

Kepler - Bug #1845

user input for port type should not be case sensitive

12/17/2004 02:24 PM - Laura Downey

Status:	Resolved	Start date:	12/17/2004
Priority:	Normal	Due date:	
Assignee:	Matthew Brooke	% Done:	0%
Category:	interface	Estimated time:	0.00 hour
Target version:	1.0.0alpha9	Spent time:	0.00 hour
Bugzilla-Id:	1845		
Description			
"string" is accepted as a port input type but "STRING" is not			

History

#1 - 12/17/2004 03:47 PM - Christopher Brooks

Ideally, we would have a combo box that would have the common types as defaults and allow the user to type something in if need be

#2 - 12/21/2004 02:04 PM - Laura Downey

Precisely, combo boxes are great little widgets that provide users a set of acceptable choices while allowing for custom input. The input however, should still be case insensitive.

#3 - 12/18/2005 11:43 PM - Christopher Brooks

I went ahead and fixed this in `ptolemy/actor/gui/PortConfigurerDialog.java` by adding a `CellFixer` class that has a `String fixCell(String value)` method. If type cell is invalid, then we call `fixCell()` with the invalid value and hopefully get back a fixed value. If a fixed value is found, we set the combobox value.

Right now the case insensitivity only works for fundamental types not arrays, matrices or records. For example "Double" gets changed to "double", but "[Double]" does not get changed to "[double]". To have this work in arrays, matrices and records would require making the parser case insensitive. I'm not so sure that changing the parser is right, it could cause backward compatibility issues.

I'm not totally sure that having the type in the combobox be case insensitive is the correct behaviour. For example, one of the lamest parts of Windows is that the so-called file system is case insensitive, yet case preserving. This causes no end of trouble. My concern here is that there is a slight chance we could be preventing the use of user defined types because "STRING" and "string" could mean two different types. Anyway, I think this a very remote possibility, and I myself have had a hard time remembering the capitalization.

Since we've had a combobox here, I've had a much easier time since the choice is there. Now that the editable portion of the combobox is case insensitive and auto correcting, this should be even easier to handle.

#4 - 12/19/2005 12:00 AM - Christopher Brooks

My bad, I wanted to leave this open pending discussion about non-fundamental types: Should `[Double]` automatically convert to `[double]`? If yes, then what about records? Should `{x=Double, y=double}` convert to `{x=double, y=double}`. Note we can have arbitrarily complex records, so this would require changing the parser.

#5 - 12/19/2005 08:06 AM - Laura Downey

The issue here involves a couple of different things.

1. when first tested, the user had to exactly type something in for the system to recognize it. That is way too much burden on the user. I very much doubt there are many users that really want to assign string and STRING to be two different types! :-)

2. the second issue is that a set of common data types should be offered to the user so they don't have to type at all if they are choosing a standard type.

3. it should really be a combo box so a user can indeed add/specify a type that is not part of the standard set (I can't remember where but I wrote a bug that talked about using combo boxes where possible in whatever situation we had that there was generally a standard set of choices but also allows for customization by the user. And if a user created a new type for example called String, then for that user String should subsequently show up as a choice. We could also consider allowing users to "globally" add new types so they are available for other users.)

#6 - 12/19/2005 10:47 AM - Christopher Brooks

Shawn wrote:

What about Boolean ?

Boolean is sort of a proper name (<http://en.wikipedia.org/wiki/Boole>) and it should be capitalized. However, the Java primitive type is "boolean" and the corresponding wrapper class is "Boolean". In Ptolemy, the type is "boolean" so as to match the other types. This could be a mistake, and perhaps PtParser should be modified, but I'm probably not going to do it.

The way the combobox gets its types is by calling `data.expr.Constant.types()` which returns the `TreeMap` of types. Thus, all the basic types in the lattice should be present in the combobox.

Laura writes:

1. when first tested, the user had to exactly type something in for the system to recognize it. That is way too much burden on the user. I very much doubt there are many users that really want to assign string and STRING to be two different types! :-)

Agreed. However, there can be problems with having multiple representations of the same thing. I think what we have now will work for the fundamental types. The question is, should the parser be modified to be case insensitive about nonfundamental types like arrays, matrices and records? If I type STRING into the combo box, then the it will change to string. However, if I type [STRING] it will not change and I'll get an error. To change this requires modifying the parser either by introducing aliases (String, string, STRING) or trying to make the comparison case insensitive.

2. the second issue is that a set of common data types should be offered to the user so they don't have to type at all if they are choosing a standard type.

We've had combo box in the Port Configuration dialog for some time. I think I updated another bug about it, but failed to update this one. This change is present in alpha8, so give it a try and let me know what you think. I realize that the "Configure Ports" dialog is not at all what the Kepler group wants, but I'm trying to push bugs along that require Ptolemy backend work so that we move closer to the final set of Kepler dialogs.

3. it should really be a combo box so a user can indeed add/specify a type that is not part of the standard set (I can't remember where but I wrote a bug that talked about using combo boxes where possible in whatever situation we had that there was generally a standard set of choices but also allows for customization by the user. And if a user created a new type for example called String, then for that user String should subsequently show up as a choice. We could also consider allowing users to "globally" add new types so they are available for other users.)

Right now, we rely in the `_type` `TreeMap` in the `Constant` class to be updated. So, if adding a type updates `_type`, then the combobox will see it as a choice. If the user creates a type called "String", and the parser recognizes it, then it will not be automatically converted to "string". The reason is that we only try to fix things if the `isValid(String value)` method fails. Currently, `isValid()` passes value to the parser, so it is up to the parser

to do the right thing.

Adding user types is another issue and should probably be a separate bug and won't be addressed here.

Right now, to add a type requires modifying the Ptolemy code.

#7 - 12/20/2005 07:19 AM - Laura Downey

Case sensitivity issue is superceded here by having a combo box with a standard set of data types recognized by the system, so the user doesn't have to guess at the case of the type they are trying to enter. If we decide to allow a user to enter a custom data type, that input will be case sensitive as it is the user creating something from scratch, not trying to guess at how the system might want the capitalization of the input. This way, the expression language retains its case sensitivity and no conversion of user input is necessary.

#8 - 12/20/2005 11:09 AM - Christopher Brooks

For the sake of simplicity, I've rolled back my case sensitivity changes so that the type combo box in the port configuration dialog is no longer case insensitive.

I'm marking this one as fixed. If anyone feels otherwise, feel free to reopen.

#9 - 03/27/2013 02:18 PM - Redmine Admin

Original Bugzilla ID was 1845