

What is it?

Remote Installer is a tool for automatic operating terminal in Linux Server via SSH protocol. It uses Groovy script language to process received data. By storing a key-value map, the tool checks received data line by line and whenever a line that includes a stored key in map, it will send the value based on the key.

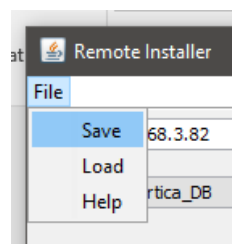
```
TreeMap<String, String> configCommand = new TreeMap<String, String>();
configCommand.put('Choose Locale','1');
configCommand.put("by typing 'quit'", '');
configCommand.put("PRESS <ENTER> TO CONTINUE", "");
configCommand.put("DO YOU ACCEPT THE TERMS OF THIS LICENSE AGREEMENT?", "y");
configCommand.put("ENTER THE NUMBER OF THE DESIRED CHOICE, OR PRESS <ENTER> TO ACCEPT THE", "1");
configCommand.put("ENTER AN ABSOLUTE PATH", extractedPath);
configCommand.put("INSTALL FOLDER IS", "y");
configCommand.put('the maximum memory that the DC', "");
configCommand.put('Disk Space Information', '');
configCommand.put('Data Aggregator Host', daIP);
configCommand.put('Installation Complete', '\necho $?');
```

How to use?

Entering your SSH host IP, username and password.

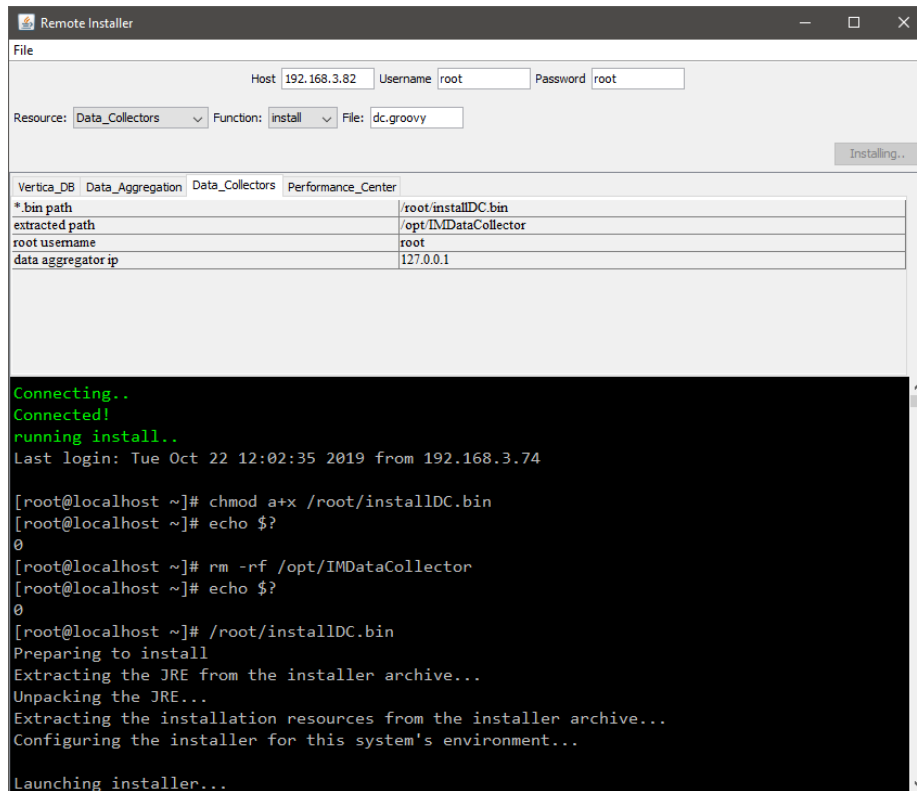
Selecting a name on The Resource List, it will show you the defined functions on The Function List, the Groovy script file name in File Text Box and the defined variables on The Variable Table below. You have to check all variables showed before starting. Modifying if any variable's values are not correct with your need (Ex: *.bin file is located in other path)

Option: You can save all of the information to a file and load it next time without setting your information again.

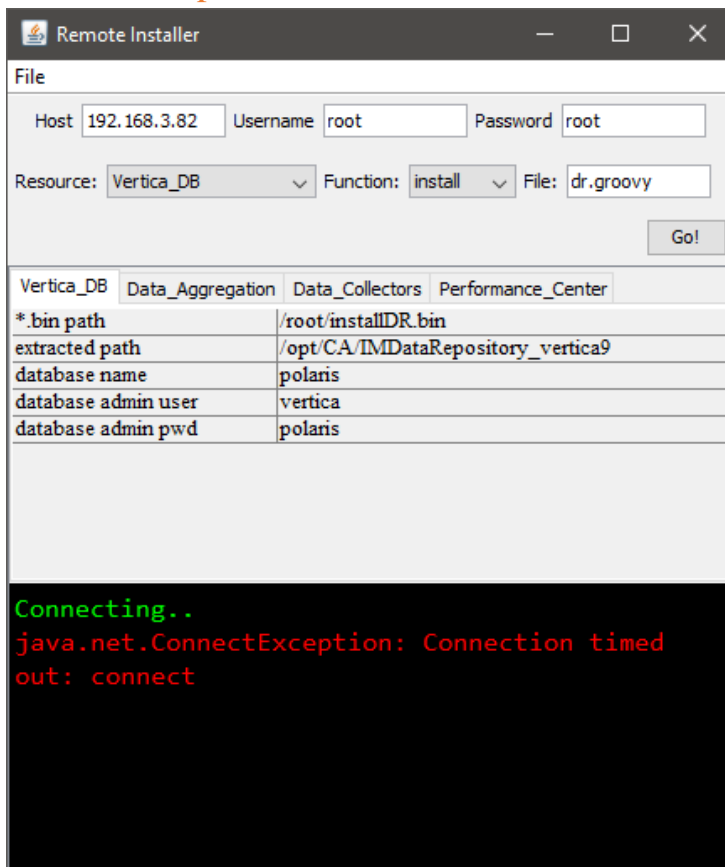


The common cases

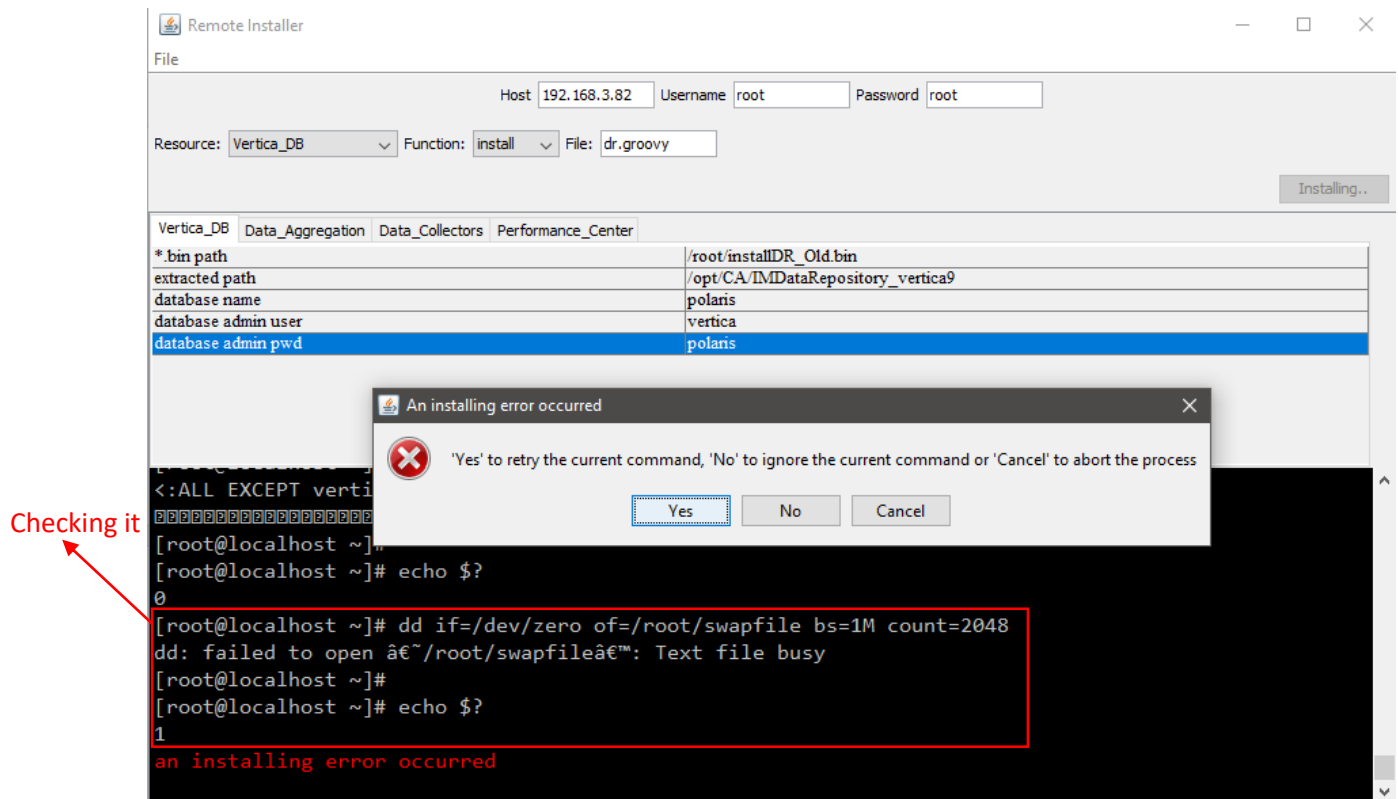
1. Running well



2. Connection problems



3. Terminal's error announcement



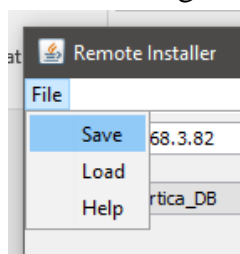
- Checking the announcement, then if the problem is important for the next activities, you have to find and fix it by fixing in the connected server or if problem is caused by variable's value, you can modify the wrong ones. Finally, pressing "Yes" to retry.
- If the problem is not important, pressing "No" to ignore it.
- If you need to stop running, you can press "Cancel"

Adding new resources

You can add new resources by altering a saved file and loading it again in next time using.

A tutorial example to add a new resource named "[new_resource](#)", following the steps below:

1. Save a configuration file



2. Open the file, now you see the format look like:

[section name]

key1 = value1

key2 = value2

3. Add some information of “[new_resource](#)” to the content

[Initiation]

Resources = Vertica_DB,Data_Aggregation,Data_Collectors,Performance_Center, [new_resource](#)

[Groovy]

Vertica_DB = dr.groovy

Data_Aggregation = da.groovy

Data_Collectors = dc.groovy

Performance_Center = pc.groovy

[new_resource](#) = [new_script](#).groovy

[Functions]

Vertica_DB = install,uninstall

Data_Aggregation = install,uninstall

Data_Collectors = install,uninstall

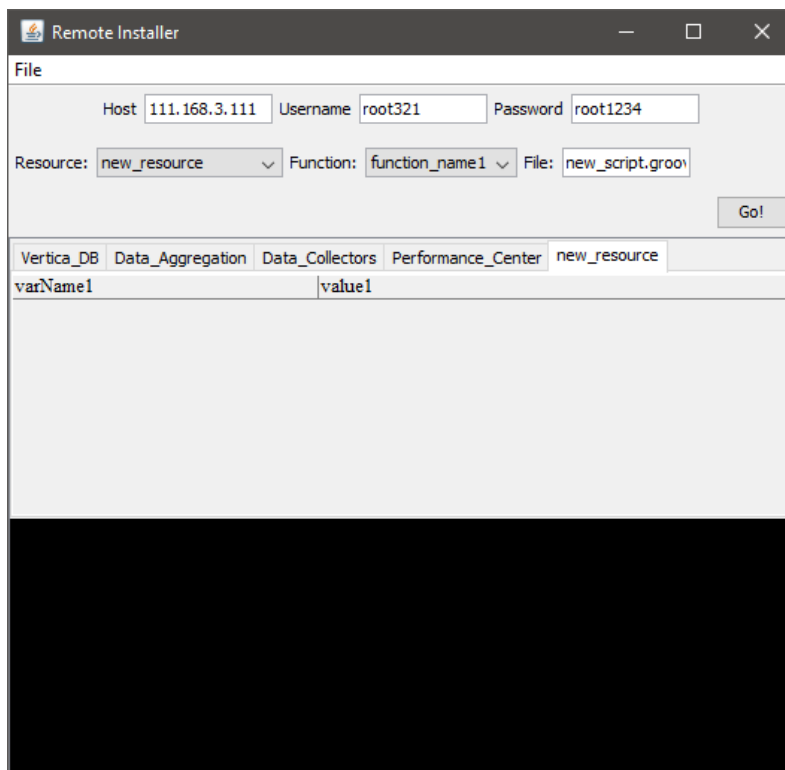
Performance_Center = install,uninstall

[new_resource](#) = [function_name1](#),[function_name2](#)

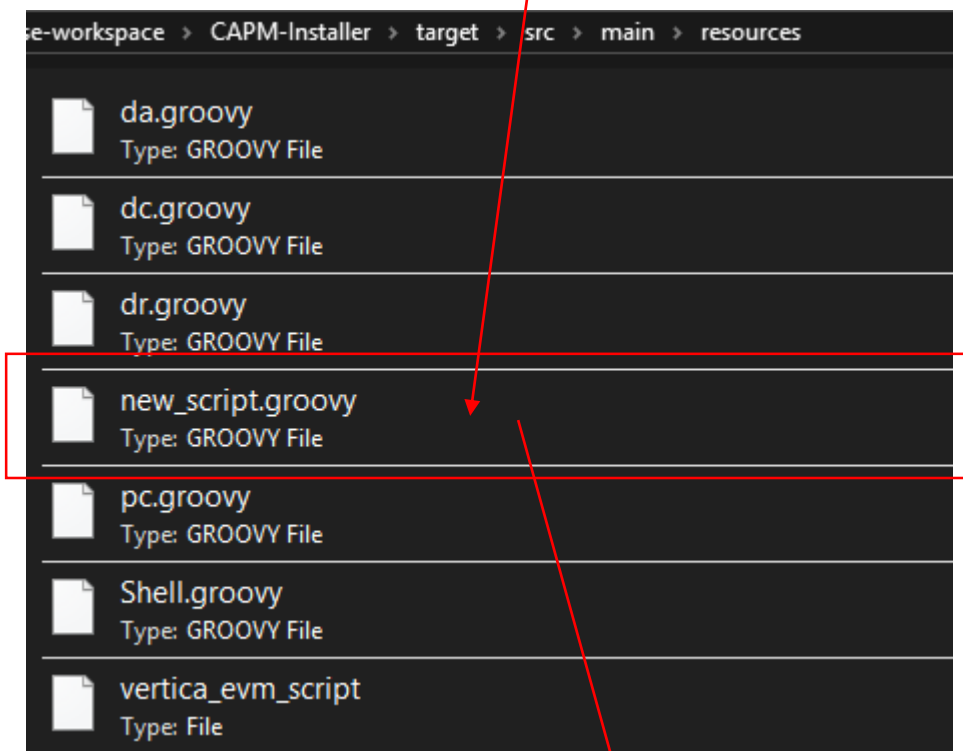
[[new_resource](#)_Variables]

[varName1](#) = value1

4. Load the modified file into the tool, now the User Interface is updated



5. Create and write a new Groovy script named `new_script.groovy` in “...\\src\\main\\resources” with the format like other groovy files in this path.



```
1 import capm.installer.TOOL.Utilities;
2
3
4 @InheritConstructors
5 class Groovy extends Shell{
6
7     //your code below!
8     String anyVarName = SharedResources.getResource('varName1');
9
10
11     def String function_name1() {
12         init();
13         if(!da_installation())
14             return response;
15         return "";
16     }
17
18     def String function_name2() {
19         init();
20         if(!da_uninstallation())
21             return response;
22         return "";
23     }
24
25     def boolean da_installation(){
26         String script = "chmod a+x $binPath\n"+
27             "rm -rf $extractedPath\n"+
28             "$binPath\n";
29
30
31 }
```