

# Actionscript 3.0 Basics

## 5 – Assets Management

# Flash Resource Management

Embedded in the application  
file.

vs

Load from external  
location.

# Embedding

## [Embed] Metadata Tag

Is a compiler directive that allows you to inject file data inbetween the as3 bytecode.  
(only for flex compiler)

# Embedding Images

```
1 [Embed(source="path/to/my-image-file.jpg")]
2 private static const MyImageFileAsset:Class;
3
4 var image:Bitmap = new MyImageAsset() as Bitmap;
5
6 [Embed(source="path/to/my-vectorial-image-file.svg")]
7 private static const MyVectorImageFileAsset:Class;
8
9 var image:Sprite = new MyVectorImageFileAsset() as Sprite;
10
```



# Embedding XML

```
11 [Embed(source="path/to/my-xml-file.xml")]  
12 private static const MyXMLFileAsset:Class;  
13  
14 var xml:XML = new XML(MyXMLFileAsset.data);  
15
```

*Bug: Embedded XML files can't have document type declarations.  
<?xml version="1.0"?>*

# Embedding Anything

```
16 [Embed(source="path/to/my-sound-file.mp3")]
17 private static const MySoundFileAsset:Class;
18
19 var sound:Sound = new MySoundFileAsset() as Sound;
20
21 [Embed(source="path/to/my-swf-file.swf", symbol="mySymbolName")]
22 private static const MySWFSymbolAsset:Class;
23
24 var sprite:Sprite = new MySWFSymbolAsset() as Sprite
25
26 var movieclip:MovieClip = new MySWFSymbolAsset() as MovieClip;
27
28 [Embed(systemFont="Ubuntu", fontName="EmbeddedUbuntu", mimeType="application/x-font")]
29 private var MyFontAsset:Class;
30
31 [Embed(source="path/to/anyFile.anyExt", mimeType="application/octet-stream")]
32 private var MyFileAsset:Class;
33
34 var data:ByteArray = new MyFileAsset() as ByteArray();
```

# Pros & Cons

Easy and straightforward  
Many formats supported  
Simpler Code  
No waiting for the assets to load

Increases the application size  
Exponentially increases the initialization time of the application.

# Loading from external location.

## Loaders

For General Purpose:

- `UrlLoader`

For Images and Swfs:

- `Loader`

For sounds:

- `Sound`

For Video

- `Video and NetStream`



# URLLoader

```
1  import flash.net.URLLoader;
2  import flash.net.URLRequest;
3
4  var loader:URLLoader = new URLLoader();
5  var request:URLRequest = ("path/to/file.ext");
6
7  loader.addEventListener(Event.COMPLETE, onLoadComplete);
8
9  loader.load(request);
10
11 function onLoadComplete(e:Event):void
12 = {
13     var loader:URLLoader = e.target as URLLoader;
14
15     trace(loader.data);
16 }
```

# Pros & Cons

Application size reduces.

While the assets load the user can still interact with the app.

No need to recompile if assets change.

Different apis for different kinds of assets.

Complex(ish) code.

Asset loading is restricted by security policies  
(no cross-domain requests for some file-types)

# Working with XML

XML & XMList

E4X Syntax

# Assets Libraries

```
1 var assetsLoader:Loader = new Loader();  
2 assetsLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, onAssetsLoaded);  
3  
4 var request:URLRequest = new URLRequest("assets.swf");  
5 var context:LoaderContext = new LoaderContext(false, ApplicationDomain.currentDomain);  
6  
7 assetsLoader.load(request, context);
```



# Working with Sound

Sound  
SoundChannel

# Thanks!

[github.com/matix/as3basics](https://github.com/matix/as3basics)

matias.figueroa@globant.com  
@matixfigueroa