# Advanced OpenFL project configuration

Configuration-as-code with project.hxp



#### FEATURED.SPEAKER



Adam Breece, FlowPlay

LEARN MORE AT https://summit.haxe.org/us/2018/



### What is a project.hxp file?

- OpenFL project configuration written in haxe
- Complex logic not possible in project.xml
- Use shared code and haxelibs

#### Example project.hxp

```
import lime.project.*;
class Project extends HXProject {
  public function new () {
       super ();
       app = { main: "piratepiq.PiratePiq", file: "PiratePiq", path: "Export"};
      meta = { title: "Pirate Pig", packageName: "com.flowplay.PiratePig", version: "1"};
       sources.push ("Source");
```

#### Sound assets

```
public static function getSounds(soundDir:String):Map<String,String> {
     var soundExt = "mp3";
     var sounds = new StringMap();
     FileSystem.readDirectory(soundDir)
        .iter(function (i:String) {
            var idRx = new EReg("(^[0-9]+)(\\." + soundExt + ")", "ig");
            if (!idRx.match(i)) return;
            var num = idRx.matched(1);
            sounds.set("sound"+num, soundDir + "/" + i );
        });
                                 <sound path="3.mp3" id="sound3" />
    return sounds;
```

#### Icon assets

```
public static function getMobileIcons(iconDir:String):Array<Icon> {
   var icons = new Array<Icon>();
   FileSystem.readDirectory(iconDir)
       .iter(function (i:String) {
           var sizeRx = \sim / (^[0-9]+) (\land.png)/ig;
           if (!sizeRx.match(i)) return;
           var size = sizeRx.matched(1);
           icons.push(new lime.project.Icon(iconDir + i, Std.parseInt(size)));
       });
   return icons;
```

### Enforcing compiler version

```
private function checkCompilerVersion() {
    if (environment["haxe_ver"].indexOf("3.4.7") == -1) {
        LogHelper.error("Incorrect haxe compiler version, expected 3.4.7");
    }
}
```

## Importing a project.xml

```
import lime.project.*;
class Project extends HXProject {
  public function new () {
       super ();
      var xmlConf = new ProjectXMLParser("project.xml");
      merge(xmlConf);
       doComplexStuff();
```

## Reducing boilerplate in Vegas World

- Games (e.g. roulette, lottery) are isolated modules
- Accessed via singletons
- Directory structure: Modules/<kind>/<name>/src/Swf<name>.hx

### Example module getter

```
Vegas World
class Kcomponent {
     public static var Lottery(get, never):Kcomponent;
     private static var kcomponentLottery:Kcomponent;
     private static function get Lottery():Kcomponent
                                                          https://github.com/elsassph/haxe-modular
         if (kcomponentLottery == null) {
             var functionLoad = function() { return Bundle.load(SwfLottery); };
             kcomponentLottery = new KcomponentDynamic(functionLoad);
         return kcomponentLottery;
     // Repeat for 92 other modules
```

#### Module getter template

```
class Kcomponent {
     ::foreach gameModules::
     public static var ::name::(get, never):Kcomponent;
     private static var kcomponent::name:::Kcomponent;
     private static function get ::name::():Kcomponent {
        if (kcomponent::name:: == null) {
          var functionLoad=function() {
                return ::if enabled::Bundle.load(Swf::name::)::else::null::end::;
           };
          Kcomponent::name:: = new KcomponentDynamic(functionLoad);
         return kcomponent::name::;
     } ::end::
```

#### Game module source discovery in project.hxp

Directory structure: Modules/<...>/<name>/src/Swf<name>.hx

```
// project.hxp
private function includeSources() {
   var modules = GameModule.allModules();
   modules.iter(function (m:GameModule) {
       LogHelper.info("", "Adding sources for module " + m.name + " at " + m.source);
       sources.push(m.source);
   });
}
```

## Copying the template

```
// project.hxp
private function genModuleGetters() {
   var text = File.getContent(MODULES_TEMPLATE_PATH);
   var template = new haxe.Template(text);
   var output = template.execute(modules);
   File.saveContent(MODULES_DESTINATION_PATH, output);
}
```

#### Generated module getter

```
class Kcomponent {
     public static var Lottery(get, never):Kcomponent;
     private static var kcomponentLottery:Kcomponent;
     private static function get Lottery():Kcomponent
         if (kcomponentLottery == null) {
             var functionLoad=function() { return Bundle.load(SwfLottery); };
             kcomponentLottery=new KcomponentDynamic(functionLoad);
         return kcomponentLottery;
     // Repeat for 92 other modules
```

#### Module getter template

```
class Kcomponent {
     ::foreach gameModules::
     public static var ::name::(get, never):Kcomponent;
     private static var kcomponent::name:::Kcomponent;
     private static function get ::name::():Kcomponent {
         if (kcomponent::name:: == null) {
           var functionLoad=function() {
                return ::if enabled::Bundle.load(Swf::name::)::else::null::end::;
           };
           Kcomponent::name:: = new KcomponentDynamic(functionLoad);
         return kcomponent::name::;
     } ::end::
```

### Filtering enabled modules

```
// project.hxp
private function includeSources() {
   var modules = GameModule.allModules();
   modules.iter(function (m:GameModule) {
       m.enabled = defines.exists(m.name);
       if (m.enabled) {
           LogHelper.info("", "Adding sources for module " + m.name + " at " + m.source);
           sources.push (m.source);
   });
```

#### Disabled module getter

```
Vegas World
class Kcomponent {
   public static var Roulette(get, never):Kcomponent;
   private static var kcomponentRoulette:Kcomponent;
   private static function get Roulette():Kcomponent
       if (kcomponentRoulette == null)
                                             Bundle.load(SwfRoulette)
          var functionLoad = function() { return null; };
           kcomponentLottery = new KcomponentDynamic(functionLoad);
       return kcomponentRoulette;
```

### Why? Reduced build times!

```
$ openfl build html5 -clean -Dlottery
$ openfl build html5 -clean
       1m1.971s
                                            0m16.887s
real
                                     real
       0m49.769s
                                            0m13.611s
user
                                    user
       0m10.820s
                                            0m3.095s
sys
                                     sys
```

#### Using other source files and haxelibs in project.hxp

```
@:compiler("-cp")
@:compiler("Build/src")
import build.GameModule;
@:compiler("-lib")
@:compiler("haxe-strings")
class Project extends HXProject {...}
```

#### Code reuse for module dependency checking

```
static function main() {
   GameModule.allModules().iter( function (m:GameModule) {
      var args = ["build", "html5", "-D" + m.name];
      var result = Sys.command("openf1", args);
      if (result != 0 ) {
          LogHelper.info("### " + m.name + " failed cross-module dependency check! ###");
           failedModules.push(m);
  });
  if (failedModules.length > 0 ) {
      LogHelper.info("### The following modules failed the cross-module dependency check:");
      failedModules.iter( function (m:GameModule) LogHelper.info(m.name) );
      Sys.exit(-1);
```

#### Caveats of project.hxp

Project.hxp code runs every time the project is evaluated

```
openfl [build, update, test, run, display]
```

Mute log output for openfl display command

```
// 'display' output is used by various tools, logging may break them
if (command == "display") {
    LogHelper.mute = true;
}
```

#### Resources

Example project @ https://github.com/atom-b/hxsummit2018

Project.hxp docs - http://www.openfl.org/lime/docs/project-files/hxp-format

Haxe modular - <a href="https://qithub.com/elsassph/haxe-modular">https://qithub.com/elsassph/haxe-modular</a>