# Client Data Logging

Fermilab
Beams Division
Accelerator Controls Department

Kevin Cahill

Version 1.0
February 28, 2003

## Introduction

Client data logging involves a client process pushing data to a data logger as opposed to the data logger pulling data from the control system on some data event.

The Tevatron clock event data logger is an example of a client logger. Each logged point represents an observed Tevatron clock event's time within the supercycle. The logged points possess a more meaningful timestamp, and the logger contains no fewer and no more logged points than the number of observed events over some time interval.

Client loggers may implement complex data event processing that is not feasible to express as a data event. Client loggers may suppress logging under some defined conditions or change the logging rate under others.

## Availability

Many client loggers are dedicated to a specific purpose and are populated by Open Access Front-end Clients sharing a Java virtual machine with the data logger thus able to reference a DataLoggerDisposition with methods that directly support client data logging.

All Java clients capable of starting a DaqJob have access to client logging capability.

## Client Logging Jobs

The following code fragment demonstrates client data logging:

```
DataSource from = new ClientLoggingSource();
ClientLoggingDisposition to = new ClientLoggingDisposition("EStats");
ClientLoggingItem item = new ClientLoggingItem(true, to);
DataEvent event = new DefaultDataEvent();
double value = 1.0;
item.queue("Z:ESTATS", new java.util.Date(), value); // optional
DaqJob job = new DaqJob(from, to, item, event, user, control);
job.start();
for (int ii=0; ii<25; ii++)
{
        item.queue("G:ESTATS", new java.util.Date(), value);
        value += 1.0;
        item.queue("Z:LTEST", new java.util.Date(), value);
        item.flush();     // flush is self-throttling
 }
```

ClientLoggingSource has no argument.

ClientLoggingDisposition specifies the name of the logger. “EStats” is the engine statistics logger and is available to all clients. Many other client loggers have a dedicated function and should be avoided.

ClientDataItem takes two arguments; the first a boolean expressing if successful callbacks are desired, and the second is an implementer of the AcnetErrorCallback interface. In this code example, the callbacks are directed to the disposition which prints a line to System.out.

As long as the job persists, the ClientDataItem is used to queue and flush data to the data logger.

The job may be run on any engine. ACNET is used as the transport mechanism when the job and logger are on differing engines.

## Custom Client Devices

Clients may create database devices to data log statistics and values specific to their application. Again, EStats, an Open Access Client Front-end as well as a data logger is an appropriate front-end for this purpose. The device “G:ESTATS” is a good example to be used as a template for the creation of these devices.