

Using Interjob communication (IJC)

Article

[Robert Cemper](#) · Jun 5, 2019



1m read

[Open Exchange](#)

Using Interjob communication (IJC)

This is a coding example working on Caché 2018.1.3 and IRIS 2020.2
It will not be kept in sync with new versions
It is also **NOT** serviced by InterSystems **Support** !

Since the very beginning of Caché and most of its predecessors
there exists a way of in-memory communication of processes.
It's called **InterJob Communication** - IJC.

I have the feeling it's forgotten by most developers now.
here is the documentation: [Communication Between Caché Processes](#)

Earlier I've written about [command pipes](#).
This is the internal variant of a PIPE.

To make this more tangible and visible for you I prepared a small example

The scenario is to run a monitoring process that receives
input from an unknown number of sensors.
(Could be Lab equipment or similar.)
The monitor should not poll his sensors nor run
in a hang loop to scan a common ^global
and work independently of any disk access.

To try it log into a terminal

```
DO ##class(IJC.Demo).%Start()
```

The Highlander principle applies for this example: There can only be one

Next open a new terminal and run

```
DO ##class(IJC.Demo).Sensor("mytext")
```

and see what happens.

It's clear that for real applications:
- the monitor might run in a background tasks
- the sensors will do something useful.

But this reduces visibility.

[#Monitoring #Caché #Ensemble #InterSystems IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

```
120 0 0 0 291
```

Log in or sign up to continue
Add reply

Using Interjob communication (IJC)

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Source URL: <https://community.intersystems.com/post/using-interjob-communication-ijc>