TINE Studio (RBE*)

VIEWING ARCHIVE DATA

*Real Best Ever

THE ARCHIVE SYSTEM

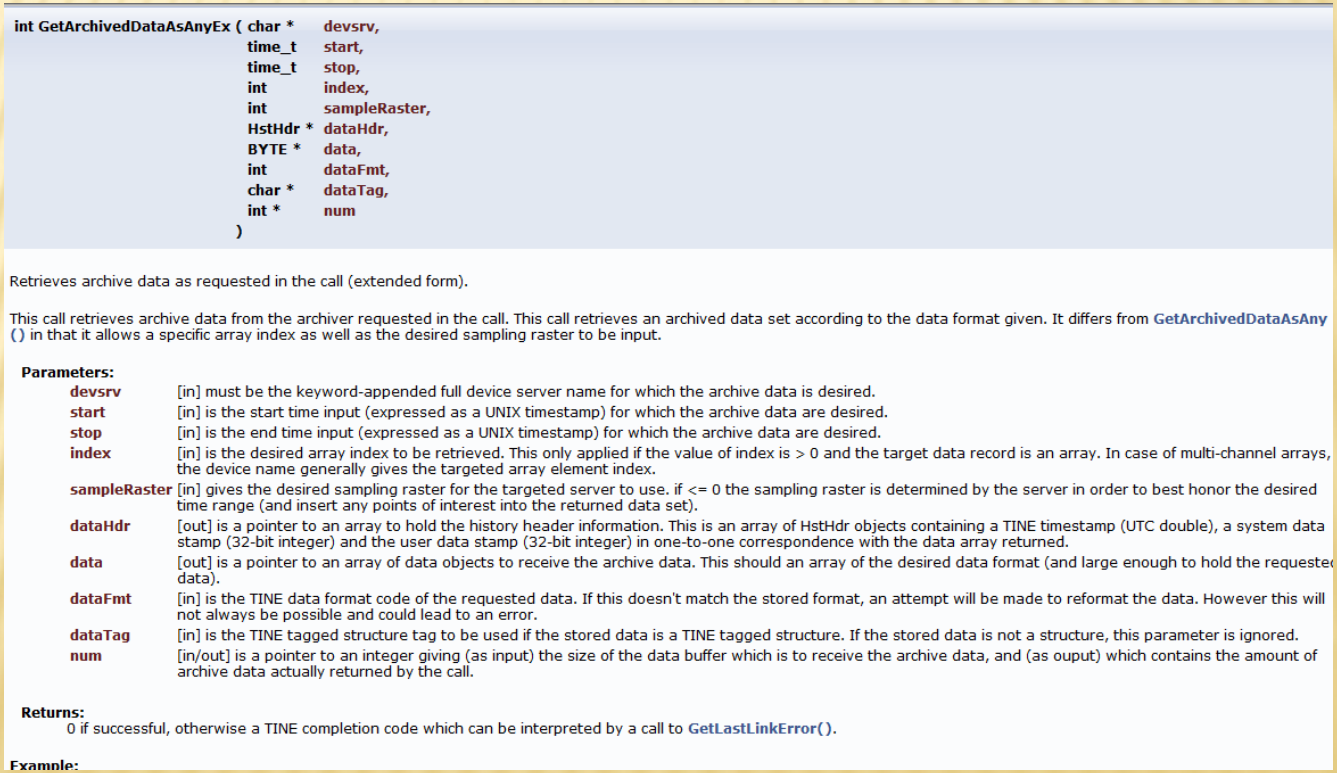
- Trends of relevant Machine Data
  - *What is happening on a regular basis …*
  - **Properties** of Control System Servers
  - Local Histories
    - Long-term storage on the server's local file system
    - Archive depth ~ months
    - Tolerance filters
      - Absolute and/or relative
  - Central Archive Server
    - Usually context specific (PETRA, DESY2, FLASH …)
    - Data is never compromised
      - Never removed from disk
      - Never adulterated (averaged, compressed, etc.)
    - Numerous filters

THE ARCHIVE SYSTEM

- Event Archive System
  - *What happened at some time due to a defined Event …*
    - Post-mortem (something bad happened)
    - State change
    - User-defined
  - That's another presentation!
- DAQ
  - *Save everything all the time!*
    - Data mining to find what you're looking for?
    - The TINE Archive System can help you out here!

INTERFACES

- APIs to the Archive System
  - C, C++
    - GetArchivedData(), GetArhivedDataAsAny(), …



```
int GetArchivedDataAsAnyEx ( char *     devsrv,
                             time_t     start,
                             time_t     stop,
                             int        index,
                             int        sampleRaster,
                             HstHdr *   dataHdr,
                             BYTE *     data,
                             int        dataFmt,
                             char *     dataTag,
                             int *      num
                           )
```

Retrieves archive data as requested in the call (extended form).

This call retrieves archive data from the archiver requested in the call. This call retrieves an archived data set according to the data format given. It differs from GetArchivedDataAsAny() in that it allows a specific array index as well as the desired sampling raster to be input.

Parameters:

devsrv	[in] must be the keyword-appended full device server name for which the archive data is desired.
start	[in] is the start time input (expressed as a UNIX timestamp) for which the archive data are desired.
stop	[in] is the end time input (expressed as a UNIX timestamp) for which the archive data are desired.
index	[in] is the desired array index to be retrieved. This only applied if the value of index is > 0 and the target data record is an array. In case of multi-channel arrays, the device name generally gives the targeted array element index.
sampleRaster	[in] gives the desired sampling raster for the targeted server to use. if <= 0 the sampling raster is determined by the server in order to best honor the desired time range (and insert any points of interest into the returned data set).
dataHdr	[out] is a pointer to an array to hold the history header information. This is an array of HstHdr objects containing a TINE timestamp (UTC double), a system data stamp (32-bit integer) and the user data stamp (32-bit integer) in one-to-one correspondence with the data array returned.
data	[out] is a pointer to an array of data objects to receive the archive data. This should an array of the desired data format (and large enough to hold the requeste data).
dataFmt	[in] is the TINE data format code of the requested data. If this doesn't match the stored format, an attempt will be made to reformat the data. However this will not always be possible and could lead to an error.
dataTag	[in] is the TINE tagged structure tag to be used if the stored data is a TINE tagged structure. If the stored data is not a structure, this parameter is ignored.
num	[in/out] is a pointer to an integer giving (as input) the size of the data buffer which is to receive the archive data, and (as ouput) which contains the amount of archive data actually returned by the call.

Returns:

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

Example:

INTERFACES

- MatLab

tine_history

Simply typing 'tine_history' at the command prompt will generate the 'usage' message shown below:

```
??? Error using ==> tine_history
tine_history usage:
tine_history('/<context>/<server>/<name>[<property>]','<stop
time>','<depth>'[,sampleInterval,index,'acquireSystemStamps'])
'ret' contains array of (timestamp,data) or (timestamp,system stamp,data)
```

We see that there are three required input parameters, namely the targeted endpoint, the 'stop time' (i.e. the most recent requested data point) and the depth of the archive trend.

All parameters are strings. The 'stop time' should be given in a string data format ('dd.mm.yyyy hh:mm:ss') or the string 'now' to refer to the current time. The 'depth' is a string begining with a number and followed by a representative unit of time ('days','hours','minutes', etc.)

Optionally one can specify a 'sampling interval'. If this is '0' or less then the targeted server will determined a suitable sampling raster so that the returned data cover the desired time interval. Internally a maximum of 8000 data points will be returned. Thus for a large time interval it could happen that stored data points are 'skipped' in order to supply data throughout the requested range. If this is not the desired behavior, an explicit sampling interval can be provided. To avoid skipping any points a value of '1' should be passed (default = '0'). One can also optionally specify an array 'index' as the 5th parameter. If this value is greater than '0' then the specific array index will be selected from the any archive record that refers to an array In the case of multi-channel arrays, this parameter is usually not necessary as the the device name will determine the array index (default = '0'). Finally one can optionally indicate in the 6th parameter whether or not to return associated 'system stamps' with the archive data. In most cases one is interested in data versus a timestamp. As most data are also archived along with the accompanying system stamp, it could be useful to retrieve this information as well. The system stamp might refer to a 'cycle number', 'pulse number', or 'shot number' depending on the nature of the accelerator. The timestamp returned is the MatLab time (and not UTC).

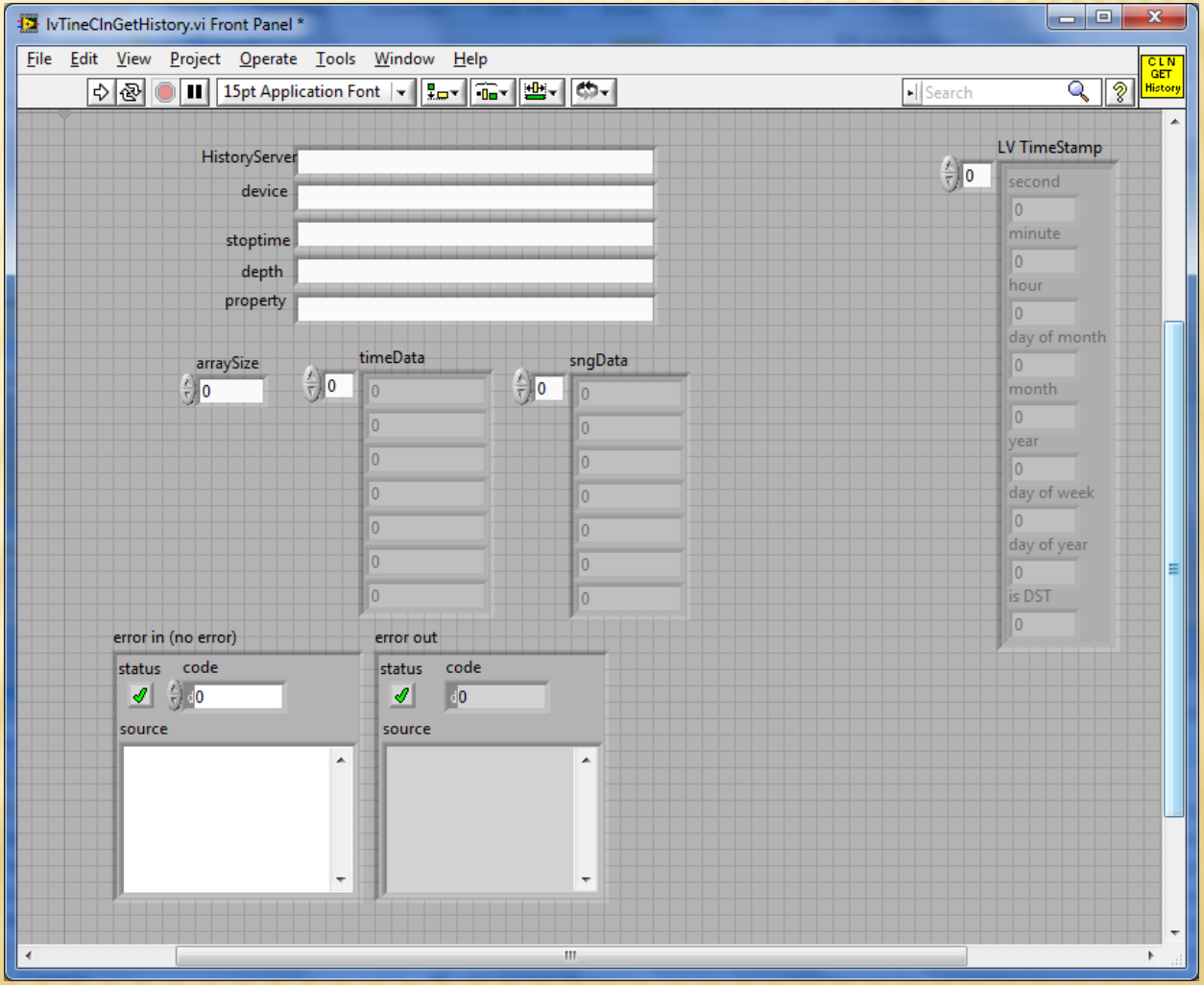
The targeted end point can be a specific server, in which case the 'local history' is obtained directly from the server (if it is available!), as in:

```
>> format long
>> tine_history('/PETRA/Idc/Buffer-0[I]','now','1hour')

ans =

  1.0e+005 *

   7.354026290012483   0.001006214523315
   7.354026290140839   0.001006183395386
   7.354026290370469   0.001006119766235
   7.354026290498942   0.001006123504639
   7.354026290728571   0.001006073532104
   7.354026290958316   0.001006025466919
```

Note that as the number involved are large it is frequently useful to make use of the 'format long' specification in MatLab

INTERFACES

- LabView:



INTERFACES

- Command Line:

thistory

Typing 'thistory' at the command line will produce the following output:

```
Retrieves a stored archive data for the keyword and device specfied
Data can be obtained from a central archive server or from a local history,
depending on how the context and keyword are specified.

A simple keyword along with a context will contact the central archive
server for the context.

        e.g. thistory thistory HERA HPDCCur now 1hour

The history depth can be specified in hours, days, weeks, or months and
is parsed as

        <number><time unit>

The history target time can be 'now', a valid unix timestamp, or a date
and time string in the form

        <day>.<month>.<year>_<hour>.<minute>.<second>

where all entries execpt <day> are optional from right to left.

        e.g. thistory HERA HPDCCur 31.05.2006 1day
          or thistory HERA HPDCCur 31.05.2006_12:00:00 1hour

A local history can be obtained by specifying the target device server
along with the context (with a leading "/")

        e.g. thistory /HERA/HEEKOLLI SOLLWERT HESL66i 31.05.2006 1day

A history snapshot of an array record at a specific time can be obtained
by specifying 'snapshot' as the depth parameter.

        e.g. thistory /HERA/HEPBPM ORBIT.X #1 7.06.2006_12:00:00 snapshot

returns the array data record stored at or after noon on June 7th 2006.
The timestamp of the record found is always displayed along with the data.

Usage : thistory <context> <keyword> <device name> <stop time> <depth>
```

INTERFACES

- Java:
  - THistory.getArchiveData() + overloads …

```
static int de.desy.tine.histUtils.THistory.getArchivedData ( String              context,
                                                             String              server,
                                                             String              property,
                                                             String              device,
                                                             double              start,
                                                             double              stop,
                                                             TCompoundDataObject[] data,
                                                             THistorySource      src,
                                                             int                 timeout
                                                           )                     [static]
```

Retrieves an archive data set.

Retrieve an archive data set (value - timestamp pairs) for the target property and device given and over the time range specified. Both the central archive and local archive are queried for the number of stored points over the interval. Data are usually retrieved from the central archive system if there are at least 500 points over the time interval in question. Otherwise the local archive system is used. By examining the THistorySource object one can see which source was chosen and how many points both sources have over the interval. For dedicated retrieval from one source or the other, please use an overloaded method which does not take the THistorySource parameter!

Parameters:

context	is the desired context
server	is the device server which manages the archived parameter. If null then the central archive will be queried for the property and device
property	is the desired archived property
device	is the desired device name
start	is the start time in seconds (UTC double, a TINE timestamp)
stop	is the stop time in seconds (UTC double, a TINE timestamp)
data	is a reference to a (doublet) data array object to contain the returned data. This can be one of FLTINT[], DBLDBL[], NAME64I[], and so on. The array should be dimensioned to the maximum number of points desired (effectively determining a raster over which the archived data will be distributed. The reference object can also be an array of HISTORY[] object instances, which have been constructed with the desired data type (usually a compound data type) to be returned. The src parameter will give the exact number of points actually stored over the interval specified. The number of points returned in the call will often be less than array dimension and less than the exact number of points stored over the interval.
src	is a reference to a THistorySource object to receive the history source information.
timeout	gives the number of milliseconds to wait for the call to complete (usually 1000 is sufficient).

Returns:

0 upon success or a TINE error code.

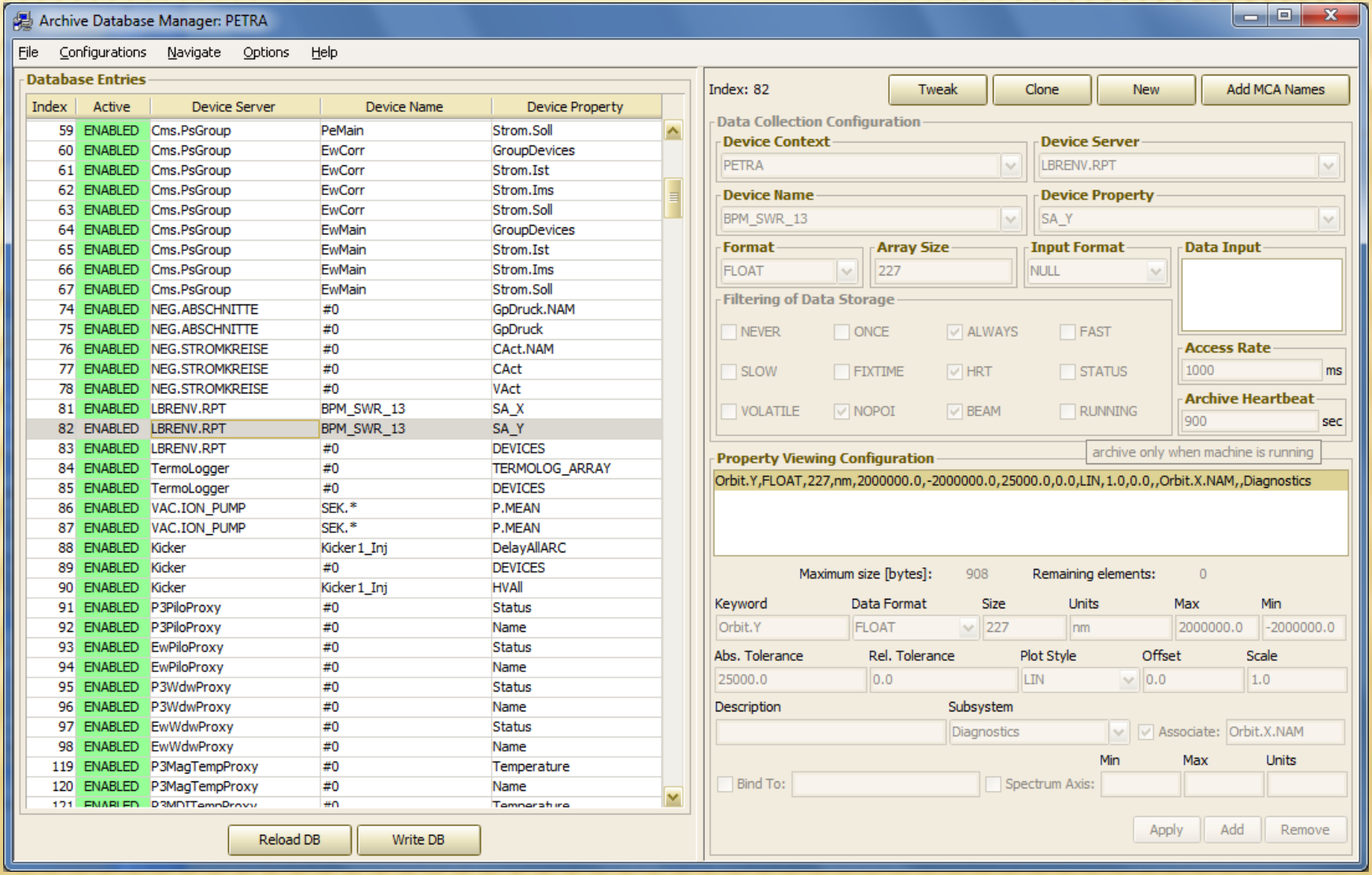
ARCHIVE VIEWER USES JAVA API

- Always checks 2 data sources
  - Local and central archives
    - Check number points in interval
    - If Central has > 500 points use it !
    - Else use source with most points.
- Optical zooming
  - There's a raster !
  - BUT: there are points of interest !
  - n.b. The archive API also allow a 'fill and increment' strategy (where sample raster = 1).
- Array Records and Snapshots
  - Waveforms (Spectra) and Multi-Channel Arrays
  - Snapshots: history of record at a specific time

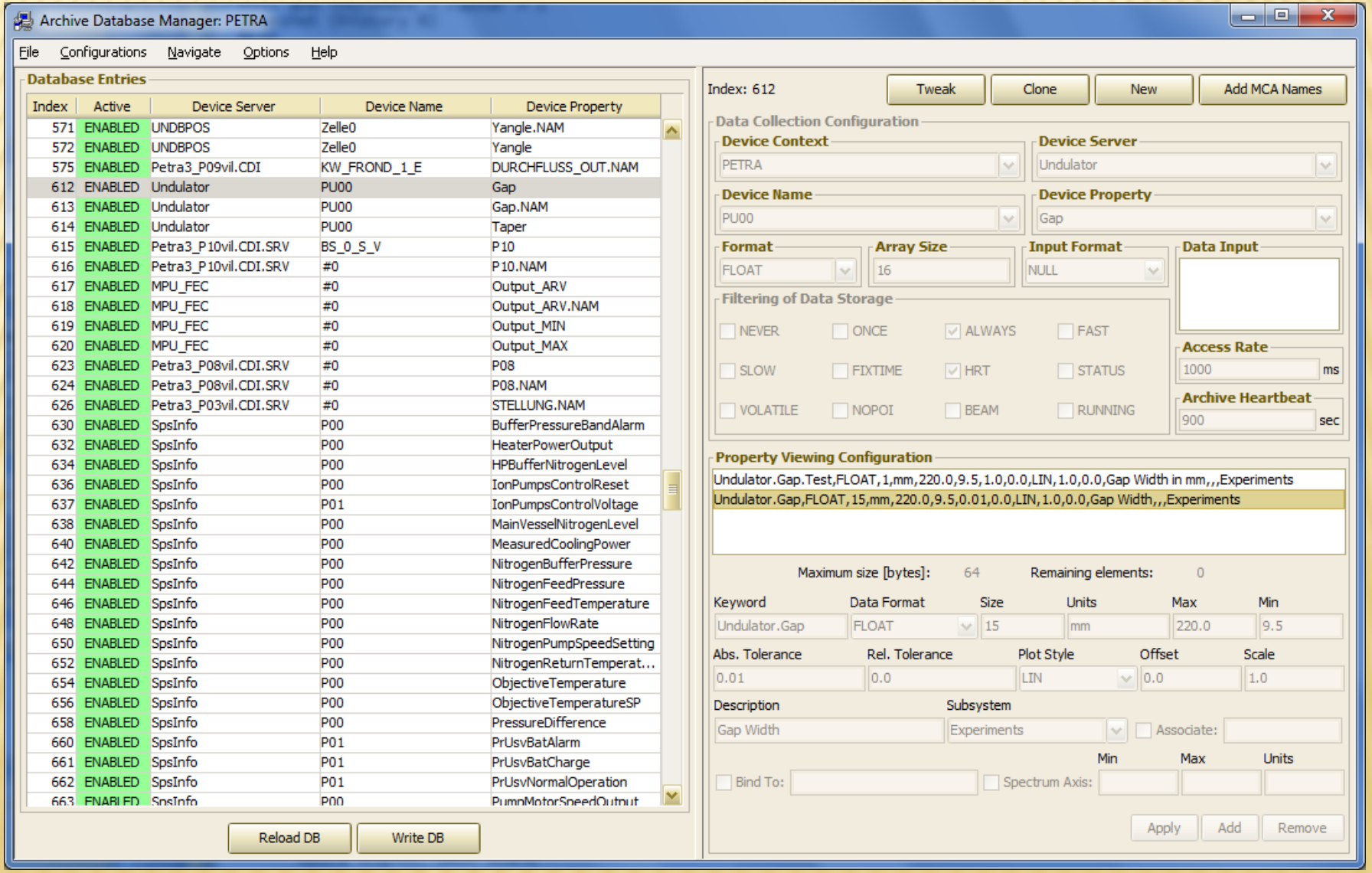
DATA ACQUISTION

- Central archive:
  - Monitors data in TIMER mode (default)
  - BUT can receive '*scheduled*' data.
  - A Record can be bound to another Record
    - *"archive me too, when you archive him"*
  - Meta Info can be associated
    - *Device names for Record A are the same for Record B*.
- Local archive:
  - Monitors data in TIMER mode
  - Can be *scheduled*!
  - Equipment Module always sees the CA_HIST access flag.

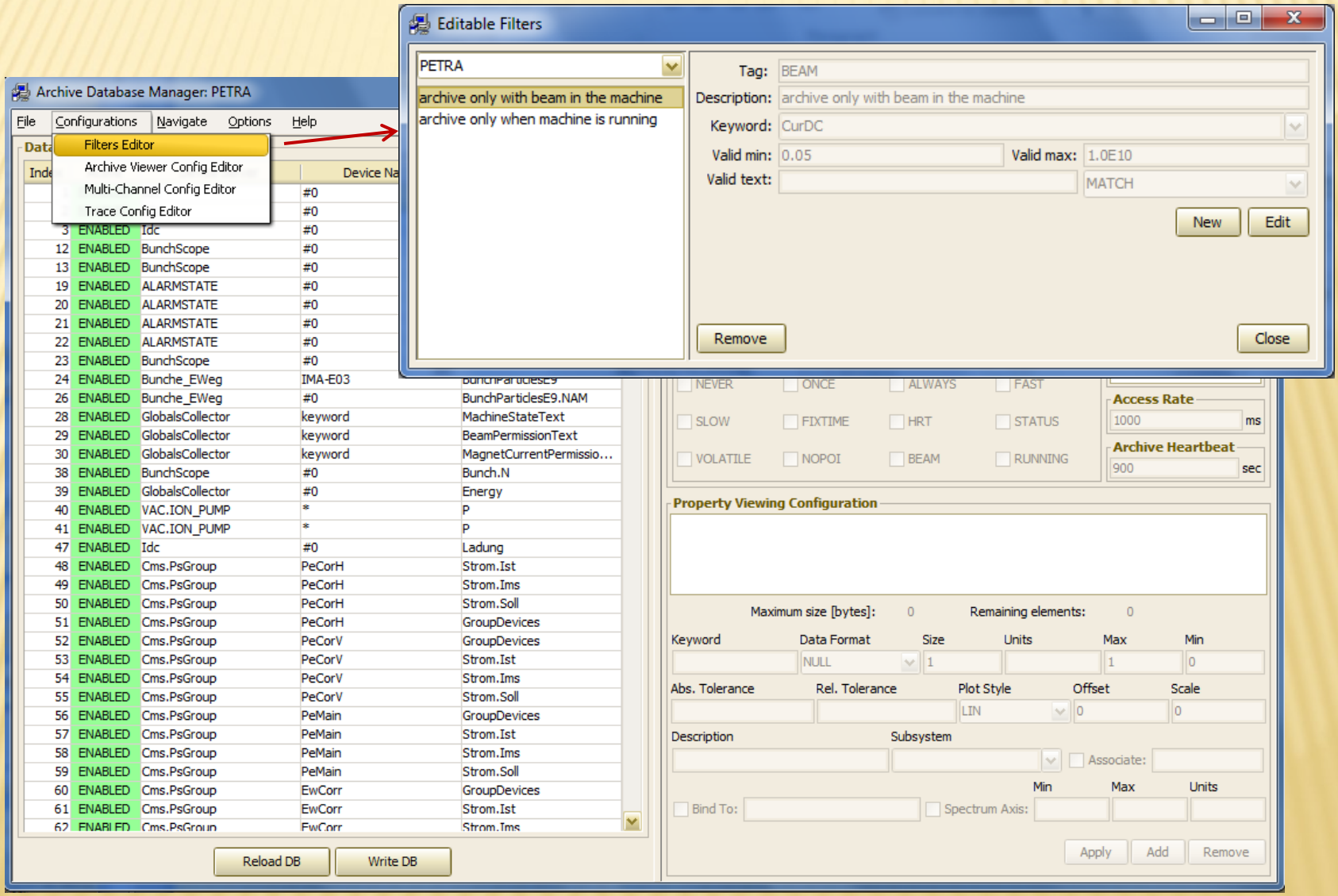
ARCHIVE DATABASE MANAGER



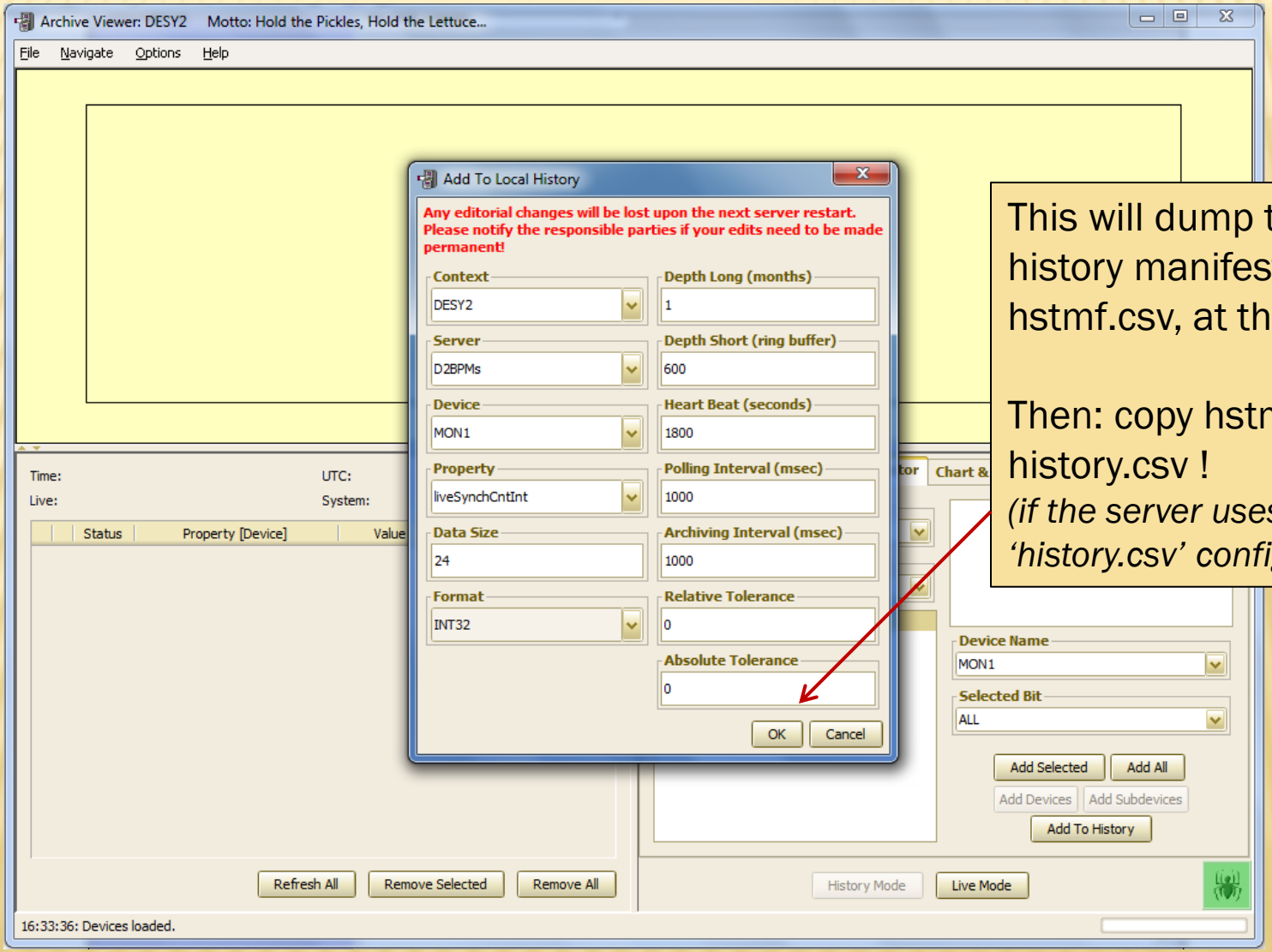
ARCHIVE DATABASE MANAGER



ARCHIVE DATABASE MANAGER



LOCAL HISTORY CONFIGURATIONS



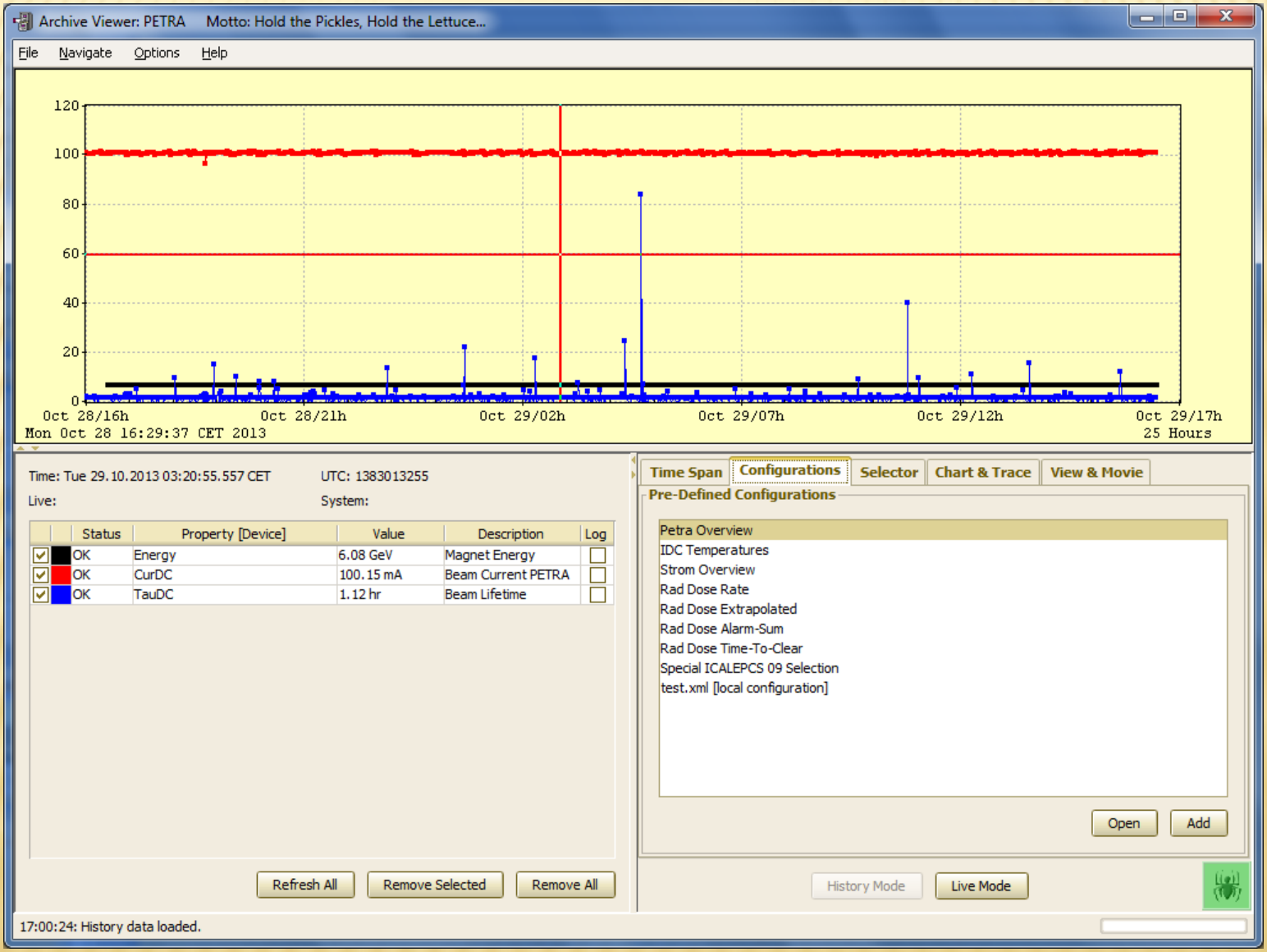
This will dump the new history manifest file, hstmf.csv, at the server !

Then: copy hstmf.csv -> history.csv !
(if the server uses 'history.csv' configuration)

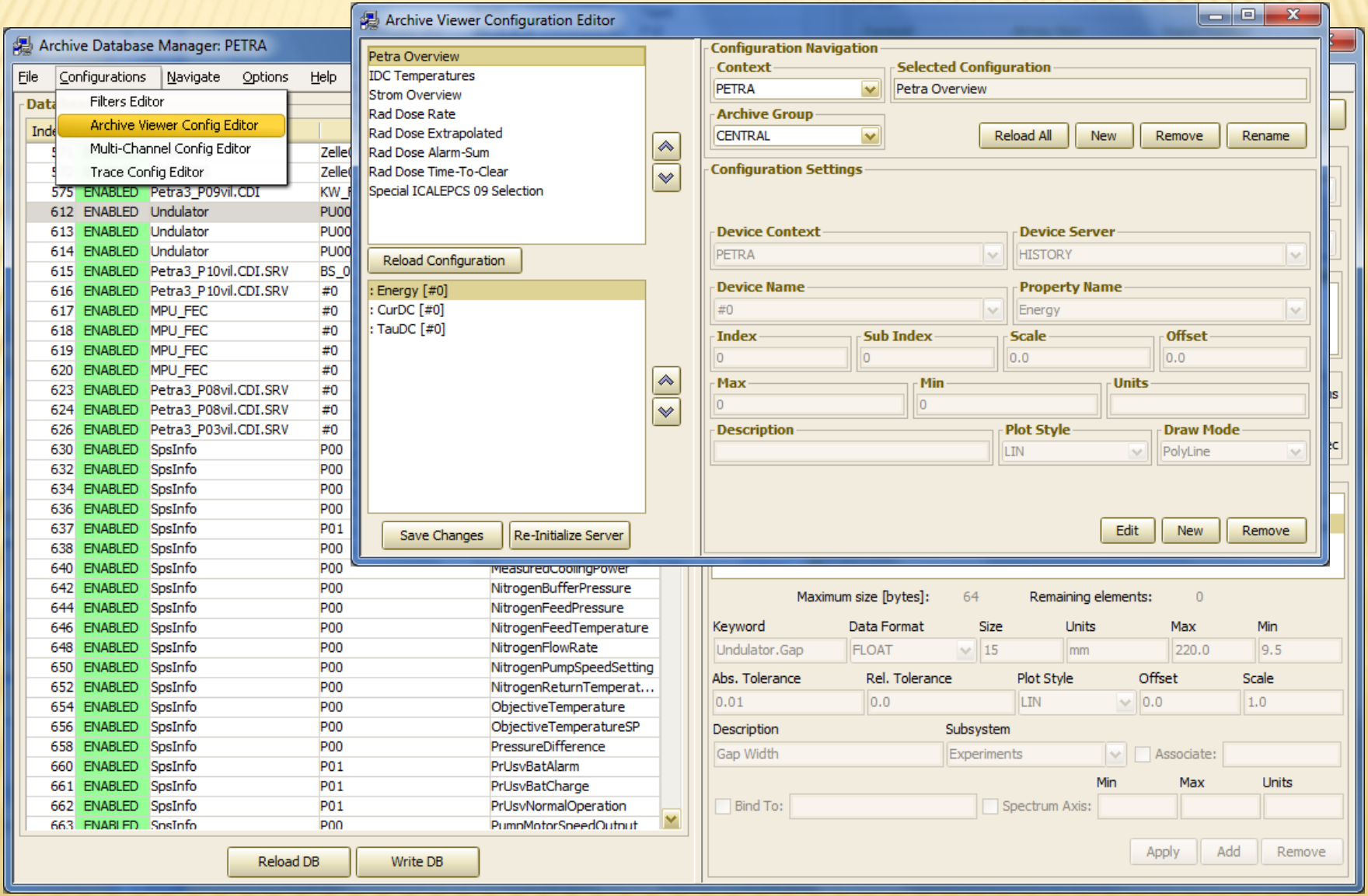
ARCHIVE VIEWER

- Feature Rich !!!
  - The application with 1 and only 1 feature can probably be made '*bug free*'
    - printf("hello world\n");
  - The application with N features has at least 2^N ways of introducing bugs even if each feature has been fully tested individually.
    - Probably more like N! * 2^N
  - => we will probably find a few things to report to cosylab !

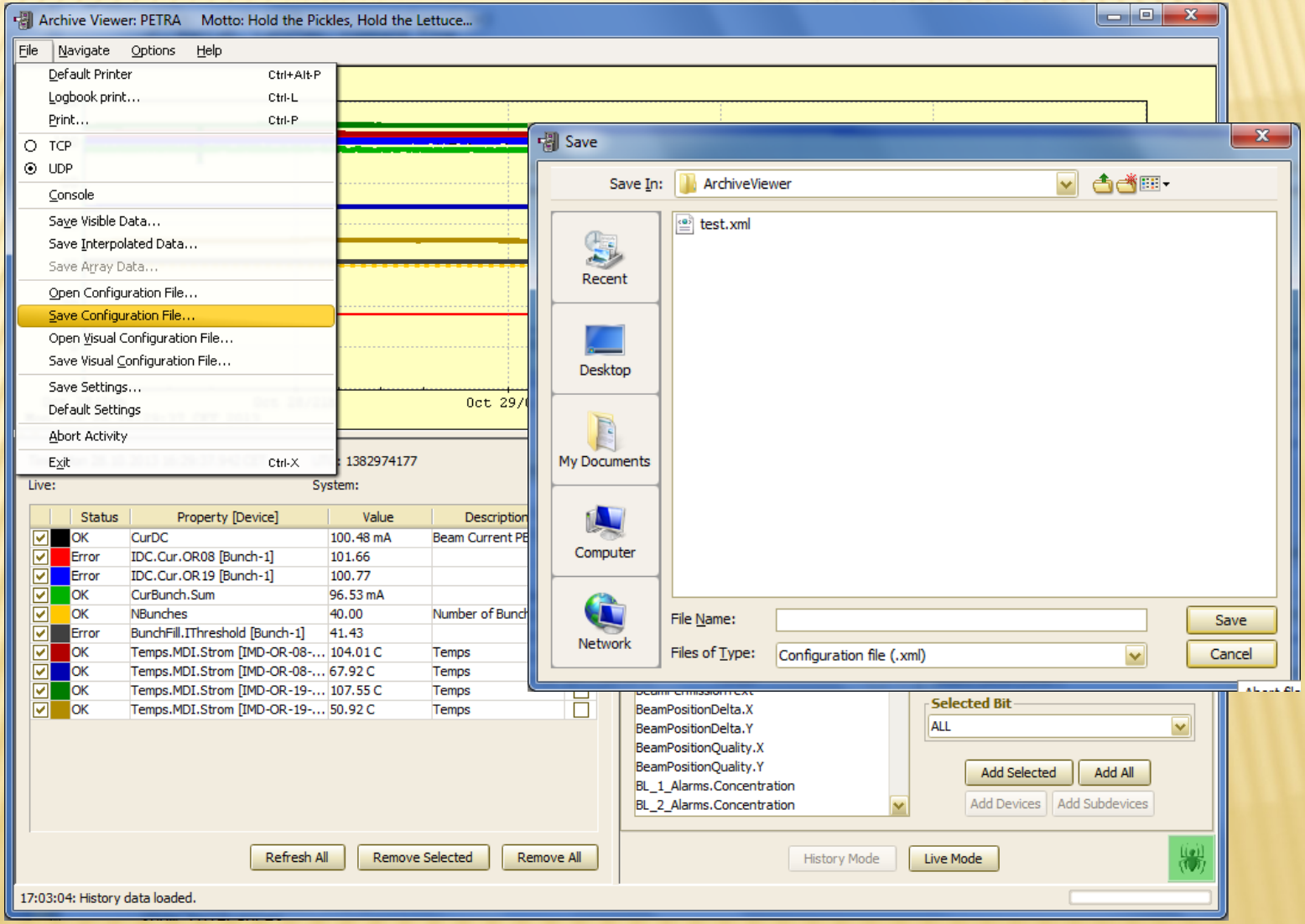
ARCHIVE VIEWER



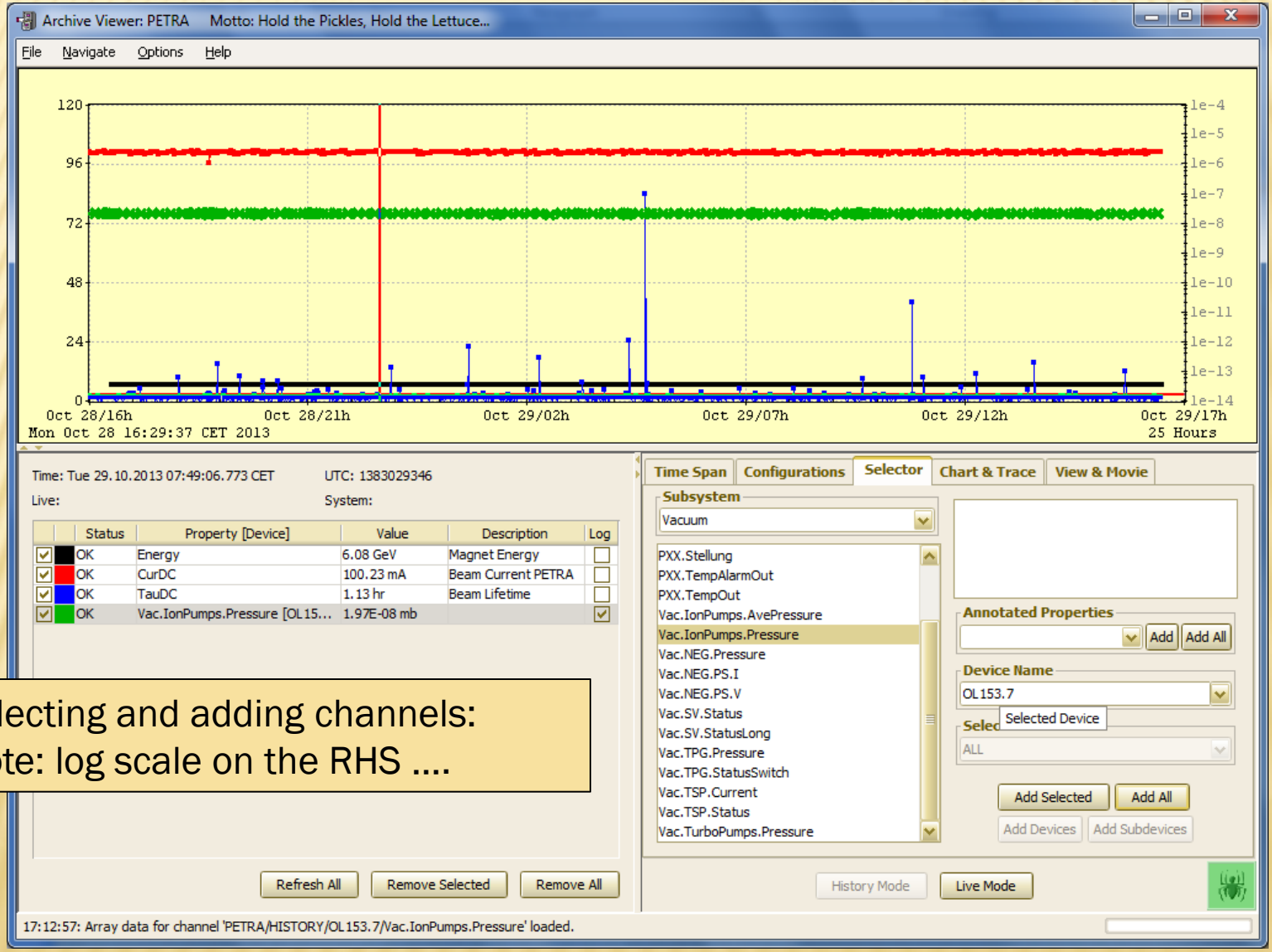
ARCHIVE DATABASE MANAGER



ARCHIVE VIEWER



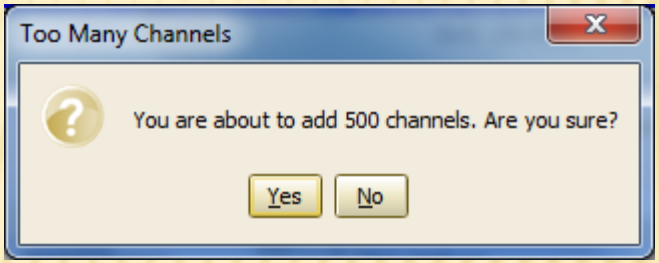
ARCHIVE VIEWER



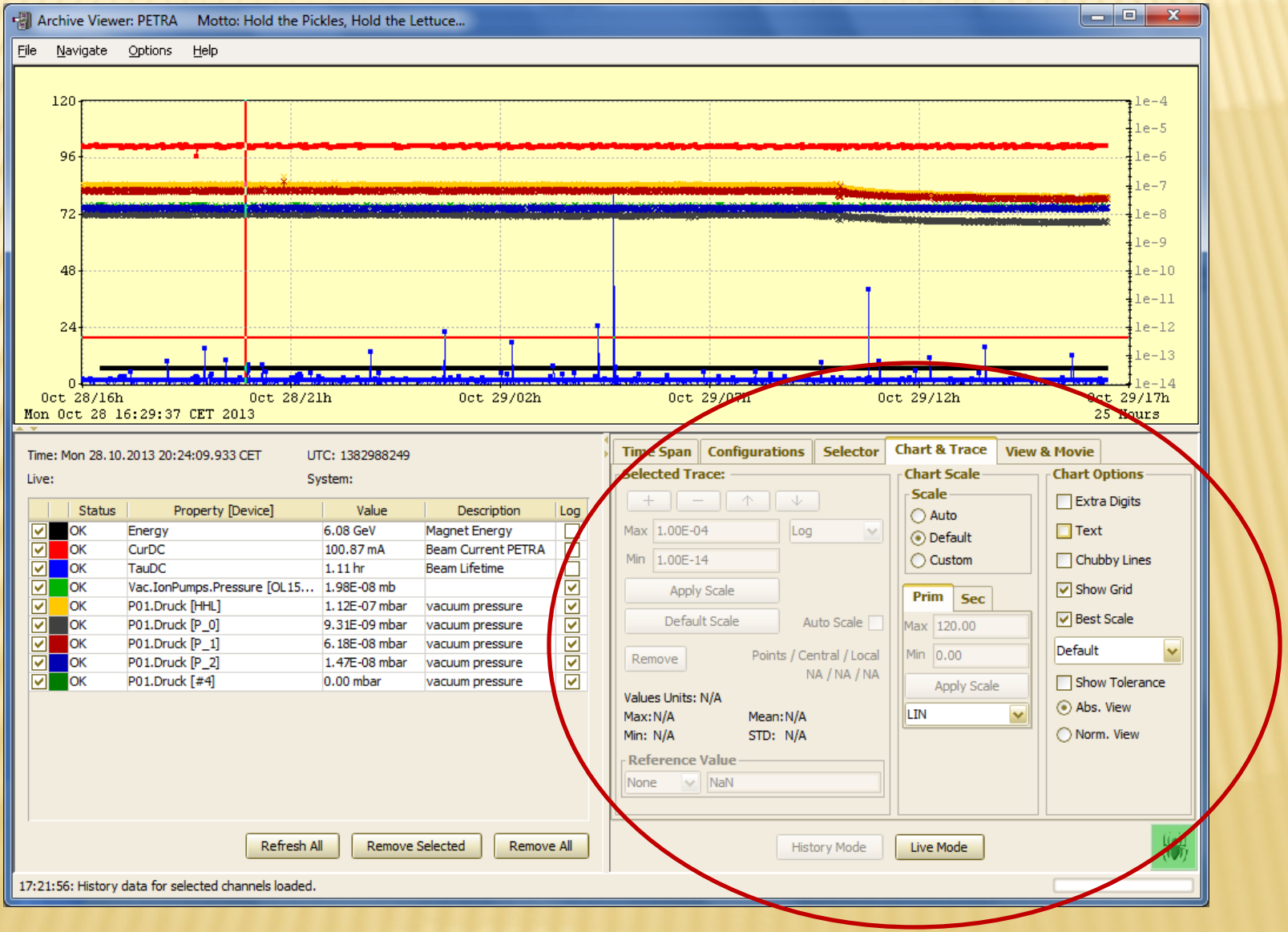
selecting and adding channels:
Note: log scale on the RHS ....

ARCHIVE VIEWER

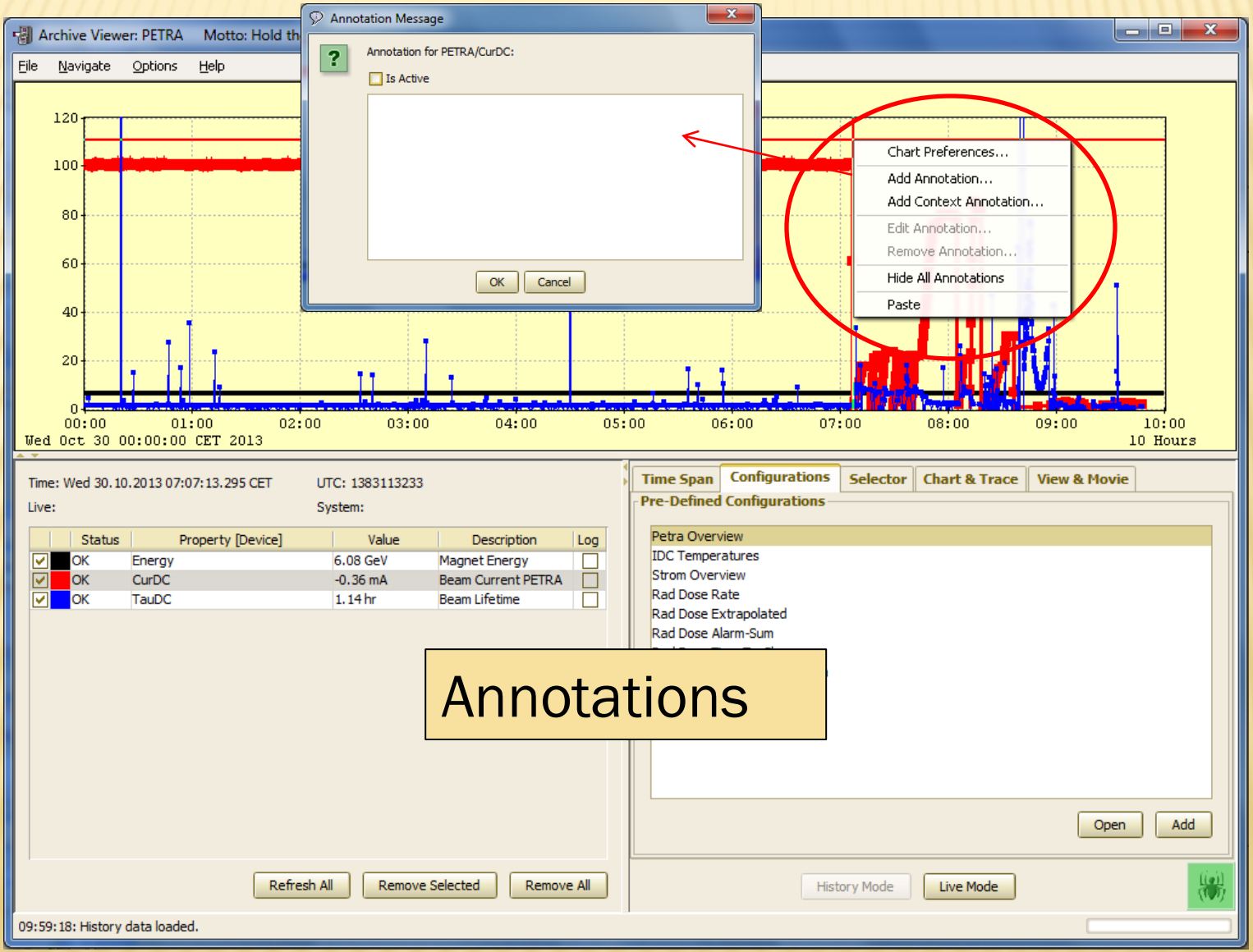
- And if you hit that ‘Add All’ button and there are more than 10 channels:



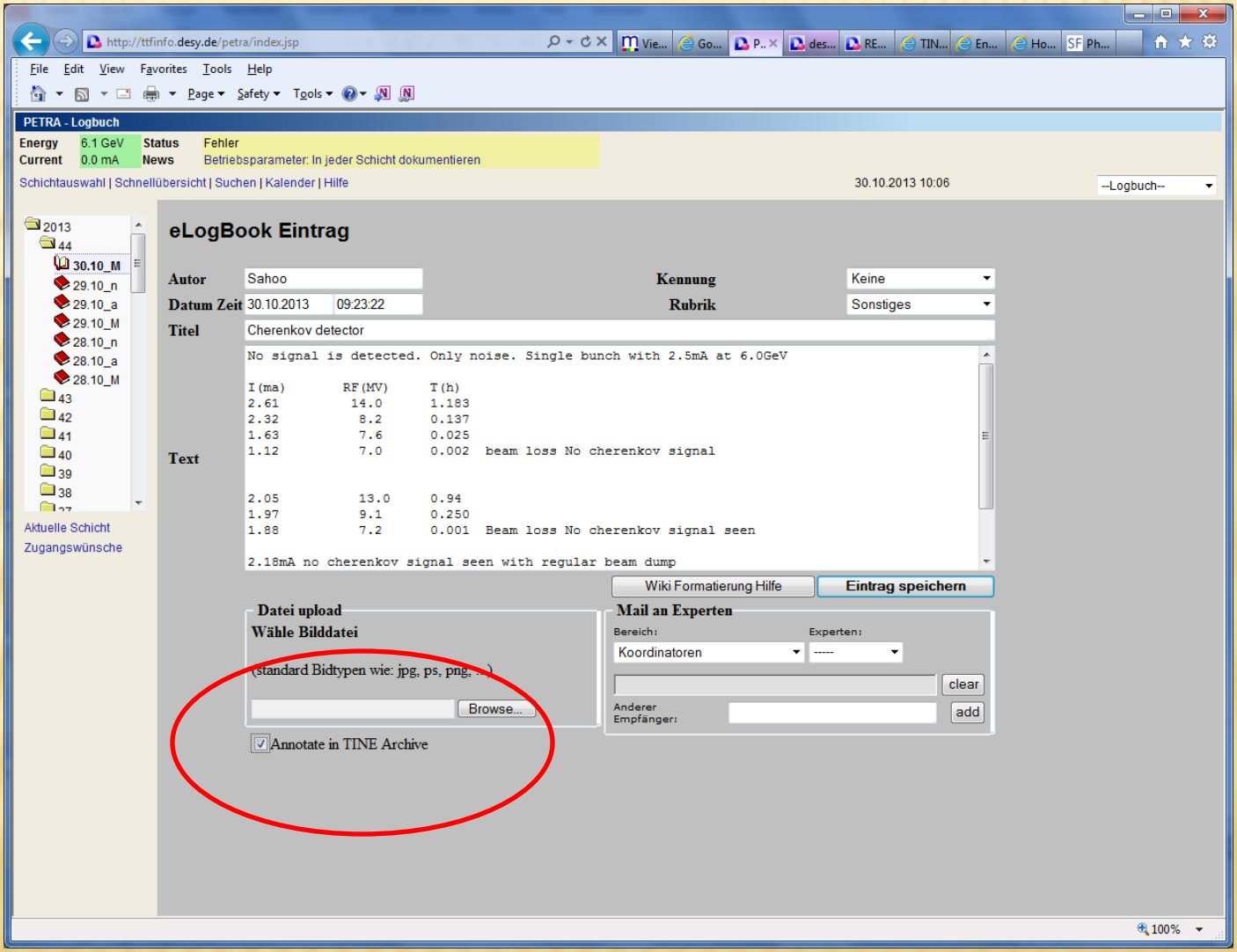
ARCHIVE VIEWER



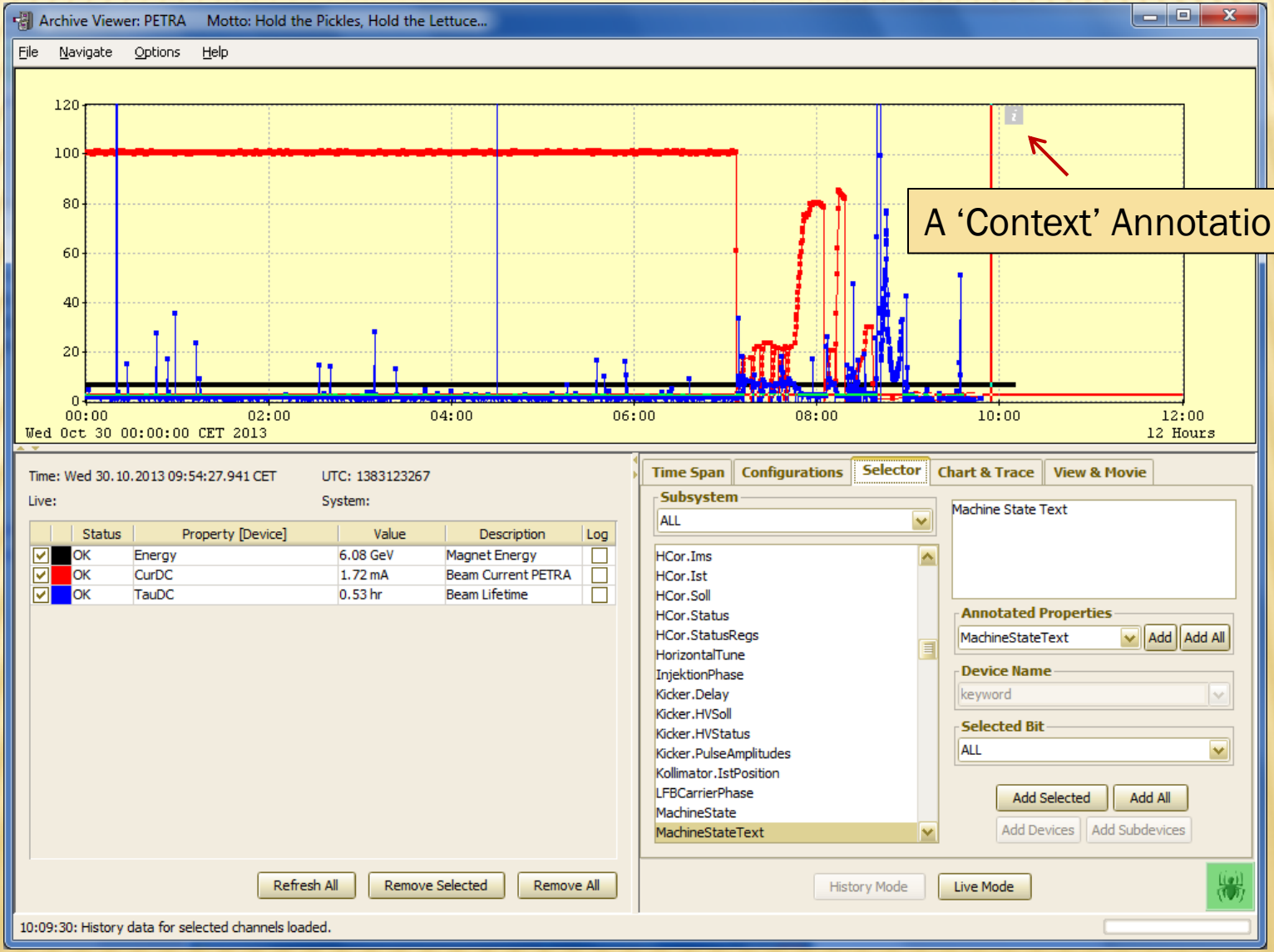
ARCHIVE VIEWER



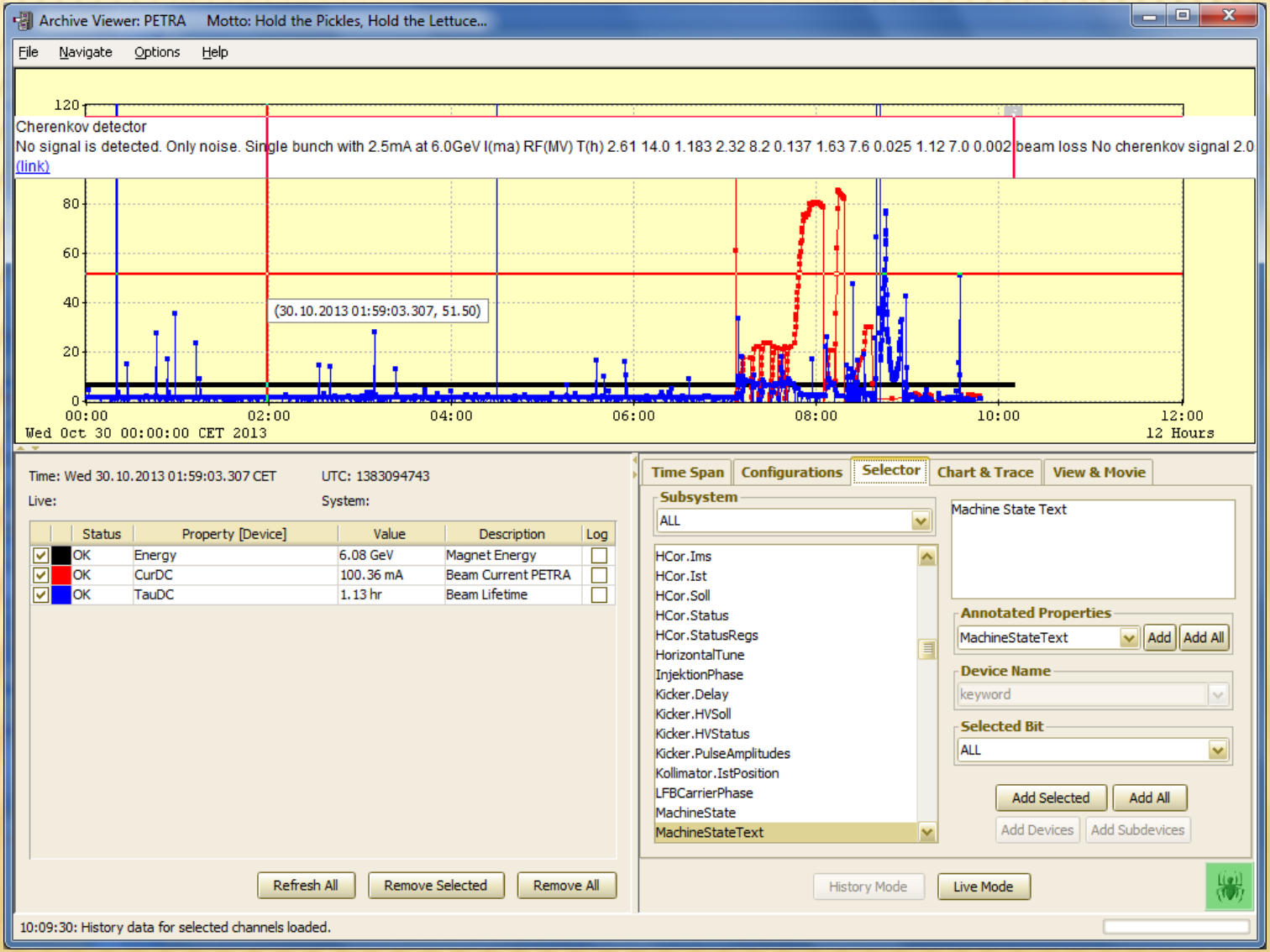
ANNOTATING FROM THE LOGBOOK



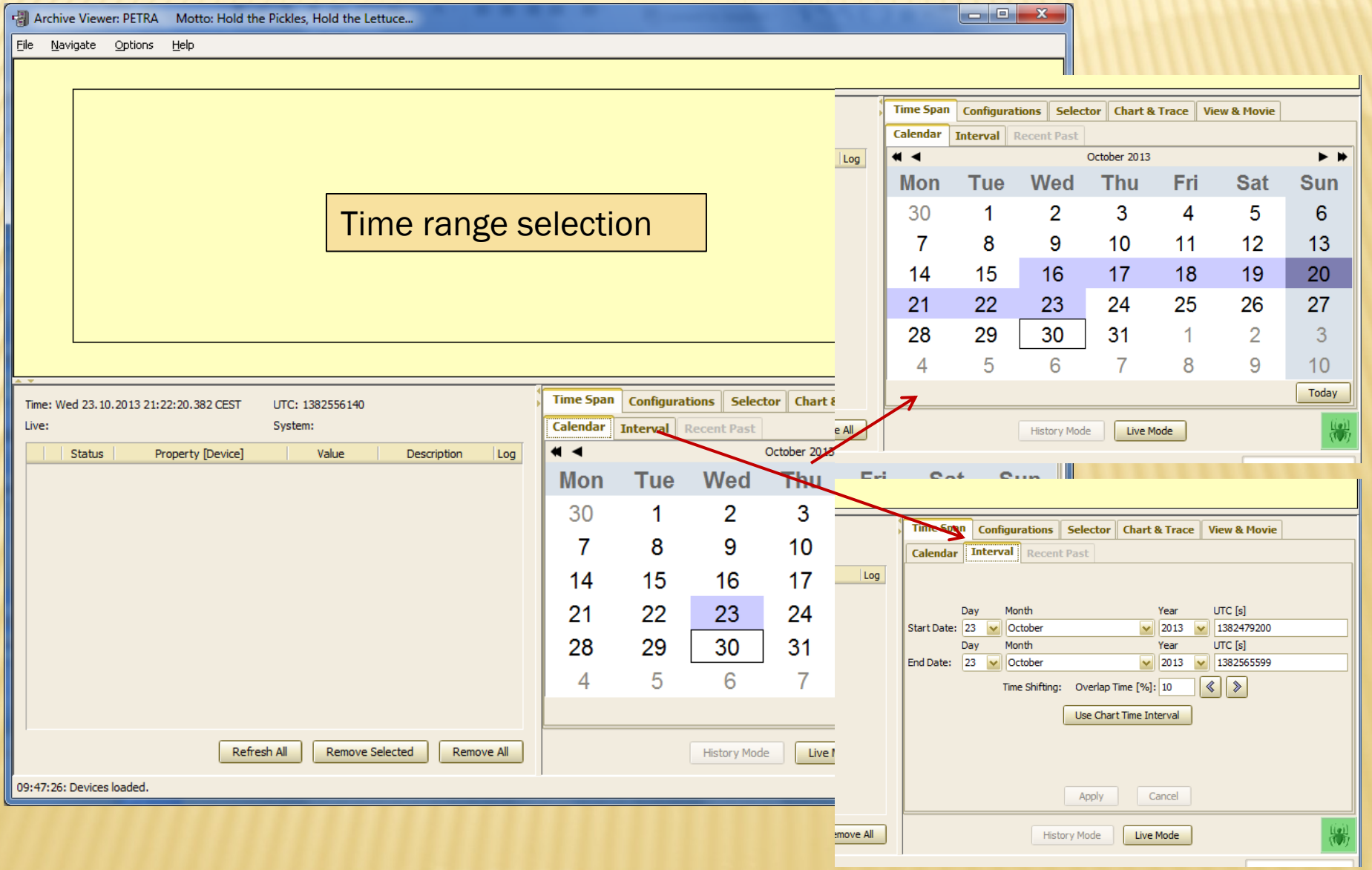
ARCHIVE VIEWER



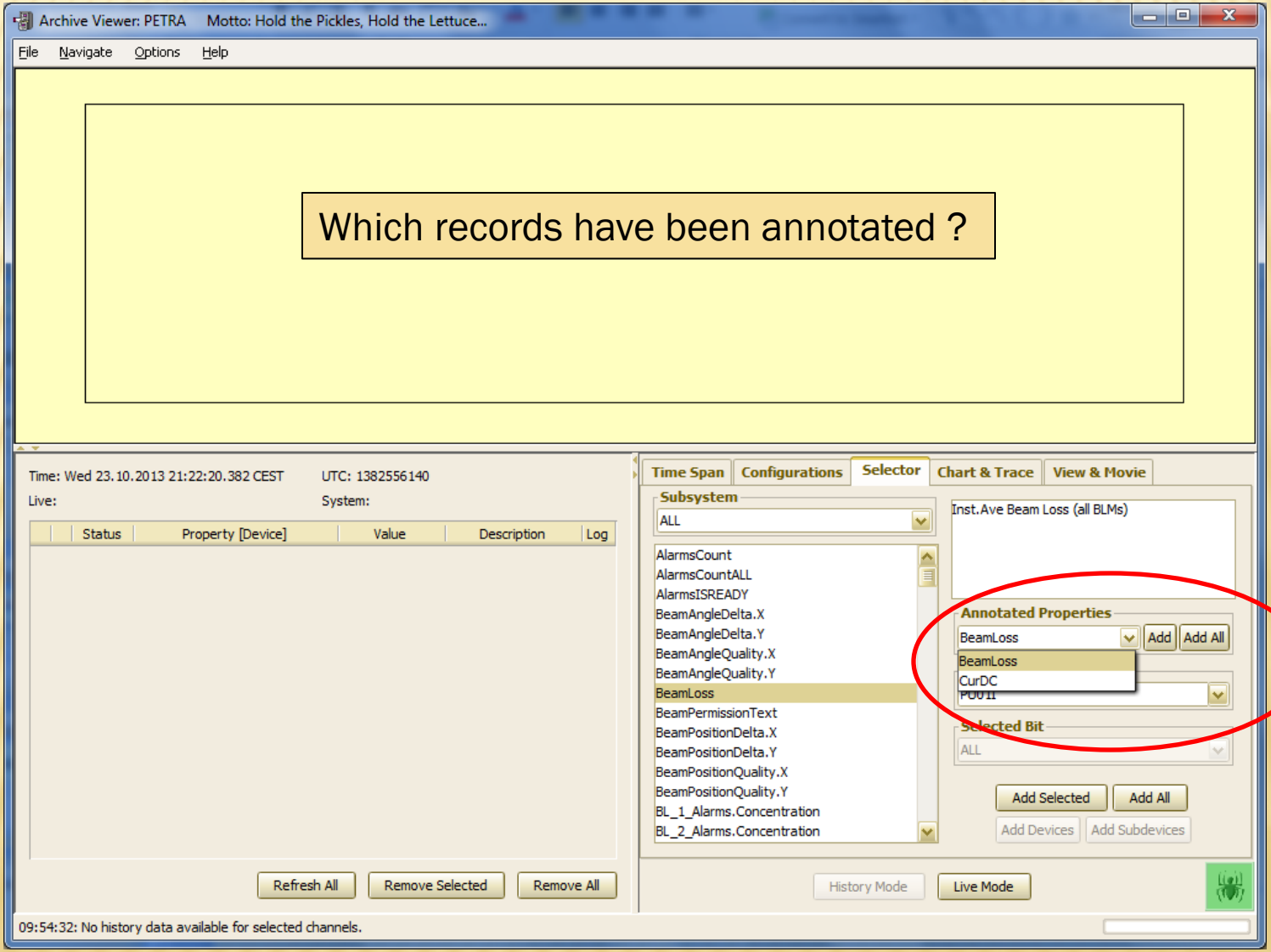
ARCHIVE VIEWER



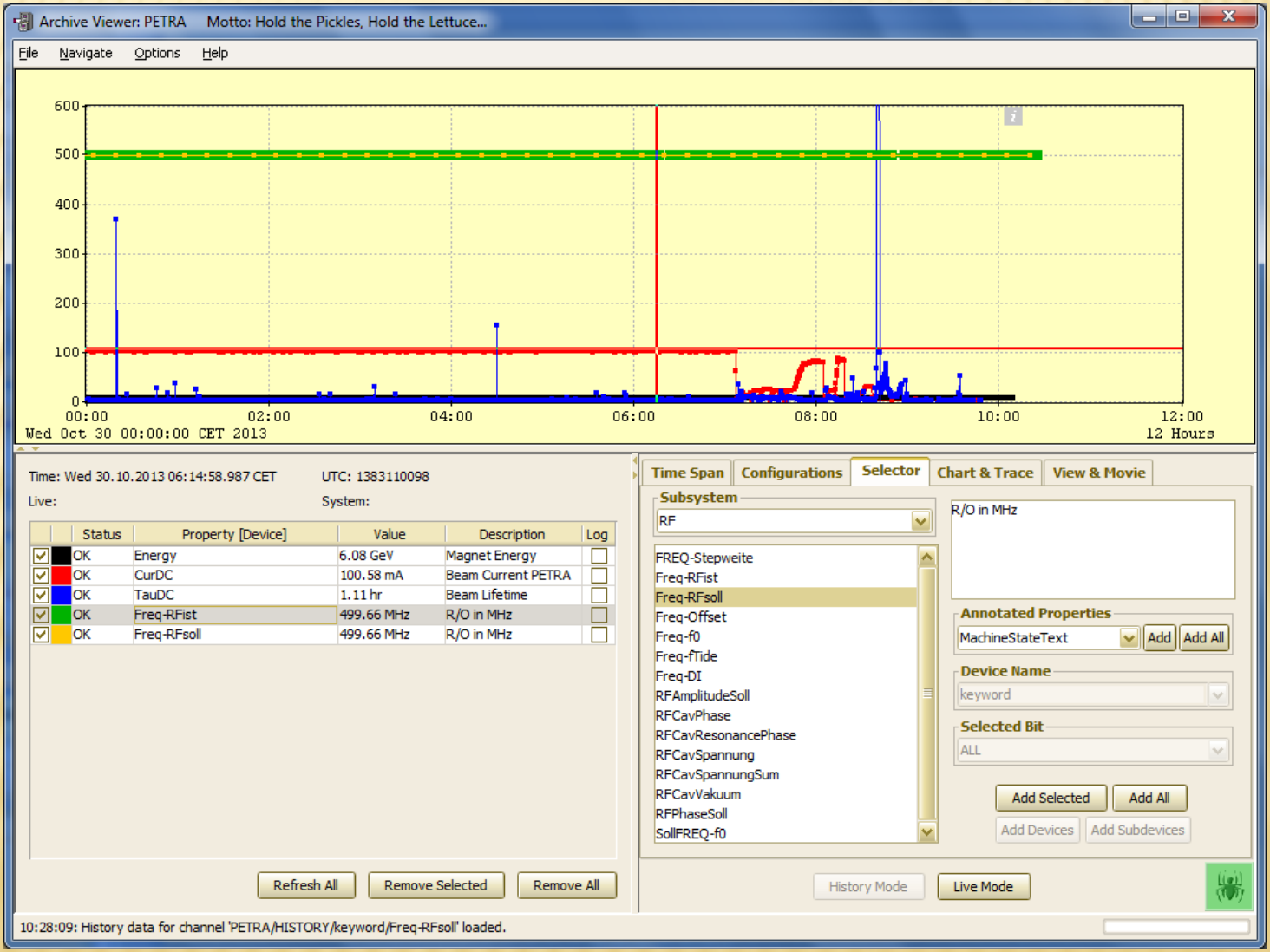
ARCHIVE VIEWER



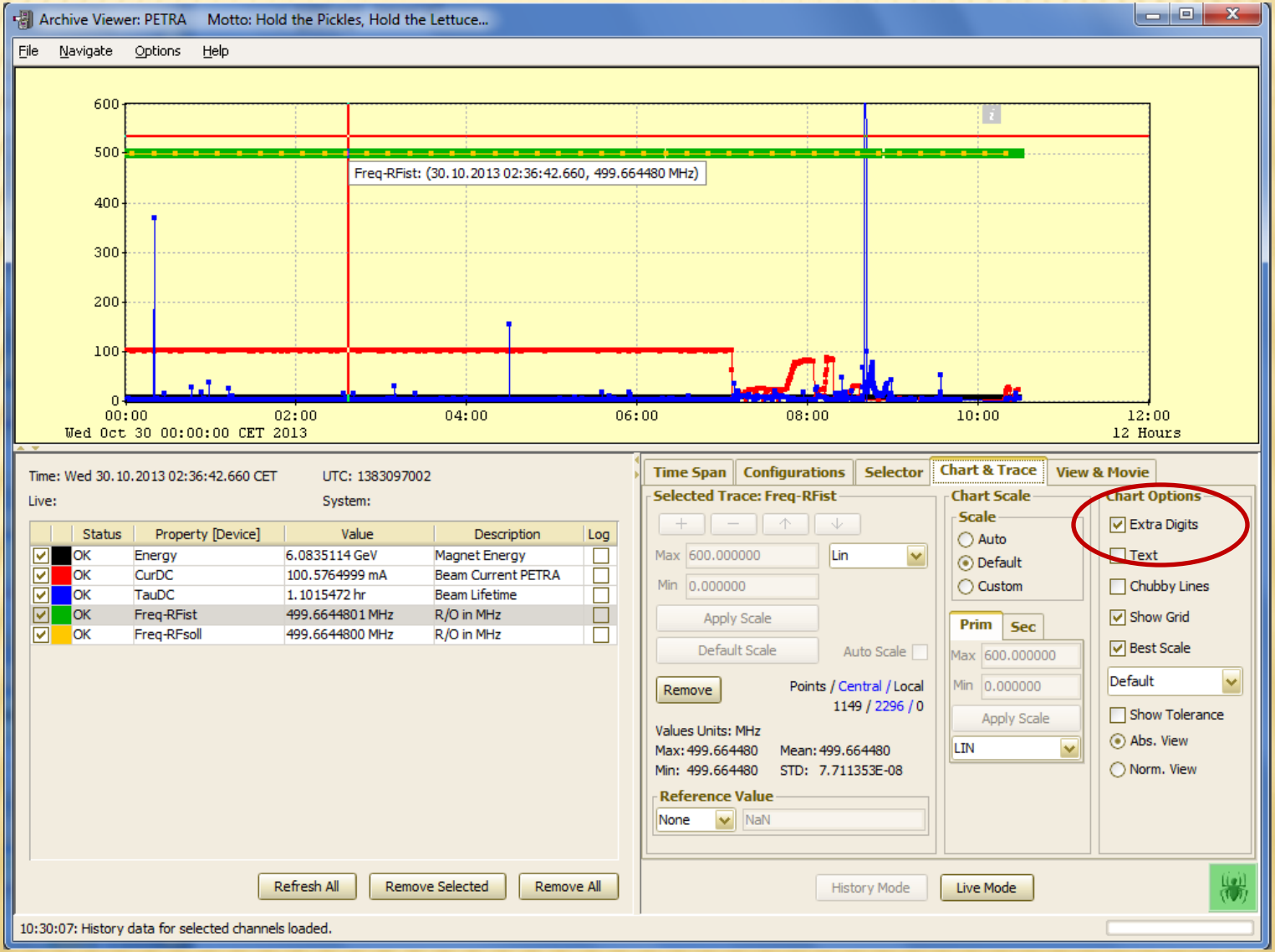
ARCHIVE VIEWER



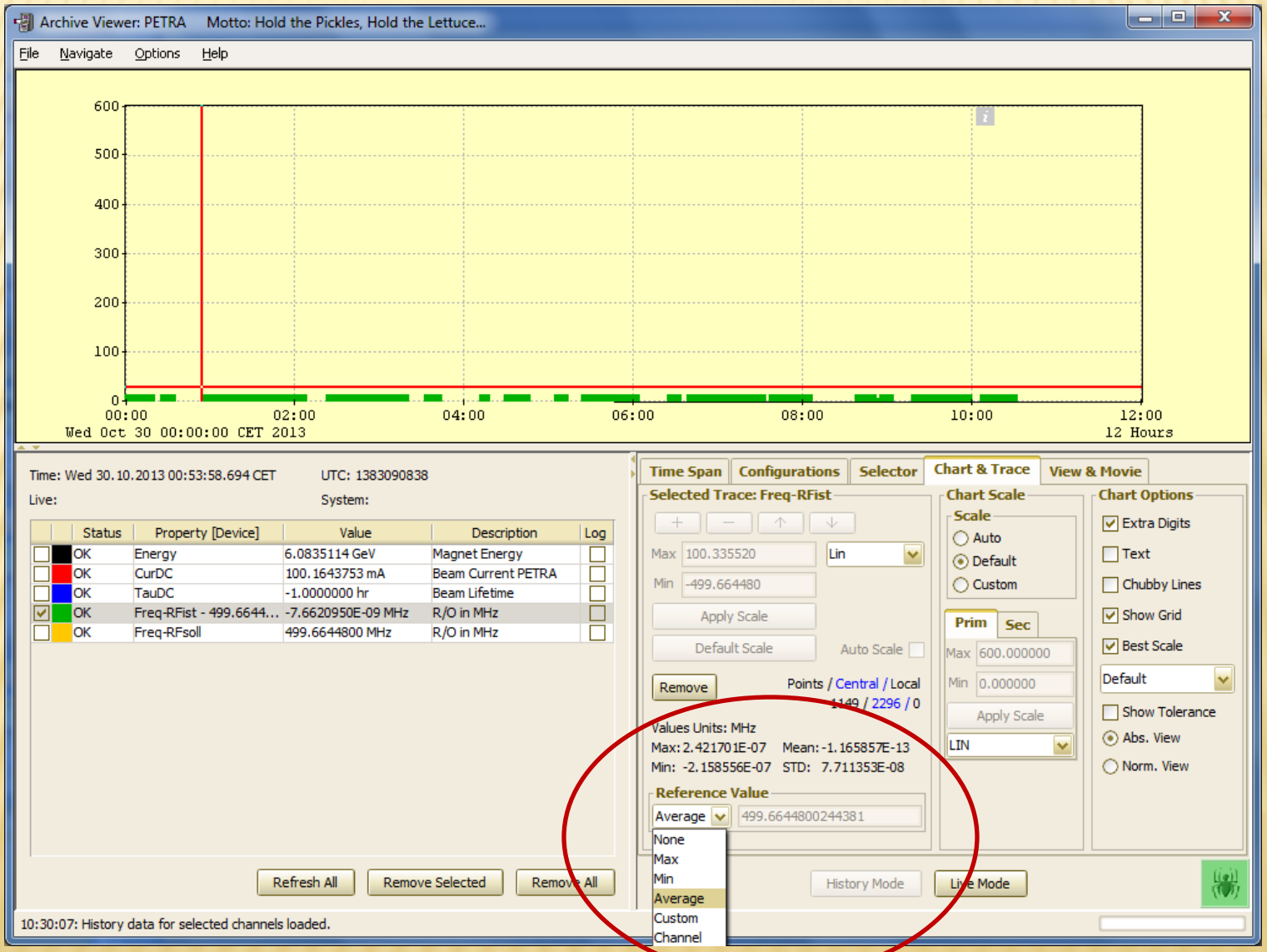
ARCHIVE VIEWER



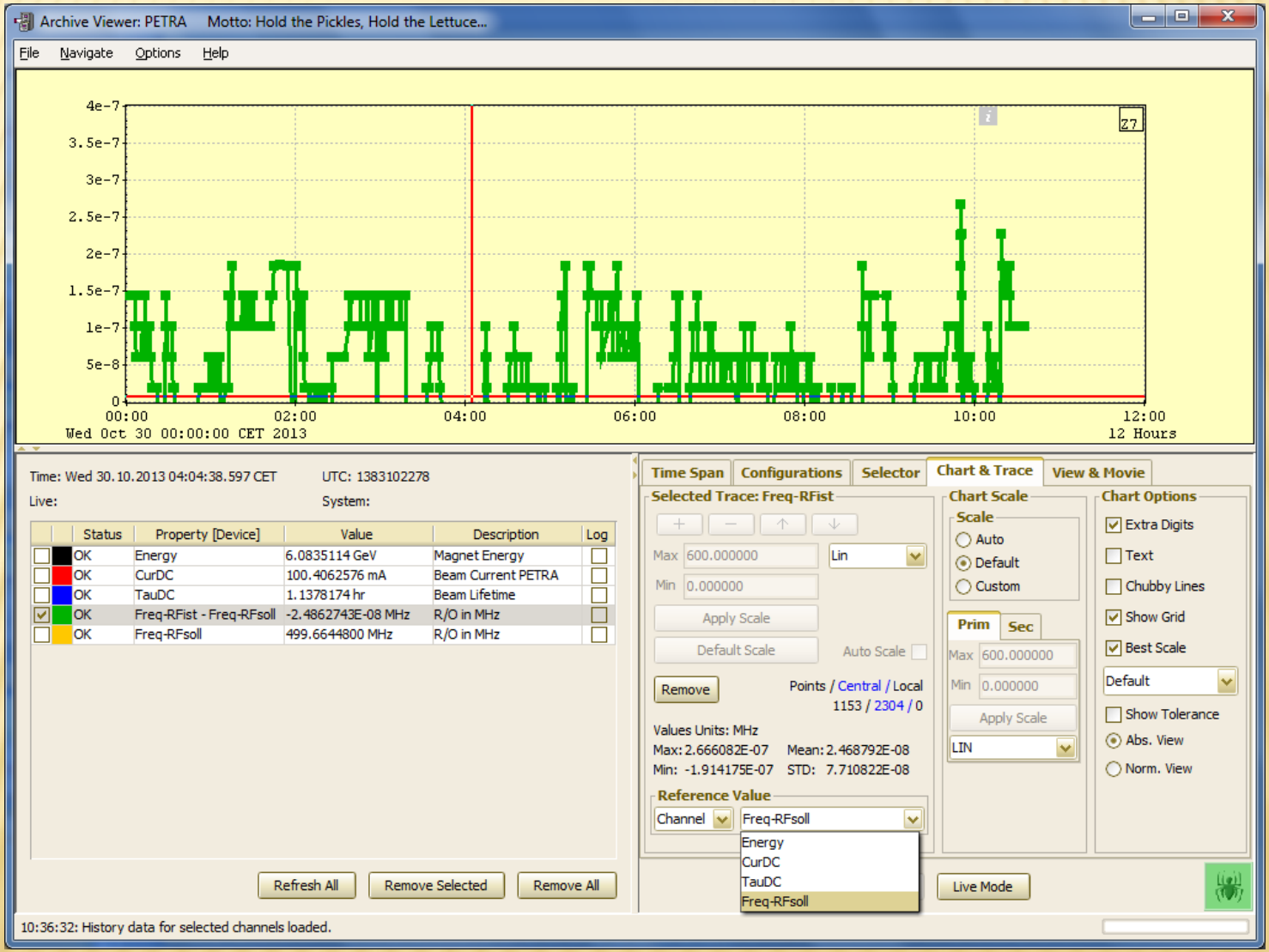
ARCHIVE VIEWER



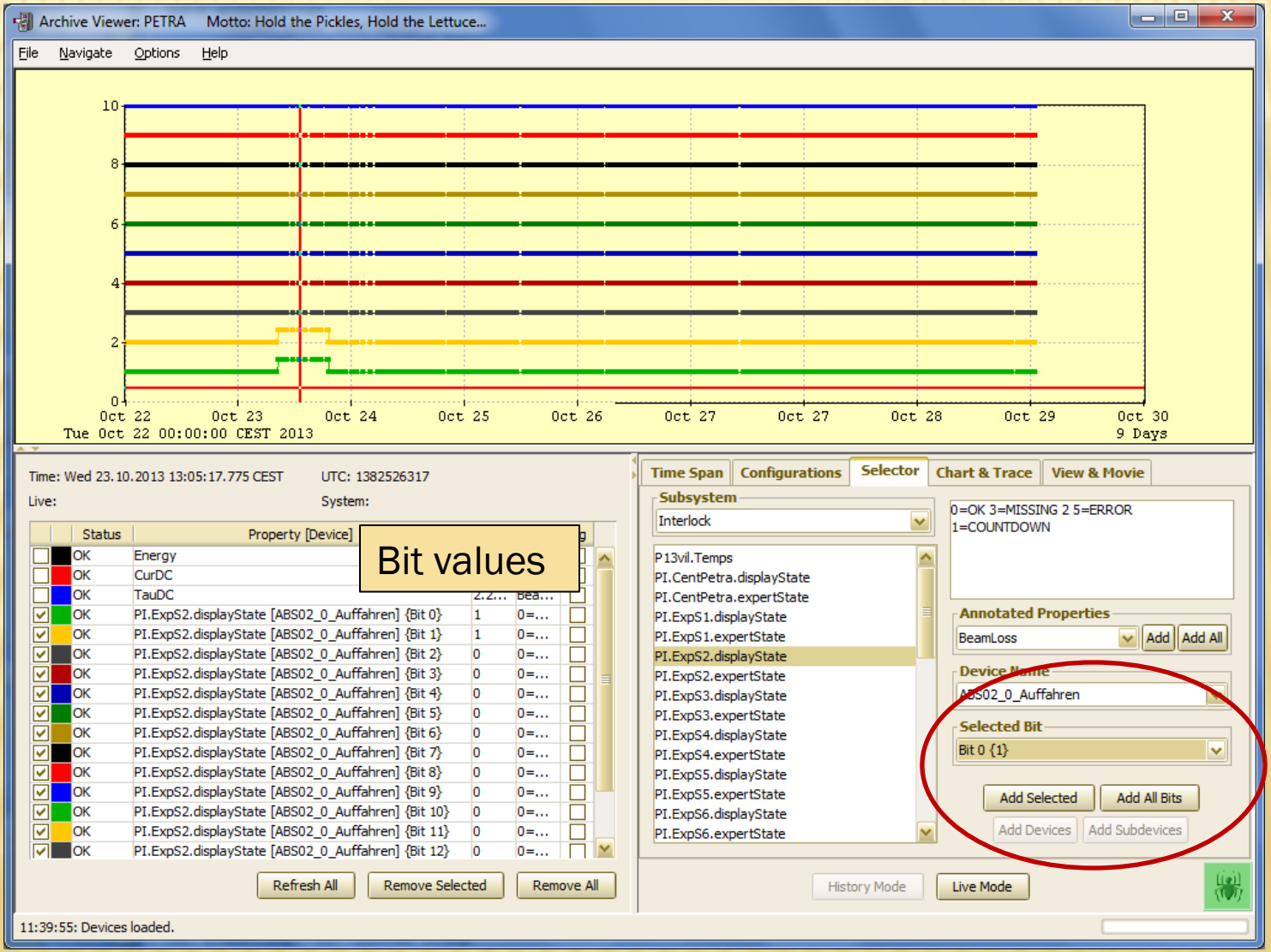
ARCHIVE VIEWER



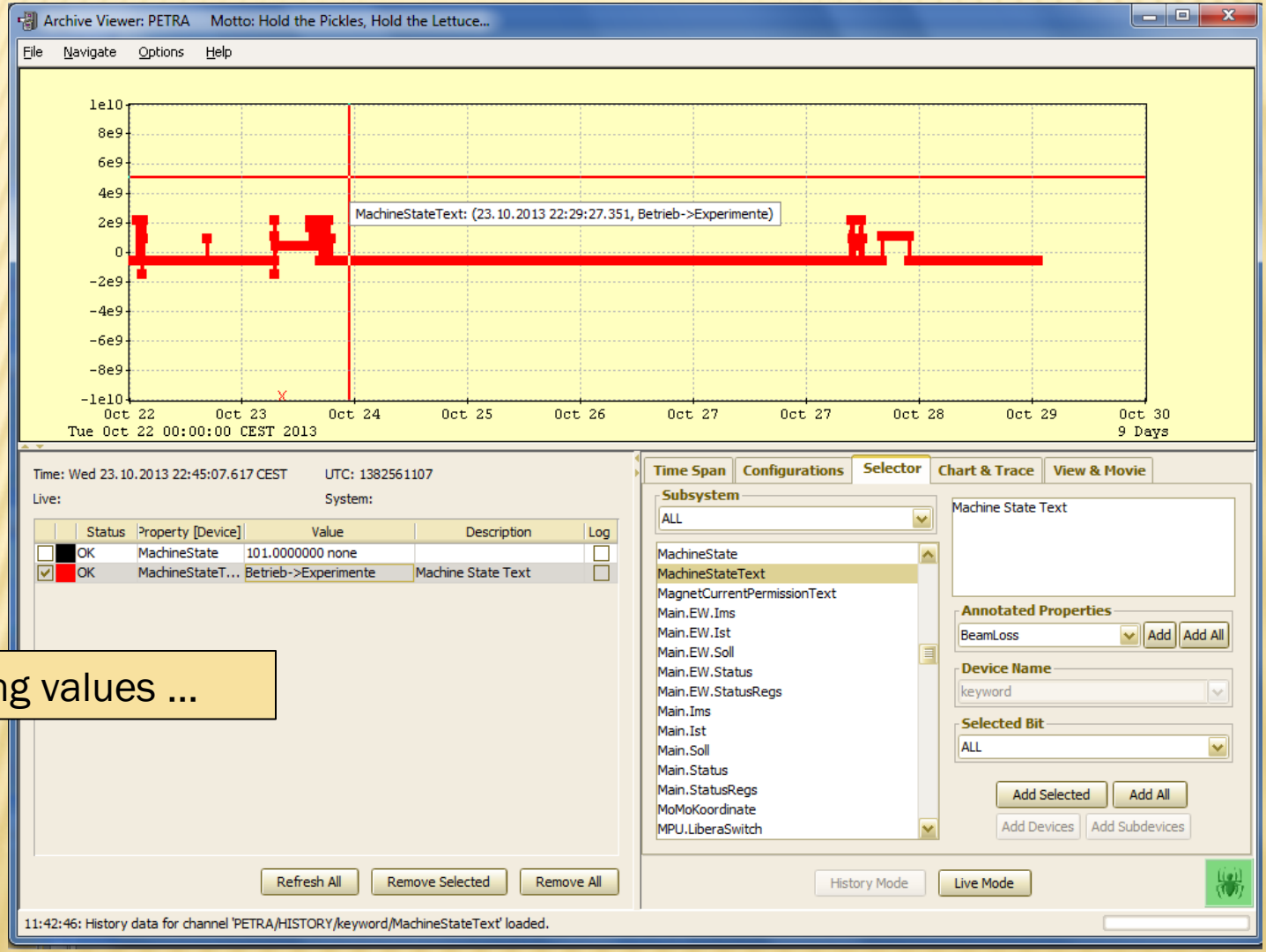
ARCHIVE VIEWER



ARCHIVE VIEWER

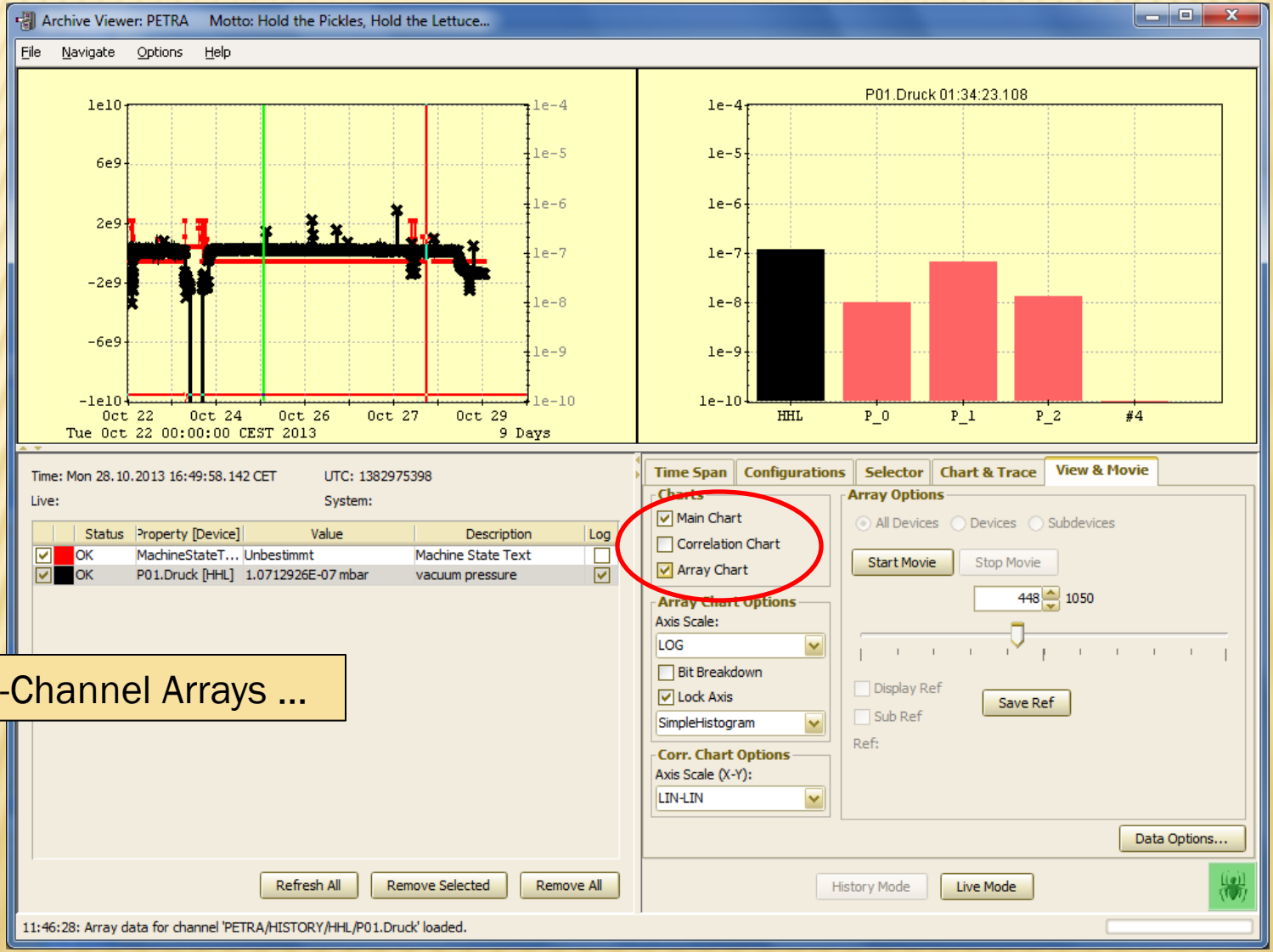


ARCHIVE VIEWER



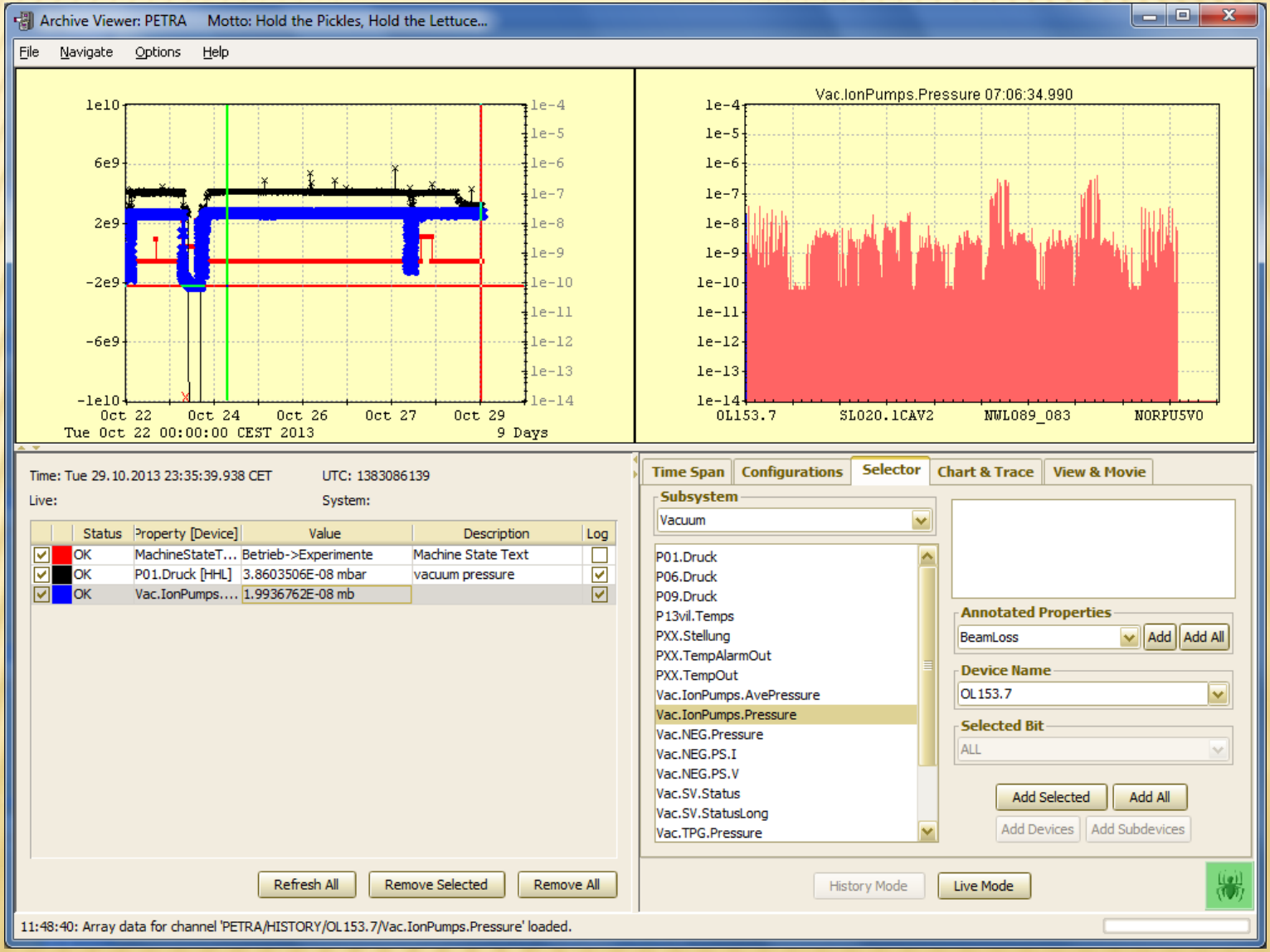
String values ...

ARCHIVER VIEWER

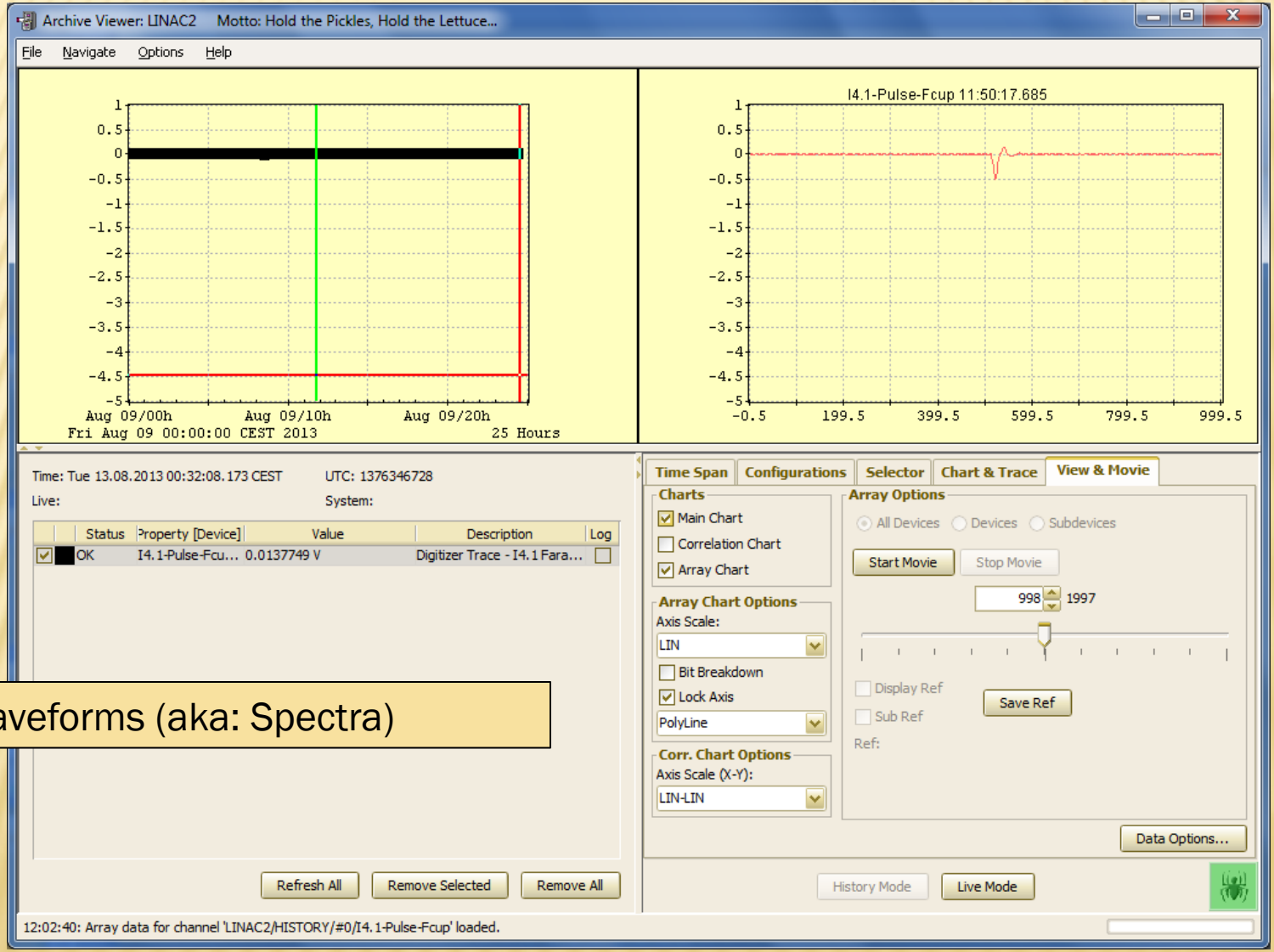


Multi-Channel Arrays ...

ARCHIVE VIEWER

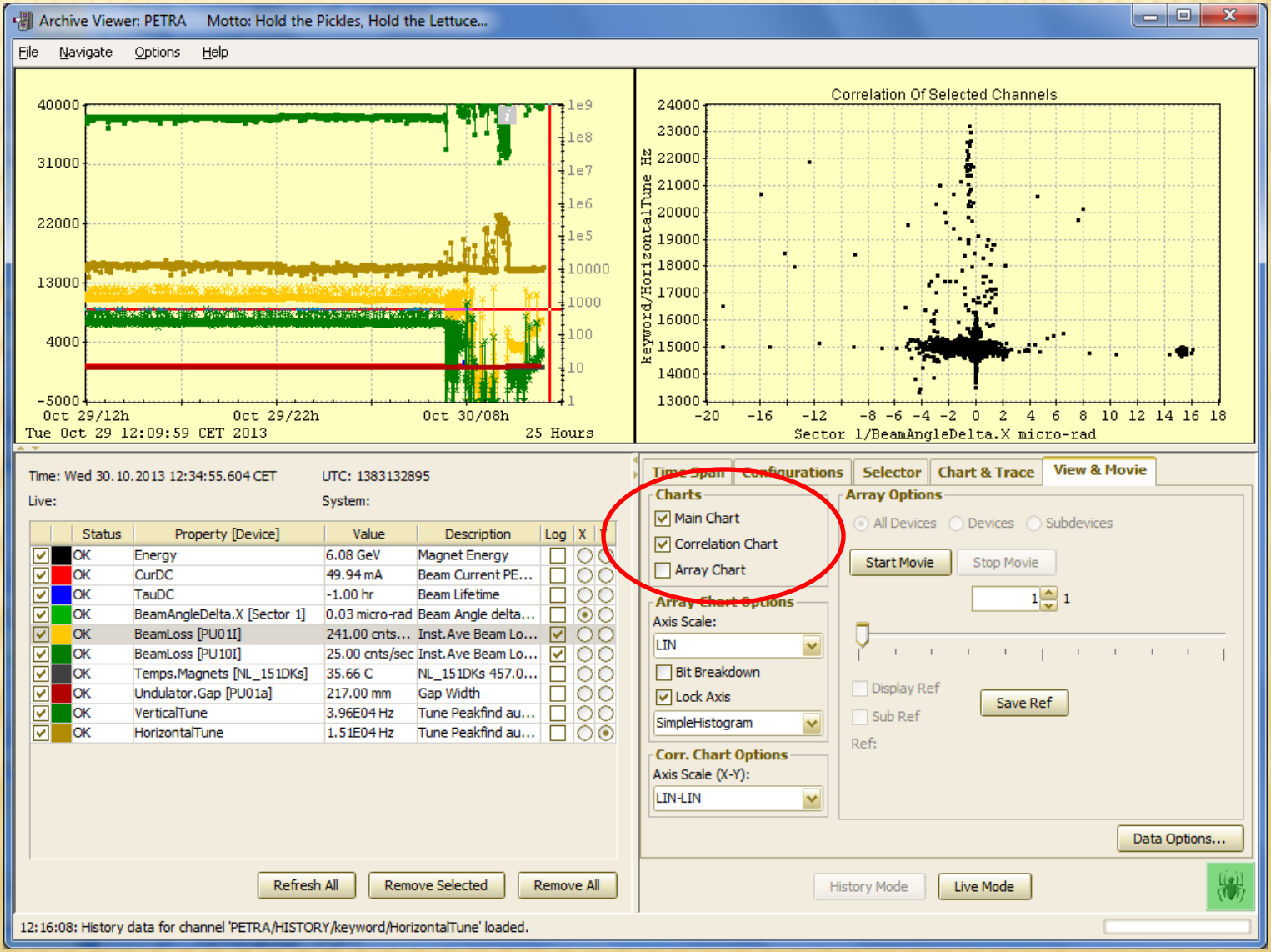


ARCHIVE VIEWER

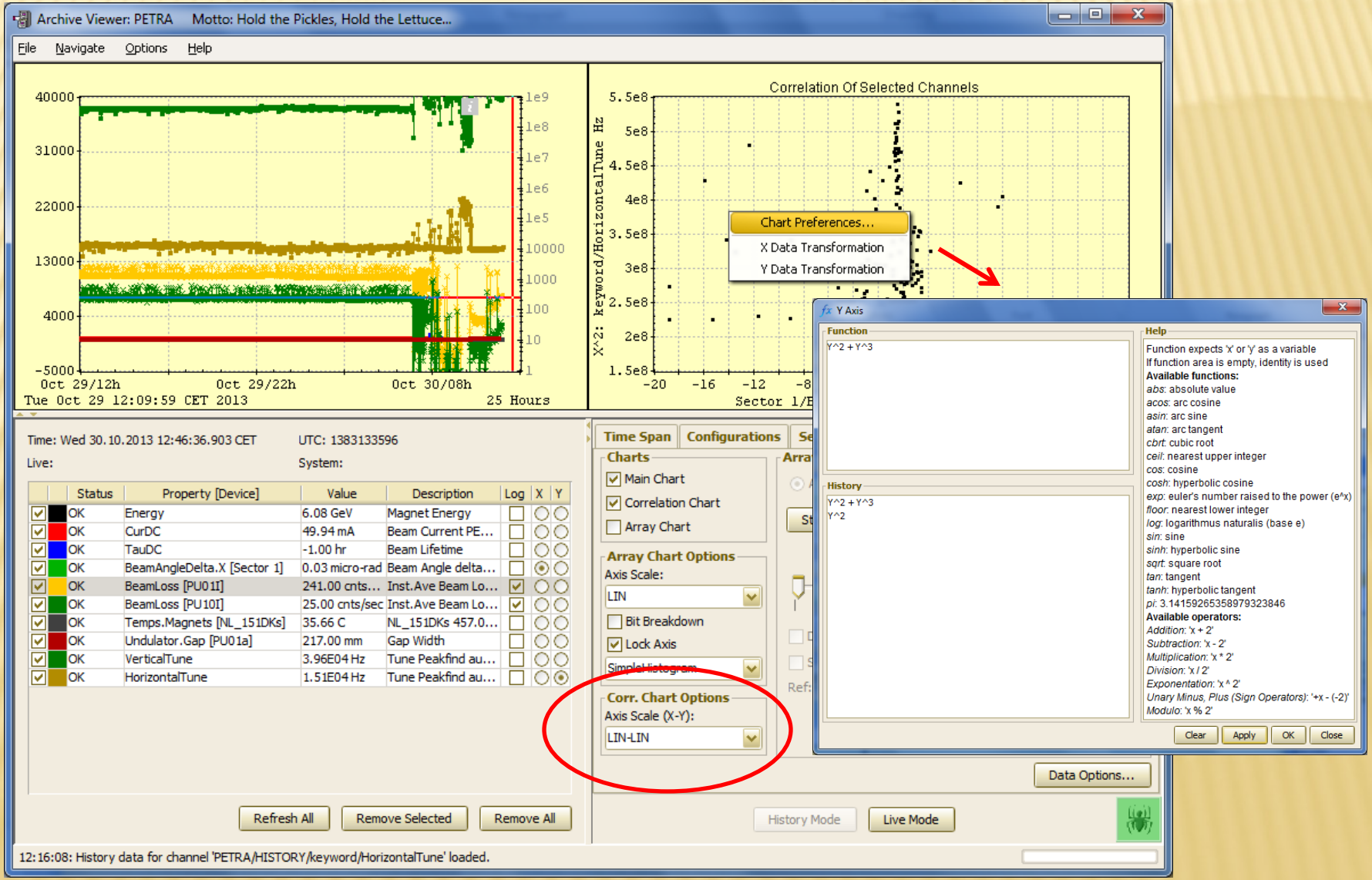


Waveforms (aka: Spectra)

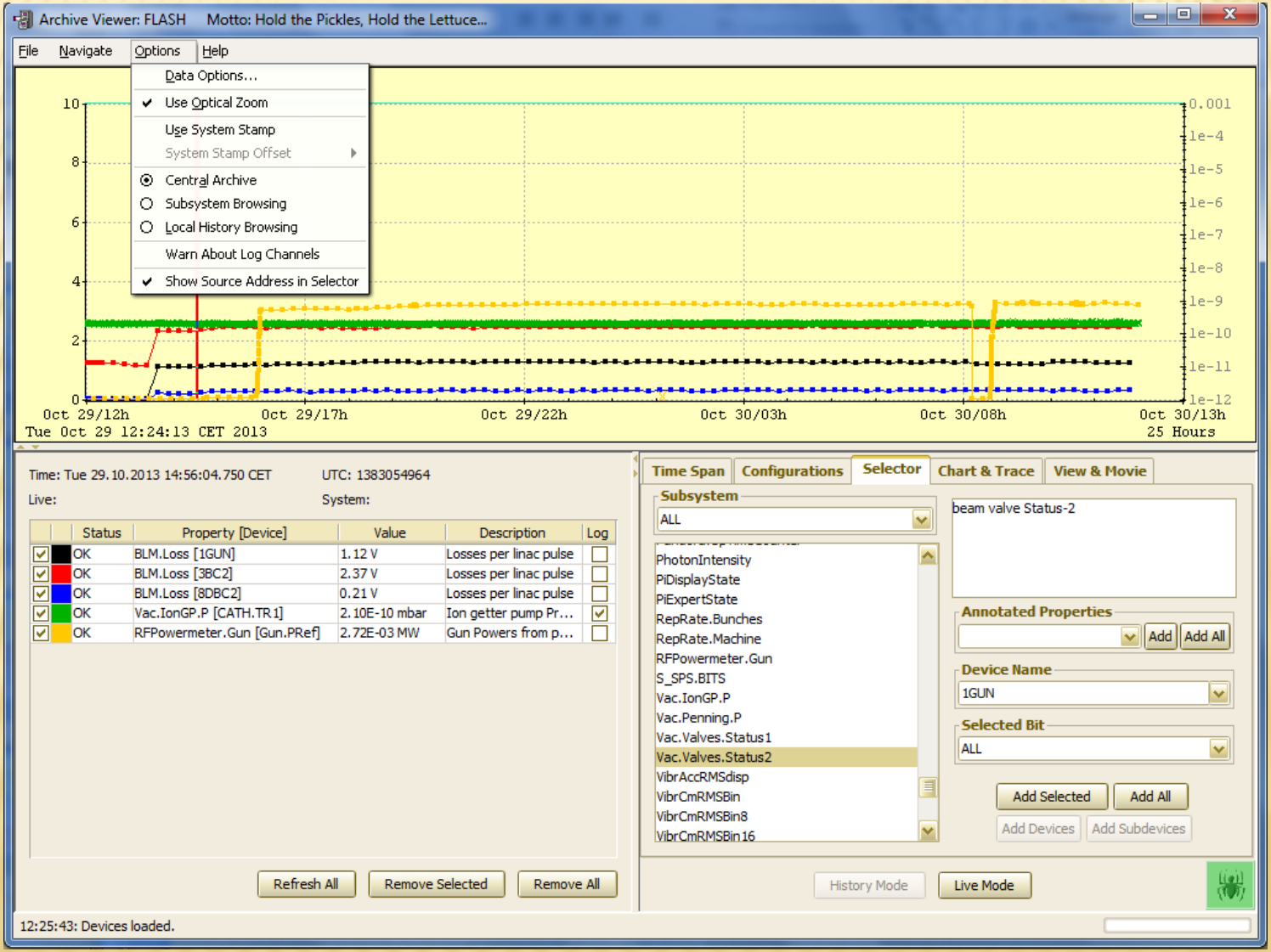
ARCHIVE VIEWER



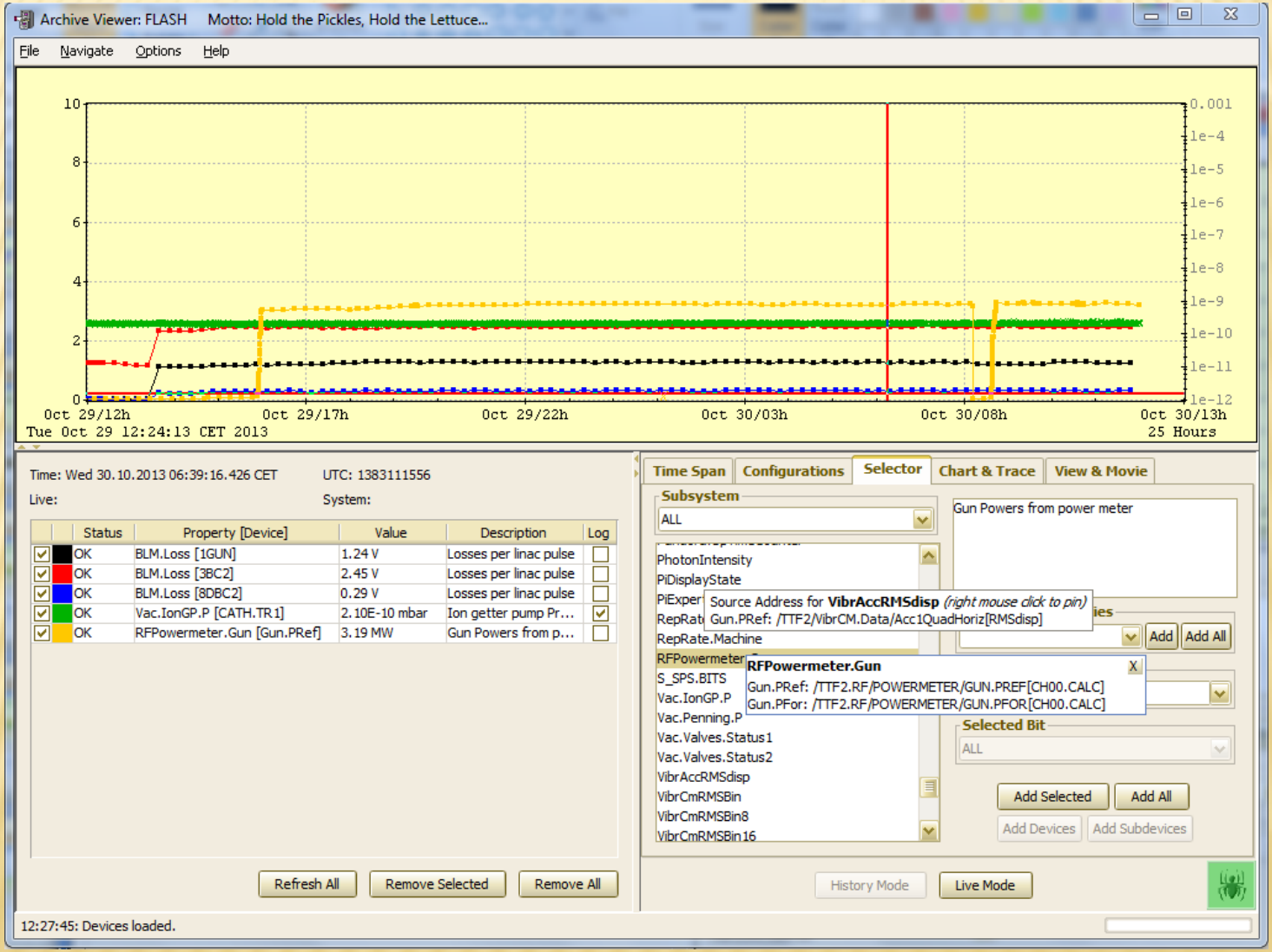
ARCHIVE VIEWER



ARCHIVE VIEWER

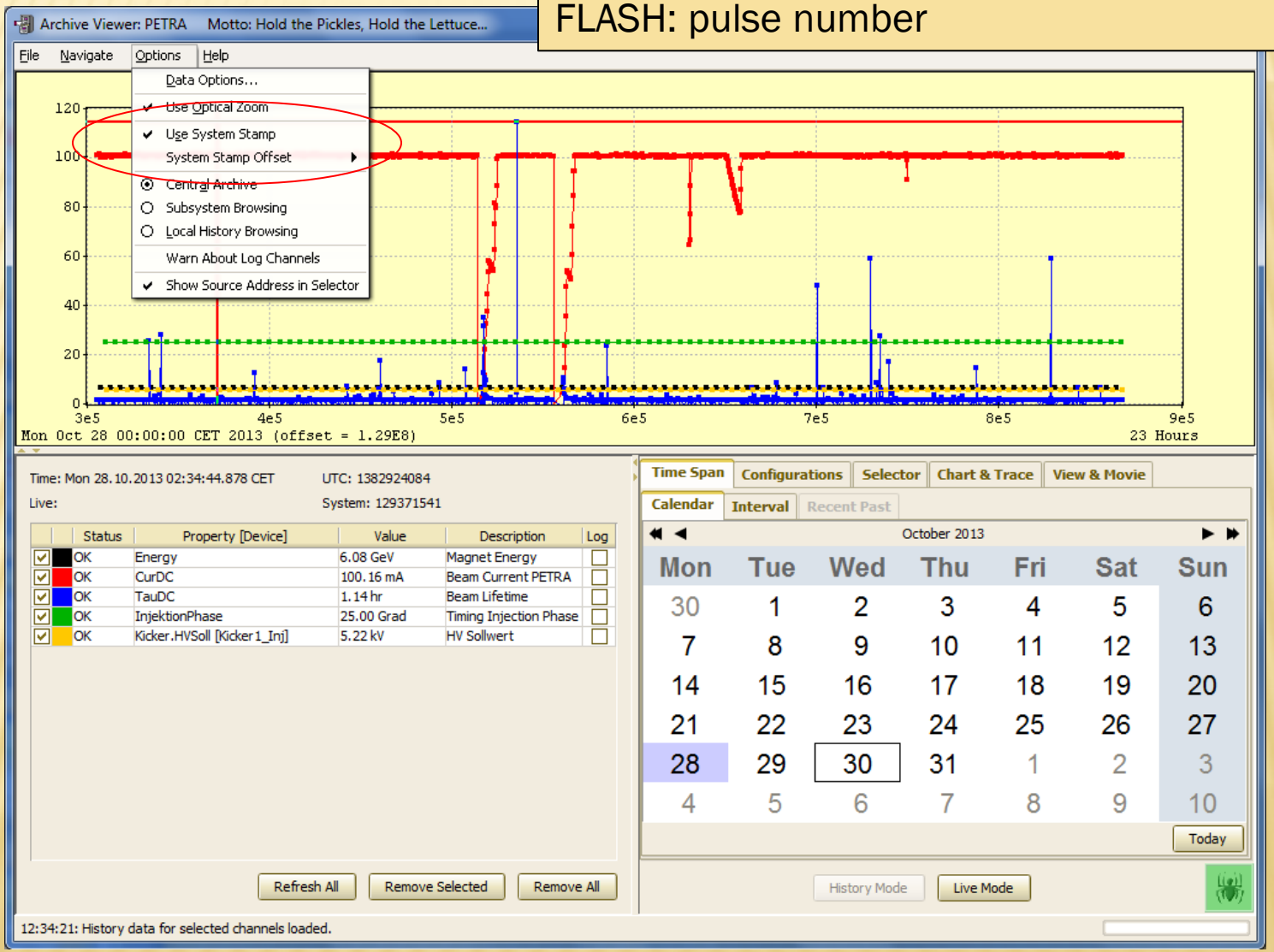


ARCHIVE VIEWER

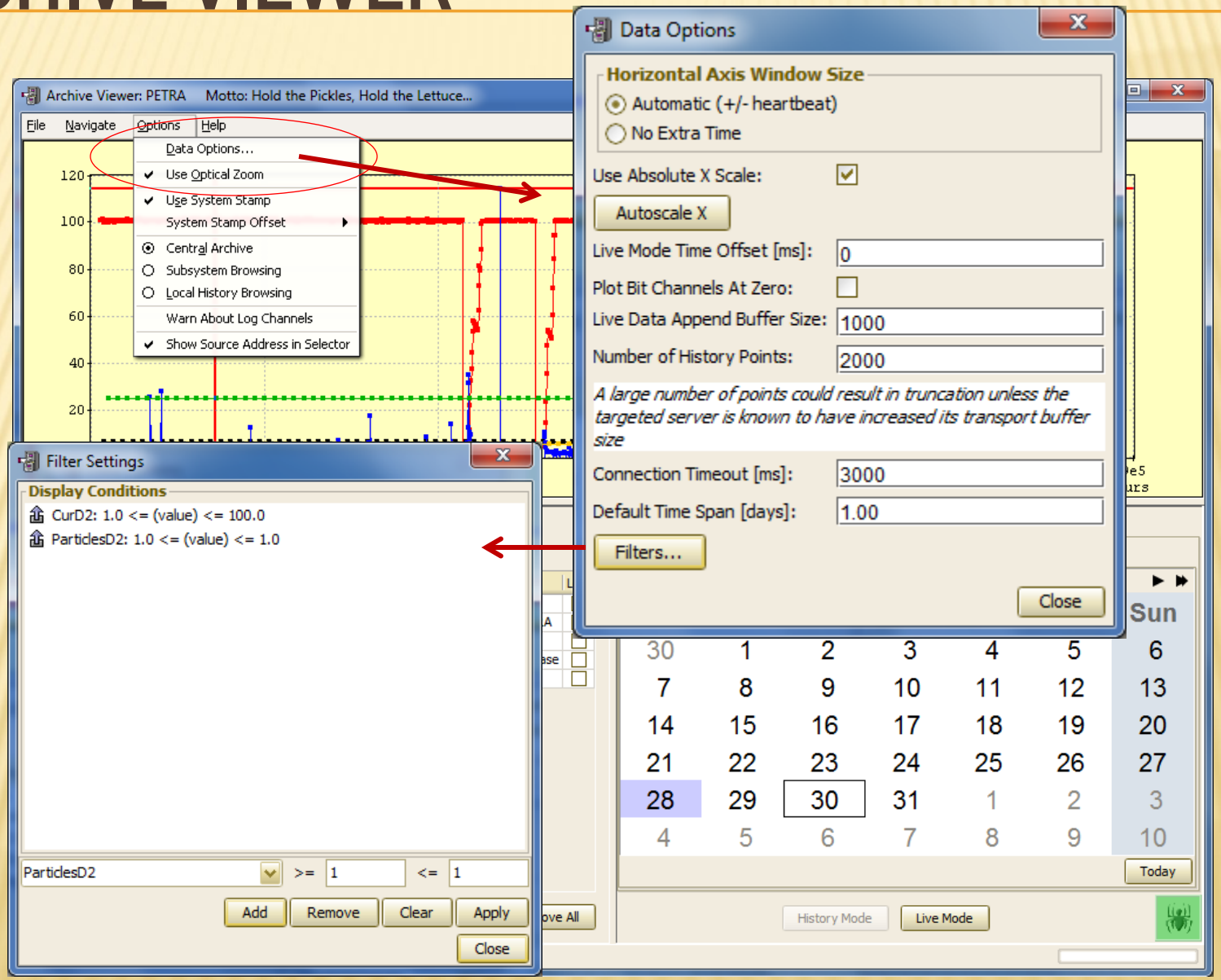


ARCHIVE VIEWER

Plotting vs. ‘system stamp’:
e.g. PETRA/DESY2/LINAC2 : cycle number
FLASH: pulse number



ARCHIVE VIEWER



ARCHIVE VIEWER

- Coming soon …
  - Save all data in range option
    - Now only saves data shown in chart(s)
  - ???