CMS Configuration and Deployment

Version 3.3.6

Table of Contents

Pre-Requisites:

Production Environment:

1. OS: RHEL5 or above with 32 Bit server version
2. Browser Support:
    - Firefox version 8.0.1 or above
    - IE 7, IE 8
    - Google Chrome version 8.0.552.0 or higher
3. Best Supported Screen Resolutions: 1152 X 864, 1024 X 768

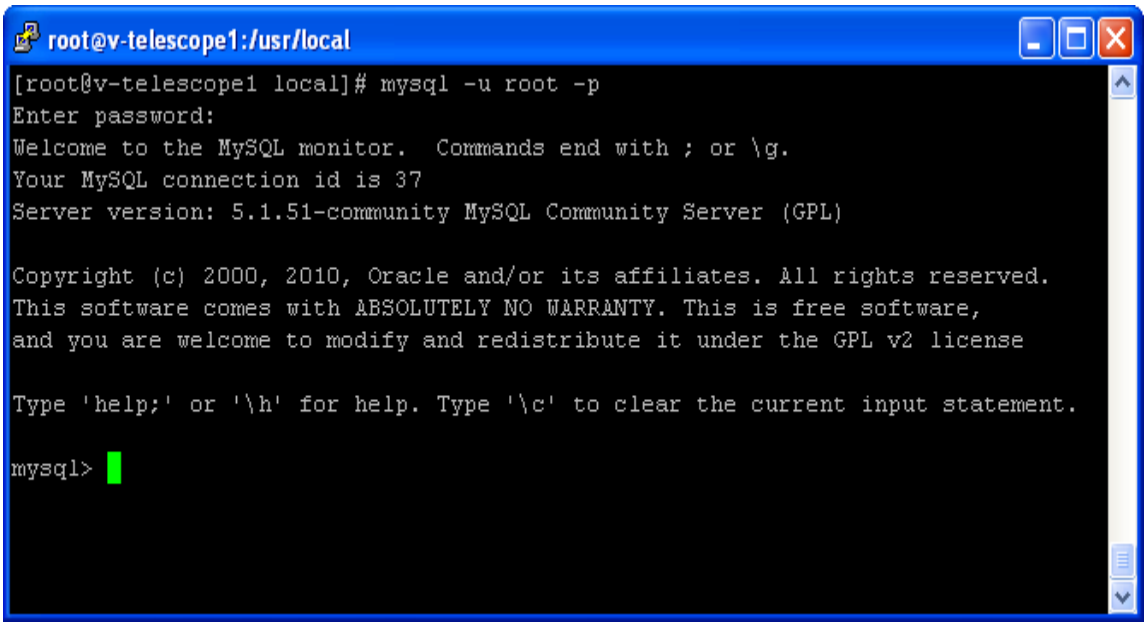
Following should have been installed in the Production Environment

1. Java 1.6 (Download JDK 1.6 from http://www.oracle.com/technetwork/java/javase/downloads/index.html)
2. JDK 1.6 (build 1.6.0_20 or higher)
3. Tomcat 6.0.29 (supported on JDK 1.6) or higher
4. MySQL server v5.5 or higher
5. Apache-activemq-5.4.3 http://activemq.apache.org/activemq-543-release.html
6. gfortran compiler (version 4.1.2.x)

Database Configuration:

For configuring database on Linux machine, follow the following steps

1. Login to Linux machine and type **mysql -u root –p**
   The system will prompt for the password, enter the root password



2. Place the ncracmsdb.sql file at a particular location.

   Example: **the ncramsdb.sql** kept at /usr/local/conf/

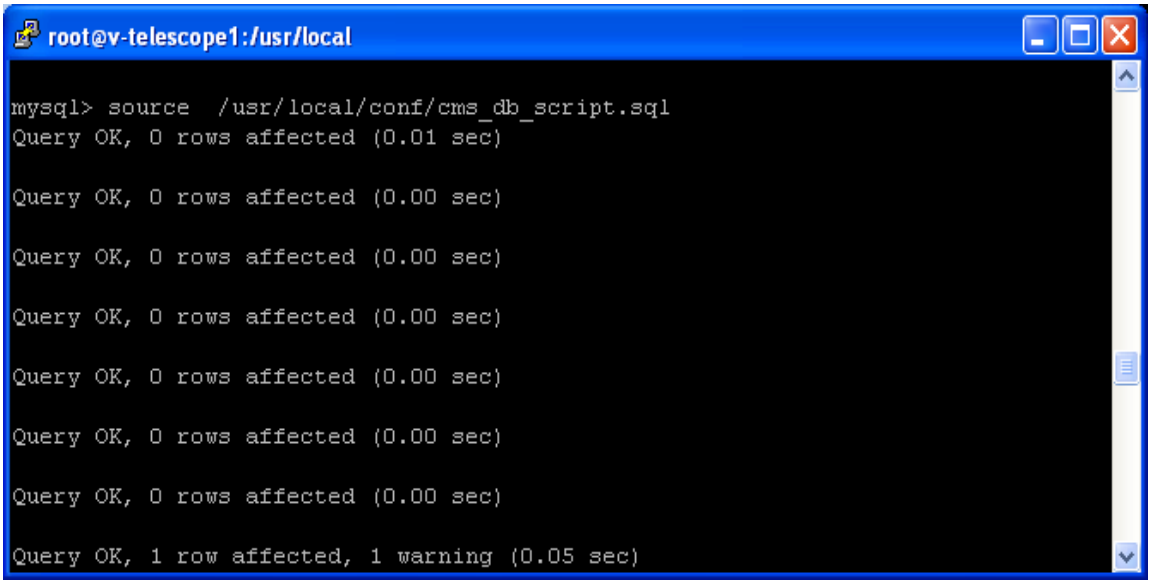
   Now run the "**source**" command at mysql prompt.

   COMMAND: **source "location of sql file"**

   Example: **source /usr/local/conf/the ncracmsdb.sql**

   This command will read the script file and database will be created on the Linux machine

Repeat the same procedure for **p_saveAlarmInfo.sql** and **p_saveMonitoringData.sql** script files as well.



3. **User can see the database created using**

   COMMAND: show databases;

   Database “ncracmsdb” created will be seen.

   To know about more database commands you can go through this link:

   http://www.yolinux.com/TUTORIALS/LinuxTutorialMySQL.html

   http://dev.mysql.com/doc/refman/5.1/en/mysql-commands.html

4. **Database configuration**

   Please refer to Appendix 3 for detailed database configuration.

Pre deployment Configurations:

The conf folder contains the required configuration files as below:

1. ncra15m.catalog
2. CMSRules.drl
3. cms.properties
4. cmsinfo.properties
5. dataAcq.properties
6. globalparameter.properties
7. manualmode.properties
8. shutdownshellscript.sh
9. startupshellscript.sh
10. batch1.txt
11. batch2.txt
12. batch3.txt
13. batch4.txt
14. init_on_powerfailure.txt
15. initAllSubsystems.txt
16. initBackEnd.txt
17. initFrontEnd.txt
18. initSentinal.txt
19. initServo.txt
20. initSigcon.txt
21. reset_backend.txt
22. reset_frontend.txt
23. reset_sentinal.txt
24. reset_servo.txt
25. reset_sigcon.txt
26. restore.txt
27. restore_backend.txt
28. restore_frontend.txt
29. restore_sentinal.txt
30. restore_servo.txt
31. restore_sigcon.txt
32. shutdownAllSubsystems.txt
33. shutdownBackend.txt
34. shutdownFrontend.txt
35. shutdownSentinal.txt
36. shutdownServo.txt
37. shutdownSigcon.txt
38. testcommoncommands.txt

39. testncracommands.txt
40. updateAcqProgress.txt
41. updateAcqStart.txt
42. updateAcqStop.txt
43. updateTrackingData.txt
44. validateServoMonitoringData.txt
45. backend_commands.xml
46. backend_engineering.xml
47. ChartRecorder.xml
48. cms_commands.xml
49. continuum.xml
50. frontend_commands.xml
51. frontend_engineering.xml
52. ncra-subsystemconfig.xml
53. planetory.xml
54. pulsar.xml
55. PulsarDisplay.xml
56. receiverstatus.xml
57. sentinal_commands.xml
58. sentinal_engineering.xml
59. servo_commands.xml
60. servo_engineering.xml
61. servoTrendPlot.xml
62. sigcon_commands.xml
63. sigcon_engineering.xml
64. spectral-line.xml
65. SpectralLineDisplay.xml
66. sun-moon.xml
67. ChartRecorder.xsl
68. htmlgenerator.xsl
69. PulsarDisplay.xsl
70. SpectralLineDisplay.xsl
71. logback.xml
72. BatchMenu.properties
73. CatalogMenu.properties
74. MetaData.xml
75. MetaData.xsl
76. updateBandCenterFrequency

Place all the above mentioned files in any folder (for e.g. /NCRA/lib/)

Edit /apache-tomcat-6.0.29/conf/catalina.properties and change value of following property to the path mentioned above. This would be referred as $librarypath henceforth in the document.

shared. loader= /NCRA/lib/

Note : As mentioned above, the location of configuration files has been changed to facilitate easy upgrade of tomcat server in future, hence please remove all cms configuration files (listed above) from apache-tomcat-6.0.29/lib folder.

Also, since CMS uses star link library for all astronomical calculations, it should be made available to CMS using procedure mentioned below -

1) Please copy libtact.so (Please see section "Appendix 2" for steps to build the .so file) file to **/NCRA/sharedlib** directory.

2) Edit apache-tomcat-6.0.29/bin/catalina.sh for following settings –

   a. Configure Shared library path

      -Djna.library.path=/NCRA/sharedlib

```
"$_RUNJAVA" "$LOGGING_CONFIG" $JAVA_OPTS $CATALINA_OPTS \
  -Djava.endorsed.dirs="$JAVA_ENDORSED_DIRS" -classpath "$CLASSPATH" \
  -Dcatalina.base="$CATALINA_BASE" \
  -Djna.library.path=/NCRA/sharedlib \
  -Dcatalina.home="$CATALINA_HOME" \
  -Djava.io.tmpdir="$CATALINA_TMPDIR" \
  org.apache.catalina.startup.Bootstrap "$@" start \
  >> "$CATALINA_OUT" 2>&1 &
```

Note that there are multiple such sections in catalina.sh file, the section to changed starts around line #344.

   b. Configure JVM settings

      Following change should be made in catalina.sh file

      JAVA_OPTS="-XX:PermSize=100m -XX:MaxPermSize=128m –Xms1024m –Xmx1024m -XX:-UseConcMarkSweepGC -XX:-UseParallelOldGC -XX:ParallelGCThreads=1 -XX:ParallelCMSThreads=1 -XX:+UseFastAccessorMethods -XX:+UseStringCache -XX:+UseCompressedStrings -XX:+OptimizeStringConcat "

      Note

      The value of -Xms, -Xmx can be increased further depending upon the amount of RAM available on server machine.

The above mentioned change (line) needs to done just at the start of the catalina.sh file (after comments section) around line # 85

```
#   LOGGING_CONFIG  (Optional) Override Tomcat's logging config file
#                   Example (all one line)
#                   LOGGING_CONFIG="-Djava.util.logging.config.file=$CATALINA_BASE/conf/logging.properties"
#
#   LOGGING_MANAGER (Optional) Override Tomcat's logging manager
#                   Example (all one line)
#                   LOGGING_MANAGER="-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager"
#
# $Id: catalina.sh 947714 2010-05-24 16:57:18Z markt $
# -----------------------------------------------------------------------------

JAVA_OPTS="-XX:PermSize=100m -XX:MaxPermSize=192m -Xmx1024m -Dcom.sun.management.jmxremote"

# OS specific support.  $var _must_ be set to either true or false.
cygwin=false
os400=false
darwin=false
case "`uname`" in
CYGWIN*) cygwin=true;;
OS400*) os400=true;;
Darwin*) darwin=true;;
esac
```

c. Extended Directory configuration

1. Create a directory named ‘endorsed’ under tomcat home directory
2. Copy following library files (found under binary/lib directory) in that directory
    1. resolver.jar
    2. serializer-2.7.1.jar
    3. xercesImpl.jar
    4. xml-apis.jar
3. Copy following line in catalina.sh somewhere around line # 260

    JAVA_ENDORSED_DIRS="$CATALINA_HOME"/endorsed

Configuration in cmsinfo.properties:

Open cmsinfo.properties and modify the line with key “CMS_USER_DOCUMENTATION”. The entry against this key should be http://<serverIPAddress>/cms-web/CMS_USER_MANUAL.docx

```
#URLs
CMS_USER_DOCUMENTATION=http://localhost:8080/cms-web/CMS_USER_MANUAL.doc
```

Configuration in cms.properties:

Open cms.properties using vi editor and modify the following as per your environment settings:

1. Database configurations

```
# MySQL
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost/ncracmsdb
jdbc.username=root
jdbc.password=root
```

    1. If ncracmsdb is installed on different machine other than the tomcat installed machine then change jdbc.url to point to the new machine. Replace localhost with the machine ip address.
    2. Change the jdbc.username and jdbc.password as per the username and password specified in mysql

2. Catalog file settings

```
########## APPCONFIG ################

# 1 - NCRA
# 2 - IUCAA
app.config=1
catalog_file=/usr/ncra/lib/ncra15m.catalog
organisation_header=NCRA TIFR Pune, INDIA
```

1. **app.config** specifies the app config. 1 is for NCRA and 2 for IUCAA.
2. **catalog_file** specifies the absolute path of default catalog file. Currently this is not being used in CMS, this information is for reference only.
3. **organization_header** specifies the title that will appear in header of CMS home page

3. Subsystem configurations

```
########## SUBSYSTEM ################

eventPort=4001
responsePort=4000
monitoringParamPort=19999

subsystemConfig=ncra-subsystemconfig.xml
```

1. **eventPort** specifies the port on which events will be received by CMS.
2. **responsePort** specifies the port on which response will be received by CMS. Incase both ports have the same port address; single port will receive the events and responses.
3. **monitoringParamPort** System Port number to receive monitoring information
4. Subsystem configuration has been shifted to a separate configuration file named **ncra-subsystemconfig.xml**. Following are details about this file

```
<?xml version="1.0" encoding="UTF-8"?>
<subsystems>
    <subsystem>
        <name>servo</name>
        <connectionurl>127.0.0.1:7775</connectionurl>
        <commandfile>servo_commands.xml</commandfile>
        <version>1</version>
        <active>true</active>
        <internal>false</internal>
        <engXML>servo_engineering.xml</engXML>
    </subsystem>
    <subsystem>
        <name>sentinal</name>
        <connectionurl>127.0.0.1:7775</connectionurl>
        <commandfile>sentinal_commands.xml</commandfile>
        <version>1</version>
        <active>true</active>
        <internal>false</internal>
        <engXML>sentinal_engineering.xml</engXML>
    </subsystem>
```

a. **name** – Specifies name of subsystem
b. **connectionurl** - Specifies the ip and port of servo wrapper.(sub-system specific wrapper)
c. **commandfile** - Specifies the command file name for subsystem. The command file should be placed in **$librarypath**.
d. **Version** – System version can be specified here
e. **active** – specifies whether subsystem is active. When set to false subsystem is considered as deactive.
f. **internal** – This is used by cms for internal use. For external subsystem this property should be always set to false.
g. **engXML**
h. Similarly frontend,backend,sigcon, cms can be configured.

4. DefaultCommandTimeout

```
#default command timeout in millisec
defaultCommandTimeout=60000
```

1. **defaultCommandTimeout** specifies the default command time out in seconds. In case a command is not specified any timeout, the default command timeout value will be used as command timeout.

5. ResponseTimeout

```
#default response timeout in mili seconds
responseTimeout=60000
```

1. **responseTimeout** specifies the default timeout for response in milliseconds.

7. Sequence no Generator Settings

```
#sequence file
seq.file=/usr/ncra/seq.dat
```

1. It will create the seq.dat file at specified location, this file maintains current unique sequence no, which is used to track and differentiate commands from each other.

6. Menu settings

```
############ MENU ####################
refresh.delay=1000
file.name=cmsinfo.properties
```

1. **refresh.delay** specifies the delay for refreshing the menu in milliseconds
2. **file.name** specifies the menu file this contains the menu name and its url. This is visible under Information Links->Help Menu page

8. Longitude,latitude and height setting

```
#######Longitude, Latitude and Height of the Antenna#####
longitude=73.49
latitude=19:05:26.35
height=560
```

1. **longitude** (in Degrees) specifies the longitude of telescope location. It is used for various astronomical calculations.
2. **latitude** (in D:M:S Format – 19:05:26.35) Used for various astronomical calculations. **height** (in float) specifies the height of antenna. It is used for various astronomical calculations.

7. Tzone and el_lim settings

```
#####timezone#####
tzone=-5:30

el_lim=17.0
```

1. **tzone** (Supported formats are +hh:mm,-hh:mm or decimal format e.g. 5.5)Used for various astronomical calculations.
2. **el_lim** (supported formats is float) Used for various astronomical calculations

8. Commandlogfilename

```
#######Default file name for command Log#######
commandlogfilename=file.xls
```

1. **Commandlogfilename** specifies the default filename of the command logs excel file which can be downloaded from command Log.

9. Broker URL

```
########### ACTIVEMQ ###################
brokerUrl=tcp://localhost:61616
```

1. This is the ActiveMQ URL. It must be set the port on which ActiveMQ is running.

10. Manual mode Subsystem and settings

```
##########Manual mode############
manualmodeSubsystem=servo
manualmode.file=manualmode.properties
```

1. **manualmodeSubsystem** specifies the name of the sub system on which manual mode commands will be executed.

2. **manualmode.file** specifies the name of the file in which manual mode commands will be present.

11. Data Acquisition Subsystem and settings

```
dataAcqSubsystem=backend
dataAcq.file=dataAcq.properties
```

1. **dataAcqSubsystem** specifies the name of the sub system on which data acquisition commands will be executed.

2. **dataAcq.file** specifies the name file where data acquisition commands will be present.

12. Plot system settings

```
plotSubsystem=backend
```

This setting is used to configure subsystem which generates pulsar and spectral plots and sends images to CMS to render. User can change this setting to reflect the plotting subsystem name.

13. Help Menu settings

```
manualmodehelpmenu=http://www.ncra.tifr.res.in/
cataloghelpmenu=http://www.ncra.tifr.res.in/
```

1. Currently not used will be removed later.

14. Time zone

```
########Global Parameter ##########
time_zone=IST[GMT+5.30]
```

a. Currently not in use. Will be removed later.

15. Server Host settings

```
serverHost=PS0672.persistent.co.in
```

a. **Server Host** for sending emails - Please change the name of this property to email server host in your organization.

16. Off Source Time Out

```
#timeout period (in milliseconds) for antenna goes off source after being on source
off_source_timeout = 180000
```

a. **off_source_timeout** specifies the maximum time out period after which alarm will be raised if antenna goes off the source after being on source.

17. AZ-EL allowable difference

```
####az-el allowable difference###
azDiffLimit=20
elDiffLimit=20
```

1. **azDiffLimit** specifies maximum allowable difference e between antenna AZ position and target AZ position.

2. **elDiffLimit** specifies maximum allowable difference e between antenna EL position and target EL position

18. Monitoring Time Out

```
### monitoring timeout it is milli second###
monitoringTimeout = 180000
```

a. **monitoringTimeout** specifies the time out period for monitoring parameters.

19. Monitoring frequency

```
monitoringfrequency=9000
```

a. **monitoringfrequency** specifies the time interval after which wrapper should send monitoring data.

20. Escape Characters

```
################## Escape Character setting #################
escapeCharacters=true
```

a. Used to handle the emdebbed commands .Currently not used for NCRA.

21. Connectivity Delay

```
connectivityDelay = 3000
```

1. **connectivityDelay** specifies the time interval between subsequent ping requests sent to the wrapper. CMS will ping each wrapper, if not getting connected it will wait for connectivityDelay time and then again ping the wrapper. The ping continues till the wrapper is connected or connectivityTimeout is reached.

22. Connectivity time out

```
connectivityTimeOut = 30000
```

1. **connectivityTimeOut** for checking all wrapper connections i.e. after this time interval cms will check the wrapper connections and if not connected will declare the particular wrapper as not connected.

Time Interval of Alarm

```
### Alarm Time interval is in milliseconds ####
timeIntervalOfAlarm = 300000
```

1. **timeintervalOfAlarm** specifies time interval between saving two same alarms rose one after another and its value is in millisecond.

23. Rules File

```
#######Rules config file############
rulesFile=CMSRules.drl
```

1. **rulesFile** specifies the rules defined for state machine configuration.

24. PolarPlotRefreshInterval

```
polarPlotRefreshInterval   = 900000
```

1. **PolarPlotRefreshInterval** specifies the interval value in mili-second which will be used by scheduler to periodically update the polar plot value. Similarly interval value is also specified for 2Dplot.

25. PolarPlotPointsLimit

```
polarPlotPointsLimit = 15
```

1. **polarPlotPointsLimit** indicates how many points user wants to plot on polar plot. This in turn shows path the object has traversed.

26. ChartRecorderUsage

```
###########chartRecorder usage 0-common for all and 1-per user##############
chartRecorderUsage=0
```

1. **ChartRecorderUsage** is defined for usage of Chart Recorder. If it '0' the chart recorder would be common for all and if '1' it would be user specific i.e per user separate chart recorder instance will run.

27. TwoDPlotUpdateInterval

```
############ 2D Plot UpdateInterval in mili seconds ##################
twoDPlotUpdateInterval = 120000
```

a. **twoDPlotUpdateInterval** specifies the interval value in mili-second which will be used by scheduler to periodically update the twoDplot.

28. Delay

```
#####Response coming from wrapper can be configured by setting delay atribute
delay=10000
```

a. **Delay** specifies the time interval between two ping request to the wrapper.

29. Pre-Obervation time

```
############ Pre-Observation settings for Astronomer #################
########## Period in milli seconds ###################################
#15 Minutes by default, infinite if -1
preObservationTime=900000
```

1. **preObservation** time before observation start time that allows astronomer to upload catalogs and validate his batch file in CMS. If "**-1"** then astronomer can login at any time to perform observation activities.

30. Acqdatapath

```
#############Astronomical Data path############################3
acqdatapath=/usr/ncra/lib/
```

1. **acqdatapath** specifies the directory in which astronomical data will be saved

31. ShutDownShellScriptPath

```
############Shell Script path##############################
shutDownShellScriptPath=/usr/ncra/lib/shutdownshellscript.sh
```

a. **ShutDownShellScriptPath** specifies the location for complete CMS shutdown s script.

32. Observation Scheduler frequency

observation_schedular_frequency specifies the time interval after which observation scheduler will be invoked in milliseconds. Observation Scheduler keeps track of active schedule for astronomer/co-astronomer. Time interval must be greater than 15 minutes i.e. 900000 ms.If it is less, then default value of 900000 ms will be assigned.

```
observation_schedular_frequency=900000
```

33. Batch Template Settings

batchTemplateDir.name specifies the directory path in which batch templates will be stored.

batchHelpMenu.file specifies the menu configuration for batch templates to be displayed to user.

```
#########Batch Template directory path############
batchTemplateDir.name=/usr/ncra/lib/BatchTemplate
batchHelpMenu.file=BatchMenu.properties
```

34. Catalog Template Settings

catalogTemplateDir.name specifies the directory path in which catalog templates will be stored.

catalogHelpMenu.file specifies the menu configuration for catalog templates to be displayed to user.

```
##########Catalog Template directory path#############
catalogTemplateDir.name=/usr/ncra/lib/Catalog
catalogHelpMenu.file=CatalogMenu.properties
```

35. Email Address configurations

    fromEmailAddress specifies the email address which will appear in "**From:**" field in emails sent out from CMS.

    criticalalarmemailalias specifies the email address to which mails will be sent out if a critical alarm is raised in or received by CMS.

```
############Email Address configurations##############
fromEmailAddress=cmsapplication@persistent.co.in
criticalalarmemailalias=iucaa@persistent.co.in
```

36. Commands supporting strict setting of global parameters:

    These commands will set the global parameter values at set per user strictly; no internal calculation will alter the values as set by user. The command is specified in the format: <subsystem_name>_<command_name>

    To add more commands separate them using comma sepearator.

```
###format is subsystemname_commandname###################
strictGlobalSetCmds=servo_rawtrack
```

Command Configurations:

To configure a command of a subsystem the corresponding subsystem's xml file needs to be modified. The xml can be modified using any standard xml editor or any text editor. Note that the text editor will not point to any error in the xml syntax. One can open this file in browser like IE, which detects few errors in xml syntax like whether document is well-formed or not.

Configuring a command for servo:

1. Find the command in the servo_commands.xml file. For e.g. position

```
<command>
    <name>position</name>
    <id>42</id>
    <syntax>ax,ang1,ang2</syntax>
    <sample>B,123:30:20,123:50:10</sample>
    <params>
        <param required="true">
            <paramname>ax</paramname>
            <type>string</type>
            <validation>
                <values>
                    <value>A</value>
                    <value>E</value>
                    <value>B</value>
                </values>
            </validation>
        </param>
        <param>
            <paramname>ang1</paramname>
            <type>angle</type>
            <validation>
                <angle>
                    <degree>
                        <min>0</min>
                        <max>360</max>
                    </degree>
                </angle>
            </validation>
        </param>
        <param>
            <paramname>ang2</paramname>
            <type>angle</type>
            <validation>
                <angle>
                    <degree>
                        <min>30</min>
                        <max>300</max>
                    </degree>
                </angle>
            </validation>
        </param>
    </params>
```

- <name> - specifies the command name
- <id> - specifies the command id.
- <syntax> - specifies the command syntax. This appears in the Expert Tab ->Syntax field. More than one parameter should be comma separated.
- <sample> -specifies the command sample. This appears in the Expert Tab-> Command Field. More than one parameter should be comma separated.
- <timeout> - This is the optional tag. It specifies timeout period for a command. If not specified defaultCommandTimeout value is considered as time out period for command.
- <params> - this tag specifies the parameter validation.
- <param> - this tag specifies validation for a single parameter

- required-“true” – specifies that this parameter is mandatory
- <paramname> - this specifies the parameter name to which validation is to be applied. This name should match to the one mentioned in syntax.
- <type> - specifies the type of parameter. Currently the following types are supported:
  integer – for integer values
  string – for a value that lies within provided string values
  long – for long values
  float – for float values
  regex – for values that can be evaluated with a regular expression
  double – for double values
- <dependency_validations> -If a particular parameters value is dependent on other parameters value <dependency_validation> tag is used. This tag is optional.

  Example: When parameter “ax” of position command has value “A” then another parameter “ang1” must lie in the range 0-360.
  This dependency validation must be defined as follows in <dependency_validations> tag.

```
<dependency_validations>
    <params>
        <param required="true">
            <paramname>ax</paramname>
            <type>string</type>
            <validation>
                <values>
                    <value>A</value>
                </values>
            </validation>
        </param>
        <param required="true">
            <paramname>ang1</paramname>
            <type>angle</type>
            <validation>
                <angle>
                    <degree>
                        <min>0</min>
                        <max>360</max>
                    </degree>
                </angle>
            </validation>
        </param>
    </params>
```

In order to make system more flexible and loosely coupled, few features have been implemented using batch scripts, .for e.g. Init and shutdown routines etc. This gives flexibility to change the logic in these scripts in future and system would be agnostic to these changes.

```
<supportedbatches>
    <batchcommand>
        <name>initServo</name>
        <id>1233</id>
        <syntax></syntax>
        <sample></sample>
        <filepath>/usr/ncra/lib/initServo.txt</filepath>
    </batchcommand>
    <batchcommand>
        <name>restoreServo</name>
        <id>412</id>
        <syntax></syntax>
        <sample></sample>
        <filepath>/usr/ncra/lib/restore_servo.txt</filepath>
    </batchcommand>
    <batchcommand>
        <name>resetServo</name>
        <id>412</id>
        <syntax></syntax>
        <sample></sample>
        <filepath>/usr/ncra/lib/reset_servo.txt</filepath>
    </batchcommand>
    <batchcommand>
        <name>shutdownServo</name>
        <id>1010</id>
        <syntax></syntax>
        <sample></sample>
        <filepath>/usr/ncra/lib/shutdownServo.txt</filepath>
    </batchcommand>
</supportedbatches>
```

- <name> - specifies the batch name
- <id> - specifies the batch id.
- <syntax> - specifies the parameter list to be sent as input parameters to batch file. This appears in the Expert Tab ->Syntax field. More than one parameter should be comma separated.
- <sample> -specifies the values for parameters mentioned above. This appears in the Expert Tab-> Command Field. More than one parameter should be comma separated.
- <filepath> - Specifies location of batch script file, please ensure this path is set correctly in your environment

2. Please see below an excerpt from batch script: initServo.txt

```
#initialization script for Servo

# init command
$cmd1 = command("servo,init");
info("done with issuing init command $cmd1");
$cmd1status =  waitforCmdCompletion($cmd1 ,10);
info("status of init command : $cmd1status");

# stowrelease command
$cmd2 = command("servo,stowrelease,B");
info("done with issuing stowrelease command $cmd2");
$cmd2status =  waitforCmdCompletion($cmd2 ,10);
info("status of stowrelease command : $cmd2status");

# doMon command
$cmd3 = command("servo,doMon");
info("done with is suing doMon command $cmd3");

if  ($cmd1status != 5  ||  $cmd2status != 5){
    error("marking batch as failed since required commands failed");
    updateBatchStatus(2);
}
```

Initialization Configuration for CMS

1. To initialize all subsystems in CMS, configure the batch command “initAllSubsystems” in cms_command.xml as displayed below. The initAllSubsystems is called when CMS is started for the first time and when CMS was shut down properly.

```
<batchcommand>
    <name>initAllSubsystems</name>
    <id>412</id>
    <syntax></syntax>
    <sample></sample>
    <filepath>/usr/ncra/lib/initAllSubsystems.txt</filepath>
</batchcommand>
```

2. The **initAllSubsystems** is a batch script containing initialization commands to be sent to each subsystem. The screenshot below displays the sample initialization script.

initAllSubsystems - Notepad

File Edit Format View Help

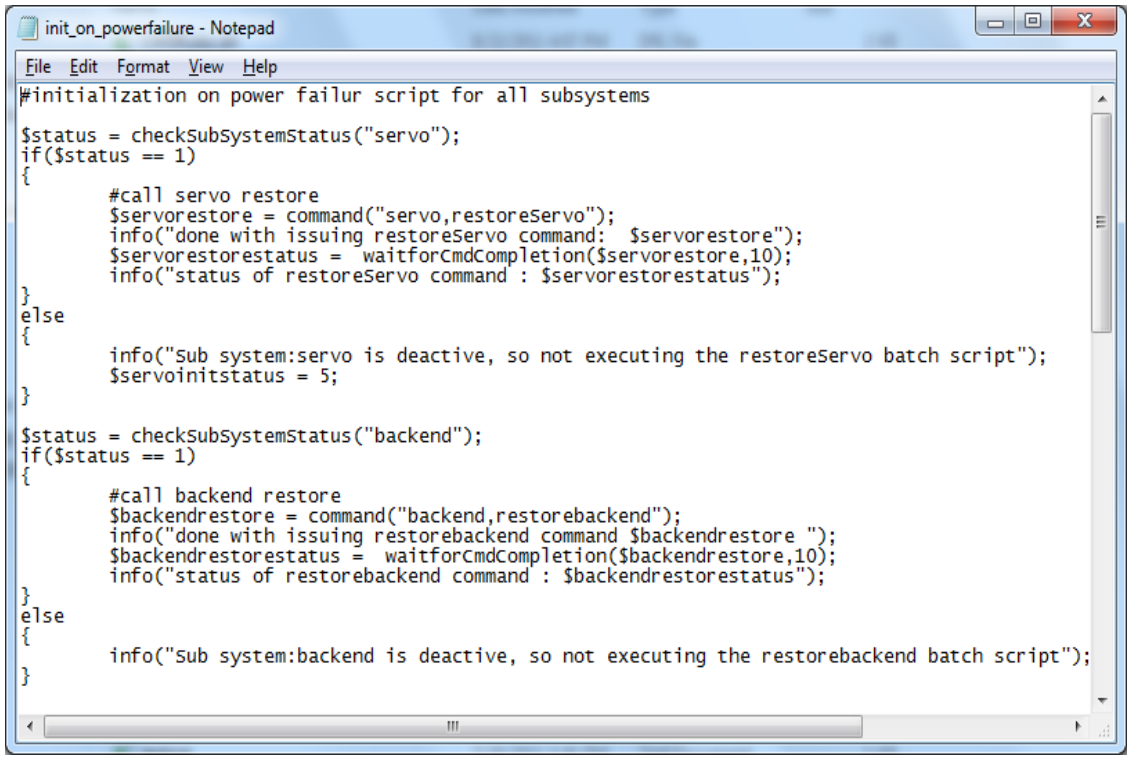
```
#initialization script for all subsystems


$status = checkSubSystemStatus("servo");
if($status == 1)
{
        #call servo init
        $servoinit = command("servo,initServo");
        info("done with issuing initServo command $servoinit");
        $servoinitstatus =  waitforCmdCompletion($servoinit,10);
        info("status of initServo command : $servoinitstatus");
}
else
{
        info("Sub system:servo is deactive, so not executing the initServo batch script");
        $servoinitstatus = 5;
}


$status = checkSubSystemStatus("backend");
if($status == 1)
{
        #call backend init
        $backendinit = command("backend,initbackend");
        info("done with issuing initbackend command $backendinit ");
        $backendinitstatus =  waitforCmdCompletion($backendinit,10);
}
else
{
        info("Sub system:backend is deactive, so not executing the initbackend batch script");
}
```

By default each batch script will be marked as successful. To mark a batch as failure use the command **updateBatchStatus (2);** the failure can be marked based on some command failure refer to initAllSubsystems.txt for the sample batch failure scenario implementation.

3. Similarly also Configure init on power failure in cms_commands.xml. The **init_on_powerfailure** script is called during CMS initialization when CMS was not properly shutdown the last time. This script contains the restoration commands in order to restore the subsystem to previous known state.

```
<batchcommand>
    <name>init_on_powerfailure</name>
    <id>412</id>
    <syntax></syntax>
    <sample></sample>
    <filepath>/usr/ncra/lib/init_on_powerfailure.txt</filepath>
</batchcommand>
```

Sample **init_on_powerfailue.txt** batch script:



For the restore command to succeed, the parameter which is being restored should be present in **t_recent_monitoring_data** table. And the parameter if being used in a command the parameter should be within validation range configured in the command.

4. Similarly for each subsystem initialization script exists which are called from the initAllSubsystems batch script.

   Consider configuring initialization of servo subsystem.

   In servo_commands.xml configure" **initServo**" command.

```
<batchcommand>
    <name>initServo</name>
    <id>1233</id>
    <syntax></syntax>
    <sample></sample>
    <filepath>/usr/ncra/lib/initServo.txt</filepath>
</batchcommand>
```

   Sample **"initServo.txt"**:

initServo.txt - Notepad

File Edit Format View Help

```
#initialization script for Servo

# init command
$cmd1 = command("servo,init");
info("done with issuing init command $cmd1");
$cmd1status =  waitforCmdCompletion($cmd1 ,10);
info("status of init command : $cmd1status");

# stowrelease command
$cmd2 = command("servo,stowrelease,B");
info("done with issuing stowrelease command $cmd2");
$cmd2status =  waitforCmdCompletion($cmd2 ,10);
info("status of stowrelease command : $cmd2status");

# doMon command
$cmd3 = command("servo,doMon");
info("done with is suing doMon command $cmd3");

if  ($cmd1status != 5  ||  $cmd2status != 5){
        error("marking batch as failed since required commands
failed");
        updateBatchStatus(2);
}
```

5. Similarly also configure restoration script for each subsystem. These scripts are called from the init_on_powerfailure batch script.

Consider restoration of servo subsystem.
In servo_commands.xml configure "**restore_servo**" command.

```
<batchcommand>
    <name>restoreServo</name>
    <id>412</id>
    <syntax></syntax>
    <sample></sample>
    <filepath>/usr/ncra/lib/restore_servo.txt</filepath>
</batchcommand>
```

Sample **"restore_servo.txt"** batch script**:**

restore_servo.txt - Notepad

File Edit Format View Help

```
#restore az_cp and el_cp

$az_cp = restore("servo,az_cp");
$el_cp = restore("servo,el_cp");

info("az_cp value restored-> $az_cp");
info("el_cp value restored0> $el_cp");

if ($az_cp ne "" && $el_cp ne ""){

$cmd = command("servo,setstowangles,B,$az_cp,$el_cp");
info("sent command setstowangles with seq $cmd");

$cmdstatus =  waitforCmdCompletion($cmd ,10);
info("status of setstowangles command : $cmdstatus");

}


if ($cmdstatus != 5){
        updateBatchStatus(2);
```

Alarm Configuration

Alarm for a particular sub system can be configured in the following way.

Add the alarm tag in respective sub-system xml.

For e.g. - Following alarm will be raised when **initAllSubystems** fails.

```
<alarm>
    <name>initAllSubsystems</name>
    <label>init failed</label>
    <id>100</id>
    <level>5</level>
    <message>cms initialization failed</message>
</alarm>
```

- <name> - specifies the alarm name, should be unique for a subsystem
- <label> - specifies the alarm label displayed on UI
- <id> - specifies the alarm id
- <level> - specifies the alarm level, should be from 1 to 5
- <message> - message displayed to user on UI when alarm is raised

Logical validation on the received response

1. When a response is received logical validation can be done to check whether the parameters received in response are valid or not.
2. An alarm can also be raised if the logic fails, say a parameter is not in range so raise a common alarm
3. For e.g. in domon for **az_cp** values are checked to determine whether they are in range or out of range, else alarm is raised for the same.
4. The logical validation is done using a batch script, the batch script is configured in response in the following way:

```
    <logicalvalidation>
        <batchcommand>
            <name>validateMonitoringData</name>
            <id>1233</id>
            <syntax>az_cp,el_cp</syntax>
            <sample>az_cp,el_cp</sample>
            <filepath>/usr/ncra/lib/validateServoMonitoringData.txt</filepath>
        </batchcommand>
    </logicalvalidation>
```

Refer to servo_commands.xml for addition of logicalvalidation.

Start up and Shutdown shell script configuration

1. The Startup script needs to be configured with the Tomcat and ActiveMQ server paths. A sample Start up script is provided in config folder. The startup script is not being used internally by CMS; the user will have to use it externally for starting Tomcat and ActiveMQ servers.

   startupshellscript.sh

```
echo "Executing cms start script";
sh /apache-activemq-5.4.2/bin/activemq start
sh /apache-tomcat-6.0.24/bin/catalina.sh start
```

   As displayed the Tomcat and ActiveMQ path must be correctly set for the script to work.

2. The Shutdown script needs to be configured with the Tomcat and ActiveMQ server paths. A sample Shutdown script is provided in config folder. The shutdown script is being used internally by CMS to shutdown the Tomcat and ActiveMQ server.

   shutdownhellscript.sh

```
echo "Executing cms shutdown script";
sh /apache-tomcat-6.0.24/bin/catalina.sh stop 15 -force
sh /apache-activemq-5.4.2/bin/activemq stop
```

   As displayed the Tomcat and ActiveMQ path must be correctly set for the script to work.

Application Deployment

The application binary cms-web.war can be deployed as below:

1. Stop tomcat server using **./catalina.sh stop** command from tomcat bin folder (<TOMCAT_HOME>/bin). Please refer to Appendix 1 section for further details.
2. Delete following artifacts (if exists from Tomcat installation directory)
    a. **cms-web.war** file from **webapps** folder
    b. **cms-web** folder from **webapps** folder
    c. **cms-web** folder from **<TOMCAT_HOME>/work/Catalina/localhost/** folder
3. Place the cms-web.war in tomcat's **webapps** folder (<TOMCAT_HOME>/webapps).
4. Start tomcat using **./catalina.sh start** command from tomcat bin folder (<TOMCAT_HOME>/bin)
5. Once the server starts, check the **webapps** folder, it should contain a folder **cms-web**
6. In case the folder is not created check for cms logs (cms-web.log, cms-core.log) or catalina logs under tomcat's log folder.
7. Once the folder is created, stop the tomcat server and proceed for Post Deployment settings.

Post Deployment Settings

1. In cms-web folder created under webapps, open **channel-config.xml** using any standard text editor.
2. Modify the properties as mentioned below:

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
        <server-protocol>http</server-protocol>
        <server-name>10.44.235.39</server-name>
        <server-port>8080</server-port>
        <context-path>cms-web</context-path>
        <additional-context-path>messagebroker/amfpolling</additional-context-path>
        <channel-name>my-polling-amf</channel-name>
</server>
```

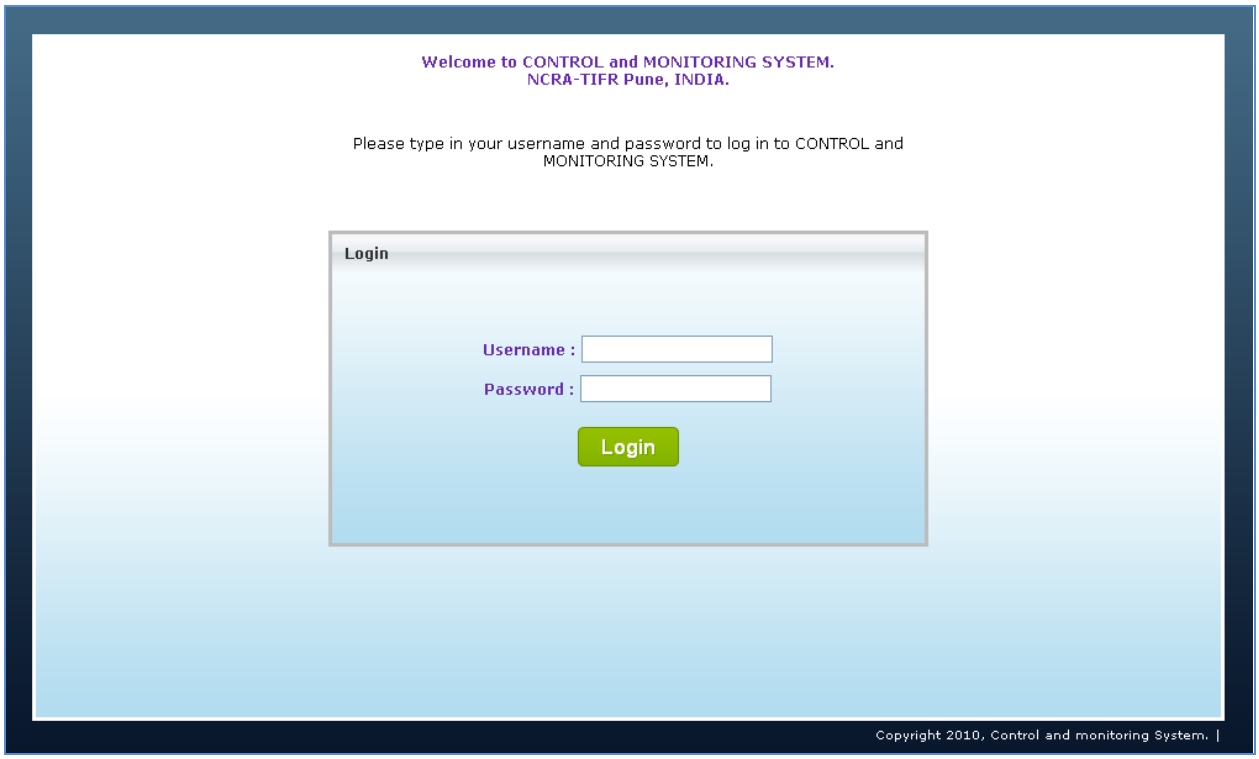
<server-name> - should contain the ip address or name of the server on which the application is deployed. Note that this should be DNS name ideally as the same would be used to access the web application. The application should be accessed using the server-name and port specified here else certain features like Message Console will not work as desired.

3. Start activemq using command **./activemq start** from **/apache-activemq-5.4.2/bin** folder

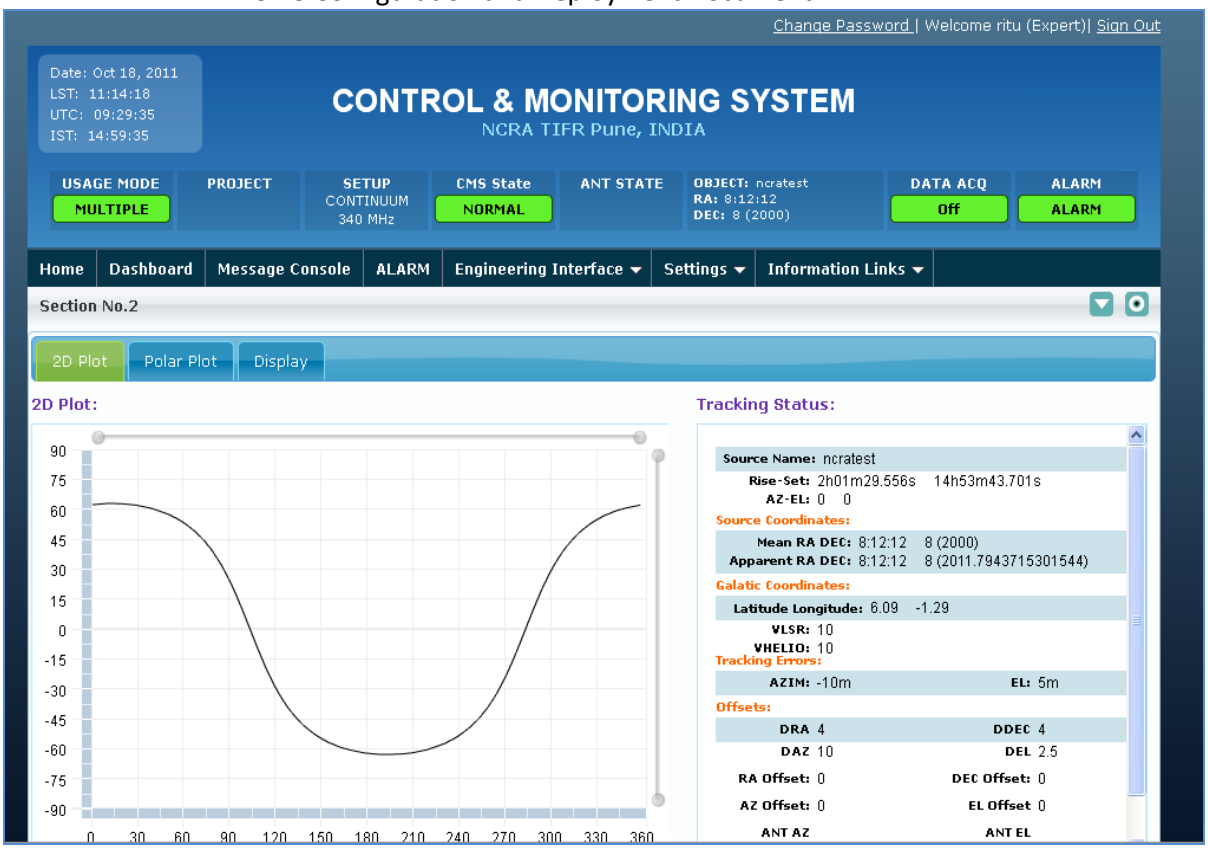
4. Once done. Start the tomcat server.

Application Flow:

1. Successful deployment will enable the user to view the GUI on the browser. To view the application specify the url **http://<server name/ip>:<port>/cms-web**
   On url submit the Login page will be displayed as shown below:



2. User needs to enter the user name and password which will determine its role. If the user enters invalid username and password, appropriate message will be displayed to the user. Default user provided is admin/admin
3. On successful login the home page is visible to the user.



4. To Logout from the application select the **Signout** link provided at the top right of the Home page.

Monitoring Simulator Settings

1. The monitoring simulator jar **ncra_monitoring_simulator.jar** can be located in the binary \MonitoringSimulator folder. This application is used to send monitoring parameters at interval of 6 sec.
2. The SubSystemResponse.xml should be placed in the same folder as the jar
3. This simulator as of now supports generating monitoring parameters for one subsystem at a time. And it reads only SubSystemResponse.xml to find out which parameters to send. So contents of SubSystemResponse.xml file should also be changed depending upon subsystem for which monitoring parameters are being generated.
4. To run this simulator type the following command:
   java -jar ncra_monitoring_simulator.jar
5. The simulator can be closed either by using kill pid option or the simulator will close on its own when cms server is shut.

Configure Monitoring parameters

To configure and test monitoring parameters for a particular sub-system following steps are to be followed:

Consider example for configuring “az_cp” for “**servo**” subsystem

1. Make entry for parameter “**az_cp**” in the “domon” response in servo_commands.xml.
   Also add validation for corresponding parameter.

```
<response>
    <name>doMon</name>
    <id>30</id>
    <params>
        <param>
            <paramname>az_cp</paramname>
            <type>angle</type>
            <validation>
                <angle>
                    <degree>
                        <min>0</min>
                        <max>360</max>
                    </degree>
                </angle>
            </validation>
        </param>
    </params>
</response>
```

2. Similarly configure “**az_cp**” in servo_engineering.xml so that the parameter status can be observed on the Servo Engineering GUI.

3. If alarm is to be generated when a corresponding parameter crosses its max limit, add alarm entry for that monitoring parameter in that particular subsystem commands .xml.
   For example if alarm is generated when “**az_cp**” reaches maximum limit.
   Configure alarm for it in servo_commands.xml.

```
<supportedalarms>
    <alarm>
        <name>az_cp</name>
        <label>Az  upper/lower limits crossed</label>
        <id>107</id>
        <level>5</level>
        <message>Az  upper/lower limits crossed</message>
    </alarm>
</supportedalarms>
```

4. For testing monitoring parameters, externally xml is provided (**SubsystemResponse.xml**) to configure monitoring parameters and test them.

   Consider testing of monitoring parameters for servo subsystem.
   Configure sub system name as "**servo**" and add monitoring parameter" **az_cp**" in the param tag.

   Similarly Configurations for other subsystem's monitoring parameter can be done.

```
<SubsystemName id="servo">
    <params>
        <param>
            <name>az_cp</name>
            <value>100</value>
            <type>angle</type>
        </param>
    </params>
</SubsystemName>
```

Appendix -1

Sometimes we've noticed that tomcat server doesn't stop even after executing ./catalina.sh command, so to ensure that tomcat is stopped completely used following command

ps –aef | grep tomcat

In case tomcat is not stopped completely one should see output like below -

root 23539 20965 0 Apr13 ? 00:02:23 /usr/java/jdk1.6.0_20/bin/java -Dcatalina.base=/var/lib/apache-tomcat-6.0.29 -Dcatalina.home=/var/lib/apache-tomcat-6.0.29 -Dwtp.deploy=/var/lib/apache-tomcat-6.0.29/webapps -Djava.endorsed.dirs=/var/lib/apache-tomcat-6.0.29/common/endorsed -classpath /var/lib/apache-tomcat-6.0.29/bin/bootstrap.jar:/usr/java/jdk1.6.0_20/lib/tools.jar -agentlib:jdwp=transport=dt_socket,suspend=y,address=localhost:52375 org.apache.catalina.startup.Bootstrap start

Please use following command to stop tomcat forcefully.

Kill -9 <PID>

Appendix 2

Building Shared Library

Copy the source code (from **NCRA/tactcalculation** folder) for astronomical calculations to a folder (e.g. /home/calculation). This location would be called as $CALCULATION_HOME henceforth in the document.

Building shared library can be divided into 2 sections –

a. Building the FORTRAN library to generate libsla.a

1) Change the current directory to $CALCULATION_HOME/SLA/sla
2) Set the environment variable **SYSTEM** to appropriate value, for Linux it should be ix86_Linux (Please refer README file from star link library for further details)
   e.g.
   export SYSTEM=ix86_Linux

3) Execute **make clean** command
4) **./mk build** command, this will create libsla.a file, copy this file to $CALCULATION_HOME folder. (Ensure that file "mk" has execute permissions)
5) Copy libsla.a library to $CALCULATION_HOME directory

b. Building our source which will use FORTRAN library for astronomical calculation

1) Change directory back to $CALCULATION_HOME
2) Execute **make clean** command first

Execute **make** command and verify that libtact.so file is created in current directory

Appendix 3

Web Server Configuration

It is recommended that web server and database servers should be installed and configured on different machines. In order to for this scheme of things to work, database server should allow remote connections from web server. Following is the command to enable this remote connection –

GRANT ALL ON ncracmsdb.* to <username>@webserverIP IDENTIFIED BY <pwd>;

Database Configuration

Due to highly concurrent nature of CMS, default MySql configuration needs to be altered to suit the need of CMS applications.

The configuration parameters to be changed or added can be located in /etc/my.cnf in Redhat Enterprise Linux platform.

Following is list of parameters from this file and their corresponding values -

max_connections=400

table_cache=1024

query_cache_size=16M

thread_cache_size=400

innodb_additional_mem_pool_size=16M

innodb_buffer_pool_size=900M

innodb_log_buffer_size=8M

innodb_flush_log_at_trx_commit=1

innodb_lock_wait_timeout=50

open_files_limit=3072

myisam_sort_buffer_size=0M

innodb_flush_method=O_DIRECT

log-queries-not-using-indexes=1

long_query_time=3

log-slow-queries=/var/log/mysql-slow.log

wait_timeout=300

Please refer to snapshot below for further details -

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
max_connections=400
table_cache=1024
query_cache_size=16M
thread_cache_size=400
innodb_additional_mem_pool_size=16M
innodb_buffer_pool_size=900M
#innodb_log_file_size=200M
innodb_log_buffer_size=8M
innodb_flush_log_at_trx_commit=1
innodb_lock_wait_timeout=50
#innodb_log_files_in_group=3
open_files_limit=3072
myisam_sort_buffer_size=0M
#default-storage-engine=INNODB
innodb_flush_method=O_DIRECT
log-queries-not-using-indexes=1
user=mysql
long_query_time=3
log-slow-queries=/var/log/mysql-slow.log
wait_timeout=300
# Default to using old password format for compatibility with mysql 3.x
# clients (those using the mysqlclient10 compatibility package).
old_passwords=1

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
~
~
~
~
~
```

Note – In order to configure higher value for **open_files_limit** parameter, corresponding operating system level settings needs to be changed as well. On RHEL 5.x platform, it is configured in **/etc/security/limits.conf** file.

- Value to be set in this is for user who runs MySql database
- Value in this file should be greater than value of **open_files_limit** parameter

Please refer to snapshot below for further reference –