DAΦNE TECHNICAL NOTE

NFN - LNF, Accelerator Division

Frascati, April 5, 2001

Note: **C-19**

DAFNE Web Server Data Access Facility

M. Masciarelli, G. Mazzitelli

Introduction

The DAFNE Web Server, http://dafne.lnf.infn.it, allows users to access raw machine elements data. The system permits to download and give access to useful files for off-line analysis. Moreover, some on-line CGI (Common Gateway Interface) present the status and give history plots of data. This note aims at giving the reader a brief introduction on how to work under the DAFNE Web Server data access facility by describing the file system and giving a reference to various formats used. The list of logged files, as well as their format, refers to the situation at the date of publication of this note.

History and Status Files

The DAFNE storing scheme has already been described [1]. The STORER process, running under the DAFNE Control System [2], writes a set of status and history files on the DAFNE Web server local disks:

- FAST.STAT is the DAFNE status file logged every 5 seconds (format in appendix C.1) and is used mostly to share information with machines not online in the DAFNE Control System;
- DAFNE.STAT is the DAFNE status file logged every 15 seconds (format in appendix C.1) equal to previous file contains the most important raw information of the devices logged.
- NEWDAFNE.STAT is the DAFNE status file logged every 15 seconds (format in appendix C.2). Data in raw format are used to create the new status files where information are decoded, and merged with past status to evaluate LIFTIME, FILL number, machine STATUS flags.

All these status files are overwritten, while history are logged, see next chapter for detailed information.

The variables of the devices logged every 5 and 15 seconds with the C.1 format as plain text, are then logged every minute in a binary file with some more detailed information (appendix A and B). In the same daily history files information on slower device, like vacuum, magnet, scraper position, and so on, are stored every 5 minutes.

The NEWDAFNE.STAT file is used by the KLOE slow control process [3] to create the ".fast" file (format in appendix C.4) where machine information are merged with detector data. Each record is tagged by the time of storing, and all process are synchronized by the NTP (Network Time Protocol) server of LNF [4]. The time difference between each record can't be uniform, because the machine writing files are not real time.

Time format is the UNIX one, seconds since first of January 1960, for most of plain text DAFNE files, PC format, seconds since first of January 1904, for the binary DAFNE files, and seconds since the midnight of the filename for all the KLOE files.
Every day a process, running 30 minutes after midnight, produce DMCV (Daily Most Common Variable) files (format in appendix C.3) useful for vacuum study versus various machine parameters.
A process running on the DAFNE Server continuously checks data validity and produces status files used by on line CGI to present data.

Files System

The DAFNE Web Server shares data between the DAFNE Control System, KLOE and DEAR experiments.
All history data files are named YYYYMMDD.XXXX, where XXXX is the file extension:

.lv	binary LABVIEW history files
.fast	fast plain text history data
.slow	slow plain text history data
.raw	fast plain text un-decoded history data
.stat	fast plain text status files
.dat	generic plain text history files

The DAFNE file system has the following structure:

/u2/data/binary/	DAFNE data binary format
/u2/data/binlog/	logs information on binary data
/u2/data/dear/	DEAR history files
/u2/data/kloe/	not used
/u2/data/estimated/	estimated DAFNE luminosity
/u2/data/fast/	fast status file
/u2/data/history/daily/	daily history of integrated quantity
/u2/data/history/raw/	raw fast logged machine data
/u2/data/history/dat/	decoded fast logged machine data
/u2/data/monitor/caron/	DEVIL and CARON history files
/u2/data/monitor/caronlog/	DEVIL and CARON log history files
/u2/data/monitor/sysreset/	Control System reset log files
/u2/data/plainFiles/	plain text daily file of the most common variable
/u2/data/runs/status/	actual status runs information
/u2/data/runs/logs/	logs of runs information
/u2/data/slow/	history of slow elements most common variable
/u2/data/temp/	temporary files
/kloe/	KLOE history files

See appendix C for the data format.
The "…log" directory mainly contains information on parameters stored.
Another way to share data is using IP network without the need of mounting any file-system on the client machine.

On the DAFNE server there is a program which sends data contained in the "FAST.STAT" file over the IP network using UDP protocol. This is useful because with a very simple and small client-software written in C, or using a virtual instrument written in LABVIEW, one can gain access to the fast data.

Logs of the RUN & Control System Monitors

The monitor area, in the data logging file system, has been dedicated to store information on Control System performance and fault. The CARON task, which is mainly devoted to send commands from the user interface to the front-end device, is continously checking the health of all the Control System process running. The information are stored and data are presented on the DAFNE web server. Any Control System reset, the procedures followed, and any possible problem, can be found in the ../sysreset directory. All the information on the Control System performance, are logged and available for off-line statistics and analysis of the performance.
A special directory as been dedicated to logs information on the runs. A form page that operators can compile from the DAFNE Control System, as well as with a web form page, contains information on the runs during operation. The history of all runs parameters is also logged for post analysis. Such data are also used as a communication tool with the experiments in case of faults. The same procedure also permits to change the information reported on the DAFNE Web Server banner, refreshed every 10 seconds, as fast communication tools.

Running off-line tools

In the download http://dafne.lnf.infn.it/private/ area users have access to data, formats, and files handling tools. Most of them are plain text files, easy to handle with common analysis tools.

Example 1:

How to load DMCV (Daily Most Common Variable) plain text file with KaleidaGraph?
Download the files of interest on your preferred folder, open the file from KaleidaGraph program, with the specification: delimiter=TAB, number =1, line skip = 0, no title. Format the first column (UNIX time) to Float, decimals =0, then with Formula Entry add 2082844800 (c0 = c0 + 2082844800). Format again the first column to date as you like. The same procedure can be applied on all other plain text file C.1, C.2 format, where the delimiter is now a space instead of a TAB.

Example 2:

How to read information on orientation of the positron beam as function of a skew quads, e.g. QSKPL204?
Download the binary file from the DAFNE Web Server area http://dafne.lnf.infn.it/private/, compile the routine reported in appendix C (present also in the download area, as well as the LABVIEW cluster type), get the right low level element name for the SPIRICON in appendix B (SPRP*001) and for the skew quad (QSKPL204). Choose the variable: 3 for the beam orientation, 1 for the current set of the skew quad, as reported in the appendix A.

Run the routine:
readRecord <filename> <element1> <element2> <varPosition1> <varPosition2>
./readRecord ./20010306.lv SPRP*001 QSKPL204 3 1
the output will be like this:

```
983919624 -4.200000e+00 0.000000e+00
983919684 3.100000e+00 5.000000e+00
983919744 7.200000e+00 11.000000e+00
983919804 5.900000e+00 8.300000e+00
983919865 5.600000e+00 8.000000e+00
983919925 7.500000e+00 12.000000e+00
```

In the first colon there is the UNIX time, that can be handled by the C routine, the second one is the beam orientation an the third one is the skew quad current set.

Expert access to history data via WWW

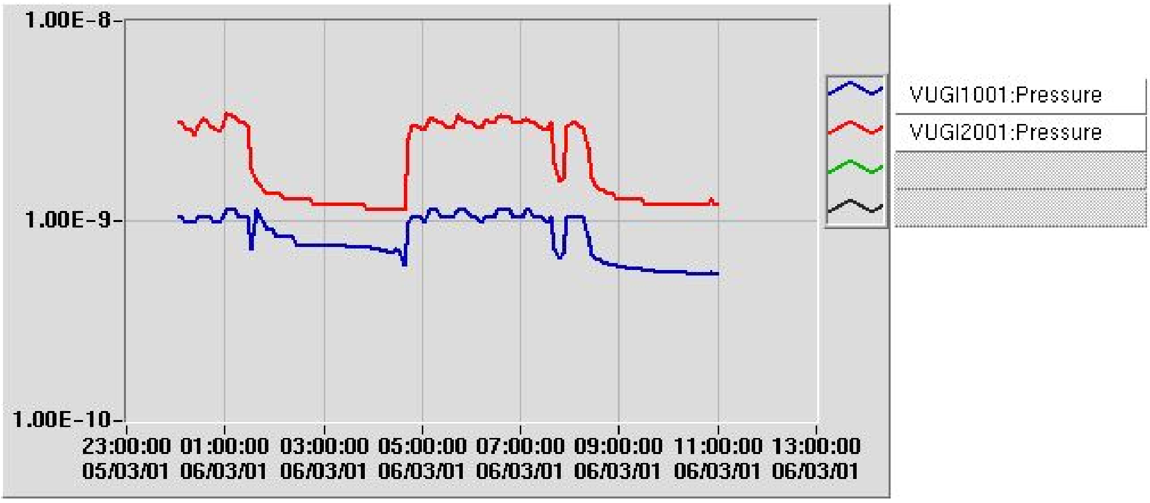
Various CGI are available in the DAFNE web site to present machine and experimental data. Some information are also presented as history plots. The most common plots and tables have been already presented [1]. A more powerful tool has been implemented to allows easy data correlation.
The tool is based on the DAFNE plot server, were some LABVIEW virtual instruments can be called to produce plots accessible via Web. In particular the "historyElements.vi" application allows to produce history of all the machine elements stored, is a specialist tool that requires the knowledge of the Controls System elements formats and name logged, reported in appendix A and B.

Usage, and query parameters:

CGI name:	historyPlot
plot:	last, YYYYMMDD.lv, if last is selected specifies the last day history otherwise the day specified is used.
VI:	name of the LABVIEW server virtual instrument to call.
elem:	name of the low level elements for which the history is required, more than one element can be specified separated by a comma.
variable:	variable to be plotted, 0 means the first variable of each elements, 7 the last one, the relative position is reported in appendix A.
plotType:	0 linear Y scale, 1 logaritmic Y scale
timeWin:	time window for the history plots, default is 0,24 if not specified.
range:	minimum and maximum of the Y scale, default is auto scale if not specified

Example of history plot query

http://dafne.lnf.infn.it/cgi-bin/historyPlot?plot=20010306.lv&VI=historyElement.vi&elem=VUGI1001&variable=0&plotType=1&timeWin=0,24&range=5e-10,5e-9



Example of mutiplot history query

http://dafne.lnf.infn.it/cgi-bin/historyPlot?plot=20010306.lv&VI=historyElement.vi&elem=VUGI1001,VUGI2001&variable=0&plotType=1&timeWin=0,24

Acknowledgment

We are very grateful to P. Valente F. Murtas and M. Bragardireanu who provided the formats for KLOE and DEAR experiments. We also thank M. Serio for the critical review of this paper.

References

[1] G. Di Pirro, G. Mazzitelli, A. Stecchi. "Data Handling tools at DAFNE" EPAC2000, 26-30 June 2000. Vienna, Austria .

[2] G. Di Pirro, G. Mazzitelli, C. Milardi, F. Sannibale, I. Sfiligoi, A.Stecchi, A. Stella. "The Evolution and Status of the DAFNE Control System". PCaPAC, 9-12 October 2000. DESY, Hamburg, German.

[3] F. Murtas, P. Valente. "The KLOE Slow Control System". KLOE memo 206.

[4] http://www.lnf.infn.it/computing/Unix/ntp/

Appendix A

DAFNE variable logged for each Control System low level Class.
ClassName:var1:var2:var3:var4:var5:var6:var7
The storing process is able to log information on 8 variable for each raw device. The class name is a DAFNE Control System flag that identifies the software controlling the device. var1 var7 are the name of the variable logged.

Magnet:
MG1:CurrSett:CurrROut:Polarity:Volt:Fault1:Fault2:Status:
MGO:CurrSett:CurrROut:Polarity:Volt:Fault1:-:Status:
CIT:-:-:-:-:Fault1:-:Status:
CHV:CurrSett:CurrROut:-:Volt:Fault1:-:Status:
COR:-:-:-:-:Fault1:-:Status:
CHN:CurrSett:CurrROut:-:Volt:Fault1:-:Status:

DC current monitor:
DCT:<I>:dI/shot:tau:Sum<I>*T:-:-:Status:

Vacuum:
VUG:Pressure:-:-:-:-:-:Status:

Luminosity monitor:
DLM:Lum:Lum*s:buckN:buckM:pedestal:SBCrSec:Status:

Timing information:
4TI:bun1-32:bun33-64:bun65-96:bun97-120:TimStatus:ACPulse:Status:

Estimated luminosity:
4LM:DLM1L0:DLM1p:DLM1e:DLM2L0:DLM2p:DLM2e:tick:

RF Frequency:
RDS:Freq:-:-:-:-:-:Status:

Ion cleaning electrodes:
ICE:HVset:HVread:-:-:-:-:Status:

Stepper motors (scrapers):
STP:set:read:-:-:-:-:Status:

Spiricon (Sincrotron light monitor):
SPR:WID:wid:orint:round:-:-:Status

KLOE Magnet field.
GSS:field:-:-:-:-:-:Status

Appendix B

DAFNE .lv binary file elements and variables. Updated every minute, or 5 minutes, in appendix A the variable logged for each element are reported.

List of elements logged each minute:

DCTEL001
DCTPS001
DLM00001
DLM00002
4TI00001
4LM00001
RDS42001
SPRE*001
SPRP*001
GSSI1001

List of elements logged each 5 minute:

VUGTM001	ICEES105	CVVPL102	DHRPS201	CHHES101	SXPES201	QUAES204
VUGTS001	ICEES106	CHHPL102	CDVES101	CVVES101	SXPES204	QUAES106
VUGTT001	ICEES107	CVVPL206	CDVES201	QSKPS101	SXPEL201	QUAES205
VUGTR001	ICEES108	CVVES201	CDHES201	CHHPS101	SXPEL204	QUAES206
VUGTL001	ICEES201	CHHPL206	CDHES101	CVVPS101	SXPEL101	QUAES209
VUGA3001	ICEES202	CHHPL205	CDVEL201	QSKPS104	SXPEL104	QUAES208
VUGA2001	ICEES203	CVVES106	CVVEI201	CHHPS104	SXPES101	QUAES207
VUGA1001	ICEES204	CHHES106	CVVEI102	CVVPS104	SXPES102	QUAES203
VUGTE001	ICEES205	CVVPL205	CHHEI201	QSKPS202	SXPES103	QUAES109
VUGTE002	ICEES206	CHHES201	CHHEI102	CHHPS202	SXPES202	QUAES110
VUGTP001	ICEES207	DHSES101	CVVEI202	CVVPS202	SXPES203	QUAES201
VUGPL101	ICEES208	DHRES102	CVVEI101	QSKPS205	SXPEL202	QUAES202
VUGPL102	ICEES209	DHREL101	CHHEI202	CHHPS205	SXPEL203	QUAES107
VUGPL103	ICEES210	DHSEL102	CHHEI101	CVVPS205	SXPEL103	QUAES108
VUGPS101	ICEI2001	DHRES201	CDHEL201	QSKPL201	SXPEL102	QUAES103
VUGPS102	ICEI2002	DHSES202	CDVEL101	CHHPL201	SXPPS101	QUAPS203
VUGPS103	ICEI2003	DHSEL201	CDHEL101	CVVPL201	SXPPS102	QUAPS202
VUGPS104	ICEI2004	DHREL202	CDHPS201	QSKPL204	SXPPS103	QUAPS207
VUGPS105	ICEEL201	CVVPS106	CDHPS101	CHHPL204	SXPPL104	QUAPS201
VUGPS201	ICEEL202	CHHPS106	CDVPS101	CVVPL204	SXPPL101	QUAPS110
VUGPS202	ICEEL203	CVVPS105	CDVPS201	QSKPL103	SXPPL204	QUAPS109
VUGPS203	ICEEL204	CHHPS105	CDHPL201	CHHPL103	SXPPL201	QUAPS103
VUGPL201	ICEEL205	CVVEL101	CHHPI101	CVVPL103	SXPPS204	QUAPS108
VUGPL202	ICEEL206	CHHEL101	CVVPI101	QSKPL106	SXPPS201	QUAPS107
VUGPL203	ICEEL207	CVVEL102	CVVPI202	CHHPL106	SXPPS104	QUAPS209
VUGEL101	ICEEL208	CHHEL102	CVVPI201	CVVPL106	SXPPL103	QUAPS208
VUGEL102	ICEEL101	CVVEL202	CHHPI201	DH*E*001	SXPPL102	QUAPS206
VUGEL103	ICEEL102	CHHEL202	CHHPI202	WGLE*001	SXPPL203	QUAPS205
VUGES101	ICEEL103	CVVEL203	CHHPI102	WGLES201	SXPPL202	QUAPS106
VUGES102	ICEEL104	CHHEL203	CVVPI102	WGLEL201	SXPPS202	QUAPS105
VUGES103	ICEEL105	CVVPS204	CDVPL201	WGLEL101	SXPPS203	QUAPS104
VUGES201	ICEEL106	CHHPS204	CDHPL101	WGLES101	QUAEL204	QUAPS102
VUGES202	ICEEL107	CVVPS203	CDVPL101	DH*P*001	QUAEL205	QUAPS101
VUGES203	ICEEL108	CHHPS203	QSKEL106	WGLP*001	QUAEL202	QUAPS204
VUGEL201	ICEEL109	CHHEL205	CHHEL106	WGLPS201	QUAEL203	QUAPL209
VUGEL202	CVVEL104	CVVEL205	CVVEL106	WGLPL201	QUAEL110	QUAPL210
VUGEL203	CHHEL104	CVVEL206	QSKEL103	WGLPL101	QUAEL108	QUAPL102
VUGI1001	CVVEL105	CHHEL206	CHHEL103	WGLPS101	QUAEL109	QUAPL101
VUGI2001	CHHEL105	CVVPS201	CVVEL103	SPLP2001	QUAEL103	QUAPL103
STPEL2U1	CVVPS102	CHHPS201	QSKEL204	SPLP1002	QUAEL104	QUAPL104
STPEL2D1	CHHPS102	CVVPL203	CHHEL204	SPLP2002	QUAEL101	QUAPL108
STPPS2I1	CVVPS103	CHHPL203	CVVEL204	SPLP1001	QUAEL102	QUAPL109
STPPS2E1	CHHPS103	CVVPL202	QSKEL201	SPLE2001	QUAEL209	QUAPL110
STPPL2U1	CVVES102	CHHPL202	CHHEL201	SPLE1002	QUAEL210	QUAPL207
STPPL2D1	CHHES102	CVVES204	CVVEL201	SPLE2002	QUAEL208	QUAPL208
STPES2I1	CVVES103	CHHES204	QSKES205	SPLE1001	QUAEL207	QUAPL203
STPES2E1	CHHES103	CVVES203	CHHES205	QUATE104	QUAEL201	QUAPL202
STPPL1I1	CVVPL105	CHHES203	CVVES205	QUATE105	QUAEL106	QUAPL201
STPPL1E1	CHHPL105	DHRPS102	QSKES202	QUAI2001	QUAEL107	QUAPL205
STPEL1I1	CVVPL104	DHSPS101	CHHES202	QUAI2002	QUAEL105	QUAPL204
STPEL1E1	CHHPL104	DHSPL102	CVVES202	QUAI2003	QUAEL206	QUAPL206
ICEES101	CVVPL101	DHRPL101	QSKES104	QUAI2005	QUAES104	QUAPL105
ICEES102	CHHPL101	DHRPL202	CHHES104	QUAI2006	QUAES105	QUAPL107
ICEES103	CVVES105	DHSPL201	CVVES104	QUAI2007	QUAES101	QUAPL106
ICEES104	CHHES105	DHSPS202	QSKES101	SXPES104	QUAES102	

Appendix C

C.1 DAFNE .raw format. Updated every 15 seconds

1.	time	[UNIX seconds]
2.	e- current	[mA]
3.	e+ current	[mA]
4.	IP1 luminosity	[cm-2s-1]
5.	IP1 lum. Int.	[nbarn-1]
6.	IP2 luminosity	[cm-2s-1]
7.	IP2 lum. Int.	[nbarn-1]
8.	bunch 1-32	word*
9.	bunch 33-64	word
10.	bunch 65-96	word
11.	bunch 97-120	word
12.	timing word	word
13.	Acc. pulse	#
14.	L0 IP1	[cm-2s-1]
15.	I0 e+ IP1 cur	[mA]
16.	I0 e- IP1 cur	[mA]
17.	L0 IP2	[cm-2s-1]
18.	I0 e+ IP2 cur	[mA]
19.	I0 e- IP2 cur	[mA]
20.	ms from start RUN	[msec]
21.	RF frequency	[Hz]
22.	Roundness e-	[σy/σx]
23.	Roundness e+	[σy/σx]
24.	KLOE field	[Gauss]

*) the bunch word in this format as well as in the KLOE fast format is 1 if the bunch is filled. The most significant bit is the first bunch.

C.2 DAFNE .dat format. Updated every 15 seconds

1.	time	[UNIX seconds]
2.	e- current	[mA]
3.	e+ current	[mA]
4.	Lum1 (IP1 or e+) rate	[Hz]
5.	Lum2 (IP2 or e-) rate	[Hz]
6.	linac mode	-1 electrons +1 positrons
7.	number of e- bunches	#
8.	e- bunch 1-32 word	bit 32-1
9.	e- bunch 33-64 word	bit 32-1
10.	e- bunch 65-96 word	bit 32-1
11.	e- bunch 96-120 word	bit 32-8
12.	number of e+ bunches	#
13.	e+ bunch 1-32 word	bit 32-1
14.	e+ bunch 33-64 word	bit 32-1
15.	e+ bunch 65-96 word	bit 32-1
16.	e+ bunch 96-120 word	bit 32-8
17.	status e-	0 no beam 1 acc inject 2 main ring inject 3 stored beam 4 colliding
18.	status e+	0 no beam 1 acc inject 2 main ring inject 3 stored beam 4 colliding
19.	status DAFNE	-3 simulated data -2 run off

		-1 unknown
		0 standby
		1 e- inject
		2 e+ inject
		3 e- stored
		4 e+ stored
		5 filled
		6 colliding
20.	fill number	#
21.	e- lifetime	[s]
		-1 not available
		0 unstable
22.	e+ lifetime	[s]
		-1 not available
		0 unstable
23.	Lum1 (IP1or e+)	[cm-2 s-1]/1e28
24.	Lum2 (IP1or e-)	[cm-2 s-1]/1e28
25.	interaction flag	0 not colliding
		1 colliding @ IP1
		2 colliding @ IP2
		3 colliding @ IP1-IP2
26.	RF frequency	[Hz]
27.	Roundness e-	[σy/σx]
28.	Roundness e+	[σy/σx]
29.	KLOE field	[Gauss]

C.3 DAFNE DMCV, daily most common variable, files format

1.	time	[UNIX seconds]		60 seconds
2.	e- current	[mA]	DCTEL001	60 seconds
3.	e+ current	[mA]	DCTPS001	60 seconds
4.	IR1 luminosity e+	[cm-2 s-1]	DLM00001	60 seconds
5.	IR1 rate e+	[Hz]	DLM00001	60 seconds
6.	IR1 luminosity e-	[cm-2 s-1]	DLM00002	60 seconds
7.	IR1 rate e-	[Hz]	DLM00002	60 seconds
8.	e- number of bunch	#		60 seconds
9.	e+ number of bunch	#		60 seconds
10.	MR vacuum IP1	[torr]	VUGI1001	300 seconds
11.	MR vacuum IP2	[torr]	VUGI2001	300 seconds
12.	MR vacuum e+	[torr]	VUGPL101	300 seconds
13.	MR vacuum e+	[torr]	VUGPL102	300 seconds
14.	MR vacuum e+	[torr]	VUGPL103	300 seconds
15.	MR vacuum e+	[torr]	VUGPS101	300 seconds
16.	MR vacuum e+	[torr]	VUGPS102	300 seconds
17.	MR vacuum e+	[torr]	VUGPS103	300 seconds
18.	MR vacuum e+	[torr]	VUGPS104	300 seconds
19.	MR vacuum e+	[torr]	VUGPS105	300 seconds
20.	MR vacuum e+	[torr]	VUGPS201	300 seconds
21.	MR vacuum e+	[torr]	VUGPS202	300 seconds
22.	MR vacuum e+	[torr]	VUGPS203	300 seconds
23.	MR vacuum e+	[torr]	VUGPL201	300 seconds
24.	MR vacuum e+	[torr]	VUGPL202	300 seconds
25.	MR vacuum e+	[torr]	VUGPL203	300 seconds
26.	MR vacuum e-	[torr]	VUGEL101	300 seconds
27.	MR vacuum e-	[torr]	VUGEL102	300 seconds
28.	MR vacuum e-	[torr]	VUGEL103	300 seconds
29.	MR vacuum e-	[torr]	VUGES101	300 seconds
30.	MR vacuum e-	[torr]	VUGES102	300 seconds
31.	MR vacuum e-	[torr]	VUGES103	300 seconds
32.	MR vacuum e-	[torr]	VUGES201	300 seconds
33.	MR vacuum e-	[torr]	VUGES202	300 seconds
34.	MR vacuum e-	[torr]	VUGES203	300 seconds
35.	MR vacuum e-	[torr]	VUGEL201	300 seconds
36.	MR vacuum e-	[torr]	VUGEL202	300 seconds
37.	MR vacuum e-	[torr]	VUGEL203	300 seconds

C.4 KLOE .fast format. Updated every 15-30 seconds
(F. Murtas, P. Valente)

1. seconds since midnight	[s]
2. current e-	[mA]
3. current e+	[mA]
4. luminosity monitor IP1 counts	[Hz]
5. luminosity monitor IP2 counts	[Hz]
6. number of bunch e-	#
7. number of bunch e+	#
8. fill number	#
9. DAFNE status	-3 simulated data
	-2 run off
	-1 unknown
	0 standby
	1 e- inject
	2 e+ inject
	3 e- stored
	4 e+ stored
	5 filled
	6 colliding
10. not used	
11. CAENET (packed)	BIT 0 HVEMC
	BIT 1 EMC
	BIT 2 HVDC
	BIT 3 DC
	BIT 4 DAQ
	BIT 5 DC CURRENT
12. trigger RUN number	#
13. ECM HV state	-2 = HV UNDEFINED
	-1 = WORNG READOUT
	0 = HV ON
	1 = ECA OFF
	2 = BAR OFF
	3 = ECB OFF
	4 = QCAL OFF
	5 = HV OFF
14. ECM LV state	-1 = WRONG READOUT
	0 = LV OK
	1 = PULSING
	2 = BAD THRESHOLDS
15. trigger luminosity	[/10^28 cm-2s-1]
16. trigger number of Bhabha	#
17. DC HV state	-1 = WRONG READOUT
	0 = HV ON
	1 = STANDBY
	2 = HV OFF
	3 = RUMPING UP
	4 = RUMING DOWN
18. DC LV state	-1 = WRONG READOUT
	0 = LV OK
	1 = PULSING
	2 = BAD THRESHOLD
	3 = BAD WIDTH
	4 = BAD DEAD TIME
19. lifetime e-	[s]
20. lifetime e+	[s]
21. daq crates state	-1 = WRONG READOUT
	0 = ALL CRATE ON
	1 = ONE OR SAME CRATE OFF
22. L3 run number	#
23. not used	

24.	L3 luminosity	[/10^28 cm-2s-1]
25.	KLOE run state	-1 NOT RUNNING 0 NOT READY 1 READY 2 PAUSED 3 RUNNING
26.	KLOE RUN number	#
27.	L3 number of Bhabha	#
28.	RUN type	-1 NOT READY 0 NORMAL 1-5 CALIBRATION
29.	RUN on disk	0 TRASH 1 DISK
30.	number of farms	#
31.	DC trigger level 1	[Hz]
32.	DC trigger level 2	[Hz]
33.	QCAL coincidence	[Hz]
34.	QCAL A	[Hz]
35.	QCAL B	[Hz]
36.	T2 yes	[Hz]
37.	T1 free	[Hz]
38.	QCAL coincidence delayed	[Hz]
39.	QCAL Bhabha delayed	[Hz]
40.	QCAL Bhabha	[Hz]
41.	TRG integrated luminosity	[nbarn-1]
42.	L3 integrated luminosity	[nbarn-1]
43.	pe1 e- bunch pattern	word
44.	pe2	word
45.	pe3	word
46.	pe4	word
47.	pp1 e+ bunch pattern	word
48.	pp2	word
49.	pp3	word
50.	pp4	word
51.	Run Size	[GByte]
52.	Event Size	[byte]
53.	ECM1	[Hz]
54.	ECM2	[Hz]
55.	ECM3	[Hz]
56.	ECM4	[Hz]

C.5 KLOE .slow format. Updated every 45 seconds

(F. Murtas, P. Valente)

1.	seconds from midnight	[s]
2.	vacuum at IP 1 (KLOE)	[torr]
3.	vacuum at IP 2 (DEAR/FINUDA)	[torr]
4.	magnet status	0 cold 1 undefined 2 300-77 K 3 77-4 K 4 warm map
5.	magnet current	[A]
6.	magnet Helium percentage in vessel	[%]
7.	magnet coil 1 temperature	[K]
8.	magnet coil 2 temperature	[K]
9.	magnet coil 3 temperature	[K]
10.	magnet coil 4 temperature	[K]

11.	gas status	0 working 1 warning 2 alarm -1 unknown
12.	gas mode	0 standby 1 closed mode 2 calibrate 3 shutdown 4 open mode (standard) 5 zero adjust 6 manual -1 unknown
13.	gas flow	[slm] (standard liters/minute)
14.	atmosferic pressure	[mbar]
15.	gas temperature	[Celsius]
16.	gas isobutane percentage	[%]
17.	gas oxygen content	[ppm]
18.	gas water content	[ppm]
19.	absolute pressure (side A)	[mbar]
20.	absolute pressure (side B)	[mbar]
21.	differential pressure (side A)	[mbar]
22.	differential pressure (side B)	[mbar]
23.	pressure of Helium inlet	[mbar]
24.	pressure of isobutan inlet	[mbar]
25.	pressure of argon inlet	[mbar]
26.	TRKMON run number	#
27.	beam position x	[cm]
28.	beam position y	[cm]
29.	beam position z	[cm]
30.	beam width x	[mm]
31.	beam width y	[mm]
32.	beam width z	[mm]
33.	Φ momentum x	[MeV/c]
34.	Φ momentum y	[MeV/c]
35.	Φ momentum z	[MeV/c]
36.	CALMON run number	#
37.	γγ endcap energy	[MeV]
38.	γγ barrel energy	[MeV]
39.	Bhabha endcap energy	[MeV]
40.	Bhabha barrel energy	[MeV]
41.	COSMON run number	#
42.	DCNOISE run number	#
43.	not used	
44.	scraper e- LONG IP2 up	[mm]
45.	scraper e- LONG IP2 down	[mm]
46.	small cells average voltage	[V]
47.	big cells average voltage	[V]
48.	number of tripped channels	#
49.	number of overcurrent channels	#
50.	number of channels off	#
51.	current of DC sector 1	[µA]
52.	current of DC sector 2	[µA]
53.	current of DC sector 3	[µA]
54.	current of DC sector 4	[µA]
55.	current of DC sector 5	[µA]
56.	current of DC sector 6	[µA]
57.	current of DC sector 7	[µA]
58.	current of DC sector 8	[µA]
59.	current of DC sector 9	[µA]
60.	current of DC sector 10	[µA]
61.	current of DC sector 11	[µA]
62.	current of DC sector 12	[µA]
63.	current of DC sector 13	[µA]

64.	current of DC sector 14	[µA]
65.	current of DC sector 15	[µA]
66.	current of DC sector 16	[µA]
67.	not used	
68.	acci. clusters >7 MeV WEST endcap	[Hz]
69.	acci. clusters >7 MeV EAST endcap	[Hz]
70.	accidental clusters >7 MeV barrel	[Hz]
71.	t-r/c endcap-endcap	[ns]
72.	t-r/c barrel-barrel	[ns]
73.	t-l/v endcap-endcap	[ns]
74.	t-l/v barrel-barrel	[ns]
75.	scraper e+ SHORT IP2 in	[mm]
76.	scraper e+ SHORT IP2 out	[mm]
77.	scraper e+ LONG IP2 up	[mm]
78.	scraper e+ LONG IP2 down	[mm]
79.	scraper e- SHORT IP2 in	[mm]
80.	scraper e- SHORT IP2 out	[mm]
81.	scraper e+ LONG IP1 in	[mm]
82.	scraper e+ LONG IP1 out	[mm]
83.	scraper e- LONG IP1 in	[mm]
84.	scraper e- LONG IP1 out	[mm]
85.	noise layer 1, sector 1	[Hz]
86.	noise layer 1, sector 2	[Hz]
87.	noise layer 1, sector 3	[Hz]
88.	noise layer 1, sector 4	[Hz]
89.	noise layer 5, sector 1	[Hz]
90.	noise layer 5, sector 2	[Hz]
91.	noise layer 5, sector 3	[Hz]
92.	noise layer 5, sector 4	[Hz]
93.	noise layer 10, sector 1	[Hz]
94.	noise layer 10, sector 2	[Hz]
95.	noise layer 10, sector 3	[Hz]
96.	noise layer 10, sector 4	[Hz]

C.6 *DEAR .dat format. Updated every 2 minutes when DEAR data acquisition is runnig*

(M. Bragardireanu)

1.	time	[Unix seconds]
2.	time	[hh:mm:ss]
3.	daily integrated luminosity	[nbarn-1]
4.	luminosity	[cm-2s-1]
5.	time between luminosity measure	[s]
6.	number of kaons since beginning of RUN	#
7.	elapsed run time	[s]
8.	integrated luminosity since the begin of the RUN	[nbarn-1]
9.	rate of coincidence vetoed by RF/4	[Hz]
10.	rate of coincidence	[Hz]
11.	rate in inner scintillator	[Hz]
12.	rate in outer scintillator	[Hz]
13.	rate first anti coincidence	[Hz]
14.	rate second anti coincidence	[Hz]
15.	rate third anti coincidence	[Hz]
16.	rate fourth anti coincidence	[Hz]
17.	rate of coincidence vetoed by anti coincidence no RF/4	[Hz]

C.7 *DAFNE estimated luminosity(IP1 is e+ lumimometer placed in the KLOE interaction region, IP2 is the e+ luminometer placed in DEAR region, but can be moved at IP1 as e- luminometer)*

1.	time	[Unix seconds]
2.	IP1 luminosity estimated	[cm-2 s-1]
3.	IP2 luminosity estimated	[cm-2 s-1]
4.	colliding flag	0 not colliding 1 colliding @ IP1 2 colliding @ IP2 3 colliding @ IP1 and IP2

C.8 *DAFNE slow elements plain files. The most useful variable of elements logged in binary files every 5 minutes are restored in plain text files.*

1.	time	[Unix seconds]
2.	IP1 vacuum	[torr]
3.	IP2 vacuum	[torr]
4.	Scraper EL201 up	[mm]
5.	Scraper EL201 down	[mm]
6.	Scraper PS201 inner	[mm]
7.	Scraper PS201 outer	[mm]
8.	Scraper PL201 up	[mm]
9.	Scraper PL201 down	[mm]
10.	Scraper ES201 inner	[mm]
11.	Scraper ES201 outer	[mm]
12.	Scraper PL101 inner	[mm]
13.	Scraper PL101 outer	[mm]
14.	Scraper EL201 inner	[mm]
15.	Scraper EL201 outer	[mm]

Apendix D

Example of a C routine to handle the binary files:

```
#include <stdio.h>
#include <time.h>

typedef struct {

  char         elementName[8];
  int          classID;
  char         fault;
  int          dataValueSize;
  double       dataValue[7];

} LVelementValue ;

typedef struct {

  double       LVtime;
  int          recordNumber;
  double       formatVertion;
  int          elementArraysSize;

} LVheader ;

typedef struct {

  LVheader header;
  LVelementValue value[400];

} LVrecord ;

int readValue(FILE *, char *,char *, int, int);

double value1, value2;
int    UNIXtime;

main(argc, argv)
     int argc;
     char **argv;
{

  FILE     *fstatus;
  FILE     *fout;
  int      status = 0;
  LVrecord buffer;
  char filein[256];
  char fileout[256];
  char selement1[256], selement2[256];
  int variable1;
  int variable2;

  if (argc>5){
    strcpy(filein   ,argv[1]);
    strcpy(selement1 ,argv[2]);
    strcpy(selement2 ,argv[3]);
```

```
    variable1 = atoi(argv[4]);
    variable2 = atoi(argv[5]);
  }else{
    printf("Usage: %s <filein> <element1> <element2> <varPos1>
<varPos2>\n",argv[0]);
    exit(1);
  }
  fstatus = fopen(filein,"rb");
  if(fstatus==NULL){
    printf("Can't open input file: %s\n",filein);
    exit (1);
  }
  while (status <= 0){
    status = readValue(fstatus, selement1, selement2, variable1,
variable2);
    if(status==0) {
      printf("%d %e %e\n", UNIXtime, value1, value2);
    }
  }
  fclose(fstatus);
  exit (0);
}

/*****************************************************************/
int readValue(FILE *fstatus, char * el1, char * el2, int var1, int var2)
/*****************************************************************/
{
  int  dummy, sizeDat, sizeEle;
  int countDat = 0,
      countEle = 0;
  int ok = -1;
  LVrecord data;

  dummy = fread(&data.header.LVtime , 8, 1, fstatus);
  if (dummy == 0) return 1;
  UNIXtime = data.header.LVtime - 2082844800;
  fread(&data.header.recordNumber, 4, 1, fstatus);
  fread(&data.header.formatVertion, 8, 1, fstatus);
  fread(&data.header.elementArraysSize, 4, 1, fstatus);
  sizeEle = data.header.elementArraysSize;
  while (countEle < sizeEle)
    {
    countEle++;
    fread(&data.value[countEle].elementName, 8, 1, fstatus);
    fread(&data.value[countEle].classID, 4, 1, fstatus);
    fread(&data.value[countEle].fault, 1, 1, fstatus);
    fread(&data.value[countEle].dataValueSize, 4, 1, fstatus);
    sizeDat = data.value[countEle].dataValueSize;
    while (countDat < sizeDat)
      {
      countDat++;
      fread(&data.value[countEle].dataValue[countDat], 8, 1, fstatus);
      if(strcmp(data.value[countEle].elementName,el1)==0 &&
var1==countDat){
        value1 = data.value[countEle].dataValue[countDat];
        ok = 0;
      }
      if(strcmp(data.value[countEle].elementName,el2)==0 &&
var2==countDat){
        value2 = data.value[countEle].dataValue[countDat];
        ok = 0;
      }
      }
    countDat = 0;
    }
  return ok;
}
```