

Article

[Robert Cemper](#) · Jul 19, 2020 2m read

Parameter passing to Language Extensions

IRIS and Caché / Ensemble offer the possibility to add [Languages Extension](#) as SystemFunctions, SystemVariables, SystemCommands

The related [documentation](#) tells you where to add the system-wide extension. Parameter passing is not covered explicitly.

In principal, it is always a call for a subroutine. And at the end, you terminate with `QUIT` returning a value or not.

SystemFunctions

You call them as any other standard system function with all required parameters in parenthesis.

eg `$ZZR(47,11)` and it will be processed. Just mentioned for completeness.

SystemVariables

There are 2 use cases: requesting a value `WRITE $ZZR` or setting a value `SET $ZZR=17`

The related [common](#) function has to detect what is going on. Like this:

```
%LANGV00 ;
ZZR(%a) public {
    if $d(%a) set ^mtemp.zzr($j)=%a quit
    quit $g(^mtemp.zzr($j))
}
```

SystemCommands

This is a little bit tricky as a distinction between multiple arguments and multiple parameters is required.

Multiple arguments are separated by a comma `","`: `ZZR arg1,arg2,..`

But if you require more than a single value as parameters

they have to be separated by colons `":"`: `ZZR par11:par12:par13, par21::par23, ::par33, par41, :par52`

or just nothing `ZZR`

The code should be prepared to handle missing parameters.

It is the decision of your code if the last parameter is interpreted as a timeout or anything else.

It is typically ended with `QUIT` without returning any value.

```
%LANGC00 ;
ZZR(%a,%b,%c) public {
    write $g(%a,"no a"),"#",!
    , $g(%b,"no b"),"#",!
    , $g(%c,"no c"),"*",!
    quit
}
```

[#ObjectScript](#) [#Tips & Tricks](#) [#Caché](#) [#Ensemble](#) [#InterSystems](#) [IRIS](#)

Source URL: <https://community.intersystems.com/post/parameter-passing-language-extensions>