Article
Amir Samary · Oct 12, 2017 4m read

How to interactively run a Windows or Unix command from inside Caché/Ensemble

Hi!

It is often necessary to run some external command such as a python program or a shell script from inside Caché/Ensemble. There are three ways of doing this:

- $ZF(-1) - Runs the command and waits for it to finish.
- $ZF(-2) - Runs the command and don't wait for it to finish.
- Using CPIPE device - Runs the command and opens a device for you to read its output or (exclusive or here!) write to its input.

$ZF(-1) is normally more interesting because it allows you to recover the return code of the program you are calling. If that is important to you, that is the way to go. But while the program is running, your Caché process will be hung and you won't be able to stop the instance cleanly until it is finished. Normally, for short processes, this is totally acceptable.

$ZF(-2) is interesting if you know for a fact that the program will take a long time to run and you obviously can't or is not interested into waiting for it to finish.

In both cases if you need to capture the output of the program you are calling, you must redirect it to a temporary file, open the file later and read from it. Also, you can't run the program interactively (like answering for prompts).

CPIPE devices are interesting if:

- You are not interested into the program return code but want the output and wait for it
- You want the output of the program but don't want to handle temporary files or...
- You want to run the program interactively so that the program can work accordingly to your input

Here is a quick an dirty example that will work on most unix/linux/mac platforms:

```
Set tDevice="|CPIPE|"

Set tCmd="bash"

Set tInstructions="vi /Users/amirsamary/test.txt"_$C(13,10)_"iHello world!"_$C(27)_":wq"_$C(13,10)_"exit"_$C(13,10)"

Open tDevice:(tCmd:"W")

Use tDevice Write tInstructions Use 0

Close tDevice
```

I am running this on Caché Terminal so that the **Use 0** command means to go back to my initial standard device. If you are running this on a job it is wise to save your current open device (that is on $IO variable) before changing to the CPIPE device. So that you can come back to it later:

```
Set tCurrentDevice=$IO
Set tDevice="|CPIPE|"

Set tCmd="bash"

Set tInstructions="vi /Users/amirsamary/test.txt"_$C(13,10)_"iHello world!"_$C(27)_":
wq"_$C(13,10)_"exit"_$C(13,10)"

Open tDevice:(tCmd:"W")

Use tDevice Write tInstructions Use tCurrentDevice
Close tDevice
```

Please change the path "/Users/amirsamary/" to a path of your choice. You will find a file test.txt with the string "Hello world" inside it. Cool ah?

If all you need is to run a command an get its output you can do something like:

```
Set tDevice="|CPIPE|"

Set tCmd="ls -l"

Open tDevice:(tCmd:"R")

Use tDevice Read line Use 0 Write line

total 2056

Use tDevice Read line Use 0 Write line

-rw-rw----  1 amirsamary  cacheusr  1048576 Aug 15 12:06 CACHE.DAT

Use tDevice Read line Use 0 Write line

-rw-rw----  1 amirsamary  cacheusr       53 Oct 10 11:30 cache.lck

Use tDevice Read line Use 0 Write line

drwxrwxr-x  2 amirsamary  cacheusr       64 Jul 31  2016 stream

USER>u device r line u 0 write line

U device R line U 0 WRITE line
         ^
<ENDOFFILE>
```

As you can see, I can continue reading from this device until I reach <ENDOFFILE> so you will probably want to to protect your code with a Try/Catch or temporarily change the behavior of EOF.

So, pick your choice accordingly to your needs. Using these methods sometimes can be confusing. One must be careful with closing the devices after using it, protecting the code for the ENDOFFILE error, etc. If all you need is to run some external program, get its return code and, maybe, a short output of less than 32000 bytes, you may want to take a look at the method RunCommandViaZF() on class %Net.Remote.Utility.

If you don't want a temporary file in the middle and your output is less than 32000 you can use

RunCommandViaCPIPE() method on the same class.

If your output is larger than 32000 than you can use RunCommandViaCPIPE() and redirect your output to a temporary file that you will have to open from yourself:

```
USER>Set tSC = ##class(%Net.Remote.Utility).RunCommandViaCPIPE("ls -l > /Users/amirsa
mary/test.txt", .tDevice)

USER>Write tSC
1

USER>Close tDevice

USER>Set oFile=##class(%File).%New("/Users/amirsamary/test.txt")

USER>Write oFile.Open("R")
1

USER>Do oFile.OutputToDevice()

total 2056-rw-rw----  1 amirsamary  cacheusr  1048576 Aug 15 12:06 CACHE.DAT-rw-rw---
-  1 amirsamary  cacheusr       53 Oct 10 11:30 cache.lckdrwxrwxr-
x  2 amirsamary  cacheusr       64 Jul 31  2016 stream

USER>
```

Don't forget to delete the file later! Take a look at the source code of these methods for more examples of using $ZF(-1) and CPIPE!

I hope that helps!

#API #Callout #Code Snippet #Object Data Model #Tips & Tricks #Caché

Source URL:https://community.intersystems.com/post/how-interactively-run-windows-or-unix-command-inside-cach%C3%A9ensemble