TINE STUDIO, MAKING LIFE EASY FOR ADMINISTRATORS, OPERATORS AND DEVELOPERS

P. Duval, M. Lomperski, DESY, Hamburg, Germany
J. Bobnar, Cosylab, Ljubljana, Slovenia

Abstract

A mature control system will provide central services such as alarm handling, archiving, location and naming, debugging, etc. along with development tools. It has become common to refer to the collection of these services as a 'studio'. Indeed Control System Studio (CSS) [1] strives to provide such services independent of the control system protocol. Such a 'one-size-fits-all' approach is likely, however, to focus on features and behavior of the most prominent control system protocol in use, providing a good fit there but perhaps offering only a rudimentary fit for 'other' control systems. TINE [2] is for instance supported by CSS but is much better served by making use of TINE Studio. We report here on the rich set of services and utilities comprising TINE Studio.

INTRODUCTION

What one ends up referring to as a control system *studio* is primarily all about *the user*. In a perfect world *control* of a machine could be *blind*, where input simply leads to output and there is no necessity of involving something as grandiose as a *studio*. In reality, this will never be the case. Namely, a user operating a machine will inevitably encounter problems which need to be quickly diagnosed and corrected. A user studying the machine will need to readily access all machine parameters. A user developing controls for a machine will also need to produce robust, functional and intuitive applications in a timely manner. A user administering a machine will need to be able to check the integrity of the controls and make modifications where necessary. We refer to the tools the user has at his disposal toward these ends as a *studio*, and where the control system is TINE, as TINE Studio.

The operator and machine physicist will make good use of TINE central services. Namely, the archive and alarm systems and their peripheral applications are major workhorses in machine control.

Likewise, developers will make use of server *wizards* and configuration editors, where front-end and middle layer servers are concerned, and rapid application development (RAD) components and GUI builders, where client applications are concerned.

Finally, control system administrators will profit from the many database management tools, report generators, and monitoring tools available.

In each case the utilities available strive to present an intuitive *cut-to-the-chase* application and take pains to avoid *popup pollution* and needless searching for results in arcane tree structures.

CENTRAL SERVICES

Archive System

An archive system can mean different things to different people. In the context of machine control, we will not refer to the data acquisition system necessary to process the myriads of data taken at an experiment. Instead we refer to the archiving of machine data versus time or versus event. The trend of relevant machine parameters can help diagnose operational problems when it is easy to browse and correlate among them. Likewise, investigating data and settings at a specific event, be it post-mortem, such as an RF trip, or otherwise is also invaluable.

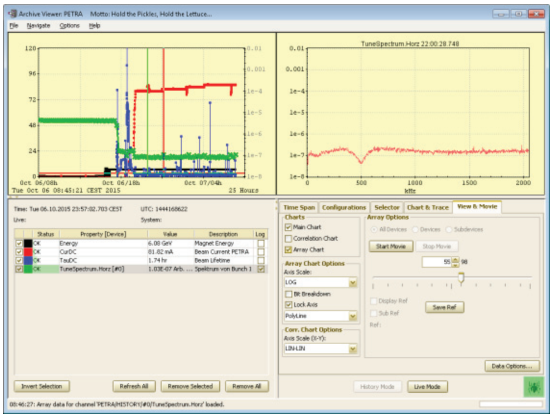
TINE allows the central archiving of any machine parameter with optional storage criteria and does not modify, reduce, or expunge the data in any way after storage. A TINE server may also provide *local* archiving of any parameter.

The TINE archive viewer uses sparse (raster) data with points of interest for rapid browsing over very large time ranges. It also makes use of *optical zooming* whereby a new data request is made whenever a zoom operation over the time interval is made: the finer the zoom, the finer the raster. At any stage the number of points in the selected time range plotted versus the number of points available is easily seen. The archive viewer also decides on the optimal source (central or local) from which to retrieve the archive data.

An archived parameter can be of any available format, be it scalar, multi-channel array, waveform trace array, or video image. The primary archive view is, however, a trend, i.e. a value versus time. Consequently if an array element is selected for viewing then a trend of that element is displayed. It should be noted that the TINE archive viewer can also display the trend versus a system stamp (which might be a pulse or cycle number) instead of a time stamp. In any case, a secondary chart is able to display a snapshot of the entire array (multi-channel or waveform) at the selected (clicked on) time in the trend chart. Similarly, video images will display a frame count in the primary trend chart and the entire image in the secondary chart.

Figure 1 shows the trends of selected parameters in PETRA, including the horizontal tune, over a 24-hour period. The secondary plot shows horizontal tune at the time (green line) clicked on by the operator.

ISBN 978-3-95450-148-9



Time: Tue 06.10.2015 23:57:02.703 CEST UTC: 1444168622

Live: System:

		Status	Property [Device]	Value	Description	Log
☑	■	OK	Energy	6.08 GeV	Magnet Energy	□
☑	🟥	OK	CurDC	81.82 mA	Beam Current PETRA	□
☑	🟦	OK	TauDC	1.74 hr	Beam Lifetime	□
☑	🟩	OK	TuneSpectrum.Horz [#0]	1.03E-07 Arb. ...	Spektrum von Bunch 1	☑

☑ Main Chart
□ Correlation Chart
☑ Array Chart

Axis Scale: LOG
□ Bit Breakdown
☑ Lock Axis
PolyLine

Axis Scale (X-Y): LIN-LIN

55 98

□ Display Ref
□ Sub Ref

08:46:27: Array data for channel 'PETRA/HISTORY/#0/TuneSpectrum.Horz' loaded.

Figure 1: Typical Archive Viewer display.

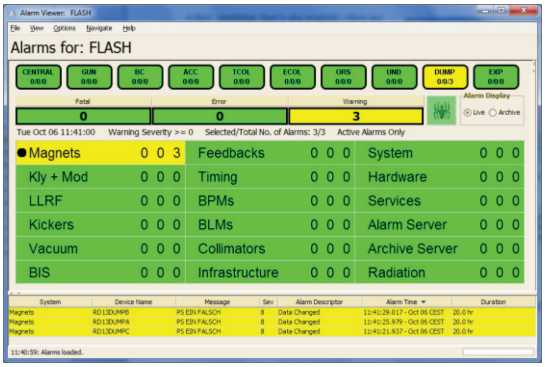
A third *correlation* chart is also available, where any two parameters can be plotted against one another (and even after applying formulae). Frequently the simple correlations the user makes with his eye in the primary trend chart are all that is needed to pinpoint operations problems.

The TINE event system allows specific data retrieval and storage based on an event trigger. This is most often employed in the case of transient recorders which have a large amount of data in memory following a post-mortem event. A trigger can then instruct the event system to lock and acquire the data and then release the lock when finished.

TINE Studio provides a generic Event Viewer for the general browsing of all events. But as transient recorders tend to be the principal event use-case, a specific Transient Recorder Viewer is also available.

Alarm System

The TINE Alarm system will in many cases provide the first indication of incipient problems and perhaps the smoking gun in cases of operations failures. An *alarm* is in general an entity that belongs to a device. It has a severity and other meta-information and might contain specific data. It also has a duration. Alarms originate at a front-end or middle layer server and are pulled by the central alarm server, where they are also archived.



Alarms for: FLASH

Fatal	Error	Warning
0	0	3

Tue Oct 06 11:41:00 Warning Severity >= 0 Selected/Total No. of Alarms: 3/3 Active Alarms Only

● Magnets 0 0 3	Feedbacks 0 0 0	System 0 0 0
Kly + Mod 0 0 0	Timing 0 0 0	Hardware 0 0 0
LLRF 0 0 0	BPMs 0 0 0	Services 0 0 0
Kickers 0 0 0	BLMs 0 0 0	Alarm Server 0 0 0
Vacuum 0 0 0	Collimators 0 0 0	Archive Server 0 0 0
BIS 0 0 0	Infrastructure 0 0 0	Radiation 0 0 0

System	Device Name	Message	Sev	Alarm Descriptor	Alarm Time	Duration
Magnets	RD13DUMPB	PS EIN FALSCH	8	Data Changed	11:41:29.017 - Oct 06 CEST	20.0 hr
Magnets	RD13DUMPA	PS EIN FALSCH	8	Data Changed	11:41:25.979 - Oct 06 CEST	20.0 hr
Magnets	RD13DUMPC	PS EIN FALSCH	8	Data Changed	11:41:21.937 - Oct 06 CEST	20.0 hr

11:40:59: Alarms loaded.

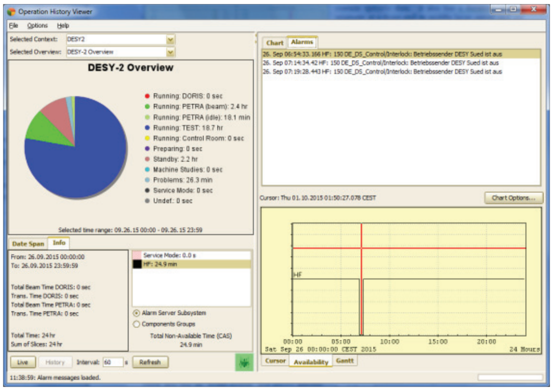
Figure 2: Typical Alarm Viewer display.

Upon start-up, the TINE Studio Alarm Viewer displays a grid-like view of the available alarm systems, showing the number of active *fatal*, *error*, and *warning* level alarms in each, as well as *regional* information. Should an alarm appear in any subsystem, a list of active alarms can be exposed by simply clicking on the alarm subsystem. Specific alarm information can then be obtained by clicking on an alarm in the alarm list (see Figure 2 above).

Operators have the ability at any time to disable an active alarm (or re-enable an alarm), as long as they provide a user name and reason in the *disable alarm* dialog.

Operation Statistics

The TINE Studio Operation History Viewer makes use of the archive and alarm systems as well as the TINE State system. If the machine states are suitably defined and declared, then the amount of time spent in any one machine state can be archived. Likewise the *subsystem-is-available* state (based on there being no fatal alarms in the subsystem) can also be archived. Together this information is used to feed the Operations History Viewer. Here the operators and machine coordinators can easily see how much time a machine has spent in any particular state over any selected time range. Likewise, the availability can easily be examined. In particular, if the machine was not available at some time, the *reasons* can be gleaned by examining the fatal alarm list during the time the machine was not available. In Figure 3 we see that the 24 minutes of non-availability were due to problems in the HF system.



Selected Context: DESY2
Selected Overview: DESY-2 Overview

● Running: DORIS: 0 sec
● Running: PETRA (beam): 2.4 hr
● Running: PETRA (idle): 18.1 min
● Running: TEST: 18.7 hr
● Running: Control Room: 0 sec
● Preparing: 0 sec
● Standby: 2.2 hr
● Machine Studies: 0 sec
● Problems: 26.3 min
● Service Mode: 0 sec
● Undef.: 0 sec

Selected time range: 09.26.15 00:00 - 09.26.15 23:59

26. Sep 06:54:33.166 HF: 150 DE_DS_Control/Interlock: Betriebssender DESY Sued ist aus
26. Sep 07:14:34.42 HF: 150 DE_DS_Control/Interlock: Betriebssender DESY Sued ist aus
26. Sep 07:19:28.443 HF: 150 DE_DS_Control/Interlock: Betriebssender DESY Sued ist aus

Cursor: Thu 01.10.2015 01:50:27.078 CEST

From: 26.09.2015 00:00:00
To: 26.09.2015 23:59:59

Total Beam Time DORIS: 0 sec
Trans. Time DORIS: 0 sec
Total Beam Time PETRA: 0 sec
Trans. Time PETRA: 0 sec

Total Time: 24 hr
Sum of Slices: 24 hr

Service Mode: 0.0 s
■ HF: 24.9 min

◉ Alarm Server Subsystem
○ Components Groups

Total Non-Available Time (CAS)
24.9 min

Interval: 60 s

11:38:59: Alarm messages loaded.

Figure 3: Typical Operation History display.

Generic Viewers

TINE Studio offers several other generic viewers, which have proved their value over the years.

Although the Archive Viewer itself can be switched to *live mode*, a trend-chart viewer, where the live trends of multiple machine parameters can be followed, is also part of the package.

Many relevant control system elements are available as *multi-channel arrays*. These are parameters representing a vector of instances of *the same thing*, with the same units and display settings, such as BPM or BLM values, vacuum pressure, etc. Such control system elements can be easily viewed, browsed, and investigated with the

ISBN 978-3-95450-148-9

TINE Studio Multi-Channel Analyser (MCA). Here both live and archive data can be correlated. Figure 4 shows the MCA viewer in history mode.

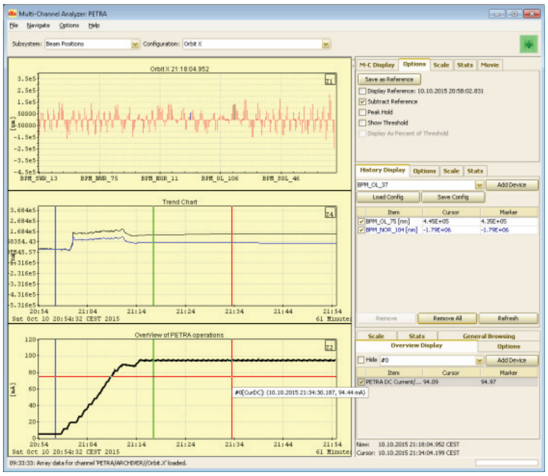

Multi-Channel Analyzer: PETRA
File Navigate Options Help
Subsystem: Beam Positions
Configuration: Orbit X
Orbit X 21:18:04.952
Trend Chart
Overview of PETRA operations
History Display
Add Device
Remove All
Refresh
Sat Oct 10 20:54:32 CEST 2015

Figure 4: An MCA display. Here: the PETRA Vacuum.

A machine control system typically features a number of servers providing oscilloscope like traces or waveforms. These are best viewed with the TINE Studio Scope Trace Viewer. The user can easily compare live data with data retrieved from either the archive system or the event archive system. See Figure 5 below.

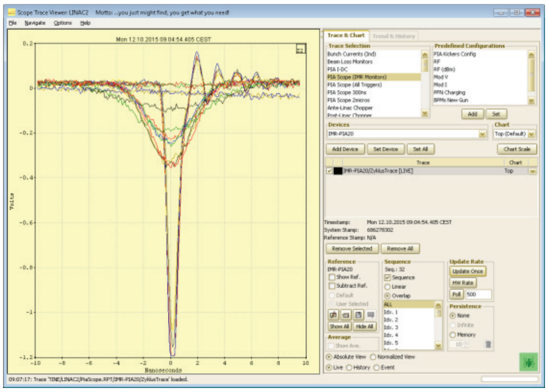


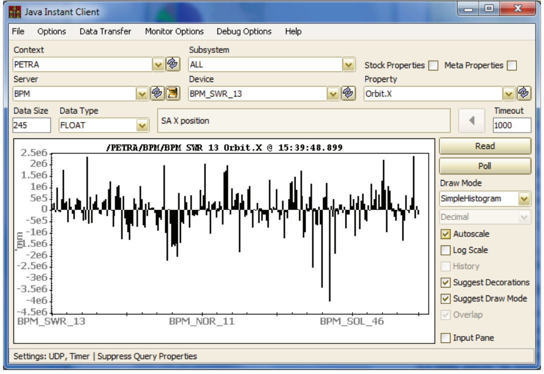
Figure 5: TINE Studio Scope Trace Viewer.

The most generic of all viewers is perhaps the TINE Instant Client. This one tool allows browsing the entire control system of all elements in the entire TINE site. Virtually all manner of control elements can also be viewed with the Instant Client (Figure 6 below), from structured data to video.

ADMINISTRATION

Database Management Utilities

Most TINE central servers make use of a local database to provide their data access instructions. This is true of the archive server, the alarm server, the event server, and the state server. In each of these cases, TINE Studio provides a database management utility, which checks consistency and simplifies the decision making process an administrator faces when he makes editorial changes.



Context: PETRA
Subsystem: ALL
Stock Properties □ Meta Properties □
Server: BPM
Device: BPM_SWR_13
Property: Orbit.X
Data Size: 245
Data Type: FLOAT
SA X position
Timeout: 1000
Draw Mode: SimpleHistogram
Decimal
☑ Autoscale
□ Log Scale
□ History
☑ Suggest Decorations
☑ Suggest Draw Mode
☑ Overlap
□ Input Pane

Settings: UDP, Timer | Suppress Query Properties

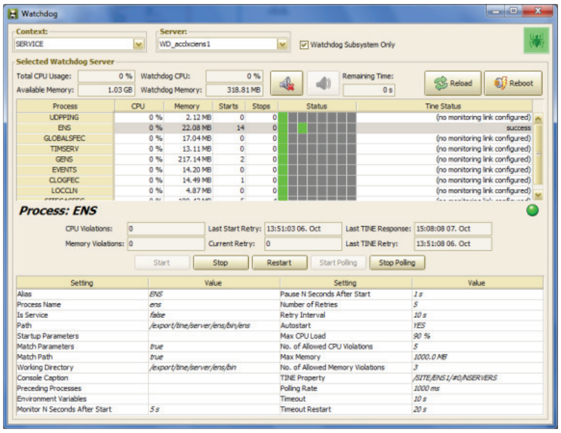
Figure 6: The TINE Instant Client.

Naming Services Utilities

TINE is a plug-and-play system. A server will automatically publish its registered information to the TINE equipment name server (ENS) as well as the TINE group equipment name server (GENS), if the server is registered as a member of a server group (a class of servers with similar devices). From time to time an administrator might wish to update the plug-and-play database maintained by the ENS. Indeed, only an administrator is allowed to assign 'importance' to a control system server. This is best accomplished by the ENS Administration Utility.

System Monitoring

It is generally the task of the control system administrator to monitor the integrity of the control system elements at large. Concerning servers, this task is itself fulfilled to zeroth order by the TINE Watchdog, which is itself a TINE server, responsible for starting, monitoring, and re-starting if necessary all other servers on the same host computer. The watchdog control GUI, shown in Figure 7, provides an interface to all watchdog servers.



Context: SERVICE
Server: WD_acclxciens1
☑ Watchdog Subsystem Only

Selected Watchdog Server
Total CPU Usage: 0 %
Watchdog CPU: 0 %
Available Memory: 1.03 GB
Watchdog Memory: 318.81 MB
Remaining Time: 0 s

Process	CPU	Memory	Starts	Stops	Status	Tine Status
UDPPING	0 %	2.12 MB	0	0		(no monitoring link configured)
ENS	0 %	22.08 MB	14	0		success
GLOBALSFEC	0 %	17.04 MB	0	0		(no monitoring link configured)
TIMSERV	0 %	13.11 MB	0	0		(no monitoring link configured)
GENS	0 %	217.14 MB	2	0		(no monitoring link configured)
EVENTS	0 %	14.20 MB	0	0		(no monitoring link configured)
CLOGFEC	0 %	14.49 MB	1	0		(no monitoring link configured)
LOCCLN	0 %	4.87 MB	0	0		(no monitoring link configured)

Process: ENS
CPU Violations: 0
Last Start Retry: 13:51:03 06. Oct
Last TINE Response: 15:08:08 07. Oct
Memory Violations: 0
Current Retry: 0
Last TINE Retry: 13:51:08 06. Oct

Setting	Value	Setting	Value
Alias	ENS	Pause N Seconds After Start	1 s
Process Name	ens	Number of Retries	5
Is Service	false	Retry Interval	10 s
Path	/export/tine/server/ens/bin/ens	Autostart	YES
Startup Parameters		Max CPU Load	90 %
Match Parameters	true	No. of Allowed CPU Violations	5
Match Path	true	Max Memory	1000.0 MB
Working Directory	/export/tine/server/ens/bin	No. of Allowed Memory Violations	3
Console Caption		TINE Property	/SITE/ENS1/#0/NSERVERS
Preceding Processes		Polling Rate	1000 ms
Environment Variables		Timeout	10 s
Monitor N Seconds After Start	5 s	Timeout Restart	20 s

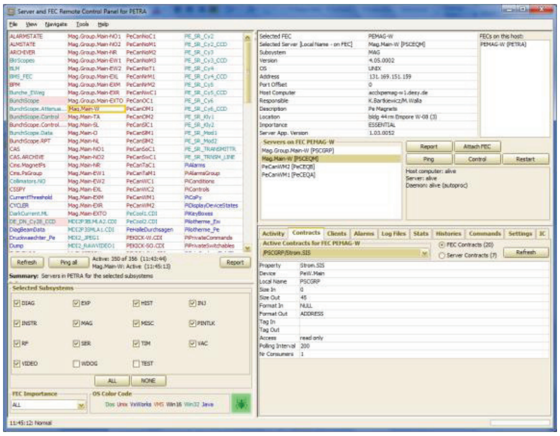
Figure 7: Watchdog control GUI.

Of major benefit to the administrator is the FEC (Front End Controller) Remote Panel. This single panel displays the running status of all servers in a particular TINE

ISBN 978-3-95450-148-9

context. Log files, alarm and other meta- information, can trivially be obtained for any selected server.

A report of the characteristics (version number, operating system, daemon statistics, ping results, etc.) of all servers in the selected context can be generated with a single button click. Figure 8 shows a typical view of the FEC remote control panel.


Server and FEC Remote Control Panel for PETRA
File View Navigate Tools Help
Refresh
Ping all
Report
Attach FEC
Ping
Control
Restart
Activity Contracts Clients Alarms Log Files Stats Histories Commands Settings IC
ALL
NONE
FEC Importance
OS Color Code
Dos Unix VxWorks VMS Win16 Win32 Java

ALARMSTATE	Mag.Group.Main-NO1	PeCanNoC1	PE_SR_Cy2
ALMSTATE	Mag.Group.Main-NO2	PeCanNoM1	PE_SR_Cy2_CCD
ARCHIVER	Mag.Group.Main-NR	PeCanNoM2	PE_SR_Cy3
BloScopes	Mag.Group.Main-EW1	PeCanNoM3	PE_SR_Cy3_CCD
BLM	Mag.Group.Main-EW2	PeCanNoT1	PE_SR_Cy4
BMS_FEC	Mag.Group.Main-EXL	PeCanNrM1	PE_SR_Cy4_CCD
BPM	Mag.Group.Main-EXM	PeCanNrM2	PE_SR_Cy5
Bunche_EWeg	Mag.Group.Main-EXR	PeCanNwC1	PE_SR_Cy5_CCD
BunchScope	Mag.Group.Main-EXTO	PeCanOC1	PE_SR_Cy6
BunchScope.Attenua...	Mag.Main-W	PeCanOM1	PE_SR_Cy6_CCD
BunchScope.Control	Mag.Main-TA	PeCanOM2	PE_SR_Kly1
BunchScope.Control....	Mag.Main-SL	PeCanSlC1	PE_SR_Kly2
BunchScope.Data	Mag.Main-OI	PeCanSlM1	PE_SR_Mod1
BunchScope.RPT	Mag.Main-NL	PeCanSlM2	PE_SR_Mod2
CAS	Mag.Main-NO1	PeCanSoC1	PE_SR_TRANSMITTR
CAS.ARCHIVE	Mag.Main-NO2	PeCanSwC1	PE_SR_TRNSM_LINE
Cms.MagnetPs	Mag.Main-NR	PeCanTaC1	PiAlarms
Cms.PsGroup	Mag.Main-EW1	PeCanTaM1	PiAlarmsGroup
Collimators.NO	Mag.Main-EW2	PeCanWC1	PiConditions
CSSPY	Mag.Main-EXL	PeCanWC2	PiControls
CurrentThreshold	Mag.Main-EXM	PeCanWM1	PiCoPy
CYCLER	Mag.Main-EXR	PeCanWM2	PiDisplayDeviceStates
DarkCurrent.ML	Mag.Main-EXTO	PeCool1.CDI	PiKeyBoxes
DE_DN_Cy28_CCD	MDI2P38MLA2.CDI	PeCool2.CDI	Pilotherme_Ew
DiagBeamData	MDI2P39MLA1.CDI	PeHalleDurchsagen	Pilotherme_Pe
Druckwaechter_Pe	MDI2_JPEG1	PEKICK-W.CDI	PiPrivateCommands
Dump	MDI2_RAWVIDEO1	PEKICK-SO.CDI	PiPrivateSwitchables

Active: 350 of 356 (11:43:44)
Mag.Main-W: Active (11:45:13)

Summary: Servers in PETRA for the selected subsystems

Selected Subsystems

☑ DIAG ☑ EXP ☑ HIST ☑ INJ
☑ INSTR ☑ MAG ☑ MISC ☑ PINTLK
☑ RF ☑ SER ☑ TIM ☑ VAC
☑ VIDEO □ WDOG □ TEST

FEC Importance: ALL

Selected FEC	PEMAG-W
Selected Server [Local Name - on FEC]	Mag.Main-W [PSCEQM]
Subsystem	MAG
Version	4.05.0002
OS	UNIX
Address	131.169.151.159
Port Offset	0
Host Computer	acckpemag-w1.desy.de
Responsible	K.Bartkiewicz/M.Walla
Description	Pe Magnets
Location	bldg 44 rm Empore W-08 (3)
Importance	ESSENTIAL
Server App. Version	1.03.0052

FECs on this host:
PEMAG-W (PETRA)

Servers on FEC PEMAG-W
Mag.Group.Main-W [PSCGRP]
Mag.Main-W [PSCEQM]
PeCanWM2 [PeCEQB]
PeCanWM1 [PeCEQA]

Host computer: alive
Server: alive
Daemon: alive (autoproc)

Active Contracts for FEC PEMAG-W
/PSCGRP/Strom.SIS
● FEC Contracts (20)
○ Server Contracts (7)

Property	Strom.SIS
Device	PeW.Main
Local Name	PSCGRP
Size In	0
Size Out	45
Format In	NULL
Format Out	ADDRESS
Tag In	
Tag Out	
Access	read only
Polling Interval	200
Nr Consumers	1

11:45:12: Normal

Figure 8: TINE Studio FEC Remote Control Panel.

DEVELOPMENT

Good development tools which go beyond a C or Java API are a godsend. But here we should distinguish between *server* development, where display is not an issue and *client* development, where it generally is.

Server Development

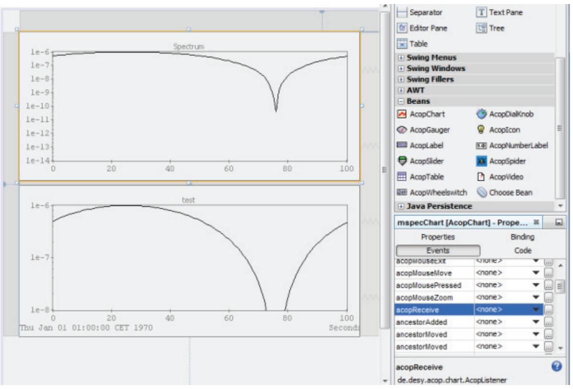
At the hardware level a TINE server can under many circumstances simply be configured as a CDI (Common Device Interface [3]) server. If this is the case then the TINE Studio CDI editor can be used to configure (and deploy and maintain) the necessary CDI database.

Often, a front end server or middle layer server will need to apply additional logic to obtain the finished data sets published by the server. Here the use of the TINE server wizard can be used to generate the configuration files and/or code in selected languages. The developer can then concentrate on making the relevant changes to the generated code.

Rapid Application Development Tools

At the client side GUI (Graphical User Interface) applications are of primary concern. Here we should also distinguish between rich clients, where programming logic (often in a high level language) enhances computation and display, and simple clients, where widgets on a panel are connected to control points but have limited interactions among themselves.

To be sure, TINE offers interfaces to high level scripting languages such as Python and MatLab, which are popular with machine physicists and are necessarily *rich*. On the widget side of the coin, there are also TINE plugs to popular panel builders such as CSS or jDDD [4]. TINE Studio itself, however, does not offer a panel builder per se. Instead GUI developers are encouraged to make use of ACOP (Advanced Component Oriented Programming [5]) components, which can be used in a number of frameworks and languages with minimal (and sometimes no) coding. For instance, *acopbeans* can easily be incorporated in java projects in either eclipse or NetBeans (see Figure 9 below). And ACOP.NET, the successor to ACOP ActiveX, can easily be added to Visual Studio. In such cases an application can become a rich client when necessary but otherwise remain a simple application created by dropping graphical widgets on a panel in a graphical development environment (and connecting them to control system endpoints).



acopMouseExit	<none>
acopMouseMove	<none>
acopMousePressed	<none>
acopMouseZoom	<none>
acopReceive	<none>
ancestorAdded	<none>
ancestorMoved	<none>
ancestorMoved	<none>

acopReceive
de.desy.acop.chart.AcopListener

Figure 9: Using ACOP in a NetBeans project.

And in the case of the COMA (Container Object MAnager [6]) class in Java, simple clients can even be created *on-the-fly* without invoking a framework.

CONCLUSION

A *studio* should strive to meet the expectations of all of categories of users, without restricting functionality or performance. This includes the developer along with the operator and the machine physicist and engineer as well as the administrator. The utilities presented to all categories of user should not only provide an intuitive interface but make available the information the user is most likely to need with a minimum of effort. After years of interviews, interactions, and feedback with precisely this set of users, TINE Studio is well on its way to realizing these goals.

REFERENCES

[1] CSS website: http://controlsystemstudio.org/
[2] TINE website: http://tine.desy.de.
[3] Duval, et al., "Common Device Interface 2.0", PCaPAC 2014, Karlsruhe, Germany, (2014) .
[4] jDDD website: http://jddd.desy.de.
[5] acopbeans website: http://public.cosylab.com/acop/site/
[6] Bartkiewicz, et al., "The Run-Time Customization of Java Rich Clients with the COMA Class", ICALEPCS 2007, Knoxville, USA, (2007).