

ACNET Package

Fermilab
Beams Division
Accelerator Controls Department

Kevin Cahill

Version 1.0
March 17, 2003

Introduction

ACNET is a messaging protocol for the Fermilab Accelerator Control System. In Java, it is carried over UDP. ACNET defines the framing of messages sent from and received on socket 6801. A single ACNET implementation can be instantiated on a node. Further, the message handling with ACNET clients is generally restricted to clients running within the same virtual machine.

This document describes its implementation with the goal of describing how ACNET and its clients work within a Data Acquisition Engine (DAE). It begins with an introduction to the package's modules.

Finally, the section entitled ACNET in a Client describes how a client not running in an engine virtual machine may use ACNET.

Modules

AcnetUtility.java

This module contains methods to stop ACNET.

AcnetNode.java, AcnetNodeTables.java

These modules contains support for addressing ACNET nodes which have a six character logical node name, a byte each of trunk and node number, an IP name, and an IP address.

AcnetConnection.java

This is ACNET's largest module and contains support for obtaining a task connection and sending and receiving ACNET messages.

AcnetHeader.java

All ACNET messages carry a header describing the message source, destination, and content. This module supports the construction and deciphering of these headers.

AcnetIPHeader.java , AcnetMCHheader.java

These modules supports message headers encompassing a restricted set of multicast message types including Tevatron clock event, software state transition, pool, and alarm.

AcnetReadThread.java

This module always has a blocking read queued to the ACNET socket. Incoming messages are routed by this module.

AcnetError.java

Message headers and contents may contain a short word of ACNET error status consisting of a byte of facility code and a byte of error status. This module defines several facilities' error codes, supports the database translation of error code to message strings, and supports error caches for statistics and reporting.

AcnauxReplier.java

This module supports statistics returns common to all control system ACNET implementations through the connected task name ACNAUX.

AcnetReplier.java

This module is an abstract class describing methods a connected task must implement to handle incoming requests.

AcnetRequest.java

This module describes an incoming ACNET request.

AcnetReply.java

This module describes an interface a connected task implements to receive ACNET replies.

AcnetClientReply.java

This module describes an interface a client application not running within the same virtual machine as an engine implements to receive ACNET replies.

AcnetKiller.java

This module supports ACNET killer messages requesting that any outstanding ACNET communications with this node should be canceled. This message type is sent to all front-ends on startup.

BounceReplier.java

This module supports bouncing ACNET messages used to measure throughput and response.

CheckPassthruThread.java

This module supported the bridge between UDP and TokenRing. It is no longer useful.

Rad50.java

ACNET task names in message headers are encoded in radix Rad50, an encoding scheme that packs 6 ASCII characters into a short word. This module supports the encoding and decoding of Rad50.

Signaler.java

This module supports synchronization objects.

Startup

The engine's `gov.fnal.controls.daq consolidate.EngineStartup` module starts ACNET using `AcnetUtility.initializeAcnet`. ACNET is stopped by the module `gov.fnal.controls.servers.DAE.DAE` using `AcnetUtility.stopAcnet`.

Using ACNET in an Engine

There are dozens of examples of connected ACNET users in the code repository. Search the repository for new `AcnetConnection` and follow the examples in code.

ACNET in a Client

ACNET capable machines have a valid entry in the ACNET node tables and have various ACNET classes writing and reading ACNET's port address. A client application is able to issue ACNET request and receive replies through a `DaqJob` on a cooperating engine.

The `AcnetClientConnection` class in the `gov.fnal.controls.daq.datasource` package describes how to use ACNET in a client application.