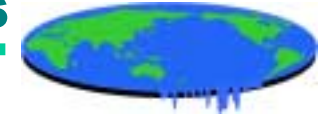


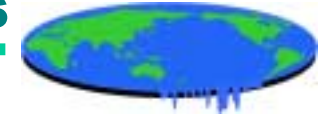
ストリーミングシステム (I) プロトコルとコンテンツ制作

(有)クロスエッジ・システムズ
渡辺 敦
nabe@crossedge.jp



1. アジェンダ

- Windows Media 9 Series
メタファイルと再生リストファイル
- Flash Communication Server
Streaming用ActionScriptの記述



2.Windows Media 9メタファイルと再生リストファイル

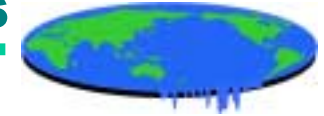
Windows Media9では従来からのASXファイルをメタファイルとして使います。また、9からは「再生リスト」と呼ばれる一部SMILをサポートしたファイル形式が導入されました。

ASXファイル(拡張子.asx)

- ・WebブラウザからWindows Media Playerに制御を移すためのメタファイル
- ・Windows Media Playerのみが読解可能なXMLマークアップ言語で記述される。

再生リストファイル(拡張子.wsx)

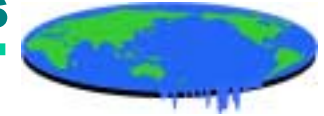
- ・W3C標準のSMIL記述とは少し異なる。
- ・Windows Media Serviceでは再生リストと呼ばれている。
- ・ローカルでは再生できない。



3.ASXファイルで可能なこと

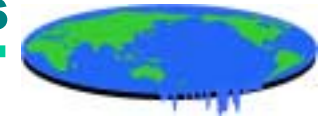
- ・タイトル、コピーライト表示などのメタ情報の付加
- ・複数のメディアの連続再生
- ・繰り返し再生
- ・再生時間の指定
- ・再生開始時間の指定
- ・静止画像の表示
- ・ハイパーリンクの付加

- ・イベント
- ・HTMLの表示

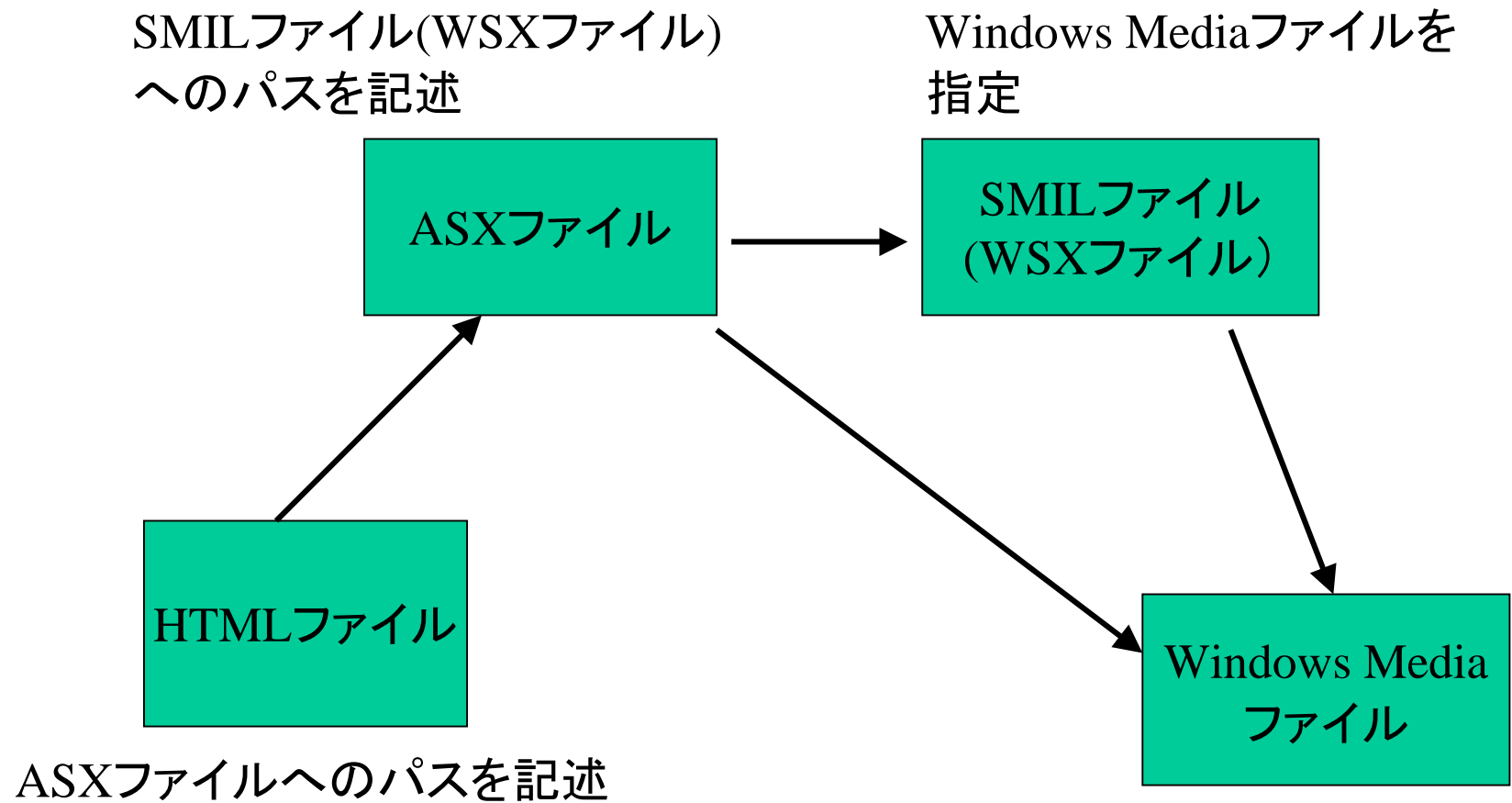


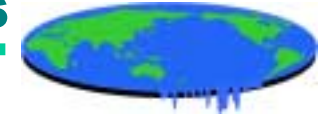
4.再生リストファイル(SMIL)で可能なこと

- ・タイトル、コピーライト表示などのメタ情報の付加
- ・複数のメディアの連続再生
- ・繰り返し再生
- ・再生時間の指定
- ・再生開始時間の指定
- ・静止画像の表示
- ・ハイパーリンクの付加



4.ASXファイルとSMILファイルの関係

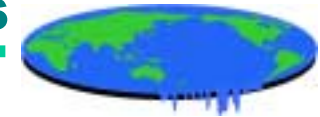




6.ASXファイルの基本的な記述方法

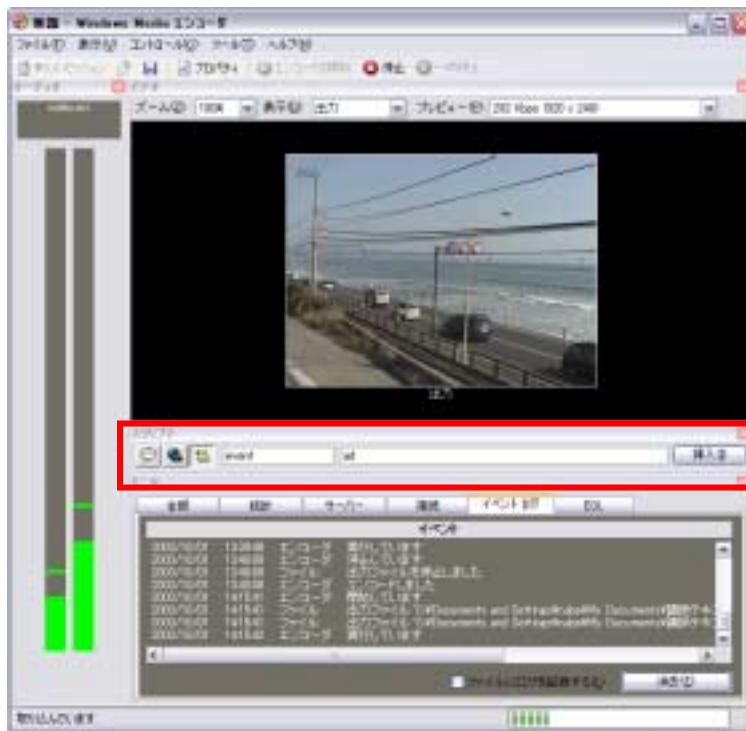
```
<asx version= " 3.0 " >  
    <entry>  
        <ref href= " video/SampleFile.wmv " />  
    </entry>  
</asx>
```

<asx>と</asx>タグの間に<entry>タグと</entry>タグを記述し、その間に<ref>タグでメディアファイルのロケーションを指定する。大文字・小文字は区別しません。



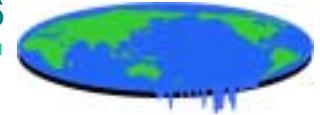
7. イベント(1)

イベントとは、Windows Mediaエンコーダーのスクリプトパネルで挿入されたラベルとASXファイル内の「EVENT」要素で定義された「name」属性を関連させて、「EVENT」要素で定義された動作をさせるものです。



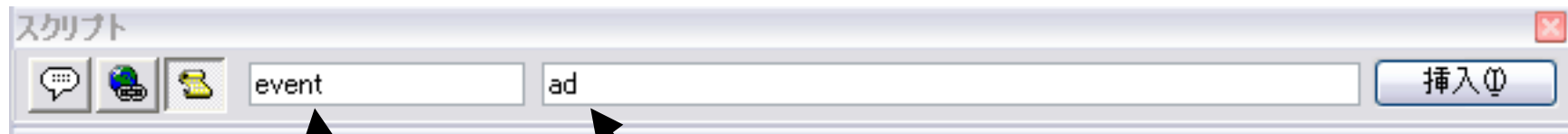
エンコーダー側

ユーザー設定スクリプトコマンド



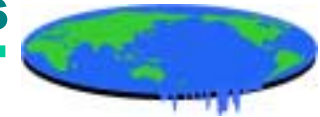
8. イベント(2)

スクリプトパネルの拡大図



「event」と入力

任意の文字列、ここでは
「ad」と入力

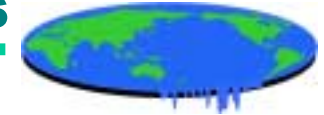


9. イベント(3)

ASXファイル側

```
<asx version="3.0">
<entry>
  <ref href="eventtest.wmv" />
</entry>
<event name="ad" whendone="resume">
  <entry>
    <ref href="ad.wmv" />
  </entry>
</event>
</asx>
```

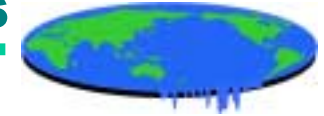
この場合、最初にeventtest.wmvが再生されますが、エンコーダー側のeventを挿入した時間で、プレイヤーは、そのname属性の値がadになっている<event>要素を実行します。したがって、この場合は、ad.wmvが再生されます。またwhendone属性がresumeになっているので、ad.wmvが終了した後は再びeventtest.wmvが一時停止した時点から再生されます。



10. イベント(4)

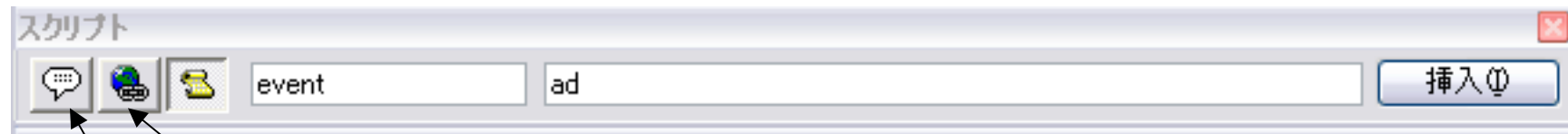
whendone属性の値

- resume 現在の項目 (イベントに割り込まれたクリップ) の再生を再開。コンテンツがライブ場合は、現在位置から再開されます。
- next イベントが発生せずにかつたかのように、次の **ENTRY** 要素が再生される。
- break 現在の項目が **REPEAT** ループ内にある場合は、繰り返しの回数が終了したかのように、ループが終了します。ループ内がない場合は、最後の項目が通常どおり終了したかのように、再生リストの最後にジャンプします。



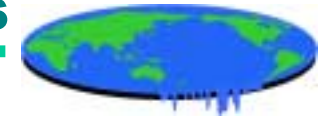
11.他のスクリプト

スクリプトパネルの拡大図



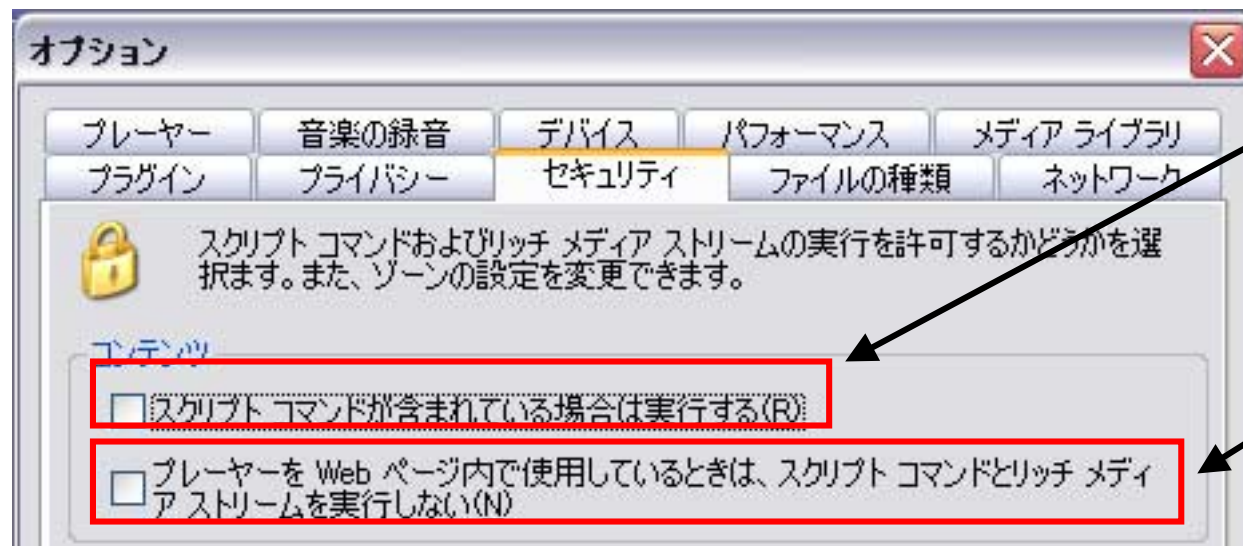
URL:メディアファイルと連動してブラウザに表示させたいURLを挿入する。

TEXT:プレーヤーに表示させたい自由形式のテキストを挿入する。



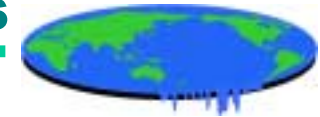
12.スクリプト挿入における注意点(1)

Windows Media Player9では、セキュリティの強化(?)がなされていて、プレイヤーの「ツール」->「オプション」->「セキュリティ」でスクリプトの実行をするかどうかを設定できます。



単体プレーヤー
での動作

Embedded型
プレーヤー
での動作



13. スクリプト挿入における注意点(2)

・単体プレーヤー

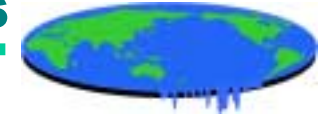
「スクリプトコマンドが含まれている場合は実行する」オプションのチェック

	デフォルト	Off	On
URLスクリプト :		×	
TEXTスクリプト :			
ユーザー設定スクリプト :			

・Embeddedプレーヤー

「プレーヤーをWebページ内で使用しているときは、スクリプトコマンドとリッチメディアストリームを実行しない」オプションのチェック

	デフォルト	Off	On
URLスクリプト :			×
TEXTスクリプト :		×	×
ユーザー設定スクリプト			



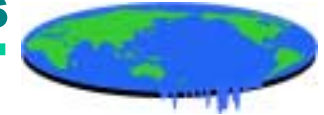
14.HTMLの表示(1)

WebコンテンツをWindows Media Playerに表示する機能をHTMLViewと呼びます。この機能はPlayer9以外では動作しません。

以下のように<param/>タグで表示したいHTMLページを指定します。

<ref/>タグではWindows Media オーディオファイルを指定しています。この場合、HTMLページがWindows Media Playerに表示され、さらにオーディオが流れます。

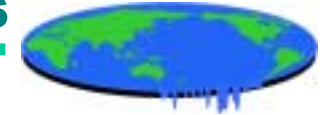
```
<asx version="3.0">  
<param name="HTMLView" value="http://www.microsoft.com"/>  
<entry>  
  <ref href="rtsp://wmt.server.com/audio.wma"/>  
</entry>  
</asx>
```



15.HTMLの表示(2)

- ・HTMLViewに表示するHTMLの中に、WMP ActiveXコントロールを追加する。
- ・エンベッド型プレイヤーと異なり、<object>タグのurlパラメータは使わない。
- ・ビデオはASXファイルの<ref/>タグで指定する。
- ・ページサイズは575x345が推奨されている。

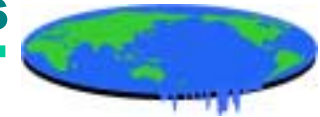
```
<html>
<body bgcolor="white">
これはテストです<br>
  <object id = "Player"
    CLASSID = "CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6">
      <param name="autoStart" value="true">
      <param name = "uiMode" value = "none">
<!-- これは使わない<param name="url" value="hoge.asx"> -->
  </object>
</body>
</html>
```

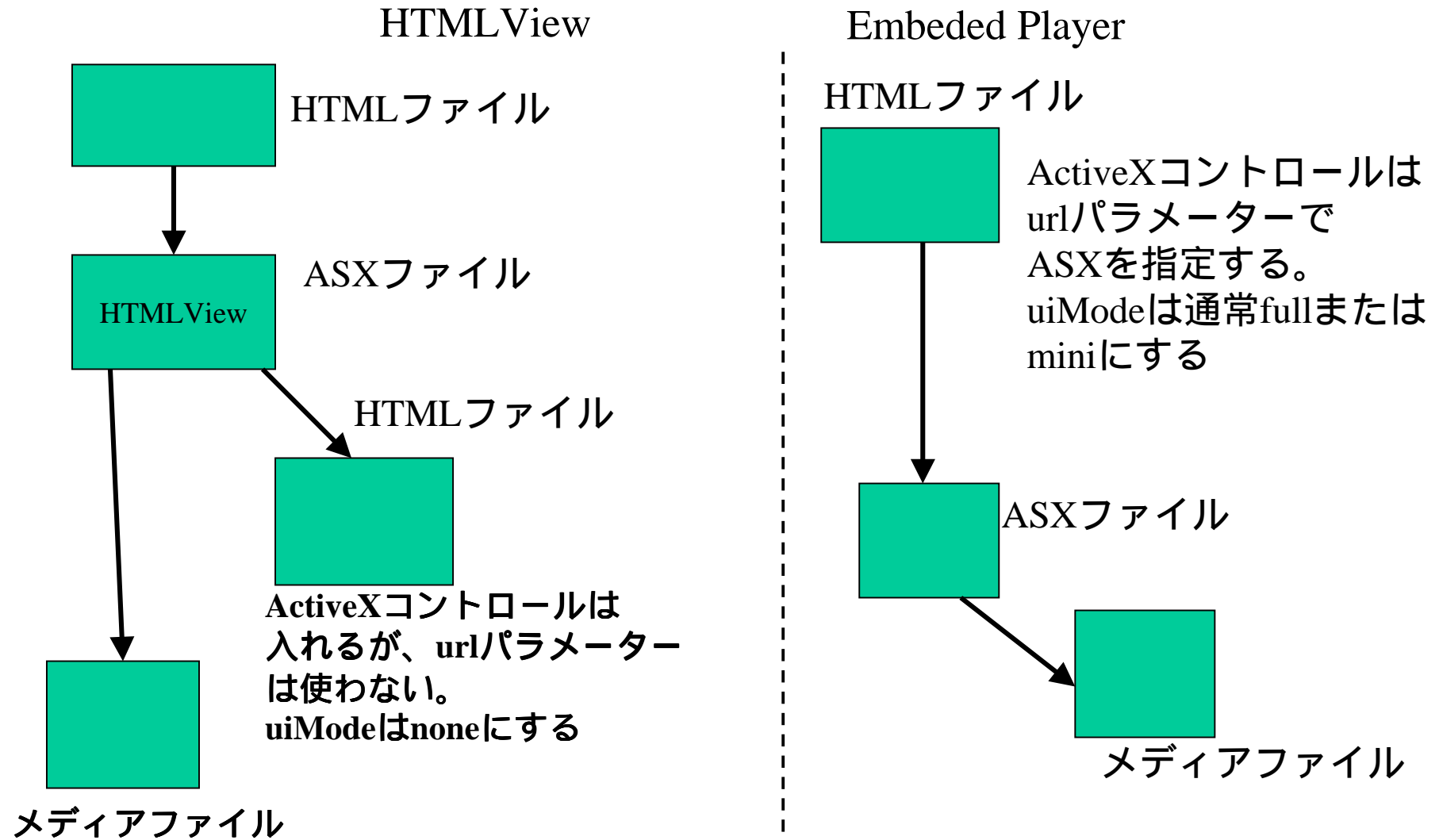
16.HTMLの表示(3)

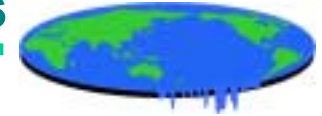
- ・ASXファイルは、<param>タグのname属性にHTMLViewを指定する。
- ・value属性で表示したいurlを指定する。

```
<asx version="3.0">  
<param name="HTMLView"  
        value="http://www.hogecom/index.html"/>  
<entry>  
    <ref href="video.wmv"/>  
</entry>  
</asx>
```



19.HTMLViewとEmbeddedプレイヤー





18.HTMLViewにおけるスクリプト挿入の注意点

- ・単体プレーヤー
「スクリプトコマンドが含まれている場合は実行する」オプションのチェック

デフォルト : Off On

URLスクリプト : ×

TEXTスクリプト :

ユーザー設定スクリプト :

HTMLView
での動作

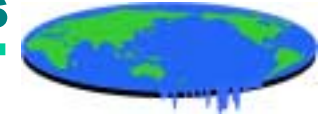
- ・Embeddedプレーヤー
「プレーヤーをWebページ内で使用しているときは、スクリプトコマンドとリッチメディアストリームを実行しない」オプションのチェック

デフォルト : Off On

URLスクリプト : ×

TEXTスクリプト : × ×

ユーザー設定スクリプト

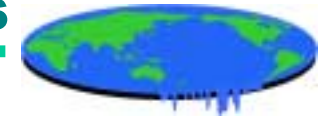


19.再生リスト(SMIL)の基本的な記述方法

以下は再生リスト(SMIL)の基本的な記述方法ですが、W3C規格のSMILの記述方法とは若干異なります。

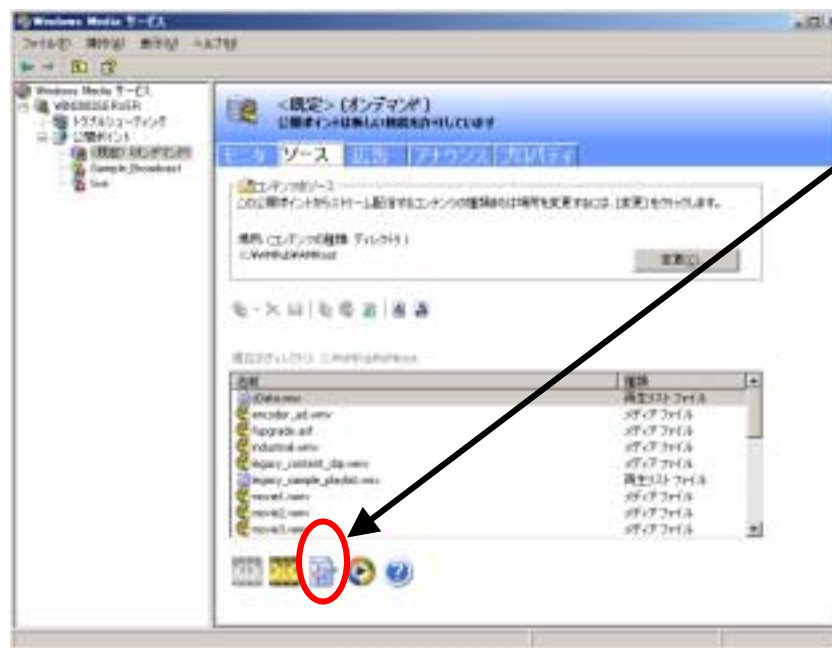
```
<?wsx version="1.0"?>
<smil>
  <media src="racecar_300.wmv" />
  <media role="Advertisement" noSkip="TRUE" src="encoder_ad.wmv" />
  <media src="snowboard_300.wmv" />
  <media src="industrial.wmv" />
</smil>
```

- ・拡張子は.wsxとします。
- ・ファイルの1行目に <?wsx version="1.0"?> と記述します。
- ・大文字小文字は区別しません。



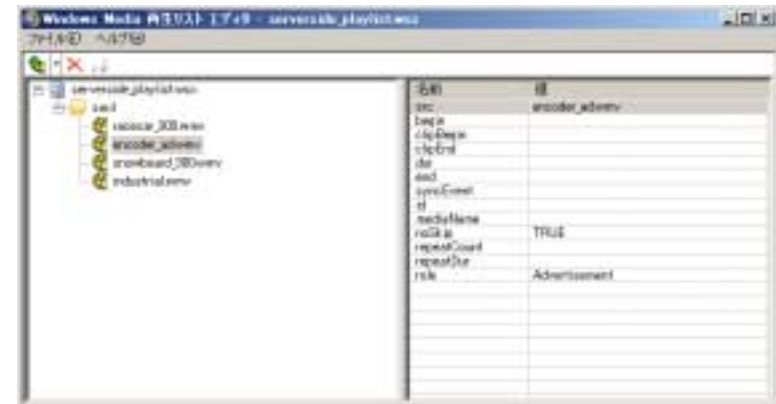
20.再生リストの編集

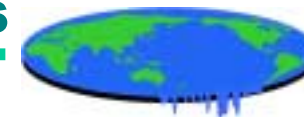
Windows Media Service9には「再生リストエディタ」が付属しています。再生リストエディタはWindows Mediaサービス管理コンソールの「ソース」タブを開き、「再生リストエディタの表示」をクリックします。



再生リストエディタの表示

エディタが開きます。

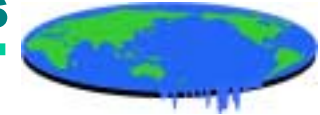




21.再生リストの要素

再生リストの要素には以下があります。

要素	機能
smil	SMILであることを示します。
media	メディアの場所を指定します。
seq	順次再生の再生順を指定します。
switch	代替コンテンツを指定します。
excl	排他的な再生順を指定します。
priorityClass	あるコンテンツが別のコンテンツの再生を中断する方法や、中断されたコンテンツの状態を制御します。
clientData	説明テキスト、バナー、ログ情報を再生リストのコンテンツと関連づけます。

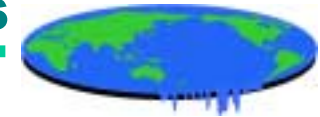


22.再生リストの具体例(1)

media要素の具体例を示します。

```
<?wsx version="1.0"?>
<smil>
  <media src="C:¥WMPub¥WMPRoot¥industrial.wmv" />
  <media src="C:¥WMPub¥WMPRoot¥encoder_ad.wmv"
    role="advertisement" noSkip="TRUE" />
</smil>
```

Src属性でメディアファイルのパスを指定します。
Role属性には任意の値を設定でき、その値でログに記録されますが、「advertisement」だけは特別な意味を持ち、この値をもつメディアがストリーム配信されるたびに、Windows Mediaサービスの管理コンソールの「モニタ」の「広告」の値が1つずつ増えます。

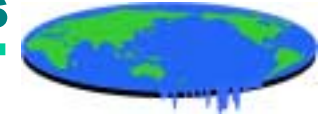


23.再生リストの具体例(2)

switch要素の具体例を示します。

```
<?wsx version="1.0"?>
<smil>
  <switch>
    <media id="media1" src="http://encoder1:8080?WMReconnect=0" />
    <media id="media2" src="Alternate.wmv" />
  </switch>
  <switch>
    <media id="media3" src="http://encoder2:8080?WMReconnect=0" />
    <media id="media4" src="Alternate.wmv" />
  </switch>
</smil>
```

この場合、通常はmedia1の再生が終了すると、media3の再生に移動しますが、Media1がうまく再生できない場合に、代わりにmedia2を再生してから、media3に移動します。その場合WMReconnect=0によってすばやく切り替わります。

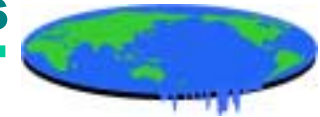


24.再生リストの具体例(3)

excl要素の具体例を示します。

```
<?wsx version="1.0"?>
<smil>
  <excl>
    <media id="media1" src="Video1.wmv" begin="media4.end" />
    <media id="media2" src="Video2.wmv" begin="0s" />
    <media id="media3" src="Video3.wmv" begin="media2.end" />
    <media id="media4" src="Video4.wmv" begin="media3.end" />
  </excl>
</smil>
```

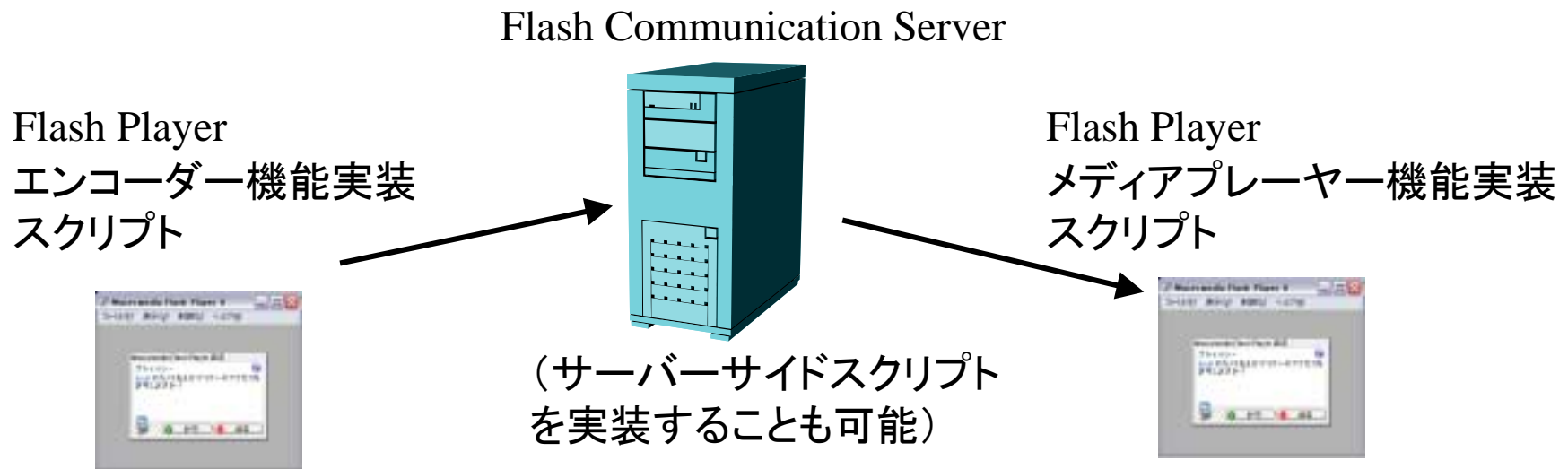
excl要素内のメディアは再生順が決まっておらず、属性によって再生順が決まります。この例では、media2が最初に再生され、media3、media4、media1の順番で再生されます。

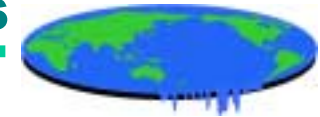


25. Flash Communication Server

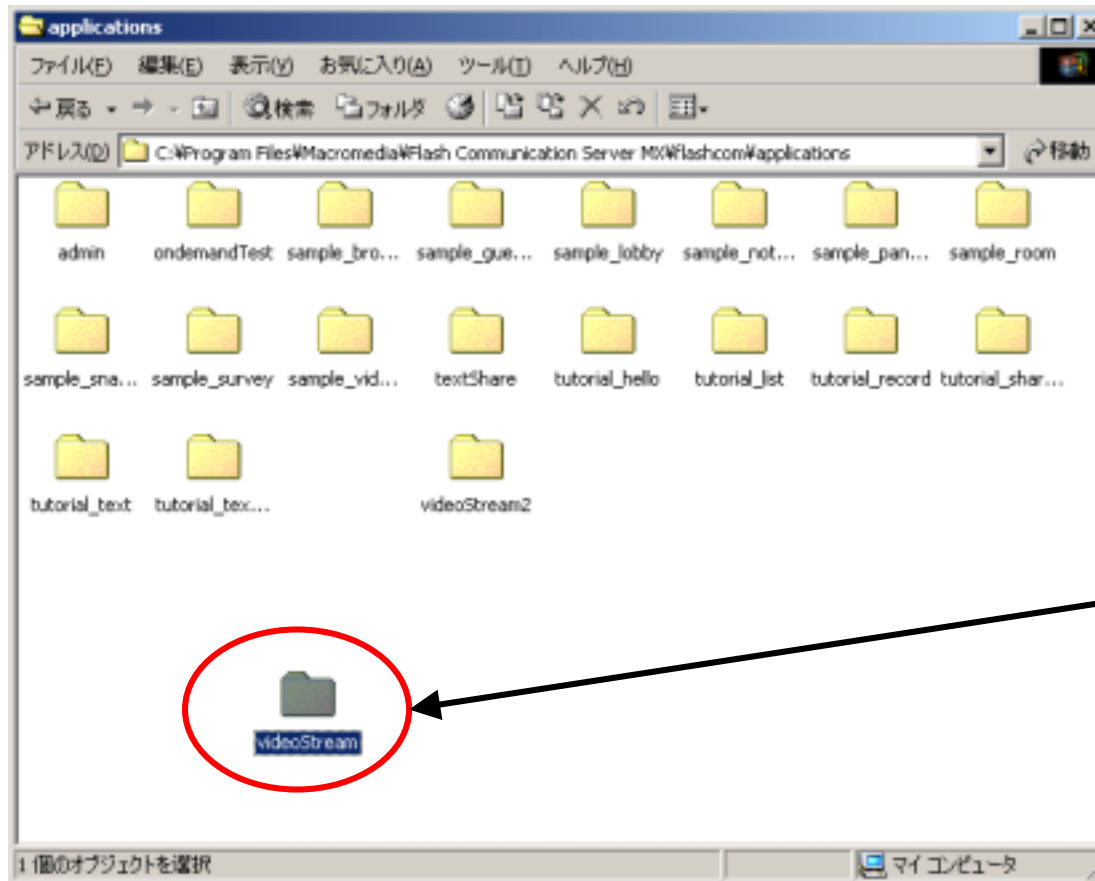
Flash Communication Serverはストリーミング専用サーバーではなく、クライアントをFlash Playerとする汎用アプリケーションサーバーである。

したがって、WMT、HelixとことなりFlash用のスクリプト(アクションスクリプト)を記述して、プレーヤーおよびエンコーダー等の機能を実装することになります。





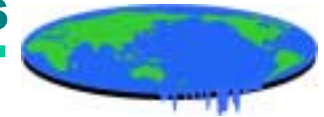
26. Flash Communication Server – サーバーの設定



ここでは、
VideoStream
としています

```
%FCS%\flashcom\applications
```

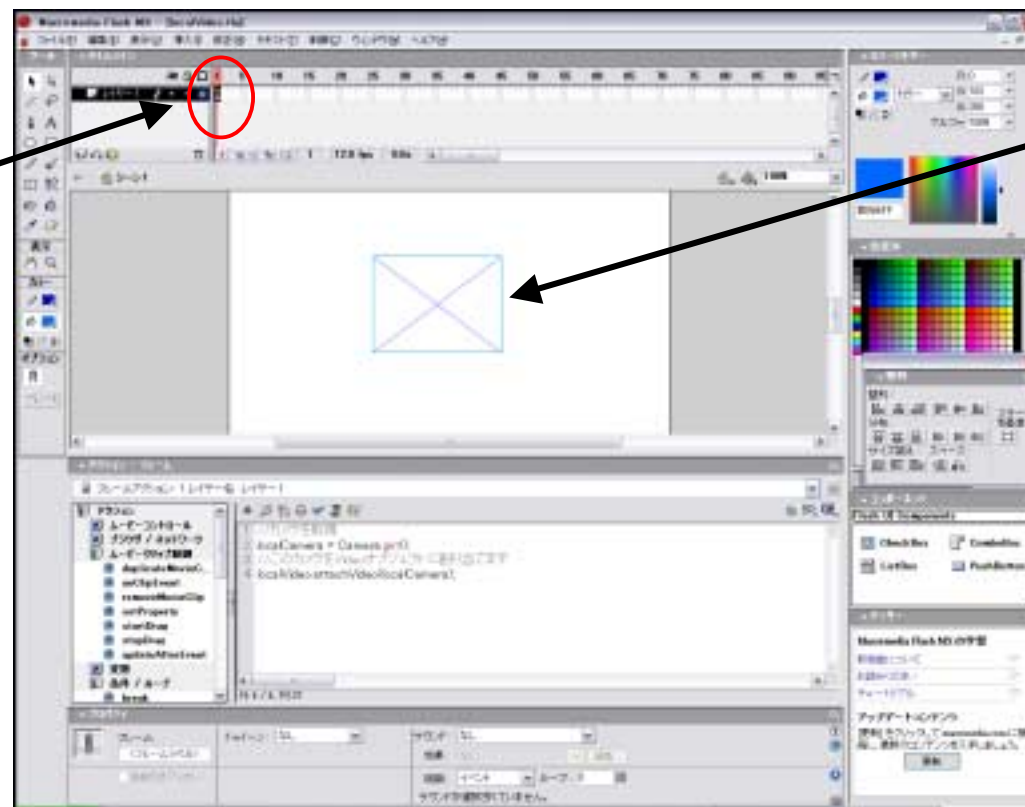
ディレクトリに、アプリケーション名をつけたディレクトリを作成します。



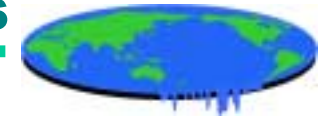
27. Flash Communication Server – エンコーダーの実装(1)

```
//カメラを取得してVideoオブジェクトに割り当てます
localCamera = Camera.get();
localVideo.attachVideo(localCamera);
```

フレーム1
にスクリプト
を書きます



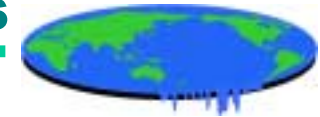
ライブラリから
新規ビデオを
作り、それを
ステージ上に
配置し、インス
タンス名を
localVideo
とする。



28. Flash Communication Server – エンコーダーの実装(2)

```
//サーバーへの接続
nc=new NetConnection();
nc.onStatus = function(response) {
    if(response.code == "NetConnection.Connect.Success") {
        publishVideo(); // ビデオの配信
    } else {
        // 接続に失敗
    }
};
nc.connect("rtmp://videoStream");//ローカルサーバーの場合
```

NetConnectionオブジェクトを生成し、イベントハンドラメソッドonStatusを定義します。ここにサーバーからの応答により、処理を分ける記述をします。接続が成功した場合は、NetConnection.Connect.Successという文字列が返されるので、ビデオの配信メソッドpublishVideo()を呼ぶようにします。このメソッドは、ユーザー定義メソッドです。定義が終了したら、実際にアプリケーション名を指定してサーバーに接続します。

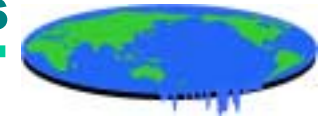


29. Flash Communication Server – エンコーダーの実装(3)

```
// ビデオの配信メソッド
function publishVideo() {
    // nc上にNetStreamオブジェクトを生成
    ns=new NetStream(nc);
    //カメラを割り当てる
    ns.attachVideo(localCamera);
    // 配信開始
    ns.publish("video1");
}
```

実際にビデオをサーバーにストリーム配信するメソッドであるpublishVideo()の定義をします。

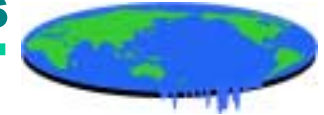
NetConnectionオブジェクト上にストリーム回線であるNetStreamオブジェクトを生成します。これに取得したカメラを割り当て、publish()メソッドで配信をします。Publish()メソッドの引数には、ストリーム名を指定します。これは受信側でストリームを識別するためのものです。



30. Flash Communication Server – メディアプレーヤーの実装(1)

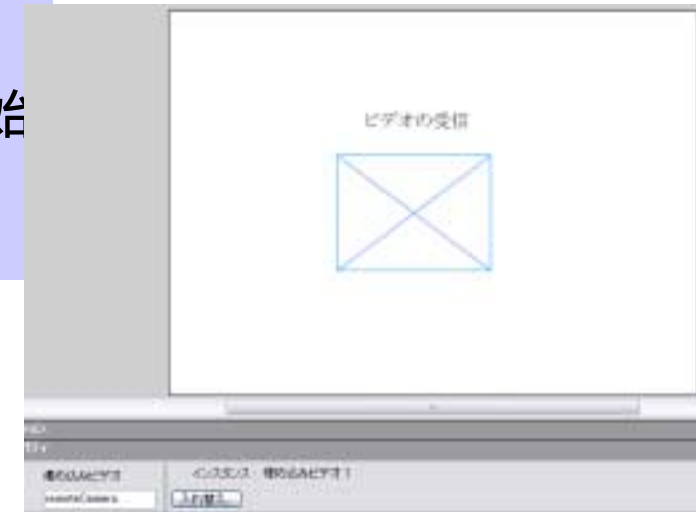
```
//サーバーへの接続
nc=new NetConnection();
nc.onStatus = function(response) {
    if(response.code == "NetConnection.Connect.Success") {
        recieveVideo(); // ビデオの受信
    } else {
        // 接続に失敗
    }
};
nc.connect("rtmp://videoStream");//ローカルサーバーの場合
```

受信側でも同様に、NetConnectionオブジェクトを生成し、onStatusイベントハンドラメソッドを定義し、connectメソッドでサーバーに接続をします。
また、受信側では、recieveVideo()というユーザー定義メソッドで受信したビデオをプレーヤーに表示します。

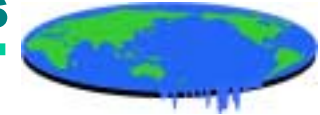


31. Flash Communication Server – メディアプレーヤーの実装(2)

```
//ビデオの受信//  
function recieveVideo() {  
    //受信用NetStreamオブジェクトの生成  
    ns = new NetStream(nc);  
    //それをVideoオブジェクトに割り当て  
    remoteCamera.attachVideo(ns);  
    //publish時の名前を指定して、受信開始  
    ns.play("video1");  
}
```



受信用のNetStreamオブジェクトをNetConnection上に生成します。それをステージ上のビデオインスタンスに割り当てて、プレイメソッドで配信時の名前を指定します。



32. Flash Communication Server – 帯域幅の制御

```
//カメラを取得してVideoオブジェクトに割り当てます
localCamera = Camera.get();
localVideo.attachVideo(localCamera);

//帯域幅の設定、デフォルトは16384Bps=131Kbps
localCamera.setQuality(4000,0); //64Kbpsに設定

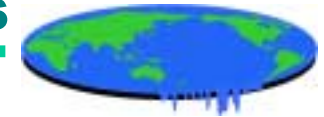
//帯域幅の取得
currentBandWidth = localCamera.bandwidth;
```

- ・ 帯域幅の設定

カメラオブジェクトのsetQuality(bandwidth, quality)メソッドで設定します。

- ・ 帯域幅の取得

カメラオブジェクトのbandwidthプロパティで取得します。

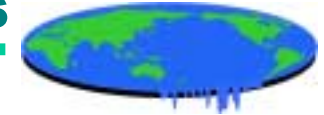


33. Flash Communication Server – 接続数の制御(1)

- クライアントの接続数はサーバーサイドスクリプトで制御します。
- Flash Communication Serverのインストールディレクトリのアプリケーションディレクトリの中に「main.asc」というファイルをつくり、その中にスクリプトを記述します。
- 「main.asc」はUTF-8で保存する必要があります。
- サーバーサイドスクリプトの記述はJavaScriptに「ECMA-262」に準拠しますが、Flash Communication Server独自のオブジェクトやメソッドを使います。



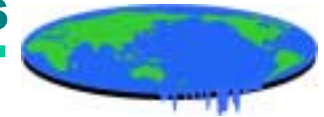
例えば、「videoStream2」というアプリケーションを定義して、そのディレクトリの中にmain.ascをおきます。



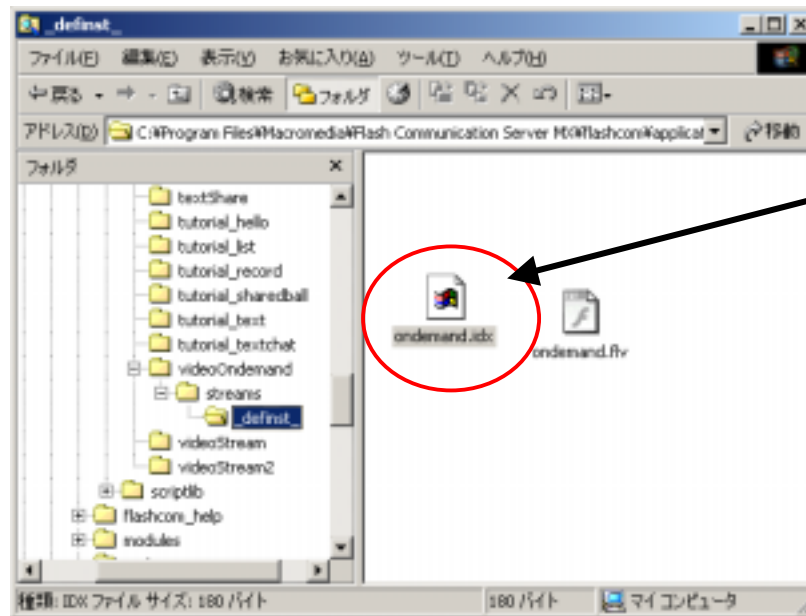
34.Flash Communication Server – 接続数の制御(2)

```
application.onConnect = function(clientUser) {  
    if(application.clients.length >=20) {  
        return false;  
    } else {  
        return true;  
    }  
}
```

- プレーヤーが接続してくると、Applicationオブジェクトが生成され、変数applicatinに格納される。
- さらに、そのonConnectイベントハンドラメソッドがClientオブジェクト引数で呼び出される。
- プレーヤーの接続数はapplication.clients.lengthプロパティで取得する。
- trueまたはfalseを返すことによって、接続を許可、解除させる。

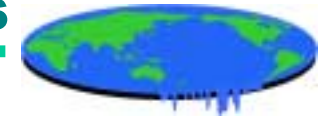


35. Flash Communication Server – オンデマンド配信(1)



この拡張子「.idx」
というファイルは、
Flvファイルにアクセス
された時に自動的に
作成される。

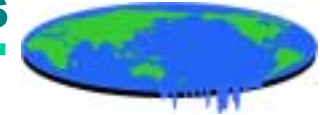
- ・ オンデマンド用のファイルは、Sorenson Squeeze for Flash MX または Flash MXを使い、拡張子「.flv」のファイルにエンコードする。
- ・ %FCS%¥flashcom¥applications¥アプリケーション名¥streams¥_definst_ というディレクトリに、オンデマンド用のファイルを置く。



35. Flash Communication Server – オンデマンド配信(2)

```
//サーバーへの接続
nc=new NetConnection();
nc.onStatus = function(response) {
    if(response.code == "NetConnection.Connect.Success") {
        publishVideo(); // ビデオの配信
    } else {
        // 接続に失敗
    }
};
nc.connect("rtmp://video0ndemand"); // ローカルサーバーの場合

//ビデオの受信//
function recieveVideo() {
    ns = new NetStream(nc);
    remoteCamera.attachVideo(ns);
    //flvの拡張子を除いたファイル名を指定して、受信開始
    ns.play("ondemad");
}
```



まとめ

- Windows Media 9
 1. Mediaプレイヤー内にHTMLの表示が可能になった。
 2. SMILが一部サポートされた。
 3. セキュリティ機能の変更でデフォルトのスクリプト動作が変わった。
- Flash Communication Server
 1. 全てAction Scriptでエンコーダ機能、プレイヤー機能を実装する。
 2. 基本的なScriptの記述は、定型である。
 3. クライアントサイド、サーバーサイドでのScriptの記述がある。