
Artículo

[Alberto Fuentes](#) · 9 jul, 2019 Lectura de 4 min

Ejecutar un comando externo (Windows / Unix) de forma interactiva desde IRIS

¡Hola! Me gustaría revisitar hoy con vosotros un artículo muy útil de Amir Samary donde se detalla cómo ejecutar un comando del sistema operativo desde IRIS / Caché / Ensemble **de forma interactiva**.

Con frecuencia es necesario ejecutar algún comando externo, como por ejemplo un programa Python o un script de bash desde Caché/Ensemble. Existen dos formas principales:

- [\\$ZF\(-100\)](#) - Ejecuta un comando como un proceso hijo. La ejecución puede ser síncrona o asíncrona. Permite redireccionar la entrada y salida estándar a ficheros. Esta función reemplaza a las obsoletas [\\$ZF\(-1\)](#), [\\$ZF\(-2\)](#).
- [CPIPE](#) - Ejecuta el comando y abre un dispositivo para leer la salida o incluso escribir de forma interactiva la entrada.

[\\$ZF\(-100\)](#) es una opción interesante para ejecutar el comando y recuperar el resultado, sin embargo, para capturarla información de salida que se produce, es necesario redirigirla a un fichero temporal. Además, no permite ejecutar un programa de forma interactiva.

La ejecución utilizando [CPIPE](#) resulta conveniente cuando no queremos tener que procesar ficheros temporales o cuando nos interesa ejecutar el programa de forma interactiva.

La mejor forma de verlo es comprobar rápidamente su funcionamiento.

Este ejemplo funcionará en la mayoría de sistemas Unix / Linux / Mac. Crea un fichero de texto con contenido "Hellow World!" utilizando el editor de texto vi de forma interactiva desde IRIS (fijaos cómo pasa las instrucciones para el editor vi incluido el Insert y el Write and Quit).

```
USER>set tCurrentDevice=$IO

USER>set tDevice="|CPIPE|"

USER>set tCmd="bash"

USER>
set tInstructions="vi /Users/afuentes/test.txt"_$c(13,10)_"i"_"Hellow World!"_$c(27)_"
":wq"_$c(13,10)_"exit"_$c(13,10)

USER>open tDevice:(tCmd:"w")

USER>use tDevice Write tInstructions Use tCurrentDevice

USER>close tDevice
```

Si lo único que precisamos es ejecutar un comando y obtener la salida, podemos hacer como en el siguiente

ejemplo en el que se muestra la salida de un listado de directorios.

```
USER>set tDevice="|CPIPE|"

USER>set tCmd="ls -l"

USER>open tDevice:(tCmd:"R")

USER>use tDevice Read line Use 0 Write line

total 1028

USER>use tDevice Read line Use 0 Write line

-rw-rw---- 1 root irisusr 1048576 Jul  1 14:58 IRIS.DAT

USER>use tDevice Read line Use 0 Write line

-rw-rw---- 1 root irisusr      40 Jul  1 14:07 iris.lck

USER>use tDevice Read line Use 0 Write line

drwxrwxr-x 2 root irisusr      64 Apr 30 19:12 stream

USER>use tDevice Read line Use 0 Write line

USE tDevice READ line USE 0 WRITE line

      ^

<ENDOFFILE>

USER>
```

Como veis, podemos continuar leyendo del dispositivo CPIPE hasta alcanzar el <ENDOFFILE>. Así que probablemente en el código habría que añadir un bloque try-catch. Fijaos como además se utiliza directamente Use 0 porque estamos ejecutándolo desde un Terminal, pero en general, al igual que en el ejemplo anterior, podemos hacer uso de \$IO para obtener el dispositivo abierto actual.

Elegiremos el método de ejecución que más nos convenga según las necesidades. En cualquier caso, merece la pena echarle un vistazo a la clase [%Net.Remote.Utility](#) y los métodos RunCommandViaZF y RunCommandViaCPIPE que encapsulan el funcionamiento de estas funciones de ejecución de comandos externos.

```
USER>set cmd="/bin/bash"

USER>set arg=2

USER>set arg(1)="-c"

USER>set arg(2)="ps -fea | grep iris"

USER>set ret = ##class(%Net.Remote.Utility).RunCommandViaCPIPE(cmd, .device, .out,,, .arg)

USER>close device
```

```
USER>set outList = $listfromstring(out, $c(13,10))

USER>write $listget(outList, 1)

root      1      0  0 04:25 ?      00:00:00 /dev/init -- /iris-
main --key /external/iris.key

USER>write $listget(outList, 2)

root      7      1  0 04:25 ?      00:00:01 /iris-main --key /external/iris.key

USER>write $listget(outList, 3)

root      772     1  0 04:25 ?      00:00:01 /usr/irissys/bin/irisdb -s/external/d
urable/mgr/ -w/external/durable/mgr/ -cc -B -C/external/durable/iris.cpf*IRIS

USER>write $listget(outList, 4)

root      773    772  0 04:25 ?      00:00:00 /usr/irissys/bin/irisdb WD
```

Tened en cuenta que si la salida de nuestro comando es mayor que el tamaño máximo de un string (3 641 144 caracteres), podemos redirigir la salida a un fichero temporal.

```
USER>set cmd="/bin/bash"

USER>set arg=2

USER>set arg(1)="-c"

USER>set arg(2)="ls -l > /tmp/ls.txt"

USER>set ret=##class(%Net.Remote.Utility).RunCommandViaCPIPE(cmd, .device, .out,,, .ar
g)

USER>write ret

1

USER>close device

USER>set file=##class(%Stream.FileCharacter).%New()

USER>set file.Filename="/tmp/ls.txt"

USER>write file.ReadLine()

total 1028

USER>write file.ReadLine()

-rw-rw---- 1 root irisusr 1048576 Jul  1 14:58 IRIS.DAT

USER>write file.ReadLine()
```

```
-rw-rw---- 1 root irisusr      40 Jul  1 14:07 iris.lck
```

[#ObjectScript](#) [#Caché](#) [#Ensemble](#) [#InterSystems](#) [IRIS](#)

URL de
fuente: <https://es.community.intersystems.com/post/ejecutar-un-comando-externo-windows-unix-de-forma-interactiva-desde-iris>