

Kepler - Bug #4152

Modules should be able to add environment variables to the runtime

06/12/2009 01:09 PM - Chad Berkley

Status:	Resolved	Start date:	06/12/2009
Priority:	Immediate	Due date:	
Assignee:	Chad Berkley	% Done:	0%
Category:	build system	Estimated time:	0.00 hour
Target version:	2.0.0	Spent time:	0.00 hour
Bugzilla-Id:	4152		
Description			
Modules such as R sometimes need to add environment variables to the runtime. R needs the R_HOME variable set in order for R to be found by the JRI system. Need to add functionality to the build to allow modules to add environment variables.			

History

#1 - 06/12/2009 01:16 PM - David Welker

Just so I understand this correctly, this is a distinct use case from setting System properties, correct? So, if you had the location stored in a system property, that would not be adequate?

#2 - 06/12/2009 01:48 PM - Chad Berkley

Right. It is distinct because it actually needs to load an environment variable.

#3 - 06/12/2009 01:50 PM - Christopher Brooks

In Java 1.5 and later the `java.lang.System.getenv()` method returns environment variables.

The Javadoc for `getenv()` says:

"System properties and environment variables are both conceptually mappings between names and values. Both mechanisms can be used to pass user-defined information to a Java process. Environment variables have a more global effect, because they are visible to all descendants of the process which defines them, not just the immediate Java subprocess. They can have subtly different semantics, such as case insensitivity, on different operating systems. For these reasons, environment variables are more likely to have unintended side effects. It is best to use system properties where possible. Environment variables should be used when a global effect is desired, or when an external system interface requires an environment variable (such as PATH)."

So, I think that if R is being invoked as a separate process or is running a C call to `getenv()`, then we would need to set the environment variables.

The Java Process class can pass environment variables to subprocesses.

I don't think there is a Java `setEnv()` command. One way to make this work would be to define a Parameter that is a Record that could be referred to by the External Execution actor (which uses `ptolemy.actor.lib.Exec`) in the environment actor parameter. This would allow subprocesses to access the updated environment variables. I don't know if a JNI call to a C library that calls the C `getenv()` function would see any change in the environment.

#4 - 06/12/2009 01:53 PM - Chad Berkley

You can, in fact, add env vars to the runtime. We do it for the PATH variable (for JNI on windows). I'm implementing this functionality now.

#5 - 06/12/2009 01:59 PM - Chad Berkley

Added the ability to the build system to parse a `$module/module-info/environment.txt` file. This file lists the environment variables that should be loaded by the java runtime at startup. Variables can be loaded either independently of the OS or they can depend on a specific OS.

The syntax is as follows:

```
<env_name>:[<env_os>]=<value>
```

An example:

R_HOME:Mac OS X=/Library/Frameworks/R.framework/Resources

or

MY_PATH=/x/y/z

#6 - 03/27/2013 02:25 PM - Redmine Admin

Original Bugzilla ID was 4152