

TINE CORE MEETING

30.08.2016

Noteworthy Changes (C-Lib and java)

- June 12: Allow local histories (and central archive) to keep and return *complex data* as well as *structures*.
 - Make use of new 'field index' in the calls:
 - specify a specific structure/data type 'field' to trend on.

```
typedef struct HstReq
{
    char *eqm; /**< target equipment module */
    char *prp; /**< target property */
    char *dev; /**< target device name */
    int index; /**< specific index */
    int fieldindex; /**< specific field index (if complex data type) */
    int sample; /**< specific raster sample */
} HstReqType;
```

Noteworthy Changes (C-Lib)

```
int loadLTSData(HstTblEntry *hst, char *fn, HstReqType *req, HstRngType rng, BYTE *data, int fmt, char *tag, int len)
{
# ifndef NO_LTS
int hFile = -1, nrecs, raster = 1;
int isSnapshot=FALSE, hasPointsOfInterest=FALSE, isPoiFile=FALSE, useSysStamp=FALSE;
int hPoolFile = -1, pidx = 0;
size_t poolPos, poolLen, plen = 0;
SINT32 cnt = 0, recsiz, reclen;
long pstart, pstop;
char fileName[256];
DTYPE d, dst;
int number=0, cc=0, p=0, sizPool=0, dstamp, sysstamp=0, usrstamp=0, datasys=rng.startstamp;
int fmtsize, poysize, depthInSeconds, indexLimit, pci=0, sizFile=0, nidx=(int)strlen(arcDbPath);
int rp, rfmt, recordPrefix = GetRecordPrefixLength(useMinimalStorage);
int foffset=0, ffmt=CF_NULL;
time_t timestamp;
double dband, dtimestamp, dlaststamp = -1, datetime = 0, dtnow = makeDateTimeStamp();
static double timebin = -1;
BYTE *tmpHstDatBuf, *nxtptr=NULL, tsBuffer[16];
# if !defined(VXWORKS) && !defined(SMALL_TINE_LIB)
struct stat sbuf;
int n, sizPoi;
char poiName[160], poi[32];
BYTE *dpoi;
HstTblEntry hstpoi;
HstReqType poireq;
HstRngType poirng;
# endif
if (rng.stoptime < rng.starttime) return 0;
}
```



```
}
else if (!isSimpleFormat((short)rfmt))
{
cc = getFieldInFormat(rfmt, NULL, hst->c.strTagOut, req->fieldindex, &foffset, &ffmt);
if (cc != 0) return -cc;
d.dFormat = ffmt;
d.data.bptr += foffset;
}
dst.dArrayLength = 1;
dst.dFormat = fmt;
}
```

Noteworthy C-Lib changes

- June 14: 'ReassignLinkData()'
 - Assign new reference for 'bound data' in an active Link
 - For use in rotating buffers ...

```
TINE_EXPORT int ReassignLinkData(int id, void *buffer, UINT32 length)
{
    int cc = 0;
    ConTblEntry *c;

    if (id < 1 || id >= nConnectionTableEntries) return invalid_index;
    if ((c=conTbl[id]) == NULL) return link_error;
    if (WaitForSystemMutex(hLinkDataMutex,-1) != 0) return mutex_error;
    if (c->allocatedBytes) ccerr(already_assigned);
    if (c->dataOut == NULL && buffer != NULL) ccerr(data_size_mismatch);
    if (c->dataOut != NULL && buffer == NULL) ccerr(data_size_mismatch);
    if (c->sizeBytesOut > length) ccerr(buffer_too_small);
    c->dataOut = buffer;
    feclog("reassigned link data buffer for link %d : %.196s",id,c->key);
err:
    ReleaseSystemMutex(hLinkDataMutex);
    return cc;
}
```

Noteworthy C-Lib changes

- July 15: make direct use of 'foregroundTTY' in SystemCycle() calls used in tcyclcr.c.
 - Established via initial call to SystemInit() or specific API (otherwise 'FALSE' => no parsing of std input in the foreground).
 - Do NOT want to do this in non-GUI Python/MatLab etc.

```
void cycleTmrTask(void)
{
    if (!gSystemInitialized)
    { /* don't spin in not initialized yet! */
        millisleep(gSystemDelay);
        return;
    }
    if (cycleTerminated) return;
    if (cycle_busy) return;
    cycle_busy = TRUE;
    SystemCycle(foregroundTTY);
    cycle_busy = FALSE;
    if (ServerExitCondition > 3) cycleTerminated = TRUE;
}
```

Noteworthy C-Lib changes

- June 16: add a 'retry' in GetSystemPropertyInformation() to accommodate 'Meta-Property' Look-alikes ...

```
TINE_EXPORT int GetSystemPropertyInformation(char *srv, char *prp, PrpQueryStruct **pqs, int *num)
{
    DTYPE dout, din, *dinpтр = NULL;
    int i, cc = 0, siz, isMeta=FALSE;
    short n, fmt;
    PrpQueryStruct *prpqs;
    XPropertyQueryStruct *xpqs;
    char *bpstr, *mpstr=NULL, *tprp=prp, *tgtprp="PROPERTIES";

    if (srv == NULL || num == NULL || pqs == NULL) return invalid_parameter;
    DTYPEZERO(dout); DTYPEZERO(din);
    if ((n = *num) <= 0)
```



```
    }
    pinfoCall:
    dout.dFormat = CF_STRUCT;
    dout.dArrayLength = n;
    strncpy(dout.dTag, "PRPQsr4", TAG_NAME_SIZE);
    dout.data.vptr = (void *)prpqs;
    cc = ExecLinkEx(srv, tgtprp, &dout, dinptr, CA_READ|CA_RETRY|CA_MUTABLE, QUERY_TO);
    if (HAS_DATA(cc))
    { /* modern server understood the query */
        *num = dout.dArrayLength;
        *pqs = prpqs;
        ccerr(0);
    }
    if (cc == non_existent_property && isMeta)
    { /* maybe not a meta-property call after all ? */
        isMeta = FALSE;
        din.data.cptr = prp;
        din.dArrayLength = 1;
        din.dFormat = CF_NAME64;
        goto pinfoCall;
    }
    if (cc == illegal_format || cc == invalid_structure_tag)
    { /* legacy server ? */
```

Noteworthy C-Lib Changes

- July 25: work around the (incredibly stupid) #define timezone in python 3.5 !
 - #if !defined(UNIX) && !defined(HAVE_GETTIMEOFDAY) around the prototype for gettimeofday !!

```
#if !defined(UNIX) && !defined(HAVE_GETTIMEOFDAY)
    TIME_EXPORT struct timeval *gettimeofday(struct timeval *t, struct timezone *tz);
#endif
```

prolog.h

```
#define HAVE_TIMEZONE
#define HAVE_GETTIMEOFDAY
#include "tine.h"
#include "tbufsrv.h"
#include "threader.h"
```

pytine.cpp

Noteworthy C-Lib Changes

- July 29 (Karol) Attempt to recognize inadvertent aliases.
 - introduce 'gDieOnFecIsAlias' and Set/Get routines (default = FALSE).
 - better check on a 'FEC match' in looking for existing link
 - “ServerB” uses same EQM Name, port and IP address for “FECB” as was formerly used for “ServerA” on “FECA”.
 - So: “ServerA” answers!
- trap problem renegotiating contract back to the normal contract length if the returned data size returns to 'normal'.

Noteworthy C-Lib Changes

- Alarm Manifests a la History Manifests ...
 - Regardless of 'how registered' (API or config file) a manifest of the current alarm and alarm watch information is dumped to file a short while after a server start.
 - Option to suppress the 'dump message' to std out.
 - (doocs 'log' files).

```
int dumpAlarmWatchManifest(char *eqm,int to_stdout)
{
    int cc = 0;
    # if defined(FS_DELIMITER) && !defined(FS_RDONLY)
    # define almwatmfhdr "LocalName, DeviceName, Property, Size, Format, Severity, SeverityHigh, SeverityLow, SeverityHighWarn, S
        "AlarmSystem, Mask, Normal, CountThreshold, High, Low, HighWarn, LowWarn, AlarmCode, AlarmCodeHigh, Alar
        "AlarmTagLow, Filter"
    char fs[512];
    AWSLstEntry *awe;
    FILE *fp = getAlmMfFile(eqm, "w", ALARM_WATCH_MANIFEST);
    if (fp == NULL) perror(file_error);
    fprintf(fp, "%s\n", almwatmfhdr);
    for (awe=almWatchList; awe != NULL; awe=awe->nxt)
    {
        if (eqm != NULL && strlen(eqm) > 0 && strncmp(eqm, awe->aws.eqm, EQM_NAME_SHORTSIZE) != 0) continue;
        getAlarmFilterString(awe->fltr, fs);
        fprintf(fp, "%.6s, %.64s, %.64s, %d, %d, %d, %d, %d, %d, %d, %d, %d, %f, %f, %f, %f, %d, %d, %.32s, %.32s, %.256s
            awe->aws.eqm, awe->aws.dev, awe->aws.prp, awe->aws.siz, awe->aws.fmt,
            awe->aws.sev, awe->aws.hisev, awe->aws.losev, awe->aws.hiwarnsev, awe->aws.lowarnsev,
            awe->aws.asys, awe->aws.mask, awe->aws.normal, awe->aws.cntThreshold, awe->aws.hi,
            awe->aws.lo, awe->aws.hiwarn, awe->aws.lowarn, awe->code, awe->codeHigh,
            awe->tagHigh, awe->tagLow, fs);
    }
    if (to_stdout) dbglog("local alarm watch info written");
err:
    if (fp != NULL) fclose(fp);
#endif
    return cc;
}
```

Noteworthy C-Lib Changes

- Alarm-specific Filters !
 - Previously: only applicable to 'watched' alarms.

```
typedef struct ADSListStruct
{
    ADS ads;
    # ifdef SMALL_TINE_LIB
    FilterLink *fltr;          /**< assigned filter */
    # endif
    struct ADSListStruct *next;
} ADSList;
```

almlib.h

```
FilterLink *getAlarmFilter(char *eqm, UINT32 code)
{
    # ifdef SMALL_TINE_LIB
    ExportListStruct *el=getExportListItem(eqm);
    if (el != NULL)
    { /* equipment module found ! */
        ADSList *lst;
        code = ALM_BASECODE(code);
        for (lst=el->adsTable; lst != NULL; lst=lst->next) if ((UINT32)code == lst->ads.alarmCode) return lst->fltr;
    }
    return NULL;
    # else
    eqm = eqm; code = code;
    # endif
    return NULL;
}
```

Noteworthy C-Lib Changes

- Clear the immediate partner in a watch table alarm if the value slides from e.g. 'warning' to 'real'.
- Automatic 'file_error' alarms if
 - Log files cannot be written
 - Local history files cannot be written.

Noteworthy C-Lib Changes

- Property Handlers (a la java servers)
 - In addition to or in lieu of the standard equipment module dispatch handler.
 - Internal equipment module for simple cases.
 - TODO: 'stock' handlers for e.g. attribute properties or commands.

Noteworthy C-Lib Changes

- Other Bug Fixes:
 - Check for 'overloaded properties' when acquiring a 'property list structure'.
 - Ensure that overloaded properties 'replicate' common information.
 - Repair 'save-and-restore' when non-WRITE overload property registered.
- Lars: added 'CA_NOCALLBACKS' for use in ExecLink()
 - Turns off callbacks during the synchronous execution of this call.

- Issue: 'replace' e.g.

```
#define out_of_range 37
```

with

```
const int out_of_range = 37;
```

???

Still To Do ...

- how to get and display and set (?) the alarmwatch and history filters
 - => work with 'parseable' strings
 - => stock property (command line also) "ALMFILTERS" (get/set almfilters)
 - input/output: alarm code and filter string
 - ---> a new 'struct' (int and 256 char string) or CF_KEYVALUE (with CODE:val and FILTER:parseable string as alternating entries ?)
 - or: CF_STRING (with CODE:val FILTER:string) (list of atomic strings)
 - read: get watchtable and specific alarm filters
 - write: apply/edit filter for watchtable or specific alarm (first check if the target alarm is being 'watched' -> apply to watchtable; else apply to specific alarm definition)
 - => stock property (command line) "HSTFILERS" (get/set hstfilters)
 - read: get local history filters
 - write: apply/edit filter for local history entry

Still To Do ...

- console command to mkhstfiles (on separate thread)
- console commands to change history or alarm settings on the fly
- stock handlers
- local history:
 - add 'CF_STRUCT' + tag to the API
- unix PIPE -> socket gets a 'temporarily unavailable' error !
- errors.log ? a la 'commands.log'

Still To Do ...

- IPv6 testing
- Release 5.0 ...