



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]

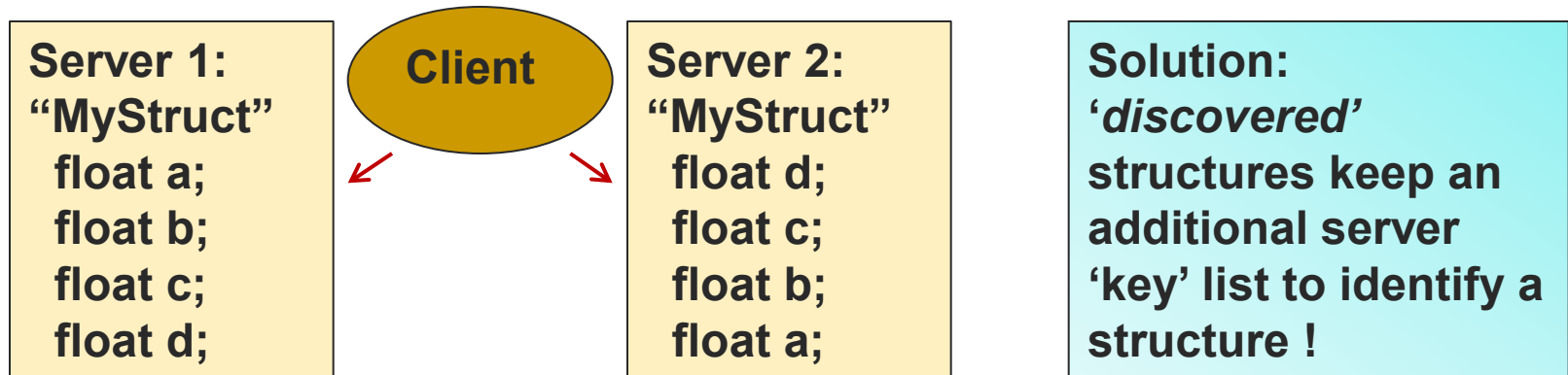
- **Improvements** in version 4.2.5
 - *Adding/Editing* history/alarm configurations *on-the-fly* !
 - New Data Type: *CF_KEYVALUE* !
 - New Console Command: *which* !
 - Potential tagged structure problem fixed.
- **MatLab API** addition: *tine_readimage()*
- **.NET** news: *C# documentation*

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 !
- Possible 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:

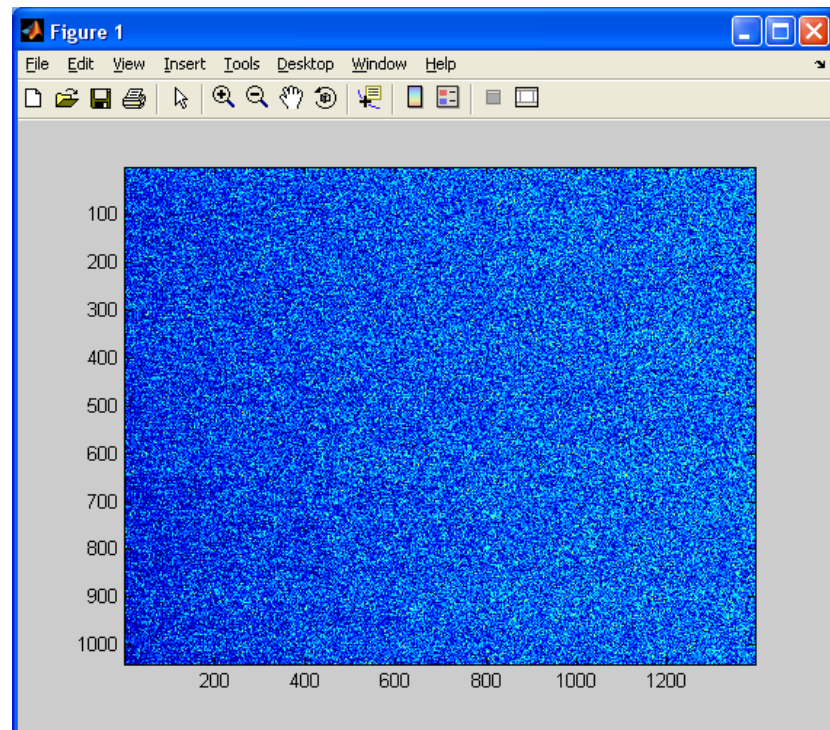
```
[a,h,s,t]=tine_readimage('/<context>/<server>/<device>[<property>']',[,polling_interval,'access_mode']);
```

'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  
isp spare1: -1  
isp spare2: -1  
isp spare3: -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 arises. 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 Archives Database Manager

Archive Database Manager

File Configurations Options Help

Arc Filters Editor
REC Archive Viewer Config Editor
Multi-Channel Config Editor
Dat Trace Config Editor

| Index | Active | Device Server | Device Name | Device Property |
|-------|-------------------------------------|---------------|-------------|---------------------------|
| 1 | <input type="checkbox"/> | TempP3 | T_Decke_0.1 | Temp1 |
| 2 | <input type="checkbox"/> | TempP3 | #0 | DEVICES |
| 3 | <input checked="" type="checkbox"/> | TempP3 | T_Decke_0.1 | Temp1 |
| 4 | <input checked="" type="checkbox"/> | TempP3 | #0 | DEVICES |
| 5 | <input checked="" type="checkbox"/> | VAC.ION_PUMP | * | P |
| 6 | <input checked="" type="checkbox"/> | VAC.ION_PUMP | * | P |
| 7 | <input checked="" type="checkbox"/> | VAC.ION_PUMP | * | P.MEAN |
| 8 | <input checked="" type="checkbox"/> | VAC.ION_PUMP | * | P.MEAN |
| 9 | <input checked="" type="checkbox"/> | VAC.TPG | SEK.GUN | dddRdDruck |
| 10 | <input checked="" type="checkbox"/> | VAC.TPG | SEK.GUN | dddRdDruck.NAM |
| 11 | <input checked="" type="checkbox"/> | VAC.TPG | SEK.GUN | dddRdSchaltFktStatInteger |
| 12 | <input checked="" type="checkbox"/> | VAC.SV | 1CATH.H | RdSvStatusInteger |
| 13 | <input checked="" type="checkbox"/> | VAC.SV | 1CATH.H | RdSvStatusInteger.NAM |
| 14 | <input checked="" type="checkbox"/> | VAC.SV | 1CATH.H | StatusWort |
| 15 | <input checked="" type="checkbox"/> | VAC.TSP | 1CATH | RdStrom |
| 16 | <input checked="" type="checkbox"/> | VAC.TSP | 1CATH | RdStrom.NAM |
| 17 | <input checked="" type="checkbox"/> | VAC.TSP | 1CATH | RdStatusInteger |
| 18 | <input checked="" type="checkbox"/> | RFRgModulator | Modulator | HvPs_1_VoltRead |
| 19 | <input checked="" type="checkbox"/> | RFRgModulator | Modulator | HvPsVoltSet |
| 20 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_101 |
| 21 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_102 |
| 22 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_103 |
| 23 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_104 |
| 24 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_105 |
| 25 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_106 |
| 26 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_201 |
| 27 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_202 |
| 28 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_203 |
| 29 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_204 |
| 30 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_205 |
| 31 | <input checked="" type="checkbox"/> | WATER | REGAE | TI_206 |

Index: 10 Tweak Edit Clone New Add MCA Names

Device Context: REGAE Device Server: VAC.TPG Device Name: SEK.GUN Access Rate: 1000

Device Property: dddRdDruck.NAM Array Size: 10 Format: NAME64 Input Format: NULL

Filter:
 NEVER ONCE ALWAYS FAST
 SLOW FIXTIME HRT STATUS
 VOLATILE NOPOI

Data Output List:
Vac.TPG.Pressure.NAM,NAME64,10,mbar,1000.0,1.0E-4,0.0,0.05,LIN,1.0,0.0,Druckwert zB 0

Reload DB Write DB

COSY News:

Generic Event Archive Viewer

Event Archive Viewer: PETRA

File Navigate Options Help

Event Context

- mmr_test_trc
- temp_mdi_intlk
- bpm_intlk
- mps_intlk
- video_test_trig
- mps_intlk_test
- MDI2_video_test
- SRINT_video_test
- SRINT2_test

Event Selection

- 07.07. 13:38:49
- 07.07. 13:51:48
- 07.07. 13:52:18
- 07.07. 13:54:10

Calendar Interval

Start Date: Day Month Year
4 Aug... 2010

End Date: Day Month Year
4 Aug... 2011

Apply Cancel

Device Servers

/PETRA/SRINT1.JPEG

Devices

Output

Properties

Frame.Sched
Frame.Sched.NSEQ

Keep Selected Property

Header Information

Thu Jul 07 13:54:10 CEST 2011
channelID : Output
eventCode : 0
arraySize : 6291456
dataType : IMAGE
SamplingRate : 6291456.0
nPreTrigVals : 0

test

Keep Update Comment

Send Trigger Stop Trigger

Trigger Status: 0

Comment: test

Chart View Text View Video

Source: SRINT1Cam1 (IMAGE_FORMAT_GRAY) # 158626 - 07.07.11 13:54:39.932
Size: 1392 px * 1040 px * 8 of 8 bpp (AOI: none) Drop: 0 (0.000%)

Format: IMAGE_FORMAT_JPEG
Flags: LITTLEENDIAN LOSSY Rotation: 0.00°

Start Movie Stop Movie

29 / 58

18:31:01: No data for PETRA/PETRA/SRINT1.JPEG/Output/Frame.Sched: Thu Jul 07 13:54:10 CEST 2011.

[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 ITaggedStructure
{
    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 ...

ITaggedStruct tts = new ITaggedStruct(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 Structures]

The screenshot displays the Java Instant Client application window. The title bar reads "Java Instant Client". The menu bar includes "File", "Options", "Data Access", "Monitor Options", "Debug Options", and "Help".

The interface is divided into several sections:

- Device Context:** A dropdown menu showing "DESY2".
- Device Subsystem:** A dropdown menu showing "ALL".
- Device Server:** A dropdown menu showing "Bunche.DE05".
- Device Name:** A dropdown menu showing "IMA-DE05".
- Data Size:** A text input field containing "1".
- Data Type:** A dropdown menu showing "STRUCT".
- Trace Header:** A text field containing "[TraceHS] Digitizer Trace Header".
- Show Stock Properties:** An unchecked checkbox.
- Device Property:** A dropdown menu showing "Trace.INFO.". A list of properties is visible below it:
 - Trace.INFO.preTrigger
 - Trace.INFO.ScaleX
 - Trace.INFO.OffsetX
 - Trace.INFO.UnitsX
 - Trace.INFO.PlotMaxY
 - Trace.INFO.PlotMinY
 - Trace.INFO.UnitsY
 - Trace.INFO.Reserved
- Autoscale:** An unchecked checkbox.
- Log Scale:** An unchecked checkbox.
- History:** An unchecked checkbox.
- Suggest Decorations:** A checked checkbox.
- Input Pane:** An unchecked checkbox.

The main display area shows a trace header: **.DESY2/Bunche.DE05/IMA-DE05 Trace.INFO @ 11:16:50.865**. Below the header is a list of tagged structure elements:

```
(0,0) [DeviceName] -> IMA-DE05
(0,1) [DeviceDesc] -> Induktiver Monitor DE05 in DESY
(0,2) [DataFormat] -> 517
(0,3) [ArraySize] -> 1000
(0,4) [preTrigger] -> 0
(0,5) [ScaleX] -> 1.0E-9
(0,6) [OffsetX] -> 0.0
(0,7) [UnitsX] -> sec
(0,8) [PlotMaxY] -> 0.1
(0,9) [PlotMinY] -> -0.5
(0,10) [UnitsY] -> Volts
```