

Kepler - Bug #5576

memory leak in ptolemy.data.expr.CachedMethod

02/17/2012 03:19 PM - Daniel Crawl

Status:	Resolved	Start date:	02/17/2012
Priority:	Normal	Due date:	
Assignee:	Daniel Crawl	% Done:	0%
Category:	general	Estimated time:	0.00 hour
Target version:	2.4.0	Spent time:	0.00 hour
Bugzilla-Id:	5576		
Description			
<p>There is a memory leak in CachedMethod where the same method is inserted multiple times into the static hash table <code>_cachedMethods</code>. You can see this by running the attached model created by Jianwu. The model has an Expression actor, which uses <code>trim()</code>, <code>isEmpty()</code>, <code>split()</code>, and <code>map()</code>. The model iterates 1000 times, and afterwards jmap reports there are 1004 CachedMethods in the heap:</p> <pre>jmap -histo:live 6959 grep CachedMethod 38: 1004 40160 ptolemy.data.expr.CachedMethod</pre> <p>Most of these CachedMethods are for <code>map()</code>. The problem is that each CachedMethod for <code>map()</code> has a different hash code. The hash code for a CachedMethod is computed by using the method's argument types and <code>map()</code>'s first argument is a <code>FunctionType</code>. The hash code for <code>FunctionType</code> is based on the hash code for <code>FieldTypeTerm</code>, which is unique for each instance since it does not override <code>Object.hashCode()</code>.</p>			

History

#2 - 06/15/2012 07:22 PM - Christopher Brooks

This is possibly fixed in r63763 in the pTII tree.
Basically, I had to add better `equals()` and `hashCode()` methods to `FunctionType` and `FunctionType.FieldTypeTerm`.

BTW - getting `equals()` and `hashCode()` right is tricky.
I found a good document at
See <http://www.technofundo.com/tech/java/equalhash.html>

- To test the bug
- 1) Download the attachment above:
<http://bugzilla.ecoinformatics.org/attachment.cgi?id=397>
 - 2) Start up Kepler, open the model, run it
 - 3) Start up `jvisualvm`
 - 4) Connect to the Kepler process, Monitor -> Perform GC, generate a heapdump
 - 5) In the heapdump window, click on classes and then search for Cached

Formerly, there were about 1000 entries, now there are many fewer.

Currently, `CachedMethod` has two entries for `map`. It looks like they are different because of the number of array elements.

I temporarily added a static `dump()` method to `CachedMethod` that merely called `System.out.println(_cachedMethods)`
I then temporarily added code to `Manager` that called `CachedMethod.dump()`

The output is below.

```
{map((function(a0:string) {key = string, value = int}),
arrayType(string,5))= map((function(a0:string) {key = string, value =int}),
arrayType(string,5)),
map((function(a0:string) {key = string, value = int}),
arrayType(string))= map((function(a0:string) {key = string, value = int}),
arrayType(string)),
string.split(string)=string.split(string),
string.isEmpty()=string.isEmpty(),
```

```
string.trim()=string.trim()})
```

I don't think there is much we can do about this, but I'm open to suggestions.

I think this is ready to be closed, I'm dispatching it back to Daniel for his approval.

#3 - 06/25/2012 11:20 AM - Daniel Crawl

I ran the attached model and this looks fixed to me. Previously, each execution added about 1000 instances of `ptolemy.data.expr.CachedMethod`. Currently there are only 2 instances regardless of how many times the model executes. Thanks, Christopher!

#4 - 03/27/2013 02:30 PM - Redmine Admin

Original Bugzilla ID was 5576

Files

cachedmethod-leak.xml	24.6 KB	02/17/2012	Daniel Crawl
-----------------------	---------	------------	--------------