Article
Jack Huser · Sep 13, 2021 6m read

Use $system.external Interface for Python

Since I saw many posts on Developer Community related to Python, and the very good articles and application written by @Eduard Lebedyuk I was wondering: "As a Object Script developer, why would I want to use an other language in Object Script? If I ever need to execute something in Object Script, I would do it in Object Script!".

I thought those functionalities to use other languages in Object Script were made only for other languages developers who have to write Object Script code.

Recently I had to parse a huge CSV file : 1.7Gb and more than 5 millions lines.

I did it in Object Script:

```
ClassMethod ReadFile(strINReadFile As %String = "") As %Status
{
	#dim tSC As %Library.Status = $$$OK
	#dim FileReader As %Library.File
	try {
		set FileReader = ##class(%Library.File).%New(strINReadFile)
		set tSC = FileReader.Open("RU")
		if $$$ISERR(tSC) { quit }
		set FileReader.LineTerminator = $$$NL
		set nbLigne = 0
		set time1 = $zh
		while (FileReader.AtEnd = 0) {
			set len = 32000
			set (strBuffer, eol) = ""
			set strBuffer = FileReader.ReadLine(.len, .tSC, .eol)
			if $$$ISERR(tSC) { quit }
			// do something with strBuffer
		}
		quit:$$$ISERR(tSC)
		set time2 = $zh
		set diff = time2 - time1
		write "execution: "_diff, !
	} catch (SysEx) {
		set tSC = SysEx.AsStatus()
	}
	if (($data(FileReader)>0) && (FileReader'="")) {
		do FileReader.Close()
	}
	quit tSC
}
```

Result was disappointing

```
USER>W ##class(JHU.Test).ReadFile("C:\Temp\GigaFile.csv")
execution: 892.108104s
1
```

Almost 15 minutes !!!

Using @Robert Cemper (Thank you so far) code results are

```
 /// Read quit
ClassMethod ReadQuick(strINReadFile As %String = "") As %Status
{
     #dim tSC As %Library.Status = $$$OK
     #dim SysEx As %Exception.AbstractException
     try {
          open strINReadFile::1
          else  set tSC=$$$ERROR($$$GeneralError, "Missing File") quit
          set eof=##class(%SYSTEM.Process).SetZEOF(1)
          use strINReadFile
          set time1=$zh
          for line=0:1 {
               read strBuffer if $zeof set diff=$zh-time1 quit
               // do something with strBuffer
           }
          close strINReadFile
          do ##class(%SYSTEM.Process).SetZEOF(eof)
          write !,"execution: "_diff,!,"lines: ",line,!
     } catch (SysEx) {
          set tSC = SysEx.AsStatus()
     }
     quit tSC
}
```

Results are

```
USER>W ##class(JHU.Test).ReadQuick("C:\Temp\GigaFile.csv")

execution: 10.047812
lines: 5000000
1
```

The same file parsing in Python would be

```
from datetime import datetime

class Test1:

   def ReadFile(self,strINFileName="") :
      if strINFileName=="":
         print("file name is empty")
         quit()
      file = open(strINFileName,"r")
      atEnd = False
      time1 = round(datetime.timestamp(datetime.now()) * 1000)
      while not atEnd:
         line=file.readline()
         if not line :
            atEnd = True
      time2 = round(datetime.timestamp(datetime.now()) * 1000)
      file.close()
```

```
    print("Execution: ",((time2-time1)/1000),"s")
```

Result was far beyond expectation

```
obj = Test1()
obj.ReadFile("C:\Temp\GigaFile.csv")
Execution:  5.222 s
```

So I wanted to parse the huge file in Object Script but using Python.

With IRIS 2021.1 comes the Interface for external languages with Python: Working with External Languages.

The call for Python Gateway using $system.external Interface is:

```
 /// Read File using Python
ClassMethod ReadFileWithPython(strINFilename As %String = "")
{
   #dim tSC As %Library.Status = $$$OK
   #dim SysEx As %Exception.AbstractException
   try {
      set gateway = $system.external.getPythonGateway()
      do gateway.addToPath("C:\Projet\Python\test1.py")
      set fooProxy = gateway.new("test1.Test1")
      do fooProxy.ReadFile(strINFilename)
   } catch (SysEx) {
      set tSC = SysEx.AsStatus()
   }
   if $$$ISERR(tSC) { write $system.Status.GetErrorText(tSC), ! }
}
```

Result is as expected

```
USER>do ##class(JHU.Test).ReadFileWithPython("C:\Temp\GigaFile.csv")
Execution:  4.387 s
```

In fact it makes Object Script more attractive and makes me want to learn more of Python.

And I'm looking forward for Embedded Python within Object Script Class or ClassMethod.
As an example the excellent article from @Henry Pereira

#ObjectScript #Python #InterSystems IRIS