TINE Release 4.0 News

(Aug 5, 2011: That was the month that was !)

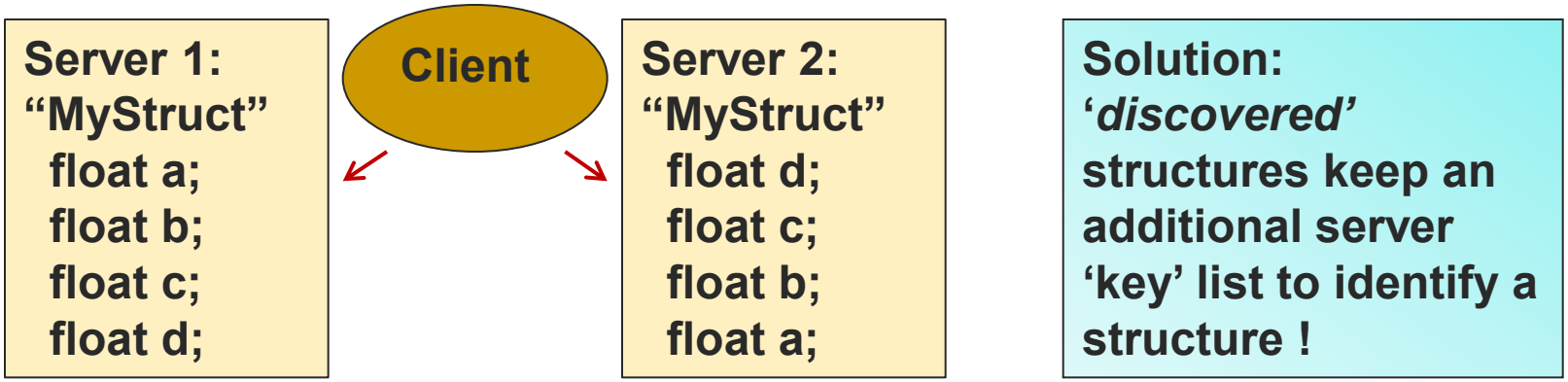
“What a long, strange trip it’s been ….”

Release 4.2.5

- Improvements in version 4.2.5
  - *Adding/Editing* history/alarm configurations *on-the-fly* !
  - New Data Type: *CF_KEYVALUE* !
  - New Console Command: *which* !
  - Potential tagged structure problem fixed.
- MatLab API addition: *tine_readimage()*
- .NET news: *C# documentation*

Release 4.2.5

- Bug Fixes (C-Lib)
  - Client Side :
    - Multiple identical links involved CF_IMAGE where not copying data correctly !
      - Situation can occur in MatLab !
    - Posible collisions using user-defined structures now checked!



Release 4.2.5

- Bug Fixes (java)
  - A notification problem with *dependent* Links during a server *timeout period* was fixed (thank you, Stefan May).
  - The *boolean* data type now behaves properly within ***ACOP*** transport ! (thank you, Kirsten).

Release 4.2.5

- Embellishments (C-Lib)
  - Console command ‘which’ now available
    - e.g. which ENS

```
>which ENS
>ENS#0 -> 131.169.120.41 @ port offset 0
>GENS -> 131.169.120.41 @ port offset 101
>
>

>which TTFMAG
>TTFMAG -> 131.169.147.189 offset 0
>
>

>which /TTF2/BLM
>/TTF2/BLM -> DCSEQM @ Fa83a9937e.67c
>Fa83a9937e.67c -> 131.169.147.126 @ port offset: 1660
>
>
```

Release 4.2.5

- Embellishments (C and java)
  - Stock Property "ADDHISTORY"
    - Adds/Edits selected Property to the local history table !
      - *Volatile*! New Record will vanish on server restart
      - Operators use the Archive Viewer to Add/Edit records and then inform those responsible.
  - Stock Properties "ALMDEFS" and "ALMWATCHTBL" now also take input!
    - Adds/Edits selected property to watched Alarm List
    - Edits selected Alarm definition (e.g. severity)
      - *Volatile*! Changes will vanish on server restart
      - Operators use the Alarm Viewer to Add/Edit records and inform those responsible.

Release 4.2.5

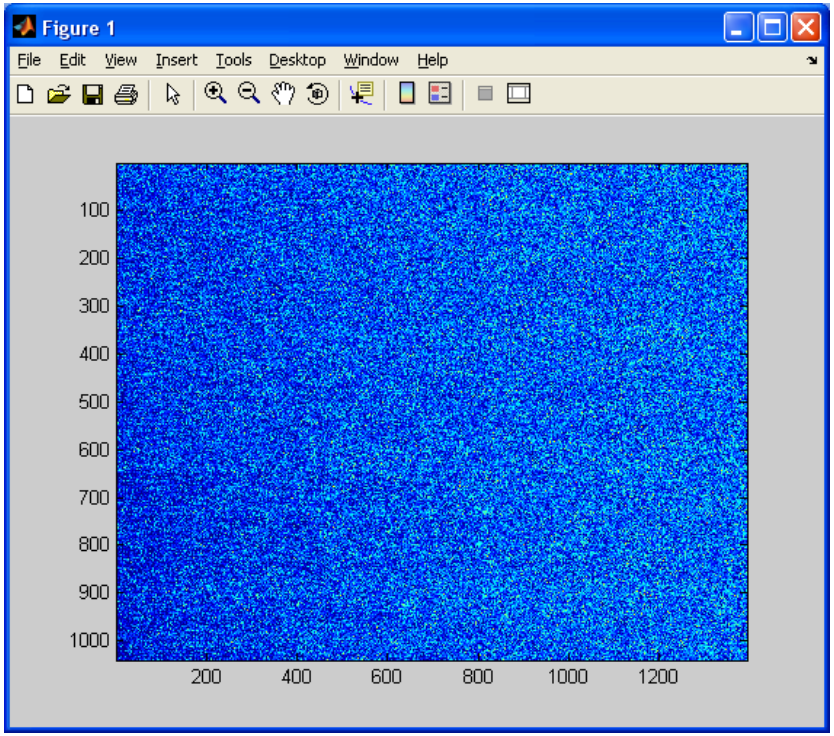
- New Format Type (from DOOCS)
  - CF_KEYVALUE
    - Behaves like a ‘free string’
    - Parse-able as “key: value”
    - Java: Class KEYVALUE
      - get/setKey(), etc.
    - What is this good for?

MatLab News

New Routine: tine_readimage()

```
??? tine_readimage usage:
 [a,h,s,t]=tine_readimage('/<context>/<server>/<device>[<property>]'[,polling_interval,'access_mode']);
'ret' contains the image, image header, status, timestamp
```

```
>> [a, hdr, sts, t] = tine_readimage('/PETRA/MDI2_RAWVIDEO1/Output[Frame.Sched]',1000,'NETWORK');
>> hdr

hdr =

                    baseTag: 861295446
                cameraPortId: 19410001
                  versionTag: 1
                 totalLength: 2895548
           timestampSeconds: '1312531657'
      timestampMicroseconds: '750000'
             cameraPortName: 'Jai Mono'
                sourceWidth: 1392
               sourceHeight: 1040
                   aoiWidth: -1
                  aoiHeight: -1
                     xStart: 0
                     yStart: 0
               bytesPerPixel: 2
      effectiveBitsPerPixel: 12
          horizontalBinning: 0
            verticalBinning: 0
               sourceFormat: 0
                imageFormat: 0
                frameNumber: 1199685
                eventNumber: 0
                     xScale: -1
                     yScale: -1
              imageRotation: 0
                    fspare1: -1
                    fspare2: -1
                    fspare3: -1
                 imageFlags: 3
                    ispare1: -1
                    ispare2: -1
                    ispare3: -1
         appendedFrameSize: 2895360
```



.NET News

Documentation for C# is there !

int tine::TLink::Attach (UInt32 mode,
TLinkCallback cb,
Int32 accessRate
) [inline]

Initiates an asynchronous link.

Asynchronous data exchange. Attach() returns immediately with a positive link index if the device name can be resolved and there are sufficient resources on the client side. Otherwise, a call to Attach() returns a negative completion code.

Parameters:

mode is the desired access mode, i.e. CM_TIMER, CM_DATACHANGE, CM_EVENT, etc. with possible modifiers (see Modes).

cb is the event callback routine which is fired when new data arrive or an error conditions arrises. This must have the prototype: void TLinkCallback(TLink lnk);

accessRate is the desired access polling interval in milliseconds. This interval is maintained at the server and is used to trigger the server's scheduler.

Returns:

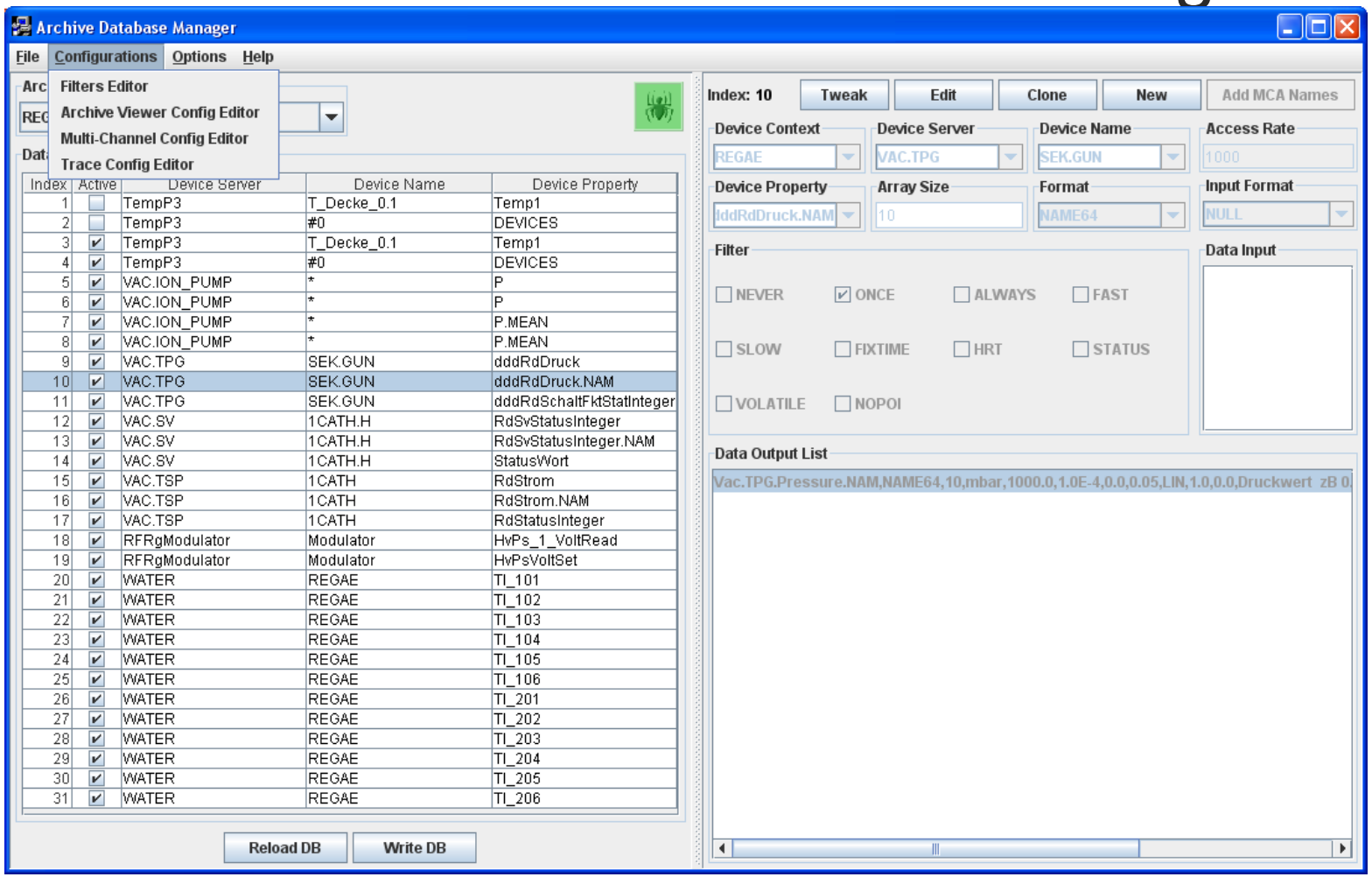
a positive link index or the negative of a TINE completion code.

Example:

```
public void lcb(tine.TLink lnk)
{
  if (lnk.GetStatus() != 0) Console.WriteLine("Link Status : " + lnk.GetStatus());
  if (bunchProfile.InvokeRequired)
    bunchProfile.Invoke(new MethodInvoker(updatePlot));
  else
    updatePlot();
}

public void startLink()
{
  cb = new TLink.TLinkCallback(lcb);
  tdt = new TDataType(currBuffer);
  tnull = new TDataType(IntPtr.Zero);

  currLink = new TLink("/PETRA/BunchScope/Bunch-1", "I.Bunch.SCH", tdt, tnull, tine.Access.CA_READ);

  int id = currLink.Attach(tine.Modes.CM_TIMER, cb, 1000);
}
```

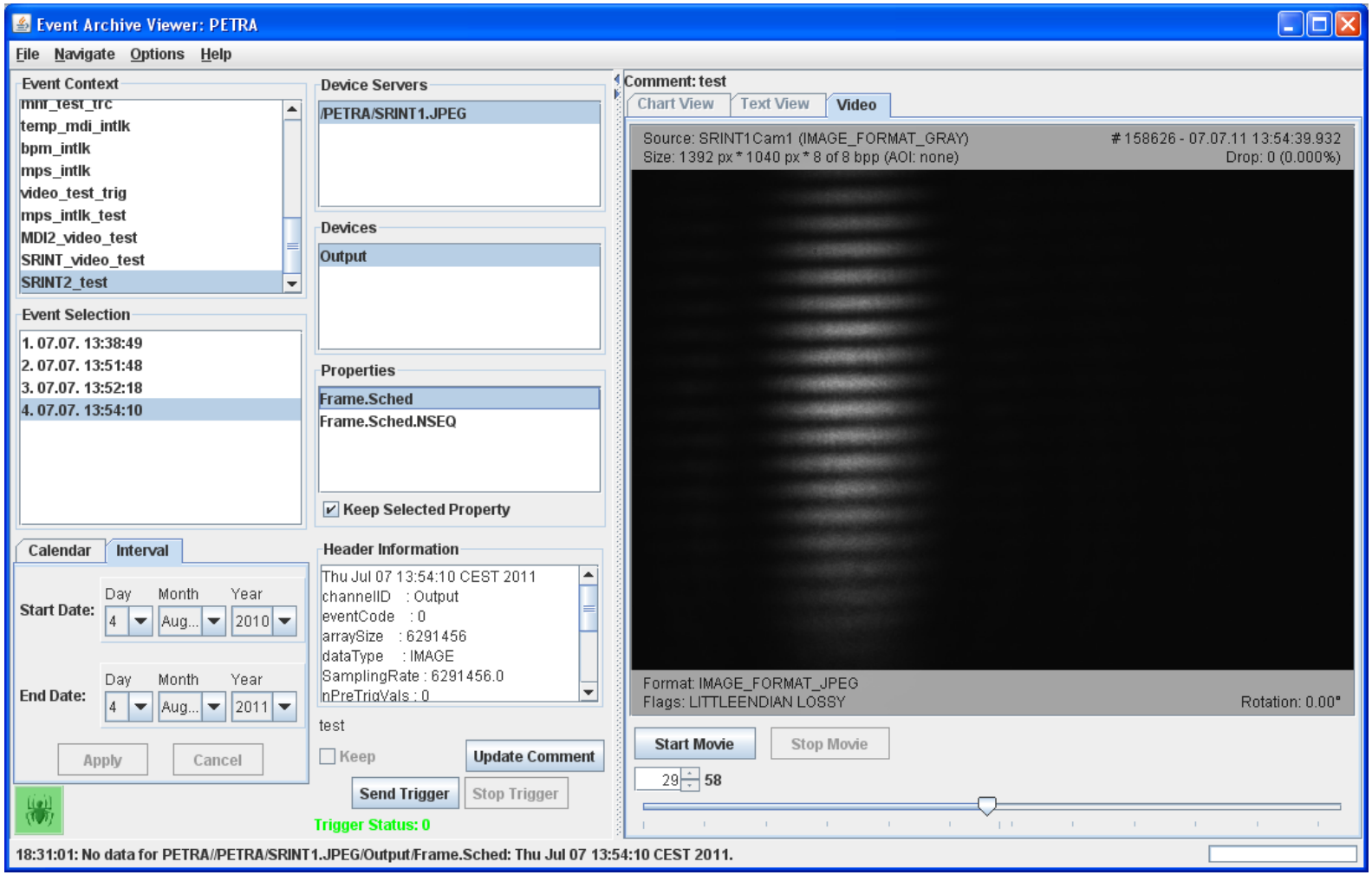
COSY News:

- Central Archive Database Manager



COSY News:

- Generic Event Archive Viewer



The Magic of Structures

- Tagged structs ultimately composed of primitives
  - byte, char (ansi), int16, int32, (int64), float, double
  - Can employ any TINE data type (*except* CF_HISTORY).
  - Deliver/Receive *atomic* data sets !
  - Field names describe the contents !

The Magic of Structures

- Compound Data Types
  - e.g. FLTINT, NAME64DBLDBL, …
  - Structure known a priori (good for SunRPC)
  - Unclear what fields contain
    - e.g. FLTINT a value-status pair or a value-timestamp pair or what?
    - e.g. FLTFLTFLTINTINTDBLNAME16
      - Yuck!
  - Like Tagged structures with generic tags "FLTINT", etc.
  - Widgets may or may not know what to do with a specific compound data type.

Tagged Structures

- C, C++:

```
typedef struct
{
  float a[3];
  long  b[2];
  short c[1];
  short reserved;
  char  d[32];
} Test1Struct;

#define Test1StructSize ((sizeof(float)*3) +\
                         (sizeof(long)*2)  +\
                         (sizeof(short)*1) +\
                         (sizeof(short)*1) +\
                         32)

/* maximum structure array length you're willing to manage: */
#define MAX_TEST1 10
#define quit(i) { printf("Register struct: out of memory\n"); exit(i); }
void registerStructs(void)
{
  /* this must follow the order of the structure explicitly! */
  if (AddFieldToStruct("TEST1",OFFSETIN(Test1Struct,a),3,CF_FLOAT,"a")) quit(1);
  if (AddFieldToStruct("TEST1",OFFSETIN(Test1Struct,b),2,CF_LONG,"b")) quit(1);
  if (AddFieldToStruct("TEST1",OFFSETIN(Test1Struct,c),1,CF_SHORT,"c")) quit(1);
  if (AddFieldToStruct("TEST1",OFFSETIN(Test1Struct,reserved),1,CF_SHORT,"reserved")) quit(1);
  if (AddFieldToStruct("TEST1",OFFSETIN(Test1Struct,d),32,CF_TEXT,"d")) quit(1);
  /* terminate the structure definition like this! */
  if (SealTaggedStruct("TEST1",sizeof(Test1Struct),MAX_TEST1)) quit(1);
}
```

Tag Name: Specified in API as "TEST1"

Tagged Structures

- Java:

```
class TEST1 extends TTaggedStructure
{
  float[] fval = new float[3];
  int[] ival = new int[2];
  short[] sval1 = new short[1];
  short[] sval2 = new short[1];
  char[] s = new char[32];
  TEST1()
  {
    super();
    addField(fval,"fval");
    addField(ival,"ival");
    addField(sval1,"sval1");
    addField(sval2,"sval2");
    addField(s,"text");
    initDone();
  }
}
//

// snippet ...

   TEST1 t1 = new TEST1();
```

Tag Name = Class Name: "TEST1"

Otherwise: use constructor which passes a tag name

Tagged Structures

- .NET (e.g. C#)

```
[StructLayout(LayoutKind.Sequential, Pack = 1, CharSet = CharSet.Ansi)]
public unsafe struct SineInfo
{
  public float amplitude;
  public float frequency;
  public float noise;
  public float phase;
  public int   numberCalls;
  [MarshalAs(UnmanagedType.ByValArray, SizeConst = 64)]
    public char[] description;
};

private const int NUM_DEVICES = 10;
private SineInfo[] sineInfoTable = new SineInfo[NUM_DEVICES];

// intermediate code omitted ...

TTaggedStruct tts = new TTaggedStruct(sineInfoTable);
```

Tag Name = struct Name: "SineInfo"

Otherwise: use constructor which passes a tag name

Tagged Structures

- MatLab:

Tag Name registered with API: “MlabInf”

```
global inf;
inf.amplitude = 100;
inf.frequency = 1;
inf.noise = 50;
inf.phase = 0;
inf.description = 'just another sine curve';
tine_register_type('MlabInf',inf);
```

Tagged Strutures