Discussion
Jack Huser · Jul 23, 2021

Performance when constructing a comma separated string

The goal is to construct a comma separated string, using a loop but the same way $listtostring(var,",") would do (i.e. "a,b,c,d,e,f,g,h" without starting or ending comma).

What is the best way to do it, concerning the readability of the code and its performances?

To do so, there are more than 5 methods, but in this example we will test 4 :

1) using a "if" in the loop

```
set str=""
for ... {
    if str'="" { set str = str_"," }
    set str=str_myval
}
```

2) using a set

```
set (str,sep)=""
for ... {
    set str=str_sep_myval
    set sep=","
}
```

3) using a $list construction and transform to string with $listtostring

```
set lst = ""
for ... {
    set lst=lst_$listbuild(myval)
}
set str = $listtostring(lst,",")
```

4) using a $select to check whether or not adding the separator

```
for ... {
    set str=str_$select(str'="":",",1:"")_myval
}
```

To do so we can create the method with checking $now() to calculate the time it takes to loop over

```
/// test String construction performance
ClassMethod TestPerformance(intINMaxLoop As %Integer = 100000)
{
    #dim tSC as %Library.Status = $$$OK
    try {
```

```
        set (time1, time2, intI) = 0
        set maxInt = intINMaxLoop
        set (now, str) = ""
        set myval = "a"

        write "Use if...", !
        set str=""
        set now = $now()
        set time1 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        for intI=1:1:maxInt {
            if str'="" { set str = str_"," }
            set str=str_myval
        }
        set now = $now()
        set time2 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        write "loop over if : "_(time2 - time1)_" ms", !

        write "Use set...", !
        set (str,sep)=""
        set now = $now()
        set time1 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        for intI=1:1:maxInt {
            set str=str_sep_myval
            set sep=","
        }
        set now = $now()
        set time2 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        write "Loop over set : "_(time2 - time1)_" ms", !

        write "Use $listtostring...", !
        set lst = ""
        set now = $now()
        set time1 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        for intI=1:1:maxInt {
            set lst=lst_$listbuild(myval)
        }
        set str = $listtostring(lst,",")
        set now = $now()
        set time2 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        write "Loop over $listbuild : "_(time2 - time1)_" ms", !

        set str=""
        write "Use $select...", !
        set now = $now()
        set time1 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        for intI=1:1:maxInt {
            set str=str_$select(str'="":",",1:"")_myval
        }
        set now = $now()
        set time2 = ($piece(now,",",1)*24*3600000)+$normalize(($piece(now,",",2)*1000
),0)
        write "Loop over $select : "_(time2 - time1)_" ms", !
```

```
    } catch (excep) {
        set tSC = excep.AsStatus()
    }
    if 'tSC { write $system.Status.GetErrorText(tSC), ! }
    else { write "ok", ! }
}
```

The result is : for smaller loop (up to 2000 items) the 5 methods are equal.

For bigger loop we can see that $select has performance issues...

The faster way is "set" which is not a surprise.

What is your favourite method?

I would rather use the "if" which sounds clearer to me when reading code. The $listbuild method is nice but we reach a <MAXSTRING> error faster.

Results

USER:23152>do ##class(User.TestJHU).TestPerformance(10000)
Use if...
loop over if : 0 ms
Use set...
Loop over set : 1 ms
Use $listtostring...
Loop over $listbuild : 0 ms
Use $select...
Loop over $select : 20 ms

USER:23152>do ##class(User.TestJHU).TestPerformance(50000)
Use if...
loop over if : 3 ms
Use set...
Loop over set : 2 ms
Use $listtostring...
Loop over $listbuild : 3 ms
Use $select...
Loop over $select : 323 ms

USER:23152>do ##class(User.TestJHU).TestPerformance(100000)
Use if...
loop over if : 6 ms
Use set...
Loop over set : 4 ms
Use $listtostring...
Loop over $listbuild : 7 ms
Use $select...
Loop over $select : 1524 ms

USER:23152>do ##class(User.TestJHU).TestPerformance(500000)
Use if...
loop over if : 39 ms
Use set...

```
Loop over set : 34 ms
Use $listtostring...
Loop over $listbuild : 48 ms
Use $select...
Loop over $select : 46021 ms


USER:23152>do ##class(User.TestJHU).TestPerformance(1000000)
Use if...
loop over if : 48 ms
Use set...
Loop over set : 39 ms
Use $listtostring...
Loop over $listbuild : 83 ms
Use $select...
Loop over $select : 187825 ms
```

(ps : I was not sure this text has the quality required for an article, so I post as "Discussion" instead)

#ObjectScript #Caché