

Data Logging

Collection and Storage

Fermilab
Beams Division
Accelerator Controls Department

Kevin Cahill

Version 1.2
April 25, 2005

Architecture

Components

There are several data logging domains utilizing varying strategies to collect accelerator data for viewing.

Distribution

The data logging components are distributed across the control system to maximize the control system's available CPU and disk resources.

Operating Systems and Languages

The legacy data logging components are written in the C language and run under the VMS or Linux operating systems. The newer data logging components are written in Java and run under the Solaris or Linux operating systems

Stored Data

The data loggers log double precision IEEE floating-point values with a long timestamp with one millisecond resolution.

Lists

A data logger supports eighteen (18) lists of devices where each list may contain sixty (60) device names. Collection rates and space allocation is list based.

Buckets

Data loggers write and read a bucket of data (250 timestamps and values) to improve write and read performance.

Databases

The data loggers utilize MySQL, a public domain relational database. A single MySQL database table is limited to four (4) Gigabytes. Each data logger list is composed of one MySQL database. The buckets of logged timestamps and values are compressed to approximately 25% using a Java compression utility. The two million bucket limit per table or list keeps the database tables under four Gigabytes.

Disk Management

As each bucket of data is written, a bucket number assigns a place in a circular, ordered list of buckets where the next bucket written for a list overwrites the oldest logged data for that list.

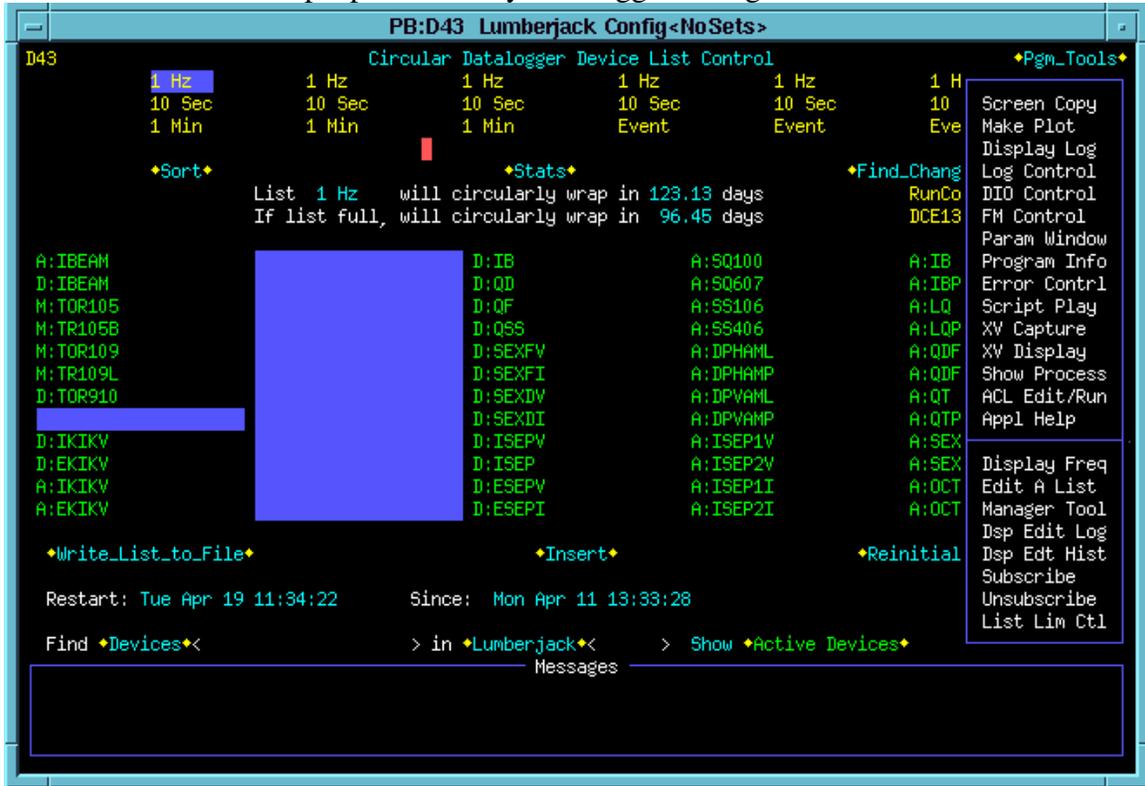
Backups

Individual data loggers are not backed up. Disk problems are infrequent but can cause a loss of data. ArkIv and Backup data loggers provide some data redundancy.

Configuring a Data Logger

Page D43, Lumberjack Devices

This application page displays the configuration of a data logger. Use the Pgm_Tools menu to list authorized people to modify data logger configurations.



PB:D43 Lumberjack Config<NoSets>

D43 Circular Datalogger Device List Control ♦Pgm_Tools♦

1 Hz					
10 Sec					
1 Min	1 Min	1 Min	Event	Event	Event

♦Sort♦ ♦Stats♦ ♦Find_Change♦

List 1 Hz will circularly wrap in 123.13 days
 If list full, will circularly wrap in 96.45 days

RunCo
DCE13

A:IBEAM	D:IB	A:SQ100	A:IB
D:IBEAM	D:QD	A:SQ607	A:IBPSI
M:TOR105	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align:center">Manager Functions</p> <p>List Authorized People</p> <p>Change Logger List Names</p> <p>Change List Data Event</p> <p>Change Logger Bucket Size</p> <p>Add a Data Logger</p> <p>Delete a Data Logger</p> <p>Swap Logger Nodes</p> <p>List Limit Override</p> </div>		A:LQ
M:TR105B			S106
M:TOR109	S406	PHAML	A:QDF
M:TR109L	PHAMP	PHAMP	A:QDFPSI
D:TOR910	PVAML	PVAMP	A:QT
	PVAMP	SEP1V	A:QTPSI
D:IKIKV	SEP1V	SEP2V	A:SEX10
D:EKIKV	SEP2V	SEP1I	A:SEX12
A:IKIKV	SEP1I	SEP2I	A:OCT10
A:EKIKV	SEP2I		A:OCT12

♦Write_List_to_File♦ ♦Insert♦ ♦Reinitialize_Lumberjack♦

Restart: Tue Apr 19 11:34:22 Since: Mon Apr 11 13:33:28

Find ♦Devices♦ < > in ♦Lumberjack♦ < > Show ♦Active Devices♦

Messages

Adding a Data Logger

Steps include:

- Select an available node. New data loggers are assigned to Java nodes.
- Choose a name.
- Choose list rates and names.
- Allocate the number of buckets for each list.

Increasing a List's Bucket Allocation

A list will circularly wrap in a longer period of time when assigned more buckets. Be aware the increase will not take effect until the current bucket assignment's value passes the prior bucket limit.

Decreasing a List's Bucket Allocation

Decreasing a list's bucket allocation may make more buckets available for other lists. The database size does not decrease until the space is reclaimed. In general, one should expect that the list's database table will be dropped and recreated resulting in the loss of logged data for that list when decreasing bucket size.

Managing Space

Thoughtful space consideration for the long term is suggested. Increased bucket allocations do not immediately effect wrap time, and decreasing bucket size is likely to be effected by clearing a database table. A maximally sized 1 Hertz list with a full complement of 60 devices will wrap in $(2,000,000 \text{ buckets} * 250 \text{ points/bucket}) / (60 \text{ list entries} * 86400 \text{ seconds/day})$ or 96.45 days.

Wrap times are directly proportionate to the number of buckets assigned to the list and inversely proportionate to the number of items in the list and the rate items are being logged.

A device list may contain more than 60 devices if some of the entries specify array devices. Each element of an array consumes the space of a device.

Data Collection

Periodic

The fastest continuous rate is 15 Hz. The slowest continuous rate is unlimited. Rates faster than 1 Hz are excluded from archiving.

Hard Event Collection

Collection on Tevatron clock event is possible if the target front-end supports collection on clock event. If hard event collection is specified to an incapable front-end an error is generated and no data will be logged.

Soft Event Collection

Soft event collection is utilized for target front-ends that do not support collection on Tevatron clock and for all collections on Tevatron clock event plus delay and for all collections on a state transition event (plus delay) unless the front-end supports the Gets32/Sets32 since the older data acquisition protocol cannot express collection on Tevatron clock event plus delay or on state transition.

Soft event collection is conducted using one-shot data collection upon detection of the event and consequently may suffer a loss of accuracy for very volatile signals. Since one-shot reads encompass more network traffic and front-end CPU cycles, data loggers limit soft event collection to no faster than one Hertz.

Specifying the Either Event Collection

A data collection event string such as “e,2,h,0” is ‘read’ as on hard event 02 with no delay. The string “e,2,s,0” is ‘read’ as on soft event 02 with no delay. The string “e,2,e,0” is ‘read’ as on hard event if the front-end supports it, otherwise on soft. When specifying list rates, a hard event string will not collect data from a front-end that does not support Tevatron clock, and a soft event string will always collect one-shot data shortly after detection of the event, but an either event string will proceed as hard when possible, soft otherwise.

Client Events

Lists may have a name and rate that specify a client event. Those lists do not have a data collection job assigned by the logger but receive values to log a client process. Examples include clock, state, and setting loggers.

Specialized Data Loggers

ArkIv

The archive logger begins a data collection at midnight from other loggers for logged values at a 5 minute interval minimum. There are several loggers excluded as sources of data to the ArkIv logger.

Backup

The backup logger begins a data collection at midnight to backup all devices at their fastest periodic rate, on each collected clock event or state transition, and most client logging. Periodic rates are clipped to no faster than 1 Hertz, and several loggers are excluded as sources of data to the Backup logger.

BLM

This client logger logs Booster loss monitor and Chipmunk readings.

CBSDA

The client logger logs shot data from completed Collider shots.

EventC

This client logger logs Tevatron clock events.

EventV

This client logger logs software state transition events.

FTP(n)

This client logger logs fast time plot data.

PBSDA

This client logger logs shot data from completed Pbar transfer shots.

QRay(n)

The quick array logger logs all elements of an array as if they were multiple values of the first array element..

Sets

This client logger logs settings across the control system.

Snap(n)

This logger logs snapshot data.

State

This client logger logs software state transitions (legacy port address).

Retrieval

Timestamps

Data loggers with multiple databases when logging a device at multiple rates will return sets of data that are not strictly time-ordered. The databases will be searched serially, returning data from each database in turn as data matches the user's request.

The data loggers support the specification the list of interest for a retrieved device and support the specification of the desired logged event string in the request structure.

Data Return Ordering

Data loggers use multiple databases to store data. The databases are searched in list order. If a device moves from a higher numbered list to a lower numbered list and retrieval is specified over a time period encompassing both list's data, the recent data will be returned before the older data. This effect is also possible in client loggers where list allocation moves at run time to spread the resource usage as equally as possible across all client logger lists.