# DPM - Feature #11367

## It has been proposed that DPM needs to support settings

01/07/2016 11:18 AM - Richard Neswold

| | | | | |
|---|---|---|---|---|
| **Status:** | Closed | | **Start date:** | 01/07/2016 |
| **Priority:** | High | | **Due date:** | |
| **Assignee:** | | | **% Done:** | 0% |
| **Category:** | Data Pool Manager | | **Estimated time:** | 0.00 hour |
| **Target version:** | DPM v1.9 | | | |

**Description**

# Abstract

Erlang DPM's responsibilities are based upon CLIB-DPM's; handling readings and supporting redirection. Settings in CLIB may have shared some code with CLIB-DPM (database look-up, scaling routines, etc.) but the actual setting was done in the client's context. It has been suggested that the new DPM should handle settings.

# Pros

- Clients would only have to track the latest DPM protocol to gain readings and settings. If DPM doesn't support settings, each client language needs to re-implement SETDAT/SETS32/whatever-follows.
- There has been talk about adding Channel Access to the bottom layer of DPM to support both ACNET and EPICs front-ends. If DPM handled settings, clients wouldn't have to know how to talk to multiple front-end architectures.
- DPM would forward settings to the setting service so clients wouldn't have to re-implement that code over and over.

# Cons

- ~~What does it mean to run settings through a data pool? You can't merge requests.~~ Settings won't get merged.
- ~~We're just adding one more hop to the setting's path.~~ The SETS32 header has an FTD field which can be used to schedule settings on an event, so the extra hop becomes irrelevant.
- DPM uses DRF2 strings to cleanly and succinctly describe a data acquisition request. DRF2 doesn't support the notion of settings. It would be nice if readings and settings had similar ways of describing the intent rather than having two unrelated APIs.

This topic needs more thorough discussion in a meeting.

---

**History**

**#1 - 03/28/2016 09:46 AM - Richard Neswold**

*- Description updated*

Update the "Cons" list to remove some concerns that Charlie King and I discussed and explained away.

We've had several requests for Settings support through DPM. The meeting needs to happen sooner than later.

**#2 - 10/27/2017 01:35 PM - Richard Neswold**

*- Category set to Data Pool Manager*

*- Target version set to DPM v1.6*

**#3 - 11/27/2017 03:39 PM - Richard Neswold**

*- Priority changed from Normal to High*

*- Target version changed from DPM v1.6 to DPM v1.7*

Raise priority and move to v1.7 release.

**#4 - 12/05/2017 09:15 AM - Richard Neswold**

*- Target version changed from DPM v1.7 to DPM v1.8*

Push this off to a later release. **Charlie** has made progress understanding how **Andrey** supported Kerberos credential passing, so effort toward this issue can benefit from what he learned.


**#5 - 04/13/2018 10:46 AM - Richard Neswold**

*- Target version changed from DPM v1.8 to DPM v1.9*


**#6 - 03/20/2019 02:52 PM - Richard Neswold**

*- Status changed from New to Closed*


Several of us had informal discussions that setting support shouldn't be part of DPM. In the past, the console DPM didn't making settings. Somehow we've attached this feature onto DPM. Moving settings into a separate service has the following benefits:

- DPM can focus on supporting all the ways we can read data
    - Accelerator data from ACNET front-ends and IOCs
    - Save/Restore data
    - Data logger data
    - FTP and snapshot data
- Setting service can focus on handing settings
    - Protocol can be tailored to setting requirements (instead of shoe-horning it into the DPM protocol)
    - Validating credentials of the request
    - Writing successful settings to the setting database
    - Handling knobbed settings

This codebase may actually run in the same VM as the DPM, but I'd recommend is not be part of the DPM package. It would be is own package and would connect to ACNET through its own handle.