TINE Release 4.0 News

(now including Java News !)

TINE Kernel

- New error : invalid_structure_size (R. Weck)
  - Tagged structure sends the right ‘tag’ but does not match the locally registered length!
  - (was sending invalid_structure_tag)
- Default ‘Burst Limit’ now 1000
  - Only ‘flow control’ parameter available.
  - How many consecutive data packets sent before breaking off to do “something else.”
  - Was 20
- New routine: fwdWildCard() (Doocs)
  - Wildcard calls requesting format CF_USTRING can be forwarded “as is” to the registered property handler.
  - How to respond to “/PETRA/VAC/*” when the property doesn’t return a simple type (e.g. an ADDRESS -> INTINTINTINT)?
  - Need a “carrier” format
    - NAME64DBLDBL can send the device name, the value, and the status for each device that matches the wildcard. (value is a simple type)
    - USTRING can kluge INTINTINTINT
- Global Synchronization Tolerance now 100 msec

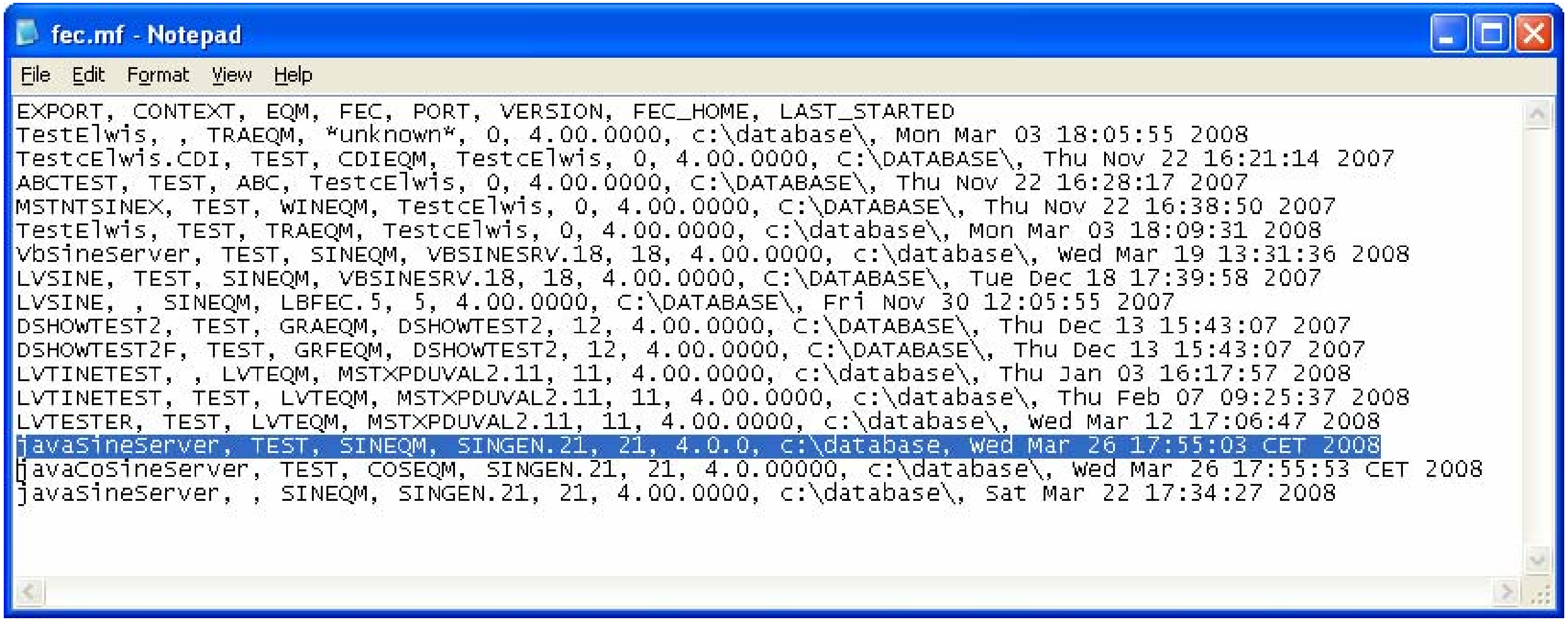
Central Services

- Group Equipment Name Server (GENS) now acquires member device names asynchronously.
- “Site” servers now in context “SITE”
  - -> Were in context “SERVICE”
  - ENS
  - GENS
  - Event Server
  - Site Globals Server
  - Time Synchronization Server
  - Central Logging Server

Java News

- Multicast catch-22 now solved on Linux 2.6 (Igor)
- Release 4.0 multicast scheme now functional !
- Java Servers can now receive an extended string size (up to 1024 chars)
- Java Servers are now case insensitive regarding devices and properties.
- Java Servers now monitor the TINE Time Synchronization server.
- Java Servers now update the TINE Server Manifest.
    - SystemDrive:\tine\cache\tine.mf (windows)
    - /var/tmp/tine/cache/tine.mf (unix, mac)

Fec Manifest Example :



```
EXPORT, CONTEXT, EQM, FEC, PORT, VERSION, FEC_HOME, LAST_STARTED
TestElwis, , TRAEQM, *unknown*, 0, 4.00.0000, c:\database\, Mon Mar 03 18:05:55 2008
TestcElwis.CDI, TEST, CDIEQM, TestcElwis, 0, 4.00.0000, C:\DATABASE\, Thu Nov 22 16:21:14 2007
ABCTEST, TEST, ABC, TestcElwis, 0, 4.00.0000, C:\DATABASE\, Thu Nov 22 16:28:17 2007
MSTNTSINEX, TEST, WINEQM, TestcElwis, 0, 4.00.0000, C:\DATABASE\, Thu Nov 22 16:38:50 2007
TestElwis, TEST, TRAEQM, TestcElwis, 0, 4.00.0000, c:\database\, Mon Mar 03 18:09:31 2008
VbSineServer, TEST, SINEQM, VBSINESRV.18, 18, 4.00.0000, c:\database\, Wed Mar 19 13:31:36 2008
LVSINE, TEST, SINEQM, VBSINESRV.18, 18, 4.00.0000, C:\DATABASE\, Tue Dec 18 17:39:58 2007
LVSINE, , SINEQM, LBFEC.5, 5, 4.00.0000, C:\DATABASE\, Fri Nov 30 12:05:55 2007
DSHOWTEST2, TEST, GRAEQM, DSHOWTEST2, 12, 4.00.0000, C:\DATABASE\, Thu Dec 13 15:43:07 2007
DSHOWTEST2F, TEST, GRFEQM, DSHOWTEST2, 12, 4.00.0000, C:\DATABASE\, Thu Dec 13 15:43:07 2007
LVTINETEST, , LVTEQM, MSTXPDUVAL2.11, 11, 4.00.0000, c:\database\, Thu Jan 03 16:17:57 2008
LVTINETEST, TEST, LVTEQM, MSTXPDUVAL2.11, 11, 4.00.0000, c:\database\, Thu Feb 07 09:25:37 2008
LVTESTER, TEST, LVTEQM, MSTXPDUVAL2.11, 11, 4.00.0000, c:\database\, Wed Mar 12 17:06:47 2008
javaSineServer, TEST, SINEQM, SINGEN.21, 21, 4.0.0, c:\database, Wed Mar 26 17:55:03 CET 2008
javaCoSineServer, TEST, COSEQM, SINGEN.21, 21, 4.0.00000, c:\database\, Wed Mar 26 17:55:53 CET 2008
javaSineServer, , SINEQM, SINGEN.21, 21, 4.00.0000, c:\database\, Sat Mar 22 17:34:27 2008
```

More Java News

- Local History can use non-fragmented files
  - Important for NTFS
- Central Logging Server API in place
  - ClsLog.log() <- submits an entry
  - ClsLog.getEntries() < retrieves entries
- Java Clients now update the local address cache
  - Same location as the manifest
- TQuery.getDevicePropertyInformation() deprecated.
  - Returned ‘old’ style information structure (with ‘short’ names).
  - Replace with TQuery.getPropertyInformation().
- Bitfield types now available (still rather ‘beta’)

Even More Java News

- New Data Types to handle archiving of ‘some’ compound data types (not yet fully tested).
  - NAME64DBL
  - NAME64DBLDBLDBL
  - DBLDBLDBL
- Efficient data retrieval requires a ‘carrier’ format
  - Stored format + timestamp
    - High resolution timestamp is a double !
  - float -> FLTINT or DBLDBL
  - int -> FLTINT or DBLDBL
  - FLTINT -> FLTINTINT or DBLDBLDBL
  - NAME16FI -> NAME64DBLDBLDBL
  - Etc.

At Some Future Time:

- Use ‘Systematic’ tagged structures to carry ‘any’ data type.
  - History calls !
    - Tag = ‘canonical’ tag + “+D” to add a double
    - e.g. “USTRING+D” if data are stored as USTRINGs.
  - or those Wildcard calls !
    - Tag = ‘canonical’ tag + “+ND” to add a name64 and double
    - e.g. “ADDRESS+ND” carries an INTINTINTINT plus a NAME64 (device name) plus a double (status) if the wildcard call asks for the list of ADDRESS types for all devices matching the wildcard pattern.

Release 4 java now late Beta !

- Merge Karol’s most recent changes
    - Tagged structures
    - fec.xml readout
    - Test BITFIELDS
    - Almost caught up with C Kernel.

Release 4.0.1 (coming soon)

- Add Video Data Type
- Add ‘deep data binding’ option :
    - No double (or triple) buffering
    - Use the caller’s data buffer address
        - And let him worry about allocation and thread-safety
    - Useful for large payloads that always change anyway (e.g. video frames)
- C-Kernel: Add alternate callback and dispatch functions which take a user-supplied void pointer.
    - Useful in both C and C++ interface.
    - e.g.

```
linkCb(int id, int cc) -> linkCbEx(void *ref,int id,int cc)

eqm(char *dev,char *prp,DTYPE *dout,DTYPE *din,short access)
-> eqm(void *ref,char *dev,char *prp,DTYPE *dout,DTYPE *din,short access)

eqmInit() -> eqmInit(void *ref)

etc.
```