National Centre for Radio Astrophysics

Tata Institute of Fundamental Research,
Pune University Campus, Pune, INDIA

Test Report of Use Cases using GMRT Configurator Tool

Date : 24- 01 -2020

Bhavesh Kunbi, Jitendra Kodilkar, Amol Chavan, Raju Uprade
GMRT – Khodad
Email :bhavesh,jitendra, rajsingh @gmrt.ncra.tifr.res.in

Revision	Date	Document Created by	Document Reviewed by	Modification/ Change
Ver. 1.0	10-01-2020	Bhavesh Kunbi Amol Chavan	Raj Uprade	
Ver. 1.1	13-01-2020	Bhavesh Kunbi Amol Chavan	Raj Uprade	Test report updated with new Configurator tool test results (**GCTBetaVersion**)
Ver. 1.2	17-01-2020	Bhavesh Kunbi Amol Chavan	Raj Uprade	Test report updated with new Configurator tool test results (**GCTBetaVersion_Config**)
GCTBetaVersion_Config	21-01-2020	Bhavesh Kunbi Amol Chavan	Raj Uprade	

INDEX

1. Background:

A Next Generation M & C System developed and released for test observation as part of the ongoing upgrade of GMRT, which aims to provide end-to-end radio telescope software solution, ranging from auto execution of scheduled observing proposal to meta-data generation for supporting science data analysis. It was developed in synergy with the SKA-TM work package, using very similar design ideas and technology choices. The GMRT Software Architecture description maps the functional modules of the SKA TM design (such as OBSMGT, TELMGT, TM.LMC) to corresponding modules of the GMRT M&C System. The M&C system has been developed using TANGO software framework to control and monitor GMRT antennas and associated systems.

The Next generation M&C system is called as Tango based GMRT Control System (TGC), which is a configuration based control system. All configuration is stored in MySQL database. Whenever we need to add new commands or any new Sub-system, We need to change the database which can result in inconsistencies and very prone to human errors. Hence need for a Configurator tool was felt, Which can generate the required configuration without having to directly change the database as well as maintain the integrity.

Configurator tool also has Git version control embed in itself which can track the various versions of the specification files.

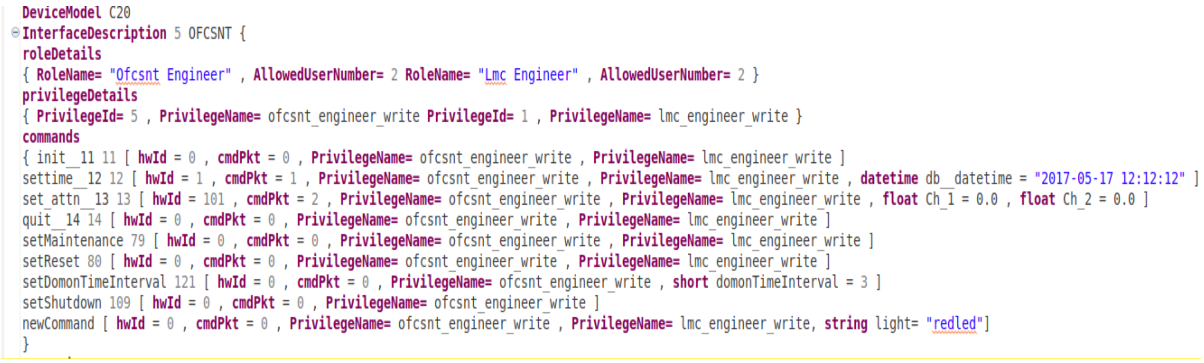
2. Configurator Tool different Use Cases test Scenarios :

With Version: GCTBetaVersion_Config (17 Jan 2020), GCTBetaVersion (13 Jan 2020) ,GCTBetaVersion_1 (3 Jan 2020) and GCTBetaVersion_2 (old)

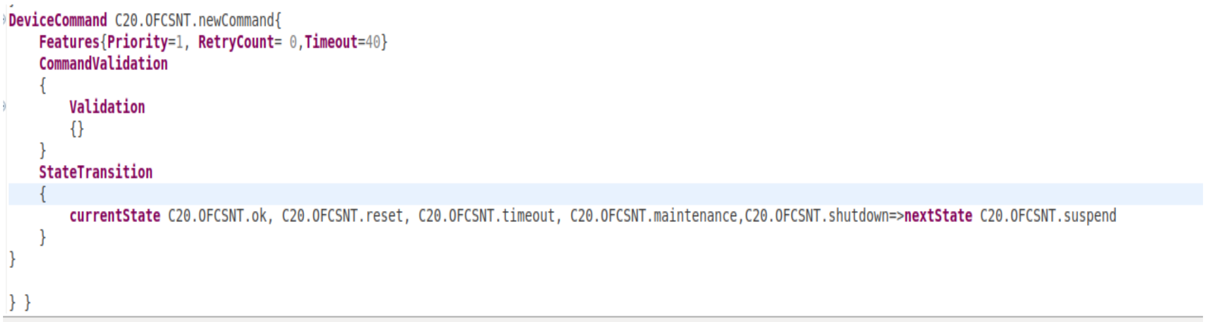
Sr. No	Test Scenarios	Testing Results (GCTBetaVersion_2)	Testing Results (GCTBetaVersion_1)	Testing Results (GCTBetaVersion)	*Testing Results (GCTBetaVersion_Config)	Remarks	MySQL Err. Code
1	Adding new command without argument. Command does not exist in command table. (Add/Delete)	OK	OK	OK	OK		
2	Addition of New Role and existing Privilege.	OK	OK	OK	OK		
3	Addition of existing Role and new Privilege.	OK	OK	OK	OK		1062-solved
4	Addition of new Role and new Privilege.	OK	OK	OK	OK		1062-solved
5	Addition of both existing Role and Privilege.	OK	OK	OK	OK		1452-solved
6	Addition/ Deletion of new subsystem, create a new file with '.mpogo' extension by right clicking on the project.	NOT OK	OK	OK	OK	→ In Generated Queries file semicolon (;) is missing. (with GCTBetaVersion_2 tool) → In GCTBetaVersion_1 it's working fine.	
7	Adding new command with Updation of Max/Min values in validation.	OK	OK	OK	OK		
8	Replication of LMC	OK	OK	OK	OK		

Note: In GCTBetaVersion_2 tool, query was generated without semicolon (;) while adding new subsystem. It was informed to TCS team, they solved the bug and provided updated version of Configurator tool. Problem was solved in GCTBetaVersion_1 (Updated Version of tool).

2. 1 . Adding new command without argument. Command does not exist in command table.

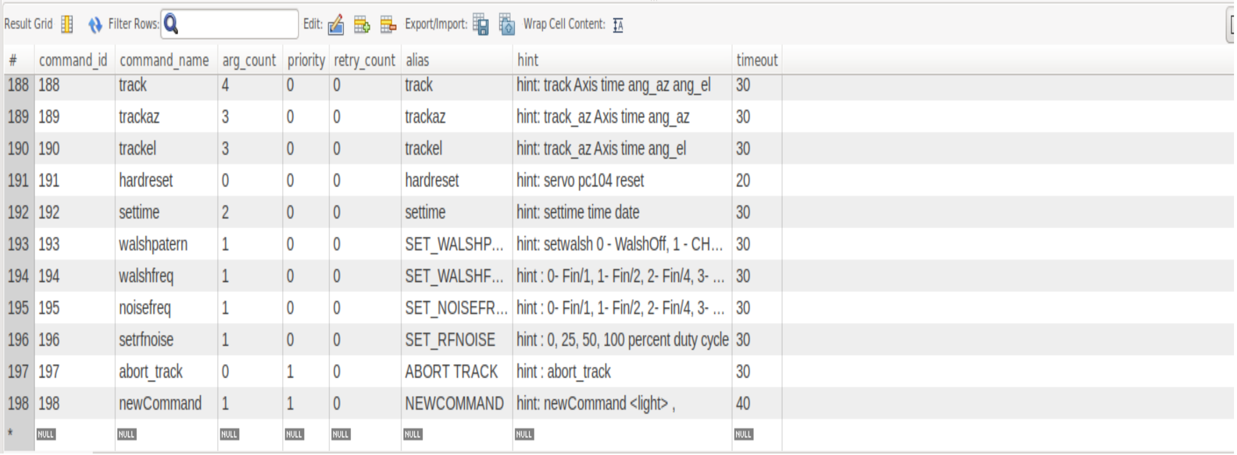


```
DeviceModel C20
InterfaceDescription 5 OFCSNT {
roleDetails
{ RoleName= "Ofcsnt Engineer" , AllowedUserNumber= 2 RoleName= "Lmc Engineer" , AllowedUserNumber= 2 }
privilegeDetails
{ PrivilegeId= 5 , PrivilegeName= ofcsnt_engineer_write PrivilegeId= 1 , PrivilegeName= lmc_engineer_write }
commands
{ init__11 11 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
settime__12 12 [ hwId = 1 , cmdPkt = 1 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write , datetime db__datetime = "2017-05-17 12:12:12" ]
set_attn__13 13 [ hwId = 101 , cmdPkt = 2 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write , float Ch_1 = 0.0 , float Ch_2 = 0.0 ]
quit__14 14 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
setMaintenance 79 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
setReset 80 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
setDomonTimeInterval 121 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , short domonTimeInterval = 3 ]
setShutdown 109 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write ]
newCommand [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write, string light= "redled"]
}
```

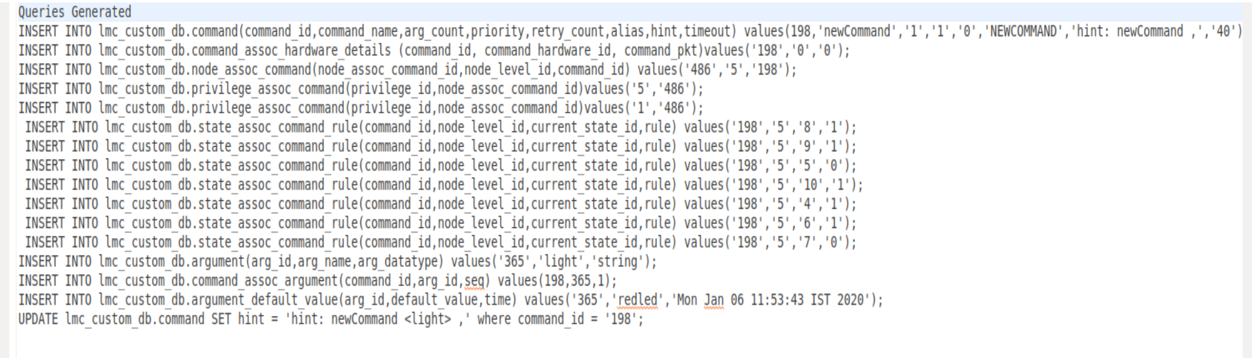


```
DeviceCommand C20.OFCSNT.newCommand{
    Features{Priority=1, RetryCount= 0,Timeout=40}
    CommandValidation
    {
        Validation
        {}
    }
    StateTransition
    {
        currentState C20.OFCSNT.ok, C20.OFCSNT.reset, C20.OFCSNT.timeout, C20.OFCSNT.maintenance,C20.OFCSNT.shutdown=>nextState C20.OFCSNT.suspend
    }
}

} }
```

Command No 198 is added in command table (MySQL workbench)

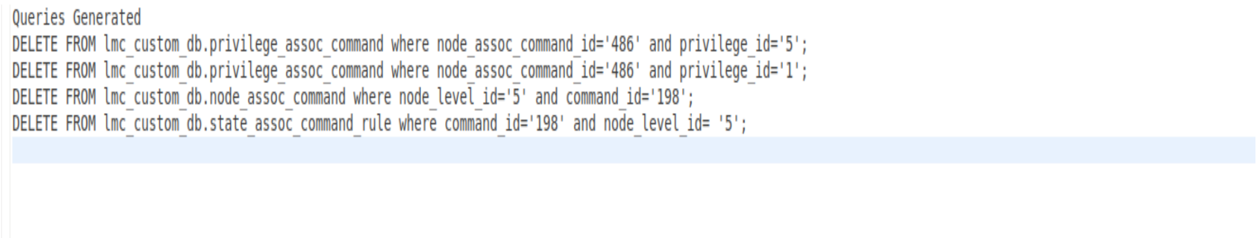


#	command_id	command_name	arg_count	priority	retry_count	alias	hint	timeout
188	188	track	4	0	0	track	hint: track Axis time ang_az ang_el	30
189	189	trackaz	3	0	0	trackaz	hint: track_az Axis time ang_az	30
190	190	trackel	3	0	0	trackel	hint: track_az Axis time ang_el	30
191	191	hardreset	0	0	0	hardreset	hint: servo pc104 reset	20
192	192	settime	2	0	0	settime	hint: settime time date	30
193	193	walshpatern	1	0	0	SET_WALSHP...	hint: setwalsh 0 - WalshOff, 1 - CH...	30
194	194	walshfreq	1	0	0	SET_WALSHF...	hint : 0- Fin/1, 1- Fin/2, 2- Fin/4, 3- ...	30
195	195	noisefreq	1	0	0	SET_NOISEFR...	hint : 0- Fin/1, 1- Fin/2, 2- Fin/4, 3- ...	30
196	196	setrfnoise	1	0	0	SET_RFNOISE	hint : 0, 25, 50, 100 percent duty cycle	30
197	197	abort_track	0	1	0	ABORT TRACK	hint : abort_track	30
198	198	newCommand	1	1	0	NEWCOMMAND	hint: newCommand <light> ,	40
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



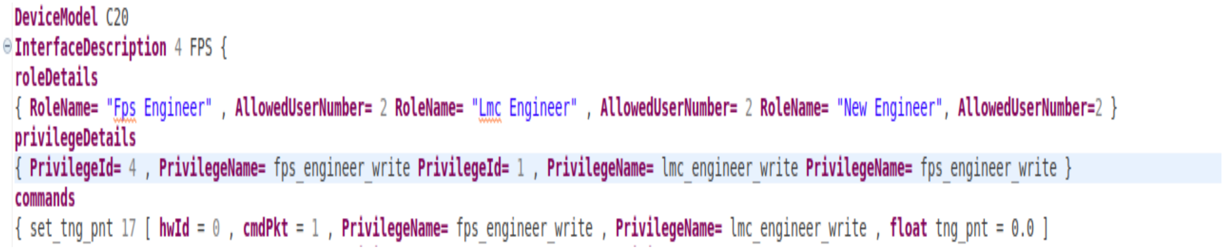
```
Queries Generated
INSERT INTO lmc_custom_db.command(command_id,command_name,arg_count,priority,retry_count,alias,hint,timeout) values(198,'newCommand','1','1','0','NEWCOMMAND','hint: newCommand ,','40')
INSERT INTO lmc_custom_db.command_assoc_hardware_details (command_id, command_hardware_id, command_pkt)values('198','0','0');
INSERT INTO lmc_custom_db.node_assoc_command(node_assoc_command_id,node_level_id,command_id) values('486','5','198');
INSERT INTO lmc_custom_db.privilege_assoc_command(privilege_id,node_assoc_command_id)values('5','486');
INSERT INTO lmc_custom_db.privilege_assoc_command(privilege_id,node_assoc_command_id)values('1','486');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('198','5','8','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('198','5','9','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('198','5','5','0');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('198','5','10','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('198','5','4','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('198','5','6','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('198','5','7','0');
INSERT INTO lmc_custom_db.argument(arg_id,arg_name,arg_datatype) values('365','light','string');
INSERT INTO lmc_custom_db.command_assoc_argument(command_id,arg_id,seq) values(198,365,1);
INSERT INTO lmc_custom_db.argument_default_value(arg_id,default_value,time) values('365','redled','Mon Jan 06 11:53:43 IST 2020');
UPDATE lmc_custom_db.command SET hint = 'hint: newCommand <light> ,' where command_id = '198';
```

2.2 After deleting new Command:

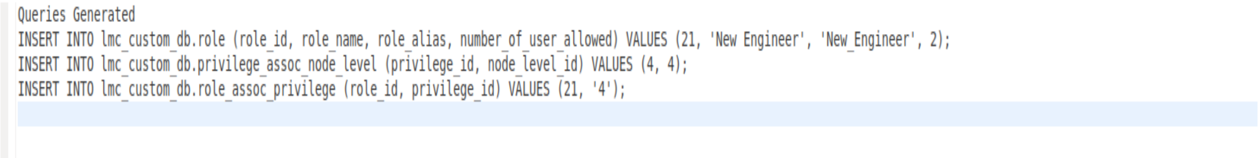


```
Queries Generated
DELETE FROM lmc_custom_db.privilege_assoc_command where node_assoc_command_id='486' and privilege_id='5';
DELETE FROM lmc_custom_db.privilege_assoc_command where node_assoc_command_id='486' and privilege_id='1';
DELETE FROM lmc_custom_db.node_assoc_command where node_level_id='5' and command_id='198';
DELETE FROM lmc_custom_db.state_assoc_command_rule where command_id='198' and node_level_id= '5';
```

2.3 . Addition of New Role and existing Privilege.

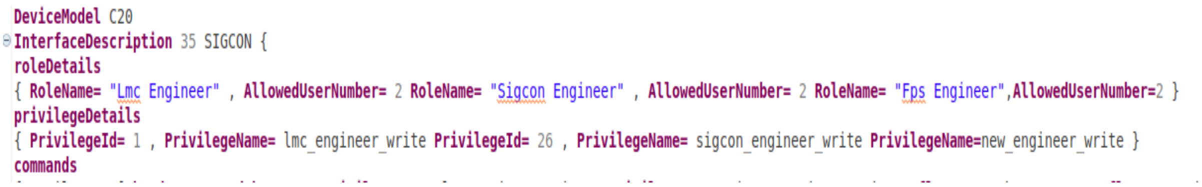


```
DeviceModel C20
InterfaceDescription 4 FPS {
roleDetails
{ RoleName= "Fps Engineer" , AllowedUserNumber= 2 RoleName= "Lmc Engineer" , AllowedUserNumber= 2 RoleName= "New Engineer", AllowedUserNumber=2 }
privilegeDetails
{ PrivilegeId= 4 , PrivilegeName= fps_engineer_write PrivilegeId= 1 , PrivilegeName= lmc_engineer_write PrivilegeName= fps_engineer_write }
commands
{ set_tng_pnt 17 [ hwId = 0 , cmdPkt = 1 , PrivilegeName= fps_engineer_write , PrivilegeName= lmc_engineer_write , float tng_pnt = 0.0 ]
```

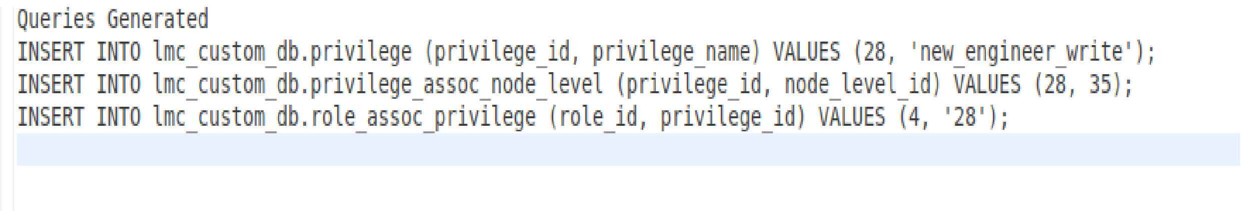


```
Queries Generated
INSERT INTO lmc_custom_db.role (role_id, role_name, role_alias, number_of_user_allowed) VALUES (21, 'New Engineer', 'New_Engineer', 2);
INSERT INTO lmc_custom_db.privilege_assoc_node_level (privilege_id, node_level_id) VALUES (4, 4);
INSERT INTO lmc_custom_db.role_assoc_privilege (role_id, privilege_id) VALUES (21, '4');
```

2. 4 . Addition of existing Role and new Privilege.

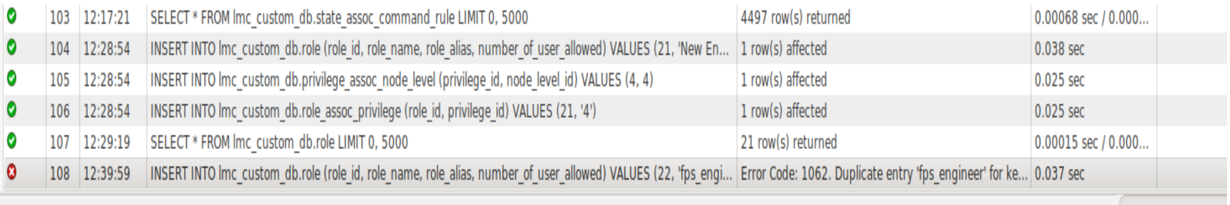


```
DeviceModel C20
InterfaceDescription 35 SIGCON {
roleDetails
{ RoleName= "Lmc Engineer" , AllowedUserNumber= 2 RoleName= "Sigcon Engineer" , AllowedUserNumber= 2 RoleName= "Fps Engineer",AllowedUserNumber=2 }
privilegeDetails
{ PrivilegeId= 1 , PrivilegeName= lmc_engineer_write PrivilegeId= 26 , PrivilegeName= sigcon_engineer_write PrivilegeName=new_engineer_write }
commands
```



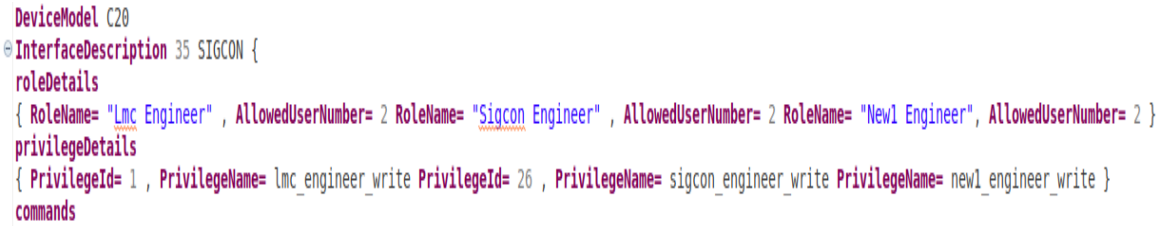
```
Queries Generated
INSERT INTO lmc_custom_db.privilege (privilege_id, privilege_name) VALUES (28, 'new_engineer_write');
INSERT INTO lmc_custom_db.privilege_assoc_node_level (privilege_id, node_level_id) VALUES (28, 35);
INSERT INTO lmc_custom_db.role_assoc_privilege (role_id, privilege_id) VALUES (4, '28');
```

Error in MySQL workbench (108) – Solved

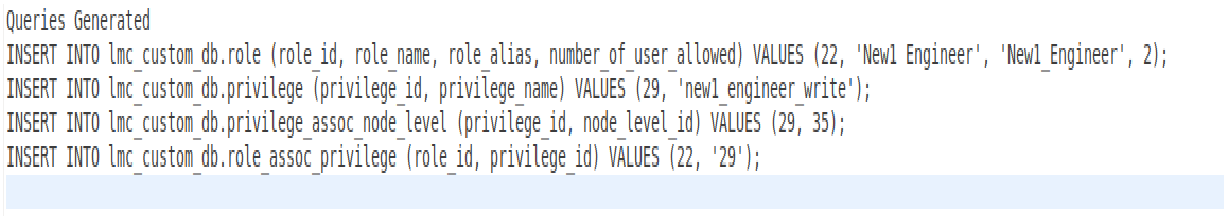


✓	103	12:17:21	SELECT * FROM lmc_custom_db.state_assoc_command_rule LIMIT 0, 5000	4497 row(s) returned	0.00068 sec / 0.000...
✓	104	12:28:54	INSERT INTO lmc_custom_db.role (role_id, role_name, role_alias, number_of_user_allowed) VALUES (21, 'New En...	1 row(s) affected	0.038 sec
✓	105	12:28:54	INSERT INTO lmc_custom_db.privilege_assoc_node_level (privilege_id, node_level_id) VALUES (4, 4)	1 row(s) affected	0.025 sec
✓	106	12:28:54	INSERT INTO lmc_custom_db.role_assoc_privilege (role_id, privilege_id) VALUES (21, '4')	1 row(s) affected	0.025 sec
✓	107	12:29:19	SELECT * FROM lmc_custom_db.role LIMIT 0, 5000	21 row(s) returned	0.00015 sec / 0.000...
☒	108	12:39:59	INSERT INTO lmc_custom_db.role (role_id, role_name, role_alias, number_of_user_allowed) VALUES (22, 'fps_engi...	Error Code: 1062. Duplicate entry 'fps_engineer' for ke...	0.037 sec

2.5 . Addition of new Role and new Privilege

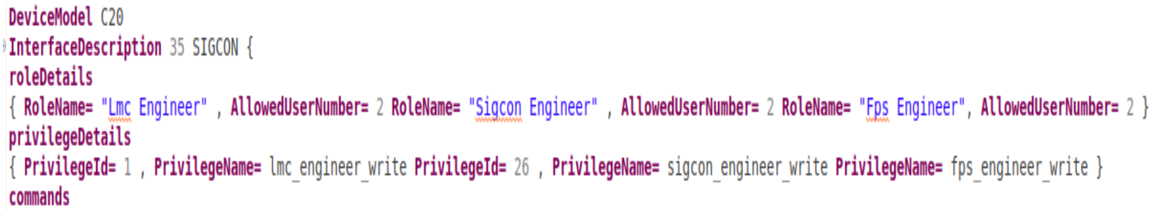


```
DeviceModel C20
InterfaceDescription 35 SIGCON {
roleDetails
{ RoleName= "Lmc Engineer" , AllowedUserNumber= 2 RoleName= "Sigcon Engineer" , AllowedUserNumber= 2 RoleName= "New1 Engineer", AllowedUserNumber= 2 }
privilegeDetails
{ PrivilegeId= 1 , PrivilegeName= lmc_engineer_write PrivilegeId= 26 , PrivilegeName= sigcon_engineer_write PrivilegeName= new1_engineer_write }
commands
```

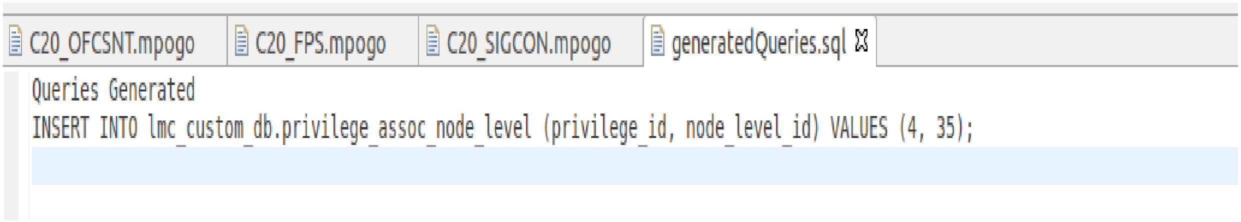


```
Queries Generated
INSERT INTO lmc_custom_db.role (role_id, role_name, role_alias, number_of_user_allowed) VALUES (22, 'New1 Engineer', 'New1_Engineer', 2);
INSERT INTO lmc_custom_db.privilege (privilege_id, privilege_name) VALUES (29, 'new1_engineer_write');
INSERT INTO lmc_custom_db.privilege_assoc_node_level (privilege_id, node_level_id) VALUES (29, 35);
INSERT INTO lmc_custom_db.role_assoc_privilege (role_id, privilege_id) VALUES (22, '29');
```

2.6 . Addition of both existing Role and Privilege.

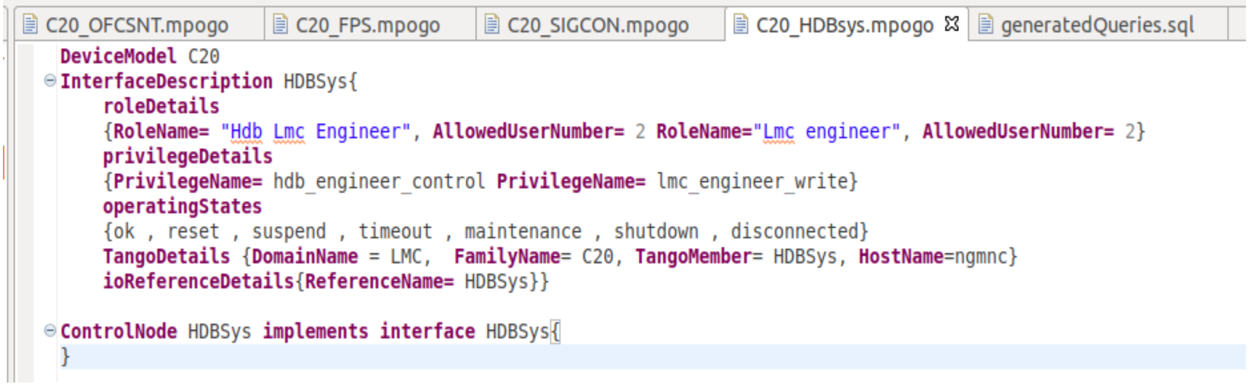


```
DeviceModel C20
InterfaceDescription 35 SIGCON {
roleDetails
{ RoleName= "Lmc Engineer" , AllowedUserNumber= 2 RoleName= "Sigcon Engineer" , AllowedUserNumber= 2 RoleName= "Fps Engineer", AllowedUserNumber= 2 }
privilegeDetails
{ PrivilegeId= 1 , PrivilegeName= lmc_engineer_write PrivilegeId= 26 , PrivilegeName= sigcon_engineer_write PrivilegeName= fps_engineer_write }
commands
```

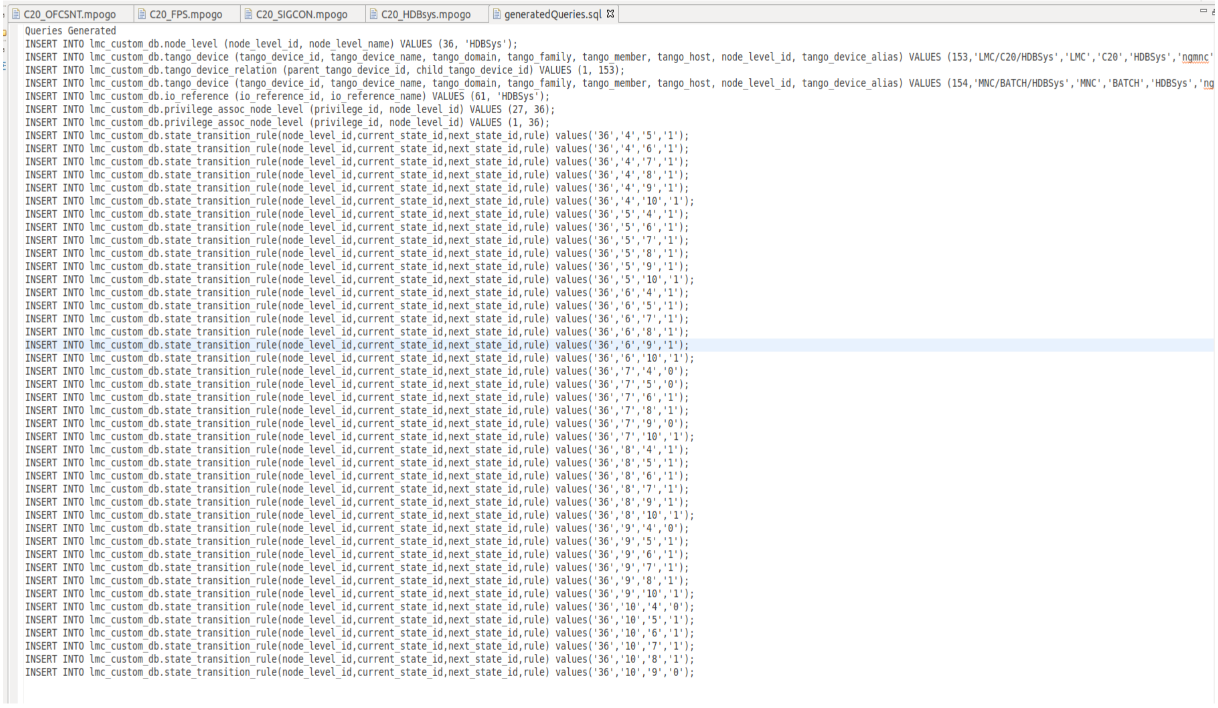


```
Queries Generated
INSERT INTO lmc_custom_db.privilege_assoc_node_level (privilege_id, node_level_id) VALUES (4, 35);
```

2.7 . Addition of A New subsystem



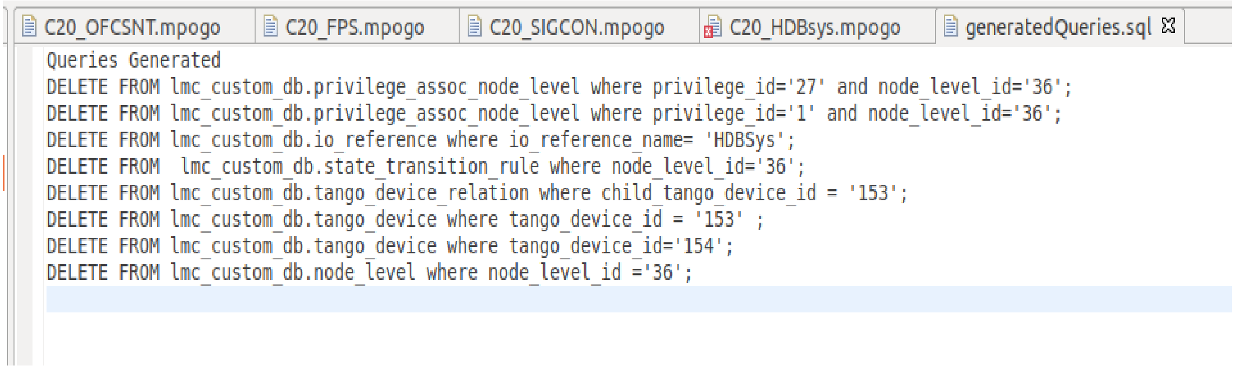
```
DeviceModel C20
InterfaceDescription HDBSys{
    roleDetails
    {RoleName= "Hdb Lmc Engineer", AllowedUserNumber= 2 RoleName="Lmc engineer", AllowedUserNumber= 2}
    privilegeDetails
    {PrivilegeName= hdb_engineer_control PrivilegeName= lmc_engineer_write}
    operatingStates
    {ok , reset , suspend , timeout , maintenance , shutdown , disconnected}
    TangoDetails {DomainName = LMC,  FamilyName= C20, TangoMember= HDBSys, HostName=ngmnc}
    ioReferenceDetails{ReferenceName= HDBSys}}

ControlNode HDBSys implements interface HDBSys{
}
```



```
Queries Generated
INSERT INTO lmc_custom_db.node_level (node_level_id, node_level_name) VALUES (36, 'HDBSys');
INSERT INTO lmc_custom_db.tango_device (tango_device_id, tango_device_name, tango_domain, tango_family, tango_member, tango_host, node_level_id, tango_device_alias) VALUES (153,'LMC/C20/HDBSys','LMC','C20','HDBSys','ngmnc'
INSERT INTO lmc_custom_db.tango_device_relation (parent_tango_device_id, child_tango_device_id) VALUES (1, 153);
INSERT INTO lmc_custom_db.tango_device (tango_device_id, tango_device_name, tango_domain, tango_family, tango_member, tango_host, node_level_id, tango_device_alias) VALUES (154,'MNC/BATCH/HDBSys','MNC','BATCH','HDBSys','ng
INSERT INTO lmc_custom_db.io_reference (io_reference_id, io_reference_name) VALUES (61, 'HDBSys');
INSERT INTO lmc_custom_db.privilege_assoc_node_level (privilege_id, node_level_id) VALUES (27, 36);
INSERT INTO lmc_custom_db.privilege_assoc_node_level (privilege_id, node_level_id) VALUES (1, 36);
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','4','5','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','4','6','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','4','7','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','4','8','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','4','9','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','4','10','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','5','4','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','5','6','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','5','7','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','5','8','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','5','9','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','5','10','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','6','4','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','6','5','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','6','7','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','6','8','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','6','9','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','6','10','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','7','4','0');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','7','5','0');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','7','6','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','7','8','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','7','9','0');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','7','10','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','8','4','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','8','5','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','8','6','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','8','7','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','8','9','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','8','10','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','9','4','0');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','9','5','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','9','6','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','9','7','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','9','8','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','9','10','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','10','4','0');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','10','5','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','10','6','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','10','7','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','10','8','1');
INSERT INTO lmc_custom_db.state_transition_rule(node_level_id,current_state_id,next_state_id,rule) values('36','10','9','0');
```

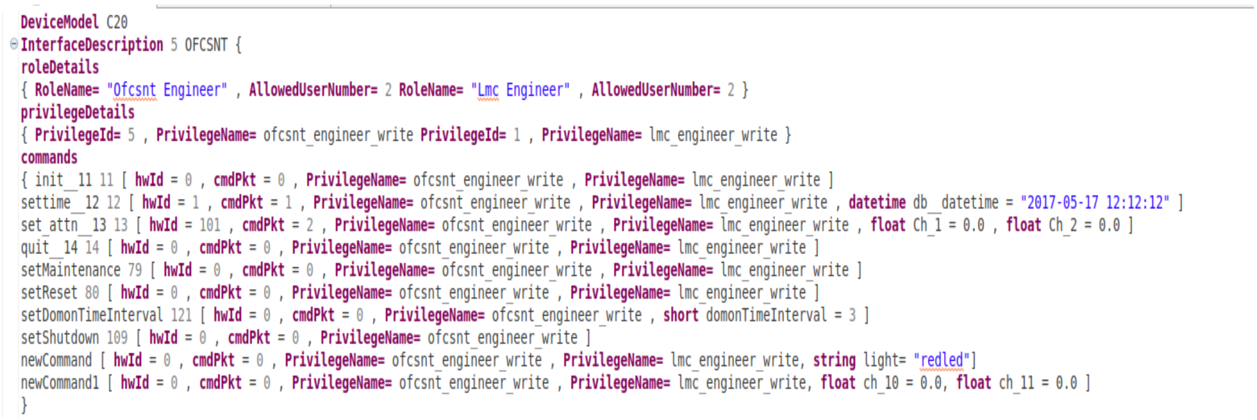
7.

2.8 After deleting new subsystem:

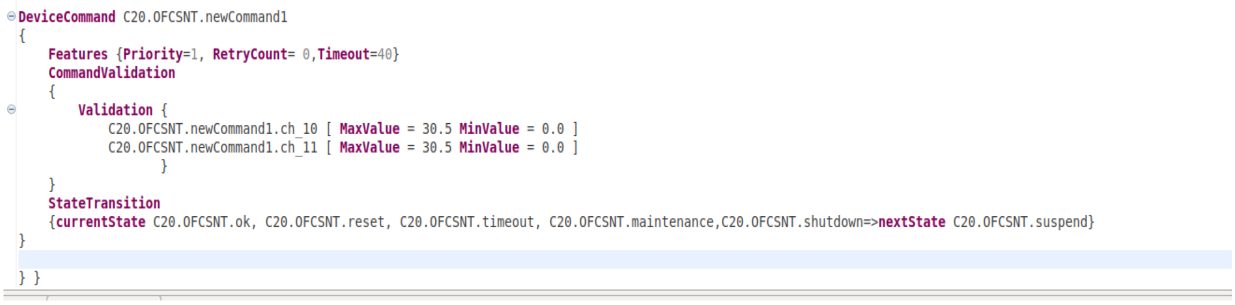


```
Queries Generated
DELETE FROM lmc_custom_db.privilege_assoc_node_level where privilege_id='27' and node_level_id='36';
DELETE FROM lmc_custom_db.privilege_assoc_node_level where privilege_id='1' and node_level_id='36';
DELETE FROM lmc_custom_db.io_reference where io_reference_name= 'HDBSys';
DELETE FROM  lmc_custom_db.state_transition_rule where node_level_id='36';
DELETE FROM lmc_custom_db.tango_device_relation where child_tango_device_id = '153';
DELETE FROM lmc_custom_db.tango_device where tango_device_id = '153' ;
DELETE FROM lmc_custom_db.tango_device where tango_device_id='154';
DELETE FROM lmc_custom_db.node_level where node_level_id ='36';
```

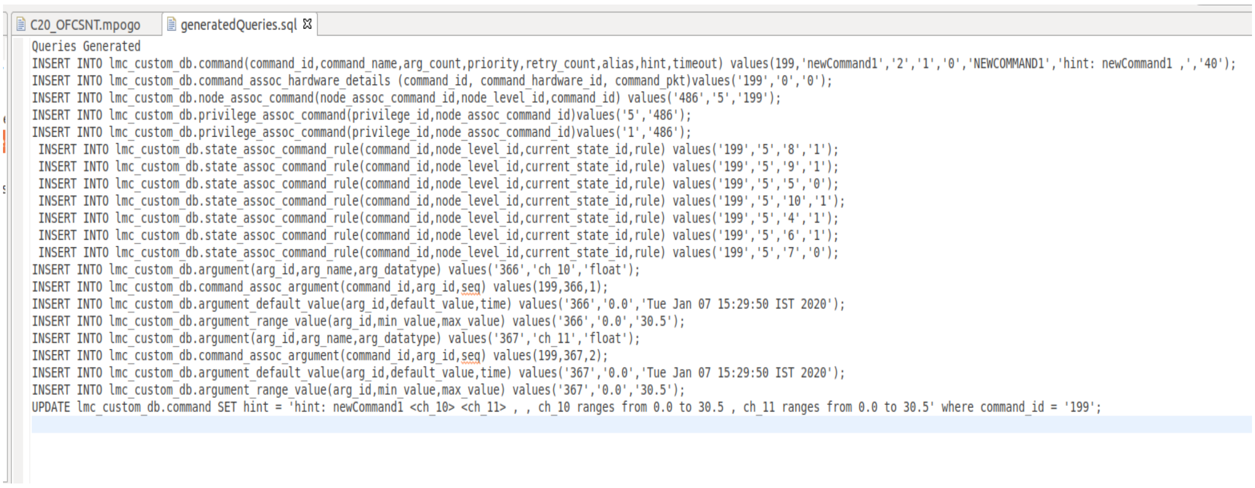
2.9 . Adding new command with Updation of Max/Min values in validation.



```
DeviceModel C20
InterfaceDescription 5 OFCSNT {
roleDetails
{ RoleName= "Ofcsnt Engineer" , AllowedUserNumber= 2 RoleName= "Lmc Engineer" , AllowedUserNumber= 2 }
privilegeDetails
{ PrivilegeId= 5 , PrivilegeName= ofcsnt_engineer_write PrivilegeId= 1 , PrivilegeName= lmc_engineer_write }
commands
{ init_11 11 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
settime_12 12 [ hwId = 1 , cmdPkt = 1 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write , datetime db_datetime = "2017-05-17 12:12:12" ]
set_attn_13 13 [ hwId = 101 , cmdPkt = 2 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write , float Ch_1 = 0.0 , float Ch_2 = 0.0 ]
quit_14 14 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
setMaintenance 79 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
setReset 80 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write ]
setDomonTimeInterval 121 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , short domonTimeInterval = 3 ]
setShutdown 109 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write ]
newCommand [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write, string light= "redled"]
newCommand1 [ hwId = 0 , cmdPkt = 0 , PrivilegeName= ofcsnt_engineer_write , PrivilegeName= lmc_engineer_write, float ch_10 = 0.0, float ch_11 = 0.0 ]
}
```



```
DeviceCommand C20.OFCSNT.newCommand1
 {
     Features {Priority=1, RetryCount= 0,Timeout=40}
     CommandValidation
     {
         Validation {
             C20.OFCSNT.newCommand1.ch_10 [ MaxValue = 30.5 MinValue = 0.0 ]
             C20.OFCSNT.newCommand1.ch_11 [ MaxValue = 30.5 MinValue = 0.0 ]
                   }
     }
     StateTransition
     {currentState C20.OFCSNT.ok, C20.OFCSNT.reset, C20.OFCSNT.timeout, C20.OFCSNT.maintenance,C20.OFCSNT.shutdown=>nextState C20.OFCSNT.suspend}
 }

 } }
```

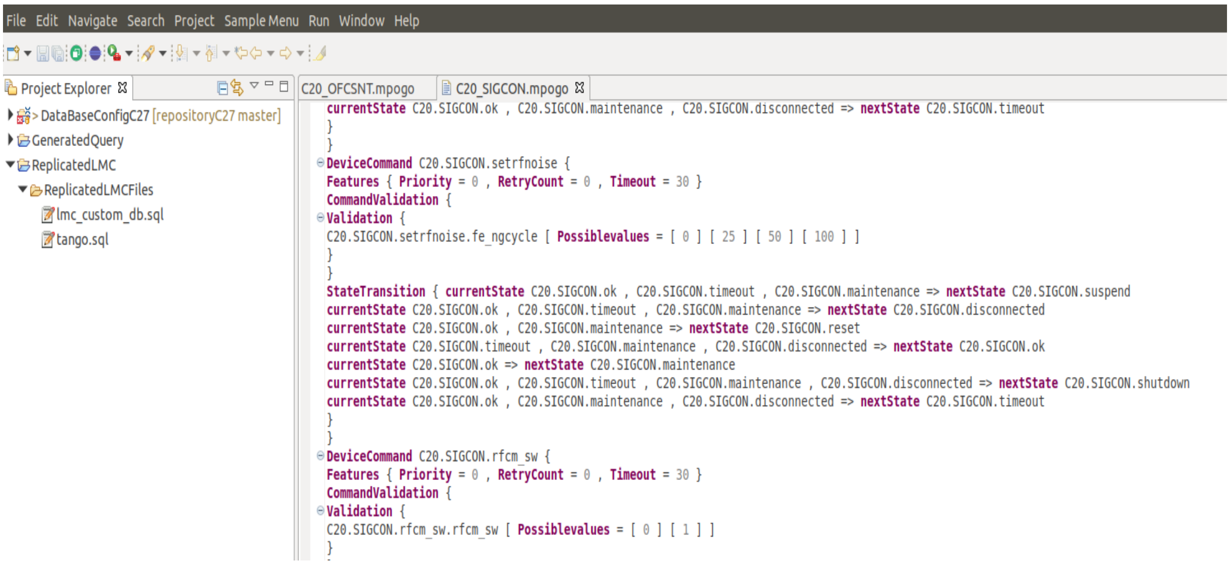


```
Queries Generated
INSERT INTO lmc_custom_db.command(command_id,command_name,arg_count,priority,retry_count,alias,hint,timeout) values(199,'newCommand1','2','1','0','NEWCOMMAND1','hint: newCommand1 ,','40');
INSERT INTO lmc_custom_db.command_assoc_hardware_details (command_id, command_hardware_id, command_pkt)values('199','0','0');
INSERT INTO lmc_custom_db.node_assoc_command(node_assoc_command_id,node_level_id,command_id) values('486','5','199');
INSERT INTO lmc_custom_db.privilege_assoc_command(privilege_id,node_assoc_command_id)values('5','486');
INSERT INTO lmc_custom_db.privilege_assoc_command(privilege_id,node_assoc_command_id)values('1','486');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('199','5','8','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('199','5','9','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('199','5','5','0');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('199','5','10','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('199','5','4','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('199','5','6','1');
 INSERT INTO lmc_custom_db.state_assoc_command_rule(command_id,node_level_id,current_state_id,rule) values('199','5','7','0');
INSERT INTO lmc_custom_db.argument(arg_id,arg_name,arg_datatype) values('366','ch_10','float');
INSERT INTO lmc_custom_db.command_assoc_argument(command_id,arg_id,seq) values(199,366,1);
INSERT INTO lmc_custom_db.argument_default_value(arg_id,default_value,time) values('366','0.0','Tue Jan 07 15:29:50 IST 2020');
INSERT INTO lmc_custom_db.argument_range_value(arg_id,min_value,max_value) values('366','0.0','30.5');
INSERT INTO lmc_custom_db.argument(arg_id,arg_name,arg_datatype) values('367','ch_11','float');
INSERT INTO lmc_custom_db.command_assoc_argument(command_id,arg_id,seq) values(199,367,2);
INSERT INTO lmc_custom_db.argument_default_value(arg_id,default_value,time) values('367','0.0','Tue Jan 07 15:29:50 IST 2020');
INSERT INTO lmc_custom_db.argument_range_value(arg_id,min_value,max_value) values('367','0.0','30.5');
UPDATE lmc_custom_db.command SET hint = 'hint: newCommand1 <ch_10> <ch_11> , , ch_10 ranges from 0.0 to 30.5 , ch_11 ranges from 0.0 to 30.5' where command_id = '199';
```

2.10. Replication of LMC

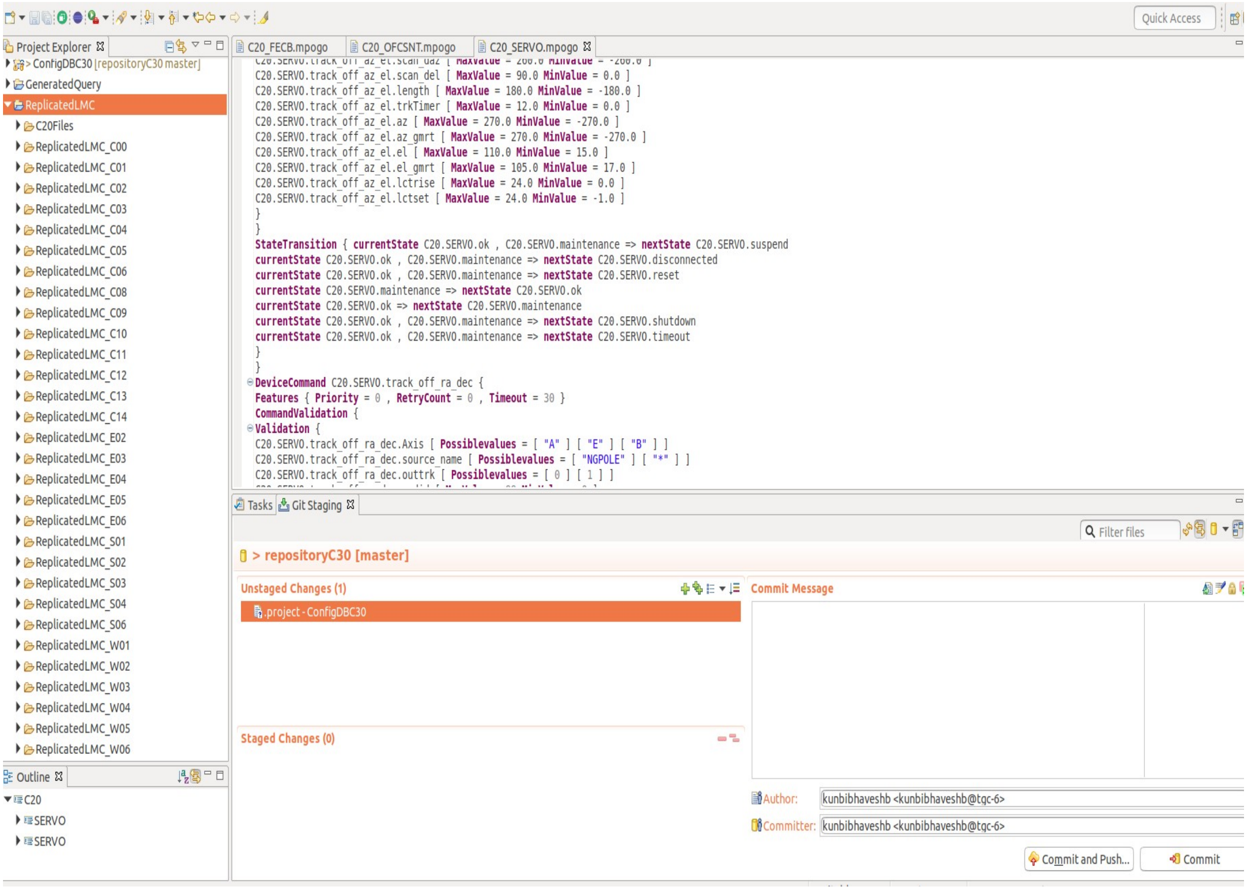
To generate ReplicatedLMC files (lmc_custom_db.sql and tango.sql) followed the steps according to the user manual (User Manual.docx file) provided by TCS.

lmc_custom_db.sql and tango.sql file was generated in ReplicatedLMC folder under project tree as shown in below image.



```
currentState C20.SIGCON.ok , C20.SIGCON.maintenance , C20.SIGCON.disconnected => nextState C20.SIGCON.timeout
}
}
DeviceCommand C20.SIGCON.setrfnoise {
Features { Priority = 0 , RetryCount = 0 , Timeout = 30 }
CommandValidation {
Validation {
C20.SIGCON.setrfnoise.fe_ngcycle [ Possiblevalues = [ 0 ] [ 25 ] [ 50 ] [ 100 ] ]
}
}
StateTransition { currentState C20.SIGCON.ok , C20.SIGCON.timeout , C20.SIGCON.maintenance => nextState C20.SIGCON.suspend
currentState C20.SIGCON.ok , C20.SIGCON.timeout , C20.SIGCON.maintenance => nextState C20.SIGCON.disconnected
currentState C20.SIGCON.ok , C20.SIGCON.maintenance => nextState C20.SIGCON.reset
currentState C20.SIGCON.timeout , C20.SIGCON.maintenance , C20.SIGCON.disconnected => nextState C20.SIGCON.ok
currentState C20.SIGCON.ok => nextState C20.SIGCON.maintenance
currentState C20.SIGCON.ok , C20.SIGCON.timeout , C20.SIGCON.maintenance , C20.SIGCON.disconnected => nextState C20.SIGCON.shutdown
currentState C20.SIGCON.ok , C20.SIGCON.maintenance , C20.SIGCON.disconnected => nextState C20.SIGCON.timeout
}
}
DeviceCommand C20.SIGCON.rfcm_sw {
Features { Priority = 0 , RetryCount = 0 , Timeout = 30 }
CommandValidation {
Validation {
C20.SIGCON.rfcm_sw.rfcm_sw [ Possiblevalues = [ 0 ] [ 1 ] ]
}
```

Generated SQL files for all 30 antennas (in updated version of Configurator tool)

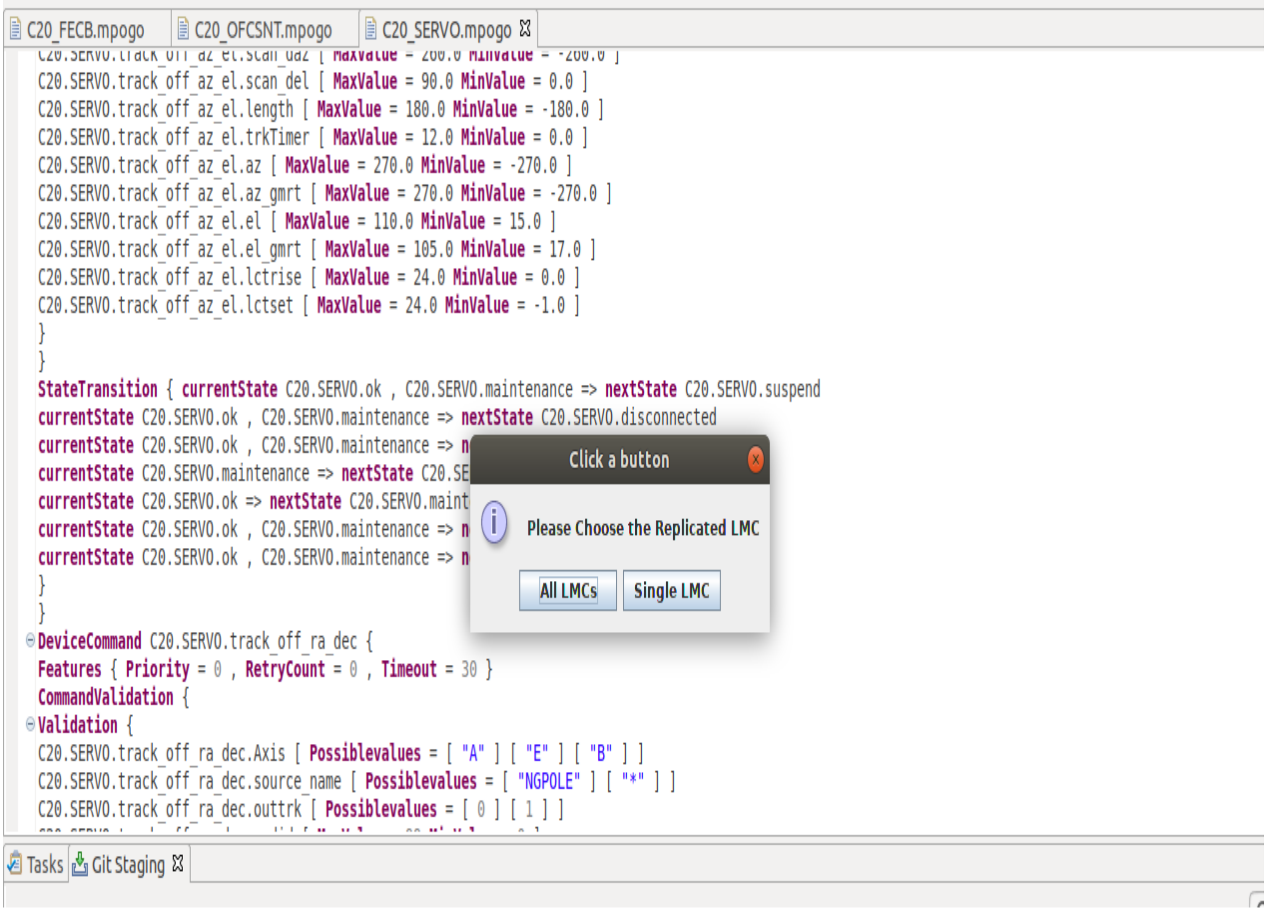


```
C20.SERVO.track_off_az_el.scan_daz [ MaxValue = 200.0 MinValue = -200.0 ]
C20.SERVO.track_off_az_el.scan_del [ MaxValue = 90.0 MinValue = 0.0 ]
C20.SERVO.track_off_az_el.length [ MaxValue = 180.0 MinValue = -180.0 ]
C20.SERVO.track_off_az_el.trkTimer [ MaxValue = 12.0 MinValue = 0.0 ]
C20.SERVO.track_off_az_el.az [ MaxValue = 270.0 MinValue = -270.0 ]
C20.SERVO.track_off_az_el.az_gmrt [ MaxValue = 270.0 MinValue = -270.0 ]
C20.SERVO.track_off_az_el.el [ MaxValue = 110.0 MinValue = 15.0 ]
C20.SERVO.track_off_az_el.el_gmrt [ MaxValue = 105.0 MinValue = 17.0 ]
C20.SERVO.track_off_az_el.lctrise [ MaxValue = 24.0 MinValue = 0.0 ]
C20.SERVO.track_off_az_el.lctset [ MaxValue = 24.0 MinValue = -1.0 ]
}
}
StateTransition { currentState C20.SERVO.ok , C20.SERVO.maintenance => nextState C20.SERVO.suspend
currentState C20.SERVO.ok , C20.SERVO.maintenance => nextState C20.SERVO.disconnected
currentState C20.SERVO.ok , C20.SERVO.maintenance => nextState C20.SERVO.reset
currentState C20.SERVO.maintenance => nextState C20.SERVO.ok
currentState C20.SERVO.ok => nextState C20.SERVO.maintenance
currentState C20.SERVO.ok , C20.SERVO.maintenance => nextState C20.SERVO.shutdown
currentState C20.SERVO.ok , C20.SERVO.maintenance => nextState C20.SERVO.timeout
}
}
DeviceCommand C20.SERVO.track_off_ra_dec {
Features { Priority = 0 , RetryCount = 0 , Timeout = 30 }
CommandValidation {
Validation {
C20.SERVO.track_off_ra_dec.Axis [ Possiblevalues = [ "A" ] [ "E" ] [ "B" ] ]
C20.SERVO.track_off_ra_dec.source_name [ Possiblevalues = [ "NGPOLE" ] [ "*" ] ]
C20.SERVO.track_off_ra_dec.outtrk [ Possiblevalues = [ 0 ] [ 1 ] ]
```

Author: kunbibhaveshb <kunbibhaveshb@tgc-6>

Committer: kunbibhaveshb <kunbibhaveshb@tgc-6>

```
--
-- Database: `tango`
--
CREATE DATABASE IF NOT EXISTS `tango` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
USE `tango`;

-- --------------------------------------------------------
```



```
C20.SERVO.track_off_az_el.scan_daz [ MaxValue = 200.0 MinValue = -200.0 ]
C20.SERVO.track_off_az_el.scan_del [ MaxValue = 90.0 MinValue = 0.0 ]
C20.SERVO.track_off_az_el.length [ MaxValue = 180.0 MinValue = -180.0 ]
C20.SERVO.track_off_az_el.trkTimer [ MaxValue = 12.0 MinValue = 0.0 ]
C20.SERVO.track_off_az_el.az [ MaxValue = 270.0 MinValue = -270.0 ]
C20.SERVO.track_off_az_el.az_gmrt [ MaxValue = 270.0 MinValue = -270.0 ]
C20.SERVO.track_off_az_el.el [ MaxValue = 110.0 MinValue = 15.0 ]
C20.SERVO.track_off_az_el.el_gmrt [ MaxValue = 105.0 MinValue = 17.0 ]
C20.SERVO.track_off_az_el.lctrise [ MaxValue = 24.0 MinValue = 0.0 ]
C20.SERVO.track_off_az_el.lctset [ MaxValue = 24.0 MinValue = -1.0 ]
}
}
StateTransition { currentState C20.SERVO.ok , C20.SERVO.maintenance => nextState C20.SERVO.suspend
currentState C20.SERVO.ok , C20.SERVO.maintenance => nextState C20.SERVO.disconnected
currentState C20.SERVO.ok , C20.SERVO.maintenance => n
currentState C20.SERVO.maintenance => nextState C20.SE
currentState C20.SERVO.ok => nextState C20.SERVO.maint
currentState C20.SERVO.ok , C20.SERVO.maintenance => n
currentState C20.SERVO.ok , C20.SERVO.maintenance => n
}
}
DeviceCommand C20.SERVO.track_off_ra_dec {
Features { Priority = 0 , RetryCount = 0 , Timeout = 30 }
CommandValidation {
Validation {
C20.SERVO.track_off_ra_dec.Axis [ Possiblevalues = [ "A" ] [ "E" ] [ "B" ] ]
C20.SERVO.track_off_ra_dec.source_name [ Possiblevalues = [ "NGPOLE" ] [ "*" ] ]
C20.SERVO.track_off_ra_dec.outtrk [ Possiblevalues = [ 0 ] [ 1 ] ]
```

Click a button

Please Choose the Replicated LMC

2.11. Updation of Command:

Sr. No	**Test Cases**	**Testing Results (GCTBetaVersion_2)**	**Testing Results (GCTBetaVersion_1)**	**Testing Results (GCTBetaVersion)**	***Testing Results (GCTBetaVersion_Config)**	**Remarks**
1	Updation of existing argument's default value in command.	OK	OK	OK	OK	
2	Updation of existing privilege details in command.	OK	OK	OK	OK	
3	Updation of possible values in command.	OK	OK	OK	OK	
4	Updation of Max/Min values in command.	OK	OK	OK	OK	
5	Updation of Hardware details values in command.	OK	OK	OK	OK	
6	Updation of specification details (Features) values in command.	NOT OK	NOT OK	OK	OK	UPDATE lmc_custom_db.command set and timeout=40 where command_id=182 Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'and timeout=40 where command_id=182' at line 1 0.00015 sec **~~Note: They will solved this Error in next updated version of Configurator tool.~~** **Error solved in new updated version of configurator tool.**
7	Updation of retry_Count in specification file.	OK	OK	OK	OK	

3. Input from GMRT team:

- Tango.sql file generated using Configurator tool (GCTBetaVersion_1) didn't had the Create and Use database query, due to which when file was sourced database was not imported into the database. Due to the limitation of ***mysqldump*** command the *Create and Use database query* was not generated.
- GMRT team also provided the suggestion to use the skip extended writing option while while writing the SQL files, So that it is visibly easier for user to compare. Suggestion is implemented in subsequent version.
- GMRT team also provided the suggestion to generate SQL files for all 30 antennae in one go because generating SQL file for all LMC one by one will not be appreciated by user of the application. (TCS team implemented this suggestion in updated version of configurator tool: **GCTBetaVersion_Config**)
- GMRT team suggested to implement dialogue box which clearly state whether we want to replicate one LMC or all 30 Antennae LMC as well as to work on memory management of GCT tool which will improve overall execution of the application.
- TCS team implemented the suggested changes in ***GCTBetaVersion_Config.***
- Inconsistencies were identified in lmc_custom_db database while importing into specification files
- This tool can further be optimized related to memory usage.

4. Conclusion:

GCT tool will be very useful in the TANGO based GMRT Control system setup, as it simplifies the generation of database queries related to addition of new commands, deletion of commands, addition of new antenna Sub-System and replication of LMCs. It will help in the reduction of human errors.

5. Future developments:

This tool can further be enhanced to use the domain specific knowledge to generate the model for a Control system, Where knowledge gathered from user and system engineers can be directly modeled using Domain Specific language. Also this tool support generation of Antenna LMC configuration, it can be further developed to generate configuration of GMRT software backend LMC, GMRT wide-band LMC, GMRT analog backend LMC and Central monitoring and Control (CMC).