスで記述する必要があります。

ドットシンタックスが使えないのは？(2)

例)

```
_parent._parent.counter = 1;
```

//2階層上にある変数counterに1を設定する

- 上記は一見よくありそうな書き方ですが、変換がうまくいっていないのか動作しません。1階層上の変数やプロパティは設定できても、2階層以上ものは設定できないのです。

スラッシュシンタックス

- スラッシュシンタックスは、Flash4の頃に使用されていた構文で、ターゲットパスを「/」を用いて参照することができます。

スラッシュシンタックス

例)

```
/:counter = 1;  
//メインタイムラインにある変数の設定
```

```
dog_mc/:counter = 1;  
//同じ階層にある「dog_mc」というムービークリップ内の変数の設定
```

```
../:couter = 1;  
//1階層上にある変数の設定
```

```
../../:couter = 1;  
//2階層上にある変数の設定
```

※HTMLのファイルパスの考え方に似ています。変数の前には「:」(コロン)をつけます。

スラッシュシンタックス

- プロパティを設定する場合も、ドットシンタックスを使用して、相対パスで2階層上のオブジェクトにはアクセスできません。
- ドットシンタックスを絶対パスで表記するか、「setProperty」というコマンドを使用して設定します。

スラッシュシンタックス

例)

```
_parent._parent.ball._x = 200;
```

↓

```
setProperty("../../", _x ,200);
```

※ドットシンタックスが使える場合は、このコマンドを使用する必要はありません。

※getPropertyというプロパティを参照するコマンドもあります。

tellTarget

- 階層の異なるMCにメソッドを送る場合には、ドットシンタックスが使えませので、「tellTarget」を使用します。

tellTarget

例)

```
_root.dog.arm.gotoAndPlay(50);
```

↓

```
tellTarget("/dog/arm"){  
  gotoAndPlay(50);  
}
```

※現在の「with」コマンドに似ています。ターゲットパスの指定が、スラッシュシンタックスになっているのに注意してください。

Call

- Flash4の時代には、「function」がありません。「function」のように一定量の命令のソースを使いまわしするには「call」を使用します。

call

例)

//フレーム50(フレーム名「meirei」)にあるキーフレーム
に記述されている命令。

```
penguin_mc._rotation = theKakudo;
```

```
//1フレーム目にある、ボタンにつけたコマンド  
on(press){  
    theKakudo = 45;  
    call(50);  
    //call("meirei");でも可  
}
```


phpとming

- PHPというサーバーサイドプログラムには「ming」というライブラリが入っています。
 - (古いPHPのバージョンの場合はライブラリを追加する必要があります。PHP4には入っています)
 - PHPが動作する環境でも、ライブラリを使用できるように設定されていなければ「ming」は適用されません。
- mingを使用すると、PHPのコードからswfを作成することができます。

mingの利点

- サーバー上で動的にswfを作成するため、PHPで外部ファイルや、URLに変数を設定して、swf内にテキストを埋め込むことができます。

SwfからPHPのコードを作成する

- 1、swfファイルから、phpを作成できるアプリケーション「swftophp」をダウンロードする

swftophp.exe

<http://www.opaque.net/ming/ming-winutils.zip>

SwfからPHPのコードを作成する(2)

2、ディレクトリを移動

今回は、ダウンロードした「swftophp.exe」を、
c:\¥htdocs へ移動させました。

※場所は特に指定があるわけではありません。

SwfからPHPのコードを作成する(3)

3、変換するswfファイルの作成

- 今回の例ではFlash Lite1.1で書き出した、ファイル名を「test.swf」とします

SwfからPHPのコードを作成する(4)

4、Dosプロンプトを立ち上げ、swftophpを起動させる

- `cd c:¥htdocs` としてディレクトリ移動
- `c:¥htdocs> swftophp.exe test.swf>test.php` としてexeファイルを起動
- `c:¥htdocs`のディレクトリに、`test.php`ができているかを確認する

SwfからPHPのコードを作成する(5)

5、出来たPHPを開き、コードを変更

- 使用されているmingのバージョンにもよるかと思いますが、通常は5以上での動作をさせる可能性があるので、

```
ming_useswfversion (4);
```

というコードを、「`$m = new SWFMovie();`」の前に書きます

SwfからPHPのコードを作成する(6)

6、PHPのコードを追加

- 「`ming_useswfversion (4);`」の前にPHPのコードを追加します。

例) 外部のファイルを開きデータを読み込み、変数を設定するコードを追加。その変数をFlashのコード内にある変数に設定する。外部書類を更新するだけで、文字が変更されるFlashが作成される。サーバーサイドで文字の埋め込みをしたFlashが作成されるため、待ち受け画面としても登録可能。