Technical Notice Reference

14-0001

Technical Note

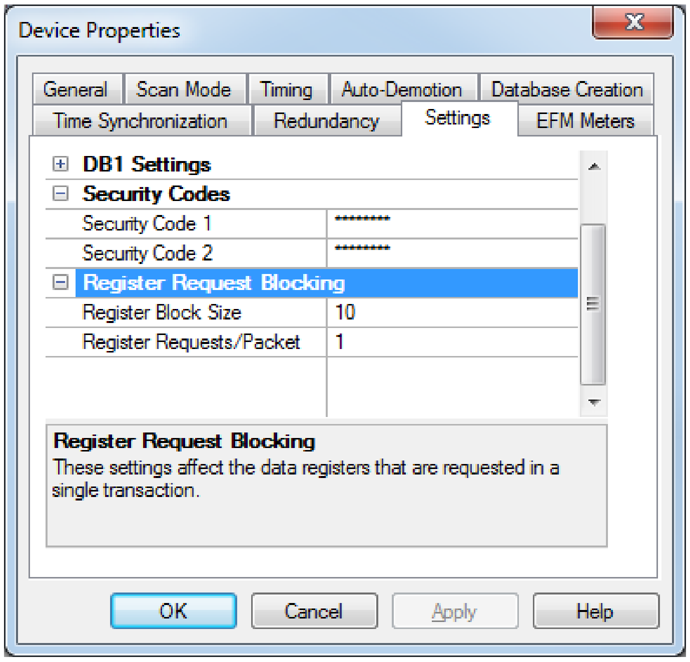
Blocking Requests in the ABB Totalflow Driver

1. Introduction

Like many KEPServerEX drivers, the ABB Totalflow Driver can be configured to group read requests together. It performs the read for these items in a single transaction with the device in an optimization commonly referred to as "Blocking." Blocking requests optimize device communications by allowing the driver to read multiple registers with a single request/response transaction. The ABB Totalflow Driver was designed with configurable block settings to allow users to enhance the driver's performance for a number of different applications. This article will discuss the driver's block settings, how these settings affect the request/response size, and present an example use case.

2. Device Settings

The ABB Totalflow Driver has two device settings that affect blocking and can be configured to optimize the number of registers read in a single device read transaction. To access these settings, right-click on the device and select **Properties**. Then, open the **Settings** tab.



DB1 Settings	
Security Codes	
Security Code 1	********
Security Code 2	********
Register Request Blocking	
Register Block Size	10
Register Requests/Packet	1

Register Request Blocking
These settings affect the data registers that are requested in a single transaction.

2.1 Register Block Size

The Register Block Size setting configures the maximum number of contiguous registers read in a single transaction. It can be configured to any value between 1 and 100. The default setting is 10.

Note: This setting is a maximum. If the driver only has active references for a block of 5 contiguous registers, a request will only be made for the 5 items and not the Register Block Size. This helps prevent performance degradation in the event that a client always reads fewer than the configured block size.

2.2 Register Requests/Packet

The Register Requests/Packet setting configures the number of block requests per transaction. This "block-of-blocks" setting allows the driver to further optimize the typical block settings that are supported. It can be configured to any value between 1 and 16. The default setting is 1, which means that only a single block request is made per transaction.

2.3 Example Use Case

The use of these two block settings is best illustrated through an example. Suppose that a client is subscribed to receive 1000 millisecond (ms) updates for the following items:

- Registers 9.0.0 through 9.0.9 (10, single-precision floats)
- Register 9.1.0 (1, 16 bit integer)
- Registers 9.2.5 through 9.2.9 (5, 32 bit integers)
- Registers 9.5.0 through 9.5.29 (30, bytes)

In this example, all the registers were in the same application (9, Holding Register); however, register blocks can be included in the same request independent of their application or array number.

The table below outlines the number of requests, Transaction Time, and Data Payload for several different Register Block Size and Register Requests/Packet cases.

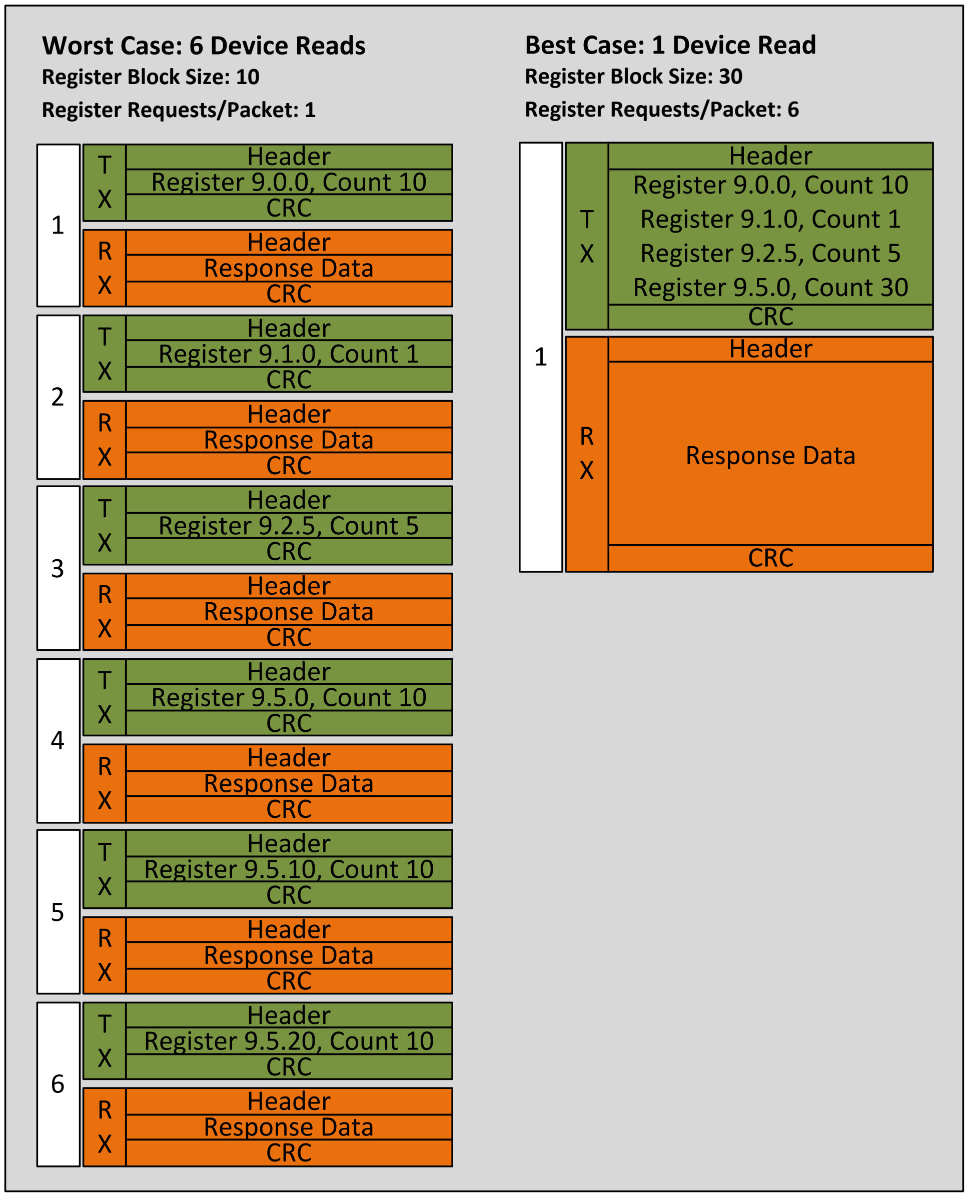
Register Block Size	Register Requests/Packet	Device Read Count	Data Payload (%)*	Transaction Time (ms)**
10 (default)	1 (default)	6	13.0	662
10 (default)	2	3	18.8	460
10 (default)	4	2	22.0	392
10 (default)	6	1	26.5	324
30	6	1	32.6	264
100	6	1	32.6	264

*The Data Payload (%) calculation method is *(D/[T+R])*100*, where:

- *D* is Register Data (bytes)
- *T* is Transmit Data (byes)
- *R* is Receive Data (bytes)

**Transaction Time is based on the theoretical throughput of a 9600 baud, 8 data bits, no parity, and 1 stop RS-232 serial port.

The image below further illustrates the requests made to read the example tags for the best and worst cases outlined in the table above. Adjusting the Register Block Size and Register Requests/Packet to reduce the number of required device reads reduces packet overhead and results in more efficient data retrieval.



As the data illustrates, adjusting the Register Block Size and Register Requests/Packet enables users to tune the application to a desired Data Payload and Transaction Time, which may vary depending on the application. Taking the time to decide the proper block settings for a particular use case is an easy way to increase the performance of a driver while balancing it with other requirements. In this case, the Transaction Time for the best optimization was nearly 2.5 times faster than the worst case! As the blocking settings were increased, the number of transactions required to read the data registers decreased. This reduced the transaction overhead, improved the efficiency of the register requests, and ultimately reduced the amount of time spent sending data over the serial link.

2.4 Block Size Considerations

Understanding why not to increase the Register Block Size and Register Requests/Packet settings to the maximum value is also important.

- Too large of a Register Block Size can result in unnecessary register data reads if the application is requesting many small register blocks. For example, if a client reads registers 9.1.0 through 9.1.9 and 9.1.40 through 9.1.49 and the Register Block Size is set to 50, the driver will perform a read of all 50 registers (9.1.0 through 9.1.49). In this case, performing two reads of 10 contiguous registers would be faster. In general, there is no reason to increase this setting beyond the largest block being requested.
- Using a large Register Requests/Packet and large Register Block Size may result in large data packets. The size of the data response may need to be balanced with the other needs of the application.