

Executing a Java Class with ProcessPanel

1. Make a Java class and package it into a JAR:

IzPack can execute Java classes provided by third party developers.

This can be useful to extend IzPack's functionality in a number of ways

Some examples:

- to control the writing of configuration files during installation or
- creating a custom log.

IzPack can display output from these classes via a ProcessPanel, or you could create a custom (possibly hidden) panel that calls your Java code.

This following example shows how to use a custom Java class with a ProcessPanel.

An external Java class must have a run() method with some parameters.

Note: This is *not* (NOT!) the run() method from the java.lang.Runnable interface. Nor is this codified as an interface in the IzPack javadoc. **Instead, this would seem to be a magic method signature that is otherwise undocumented.**

a. Contents of a minimal file that matches the requirements:

a. Contents of a minimal file that matches the requirements:

```
package org.callimachusproject;
import com.izforge.izpack.util.*;
public class HelloWorld {
    public void run(AbstractUIProcessHandler
handler, String[] args) {
        handler.logOutput("Hello, World!",
false);
    }
}
```

This class uses the *com.izforge.izpack.util.AbstractUIProcessHandler* class to log output into a calling ProcessPanel. See the IzPack javadoc for details.

b. Compile your Java class

You will obviously need to include the standalone-compiler.jar on your classpath, in order to compile your class:

Compile your Java class

```
$ javac -cp
"/path/to/izpack-standalone-compiler.jar:."
org/callimachusproject/HelloWorld.java
```

c. Package your Java class into a separate JAR

IzPack can only reference JAR files not individual class files from a ProcessPanel.

Package your Java class into a separate JAR

```
$ jar cvf hello.jar
org/callimachusproject/HelloWorld.class
added manifest
adding:
org/callimachusproject/HelloWorld.class(in =
470) (out= 316)(deflated 32%)
$ cp hello.jar /path/to/installer/
```

NB: You should eventually use Maven to automate the compilation, packaging and moving of your JAR file.

2. Set up the install.xml file

a. Add <resources> entry.

In the <resources> section of your install.xml, reference an external file called "*ProcessPanel.Spec.xml*".

The *ProcessPanel.Spec.xml* file holds the XML configuration for the external Java class you want to execute.

install.xml file

```
<installation>
  ...

  <resources>
    <res id="ProcessPanel.Spec.xml"
src="installer/ProcessPanel.Spec.xml" />
    ...
  </resources>

  ...
</installation>
```

b. Reference your JAR

In the install.xml file's top level, reference the JAR file containing your class.

This is partially documented at:

<http://izpack.org/documentation/installation-files.html#the-jar-merging-element-jar>

NB: The path to the JAR is the path at *compile time*.

install.xml file

```
<installation>

  <resources>
    <res id="ProcessPanel.Spec.xml"
src="installer/ProcessPanel.Spec.xml"/>
  </resources>

  ...

  <jar src="path/to/hello.jar"
stage="install"/>

  ....

</installation>
```

A good place to put this is near the <resources> definition.

c. Use a ProcessPanel to execute your Java class

In the install.xml file in the panels section, use a ProcessPanel to execute your Java class.

install.xml file

```
<installation>

  <resources>
    <res id="ProcessPanel.Spec.xml"
src="installer/ProcessPanel.Spec.xml"/>
  </resources>

  ...

  <jar src="path/to/hello.jar"
stage="install"/>

  ....

  ...
  <panels>
    ...
    <panel classname="ProcessPanel"/>
  </panels>

  ....
</installation>
```

3. Create the ProcessPanel.Spec.xml file

This is partially documented in <http://izpack.org/documentation/panels.html#processpanel> **DOES THIS NEED TO BE ADDED TO THE VERSION 5 DOCS?**

ProcessPanel.Spec.xml

```
<processing>
  <logfiledir>$INSTALL_PATH</logfiledir>
  <job name="setup">
    <executeclass
name="org.callimachusproject.HelloWorld"></e
xecuteclass>
  </job>
  <onFail previous="true" next="false" />
  <onSuccess previous="false" next="true" />
</processing>
```

IzPack javadoc

The IzPack javadoc is available in the distribution (although apparently not online). There is an RPM for Linux users.
