5 Gould Road, PO Box 2115
New London, NH 03257
Voice: (603) 526-9088
info@canarysystems.com
www.canarysystems.com

# Guidelines for Creating CR1000 Instruction Files

MultiLogger Application Note #16

## ***Overview***

MultiLogger version 4.0 includes support for the CR1000. Numerous instruction files (extension .CR1) can be created to provide support for more types of measurements and other actions. The following instruction files can be modified/created, along with their corresponding section in multilogger.ini (in the \CR1000 folder):

- **Data Output – Section [Data Output]**
- **Alarm Action – Section [Alarm Actions]**
- **Output Device – Section [Storage Devices]**
- **Gage Types – Section [Gage Types]**
- **Upper Channel Devices – Section [Upper Channel Devices]**
- **Check Alarms Type – Section [Alarm Type]**
- **Processing Files – Section [Processing Files]**

All of these files follow basically the same format, they are named with the extension .CR1 to denote CR1000 CRBasic programming, they consist of sections of CRBasic programming with comments as needed and they include references to variable names utilized by MultiLogger.

The variable name references can be any of several types, as follows:

1. Input Storage Labels – These are the default locations and labels used when first creating a new configuration file. They are retrieved from the \CR1000\multilogger.ini file and then copied to the configuration file. There may be occasions where due to changes in MultiLogger and changes to the default labels you will need to use a text editor and delete the section from your configuration files.

```
[Input Storage Labels]
Location#1=Logger_ID,The Logger ID,true,false,0,0
Location#2=Year,The current year,true,false,0,0
Location#3=Julian_Day,The current day,true,false,0,0
Location#4=Time_HHMM,The current time,true,false,0,0
Location#5=Seconds,The current seconds,true,false,0,0
Location#6=Decimal_Day,The time formatted as Decimal Day,true,false,0,0
Location#7=Elapsed_Hours,Elapsed Hours Location,true,false,0,0
Location#8=Elapsed_Minutes,Elapsed Minutes Location,true,false,0,0
Location#9=Elapsed_Seconds,Elapsed Seconds Location,true,false,0,0
Location#10=Battery_Volts,The battery voltage,true,false,0,0
Location#11=Panel_Temp,The panel temperature,true,false,0,0
Location#12=Program_Mode,Program Mode,false,false,0,0
Location#13=Complete_Scans,Completed Scans,false,false,0,0
Location#14=ReadFlag,The take readings flag,false,false,0,0
Location#15=Flag(1),Flag Location 1,false,false,0,0
Location#16=Flag(2),Flag Location 2,false,false,0,0
Location#17=Flag(3),Flag Location 3,false,false,0,0
Location#18=Flag(4),Flag Location 4,false,false,0,0
Location#19=Flag(5),Flag Location 5,false,false,0,0
Location#20=Flag(6),Flag Location 6,false,false,0,0
Location#21=Flag(7),Flag Location 7,false,false,0,0
Location#22=Flag(8),Flag Location 8,false,false,0,0
```

```
Location#23=Reading_Loc,Reading Location,false,false,0,0
Location#24=GageType_Loc,Gage Type Location,false,false,0,0
Location#25=Channel,Mux Channel,false,false,0,0
Location#26=MuxNumber,Mux Number,false,false,0,0
Location#27=Read_Time,Length of program read,false,false,0,0
Location#28=Int_Counter,The Interval counter for log
intervals,false,false,0,0
Location#29=Int_Current,The Current Interval being
executed,false,false,0,0
Location#30=Int_Iterations,The Current Iterations being
executed,false,false,0,0
Location#31=UserLoc(1),User Location 1,false,false,0,0
Location#32=UserLoc(2),User Location 2,false,false,0,0
Location#33=UserLoc(3),User Location 3,false,false,0,0
Location#34=UserLoc(4),User Location 4,false,false,0,0
Location#35=UserLoc(5),User Location 5,false,false,0,0
Location#36=UserLoc(6),User Location 6,false,false,0,0
Location#37=UserLoc(7),User Location 7,false,false,0,0
Location#38=UserLoc(8),User Location 8,false,false,0,0
Location#39=UserLoc(9),User Location 9,false,false,0,0
Location#40=UserLoc(10),User Location 10,false,false,0,0
Location#41=UserLoc(11),User Location 11,false,false,0,0
Location#42=UserLoc(12),User Location 12,false,false,0,0
Location#43=UserLoc(13),User Location 13,false,false,0,0
Location#44=UserLoc(14),User Location 14,false,false,0,0
Location#45=UserLoc(15),User Location 15,false,false,0,0
Location#46=UserLoc(16),User Location 16,false,false,0,0
```

2. **DIM Declarations** – These are variables used internally by the MultiLogger program generator, but you may also add other declarations depending on your programming requirements. Keep in the mind that these variables DO NOT show up in the Text Monitor, nor will they even be available. DIM variables do not show up in the PUBLIC table of the CR1000, they are intended for use exclusively by the program.

   Note: These variables are particularly useful when constructing Processing Files or Alarm Type instruction files. For example, the ScratchLoc array is used to hold temporary results of some processing. The MuxReadingLoc is used to hold the measurements from Direct Connect Channels and Multiplexer Channels prior to their being moved to their final destination locations. Use the indexing characters ++ to denote the location for a specific channel. The indexing characters can also be followed by increments or decrements, using the + or – symbols respectively, followed by the value. For example MuxReadingLoc(+++1) will retrieve the value from the next channel, likewise MuxReadingLoc(++-1) will retrieve the value from the preceding channel.

   The format of this section is as follows:
   Name, Dimension (0 = single value, no array), Type (0=float, 1=single precision, 2=Boolean and 3=string)

```
[DIM Declarations]
Variable=ScratchLoc,16,0
Variable=MuxReadingLoc,48,0
Variable=sInBuf,255,3
Variable=sOutBuf,255,3
Variable=SDM1Loc,8
Variable=SDM2Loc,8
Variable=SDM3Loc,8
Variable=SDM4Loc,8
```

3. **CONST Declarations** – These are CONSTant declarations. MultiLogger does not presently use any of these declarations but they can be utilized in custom programming.

```
[CONST Declarations]
```

4. **Input Storage Assignments** – These are keyword based references to the default Input Storage Labels. This is important because label names can be changed which will break programming that refers to them. Use the keyword reference instead, if the label name is changed MultiLogger will still be able to find the correct label name for use in the program.

```
[Input Storage Assignments]
Assignment=mlLoggerID,1
Assignment=mlYear,2
Assignment=mlJulian,3
Assignment=mlTime,4
Assignment=mlSeconds,5
Assignment=mlDecimalDay,6
Assignment=mlElapsedHours,7
Assignment=mlElapsedMinutes,8
Assignment=mlElapsedSeconds,9
Assignment=mlBattery,10
Assignment=mlPanelTemp,11
Assignment=mlProgramMode,12
Assignment=mlCompleteScans,13
Assignment=mlReadFlag,14
Assignment=mlFlag1,15
Assignment=mlFlag2,16
Assignment=mlFlag3,17
Assignment=mlFlag4,18
Assignment=mlFlag5,19
Assignment=mlFlag6,20
Assignment=mlFlag7,21
Assignment=mlFlag8,22
Assignment=mlReading,23
Assignment=mlGageType,24
Assignment=mlChannel,25
Assignment=mlMuxNumber,26
Assignment=mlReadTime,27
Assignment=mlIntCount,28
Assignment=mlIntCurrent,29
Assignment=mlIntIterations,30
Assignment=mlUser1,31
Assignment=mlUser2,32
Assignment=mlUser3,33
Assignment=mlUser4,34
Assignment=mlUser5,35
Assignment=mlUser6,36
Assignment=mlUser7,37
Assignment=mlUser8,38
Assignment=mlUser9,39
Assignment=mlUser10,40
Assignment=mlUser11,41
Assignment=mlUser12,42
Assignment=mlUser13,43
Assignment=mlUser14,44
Assignment=mlUser15,45
Assignment=mlUser16,46
Assignment=Mux0Loc,47
```

An explanation for each of these assignments will be helpful to understand how to utilize the Input Storage Assignments list.

mlLoggerID – The location storing the Datalogger ID.
mlYear – The location storing the Year of the last program scan.
mlJulian – The location storing the Julian Day of the last program scan.
mlTime – The location storing the Time in HHMM format of the last program scan.
mlSeconds – The location storing the Seconds of the last program scan.
mlDecimalDay – The Decimal Day of the last program scan.
mlElapsedHours[1] – The location storing Elapsed Hours.
mlElapsedMinutes[1]– The location storing the Elapsed Minutes.
mlElapsedSeconds[1] – The location storing the Elapsed Seconds.
mlBattery – The location storing the battery voltage measurement.
mlPanelTemp – The location storing the panel temperature measurement.
mlProgramMode – The location storing the Program Mode setting.
mlCompleteScans – The location storing the number of completed program scans.
mlReadFlag – The status of the read flag, used to determine when to take measurements.
mlFlag1[2] – The location storing the status of Flag(1), where zero = low, non-zero = high.
mlFlag2[2] – The location storing the status of Flag(2), where zero = low, non-zero = high.
mlFlag3[2] – The location storing the status of Flag(3), where zero = low, non-zero = high.
mlFlag4[2] – The location storing the status of Flag(4), where zero = low, non-zero = high.
mlFlag5[2] – The location storing the status of Flag(5), where zero = low, non-zero = high.
mlFlag6[2] – The location storing the status of Flag(6), where zero = low, non-zero = high.
mlFlag7[2] – The location storing the status of Flag(7), where zero = low, non-zero = high.
mlFlag8[2] – The location storing the status of Flag(8), where zero = low, non-zero = high.
mlReading – The location storing the last reading obtained.
mlGageType – The location storing the gage type of the last reading obtained.
mlChannel – The current Channel.
mlMuxNumber – The current Mux Number (used for Test mode).
mlReadTime – The length of time required to obtain all measurements, in seconds.
mlIntCount – The current interval count (used for logarithmic interval timing).
mlIntCurrent – The current iterations count (used for logarithmic interval timing).
mlIntIterations – The current interval iterations (used for logarithmic interval timing).
mlUser1[3] – The location referenced to User Location 1.
mlUser2 – The location referenced to User Location 2.
mlUser3 – The location referenced to User Location 3.
mlUser4 – The location referenced to User Location 4.
mlUser5 – The location referenced to User Location 5.
mlUser6 – The location referenced to User Location 6.
mlUser7 – The location referenced to User Location 7.
mlUser8 – The location referenced to User Location 8.
mlUser9 – The location referenced to User Location 9.
mlUser10 – The location referenced to User Location 10.
mlUser11 – The location referenced to User Location 11.
mlUser12 – The location referenced to User Location 12.
mlUser13 – The location referenced to User Location 13.
mlUser14 – The location referenced to User Location 14.
mlUser15 – The location referenced to User Location 15.
mlUser16 – The location referenced to User Location 16.

Notes:
[1] These locations will only update when the Single Interval is set to Logarithmic Intervals.
[2] Flags5-8 are used by various MultiLogger functions relating to alarm functionality so their use should be avoided. Use Flags1-4 for user defined functions or control.
[3] When using the VWDSP avoid use of User Location 1, this is used to store the VW measurement quality value.

5. **Reserved MultiLogger keywords** – These refer to specific settings stored in the configuration file. For example, if you want to include a mathematical operator in a Processing File that requires the Gage Factor for the selected channel then include mlGageFactor in the Processing File.

   mlChannel – The destination location for the current channel.
   mlDataloggerID – The ID for the current datalogger.
   mlReadTime1 – The Read Time #1 entered on the MultiLogger Program tab.
   mlReadTime2 – The Read Time #2 entered on the MultiLogger Program tab.
   mlReadTime3 – The Read Time #3 entered on the MultiLogger Program tab.
   mlReadTime4 – The Read Time #4 entered on the MultiLogger Program tab.
   mlReadTime5 – The Read Time #5 entered on the MultiLogger Program tab.
   mlReadTime6 – The Read Time #6 entered on the MultiLogger Program tab.
   mlReadTime7 – The Read Time #7 entered on the MultiLogger Program tab.
   mlReadTime8 – The Read Time #8 entered on the MultiLogger Program tab.
   mlLogInterval1Length – The Log Interval #1 Length entered on the MultiLogger Program tab.
   mlLogInterval2Length – The Log Interval #2 Length entered on the MultiLogger Program tab.
   mlLogInterval3Length – The Log Interval #3 Length entered on the MultiLogger Program tab.
   mlLogInterval4Length – The Log Interval #4 Length entered on the MultiLogger Program tab.
   mlLogInterval5Length – The Log Interval #5 Length entered on the MultiLogger Program tab.
   mlLogInterval6Length – The Log Interval #6 Length entered on the MultiLogger Program tab.
   mlLogInterval7Length – The Log Interval #7 Length entered on the MultiLogger Program tab.
   mlLogInterval8Length – The Log Interval #8 Length entered on the MultiLogger Program tab.
   mlLogInterval1Iterations – The Log Interval #1 Iterations entered on the MultiLogger Program tab.
   mlLogInterval2Iterations – The Log Interval #2 Iterations entered on the MultiLogger Program tab.
   mlLogInterval3Iterations – The Log Interval #3 Iterations entered on the MultiLogger Program tab.
   mlLogInterval4Iterations – The Log Interval #4 Iterations entered on the MultiLogger Program tab.
   mlLogInterval5Iterations – The Log Interval #5 Iterations entered on the MultiLogger Program tab.
   mlLogInterval6Iterations – The Log Interval #6 Iterations entered on the MultiLogger Program tab.
   mlLogInterval7Iterations – The Log Interval #7 Iterations entered on the MultiLogger Program tab.
   mlLogInterval8Iterations – The Log Interval #8 Iterations entered on the MultiLogger Program tab.
   mlZeroReading – The Zero Reading for the current channel.
   mlGageFactor – The Gage Factor for the current channel.
   mlOffset – The Offset for the current channel.
   mlPolyCoefficientA – The Poly Coefficient A for the current channel.
   mlPolyCoefficientB – The Poly Coefficient B for the current channel.
   mlPolyCoefficientC – The Poly Coefficient C for the current channel.
   mlTempFactor – The Temperature Factor for the current channel.
   mlInitialTemp – The Initial Temperature for the current channel.
   mlLowLimit – The Alarm Low Limit for the current channel.
   mlHighLimit – The Alarm High Limit for the current channel.

***Examples***

Consider an example **Alarm Type | Rate of Change/High Level**

The instruction file is: **rateofchangehighlevel.cr1**

Load this file (found in the \CR1000 folder) into MLEditor or a text editor such as Notepad.

```
'Make sure its not first scan
if mlCompleteScan > 0 then

      'first get our rate
      ScratchLoc(1) = MuxReadingLoc(++) - mlChannel

      'check against rate (absolute values)
      if ABS(ScratchLoc(1)) > ABS(mlLowLimit) then Flag(8) = -1

endif

'check against high level
if MuxReadingLoc(++) > mlHighLimit then Flag(8) = -1
```

The first program line "`if mlCompleteScan > 0 then`" is used to make sure that we actually have a previous measurement to compare for the Rate of Change alarm. The mlCompleteScan location records the number of complete scans, at this point in the program it will be 0 for the first scan.

The second program line "`ScratchLoc(1) = MuxReadingLoc(++) - mlChannel`" will use a temporary location to store the results of subtracting the current reading in MuxReadingLoc(++) for this channel from the previously measured value stored in mlChannel. Keep in mind that mlChannel instructs the program generator to retrieve the actual label for that channel, for example channel #2 on multiplexer #2 will have a label of "Mux2CH(2)".

The third program line "`if ABS(ScratchLoc(1)) > ABS(mlLowLimit) then Flag(8) = -1`" will take an absolute value for the change in value and compare it to the absolute value of the mlLowLimit for the channel, if it's true then Flag(8) is set high.

The fourth program "`endif`" marks the end of the if statement to check whether it was the first scan or not.

The fifth program line "`if MuxReadingLoc(++) > mlHighLimit then Flag(8) = -1`" will compare the current reading in MuxReadingLoc(++) to the mlHighLimit for the channel, if it evaluates as higher then the alarm Flag(8) is set high.

Consider an example **Processing File | Convert 6 Channels to Load**

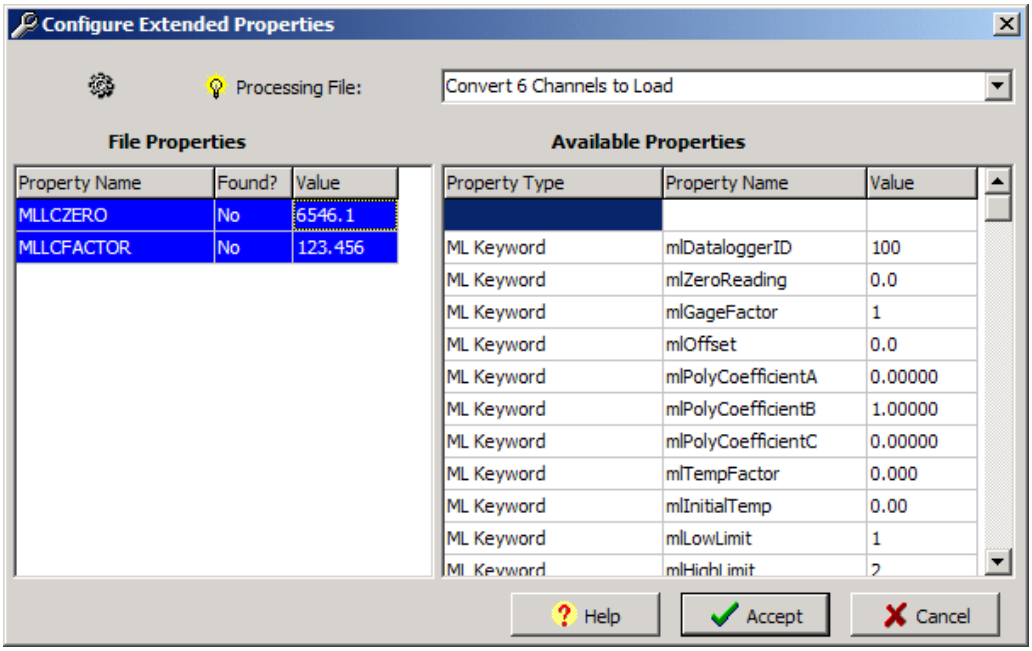
The instruction file is: **convert6channelstoload.cr1**

Load this file (original found in the \CR1000 folder) into MLEditor or a text editor such as Notepad.

```
'Convert 6 consecutive channel measurements to load
'Usually used to convert 6 readings of a load cell to load

'Get average of channel measurements
AvgSpa(ScratchLoc(1),6,MuxReadingLoc(++))

'Subtract Zero Reading (Extended Property to be user configured)
ScratchLoc(1) = ScratchLoc(1) - mlLCZero

'Multiply by Gage Factor (Extended Property to be user configured)
UserLoc(++) = ScratchLoc(1) * mlLCFactor
```

The first program line "`AvgSpa(ScratchLoc(1),6,MuxReadingLoc(++))`" will calculate the spatial average for the next 6 measurements beginning with the current channel. As such this Processing File should always be selected for the 1st channel of the 6 channels to be used for the calculations. For example if this Processing File is selected for channel #11 the MuxReadingLoc(++) will evaluate to MuxReadingLoc(11) and channels 11-16, MuxReadingLoc(11) – MuxReadingLoc(16), will be averaging for the subsequent calculations.

The second program file "`ScratchLoc(1) = ScratchLoc(1) - mlLCZero`" will subtract the value for **mlLCZero** from the spatial average calculated previously. The value **mlLCZero** must be entered using the Extended Properties form, shown at right.




The third program line "`UserLoc(++) = ScratchLoc(1) * mlLCFactor`" will multiply the result by **mlLCFactor** to produce load and then move the result to the correspond UserLoc. The value **mlLCFactor** must be entered using the Extended Properties form, shown above. The UserLoc destination will correspond to the channel selected, for example channel 11 result will be moved to UserLoc(11). Be sure to configure this location to be stored using the Program | Input Locations form, scroll down the list to find the User Locations.

## ***CR1000 Program Generation Flowchart***

The following flowchart details the order in which MultiLogger generates the CR1000.