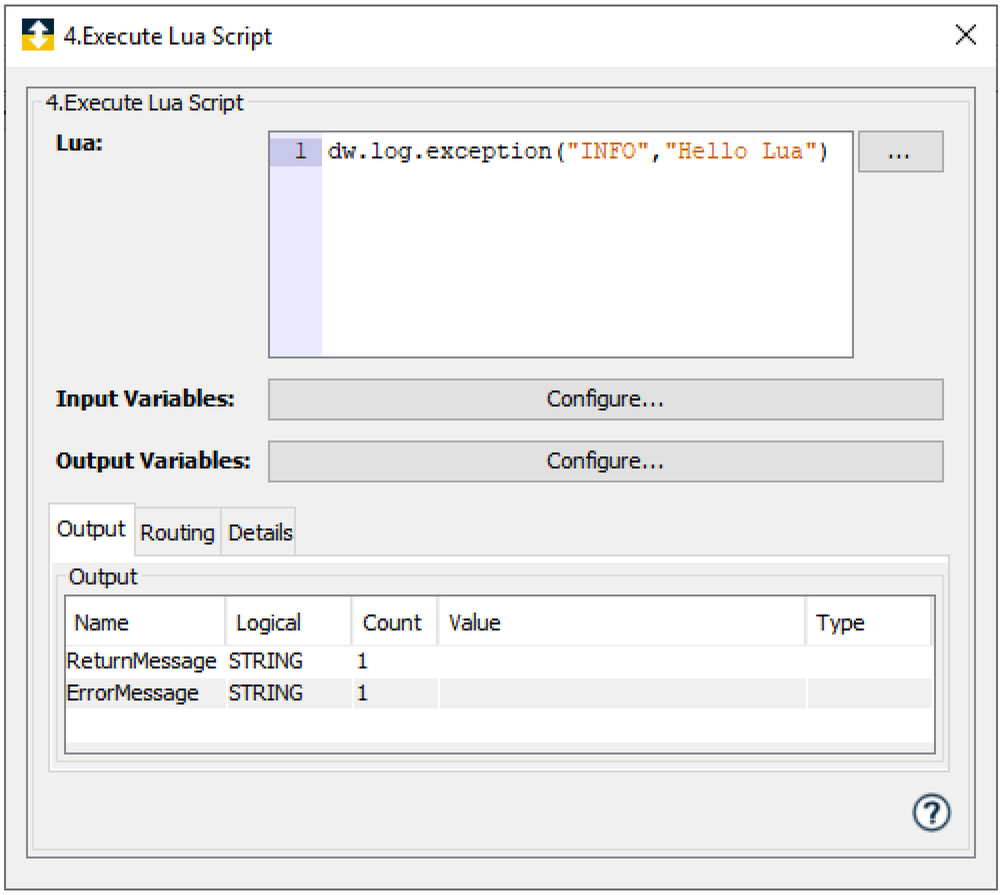
You are here: IoT Gateway User Guide > Projects and triggers > Trigger actions reference > Execute Lua Script

Topic updated on September 21, 2023

Execute Lua Script

The **Execute Lua Script** action executes a custom Lua script defined in the action. The Lua script contains scripting code that normally executes inside a Lua function or it contains a mix of scripting code and functions themselves. To execute a Lua function defined in a file that has been uploaded to the staging directory, see Execute Lua Function From File.



```
dw.log.exception("INFO","Hello Lua")
```

Name	Logical	Count	Value	Type
ReturnMessage	STRING	1		
ErrorMessage	STRING	1		

Parameter descriptions

Parameter	Description
Lua	The custom Lua script is entered directly in this parameter. This can contain lines of Lua scripting code or entire Lua functions.
Input Variables	Variables that will be passed into the Lua script when execution of the script begins. When an input variable is added using the **Configure...** Variables window, it is also added to the Input tab.
Output Variables	Variables that will be returned by the Lua script when execution of the script ends. When an output variable is added using the **Configure...** Variables window, it is also added to the Output tab.

Input tab

Parameter	Description
Input Variables	Input variables will appear and can be mapped to variables when **Input Variables** parameters are added using the **Configure...** Variables window.

Output tab

Parameter	Description
Output Variables	Output variables will appear and can be mapped to variables when **Output Variables** parameters are added using the **Configure...** Variables window.
ReturnMessage	A return message set by the Lua script. The script can return a numeric code or string.
ErrorMessage	An error message set by the Lua script.

For additional information on the runtime's support of Lua, see Extending the system using Lua scripting. For a usage example of the **Execute Lua Function From File** action, see Executing a Lua function from a Lua script file.

Example 1: LUA Script

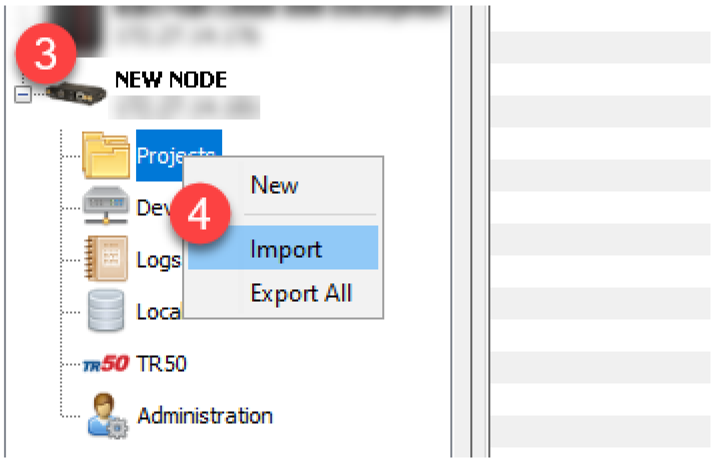
```
function logsimple()
   dw.log.debug("INFO","Log Message")
end
logsimple();
dw.log.debug("INFO","Log Message2");
```

Example 2: LUA Hexadecimal to Decimal converter Script

The LUA Script is defined to convert a Hexadecimal value to a Decimal value and write it to a device output variable. To use the sample, do the following:

1. Download the project Project_Lua.dwx
2. Open the Workbench. To learn about workbench, see Workbench
3. Expand the node to which you want to import the downloaded project

4. Right click on the Projects icon and click **Import**



   Import File Location window opens

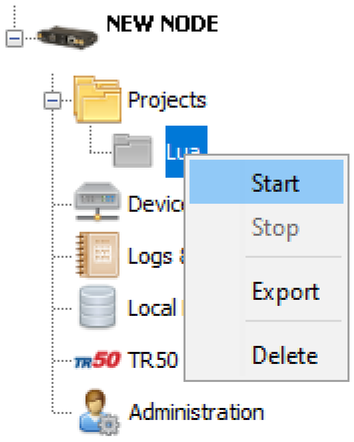
5. Select the downloaded file (Project_Lua.dwx)

6. Click **Select**

   Import window appears and to view the dependencies click the expand button.

7. Click **Import**

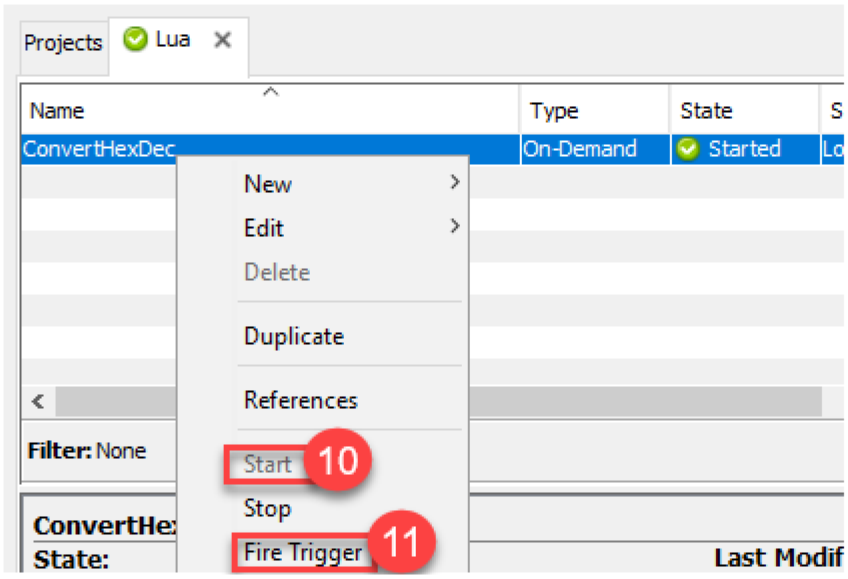
8. Right click on the imported project and click **Start**



9. Click on the Imported Project and you will view the trigger (*converHexDec*)

10. Right click and select **Start** to start the trigger

11. Right click and select **Fire Trigger**



The trigger succeeds and you will see the success count increase.

Trigger insight

1. Double click on the trigger(*converHexDec*) to view the trigger definition

   The trigger window appears

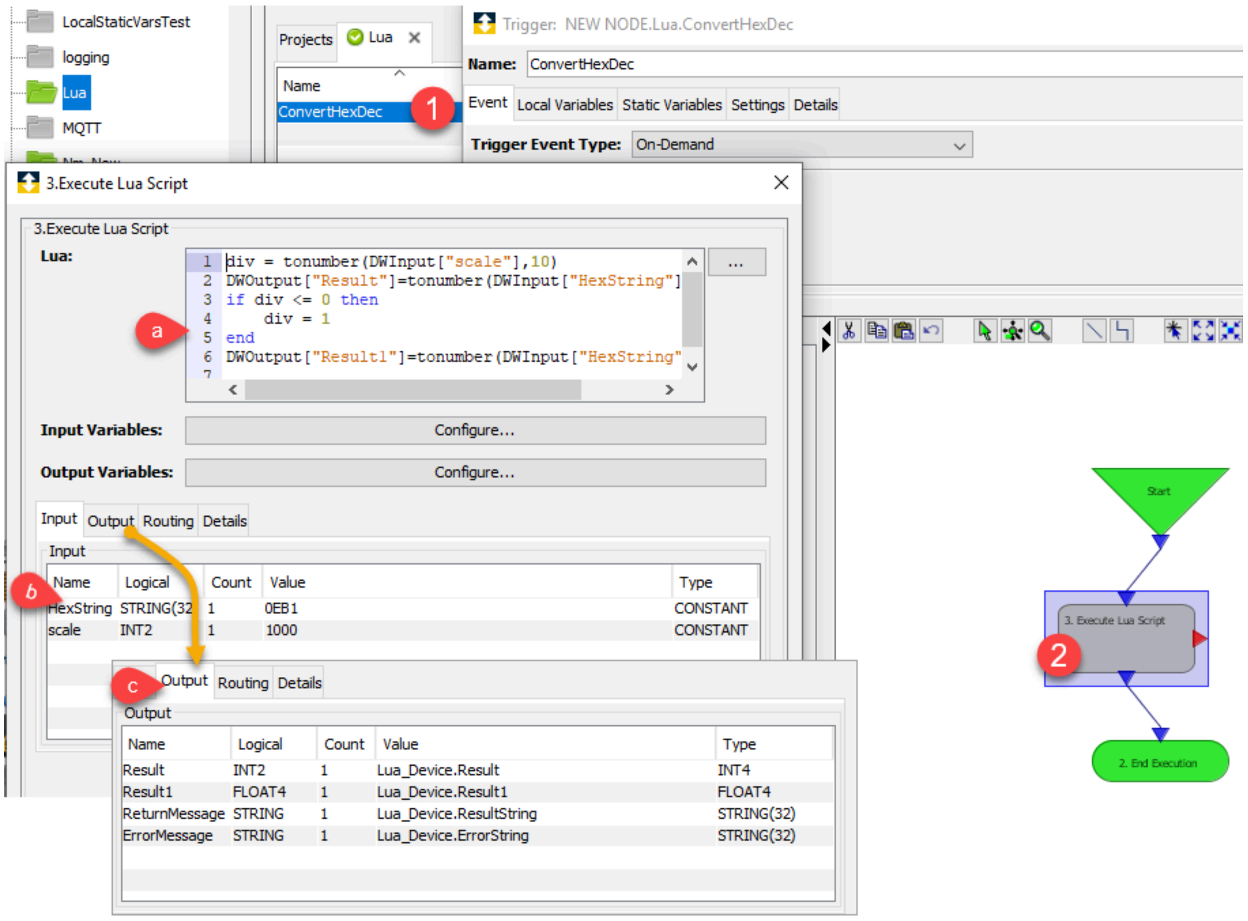
2. Double click on the Execute Lua Script Action

   You will see the following

   a. Lua script

   b. Input Hex String to get converted using the Lua script

c. Click output to see the device output variable to which the output is written



Name: ConvertHexDec

Trigger Event Type: On-Demand

Lua:

```
div = tonumber(DWInput["scale"],10)
DWOutput["Result"]=tonumber(DWInput["HexString"]
if div <= 0 then
    div = 1
end
DWOutput["Result1"]=tonumber(DWInput["HexString"
```

Input Variables: Configure...

Output Variables: Configure...

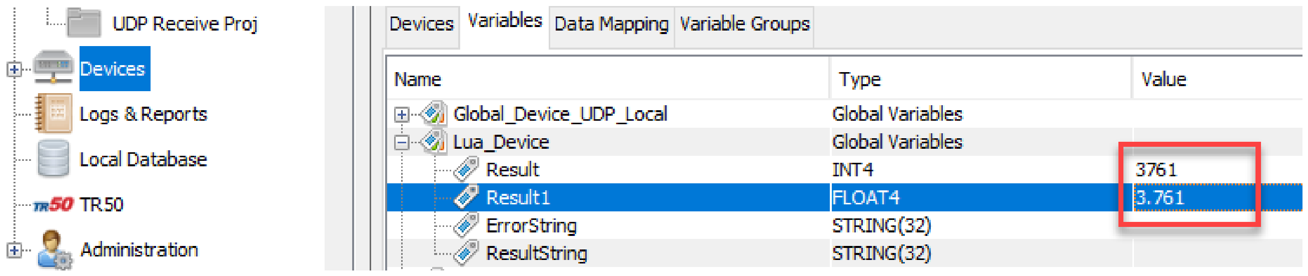
Input

Name	Logical	Count	Value	Type
HexString	STRING(32	1	0EB1	CONSTANT
scale	INT2	1	1000	CONSTANT

Output

Name	Logical	Count	Value	Type
Result	INT2	1	Lua_Device.Result	INT4
Result1	FLOAT4	1	Lua_Device.Result1	FLOAT4
ReturnMessage	STRING	1	Lua_Device.ResultString	STRING(32)
ErrorMessage	STRING	1	Lua_Device.ErrorString	STRING(32)

3. To view the result, click the **Device**, go to the **Variables** tab and click on the output variable to see the output



Name	Type	Value
Global_Device_UDP_Local	Global Variables	
Lua_Device	Global Variables	
Result	INT4	3761
Result1	FLOAT4	3.761
ErrorString	STRING(32)	
ResultString	STRING(32)	

About Telit | Contact Us | Legal Notices | Terms of Service | Privacy Policy