

## GDP - Bug #54

### Multiple multi-reads (of size 1, typically) do not return the correct number of events.

09/23/2016 07:30 PM - Nitesh Mor

<b>Status:</b>	Closed	<b>Start date:</b>	09/23/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Eric Allman	<b>% Done:</b>	0%
<b>Category:</b>	libgdp	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
<p>For a multi-read of size 1, a client should receive 2 events (GDP_EVENT_DATA followed by GDP_EVENT_EOS). In case of N multi-reads of size 1, this translates to 2*N events. But this doesn't always happen. In as small as 2 multi-reads, the number of actual events range from 2 to 4 across different runs.</p> <p>Attached is a sample Python program to reproduce it; it attempts to implement an asynchronous read by calling multi-reads of size 1. The same behavior is seen in C as well.</p> <p>The server returns the correct number of events (verified with WireShark). The bug is in the client library processing. My hunch is the bug lies in the processing of the PDU starting at <a href="#">source:gdp/gdp_main.c#L425</a>.</p> <p>In <a href="#">source:gdp/gdp_priv.h#L409</a>, the assumption is that a request can not have a WAITING status, while maintaining an active subscription (or multi-read, for that matter). The logic in <a href="#">source:gdp/gdp_main.c#L425</a> follows this assumption. However with quite a few multi-reads of size 1, this assumption fails: a reader may be WAITING on <code>_gdp_invoke</code> of a new multi-read command when a PDU as a result of a previous multi-read arrives.</p> <p>Attached is a debug log for 2 multi-reads of size 1, which should see 4 events but only 2 events are passed back to the client. See line#150, where <code>req-&gt;state</code> is WAITING; the PDU that was received is the data from result of multi-read and the control flow goes to the if block at <a href="#">source:gdp/gdp_main.c#L425</a> rather than to <a href="#">source:gdp/gdp_main.c#L443</a></p>			

## History

### #1 - 09/27/2016 10:25 PM - Nitesh Mor

commit:4d6ff457 seems to have it fixed. I can no longer reproduce the bug.

Leaving this open though, just in case you have more sub-issues to be addressed that I might not be aware of. Feel free to close it.

### #2 - 09/28/2016 10:51 AM - Eric Allman

- Status changed from New to Feedback

Should have been fixed in commit:4d6ff457. However, it turns out that small multi-reads still serialize behind the actual read command, so commit:8204d4b2 adds `gdp_gcl_read_async`, which asynchronously returns exactly one event PDU for each read.

### #3 - 09/29/2016 05:01 PM - Nitesh Mor

- Status changed from Feedback to Closed

Verified `read_async` works. Closing this. Thanks

## Files

reader_test_async.py	1.4 KB	09/24/2016	Nitesh Mor
multi-read.debug.log	19.7 KB	09/24/2016	Nitesh Mor