

# The TINE Control System

Overview + Notes on Release 4.0



# First, some general observations

- A Control System is more than
  - clicking a button “here” and making something happen “there”.
  - taking data “there” and displaying it “here”.
- Every aspect of the machine/facility under control should be part of the control system.
  - What do you mean “off-line” analysis? (Why isn’t it on-line?)
  - If the “going gets tough”, does the control system quit?
- The Control System can have different flavors
  - Different platforms, io busses, etc.
  - Different “control systems”
    - If TINE and EPICS are used to control a machine, then the control system is “TINE and EPICS” not TINE or EPICS.
- The Presentation layer shall not be ignored!

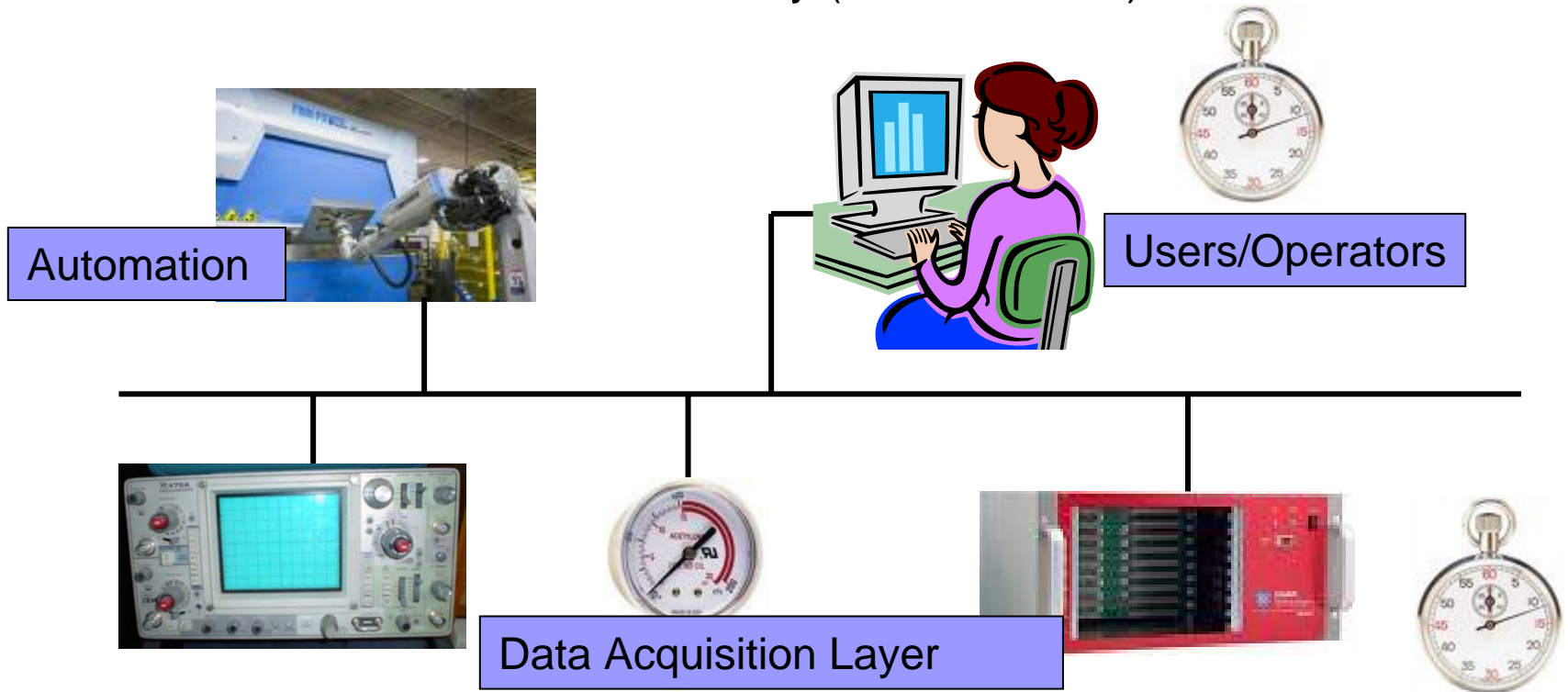
# Control Systems

(one way or another) have to deal with ...

- **Distributed** end points and processes
- **Data Acquisition** (front end hardware)
- **Real-time** needs (where necessary)
- **Process control** (automation, feedback)
- **Central Services** (Archive, Alarm, Name Resolution, ...)
- **Security** (who's allowed to do what from where?)
- **States** (Finite State Machines, sequencing, automation...)
- **Time synchronization** (time stamps, cycle ids, etc.)
- **Databases** (configuration, machine data, post-mortem data, ...)
- **Statistics** (control system itself, operation, ...)
- **Logging** (central, local, application, ...)
- **Data transport** (data flow, *control system protocol*, scalability)

# Control Systems Protocols ...

- How to move information efficiently (does it scale?) ....



current Data timestamp: Sat Feb 3 16:50:45 2007 666 msec (UTC: 1170517845.666)

# TINE Bells and Whistles ...

The image displays a collection of software windows used for system monitoring and control:

- Archive Viewer:** Shows a line graph of system data over time.
- FEC Remote Control:** A table listing various subsystems and their status.
 

Subsystem	Status	Subsystem	Status	Subsystem	Status
ABSCALC.7	HEMARCH	HESTATSRV	MHF		
ADDA	HEMARCHX	HETBUNCH	MPYI		
CVTLPOL	HEMON.5	HETCOOL.5	MPYI		
DIAGHIST.4	HEMON.6	HETIME.5	MPYI		
DIAGHIST.4.100	HEMON.7	HETMON.5	MPYI		
ELPRSTATSRV	HEMON.8	HETMON.6	MPYI		
ETSrvr	HEMON.9	HETMON.7	OSZD		
ETSrvr RPT	HENEG.10	HETMON.8	PHF		
FB-Q-HET	HENEG.11	HETRCRFFB.1	QUAT		
FBUNLUX1	HENEG.5	HETRCRPN.1	RELI		
H1CJCPROXY	HENEG.6	HETRCRFO.1	RMS		
H1USERVFR	HENEG.7	HETRCRFS.1	VWE		
HIERU.S	HEORBFK.9	HETRCRFLC.1	VWE		
HIEBRU.S	HEBPMP	HETRCRFLW.1	VWN		
HIECASR.S	HEPBUNCH	HETRCRFRW.1	VWN		
HIECOOL.5	HEPCOOL.5	HETSFN	VWS		
HIECOOL.6	HEPHAKO.5	HETSP0	VWS		
HIECOOL.7	HEPHTRC	HETSPREAM	VWW		
HIECOOL.8	HEPIC1	HETSFS	VWW		
HIECOOL.9	HEPKOLLI	HETSFWI	VWW		
HIEDIAL.OG	HEPMARCH	HEVACN.10	ZLUS		
HIEBLM	HEPOSHI	HEVACN.11	ZME		
HIEBUNCH	HEPOSIZEUS	HEVACN.6	ZSPS		
HEHFHF	HEPTRANSF	HEVACN.7	bwis		
HIECIDC.1	HEPTRANSFVW	HEVACS.10	bwis		
HIEPALLIS	HEPACPROXY	HEVACFS.11	bwis		
HIEHMFS	HEPSPROXY	HEVACS.8	Imbis		
HIEHPSA	HEPSPROXY	HEVACS.9	Imbis		
- DORIS Overview Operation History:** Includes a pie chart showing system status distribution.
- Transient Recorders:** Shows a graph of transient events.
- Device Context:** A control panel for the HERA device with various subsystems checked.
  - Selected Subsystems: GPS, IM, SER, XXX, HERA, INJ, RF, HST, SUB, RPI, EXP, TUNE, MAG, PVAK, TEST, MEX, VAC, CRVO.
- DORIS Alarm Viewer:** Displays a table of active alarms.
 

System	Alarms	System	Alarms	System	Alarms
Test	0	Trigger Mod	0	HF	2
Wiggler	0	Magnet	0	SeK1	0
Bunche	0	Neb Bunche	0	HCon	0
VCor	0	Lage Regl	1	Vac	0
Peak-Strip	0	AM-Gen	0	Timing	0
FotoMon	0	Schime	0	Profile	0
Tim. Mon	0	Radio	0	Zyklus Gen	0
I-Hist	0	Per Interlock	0	BL Interlock	1
DatenTV	0	Orbit	0	WasserHF	1
HF-Dump	4	Tune Cntl	0	Feedback	0
Temp	0	Scope	0	IDC	0
Scrapet	0	System	1	Hardware	2
- Overview of HERA Operations:** A graph showing operational parameters over time.

We'll cover these as we go along!

# TINE\* : multi-platform

- DOS
- Win16, Win32 (9x, NT, 2K, XP, ...), Win64 ?
- Unix (Solaris, HP, OSF, SGI, Ultrix, ...)
- Linux, FreeBSD (32 bit, 64 bit)
- ELINOS
- MAC OS X
- VxWorks
- VMS (Vax, Alpha)
- LynxOS
- NIOS (plugs, single-threaded LWIP, ...)
- Java

Three-fold Integrated Networking Environment (this is not epics)

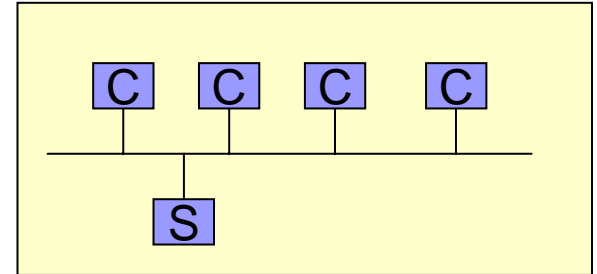
# TINE: multi-protocol

- IPX (dead and no one has noticed?)
- **UDP** (most common)
- TCP/IP (upon request)
- Pipes (client-server on same UNIX machine)
- Windows Messages (client-server on same Windows machine)

# TINE: multi-architecture

## ■ Client-Server (classic)

- Transaction based
- Synchronous data access only
- The “**N-Client**” Problem ? (do threads help?)
- ExecLink(“/HERA/BPM/WL167MX”, “ORBIT.X”, ...)



## ■ Publisher-Subscriber (nearly classic)

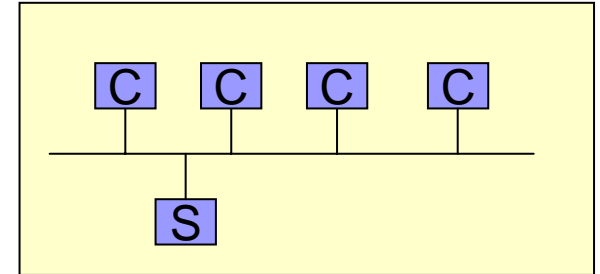
- **Connection Tables !**
- Synchronous/Asynchronous data access
- The “**10N-Client**” Problem ?
- AttachLink(..., CM\_REFRESH, 1000, linkCb)
- Callback events !



# TINE: multi-architecture

## ■ Producer-Consumer

- Asynchronous data messages (Multicast)
- The “**N-Producer**” Problem ?
- `recvNetGlobal(“HPMAGEN”) or`
- `AttachLink(“HPMAGEN”,...,CM_RECEIVE, 1000, linkCb)`



## ■ Publisher-Consumer (Producer-Subscriber?)

- Like Publisher-Subscriber but:
- Multicast group is a single connection Table entry
- **N = 1 !!!**
- `AttachLink(...,CM_REFRESH|CM_NETWORK,1000, linkCb)`

# TINE Core:

- Kernel written in C (straight-up)
  - Just like your operating system!
  - OO APIs are on top of this
- Network transport based on sockets
  - So are CORBA, SunRPC, ChannelAccess
- Java kernel written in java
  - No JNI (except CDI interface)

# TINE Servers

(Publishers, Producers)

- A Front End Controller (**FEC**) :
  - Has one or more **Equipment Modules** (device servers)
    - *Handles* requests for data and commands
    - Has a *well-known identity*
    - Has a unique *export name* for each running instance
    - Is *always running*
    - Can be a member of a device *Group*

# TINE Clients

(Subscribers, Consumers)

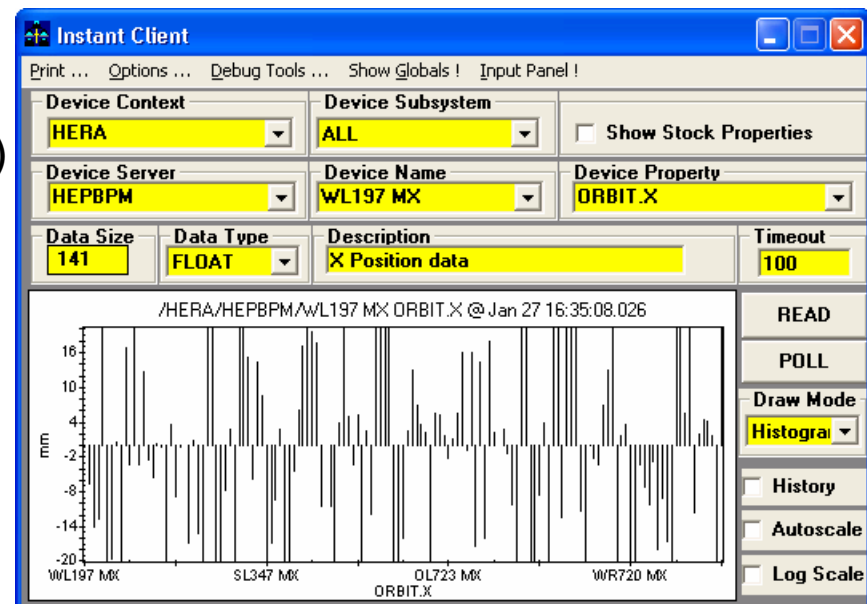
- TINE Client Process:

- *anonymous* -> i.e. Not known to the system
- *Not unique*
- Can *come and go*

# TINE Naming Convention

- Hierarchical
- Device is specified by :
  - **Device Context** (*Facility* in DOOCS, *Domain* in TANGO)
  - **Device Server** (or **Group**) (*Device* in DOOCS, *Family* in TANGO)
  - **Device Name** (*Location* in DOOCS, *Member* in TANGO)
- Data endpoint is specified by:
  - **Property** (*Property* in DOOCS, *Attribute/Command* in TANGO)
    - Are really 'methods' or 'calls'

Note: **Device Subsystem** is not part of the name space, but is a browseable element!



# TINE ENS: Plug and Play



I want to be known to the system as “BPM.P” in the context “HERA”



Does BPM.P already exist for HERA?

-> **Yes** : Is same address as already registered?

-> **Yes: Accept** and increment reboot count

-> **No**:

Is the currently listed BPM.P for HERA running ?

**Yes : Refuse** and send “in-use” message

**No : Accept** and update database

-> **No : Accept** and update database

Equipment Name Server (ENS)

- Forward accepted requests to secondary name servers



# Address Redirection

The image shows two instances of the 'Instant Client' software. The left instance shows a configuration window with the following settings:

- Device Context: TTF
- Device Subsystem: ALL
- Device Server: QUAD
- Device Name: Q9ACC3
- Device Property: BITS2AMPS
- Data Size: 1
- Data Type: FLOAT
- Timeout: 100
- Draw Mode: Text dun
- Autoscale: unchecked
- Log Scale: unchecked

The right instance shows the same configuration window with the following settings:

- Device Context: TTF
- Device Subsystem: ALL
- Device Server: QUAD
- Device Name: Q9ACC3
- Device Property: BITS2AMPS
- Data Size: 1
- Data Type: FLOAT
- Timeout: 100
- Draw Mode: Text dun
- Autoscale: unchecked
- Log Scale: unchecked

The main window displays the following text:

```
/TTF/QUAD/Q9ACC3 BITS2AMPS @ Jan 27 12:28:54.013  
( 0) 3.662221E-03
```

A dialog box titled 'Front End Information for : TTF/QUAD' is open, displaying the following information:

```
Device Server : QUAD -> TTMAG1-19  
Local equipment module name : QUAD -> TCPDAT  
Front End Computer (FEC) name: TTFMAG1 -> TTMAG1-19  
Property : BITS2AMPS -> BITS2AMPS  
IP Address: 131.169.147.189  
IPX Address: 000000000000[00000000]  
Port Offset: 0  
Host Computer: accsottfmg2.desy.de
```

An 'OK' button is visible at the bottom of the dialog box.

Some registered devices  
are not handled directly by  
FEC TTFMAG1 !

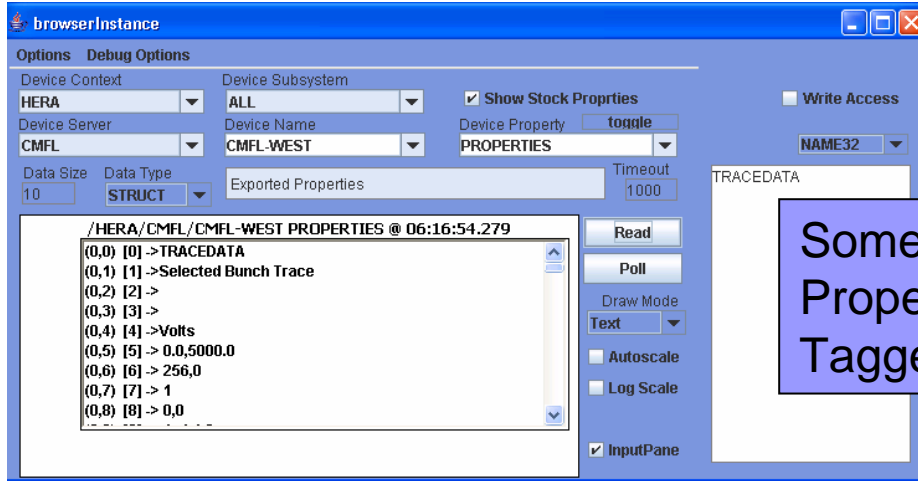


# TINE Data Types

- Primitives (byte, short, int\*, long\*, float, double, char\*)
- Fixed-length Strings (“NAME8”, “NAME16”, “NAME32”, ...)
- Doublets (FLTINT, “INTINT”, “DBLDBL”, “NAME32INT”, ...)
- Triplets (“FLTINTINT”, “NAME16FLTINT”, ...)
- Quadruplets (“FLTINTINTINT”, “INTINTINTINT”, ...)
- Specials (“USTRING”, “UNAME”, SPECTRUM, VIDEO...)
  
- **User-defined “Tagged Structures” !!!**
  - Structure registered at both ends (client, server)
    - Client can ‘discover’ structure fields !

\*Platform dependent

# TINE Tagged Structures

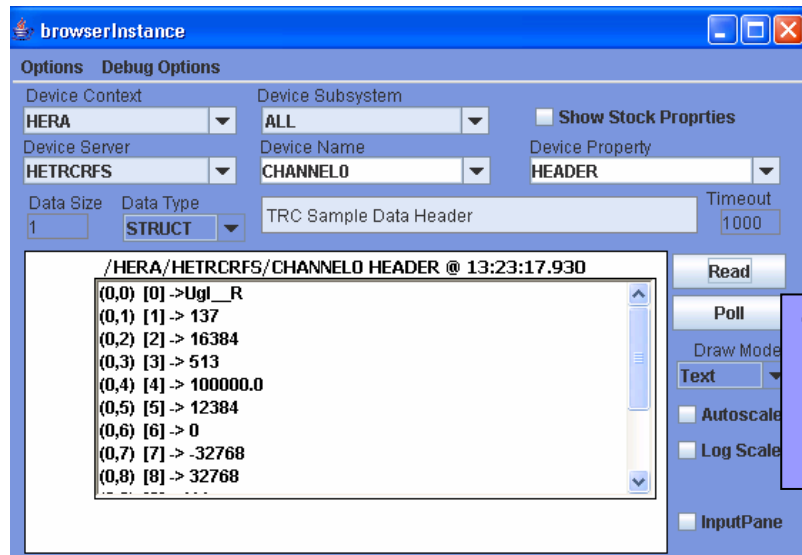


The screenshot shows the 'browserInstance' application window. The 'Options' tab is active, displaying configuration for 'HERA' device context, 'CMFL' device server, and 'CMFL-WEST' device name. The 'Device Property' is set to 'PROPERTIES'. The 'Data Type' is 'STRUCT' and 'Data Size' is '10'. The 'Exported Properties' field is empty. The main display area shows a list of properties for '/HERA/CMFL/CMFL-WEST PROPERTIES @ 06:16:54.279':

```
(0,0) [0] ->TRACEDATA
(0,1) [1] ->Selected Bunch Trace
(0,2) [2] ->
(0,3) [3] ->
(0,4) [4] ->Volts
(0,5) [5] -> 0.0,5000.0
(0,6) [6] -> 256,0
(0,7) [7] -> 1
(0,8) [8] -> 0,0
```

Buttons for 'Read', 'Poll', 'Draw Mode' (set to 'Text'), 'Autoscale', 'Log Scale', and 'InputPane' are visible on the right.

Some Stock Properties use Tagged Structures.



The screenshot shows the 'browserInstance' application window. The 'Options' tab is active, displaying configuration for 'HERA' device context, 'HETRCRFS' device server, and 'CHANNELO' device name. The 'Device Property' is set to 'HEADER'. The 'Data Type' is 'STRUCT' and 'Data Size' is '1'. The 'Exported Properties' field is 'TRC Sample Data Header'. The main display area shows a list of properties for '/HERA/HETRCRFS/CHANNELO HEADER @ 13:23:17.930':

```
(0,0) [0] ->Ugl_R
(0,1) [1] -> 137
(0,2) [2] -> 16384
(0,3) [3] -> 513
(0,4) [4] -> 100000.0
(0,5) [5] -> 12384
(0,6) [6] -> 0
(0,7) [7] -> -32768
(0,8) [8] -> 32768
```

Buttons for 'Read', 'Poll', 'Draw Mode' (set to 'Text'), 'Autoscale', 'Log Scale', and 'InputPane' are visible on the right.

Simple Object that ships 'together' not in pieces!

Transient Recorder Servers use Tagged structures for headers.

# TINE Tagged Structures

```
{  
/* this must follow the order of the structure explicitly! */  
if (addFieldToStruct("SineInfo", OFFSETIN(SineInfo, amplitude), 1, CF_FLOAT, "amplitude")) quit(1);  
if (addFieldToStruct("SineInfo", OFFSETIN(SineInfo, frequency), 1, CF_FLOAT, "frequency")) quit(1);  
if (addFieldToStruct("SineInfo", OFFSETIN(SineInfo, noise), 1, CF_FLOAT, "noise")) quit(1);  
if (addFieldToStruct("SineInfo", OFFSETIN(SineInfo, phase), 1, CF_FLOAT, "phase")) quit(1);  
if (addFieldToStruct("SineInfo", OFFSETIN(SineInfo, numberCalls), 1, CF_LONG, "numberCalls")) quit(1);  
if (addFieldToStruct("SineInfo", OFFSETIN(SineInfo, description), 64, CF_TEXT, "description")) quit(1);  
/* terminate the structure definition like this! */  
if (sealTaggedStruct("SineInfo", sizeof(SineInfo), NUM_DEVICES)) quit(1);  
}
```

Registered at Server

The screenshot shows the 'Instant Client' window with the following configuration:

Device Context: TEST	Device Subsystem: ALL	<input checked="" type="checkbox"/> Show Stock Properties	
Device Server: WinSineServer	Device Name: SineGen0	Device Property: STRUCTFORMAT	
Data Size: 512	Data Type: NAME16II	Description: Registered structure information	Timeout: 1000

The main display area shows the following structure definition:

```
/TEST/WinSineServer/SineGen0 STRUCTFORMAT @ Sep 24 22:49:45.000  
( 0) amplitude 1 517  
( 1) frequency 1 517  
( 2) noise 1 517  
( 3) phase 1 517  
( 4) numberCalls 1 515  
( 5) description 64 516  
( 6) SineInfo 84 767
```

Discovered by Client

The screenshot shows the 'Instant Client' window with the following configuration:

Device Context: TEST	Device Subsystem: ALL	<input type="checkbox"/> Show Stock Properties	
Device Server: WinSineServer	Device Name: SineGen0	Device Property: SineInfo	
Data Size: 10	Data Type: STRUCT	Description: Sine Curve Information Set	Timeout: 1000

The main display area shows the following data:

```
/TEST/WinSineServer/SineGen0 SineInfo @ Sep 24 23:06:51.217  
[ 0 -> 0] 256  
[ 0 -> 1] 1  
[ 0 -> 2] 30  
[ 0 -> 3] 0  
[ 0 -> 4] 1026  
[ 0 -> 5] Sine Generator 0 at your disposal  
[ 1 -> 0] 256  
[ 1 -> 1] 1  
[ 1 -> 2] 30  
[ 1 -> 3] 0
```

# TINE Array Types

- **AT\_UNKNOWN** - type not given
- **AT\_SCALAR** - just a number
- **AT\_SINGLE** - a single column array
- **AT\_DOUBLE** - a double array
- **AT\_CHANNEL** - attribute
  - - array elements refer to device channels
  - - display as histogram (only y-axis units)
- **AT\_SPECTRUM (AT\_TRACE, AT\_WAVEFORM)** - attribute
  - - array elements define a trace
  - - display as line trace (y-axis and x-axis units)
- **AT\_STRUCTURED (AT\_COLLECTION)** - attribute
  - - array elements define a collection of things with the same data type
  - - display type unknown
- **AT\_IMAGE** - a byte blob defining an image



# TINE: Getting started ...

- How do I make a TINE Server?
- How do I connect to my hardware?
- How do I access the data from my server?

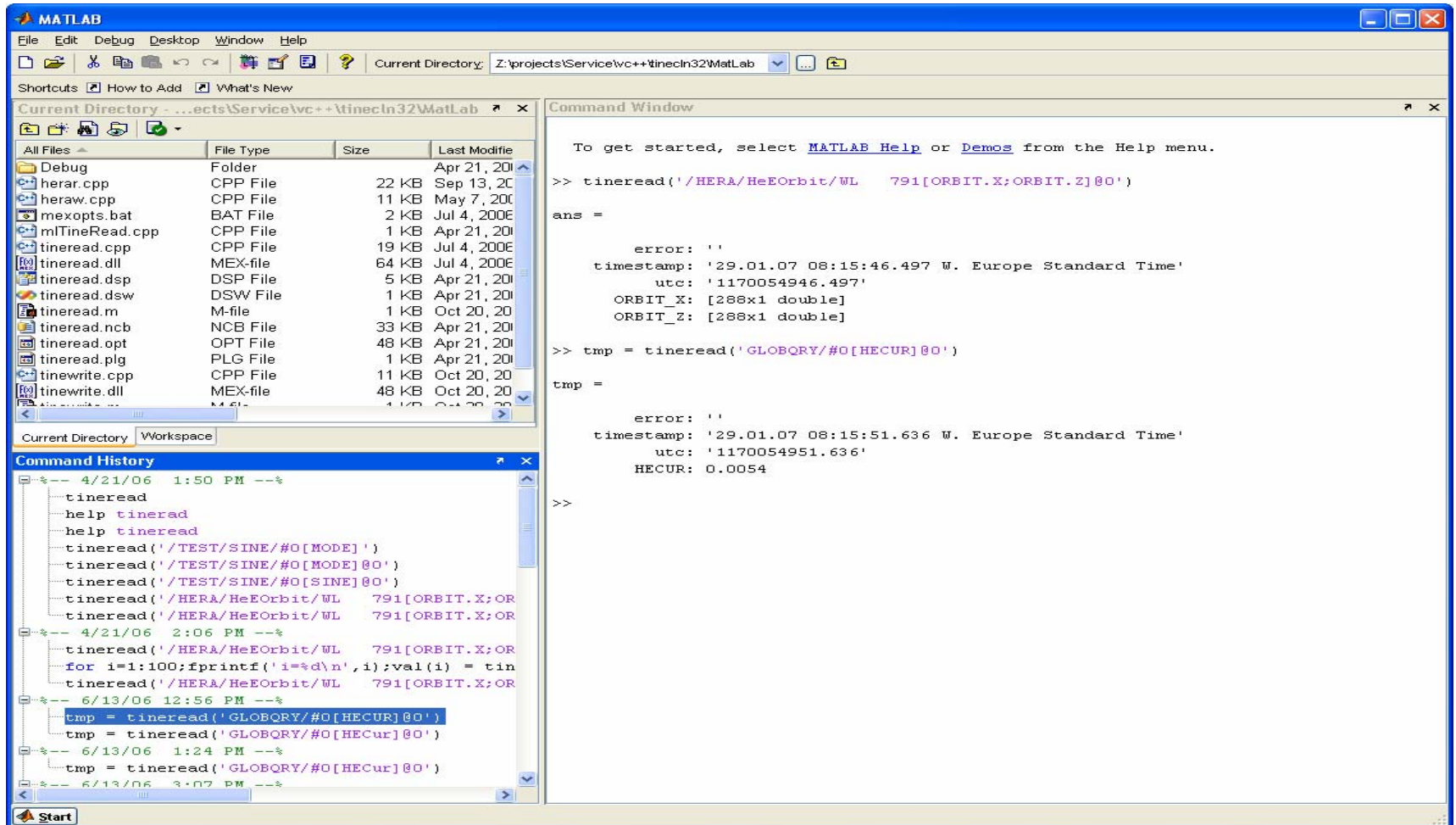


# TINE API

## (Application Programmer's Interface)

- C, C++
- VB
- Java
- C#, VB.NET (rudimentary, but more coming soon!)
- Qt, Lazarus (Visual Pascal) anybody interested?
- Command line scripts
- Plus ...

# TINE and MatLab ...



The screenshot displays the MATLAB environment with the following components:

- Current Directory:** Z:\projects\Service\vc++\tinech32\MatLab
- Command Window:**

```
To get started, select MATLAB Help or Demos from the Help menu.  
>> tineread('/HERA/HeEOorbit/WL 791[ORBIT.X:ORBIT.2]@0')  
  
ans =  
  
error: ''  
timestamp: '29.01.07 08:15:46.497 W. Europe Standard Time'  
utc: '1170054946.497'  
ORBIT_X: [288x1 double]  
ORBIT_2: [288x1 double]  
  
>> tmp = tineread('GLOBQRY/#0[HECUR]@0')  
  
tmp =  
  
error: ''  
timestamp: '29.01.07 08:15:51.636 W. Europe Standard Time'  
utc: '1170054951.636'  
HECUR: 0.0054  
  
>>
```
- Command History:**

```
4/21/06 1:50 PM --%  
tineread  
help tineread  
help tineread  
tineread('/TEST/SINE/#0[MODE]')  
tineread('/TEST/SINE/#0[MODE]@0')  
tineread('/TEST/SINE/#0[SINE]@0')  
tineread('/HERA/HeEOorbit/WL 791[ORBIT.X:ORBIT.2]@0')  
tineread('/HERA/HeEOorbit/WL 791[ORBIT.X:ORBIT.2]@0')  
4/21/06 2:06 PM --%  
tineread('/HERA/HeEOorbit/WL 791[ORBIT.X:ORBIT.2]@0')  
for i=1:100;fprintf('i=%d\n',i);val(i) = tineread('/HERA/HeEOorbit/WL 791[ORBIT.X:ORBIT.2]@0')  
6/13/06 12:56 PM --%  
tmp = tineread('GLOBQRY/#0[HECUR]@0')  
tmp = tineread('GLOBQRY/#0[HECUR]@0')  
6/13/06 1:24 PM --%  
tmp = tineread('GLOBQRY/#0[HECUR]@0')  
6/13/06 3:07 PM --%
```

# TINE and LabView ...

HelpOnTimePolymorphicVIs.vi Block Diagram \*

File Edit View Project Operate Tools Window Help

13pt Application Font

There are only 4 Library's  
Client Get & Client Put  
Server Pull & Server Push

You should mostly use the Top-Level VIs:  
lvTime\_xxxx\_AnyType.vi

CLN GET Any Data      CLN PUT Any Data      Pull Server      Push Server

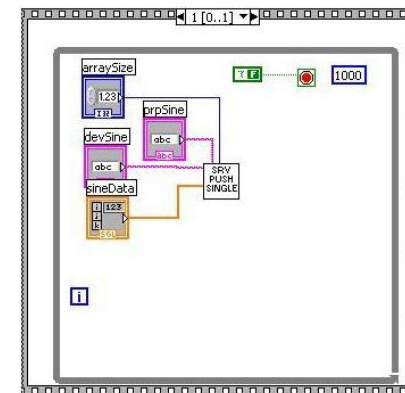
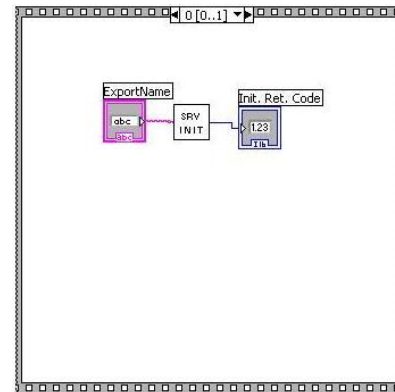
lvTimeClnGetAnyType.vi      lvTimeClnPutAnyData.vi      PullAnyData.vi      PushAnyData.vi

**You'll then get an instance of one of the VIs above, which will morf according to the pin connections.**

CLN GET STR	CLN PUT STR	SRV PULL String	SRV PUSH String
CLN GET SHORT	CLN PUT SHORT	SRV PULL SHORT I16	SRV PUSH SHORT I16
CLN GET SHORT I32	CLN PUT SHORT I32	SRV PULL SHORT I32	SRV PUSH SHORT I32
CLN GET LONG	CLN PUT LONG	SRV PULL LONG I32	SRV PUSH LONG I32
CLN GET LONG I64	CLN PUT LONG I64	SRV PULL LONG I64	SRV PUSH LONG I64
CLN GET SINGLE	CLN PUT SINGLE	SRV PULL SINGLE	SRV PUSH SINGLE
CLN GET SGL	CLN PUT SGL	SRV PULL SGL	SRV PUSH SGL
CLN GET DBL	CLN PUT DBL	SRV PULL DBL	SRV PUSH DBL
CLN GET DBL I64	CLN PUT DBL I64	SRV PULL DBL I64	SRV PUSH DBL I64

**Example:**

Here you should simply edit the constant "Representation" ==>> . Thus change to I16; I32; SGL or DBL. You'll then see how the VIs change. It's also possible to convert to String. However, note that there are currently no String- Arrays available. Hence the corresponding sub VIs are indeterminant.





# TINE Server Wizard Panel

(Jump-start your server application!)

The screenshot displays the TINE Device Server Setup Wizard interface. The main window is titled "TINE Device Server Setup Wizard" and contains several sections:

- Device Server Information:** Includes fields for Export Name (ROBOTSRV), Local Name (ROBEQM), Number of Devices (1), and Project Directory (H:\ROBOTSRV).
- Property Information:** Includes fields for Property Name (STATUS), Description (The Status), Access (READ), Max Value (1), Data Format (NULL), Data Input (String), Data Output (String), Max Data Size (0), and Output Array Type (UNKNOWN).
- History Panel:** Includes fields for Calling Device Name (#0), Archive Rate (secs) (10), Polling Rate (msec) (1000), Archive Heartbeat (secs) (18000), Ringbuffer Depth (600), and Long term Depth (months) (1).
- Alarm Watch Panel:** Includes fields for Calling Device Name (#0), Alarm System ID (0), Severity (high) (10), Counts Threshold (3), Value too high (0), and Value too low (0).

A yellow box labeled "Panel" with an arrow points to the Property Information section. A yellow box labeled "Tree" with an arrow points to a file tree window that is open over the main wizard. The file tree shows a directory structure:

- HE\_EORB\_STAB
  - HeeOrb
    - Stabilizer
      - GETSTATUS
      - StartStop
      - GETCLASS

The "StartStop" file is selected in the tree. The file tree window also shows a "Property name" field (StartStop), a "Description" field (Start/Stop correction), "Data In type" (LONG), "Data In size" (1), "Data Out type" (LONG), and "Data Out size" (1). Buttons for "Cancel" and "Ok" are visible at the bottom of the file tree window.

# Generated Code Example (C)

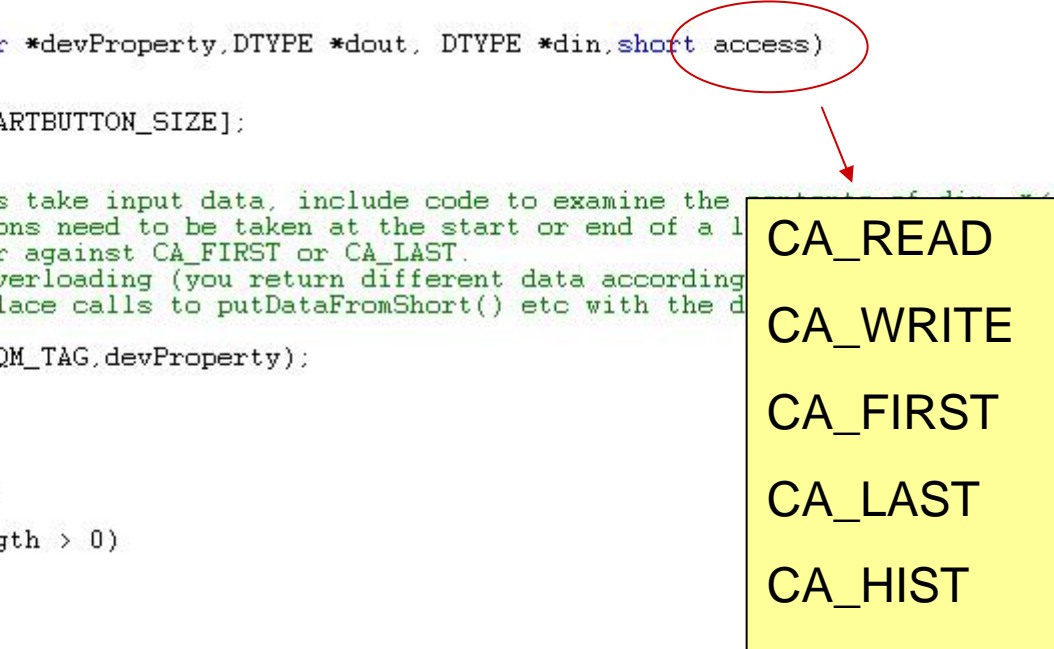
## Call handler

```
int robeqm(char *devName, char *devProperty, DTYPE *dout, DTYPE *din, short access)
{
    int devnr, prpid, i, cc;
    short l_startbutton[PRP_STARTBUTTON_SIZE];
    short l_moverobot_start;

    /* TODO: If READ properties take input data, include code to examine the
    /*      If different actions need to be taken at the start or end of a l
    /*      'access' parameter against CA_FIRST or CA_LAST.
    /*      If allow format overloading (you return different data according
    /*      format), then replace calls to putDataFromShort() etc with the d

    prpid = GetPropertyId(ROBEQM_TAG, devProperty);

    switch (prpid)
    {
        case PRP_STARTBUTTON:
            if (access & CA_WRITE)
            {
                if (din->dArrayLength > 0)
                    ,
            }
    }
}
```



CA\_READ  
CA\_WRITE  
CA\_FIRST  
CA\_LAST  
CA\_HIST  
CA\_ALARM  
...



# TINE: Connecting to Hardware

- Okay, I've gotten started
- Now how do I connect to my hardware?

# TINE Device Layer

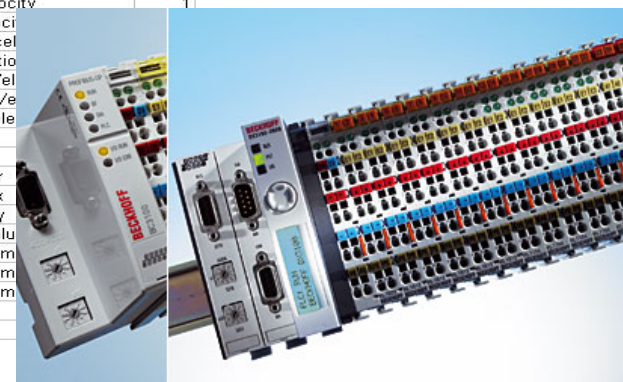
- Middle layer Servers acquire data from other Servers (not connected to hardware)
- “Do it yourself” + your hardware API
- EPICS IOCs (asyn drivers) + Epics2Tine
- LabView IVIs + TINE LabView
- DOOCS + (turn on that TINE thread!)
  
- **CDI** (Common Device Interface) !!!
- **TICOM** (TINE CanOpen Manager)

# CDI:Petra3/EMBL Motor Server with Beckhoff PLCs

- Bus Plugs (ADS DLL) for Beckhoff PLCs for Windows
- A PLC program (Motor and div IOs) in TwinCat
- Use template in CDI database to register devices at initialization:

c diaddr.csv											
A	B	C	D	E	F	G	H	I	J	K	
NUMBER	NAME	BUS	LINE	INDEX	ADDRESS	ACCESS	INPUT	FORMAT	LONG_NAME	LIMIT	
11	0 MOTOR:Start	TEMPLATE	0	0				short	Motor[].Run	01:01	
12	0 MOTOR:CMD	TEMPLATE	0	0				short	Motor[].inCmd	06:04	
13	0 MOTOR:fltCMD	TEMPLATE	0	0				float	Motor[].inRealCmd	06:04	
14	0 MOTOR:TgtPos	TEMPLATE	0	0		RD		long	Motor[]. SollPositionAbs	01:01	
15	0 MOTOR:CurPos	TEMPLATE	0	0		RD		long	Motor[].IstPositionAbs	1	
16	0 MOTOR:Status	TEMPLATE	0	0		RD		Short	Motor[]. Status	1	
17	0 MOTOR:mSteps	TEMPLATE	0	0		RD		Short	Motor[].microSteps	1	
18	0 MOTOR:fSteps	TEMPLATE	0	0		RD		Short	Motor[].fullSteps	1	
19	0 MOTOR:Rps	TEMPLATE	0	0		RD		float	Motor[]. Rps	1	
20	0 MOTOR:Rms	TEMPLATE	0	0		RD		float	Motor[].Rms	1	
21	0 MOTOR:Rfs	TEMPLATE	0	0		RD		float	Motor[].Rfs	1	
22	0 MOTOR:mVeloc	TEMPLATE	0	0		RD		float	Motor[].travelVelo	1	
23	0 MOTOR:rCurPos	TEMPLATE	0	0		RD		float	Motor[].rlstPositi	1	
24	0 MOTOR:rTgtPos	TEMPLATE	0	0		RD		float	Motor[].rSollPositi	1	
25	0 MOTOR:maxVel	TEMPLATE	0	0		RD		float	Motor[].Max_Velocity	1	
26	0 MOTOR:minVel	TEMPLATE	0	0		RD		float	Motor[].Min_Veloci	1	
27	0 MOTOR:maxAcc	TEMPLATE	0	0		RD		float	Motor[].Max_Accel	1	
28	0 MOTOR:thrAcc	TEMPLATE	0	0		RD		float	Motor[].Acceleratio	1	
29	0 MOTOR:LwrLmtV	TEMPLATE	0	0		RD		float	Motor[].limitMinVel	1	
30	0 MOTOR:UprLmtV	TEMPLATE	0	0		RD		float	Motor[].limitMaxVe	1	
31	0 MOTOR:UprLmtA	TEMPLATE	0	0		RD		float	Motor[].limitAccele	1	
32	0 MOTOR:regCont	TEMPLATE	0	0		RD		Short	Motor[].regAry	1	
33	0 MOTOR:Error	TEMPLATE	0	0		RD		Short	Motor[]. Error	1	
34	0 MOTOR:NumError	TEMPLATE	0	0		RD		Short	Motor[]. NumError	1	
35	0 MOTOR:datIdx	TEMPLATE	0	0		RD		Short	Motor[]. dataIndex	1	
36	0 MOTOR:istAry	TEMPLATE	0	0		RD		long	Motor[]. IstPosAry	1	
37	0 MOTOR:datAry	TEMPLATE	0	0		RD		long	Motor[]. analogValu	1	
38	0 MOTOR:regStat	TEMPLATE	0	0		RD		byte	MotorReadReg[. m	1	
39	0 MOTOR:regPosi	TEMPLATE	0	0		RD		short	MotorReadReg[. m	1	
40	0 MOTOR:regExSta	TEMPLATE	0	0		RD		short	MotorReadReg[. m	1	
41	1 Motor1	TWINCAT	1	0	1.0:<MOTOR>			Short			
42	2 Motor2	TWINCAT	1	1	2.0:<MOTOR>			Short			
43	5 SedCnt	SEDPC:1:37E	1	0	15.145.0			Short			

<http://www.beckhoff.com>



# CDI API Details

- **Device Name** (name or number):

- “localhost/cdi/#1”
- “localhost/cdi/#1-#100”
- “localhost/cdi/#1,#3-#10,#99”
- “localhost/cdi/pump1 – pump100”
- ...

- **Device Properties** (methods)

- “RECV”
- “SEND”
- “RECV.SEND.ATOM”
- “SEND.RECV.ATOM”
- “RECV.CLBR”
- “SEND.RECV.CLBR”
- “BUSADDR”
- “BUSNAME”
- “BUSSCAN”
- “BUSERRORS”
- ...

- **Templates + Property-Query Precedence**

- Device Name “dev1.field1” provides Property “field1” with a property specific device list.

Can Use Device Name  
or Device Number !

Read/Write Raw or  
Calibrated data

Atomic pair-wise  
access

Device information

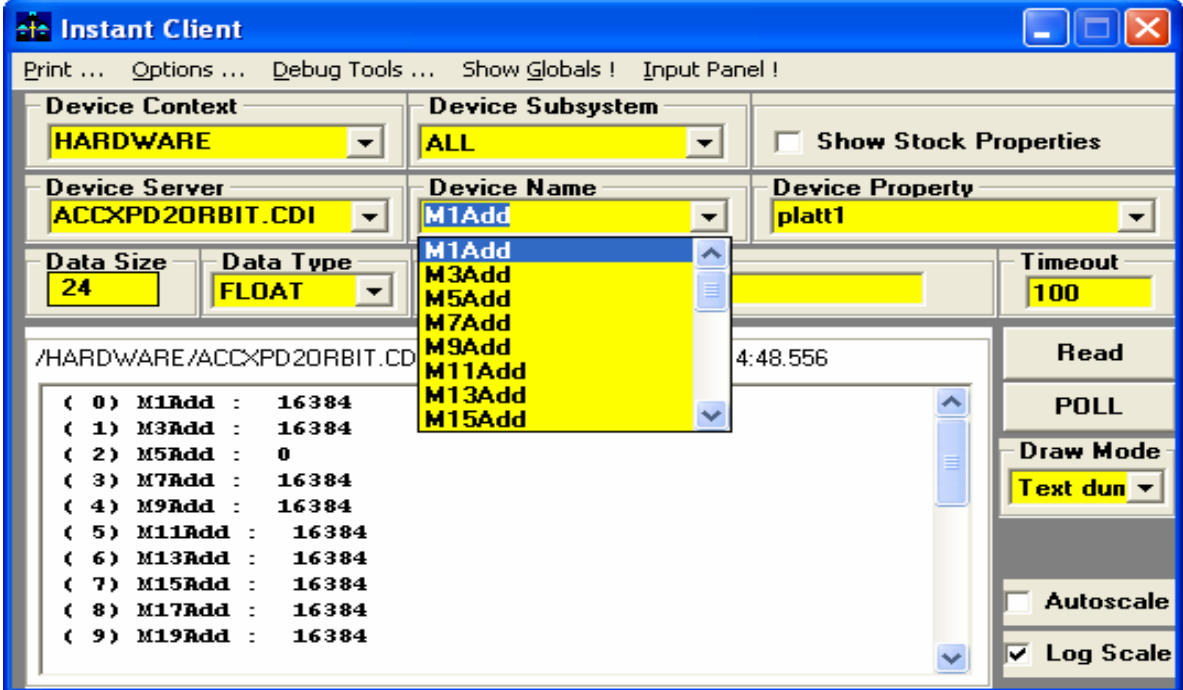
# CDI API Details

You get a (Hardware) Device Server for free!

Template Devices:

“abcd.xyz”

refactored into  
Property Query  
precedent List =>



The screenshot shows the 'Instant Client' application window. The title bar includes 'Print ... Options ... Debug Tools ... Show Globals ! Input Panel !'. The main interface is divided into several sections:

- Device Context:** Set to 'HARDWARE'.
- Device Subsystem:** Set to 'ALL'.
- Device Server:** Set to 'ACCXPD20RBIT.CDI'.
- Device Name:** A dropdown menu is open, showing a list of memory addresses: M1Add, M3Add, M5Add, M7Add, M9Add, M11Add, M13Add, and M15Add.
- Device Property:** Set to 'platt1'.
- Data Size:** Set to '24'.
- Data Type:** Set to 'FLOAT'.
- Timeout:** Set to '100'.
- Read/POLL:** A 'Read' button is visible, and the 'POLL' option is selected.
- Draw Mode:** Set to 'Text dun'.
- Autoscale:** A checkbox that is currently unchecked.
- Log Scale:** A checked checkbox.

The main display area shows a list of memory addresses and their values:

```
/HARDWARE/ACCXPD20RBIT.CD  
( 0) M1Add : 16384  
( 1) M3Add : 16384  
( 2) M5Add : 0  
( 3) M7Add : 16384  
( 4) M9Add : 16384  
( 5) M11Add : 16384  
( 6) M13Add : 16384  
( 7) M15Add : 16384  
( 8) M17Add : 16384  
( 9) M19Add : 16384
```

# TICOM (TINE CanOpen Manager)

TICOM Viewer, version: 1.0

Device Context: TEST Device Subsystem: ALL Device Server: TICOM

CANopen bus

- SYNC producer
- Time Producer
- Devices
  - CAN\_VME\_0x07
  - CAN\_VME\_0x08
  - KnielEnergy3000
  - CAN\_VME\_0x09

Node Basic Information Node Data

OD access

Index: 0x 1000 Subindex: 0x 0

Read Write

Status:

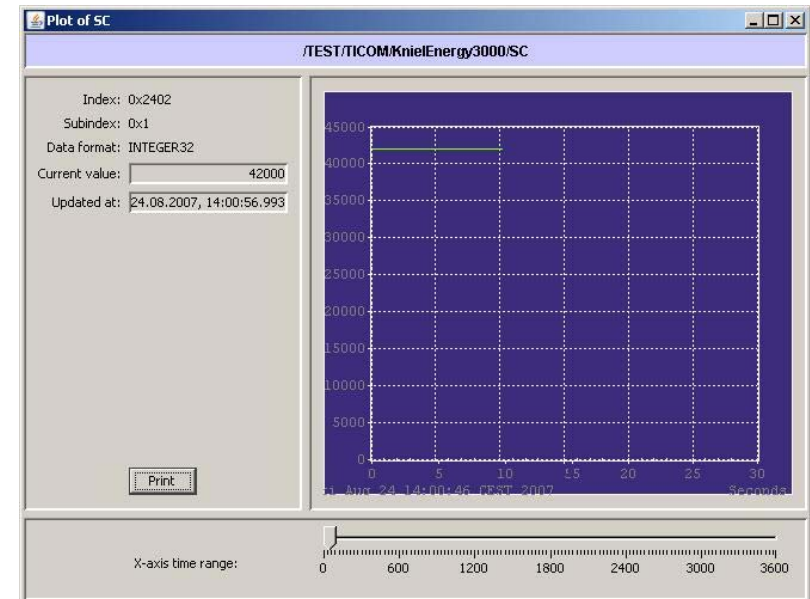
Transmit Process Variables (TPDO)

Variable	Value	Update time	Plot
DEV_STATE	4	24.08.2007, 14:00:05.939	show
FAIL_STATE	0	24.08.2007, 14:00:05.939	show
FLAG_STATE	0	24.08.2007, 14:00:05.939	show
AV	0	24.08.2007, 14:00:05.951	show
SV	90000	24.08.2007, 14:00:05.951	show
AC	0	24.08.2007, 14:00:05.955	show
SC	42000	24.08.2007, 14:00:05.955	show
AP	0	24.08.2007, 14:00:05.959	show

Read Process Variables (RPDO)

Variable	Value	Variable status	Action
OUT	0		send
SB	0		send
SV	0		send
SC	0		send

Success







# TINE: Getting Data from a Server

- Okay, I've got a server
- And it's connected to my hardware
- How do I access the data?

# A Word or Two About the Client API ...

- Fundamental API is :
  - Link based and Not Channel based !
  - **NOT** 'get', 'set', and 'monitor' !!!
  - Think of 'calls' a la RPC or RMI ...
  - Synchronous data acquisition
  - Asynchronous data acquisition
    - Callbacks, events



# Client API: Synchronous Calls

**ExecLink**(devName, devProperty, dout, din, access, timeout)

# Client API: Synchronous Calls

/<context>/<server>/<device>

e.g.: "/PETRA/Vacuum/WLB.HP141"

Device Property or Method

e.g.: "Pressure"

**ExecLink**(devName, devProperty, dout, din, access, timeout)



# Client API: Synchronous Calls

/<context>/<server>/<device>  
e.g.: "/PETRA/Vacuum/WLB.HP141"

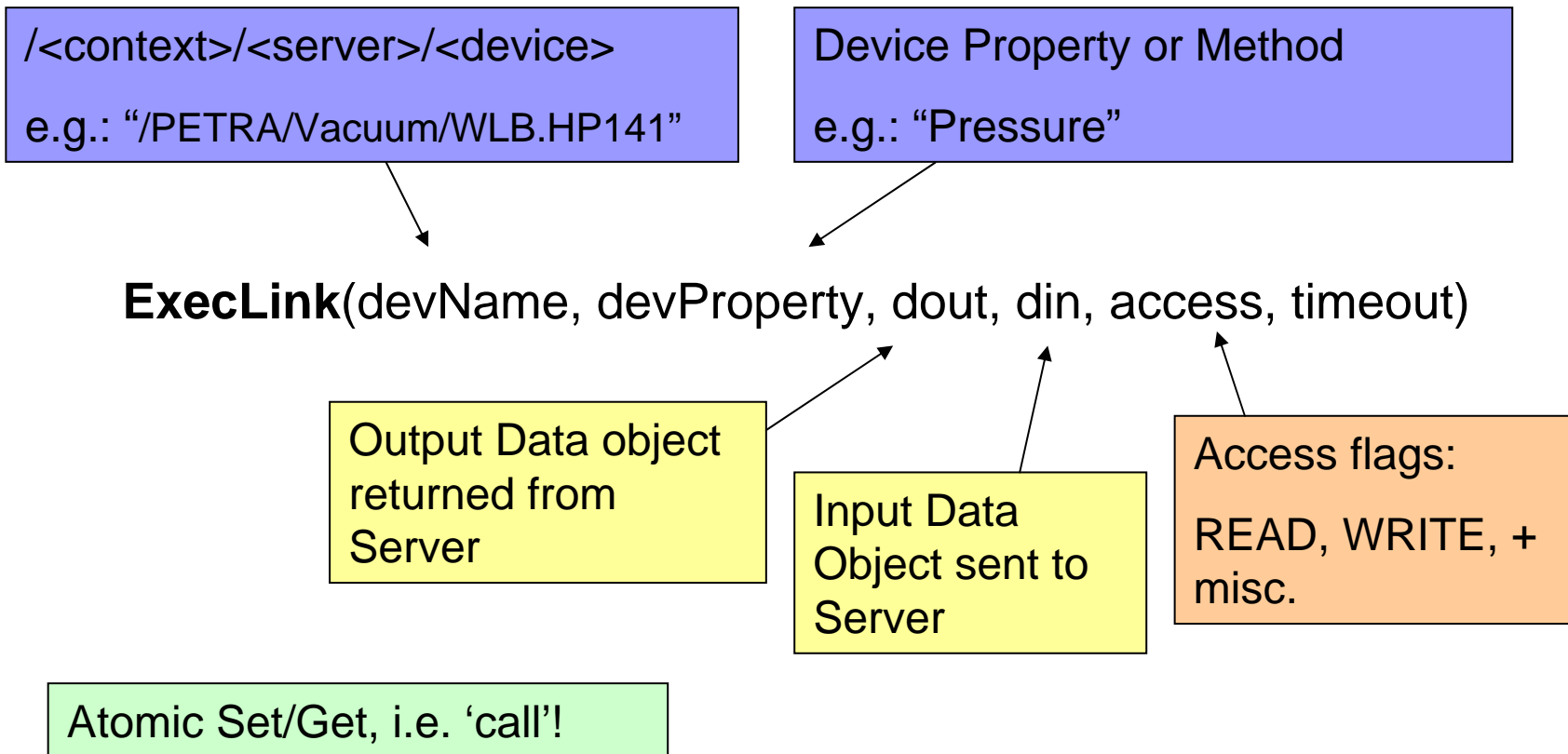
Device Property or Method  
e.g.: "Pressure"

**ExecLink**(devName, devProperty, dout, din, access, timeout)

Output Data object  
returned from  
Server

Input Data  
Object sent to  
Server

# Client API: Synchronous Calls



# Client API: Synchronous Calls

```
int Get( devName, devProperty, dout, timeout )  
{  
    din = NULL;  
    access = CA_READ;  
    ExecLink(devName, devProperty, dout, din, access, timeout)  
}
```

'Get' Wrapper :

# Client API: Synchronous Calls

```
int Set( devName, devProperty, din, timeout )
{
    dout = NULL;
    access = CA_WRITE;
    ExecLink(devName, devProperty, dout, din, access, timeout)
}
```

'Set' Wrapper :



# Client API: Synchronous Calls

e.g. A call to the orbit correction server:

din = tagged struct with optics, current orbit, beam parameters;

dout = tagged struct with new optics, projected corrected orbit;

**ExecLink**("/SERVICE/ORBCOR", "EFFCOR", dout, din, CA\_READ, 1000);

e.g. "Command" Properties: "RESET", "INIT", "START", ...

# Client API: Asynchronous Calls

Analogous to synchronous parameters ...

A light blue rectangular box with a thin black border contains the text "Analogous to synchronous parameters ...". Five black arrows point downwards from the box to the parameters devName, devProperty, dout, din, and access in the function signature below.

**AttachLink**(devName, devProperty, dout, din, access, pollrate,  
void (\*callback)(int,int), callbackID, mode)

# Client API: Asynchronous Calls

Analogous to synchronous parameters ...

**AttachLink**(devName, devProperty, dout, din, access, pollrate,  
void (\*callback)(int,int), callbackID, mode)

Callback with callback id  
and status code ...

CM\_CANCEL  
CM\_SINGLE  
CM\_REFRESH  
CM\_POLL  
CM\_EVENT  
CM\_NETWORK  
CM\_GROUPED  
CM\_WAIT  
+ ...

# Client API: Asynchronous Calls

```
int Monitor( devName, devProperty, dout, rate, callback, id )
{
    din = NULL;
    access = CA_READ;
    mode = CM_REFRESH;
    AttachLink(devName, devProperty, dout, dn, access, pollrate,
        void (*callback)(int,int), callbackID, mode)
}
```

Monitor Wrapper

# A Note about Scheduling

- What do I mean by an 'event'?
  - Asynchronous links => notification events at the client
  - But the client-side call gives a 'polling' parameter to the server.
  - What about latency? (How much time between data acquisition and report to client?)
  - A TINE Server can call the Scheduler!
    - Irrespective of the polling interval requested by the client.
    - No (i.e. minimal) latency !
  - When the Server calls the scheduler => Event!

# Writing GUI Applications (rich clients)

- Use what you've got + TINE Client API for your platform, or ...
- VC++ or VB plus ACOP\* ActiveX control (HPVee, LabView), or direct TINE library calls
- Java + ACOP\* beans (eclipse, net beans), or direct TINE calls
- LabView + TINE client VIs
- MatLab + TINE Client API
- .NET is on the way ...

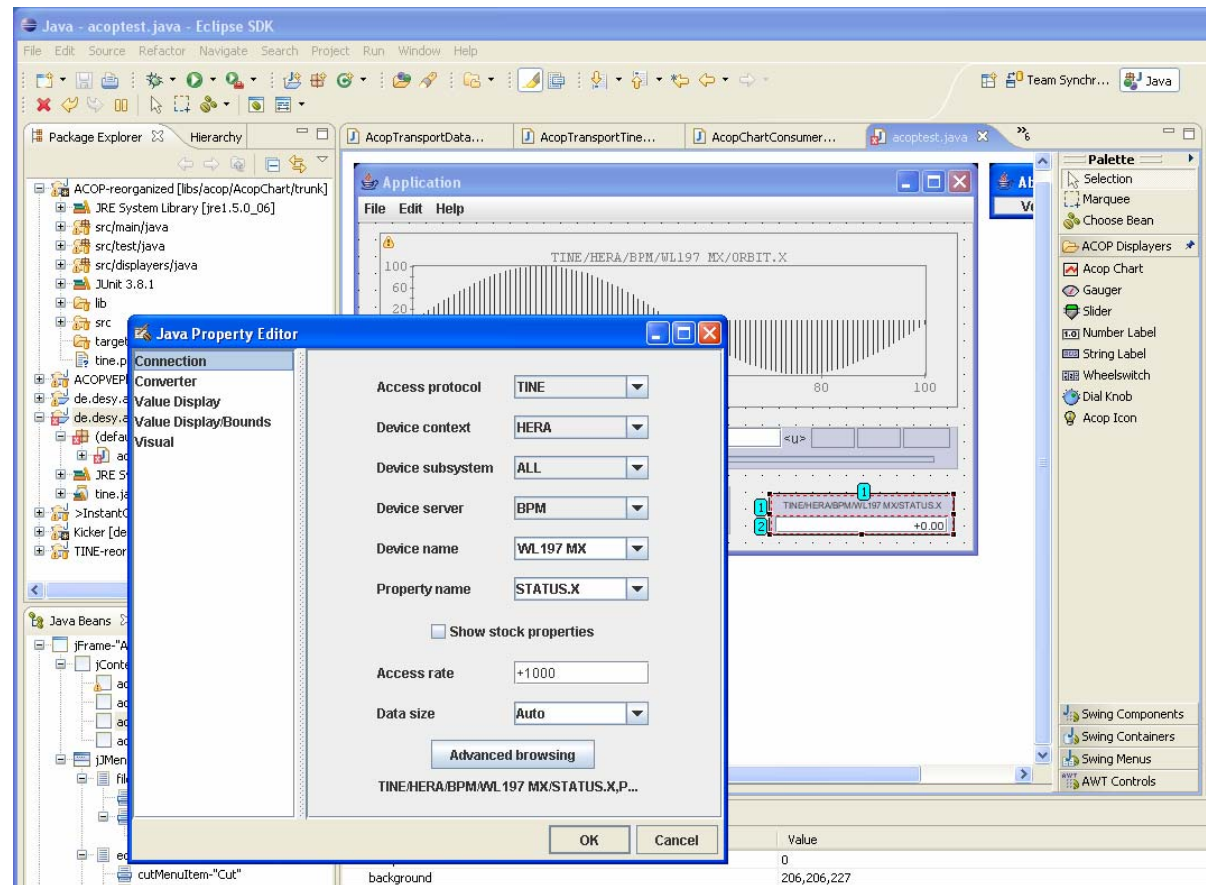
\*Advanced Component Oriented Programming

# TINE and Java ACOP

No Frameworks !

Use Eclipse,  
NetBeans, or  
whatever ...

Browse  
Control system  
at design-time  
with property  
panels or  
customizer ...



# TINE Rich Clients (Java)

The image displays three Java-based control interfaces for the DESY 2 accelerator system.

### DESY 2 Zyklusgenerator

File: Maschine(Desy 2) Optionen Hilfe  
 Geladenes File: PETRA e- (vom 02-20-2007 08:51:44)

Betriebsmode	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Linac 2																								
Linac 2 Pia																								
Pia 125 MHz																								
Pia Desy 2																								
Desy 2																								
Desy 2 Doris																								
Desy 2 Petra																								
LTG Vortrg.																								
LTG Doris																								
LTG Petra																								
Reserve																								
Schleife																								

Buttons: Betrieb, Service  
 List: PETRA e-, DORIS e+, PETRA e+, File 1, Teststrahl 80 mS  
 Status: MstXpDuval2 | 24.08.07 15:49:27

### D2Frequenzkorrektur

File: Maschine(Desy 2) Optionen Hilfe  
 Desy 2 Frequenzkorrektur und Synchronisation

Frequenz: 499.666.502 Hz  
 Offset: 0 Hz  
 FM: AUS  
 Synchronisation: Desy 2 Synchron: EIN, Desy 2 Referenz: Petra  
 Synchro Trigger: Anwahl Betrieb: Elektronen  
 Phasenschieber Bedienung: Anwahl Betrieb: Doris  
 Injektionsphase: 0 Grad  
 Status: MstXpDuval2 | 24.08.07 15:47:32 | Operations Mode [Betrieb] | Serveranwahl [Default] | (4)Kein

### KickerManager

File: Maschine(\$not applicable\$) Optionen Hilfe  
 DeviceTable: DESY2

DevName	Netz	HV	Soll	Delay	Seiki Modul
KICK 2 -	EIN	EIN	4.596 KV	4.628 µS	1 16 48
KICK 2 +	EIN	EIN	1.944 KV	1.235 µS	1 10 32
SEPT J+	EIN	-	0.513 KA	372.9 µS	1 25 80
KI 40 E	AUS	AUS	14.002 KV	11.106 µS	1 9 32
SE 42 E	AUS	-	5.563 KA	670 µS	1 25 96
HBB E	AUS	-	47.802 A	35.2 mS	1 20 160
VBB44ER	EIN	-	27.497 A	37.33 mS	1 20 80
VBB46ER	EIN	-	23.761 A	36.75 mS	1 20 96
KI 24 R	EIN	AUS	4.947 KV	1.565 µS	1 16 32
SE 26 AR	EIN	-	4.425 KA	810 µS	1 25 48
SE 26 BR	EIN	-	5.021 KA	811 µS	1 25 64
HBB R	EIN	-	14.481 A	34.25 mS	1 20 192
KI 16 P	EIN	EIN	4.231 KV	3.214 µS	1 9 48
HKI 36	EIN	AUS	0.112 KV	1.969 µS	1 10 48
VKI 12	EIN	AUS	1.148 KV	1.729 µS	1 17 32

Buttons: Netz, HV, Trigger, Trg. Modul, Error, Delay Modul  
 Status: MstXpDuval2 | 24.08.07 15:46:06 | Operations Mode [Betrieb] | Serveranwahl [Default] | (5)KeinInDatei

### Pulse e-

Fri Aug 24 15:46:06 CEST 2007

Reference Pulse: Sollwert 4.596 KV, Delay 4.628 µS



# Configuring GUI Clients (simple clients)

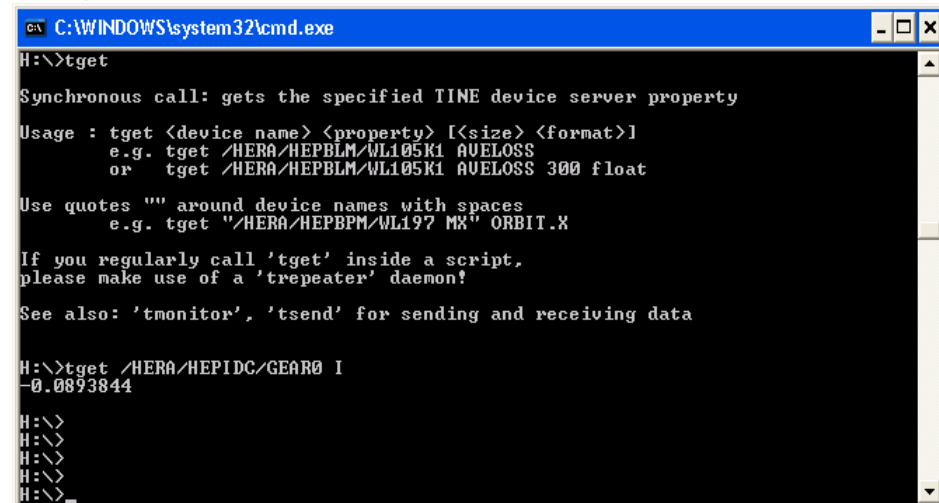
- JoiMint + TINE
- CSS + TINE
- DOOCS DDD + TINE
- ACOP Family of beans + TINE \*

\*More later ....

# TINE command-line tools

- tservers (queries the ENS for registered servers)
- tdevlist (queries a server for its devices)
- tproplist (queries a server for its properties)
- tinfo (queries a server for property information)
- tget (synchronous read-only call to server)
- tsend (synchronous write/read call to server)
- tmonitor (asynchronous read-only poll to server)
- thistory (queries the archive server)

With 'tget' in tcl scripts  
PLEASE use a tineRepeater !



```
C:\WINDOWS\system32\cmd.exe
H:\>tget

Synchronous call: gets the specified TINE device server property

Usage : tget <device name> <property> [<size> <format>]
        e.g. tget /HERA/HEPBLM/WL105K1 AVELOSS
        or   tget /HERA/HEPBLM/WL105K1 AVELOSS 300 float

Use quotes "" around device names with spaces
        e.g. tget "/HERA/HEPBPM/WL197 MX" ORBIT.X

If you regularly call 'tget' inside a script,
please make use of a 'trepeater' daemon!

See also: 'tmonitor', 'tsend' for sending and receiving data

H:\>tget /HERA/HEPIDC/GEAR0 I
-0.0893844

H:\>
H:\>
H:\>
H:\>
H:\>
```

# TINE Web Applications

- TINE Client Servlet
- PHP Interface (Daresbury)
- Web2C (**Web-based controls client**)
  - Thin Ajax Client

http://webmex.desy.de:8080/servlet/WebRepeater?setname=DORIS - Microsoft Internet Explorer

Address http://webmex.desy.de:8080/servlet

DC Current = 131.64815, mA at  
 DC Lifetime = 15.339431, hr at  
 Doris Beam Energy = 4.4423137  
 DORIS Declared State = Sync R  
 DORIS TV Text = SYNCHROT

BLine Beam X = 0.0,0.0,0.0,0.0,0.0,0.22949217,0.6408691,4.205322,0.034179684,0.013427733,0.252

BLine Beam Y = 0.09643554,-0.0053710934,0.002441406,0.019002441406,-0.010986327,0.010001220703,0.001220703,0.017

BLine Monitor X = -0.63354486,2.8991697,2.5390623,0.0,0.0,0.0,1.0351561,6.2182612,2.8942869

BLine Monitor Y = 3.4020994,1.406372,0.6384277,1.4184569,-0.5273437,-1.4855956,-1.904290,1.2341307,0.23071288,0.0,3.002

**Trend**

30.0  
27.5  
25.0  
22.5  
20.0  
17.5  
15.0  
12.5  
10.0  
7.5  
5.0  
2.5  
0.0

15:16:00 15:16:30 15:17:00 15:17:30  
Date

—tine:///TEST/LxSineGen/SineGen0/Sine(0)  
 —tine:///TEST/LxSineGen/SineGen1/Sine(0)

Ventil auf Ventil zu

Fri Aug 24 15:17:38 CEST 2007

Go to DESY Home Page

Device	Wert	Status
Magnet 1	1.00	
Magnet 2	29.00	
Magnet 3	42.40	
Magnet 4	30.96	
Magnet 5	42.53	
Magnet 6	34.09	

Messwert: 45.01

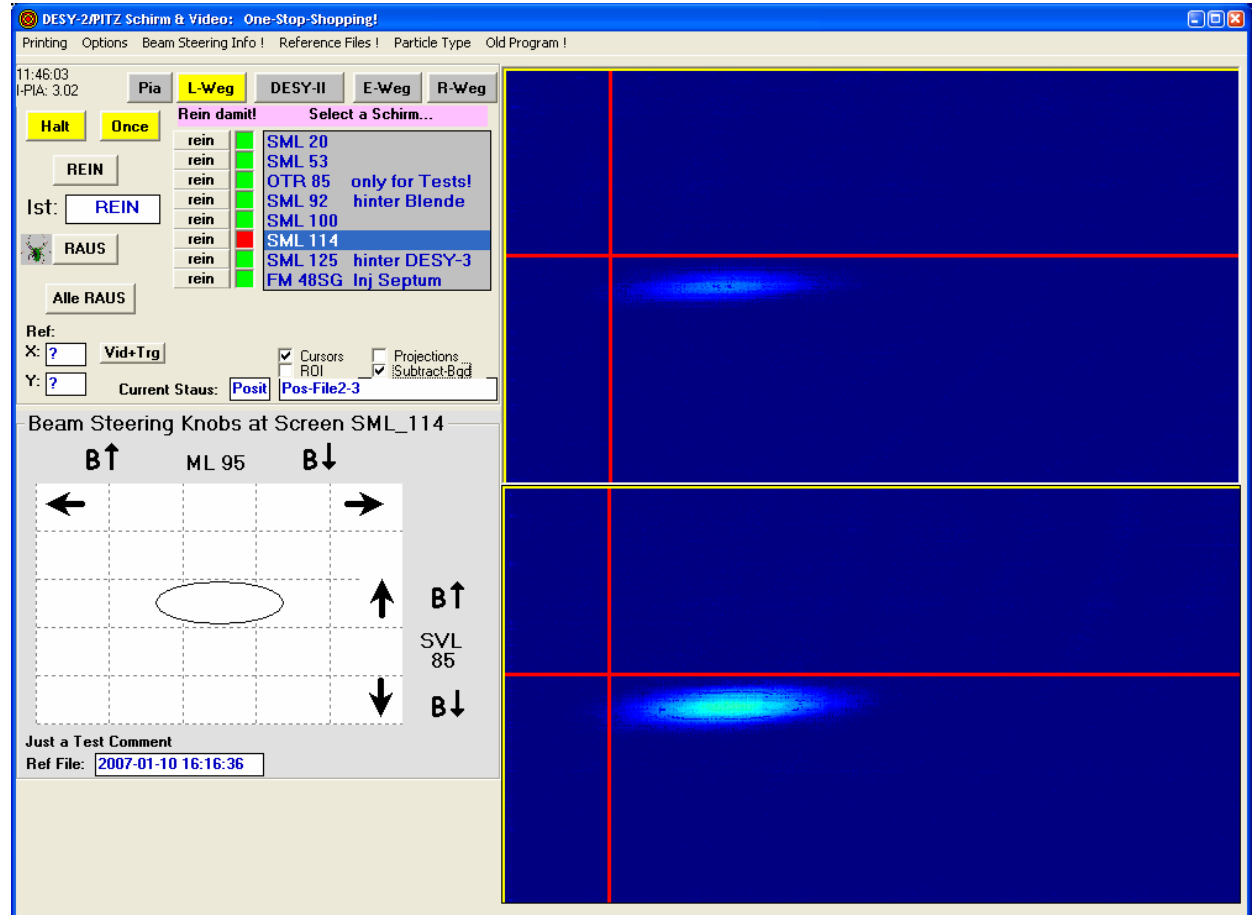
\* \* \* \* \* T h a n k \* Y o u \* H E R A \*

# TINE Video (Multicasting + Scheduling)

.5 Mbyte Video  
Frames @ 10 Hz  
multicast (100 Mb  
ethernet). (also runs  
fine @ 20 Hz)

Uses the  
CM\_NETWORK  
switch.

Server calls the  
Scheduler when a  
new frame is  
grabbed => as real-  
time as it gets !!!  
(beat that EPICS or  
TANGO!)

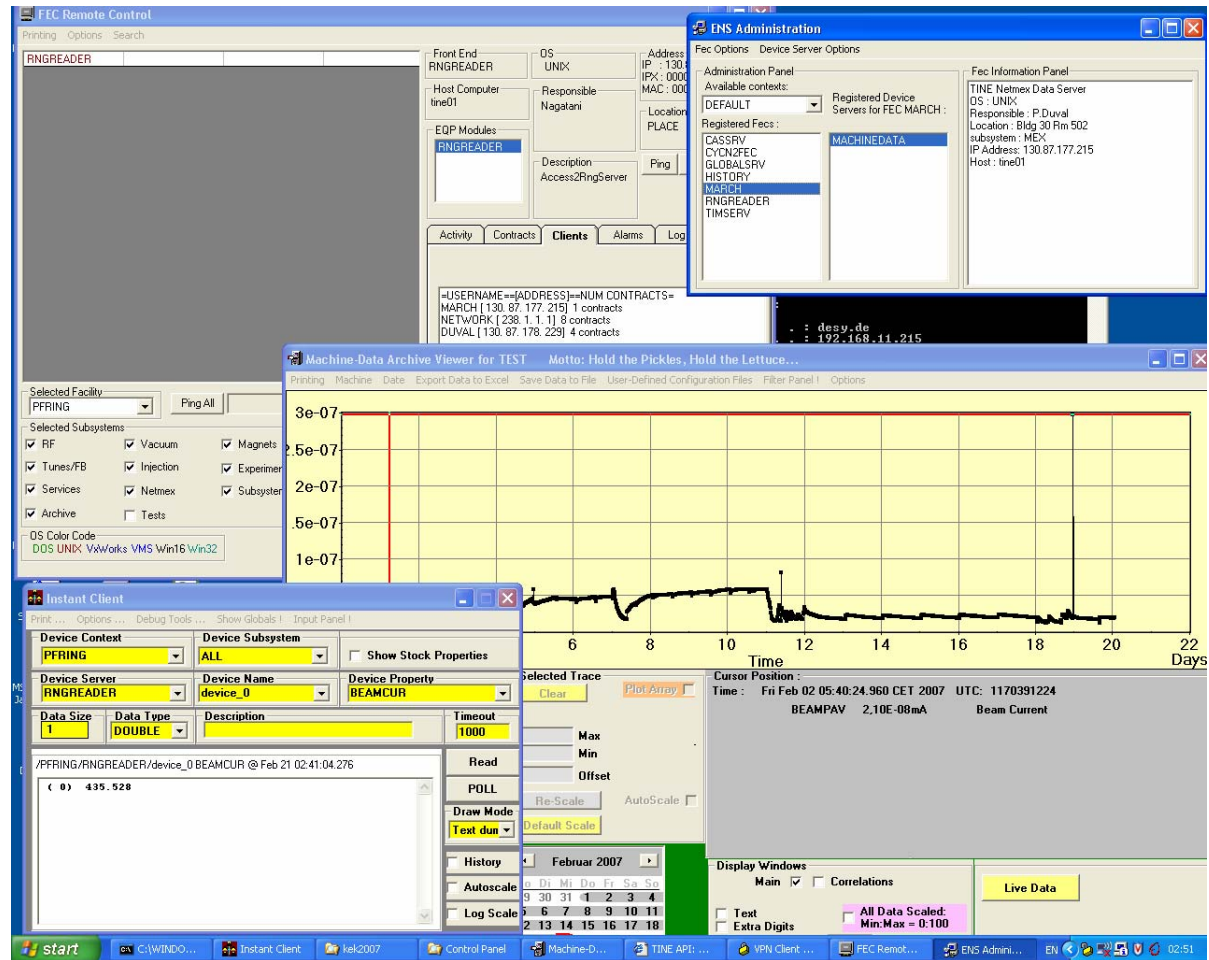


# TINE @ KEK

PF Beam Lines:

Beam parameters multicasted to 70 beam lines

Archive Services



# Connectivity to other systems

- Already embedded in DOOCS
- Epics2Tine runs on any EPICS ioc
- Connect to STARS/COACK via STARS bridge (Japan)
- Tango2Tine (July 2007)
- Anything else needs a gateway

# DOOCS and TINE

- Always a “special” relationship between the two.
- Same (mostly) set of data types.
- Same (basically) naming hierarchy.
- Large set of commonly supported features
  - local histories
  - Meta properties
  - Wildcard calls
- Differences in APIs (not a big deal: use the one you like!)
- Differences in data transfer philosophy (under discussion).
- Efforts to merge are well underway!
- Imagine:
  - Download anything from doocs.desy.de; Install it and use it.
  - Download anything from tine.desy.de; Install it and use it.
  - And it all fits together seamlessly (no tweaking)!

We're not there yet!

# TINE @ DESY





# HERA

- Large machine!
- 6.3 Km Proton-electron storage ring, collider
- Experiments at ZEUS, H1, Hermes
- superconducting proton ring, RF cavities => QPS
- > 100000 control points
- Cryogenics control system
- Principal devices :

Device Type	Nr. of Units
Magnet PSCs	2000
RF Systems	230
Vacuum	3000
BPMs	800
Other beam measurement instrumentation	2000
Quench Protection System	2000
Other diagnostic instrumentation	1000
Air conditioning, water cooling	500

# HERA FECS:

Shutdown on  
June 30, 2007

~160 Device  
Servers;  
Many Operating  
Systems !

The screenshot displays the 'FEC Remote Control' application window. It features a main table of servers with columns for name, IP, and status. Below the table is a 'Device context' section for 'HERA', showing 'Active: 88 of 161 (13:59:04)'. This section includes a grid of checkboxes for various subsystems like QPS, TIM, SER, XXX, HERA, INJ, RF, HIST, SUB, RPT, EXP, TUNE, MAG, PVAK, DIAG, MEX, VAC, and CRYO. At the bottom, there are tabs for 'Activity', 'Contracts', 'Clients', 'Alarms', 'Log File', and 'Stats'. The 'Log File' tab is active, showing a list of log entries with timestamps and descriptions of synchronization events.

Server Name	IP Address	Status
ABSCALC.7	HEMARCH	HETSTATSRV
ADDA	HEMARCHX	HETBUNCH
CVTLPOL	HEMON.5	HETCOOL.5
DIAGHIST.4	HEMON.6	HETIME.5
DIAGHIST.4.100	HEMON.7	HETMON.5
ELPRSTATSRV	HEMON.8	HETMON.6
ETSrver	HEMON.9	HETMON.7
ETSrver.RPT	HENEG.10	HETMON.8
FB-Q-HET	HENEG.11	HETRCRFFB.1
FEBLINUX1	HENEG.5	HETRCRFN.1
H1CJCPROXY	HENEG.6	HETRCRFO.1
H1LUSERVER	HENEG.7	HETRCRFS.1
HEBRU.5	HEORBFX.9	HETRCRFSLC.1
HEBRU.6	HEBPBM	HETRCRFWL.1
HECASSRV	HEPBUNCH	HETRCRFWR.1
HECOOL.5	HEPCMFL	HETSFN
HECOOL.6	HEPHAKO.5	HETSFO
HECOOL.7	HEPHFTRC	HETSFPBEAM
HECOOL.8	HEPIDC.1	HETSFS
HECOOL.9	HEPKOLLI	HETSFW1
HEIALOG	HEPMARCH	HEVACN.10
HEEBLM	HEPQPSH1	HEVACN.11
HEEBUNCH	HEPQPSZEUS	HEVACN.6
HEEHFFB	HEPTRANSF	HEVACN.7
HEEIDC.1	HEPTRANSF-VXW	HEVACS.10
HEEKOLLI.5	HEPVACPROXY	HEVACS.11
HEEMHFS	HEQPSPROXY	HEVACS.6
HEEPTICI	HERA52	HEVACS.9
HEETRANF-VXW	HERA208	HEVAKHIST.4
HEETRANSF	HEREFORB.0	HEVAKHIST.4.100
HEEVENT	HEREFORB.1	HEWIRE.5
HEFECSTATSRV	HESEKI.5	HEXFEREFF
HEGLOBSLRV	HESEKI.6	HE_EORB_STAB
HEGLOBSLRV2	HESEKI.7	HE_PORB_STAB
HEHISTORY	HESRDET.4	HPCHROM
HEHISTORY.RPT	HESRDETS.4	HPTUNE
HEHIT.5	HE SRLUMIS.0	ILMAG.9
HEIDCSP.1	HE SRLUMIS.0	MEX-ZEUS4
HEMAG	HE SRROST.0	MEX-HERMES
HEMAGHIST.0	HE SRRWEST.0	MHFHISTORY

**Device context: HERA**  
Active: 88 of 161 (13:59:04)

**Selected Subsystems:**

<input checked="" type="checkbox"/> QPS	<input checked="" type="checkbox"/> INJ	<input checked="" type="checkbox"/> EXP	<input checked="" type="checkbox"/> DIAG
<input checked="" type="checkbox"/> TIM	<input checked="" type="checkbox"/> RF	<input checked="" type="checkbox"/> TUNE	<input checked="" type="checkbox"/> MEX
<input checked="" type="checkbox"/> SER	<input checked="" type="checkbox"/> HIST	<input checked="" type="checkbox"/> MAG	<input checked="" type="checkbox"/> VAC
<input checked="" type="checkbox"/> XXX	<input checked="" type="checkbox"/> SUB	<input checked="" type="checkbox"/> PVAK	<input checked="" type="checkbox"/> CRYO
<input checked="" type="checkbox"/> HERA	<input checked="" type="checkbox"/> RPT	<input type="checkbox"/> TEST	

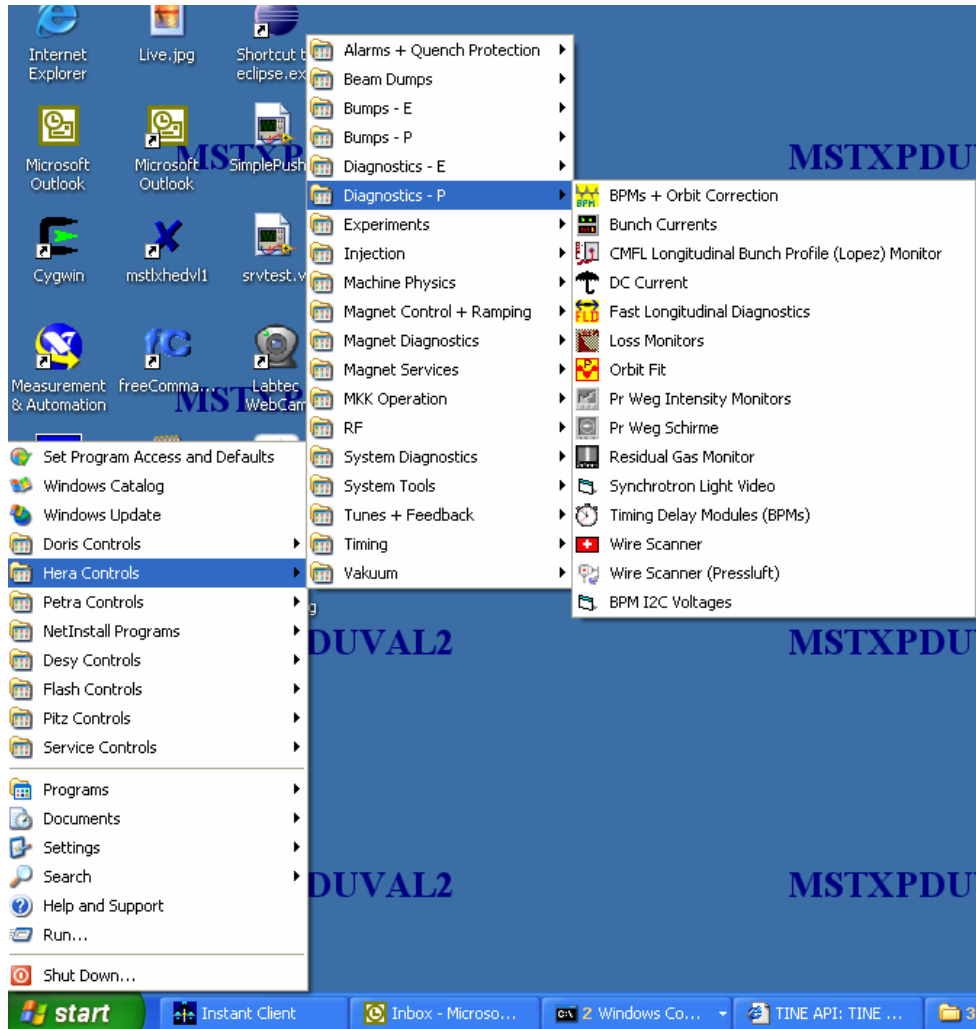
**OS Color Code:** Dos Unix VxWorks VMS Win16 Win32 Java  
**FEC Importance:** ALL

**Log File Activity:**

```

3:25:27.466 CDT[HEVACN.10] global synchronization offset: -1.37107 sec
18:09:07 23:25:34.506 CDT[HEVACN.7] largest correction: 4.56881 sec (nr. corrections 23)
18:09:07 23:25:34.506 CDT[HEVACN.11] largest correction: 4.55891 sec (nr. corrections 23)
18:09:07 23:25:27.466 CDT[HEVACN.1] global synchronization offset: -1.37107 sec
18:09:07 23:25:34.646 CDT[HEVACN.1] largest correction: 4.13158 sec (nr. corrections 21)
18:09:07 23:25:34.656 CDT[HEVACN.10] largest correction: 4.56365 sec (nr. corrections 19)
19:09:07 06:27:10.148 CDT[HEVACN.6] global synchronization offset: -1.26262 sec
19:09:07 06:27:10.198 CDT[HEVACN.6] largest correction: 7.79293 sec (nr. corrections 41)
20:09:07 06:08:31.314 CDT[HEVACN.6] global synchronization offset: -2.25703 sec
20:09:07 06:08:31.625 CDT[HEVACN.6] largest correction: 1.93445 sec (nr. corrections 2)
20:09:07 06:08:35.315 CDT[HEVACN.7] global synchronization offset: -1.02061 sec
20:09:07 06:08:35.395 CDT[HEVACN.7] largest correction: 7.09 sec (nr. corrections 10)
20:09:07 06:08:35.315 CDT[HEVACN.11] global synchronization offset: -1.00061 sec
20:09:07 06:08:35.445 CDT[HEVACN.11] largest correction: 7.09 sec (nr. corrections 10)
20:09:07 21:55:36.065 CDT[HEVACN.7] global synchronization offset: -1.18025 sec
20:09:07 21:55:36.325 CDT[HEVACN.7] largest correction: 3.68691 sec (nr. corrections 17)
20:09:07 21:55:32.065 CDT[HEVACN.6] global synchronization offset: -3.60867 sec
20:09:07 21:55:33.897 CDT[HEVACN.6] largest correction: 8.2438 sec (nr. corrections 38)
20:09:07 21:55:36.065 CDT[HEVACN.11] global synchronization offset: -1.18025 sec
20:09:07 21:55:36.385 CDT[HEVACN.11] largest correction: 3.68691 sec (nr. corrections 17)
20:09:07 21:55:36.065 CDT[HEVACN.1] global synchronization offset: -1.18025 sec
20:09:07 21:55:36.065 CDT[HEVACN.10] global synchronization offset: -1.18025 sec
    
```


# HERA Console Applications



< 200 Console Applications available

~ 20 Console Applications needed for Normal Operation

Console Applications generally "Rich Clients"



# TINE Release 4.0.0 Highlights

- Everything should work as before (whew!)
  - Backwards and forwards compatible to Release 3.xx
- Bigger, Better, Faster, Stronger ....
- Lots of refactoring, optimizing, ...
- Uh .... So, what's new ?

# TINE Release 4.0.0 Highlights

- Allowed Name lengths greatly increased!
  - Registered Device Names, Properties -> 64 chars
    - Device Name String up to 1025 chars
    - e.g. “M1Adc.rstTrg,M3Adc.rstTrg,M5Adc.rstTrg,...”
  - Structure, Bitfield Tags up to 16 chars
- Case Insensitivity
  - e.g. No difference between “TEST” and “Test”
  - e.g. No difference between “NR 64 MO” and “nr 64 mo”
  - e.g. No difference between “RESET” and “reset”

# TINE Release 4.0.0 Highlights

## ■ Server Configuration

### □ API Configuration as before

- RegisterFecInformation(), RegisterProperty(), etc.

### □ .csv Configuration as before

- FEC\_HOME -> fecid.csv

#### □ Subdirectories for Equipment Modules

- exports.csv, history.csv, alarms.csv, devices.csv

### □ **.xml Configuration !!**

- Single xml file : fec.xml

# fec.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
- <COMPUTER>
- <FEC>
  <NAME>MSTXPDUVAL03.23</NAME>
  <PORT_OFFSET>23</PORT_OFFSET>
- <EQM>
  <NAME>BPMEQM</NAME>
  <SERVER>DoBeam</SERVER>
  <CONTEXT>DORIS</CONTEXT>
  <SUBSYSTEM>DIAG</SUBSYSTEM>
  <DEVICE_SPACE>42</DEVICE_SPACE>
- <DEVICE>
  <NAME>NR 3 MO</NAME>
</DEVICE>
- <DEVICE>
  <NAME>NR 6 MO</NAME>
</DEVICE>
- <DEVICE>
  <NAME>NR 7 MO</NAME>
</DEVICE>
- <PROPERTY>
  <ID>1</ID>
  <NAME>OrbitX</NAME>
  <DESCRIPTION>[- 10:10 mm]Horizontal Orbit</DESCRIPTION>
  <SIZE_OUT>42</SIZE_OUT>
  <DTYPE_OUT>float.CHANNEL</DTYPE_OUT>
  <ACCESS>READ</ACCESS>
  <REDIRECTION />
</PROPERTY>
- <PROPERTY>
  <ID>2</ID>
  <NAME>OrbitY</NAME>
  <DESCRIPTION>[- 10:10 mm]Vertical Orbit</DESCRIPTION>
  <SIZE_OUT>42</SIZE_OUT>
  <DTYPE_OUT>float.CHANNEL</DTYPE_OUT>
  <ACCESS>READ</ACCESS>
  <REDIRECTION />
</PROPERTY>
</FOM>
```

# TINE Release 4.0.0 Highlights

## ■ New Data Formats

### □ CF\_XML

- Sent as a text string

### □ CF\_VIDEO

- Video header + frame

### □ CF\_BITFIELD8, CF\_BITFIELD16, CF\_BITFIELD32, CF\_BITFIELD64

- Data type: DBITFIELD

- bitfield segments from 1 bit to full range have names

- Bitfield Registry

- e.g. `addFieldToBitField("thisfec","StsBits",0xf0,"field3");`
- Property "Status" registered with format CF\_BITFIELD16
- "Status.field3" gives 2<sup>nd</sup> Nibble of the Status Word!



# TINE Release 4.0.0 Highlights

- Expanded Data Object (DTYPE)
  - dArrayLength (as before)
  - dFormat (as before)
  - dTimeStamp (as before)
  - dTag (as before, but now longer)
  - **dStamp** (a user supplied integer tag)
  - **sysStamp** (a systematic integer tag : e.g. cycle number, run number)
  - **xferReason** :
    - CX\_NULL, CX\_RESPONSE, CX\_STALE, CX\_HEARTBEAT, CX\_EVENT, CX\_TIMER, etc.

# TINE Release 4.0.0 Highlights

- Expanded Alarm Message Structure
  - timestamp (secs + **usecs**)
  - **starttime (secs + usecs)**
  - code (as before)
  - status (as before)
  - data (**64 bytes!** – was 6)

# TINE Release 4.0.0 Highlights

## ■ Dynamic Client-side Name Caching

### □ Name resolution:


- First ask the configured ENSes
- Then consult the dynamic Name cache
- Then consult the static Name cache (if present)

□ Once a Client has acquired an Address the local dynamic cache is updated!

□ Upon ENS failure, the last known address is probably as good as anything else!

# TINE Release 4.0.0 Highlights

- Revised Multicast Address Scheme (Kars Ohrenberg)
  - Globals multicast (Producers)
  - Publisher multicast
  - Services multicast
- Classic (old) way:
  - Each has a single multicast group
- Standard (new) way:
  - Each server on the control net has its own multicast group!
  - No more 'N-Producer' problem!
  - Services multicast still uses a single systematically known multicast group.



# TINE Release 4.0.0 Highlights

- Local history system to use “worst-case” non-fragmented files
- Time Synchronization to 100 msec.
  - Requires client-side daemon
- Forced transfer efficiency of multi-channel arrays, bitfields, user-defined structures.
- Adjustable Local History, Alarm settings from remote location.

# TINE Installation Issues

- <http://tine.desy.de>
  - Visit the download section and chose your platform.
  - Use setup tools available.
  - Installation takes a few minutes
  - Don't expect too many miracles (you might have to read the README.txt)

TINE API: TINE API for Console Applications - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://adweb.desy.de/mst/tine/>

[Main Page](#) | [Features](#) | [Central Services](#) | [csv-Files](#) | [Types](#) | [Transfer](#) | [Access](#) | [API-C](#) | [API-VB/ActiveX](#) | [API-Java](#) | [Examples](#) | [Downloads](#)

## TINE (Three-fold Integrated Networking Environment)

Note: Remember: *This Is Not Epics!*  
But you can run [EPICS](#) iocs on TINE using Epics2Tine.  
TINE is embedded in [DOOCS](#), so you can also run DOOCS clients and servers using TINE.  
TINE can also be used in a [STARS](#) system and via a STARS-bridge in a [COACK](#) system.  
But you might want to go native ...

General	APIs	Services	Examples & Tutorials	Recipes
<a href="#">Bird's Eye View</a>	<a href="#">C API</a>	<a href="#">Alarm System</a>	<a href="#">Getting Started</a>	<a href="#">DOS</a>
<a href="#">Overview</a>	<a href="#">Visual Basic API</a>	<a href="#">Archive System</a>	<a href="#">TINE Server Wizard</a>	<a href="#">UNIX</a>
<a href="#">Features</a>	<a href="#">Java API</a>	<a href="#">Post Mortem System</a>	<a href="#">Console Server (C)</a>	<a href="#">VxWorks</a>
<a href="#">Configuration</a>	<a href="#">Java API (ACOP)</a>	<a href="#">State Server</a>	<a href="#">Console Client (C)</a>	<a href="#">VMS</a>
<a href="#">Data Types</a>	<a href="#">Buffered API</a>	<a href="#">Dialog Server</a>	<a href="#">GUI Server (VB)</a>	<a href="#">Win16</a>
<a href="#">Transfer Modes</a>	<a href="#">LabView API</a>	<a href="#">Name Server</a>	<a href="#">GUI Client (VB)</a>	<a href="#">Win32</a>
<a href="#">Access Flags</a>	<a href="#">MatLab API</a>	<a href="#">Remote Services</a>	<a href="#">Console Client (Java)</a>	<a href="#">Workshop Tutorial (Buffered Server)</a>
<a href="#">Array Types</a>		<a href="#">Network Globals</a>	<a href="#">GUI Client (Java)</a>	<a href="#">Workshop Tutorial (Standard Server)</a>
<a href="#">Time Stamps</a>		<a href="#">Time Synchronization</a>	<a href="#">Console Server (Java)</a>	<a href="#">Workshop Tutorial (Clients)</a>
<a href="#">Naming Conventions</a>		<a href="#">Security</a>	<a href="#">Trouble Shooting</a>	<a href="#">CDI Tutorial</a>
<a href="#">Data Flow Tips</a>		<a href="#">Netmex</a>		
		<a href="#">Command Line</a>		

TINE is fully supported by [ACOF](#), [Abeans](#), [JoiMint](#), and will be supported by [Control System Studio](#).

You may want to have a look at the release note for versions [3.20](#), [3.30](#) or [3.31](#) or take a quick look at a [Bird's Eye View](#) of TINE.

Download TINE [Download page](#)

Local intranet

# TINE Installation Issues

- Installation from the <http://tine.desy.de> is actually for a TINE Site Administer.
  - Individual Installs at a Site should come from the Site; i.e. after
    - ENS location is established
    - deployment policies have been decided
    - etc.
- Install packages provide
  - Libraries, source code
  - core applications.
  - core servers.
    - ENS
    - Archive
    - Alarm
    - Event
    - etc.
  - examples
  - tools
    - Command line
    - LabView (where applicable)
    - MatLab (where applicable)

# TINE Installation Issues

- Site Deployment policies

- What is centrally supported at a site?
  - e.g. Labview 7.1 vs. Labview 8.2, etc.
  - e.g. Java 1.5 vs. Java 1.6
  - Flavors of Linux, Unix (different libc, default shells, etc.)
  - Flavors of Windows
    - (3 cheers for Microsoft! Win32 API works from win95 to winXp with all service packs thrown in)
  - VxWorks builds
    - with/without floating point
    - with/without multicast support
    - etc.
  - And so on ...



# Where to get TINE?

- <http://tine.desy.de>
  - Release 3.31.23
  - Release 4.0.0 (next week)
    - All platforms except java (in ~ 1 month)

