TINE CORE MEETING

28.11.2017

Feature Additions 4.6.2

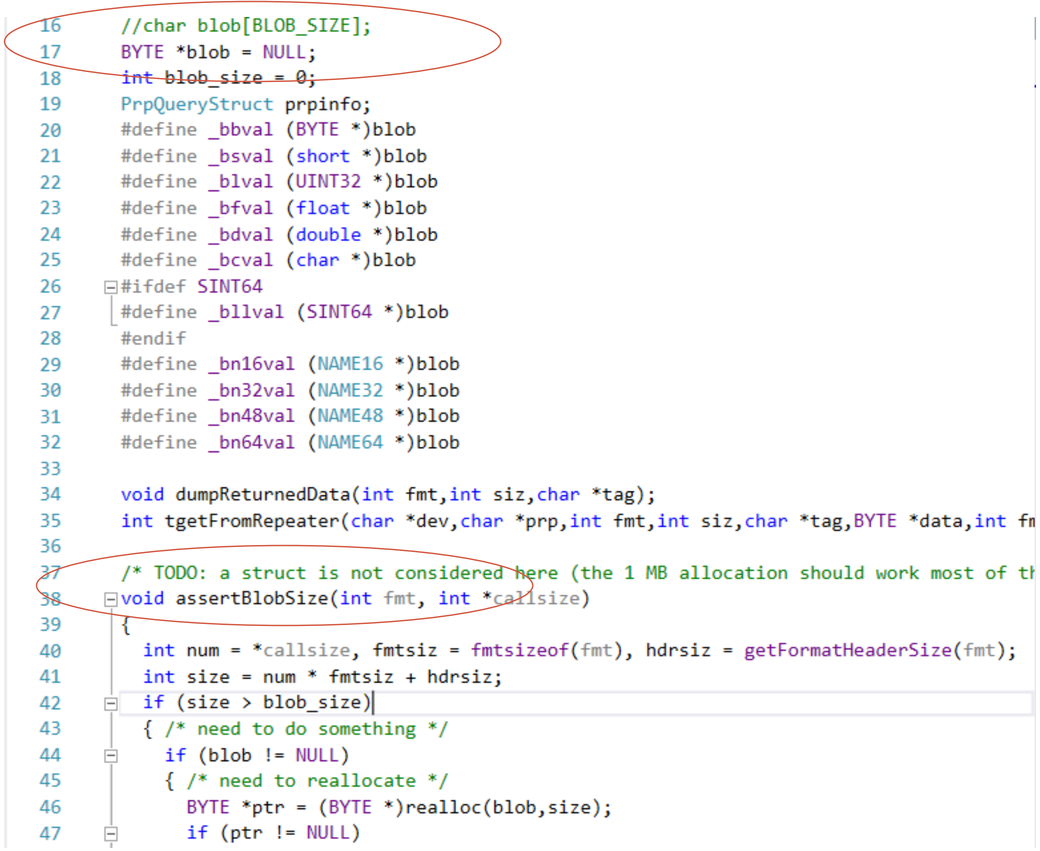
- To do:
    - Need a ‘contacts’ server.
        - Input: ‘a follower’ or a ‘followers list’
        - Output: associated email address(es)
            - Check : if ‘follower’ is a FEC then determine the ‘responsible list’
            - => a ‘follower’ could resolve to > 1 contact
        - Input: “send message to followers”
        - Output: determine the contacts list and email the message !

Andreas is working on it !

Bug Fixes and Features 4.6.2

- Utilities:
  - command line: talarms :
    - Seg fault when MPI tried to use it …

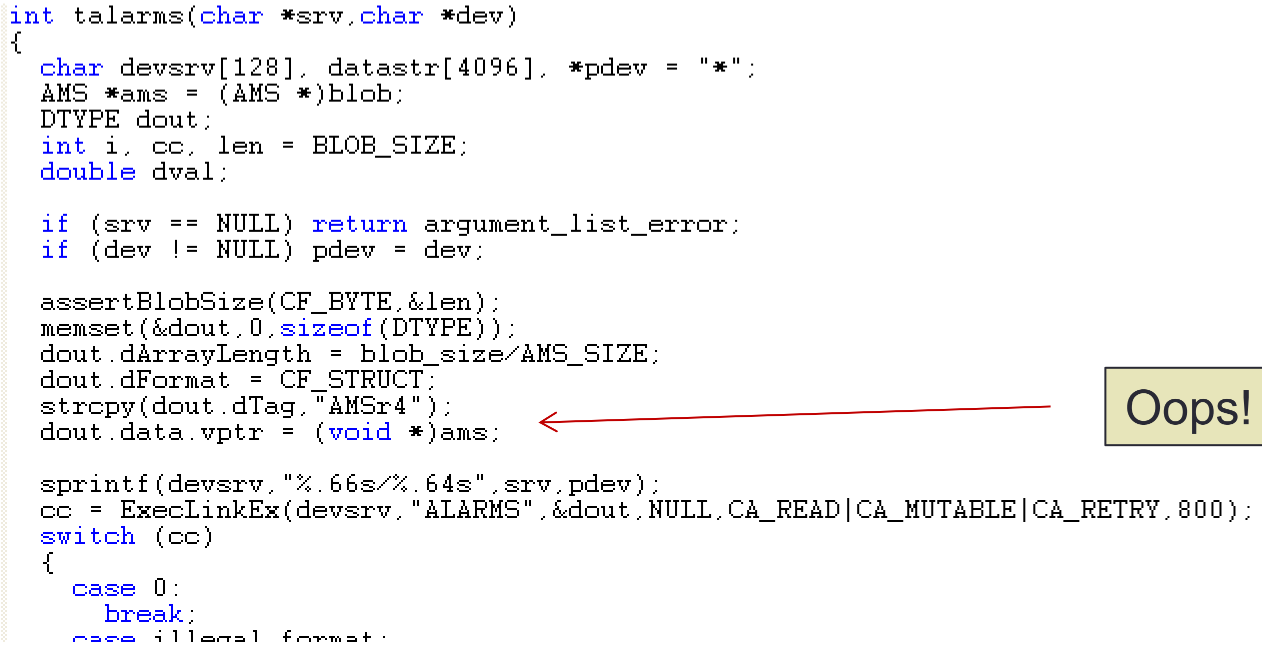
Resulted from a refactoring of the ‘blob’ working area that the command line tools use :



```c
//char blob[BLOB_SIZE];
BYTE *blob = NULL;
int blob_size = 0;
PrpQueryStruct prpinfo;
#define _bbval (BYTE *)blob
#define _bsval (short *)blob
#define _blval (UINT32 *)blob
#define _bfval (float *)blob
#define _bdval (double *)blob
#define _bcval (char *)blob
#ifdef SINT64
#define _bllval (SINT64 *)blob
#endif
#define _bn16val (NAME16 *)blob
#define _bn32val (NAME32 *)blob
#define _bn48val (NAME48 *)blob
#define _bn64val (NAME64 *)blob

void dumpReturnedData(int fmt,int siz,char *tag);
int tgetFromRepeater(char *dev,char *prp,int fmt,int siz,char *tag,BYTE *data,int fm

/* TODO: a struct is not considered here (the 1 MB allocation should work most of th
void assertBlobSize(int fmt, int *callsize)
{
  int num = *callsize, fmtsiz = fmtsizeof(fmt), hdrsiz = getFormatHeaderSize(fmt);
  int size = num * fmtsiz + hdrsiz;
  if (size > blob_size)
  { /* need to do something */
    if (blob != NULL)
    { /* need to reallocate */
      BYTE *ptr = (BYTE *)realloc(blob,size);
      if (ptr != NULL)
```

Bug Fixes and Features 4.6.2

- Originally:

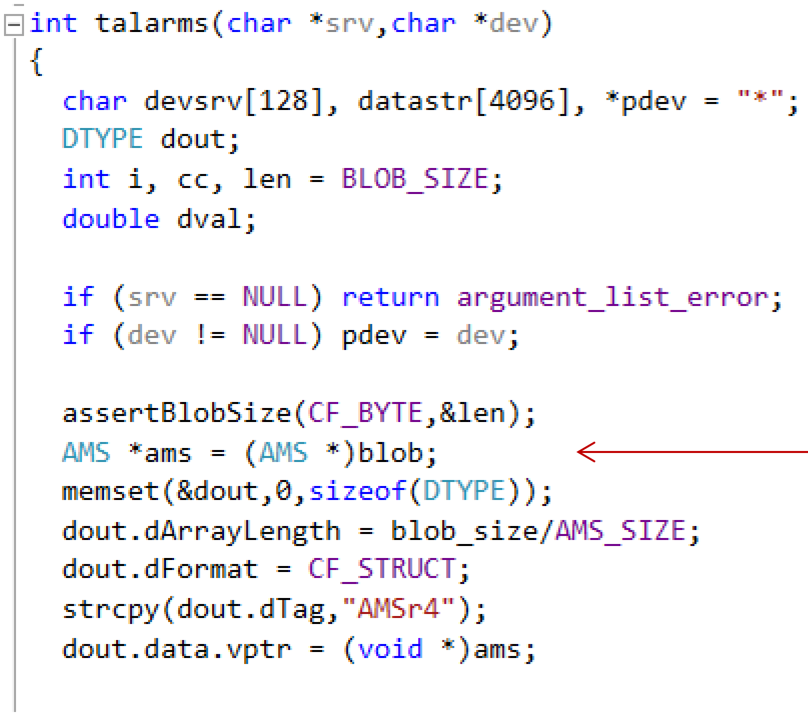


```
int talarms(char *srv,char *dev)
{
  char devsrv[128], datastr[4096], *pdev = "*";
  AMS *ams = (AMS *)blob;
  DTYPE dout;
  int i, cc, len = BLOB_SIZE;
  double dval;

  if (srv == NULL) return argument_list_error;
  if (dev != NULL) pdev = dev;

  assertBlobSize(CF_BYTE,&len);
  memset(&dout,0,sizeof(DTYPE));
  dout.dArrayLength = blob_size/AMS_SIZE;
  dout.dFormat = CF_STRUCT;
  strcpy(dout.dTag,"AMSr4");
  dout.data.vptr = (void *)ams;

  sprintf(devsrv,"%.66s/%.64s",srv,pdev);
  cc = ExecLinkEx(devsrv,"ALARMS",&dout,NULL,CA_READ|CA_MUTABLE|CA_RETRY,800);
  switch (cc)
  {
    case 0:
      break;
    case illegal_format:
```

Oops!

Bug Fixes and Features 4.6.2

- First Fix:



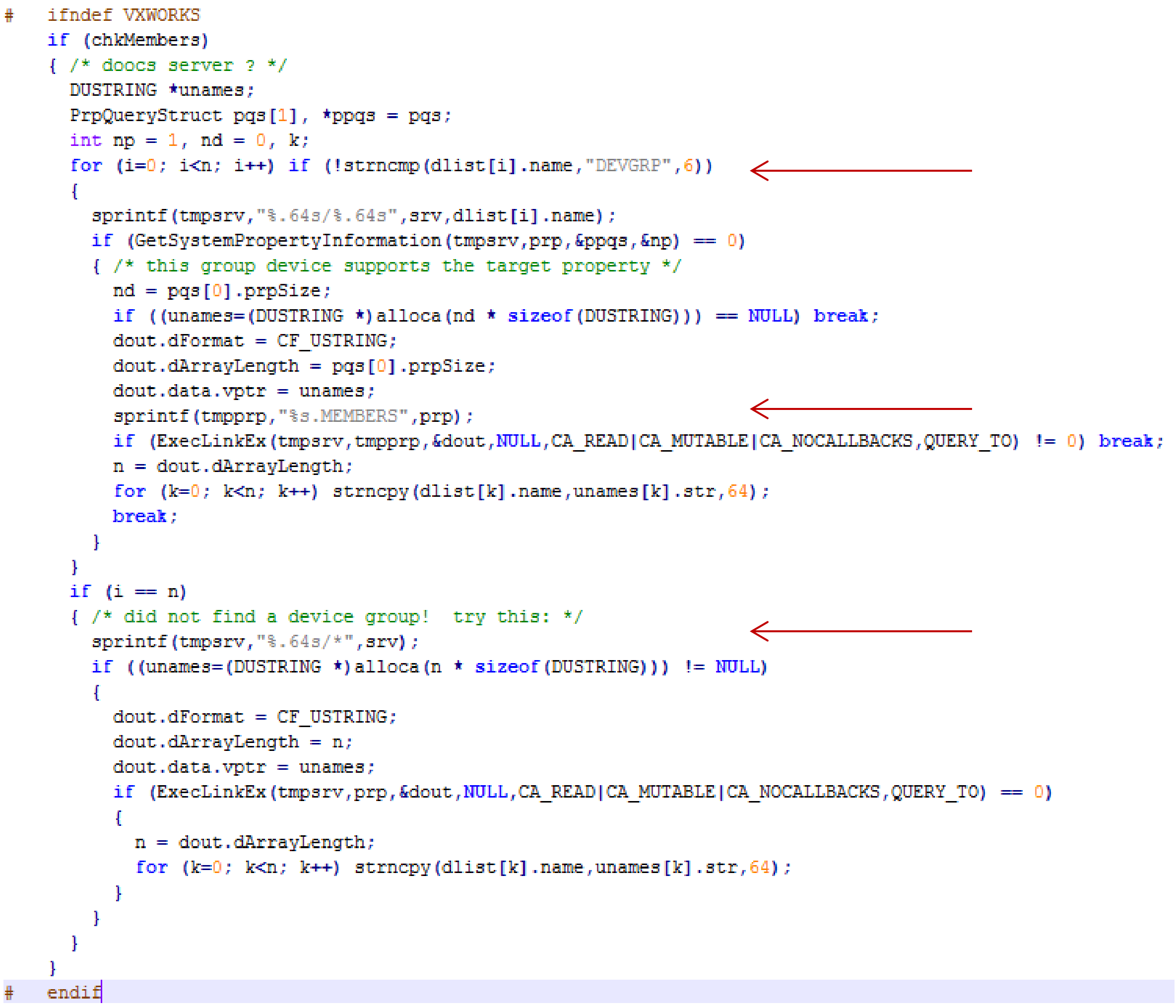
```
int talarms(char *srv,char *dev)
{
  char devsrv[128], datastr[4096], *pdev = "*";
  DTYPE dout;
  int i, cc, len = BLOB_SIZE;
  double dval;

  if (srv == NULL) return argument_list_error;
  if (dev != NULL) pdev = dev;

  assertBlobSize(CF_BYTE,&len);
  AMS *ams = (AMS *)blob;
  memset(&dout,0,sizeof(DTYPE));
  dout.dArrayLength = blob_size/AMS_SIZE;
  dout.dFormat = CF_STRUCT;
  strcpy(dout.dTag,"AMSr4");
  dout.data.vptr = (void *)ams;
```

Oops! This is C, not C++.

Bug Fixes and Features 4.6.2

Inside ‘GetSystemDevices() :

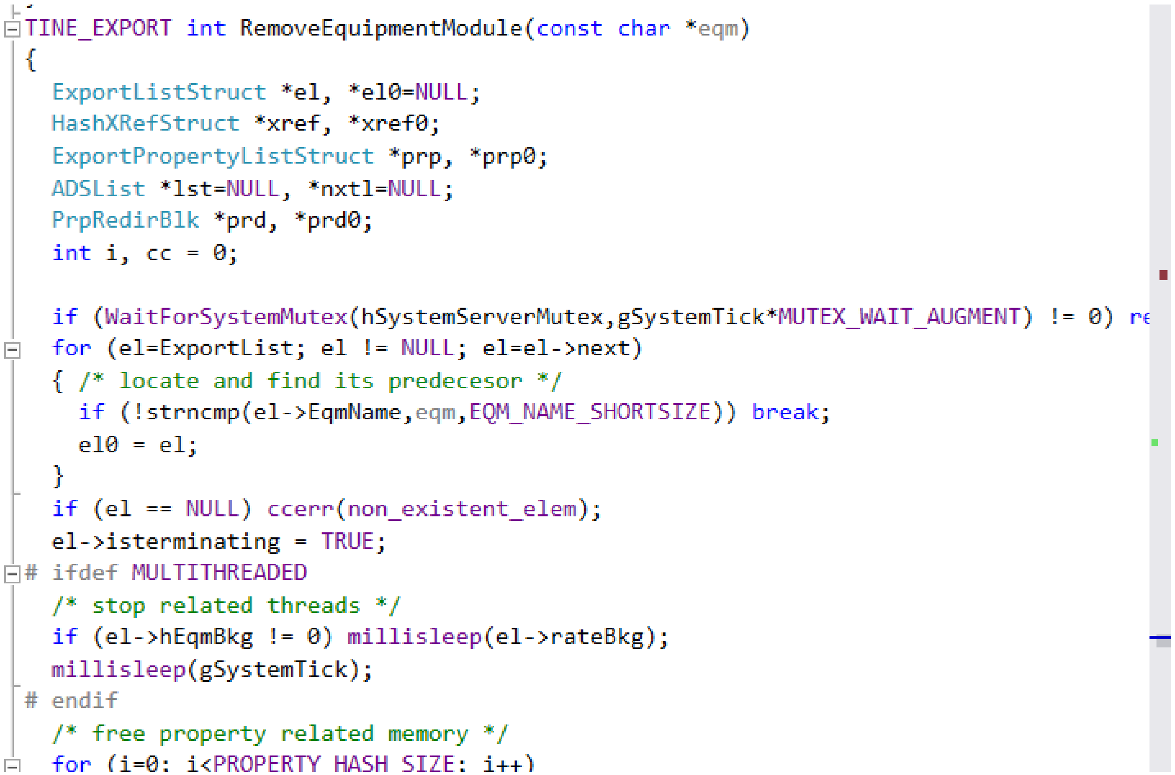
Try to ‘learn’ channel names from a doocs array :



```
#   ifndef VXWORKS
    if (chkMembers)
    { /* doocs server ? */
      DUSTRING *unames;
      PrpQueryStruct pqs[1], *ppqs = pqs;
      int np = 1, nd = 0, k;
      for (i=0; i<n; i++) if (!strncmp(dlist[i].name,"DEVGRP",6))
      {
        sprintf(tmpsrv,"%.64s/%.64s",srv,dlist[i].name);
        if (GetSystemPropertyInformation(tmpsrv,prp,&ppqs,&np) == 0)
        { /* this group device supports the target property */
          nd = pqs[0].prpSize;
          if ((unames=(DUSTRING *)alloca(nd * sizeof(DUSTRING))) == NULL) break;
          dout.dFormat = CF_USTRING;
          dout.dArrayLength = pqs[0].prpSize;
          dout.data.vptr = unames;
          sprintf(tmpprp,"%s.MEMBERS",prp);
          if (ExecLinkEx(tmpsrv,tmpprp,&dout,NULL,CA_READ|CA_MUTABLE|CA_NOCALLBACKS,QUERY_TO) != 0) break;
          n = dout.dArrayLength;
          for (k=0; k<n; k++) strncpy(dlist[k].name,unames[k].str,64);
          break;
        }
      }
      if (i == n)
      { /* did not find a device group!  try this: */
        sprintf(tmpsrv,"%.64s/*",srv);
        if ((unames=(DUSTRING *)alloca(n * sizeof(DUSTRING))) != NULL)
        {
          dout.dFormat = CF_USTRING;
          dout.dArrayLength = n;
          dout.data.vptr = unames;
          if (ExecLinkEx(tmpsrv,prp,&dout,NULL,CA_READ|CA_MUTABLE|CA_NOCALLBACKS,QUERY_TO) == 0)
          {
            n = dout.dArrayLength;
            for (k=0; k<n; k++) strncpy(dlist[k].name,unames[k].str,64);
          }
        }
      }
    }
#   endif
```

Feature Additions 4.6.2

- Extra mutex in RemoveEquipmenModule
  - Both FreeEQPModules() and freeServerMemory() call this !
    - Was a problem with buffered server attaching to a fec.xml database.



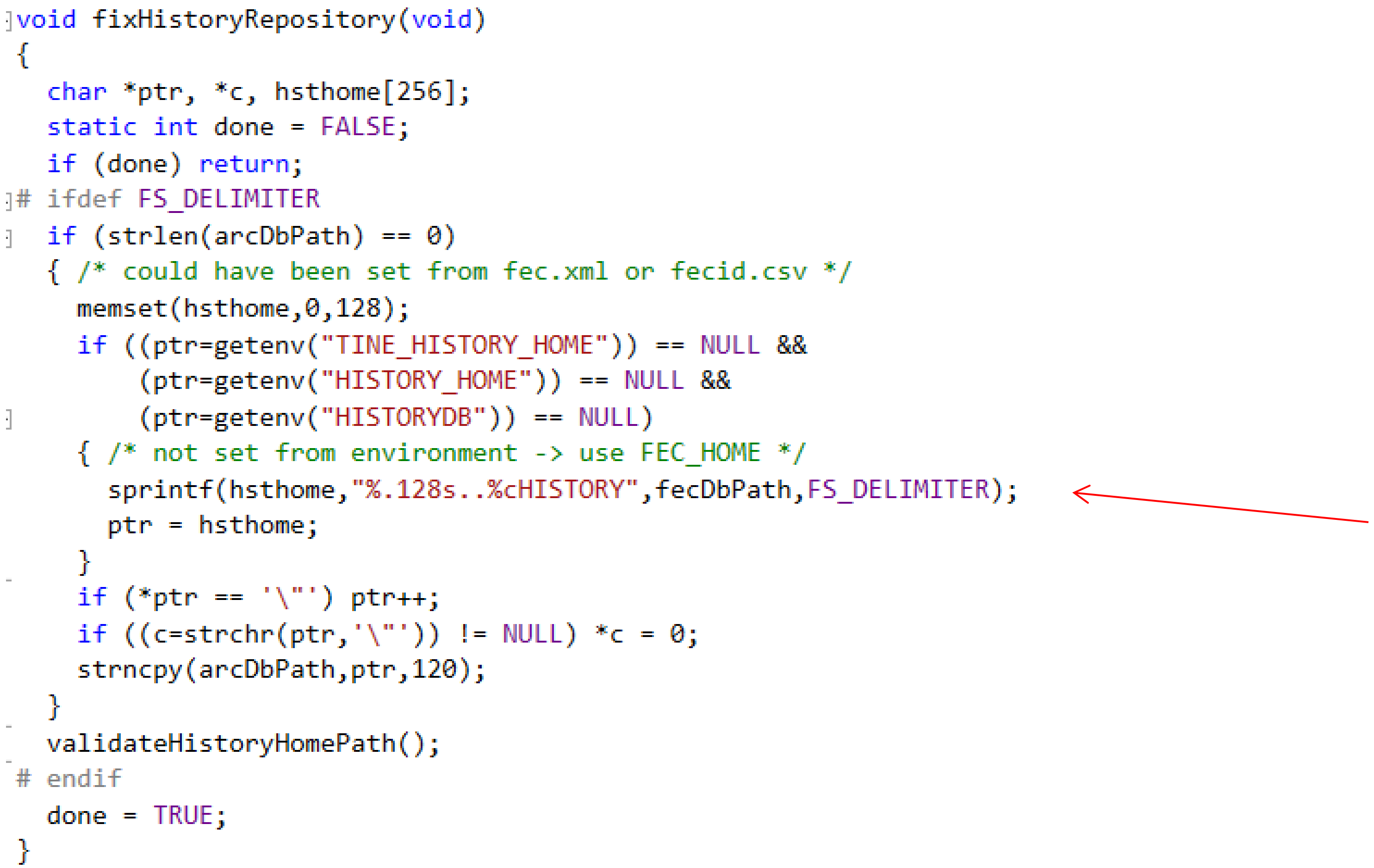
```
TINE_EXPORT int RemoveEquipmentModule(const char *eqm)
{
  ExportListStruct *el, *el0=NULL;
  HashXRefStruct *xref, *xref0;
  ExportPropertyListStruct *prp, *prp0;
  ADSList *lst=NULL, *nxtl=NULL;
  PrpRedirBlk *prd, *prd0;
  int i, cc = 0;

  if (WaitForSystemMutex(hSystemServerMutex,gSystemTick*MUTEX_WAIT_AUGMENT) != 0) re
  for (el=ExportList; el != NULL; el=el->next)
  { /* locate and find its predecesor */
    if (!strncmp(el->EqmName,eqm,EQM_NAME_SHORTSIZE)) break;
    el0 = el;
  }
  if (el == NULL) ccerr(non_existent_elem);
  el->isterminating = TRUE;
# ifdef MULTITHREADED
  /* stop related threads */
  if (el->hEqmBkg != 0) millisleep(el->rateBkg);
  millisleep(gSystemTick);
# endif
  /* free property related memory */
  for (i=0; i<PROPERTY_HASH_SIZE; i++)
```

Bug Fixes and Features 4.6.2

- Local History tweaks …
  - Add a transition point when ‘next’ data point is out of tolerance.
  - Trap a M. Lomperski crazy setting:
    - Long-term history depth = 1200 (because he always wants to keep it around)
    - 1200 months = 100 years -> file name naming system using the last two YY characters sees ’17’ which is ‘this year’.
    - => don’t allow local history > 20 years !

Bug Fixes and Features 4.6.2

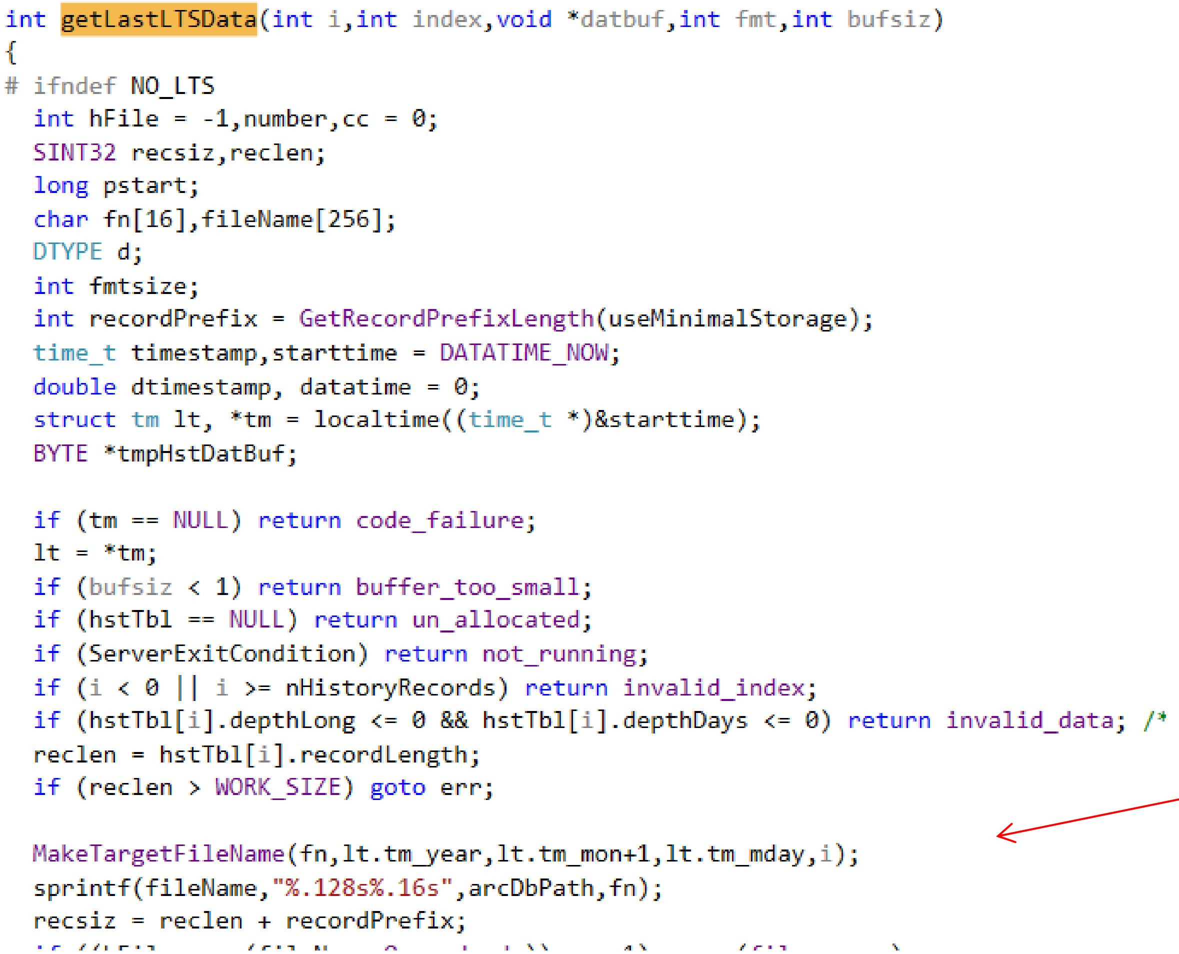
- Local History tweaks …



```
void fixHistoryRepository(void)
{
  char *ptr, *c, hsthome[256];
  static int done = FALSE;
  if (done) return;
# ifdef FS_DELIMITER
  if (strlen(arcDbPath) == 0)
  { /* could have been set from fec.xml or fecid.csv */
    memset(hsthome,0,128);
    if ((ptr=getenv("TINE_HISTORY_HOME")) == NULL &&
        (ptr=getenv("HISTORY_HOME")) == NULL &&
        (ptr=getenv("HISTORYDB")) == NULL)
    { /* not set from environment -> use FEC_HOME */
      sprintf(hsthome,"%.128s..%cHISTORY",fecDbPath,FS_DELIMITER);
      ptr = hsthome;
    }
    if (*ptr == '\"') ptr++;
    if ((c=strchr(ptr,'\"')) != NULL) *c = 0;
    strncpy(arcDbPath,ptr,120);
  }
  validateHistoryHomePath();
# endif
  done = TRUE;
}
```

Bug Fixes and Features 4.6.2

- Local History tweaks …

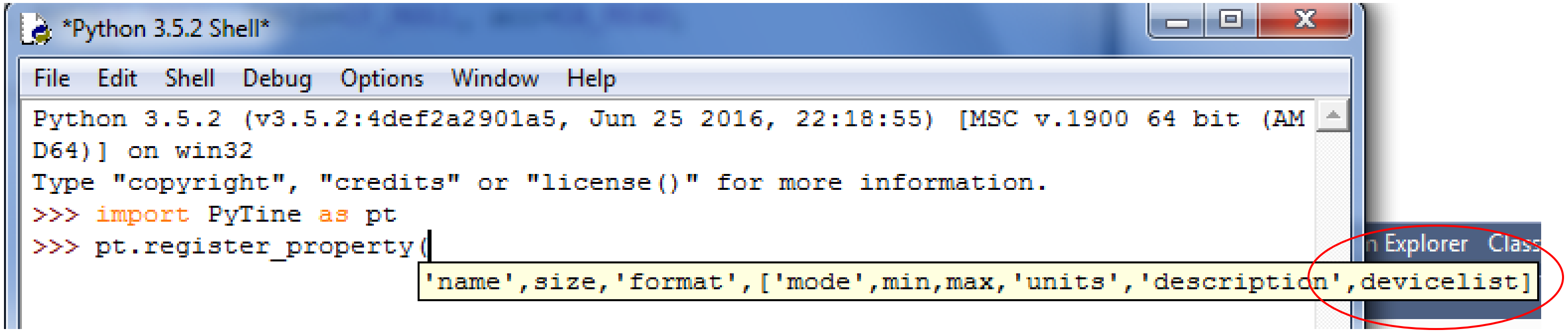


```
int getLastLTSData(int i,int index,void *datbuf,int fmt,int bufsiz)
{
# ifndef NO_LTS
  int hFile = -1,number,cc = 0;
  SINT32 recsiz,reclen;
  long pstart;
  char fn[16],fileName[256];
  DTYPE d;
  int fmtsize;
  int recordPrefix = GetRecordPrefixLength(useMinimalStorage);
  time_t timestamp,starttime = DATATIME_NOW;
  double dtimestamp, datatime = 0;
  struct tm lt, *tm = localtime((time_t *)&starttime);
  BYTE *tmpHstDatBuf;

  if (tm == NULL) return code_failure;
  lt = *tm;
  if (bufsiz < 1) return buffer_too_small;
  if (hstTbl == NULL) return un_allocated;
  if (ServerExitCondition) return not_running;
  if (i < 0 || i >= nHistoryRecords) return invalid_index;
  if (hstTbl[i].depthLong <= 0 && hstTbl[i].depthDays <= 0) return invalid_data; /*
  reclen = hstTbl[i].recordLength;
  if (reclen > WORK_SIZE) goto err;

  MakeTargetFileName(fn,lt.tm_year,lt.tm_mon+1,lt.tm_mday,i);
  sprintf(fileName,"%.128s%.16s",arcDbPath,fn);
  recsiz = reclen + recordPrefix;
```

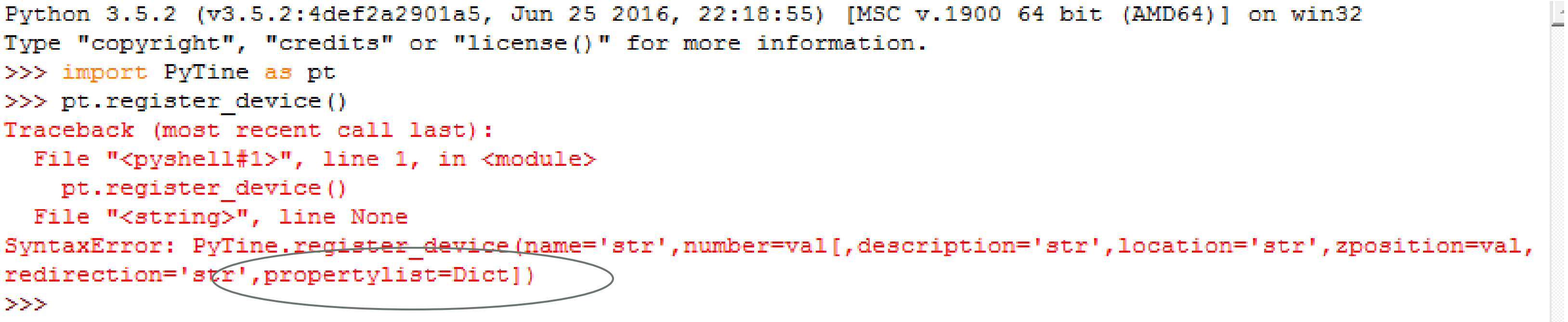
Filename refers to 'today'

Feature Additions 4.6.2

- Python (server API) …
  - add ‘lists’ to property and device registration …



```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import PyTine as pt
>>> pt.register_property(
                        'name',size,'format',['mode',min,max,'units','description',devicelist]
```



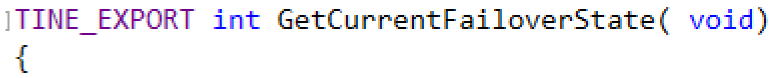
```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import PyTine as pt
>>> pt.register_device()
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    pt.register_device()
  File "<string>", line None
SyntaxError: PyTine.register_device(name='str',number=val[,description='str',location='str',zposition=val,
redirection='str',propertylist=Dict])
>>>
```

Feature Additions 4.6.2

- Python (3.5):
  - 32-bit windows PyTine.pyd (for KEK)

Feature Additions 4.6.2

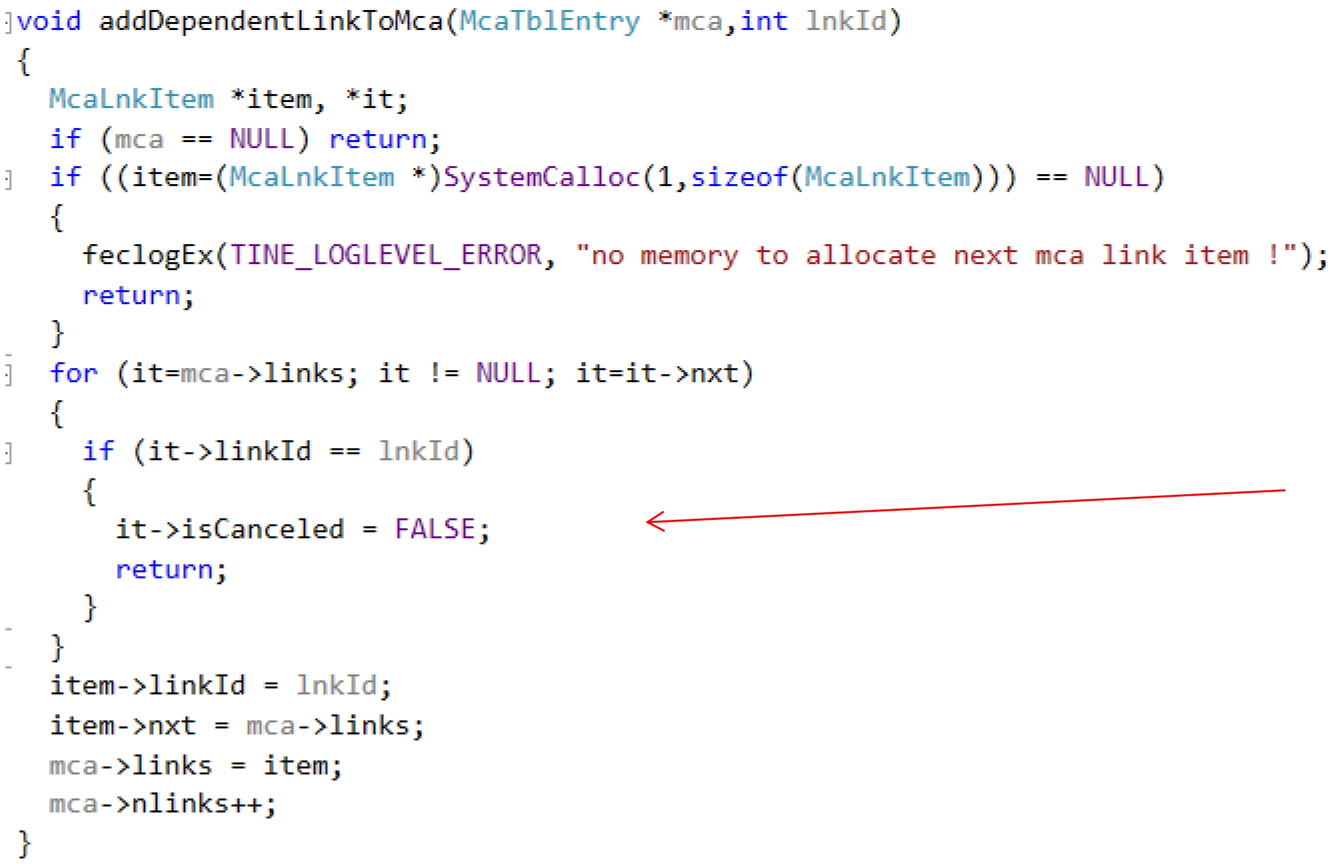
- Master/Slave
  - failover slaves now signal the master to re-export modules if it appears to have been simply unresponsive
    - New stock property ‘REEXPORT’
  - New API calls to determine if ‘I’m a slave who is now running as master’
    - New stock property “FAILOVERSTATE”
    - API :



```
]TINE_EXPORT int GetCurrentFailoverState( void)
{
```

Feature Additions 4.6.2

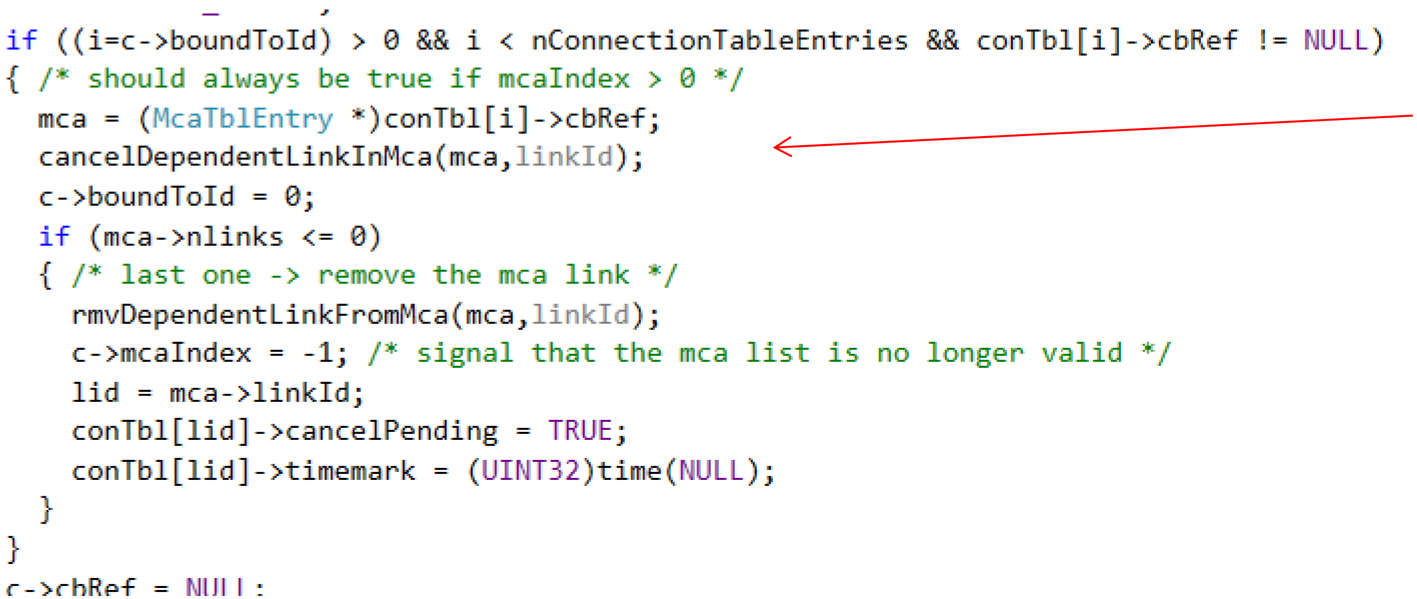
- Bug(s) of the month …
  - If we find it in the list during an ‘add’ make SURE it’s not canceled!



```
void addDependentLinkToMca(McaTblEntry *mca,int lnkId)
{
  McaLnkItem *item, *it;
  if (mca == NULL) return;
  if ((item=(McaLnkItem *)SystemCalloc(1,sizeof(McaLnkItem))) == NULL)
  {
    feclogEx(TINE_LOGLEVEL_ERROR, "no memory to allocate next mca link item !");
    return;
  }
  for (it=mca->links; it != NULL; it=it->nxt)
  {
    if (it->linkId == lnkId)
    {
      it->isCanceled = FALSE;
      return;
    }
  }
  item->linkId = lnkId;
  item->nxt = mca->links;
  mca->links = item;
  mca->nlinks++;
}
```

Feature Additions 4.6.2

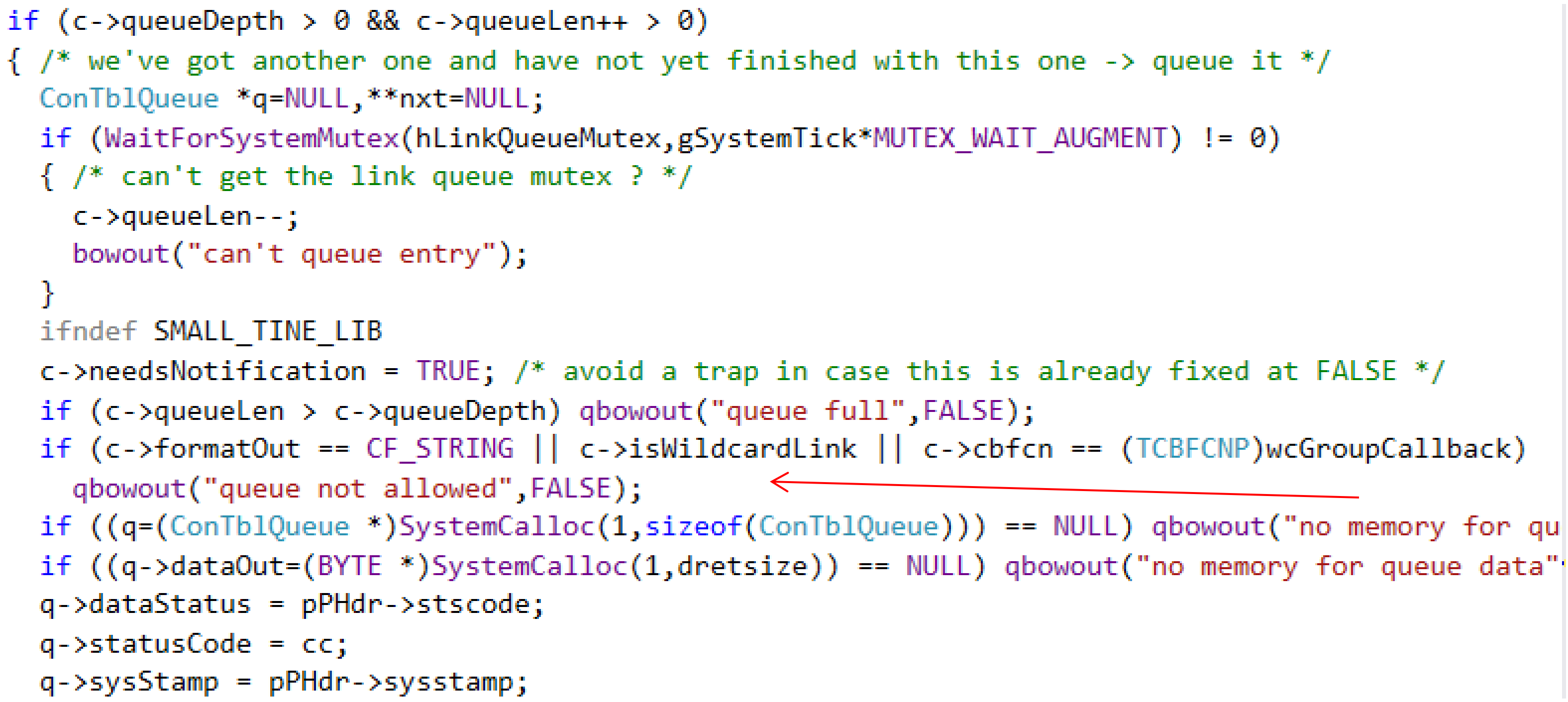
- Bug(s) of the month …
  - in _closeLink() …



```
if ((i=c->boundToId) > 0 && i < nConnectionTableEntries && conTbl[i]->cbRef != NULL)
{ /* should always be true if mcaIndex > 0 */
  mca = (McaTblEntry *)conTbl[i]->cbRef;
  cancelDependentLinkInMca(mca,linkId);
  c->boundToId = 0;
  if (mca->nlinks <= 0)
  { /* last one -> remove the mca link */
    rmvDependentLinkFromMca(mca,linkId);
    c->mcaIndex = -1; /* signal that the mca list is no longer valid */
    lid = mca->linkId;
    conTbl[lid]->cancelPending = TRUE;
    conTbl[lid]->timemark = (UINT32)time(NULL);
  }
}
c->cbRef = NULL;
```

Feature Additions 4.6.2

- Bug(s) of the month …
  - link queues : (don’t allow if we’re accessing a true string) …



```
if (c->queueDepth > 0 && c->queueLen++ > 0)
{ /* we've got another one and have not yet finished with this one -> queue it */
  ConTblQueue *q=NULL,**nxt=NULL;
  if (WaitForSystemMutex(hLinkQueueMutex,gSystemTick*MUTEX_WAIT_AUGMENT) != 0)
  { /* can't get the link queue mutex ? */
    c->queueLen--;
    bowout("can't queue entry");
  }
  ifndef SMALL_TINE_LIB
  c->needsNotification = TRUE; /* avoid a trap in case this is already fixed at FALSE */
  if (c->queueLen > c->queueDepth) qbowout("queue full",FALSE);
  if (c->formatOut == CF_STRING || c->isWildcardLink || c->cbfcn == (TCBFCNP)wcGroupCallback)
    qbowout("queue not allowed",FALSE);
  if ((q=(ConTblQueue *)SystemCalloc(1,sizeof(ConTblQueue))) == NULL) qbowout("no memory for qu
  if ((q->dataOut=(BYTE *)SystemCalloc(1,dretsize)) == NULL) qbowout("no memory for queue data"
  q->dataStatus = pPHdr->stscode;
  q->statusCode = cc;
  q->sysStamp = pPHdr->sysstamp;
```

Feature Additions 4.6.2

- Many new Acop.NET controls
- Fixes and features in tine.NET DLL.
- …

Still To Do …

- IPv6 testing
- Release 5.0 …