URLLoader & XML

URLLoader Class

- uses a request -response approach
- process cannot be intiated by server
- if data change in DB the app must request again
- loads external files
- can be text, text containing name/value pairs or well formed XML
- all info MUST finish loading before it is avaialbe in the swf app

Properties of the URLLoader

Property	Data Type	Description
bytesloaded	uint	number of bytes loaded so far
bytesTotal	uint	total number of bytes to be downloaded, populated when complete
data	Mulitple	data recieved when load() completer
dataFormat	String	format expected for data: URLLoaderDataFormat.TEXT, URLLoaderDataFormat.BINARY, URLLoaderDataFormat.VARIABLES

DataFormat

- TEXT - expect string
- BINARY - expects ByteArray
- VARIABLE - epects name/value pairs
- for XML the default is TEXT
- to use E4X it must be cast as a XML object
- the URL can be a static XML doc or a server side file that create a XML Stream

Methods of URLLoader

Method	Parameters	Description	Return
close()		closes the current load event	Nothing
load()	request:URLRequest	sends & loads data to the specified URLRequest	Nothing
URLLoader()	request:URLRequest	Constructor method	Nothing

Events of URLLoader

Event	Type	Description
complete	Event	fired after all data is received and placed in data property
httpStatus	HTTPStatusEvent	FIRED IF LOAD(0 ATTEMPT TO DO SO OVER HTPP
ioError	IOErrorEvent	fired if load() result in error and ends download
open	Event	fire once load has started after the call
progress	ProgressEvent	fired as download data is recieved & decoded
securityError	SecurityErrorEvent	fired if load attempts to load data outsifde the security sandbox

Limits

- restricted to the get and post method
- cannot use to do put or delete

Creating

- step 1 create the loader with the constructor method
- you can make the request at the same time as instanitating but calling the load() is more flexible(and reliable)
  - var myLdr:URLLoader = new URLLoader();
- you can specify data type
  - myLdr.dataFormat = URLLaoderDataFormat.TEXT;
- you do not need to do this TEXT is default for XML

Text

Request

```
var myRequest:URLRequest = new URLRequest("myFile.xml")
var myLdr:URLLoader = new URLLoader();;
myLdr.load(myRequest)
```

default load method is GET so you can set POST

```
var myRequest:URLRequest = new URLRequest("myFile.xml")
myRequest.method = URLRequestMethod.POST;
var myLdr:URLLoader = new URLLoader();;
myLdr.load(myRequest)
```

To Request XML Stream

```
var strURL:String =”http://www.domain.com/xmlStream.ext”
var myRequest:URLRequest = new URLRequest(strURL)

var myLdr:URLLoader = new URLLoader();;
myLdr.load(myRequest)
```

- you MUST use the fully qualified path to request a XML Stream generate by a server-side file
- include path allow the server to do the parsing before it is deliverd to your swf

Send Vars

```
var strURL:String =”http://www.domain.com/xmlStream.ext”
var myRequest:URLRequest = new URLRequest(strURL)
var params:URLVariables = new URLVariables();
params.continent = “Europe”
params.timeStamp = new Date().getTime();
request.data = params;
```

Flash Security

- based on the location of the swf and the data being load
- a swf can acces any data from a sub domain as it's own location
- you cannot load data from a different domain or subdomain

Security SandBox

- Flash equates to the exact domain
- these are considered separate sandboxes
  - http://www.friendsofed.com
  - http://friendsofed.com
  - http://eaxmples.friendsofed.com
  - http://65.19.150.101
- if though the IP may resolve to the 1st one on the list it is considered a separte sandbox

Cross-Domain policy

- to over come this we have the cross domain policy file
- it is a xml file called “crossdomain.xml”
- it lives on the server that hosts the external data
- it grant permission to certian domains acces to the files

Cross Domain

```
<?xml version =”1.0”?>
<cross-domain-policy>
<allow-access-from domain = “www.friendofed.com”/>
<allow-access-from domain = “*.friendofed.com”/>
<allow-access-from domain = “65.19.159.101”/ >
</cross-domain-policy>
```

Wildcard

```
<?xml version =”1.0”?>
<cross-domain-policy>
<allow-access-from domain = “*”/>
</cross-domain-policy>
```

Issues

- FP 10 will only recognise cd when content type set to ‘text” (text/*)
- or application/xml , application/xhtml+xml
- so check server settings
- FP will reject files that are not well formed
- root elemtent MUST be <cross-domain-policy>
- no element can contain text children
- no chacater before the open or after the close tag

Using Proxy

- if you cannot get cross domain you can use a server side proxy file
- as long as the server side proxy file is in same domain as SWF you will be able to load dat

proxy

```
<?php

$post_data = $HTTP_RAW_POST_DATA;

$header[] = "Content-type: text/xml";
$header[] = "Content-length: ".strlen($post_data);

$ch = curl_init( $_GET['url'] );
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 10);
curl_setopt($ch, CURLOPT_HTTPHEADER, $header);

if ( strlen($post_data)>0 ){
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
}

$response = curl_exec($ch);

if (curl_errno($ch)) {
    print curl_error($ch);
} else {
    curl_close($ch);
    print $response;
}


?>
```

http://kb.adobe.com/selfservice/viewContent.do?externalID=tn_16520&sliceID=1