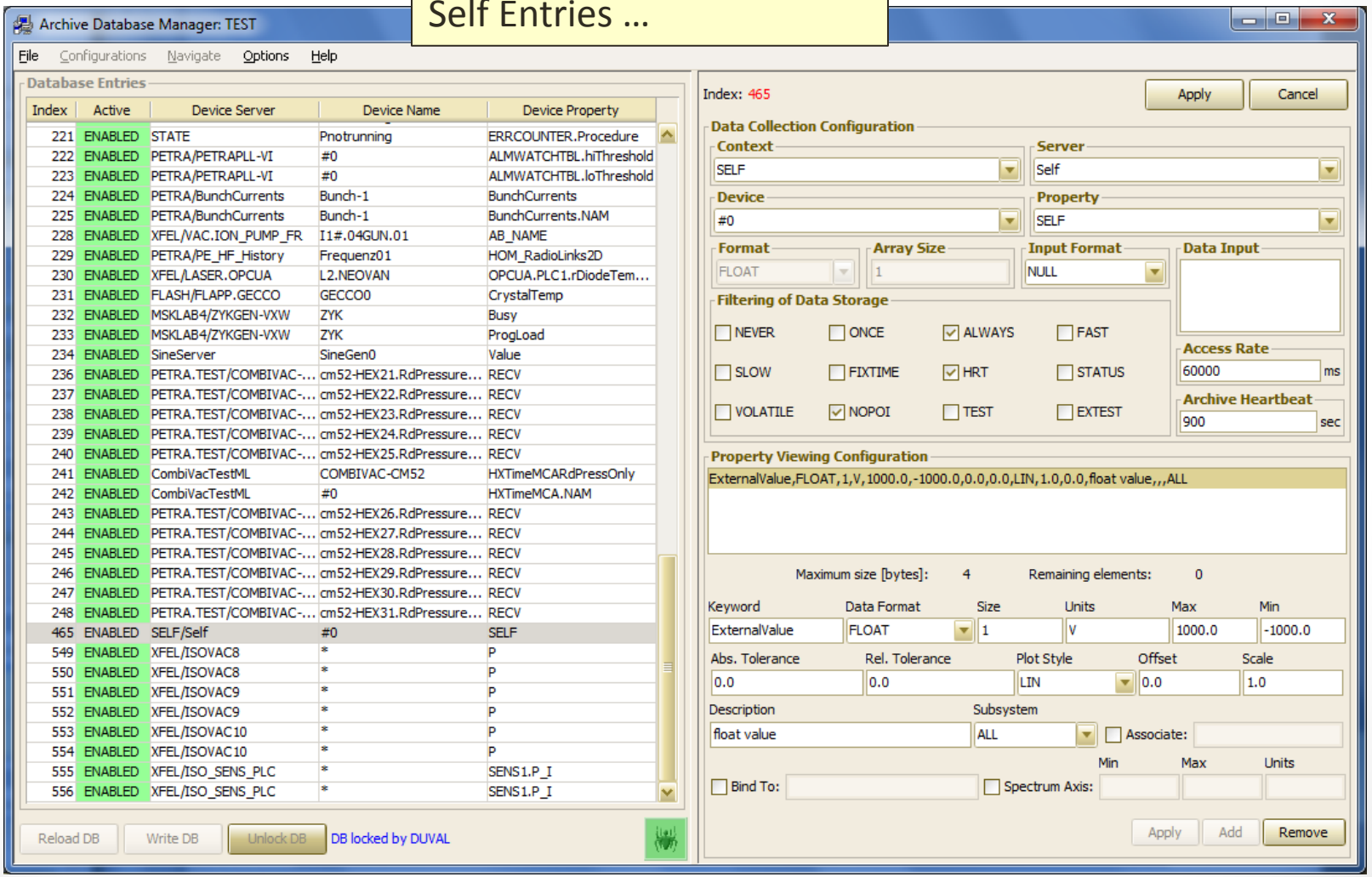
TINE Studio News

June 12, 2019

Archives, Alarms, and other interesting tales …

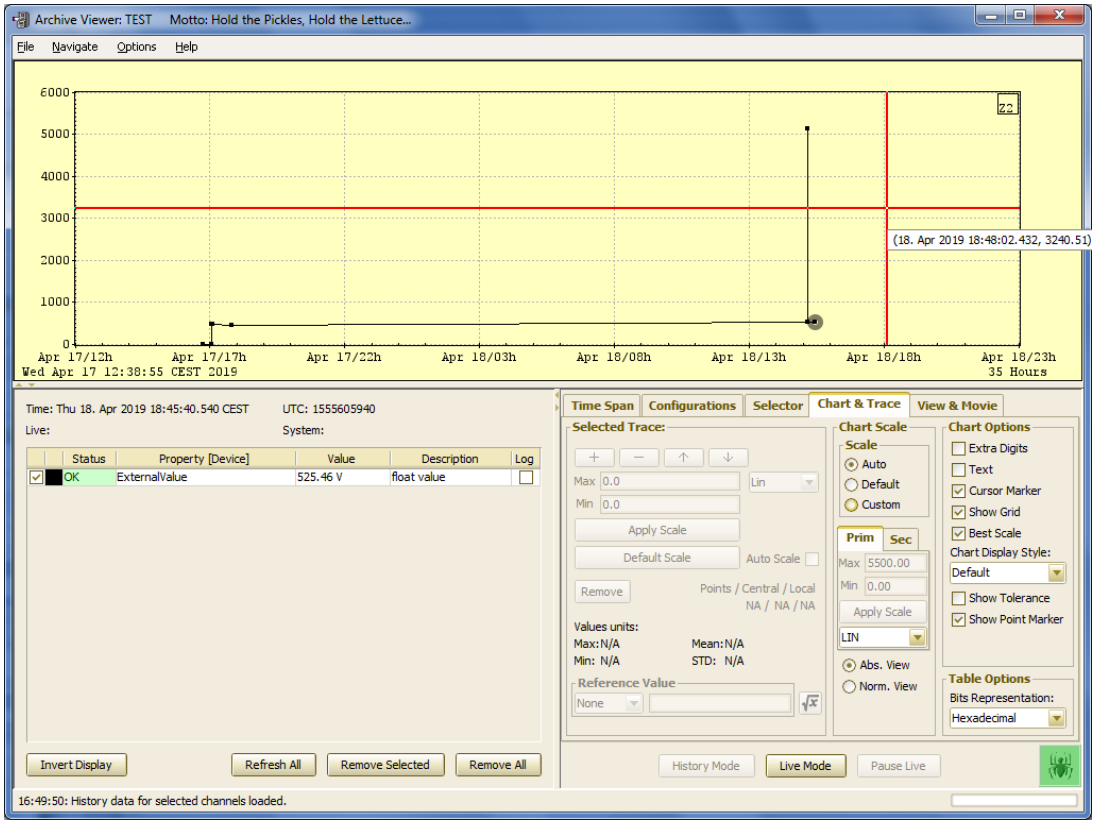
Archive Viewer

Self Entries …



Archive Viewer

- Send ‘WRITE’ call with property = KEYWORD (here: ‘ExternalValue’) and with the data, timestamp, and system stamps to archive :



Archive Viewer

```
int PutArchivedData ( char *  devsrv,
                      BYTE *  data,
                      int     dataFmt,
                      int     num,
                      double  dataTime,
                      int     sysStamp,
                      int     usrStamp
                    )
```

puts data and timestamps into the central archive system for 'SELF' entries

Supported data types for self entries include (arrays of) any numerical type and nothing else.

Parameters:

devsrv	must be the full device server name for the target central archiver. The context is primarily used.
data	is a reference to the data array to be sent to the archive server.
dataFmt	is the format of the data passed
num	is the number of elements in the data array reference
dataTime	is the UTC double timestamp of the data passed.
sysStamp	is the system stamp of the data passed.
usrStamp	is the user stamp of the data passed.

Returns:

0 if successful, otherwise a TINE completion code which can be interpreted by a call to **GetLastLinkError()**.

References **DUNION::bptr**, **DTYPE::dArrayLength**, **DTYPE::data**, **DTYPE::dFormat**, and **ExecLinkEx()**.

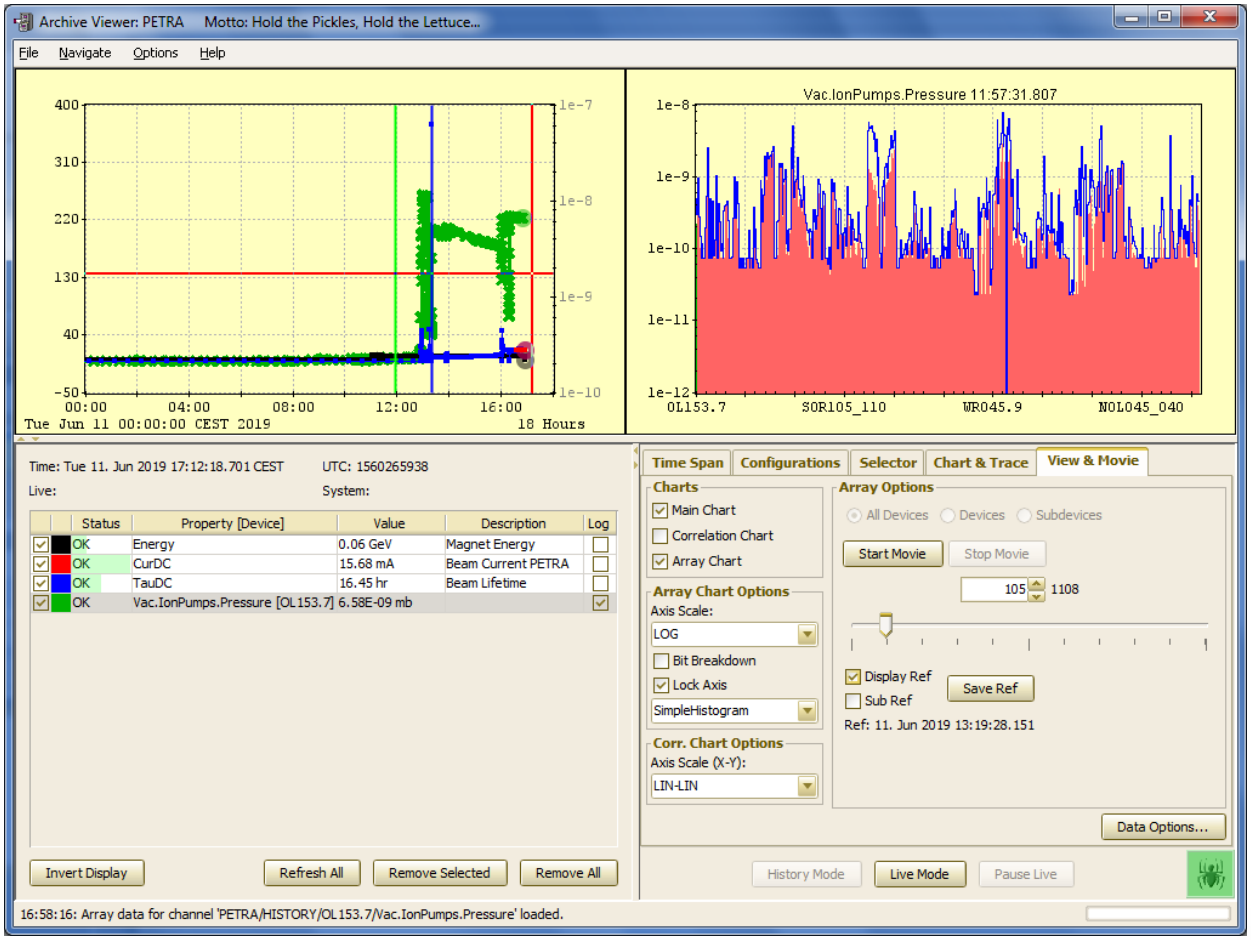
```
./tputhistory

Puts data and timestamps into the central archive system for 'SELF' entries
Usage :
        e.g. tputhistory PETRA MySelfEntry 1 2 3 4 5 /t=now /s=42 /u=1

        Usage : tputhistory <context> <keyword> [/t=<timestamp> /s=<systemstamp> /u=<userstamp>]
```

Archive Viewer

- MCA and Trace Records …
  - Can save an entire array of something at a given time (and this is more than useful !) …

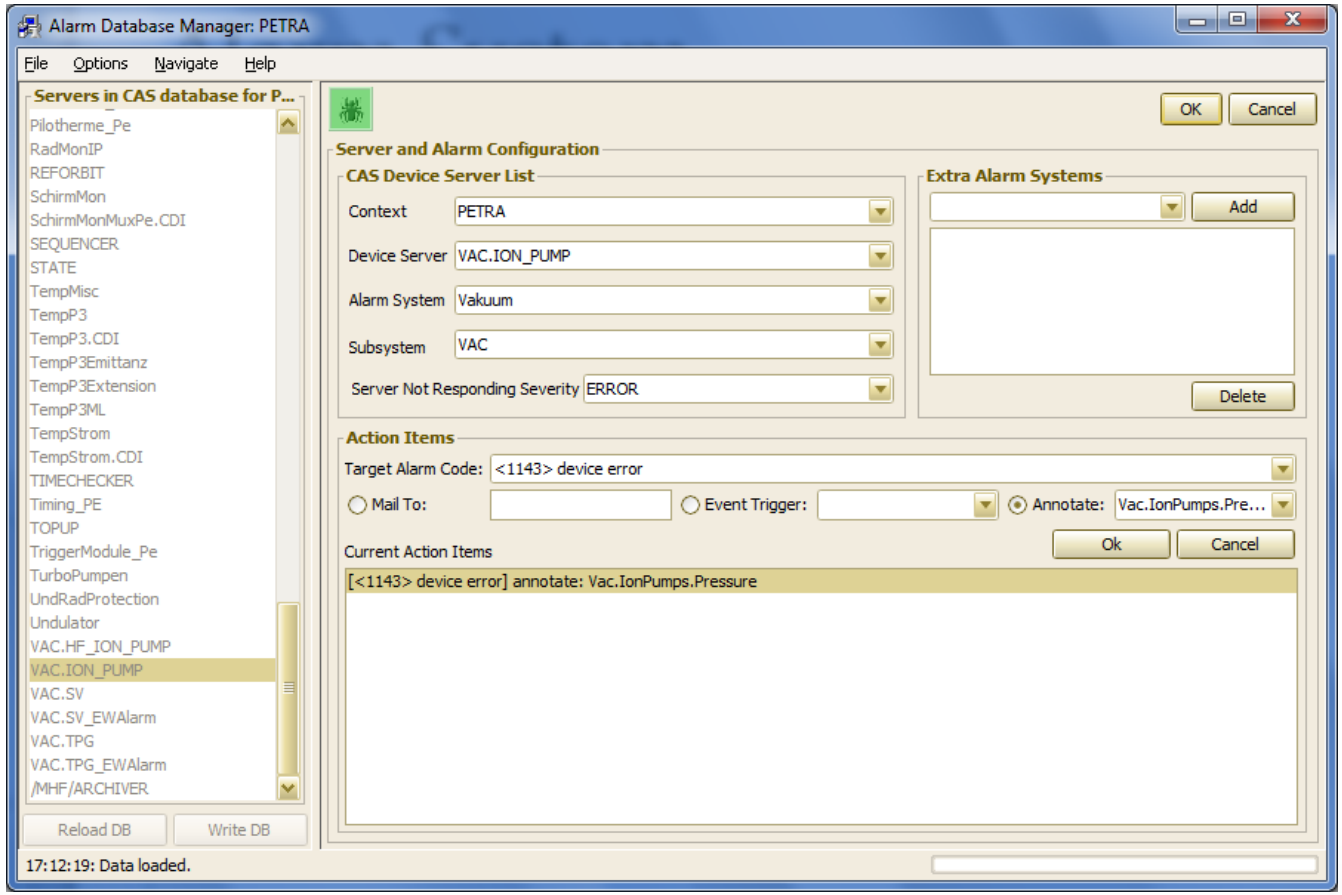


Archive Viewer

- MCA : an array of all devices of some type with the same settings and units.
  - But: need to get the channel names correctly!
    - Common problems:
      - someone removes or adds a BPM, BLM, Vacuum Pump, etc. in the middle. (declare as ‘volatile’ in the Archive Database Manager).
      - The MCA grows or shrinks in length (add some padding to the record).
    - These issues are easily dealt with
  - The latest twist:
    - New MCA Keyword ‘Keyword_A’ needs to supersede and map to N previous Keywords (e.g. ‘Keyword_1’, ‘Keyword_2’, and ‘Keyword_3’)
      - Can make use of the ‘volatile’ mapping logic already in place if the central archive knows how to map ‘Device_1A’ from ‘Keyword_A’ to ‘some_device’ from one of ‘Keyword_1’, ‘Keyword_2’, or ‘Keyword_3’) prior to some date.
      - Implemented and tested!

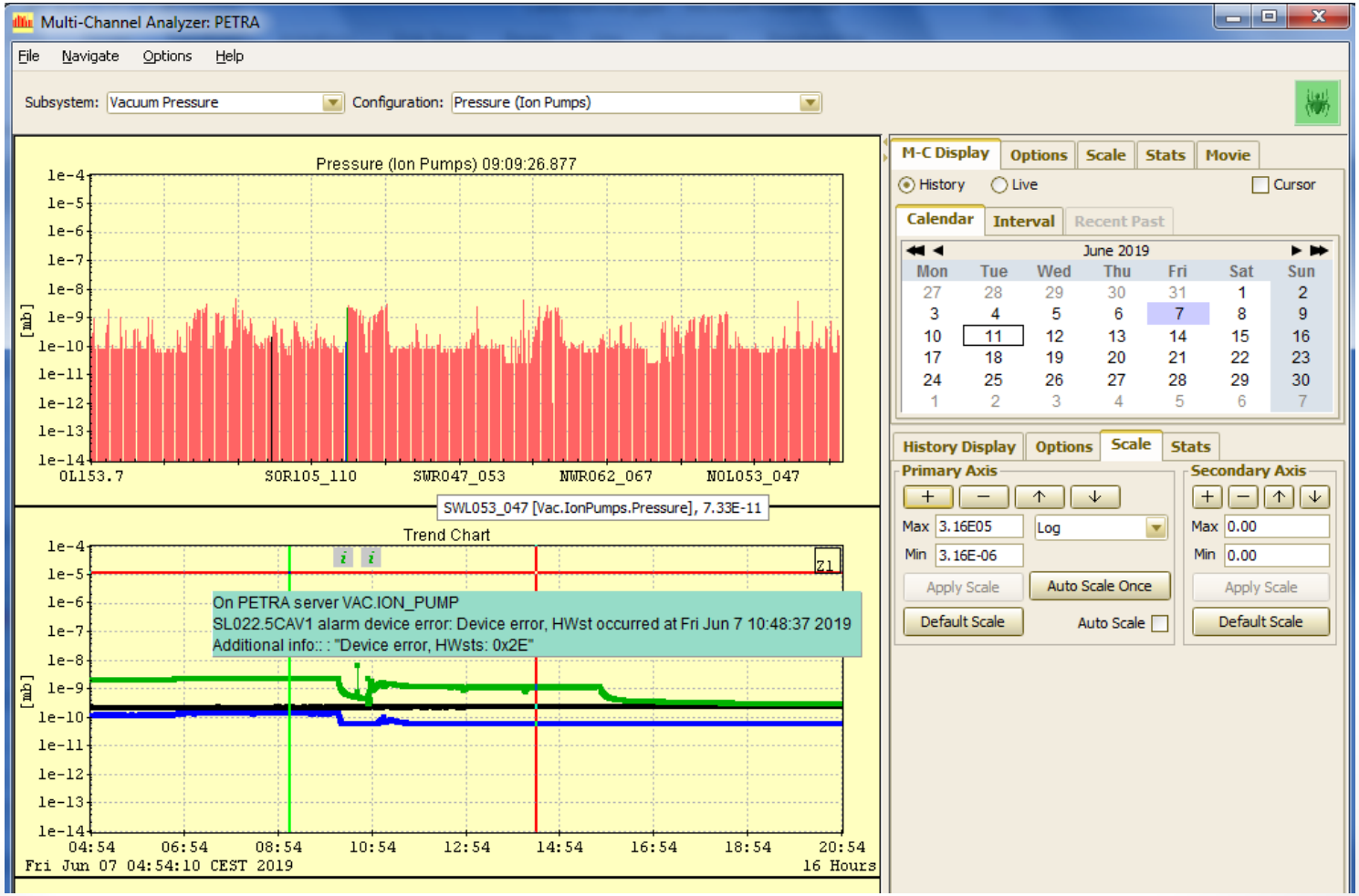
Alarm System

- (We should have thought of this years ago !)
- New Action Item for a particular alarm : **Archive Annotations !**



Alarm System

- Multi-Channel Analyzer now features ‘annotations’ :

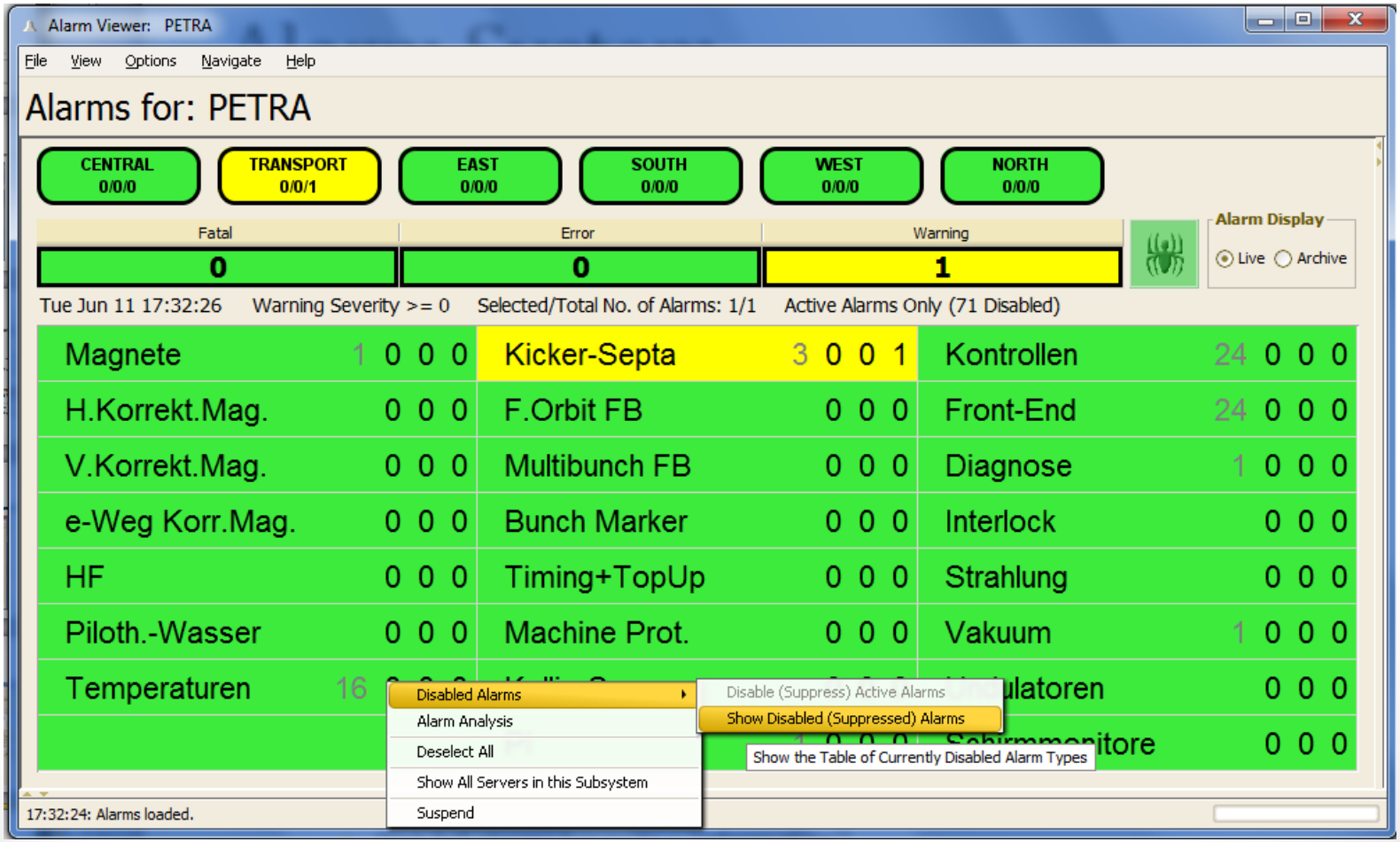


Alarm System

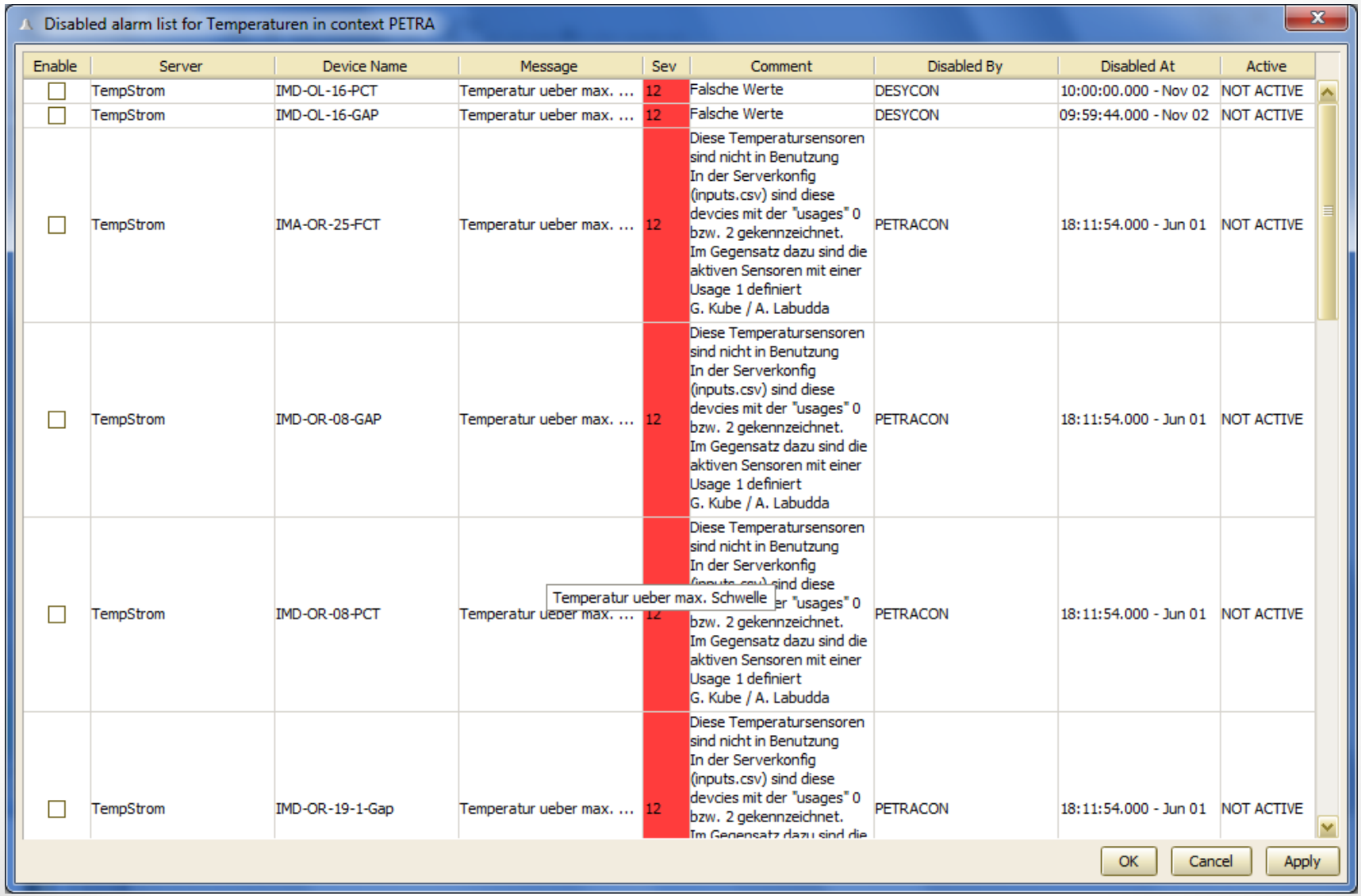
- Brief review of things which often get forgotten …
  - An Alarm belongs to a registered device !
  - Each server maintains a ‘local alarm server’
    - So if you set alarms you can see them in e.g. the instant client or the FEC Remote panel.
  - BUT: the central alarm server must be instructed to acquire alarms from a server
    - option: the CAS can automatically check form any new ‘important’ server when it starts up.
  - Alarms can be set and cleared by the server programmer
    - ClearAlarm() increments a ‘clear counter’ but does NOT immediately remove an alarm
    - RemoveAlarm() DOES immediately remove an alarm.
  - An alarm ‘watch’ file can be configured to regularly check readback values of a given property against a given threshold.
  - Some alarms are automatic (can optionally turn off automatic alarms)
    - Link Error alarms (severity = ‘information’)
    - Disk full alarms (severity = ‘information’)

Alarm System

- Alarms require a lot of feedback between those who set them and those who have to look at them …

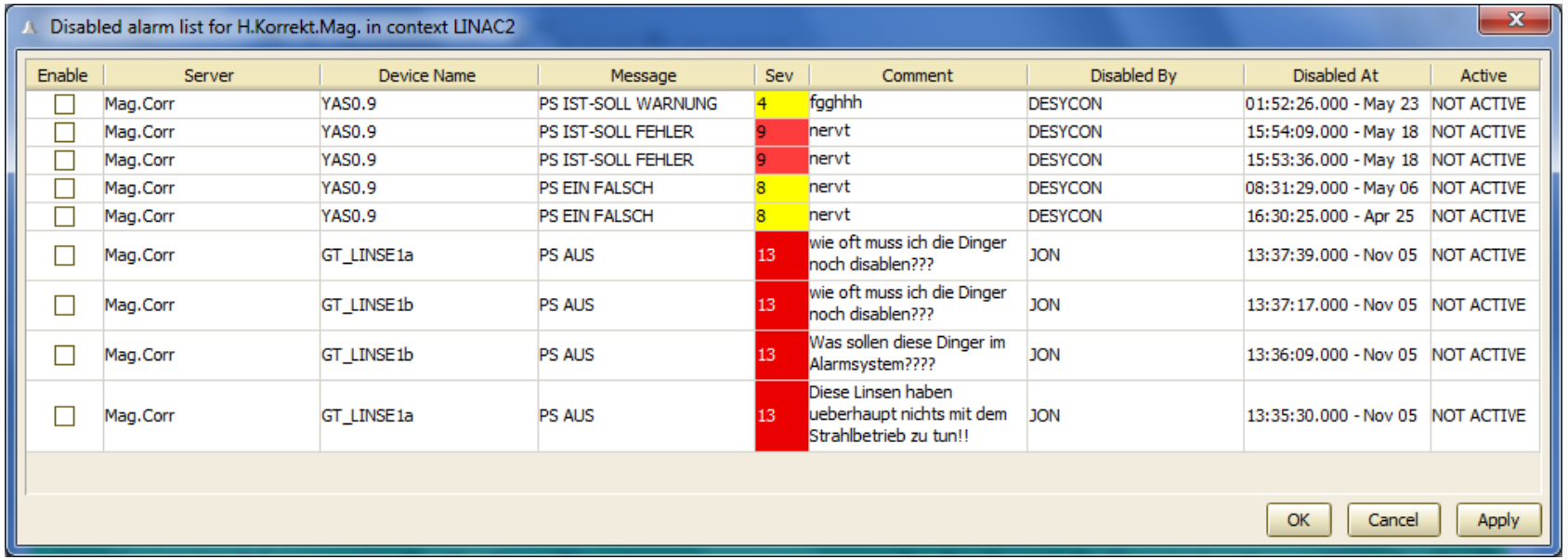


Alarm System



Alarm Viewer

- Or how about :



The operators want to ‘get to green’ but they are giving us feedback !