Tip of the Month :

- How to use the Local and Central Alarm Servers

revisit “Tip of the Week” from March 28, 2008

TINE Alarms
(a short review)

- An alarm belongs to a ‘device’
- An alarm has a timestamp
  - Last alarm signal
- An alarm has a start time
  - First alarm signal
- An alarm has a code
  - defines the alarm
- An alarm can have data
  - Up to 64 total bytes (any format)

- Alarm identified by
  - /context/server/device + code + starttime

TINE Alarms

- An alarm can be persistent
  - Always there (until someone takes care of it)
  - e.g. hardware readout error
- An alarm can be transient
  - A change of state from okay to not okay
  - e.g. beam dump, quench, RF trip
- An alarm can be oscillating
  - There for a while then not there then back again.
  - e.g. intermittent hardware error (a flickering sedac error).

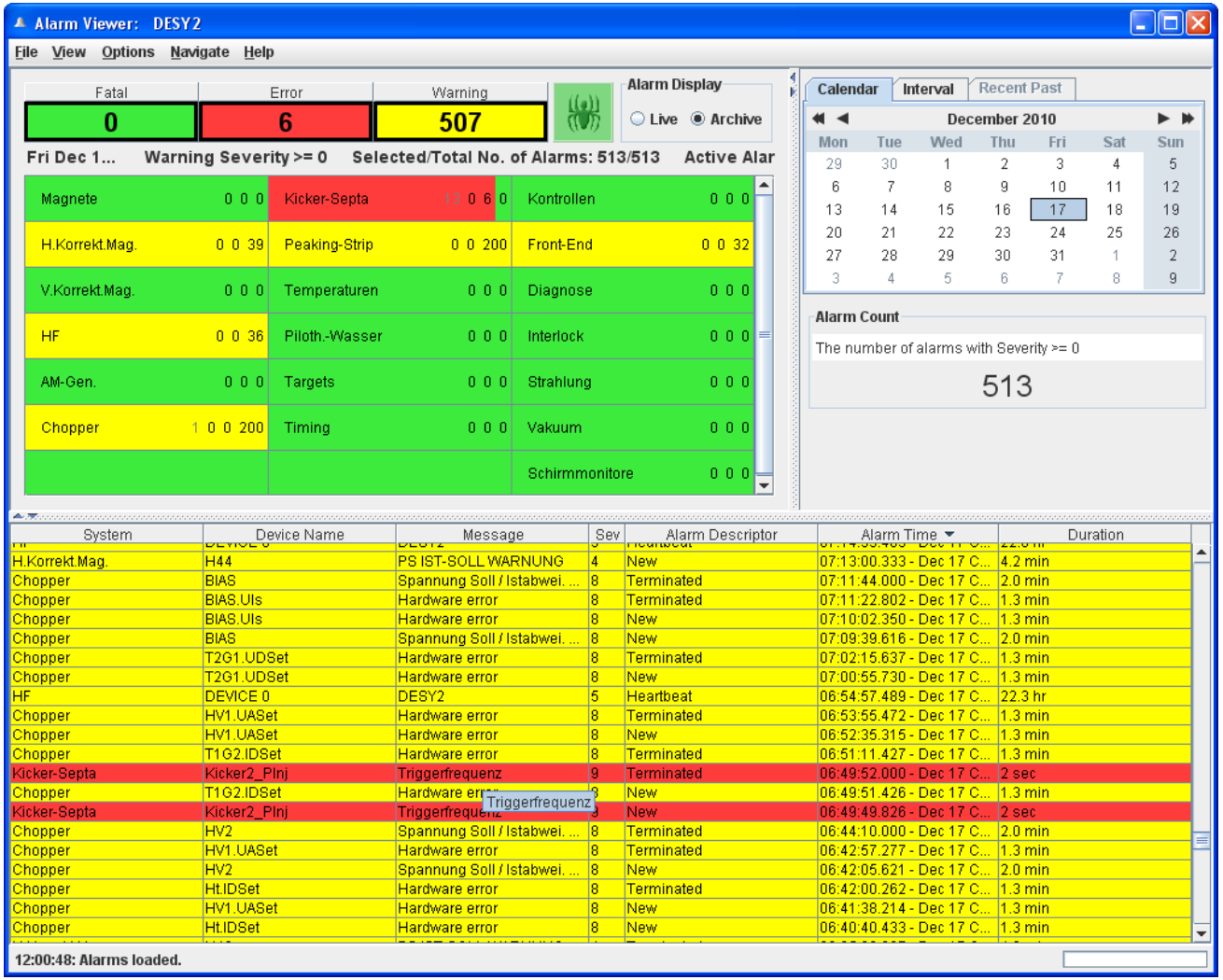
TINE Alarm Definition

- Alarm Code cross references static alarm information
  - Severity
    - (Release 4.0: can override dynamically !)
  - System ID
    - Usually = 0 (=> let the CAS decide which system)
  - Tag
    - Short description of alarm
  - Alarm Text, Device Text, Data Text, url
  - Data Type and Size

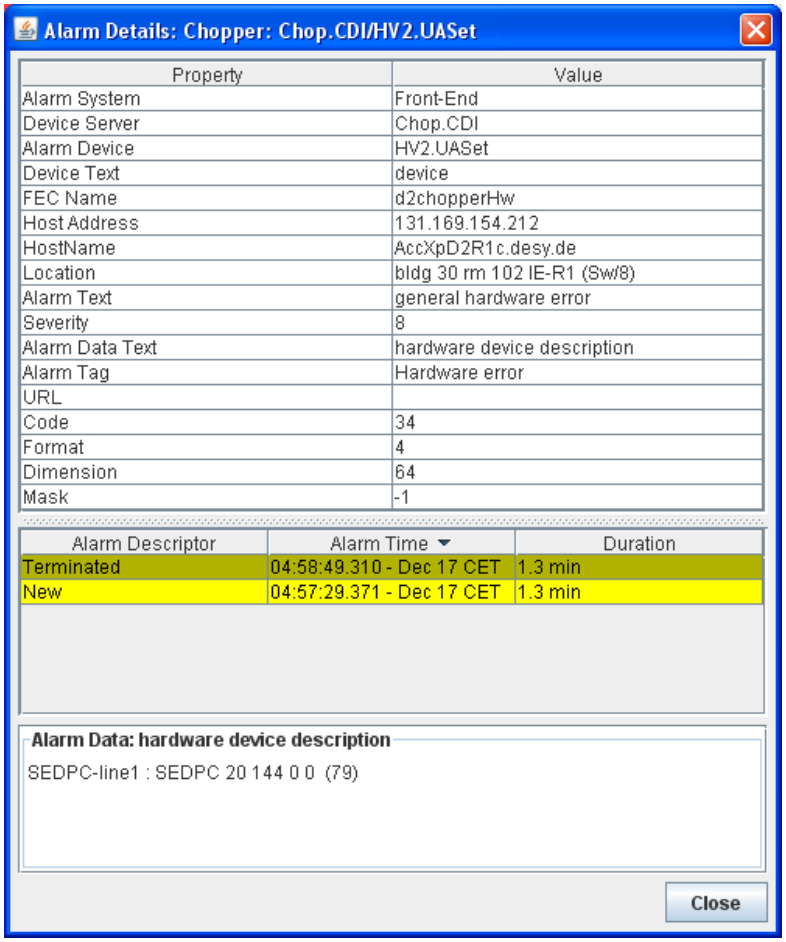
TINE Alarm Severities

- Range of 0 to 15
  - 0 = test alarm
  - 15 = you can't possible have operations unless you fix this
  - Typically:
    - 0 => not really an alarm (not handled at CAS)
    - 1 – 6 => information (not archived)
    - 7 – 12 => warning
    - > 12 => fatal

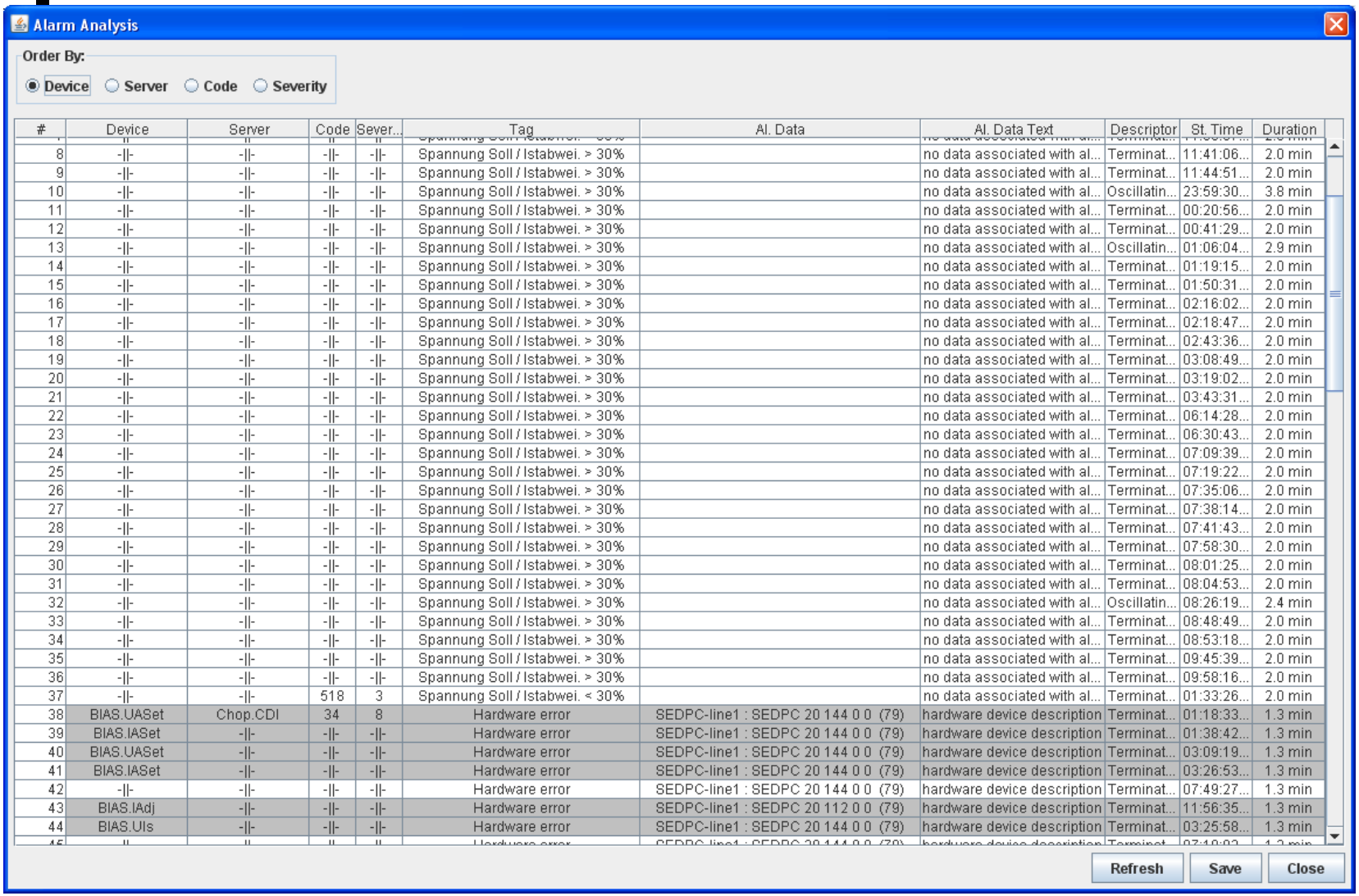
The New Alarm Viewer :



The New Alarm Viewer :



The New Alarm Viewer :



#	Device	Server	Code	Sever...	Tag	Al. Data	Al. Data Text	Descriptor	St. Time	Duration
8	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	11:41:06...	2.0 min
9	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	11:44:51...	2.0 min
10	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Oscillatin...	23:59:30...	3.8 min
11	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	00:20:56...	2.0 min
12	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	00:41:29...	2.0 min
13	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Oscillatin...	01:06:04...	2.9 min
14	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	01:19:15...	2.0 min
15	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	01:50:31...	2.0 min
16	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	02:16:02...	2.0 min
17	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	02:18:47...	2.0 min
18	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	02:43:36...	2.0 min
19	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	03:08:49...	2.0 min
20	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	03:19:02...	2.0 min
21	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	03:43:31...	2.0 min
22	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	06:14:28...	2.0 min
23	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	06:30:43...	2.0 min
24	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	07:09:39...	2.0 min
25	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	07:19:22...	2.0 min
26	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	07:35:06...	2.0 min
27	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	07:38:14...	2.0 min
28	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	07:41:43...	2.0 min
29	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	07:58:30...	2.0 min
30	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	08:01:25...	2.0 min
31	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	08:04:53...	2.0 min
32	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Oscillatin...	08:26:19...	2.4 min
33	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	08:48:49...	2.0 min
34	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	08:53:18...	2.0 min
35	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	09:45:39...	2.0 min
36	-||-	-||-	-||-	-||-	Spannung Soll / Istabwei. > 30%		no data associated with al...	Terminat...	09:58:16...	2.0 min
37	-||-	-||-	518	3	Spannung Soll / Istabwei. < 30%		no data associated with al...	Terminat...	01:33:26...	2.0 min
38	BIAS.UASet	Chop.CDI	34	8	Hardware error	SEDPC-line1 : SEDPC 20 144 0 0 (79)	hardware device description	Terminat...	01:18:33...	1.3 min
39	BIAS.IASet	-||-	-||-	-||-	Hardware error	SEDPC-line1 : SEDPC 20 144 0 0 (79)	hardware device description	Terminat...	01:38:42...	1.3 min
40	BIAS.UASet	-||-	-||-	-||-	Hardware error	SEDPC-line1 : SEDPC 20 144 0 0 (79)	hardware device description	Terminat...	03:09:19...	1.3 min
41	BIAS.IASet	-||-	-||-	-||-	Hardware error	SEDPC-line1 : SEDPC 20 144 0 0 (79)	hardware device description	Terminat...	03:26:53...	1.3 min
42	-||-	-||-	-||-	-||-	Hardware error	SEDPC-line1 : SEDPC 20 144 0 0 (79)	hardware device description	Terminat...	07:49:27...	1.3 min
43	BIAS.IAdj	-||-	-||-	-||-	Hardware error	SEDPC-line1 : SEDPC 20 112 0 0 (79)	hardware device description	Terminat...	11:56:35...	1.3 min
44	BIAS.UIs	-||-	-||-	-||-	Hardware error	SEDPC-line1 : SEDPC 20 144 0 0 (79)	hardware device description	Terminat...	03:25:58...	1.3 min

Alarm Configuration

- Static information from either
  - ‘alarms.csv’ (relative to the equipment module)
  - ‘fec.xml’ (the <ALARM_DEFINITION> tag)
  - API Calls (java device server wizard)

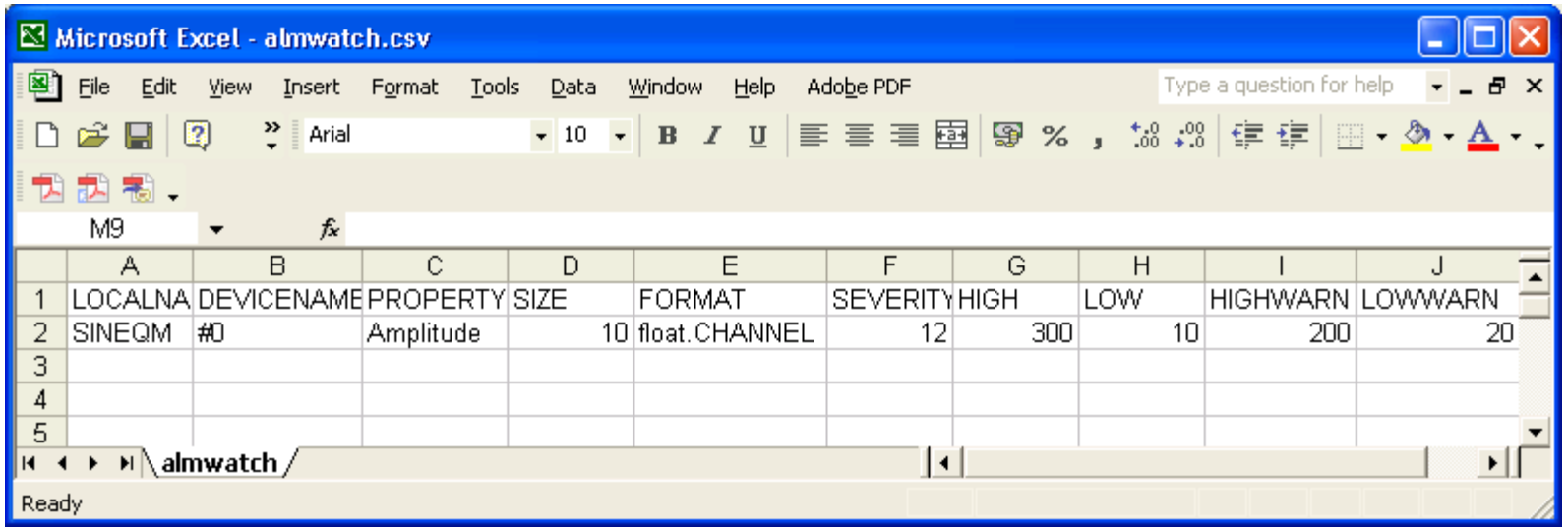
Automatic Alarm Generation

- An Alarm Watch Table
  - Threshold alarms
    - readback is out of bounds !
    - value_too_high, value_too_low, warn_too_high, warn_too_low
  - Value Mask alarms
    - masked readback does not match the ! ‘normal’ value
    - invalid_data
- Configure via either
  - ‘almwatch.csv’ (relative to the equipment module) or
  - ‘fec.xml’ (the <ALARM> tag) or
  - API Calls
- You do NOT have to Set/Clear these alarms yourself!

Alarm Watch Table

almwatch.csv

Instruct the Local Alarm Server which properties should be monitored and where the thresholds are :



	A	B	C	D	E	F	G	H	I	J
1	LOCALNA	DEVICENAME	PROPERTY	SIZE	FORMAT	SEVERITY	HIGH	LOW	HIGHWARN	LOWWARN
2	SINEQM	#0	Amplitude	10	float.CHANNEL	12	300	10	200	20
3										
4										
5										

Alarm Watch Table

```
- <PROPERTY>
    <NAME>Sine</NAME>
    <DEVICE_SET />
    <EGU>V</EGU>
    <XEGU>r</XEGU>
    <MAX>1000</MAX>
    <MIN>0</MIN>
    <XMAX>8092</XMAX>
    <XMIN>0</XMIN>
    <ID>0</ID>
    <DESCRIPTION>Sine Curve</DESCRIPTION>
    <SIZE_IN>0</SIZE_IN>
    <DTYPE_IN>null</DTYPE_IN>
    <SIZE_OUT>1024</SIZE_OUT>
    <DTYPE_OUT>float.SPECTRUM</DTYPE_OUT>
    <ACCESS>READ</ACCESS>
    <REDIRECTION />
  - <ALARM>
      <DEVICE_NAME>SineGen0</DEVICE_NAME>
      <SEVERITY_HIGH>12</SEVERITY_HIGH>
      <SEVERITY_LOW>12</SEVERITY_LOW>
      <SEVERITY_HIGH_WARN>10</SEVERITY_HIGH_WARN>
      <SEVERITY_LOW_WARN>10</SEVERITY_LOW_WARN>
      <SYSTEM>100</SYSTEM>
      <VALUE_MASK />
      <VALUE_NORMAL />
      <COUNT_THRESHOLD>3</COUNT_THRESHOLD>
      <VALUE_HIGH>800</VALUE_HIGH>
      <VALUE_LOW>50</VALUE_LOW>
      <VALUE_HIGH_WARN>700</VALUE_HIGH_WARN>
      <VALUE_LOW_WARN>100</VALUE_LOW_WARN>
    </ALARM>
  - <HISTORY>
      <DEVICE_NAME>SineGen0</DEVICE_NAME>
      <TOLERANCE>10%</TOLERANCE>
```

fec.xml

Can also mask a readback value and compare versus a ‘normal’ value

Alarm Watch Table

API :

```
int AppendAlarmWatchTable ( char *              eqm,
                            char *              prp,
                            char *              dev,
                            int                 siz,
                            int                 fmt,
                            int                 atyp,
                            int                 sev,
                            int                 sys,
                            ALM_THRESHOLDS *    thr
                          )
```

Inserts a property to be monitored into the local alarm server's Watch Table.

Certain alarms are to be set whenever the value of a property exceeds a defineable threshold. Such alarms can be managed automatically by the local alarm server if the alarm criteria are entered into the alarm watch table. This can be achieved by calling this routine (or supplying a startup configuration file almwatch.csv).

Parameters:

eqm is the 6-character local equipment identifier name, which is internal to server.
prp is the property which is to be called by the local alarm server
dev is the device name associated with the property to be called by the local alarm server
siz is the data array size to be called by the local alarm server
fmt is the TINE data format to be called by the local alarm server
atyp is the TINE data array type to be applied to the property called by the local alarm server
sev is the severity of the alarm issued when the data returned by the call exceed the given thresholds.
sys is the alarm system identifier to be associated with the alarm.
thr is an **ALM_THRESHOLDS** object specifying the threshold criteria for setting the alarm

Note: Information can also be entered into the alarm watch table by supplying the startup configuration file almwatch.csv, where the input parameters are given in the relevant columns. This is frequently the preferred way of inputting such alarm information, since if does not involve hard-coding such alarm criteria.

Returns:

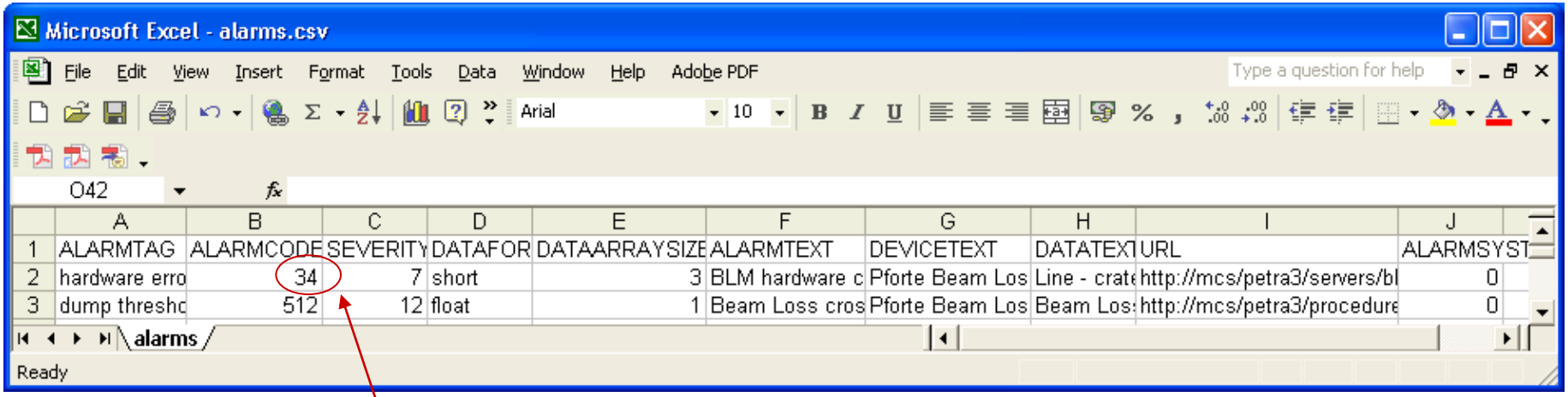
0 upon success, otherwise a TINE error code.

Alarm API :

- Make sure your alarms are defined !
  - alarms.csv (or fec.xml, or API)
- Make use of ClearAlarm()/SetAlarm() inside your I/O loop.
  - ClearAlarm() at the start of the loop
    - Increments the ‘clear counter’
  - If the alarm is still active then SetAlarm()
    - Resets the ‘clear counter’

Alarm Definitions

alarms.csv :



#definition from errors.h

Alarm Definitions

fec.xml :

```
- <EQM>
    <NAME>BLMEQM</NAME>
    <SERVER>BLM</SERVER>
    <CONTEXT>PETRA</CONTEXT>
    <SUBSYSTEM>DIAG</SUBSYSTEM>
    <GROUP />
    <GROUP_INDEX />
  - <ALARM_DEFINITION>
      <TAG>hardware error</TAG>
      <DATA_FORMAT>short</DATA_FORMAT>
      <ALARM_TEXT>BLM hardware cassette readback error</ALARM_TEXT>
      <DEVICE_TEXT>Pforte Beam Loss Monitor</DEVICE_TEXT>
      <DATA_TEXT>Line - crate - subaddress</DATA_TEXT>
      <URL>http://mcs/petra3/servers/blm/hardware-trouble.html</URL>
      <ALARM_CODE>34</ALARM_CODE>
      <ALARM_MASK />
      <SEVERITY>7</SEVERITY>
      <ALARM_SYSTEM />
      <DATA_SIZE>3</DATA_SIZE>
    </ALARM_DEFINITION>
  - <ALARM_DEFINITION>
      <TAG>dump threshold reached</TAG>
      <DATA_FORMAT>flot</DATA_FORMAT>
      <ALARM_TEXT>Beam Loss crossed alarm threshold and triggered dump</ALARM_TEXT>
      <DEVICE_TEXT>Pforte Beam Loss Monitor</DEVICE_TEXT>
      <DATA_TEXT>Beam Loss (cnts/sec) that crossed the alarm threshold</DATA_TEXT>
      <URL>http://mcs/petra3/procedures/restart-following-dump.html</URL>
      <ALARM_CODE>512</ALARM_CODE>
      <ALARM_MASK />
      <SEVERITY>12</SEVERITY>
      <ALARM_SYSTEM />
      <DATA_SIZE>3</DATA_SIZE>
    </ALARM_DEFINITION>
```

Alarm Definitions

API Call :

```
int AppendAlarmInfoTable ( char * eqm,
                           ADS * ads
                         )
```

Inserts an alarm definition into the alarm definition table.

As an alternative to the <local name>-alarms.csv configuration file, the front end server can make use of this API call in order to fill in the alarm definition table describing locally generated alarms. This is particularly useful for embedded platforms where there is no file system, or where a TINE server is used as a translation layer and needs to map a given alarm system onto the TINE alarm system.

Parameters:

eqm is the 6-character local equipment identifier name, which is internal to server.
ads is a pointer to an Alarm Definition Structure (ADS) containing the alarm table information which is to be appended to the alarm definition table.

Returns:

0 upon success, otherwise a TINE error code.

```
strncpy(ads.alarmTag,"Threshold exceeded",16);
ads.alarmCode = 512;
ads.alarmMask = 0xff;
ads.severity = 7;
ads.alarmDataFormat = BFMT(CF_FLOAT);
ads.alarmDataArraySize = 1;
ads.alarmSystem = 640;
strncpy(ads.alarmText,"Take action immediately!",40);
strncpy(ads.dataText,"Current threshold setting",40);
strncpy(ads.url,"alarm512.html",40);
if ((cc=appendAlarmInfoTable(EQPMODNAME,&ads)) != 0)
{
  feclog("appendAlarmInfoTable : %s",erlst[cc]);
}
```

Java: from TEquipmentModule

```
public int addAlarmDefinition(int code,TAlarmDefinition adef)
```

Alarm API Example : C

```
#define dump_threshold_reached 512
void hdwIo(void)
{
  int i;
  ClearAlarm(BLMEQM_TAG,-1); // clear them all

  for (i=0; i<gNumBlms; i++)
  {
    if ((cc=rdBlmLoss(i)) != 0)
    { // there was a readout problem
      SetAlarm(BLMEQM_TAG,i,hardware_error,blm[i].addr);
      continue;
    }
    if (blm[i].loss > blm[i].dumpThreshold)
    {
      SetAlarm(BLMEQM_TAG,i,dump_threshold_reached,blm[i].loss);
    }
  }
}
```

Pass the hardware address that caused the problem

Pass the readback value that crossed the threshold

Alarm API : java

Many convenient constructors for setAlarm()

```
}
/**
 * \brief sets an alarm with the given data set
 *
 * @param code The alarm code to apply to the alarm
 * @param data The alarm data associated with the alarm
 * @return
 */
public TAlarm setAlarm(int code,TDataType data)
{
  return setAlarm(code,data.getDataBuffer(),TAlarmDescriptor.NEW);
}
/**
 * \brief sets an alarm with the given data set
 *
 * @param code The alarm code to apply to the alarm
 * @param data The alarm data associated with the alarm
 * @return
 */
public TAlarm setAlarm(int code,float data)
{
  float[] fdata = new float[1];
  fdata[0] = data;
  TDataType td = new TDataType(fdata);
  return setAlarm(code,td.getDataBuffer(),TAlarmDescriptor.NEW);
}
/**
 * \brief sets an alarm with the given data set
 *
```

Alarm API : java

```
/**
 * Called in the background task.
 *
 * Note the calls to the local alarm server.
 * clearAlarm() increments the clear counter and is called when the routine is entere
 * setAlarm() is called when an alarm state is detected.
 * This is just an example.  Alarms for 'value too high', etc. are better handled
 * through the use of an 'almwatch.csv' configuration file.
 *
 */
public void update()
{
  clearAlarm(512);
  incrementPhase();
  for (int i = 0; i < 1024; i++)
    myData[i] = amplitude * Math.sin(phase + frequency * 6.28 * ((double) i / 1024.0)
  if (amplitude > 100) setAlarm(512);  // amplitude too high !
}
public double getPhase()
{
  return phase;
```

Some Notes

- ClearAlarm()
  - does not remove the alarm
  - Increments the clear counter
  - SetAlarm() resets the clear counter
  - If clear counter increase by more than 1 prior to the next SetAlarm() the alarm is marked as ‘oscillating’
  - Clear counter > 8 => alarm has terminated !
- Transient Alarms need to call SetAlarmEx()
  - Can pass the alarm flag ‘almINSTANT’
  - Immediately flagged as terminated
- Alarms stay in the local alarm table for the duration of the Alarm Termination Window (default = 30 seconds)
  - Longer if a configured CAS has not read the alarm
- CAS can react to (configured) alarm signals
  - Trigger events
  - Send email
  - Send to central logger

Reading Alarms from a Server

- Best Practice:
  - Monitor the Alarms ‘Snapshot’
  - Stock Property “NALARMS”
    - 5 long integer values
      - Total number of alarms
      - UTC Timestamp of the most recent
      - Highest severity
      - Number at the most recent timestamp
      - Number at the highest severity
  - Can incrementally update an alarm cache using this snapshot (CAS)
    - Use DATACHANGE mode
    - If something’s different : get the most recent alarm set.
- Alarm Viewer gets all alarm information from the CAS !
- Java : TAlarmSystem query class with lots of static methods to get alarm information !

Reading Alarms in java

You probably won't be doing this, but just for fun :

- TLink lnkNalms = TAlarmSystem.monitorNumberOfAlarms("PETRA",null,"RF",**12**,cbNalms)
  - Monitors number of alarms from the PETRA CAS, alarm subsystem "RF" with severity >= 12
- TLink lnkNalms = TAlarmSystem.monitorNumberOfAlarms("PETRA","ELWIS3",null,**12**,cbNalms)
  - Monitors number of alarms from server "ELWIS3" with severity >= 12 directly
- Inside cbNalms(lnk) :
  - timeAlm = lnk.getLastTimeStamp();
  - TAlarmMessage[] almsNew = getAlarms("PETRA","RF",timeAlm,timeNow,12);
  - Join or filter the 'almsNew' list to a cached alarm list.
    - e.g. remove anything older than 1 hour
    - e.g. don't include 'terminated' alarms in the list
    - etc.