

Example of Basic Script

Presented below is a script which can be used as a shell to get started in developing your own custom check/script.

This script template is located in the documentation folder in the YourProgramFiles/GeoSystemsMonitor/Documentation and it is called ExampleScript.py. Each section is marked with double `##` signs. This script is ready to go with the exception of adding your own variables/parameters needed, and the actual check to be performed.

To add your own variables/parameters, you will first add them in the **varList**

```
varList = ["port", "server", "urlCheck", "debug", "enginePID", "enginePort", "engineHost"]
```

For example if you wanted to add a variable called '**monitor**' that my script wants to use and pass from the engine.

```
varList = ["port", "server", "urlCheck", "debug", "enginePID", "enginePort", "engineHost",  
"monitor"]
```

Then add '**monitor**' to the following section

```
for argv in theArgvList:
```

```
    argvL = argv.split("^")
```

```
    var = argvL[0]
```

```
    val = argvL[1]
```

```
    if var in varList:
```

```
        if var == "port":
```

```
            port = int(val)
```

```
        elif var == "server":
```

```
            server = val
```

```
        elif var == "urlCheck":
```

```
            urlCheck = val
```

```
        elif var == "debug":
```

```
            debug = val
```

```
        elif var == "enginePID":
```

```
            enginePID = val
```

```
        elif var == "enginePort":
```

```
            enginePort = int(val)
```

```
        elif var == "engineHost":
```

```
engineHost = val

elif var == "monitor":

    monitor = val
```

You now have a script accepting variables passed into it from the engine.

Next you will want to do your own 'check' within the 'Section to perform Point Check'.

Make sure you set the 'status' variable when finished that gets passed back into the engine.

To have a unique debug file for each type of script, change the debug file name:

```
#Debugfile
```

```
if debug:
```

```
        debugfile = open('ExampleDebugLog.txt','a')
```

For testing your script you can change the debug parameter to True

```
# Debug either True or False 'Capital'
```

```
debug = False
```

ExampleScript.py

```
# This Python script is an Example for the GeoSystems Monitor
```

```
# This is a shell to demonstrate the sections for a point check
```

```
#
```

```
#
```

```
# Copyright VESTRA Resources Inc.
```

```
# Davin Laine
```

```
## Section Import Libraries ##
```

```
import sys,os,time,urllib2
```

```
from socket import *
```

```
## Section to initialize variables ##
```

```
urlCheck = 'http://testserver/testapp'
```

```
server = 'testserver'
```

```
# Debug either True or False 'Capital'
```

```
debug = False
```

```
#Timeout in seconds
```

```
timeout = 30
```

```
setdefaulttimeout(timeout)
```

```
# return arguments
```

```
returnargs = False
```

```
# sets initial variables
```

```
urlResponse = " ; errorcode = " ; urlStatus = "
```

```
# GIS Engine variables -
```

```
engineHost = '10.1.1.111'
```

```
enginePort = 8888
```

```
enginePID = 2222
```

```
# time, new-line, space variables
```

```
tme = time.ctime()
```

```
nl = '\n'
```

```
sp = ' '
```

```
## Section to parse Parameters passed into the script ##
```

```
# *****
```

```
# The following reads arguments passed into the script
```

```
# and assigns parameters to the variables, overriding
```

```
# variables set from above
```

```
# *****
```

```
varList = ["port", "server", "urlCheck", "debug", "enginePID", "enginePort", "engineHost"]
```

```
theArgvList = []
```

```
argvLen = len(sys.argv)
```

```
for argvNum in range(1,argvLen):
```

```

theArgvList.append(sys.argv[argvNum])

for argv in theArgvList:

    argvL = argv.split("^")
    var = argvL[0]
    val = argvL[1]

    if var in varList:

        if var == "port":

            port = int(val)

        elif var == "server":

            server = val

    elif var == "urlCheck":

        urlCheck = val

    elif var == "debug":

        debug = val

    elif var == "enginePID":

        enginePID = val

    elif var == "enginePort":

        enginePort = int(val)

    elif var == "engineHost":

        engineHost = val

# *****

# end of argument parsing

# *****

#Debugfile

if debug:

    debugfile = open('ExampleDebugLog.txt','a')

```

```

if debug:

    debugfile.write(time.ctime() + sp + 'Entry-----')

    debugfile.write(' Server: %s  Url: %s'%(server,urlCheck))

    debugfile.write(nl)

    debugfile.write(time.ctime() + sp + 'Passed Parameters ')

    debugfile.write(' engineHost: %s  enginePort: %s  enginePID: %s'
%(engineHost,enginePort,enginePID))

    debugfile.write(nl)


## Section to perform Point Check ##


# Start time for service - Used for service response - Performance
startSvctime = time.clock()


# **** Try's to access URL and looks for an exception, reads response if available ****
try:

    # Actual Check for point

    # Actual Check for point

    status = 'good'


except IOError, e:

    if hasattr(e,'code'):

        errorcode = e.code

    elif hasattr(e,'reason'):

        errorcode = e.reason

    status = 'error'


# Total time for service response - Performance
serviceTime = "%.3f" % (time.clock() - startSvctime)


if debug:

    debugfile.write(time.ctime())

    debugfile.write(' status: ' + urlStatus + sp + 'response: ' + urlResponse)

    debugfile.write(nl)

```

```

        debugfile.write(time.ctime())

        debugfile.write(' errorcode: ' + str(errorcode))

        debugfile.write(nl)

# if above checks are not successful, set 'unknown' for status
if status == '':

    status = 'unknown'

## Section to set return parameters to engine ##
# Builds return arguments to GIS Monitor Engine
# server, url and status

returnargs = sp + 'server=' + server + sp + 'status=' + status + sp + 'url=' + urlCheck + sp +
'serviceTime=' + str(serviceTime) + sp + 'enginePID=' + str(enginePID)

if debug:

    debugfile.write(time.ctime())

    debugfile.write(' returnargs:' + sp + returnargs)

    debugfile.write(nl)

# for trouble shooting to console, uncomment below line
print 'server is: %s - status is: %s - errorcode: %s' % (server, status, errorcode)


# *****

# Makes a socket connection to GIS Monitor Engine and sends returnargs

# *****


## Section to send return parameters to engine ##
try:

    sHost = socket(AF_INET, SOCK_STREAM)

    sHost.connect((engineHost, enginePort))

    sHost.send(returnargs)

    sHost.close()

    if debug:

```

```

        debugfile.write(time.ctime())

        debugfile.write(' Sockethost:' + sp + 'sent')

        debugfile.write(nl)

except error, e:

    if hasattr(e,'args'):

        errorcode = e.args

    elif hasattr(e,'message'):

        errorcode = e.message

    sHostStatus = 'error'

    if debug:

        debugfile.write(time.ctime())

        debugfile.write(' EngineSocketHostError:' + sp + str(errorcode))

        debugfile.write(nl)

# *****

# End of socket connection to GIS Monitor Engine

# *****

if debug:

    debugfile.close()

sys.exit(returnargs)

```

Article ID: 327

Last updated: 27 Jul, 2018

Revision: 3

GeoSystems Monitor Enterprise -> Product Guide v4.0 -> Developing Custom Checks -> Example of Basic Script

<http://www.vestra-docs.com/index.php?View=entry&EntryID=327>