ADDI NG FLEXIBLE SUBSCRIPTION OPTIONS TO EPICS*

Abstract
The need for a mechanism to control and filter subscriptions to control system variables by the client was described in a paper at the ICALEPCS2009 conference [1]. The implementation follows a plug-in design that allows the insertion of plug-in instances into the event stream on the server side. The client can instantiate and configure these plug-ins when opening a subscription, by adding modifiers to the channel name using JSON notation [2]. This paper describes the design and implementation of a modular server-side plug-in framework for Channel Access, and shows examples for plug-ins as well as their use within an EPICS control system.

Server-Side Plug-Ins — Instantiated and Configured by the Channel Access Client Using JSON Modifiers

- myPV.A("ts":{})
- myPV.RVAL("dbnd":{"m":"rel","d":7.5})
- myArray.VAL[-5:2:8]
- myPV.VAL("sync":{"m":"while","s":"red"})

Extending the Existing Update Mechanism in a Flexible and Modular Way

Plug-Ins Can Manipulate Data Updates in Many Ways
- Manipulate the data
- Manipulate the meta data (alarm, timestamp)
- Change the type of data
- Change the size of arrays
- Drop updates
- Insert updates

Flexible Plug-In Options Control Synchronization with External Events and System States
External events from timing/event systems or other software modules set or reset system states.

- The „sync“ plug-in forwards or drops data updates depending on state transitions.

Low Priority CA Context
for CPU intensive, atomic operations that work on single updates

High Priority Database Context
for real time relevant, throttling, or average-type operations

*Work supported by U.S. Department of Energy (under contracts DE-AC02-06CH11357 and DE-AC02-98CH10886), German Bundesministerium für Bildung und Forschung and Land Berlin.


WEPKS020 ICALEPCS2011 Conference, Grenoble, France.