

Deploying ForceSPIRE

Introduction

This document describes how to compile and bundle ForceSPIRE for release on Mac OSX and Windows Platforms. The steps in this document go over how to update the stable branch, compile it and then bundle the jar into something nice for various operating systems.

Select and Update Stable Branch

Since the Mercurial repository has two branches, *stable* and *default* (development branch) you need to first make sure you're in the stable branch to build a release.

Switching branches

The following command switches to the stable branch:

```
hg update -C stable
```

Note: make sure you commit your work before you switch branch or you might lose it. You can check what branch you're in by doing

```
hg branch
```

And see the branches available with

```
hg branches
```

Updating a branch

Next if you've been working in the default aka dev branch you might wanna bring your cool features into the stable branch if they're ready and almost bug free. To do that you have to merge.

However to merge between two branches you need to specify which revision you want to merge. Here's an easy way to find out:

```
hg heads
```

This will give you output looking somewhat like this:

```
changeset: 366:249f623c85d0
tag:       tip
user:      Patrick Fiaux <pfiaux@vt.edu>
date:      Wed Nov 16 23:20:54 2011 -0500
summary:   fixed bugs in document softdata printing

changeset: 342:23be93a2f3d0
branch:    stable
```

```
user:      Patrick Fiaux <pfiaux@vt.edu>
date:      Wed Nov 09 09:48:29 2011 -0500
summary:   Added tag v0.3.1 for changeset 50a949ce099d
```

I've highlighted the important parts in yellow. Namely you want to identify the head of the dev branch. In this case the tip (latest commit) so from the print out we know that:

- The dev branch (default doesn't display) is on revision 366
- The sable branch is on revision 342

So while we're in the stable branch we want to merge it with 366 to get all the latest goodies. Here's how you do it:

```
hg merge -r 366
```

Of course now you want to check what was merged see if you need to change anything back or resolve conflicts. (like setting a different title in forcespireApp.java BETA instead of DEV)

If you didn't want the latest changes because say the last 2 or 3 commits are experimental or crash the app you can do

```
hg log -l 10
```

To get the last 10 commits made and looking at the summary you might decide that maybe 362 is the one you want to merge.

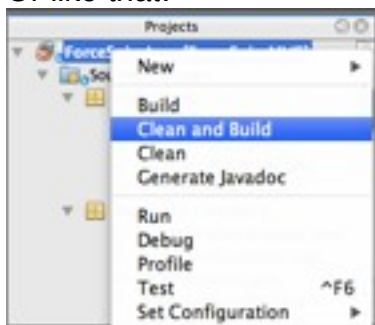
So by now you should be all up to date and ready to compile a distribution.

Building a jar

There's this handy feature in netbeans called "Clean and Build" you might find it like this:



Or like that:



Use it. Then open up your project folder and go to the "dist" (stands for distribution) folder:



That jar and that lib folder is all you need. You can double click it and run the app or use something like this:

```
java -Xms1024m -Xmx1024m -jar ForceSpireJava.jar
```

Or like this:

```
javaw -Xms1024m -Xmx1024m -jar ForceSpireJava.jar
```

If you're on windows and you don't want the console with it. The xms and xmx just increases the heap size allowed so that it can eat up more ram and load bigger files.

Bundle the jar on Mac OSX

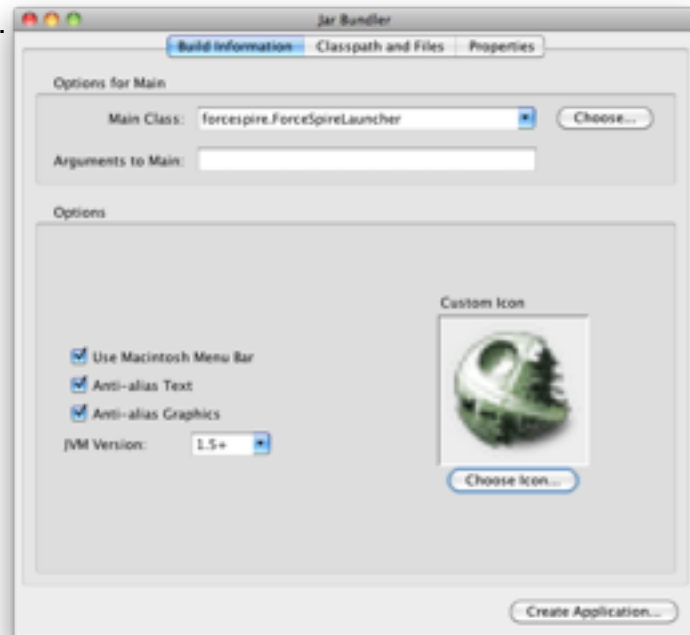
Let's face it having one file and a folder with 2 files in it isn't so user friendly because you know some one will forget to move the lib folder and then complain when it crashes because it can't find the libraries to parse entities...

Using the Jar Bundler

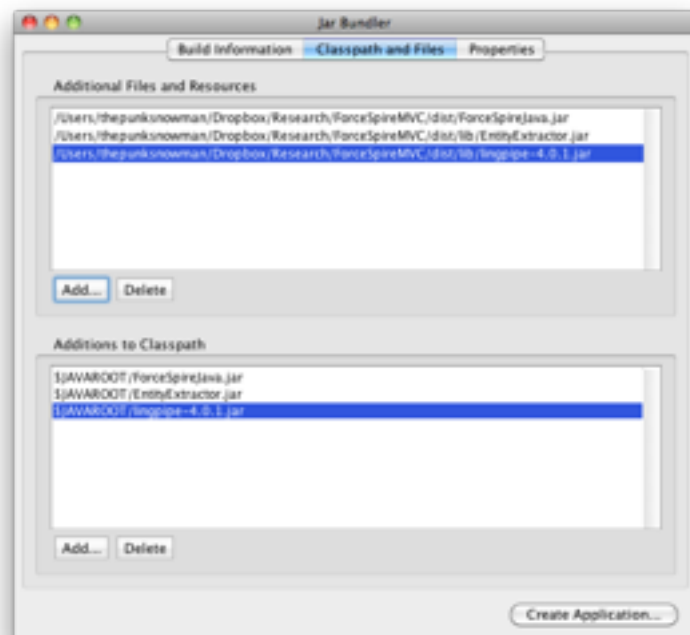
You have to be on a mac to do this.

1. Start up the Jar Bundler (use spotlight to find it).
2. Build Information. Select the jar you just made as the main class.

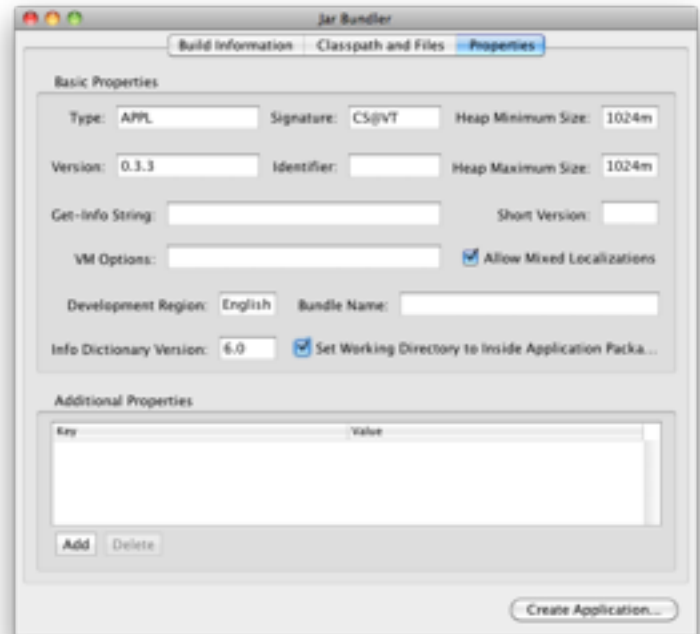
The icon is in the graphics folder on the root of the repository/project



3. Classpath and Files. Manually add the 2 jars in the lib folder:
It should look like this:



4. Properties. Set the heap sizes to 1024m
Set the rest of the settings as you please.



5. Hit Create Application. Pick a location and name, “ForceSPIRE” is a good bet.

Creating a Disk Image

If you give people a dmg file and all they do is use the mouse (no console or anything) they feel like you’re the real deal so here’s how it works:

1. Move the App Bundle/Package you just made into a new folder names “ForceSPIRE”
2. Start the Disk Utility app
3. Go to File > New > Disk Image From Folder
4. Select the “ForceSPIRE” folder you just made
5. Hit go wit the normal settings and it should pump out a .dmg file for you. (and the compression should cut the size down by like half)

The cake is a lie.

Bundle the jar on Windows

Now the hard part. We all know Macs are for idiots and their dumbed down interface clearly just doesn't make people smarter but us windows user are of superior intellect so it's OK for us to have 3 files and 2 of which are in a folder right?

Yes, but this command line part mentioned above... not so sure about that. Command line is for geeks, nerds and system administrators right? All those unix people. We don't use command lines on windows so we'll just have to bundle the jar anyways.

Launch4j

To create a windows executable Launch4j offers a GUI and the ability to save a config for reuse.

<http://launch4j.sourceforge.net/>

Setup and bundling

Launch4j has config files but they're not exactly multi setup friendly and don't really use relative paths so unless you have the same exact path on your machine as the one who set up the config you can't reuse it.

Therefore I'll just list the fields that need to be filled out:

1. Basic Tab

Output Path:

```
dist/ForceSPIRE.exe
```

This works well because it keeps it within range of the libs needed to run.

Jar:

```
dist/ForceSpireJava.jar
```

Next the icon:

```
graphics/DeathStar.ico
```

2. JRE

Min JRE version:

```
1.6.0_26
```

Initial heap size:

```
1024
```

Max heap size:

```
1024
```

3. Compile

The rest of the config isn't needed. Just hit the gear button and it should generate an exe!

Final Step

At the time this document was written there's just no easy way to bundle the library jars inside the exe (without using yet another tool like One-Jar, which just makes it too complicated).

So here's what the dist folder should look like:

```
lib/  
ForceSPIRE.exe  
ForceSpireJava.jar  
README.txt
```

Select the lib folder and the ForceSPIRE executable and compress them into a zip using your favorite compression utility.