Question
[Laura Cavanaugh](#) · May 11, 2020

Clones, oref reference count, and scope

Hello all, I have a question about constucting thousands of clones, and scope.

In my code, I'm looping through a database, say 200k+ objects, and creating a clone of each object (we need to evaluate a modified clone of the object, but not account for what's on disk).

I see this a lot in the documentation:

" Whenever you set a variable or object property to refer to a object, its reference count is automatically incremented. When a variable stops referring to an object (if it goes out of scope, is killed, or is set to a new value), the reference count for that object is decremented. When this count goes to 0, the object is automatically destroyed (removed from memory) "

After I'm done with an object and its clone, I kill them, but over time, I keep getting a STORE error. I'm wondering if IS has a java-like garbage collector (as in "the object is automatically destroyed"), and if so, is this keeping up with the clone construction and killing, over a 200k+ loop? Is it an immediate thing, or is there a garbage collector crawling around looking for un-referenced objects?

Or, am I missing killing an object?

All objects and clones are killed after each loop iteration. The process has only a short list of variables. The reference count gets higher and higher (I assume it is, when an object's oref is [43234@ Package.Class.cls](#)), but I'm pretty sure I'm killing every variable that might reference an object.

Just curious about the speed of the garbage collection, and what might contribute to the high oref reference count.

Thanks,

Laura

[#ObjectScript](#) [#Caché](#)

Source URL:https://community.intersystems.com/post/clones-oref-reference-count-and-scope