



```
from java.lang import *  
  
from java.util import *  
  
from com.ti.debug.engine.scripting import *  
  
from com.ti.ccstudio.scripting.environment import *  
  
# Create our scripting environment object - which is the main entry point into any  
script and  
  
# the factory for creating other Scriptable Servers and Sessions  
  
script = ScriptingEnvironment.instance()  
  
# Create a log file in the current directory to log script execution  
  
script.traceBegin("BreakpointsTestLog_python.xml", "DefaultStylesheet.xml")  
  
  
  
# Set our TimeOut  
  
script.setScriptTimeout(15000)  
  
  
  
# Log everything  
  
script.traceSetConsoleLevel(TraceLevel.ALL)  
  
script.traceSetFileLevel(TraceLevel.ALL)  
  
  
  
# Get the Debug Server and start a Debug Session  
  
debugServer = script.getServer("DebugServer.1")
```



```
debugServer.setConfig("../bin/target-config/awr1843.ccxml");

debugSessionOne = debugServer.openSession("Texas Instruments XDS110 USB
Debug Probe_0/Cortex_R4_0")

debugSessionTwo = debugServer.openSession("Texas Instruments XDS110 USB
Debug Probe_0/C674X_0")

debugSessionOne.target.connect()

debugSessionTwo.target.connect()

# Load a program

# (ScriptingEnvironment has a concept of a working folder and for all of the APIs
which take

# path names as arguments you can either pass a relative path or an absolute path)

debugSessionOne.memory.loadProgram("../bin/target-config/xwr18xx_mrr_ti_design_
mss.xer4f")

debugSessionTwo.memory.loadProgram("../bin/target-config/xwr18xx_mrr_ti_design_
dss.xe674")

debugSessionOne.memory.loadBinaryProgram("../bin/target-config/ROC_LCP_FLAS
H_debug.bin",0x510e0000)

# Run the target. Should halt at our breakpoint.

debugSessionOne.target.run()

debugSessionTwo.target.run()
```



```
# All done
#debugSessionOne.terminate()
#debugSessionTwo.terminate()
#debugServer.stop()
```

```
waitUntil: RETURN
load: Program load successful
getBoolean: ENTRY ID: AutoRunToLabelOnRestart
getBoolean: RETURN true
waitForHaltIfPropertySet: Waiting for halt
waitUntil: ENTRY timeout: 15000 (ms)
log: Target has halted at 0x20000140
waitUntil: RETURN
loadProgram: RETURN
loadBinaryProgram: ENTRY sFileName: ../bin/target-config/ROC_LCP_FLASH_debug.bin
loadBinary: Requesting program load
waitUntil: ENTRY timeout: 15000 (ms)
waitUntil: RETURN
loadBinary: Program load successful
getBoolean: ENTRY ID: AutoRunToLabelOnRestart
getBoolean: RETURN true
loadBinaryProgram: RETURN
run: ENTRY
go: Requesting target execution
waitUntil: ENTRY timeout: 15000 (ms)
Debug: Launched the Initialization Task
PMIC is configured correctly
Debug: Initialized the mmWave module
Debug: Synchronized the mmWave module
SEVERE: Timed out after 15000ms
SEVERE: com.ti.debug.engine.scripting.Target.run(): Timed out after 15000ms
Traceback (most recent call last):
  File "C:\ti\ccs1100\ccs\ccs_base\scripting\examples\DebugServerExamples\my_script.py", line 39, in <module>
    debugSessionOne.target.run()
    at com.ti.debug.engine.scripting.ExecutionOperation.go(ExecutionOperation.java:68)
    at com.ti.debug.engine.scripting.Target.run(Target.java:818)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
    at java.base/java.lang.reflect.Method.invoke(Unknown Source)
com.ti.ccstudio.scripting.environment.ScriptingException: com.ti.debug.engine.scripting.Target.run(): Timed out after 15000ms
C:\Tool\Python\bin>
```