



How do the Data Output options work?
(And why don't they always work properly!)

Frequently Asked Question #12

MultiLogger allows you to program the data acquisition system to store data at intervals other than the specified **Single Interval**. For example, a system may be programmed to read all instruments every 15 minutes by entering a Single Interval value of 900, but only store data once a day (at midnight) by selecting **Once a Day** for the Data Output option.

Most of the Data Output options work by using a special programming instruction, instruction number 92, or P92 for short. This instruction is used to check when a certain time interval is met, for example see the following programming that is used by the **Once a Day** Data Output option: (this programming can be loaded into the Mleditor program supplied with MultiLogger by clicking on the gear button located to the left of the Data Output drop-down list)

```
;Output Data once a day
P92  If time is  ;
1:[ 0           ]      Minutes (Seconds --) into a  ;
2:[ 1440        ]      Interval (same units as above)  ;
3:[ 30          ]      Command Code Option (Then Do) ;

P86  Do  ;
1:[StoreSub    ]      Command Code Option  ;

P95  End  ;
```

Note the structure of the P92, the first parameter describes when the interval is satisfied, the second parameter describes the interval, the third parameter describes what is to be performed in the event of the interval being satisfied. In the example above the first and second parameters are satisfied at midnight each day because that is the 0 point in a 1440 minute, or daily, interval. The command code 30 instructs the system to perform the P86 instruction (or others if they were included before the P95), which calls the subroutine to store data. (See the CR10X Operators Manual for a more comprehensive discussion of P92.)

Incidentally, for all you CR10X Programming Buffs out there, this programming could be simplified to a single instruction as follows:

```
;Output Data once a day
P92  If time is  ;
1:[ 0           ]      Minutes (Seconds --) into a  ;
2:[ 1440        ]      Interval (same units as above)  ;
3:[StoreSub    ]      Command Code Option (Then Do) ;
```

If the Single Interval is set to 900, or 15 minutes, then measurements will be taken every 15 minutes, on the even 15 minute interval (i.e. 1200, 1215, 1230, etc.), throughout the day. At midnight the measurements will be stored because the above programming will execute at the completion of measurements.

However, what if the measurements require more than 1 minute to complete? In this case it will actually be 12:01 (or later) when the Data Output instructions are executed, so it is actually at minute 1 (or later) in the 1440 minute interval. What does this mean? **YOUR DATA WILL NOT BE STORED!**

How can this be remedied?

Before proceeding it would be a good idea to find out exactly how long it does take to read all the instruments. MultiLogger includes a special location called the **Read Timer**, this location can be monitored to discover how long it takes to complete all of the measurements. Load the configuration, or Logger, form for the system. Add the location **Read Timer** to the Text Monitor, this can be added by selecting the menu item **Configure | Text Monitor**. Note: First check the locations listed on your Text Monitor, it may already be present. Start the Monitor, while the system is running it will indicate the time that was required, in seconds, to complete the last set of measurements.

Now for the remedies:

- First, the program can be optimized to reduce the time required for all measurements to complete. For example, oftentimes channels where there is no instrument connected specify an instrument. Even though physically there may not be an instrument connected to a specific channel the system will still execute the measurements to read it. Remove these unused channels from your channel configuration (select None).
- Second, the Data Output programming can be modified to support this circumstance, specifically by modifying the instructions or by replacing them with new instructions.

For example, lets assume the program requires exactly (or close to exactly) 72 seconds to execute. The example programming shown previously could be changed to:

```
;Output Data once a day
P92  If time is  ;
1:[1           ]      Minutes (Seconds --) into a  ;
2:[1440        ]      Interval (same units as above)  ;
3:[30          ]      Command Code Option (Then Do) ;

P86  Do  ;
1:[StoreSub    ]      Command Code Option  ;

P95  End  ;
```

Note the first parameter of the P92 has been changed to 1, at minute 1, or at 12:01AM, data will be stored. Another possibility is to change the P92 to a compare instruction, the P89.

```
;Output Data once a day
P89  If (X<=>F)  ;
1:[TimeLoc      ]      X Loc ;
2:[1            ]      Comparison Code Option (=)  ;
3:[0            ]      F      ;
4:[30           ]      Command Code Option (Then Do) ;

P86  Do  ;
1:[StoreSub    ]      Command Code Option  ;

P95  End  ;
```

In this example we will make a check for a specific time of day. The TimeLoc is updated at the start of measurements so for the measurement initiated at Midnight it will contain 0, regardless of how long the program requires to execute. This technique is well suited to situations where you may not know how long the program will take or the time could vary considerably. As can be seen in the parameters following the P89, the TimeLoc (parameter 1) is checked with comparison code 1 (parameter 2), for equal, to 0 (parameter 3), or midnight. If it matches then the StoreSub is executed.

Please contact Canary Systems or your software vendor if you require further assistance.