

Question

[Joel Solon](#) · Feb 18, 2016

Variable Number of Arguments to a Method

This morning on the old Caché Google Group, someone posed the following question, which I've decided to answer here, because it's interesting!

Is there a way to iterate ClassMethod's params, and get param's names and values?

The first answer I can come up with is: it's not easy! In any method, you could try to write code like this (where methodName is the name of your method):

```
set method = ##class(%Dictionary.MethodDefinition).IDKEYOpen($classname(), method
Name)
set args = method.FormalSpec
for i=1:1:$length(args, ",") {
    set arg = $piece($piece(args, ",", i), ":", 1)
    write !, arg , " = ", @arg
}
```

But the problem is that the @arg won't work, because indirection doesn't have access to the private variables of the method, so you'll get an <UNDEFINED>. You could decide to make the method not use ProcedureBlock ([ProcedureBlock = 0]), or list all the arguments of the method in the PublicList of the method, so that the arguments are all public so that @arg works, but those seem like bad ideas to me.

So is the answer just: No? Well, not exactly. Another way is to use the "Variable Number of Arguments to a Method" technique, documented here:

<http://docs.intersystems.com/cache20152/csp/docbook/DocBook.UI.Page.cls?...>

As you'll see from the docs, the method signature has to use three dots, like this:

```
Method Test(args... as %String)
```

It's still not a generic solution; it forces you to write the method in a certain way ahead of time, if you want to be able to iterate through the arguments. But it does solve the problem in a straightforward way.

Anybody have any other ideas? Fire away!

[#Languages](#) [#Object Data Model](#) [#ObjectScript](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/variable-number-arguments-method>