

Artículo

[Dani Fibla](#) · Jun 11, 2020 Lectura de 4 min

Resumen sobre el alcance de las variables locales

Este es un resumen de un tema que aparece en varios lugares de la documentación en línea, sobre todo como observaciones, pero nunca como un capítulo dedicado.

Érase una vez... No, no es un cuento para niños.

En los comienzos de Caché (y antes), tenías tu partición para ejecutar tu código. Parte de esa partición era un espacio con todas tus variables locales bien ordenadas según %,A,..Z,a,...z

Además, cualquier valor o información que tuvieras que almacenar localmente estaba allí, visible y disponible para cualquier parte del código que se ejecutara en tu partición. No había problema si lo usabas con un equipo de desarrolladores que trabajaban juntos, con documentación completa y una excelente disciplina.

[Lo siento, no debía de ser un cuento para niños].

De hecho, trabajar con paquetes de software (propios o ajenos) podía convertirse en una pesadilla y encontrar un uso no conflictivo de las variables era más difícil que el código en sí. No hace falta mencionar que los nombres significativos se convirtieron en una excepción. La única ayuda en ese momento era el comando NEW para empujar variables en la pila y recuperarlas más tarde.

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOS_cnew

La solución real en Caché 5.0 (2002) fue crear un ámbito aislado para procedimientos y métodos, con un espacio vacío e independiente para las variables dedicadas a estos BLOQUES de código. Ya no hay más problemas con la protección contra el uso indebido o variables con el mismo nombre.

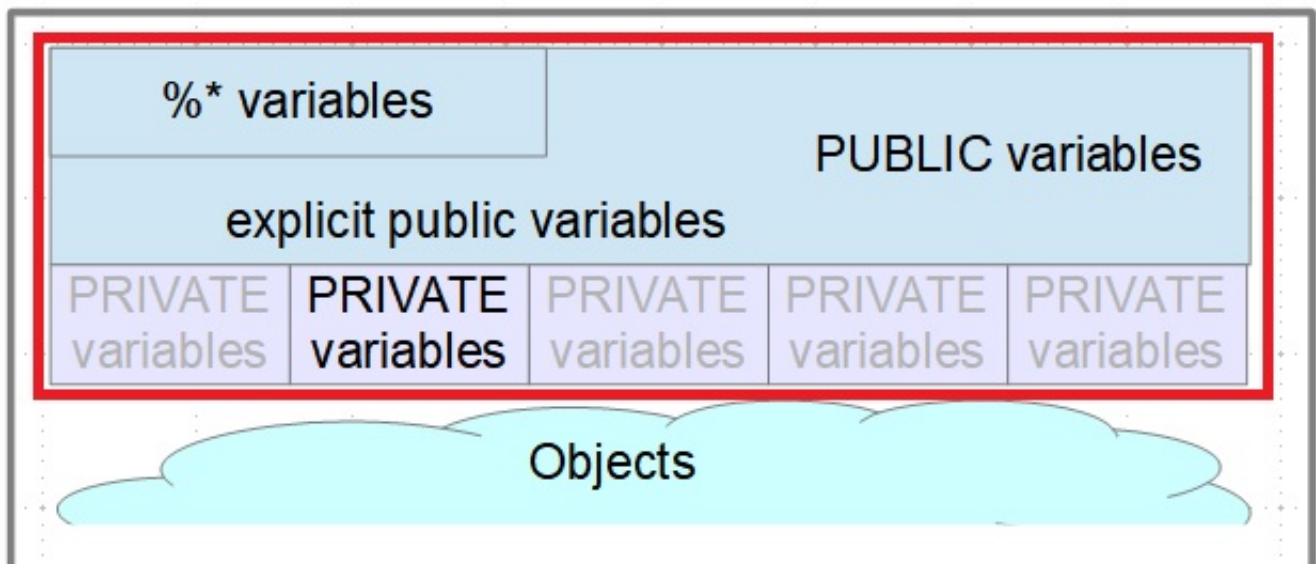
Para las definiciones de Clases y Métodos, el parámetro se denominó ProcedureBlock .

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=ROBJ_method_procedureblock

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GORIENT_class_method_keywords

El alcance tradicional de las variables se denominó PUBLIC (público) y el nuevo alcance se denominó PRIVATE (privado).

Con esta regla básica: cualquier cosa que empiece con % es una variable "PUBLIC". Cualquier otra cosa, requiere una configuración explícita.



La palabra clave ProcedureBlock (Default=1) controla si está disponible un ámbito PRIVATE para variables.
¡ Qué mejora!

Implementando funciones fáciles de usar, los procedimientos se volvieron bastante sencillos con alcance.
Además, la implementación de Métodos en las Clases de Objetos se volvió notablemente sencilla.

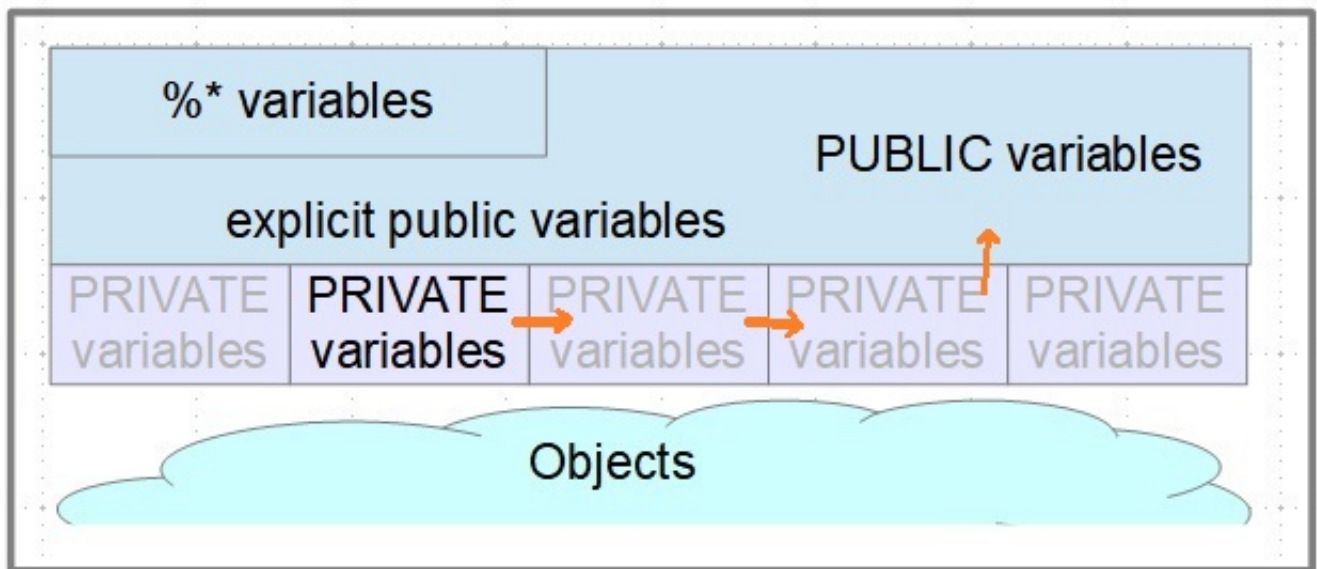
Ahora, en paralelo a todas las variables PUBLIC, se puede tener un gran conjunto de variables PRIVATE.

(¡ Ya no hay excusas para las variables con nombre!). Para que quede claro: al conjunto de variables PRIVATE se aplica el principio de Highlander:

¡ Solo puede haber una! <http://wiki.c2.com/?HighlanderPrinciple>

Hay 1 conjunto de variables Public y 1 conjunto de variables Private activas en un momento dado.

Pero, como en la vida real, hay una pequeña excepción: si pasas una variable por referencia a un método/procedimiento tienes una pequeña puerta de regreso al alcance de llamada.



La variante `ProcedureBlock=0` o `Not ProcedureBlock` apuntaba principalmente al código heredado que se basaba en la lógica antigua. PERO: de vez en cuando hay algunos rincones oscuros del código generado que todavía lo requieren/utilizan.

En general, los conflictos con el alcance variable son poco habituales.

Con 2 excepciones y 1 solución alternativa:

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCOS_usercode_indirxecjob

Xecute solo funciona con variables Public en su definición básica.

Hay una extensión que permite el paso de parámetros de forma similar a una llamada de procedimiento

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOS_cxecute

La función `$Xecute()` proporciona otra extensión en esta dirección.

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=RCOS_fxecute

La indirección solo funciona con variables public. No hay solución alternativa.

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCOS_operators#GCOS_operators_indirection

Otra buena razón para mantenerse alejado de la indirección.

Encontrarás más documentación relacionada con el alcance de las variables aquí:

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCOS_C16865

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GORIENT_ch_cos#GORIENT_cos_sc_ope

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCOS_usercode#GCOS_usercode_ars_byref

[#Principiante](#) [#Caché](#)

URL de fuente: <https://es.community.intersystems.com/post/resumen-sobre-el-alcance-de-las-variables-locales>