

Jar のマニフェストをいじる

マニフェスト中の、特定のキーの値が欲しいとき

```
Manifest manifest = jf.getManifest();
Attributes ma = manifest.getMainAttributes();
String classpath = (String)(ma.getValue("Class-Path"));
```

COINS を実行可能 JAR から利用する

実行可能な Jar ファイルでは、manifest の Class-Path で必要なライブラリへのパスを通すことができる。

したがって、COINS を自分でつくったアプリケーションから使う場合、COINS をビルドして jar にしたファイル

(COINS で配布されている jar ではなくて classes の下だけを jar にしたもの)

に対して、manifest で class-path を指定すればよいはずである。

が、Unix 系 (Linux、FreeBSD、MacOSX) ではうまく動かなかった。

なぜか WindowsXP ではうまく動いた。

これは、manifest 中の Class-Path が Java のプロパティである java.class.path に含まれないために、COINS の中でクラスを動的にロードしようとしても当該クラスをみつけることができないことに起因していた。

というわけで、これを回避するために、ちょっとハックしてみた。

COINS のディレクトリの中で、

```
patch -p0 < coins-1.4.2.2-ja-searchClass_resursive.patch
```

とすれば適用できる。

COINS を実行可能 JAR から利用するための作業工程

たとえば、

```
wallet.jar
lib/coins.jar
```

というようなディレクトリ構成にしている

wallet.jar は実行可能 jar ファイルで、manifest ファイルで次のように

```
Manifest-Version: 1.0
Main-Class: net.wasamon.wallet.WalletGUI
Class-Path: lib/coins.jar
```

Class-Path に lib/coins.jar をちゃんと指定していても

```
java -jar wallet.jar
```

として Linux や FreeBSD、MacOSX で起動しようとするとき、次のような例外が発生して実行できない。

ちなみに Windows でダブルクリックするとなぜか実行できる。

```
Exception in thread "AWT-EventQueue-0" java.lang.NoClassDefFoundError: coins.driver.SuffixFactory
at coins.driver.CompilerDriver.initializeSuffixFactory(CompilerDriver.java:81)
at coins.driver.CompilerDriver.<init>(CompilerDriver.java:184)
at net.wasamon.wallet.WalletDriver.doCompile(WalletDriver.java:156)
at net.wasamon.wallet.WalletGUI.actionPerformed(WalletGUI.java:204)
```

ちなみに、

```
java -cp wallet.jar:lib/coins.jar net.wasamon.wallet.WalletGUI
```

とした場合には、ちゃんと実行できる。

もちろん、coins.driver.SuffixFactory は coins.jar に含まれている。

で、してみると、coins.driver.CompilerDriver.initializeSuffixFactory で
getAttachedFileInputStream(Class String) という関数を呼び出していて、
このメソッドが原因となる例外が発生している。

このメソッドでは、

System.getProperty("java.class.path") で得られる

パスから、該当するクラスを検索するために

ディレクトリあるいは、jar ファイルであっても求めるクラスを探す

searchClass メソッドを呼び出し、この戻り値が全て偽の場合に、例外が投げられる。

ここで問題となるのは、

```
java -jar wallet.jar
```

として起動した場合には、System.getProperty("java.class.path") で得られるのが
wallet.jar だけであるということ。

たしかに、これじゃあ、coins.jar の方は検索していないわけだから見つかるはずもない。

なので manifest で指定した Class-Path ファイルについても検索してもらう必要がある。

たとえば、こんな感じ。

```
Manifest manifest = jf.getManifest();
Attributes ma = manifest.getMainAttributes();
String classpath = (String)(ma.getValue("Class-Path"));
```

Attributes には、get(Object) というメソッドもあるが、getValue(String) でないとダメ。

で後は、ここで得られた classpath を空白で分割して、

それぞれについて再帰的に searchClass を検索すればいい。

幅優先で探索するべきなのかもしれないけど、まあ、とりあえず深さ優先で。

ちなみに、オリジナルの searchClass は、真偽値を返すメソッドだが、

これでは、新しく manifest 中の Class-Path から得たパスを返すことができないので、

真の代わりにパスを文字列で返し、偽の場合は null を返すようにした。