
Article

[Steve Wilson](#) · Feb 3, 2017 3m read

Using Named Pipes in InterSystems Caché

Points to remember before you start:

1. It is not possible in a COS (Caché Object Script) job/process context to have multiple Named Pipes. It is a one Named Pipe per job/process limited line of communication.
2. Named Pipes, in Caché, like most pipes on most operating systems are Unidirectional. That means you open them for either Read or Write, but not both.
3. If you need a two-way bidirectional conversation between servers or jobs then consider using TCP connections or the IJC (inter Job Communication) paired ports.
4. The only real advantage a Named Pipe has over a TCP connection is the reduced Network impact, but it does also remove the need for one end to be running all the time – less likely to fail due to a temporary gap in process/job visibility – because writes are buffered, so if you want to send something and don't know (or care) when the other end receives and processes it then use a Named Pipe.
5. When trying to understand Named Pipe documentation in a COS context, the “Server” is the listening/receiving/Reading device end of the pipe and the non-server (local or otherwise) is the sending/writing end. Opening a Named Pipe with “ ” (empty string) for the host parameter is a Server Named Pipe Open, so trying to Write to that device will just hang the process. In other words, opening a server Named Pipe is the same as opening a pipe for READ only and opening a Named Pipe with a “.” or a string for the Host parameter is the same as opening for WRITE only.
6. Opening a Named Pipe is the same as opening a file, but in a non-UNIX environment you must have both a WRITE and a READ (non-server OPEN and server OPEN) before you can close the Named Pipe. Opening a Named Pipe for READ or WRITE without the corresponding end in another process will just hang/wait until the other end is opened. Imagine trying to save/close a NULL (not just empty but actually null) file – it doesn't even make sense.
7. Closing a Named Pipe without writing anything to it will cause an error at the READ (server) end. This is like hitting an EOF at the beginning of the file.
8. Named Pipes are primarily used for printing – sending (writing) files to print devices – so if you are looking to use them for anything else you may be straying down the wrong system design path.

A simple example to help us understand how to use Named Pipes in a COS routine can be shown using two separate Cache Terminal sessions to act as two “ ends ” . Run a process with a “ server ” Named Pipe type OPEN in one session and a corresponding “ client ” (WRITE) OPEN in the other. The two routines look like this...

...and...

...and when the server and then client are run this gives us outputs something like this...

...and...

...the only thing not shown here is the pause/wait that we have on the “ server ” while we wait for the “ client ” input of the string and then hit return.

References:

For more information on using Pipes in Cache and Ensemble

...

<http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GIODinterprocomm#GIODinterproccommpipesandcommandpipes>

For more information on Named Pipes in general, this Wiki page is good...

<https://en.wikipedia.org/wiki/Namedpipe>

[#Code Snippet](#) [#ObjectScript](#) [#Tips & Tricks](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/using-named-pipes-intersystems-cach%C3%A9>