How to write custom actions triggered both by the local agent as well as a central point with the ability to use Nimsoft to manage and report on these actions which may include scripting.

nexec probe: (command execution probe, local system or remote)

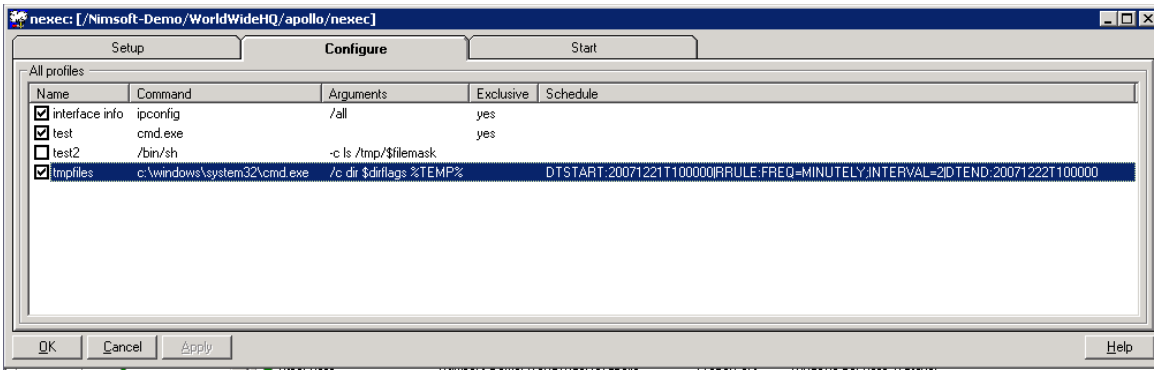
Description: run command anywhere -> on any machine and from any system

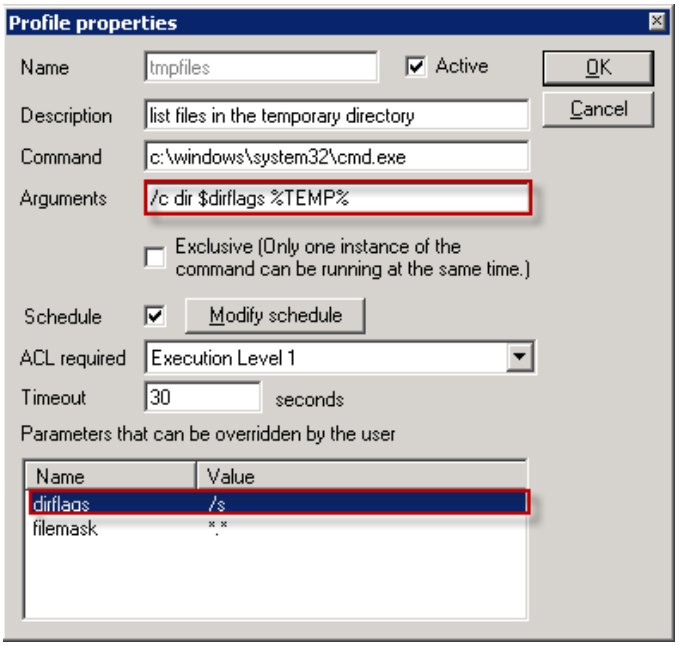
Features:

- Control what person or group is allowed to run the command
- Schedule a command to be run via nexec profile(s)
- Parameters: if you want to use a generic command, e.g., dir, you can set parameters to use in the command execution, for example:

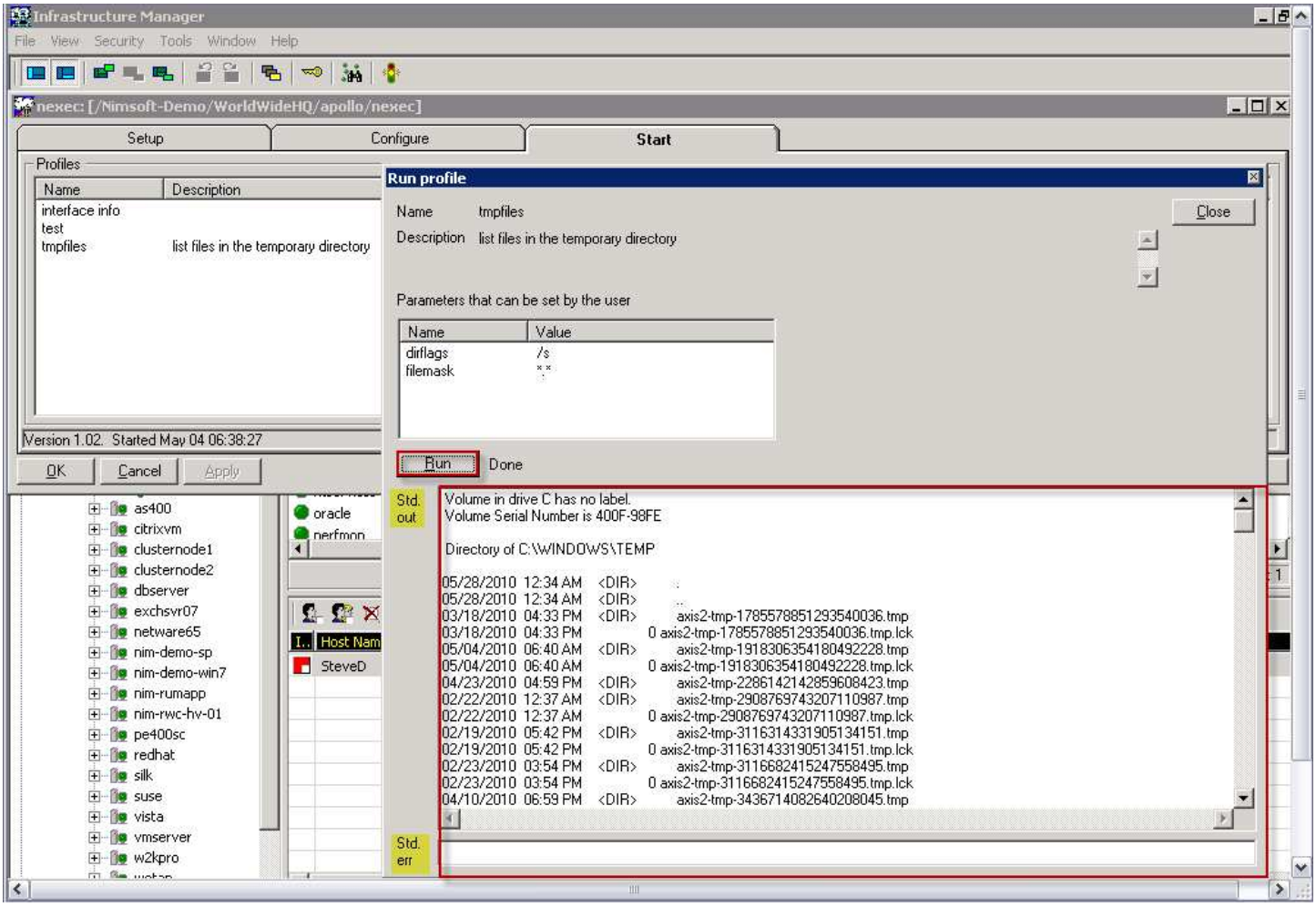
Parameter Name	Value
dirflags	/s

then call it in the command using $dirflag

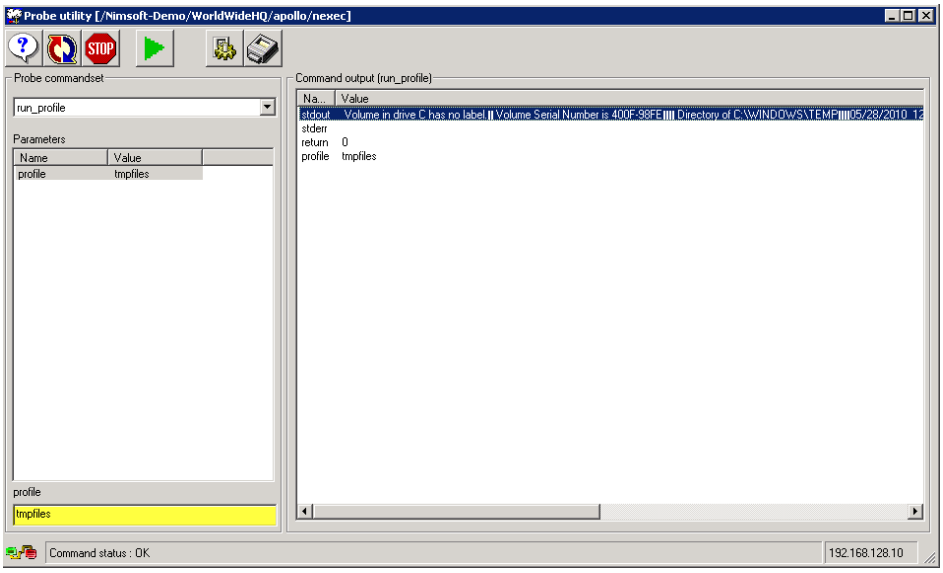




To test the command you can Select the Start Tab and then select the profile you created to execute the command, then rt-click it and Run it. You can then see the output as well as any errors if there's a problem, e.g., cant find a dir/file, bad command syntax, etc.



Note: You can also select the nexec probe and press Ctrl-P, to interact directly with the probe to make a request and list the profiles or run one of the profiles to see the output. Note that you have to enter the name of the profile you want to run.

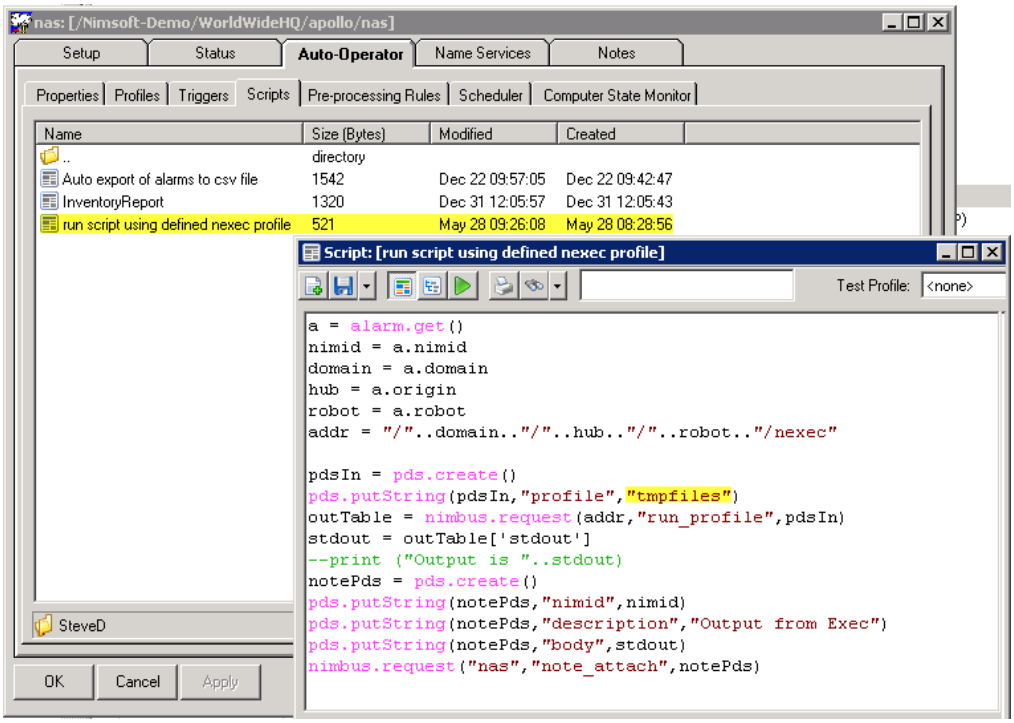


Useful feature: You can add the stdout/stderr to a note contained in an alarm.

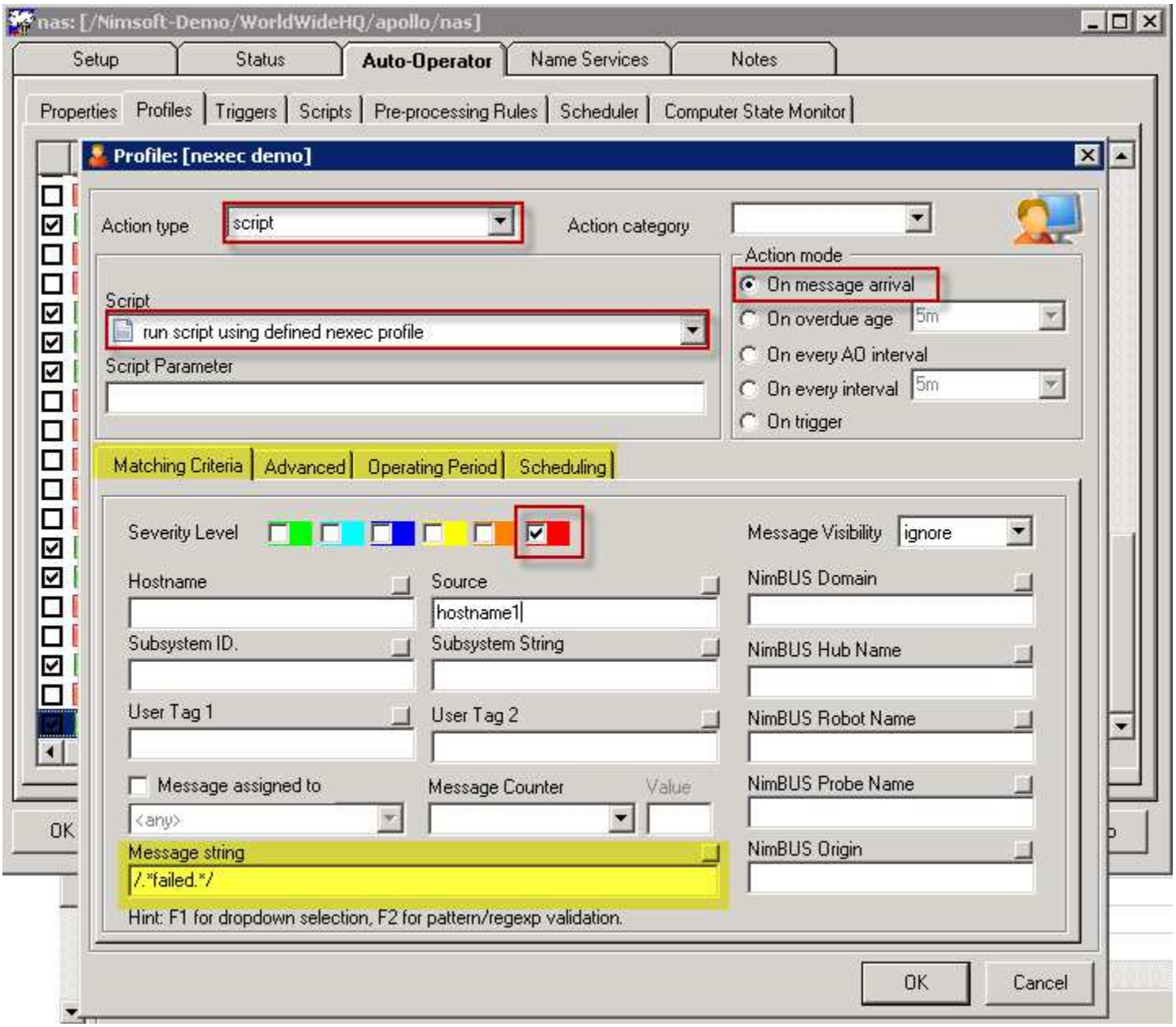
Automatic execution of actions, e.g., using commands/scripting

Nas probe (Nimsoft Alarm Server)

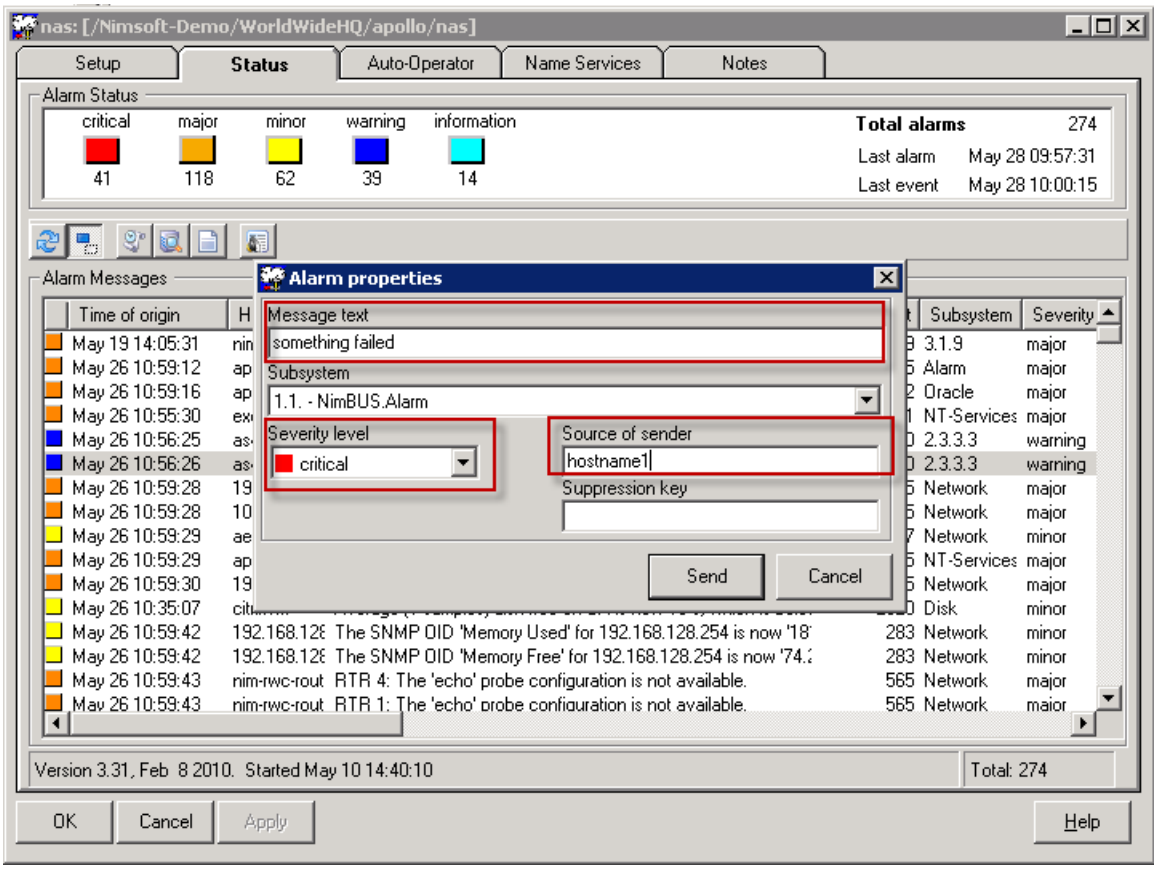
You have many options available to you to control when you run a LUA script to execute the command(s) by using the features of nas, e.g., based on operating period, schedule, excluding maintenance periods etc. First you create/add the script to your nas probe under scripts then under the Auto-Operator profiles…



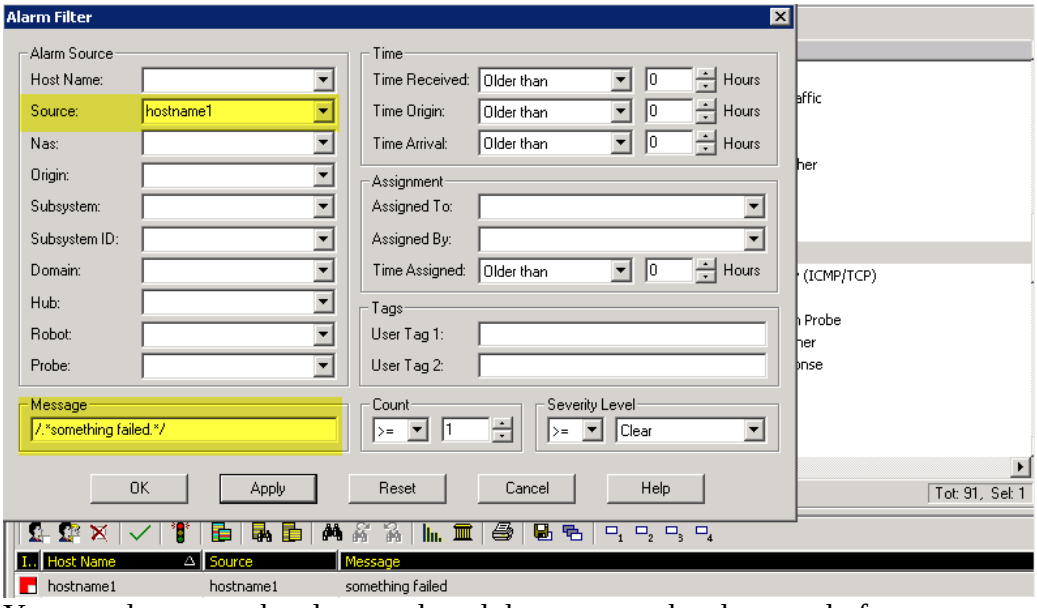
and then you can choose when you want to run the specific script that executes the custom action/command, e.g., when you receive a message with the message text of "something failed" and it's a **critical** alarm.



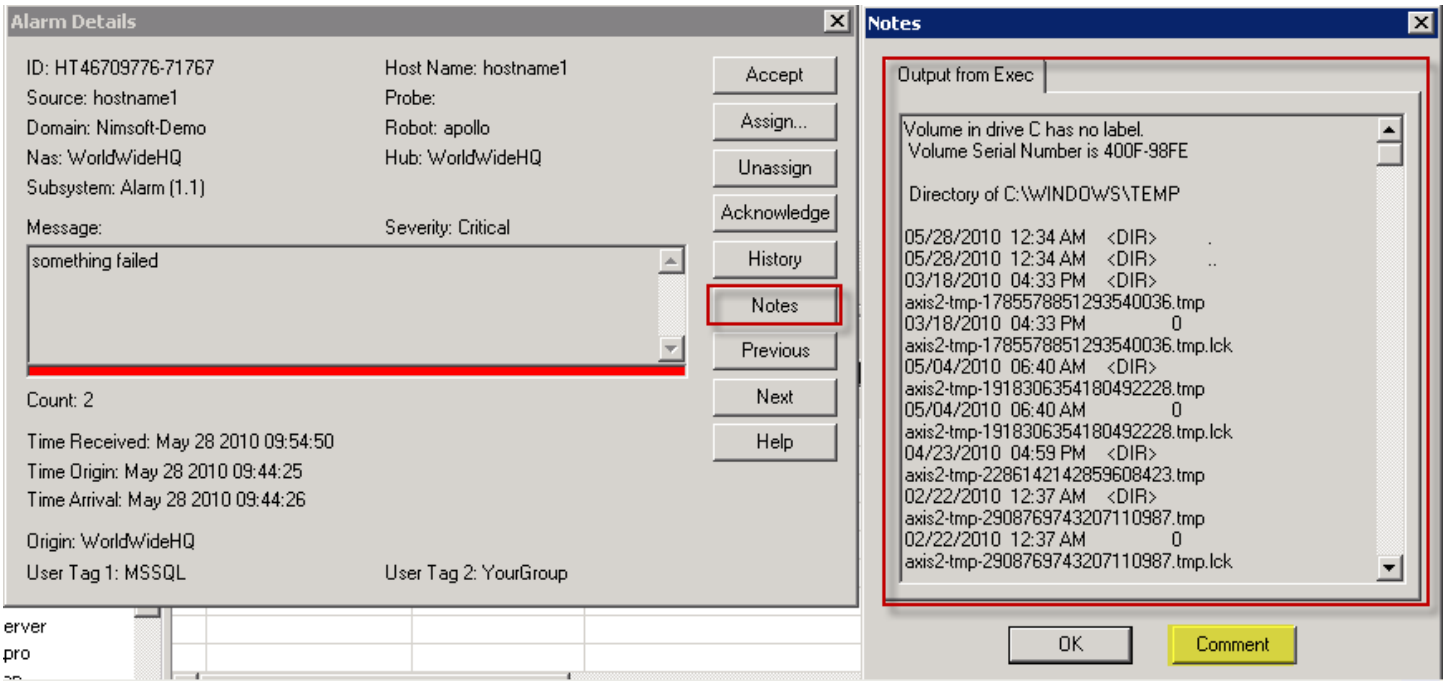
To test your Auto-Operator profile to see if it is successfully executing the action, you can go to the Status Tab in the nas and Send a Test Alarm via rt-click.



In the alarm sub-console in the Infrastructure Manager, you can use the alarm filtering feature to see if you got the test alarm:



You can then open the alarm and read the notes on the alarm and of course, operators can add their own comments to it.



Summary

- nexec probe can be deployed to the various robots with profiles configured to run the desired scripts/commands. These profiles can be executed via LUA script from the nas (central point) via nimbus.request() calls in response to specific alarms.

- If you want to "trigger" actions locally rather than from the nas using a script you can also use probes if they support actions – it depends on the probe. The only local 'agent' ones possible are through probes that provide that capability for example, logmon, dirscan, processes, can run commands and it does not require anything else.

- If these are a set of standard options then configuring them via nexec and then calling them via a callback from nas/LUA. It depends on what reporting you are looking for on the actions.

- If nexec probe is not an option, then possibly a custom on-demand probe will work that can run the required action.

We have created scripts/probes to run commands remotely (e.g. via ssh or pscript), but then authentication becomes the key – see rsp probe.

Reference:
Nexec demo script with comments

```
-- Get the alarm object
a = alarm.get()
-- Extract the NimID from the alarm object
nimid = a.nimid
-- Extract the address details from the alarm object
domain = a.domain
```

```
hub = a.origin
robot = a.robot
-- Construct the address as /domain/hub/robot/probe
addr = “/”..domain..”/”..hub..”/”..robot.."/nexec"

-- Create a PDS object to use to send details
pdsIn = pds.create()
-- Add a string key/value pair to the PDS to tell the probe which profile to run
pds.putString(pdsIn,"profile","interface info")
-- Run the “run_profile” callback of the nexec probe
outTable = nimbus.request(addr,"run_profile",pdsIn)
-- Take the “stdout” parameter from the returned table
stdout = outTable['stdout']
-- Create a new PDS to use to construct a note
notePds = pds.create()
-- Add key/value pairs to the PDS to define the NimID, Description and body
pds.putString(notePds,"nimid",nimid)
pds.putString(notePds,"description","Output from Exec")
pds.putString(notePds,"body",stdout)
-- Run the “note_attach” callback of NAS to attach a note to the NimID
nimbus.request("nas","note_attach",notePds)
```