|  |
| --- |
| Circle Language Construct Drafts | Relationships |

## Referrers Implementation

**[ This article may not be up to date. ]**

*Referrers* is a coding concept. A concept is something programmed inside the new computer language itself, which can extend any object, because it operates using the reflective data of an object, which makes it adaptable to any object.

The Referrers coding concept extends the system interface of any related object with an attribute Referrer ID in Object. The Object . Set will be pre-extended with removing itself from the original target object’s list of referrers, using the previously stored Referrer ID in Object attribute. The Object . Set also gets post-extended with adding itself to the new target object’s list of referrers. After that the position in this list is assigned as the Referrer ID in Object.

The Referrers coding concept also adds a Referrers list to every object. An *object* stores the list of referrers, but a *symbol* can also store its own list of referrers if it is a pointer *pointed to*. The Referrer list’s Add method will return the added item’s position in the list, so it can be recorded by the Referrer as a sub-object’s Referrer ID in Object.

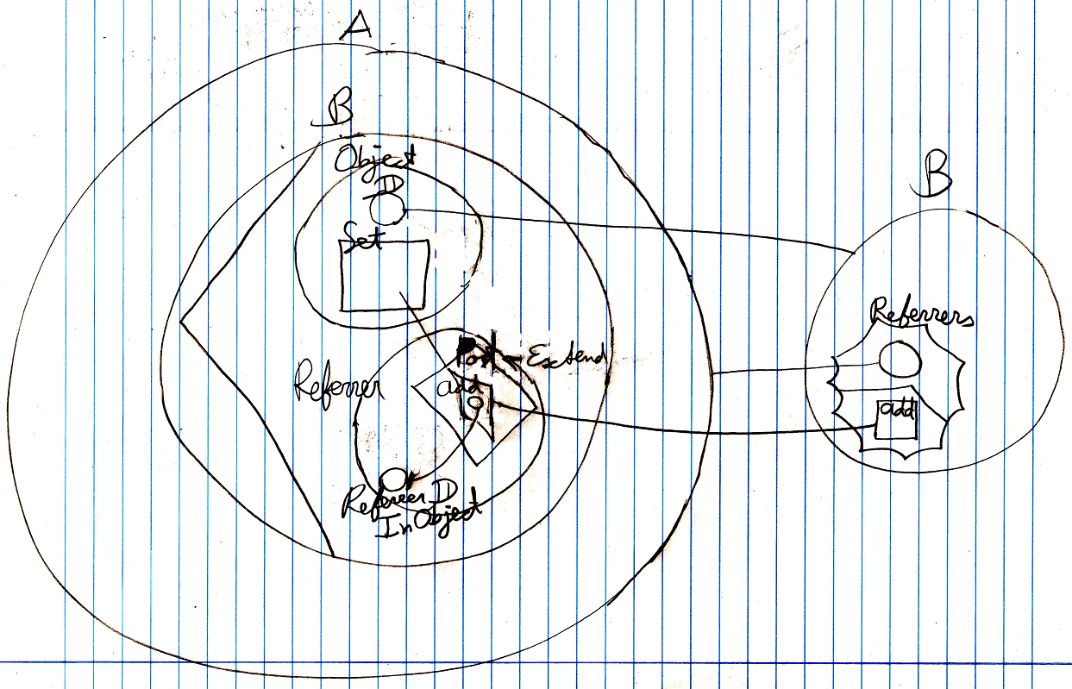
If an object does not support the concept of Referrers, then it will not have a Referrers list.

An object, that does not support the concept of Referrers, also will not register itself as a referrer in objects that it references. So that will make the object sort of stealthy, and not known to the objects that it references.

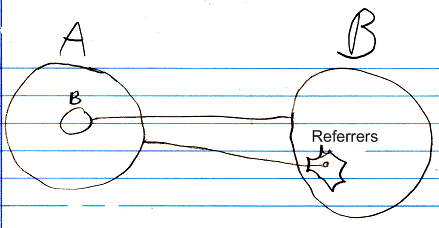
### Diagrams

As explained in the article *Referrers*, the implementation of referrers as a concept requires extension of related item’s Object . Set procedure, so that a referrer is added to the other object’s Referrers list and the Referrer ID In Object is recorded.

The adapted system interface would look something like this in a diagram:



It is just a demonstration of extension of the system interface. Not all the details are shown above. By showing the system interface you can see exactly see how the Referrers list is updated. Fortunately you can hide the system interface, so that it looks simple again:



Showing the system interface gives you a view on the inner workings of the Referrers concept. Further details won’t be discussed here. It is just an example of a concept extending of the system interface.