Alarm System

Change log

Issues	Date	Comment	Author
	03/04/2009	Updated to version 1.1	Graziano Scalamera
	05/05/2016		Graziano Scalamera

Table of Contents

Executive Summary

This document describes the status and the activities about the Alarm System.

Introduction

An alarm is an asynchronous notification that some event has happened or that a given state has been reached in the plant. The alarm system is in charge of providing fast and effective method to manage this information. It should be flexible enough to allow some processing to be made, for example to compare a variable with a predefined value or to evaluate some status and, according to the result, decide whether a condition is met or not, namely the alarm is active or not. The most effective way to gather all of the necessary input information from the control system is to exploit the event-driven communication paradigm available within TANGO.

Overview

The Alarm Server is a Tango Device Server that evaluates alarm formulas containing Tango Attributes of other Device Servers. In order to get the last updated values of each Attribute present in formulas, the Alarm Server subscribes to Tango Change Events of these Attributes. The status of alarms is returned in an Attribute. Alarms with the corresponding formulas are loaded with a Tango Command.

As a consequence of a change state of an alarm, a Tango Command configured for this alarm can be executed.

The AlarmMail Device Server can be configured as the action associated to alarm rules to notify alarms via email.

TANGO device

Alarm device

The Alarm device is:

state	status	description
RUNNING	Alarm server is running	

spectrum attributes			
attribute name	data type	max length	R/W type
alarm: Read: array of alarm strings	DEV_STRING	1024	READ

commands		
command name	argin type	argout type
Ack: acknowledge all alarm names passed as argin	DEVVAR_STRING_ARRAY	DEV_VOID
Configured: return array of configured alarms filtered by name with argin	DEV_STRING	DEVVAR_STRING_ARRAY
Load: load a new alarm string	DEV_STRING	DEV_VOID
Modify: modify an existing alarm string	DEV_STRING	DEV_VOID
Remove: remove an alarm	DEV_STRING	DEV_VOID
Silence: silence all alarm names passed as argin: they won't trigger sound till the silence_time period configured for each alarm has expired	DEVVAR_STRING_ARRAY	DEV_VOID
StopNew: reset the "NEW" field in the alarm attribute	DEV_VOID	DEV_VOID

device properties		
property name	type	description
AlarmStatus	Array of string	Status of stored alarms for startup initialization
DbHost	string	Host for MySQL DB
DbUser	string	Username for MySQL DB
DbPasswd	string	Password for MySQL DB
DbName	string	DB name for MySQL DB
DbPort	string	Port for MySQL DB
ErrThreshold	DEV_LONG	Threshold for Tango errors for being logged as Internal Errors
InstanceName	string	Name used to associate configured alarm rules to this instance of the alarm server
GroupNames	Array of string	Labels for Group mask, first is for mask 0x00

Attributes:

- alarm

Read an array of alarm strings. It returns alarms that are in alarm table, possible status are ALARM/NACK, ALARM/ACK, NORMAL/NACK. Then there are the internal alarms that are alarms regarding the internal working of the server or error related to events (when there are problems with the event system due to network problems, servers which contains attributes to read are shutdown, notifd is not running, ...). If there aren't alarms to return, the attribute quality is set to “WARNING” and it is returned a string with useless values.

The alarm string is composed by some fields separated by '\t' character, the fields are the following:

- timestamp (seconds since 1/1/1970) of the instant when this alarm changed status
- micro seconds of the instant when this alarm changed status
- alarm name
- status (“NORMAL” or “ALARM”)
- acknowledge (“ACK” or “NACK”)
- count: if the status is “ALARM” the count field contains the number of subsequent events all evaluating the “ALARM” status, otherwise it is 0
- severity level (“fault”,“warning” or “log”)
- silence time: -1 if not silenced otherwise minutes of silenced state remaining (in which no new alarm condition can trigger sound alarm)
- group(s) (if more than one, groups are separated by the '| ' character)
- message (optional)
- new status: if it is the first time this alarm is read in the “ALARM” status, it is added the string “NEW” as the last field

example:

```
Read [0]   1205243098  154618  b/test/test/curr  NORMAL  NACK  0  log  -1  gr_ps  Curr!!
Read [1]   1204104018  719978  b/test/test/volt  ALARM  NACK  2  fault  5  gr_ctrl  Volt!! NEW
Read [2]   1204104018  83086  b/test/test/state  ALARM  NACK  1  log  -1  gr_id|gr_ps  state=FAULT!
Read [3]   1206954173  726704  internal_error_0  ALARM  NACK  3  log  -1  gr_none  Failed to read initial value of
sr/test/test/state = Device sr/test/test/state is not exported
Read [4]   1206954173  735599  internal_error_1  ALARM  NACK  0  log  -1  gr_none  Failed to read initial value of
sr/test/test/stat = Device sr/test/test/stat is not exported
```

Commands:

- Ack

  Acknowledge alarms that are present in alarm table. Possible argins are the alarm names returned by the “alarm” attribute. Alarms in status NORMAL/ACK are removed from the alarm table. Internal alarms are removed from the alarm table when are acknowledged.

- Configured

Read an array of configured alarms. Argin is a filter on the alarm name, if it is the null string, all alarms configured are returned. The string is composed by some fields separated by '\t' character, the fields are the following:

- timestamp of the instant when this alarm was added
- alarm name
- alarm formula
- time threshold (period of time during which the formula has to evaluate true and after which the alarm changes status to ALARM)
- severity level
- silence time in minutes, -1 if not possible to silence
- group(s) (if more than one, groups are separated by the '| ' character)
- message (optional)
- actions (two Tango commands separated by the ";" character)

example:

Read [0]   1205243094 b/sec/test/curr (b/test/test/current > 100) 15 log -1 gr_ps Curr!! b/sec/dev/notif_curr;

Read [1]   1204104016 b/sec/test/volt (b/test/test/voltage != 10) 10 fault 60 gr_ctrl Volt!! ;b/sec/dev/notif_volt_ok

Read [2]   1204105328 b/sec/test/state (b/test/test/state == FAULT) 0 log -1 gr_id|gr_ps state=FAULT! ;

- Load

  Load a new alarm. Argin is a string with the following fields separated by one or more spaces or tabs: alarm name, formula, time threshold (in seconds, optional, defaults to 0), level, silence time (in minutes, optional, default -1=not possible to silence), one or more groups between that configured in "GroupNames" Device Property, separated by "|", the message, actions to be executed (one or two full paths of Tango commands separated by ";", optional, defaults to no action).

examples:

1. ***b/sec/test/curr (b/sec/test/current > 100) log gr_ps|gr_ctrl "Alarm Curr!"***

   alarm name=b/sec/test/curr

   formula=(b/test/test/current > 100)

   time threshold=0 seconds

   level=log

   silence time=-1 minutes (not possible to silence)

   groups=gr_ps,gr_ctrl

   message="Alarm Curr!"

   actions=no action to be executed

   note: to pass this string as an argument in Jive the " character needs to be escaped, so this is the string to be used:

   "b/sec/test/curr (b/sec/test/current > 100) log gr_ps|gr_ctrl \"Alarm Curr!\""

2.

b/sec/test/dev ((b/sec/test/current > 100) && (b/sec/test/voltage > 20)) 30 log 60 gr_all “Alarm Device!” b/sec/dev/notif_alarm;b/sec/dev/notif_normal

alarm name=b/sec/test/dev

formula=((b/sec/test/current > 100) && (b/sec/test/voltage > 20))

time threshold=30 seconds (after 30 seconds formula evaluates true, state changes to ALARM)

level=log

silence time=60 minutes (during which no new alarm triggers sound alarm)

groups=gr_all

message=”Alarm Curr!”

actions=command b/sec/dev/notif_alarm is executed when changing state to ALARM

command b/sec/dev/notif_normal is executed when changing state to NORMAL

- Modify

  Same syntax as the Load command. Alarm name has to exist already. If the formula is not changed the existing alarm rule is updated, otherwise existing alarm rule is removed and the new one added

- Remove

  Remove an alarm from configuration. Argin is the alarm name.

- StopNew

  Reset the field containing “NEW” in the “alarm” attribute (used to trigger sound alarm on QTango Alarm GUI).

Device properties:

- AlarmStatus: contains an array of strings which is the alarm status at the instant of the stop of the server. When the server is started the status is loaded from this device property
- DbHost, DbUser, DbPasswd, DbName, DbPort are the configuration necessary to access the database
- ErrThreshold: is the threshold for events arriving with the error flag set before being logged as internal alarms
- InstanceName: is the name used to associate configured alarm rules to this instance of the server
- GroupNames: contains an array of string with the names to be used in the alarm formulas. First name is reserved for the name of the group none (that is the group with mask 0x0000). Group all (correspondig to the mask 0xffff) is named by default gr_all. Not more than 32 groups can be defined.

DB

MySQL db is used to log alarms history and for saving alarms configuration.

There are two tables: Alarms and Description.

Alarms

id_alarms	int unsigned not null auto_increment PRIMARY_KEY
time_sec	int unsigned not null
time_usec	int(6) unsigned not null
status	enum ('ALARM', 'NORMAL') not null
ack	enum ('ACK', 'NACK') not null
id_description	int unsigned not null
attr_values	varchar(1024)

In Alarm table every change of status and of ack is logged. When the status is “ALARM”, “attr_values” field contains a list of couples “event_name=value” separated by “;”, these are the values that have determined the “ALARM” status.

Description

id_description	int unsigned not null auto_increment PRIMARY_KEY
name	varchar(128) not null
instance	varchar(64) not null
active	int(1) unsigned not null
time_sec	int unsigned not null
formula	varchar(1024) not null
time_threshold	int unsigned
silent_time	int default -1
level	enum ('log', 'warning', 'fault') not null
grp	varchar(128) not null
msg	varchar(255)
action	varchar(255)

In Description table are logged alarm rules configured. Field “active” says if this rule is actually used of it was used in the past but is no more active.

Field instance says at which instance of the alarm server belongs this alarm.

Field action contains the actions to be executed when this alarm changes state. Actions are Tango commands expressed with the full path of the command. There are two possible actions: the first can be executed when changing to ALARM state, the second when changing to NORMAL state. This two actions are separated by the

";" character.

Formula

Formula can contain attribute names, numbers, Tango States, logical and mathematical operators. Expressions can be enclosed by round brackets.

- Attribute names have to be of four fields separated by "/", ex: b/dev/test/curr
- Numbers can be real numbers (ex: 35, -23.5, ...) or hexadecimal (ex: 0xaf, 0x1A)
- Tango States allowed are ON, OFF, CLOSE, OPEN, INSERT, EXTRACT, MOVING, STANDBY, FAULT, INIT, RUNNING, ALARM, DISABLE, UNKNOWN
- Operators allowed are the following in order of precedence: *, /, +, -, <<, >>, <= , >=, >, <, ==, !=, &, |, ^, &&, ||. Unary operators abs, ! are allowed

  example:

  *(b/dev/test1/val * 2.5 >= abs(b/dev/test2/val) && (b/dev/test3/state == OFF) || b/dev/test4/stat & 0xA0)*

  note: FQDN names are supported (tango://server:10000/a/b/c/state)

Actions

For each alarm two actions can be configured: a Tango Command to be executed when the state changes to ALARM, and one to be executed when the state changes to NORMAL. These two commands can be specified as the last field of the alarm string given to the "Load" command and have to be separated by the ";" character. This field isn't required so to the Load command the following can be passed:

1.

   b/sec/test/curr (b/sec/test/current > 100) 0 log gr_all "Alarm Curr! b/sec/dev1/action;b/sec/dev2/action"

   actions=command b/sec/dev1/action is executed when changing state to ALARM

   command b/sec/dev2/action is executed when changing state to NORMAL

2.

   b/sec/test/curr (b/sec/test/current > 100) 0 log gr_all "Alarm Curr!" ***b/sec/dev1/action;***

   actions=command b/sec/dev1/action is executed when changing state to ALARM

   no command is executed when changing state to NORMAL

3.

   b/sec/test/curr (b/sec/test/current > 100) 0 log gr_all "Alarm Curr!" ***;b/sec/dev2/action***

   actions=no command is executed when changing state to ALARM

   command b/sec/dev2/action is executed when changing state to NORMAL

4.

   b/sec/test/curr (b/sec/test/current > 100) 0 log gr_all "Alarm Curr!" ***;***

   actions=no command is executed when changing state to ALARM

   no command is executed when changing state to NORMAL

5.

   b/sec/test/curr (b/sec/test/current > 100) 0 log gr_all "Alarm Curr!"

   actions=no command is executed when changing state to ALARM no command is executed when changing state to NORMAL

The Tango Commands used as actions have to be either: with Tango::DEV_VOID or Tango::DEV_STRING as argin type. In the second case the value passed to the command by the Alarm Server is made by concatenating the alarm rule name, the alarm groups, the alarm message, and the values of the attributes involved in the formula in the form attr_name=value comma separated and the formula. The separator character is ";".

example:

for the alarm loaded with the following

b/sec/test/dev ((b/sec/test/current > 100) && (b/sec/test/voltage > 20)) 30 log gr_all "Alarm Device!" b/sec/dev/notif_alarm;b/sec/dev/notif_normal

the value passed to b/sec/dev/notif_alarm if it accepts a Tango::DEV_STRING as input would be

name=b/sec/test/dev;groups=gr_all;msg=AlarmDevice!;values=b/sec/test/current=101.75,b/sec/test/voltage=22.6;formula=((b/sec/test/current > 100) && (b/sec/test/voltage > 20))

AlarmMail device

The AlarmMail device is:

commands		
command name	**argin type**	**argout type**
SendAlarm: send email to notify Alarm state	DEV_STRING	DEV_VOID
SendNormal: send email to notify Normal state	DEV_STRING	DEV_VOID

device properties		
property name	**type**	**description**
email_name	string	Name used as sender in the email
email_smtp	string	SMTP server address
email_user	string	Sender email address
receivers	Array of string	Array of group (as defined in GroupNames Tango Alarm device property)= comma separated email addresses *example: gr_ctrl=name1@emal.com,name2@email.com*

Commands:

- SendAlarm

  Send emails to notify Alarm state. It expects the argin string to be formatted as key1=value1;key2=value2;...

  Used keys are:

  - name: alarm rule name
  - groups: comma separated list of group names
  - msg: message associated to alarm

  - values: comma separated list of attribute name=value that triggered the alarm state change
  - formula: formula of the alarm

  Example of the email sent:

```
NEW ALARM DETECTED:
Name:    archiving/hdb++archiver2/llrf-kg04/attributenoknumber
Message: At least one attribute in archiving error
Values:  archiving/hdb++archiver2/llrf-kg04/attributenoknumber[0]=35,
Formula: (archiving/hdb++archiver2/llrf-kg04/attributenoknumber > 0)
```

- SendNormal

  Send emails to notify Normal state. It expects the argin string to be formatted with the same format as for the SendAlarm command

  Example of the email sent:

```
RESTORED TO NORMAL CONDITION:
Name:    archiving/hdb++archiver2/llrf-kg04/attributenoknumber
Message: At least one attribute in archiving error
Values:  archiving/hdb++archiver2/llrf-kg04/attributenoknumber[0]=0,
Formula: (archiving/hdb++archiver2/llrf-kg04/attributenoknumber > 0)
```