GLAST LAT SYSTEM SPECIFICATION	Document # **LAT-TD-01340-02**	Date 13 March 2003
	Author(s) Joanne Bogart, Eduardo do Couto e Silva	
	Subsystem/Office Science Analysis Software	
Document Title **LAT SAS Engineering Model Calibration Infrastructure**		

Change History Log

Revision	Effective Date	Description of Changes
02	March 13, 2003	Add color-coding to the metadata description. Changes to section 7.2 and 7.3 including addition of discussion of mirror sites.

Gamma Ray Large Area Space Telescope (GLAST)
Large Area Telescope (LAT)
Science Analysis Software Subsystem
The LAT SAS Engineering Model Calibration Infrastructure

Contents

1. Purpose

This document describes the prototype calibration infrastructure to be implemented for the Engineering Model activities.

2. Scope

This document presents details on the calibration infrastructure required for the LAT.

3. Acronyms and Definitions

3.1. Acronyms

ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
EM	Engineering Model
FEE	Front-End Electronics
I&T	GLAST LAT Integration and Test Subsystem
SAS	Science Analysis Software Subsystem
SVAC	Science Verification Analysis and Calibration
TDS	Transient Data Storage

3.2. Definitions

Gaudi: A software framework specialized for offline experimental event analysis.

Subsystem: A functional subdivision of the LAT, e.g. I&T, SAS.

System: A functional subdivision consisting of two or more subsystems

4. Applicable Documents

LAT-MD-00573 " LAT I&T SVAC Plan for the Engineering Model"

“Gaudi Developer’s Guide”

5. Requirements

- Archive all calibration data with even a remote chance of being of future interest.
- Provide transparent access to "the right" calibration data for analyzing a given event (e.g. during event reconstruction).
- Support queries for access to a particular calibration data set.

- Facilitate others forms of query, such as per-channel histories.
- Facilitate automated report generation.

5.1. "The Right" Calibration Data

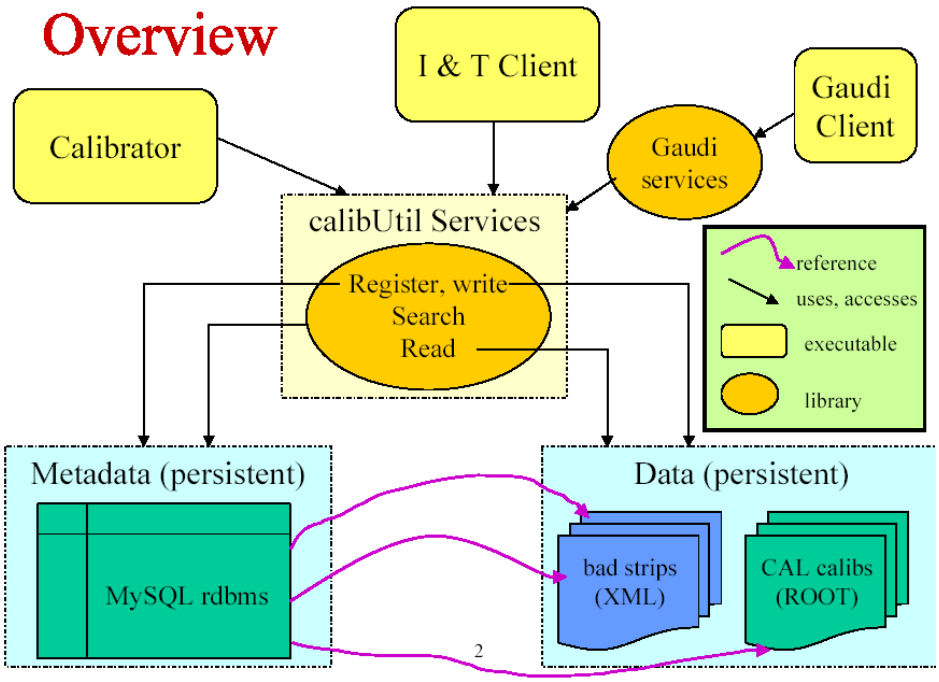
During event processing, the right calibration data

- must be of the right type (won't do to fetch tracker alignment when what is needed is CAL pedestals!)
- must be for the right instrument
- should have a period of validity which includes the time the event was taken
- should be of production quality (that is, the calibration output should be an accurate representation of the state of the instrument)
- must be in a compatible format, one the client program can handle

6. Client Support Overview

The primary calibration database (SAS database) and related services are developed and maintained by the SAS group to support the necessary inputs to the reconstruction software. Relationships between the relevant entities for the three supported client types (Calibrator, I & T client, Gaudi client) are shown in the diagram below.

Figure 1. Client Support Overview



- The **Calibrator** independently creates bulk data in one of the accepted formats, and then uses services depicted here to register the data (that is, create a new metadata record describing it in the MySQL database).

- **I & T clients** need low-level read access to both the bulk data and the metadata.
- **Gaudi clients** (that is, event processing clients) don't use the low-level calibUtil services directly, but only via Gaudi constructs such as conversion services.

7. Elements of Implementation

The building blocks provided by SAS, as shown in the figure above, include:

- A place to store (bulk) calibration data. Most likely this will be SLAC-provided nfs disk space
- A registry for calibration data sets. Implementation used is a MySQL database served by a SLAC node.
- Low-level software services to access the registry and the bulk data. These reside in the calibUtil software package.
- Transparent access to the "right" calibration data for event-processing clients. These are implemented in the software packages CalibData and CalibSvc (labeled Gaudi services in the figure).

7.1. Data and Metadata

The calibration infrstructure deals with two kinds of information: the calibration data proper, and information *about* a particular calibration data set, such as its period of validity. The latter will be called the **metadata** for the data set. The entire collection of metadata resides in the **Calibration Registry**. Metadata include, but are not limited to, criteria used in searching for a particular data set. Metadata fields are sufficiently generic that nearly all of them will be meaningful for all or almost all calibration types. All of this (highly structured, searchable) argues for keeping the metadata in a relational database. Current plans are to use MySQL, which has several desirable properties, including some **advantages** over Oracle.

- It implements almost all standard SQL features (perhaps all by now), and certainly all that we would need; performance is reputed to be quite good.
- It runs on all of our platforms.
- It's **open source** and **free**.
- It has a relatively **friendly API**.

Calibration data, as opposed to metadata, come in many forms. Output from a calibration procedure may be small or large, of fixed or variable size, more or less structured, depending on the procedure and the state of the intrument at the time of the calibration. A relational database representation would be suitable, perhaps optimal, for some procedures, but not others. Fortunately there is no requirement or even preference that all calibration data be stored the same way. It would be sensible to stick to a small number of supported formats since software resources are not infinite. Two or three (ROOT file, XML file, MySQL table??) ought to be enough to avoid any serious awkwardness in representing and accessing the data.

7.2. Gaudi Data Service Paradigm

A Gaudi data service, as described in the Gaudi Developer's Guide, will be implemented for calibration data. Such a service handles all the details of finding the right data set and insuring that it is available in an in-memory representation (in Gaudi terminology, the Transient Data Store, or TDS). The client application accesses the data exclusively via TDS classes; it has no need to know anything about the persistent form of the data, nor does it need to know how to find a particular data set among several of the same kind.
Components of the Calibration Data Service include

- a Data Service. The Data Service (implementing the Gaudi IDataProviderSvc interface, or something similar) has overall responsibility for making the calibration data avalilable in the TDS (in memory) if the data is not laready present in the TDS. The Data Service will do this by delegating most of the work to a MySQL conversion service and ultimately to another conversion service which knows about the physical form the bulk calibration data is in.

- Conversion services. For each supported persistent form a class which is responsible for converting data sets in that form to an in-memory representation. Typically a conversion of a particular dataset is delegated to a converter.

- Gaudi Converters. Classes which, given access to the persistent form of a data set, can convert it to its TDS representation. The number of these required will depend on the similarity (or not) of the different calibration output data sets, but there will be at least one Converter for each supported persistent form.

The Data Finder makes use of the remaining classes as needed to get the data into the TDS without any intervention from the client.

7.3. Accessibility and Portability

Implicit in several of the high-level requirements is the assumption that the Calibration Services will be available in any environment in which clients might run, be that at SLAC, at another collaborating institution, or at 30,000 feet on a laptop. (This most likely excludes archiving or any other applications involving writes.) Three techniques will be employed to achieve this:

1. Access over the network to the central data store

2. Mirrorwing of the complete store at a samll numebr of overseas sites.

3. Replication of some part of the central store for local use.

Network access to the central store or a supported mirror site is normally preferred when feasible. We expect the calibration data store, broadly speaking, to contain only two kinds of objects visible to applications: MySQL data and (maybe) physical files, such as ROOT files. MySQL has a client/server architecture in which network access is transparent except for possible performance degradation if the network connection is poor. Physical files can be made readily network accessible for reading by, for example, keeping them in a dedicated anonymous ftp area.
In case there is no network connection or only a poor one, users may copy needed files and extract parts of the MySQL database offline. They would have to maintain their own pool of data files and something to substitute for the central MySQL database as metadata server. Client programs would have to be told, perhaps via Gaudi job options, to use the local metadata server and local files.

8. Calibration Services

8.1. Data Archiving

There are two parts to archiving a calibration data set: create the persistent form of the data in one of the supported formats, then register the data set in the Calibration Registry. How the persistent form is created will vary, depending on the calibration procedure. Registering the calibration just consists of adding a row (the metadata) to a MySQL table. Both operations must be supported both for Gaudi and non-Gaudi applications. Utilities will be provided to write the metadata and to do at least the generic parts of writing the persistent form of the data.
The contents of the metadata will include fields used primarily for **selection** and others needed for **i/o**.

Table 1. Metadata Description

Field name	Explanation, Typical contents
Calibration type	E.g., TKR alignment, TKR noisy channel, CAL light asymmetry, ACD pulse height calibration (for purpose of defining Veto threshold) etc.
Flavor	String, which may be used to identify calibrations, intended for particular applications or analyses. If unspecified, will default to "vanilla".
Serial number	Unique serial number for this record and the corresponding calibration output (TBD: is the serial number unique over *all* records, just for records for this calibration type, or do we want two serial numbers?)
Software version or data format version	Intent is to supply enough information so that programs wishing to read the data can tell whether the data will be readable.
Data format	Refers to medium used for persistent data. May be one of a small list of possibilities, such as MySQL records or ROOT file
Persistent data identifier	Will depend on the format. For ROOT files, this field might be a file spec. For rdbms data, could be an index.
Validity start time	Demarcate time interval for which calibration is known to be valid
Validity end time	Demarcate time interval for which calibration is known to be valid
Procedure completion time	Time the dataset was made
Instrument calibrated	E.g. engineering model, flight instrument, etc.
Calibration procedure level	Possible values will come from a predefined list including 'test', 'development', 'production' and 'superseded'. These terms refer to the procedure being used and say nothing about whether the component being calibrated was deemed acceptable.
Calibration status	Possible values will come from a predefined list including 'OK' and things like 'incomplete' or 'aborted'
Data size	Could be number of bytes, number of records, or something else. Precise meaning will depend on data format and calibration type.
Creator	Could be hardware procedure or software algorithm. Should also include procedure or software version information.
Input description	Name, version, etc. of any input data sets.
Comment field	Anything that might be of interest but doesn't fit in the other categories.

8.2. Data Access

The access to calibration data during event processing is presumed to take place only in a Gaudi process, so Gaudi concepts and tools are freely used. In the course of analyzing an event, if an application requests calibration data of a particular sort the following sequence ensues.

- By the standard Gaudi mechanism a check will be made to see if the requested data (i.e., data of the proper calibration type and flavor) already exists in the TDS.
- If data of the requested calibration type and flavor is not already in TDS, the Data Provider (through its proxies) will attempt to come up with it by searching the Calibration Registry. If found, the information in the metadata

record will be used to dispatch the correct Conversion Service and Converter. The Driver is supplied as part of Calibration Services. Some part of the Converter may have to be supplied by the Subsystem.

At this point data of the correct type is in the TDS, but, unless just fetched, it might not be current. The client will make another request (updateObject) to the Data Provider, causing current data to be fetched if necessary. .

8.3. Data queries

Assuming enough information is known to uniquely specify the desired calibration output, the metadata may be fetched directly from the MySQL database by SQL commands or via a C++ utility to be supplied. This utility will not require the Gaudi environment. From the metadata one can determine how the persistent data is to be read and where to find it.

Per-channel history is not easily extracted from the primary data structures (Calibration Registry plus the collection of all persistent data). Rather than recomputing it each time it is requested, we could keep additional derived data indexed by channel or whatever hardware division is most convenient. The data would be updated whenever a new calibration record of a calibration type relevant to this hardware is entered into the Calibration Registry, or maybe only for records meeting certain criteria, such as procedure level = production and status = OK. The updating procedure should be invoked automatically.

8.4. Automatic Report Generation

If the information of interest is contained within the metadata automated report generation will be straightforward. It would be well worth adding a few more fields to the metadata if all summary information needed for standard reports could be captured this way.