|  |
| --- |
| Circle Language Spec: Events |

## Black Boxed Events

Details about black boxes are covered in the chapter *Black Boxes*.

However, two situations are explicitly denoted here:

- Making events Private

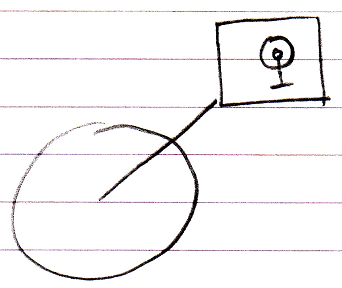
- Accessing the Privates of an event parameter

Just like any other member of an object, events can also be made Private. In that case only Friend objects can pick up this event. Below you will find the involved notations. Details are not covered. More information about black boxing can be found in the *Black Boxing* chapter.

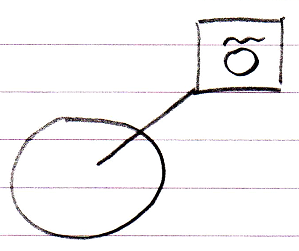
|  |  |  |
| --- | --- | --- |
| Public Event Connector | Friend Event Connector | Private Event Connector |
|  |  |  |
|  |  |  |
| Public Event Connection | Friend Event Connection |  |
|  |  | – |

There are two reasons why the Public event connector does not get an access mark: Public is sort of the default, and the event connection is outward so it does not need an access symbol. The others do need an access mark, because it is the only way to express the access modification.

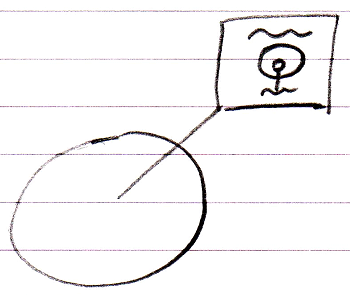
The second black boxing situation to be explicitly mentioned is accessing the Privates of an event parameter. By default you only get to access the Public members of an event parameter. This is what makes it the black box principle.



To be able to access the Private members of the event parameter, the event procedure must be declared Friend of the event parameter.



Then the event procedure can access the Private members of the event parameter:



The Friend declaration is an agreement between the event sender and the class of the event parameter. The Friend declaration is part of the event interface, defined by the event sender. So the event receiver can not just declare itself Friend of the event parameter whenever it feels like it.