

MOOC Framework - Bug #23100

Sets32 Settings Fail from FSM to Mooc fail with MOOC_ACNET_COMM_ERROR

08/12/2019 11:34 AM - Dennis Nicklaus

Status:	New	Start date:	08/12/2019
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	MOOC Framework	Estimated time:	0.00 hour
Target version:			

Description

When running the CW Coupler Conditioning FSM for the first time in a year or more, we had a problem with the settings to the MOOC device E:CWSGPA on node HINSGP.
The FSM looked like it was working and should have been updating the value of E:CWSGPA, but the value wasn't updating. D54 settings log showed the settings were going out from the FSM OAC, FSMDEV.
FSMDEV is hosted on dce09, and its log showed settings were returning MOOC_ACNET_COMM_ERROR (57 -18) for that device. That error appears in the sets32 MOOC code in two places, both related to checking the length of the acnet packet.

We (Jerry) have since moved the device from HINSGP to an erlang frontend (it's a GPIIP device read out over ethernet), so this particular device doesn't return the error any longer.

But it seems it should be easy enough to re-create the error and this issue is to investigate why we're producing this error. I swear this used to work. And the settings work fine to the Erlang device/node now.

Linden, how does the FSM make its setting requests?

Running stats() on HinsGP shows:

AS statistics

```
Retdat message recieved      : 0      Retdat ACnet mem. releasing  : 30470
Retdat flashed request       : 0      Retdat cancels              : 15225
Retdat cancel errors        : 0      Retdat one shots           : 0
Retdat one shot errors      : 0      Retdat multiple requests   : 15245
Retdat multiple req errors  : 0      Retdat error replies       : 20
Setdat message recieved     : 261    Setdat ACnet mem. releasing : 261
Setdat flashed requests     : 0      Setdat message errors      : 0
Setdat send replies        : 0      Lists recieved by the reader : 0
Cancels recieved by reader  : 15225  DAS unidentified cancel    : 60
Triggers defined by DAS     : 15225  Triggers defining errors   : 0
Triggers undefined by DAS   : 15165  Messages queued by DAS     : 0
Messages queuing errors, DAS : 0      Memory allocations by DAS  : 3779559
Memory allocation errors, DAS : 0      Memory deallocations by DAS : 3764334
DAS replies                 : 3764330 Requests to the DB          : 4
Requests processed by writer : 0      Request to ACNET from writer : 4
Acnet errors in Writer sends : 0      Successfull sends by Writer : 4
Corrupted message from ACNET : 257
```

which indicates it is this code in das.c sets32_rq() which produces the error:

```
/* The length better be exactly zero at this point. Otherwise the
   * packet was too big or too small. */

if (len != 0) {
    das_peg1(RDT_CORRUPT);
}
```

Slightly-annoying-but=probably-doesn't-need-to-be-fixed side issue: After we moved the device from one front-end to another, the FSM didn't notice the change and still kept trying to set the old device.

To investigate further, we need to select a set-able device and make a sets32 request to it. HinsGP is running vxworks 6.4 on a 5500.

