

Re: Compile java sources inside J2EE component

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2004-09/0642.html>

From: Chris Uppal (chris.uppal_at_metagnostic.REMOVE-THIS.org)

Date: 09/07/04

Date: Tue, 7 Sep 2004 10:50:29 +0100

Stefan Siegl wrote:

> *So much for the introduction ;). After creating the source files I need
> to compile them at runtime, is there a possibility to do so?*

Yes. You can invoke Sun's compiler directly (I can't remember the precise class off-hand, but it's mentioned in another thread in this newsgroup today). Andrew's warning is to be taken seriously, but I wouldn't let it worry /me/ — I believe that Sun are intending to provide a public API to compilation soon (and that it only missed 1.5 because of time pressure), so I'd hope that if Sun do remove the private classes, then they'll replace them with something else you can use. You can always fall back to using javac if it really becomes necessary, anyway.

However, if /I/ were doing this then I think I'd look at the option of interpreting the meta-model directly, rather than generating new classes. I think that would be easier, and I can't imagine it causing performance problems. Still, there's nothing wrong with generating classes, if that's what appeals to you, so...

> *As I said before if the model changes, the new source code files need to
> be compiled again. Is there a way to "unload" possible classes that were
> created for the last model, so that there won't be a clash with the
> newly created classes?*

You need to load all the "disposable" classes through a specially created classloader (the standard URLClassLoader, or whatever it's called, should work OK, but you might want to write your own anyway). When you drop the reference to the classloader, and when there are no more instances of the classes themselves alive, then the classes will be removed from the runtime. When you want to create a new batch of classes, you create a new classloader to load them, so there's no interference with any defunct classes that might still be hanging around.

You'll probably find it worthwhile to define a bunch of interfaces that are /not/ disposable, and which the various generated classes implement. Otherwise you'll have difficulty "talking about" instances of the disposable classes in other contexts.

comp.lang.java.programmer: Re: Compile java sources inside J2EE component

One thing that isn't helped by classloaders is migrating the state of the system as it is modified. If you create a new "version" of a class, the instances of the existing class won't be migrated. You'll have to transfer the state of the system from the old collection of objects to the new. Somehow...

Again, I think that migration would probably be easier if you were interpreting the meta-model, and that's one reason why I'd consider that first.

-- chris