



TINE Release 4.0 News

(Dec 4, 2009: That was the month that was !)

“What a long, strange trip it’s been”

TINE Release 4.1 (bug fixes ...)

From last time :

■ C Library:

- *More threads !*
 - Globals thread
 - Server Multicast thread
 - Lazy-scheduling thread
 - TCP threads (send and rcv)
 - **Server cycle / client cycle on different threads**
- *In addition to :*
 - Shm thread
 - CM_STREAM threads
 - Background task threads
 - Stock property thread (some stock properties)
 - Optional Eqm thread

**Dependency on
configuration
style !**

TINE Release 4.1

(bug fixes, new features ...)

■ Java:

- multiple identical links
 - synchronous calls in callbacks
 - ->deadlock problem resolved.
 - resulted in ALL links disappearing ...
 - data copying problem with CF_STRUCT
- getHistory() using **CF_HISTORY**
 - Get *ANY* type of stored data
- MulticastMessenger
 - Simulate Drag-and-drop for LabView ...

Central Services

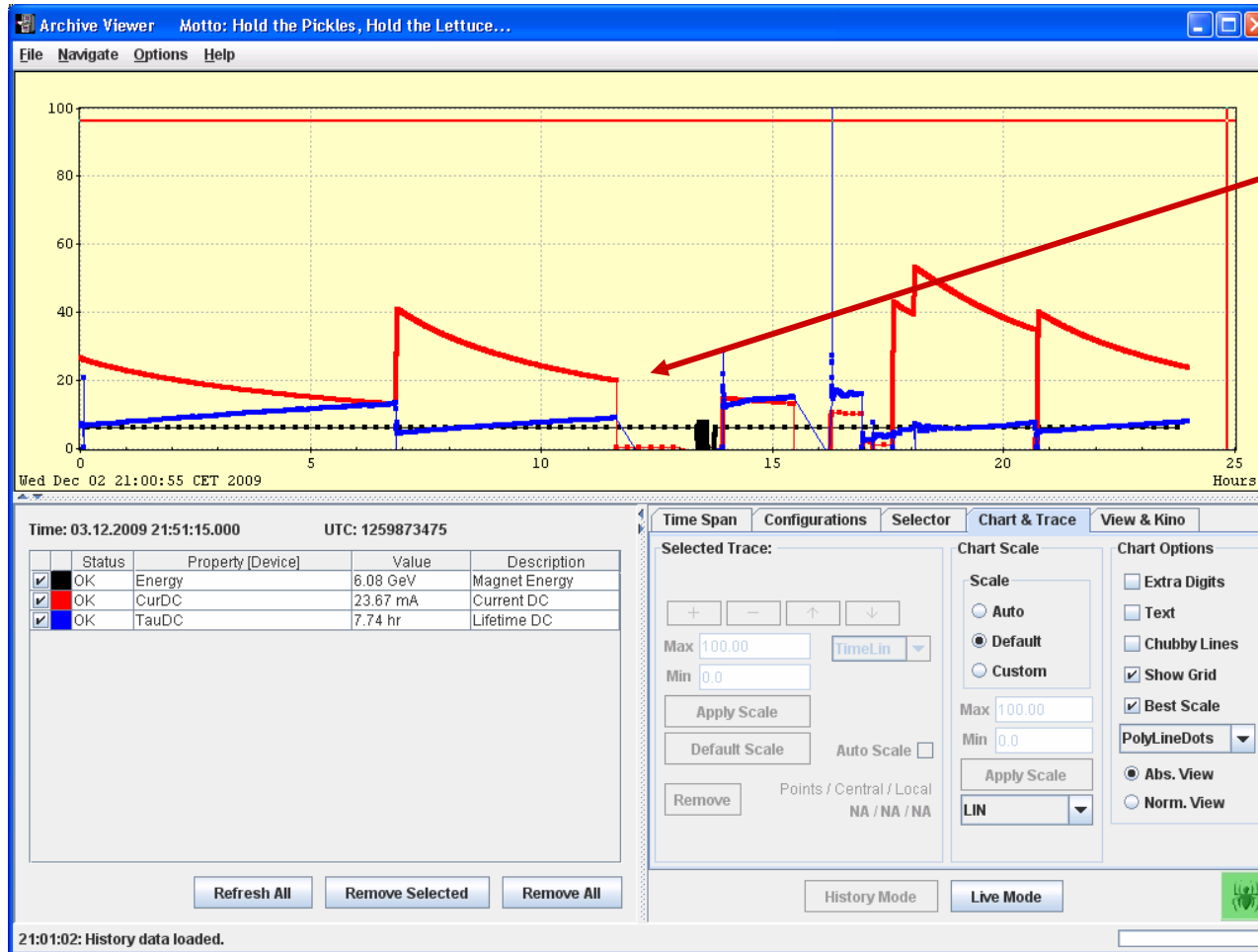
From last time :

■ Central Archive

- Recognized '*point of interest*' will generate a '*transition data point*' if necessary.
 - archived data filtered as we go along.
 - sudden jump from a 'constant' value could lead to a displayed 'ramp' if data points are connected with a poly-line.
 - **Insert a 'best-guess' at 'now' – polling interval if necessary.**

Now uses last collected record !

Central/Local Archive



Make sure this shows an accurate plot !

Central Services: Alarms

- CAS now interfaces with Alarm Viewer in *incremental* mode correctly!
 - Problem with sub-system timestamps resolved!
 - Number of alarms for “off-axis” server alarms.
 - Server is assigned an alarm system but also sets alarms for other alarm systems !
 - Now capable of handling requests for up to 5000 alarms !
 - Was limited to 1000 (due to nature of alarm message structure registration).
 - TODO: make use of a ‘DATA CHANGE’ window
 - DataChange alarms within a window of 1 minute (?) will not update the alarm timestamp!
 - Avoid alarm avalanche at the CAS due to minor (irrelevant?) data changes.

Central Services: Name resolution

- Acquiring the address of the ENS

- environment variable TINE_HOME points to location containing :

1

- cshosts.csv (contains address of configured ENSes)
- eqpdbname.csv, fecaddr.csv (local database with all relevant addresses).

2

- environment variable TINE_ENS gives the address of a single ENS

- multicast call asking for the address of *the* ENS.

3

- Like an execute() or ExecLink()
- The first ENS to answer is 'the' ENS

4

- request the DNS to resolve 'tineens' on the local domain.
 - Also gives a single ENS.

- No ENS or 'host unknown'?

5,6

- Look at local cache of 'last valid addresses'
- Look at local database in the TINE_HOME repository.

- If so configured: send multicast address resolution request

7

[autoLinkErrorAlarms]

- (middle-layer) servers with links to other servers
 - Set alarm with a link 'disappears'
 - Enabled by default!
 - Java: `TLinkFactory.getInstance().setAutoLinkErrorAlarms(false);`
 - C:
 - `SetAutoLinkErrorAlarms(false);`
 - Problems:
 - Alarms belong to a device!
 - No way to automatically associate a link error with a particular device.
 - Take first device of first equipment module, substitute server name for device name.

autoLinkErrorAlarms

Alarm Viewer: TEST

File View Options Navigate Help

Fatal	Error	Warning	Alarm Display
0	0	3	<input checked="" type="radio"/> Live <input type="radio"/> Archive

Thu Dec 03 22:32:31 Warning Severity >= 0 Selected/Total No. of Alarms: 3/3 Active Alarms Only

System	Fatal	Error	Warning	Device Name	Fatal	Error	Warning	System	Fatal	Error	Warning
Magnet Server	0	0	0	BPM	0	0	0	System	0	0	3
RF	0	0	0	BLM							
VAC	0	0	0	SineServers							
Temp.Yury	0	0	0	Pilotherm							
Temp.Susen	0	0	0	Current Monit							

System	Device Name	Message
System	WinSineServer	Link error
System	WinSineServer	Link error
System	TempP3	Not Responding

22:32:10: Alarms loaded.

```

Temp.Yury (6) alarms -> acquire 16
300 WinSineServer Not Responding Te
Temp.Susen (6) alarms -> acquire 16
5001 SineGen0 Hardware error Termin
SYS:Temp.Yury 16 alarms requested;
SYS:Temp.Susen 16 alarms requested;
  
```

BufferSize
:TimeToLive
AEOUT_GRACE_INTERVAL
eUserName
jllLink
_THRESHOLD
dMcaLink(TMcaLink)

Alarm Details: System: WinSineServer/WinSineServer

Property	Value
Alarm System	System
Device Server	WinSineServer
Alarm Device	WinSineServer
Device Text	device
FEC Name	SINEGEN.7
Host Address	131.169.9.107
HostName	McsXpDuval01.desy.de
Location	Bldg 30 Room 502
Alarm Text	link to dependent server is not open
Severity	8
Alarm Data Text	link which has the problem
Alarm Tag	Link error
URL	
Code	65
Format	4
Dimension	64
Mask	-1

Alarm Descriptor	Alarm Time	Duration
Oscillating	22:32:10.836 - Dec 03	32 sec
Oscillating	22:32:08.836 - Dec 03	30 sec
Oscillating	22:32:06.836 - Dec 03	28 sec
Oscillating	22:32:04.836 - Dec 03	26 sec
Oscillating	22:32:02.836 - Dec 03	24 sec
Oscillating	22:32:00.836 - Dec 03	22 sec

Alarm Data: link which has the problem

```

/TEST/SoSineServer/SineGen1[Amplitude]
  
```

Close

[TINE 4.1.1 (or maybe 4.1.2)]

- Structures with variable length formats allowed !
 - CF_STRING, CF_IMAGE, CF_SPECTRUM
 - Works, but not yet *finalized!*
 - Implement in C-Lib first (more difficult), then in java!
- Allow arrays of these types!

[The Problem with Strings ...]

- What is a 'String'?
 - Not a primitive!
 - In C (your operating system):
 - String = char *str;
 - i.e. an array of 'char's terminated with a '0'
 - String class (java, MFC, STL, VB, whatever)
 - str1 = str2 + str3;

```
str1 = (char *)malloc(strlen(str2) +
strlen(str3) + 1);
strcpy(str1, str2);
strcat(str1, str3);
free(str1); free(str3);
```

[The Problem with Strings ...]

- We like to use ‘**fixed-length**’, i.e. fixed-capacity strings:
 - CF_NAME16, CF_NAME32, CF_NAME64, etc.
 - Know *a priori* what the capacity is!
- An array of NAME64 things is easy to traverse, allocate, etc.
- A structure containing a NAME64 is also known *a priori* and is easy to serialize.

```
typedef struct
{
    float fval;
    int ival;
    NAME64 nval;
}
```


← Length = 72 bytes
(ascii char set)

[The Problem with Strings ...]

- Structures with 'free strings'
 - Length not known *a priori*!

```
typedef struct
{
    float fval;
    int ival;
    char *sval;
}
```

Length = 12 bytes (32-bit OS)
Length = 16 bytes (64-bit OS)



sval = "hello world!" or

sval = "A Tale of Two Cities\n\t by Charles Dickens\n\n\t It was the best of times, it was the worst of times ..."

[The Problem with Strings ...]

- Can be really 'fun' ...

```
typedef struct
{
    float fval;
    int ival;
    char *strArray[10];
    IMAGE imgArray[4];
    float anotherFloat;
    SPECTRUM specArray[4];
}
```

Format types with variable length (not known a priori).

A known length, but with an alignment problem ...

[The Problem with Strings ...]

- Current solutions
 - CF_STRING : an array of **free** strings
 - C and Java
 - `char *strarr[N];` or `String[] strarr = new String[N];`
 - CF_STRUCT containing CF_STRING, CF_IMAGE
 - C only (so far)
- Expand necessary **capacity** at both client and server when necessary
- Carry extended, i.e. **free** bytes following the fixed bytes
- Update the pointer (**reference**) to the free parts.
 - No bound or deeply-bound data
 - Caller required to copy data upon update (because the reference(s) could wander or disappear after the next call or update).

[The Problem with Strings ...]

- Other variable length types:

- DIMAGE

- Length gives number of frame bytes

- DSPECTRUM

- Length gives number of float elements

Now

- How to have arrays of these things?

- AIMAGE (adjustable IMAGE type)

- Length gives the number of images

- ASPECTRUM

- Length gives the number of spectra

Want

[Local Struct gives the 'Length']

- IMAGE

```
typedef struct
{
    TImageSourceHeader sourceHeader;
    TImageFrameHeader frameHeader;
    UINT32 frameBufferSize;
    BYTE *frameBuffer;
} DIMAGE;
```

- SPECTRUM

```
typedef struct
{
    char comment[80];
    SINT32 tm;
    float start;
    float inc;
    SINT32 status;
    UINT32 spectBufferSize;
    float *spectBuffer;
} DSPECTRUM;
```

[TINE 4.1.2 (maybe 4.1.3?)]

■ Roadmap

- Allow archiving of variable length types.
- Allow 'registered' long device names!
 - Device names can be up to 1024 chars long, BUT only 64 chars are 'registered'
 - Queries, redirection strings, archive/alarm headers, etc. all assume a device name capacity of 64 bytes.
 - Is possible in principle, but will be a major headache.

TINE 4.0.8: Site Deployment

- 'Office Deployment'
 - Always use 'latest and greatest'
 - Test as much as possible and release
 - 'extreme program' through the rough spots
 - 'The way of greatest progress'
 - But ...
 - Critical servers and applications need to start in a 'known' configuration
- 'Field Deployment'
 - 'Stable' repository updated less frequently, on maintenance periods, and after numerous 'acid' tests.

Deployment Proposal (win):

- Existing environment variables
 - USE_DEBUG_DLLS
 - Gets set of debug libraries (with symbol tables)
 - Include the 'office' jar files as well
 - ACC_TYPE
 - PETRA, DESY2, etc.
 - Get libraries from a 'stable' repository (?)
- 'Urgent' updates
 - Set the USE_DEBUG_DLLS variable
 - Get the most recent builds of all libraries from office repository
- Maintenance Days
 - Synchronize 'stable' repository with office repository
 - Remove all 'USE_DEBUG_DLLS' env variables for the ACC stations.
 - Reboot everything and keep your eyes open ...
- Non win domain stations:
 - Updated always at the discretion of the responsible parties

[UPDATE Notification Proposal]

- Use the current Users-Meeting Mailing list (until people start complaining)
- 'Each' new build of relevant DLL or jar File notifies those on the list
 - Short comment as to what changed (same as SVN comment)
 - 'Attention' levels
 - Green
 - Very minor or no changes (maybe just a re-build)
 - No side-effects expected anywhere (but of course, you never know)
 - Yellow
 - Some re-factoring of code sections
 - Possible side-effects in noted areas (checked as far as possible, but please make sure everything in the areas noted still works)
 - Red
 - Re-factoring in protocol or configuration sections, performance enhancements
 - Possible side-effects in startup or behavior (checked as far as possible, but please make sure everything works as soon as possible).