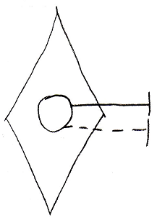
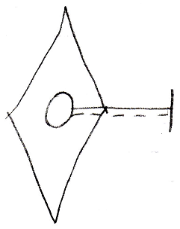
|  |
| --- |
| Circle Language Spec: Black Boxes |

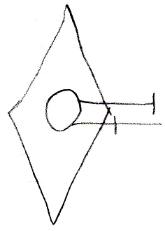
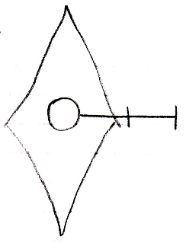
## Simplified Access Control Notation

There are only a couple of ways, in which expression of access control connectors in a diagram will be simplified.

The first method is merging the access connectors of different aspects together.

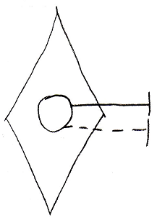
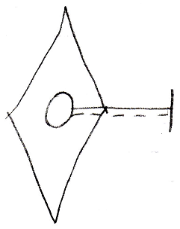
 

The second method is merging the access connectors of a Get and Set together.

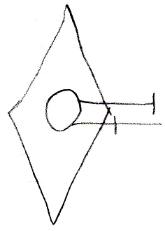
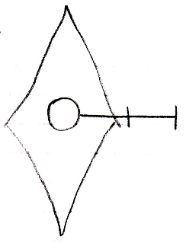
 

There will be rules about when you can apply these methods of simplification.

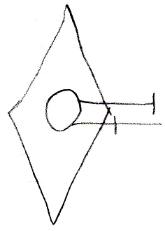
Merging access connectors of different aspects together only happens for access connectors with the same access direction and same access mark position.

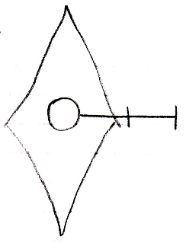
Merging Get and Set connectors happens only when they are both accessible.

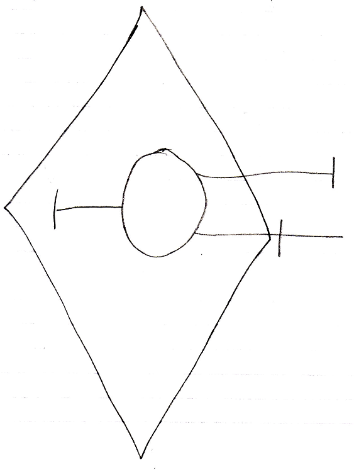
Furthermore, the merging happens in respect to the most likely chronological order of access: Set Public, Get Private, Set Private, Get Public. This means, that two access connectors will only merge if there is nothing chronologically in between them, that is accessible.



The accesss connectors above will merge, because nothing is chronologically in between.

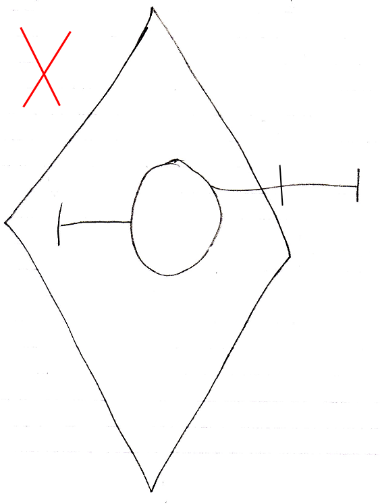


But if something were in between, for instance, an Private Set connector, then the merge would be blocked by it:



In the picture above, the Public Get and Set connectors will not be merged, because the Private Set connector is chronologically in between.

The access connectors do not need to be *positioned* in chronological order in the diagram, but merging connectors together suggests nothing can be in between them, and it is just clearer if you keep them separated when something can get chronologically in between.



Here are all the possible access connectors for parameters, now with the easier to read parameter access control literals and in a more chronological order. Actually, for now, they are not much different from the unsimplified versions, but in the future (from 2008-09-29), when the terms In, Out and Thru are more exactly defined, the terms may very well become a lot simpler.

|  |  |
| --- | --- |
| Object | Command |
|  |  |
|  |  |
| Object Set Public | Object Get Private |
|  |  |
|  |  |
| Object Set Private | Object Get Public |
|  |  |
|  |  |
| Class Set Public | Class Get Private |
|  |  |
|  |  |
| Class Set Private | Class Get Public |
|  |  |
|  |  |
| Value Set Public | Value Get Private |
|  |  |
|  |  |
| Value Set Private | Value Get Public |
|  |  |
|  |  | |
| Clone 2 Set Public | Clone 2 Get Private | |
|  |  | |
|  |  | |
| Clone 2 Set Private | Clone 2 Get Public | |
|  |  | |
|  |  |
| Data Set Public | Data Get Private |
|  |  |
|  |  |
| Data Set Private | Data Get Public |
|  |  |

|  |  |
| --- | --- |
| Execute Public | Execute Private |
|  |  |
|  |  |
| New In | New Out |
|  |  |
|  |  |
| Annul In | Annul Out |
|  |  |