

## Kepler - Bug #3962

### RExpression2 - support RecordTokens with length=1

04/07/2009 02:38 PM - ben leinfelder

<b>Status:</b>	Resolved	<b>Start date:</b>	04/07/2009
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	ben leinfelder	<b>% Done:</b>	0%
<b>Category:</b>	actors	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.0.0	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	3962		
<b>Description</b>			
Ptolemy expression: {a = 1, b = "b", c = 3} results in a conversion error when going from Token->Java/R			

#### History

##### #1 - 04/07/2009 02:52 PM - Derik Barseghian

Hmm, this looks familiar. I think I fixed this, or something similar in the original RExpression actor. Not sure if that's in anyway helpful for RExpression2, but if so, grep your mail for "No longer ignore recordtokens with length 1 in conversion to dataframe.", a cvs commit on April 25, 2008 to see the relevant section.

##### #2 - 04/13/2009 10:44 AM - ben leinfelder

I can see in the RExpression2 code that I'm assuming RecordTokens contain ArrayTokens. But when there's only one Token, there's a class cast exception (duh).

##### #3 - 04/13/2009 03:13 PM - ben leinfelder

RExpression2 actor now accepts the length=1 RecordToken, but slightly mutates the structure during the conversion.

Ptolemy -> JRI -> Ptolemy results in the following Token: {a = {1}, b = "b", c = {3}}

Notice that the the numeric tokens (a and c) are now lists, whereas the string token is a single token.

JRI *only* assigns numeric values using arrays of them. Even treating the String as a String[] doesn't force "b" to be a list - this seems like a JRI limitation.

I think we can live with what's returned. The alternative is to treat the numbers as strings so that it's a uniform data structure.

Subsequent passes through multiple RExpression2 actors preserves the data structure as such: {a = {1}, b = "b", c = {3}}  
so at least it's predictable from that standpoint.

##### #4 - 03/27/2013 02:25 PM - Redmine Admin

Original Bugzilla ID was 3962