
Native

Under the Skin

Why go native?

- Access platform features early.
- Faster data processing.
 - Faster mathematical operations.
 - Faster memory access.
- Access to existing libraries.

Why go native?

- Faster

Speed

- Can access ASM libraries.
- Can access SIMD operands.
- Code compiled by sane compilers. (Sometimes)
- Box2D, Nexus One, braindead ANE:

	AS Only	ANE
122 objects	~110ms	~25ms



The Good



The Bad



The Ugly

by ROB WORD

The Good

- If you use C or C++, your extension is pretty portable.

native

```
graph TD; native --- android; native --- box2d; native --- ios; native --- shared;
```

android (0 source files)

box2d (136 source files)

ios (0 source files)

shared (4 source files)

The Good

- For the ‘droids - it kicks the pants off JNI.

```
jclass cls = (*env)->GetObjectClass(env, obj);
jmethodID mid = (*env)->GetMethodID(env, cls, "callback", "(I)V");
if (mid == 0)
{
    return;
}
(*env)->CallVoidMethod(env, obj, mid, depth);
```

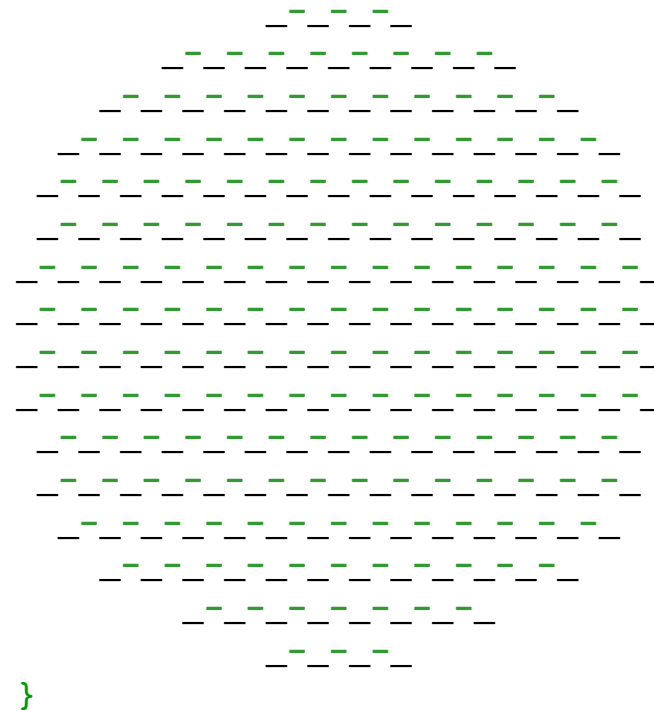
VS

```
FRECallObjectMethod(store, "callback", 0, NULL, &methodReturn, NULL);
```

The Good

- You get to play with C, which is awesome.

```
#define _ -F<00||--F-OO--;
int F=00,OO=00;main(){F_OO();printf("%1.3f\n",4.*-F/OO/OO);}F_OO()
{
```



The Bad

- Probably going to double your development environments.



The Bad

- Have to remember how to code in stupid languages. '\0' terminated strings dude.

```
char strA[] = "Dammit";  
char strB[] = " Janet";  
char strCombined[256] = "\0";
```

```
strncat(strCombined, strA, 256);  
strncat(strCombined, strB, 256);
```

The Bad

- Switching mental contexts.

```
public function foo():void
{
    int wut = inAS() ? 0 : 1;
}
```

The Ugly

- Code it wrong, explode your app.
- Lots of boiler plate code.
- Errors are cryptic.
- Say goodbye to source level debugging.

Design Sanity

ActionScript

- Make your interface an interface.
- Package your shareable source separately.
- Create at least a stub AS only version. Make it functional if you can.

ActionScript

- Do ActionScript things in ActionScript if you can.

```
public function getBody(  
    bodyID:uint,  
    createIfNeeded:Boolean = false):b2BodyProxy  
{  
    if(_bodies[bodyID] == null && createIfNeeded)  
    {  
        _bodies[bodyID] = new b2BodyProxy();  
    }  
  
    return _bodies[bodyID];  
}
```

Native

- Write once, compile everywhere.
- Abuse the `ActionScriptData` object.
- Take advantage of the fact you control the `ActionScript` interface for validation.
- Keep the interface boundary thin, build complexity at the sides.

All Together

- You control the ActionScript and native boundaries. Make them dance a beautiful dance.
- Abuse the Status event for communication.
 - Logging
 - Errors
 - General WTF investigation (printf is your debug)

Gotchas and Tips

iOS + AIR × ANE = suck

- Optimisation level can break things.
- AIR recompilation step can break function pointers.
- Choose the right compiler.

Script Everything

Android is *your* friend

ANE is newly open

- Documentation isn't perfect.
- Some things don't work.
- Very small, but growing, community.

Unanswered Questions



Including VM/native boundary,
is ANE faster than Alchemy?

Flash + ANE is more accessible
to developers than Java + JNI. It's
also cross platform.

Implications for Android?

1. HaXe
2. Air Native Extensions
3. NME
4. ???
5. Profit

Buildr, Buildr, Buildr

Libraries

- Third party platform libraries
 - Facebook
 - Open Feint
 - Analytics
 - BLAS, LAPACK
 - fmod