How to use a .ps1 file to run FCAT

This document walks through the process of running FCAT from a windows OS (i.e. the BlueJay VM) using a powershell script (i.e. a .ps1 file) and specifically references the file "EXAMPLE_FCAT_commands.ps1" that can be found in the rem-tools folder on the shared drive.

1. Make a copy of the "EXAMPLE_FCAT_commands.ps1" file to your user profile and rename it to be specific to your project.
2. Update the variables in the upper part of the .ps1 script. The variables you will need to update if you are doing a regular REM project include (you may have to update other variables depending on your project):
    a. $script_dir (if you keep all your .ps1 files in the same folder, you'll only have to update this once)
    b. $project_code
    c. $aoi_file
    d. $tx_file
    e. $global_kcp
    f. $regen_kcp
    g. $num_simulations
    h. $ABP
3. Uncomment the lines needed to run the fvs microservice (FVS set-up). Ensure all lines for all other steps are commented out (in a .ps1 file use a "#" to comment out a line, commented out lines will not run). You only want fvs to run, once completed you will move on to rfvs and so on and so forth. For each FCAT microservice, in turn, you will uncomment:
    a. $host_ip (only for currawong or chickadee)
    b. $port
    c. $xxx_cmd (only the version without logging properly functions, 03/01/2024)
    d. echo $xxx_cmd
    e. clojure…….

```
##########################################################################
# Uncomment these commands as needed; they should not need to be edited. #
# For projects please run steps only on currawong and chickadee. Grouse  #
# and goose are for development only, only run something on these VMs if #
# asked to or if you have confirmed they are in a functioning state.     #
##########################################################################
##### 1. FCAT-fvs (FVS set-up) #####
## currawong host IP
$host_ip = "10.1.30.142"
## grouse host IP
#$host_ip = "10.1.30.120"
$port = 1337
## command with logging
#$fvs_cmd = -join(' {"""scriptArgs""": {"""cell-size""": 30, """project-code""": """', $pr
'""", """buffer""": ', $buffer_m, ', """include-dead-trees""": ', $include_dead_trees, ',
"""jobId""": 1, """responseHost""": """', $response_ip, '""", """responsePort""": ', $resp
## command without logging
$fvs_cmd = -join(' {"""scriptArgs""": {"""cell-size""": 30, """project-code""": """', $pro
'""", """buffer""": ', $buffer_m, ', """include-dead-trees""": ', $include_dead_trees, ',
"""jobId""": 1}')
echo $fvs_cmd
clojure -M:default-ssl-opts:run -h $host_ip -p $port $fvs_cmd
```

This ^^^ is what the script should look like when you are about to run the fvs microservice (FCAT step 1). You have uncommented only the lines needed to run fvs, all other lines for all other microservices are commented out.

4. Save the changes you have made to the .ps1 script.
5. Open a windows powershell and cd into the directory where your .ps1 file is stored, press 'enter'. Type ".\" followed by the full name of your .ps1 file, press 'enter' (this will look like ".\EXAMPLE_FCAT_commands.ps1"). After a few seconds, you'll see the JSON (the $xxx_cmd) echo'd back to you and a message that your job has been queued. This message means there were no errors in the command and it has been sent off to do its work.
6. Give FCAT time to run step 1, you can monitor the file production in the shared drive as a way to "status check" FCAT (see the document "Signs that each FCAT step has completed running.pdf" in the rem-tools folder). You do not want to start a microservice if the previous microservice has not completed running – each subsequent microservice is dependent on using outputs from previous microservices as inputs.
7. Once fvs is complete, go back to the .ps1 script. Comment out the lines for fvs and uncomment the lines for rfvs. *Note that you should not uncomment lines that are explanatory only. This is what it should look like when you are ready to run step 2:

```
##### 2. FCAT-rfvs (execute FVS) ####
## currawong host IP
$host_ip = "10.1.30.142"
## grouse host IP
#$host_ip = "10.1.30.120"
$port = 1338
## command with logging
#$rfvs_cmd = -join(' {"""scriptArgs""": {"""project-code""": """', $project_code, '"""
$exec_wildfires, ', """exec-project""": ', $exec_project, ', """exec-baseline""": ', $
', $response_port, '}')
## command without logging
$rfvs_cmd = -join(' {"""scriptArgs""": {"""project-code""": """', $project_code, '""",
$exec_wildfires, ', """exec-project""": ', $exec_project, ', """exec-baseline""": ', $
echo $rfvs_cmd
clojure -M:default-ssl-opts:run -h $host_ip -p $port $rfvs_cmd
```

8. Save your changes to the .ps1 script.
9. Return to the powershell, here you'll notice the .ps1 script has cd'd into the directory where your .ps1 script is saved so now all you have to type is "./name_of_your_ps1_file.ps1" followed by enter and the script will run (hint: use the 'up' arrow on your keyboard to avoid having to retype the file name). (Notice the "cd $script_dir" at the bottom of the .ps1 script – that line is what leaves the powershell cd'd into the folder where your .ps1 script lives.) You'll see the JSON for rfvs echo'd back to you and the job queued message again. *Note: the rfvs microservice (step 2) cannot correctly be run more than once without first deleting the fvs/runs folder. If you need to re-run rfvs, first delete the fvs/runs folder OR start a new version of the project.
10. Give FCAT time to run rfvs. Again, you can monitor the file production in the shared drive as a way to "status check" FCAT.
11. Once rfvs is complete, return to the .ps1 file and comment out the lines for rfvs and uncomment the lines for gridfire-1. Save your changes to the .ps1 script, return to the powershell, type the .ps1 file name like you did in step 9 and hit enter.
12. Repeat this process for the remaining FCAT microservices.