



# PC400

---

Datalogger Support Software  
Version 4.10

# Campbell Scientific, Inc. Software End User License Agreement (EULA)

---

**COPYRIGHT:** This software is protected by United States copyright law and international copyright treaty provisions. This software may not be sold, included or redistributed in any other software, or altered in any way without prior written permission from Campbell Scientific. All copyright notices and labeling must be left intact.

**NOTICE OF AGREEMENT:** Please carefully read this EULA. By installing or using this software, you are agreeing to comply with the following terms and conditions. If you do not want to be bound by this EULA, you must promptly return the software, any copies, and accompanying documentation in its original packaging to Campbell Scientific or its representative.

**LICENSE FOR USE:** This software is protected by United States copyright law and international copyright treaty provisions. The installation and use of this software constitutes an agreement to abide by the provisions of this license agreement.

Campbell Scientific distributes this software free of charge to enable users to work with Campbell Scientific data acquisition equipment. This software may be freely copied; however, neither the software nor any of its files may be sold or included in any other software. The software may not be altered in any way without prior written permission from Campbell Scientific.

# Limited warranty

---

The following warranties are in effect for ninety (90) days from the date of shipment of the original purchase. These warranties are not extended by the installation of upgrades or patches offered free of charge:

Campbell Scientific warrants that the installation media on which the software is recorded and the documentation provided with it are free from physical defects in materials and workmanship under normal use. The warranty does not cover any installation media that has been damaged, lost, or abused. You are urged to make a backup copy (as set forth above) to protect your investment. Damaged or lost media is the sole responsibility of the licensee and will not be replaced by Campbell Scientific.

Campbell Scientific warrants that the software itself will perform substantially in accordance with the specifications set forth in the instruction manual when properly installed and used in a manner consistent with the published recommendations, including recommended system requirements. Campbell Scientific does not warrant that the software will meet licensee's requirements for use, or that the software or documentation are error free, or that the operation of the software will be uninterrupted.

Campbell Scientific will either replace or correct any software that does not perform substantially according to the specifications set forth in the instruction manual with a corrected copy of the software or corrective code. In the case of significant error in the installation media or documentation, Campbell Scientific will correct errors without charge by providing new media, addenda, or substitute pages. If Campbell Scientific is unable to replace defective media or documentation, or if it is unable to provide corrected software or corrected documentation within a reasonable time, it will either replace the software with a functionally similar program or refund the purchase price paid for the software.

All warranties of merchantability and fitness for a particular purpose are disclaimed and excluded. Campbell Scientific shall not in any case be liable for special, incidental, consequential, indirect, or other similar damages even if Campbell Scientific has been advised of the possibility of such damages. Campbell Scientific is not responsible for any costs incurred as a result of lost profits or revenue, loss of use of the software, loss of data, cost of re-creating lost data, the cost of any substitute program, telecommunication access costs, claims by any party other than licensee, or for other similar costs.

This warranty does not cover any software that has been altered or changed in any way by anyone other than Campbell Scientific. Campbell Scientific is not responsible for problems

caused by computer hardware, computer operating systems, or the use of Campbell Scientific's software with non-Campbell Scientific software.

Licensee's sole and exclusive remedy is set forth in this limited warranty. Campbell Scientific's aggregate liability arising from or relating to this agreement or the software or documentation (regardless of the form of action; e.g., contract, tort, computer malpractice, fraud and/or otherwise) is limited to the purchase price paid by the licensee.

# Table of contents

---

<b>1. PC400</b>	<b>1</b>
1.1 Who Should Use PC400?	1
1.2 What's new in Version 4?	1
1.2.1 Version 4.10	1
1.2.2 Version 4.9	1
1.2.3 Version 4.8	2
1.2.4 Version 4.7	2
1.2.5 Version 4.5	3
1.2.6 Version 4.4	3
1.2.7 Version 4.3	3
1.2.8 Version 4.2	3
1.2.9 Version 4.1	4
1.2.10 Version 4.0	4
<b>2. System Requirements</b>	<b>5</b>
<b>3. Installation, Operation, and Backup Procedures</b>	<b>5</b>
3.1 Installation	5
3.2 Command Line Arguments	6
3.3 PC400 Operation and Backup Procedures	6
3.3.1 PC400 Directory Structure	6
3.3.1.1 Program Directory	6
3.3.1.2 Working Directories	7
3.3.2 Backing up PC400	8
3.3.2.1 Performing a Backup	8
3.3.2.2 Restoring the Network from a Backup File	8
3.3.3 Loss of Computer Power	8
3.3.4 Program Crashes	9
<b>4. The PC400 Main Screen</b>	<b>9</b>
4.1 Clock/Program Tab	11
4.1.1 Current Datalogger Profile	12
4.1.2 Datalogger Clock	12
4.1.3 Datalogger Program	13

4.1.4 EZSetup Wizard .....	13
4.2 Monitor Data Tab .....	14
4.2.1 Monitoring Values .....	14
4.2.2 Monitoring and Controlling Ports and Flags .....	15
4.2.3 Options .....	17
4.3 Collect Data Tab .....	18
4.4 Pull-down Menus .....	19
4.4.1 View Menu .....	19
4.4.2 Datalogger Menu .....	20
4.4.2.1 Connect/Disconnect .....	20
4.4.2.2 Update Table Definitions .....	20
4.4.2.3 Station Status .....	21
4.4.2.4 File Control .....	23
4.4.2.5 Terminal Emulator .....	28
Datalogger Remote Terminal Mode .....	28
Comms Watch .....	28
Talk Through Mode .....	28
COM Port Feedback Test .....	29
Peripheral Device Test .....	29
4.4.3 Network Menu .....	29
4.4.3.1 Add/Delete/Edit/Rename Datalogger .....	29
4.4.3.2 Backup/Restore Network .....	30
4.4.3.3 Computer's Global PakBus Address .....	30
4.4.4 Tools Menu .....	30
4.4.4.1 Stand-alone Applications .....	30
4.4.4.2 Options .....	31
<b>5. Creating and Editing Datalogger Programs .....</b>	<b>31</b>
5.1 Short Cut .....	32
5.1.1 Creating a Program Using Short Cut .....	33
5.1.1.1 Step 1 – Create a New File or Open Existing File .....	33
5.1.1.2 Step 2 – Select Datalogger .....	33
5.1.1.3 Step 3 – Choose Sensors to Monitor .....	37
5.1.1.4 Step 4 – Set up Output Tables .....	44
5.1.1.5 Step 5 – Set up Advanced Outputs .....	46
5.1.1.6 Step 6 – Select Outputs .....	48
5.1.1.7 Step 7 – Generate the Program in the Format Required by the Datalogger .....	50

5.1.2 Short Cut Settings .....	51
5.1.2.1 Program Security .....	51
5.1.2.2 Datalogger ID .....	52
5.1.2.3 Power-up Settings .....	52
5.1.2.4 Select CR200 Compiler .....	52
5.1.2.5 Sensor Support .....	53
5.1.2.6 First Notch Frequency (fN1)/Integration .....	53
5.1.2.7 Font .....	53
5.1.2.8 Set Working Directory .....	53
5.1.2.9 Enable Creation of Custom Sensor Files .....	54
5.1.3 Editing Programs Created by Short Cut .....	54
5.1.4 New Sensor Files .....	54
5.1.5 Custom Sensor Files .....	54
5.2 CRBasic Editor .....	55
5.2.1 Inserting Instructions .....	56
5.2.1.1 Parameter Dialog Box .....	56
5.2.1.2 Right-Click Functionality .....	57
5.2.2 Toolbar .....	58
5.2.2.1 Compile .....	61
5.2.2.2 Compile, Save, and Send .....	61
5.2.2.3 Conditional Compile and Save .....	63
5.2.2.4 Templates .....	64
5.2.2.5 Program Navigation using BookMarks and GoTo .....	64
5.2.2.6 CRBasic Editor File Menu .....	65
5.2.2.7 CRBasic Editor Edit Menu .....	65
5.2.2.8 CRBasic Editor View Menu .....	66
Editor Preferences .....	66
Instruction Panel Preferences .....	68
Other Options .....	68
5.2.2.9 CRBasic Editor Tools Menu .....	69
Edit Instruction Categories .....	69
Constant Customization .....	70
Other Options .....	72
5.2.3 CRBasic Programming .....	73
5.2.3.1 Programming Sequence .....	73
5.2.3.2 Program Declarations .....	75
5.2.3.3 Mathematical Expressions .....	75


5.2.3.4 Measurement and Output Processing Instructions .....	76
5.2.3.5 Line Continuation .....	76
5.2.3.6 Inserting Comments into Program .....	76
5.2.3.7 Example Program .....	77
5.2.3.8 Data Tables .....	77
5.2.3.9 The Scan—Measurement Timing and Processing .....	80
5.2.3.10 Numerical Entries .....	80
5.2.3.11 Logical Expression Evaluation .....	81
What is True? .....	81
Expression Evaluation .....	81
Numeric Results of Expression Evaluation .....	82
5.2.3.12 Flags .....	82
5.2.3.13 Parameter Types .....	82
Expressions in Parameters .....	83
Arrays of Multipliers and Offsets for Sensor Calibration .....	84
5.2.3.14 Program Access to data Tables .....	84
<b>6. View Pro .....</b>	<b>85</b>
6.1 View Pro Toolbar .....	86
6.1.1 Selected Graph .....	87
6.2 Right-Click Menu .....	88
6.2.1 Selection Definition .....	89
6.2.2 Format Column .....	90
6.3 Opening Files .....	92
6.3.1 Data Files .....	92
6.3.1.1 File   Open .....	92
View   Autosize Columns .....	93
View   Show Full Header .....	93
View   Row Shading .....	93
View   Lock Timestamp Column .....	93
View   View File Information .....	94
View   Start Graphs Maximized .....	94
View   Array Definitions .....	94
View   Keep Data on Top .....	95
View   Keep Graph on Top .....	95
View   Background Color .....	95
View   Font .....	95



Window   Cascade .....	96
Window   Tile Horizontally .....	96
Window   Tile Vertically .....	96
6.3.2 Other Files .....	96
6.3.2.1 File   Open as Text .....	96
6.3.2.2 File   Open as Hex .....	96
6.3.2.3 Import CSV .....	96
6.4 Viewing Database Tables .....	99
6.4.1 File   View LoggerNet Database Table .....	99
6.4.2 Selecting a Database .....	99
6.4.3 Selecting a Table .....	104
6.4.4 Installing PostgreSQL ODBC Drivers .....	105
6.4.5 Installing Oracle ODBC Drivers .....	114
6.5 Data View Options .....	118
6.5.1 Line Graph .....	118
6.5.1.1 Common vs. Independent Y-Axes .....	123
6.5.1.2 Line Graph Options .....	123
6.5.1.3 Line Graph Trace Options—Display .....	125
6.5.1.4 Line Graph Trace Options—Y Axis .....	126
6.5.1.5 Line Graph Trace Options—Marks .....	126
6.5.2 Histogram .....	127
6.5.2.1 Histogram Options .....	131
6.5.2.2 Histogram Selection Properties—Display .....	131
6.5.2.3 Histogram Selection Properties—Y Axis .....	132
6.5.2.4 Histogram Selection Properties—X Axis .....	133
6.5.3 XY Plot .....	134
6.5.3.1 XY Plot Options—Visual Display .....	136
6.5.3.2 XY Plot Options—Y Axis .....	136
6.5.3.3 XY Plot Options—X Axis .....	137
6.5.3.4 XY Plot Series Options .....	138
6.5.4 Rainflow Histogram .....	139
6.5.4.1 Rainflow Histogram Options .....	142
6.5.4.2 Rainflow Histogram Selection Properties-Display .....	143
6.5.4.3 Rainflow Histogram Selection Properties-Axes .....	144
6.5.5 FFT .....	144
6.5.5.1 FFT Options .....	148
6.5.5.2 FFT Selection Properties-Display .....	148

6.5.5.3 FFT Selection Properties-Y Axis .....	150
6.5.5.4 FFT Selection Properties-X Axis .....	150
6.6 Printing .....	151
6.6.1 Printing Options .....	151
6.6.2 Print Preview .....	152
<b>7. Utilities .....</b>	<b>153</b>
7.1 CardConvert .....	153
7.1.1 Input/Output File Settings .....	153
7.1.2 Destination File Options .....	154
7.1.2.1 File Format .....	154
7.1.2.2 File Processing .....	155
7.1.2.3 File Naming .....	156
7.1.2.4 TOA5/TOB1 Format .....	157
7.1.3 Converting the File .....	157
7.1.3.1 Repairing/Converting Corrupted Files .....	157
7.1.4 Viewing a Converted File .....	158
7.1.5 Running CardConvert From a Command Line .....	158
7.2 Device Configuration Utility .....	159
7.2.1 Main DevConfig Screen .....	161
7.2.2 Downloading an Operating System .....	162
7.2.2.1 Manage OS Tab .....	162
7.2.2.2 Send OS Tab .....	163
7.2.3 Terminal Tab .....	166
7.2.4 The Unknown Device Type .....	167
7.2.5 Off-line Mode .....	168
7.2.6 Backing up and Restoring a Datalogger .....	168
7.2.7 Data Recovery .....	170
7.3 File Format Convert .....	171
7.3.1 Options .....	173
7.4 Split .....	174
<b>Appendix A. Glossary .....</b>	<b>175</b>
<b>Appendix B. Log Files .....</b>	<b>191</b>
B.1 Event Logging .....	191
B.1.1 Log Categories .....	191
B.1.2 Enabling Log Files .....	192

B.1.3 Log File Message Formats .....	192
B.1.3.1 General File Format Information .....	192
B.1.3.2 Transaction Log Format .....	193
B.1.3.3 Communications Status Log Format .....	226
B.1.3.4 Object State Log Format .....	229
<b>Appendix C. Split .....</b>	<b>231</b>
Split .....	232

**NOTE:** Most information specific to our older Edlog/Array-based dataloggers has been removed from this manual. For information on these dataloggers, please refer to the online help or a previous PC400 manual: [www.campbellsci.com/old-manuals](http://www.campbellsci.com/old-manuals) .

# 1. PC400

---

PC400 is Campbell Scientific's free software package that supports Campbell Scientific dataloggers. It allows you to communicate with your datalogger network over a direct connection, telephone modem, Internet connection, RF link, or MD9 link. Programming tools are included for Campbell Scientific's PakBus dataloggers, as well as our older mixed-array dataloggers. A manual data collection tool is available, and once data is collected from the datalogger it can be analyzed using the View Pro file viewer or processed further using the Split report generator.

## 1.1 Who Should Use PC400?

PC400 is designed for users who have small- to medium-sized datalogger networks. Some users find that they do not need all of the functionality offered in our LoggerNet software. PC400 was created to support those users. PC400 may fit your needs if:

You require only one level of communication to any one datalogger in your network. For instance, PC400 supports phone modem communication or RF communication, but it does not support phone-to-RF communication links (two levels).

You will be collecting data from your datalogger(s) manually (scheduled data collection is not supported in PC400).

## 1.2 What's new in Version 4?

### 1.2.1 Version 4.10

Added support for the CR1000Xe datalogger. Miscellaneous performance improvements and bug fixes.

### 1.2.2 Version 4.9

Skipped

## 1.2.3 Version 4.8

One of the main efforts in the development of PC400 4.8 has been to add the CR350 Series datalogger as a unique device. Previously, the CR350 was included as part of the CR300Series. Any existing CR350 Series dataloggers that were created using the CR300Series option can continue to be used as is.

Automatic software update checks are now a global feature across all Campbell Scientific applications that support this function. This means automatic software update checks may be activated or deactivated upon initial software installation and then the setting will be set the same (on or off) for all other Campbell Scientific software applications that support automatic updates. Automatic update checks may be turned on or off at any time using the Software Updater **File** menu.

## 1.2.4 Version 4.7

One of the main efforts in the development of PC400 4.7 has been support for the new GRANITE Data Logger Modules and the new CR350 datalogger.

In addition, a Neighbor PakBus Address field has been added to the Datalogger Settings screen in the EZSetup Wizard. This allows you to specify the neighbor you will go through to connect to your datalogger. For example, this can be used to enter the PakBus address of a Konect PakBus router.

Also, a TCP Password field has been added to the Security Settings screen in the EZSetup Wizard to control IP access to a datalogger.

A Software Updater has been added to PC400 and Device Configuration Utility that will automatically notify you when a new release is available. **New Version** will appear in the menu bar when PC400 or Device Configuration is opened. Click **New Version** to display the new version available. Click on the new version to be taken to the Software Updater and update your software.

Users of Campbell Scientific's LNDB database software who are using a PostgreSQL or Oracle database can now access the data directly from the database with View Pro. In addition, View Pro now allows you to set graphs to start maximized, set default settings for all new line graphs, and jump to the beginning or end of a line graph.

## 1.2.5 Version 4.5

The main effort in the development of version 4.5 has been support for the new CR1000X Series datalogger. In addition, the Short Cut user interface has been redesigned in an effort to make it more intuitive to users.

**NOTE:** Beginning with this version of PC400, Windows XP and Windows Vista are no longer supported.

## 1.2.6 Version 4.4

The main effort in the development of version 4.4 has been support for the new CR300 Series datalogger. The ability to change the number of rows and columns was added to the Monitor Data tab. The data file viewer was upgraded from View to View Pro.

## 1.2.7 Version 4.3

The main effort in the development of version 4.3 has been support for the new CR6 Series datalogger. Support was also added for the CRVW Series Vibrating Wire Recording Interface.

## 1.2.8 Version 4.2

Starting in version 4.2, PC400 now supports display and input of Unicode characters/strings in many areas of the product. Unicode is a universal system for encoding characters. It allows PC400 to display characters in the same way across multiple languages and countries. To support Unicode, an Insert Symbol dialog box has been added to the CRBasic Editor. This allows you to insert Unicode symbols into your CRBasic program for use in Strings and Units declarations.

PakBus Encryption is now supported for communication between PC400 and CR1000, CR3000, and CR800 series dataloggers. Note that the datalogger must be running OS 26 or later in order for PakBus Encryption to be used. A PakBus Encryption Key must be entered in both the datalogger's device settings and PC400's EZSetup Wizard. AES-128 encryption is used.

Boolean values displayed on the Monitor Data tab now have an LED icon next to them to allow for easy toggle.

A Constant Customization feature has been added to the CRBasic Editor. This allows you to define values for one or more constants in a program prior to performing a conditional compile. The constants can be set up with an edit box, a spin box field for selecting/entering a value, or with a list box. A step increase/decrease can be defined for the spin box, as well as maximum and minimum values.

The CRBasic Editor now allows you to Save and Open Display Settings. Display settings affect the look and feel of the CRBasic Editor. This includes font and background, as well as syntax highlighting.

View has a new View Record option in the right-click menu that can be used to view an entire record in a new window.

## 1.2.9 Version 4.1

Starting in version 4.1 the CRBasic Editor has the capability to open a read-only copy of any file. This gives you the ability to open multiple copies of a program and examine multiple areas of a very large program at the same time. You can also now continue an instruction onto multiple lines by placing the line continuation indicator (a single space followed by an underscore "\_") at the end of the each line that is to be continued. Also, bookmarks in a CRBasic program are now persistent from session to session.

The Device Configuration Utility has a new off-line mode which allows you to look at the settings for a certain device type without actually being connected to a device.

The ability to lock the timestamp column on the left of the data file has been added to View. This keeps the timestamp visible as you scroll through columns of data.

Split has a new "Time Sync to First Record" option that can be used with the time-sync function to avoid blank lines at the start of the output file. Also, a range of time values rather than a single time can now be entered in a Split Copy Condition.


CardConvert can now be run from a command line without user interaction.

An option to provide feedback on PC400 has been added to the PC400 Toolbar's Help menu.

## 1.2.10 Version 4.0

Version 4.0 of PC400 includes a new data file viewer, View. View maintains the ease of use of our previous data file viewer with enhanced capabilities, including the ability to view an unlimited number of traces on a graph.

You also now have the ability to launch the Device Configuration Utility from PC400's toolbar.

**NOTE:** Most information specific to our older Edlog/Array-based dataloggers has been removed from this manual. For information on these dataloggers, please refer to the online help or a previous PC400 manual: [www.campbellsci.com/old-manuals](http://www.campbellsci.com/old-manuals) .

## 2. System Requirements

---

PC400 runs on Windows 10 and Windows 11. PC400 runs on both 32-bit and 64-bit versions of these operating systems. In addition, TCP/IP service must be installed and enabled on the computer for PC400 to run.

**NOTE:** Contact your IT department for help with enabling TCP/IP service.

## 3. Installation, Operation, and Backup Procedures

---

3.1 Installation .....	5
3.2 Command Line Arguments .....	6
3.3 PC400 Operation and Backup Procedures .....	6
3.3.1 PC400 Directory Structure .....	6
3.3.2 Backing up PC400 .....	8
3.3.3 Loss of Computer Power .....	8
3.3.4 Program Crashes .....	9

### 3.1 Installation

Run the executable file, PC400\_version.exe, to begin the installation. Follow the prompts on the screen to complete the installation.

A shortcut to launch PC400 is added to your computer's Start menu. If the default directories are used, PC400 executable files and help files are placed in the C:\Program Files (x86)\CampbellSci\PC400 directory.

Working directories will also be created under C:\Campbellsci for PC400 configurations and data files, user programs, and settings for the accessory applications and utilities.



## 3.2 Command Line Arguments

Command line arguments allow you to change PC400's default behavior when it is started from a shortcut.

Currently, there are two command line arguments:

**/WorkDir**      Sets the working directory to something other than the default. Usage:

```
"C:\Program Files (x86)\CampbellSci\PC400\PC400.exe"  
/WorkDir=C:\CampbellSci\test
```

where "C:\Program Files (x86)\CampbellSci\PC400\PC400.exe" is the directory and filename for the PC400 executable and C:\CampbellSci\test is the working directory to be used. Note that if there are multiple-word directories in the directory string, the entire string must be enclosed in quotation marks. For example, /WorkDir="C:\CampbellSci\Working Dir".

**/IPPORT=XXXXX**      Causes the server to use port XXXXX for TCP/IP communications with clients. This is useful if some other software is using PC400's default port of 6789. Usage:

```
"C:\Program Files (x86)\CampbellSci\PC400\PC400.exe" /ipport=12345
```

## 3.3 PC400 Operation and Backup Procedures

This section describes some of the concepts and procedures recommended for routine operation and security of the PC400 software. If software and computer systems were perfect this section would not be necessary. However, since this software is required to run with predictable results in the real world on real computers, the following guidelines and procedures will be helpful in minimizing possible problems that may occur.

3.3.1 PC400 Directory Structure .....	6
3.3.2 Backing up PC400 .....	8
3.3.3 Loss of Computer Power .....	8
3.3.4 Program Crashes .....	9

### 3.3.1 PC400 Directory Structure

#### 3.3.1.1 Program Directory

As described in the installation procedures, all of the files for program execution are stored in the C:\Program Files (x86)\CampbellSci\PC400 directory. This includes the executables, DLLs, and

most of the application help files. This directory does not need to be included in back up efforts. PC400 and its applications rely on registry entries to run correctly; therefore, any restoration of the program should be done by reinstalling the software.

### 3.3.1.2 Working Directories

Each major application keeps its own working directory. The working directory holds the user files created by the application, as well as configuration and initialization (\*.INI) files. Because of this working directory scheme, when you use File | Open in Short Cut, CRBasic Editor, etc., you may find yourself in an unfamiliar directory and may have to navigate to a different directory to find existing data files, datalogger programs, etc.

This scheme was implemented because we use the underlying tools and many of the applications (the server itself, library files, datalogger program editors, etc.) in a number of different products. By providing a common working directory for each major application, we hope to make it easier to keep track of files and information as you move from one product to another.

By default, the files that you create in each of the applications will be stored in their respective folders in the working directory. You can override that default and store the files in a different location. Each application “remembers” the last directory in which a file was saved and will default to that directory until a different directory is selected.

Note that most all applications have one or more sub-directories in which configuration files are saved.

**Lib directory** – The Lib directory is a library directory for several of the PC400 applications. The **Compilers** folder holds all of the compilers for the CRBasic Editor, except for the CR200 compilers, which are stored in the **CR200Compilers** folder. The **CRBasicDefFiles** folder holds the definition files and help files for all dataloggers supported by the CRBasic Editor. The definition files are the files which provide the unique instructions and parameters for each datalogger. The **DevConfigLib** folder contains the files for each datalogger that will be used by the Device Configuration Utility.

**PC400 directory** – The ASCII data files that are saved to disk as a result of data collection from the dataloggers are stored to the PC400 directory with a \*.dat extension. The **Logs** directory holds the logs that are created when communication takes place between the server and client applications, and the server and the dataloggers. These logs are used to help troubleshoot communication problems. The **Sys** directory holds the network map description (CsiLgrNet.xml) and the binary data cache. (The data cache is a repository for the data which is collected from the dataloggers by the server, and which each client application accesses when processing that data.)

## 3.3.2 Backing up PC400

As with any computer system that contains important information, the data stored in the LoggerNet working directory should be backed up to a secure archive on a regular basis. This is a prudent measure in case a file becomes corrupted, the hard disk crashes, or the computer suffers some other hardware failure that prevents access to the stored data on the disk.

The maximum interval for backing up data files depends primarily on the amount of data maintained in the datalogger memory. The datalogger's final storage is configured as ring memory that will overwrite itself once the storage area or table is full. If the data is backed up more often than the oldest records in the datalogger are overwritten, a complete data record can still be maintained by restoring the data from the backup and then re-collecting the newest records from the datalogger.

### 3.3.2.1 Performing a Backup

PC400 provides a simple way to back up the network map, the PC400 data cache, and the initialization files for each of the applications. The network map will restore all settings and data collection pointers for the dataloggers and other devices in the network. The data cache is the binary database which contains the collected data from the datalogger. Initialization files store settings such as window size and position, configuration of the data display, etc.

From the PC400 menu, choose **Network > Backup/Restore Network**, and then press **Backup**.

The backup file is named PC400.bkp and is stored in the C:\CampbellSci\PC400 directory (if you installed PC400 using the default directory structure). You can, however, provide a different file name if desired.

### 3.3.2.2 Restoring the Network from a Backup File

To restore a network from a backup file, choose **Network > Backup/Restore Network**. Select the \*.bkp file that contains the network configuration you want to restore, and press **Restore**. Note that this process DOES NOT append to the existing network — the existing network will be overwritten when the restore is performed.

## 3.3.3 Loss of Computer Power

The PC400 communications server writes to several files in the \SYS directory during normal operations. The most critical files are the data cache table files and the network configuration files. The data cache files contain all of the data that has been collected from the dataloggers by the PC400 server. These files are kept open (or active) as long as data is being stored to the file.

The configuration files contain information about each device in the datalogger network, including device settings and other parameters. These files are written to frequently to make sure that they reflect the current state and configuration of each device. The configuration files are only opened as needed.

If computer system power is lost while the PC400 server is writing data to the active files, the files can become corrupted, making the files inaccessible to the server.

While loss of power won't always cause a file problem, having files backed up as described above will allow you to recover if a problem occurs. If a file does get corrupted, all of the server's working files need to be restored from backup to maintain the synchronization in the server state.

### **3.3.4 Program Crashes**

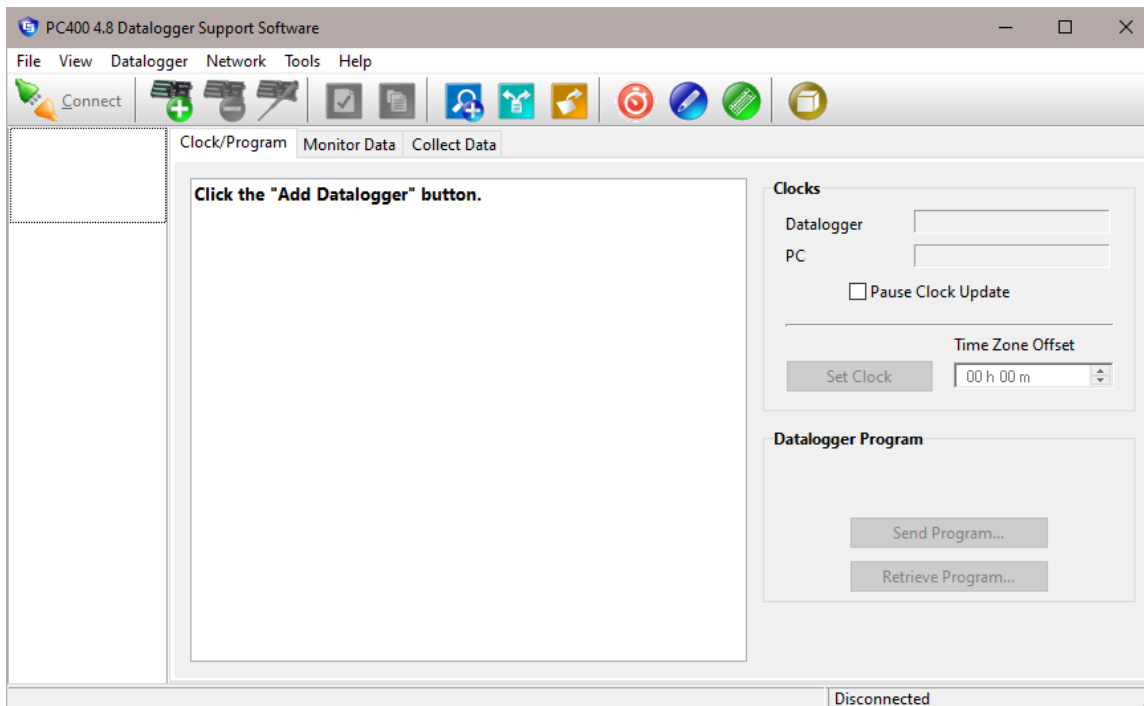
If the communication server crashes, there is a possibility that files can be corrupted (note, however, that corruption is much less likely with a program crash than during a power loss, since the computer operating system remains in control and can close the files left open by the failed program). If, after a program crash, the server does not run properly, you may need to restore the data from backup.

If you have problems restarting the PC400 server after a program crash or it crashes as soon as it starts, make sure that the PC400 server has not left a process running. You can check this by going to the Windows Task Manager and selecting the Process tab. In the list of processes look for the Toolbar or one of the client applications. If one of these processes exists but the Toolbar is not running, select this process and click End Process; you will be asked to confirm the end process.

## **4. The PC400 Main Screen**

---

To start PC400 go to the start menu of the computer and select PC400. You can alternatively use the shortcut on the desktop if you elected to create one during the installation process.



PC400 offers an integrated main screen, with three tabs for basic communications functions (**Clock/Program**, **Monitor Data**, and **Collect Data**), and buttons from which to launch standalone applications to work with data files or create datalogger programs.

Setting up and configuring PC400 to communicate with dataloggers is done with the EZSetup Wizard. This wizard appears automatically the first time you run PC400. To add additional dataloggers, click the **Add** button on the main toolbar to bring up the EZSetup Wizard again. Editing existing datalogger and communications settings is also done through this wizard, via the **Edit** button.

PC400 supports the CRBasic series of dataloggers, including the CR1000X, CR6 series, CR350, CR300 series, CR1000, CR800 series, CR3000, CR5000, CR9000, and CR200/205. PC400 also supports the Edlog dataloggers (CR500, CR510, CR10, CR10X, 21X, CR23X, and CR7 dataloggers) using any of their mixed-array, table-data, or PakBus operating systems.

PC400 supports one level of communication to any one datalogger in your network. For instance, PC400 supports phone modem communication or RF communication, but it does not support phone-to-RF communication links (two levels).

In order to be easy to use, PC400 relies on user-attended communications. It does not provide for automated scheduled data collection or automated clock checks. It also doesn't support remote connections from other PCs.

In addition to the communications functions, PC400 provides Split, View Pro, and CardConvert for working with data files, Short Cut for generating programs, and Edlog and CRBasic Editor for

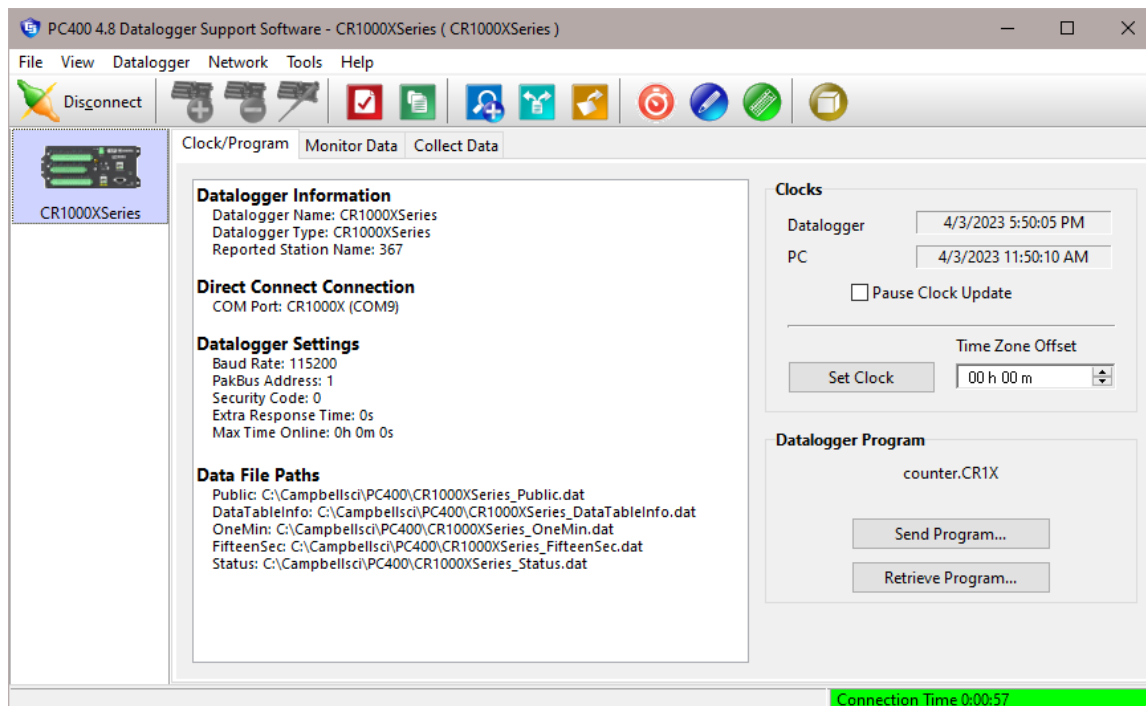
more advanced datalogger program editing. These tools are accessed via the buttons on the main screen or the pull-down menu selections under the Tools menu item.



Help for each application is available from the Help menu item, or by moving the focus to a control (by clicking on or tabbing to a control) and pressing F1.

To exit PC400, either click the [X] in the upper right hand corner of the main screen, or select **Exit** under the **File** menu.

## 4.1 Clock/Program Tab

When PC400 is opened, the main window is displayed with the Clock/Program tab active. Your datalogger stations will be represented on the left side of the window. (If no dataloggers are shown, you will need to configure your station(s) for communication using the EZSetup Wizard.). The top right portion of the main window shows buttons that, when pressed, launch the applications in PC400.




To communicate with a datalogger, on the Clock/Program tab select the datalogger with the mouse and press the connect button (  ). Once communication is established, the icon on this button will change to  which can then be used to disconnect from the datalogger. When communication is established, the two remaining tabs, Monitor Data and Collect Data, will become available. Communication will also be indicated by the update of the **Datalogger Clock** fields and the status indicator on the bottom right of the window. When this status indicator is Green, PC400 is communicating with the datalogger. A timer will be displayed, indicating the


amount of time PC400 has been communicating with the station. When the status indicator is Red, communication has failed. When it is Yellow, an attempt to communicate with the datalogger is in progress.

## 4.1.1 Current Datalogger Profile


The left side of the Clock/Program tab lists all of the dataloggers that have been set up for communication with PC400. When a datalogger is selected (evidenced by a blue box surrounding the picture of the datalogger), summary information for that datalogger will be displayed.

Press the connect button (  ) to establish the communication with the selected datalogger. If your system is not set up properly and communication cannot be established, an error message will be displayed. Ensure that all cables are secure, power is supplied to the datalogger, the proper COM port is selected, and no other software or device is using the COM port.

Press the add button (  ) to open the [EZSetup Wizard](#) (p. 13). This wizard will walk you through setting up communication with a new datalogger station.

A datalogger can be deleted from the Clock/Program tab. Highlight the datalogger you want to delete with the mouse pointer, and then press the delete button (  ). You will be asked to confirm the deletion of the datalogger.

The name of a datalogger can be changed to a new name. Select the desired datalogger with the mouse pointer, and then select **Network | Rename Datalogger**. The datalogger's name field will be highlighted; type in a new name over the old.

The current communication settings for a datalogger can be changed by highlighting that datalogger and pressing the edit button (  ). This will open the EZSetup wizard. If you know the window that contains the setting you want to change, you can select that window from the **Progress** portion of the wizard and jump to that window immediately, without going through the preceding steps.

## 4.1.2 Datalogger Clock

The Datalogger Clock portion of the Clock/Program tab displays the current time and date of your datalogger and the current time and date of your computer. You can synchronize the datalogger clock with the computer clock by pressing the **Set Clock** button. You can stop the clock updates by selecting the **Pause Clock Update** check box. Pausing the clock update may improve performance on slower communication links, because fewer commands will be sent to the datalogger from the software. Note, however, that CR10(X), CR500/510, CR23X, CR7, and 21X

dataloggers will time out and drop the communications link if no communication takes place with the datalogger for 40 seconds.

The datalogger clock can be offset from the computer clock in one hour increments, up to plus or minus 23 hours. Use the up arrow or down arrow to select an offset, in hours, (or type in a new value over the old), then press Set Clock to implement the change.

## 4.1.3 Datalogger Program

This portion of the Clock/Program tab indicates the name of the program running in the datalogger and is used to send a new program to the datalogger. To send a new program, press the **Send Program** button. A dialog box will be displayed. Search for and highlight the program you want to transfer to the datalogger and choose **Open**. A message will be displayed asking you to confirm the program send operation.

The program running in a datalogger can be retrieved and stored to a file on the PC by pressing the **Retrieve Program** button. Note that this option is not available for CR200 dataloggers, because the running program is a compiled binary image.


CR10X, CR510, CR23X, CR10, CR7, and 21X dataloggers (with the default, mixed-array OS) will also have an **Associate Program** button. When pressed, this button displays a browse dialog box from which to select a DLD file to associate with the datalogger. PC400 will assign input location labels from the DLD file displayed in this field to the values on the Monitor Data tab.

**WARNING:** Downloading a new program to the datalogger can erase stored data!

The output tables in table-based dataloggers will be reset to 0 records (e.g., data erased). The data in mixed-array dataloggers will be erased if memory reallocation is required to accommodate the new program.

You should always collect any uncollected data from the datalogger before initiating a program transfer.

## 4.1.4 EZSetup Wizard

PC400 simplifies the task of setting up your datalogger by providing a wizard that walks you through the steps required to define the datalogger type, its settings, and its communication link settings. The EZSetup Wizard is launched automatically the first time PC400 is opened. If you close the wizard and want to add a new station, press the add button (  ) from PC400's main window. The EZSetup Wizard's Introduction window is displayed.



You can back up through the wizard screens by pressing **Previous**.

The Monitor Data tab displays the current measurement values (i.e., input locations or public variables) and final storage data for the datalogger. This window can also be used to monitor or set ports and flags in the datalogger.

[illegible]

When the Monitor Data tab is first opened for a datalogger, values from the Public or Input Location table are displayed. Final storage table values can also be added to the Monitor. Press

the **Add** button to display a window with a list of all tables in the datalogger program. When a table is highlighted, the data values for that table are displayed in the field on the right. Choose one or more values from the list and press the **Paste** button to add them to the Monitor. You can also add values to the Monitor by dragging and dropping them to the desired location.

**NOTE:** For mixed-array dataloggers, if you see 28 default input locations listed in the monitor and do not see a list of the tables when the **Add** button is pressed, you will need to associate a program with the datalogger. This is accomplished by using the **Associate Program** button on the Clock/Program tab.

One or more values can be removed from the Monitor Data tab by highlighting the value(s) and pressing the **Delete** button. Values can be moved to different cells by dragging and dropping them.

The values of input locations (or variables) and the state of ports and flags can be changed from the Monitor Data tab. To edit an input location value, double-click the displayed value, enter the new value, and press the **Enter** key. When a value is enabled for editing, if the **ESC** key is pressed, the change will be canceled. The state of a port or flag can be changed by clicking the LED icon to the right of the field value. A black LED indicates that the port or flag is low; green indicates that it is high.

The font size on the Monitor Data tab cannot be changed directly. However, you can drag a corner of the PC400 window to resize it. This will also cause the Monitor Data tab font to be resized to correspond to the new window size.

## 4.2.2 Monitoring and Controlling Ports and Flags

When the Port/Flag button is selected, a window appears displaying the ports and flags in the datalogger.

The state of a port or flag can be changed by clicking the LED icon to the left of the field label. A black LED indicates that the port or flag is low; green indicates that it is high. Custom labels can be assigned to the ports and flags by double-clicking within the label field and typing in a new label.

Program variables that are declared as Boolean can also be placed on this display, for dataloggers that support data types. For these dataloggers, an **Add** button is available that, when pressed, lists all of the tables in the datalogger. When a table is highlighted on the left side of the window, any variables that are declared as Boolean in the program will be displayed on the right side of the window.

Our current dataloggers do not have predefined flags. The first time a program is sent to the datalogger, PC400 will look for a Public array with the name of Flag in the program. If a Flag array

is found, the declared flags will be added to the Ports and Flags dialog box. The number of flags that will be added is limited by the number of cells available on the Ports and Flags display. Ports that can be toggled from this display will be displayed in the first column and the remaining cells will be available to display flags and other Boolean values in the program.

To return the Ports and Flags display to its original state, press the **Defaults** button. This will reset all labels to their original names, update the number of flags based on the currently running program, and remove any Boolean values placed on the screen.

### Right Click Menu Options

Pressing the right mouse button while in a name field in the Ports and Flags window will bring up a short cut menu with the following options:

**Add** - Adds a new Boolean value to the window.

**Delete** - Deletes the selected value.

**Rename** - Allows you to rename the selected value.

**Save Configuration**- Saves the Ports and Flags configuration to a file that can be loaded in the future.

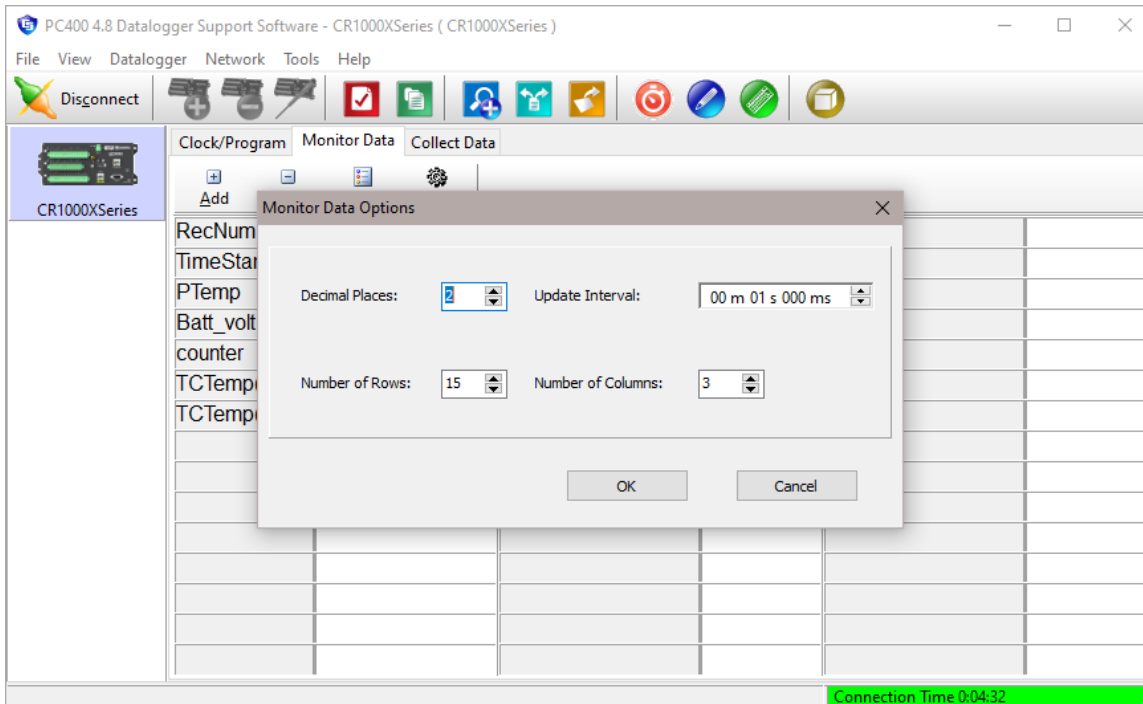
**Load Configuration**- Loads a previously saved Ports and Flags configuration file.

**NOTE:** If a Flag array has been declared and subsequently aliased, it will no longer show up on the Ports and Flags display by default. However, if it has been declared as Boolean, it may still be added manually.

**NOTE:** A control port must first be configured for output in the datalogger program before it can be toggled on or off. Consequently, if you select a port and it doesn't appear to change, your program may not have the port configured for output (refer to your datalogger operator's manual).

**NOTE:** A Boolean variable is a variable that can have one of two states: high/low, off/on, -1/0, true/false. Variables can be declared as Boolean with the Public or Dim statement.

## 4.2.3 Options



### Decimal Places

The number of decimal places displayed for the values on the Monitor Data tab is controlled by the **Decimal Places** field. Use the up and down arrows to the right of the field to increase or decrease the number of decimal places, or type a value directly into the field.

### Update Interval

For mixed-array dataloggers, input locations will be updated based on the interval specified in the **Update Interval** field. Final Storage Locations may also be viewed on the Monitor Data tab, but they will be updated only when data is collected from the datalogger.

For table-based dataloggers, both public variables and final storage data will be updated based upon the interval specified in the **Update Interval** field.

**NOTE:** Measurement and final storage values can be updated on the Monitor Data tab no more quickly than they are updated in the datalogger program.

### Number of Rows

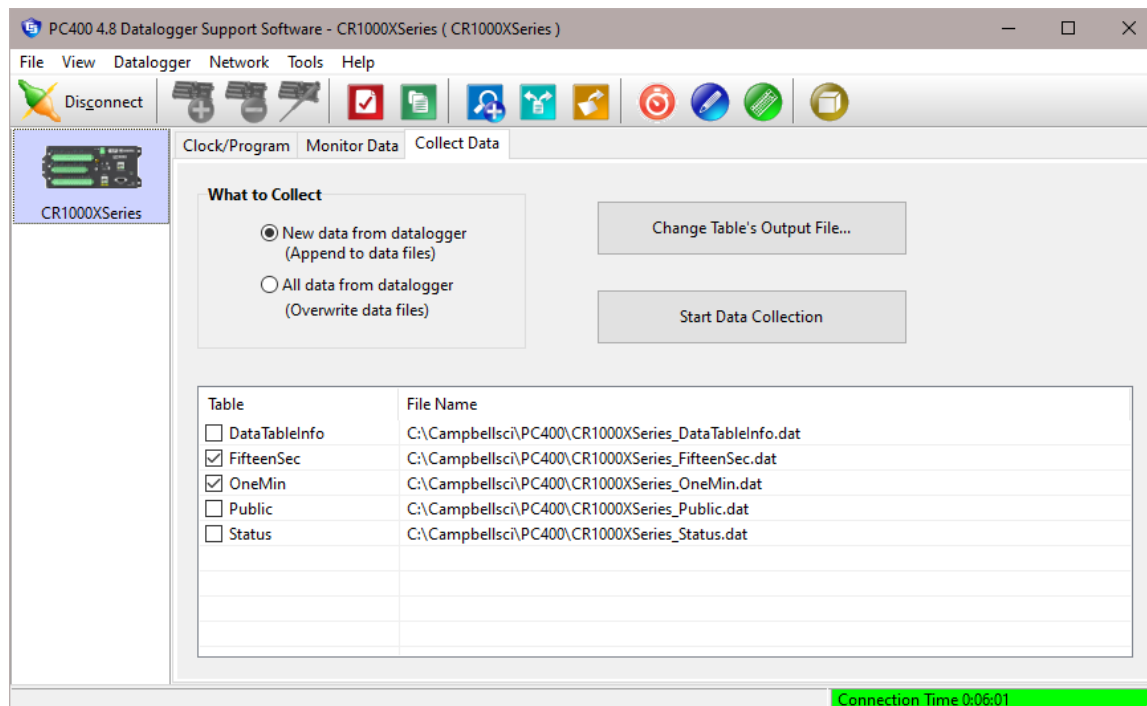
Determines the number of rows viewed. Use the arrows to the right of the field, or type in a number directly, to specify this value. The maximum number of rows is 100.

## Number of Columns

Determines the number of columns viewed. Use the arrows to the right of the field, or type in a number directly, to specify this value. The maximum number of columns is 10.

## 4.3 Collect Data Tab

From the Collect Data tab you can choose what data to collect and where to store the retrieved data.



## Data Collection

Pressing the **Start Data Collection** button initiates the retrieval of stored data from the datalogger for all tables or final storage areas that have their check box selected. The Progress indicator provides a visual cue that data is being collected successfully. You can stop the data collection process at any time by pressing Cancel on the Progress indicator.

## What to Collect

There are two options for data collection. Choosing **New data from datalogger** collects only the data stored since the last data collection and appends this new data to the end of the existing file. Choosing **All data from datalogger** collects all of the data in the selected table or final storage area and overwrites (i.e., replaces) the existing file.

## Tables/Final Storage Area to Collect

This information is used to select what data to collect from the datalogger. Different fields will be shown for different datalogger types.

Mixed-array dataloggers store their data in final storage memory. The final storage memory for some dataloggers can be configured for two final storage areas. If the datalogger type you are connected to can be configured for two final storage areas, you will see two check boxes: final storage area 1 and final storage area 2 (otherwise, you will see only final storage area 1). Select the check box for the final storage area(s) that you want to collect. A default is given for the file name and the directory in which the file will be stored, but these can be changed by selecting the button to the right of the field containing the file name.

Table-based dataloggers store their data in multiple tables within datalogger memory. If the datalogger type you are connected to is a table-based datalogger, you will see a list of the tables that are available for collection. Select the check box to the left of the table name for the tables that you wish to collect. You can change the directory or name under which the file is stored by highlighting a table and pressing the **Change Table's Output File** button or by double-clicking the table name. The default file extension is \*.dat. This can be changed to cxv, txt, or prn by choosing the option from the Save As Type list box

## Scheduled Data Collection

PC400 supports only manual data collection. If you need more advanced data collection features, such as automatically collecting data on a predefined schedule, consider upgrading to LoggerNet.

# 4.4 Pull-down Menus

Access to almost all of the buttons and tabs on PC400's screens is also available via pull-down menus. Many of these are described in detail in other sections.

## 4.4.1 View Menu

PC400 can display the user interface component text (for buttons, dialog boxes, etc.) in an alternate language if a separate language package has been installed. If a language package is installed on your machine, the View menu will have a Languages entry and you will see the language in the subsequent pick list. When a new language is chosen, the main PC400 user interface will immediately reflect the change. Opened auxiliary applications will not reflect the change until they are closed and reopened.

The main PC400 user interface and most auxiliary applications are capable of displaying text for alternate languages. However, Split and Edlog do not have alternate language support at this time.



**NOTE:** Available language packages are provided by Campbell Scientific's international representatives or on the CSI web site. They are not included in a standard PC400 installation.

## 4.4.2 Datalogger Menu

4.4.2.1 Connect/Disconnect .....	20
4.4.2.2 Update Table Definitions .....	20
4.4.2.3 Station Status .....	21
4.4.2.4 File Control .....	23
4.4.2.5 Terminal Emulator .....	28

### 4.4.2.1 Connect/Disconnect

When PC400 is opened, the main window is displayed with the Clock/Program tab active. Your datalogger stations will be represented on the left side of the window. (If no dataloggers are shown, you will need to configure your station(s) for communication using the EZSetup Wizard.).

To communicate with a datalogger, on the Clock/Program tab select the datalogger with the mouse and choose **Connect** from the Datalogger Menu. Once communication is established, the menu item will change to **Disconnect**, which can be used to disconnect from the datalogger. (The Connect and Disconnect functions are also available from the toolbar. Press the connect button (  ) to connect to a datalogger. Once communication is established, the icon on this button will change to  , which can be used to disconnect from the datalogger.)

Communication will be indicated by the update of the **Datalogger Clock** fields and the status indicator on the bottom right of the window. When this status indicator is Green, PC400 is communicating with the datalogger. A timer will be displayed, indicating the amount of time PC400 has been communicating with the station. When the status indicator is Red, communication has failed. When it is Yellow, an attempt to communicate with the datalogger is in progress.

### 4.4.2.2 Update Table Definitions

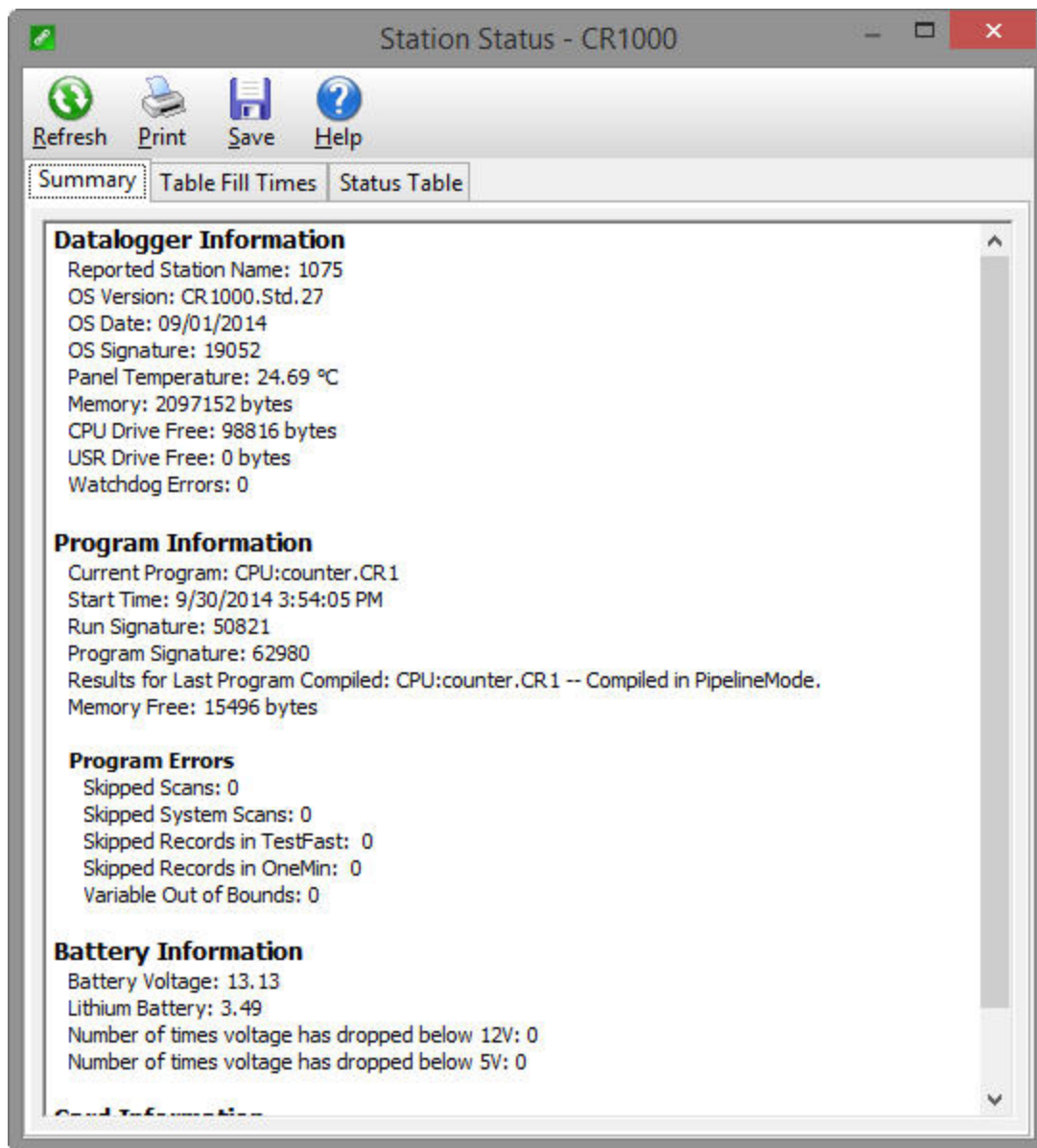
When a table-based datalogger is first set up or when a new program is sent, PC400 queries the datalogger for its table definitions. This information is used when displaying data in the Monitor

Data tab and when collecting data.

If a new program is sent by another user or by using some other program, PC400 will not be aware of the changes in the datalogger's table definitions. To update the table definitions for a datalogger, connect to that datalogger and select **Update Table Definitions** from the Datalogger menu.

### 4.4.2.3 Station Status

Information about the datalogger program, the execution of the program, battery voltage, internal temperature, etc. can be viewed from the **Datalogger | Station Status** menu item.





The window has three tabs. The **Summary** tab provides an overview of important status information in the datalogger, including the information about the datalogger model and its firmware, program details, program errors, battery voltage levels, and card memory (if one is present).

**NOTE:** Only the Summary tab is available for array-based dataloggers.

The **Table Fill Times** tab lists the tables in the datalogger, along with the maximum number of records the table can hold, and the estimated amount of time that it will take the table to fill. A data table can be reset from this window by pressing the **Reset Tables** button.

**NOTE:** For extended-memory dataloggers, auto-allocated data tables are automatically written to the extended internal memory (which is 72 MB), unless CardOut() or Tablefile() is used. In the case of CardOut() or Tablefile(), data from the CPU is streamed to the card in 1 KB frames and the internal extended memory is not used. Therefore, on extended-memory dataloggers, table fill times for auto-allocated tables on the CPU are greater if CardOut() or Tablefile() is not used. However, note that total final data storage for the table is greatly extended with external memory (up to 2 GB per table). In order to tell if the datalogger has extended internal memory, view the datalogger CPU Bytes Free in File Control. Dataloggers with extended internal memory show 30 MB Bytes Free for an empty CPU, compared to 1 MB Bytes Free for dataloggers that do not have extended internal memory.

**NOTE:** No Table Fill Time statistics will be shown for a CR200 Series datalogger, because they cannot be calculated for this datalogger model.

**NOTE:** For the CR10XTD, CR10XPB, CR510TD, CR510PB, CR23XTD, and CR23XPB, the Time of Fill will not be shown and you will not have the option to Reset Tables.

**NOTE:** Resetting a table will erase the data in the datalogger and in the data cache.

The **Status Table** tab lists all of the Status Table fields in the datalogger along with their values. By default, all of the fields in the Status Table are displayed. To select only certain status data to be viewed, press the **Select Fields** button. This will display a list of the status data available in the datalogger. Select one or more of the fields and then press **OK**. The current values will be displayed in the table. If you select a cell within the Status Table and **right click**, a short cut menu will be displayed. From this menu you can **select fields** or **view/modify** a value (if it is a writable value).

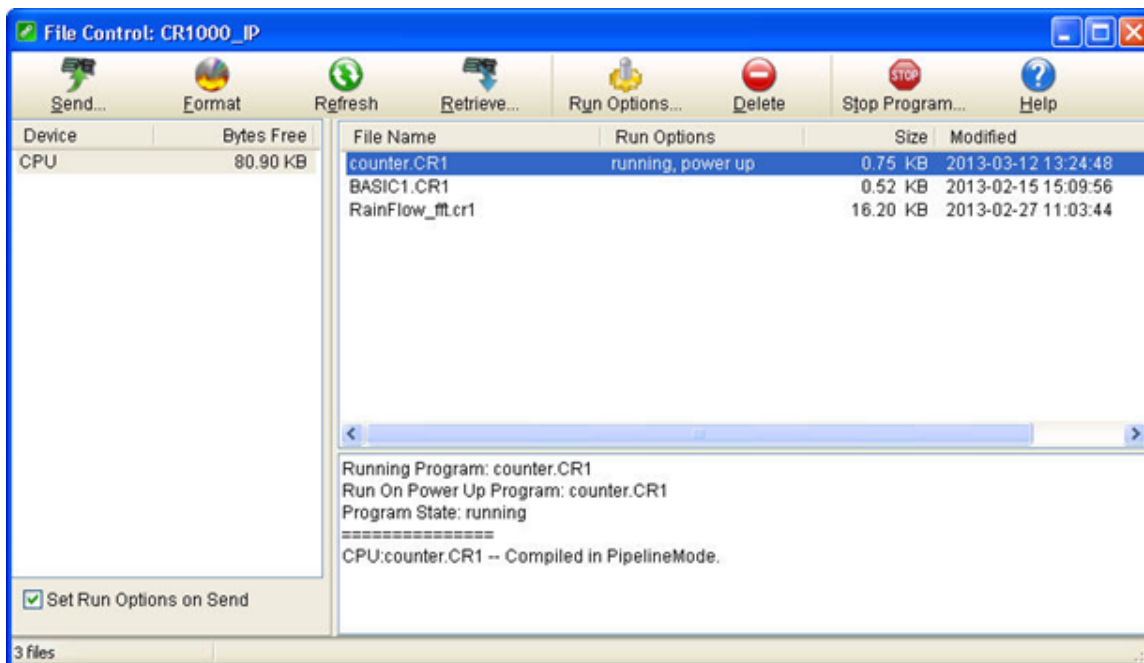
Refer to individual datalogger manuals for a list of fields included in the Status Table and a description of each field.

Press the **Refresh** button to prompt PC400 to query the datalogger and update the values again, the **Print** button to print the information in the current tab, or the **Save** button to save the information in the tab being displayed to a file. (Note that you cannot save or print the information on the Table Fill Times tab.)

By default, certain values in the Status Table are checked each time a program is sent. If a problem is found during the status check, the Status Table will pop-up along with a warning indicating the problem that has been identified. Through the **Tools | Options** menu item, automatic status checks can also be generated each time a connection to the datalogger is made and/or on a specified interval. From this menu, the automatic status checks can also be turned off completely.

#### 4.4.2.4 File Control

This option is used to manipulate the files stored on a CRBasic-programmed datalogger (excluding the CR200).



The **Devices** (e.g., CPU, CRD, USB, CS9, USR Drive) for the datalogger to which you are connected are shown on the left-hand side of the window. When you highlight a device in this field, the files stored on that device are listed on the right-hand side of the window.

**NOTE:** For GRANITE Data Logger Modules, an SC115 is specified as CS9 and USB refers to the integrated USB Host. For all other data loggers, USB refers to the SC115.

The **File Name** field provides the name of the file stored on the device. The **Run Options** field notes whether the file is set to Run Now (**running**), Run On Power-up (**power up**), or both Run Now and Run On Power-up (**running, power up**). The **Size** indicates the size of the file. **Modified** provides the date the file was last modified. The **Attributes** field indicates whether the file can be read from (R) and/or written to (W).

At the bottom of the right-hand side of the window is a summary box that indicates the Running Program, the Run On Power Up Program, the current Program State (running, stopped, or no program), and the last compile results.

### File Control Options

**Send** - Displays a standard file dialog box from which to choose a file to be sent to the selected device. If the Set Run Options on Send check box is selected, you will be asked to specify the Run Options for the file being sent as described below under Run Options.

**Format** - Formats the selected device. This option should be used with caution, since all files stored on the device will be lost.

**Refresh** - Press this button to update the devices and list of files for the datalogger. If a change is made via the keyboard or under program control, it will not be reflected in this window unless the screen is refreshed.

**Retrieve** - Highlight a file in the **File Name** field and press this button to display a standard file dialog box in which you can enter a directory and file name for the retrieved file.

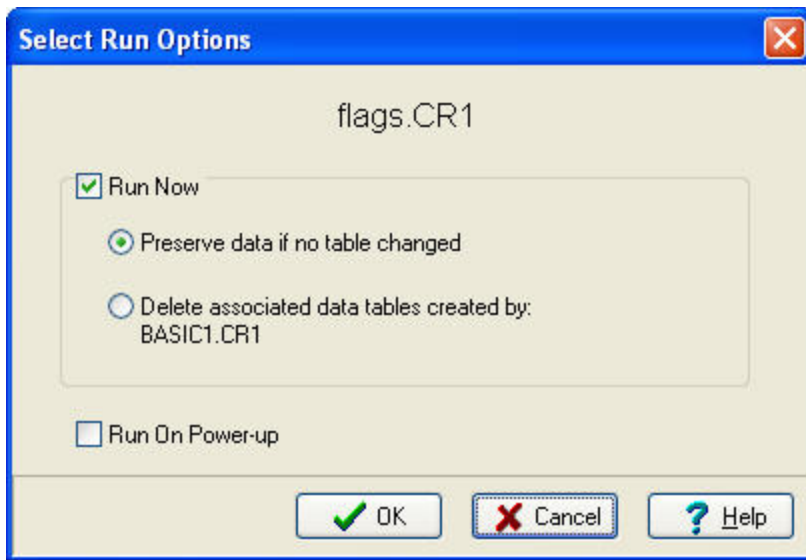
**NOTE:** File Control should not be used to retrieve data from a CF card created using the CardOut instruction. Using File Control to retrieve the data file can result in a corrupted data file.

**Run Options** - This button allows you to set the run options of a program file. Select the file from the **File Name** field, and press **Run Options**. From the resulting dialog box, select the run options.

#### Run Now

The Run Now run options are different for the different datalogger types.

CR1000X/CR6-series/CR300-series/CR350-series/GRANITE 6/GRANITE 9/GRANITE 10/CR1000/CR3000/CR800-series Datalogger Run Now Options



When Run Now is checked, the program is compiled and run in the datalogger. You may choose to preserve existing data tables on the datalogger's CPU if there has been no change to the data tables (**Preserve data if no table changed**) or to delete data tables on the CPU that have the same name as tables declared in the new program (**Delete associated data tables**).

**WARNING:** Neither of these options affects existing data files on a card if one is being used. If a data table exists on the card that has the same name as one being output with the new program, the message will be returned "Data on Card is from a different program or corrupted". Data will not be written to the card until the existing table is deleted. Data tables on the card that have different names than those declared in the new program will be maintained and will not affect card data storage when the new program is running.

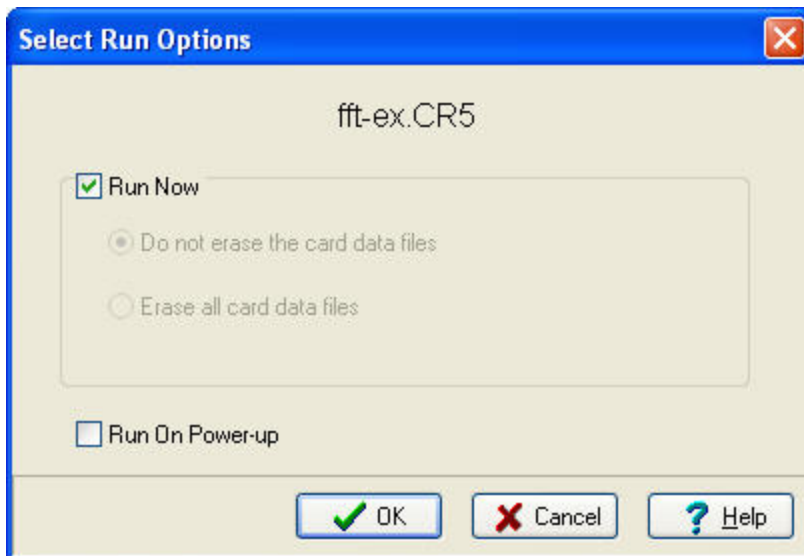
When using the **Preserve data if no table changed** option, existing data and data table structures are retained unless one of the following occurs:

- Data table name(s) change
- Data interval or offset change
- Number of fields per record change
- Number of bytes per field change
- Number of records per table (table size) change
- Field type, size, name, or position change

To summarize, any change in data table structure will delete all tables on the datalogger's CPU, regardless of whether or not the Preserve Data option was chosen. If the Preserve Data option was chosen but the datalogger was unable to retain the existing data, the following message will appear in the Compile Results: Warning: Internal Data Storage Memory was re-initialized.

### CR9000(X)/CR5000 Datalogger Run Now Options

The Run Now options and behavior for the CR9000(X) and CR5000 dataloggers are different from the CR1000X, CR6-series, CR300-series, CR350-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, and CR800-series dataloggers. Below is a dialog box for a CR5000 datalogger.



When Run Now is checked, the program is compiled and run in the datalogger. All data tables on the CPU are erased. You have the option of whether or not to erase data files stored on a card.

### Run On Power-up

The file will be sent with the Run On Power-up attribute set. The program will be run if the datalogger loses power and then powers back up.

### Run Always

Run Now and Run On Power-up can both be selected. This sets the program's file attribute in the datalogger as Run Always. The program will be compiled and run immediately and it will also be the program that runs if the datalogger is powered down and powered back up.

## Restarting a Stopped Program

Pressing **Run Options** to restart a stopped program in a CR1000X, CR6-series, CR300-series, CR350-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, or CR800-series datalogger displays a different dialog box. From the dialog box you can choose **Restart Program** to begin running the selected program immediately. You can also select an option button to determine whether or not the data tables previously created by the program are erased or retained.

**NOTE:** CR1000X, CR6-series, CR300-series, CR350-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, and CR800-series dataloggers -- A program marked as "Run on power up" can be disabled when power is first applied to the datalogger by pressing and holding the DEL key.

**Delete** - Highlight a file from the list of files and press **Delete** to erase the file from datalogger memory.

**Stop Program** - Press this button to stop the execution of the program currently running in the datalogger. Select the option to stop the program and retain the data files, or to stop the program and delete data files. The option to retain or delete data files includes those in internal datalogger memory and those on a card written by the CardOut instruction. In most cases, data written to a card using the TableFile instruction will not be erased. However, when writing data to the card using the TableFile instruction with option 64, the file currently in use will be erased while other files will be maintained.

If you select the option to Delete Data, you also have the option of whether or not to clear the Run On Power-up option for the file. Select the check box to clear the Run On Power-up option. Clear the check box to leave the Run On Power-up option of the file unchanged.

If a CR1000X, CR6-series, CR300-series, CR350-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, or CR800-series program has been stopped, you can use the **Run Options** button to restart it. If data files were not deleted when the program was stopped, you will once again be able to choose whether to retain or erase the data files. Instead of restarting the stopped program, you can choose a new program file to run by selecting a different file in the **File Name** field before pressing **Run Options**.

## Right Click Menu Options

When a file name is selected, pressing the right mouse button displays a menu with the **Retrieve File**, **Delete File**, **Rename File**, **View File** (retrieves the file and opens it in the CRBasic Editor), and **Run Options** choices.

**NOTE:** The View File option can be used to edit a program on your datalogger. After making the desired edits and saving it to your computer, you will need to send the edited program to the datalogger.

## 4.4.2.5 Terminal Emulator

This option allows you to communicate with a datalogger or other device in a remote terminal mode. It is used primarily for troubleshooting. The device to be tested is selected from the **Select Device** drop-down list box. When **All Caps Mode** is selected, all alpha characters that are typed will be sent to the device in upper case. A datalogger requires upper case commands; however, some modems or other communication devices recognize lower case commands.

Press the **Clear** button to erase all text from the Terminal Emulator window. Press **Close Terminal** to end the terminal emulator session.

Choose **Edit | Copy** from the Terminal Emulator menu or right-click in the window and select **Copy** to copy the entire contents of the terminal emulator window (up to 1000 lines) to the Windows clipboard. Choose **Edit | Paste** from the Terminal Emulator menu or right-click in the window and select **Paste** to place the contents of the Windows clipboard into the Terminal Emulator window.

## Datalogger Remote Terminal Mode

Press the **Open Terminal** button. Press **Enter** a few times. The datalogger will respond with **datalogger\_name>** (for example, **CR1000X>**), and you can begin typing in commands. Type an **H** and **Enter** to return the terminal emulator menu.

## Comms Watch

Comms Watch (also known as sniffer mode) allows you to view communications traffic on any communications port, except the port you are using to view the communications. To enter Comms Watch:

1. Type **W**, **Enter**
2. Type the number corresponding to the port you want to watch, **Enter**.
3. Type **Y**, **Enter** (for ASCII).


The communications on the selected communication port will be displayed. Lines preceded with a **T** are being transmitted; lines preceded with an **R** are being received.

When you are finished, press **Esc** to exit Comms Watch.

## Talk Through Mode

Talk through mode allows you to issue commands from the keyboard that are passed through the datalogger port to the connected device. There are terminal mode commands for both Serial

Talk Through (P) and SDI12 Talk Through (SDI12). In this way, you can manually interrogate sensors and enter settings in probes connected to the datalogger.

For how-to instructions for communicating directly with an SDI-12 sensor using a terminal emulator, watch this video: <https://www.campbellsci.com/videos/sdi12-sensors-transparent-mode>  .

## COM Port Feedback Test

To perform a feedback test:

- ❖ Select the COM port to be tested.
- ❖ Press the **Open Terminal** button.
- ❖ Connect the Transmit and Receive lines (2 and 3) of the serial port cable using a small Philips head screw driver or paper clip.
- ❖ Type characters on the computer keyboard.

If the characters are echoed back to the screen, the COM port is working.

The characters on the screen can be cleared by selecting the **Clear** button.

## Peripheral Device Test

The Terminal Emulator can also be used to test a peripheral device attached to the COM port such as a phone modem. Commands can be sent to test initialization strings or to manually dial a modem to connect to the datalogger.

To test a peripheral device, select it from the drop-down list box and press the **Open Terminal** button. Issue characters using the computer keyboard. The device must be in command mode.

The commands to be entered are specific to a device. Refer to the device's user manual for details.

## 4.4.3 Network Menu

[4.4.3.1 Add/Delete/Edit/Rename Datalogger](#) .....29

[4.4.3.2 Backup/Restore Network](#) ..... 30

[4.4.3.3 Computer's Global PakBus Address](#) ..... 30

### 4.4.3.1 Add/Delete/Edit/Rename Datalogger

The Add, Delete, and Edit options perform the same functions as the buttons on the main toolbar. The Rename Datalogger can be used to change the name of a datalogger.



### 4.4.3.2 Backup/Restore Network

This function can be used to save a copy of the network map to a file, and then to restore the network if necessary. The settings for all the devices in the network will be saved.

A default is given for the directory and file name to be used for the backup or restore. This can be changed by typing over the default directory and/or file name or selecting the button to the right of the field containing the file name and browsing to the desired directory and file name.

To backup the current network map, select the file name to which the backup will be stored, and then press the **Backup** button. The network map will be saved to the chosen file, and a message will appear indicating that the network has been backed up.

To regenerate the network map from a backup file, select the name of the backup file to restore from the dialog box, and then press the **Restore** button. Note that this backup will replace the existing network (it does not add to the existing network).

### 4.4.3.3 Computer's Global PakBus Address

All devices in a PakBus network, including any PC being used to communicate with the network, must have a unique PakBus address. To change the PakBus address of the computer running PC400, select **Computer's Global PakBus Address** from the Network menu. Valid addresses are 0 through 4094. The default address for PC400 is 4093. It is good practice to set the PakBus address for all PCs that connect to a datalogger network to different PakBus values.

**NOTE:** Campbell Scientific's LoggerNet software uses 4094, RTDAQ uses 4090, and the Device Configuration Utility uses 4089.

## 4.4.4 Tools Menu

4.4.4.1 Stand-alone Applications .....	30
4.4.4.2 Options .....	31

### 4.4.4.1 Stand-alone Applications

Use these menu items to launch the stand-alone tools, including Split, View Pro, CardConvert, Short Cut, Edlog, and CRBasic Editor. Some users prefer using the keyboard to the mouse, and the Alt-T keystroke provides this access. These stand-alone applications are described in more detail in their own sections of this manual

#### 4.4.4.2 Options

The **Tools | Options** menu item is used to set the following options:

- **Automatically Check Datalogger Status** - This option allows you to determine when the datalogger status is automatically checked. The options include **After Connection**, **After Program Send**, and/or **On Interval**. If none of the options are selected, automatic checking of the datalogger status will be disabled completely.

The status values checked include the following: PakBus Address, WatchDog Errors, Skipped Scans, Skipped Slow Scans, Skipped Records, Variable Out of Bounds, Program Errors, Battery Voltage, Lithium Battery, Number of times voltage has dropped below 12V, Number of times voltage has dropped below 5V, Stack Errors, and Calibration Errors. Note that not all status values are applicable to all dataloggers. Only those status values applicable to the current datalogger will be checked.

If a problem is found during a datalogger status check, the status table will pop-up along with a warning indicating the problem that has been identified.

- **Show Hints** - The **Show Hints** option is used to turn on/off the tool tips that appear when your mouse hovers over a control (button or field). When a check mark appears beside the menu item, the tool tips will be displayed. When the check mark does not appear beside the menu item, no tool tips will be displayed. This menu item is a toggle; select it to change its state.

## 5. Creating and Editing Datalogger Programs

Dataloggers must be programmed before they can make measurements. PC400 offers three options for programming dataloggers. Short Cut, CRBasic Editor, and Edlog.

Short Cut (also referred to as SCWIN) is an application for generating programs for all of Campbell Scientific's dataloggers and preconfigured weather stations except the CR7 and CR9000. Users do not have to know individual program instructions for each datalogger. Short Cut not only generates a program for the datalogger, but also a wiring diagram that can be left with the datalogger for field servicing.

The CRBasic Editor is the full-featured program editor for the CR1000X-series, CR6 series, CR300-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, CR800-series, CR200-series, CR5000, CR9000, and CR9000X dataloggers. It is a full-featured editor that requires the user to understand the program instructions for the datalogger, but it can be used to develop more complex programs than what can be created using SCWIN.

The CR7, CR10, 21X, CR500, CR510, CR10X, and CR23X dataloggers are programmed using the Edlog editor. Edlog supports all operating systems for these dataloggers, including the table-data or "TD" and PakBus or "PB" versions. Like the CRBasic Editor, it requires that the user have more knowledge of datalogger program instructions than SCWIN.

In addition to the above programming tools, the Transformer utility is offered in LoggerNet for those users of CR10X or CR23X dataloggers who need to develop programs for the CR800-series, CR1000 or CR3000 dataloggers.

## **5.1 Short Cut**

The Short Cut program generator creates programs for Campbell Scientific dataloggers in seven easy-to-follow steps. Using a wizard-like interface, you create a new or open an existing program, select the datalogger, choose which sensors you wish to measure, select the measurement interval and the frequency of data stored in datalogger memory, specify advanced output options (that is, to store data based on a flag or the value of a measurement), specify what processing to perform on raw measurements for final storage, and finally generate the program. Short Cut also generates a wiring diagram for connecting your sensors to the datalogger.

Short Cut was designed to help the beginning datalogger programmer create datalogger programs quickly and easily. Short Cut effectively insulates the user from having to know the nuances of datalogger programming and the Edlog versus CRBasic programming languages. It supports the most commonly sold sensors from Campbell Scientific, as well as generic measurements (such as differential voltage, bridge, and pulse), commonly used calculation and control functions (such as heat index calculation, alarm conditions, and simple controls), and multiplexer analog channel expansion devices.

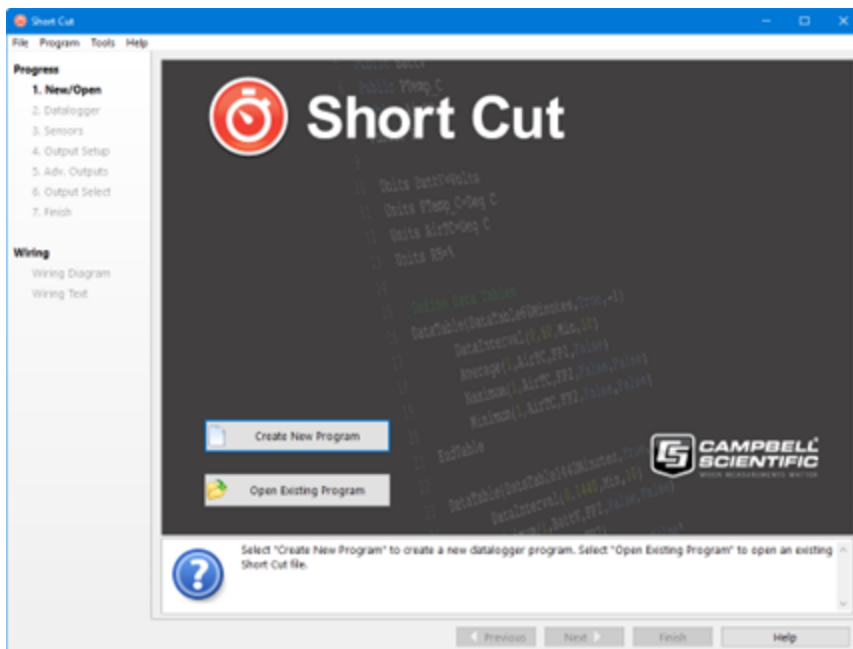
Short Cut cannot be used to edit existing Edlog, CRBasic, or Short Cut for DOS programs. Program editing and more complex datalogger programming functions should be accomplished using our Edlog or CRBasic Editor programming tools.

Short Cut was designed with extensive built-in help. Help can be accessed at any time by pressing **F1**. There are also **Help** buttons on most screens. You can also open the Help by selecting **Short Cut Help** from Short Cut's Help menu. Help for each sensor can be accessed by searching the Help Index or pressing the **Help** button from the sensor form.

After generating the program, you can send it to the datalogger from the **Results** tab of Short Cut's Finish screen or from LoggerNet's Connect Screen or from PC400 or RTDAQ's Clock/Program tab.

### 5.1.1 Creating a Program Using Short Cut

On opening, Short Cut presents a wizard that walks you through the steps of creating a datalogger program.

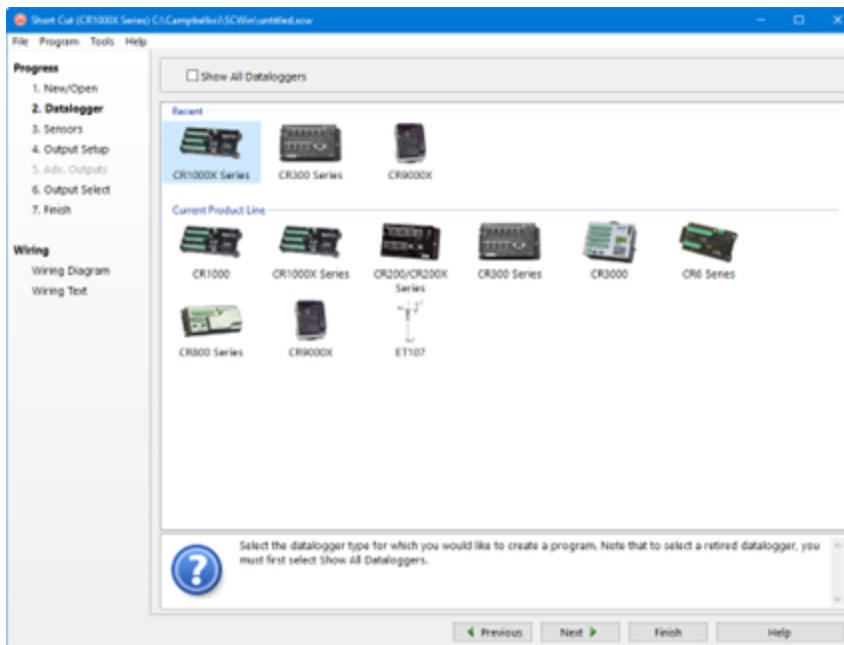


### 5.1.1.1 Step 1 – Create a New File or Open Existing File

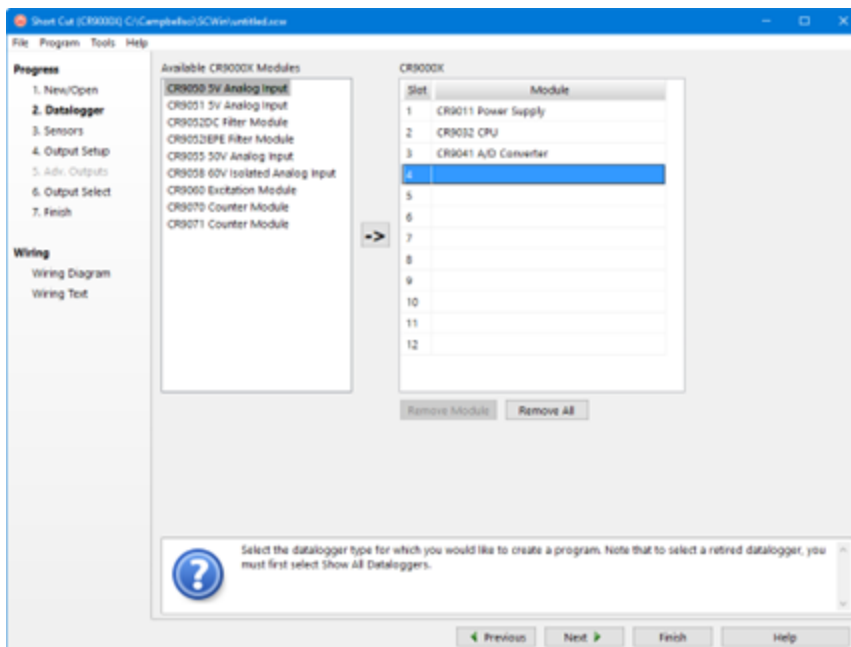
To begin creating a new program, press **Create New Program**. To open an existing program, press **Open Existing Program** and select a file from the resulting browser window.

### 5.1.1.2 Step 2 – Select Datalogger

Select the datalogger model for which to create a program and press **Next**. (By default, only our current product line is shown. Select **Show All Dataloggers** to include our retired product line.)



If you are creating a program for a CR9000X, the CR9000X Configuration screen will appear when you press **Next**.

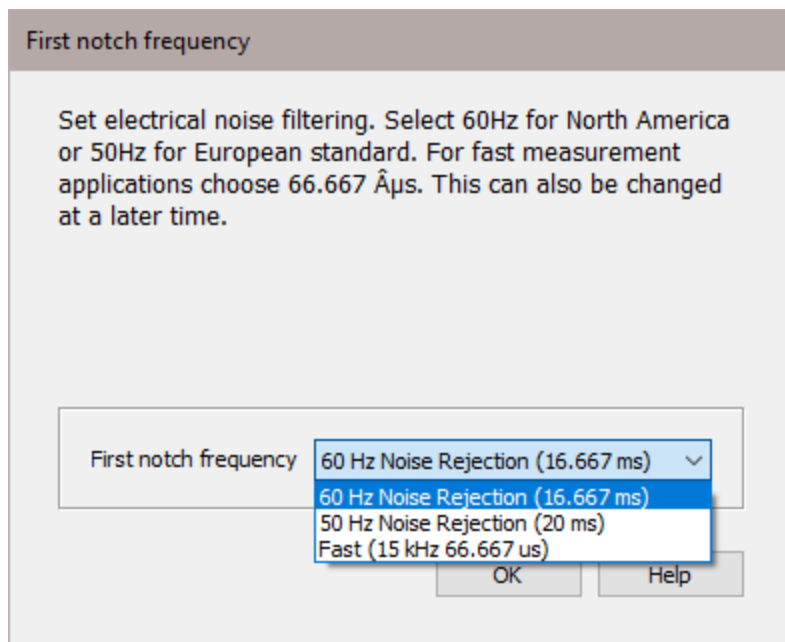


From this screen, you indicate which CR9000X modules are inserted into which CR9000X slots. To add a module, select the module by clicking on it in the **Available CR9000X Modules** list, select the **Slot** by clicking on the slot number, then press the arrow key.

To remove a module, select the slot containing it and then press the **Remove Module** button.

**NOTE:** Whenever you are working with a CR9000X program, this dialog box can be brought up by choosing **Datalogger** from the **Progress** panel and then pressing **Next**. However, the **Remove Module** button is only available when a new program is being created. Once the **Next** button on the screen has been pressed, modules can be added but they cannot be removed.

The next dialog box that is displayed is used to select the **First notch frequency** to apply to the measurements in the program. This can be used to filter out AC signals that might affect the accuracy of your measurements (such as noise from fluorescent lighting or a generator). Typically 60 Hz rejection is used for North America and 50 Hz rejection is used for countries following European standards. Fast (15 kHz) integration should be used when you need an execution speed that cannot be accomplished using one of the other options.

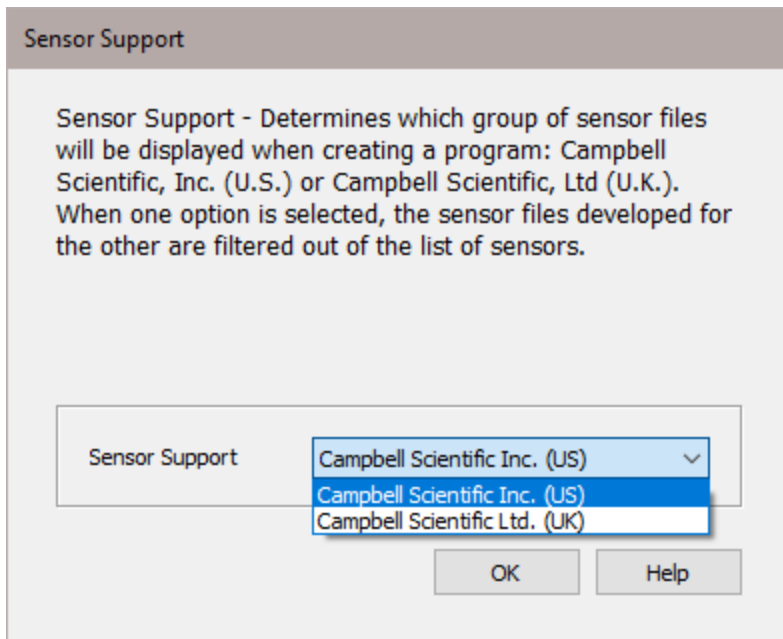


This dialog box will be displayed the very first time you create a program for a specific datalogger type; it will not be displayed thereafter. With each subsequent program you create, the first notch frequency you chose when the datalogger was initialized in Short Cut will be used. However, you can change the first notch frequency from the **Program** menu. If you make this change, the setting will remain in effect for all programs for that datalogger type (whether they are new programs or edited programs) until it is changed again.

**NOTE:** For the CR1000X series, CR6 series, CR300 series, and GRANITE Data Logger Modules, the integration setting is named first notch frequency (fN1).

The last dialog box displayed is the **Sensor Support** dialog box. (This dialog box will not be displayed when creating a CR9000X program.) This is used to select which group of sensor files

will be displayed when creating a program: Campbell Scientific, Inc. (CSI, USA) or Campbell Scientific, Ltd. (CSL, UK). The standard set of Short Cut sensor files was created by CSI; however, CSL has created some additional files that are customized for their client base. When one option is selected, the sensor files developed specifically for the other are filtered out.



This setting is similar to the **First notch frequency** setting in that the dialog box will be displayed only the first time you create a program for a specific datalogger type, and the setting will apply to all programs created or edited for that datalogger, unless it is changed via the **Program** menu. Note that programs containing sensor files that are filtered from the list of **Available Sensors** will still load and work correctly in Short Cut.

**NOTE:** The **First notch frequency** and the **Sensor Support** settings are persistent settings for each datalogger model. The first time you create a program for a particular datalogger model, you will be presented with these two dialog boxes. The state of these settings is saved between Short Cut sessions. Any subsequent new or edited programs that are generated after a setting has been changed will reflect the change as well.

Each time you create the first program for a datalogger model you will be presented with these dialog boxes (e.g., the first time you create a CR10X program, you must initialize these settings; the first time you create a CR1000 program, you must initialize these settings).

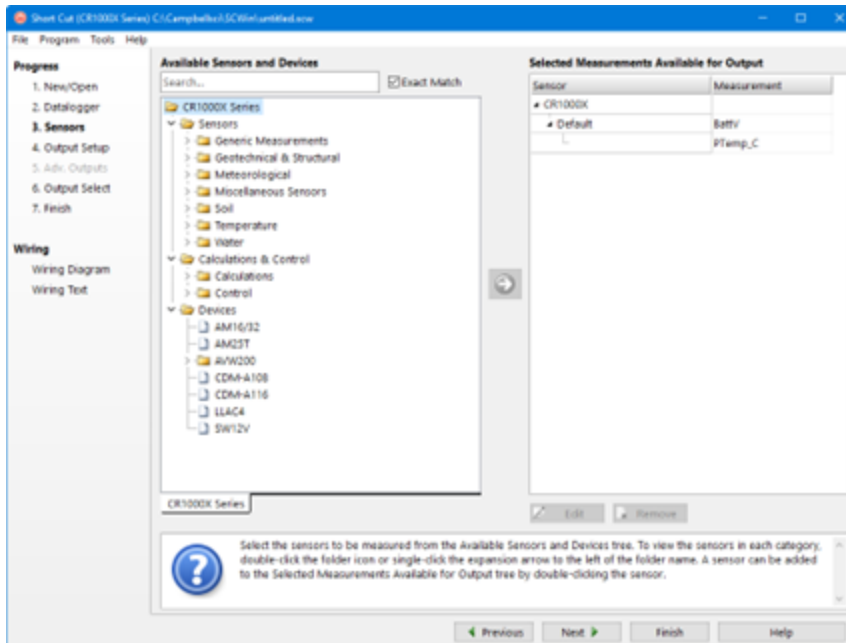
The settings can be changed at any time and the datalogger program will be regenerated to use the new setting when you click **Finish** on the Home screen.

After making your selections, note that the title bar shows the datalogger type.

Once you have saved the file, the filename will replace “untitled.scw”.

### 5.1.1.3 Step 3 – Choose Sensors to Monitor

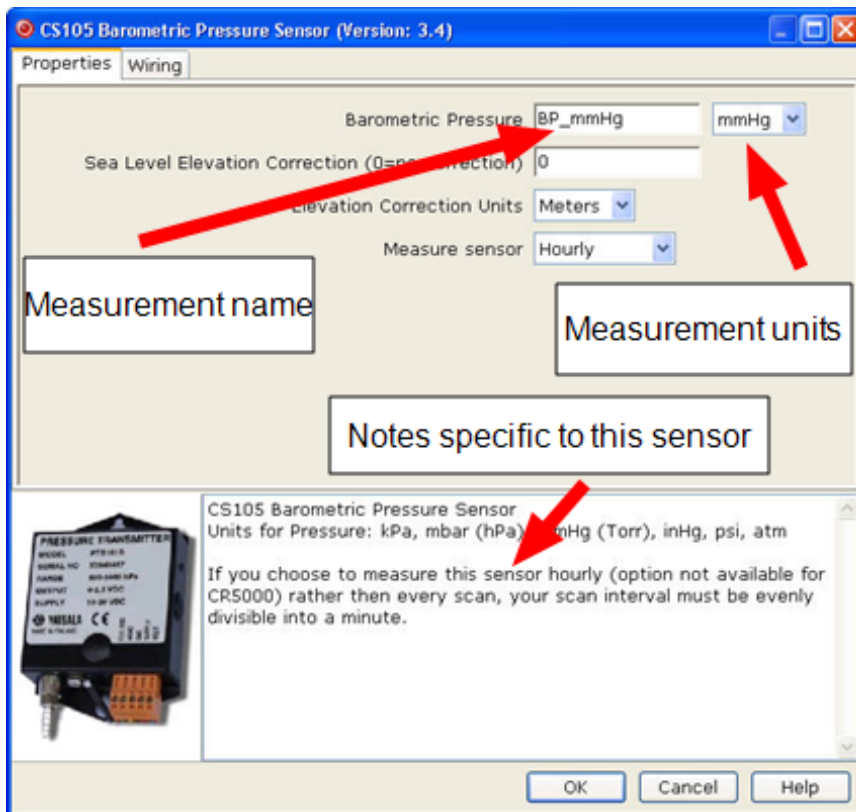
In step 3, you tell Short Cut which sensors you’ll be measuring. Short Cut organizes sensors into application groups:



Some major groups have subgroups. Double-clicking the Meteorological group folder shows several subgroups of meteorological sensors. Double-click a subgroup to show the available sensors. Refer to the documentation for your sensors for the name of the sensors you have. If your sensor is not shown, you may be able to measure it with a generic measurement. Contact your Campbell Scientific application engineer for more assistance, if needed.

You “add” sensors to your program by double-clicking them or selecting them and clicking the arrow in the middle of the screen. Most sensors will require you to at least review the default settings for that measurement, including the measurement name, units, etc. An example of choosing the CS105 Barometric Pressure Sensor is below.





Note that this sensor not only offers a custom name field and units, but also allows you to correct for sea level, a common practice in measuring atmospheric pressure. In the middle of the screen, look over the notes (or refer to the Help for this sensor), for this sensor may require other sensors or have limitations. When you choose **OK**, Short Cut adds the necessary instructions with appropriate multipliers and offsets.

In some cases, multiple sensors of the same type can be added at one time. These sensors will have a **How many sensors?** parameter as the first parameter on the form as shown below. The maximum number of sensors that can be added will be indicated. The maximum will vary, depending upon the sensor and the number of other sensors already configured in the program. If the sensor form includes calibration and/or conversion parameters (e.g., multiplier, offset, gage factor), there will be a **Set** button next to these parameters. Pressing this button will allow you to set unique values for each sensor.

**Half Bridge (Version: 3.0)**

Properties **Wiring**

How many HalfBr sensors? (Max=3)

Total Bridge Resistance (ohm)

Excitation Voltage (mV)

Sensors Per Excitation Channel

Measurement Result

Range of Sensor Voltage

Reverse Excitation to cancel offsets? ☒ True

Measurement Integration

Settling Time, us (0 for default)

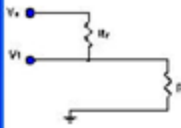
Multiplier, Offset

Optional Field Calibration

☐ Two Point, Multiplier and Offset

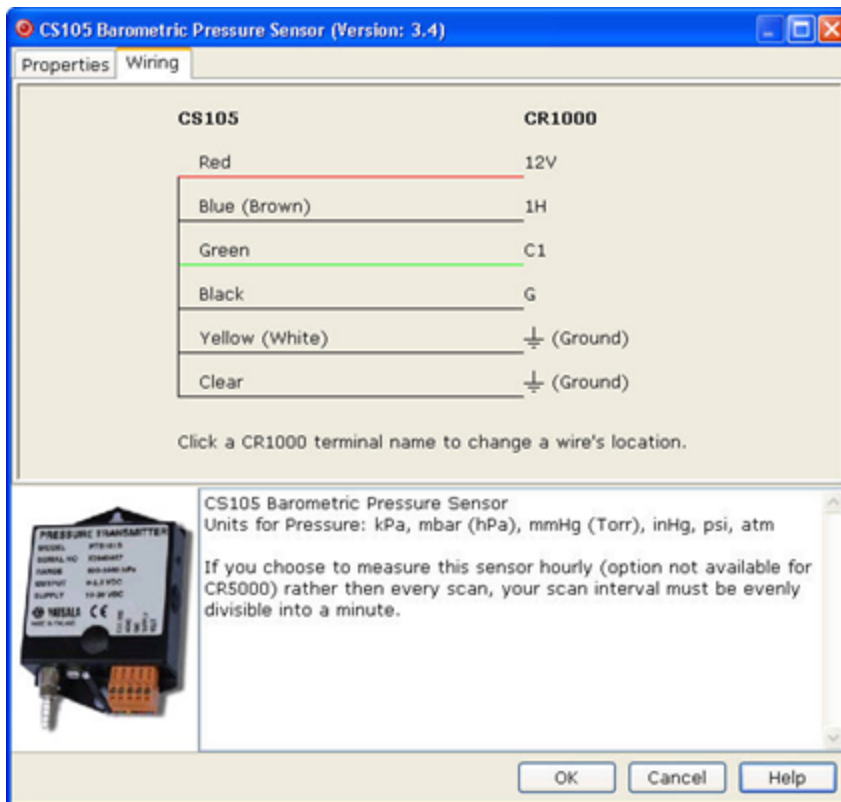
☐ Zeroing Calibration

Select Zeroing Calibration Group



This Half Bridge measurement applies an excitation voltage and makes a voltage measurement of the bridge output. Optionally, it then reverses the polarity of the excitation voltage and makes another measurement (e.g., excites first with +1000 millivolts then with -1000 millivolts). The voltage

Click on the **Wiring** tab of a sensor's parameter form to show the wiring for the sensor (or the first sensor in a sensor group).



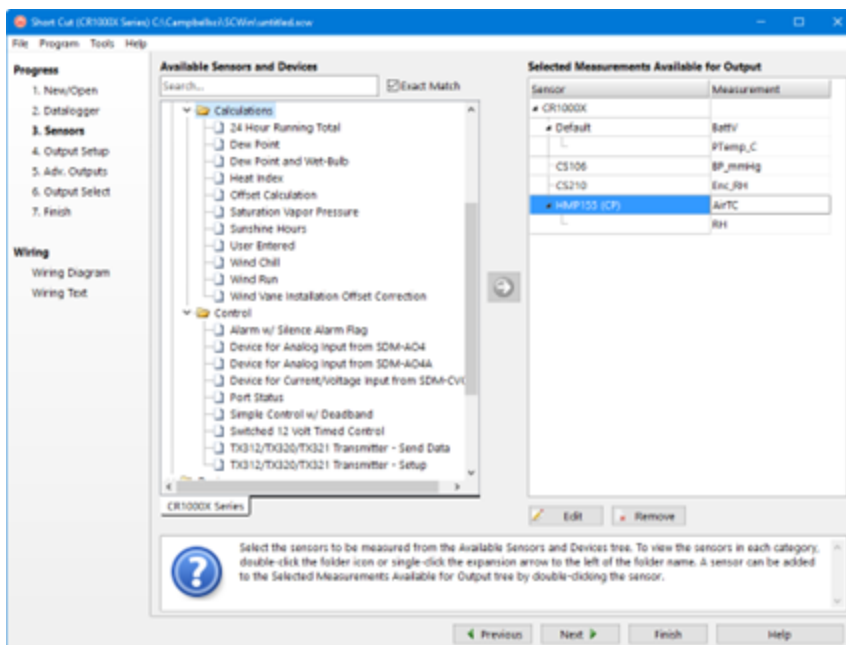
Each wire's caption/color is shown on the left side of the wire. The location where the wire will be connected to the device is shown on the right side (under the device). You can change a caption/color by clicking on the caption/color label. A wiring location can also be changed by clicking on the wiring location.

**NOTE:** Changes to the wiring location for a sensor group can only be made when the group is first added. To make changes to a wiring location at a later time, you will need to change the number of sensors to one, press **OK**, reopen the parameter form, make the desired wiring location changes, and then change the number of sensors back to the desired number.

**NOTE:** Not all sensors support changes to the wire caption/color and wiring location. When hovering over a wire caption/color or wiring location, the mouse cursor will change to indicate that the property can be changed. Changes are generally supported for generic sensors and other sensors that do not use special wiring connections.

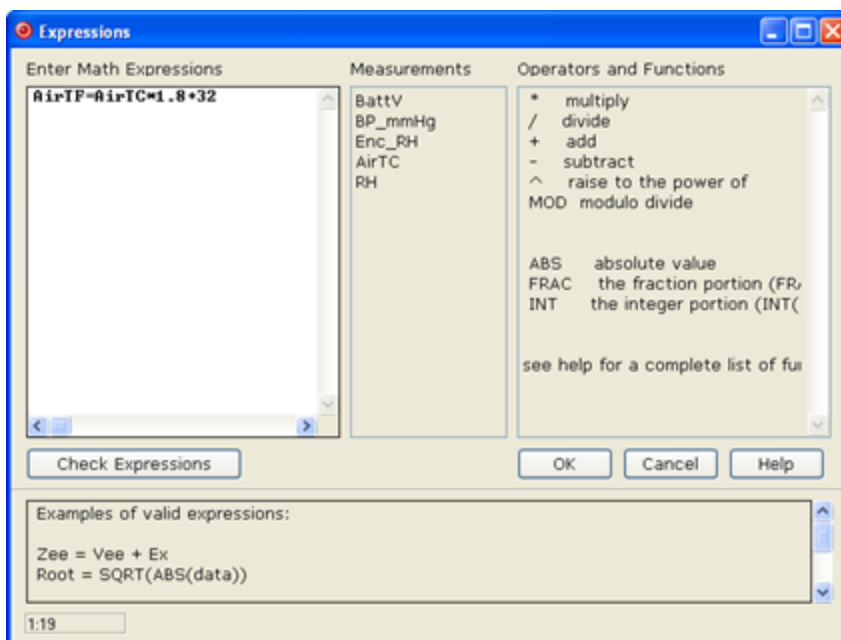
At any time, you may choose a measurement label on the right side of the Sensors screen and edit it or remove it.

In addition to actual sensors, Short Cut provides functionality to perform various calculations and effect some simple control:



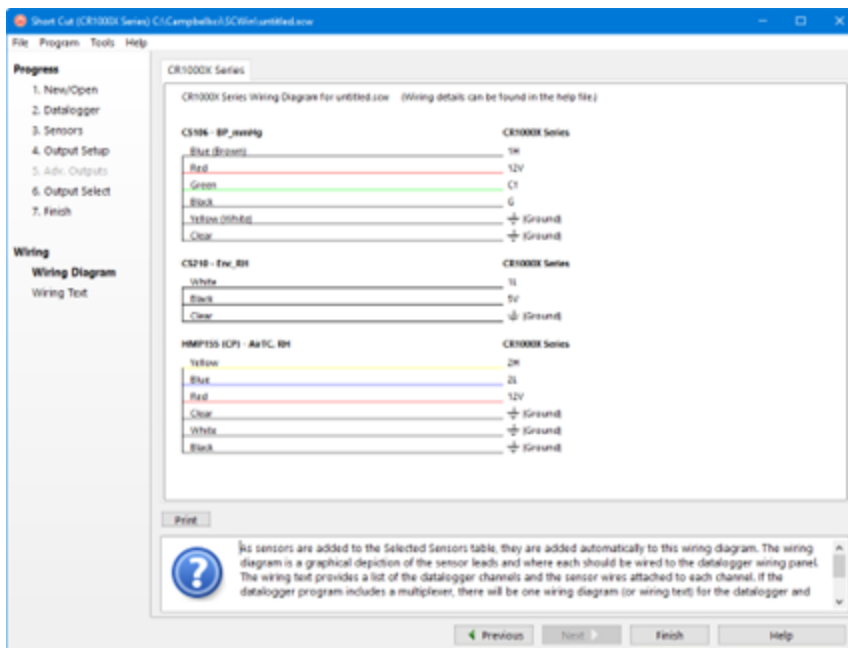
Some of these calculations may require additional sensors, or sensor measurements stored in particular units. See the help for each calculation to determine the necessary inputs. Note that there is also a User Entered calculation available in the Calculations folder. With it you can enter your own custom calculation.

In the example below, a new measurement, AirTF, is being created by performing calculations on an existing measurement, AirTC:

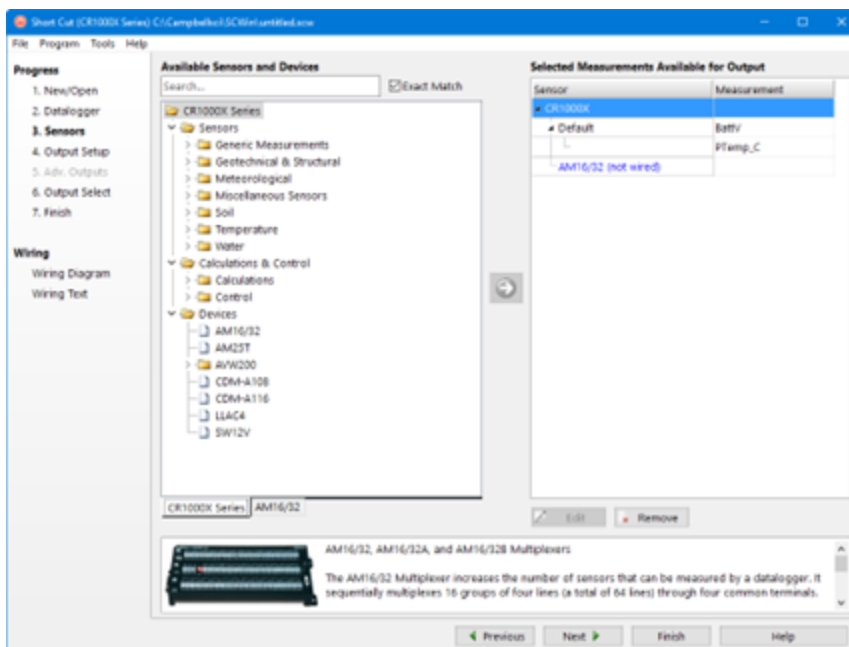


Refer to the online help for complete information on creating User Calculation.

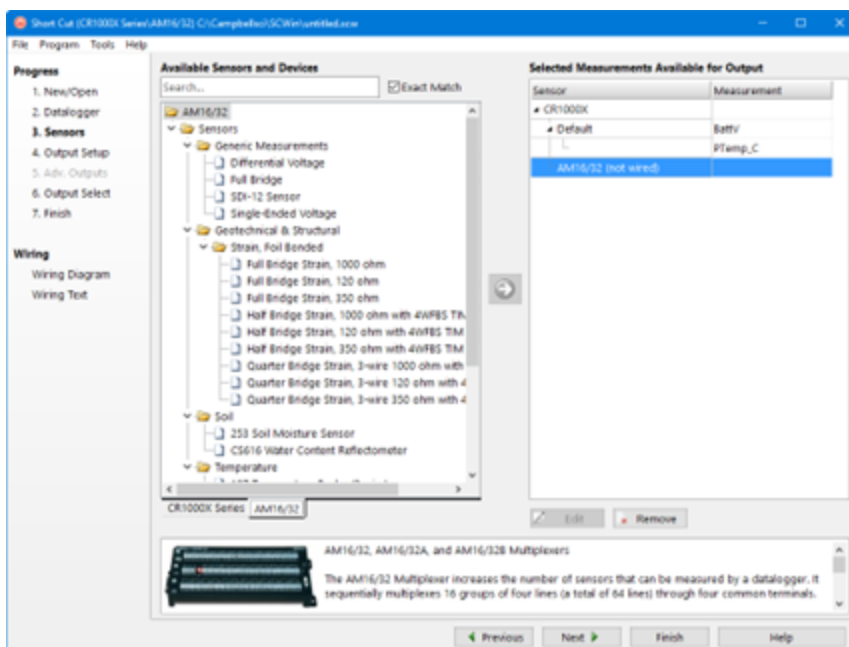
Short Cut provides you with a wiring diagram by clicking on **Wiring Diagram** on the left side of the Sensors window. In the example below, Short Cut was told to measure a CS106 Barometric Pressure sensor, a CS210 enclosure relative humidity sensor, and an HMP155 Air Temperature and Relative Humidity sensor. Each sensor was allocated the necessary terminals. Short Cut will not let you add more sensors than there are terminals on that datalogger or device. You can print this diagram (or the textual equivalent) by choosing the **Print** button. Many users find it handy to leave a printed wiring diagram in the enclosure with the datalogger in case a sensor has to be replaced.



Short Cut can also create programs for dataloggers using a variety of interface devices, including multiplexers and special interfaces for sensors. Add these devices by selecting them from the Devices folder in **the Available Sensors and Devices** tree.



Once you've added a device, such as the AM16/32 multiplexer, a tab is added to the screen for that device, and the sensors available for that device are shown:



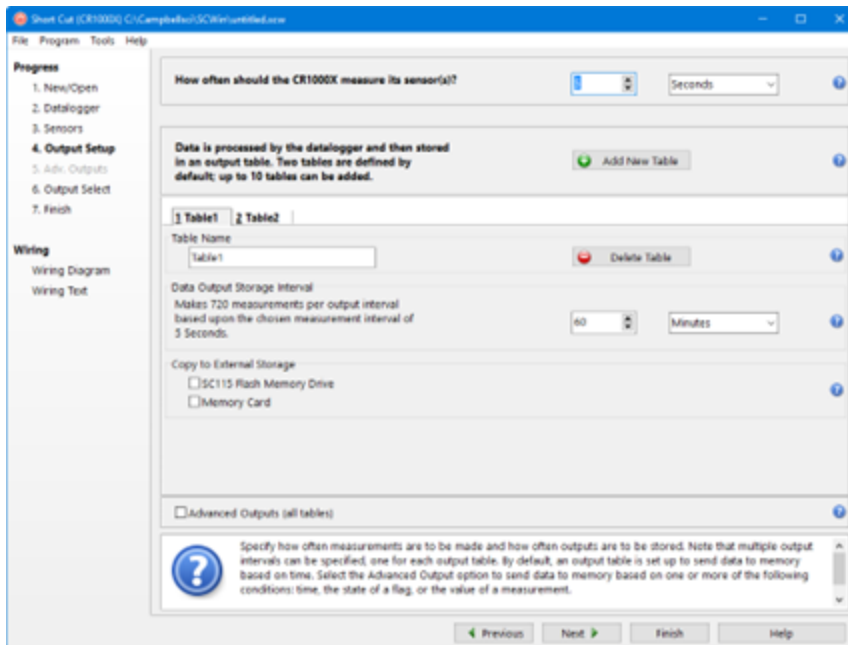
You can then add sensors to that device just as you would to the main datalogger.

Note that, once you add a sensor to a multiplexer, it may limit what kind of sensors can be added thereafter, as each sensor on the multiplexer must share the same wiring between the multiplexer and the datalogger.

After adding all the desired sensors, click **Next**.

### 5.1.1.4 Step 4 – Set up Output Tables

The fourth step in creating a program is to set up the output tables for the sensor measurements you have selected. The output tables must be completed or no data will be stored in the datalogger's memory.



In the **How often should the datalogger measure its sensor(s)?** field, specify how often the datalogger will execute the instructions in its program. This is known as the measurement or scan interval.

When choosing a scan interval, remember that faster scan intervals will use more power. For most applications, a 10 to 60 second scan interval is sufficient. If faster scan intervals are required for your application, make sure there is sufficient time for the execution of all instructions in the program (refer to the section in the datalogger manual on Execution Intervals for additional information).

**NOTE:** By default, data is sent to memory based on time. Data can also be sent to memory based on one or more of the following conditions: time, the state of a flag, or the value of a measurement. This is set up from the **Advanced Outputs** screen. To use the **Advanced Outputs** screen, select the **Advanced Outputs (all tables)** check box at the lower left of the **Output Setup** screen. The **Data Output Storage Interval** field will be removed from the **Output Setup** screen (and moved to the **Advanced Outputs** screen). After completing the fields on the **Output Setup** screen and pressing **Next**, Short Cut will advance to the **Advanced Outputs** screen.

Two tables are defined by default. Additional tables can be added by pressing the **Add Table** button. Short Cut limits the number of output tables to 10. An output table can be removed by clicking on the table to make it the active table and pressing the **Delete Table** button.

Steps for completing the standard table output are given below:

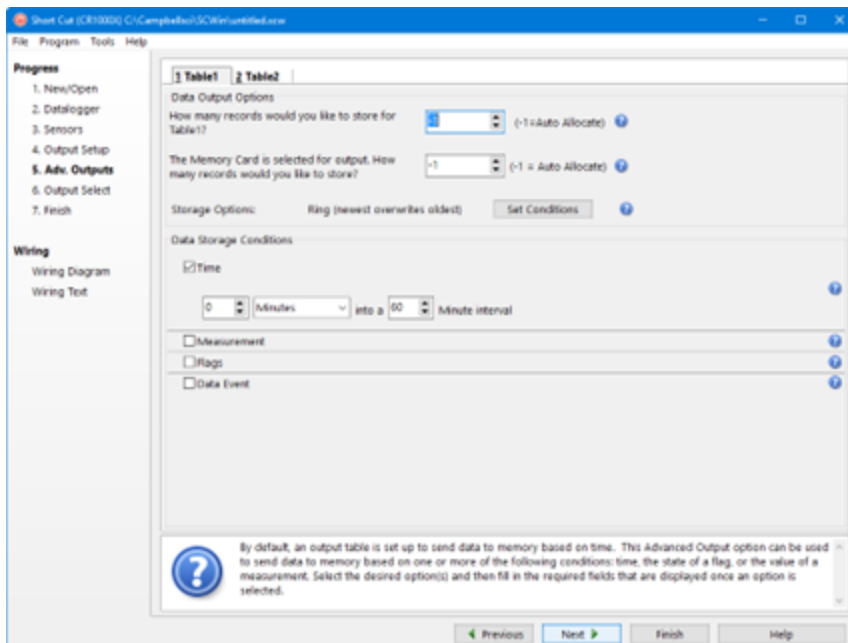
- Name the output.
  - **Mixed-array dataloggers:** The **Array ID** field is for the identification number that will be used by the datalogger to identify the output array. You can accept the default ID number, or type a new number in the field (1 to 511 are valid options). A unique array ID must be used for each output table.
  - **Table-based dataloggers:** The **Table Name** field is the name that will be used for the data table in the datalogger. You can accept the default Name of Table1, Table2, etc., or type a new name in the field. The table name can be up to 20 characters.
- The **Data Output Storage Interval** field and the adjacent drop-down list are used to set the interval at which data will be stored to memory. The default output intervals are 60 minutes (Table1) and 1440 minutes (Table2), but they can be changed. (This field is removed from this screen if the **Advanced Outputs (all tables)** checkbox is selected. In this case, it can be set from the **Advanced Outputs** screen along with any other conditions to be met for data to be stored.)
- Table-based dataloggers that support output to a PCMCIA, microSD, or compact flash card will have a **Memory Card** check box. When this box is selected, the table will be stored to a card inserted in the datalogger, as well as to datalogger memory.
- Table-based dataloggers that support output to the SC115 will have an **SC115 Flash Memory Drive** check box. When this box is selected, new data will be copied to an SC115 when it is plugged into the CS I/O port of the datalogger. (This mode of operation. (This mode of operation is referred to as Data Collection Mode. See the SC115 manual for more information.)
- Dataloggers that support publishing to CampbellCloud will have an **MQTT Publish** check box. If this check box is selected, data from the current table will be published using MQTT. Data will be published every 10 minutes or at the Data Output Storage Interval if it is greater than 10 minutes. If **Advanced Outputs** options are used, the data will be published every 10 minutes. Note that the datalogger must be CampbellCloud enabled and on-boarded before the data will be received by CampbellCloud.
- GRANITE Data Logger Modules will have a **USB Mass Storage Drive** check box. If this box is selected, new data will be copied to a USB mass storage device when it is plugged into the



USB host port of the datalogger. Once the new data has been copied to the drive, no more data is copied until the drive is removed and plugged in again.

### 5.1.1.5 Step 5 – Set up Advanced Outputs

Selecting the **Advanced Outputs** check box at the bottom left of the **Output Setup** screen enables the **Advanced Outputs** screen (the fifth step in the **Progress** panel). This screen allows you to send data to memory based on time, the state of a flag, or the value of a measurement. The options are set separately for each table. Be cautious in using more than one check box, for the logic for the check boxes in the advanced mode are inclusive — that is, they must all be true in order for any output to be stored.



- Data Output Options (table-based dataloggers only)

In the **How many records would you like to store for tablename?** field, enter the maximum number of records that should be stored in the table. Once the maximum number of records have been stored in the table, the oldest record will be removed when a new record is added.

Instead of specifying a fixed table size, you can let the datalogger set table size automatically (autoallocate) by using the default value (0 or –1, depending upon the datalogger). When table size is autoallocated, the datalogger will first assign memory to any fixed-size tables and then will divide its remaining memory among the autoallocated tables so that all tables are filled at approximately the same time.

**NOTE:** Consideration should be given when configuring tables with conditional output for autoallocation, because it may not be the most efficient use of datalogger memory. If output is not based on a time interval, the datalogger will assume the output interval to be the same as the execution interval. This may result in the datalogger allocating memory for a very large table for the conditional data, and much smaller tables for the remaining tables. Therefore, you may want to specify a fixed size for conditional tables.

If the **Memory Card** checkbox was selected on the **Output Setup** screen, you must also specify **How many records would you like to store?** to the memory card.

**Storage Options** – By default, data table memory is set up as “ring memory.” Once the maximum number of records has been stored to a table, each new record will overwrite the oldest record in the table. However, you can set a data table to fill and stop (the data table fills and then no further data is stored) by pressing **Set Conditions** and selecting **Fill/Stop (stop when full)** from the list box. Other options that can be set from this dialog box are to store a file mark in the data table each time a specified flag goes high, or to set a flag high when the table is full.

- **Data Storage Conditions**

Check the appropriate box for one or more of the output conditions:

**Time** – Enter the number of minutes (or milliseconds, seconds, hours, or days for certain datalogger models) into an Interval for when the output should occur. The first **Time** field provides an offset into the interval; the second **Time** field is the interval on which output should occur. For instance, if the output is set to 0 minutes into a 60-minute interval, data will be stored to memory at the top of every hour. If the output is set to 15 minutes into a 60-minute interval, data will be stored to memory at 15 minutes past the hour, each hour.

**Measurement** – Use the first list box to select the measurement to evaluate. The second list box contains the list of comparators (= equal to, <> not equal to, >= greater than or equal to, <= less than or equal to, < less than, or > greater than). The value to test the measurement against is entered into the last field. If an input location called Temp is selected in the first field, >= is selected as the comparator, and 27 is entered in the numeric field, data will be stored to memory each time Temp is greater than or equal to 27. For table-based dataloggers, when the Measurement option is set, the table size should be set to a fixed value instead of autoallocate (-1). See note above.

**Flags** – Use the first list box to select the flag and the second list box to select the state of the flag that will cause data to be stored to memory. If Flag 8 is selected from the first list, and High is selected from the second, data will be stored to memory each time Flag 8 is

high during program execution. For table-based dataloggers, when the Flags option is set, the table size should be set to a fixed value instead of autoallocate (-1). See note above.

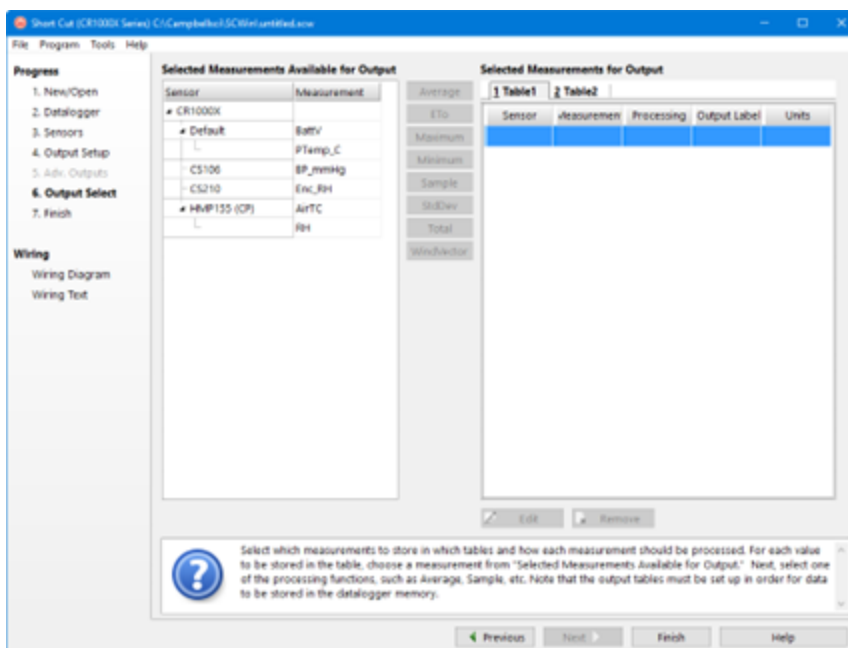
**Data Event** – This option is used to conditionally store data to a table, based on the value of a variable (table-based dataloggers only).

- The **Records Before** field is used to enter the number of records that should be stored prior to the condition being met (the datalogger will keep this number of records in memory). The **Trigger** is the variable that will be monitored for the specified condition. Use the drop-down list box to select the trigger from the list of variables in the program. The two remaining fields for the trigger are used to specify the value for the variable that will trigger the condition.
- An **Optional Stop Trigger** can also be specified, which will stop the storage of data to the table. If a stop trigger is not specified, data will be stored to the table indefinitely. If a stop trigger is specified, you can also specify the number of records to continue storing to the table after the stop trigger condition is true in the **Records After** field.
- Fields below the trigger criteria indicate the total number of records that will be stored to the table when the trigger condition is met.

When the **Data Event** option is set, the table size should be set to a fixed value instead of autoallocate (-1). See note above.

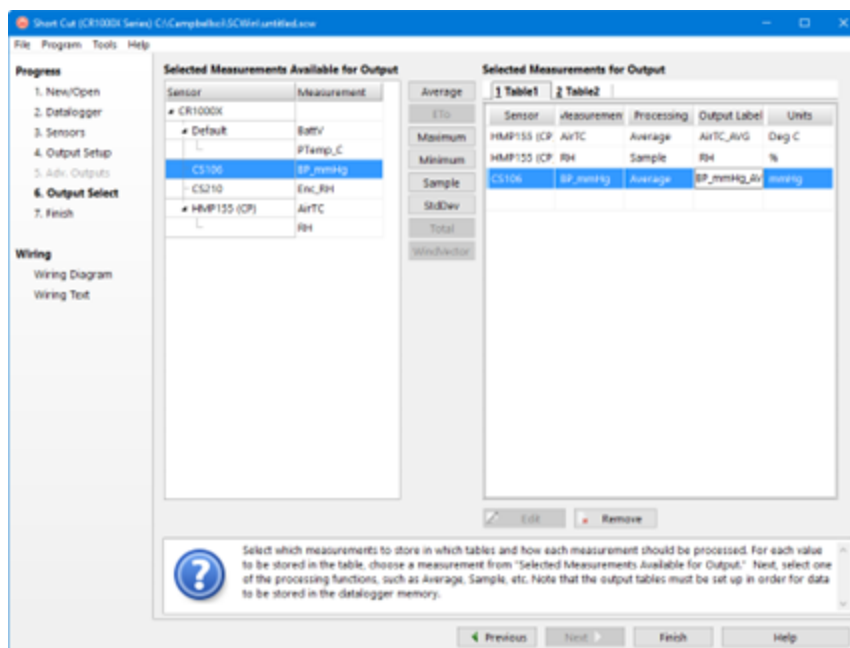
### 5.1.1.6 Step 6 – Select Outputs

After setting up the data storage conditions, you can choose what data to store in each table:



On the left, Short Cut will show the sensors you've added to be measured, with the measurement labels you've used. On the right is a multi-tabbed grid that shows the output tables.

To store a measurement to final storage, simply click on a measurement label on the left, choose the data processing you want for that measurement by clicking one of the enabled buttons in the middle, and Short Cut adds the necessary instructions to save that data. In the example below, average air temperature, a sample of relative humidity, and average barometric pressure were selected. Short Cut enables the most logical outputs for each measurement. If you require an output that is not enabled you can right-click on the measurement to get a pop-up menu containing all output options. You can also select a block of measurements (left-click and shift+left-click) to do the same output on all of them. Note however that only output options common to all of the selected measurements will be enabled.



Note that outputs for a sensor don't have to be added in the same sequence as the measurement. You can even drag and drop the outputs to rearrange their order. Note also that multiple outputs can be added for any one sensor. For example, you may want to store the maximum and minimum air temperature as well as the average.

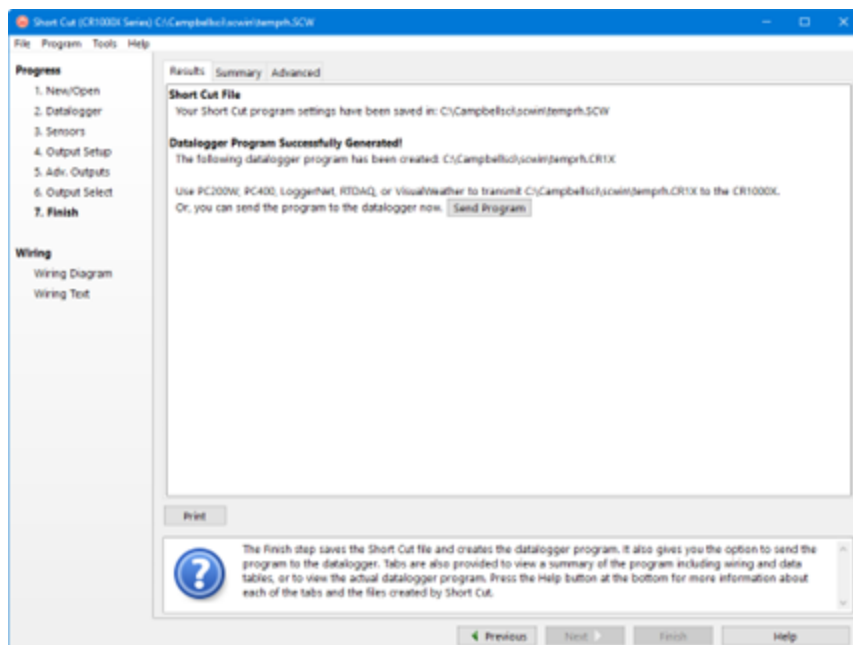
An **Output Label** can be changed by selecting it and making the desired modifications.

If **Advanced Outputs** was selected on the **Output Setup** screen, there will also be a column for **Resolution**. By default, data is stored in low resolution (2-byte floating point numbers). You can instead select high resolution to have data stored as 4-byte floating point numbers.

When you've configured all of your outputs, click **Finish**.

### 5.1.1.7 Step 7 – Generate the Program in the Format Required by the Datalogger

The **Finish** button completes the process. If you haven't yet saved the program, Short Cut asks for a program name and offers the default directory within its program working directory (default is C:\Campbellsci\SCWin). Short Cut also displays a Results, Summary, and Advanced window:



The **Results** tab provides information on the files that were created. If a program was created successfully, a **Send Program** button will also be displayed which allows you to send the program to the datalogger.

The files generated by Short Cut are as follows:

- *ProgramName*.SCW ("Example.SCW" in this example) at the top of the screen is the file in which Short Cut keeps all of your selections for datalogger, sensors, outputs, etc.
- For CR10, CR10X, CR500/510, CR23X, and 21X dataloggers (including mixed-array, table-data and PakBus operating systems), *ProgramName*.DLD is the ASCII text file that must be sent to the datalogger for it to make the measurements and store the data you want. For CR1000X-series, CR1000, CR6-series, CR3000, CR5000, CR800-series, CR300-series, and CR9000X dataloggers, this file will be the .CR1X, .CR1, .CR6, .CR3, .CR5, .CR8, .CR300, or .C9X file. For GRANITE Data Logger Modules, this file will be a .CRB file. For CR200 dataloggers, this file will be a .BIN (binary image) file.

- *ProgramName*.DEF is the text file that describes the wiring for the sensors and devices to the datalogger, measurement labels, flag usage, and the output expected. You can view the contents of the DEF file by clicking the **Summary** button on the Results screen.
- For mixed-array dataloggers, *ProgramName*.FSL is a text file containing output labels (created for mixed-array dataloggers only). This file can be used by Split or View or other software to provide column headers for the data file.

The **Summary** tab displays the information in the DEF file as described above.

The **Advanced** tab (for CRBasic dataloggers) displays the CRBasic program that was generated. It includes a **CRBasic Editor** button which opens the program for editing in the CRBasic Editor. Note that any changes made to the generated program in the CRBasic Editor will not be reflected in Short Cut or future programs generated by Short Cut.

Note that, while Short Cut can generate a program file for the datalogger, you must use datalogger communication software to transmit that program to the datalogger. (This is true even when pressing the **Send Program** button from Short Cut's Finish screen. Short Cut relies on the datalogger communication software to transmit the program.)

## 5.1.2 Short Cut Settings

The Program and Tools menus on the Short Cut menu offer several settings that may prove useful.

### 5.1.2.1 Program Security

Some dataloggers allow you to set security by entering one or more numbers into their security fields. You can allow different levels of access (e.g.; only allow data retrieval, or also allow monitoring of values, or also allow sending a new program or setting the clock) by entering multiple levels.

Datalogger security is not meant to be extremely tight. Rather, it is designed to prevent honest people from making mistakes.

Notwithstanding its intention, one mistake you can make is to set security and then forget the values. If you send a program with security set, you will then need to add that security setting to LoggerNet's Setup Screen or RTDAQ or PC400's EZSetup Wizard for that datalogger. If you don't, you may find that you can no longer communicate with the datalogger. Should this happen and you forget the security code and have lost the Short Cut program file, you may have to visit the datalogger site and cycle power on the datalogger to be able to communicate with it. Most dataloggers that offer security will communicate over their CS I/O port directly with a

keyboard/display or PC in the first few seconds of powering up. See the datalogger manual for a full description of the security features.

### **5.1.2.2 Datalogger ID**

Mixed-array dataloggers keep a memory location available for a datalogger ID value. This is typically an integer that you can read from within the program and store into final storage to keep track of the identity of the datalogger that created the data. Valid Datalogger IDs are 1 through 12 and 14 through 254. Use the Datalogger ID instruction in Short Cut (found under Miscellaneous Sensors) to use the ID in the datalogger program.

### **5.1.2.3 Power-up Settings**

Some dataloggers offer the option to retain interim measurements or calculations or the states of flags or ports when they power-up from a low battery or loss of power condition. This may be useful when calculations are used to control devices. You may, for example, want to ensure that pumps or controls are off when a datalogger powers up so as to make the control decision based on a fresh measurement. See the datalogger manual for a full description of this feature.

### **5.1.2.4 Select CR200 Compiler**

Use this setting to select the directory and executable name that will be used to pre-compile the CR200/205 program to check for errors.

Most Campbell Scientific dataloggers are sent an ASCII program file, which they then compile into machine code. The CR200/205 does not have enough memory and processing capability to do this compilation, so it's necessary to compile the program file into the binary version used by the datalogger itself. This compilation is done by Short Cut to check for errors in the program before sending it. It's done again by LoggerNet, RTDAQ, or PC400 when sending the program to the datalogger. Compilation is performed using a special executable that mimics the functions and capability in the datalogger's operating system. Therefore, the compiler executable must match the datalogger's operating system or the datalogger may fail to run the compiled binary (\*.BIN) program. LoggerNet, RTDAQ, PC400, and Short Cut are installed with precompilers for all of the released versions of the CR200/205 operating systems. If, at some time in the future, you acquire a newer CR200/205, or choose to install a later operating system, you must make sure you also have the compiler executable that matches. These compiler executables are typically installed in a library directory. By default, this directory would be installed as:

C:\Campbellsci\Lib\CR200Compilers

If you receive an operating system update, you should copy the compiler associated with it to this directory. If, for some reason, you put the compiler in a different directory, this menu item provides a way to choose that compiler executable.

### **5.1.2.5 Sensor Support**

The Sensor Support option is used to select which group of sensor files will be displayed when creating a program: Campbell Scientific, Inc., (CSI) or Campbell Scientific, Ltd. (CSL). The standard set of Short Cut sensor files was created by CSI; however, CSL has created some additional files that are customized for their client base. When one option is selected, the sensor files developed specifically for the other are filtered out.

This dialog box is displayed the very first time you create a program for a specific datalogger type; it will not be displayed thereafter. With each subsequent program you create, the group of sensor files that you chose when the datalogger was initialized in Short Cut will be used. However, you can change this setting at any time. If you make a change, the setting will remain in effect for all programs for that datalogger type (whether they are new programs or edited programs) until it is changed again.

### **5.1.2.6 First Notch Frequency ( $f_{N1}$ )/Integration**

Some dataloggers have parameters available in their measurement instructions to provide integration for rejection of noise due to AC electrical signals. These parameters will be used by Short Cut if possible, but the frequency of this noise varies. In most of North America, the AC frequency is 60 Hz. However, in many countries the frequency is 50 Hz. If you know the frequency of this AC noise, you can select one or the other frequency. Fast integration should be used when you need an execution speed that cannot be accomplished using one of the other options. This setting remains in effect for other programs generated by Short Cut until you change it.

### **5.1.2.7 Font**

This setting is accessed from the Options menu item of the Tools menu. Use this setting to change the appearance of the font used by Short Cut. Most windows other than the wiring descriptions (which require a non-proportional font to make sure wiring diagrams are aligned) will use this font.

### **5.1.2.8 Set Working Directory**

This setting is accessed from the Options menu item of the Tools menu. This setting changes the directory that Short Cut offers as a default for your programs. Upon installation, the default is set to C:\CampbellSci\SCWIN.



### 5.1.2.9 Enable Creation of Custom Sensor Files

This setting is accessed from the Options menu item of the Tools menu. It allows the user to create custom sensor files as described in [Custom Sensor Files](#) (p. 54).

## 5.1.3 Editing Programs Created by Short Cut

Short Cut is very flexible and has many features. It does not, however, support all of the functionality in Campbell Scientific dataloggers. Some users will need to develop programs with capabilities beyond that offered by Short Cut, but will want to take advantage of the library of instructions and settings known to a program generator in order to get a head start.

For Edlog dataloggers, the easiest method is to Document the DLD file from within Edlog (discussed later in this section). Short Cut creates a .DLD file to send to the datalogger that includes input location and final storage labels. Documenting a .DLD file causes Edlog to use the same labels and to show you the individual instructions being used to carry out the program. You can then add and delete instructions from within Edlog to add functionality to the program. Short Cut cannot import the files created by Edlog, however. Short Cut reads only its own SCW-formatted files.

For CRBasic dataloggers, you can use the CRBasic Editor to open the .CR# files directly. Again, Short Cut will not be able to open the files you've edited with the CRBasic Editor, since they are not an SCW file.

## 5.1.4 New Sensor Files

Short Cut was designed with future flexibility in mind. Datalogger and sensor support is provided as individual files and not part of the SCWIN executable. As new dataloggers and sensors become available, new definition files will be created to add and modify the necessary features known to Short Cut. To update these files, you can download the latest version of Short Cut from the Campbell Scientific website:

[www.campbellsci.com/downloads](http://www.campbellsci.com/downloads)

It is also possible to have custom sensor files created for sensors your organization uses that are not included with Short Cut. Contact your Campbell Scientific applications engineer for details.

## 5.1.5 Custom Sensor Files

The creation of custom sensor files can be enabled from Short Cut's Tools | Options menu item. Once enabled, custom sensor files can be created by right-clicking on a sensor in the Available Sensors and Devices list and choosing Create Custom Sensor.

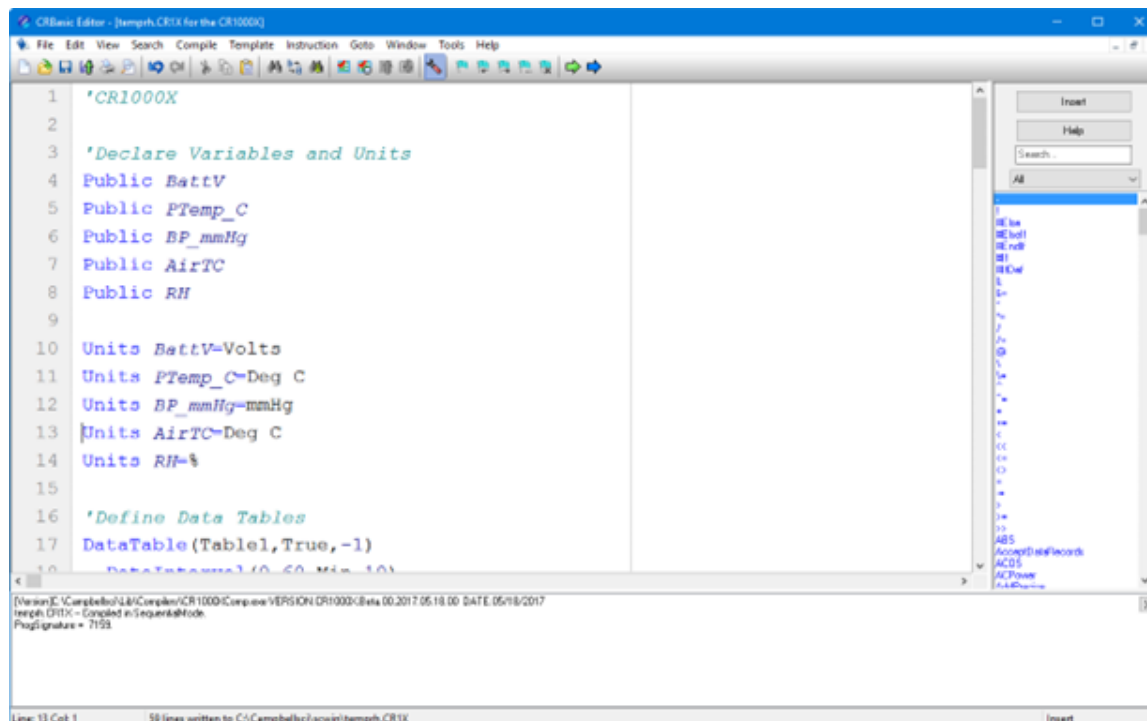
The resulting dialog box will allow the user to make changes to the chosen sensor file and then save it with a new name. (See Short Cut's Online Help for additional information on changes that can be made.) By default, custom sensor files will be created in C:\CampbellSci\SCWin\SENSORS, which is a different location than that of Short Cut's included sensor files.

Once the custom sensor file has been saved, it will be added to the Available Sensors list.

## 5.2 CRBasic Editor

The CRBasic Editor is a programming tool which can be used with the CR1000X-series, CR6-series, CR300-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, CR800-series, CR200-series, CR5000, CR9000, and CR9000X dataloggers. It is intended for use by experienced datalogger programmers who need more flexibility and control over the datalogger operation than what can be achieved using Short Cut. This programming language is similar in syntax, program flow, and logic to the Structured BASIC programming language.

As shown below, the CRBasic Editor's main window is divided into three parts: the Program Entry Window, the Instruction Panel, and the Message area. The Instruction Panel on the right side is a list that comprises the instructions for a particular datalogger in the CRBasic language. Instructions can be selected from this list or entered directly into the Program Entry Window on the left. The Message area at the bottom becomes visible after a program is compiled and shows results of the compile and any errors detected.



## 5.2.1 Inserting Instructions

An instruction can be easily inserted into the program by highlighting it in the Instruction Panel list and pressing **Insert** or by double-clicking the instruction name. If an instruction has one or more parameters, an instruction dialog box will be displayed to facilitate editing the parameters. Complete the information in the parameter fields and press **Insert** to paste the instruction into the program. (You may disable this instruction dialog box by clearing the option in the **View > Instruction Panel Preferences > Show Instruction Dialog** check box.)

You can filter the list of instructions available in the Instruction Panel by clicking the drop-down arrow to the right of the text box above the list. This will allow you to display only instructions of a specific type such as **Measurement** or **Program Structure/Control**. This provides a smaller list to select from and makes it easier to find the instruction you want. Switch back to **All** to see all of the instructions available. You can create custom instruction filter lists as described later in this section.

### 5.2.1.1 Parameter Dialog Box

The **Parameter** dialog box will appear when an instruction is added that has one or more parameters or when the cursor is placed on an existing instruction and the right mouse button is pressed. This dialog box contains a field for each of the parameters in the instruction. Edit these fields as necessary and then press **Insert** to paste the instruction into the program.

Below is an example of the **Parameter** dialog box for the differential voltage instruction (*VoltDiff*).

Parameter Type	Value	Comment
Destination	DiffVolt1	
Repetitions	8	
Voltage Range	mV2500	
DiffChan	1	
RevDiff	True	Measure second time Low referenced to High
SettlingTime	0	
Integration	_60Hz	16.667 ms integration
Multiplier	1.0	
Offset	0.0	

Variables:  
[Dropdown Menu]  
[Insert] [Cancel] [Help]  
[Prev] [Next]

The **Prev** (Previous) and **Next** buttons can be used to move to the next (or previous) instruction with the parameter entry box opened.

#### Short Cuts for Editing the Parameters

Right-clicking or pressing **F2** on a parameter that uses a variable as an input type will display a list of variables that have been defined in the program. A sample list is shown below.



The variable list is sorted by variable type and then alphabetically by name. In the list above, the first green A denotes that the variable *AIRCOOL* is set up as an Alias.

Constants are listed with a blue C, Dimensioned variables are listed with a red D, and Public variables are listed with a black P.

At any time you can press **F10** to bring up the list of variables, regardless of the input type for the selected parameter. Also, defined variables can be selected from the **Variables** drop-down list box at the upper right of the **Parameter** dialog box.

Pressing **F9** at any time will also bring up a list of variables. However, when a variable is chosen from the list brought up by **F9**, it will simply be inserted at the cursor without overwriting anything.

Right-clicking or pressing **F2** on a parameter that has a finite number of valid entries will bring up a list of those available options.

Right-clicking or pressing **F2** on a parameter that does not fall within the two categories above will bring up help for that parameter.

Pressing **F1** with any parameter selected will bring up help for that parameter along with a list of possible options where appropriate.

#### **Changing Default Parameters Values for an Instruction**

Each instruction offers default values for each parameter. For instance, in the **Parameter** box above, the default for the Range is mV5000. If you wanted to edit this so that each time you inserted the VoltDiff instruction the Range value defaulted to mV1000, you would highlight the instruction in the **Instruction Panel**, select **Instruction > Edit Instruction Defaults** from the menu, and make the change in the resulting dialog box.

### **5.2.1.2 Right-Click Functionality**

The result of a right-click action varies, depending upon your cursor location.

Right-click an instruction name to show the **Parameter** dialog box to edit the instruction parameters.

Right-click a parameter that uses a variable as an input type to bring up a list of variables that have been defined in the program as described in the previous section.

Right-click a parameter that has a finite number of valid entries to bring up a list of those available options. You can change the option by clicking the desired option.

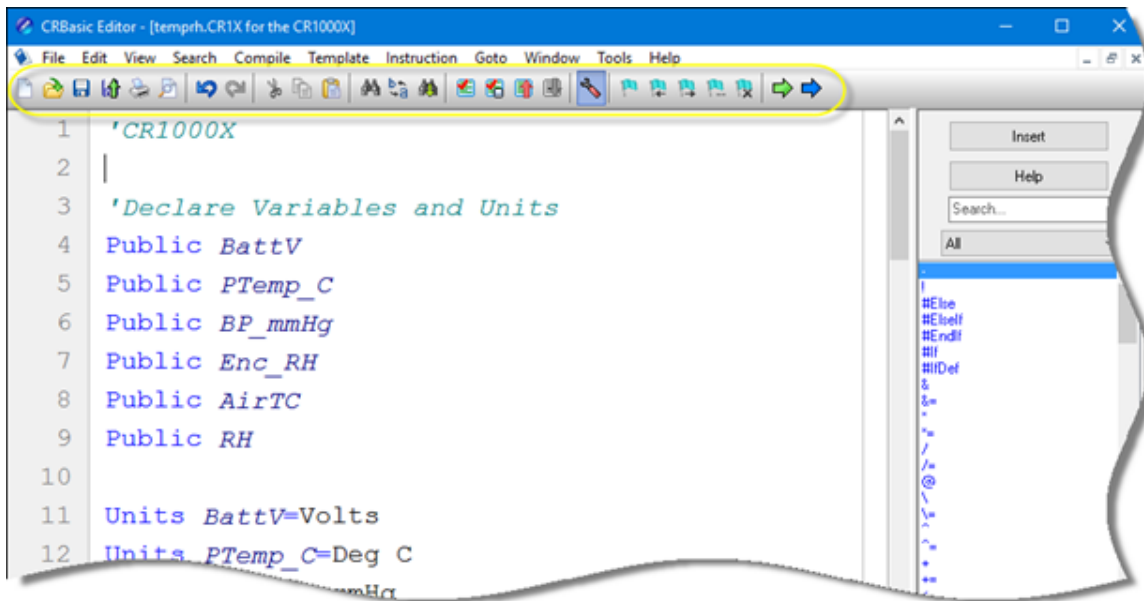
Right-click another type of parameter to bring up help for that parameter.

Right-click a block of text that is highlighted to bring up a short cut menu with the following options:

- **Comment/Uncomment Block:** Only one of these options will be available, depending upon the status of the highlighted text. If the text has been marked as a comment, you can choose to uncomment it. If the text is not commented, you can choose to make it into a comment. Commented text has a single quote ( ' ) at the beginning of the line. Comments are ignored by the datalogger's compiler.
- **Decrease/Increase Indent:** You can increase or decrease the indentation of the selected text. The spacing is increased or decreased by one.
- **Cut/Copy/Paste/Delete:** Standard editing functions can be accessed through this menu.
- **Save as .CRB File:** Saves highlighted text to a file with a \*.CRB extension. This file is referred to as a "library file". The file can then be reused by inserting it into another CRBasic program.
- **Insert File:** Inserts a library file into the current program overwriting the highlighted text.

## **5.2.2 Toolbar**

The toolbar of the CRBasic Editor provides easy access to frequently used operations.



**New** – Creates a new program window to start writing a new program. If you have defined a default template, the new program will start with the defined template instructions.



**Open** – Brings up a File Open dialog to select a program file to open. File extension filters are provided to list only files of a certain type such as .cr5 files for CR5000 programs. Data files (\*.dat) can also be opened.



**Save** – Saves any changes to the currently opened program. If this is a new program and has not been saved yet, a Save As dialog will prompt you for the file name and location to save the file. A table definition file (\*.tdf) of the same name as the saved program will also be created. Refer to the online documentation for more information about using table definition files.



**Compile, Save, and Send** – Saves any changes to the currently opened program, checks it for errors with the pre-compiler, and sends the file to the datalogger via LoggerNet, PC400, or RTDAQ. LoggerNet, PC400, or RTDAQ must be running for this function to work properly.



**Print** – Prints the currently opened program.



**Print Preview** – Opens a Print Preview screen that will show what the program will look like when printed. You can check and set the margins and printer options.



**Undo** – Each time the Undo button is clicked it will step back through the last changes made to the program.



**Redo** – Cancels the undo and steps forward restoring the changes.



**Cut** – Removes the selected part of the program and puts it on the clipboard to be pasted elsewhere.



**Copy** – Places a copy of the selected part of the program on the clipboard to be pasted elsewhere.



**Paste** – Inserts a copy of the contents of the clipboard into the program at the cursor location.



**Find** – Brings up a Find dialog to specify a text string to search for in the program. Click the **Find Next** button or press F3 to go to successive occurrences of the text.



**Replace** – Brings up a Find and Replace dialog that allows you to specify a text string to search for and a text string to replace it with. You can replace all occurrences of the text or check them one at a time to make sure they should be replaced.



**Find Next** – Finds the next occurrence of the text string specified in the Find dialog.



**Compile** – Starts the compiler to check the current program for errors and consistency. Compile results and errors will be displayed in the message area at the bottom of the screen.



**Save and Compile** – Saves and then compiles the opened file.



**Previous Error** – Moves the cursor to the part of the program where the previous error was identified.



**Next Error** – Moves the cursor to the part of the program where the next error was identified.



**Instruction Panel** – Controls whether the Instruction Panel is displayed. Hiding the Instruction Panel allows more room in the window to view the program.



**Toggle Bookmark** – Adds a bookmark to the line where the cursor resides. If a bookmark already exists, it will remove the bookmark.



**Previous Bookmark** – Moves backward to the previous bookmark in the program.



**Browse Bookmarks** – Displays a list of all bookmarks in the program. When a bookmark is selected, the cursor moves to that line in the program.



**Clear Bookmarks** – Erases all bookmarks from the program.



**GoTo** – Moves the cursor to a particular section of the program. Choose the section type from the list box that appears.



**User-Defined Functions and Subroutines** – Provides a list box containing all

of the user-defined functions and subroutines. Functions are identified with a purple F. Subroutines are marked with a black S. Clicking on a name in the list box moves the cursor to the start of that function or subroutine.

## 5.2.2.1 Compile

**Compile** is a function provided by the CRBasic Editor to help the programmer catch problems with the datalogger program. **Compile** is available from the toolbar and the **Compile** menu.

When the Compile function is used, the CRBasic Editor checks the program for syntax errors and other inconsistencies. The results of the check will be displayed in a message window at the bottom of the main window. If an error can be traced to a specific line in the program, the line number will be listed before the error. You can double-click an error preceded by a line number and that line will be highlighted in the program editing window. To move the highlight to the next error in the program, press the **Next Error** button or choose **Next Error** from the **Compile** menu. To move the highlight to the previous error in the program, press the **Previous Error** button or choose **Previous Error** from the **Compile** menu.

It is important that the compilers used for checking programs match the OS version loaded in the datalogger, otherwise errors may be returned when the program is sent. When a CR200 program is being edited, the **Pick CR200 Compiler** menu item is available. This item opens a dialog box from which a compiler can be selected for the CR200 datalogger.

The error window can be closed by selecting the **Close Message Window** menu item from the **View** menu, or by clicking the **X** in the upper right corner of the message window.

**NOTE:** For the CR1000X-series, CR6-series, CR300-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, CR800-series, CR5000, and CR9000X dataloggers, the Compile function only verifies the integrity of the program. Actual compilation of the program takes place in the datalogger. When using the CR200 datalogger, however, this function creates a binary image of the program to be loaded to the datalogger at a later time. This function is not available for the CR9000 datalogger.

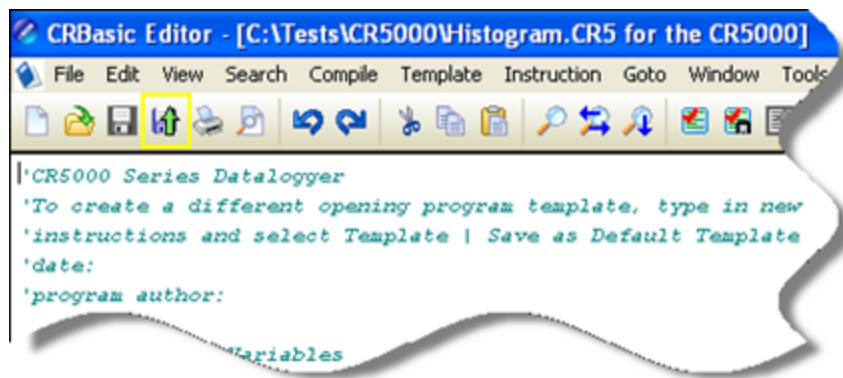
## 5.2.2.2 Compile, Save, and Send

The CRBasic Editor allows you to send a program to a datalogger that has already been defined on the network map in LoggerNet, PC400, or RTDAQ. This only works if LoggerNet, PC400, or RTDAQ is running at the time you attempt to send the program.

This function first checks the program for errors using the pre-compiler, then saves the program (using the current name, or by prompting the user for a name if the program is new). After the compile and save, this function sends the program to a user-specified datalogger. To do this, use

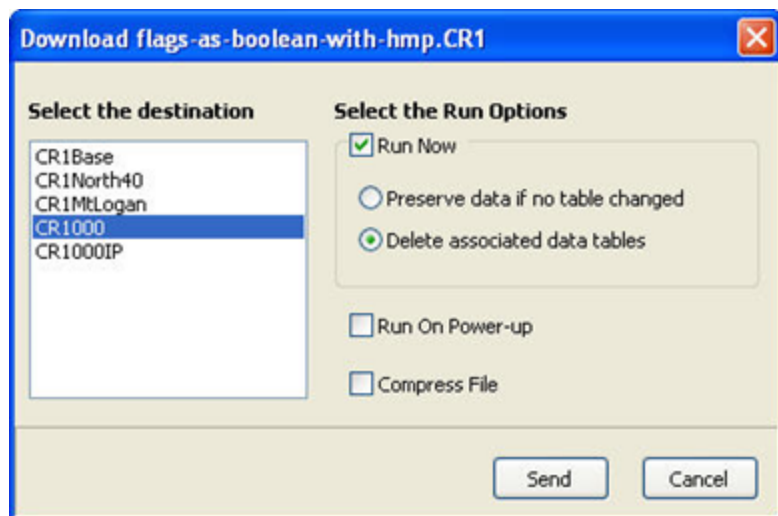


the **Compile, Save and Send** item on the **File** menu or **Compile** menu, or you can press the corresponding button on the toolbar.



**NOTE:** When a file is sent to the datalogger using Compile, Save, and Send and the software is not actively connected to the datalogger, the software connects to the datalogger, sends the file, retrieves table definitions, and then disconnects. There will be little indication in the software that a connection was established.

When this function is chosen a dialog box is displayed. Below is the dialog box for a CR1000 datalogger:



The **Select the destination** list shows all dataloggers configured within LoggerNet, PC400, or RTDAQ that may receive a program matching the extension of the current CRBasic program to be sent. Assume, for example, that you have three CR1000s and some other dataloggers in your LoggerNet, PC400, or RTDAQ network map. When you send a \*.CR1 program, this screen will show only the three CR1000 dataloggers. Any other dataloggers will be excluded from the list in this case, even when they are defined in the network map, because those dataloggers are not associated with \*.CR1 programs. A program with the extension of .DLD will be associated with all CRBasic-programmed datalogger types.

Select the datalogger to send the file to, and then select the Run Options.

**NOTE:** The CR200 does not have an on-board compiler. A compiled binary (\*.bin) file is sent to the datalogger. Run options are not applicable to this datalogger and are therefore disabled.

### Compress File

If the **Compress File** check box is selected, a renamed version of the CRBasic program which has all unnecessary spaces, indentation, and comments removed in order to minimize the file size will be sent to the datalogger instead of the original program.

### Sending the Program

To send the file and perform the associated functions you have selected in the screen, press the **Send** button. If LoggerNet, PC400, or RTDAQ is not running, an error message will appear indicating that there is no communications server currently running. If LoggerNet, PC400, or RTDAQ is running and the program compiles properly on the hardware, you will receive a message indicating that the program is now running on the datalogger. If something goes wrong when sending the program, a message will appear indicating the error conditions. This may be a hardware-level compile error or another failure as reported to the software by the datalogger's program load and run process.

Press **Cancel** if you do not wish to send the program to the datalogger.

**NOTE:** When sending a program with the Compile, Save, and Send feature to a CR9000X datalogger while you are connected to the datalogger, you may get a disconnect message or similar notification. This is unique to the CR9000X datalogger and does not indicate any problem with the sending of the program. You can simply reconnect to the datalogger and continue your work.

## 5.2.2.3 Conditional Compile and Save

The **Conditional Compile and Save** option is used to generate a new CRBasic program from code that uses conditional compile syntax (#If/Else/Elseif statements) or constant customization. (See conditional compilation in the CRBasic Editor's online help for more information on conditional compile syntax. [Constant Customization](#) (p. 70) See for more information on constant customization.)

When a program is compiled that uses conditional syntax, any conditional compilation statements that do not evaluate as true are removed from the program and the program is compiled. When a program is compiled that uses constant customization, the constant values selected in the **Tools > Customize Constants** menu item are used when compiling the new

program. In either instance, you are prompted to save the file under a user-specified name or the file will be saved under the name of the original program with `_CC#` appended. The `#` is a number that increments to create a unique filename. For instance, if the program name is `myprogram.cr1`, the first time it is compiled the default name will be `myprogram_CC1.cr1`. If `myprogram_CC1.cr1` exists, the program will be named `myprogram_CC2.cr1`.

## 5.2.2.4 Templates

The use of templates can be a powerful way to quickly create a set of similar datalogger programs. All or part of a program can be saved so that it can be used when creating new programs. These files are called templates. The **Template** menu provides access to create and use templates.

- **Save as Template** – Saves the comments and instructions in the active file as a template. To save part of a program as a template, copy the selected part to a new program file and then **Save as Template**.
- **Save as Default Template** – Saves the comments and instructions in the active file as a template that will be used each time **File > New** is selected for that type of datalogger.
- **Delete** – When selected, a list of all dataloggers is displayed. Select a datalogger to open a dialog box containing a list of saved templates. A template can then be highlighted and deleted from disk.
- **(Datalogger Types)** – When a datalogger type is selected, a list of all templates is displayed.

**NOTE:** Template files are associated with a specific datalogger type. For example, templates for a CR5000 cannot be used for CR9000X programming and vice versa. Each datalogger has its own set of instructions that may be different than the other.

## 5.2.2.5 Program Navigation using BookMarks and GoTo

Bookmarks are lines of code in the program that the user marks, which can be quickly navigated to using the **Next**, **Previous**, and **Browse Bookmark** functions. Buttons for the bookmark function are available on the toolbar or in the **GoTo > Bookmarks** menu. Selecting the **Toggle Bookmark** option will add a bookmark to a line. Selecting it a second time will remove the bookmark. When a line is bookmarked, the entire line will be highlighted with a color (the color can be changed using the **View > Editor Preferences** menu item). You can then navigate from bookmark to bookmark by selecting **Previous** or **Next**. All bookmarks can be removed from the program by selecting **Clear Bookmarks**. Bookmarks are persistent when you close a program (i.e., they are saved and will exist the next time the program is opened).

Programs have certain common instructions, such as the declaration of variables, data table definitions, subroutine and/or function declarations, the BeginProg/EndProg statements, and Scan/NextScan. The **Goto** function is used to move the cursor to the next occurrence of a common instruction in the program (**GoTo > Navigation** or choose the **GoTo** button from the toolbar). In addition, you can move to a particular line number in the program by selecting **GoTo > Go To Line**.

### 5.2.2.6 CRBasic Editor File Menu

Many of the functions available from the CRBasic Editor Toolbar are found in this menu. They have been discussed previously. Other options include:

- **Open as Read-Only** – Opens a copy of a program file in read-only view. In this mode, the file cannot be edited. However, you can copy text from a read-only file into another file. Read-only allows the same program to be opened twice – once in regular view and once in read-only view. This allows the user to examine multiple areas of a very large program at the same time.
- **Save and Encrypt** – Encrypts the active file. Encrypted files can be compiled in the datalogger but cannot be read by a user. (Refer to FileEncrypt in the CRBasic Editor's online help for dataloggers that support file encryption.)

### 5.2.2.7 CRBasic Editor Edit Menu

This menu item allows you to edit and manipulate the text currently being displayed in the Editor. Standard text editing functions such as **Cut**, **Copy**, **Paste**, **Delete**, **Select All**, **Undo** and **Redo** are found in this menu. Other options include:

- **Create Compressed File** – Creates a new file with a *\_str* extension. All user comments and line spacing in the program are removed from the file. Removing comments and spaces can significantly reduce the file size in larger programs.
- **Rebuild Indentation** – Reworks the indentation of loops, If/Then/Else statements and other logic nesting, and removes blank lines based on the Vertical Spacing rules (**Options > Editor Preferences, Vertical Spacing tab**).
- **Save As CRB** – Saves highlighted text to a file with a \*.CRB extension. This file is referred to as a library file. The file can then be reused by inserting it into another CRBasic program.
- **Insert File** – Inserts a library file (\*.CRB) into the current program at the location of the cursor.

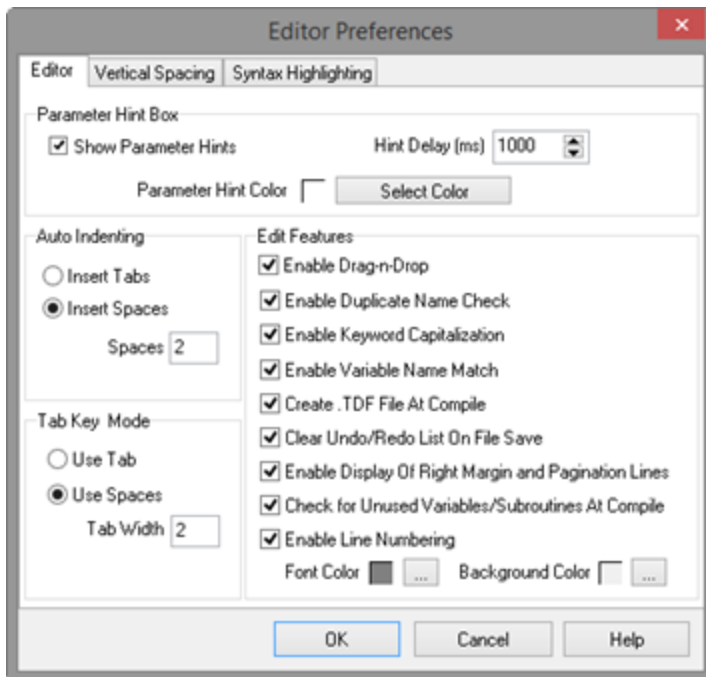
## 5.2.2.8 CRBasic Editor View Menu

This menu item allows you to specify the files used in the CRBasic Editor and customize its look and syntax highlighting.

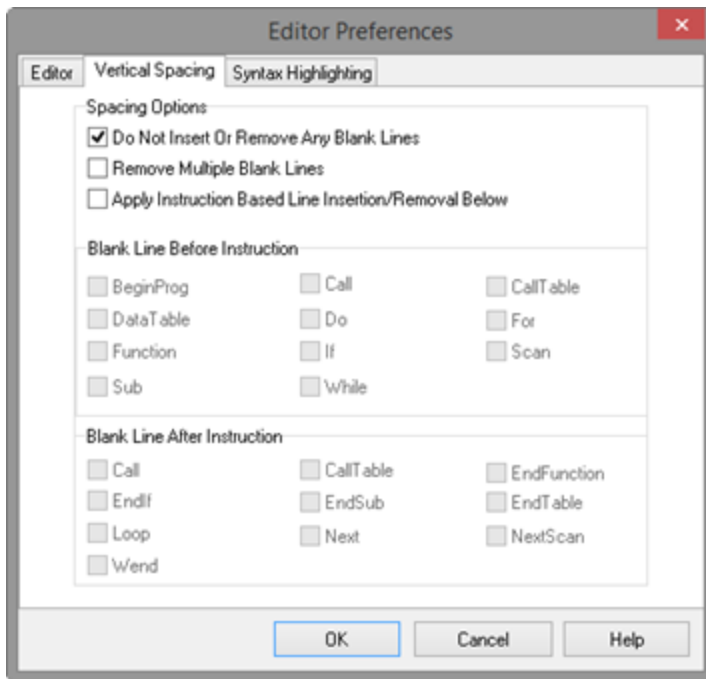
### Editor Preferences

This option sets up the appearance options for the text instructions and the behavior of pop-up hints.

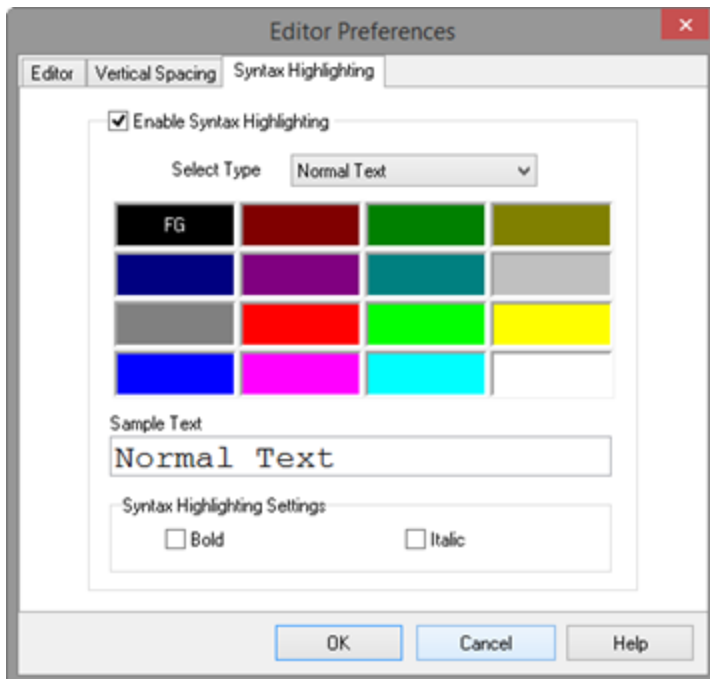
The **Editor** tab allows the user to toggle on or off the pop-up hints for parameters in instructions, set the amount of time the cursor must hover over the instruction before the pop-up hint appears, and the background color of the pop-up hint. This is also used to choose whether CRBasic automatic instruction indenting indents using tabs or spaces, and set the number of spaces if that option is chosen. Other options relating to the use of the tab key, capitalization, name checking, and line numbers are also available. Press the **Help** button for more information.



The **Vertical Spacing** tab is used to set up the rules for the CRBasic Editor's **Rebuild Indentation** function (**Edit > Rebuild Indentation**). You can control whether blank lines are inserted before or after certain instructions, and how the CRBasic Editor will process multiple blank lines in the program. If **Do Not Insert or Remove Any Blank Lines** is selected, all other fields on this tab will be disabled. If either of the other two line options is chosen, the remaining fields will be available for the user to customize as desired.



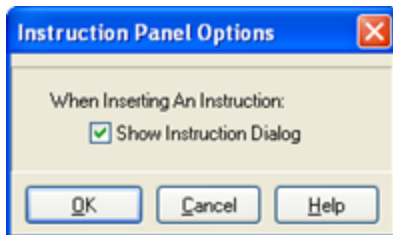
The **Syntax Highlighting** tab sets up the appearance of different text elements in the program using different font styles and colors. You can customize the appearance of the text by giving normal text, keywords, comments, operators, numbers, strings, and parentheses each a different font style and color to make the program easier to read and edit. Text colors and styles can be disabled by clearing the **Enable Syntax Highlighting** check box.



Note that if special formatting (font style, color) is assigned to **Matched Parentheses**, when your cursor is on an opening or closing parenthesis it will be highlighted with the formatting, and the "other half" of that parenthesis will also be highlighted. When your cursor moves off the parenthesis, the formatting will return to normal text.

## Instruction Panel Preferences

This option determines whether or not the instruction dialog box will be displayed when the user inserts an instruction.



## Other Options

**Font** – Displays a font selection dialog to select the font typeface and size for the text in the CRBasic Editor. Font style and color are set under **Editor Preferences**.

**Background Color** – Displays a color selection dialog to set the color of the CRBasic program window.

**Wrap Text When Printing** – When this option is selected, long lines that extend past the right margin will be wrapped to the next line. This option affects printing, as well as the **Print Preview** mode. A check mark will appear next to the option in the menu when it is selected.

**Display on Startup** – This option determines what is displayed when the CRBasic Editor is opened. Select **Start Page** to show a dialog box that allows you to select a recently used program, open any existing program, or choose a datalogger to write a new program for. Choose **Last Window Used** to open the datalogger program that was active in the CRBasic Editor when it was last closed. Select **New Window** to open the template for the datalogger that was last active in the CRBasic Editor.

**Close Message Window** – After you have pre-compiled your program with the **Compile > Compile** menu item, or using the toolbar, a message window opens up at the bottom of the CRBasic Editor main screen. This option will close down that message window.

**View Instruction Panel** – Select this option to View or Hide the instruction panel which displays a list of available instructions which can be used in your datalogger program based on the pre-defined instruction filter selected with the drop-down selection box.

## 5.2.2.9 CRBasic Editor Tools Menu

This menu item allows you to use special tools associated with the operation of the editor.

### Edit Instruction Categories

**Edit Instruction Categories** allows the user to create one or more custom list of instructions. If a category of instructions is selected from the Instructions Panel, the entire list of instructions in the Editor will be filtered to show only those instructions in the selected category. (Note: The default categories cannot be edited or deleted.)

To create a new list, first select the **Add New Category** button and provide a name for the user-created category. Next, ensure the category name is selected and click the **Edit Category** button to bring up the **Select Instructions** dialog (shown below). Instructions that should be included in the new list are indicated by a check in the box to the left of the instruction name. This feature allows the user to display a filtered instruction list containing only those instructions most often used. Press **OK** to save the list.

Edit Instruction Categories allows the user to create one or more custom list of instructions. If a category of instructions is selected from the Instructions Panel, the entire list of instructions in the Editor will be filtered to show only those instructions in the selected category. (Note: The default categories cannot be edited or deleted.)

To create a new list, first select the Add New Category button and provide a name for the user-created category. Next, ensure the category name is selected and click the Edit Category button to bring up the Select Instructions dialog (shown below). Instructions that should be included in the new list are indicated by a check in the box to the left of the instruction name. This feature allows the user to display a filtered instruction list containing only those instructions most often used. Press OK to save the list.

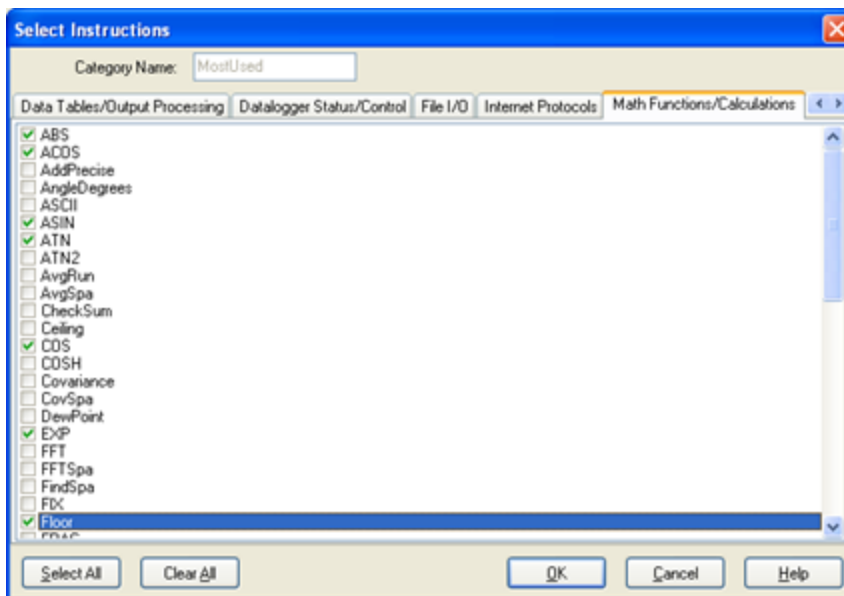
Edit Instruction Categories allows the user to create one or more custom list of instructions. If a category of instructions is selected from the Instructions Panel, the entire list of instructions in the Editor will be filtered to show only those instructions in the selected category. (Note: The default categories cannot be edited or deleted.)

To create a new list, first select the Add New Category button and provide a name for the user-created category. Next, ensure the category name is selected and click the Edit Category button to bring up the Select Instructions dialog (shown below). Instructions that should be included in the new list are indicated by a check in the box to the left of the instruction name. This feature allows the user to display a filtered instruction list containing only those instructions most often used. Press OK to save the list.



Edit Instruction Categories allows the user to create one or more custom list of instructions. If a category of instructions is selected from the Instructions Panel, the entire list of instructions in the Editor will be filtered to show only those instructions in the selected category. (Note: The default categories cannot be edited or deleted.)

To create a new list, first select the Add New Category button and provide a name for the user-created category. Next, ensure the category name is selected and click the Edit Category button to bring up the Select Instructions dialog (shown below). Instructions that should be included in the new list are indicated by a check in the box to the left of the instruction name. This feature allows the user to display a filtered instruction list containing only those instructions most often used. Press OK to save the list.



## Constant Customization

The Constant Customization feature allows you to define values for one or more constants in a program prior to performing a conditional compile (**Compile > Conditional Compile and Save** menu item). The constants can be set up with an edit box, a spin box field for selecting/entering a value, or with a list box. A step increase/decrease can be defined for the spin box, as well as maximum and minimum values.

To set up Constant Customization, place the cursor on a blank line within the CRBasic Editor and choose **Tools > Set Up Constant Customization Section**. This will insert two comments into the program:

*'Start of Constants Customization Section*

*'End of Constants Customization Section*

Within these two comments, define the constants. Following each constant, use the keywords noted below formatted as a comment, to set up edit boxes, spin boxes, or list boxes for the constant values. The fields are edit boxes by default. If a maximum/minimum are defined for a constant, the field will be a spin box. If a discrete list is defined for the constant, the field will be a list box.

Constant Name=(value)	Sets a default value for a constant
Min=(value)	Sets a minimum value for a spin box control
Max=(value)	Sets a maximum value for a spin box control
Inc=(value)	The number of steps for a value each time the up or down control on the spin box is selected
Value=(value)	Defines a pick list value

The Constant Customization syntax may be best understood by looking at an example. Consider the following program code:

```
'Start of Constants Customization Section
Const SInterval=10
'Min=5
'Max=60
'Inc=5

Const SUnits = sec
'value=sec
'value=min

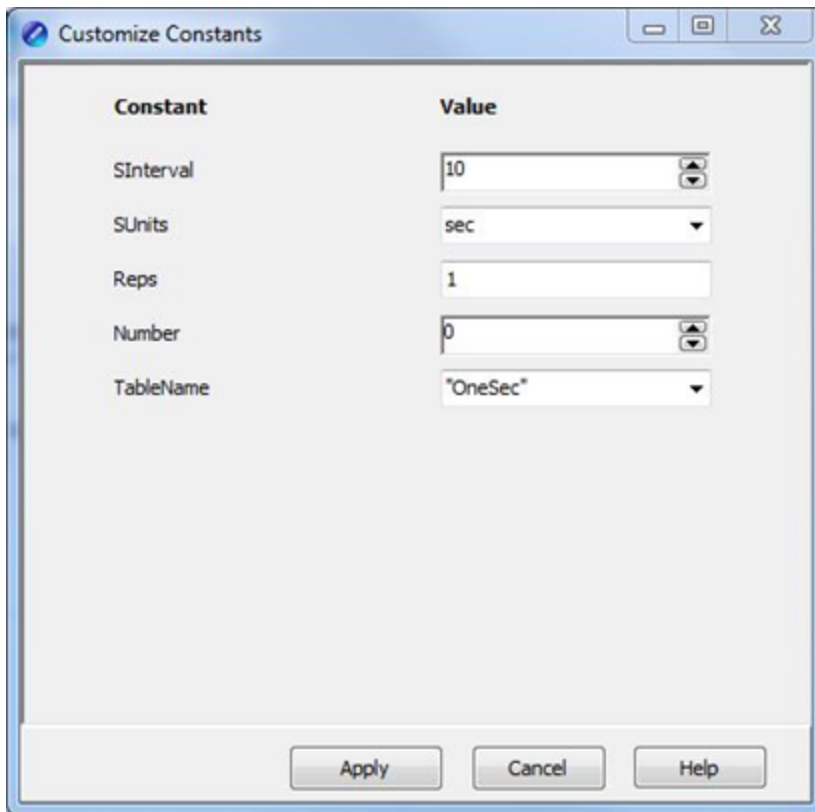
Const Reps=1

Const Number=0
'Min=-100
'Max=100

Const TableName="OneSec"
'value="OneMin"
'value="OneHour"
'value="OneDay"

'End of Constants Customization Section
```

This code will create the following constant customization dialog box:



The constant SInterval is defined with a default value of 10, a maximum of 60 and a minimum of 5, with a step of 5 each time the up or down control is selected.

The constant SUnits has a list box with sec and min; sec is the default.

The constant Reps is defined with a default value of 1. It is an edit box, into which any value can be entered.

The constant Number is defined with a default value of 0, a minimum of -100 and a maximum of 100. The value will increase by 1 each time the up or down control is selected.

The constant TableName is defined with a list box of "OneSec", "OneMin", "OneHour", and "OneDay"; the default value is "OneSec".

Before compiling the program, open the Customize Constants dialog box, select the constant values you want to compile into the program, and then perform the Conditional Compile and Save.

## Other Options

**Associate Files** – This option is used to set up file associations within the Windows operating system, so that if a program file is double-clicked while in Windows Explorer, that file will be opened in the CRBasic Editor.

Check one or more boxes for file extension(s) you want to associate and press the **Associate Files** button.

**Show Keyboard Shortcuts** – This option displays a list of the functions of the CRBasic Editor which are accessible via the keyboard. The list can be copied to the clipboard for printing or other uses.

**Show Tables** – This option displays details about the output tables and the items they store as they are defined in the current CRBasic program. The list can be copied to the clipboard for printing or other uses.

**Set Datalogger Type** – This option displays a list of dataloggers so the user can select the instruction set, compiler, and help files to use when the program extension is .DLD or .CRB (e.g., myprogram.DLD, or myprogram.CRB).

**Insert Symbol** – Opens a dialog box that lets you insert Unicode symbols into your CRBasic program for use in strings and units declarations.

**Set DLD Extension** – This option selects which datalogger's pre-compiler will be used when performing a pre-compile check on a DLD program which uses conditional compile statements. A CRBasic program must be named with the DLD extension for this item to be active.

**Open Display Settings File** – Opens a previously saved display setting file.

**Save Display Settings File** – The look and feel of the CRBasic Editor can be changed from the default. The Font and Background can be changed, as well as the syntax highlighting. These changes can be saved to a file (with an ini extension) using the Save Display Settings File menu item. The file can be reloaded on the same or different computer running CRBasic using the Open Display Settings File.

## **5.2.3 CRBasic Programming**

CRBasic is a programming language that has some similarities to a structured BASIC. There are special instructions for making measurements and for creating tables of output data. The results of all measurements are assigned variables (given names). Mathematical operations are written out much as they would be algebraically. This section provides a summary of a program, its syntax, structure, and sequence. Refer to the datalogger users manual or the on-line help for detailed information on program instructions.

### **5.2.3.1 Programming Sequence**

The structure of a datalogger program requires that variables, data tables, and subroutines be declared before they can be used. The best way to do this is to put all the variable declarations and output table definitions at the beginning, followed by the subroutines, and then the

program. Below is the typical layout of a program. Note that the CRBasic online help has example code for each instruction to demonstrate the use of the instruction in a program.

<b>Declarations</b>	<i>Make a list of what to measure and calculate.</i>
<b>Declare constants</b>	<i>Within this list, include the fixed constants used,</i>
<b>Declare Public variables</b>	<i>Indicate the values that the user is able to view while the program is running,</i>
<b>Dimension variables</b>	<i>the number of each measurement that will be made,</i>
<b>Define Aliases</b>	<i>and specific names for any of the measurements.</i>
<b>Define data tables</b>	<i>Describe, in detail, tables of data that will be saved from the experiment.</i>
<b>Process/store trigger</b>	<i>Set when the data should be stored. Are they stored when some condition is met? Are data stored on a fixed interval? Are they stored on a fixed interval only while some condition is met?</i>
<b>Table size</b>	<i>Set the size of the table in RAM.</i>
<b>Other on-line storage devices</b>	<i>Should the data also be sent to the external storage?</i>
<b>Processing of Data</b>	<i>What data are to be output (current value, average, maximum, minimum, etc.).</i>
<b>Define Subroutines</b>	<i>If there is a process or series of calculations that needs to be repeated several times in the program, it can be packaged in a subroutine and called when needed rather than repeating all the code each time.</i>
<b>Program</b>	<i>The program section defines the action of datalogging.</i>
<b>Set scan interval</b>	<i>The scan sets the interval for a series of measurements.</i>
<b>Measurements</b>	<i>Enter the measurements to make.</i>
<b>Processing</b>	<i>Enter any additional processing with the measurements.</i>
<b>Call Data Table(s)</b>	<i>The Data Table must be called to process output data.</i>
<b>Initiate controls</b>	<i>Check measurements and Initiate controls if necessary.</i>
<b>NextScan</b>	<i>Loop back (and wait if necessary) for the next scan.</i>
<b>End Program</b>	

### 5.2.3.2 Program Declarations

Variables must be declared before they can be used in the program. Variables declared as **Public** can be viewed in display software. Variables declared using DIM cannot be viewed. Variables assigned to a fixed value are used as constants.

For example, in a CRBasic program there may be multiple temperature (or other) sensors that are wired to sequential channels. Rather than insert multiple instructions and several variables, a variable array with one name and many elements may be used. A thermocouple temperature might be called TCTemp. With an array of 20 elements the names of the individual temperatures are TCTemp(1), TCTemp(2), TCTemp(3), ... TCTemp(20). The array notation allows compact code to perform operations on all the variables. For example, to convert ten temperatures in a variable array from C to F:

```
For I=1 to 10
  TCTemp(I)=TCTemp(I)*1.8+32
Next I
```

Aliases can also be created that will allow an element of an array or another data result to be referred to by a different name. To continue the example above, TCTemp(3) could be renamed using the following syntax:

```
Alias TCTemp(3) = AirTemp
```

In the display software, the more descriptive alias, AirTemp, would be used for the cell name.

### 5.2.3.3 Mathematical Expressions

Mathematical expressions can be entered algebraically into program code to perform processing on measurements, to be used for logical evaluation, or to be used in place of some parameters.

As an example of **Measurement Processing**, to convert a thermocouple measurement from degrees Celsius to degrees Fahrenheit, you could use the following expression:

```
TCTempF=TCTemp(1)*1.8+32
```

Logical Evaluation expressions could be used to determine the flow of a program:

```
If TCTemp(1) > 100 Then
  Call Subroutine1
Else
  'enter code for main program
EndIf
```

Many parameters will allow the entry of expressions. In the following example, the DataTable will be triggered, and therefore data stored, if TCTemp(1)>100.

```
DataTable(TempTable, TCTemp(1)>100, 5000)
```

### 5.2.3.4 Measurement and Output Processing Instructions

Measurement instructions are procedures that set up the measurement hardware to make a measurement and place the results in a variable or a variable array. Output processing instructions are procedures that store the results of measurements or calculated values. Output processing includes averaging, saving maximum or minimum, standard deviation, FFT, etc.

The instructions for making measurements and outputting data are not found in a standard basic language. The instructions Campbell Scientific has created for these operations are in the form of procedures. The procedure has a keyword name and a series of parameters that contain the information needed to complete the procedure. For example, the instruction for measuring the temperature of the CR5000 input panel is:

```
PanelTemp(Dest, Integ)
```

PanelTemp is the keyword name of the instruction. The two parameters associated with PanelTemp are: Destination, the name of the variable in which to put the temperature; and Integration, the length of time to integrate the measurement. To place the panel temperature in the variable RefTemp (using a 250 microsecond measurement integration time) the code is:

```
PanelTemp(RefTemp, 250)
```

### 5.2.3.5 Line Continuation

Line continuation allows an instruction or logical line to span one or more physical lines. This allows you to break up long lines of code into more readable “chunks”. Line continuation is indicated by one white space character that immediately precedes a single underscore character as the last character of a line of text. Following is an example of line continuation:

```
Public Temp, RH, WindSp, WindDir, _  
BatteryV, IntrRH, IntTemp, RainTot, _  
RainInt, Solar
```

### 5.2.3.6 Inserting Comments into Program

It is often useful to provide comments in your datalogger program so that when you review the program at a later date, you will know what each section of code does. Comments can be inserted into the program by preceding the text with a single quote. When the program is compiled, the datalogger compiler will ignore any text that is preceded by a single quote. A comment can be placed at the beginning of a line or it can be placed after program code. If Syntax Highlighting is enabled (**Options > Editor Preferences > Syntax Highlighting**), commented text will appear formatted differently than other lines of code.

```
'CR1000X
```

*'The following program is used to measure 4 thermocouples*

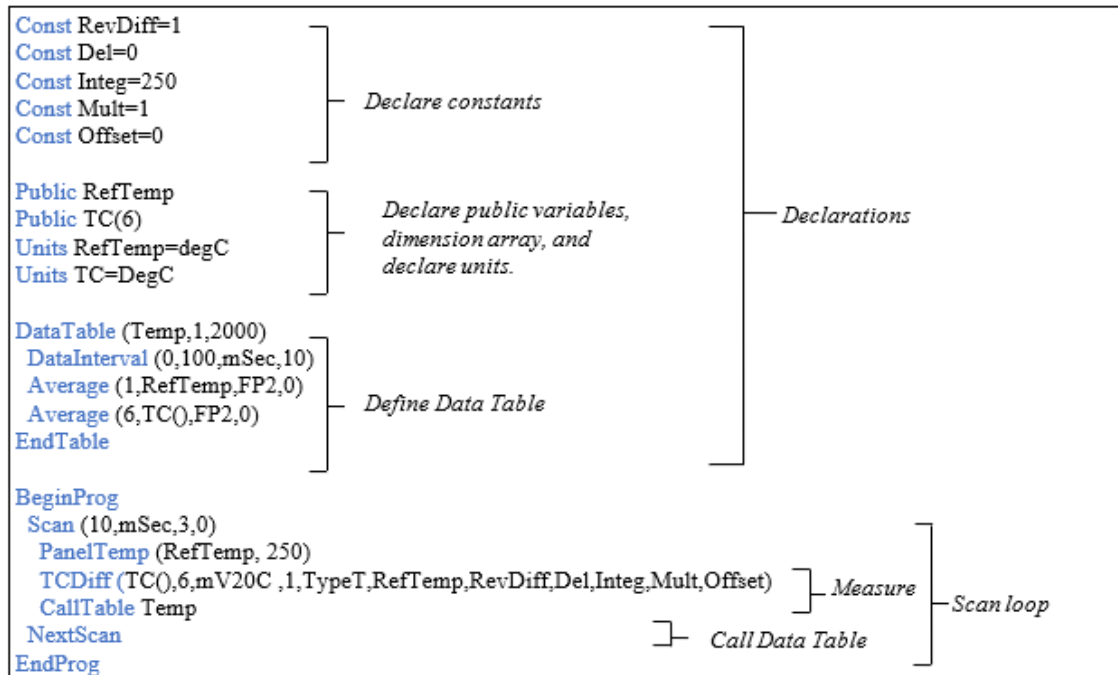
*'VARIABLE DECLARATION*

```
Dim TCTemp(4) 'Dimension TC measurement variable
Alias TCTemp(1)=EngineCoolantT 'Rename variables
Alias TCTemp(2)=BrakeFluidT
Alias TCTemp(3)=ManifoldT
Alias TCTemp(4)=CabinT
```

In the sample code above, the datalogger compiler will ignore the commented text.

### 5.2.3.7 Example Program

The following program will serve as a programming example in the following sections to illustrate the concepts and program structure. This is a program for a CR5000 datalogger. Note that other dataloggers may have slightly different parameters for some instructions.



### 5.2.3.8 Data Tables

Data storage follows a fixed structure in the datalogger in order to optimize the time and space required. Data are stored in tables such as:



TOA5 TIMESTAMP TS	StaName RECORD RN	Temp RefTemp_Avg degC Avg	TC_Avg(1) DegC Avg	TC_Avg(2) degC Avg	TC_Avg(3) degC Avg	TC_Avg(4) degC Avg	TC_Avg(5) degC Avg	TC_Avg(6) degC Avg
1995-02-16 15:15:04.61	278822	31.08	24.23	25.12	26.8	24.14	24.47	23.76
1995-02-16 15:15:04.62	278823	31.07	24.23	25.13	26.82	24.15	24.45	23.8
1995-02-16 15:15:04.63	278824	31.07	24.2	25.09	26.8	24.11	24.45	23.75
1995-02-16 15:15:04.64	278825	31.07	24.21	25.1	26.77	24.13	24.39	23.76

The user's program determines the values that are output and their sequence. The datalogger automatically assigns names to each field in the data table. In the above table, TIMESTAMP, RECORD, RefTemp\_Avg, and TC\_Avg(1) are fieldnames. The fieldnames are a combination of the variable name (or alias if one exists) and a three letter mnemonic for the processing instruction that outputs the data. Alternatively, the FieldNames instruction can be used to override the default names.

The data table header may also have a row that lists units for the output values. The units must be declared for the datalogger to fill this row out (e.g., Units RefTemp = degC). The units are strictly for the user's documentation; the datalogger makes no checks on their accuracy.

The above table is the result of the data table description in the example program:

```
DataTable(Temp,1,2000)
  DataInterval(0,10,msec,10)
  Average(1,RefTemp,fp2,0)
  Average(6,TC(1),fp2,0)
EndTable
```

All data table descriptions begin with DataTable and end with EndTable. Within the description are instructions that tell what to output and the conditions under which output occurs.

```
DataTable(Name, Trigger, Size)
DataTable(Temp,1,2000)
DataInterval(TintoInt, Interval, Units, Lapses)
DataInterval(0,10,msec,10)
```

The DataTable instruction has three parameters: a user specified name for the table, a trigger condition, and the size to make the table in RAM. The trigger condition may be a variable, expression, or constant. The trigger is true if it is not equal to 0. Data are output if the trigger is true and there are no other conditions to be met. No output occurs if the trigger is false (=0). The size is the number of records to store in the table. You can specify a fixed number, or enter -1 to have the datalogger auto allocate the number of records. The example creates a table name Temp, outputs any time other conditions are met, and retains 2000 records in RAM.

```
DataInterval(TintoInt, Interval, Units, Lapses)
DataInterval(0,10,msec,10)
```

DataInterval is an instruction that modifies the conditions under which data are stored. The four parameters are the time into the interval, the interval on which data are stored, the units for time,

and the number of lapses or gaps in the interval to track. The example outputs at 0 time into (on) the interval relative to real time, the interval is 10 milliseconds, and the table will keep track of 10 lapses. The DataInterval instruction reduces the memory required for the data table because the time of each record can be calculated from the interval and the time of the most recent record stored. The DataInterval instruction for the CR200 does not have lapses.

**NOTE:** Event driven tables should have a fixed size rather than allowing them to be allocated automatically. Event driven tables that are automatically allocated are assumed to have one record stored per second in calculating the length. Since the datalogger tries to make the tables fill up at the same time, these event driven tables will take up most of the memory leaving very little for the other, longer interval, automatically allocated data tables.

The output processing instructions included in a data table declaration determine the values output in the table. The table must be called by the program using the CallTable (Tablename) instruction in order for the output processing to take. That is, each time a new measurement is made, the data table is called. When the table is called, the output processing instructions within the table process the current inputs. If the trigger conditions for the data table are true, the processed values are output to the data table. In the example below, several averages are output.

*Average(Reps, Source, DataType, DisableVar)*

*Average(1, RefTemp, fp2, 0)*

*Average(6, TC(1), fp2, 0)*

Average is an output processing instruction that will output the average of a variable over the output interval. The parameters are repetitions (the number of elements in an array to calculate averages for), the Source variable or array to average, the data format to store the result in (see following table), and a disable variable that allows excluding readings from the average if conditions are not met. A reading will not be included in the average if the disable variable is not equal to 0; the example has 0 entered for the disable variable so all readings are included in the average.

Formats for Output Data				
Code	Data Format	Size	Range	Resolution
FP2	Campbell Scientific floating point	2 bytes	±7999	13 bits (about 4 digits)
IEEE4	IEEE four byte floating point	4 bytes	1.8 E –38 to 1.7 E 38	24 bits (about 7 digits)
LONG	4 byte Signed Integer	4 bytes	–2,147,483,648 to +2,147,483,647	1 bit (1)

### 5.2.3.9 The Scan—Measurement Timing and Processing

Once the measurements and calculations have been listed and the output tables defined, the program itself may be relatively short. The executable program begins with BeginProg and ends with EndProg. The measurements, processing, and calls to output tables bracketed by the Scan and NextScan instructions determine the sequence and timing of the datalogging.

BeginProg

Scan(1,MSEC,3,0)

PanelTemp(RefTemp, 250)

TCDiff(TCC),6,mV50,4,1,TypeT,RefTemp,RevDiff,Del,Integ,Mult,Offset)

CallTable Temp

NextScan

EndProg

The Scan instruction determines how frequently the measurements within the scan are made:

Scan(Interval, Units, BufferOption, Count)

Scan(1,MSEC,3,0)

The Scan instruction has four parameters (the CR200 datalogger's Scan instruction has only two). The Interval is the time between scans. Units are the time units for the interval. The BufferSize is the size (in the number of scans) of a buffer in RAM that holds the raw results of measurements. Using a buffer allows the processing in the scan to at times lag behind the measurements without affecting the measurement timing (see the scan instruction in the CR5000 help for more details). Count is the number of scans to make before proceeding to the instruction following NextScan. A count of 0 means to continue looping forever (or until ExitScan). In the example the scan is 1 millisecond, three scans are buffered, and the measurements and output continue indefinitely.

### 5.2.3.10 Numerical Entries

In addition to entering regular base 10 numbers there are 3 additional ways to represent numbers in a program: scientific notation, binary, and hexadecimal.

Formats for Entering Numbers in CRBasic		
Format	Example	Value
Standard	6.832	6.832
Scientific notation	5.67E-8	5.67X10-8
Binary:	&B1101	13
Hexadecimal	&HFF	255

The binary format makes it easy to visualize operations where the ones and zeros translate into specific commands. For example, a block of ports can be set with a number, the binary form of which represents the status of the ports (1= high, 0=low). To set ports 1, 3, 4, and 6 high and 2, 5, 7, and 8 low; the number is &B00101101. The least significant bit is on the right and represents port 1. This is much easier to visualize than entering 72, the decimal equivalent.

## 5.2.3.11 Logical Expression Evaluation

What is True? .....	81
Expression Evaluation .....	81
Numeric Results of Expression Evaluation .....	82

### What is True?

Several different words are used to describe a condition or the result of a test. The expression,  $X > 5$ , is either true or false. However, when describing the state of a port or flag, on or off or high or low is more intuitive. In CRBasic there are a number of conditional tests or instruction parameters, the result of which may be described with one of the words in the following table. The datalogger evaluates the test or parameter as a number; 0 is false, not equal to 0 is true.

Synonyms for True and False		
Predefined Constant	True (-1)	False (0)
Synonym	High	Low
Synonym	On	Off
Synonym	Yes	No
Synonym	Trigger	Do Not Trigger
Number	$\neq 0$	0
Digital port	5 Volts	0 Volts

### Expression Evaluation

Conditional tests require the datalogger to evaluate an expression and take one path if the expression is true and another if the expression is false. For example:

If  $X \geq 5$  then  $Y = 0$

will set the variable Y to 0 if X is greater than or equal to 5.

The datalogger will also evaluate multiple expressions linked with and or or. For example:

If  $X \geq 5$  and  $Z=2$  then  $Y=0$

will set  $Y=0$  only if both  $X \geq 5$  and  $Z=2$  are true.

If  $X \geq 5$  or  $Z=2$  then  $Y=0$

will set  $Y=0$  if either  $X \geq 5$  or  $Z=2$  is true (see And and Or in the help). A condition can include multiple and and or links.

## **Numeric Results of Expression Evaluation**

The datalogger's expression evaluator evaluates an expression and returns a number. A conditional statement uses the number to decide which way to branch. The conditional statement is false if the number is 0 and true if the number is not 0. For example:

If 6 then  $Y=0$ ,

is always true,  $Y$  will be set to 0 any time the conditional statement is executed.

If 0 then  $Y=0$

is always false,  $Y$  will never be set to 0 by this conditional statement.

The expression evaluator evaluates the expression,  $X \geq 5$ , and returns  $-1$ , if the expression is true, and 0, if the expression is false.

$W=(X>Y)$

will set  $W$  equal to  $-1$  if  $X>Y$  or will set  $W$  equal to 0 if  $X \leq Y$ .

The datalogger uses  $-1$  rather than some other non-zero number because the and and or operators are the same for logical statements and binary bitwise comparisons. The number  $-1$  is expressed in binary with all bits equal to 1, the number 0 has all bits equal to 0. When  $-1$  is anded with any other number the result is the other number, ensuring that if the other number is non-zero (true), the result will be non-zero.

## **5.2.3.12 Flags**

Any variable can be used as a flag as far as logical tests in CRBasic are concerned. If the value of the variable is non-zero the flag is high. If the value of the variable is 0 the flag is low. LoggerNet, PC400, or RTDAQ looks for the variable array with the name **Flag** when the option to display flag status is selected from the Connect Screen. If a **Flag** array is found, as many elements of that array which can fit will be displayed in the Port and Flags dialog box.

## **5.2.3.13 Parameter Types**

Instruction parameters allow different types of inputs. These types are listed below and specifically identified in the description of the parameter in the following sections or in CRBasic

help.

- Constant
- Variable
- Variable or Array
- Constant, Variable, or Expression
- Constant, Variable, Array, or Expression
- Name
- Name or list of Names
- Variable, or Expression
- Variable, Array, or Expression

The following table lists the maximum length and allowed characters for the names for Variables, Arrays, Constants, etc.

Rules for Names		
Name for	Maximum Length (number of characters)	Allowed characters
Variable or Array	39	Letters A-Z, upper or lower case, underscore "_", and numbers 0-12. The name must start with a letter. CRBasic is not case sensitive.
Constant	39	
Alias	39	
Data Table Name	20	
Field name	39	

## Expressions in Parameters

Many parameters allow the entry of expressions. If an expression is a comparison, it will return –1 if the comparison is true and 0 if it is false. An example of the use of this is in the DataTable instruction where the trigger condition can be entered as an expression. Suppose the variable TC (1) is a thermocouple temperature:

`DataTable(Name, TrigVar, Size)`  
`DataTable(Temp, TC(1)>100, 5000)`

Entering the trigger as the expression, TC(1)>100, will cause the trigger to be true and data to be stored whenever the temperature TC(1) is greater than 100.

## Arrays of Multipliers and Offsets for Sensor Calibration

If variable arrays are used for the multiplier and offset parameters in measurements that use repetitions, the instruction will automatically step through the multiplier and offset arrays as it steps through the channels. This allows a single measurement instruction to measure a series of individually calibrated sensors, applying the correct calibration to each sensor. If the multiplier and offset are not arrays, the same multiplier and offset are used for each repetition.

`VoltSE(Dest, Reps, Range, SEChan, Delay, Integ, Mult, Offset)`

*'Calibration factors:'*

`Mult(1)=0.123 : Offset(1)= 0.23`

`Mult(2)=0.115 : Offset(2)= 0.234`

`Mult(3)=0.114 : Offset(3)= 0.224`

`VoltSE(Pressure(), 3, mV1000, 6, 1, 1, 100, Mult(), Offset())`

Note that one exception to this is when the Multiplier or Offset points to an index into the array, then the instruction will not advance to the next Multiplier or Offset but use the same for each repetition. For instance in the above example, if `Mult(2)` and `Offset(2)` were used, the instruction would use 0.115 and 0.234 for the Multiplier and Offset, respectively, for each repetition. To force the instruction to advance through the Multiplier and Offset arrays while still specifying an index into the array, use the syntax `Mult(2)()` and `Offset(2)()`.

### 5.2.3.14 Program Access to data Tables

Data stored in a table can be accessed from within the program. The format used is:

`Tablename.FieldName(fieldname index, records back)`

Where `Tablename` is the name of the table in which the desired value is stored. `FieldName` is the name of the field in the table. The `fieldname` is always an array even if it consists of only one variable; the `fieldname index` must always be specified. `Records back` is the number of records back in the data table from the current time (1 is the most recent record stored, 2 is the record stored prior to the most recent). For example, the expression:

`Tdiff=Temp.TC_Avg(1,1)-Temp.TC_Avg(1,101)`

could be used in the example program to calculate the change in the 10 ms average temperature of the first thermocouple between the most recent average and the one that occurred a second (100 x 10 ms) earlier.

In addition to accessing the data actually output in a table, there are some pseudo fields related to the data table that can be retrieved:

`Tablename.record(1,n)` = the record number of the record output `n` records ago.

*Tablename.output(1,1)* = 1 if data were output to the table the last time the table was called, = 0 if data were not output.

*Tablename.timestamp(m,n)* = element m of the timestamp output n records ago where:

- *timestamp(1,n)* = microseconds since 1990
- *timestamp(2,n)* = microseconds into the current year
- *timestamp(3,n)* = microseconds into the current month
- *timestamp(4,n)* = microseconds into the current day
- *timestamp(5,n)* = microseconds into the current hour
- *timestamp(6,n)* = microseconds into the current minute
- *timestamp(7,n)* = microseconds into the current second

*Tablename.eventend(1,1)* is only valid for a data table using the DataEvent instruction, *Tablename.eventend(1,1)* = 1 if the last record of an event occurred the last time the table was called, = 0 if the data table did not store a record or if it is in the middle of an event.

*TableName.EventCount* = the number of data storage events that have occurred in an event driven DataTable.

*TableName.Tablefull* = 1 to indicate a fill and stop table is full or a ring-mode table has begun overwriting its oldest data, = 0 if the data table is not full/begun overwriting oldest data.

*TableName.TableSize* = the size allocation, in number of records, of the selected DataTable.

**NOTE:** The values of *Tablename.output(1,1)* and *Tablename.eventend(1,1)* are only updated when the tables are called.

## 6. View Pro

View Pro is a program that can be used to open data files (\*.DAT) or other CSI file types (\*.DLD, \*.CSI, \*.PTI, \*.FSL, \*.LOG, \*.CRX, etc.). It can also be used to view data from a LoggerNet database table.

Use the **File | Open** menu option to view a data file. (TOAC11, TOA5, TOB1, TOB2, TOB3, printable ASCII, comma separated ASCII, and array-based datalogger binary data files can be opened with View Pro.) Use the **File | View LoggerNet Database Table** menu option to view a database table. This opens a text view of your data. Data values from the text view can then be chosen to display graphically on a line graph, Histogram, XY Plot, Rainflow Histogram, or FFT as appropriate for the



data type. Multiple instances of each of these graphical displays can be opened. Both numeric data and graphs can be sent to a printer. Graphs can be saved to disk in a choice of formats.

**NOTE:** View Pro cannot display FFTs or Histograms from a TOAC11 data file.

Data values from array-based data files cannot be graphed unless an \*.FSL file has been associated with the data file or the [View | Array Definitions](#) (p. 94) dialog box has been used to generate timestamps for the data file.

ASCII Table Data, No Header files cannot be graphed in View Pro.

Use the **File | Open As Text** menu option to view other CSI file types.

View Pro options are accessed by using the menus or by selecting the toolbar icons. If you move and hold the mouse over a toolbar icon for a few seconds, a brief description of that icon's function will appear.

## 6.1 View Pro Toolbar

View Pro includes the following toolbar icons:



**Open.** Brings up a dialog box from which you can choose a data file to open.



**Copy.** Copies selected text to the clipboard. Text is selected by dragging the mouse pointer across the desired selection. Multiple columns in a data file can be selected by dragging the mouse pointer across the column headings.



**Cascade.** Rearranges all open, non-minimized data file windows so that the title bar of each window is visible. Windows cascade down and to the right starting from the upper left corner.



**Tile Vertically.** Rearranges all open, non-minimized data file windows as non-overlapping vertical tiles. This makes them all visible at the same time.



**Tile Horizontally.** Rearranges all open, non-minimized data file windows as non-overlapping horizontal tiles. This makes them all visible at the same time.



**Refresh Current File.** Refreshes an open data file. This is useful if you are viewing a file, and additional data has been stored since the file was first opened.



**Print Preview.** Displays how the currently selected data file will appear when it is printed.



**Print.** Brings up a dialog box that allows you to print the currently selected data file.



**New Line Graph.** Brings up a Line Graph window from which you can graph data values on the y-axis against their timestamps on the x-axis.



**New Histogram.** Brings up a Histogram window from which you can view Histogram data values.



**New XY Plot.** Brings up an XY Plot window from which you can plot data values on the y-axis against another specified data value on the x-axis.



**New Rainflow Histogram.** Brings up a Rainflow Histogram window from which you can view Rainflow Histogram data values.



**New FFT.** Brings up an FFT window from which you can view FFT data values.



**Keep Selected Graph on Top.** When data is being graphed and this option is selected, the currently selected graph will always be at the forefront of the View Pro program.

This is a toggle button. The button icon will have a green check mark through it when the option is currently selected.



**Reset Grid for New Selections.** If no graphs are open, this button will clear all selections in all open files. If one or more graphs are open, this button will change the Selected Graph to "None" and clear all selections in all open files for the "None" selection set.

Selected  
Graph

**Selected Graph.** Indicates which graph is currently selected. The drop-down list can be used to switch the currently [Selected Graph](#) (p. 87) to a different graph.

This control is disabled if no graphs are opened.



**Bring Selected Graph to Front.** Brings the currently selected graph to the forefront of the View Pro program.

This control is disabled if no graphs are opened.

## 6.1.1 Selected Graph

The **Selected Graph** field on the View Pro toolbar can be used to navigate between all opened graphs. When a graph is selected from the drop-down list, that graph will be brought to the front and the selections corresponding to that graph will be highlighted in the data file(s). As selections are made in a data file, they will be added to the currently-selected graph.

**NOTE:** If the Keep Data on Top option has been selected from the View menu, you will need to disable it before pressing the Bring the selected graph to the front button in order to bring the selected graph to the front.

You will only be able to make selections that correspond to the currently-selected graph. For example, if the currently-selected graph is a Histogram, you will only be able to select histogram records in a data file.

In order to begin a new graph, set the **Selected Graph** field to None. All highlighting in the data file(s) will be cleared, and you can begin making selections for a new graph. Once you have selected your data, press the button for the type of graph you wish to create.

With the **Selected Graph** field set to None, you can add selections to any graph by making the selections, right-clicking, choosing **Add Selections to Graph**, and then selecting the desired graph.

## 6.2 Right-Click Menu

The following options are available from the floating menu that is displayed when you right-click in a data file. (Note that for the following descriptions, the current selection in the data file is indicated by a dashed box around the selection, and the selected graph is indicated at the top right of the main window.)

**Define Selection** – This option allows you to modify the time span, record span, and color of the current selection. See [Selection Definition](#) (p. 89) for more information.

**Copy Selection** – This option copies selected text to the Windows clipboard

**Add Selections to Graph** – Adds the current selections to a graph. Click on or hover over this menu item to see a list of the opened graphs. Choose a graph to add the selections to. Note that this can be used to add selections made when is set to None to a graph. It can also be used to move selections from the currently-selected graph to a different graph.

**Clear All Selections** – Clears all selections in the data file.

**Clear Selection** – Clears the selection with focus. (This is the selection that has the dashed box around it. Left or right-click on a selection to give it the focus.)

**Format Column** – This option allows you to format a column in a binary (TOB) or CSIXML file. See [Format Column](#) (p. 90) for more information.

**Autosize Columns** – When a data file is opened, the columns are autosized to fit the data. Column sizes can be changed by dragging a column divider bar to the desired location. If column

sizes have been changed, choosing this option will return them to their default sizes. This function can also be accomplished from the View | Autosize Columns menu item.

**Add Bookmark** – The user can quickly navigate to a bookmarked record by using the Goto Bookmark option. Choosing the Add Bookmark option will add a bookmark to a record. When a record is bookmarked, a numbered circle (beginning at 0) will appear to the left of the record.

A bookmark can also be added to the top visible record by typing Ctrl-Shift-n, where n is the number of the bookmark.

Note that bookmarks are not persistent and will be gone once the data file or View Pro is closed.

**Goto Bookmark** – If the data file contains one or more bookmarks, hovering over the Goto Bookmark menu item will bring up a list of the current bookmarks. Selecting a bookmark from the list will automatically move the data file to that record.

The shortcut Ctrl-n, where n is the number of the bookmark, can also be used to move the data file to the desired bookmark.

**Delete Bookmark** – If the data file contains one or more bookmarks, hovering over the Delete Bookmark menu item will bring up a list of the current bookmarks. Selecting a bookmark from the list will delete the bookmark.

**Delete All Bookmarks** – Deletes all bookmarks from the data file.

**View Record** – Brings up the current record in a Record View window showing each column heading and the data value. (The Record View window can be brought up directly by clicking to the left of the record.)

## 6.2.1 Selection Definition

Define the records to be included in the selection by initial and ending time, initial time and time span, initial and ending record, or initial record and record span. If any of the date/time or record number options are changed, the other options will automatically adjust to reflect the change.

Note that changes to the records included will not be reflected in the data panel, until the Apply button is pressed.

**NOTE:** The Time Span control will not allow a time span less than the interval existing between the initial record and the next record. For example, for a data file with a 5-minute data interval, the Time Span cannot be less than 5 minutes.

If the Ending Record is less than or equal to the Initial Record when Apply is pressed, the Ending Record will automatically be set to one greater than the Initial Record.

Set the color to be used for the selection by pressing the ... button and choosing a color.

## 6.2.2 Format Column

**NOTE:** Format Column is only enabled for binary (TOB) and CSIXML files. With a binary or CSIXML file open, Format Column is accessed by right-clicking on a data column.

### Date/Time Format Options

Choose the format in which you would like the date and time to be formatted. You can, alternatively, use the Format Code field to specify how the date and time are to be formatted. When one of the predefined format options is selected, the format code will automatically change to represent the selected format.

Note that this field is only enabled, when the column being formatted contains a time.

### Format Code

Indicate how you would like the date and time to be formatted, according to the following codes:

c	Displays the date using the format given by the ShortDateFormat global variable, followed by the time using the format given by the LongTimeFormat global variable. The time is not displayed if the fractional part of the DateTime value is zero.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat).
dddd	Displays the day as a full name (Sunday-Saturday).
ddddd	Displays the date using the short date format.
dddddd	Displays the date using the long date format.
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec).
mmmm	Displays the month as a full name (January-December).
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).

h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (0-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
z	Displays the millisecond without a leading zero (0-999).
zzz	Displays the millisecond with a leading zero (0-999).
am/pm	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
/	Displays the date separator character.
:	Displays the time separator character.
'xx' or "xx"	Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

Format specifiers may be written in upper case as well as in lower case letters--both produce the same result.

Note that the **Format Code** field is only enabled, when the column being formatted contains a time.

### Decimal Places

Specify the number of digits that will be shown after a decimal place.

### Leading Zeroes

Specify the minimum number of digits that will be shown before a decimal place. Leading zeroes will be added, if necessary.

### Scientific Notation

Select the check box to display the data values in scientific notation.


## 6.3 Opening Files

6.3.1 Data Files .....	92
6.3.2 Other Files .....	96

### 6.3.1 Data Files

6.3.1.1 File   Open .....	92
---------------------------	----

#### 6.3.1.1 File | Open

To open a data file, press the  icon or select **File | Open** from the menu. TOAC11, TOA5, TOB1, TOB2, TOB3, CSIXML, printable ASCII, comma separated ASCII, and array-based datalogger binary data files can be viewed.

Files with a particular extension can be configured in Windows to be opened by View Pro automatically when double-clicked in Windows Explorer. Refer to Assigning Data Files to View Pro|tag=AssignFiles for more information.

**NOTE:** If you are viewing a CSIXML file and it was collected using the **Midnight is 2400** option, midnight will not be displayed as 2400 in View Pro.

#### Array-based Data Files

When opening a data file from an array-based datalogger, you will be given the option of loading an FSL (Final Storage Label) file. The FSL file will be used to provide column headings. (The \*.FSL file is created when a datalogger program is compiled in Edlog or ShortCut.)

If a data file is opened that contains multiple arrays, the entire data file will be opened in one window. In addition, each array will be opened in a separate window. The window containing the entire data file is for viewing only. Data must be graphed from the individual array windows.

**NOTE:** Individual array windows will not have an array ID column. The array ID will be identified in the window's title bar.

Array-based data files do not contain timestamps. If an FSL file is associated with the data file, View Pro will try to extract timestamps from the appropriate columns. You can select [View | Array Definitions](#) (p. 94) to specify how the timestamps are created. Note that if no timestamps are used, data cannot be graphed.

## Scrolling Data Files

A data file can be scrolled using the scroll bars at the top and right of the window. A data file can also be scrolled vertically by clicking and holding with the mouse just above the bottom scroll bar. It can be scrolled horizontally by clicking and holding with the mouse just above the right scroll bar.

## Refreshing Data Files

An opened data file can be refreshed by pressing the  icon on the toolbar. This is useful if you are viewing a file, and additional data has been stored since the file was first opened.

## Low Memory Errors

Depending upon your computer's hardware specifications and operating system, you may experience low memory errors when trying to open large data files with View Pro. It may be necessary to close other applications in such instances.

View Pro can be run as a stand-alone program by double-clicking the View\_Pro.exe file in Windows Explorer. This will allow you to open larger data files without memory errors.

## View | Autosize Columns

When a data file is opened, the columns are autosized to fit the data. Column sizes can be changed by dragging a column divider bar to the desired location. If column sizes have been changed, choosing this menu item will return them to their default sizes.

## View | Show Full Header

Shows full header information for each column in the data files, including units and field names, if available in the \*.DAT file.

This is a toggle menu item. There will be a check mark next to the item, when it is active. Deactivate it by selecting it again.

## View | Row Shading

Shades every other row of the data file.

This is a toggle menu item. There will be a check mark next to the item, when it is active. Deactivate it by selecting it again.

## View | Lock Timestamp Column

Locks the timestamp column on the left of the data file so that it remains visible as you scroll through the columns in the data file.



This is a toggle menu item. There will be a check mark next to the item, when it is active. Deactivate it by selecting it again.

## **View | View File Information**

Displays information on the currently selected data file including file name and file format, and information about that datalogger and program that generated the file such as station name, table name, datalogger model, datalogger OS version, program name, and program signature.

## **View | Start Graphs Maximized**

When this option is selected, all new graphs will be started in a maximized state.

This is a toggle menu item. There will be a check mark next to the item, when it is active. Deactivate it by selecting it again.

## **View | Array Definitions**

Specifies the FSL file to be used and how the timestamps will be created for an array from an array-based datalogger data file.

**NOTE:** If a data file is opened that contains multiple arrays, each array will be opened in a separate window. The array definitions for each array are set individually.

Data values from array-based data files cannot be graphed unless an \*.FSL file has been associated with the data file or the Array Definitions dialog box has been used to generate timestamps for the data file.

### **FSL File**

Select the FSL (Final Storage Label) file that will be used to provide column headings. The \*.FSL file is created when a datalogger program is compiled in Edlog or ShortCut.

### **No Timestamp**

No timestamps will be used. Note that arrays without timestamps cannot be graphed.

### **Extract Timestamp using RealTime fields**

The timestamps will be extracted from the real time fields of the array. (Real time fields are placed into the array by using the P77 instruction in the Edlog program.) Select a field from the drop-down list for **Year**, **Day**, **Time**, and **Seconds**. If there is no array field to match a value, you may choose *Missing* or *Use Fixed Value*. If *Missing* is chosen, the value (year, day, time, or seconds) will

not be part of the timestamps. If *Use Fixed Value* is chosen, you must specify a **Fixed Value** that will be used for the year, day, time, or seconds for all timestamps.

### Generate Timestamps


The timestamps will be generated based on the date and time of the first record and the interval between records. Enter the date and time of the first array record in **Initial Date & Time**. Then enter the interval between records in **Data Interval**.

## View | Keep Data on Top

When data is being graphed and this option is selected, the Data Display will always be at the forefront of the View Pro program. This is a toggle item. There will be a check mark to the left of the menu item when it is turned on. Note that if this menu item is enabled, the **Keep Graph on Top** item will automatically be disabled.

## View | Keep Graph on Top

When data is being graphed and this option is selected, the currently selected graph will always be at the forefront of the View Pro program. This is a toggle item. There will be a check mark to the left of the menu item when it is turned on. Note that if this menu item is enabled, the **Keep Data on Top** item will automatically be disabled.

This option is also available from a toolbar button, .

## View | Background Color

This option allows you to change the background color for the currently selected data file.

## View | Font


This option allows you to change the font used on the Data Panel display when viewing data. The selected font is also used when a file is printed.

The following characteristics can be set:


- Font Type (Courier, Courier New, Fixedsys, Lucida Console, MS LineDraw, Terminal, etc.)
- Font Style (regular, italic, bold, bold italic)
- Size
- Effects (strikeout, underline)
- Color

- Script (Western, Hebrew, Arabic, Greek, Turkish, Baltic, Central European, Cyrillic, Thai, Vietnamese)


## Window | Cascade

This menu option rearranges all open, non-minimized data file windows so that the title bar of each window is visible. Windows cascade down and to the right starting from the upper left corner. This function can also be accomplished by pressing the  button on the main toolbar.

## Window | Tile Horizontally

This menu option rearranges all open, non-minimized data file windows as non-overlapping horizontal tiles. This makes them all visible at the same time. This function can also be accomplished by pressing the  button on the main toolbar.

## Window | Tile Vertically

This menu option rearranges all open, non-minimized data file windows as non-overlapping vertical tiles. This makes them all visible at the same time. This function can also be accomplished by pressing the  button on the main toolbar.

## 6.3.2 Other Files

6.3.2.1 File   Open as Text .....	96
6.3.2.2 File   Open as Hex .....	96
6.3.2.3 Import CSV .....	96

### 6.3.2.1 File | Open as Text

To open a file that is not a data file (e.g., \*.DLD, \*.CSI, \*.PTI, \*.FSL, \*.LOG, \*.CRX) or to open a data file without time-stamp processing, select **File | Open As Text** from the menu. Files opened with this option cannot be graphed.

### 6.3.2.2 File | Open as Hex

This option is used to open the file in a hexadecimal format. This may be useful when viewing binary files.

### 6.3.2.3 Import CSV

The Import CSV dialog box is opened from the File | Import CSV menu item. It is used to import a CSV (Comma Separated Value) file into View Pro.

When you select the File | Import CSV menu item, a browser will be displayed allowing you to browse to the CSV file to be imported. By default, only files with a .csv extension will be shown. If your file does not have a .csv extension, you will need to use the drop-down list box to select .txt files or all files.

The data preview grid, shown below, is used to specify how many lines are in the header, the file delimiter, the date/time format, and the header line to be used for column names.

Note that only the first 14 rows of the file, including header lines, can be seen in the data preview grid. The scroll bars can be used to scroll through these first 14 rows and all columns of data in the file.

**Import CSV**

File... C:\Data\USUClimateData.csv

**File Details**

Header Line Count: 2

File Delimiter: Comma

**Date/Time Format Options**

☒ Has Date/Time Column

Derived Date/Time Format: yyyy-mm-dd

**Names**

Date	Id	Name	State	Latitude	Longitude	Elevation	Maximum
				Degrees	Degrees	Feet	Degrees
2010-01-01	427271	RICHMOND	UT	41.9064	-111.81	4680	32.0
2010-01-02	427271	RICHMOND	UT	41.9064	-111.81	4680	37.0
2010-01-03	427271	RICHMOND	UT	41.9064	-111.81	4680	38.0
2010-01-04	427271	RICHMOND	UT	41.9064	-111.81	4680	27.0
2010-01-05	427271	RICHMOND	UT	41.9064	-111.81	4680	31.0
2010-01-06	427271	RICHMOND	UT	41.9064	-111.81	4680	36.0

Import Cancel Help

## File

The name of the file to be imported. Press the **File** button to bring up a browser to select the desired file.

## Header Line Count

CSV files may have multiple header lines or no header line. Use the **Header Line Count** field to designate how many header lines your file contains before the data begins.

## File Delimiter

Although CSV files are by definition comma delimited, other file delimiters (tab or space) can be selected in the **File Delimiter** drop-down list box.

### **Date/Time Format Options**

When importing a CSV file, View Pro will attempt to derive a timestamp from data in the first column, if common timestamp delimiters exist in the data. If View Pro is able to derive the timestamp, the **Has Date/Time Column** check box will be checked and the derived timestamp format will be displayed in the **Derived Date/Time Format** field using the codes shown below. Any errors in the derived format can be corrected in this field.

If the first column contains a timestamp, but does not have the common timestamp delimiters that allow View Pro to determine that it is a timestamp, you can manually check the **Has Date/Time Column** check box and then input the appropriate codes in the **Derived Date/Time Format** field to designate the format of the timestamp.

If the first column of data does not contain a timestamp, leave the **Has Date/Time Column** check box unchecked. You will still be able to import the file into View Pro and view the data. However, you will not be able to graph the data.

#### *Date and Time Format Codes*

yy = Year last 2 digits

yyyy = Year as 4 digits

m = Month number no-leading 0

mm = Month number as 2 digits

mmm = Month using short form (Jan)

mmmm = Month using long form (January)

d = Day number no-leading 0

dd = Day number as 2 digits

ddd = Day using short day names (Sun)

dddd = Day using long day names (Sunday)

h = Hour with no leading 0's

hh = Hour as 2 digits

n = Minute with no leading 0's

nn = Minute as 2 digits

s = Seconds with no leading 0's

ss = Seconds as 2 digits

z = Milli-seconds with no leading 0's

zzz = Milli-seconds as 3 digits

### **Names**

The header line that contains column names is designated by selecting an option button on the left of the preview grid under **Names**.

### Import

After all of the settings have been specified, press the Import button to import the CSV file into View Pro.

## 6.4 Viewing Database Tables

6.4.1 File   View LoggerNet Database Table .....	99
6.4.2 Selecting a Database .....	99
6.4.3 Selecting a Table .....	104
6.4.4 Installing PostgreSQL ODBC Drivers .....	105
6.4.5 Installing Oracle ODBC Drivers .....	114

### 6.4.1 File | View LoggerNet Database Table

This option allows you to view data from a LoggerNet database table. The first time the option is selected you will need to select a database (see [Selecting a Database](#) (p. 99)) and then select a database table (see [Selecting a Table](#) (p. 104)).

### 6.4.2 Selecting a Database

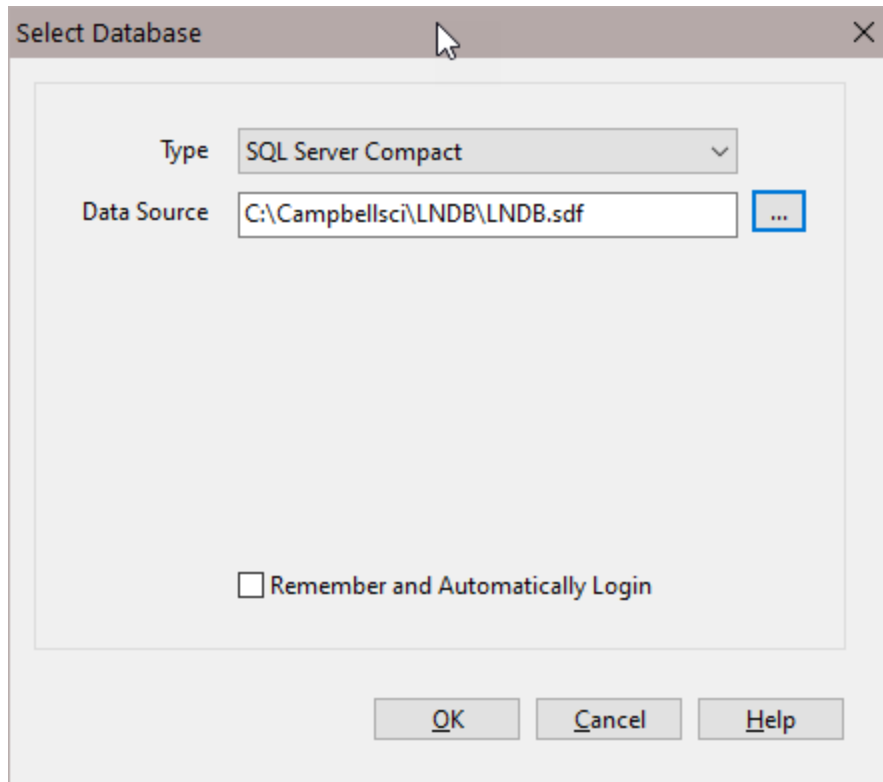
The Select Database dialog box comes up the first time you select **File | View LoggerNet Database Table**. (After a database has been selected, this menu item will bring up the [Selecting a Table](#) (p. 104) dialog box. To view a table from a different database, you will need to press the Change Database button on the Select Table dialog box.)

View Pro supports SQL Server Compact, SQL Server, MySQL, PostgreSQL (ODBC), and Oracle (ODBC) databases.

**NOTE:** To use a PostgreSQL or Oracle database, ODBC drivers are required. See [Installing PostgreSQL ODBC Drivers](#) (p. 105) or [Installing Oracle ODBC Drivers](#) (p. 114).

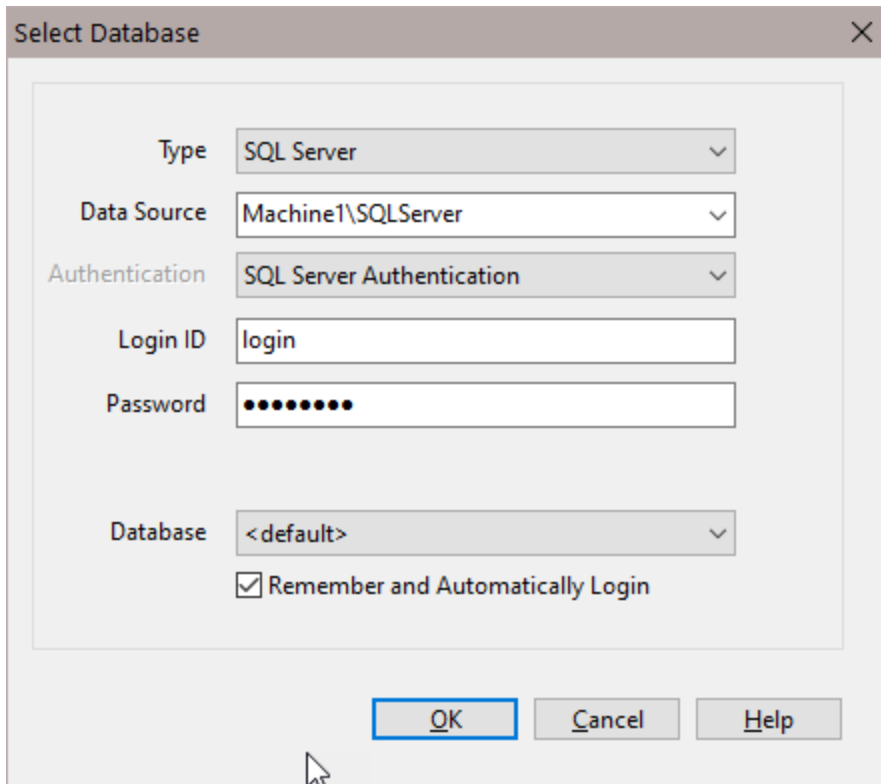
The information to enter changes depending on the database type as described below:

## SQL Server Compact



SQL Server Compact is an embedded database that just requires the selection of a filename. Press the Browse button to the right of the Data Source field to browse to the desired database.

## SQL Server



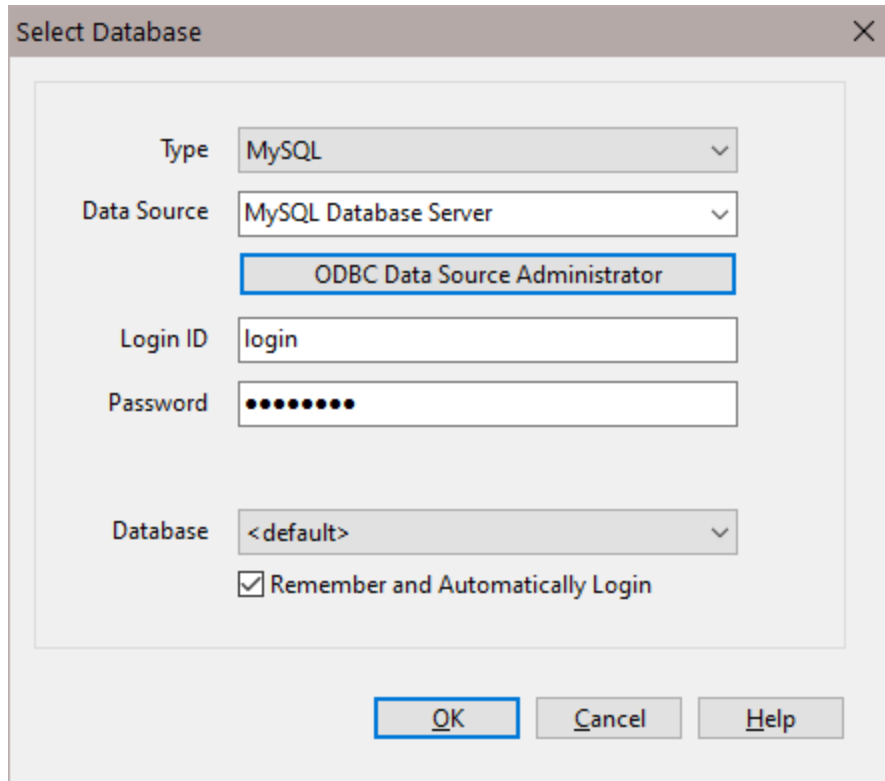
The screenshot shows a 'Select Database' dialog box with the following fields and options:

- Type:** A dropdown menu set to 'SQL Server'.
- Data Source:** A dropdown menu set to 'Machine1\SQLServer'.
- Authentication:** A dropdown menu set to 'SQL Server Authentication'.
- Login ID:** A text box containing 'login'.
- Password:** A text box filled with ten dots.
- Database:** A dropdown menu set to '<default>'.
- Remember and Automatically Login:** A checked checkbox.
- Buttons:** 'OK', 'Cancel', and 'Help' buttons at the bottom.

To configure a connection to SQL Server you must select a SQL Server instance. The list of published SQL Server instances is shown in the Data Source combo box. You can also type into the Data Source combo box, because the desired server might not be published. Windows Authentication or SQL Server Authentication can be selected. Windows Authentication does not require a username and password, but rather uses Windows user accounts to authenticate valid users. SQL Server Authentication requires a login ID and Password and is independent of Windows user accounts. You can select the <default> database or select a specific database from the Database combo box.



## MySQL



The screenshot shows a 'Select Database' dialog box with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Type:** A dropdown menu with 'MySQL' selected.
- Data Source:** A dropdown menu with 'MySQL Database Server' selected. Below it is a button labeled 'ODBC Data Source Administrator'.
- Login ID:** A text input field containing 'login'.
- Password:** A text input field with masked characters (dots).
- Database:** A dropdown menu with '<default>' selected.
- ☒ Remember and Automatically Login
- At the bottom are three buttons: 'OK', 'Cancel', and 'Help'.

The MySQL connection is an ODBC connection. You must use the Windows ODBC Data Source Administrator to configure the database connection. Currently only system data sources are supported and show in the Data Source combo box. The Login ID and Password may be optional. They will be set to blank in the connection string. It has been found that when set to blank, the login id and password configured in the system data source are used. You can select the <default> database (default as configured in the data source) or select a different database.

### PostgreSQL (ODBC)

**NOTE:** To use a PostgreSQL database, ODBC drivers are required. See [Installing PostgreSQL ODBC Drivers](#) (p. 105).

Select Database

Type: PostgreSQL (ODBC)

Data Source: PostgreSQL35W

ODBC Data Source Administrator

Login ID: login

Password: .....

Database: mypgdatabase

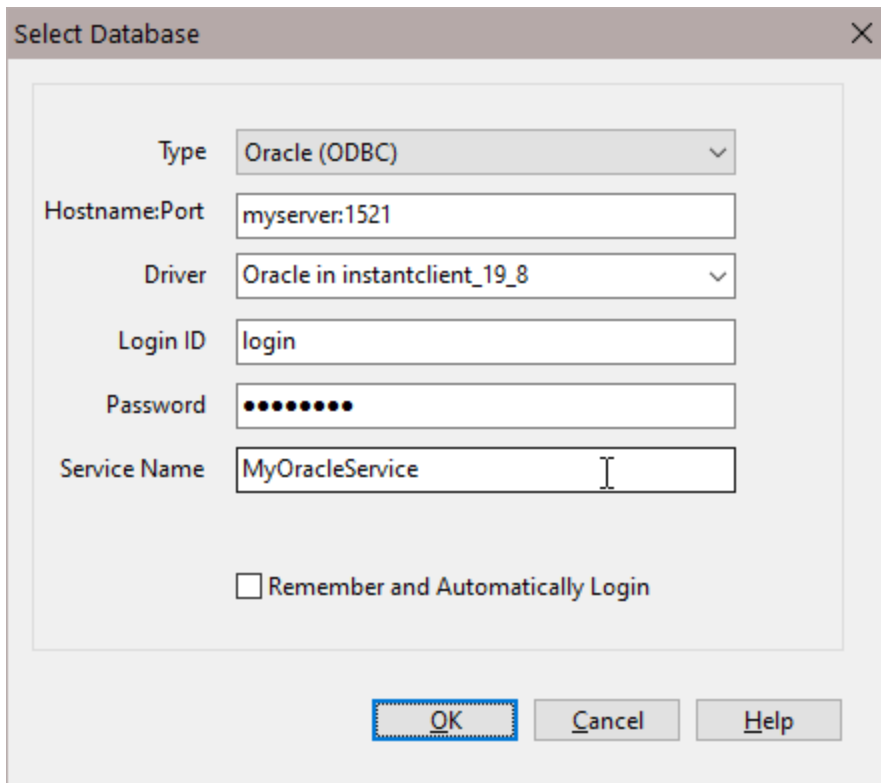
☐ Remember and Automatically Login

OK Cancel Help

In **Data Source**, select the DSN (name) which was created in the ODBC administrator when the ODBC drivers were installed. Provide the **Login ID**, **Password**, and **Database** name for making the connection. The database administrator for the LNDB PostgreSQL database should be able to provide this information.

### Oracle (ODBC)

**NOTE:** To use an Oracle database, ODBC drivers are required. See [Installing Oracle ODBC Drivers](#) (p. 114).



The screenshot shows a 'Select Database' dialog box. It contains the following fields and controls:

- Type:** A dropdown menu set to 'Oracle (ODBC)'.
- Hostname:Port:** A text field containing 'myserver:1521'.
- Driver:** A dropdown menu set to 'Oracle in instantclient\_19\_8'.
- Login ID:** A text field containing 'login'.
- Password:** A text field with masked characters (dots).
- Service Name:** A text field containing 'MyOracleService'.
- Remember and Automatically Login:** An unchecked checkbox.
- Buttons:** 'OK', 'Cancel', and 'Help' buttons at the bottom.

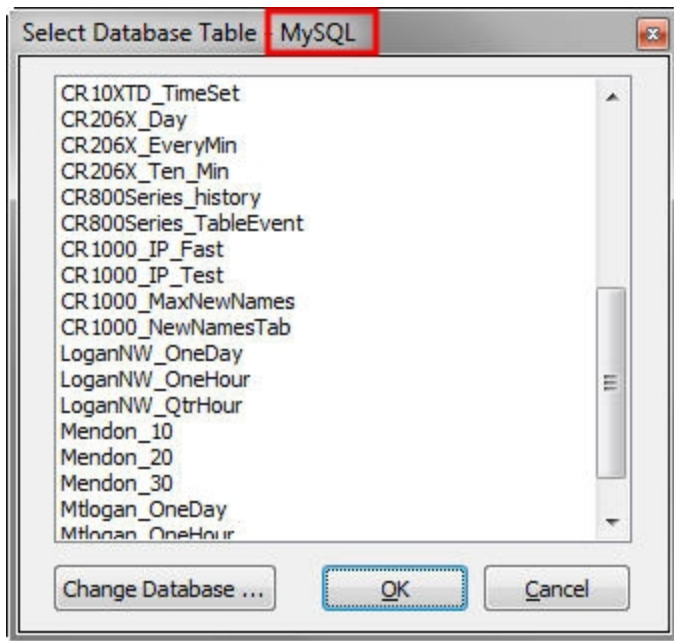
In **Hostname:Port**, enter the IP address or DNS name of the host (server) and port if not the default. (The default Oracle port is 1521.) Choose the driver from the **Driver** dropdown list ("Oracle in instantclient\_19\_8" in the driver installation example). Enter the **Login ID**, **Password**, and **Service Name**. The host:port and service name can be obtained from the database administrator for the Oracle LNDB database.

### Remember and Automatically Login

If you select the **Remember and Automatically Login** check box, the Login ID and Password will be remembered and the next time the application starts an attempt will be made to login without showing the dialog again.

## 6.4.3 Selecting a Table

The Select Database Table dialog box comes up once a database has been selected. The currently active database is indicated in the title bar as highlighted in the following screenshot. (A database is selected from the Select Database dialog box which comes up the first time you select File | View LoggerNet Database Table. Once a database has been selected, the Select Tables dialog box will come up immediately when the File | View LoggerNet Database Table menu item is selected. To view a table from a different database, you will need to press the Change Database button on the Select Table dialog box.)



Select the database table that you wish to view and then press the OK button.

## 6.4.4 Installing PostgreSQL ODBC Drivers

To use an LNDB database that is stored in a PostgreSQL server with View Pro, you will need to install the official PostgreSQL ODBC driver on your computer.

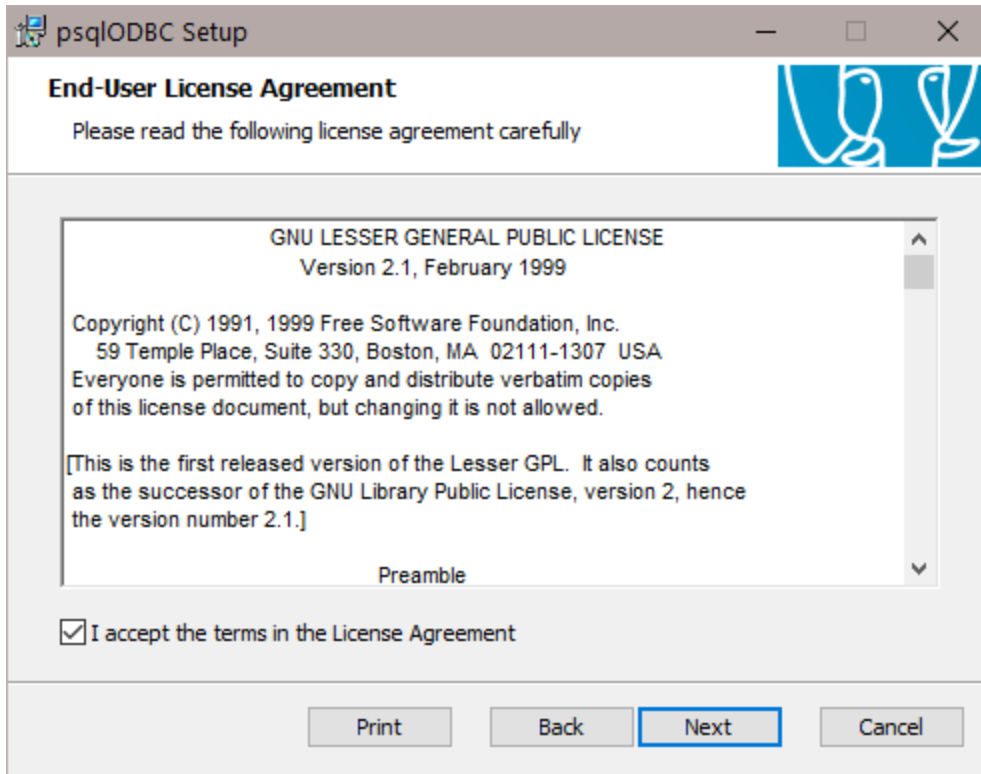
**NOTE:** View Pro is different from the LNDB Manager (and LNDB in general), which can connect to the PostgreSQL database without an ODBC driver.

The official ODBC driver is called psqLODBC. The home page for the driver is <https://odbc.postgresql.org>. It is very important to install the 32-bit version of the ODBC driver. (It shouldn't cause any trouble if the 64-bit driver is installed in addition to the 32-bit driver, but only the 32-bit version will work with View Pro.)

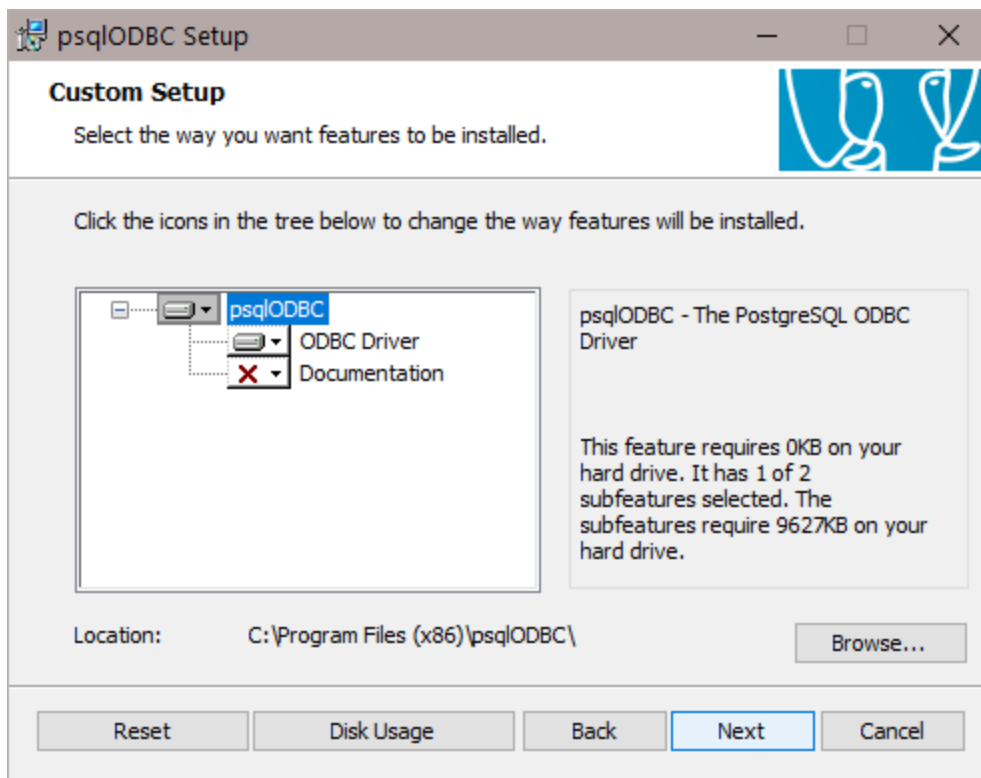
From the home page:

1. Click the **PostgreSQL downloads site** link.
2. Click the **msi** folder.
3. Scroll down to the bottom to see the newest driver installers.
4. Find the newest -x86 driver, for example, psqldbc\_12\_02\_0000-x86.zip. The **x86** designation on the end of the filename ensures that it is 32-bit. (The .zip file without x86 or x64 at the end of the filename installs both the 32-bit and 64-bit driver and is about twice the size.)

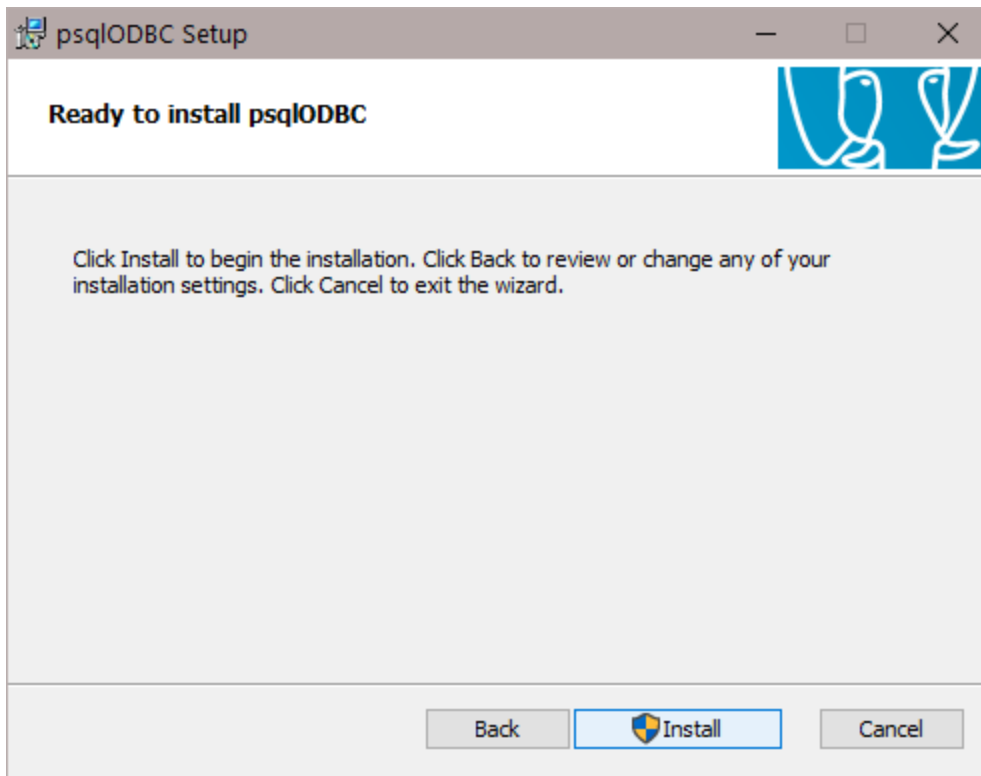
5. Click the appropriate .zip file to download the file.
6. Right-click the .zip file to extract the .msi file from it.
7. Double-click the .msi file to run the driver installation process.
8. Click **Next** to begin the installation.
9. Accept the license agreement and click **Next**.



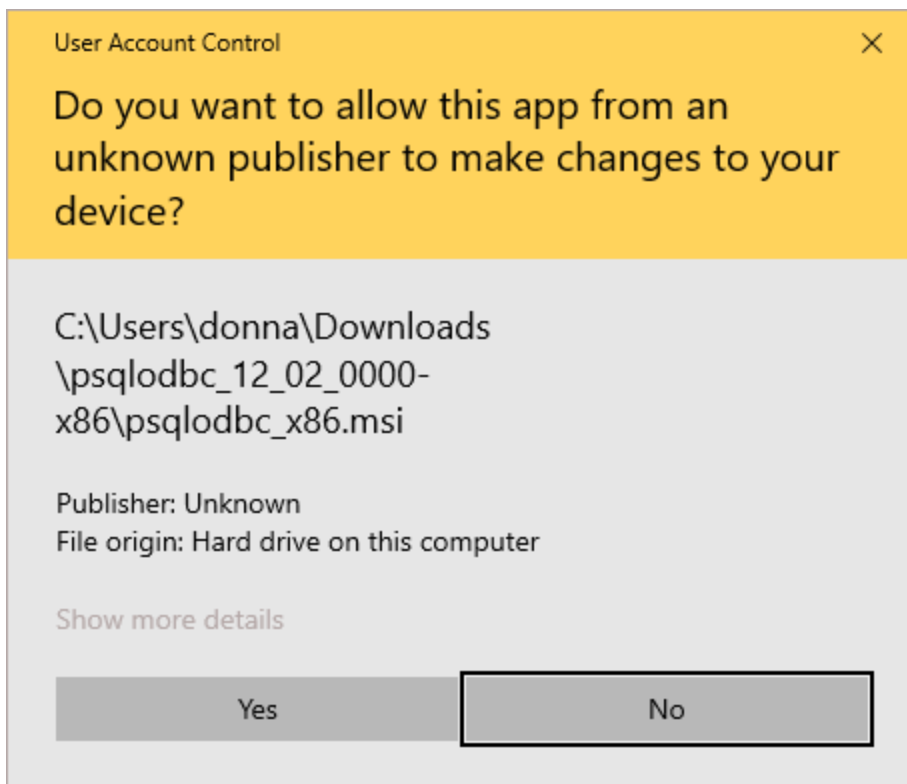
10. Click **Next** to accept the default setup.



11. Click **Install**.

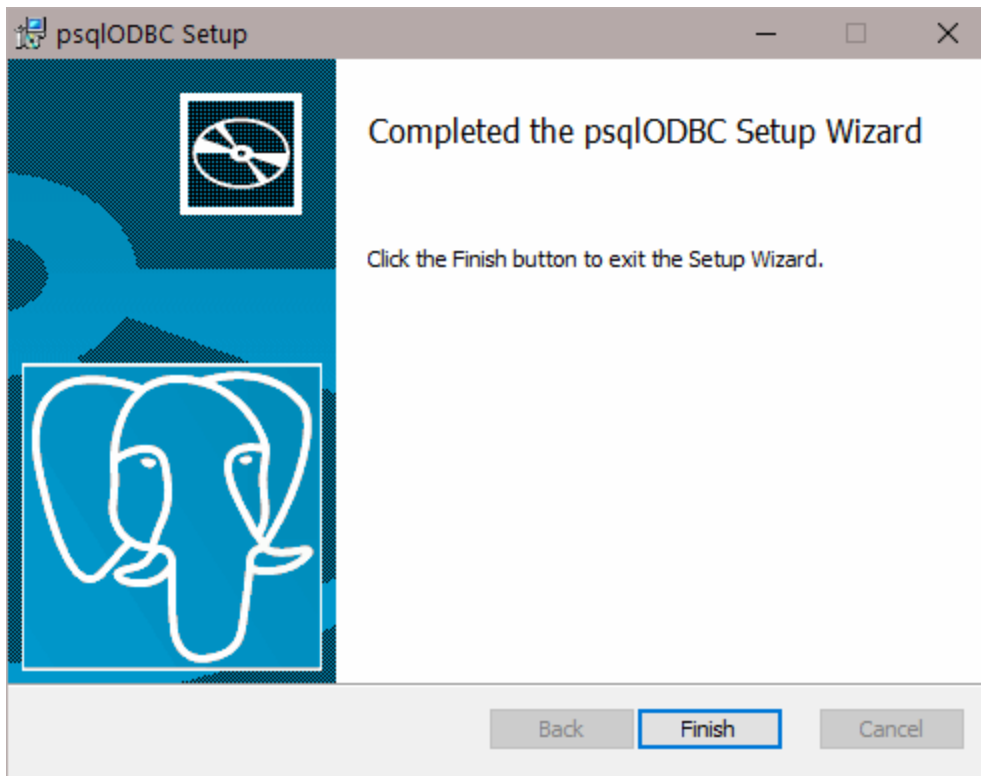


12. You may see a Windows warning similar to this. Click **Yes** to continue.

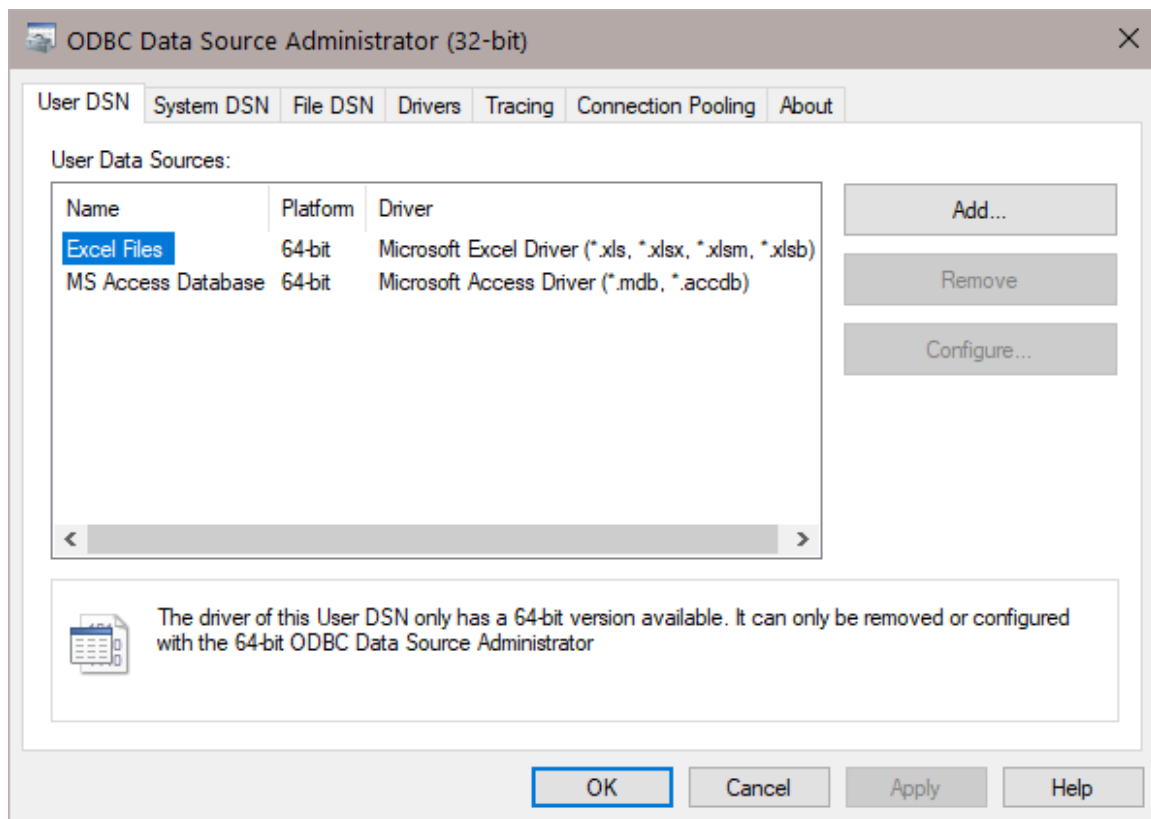




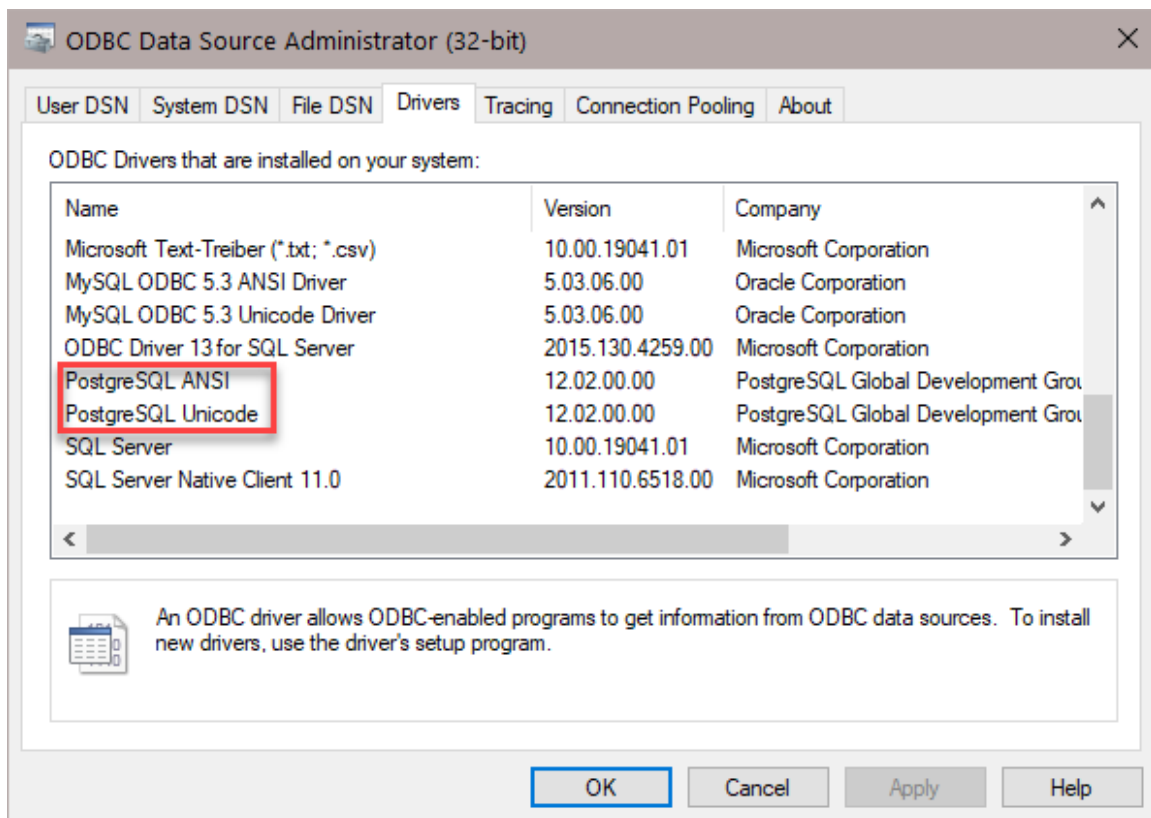
13. Click **Finish** to close the installation wizard.



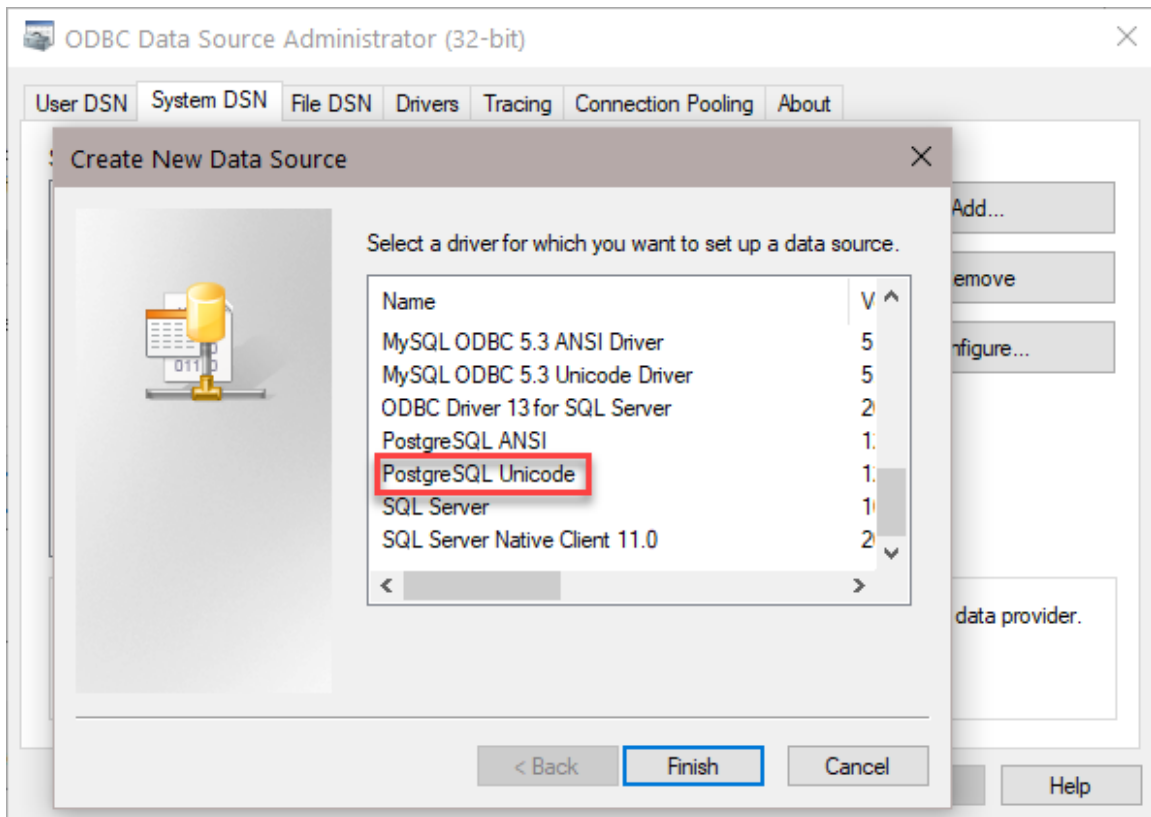
14. Once the installation is complete, run the 32-bit (Windows/Microsoft) ODBC Data Source Administrator. (Type ODBC in the Windows search bar. From the choices, make sure to select the 32-bit version.)



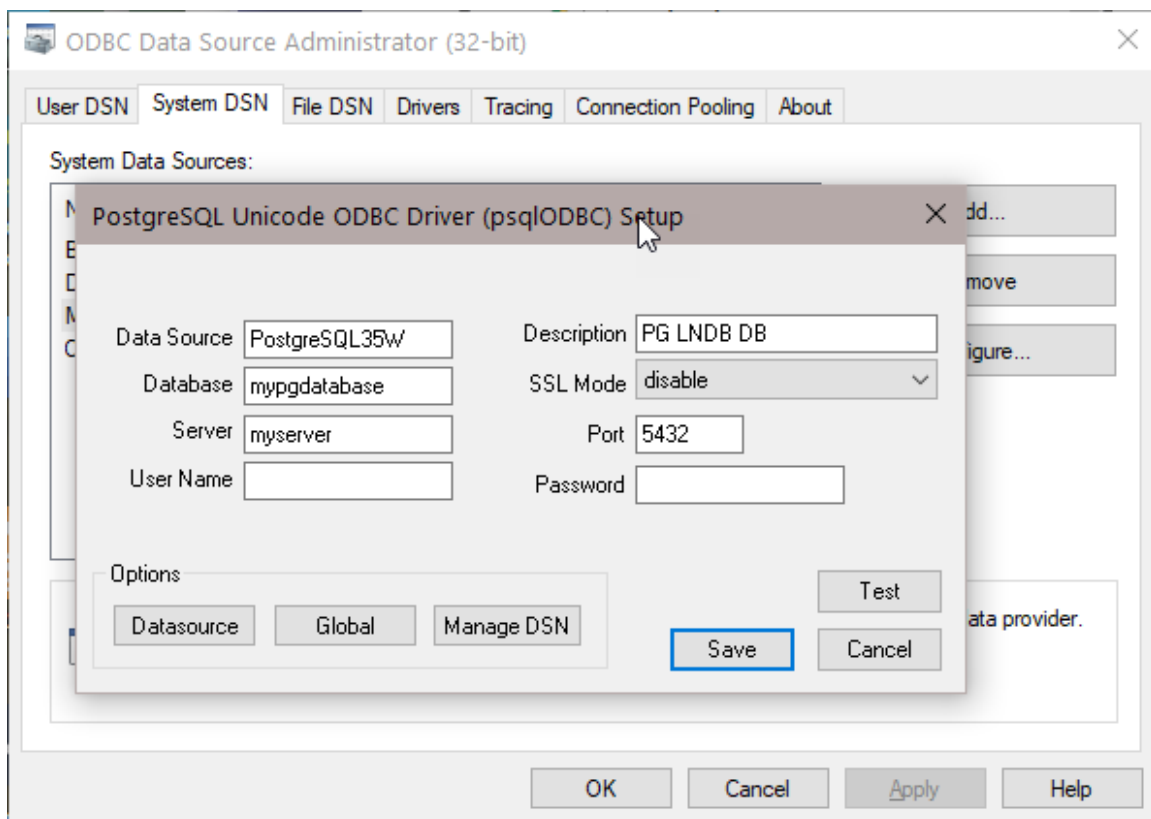
15. Click the **Drivers** tab and verify the drivers highlighted here:



16. To set up a DSN (Data Source Name), click the **System DSN** tab. Click **Add**. Select **PostgreSQL Unicode** and click **Finish**.



17. The following screen is displayed for configuring the Data Source. In **Data Source**, type the name (user's choice) by which this connection will be known on the system. (You will select this name from a drop-down list later in View Pro.) In **Database**, type the name of the database. Type the **Server** and **Port** where the PostgreSQL database service is running. (The default PostgreSQL port is 5432.) The **User Name** and **Password** can be provided later in View Pro. However, they can be temporarily added here in order to use the **Test** button to ensure the connection to the database is working. Your PostgreSQL database administrator can provide you with the Database, Server, Port, User Name, and Password, if you don't know these.



Your PostgreSQL ODB data source set up is now complete. See [Selecting a Database](#) (p. 99) and [Selecting a Table](#) (p. 104) for how to use the data source in View Pro.

## 6.4.5 Installing Oracle ODBC Drivers

To use an LNDB database that is stored in an Oracle server with View Pro, you will need to install an official Oracle ODBC driver on your computer.

Oracle provides multiple official ODBC drivers for use with Oracle databases. View Pro can work with any of these drivers, but it must be a 32-bit version. Also, the driver must have a version number at least as large as the version of the Oracle database server being used (12c, 18c, 19c

etc.) Consult with the database administrator of your Oracle database to determine if an ODBC driver is already installed on your computer, whether the administrator can install it for you, or whether you should do it yourself. Unlike ODBC drivers provided from other vendors, Oracle ODBC drivers include the version number in the name, as well as an indicator of what kind of client installed the driver. This means that multiple Oracle ODBC drivers can show up on a system, each with a unique and separate name. If multiple drivers are installed on your machine, determine which driver to use by consulting with your database administrator.

Two methods for getting a new 32-bit ODBC driver installed include using the Oracle Universal Installer (OUI) or using the Oracle InstantClient. Instructions for the InstantClient install are provided here. Once the driver is installed, you will specify that driver directly inside of View Pro when setting up the Oracle connection. It is not necessary to set up a Data Source (DSN) configuration when using the Oracle ODBC driver. A DSN could be used for troubleshooting purposes, if needed.

The InstantClient ODBC driver has a smaller file-size footprint on the machine where it is installed. You must obtain two .zip files to enable the installation.

1. Go to the Oracle InstantClient download area, <https://www.oracle.com/database/technologies/instant-client/microsoft-windows-32-downloads.html>, to obtain the two installers: Basic Light Package (Base) and the ODBC Package (Development and Runtime - optional package).
2. Select the newest versions of the files at the top of the window. For example, in the following screenshot, the names of the two newest .zip files are: instantclient-basiclite-nt-19.8.0.0.0dbru.zip and instantclient-odbc-nt-19.8.0.0.0dbru.zip. You may have to create a

no-cost Oracle account to download these files.

Instant Client for Windows 32-bit

oracle.com/database/technologies/instant-client/microsoft-windows-32-downloads.html

ProductsResourcesSupportEventsView Accounts

Database / Technologies / Instant Client / Instant Client for Windows 32-bit

## Instant Client Downloads for Microsoft Windows 32-bit

See the [Instant Client Home Page](#) for more information about Instant Client.

The installation instructions are at the foot of the page.

Client-server version interoperability is detailed in Doc ID 207303.1. For example, Oracle Call Interface 18.3 can connect to Oracle Database 11.2 or later. Some tools may have other restrictions.

Permanent links to the latest packages are: [Basic](#), [Basic Light](#), [SQL\\*Plus](#), [Tools](#), [SDK](#), [JDBC Supplement](#), [ODBC](#).

### Version 19.8.0.0.0

Base - one of these packages is required

Name	Download	Description
Basic Package	<a href="#">instantclient-basic-nt-19.8.0.0.0dbru.zip</a>	All files required to run OCI, OCCI, and JDBC-OCI applications (77,279,917 bytes) (cksum - 1413024006)  The 19.8 Basic package requires the Microsoft Visual Studio 2017 Redistributable.
Basic Light Package	<a href="#">instantclient-basclite-nt-19.8.0.0.0dbru.zip</a>	Smaller version of the Basic package, with only English error messages and Unicode, ASCII, and Western European character set support (37,025,587 bytes) (cksum - 2147043239)  The 19.8 Basic package requires the Microsoft Visual Studio 2017 Redistributable.

### Tools - optional packages

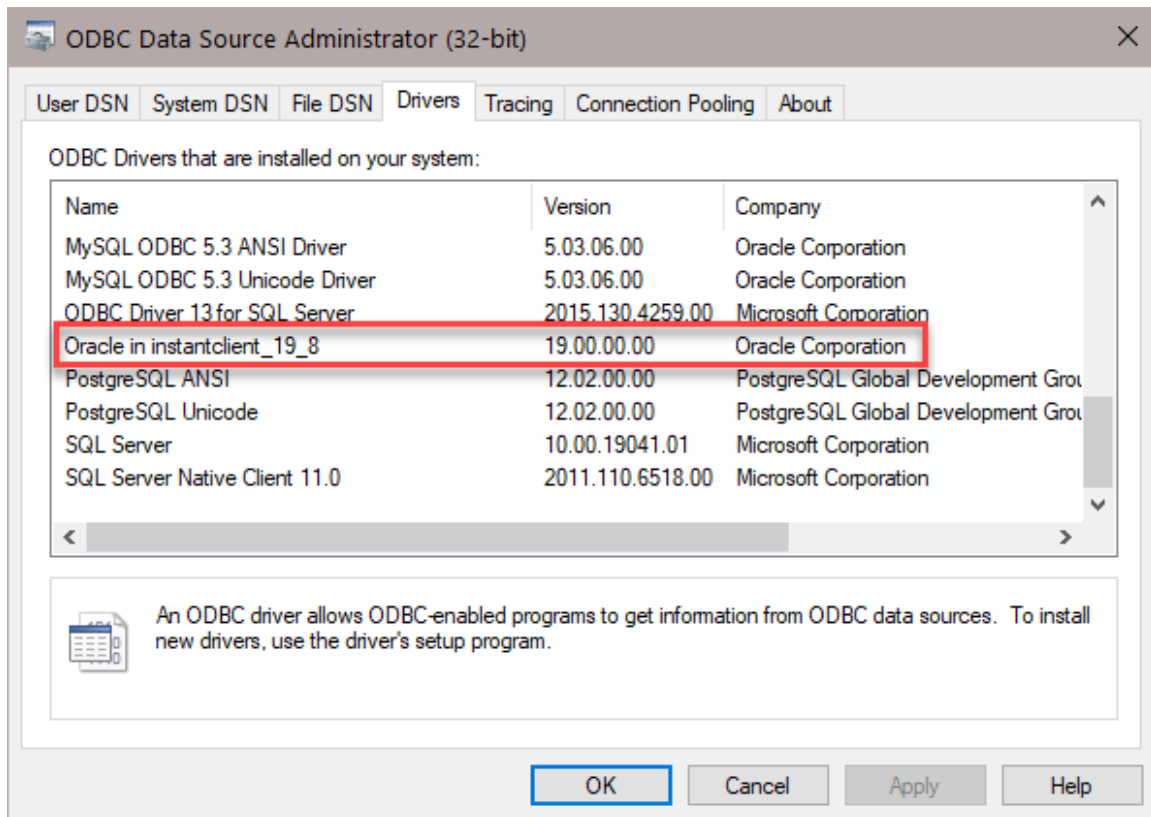
Name	Download	Description
SQL*Plus Package	<a href="#">instantclient-sqlplus-nt-19.8.0.0.0dbru.zip</a>	The SQL*Plus command line tool for SQL and PL/SQL queries (886,517 bytes) (cksum - 1212080256)
Tools Package	<a href="#">instantclient-tools-nt-19.8.0.0.0dbru.zip</a>	Includes Data Pump, SQL*Loader and Workload Replay Client (1,094,122 bytes) (cksum - 2335115878)

### Development and Runtime - optional packages

Name	Download	Description
SDK Package	<a href="#">instantclient-sdk-nt-19.8.0.0.0dbru.zip</a>	Additional header files and an example makefile for developing Oracle applications with Instant Client (1,503,154 bytes) (cksum - 2759323101)
JDBC Supplement Package	<a href="#">instantclient-jdbc-nt-19.8.0.0.0dbru.zip</a>	Additional support for Internationalization under JDBC (1,583,355 bytes) (cksum - 301921892)
ODBC Package	<a href="#">instantclient-odbc-nt-19.8.0.0.0dbru.zip</a>	Additional libraries for enabling ODBC applications (788,746 bytes) (cksum - 3276888572)
Precompiler Package	Precompiler Downloads	Additional files for Pro*C and Pro*COBOL

PC400 116

3. Extract the contents of both .zip files into the same folder (as contained in the zip file). For example, in the preceding screenshot, the folder is "instantclient\_19\_8". This folder is the Oracle home folder, which can be located anywhere. It is sometimes placed in "C:\\" (root) or "C:\Oracle". If the folder already exists previous to this installation, use a different folder name for the contents, such as "instantclient\_CSI". The name of this folder shows up in the name of the ODBC driver once installed.
4. Open the folder in Windows Explorer, right click on the file "odbc\_install.exe", and select **Run As Administrator**.
5. Once the installation is complete, run the 32-bit (Windows/Microsoft) ODBC Data Source Administrator. (Type ODBC in the Windows search bar. From the choices, make sure to select the 32-bit version.)
6. Click the **Drivers** tab and verify the driver highlighted in the following screenshot is present. Note that the name may be different based on the driver version and folder name.



For testing or troubleshooting, a DSN can be configured in the System DSN tab, but this is not required.

See [Selecting a Database](#) (p. 99) and [Selecting a Table](#) (p. 104) for information on setting up the database in View Pro.

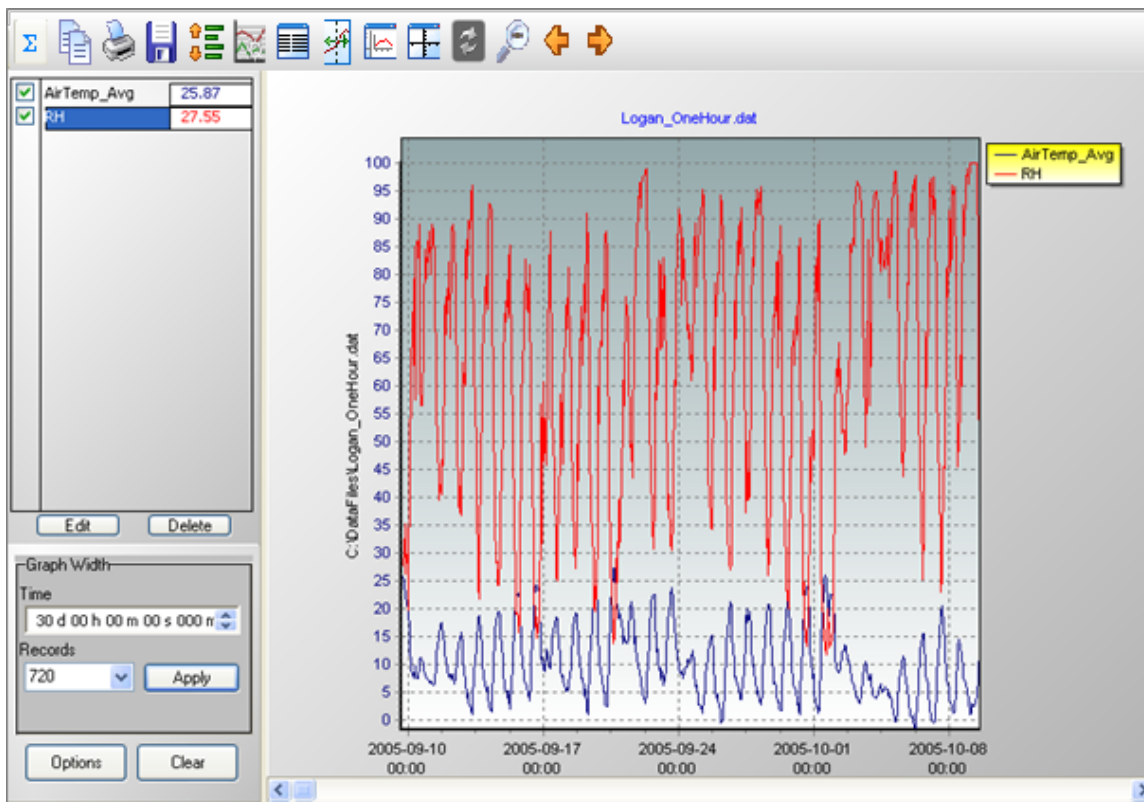


## 6.5 Data View Options

6.5.1 Line Graph .....	118
6.5.2 Histogram .....	127
6.5.3 XY Plot .....	134
6.5.4 Rainflow Histogram .....	139
6.5.5 FFT .....	144

### 6.5.1 Line Graph

From the Line Graph screen, you can graph data values on the y-axis against their timestamps on the x-axis.



Multiple Line Graph screens can be created.

#### Selecting Data to be Graphed

To select the data value(s) to be graphed, click the column heading(s) in the data file with a single mouse click. The selection will be highlighted and will automatically be added to the currently

selected graph. ((The currently selected graph is indicated on the main toolbar, and can be changed from the drop-down list.) Click and drag over multiple column headings to select multiple columns. (Note the columns will not switch to multiple colors until you release the mouse button.)

A partial column can be added by dragging the mouse pointer over the desired values. (Note that once a partial column is selected, it can be extended by holding the Shift key and clicking at a point below the current selection. The selection will be extended to that point.) Multiple partial columns can be added by holding the Ctrl key and dragging the mouse pointer over the desired values.

Data from multiple data files can be displayed in a single graph. This is done by opening multiple data files and selecting data to be graphed as described above. This may be useful when comparing data from multiple datalogger stations.

### **Creating Multiple Line Graphs**

To open an additional Line Graph, select "None" from the **Selected Graph** drop-down menu on the main View Pro toolbar. The highlighting in the data file will be cleared. Select the data that you would like graphed as described above and then press the Line Graph button. A new Line Graph will be created with your selection(s) graphed. You may then continue adding selections to the Line Graph as described above.

An unlimited number of Line Graphs can be opened using this same process.

You can navigate between multiple graphs by clicking on a graph or by choosing a graph from the **Selected Graph** drop-down list on the View Pro toolbar and pressing the Bring the selected graph to the front button.

**NOTE:** The highlighted selections in the data files will always indicate the values being graphed in the currently selected graph.

### **Edit**

Brings up a dialog box to set properties for the selected trace. (A trace is selected by clicking its name in the list above the Edit and Delete buttons.) This dialog box can be used to set properties for Display (name, color, line width, line style, and symbol style), Y Axis (scaling, limits, and title), and Marks.

### **Delete**

Deletes the selected trace from the graph. (A trace is selected by clicking its name in the list above the Edit and Delete buttons.)

### **Graph Width**

The Graph Width can be set either as a function of time or number of records.

**NOTE:** If the Graph Width **Time** or **Records** field is changed, when the Apply button is pressed, the other field will automatically be changed correspondingly.

**NOTE:** The number of **Records** to be displayed cannot be set to a number less than two. If a number less than two is entered, it will automatically be changed to two, when the Apply button is pressed.

## Time

The amount of time that will be shown on the graph in days, hours, minutes, seconds and milliseconds. Each element (days, hours, minutes, seconds, and milliseconds) can be highlighted separately. Once an element is highlighted, type in the desired value or use the arrow keys to increase or decrease. After entering the desired value, press the Apply button to make the change take effect.

## Records

The number of records that will be shown on the graph. Highlight the field and then type in the desired value or click on the arrow to pick a value from the drop-down list. After entering the desired value, press the Apply button to make the change take effect.


## Options

Brings up the Options dialog box for the graph. This dialog box can be used to set options for chart colors, margins, title, legend, points, and plotting options (line only, points and line, points only).

## Clear

Press this button to clear all traces and data contained in the graph.



## Zoom Feature

You can zoom in on a particular area of a Line Graph by holding the left mouse button and dragging the mouse pointer from top-left to bottom-right (or bottom-left to top-right) over the area to be zoomed. Pressing the Undo Zoom toolbar icon  or dragging the mouse pointer from bottom-right to top-left (or top-right to bottom-left) will undo the zoom.

## Scrolling Through the Graph

You can scroll through the graph by using the scroll bar at the bottom. You can also use the Page Up and Page Down buttons on your keyboard to scroll through the graph a **Graph Width** at a time.

Scrolling the graph will scroll the data on the data panel as well. Conversely, scrolling through a data panel will also scroll the currently selected graph.

Graph scrolling can be disabled by pressing the Lock Scrolling icon  on the toolbar. The icon will change to . Pressing the icon again will re-enable scrolling. When locked, the graph will not scroll. However, moving the scroll bar on either the graph or the data file will still scroll the data file.

## Right-Click Menus

Right-clicking on the graphical display area will bring up a menu from which you can choose **Export** to save the graph in a choice of formats, **Copy to Clipboard** to place the graph on the clipboard, **Print** to print the graph, or **Options** to bring up the graph's Options dialog box.

Right-clicking on a trace name in the list above the Edit and Delete buttons brings up a menu from which you can choose **Edit Selection** to bring up the trace options dialog box, **Delete Selection** to delete the selection from the graph, **Selection Summary** to see information about the trace, the data file, and the datalogger and program that generated the data file, **Assign to Left Axis** to show the selected trace's scale on the left y-axis, **Assign to Right Axis** to show the selected trace's scale on the right y-axis, **Check All** to check all traces and make them visible on the graph, **Uncheck All** to uncheck all traces and remove them from the graph. (See [Common vs. Independent Y-Axes](#) (p. 123) for more information on the y-axes.)

## Line Graph Toolbar Icons

The Line Graph includes the following toolbar icons:



**Statistics.** Displays statistics for each trace including Average, Standard Deviation, Minimum and Maximum. Note that these statistics are for the data displayed in the graph. They are not statistics for the entire column(s) of data. Note that an asterisk next to a statistics value indicates that the trace contains one or more bad data values (i.e., NAN, INF, etc.)

The Statistics button is disabled when a graph is in a zoomed state.



**Copy to Clipboard.** Places the graphic on the Windows clipboard. It can then be pasted into other applications.



**Print.** Prints the graph. Print options can be set before printing begins.



**Export.** Allows the graph to be exported in a choice of text or graphical formats.



**Lock Scrolling.** Locks and unlocks the scroll bar at the bottom of the graph.

When unlocked, moving the scroll bar on either the graph or the data file will scroll both the graph and the data file.

When locked, the graph will not scroll. However, moving the scroll bar on either the graph or the data file will still scroll the data file.

This is a toggle button. When the lock is currently enabled, there will be a



lock on top of the icon.

**Graph Options.** Opens a dialog box from which you can set properties for the graph including colors, margins, titles, legend, etc. This dialog box can also be brought up by pressing the Options button.



**Show Table.** Brings the main View Pro window in front of other windows, making the data file(s) visible.



**Show/Hide Graph Cursor.** A toggle button that shows and hides the graph cursor. The graph cursor is a vertical line extending from the top to the bottom of the graph display. When visible, you can click and drag the cursor across the graph. Data values at the current cursor position will be shown in the table. Timestamp at the current cursor position will be shown in the status bar.



**Show/Hide Gradient.** A toggle button that turns on and off the gradient background of the graph. It may be useful to hide the gradient, when printing the graph.



(Common)

**Common/Independent Axes.** When multiple data values are being graphed, determines whether they have common y-axes or independent y-axes.

When using common y-axes, one scale will apply to all traces assigned to the left y-axis and one scale will apply to all traces assigned to the right y-axis.



(Independent)

When using independent y-axes, the scale shown on each y-axis will apply only to the last selected trace assigned to that axis. (A trace is selected by clicking on its name in the list above the Edit and Delete buttons.)

Refer to [Common vs. Independent Y-Axes](#) (p. 123) for more information.



**Synchronize Axes.** Only enabled when data is being graphed from multiple data files and all of the data files have an overlapping time period.

When a graph contains traces from multiple data files, a box with a drop-down list will appear in the Graph Width options box. The data file chosen from the drop-down list indicates which data file and graph will be scrolled by the graph scroll bar. After using the scroll bar to scroll the indicated graph, the Synchronize Axes button may be pressed to synchronize the timestamps of the remaining graph(s) and data file(s) so that they are all displaying data from the same time period.



**Undo Zoom.** Returns the graph to its original state after zooming.



Jumps the graph to the position showing a full screen of data ending with the last record on the right-most part of the graph.



Jumps the graph to the position showing a full screen of data beginning with the first record on the left-most part of the graph.

### 6.5.1.1 Common vs. Independent Y-Axes

The Line Graph has a toolbar button that determines whether data values being graphed have common y-axes or independent y-axes. This is a toggle button. The icon on the button changes depending on which mode is currently active as shown below.

#### Common Y-Axes ( )

When using common y-axes, the same scale will apply to all traces assigned to an axis. Traces can be assigned to an axis by selecting the trace, right-clicking, and then choosing Assign to Left Axis or Assign to Right Axis. (A trace is selected by clicking on its name in the list above the Edit and Delete buttons.) The scale for the common left y-axis will spread from the lowest value of any trace assigned to the left y-axis to the highest value of any trace assigned to that axis. The scale for the common right y-axis will spread from the lowest value of any trace assigned to the right y-axis to the highest value of any trace assigned to that axis. By default, all traces are assigned to the left y-axis and the scale will spread from the lowest value of any trace being graphed to the highest value of any trace being graphed.

The graph legend will show an L or R next to the name of each trace to indicate whether that trace is assigned to the left or right y-axis.

The default y-axis title will be the name of the data file from which data values are being graphed.

#### Independent Y-Axes ( )

When using independent y-axes, the scale shown on each y-axis will apply only to the last selected trace that is assigned to that axis. (Traces can be assigned to an axis by selecting the trace, right-clicking, and then choosing Assign to Left Axis or Assign to Right Axis.) The scale will only show the value spread for the specified trace, but all traces will maintain their relative positions on the graph. The y-axis title will be the name of the last selected trace.

### 6.5.1.2 Line Graph Options

After setting the options described below, press OK to close the dialog box and keep the changes. Pressing Cancel will close the dialog box and abandon the changes.

#### Background

Determines the color for the back wall of the graph. Select **Solid** and press the ... button to select a solid color. Select **Gradient** to give the graph's back wall a gradient. Press the Edit Gradient button to display a Gradient Editor that can be used to select a default or customized gradient.

## Chart Background

Determines the color for the background of the graph. Select **Solid** and press the ... button to select a solid color. Select **Gradient** to give the graph's background a gradient. Press the Edit Gradient button to display a Gradient Editor that can be used to select a default or customized gradient.

## Legend Background

Determines the color for the legend background of the graph. Select **Solid** and press the ... button to select a solid color. Select **Gradient** to give the legend's background a gradient. Press the Edit Gradient button to display a Gradient Editor that can be used to select a default or customized gradient.

## Titles

### Show Graph Title

Determines whether a title is shown for the graph. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## Margins

Sets the left, top, bottom, and right margins of the graph as percentages of the available space.

## Legend

### Show Legend

Check this box to display a legend for the chart.

## Position

Determines where the legend appears on the chart. Choose Right or Top.

## X Axis

### Use record number on X Axis

Select this check box to display record numbers rather than timestamps on the X Axis. This may be useful when displaying data with gaps in the timestamps. With this option selected, you will see continuous data without the gaps. It may also be useful when displaying data with duplicate timestamps.

## Plotting Options

Determines how the data points will be drawn. Specify **Line Only** to simply draw a line between the data points; **Points and Line** to draw a line between the data points and draw a symbol at each point; or **Points Only** to draw a symbol at each data point without connecting the points with a line.

Note that when switching between **Line Only** and either of the other two options, the graph will temporarily be cleared and no traces will be visible until the OK button is pressed. This is because a different component is used for these options.

### **Graph Width**

The Default Graph Width can be set either as a function of time or number of records.

### **Set as Defaults**

Sets these settings as the defaults for a line graph. This applies to all of the settings on this page, For example if you change the **Chart Color** to **Solid** and the color to white and click **Set as Defaults**, every new line graph will be created with these settings.

## **6.5.1.3 Line Graph Trace Options—Display**

### **Selection**

#### **Name**

Determines the name that will be used to identify the trace in the list on the left side of the graph which shows the data values being graphed.

#### **Color**

Opens a dialog box from which to choose a color for the line and symbols used to display the data for this trace.

### **Trace**

#### **Line Width**

Sets the width of the line that is drawn between each point on the trace. The value must be between 1 and 10.

#### **Line Style**

Determines the style of line for the trace. Choose Solid Line, Dash, Dot, Dash Dot, or Dash Dot Dot.

#### **Symbol Style**

Sets the style of the symbol that is used to represent each data point for this trace. The available styles are Rectangle, Circle, Triangle, Down Triangle, Cross, Diagonal Cross, Star, Diamond, Small Dot, Nothing, Left Triangle, and Right Triangle.

Note that the symbols will not be shown if the **Plotting Options** for the graph is set to Line Only.



## 6.5.1.4 Line Graph Trace Options—Y Axis

### Scaling Option

#### Automatic Scaling

Configures the Y Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

#### Use Custom Limits

Allows the user to set the maximum and minimum values in the Custom Limits fields.

#### Logarithmic Scale

Configures the Y Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

Note that this option is not available if the trace has any negative values within the **Graph Width**. If **Logarithmic Scale** is selected and the graph is then scrolled such that there is at least one negative value within the **Graph Width**, the **Logarithmic Scale** will be automatically disabled.

### Custom Limits

#### Max Value

The maximum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

#### Min Value

The minimum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

### Axis Title

Determines whether a title is shown for the Y Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## 6.5.1.5 Line Graph Trace Options—Marks

### Show Marks

Determines whether marks are shown on the graph to display the value of data points. Select the check box to show marks.

### Round Frame

Sets the shape of the frame around each mark. Select the check box to make the frame round. Clear the check box to make the frame square.

## Transparent

Determines whether a frame is shown around each mark. Select the check box to display a frame around each mark. Clear the check box to display only text.

## Draw Every

Determines whether marks are shown for all or only some data points. The default value is one, which will show marks for every data point. Entering a value of  $n$  will cause marks to be shown only for every  $n$ th data point.

## Color

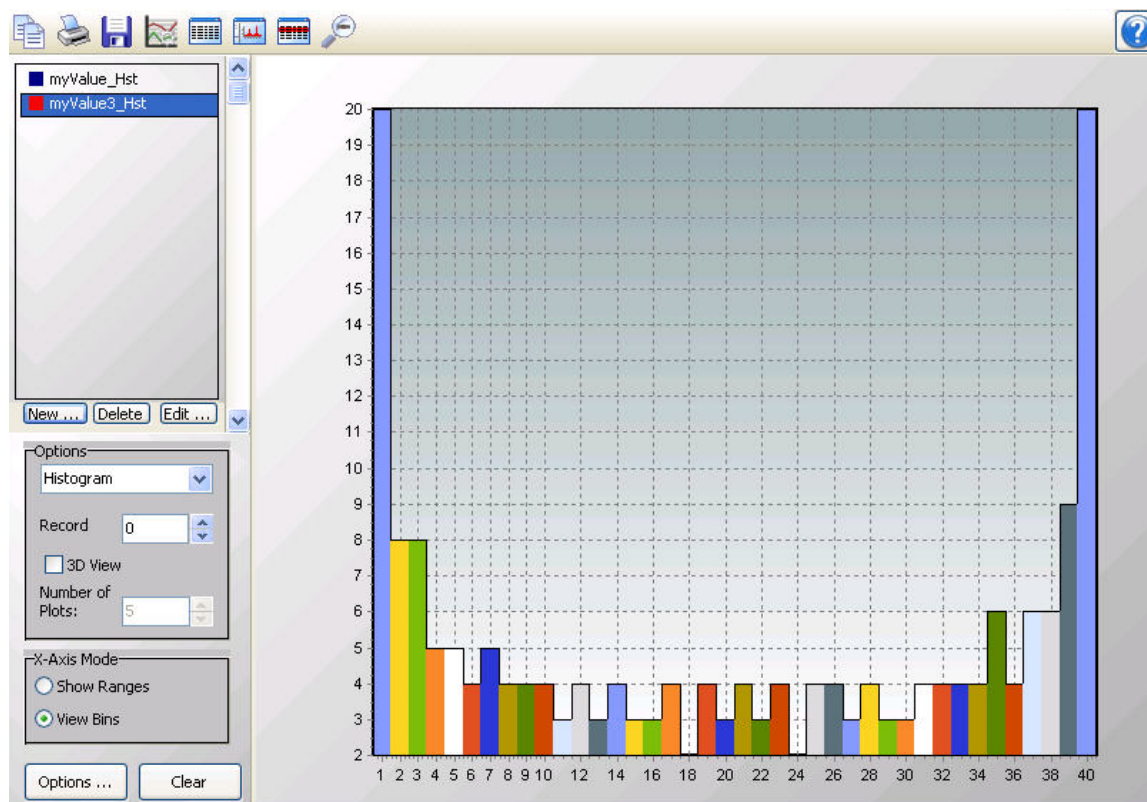
Press the button to open a dialog box from which to choose a color for the frames around the marks.

## Style

Determines the data contained in each mark. Choose Y-Value, X-Value, or X and Y Value.

## 6.5.2 Histogram

From the Histogram screen, you can view histogram data. The Histogram button on the toolbar will be enabled if there is at least one valid histogram in the currently selected data file.



**NOTE:** View Pro does not create histogram data from time series information. It only displays histogram data contained in a \*.DAT file. Histogram data in a \*.DAT file is created by using the CRBasic Histogram instruction in a CRBasic program Data Table.


View Pro cannot display histograms from a TOAC11 data file.

Multiple Histogram screens can be created.

## Selecting Data to be Viewed

### From a Histogram Screen

When a Histogram screen is first opened with no histogram records selected in the data file, the Histogram Setup dialog box will open which allows you to set up the Histogram. The first option is a drop-down list that shows the available histograms in the currently selected data file. Select the histogram that you would like to view. The second option allows you to choose which record of the histogram you would like to view initially. Type in a number directly or use the arrow keys to the right of the box to change the value.

This dialog box can also be opened from a button, , on the Histogram toolbar. This allows you to change the options for the histogram record that is selected in the list on the left side of the Histogram screen.

Additional histogram records can be added by pressing the New button. (These additional records can be from either the same histogram or a different histogram in your data file.) You can then choose which histogram record is being displayed by selecting it in the list.

### From the Data Grid

You can also select histogram records directly from a data file to be displayed on a Histogram screen. Clicking on any data value in a histogram record will select that histogram record. Histogram records can be selected before the Histogram screen is opened with the [Selected Graph](#) (p. 87) set to None. When the Histogram screen is opened, all selected histogram records will be listed on the left side of the Histogram screen. A histogram record can then be displayed by clicking on it in the list. Once the Histogram screen is opened, additional histogram records can be added to the Histogram screen by selecting them in the data file as described above.

**NOTE:** All histogram records from the same histogram will have the same default name in the list. They can be distinguished by the colored boxes next to their names. Each box is the same color with which that histogram record is highlighted in the data file. It is also the color with which that histogram record is displayed if the “Use Selection Color” option is chosen in the Selection Properties dialog box. The color associated with a histogram record can also be

changed from this dialog box. (The Selection Properties dialog box is opened by clicking on the histogram record in the list and then pressing the Edit button.)

### **New**

Brings up the Histogram Setup dialog box to allow you to add a new histogram record to the display.

### **Delete**

Deletes the selected histogram record from the Histogram.

### **Edit**

Brings up a dialog box to set properties for the selected histogram record. This dialog box can be used to set properties for Display (name, color, marks) , Y Axis (scaling, limits, and title), and X Axis (scaling, limits, title).

### **Options**

Determines the graph type. Select Area, Histogram, Line, or Bar from the drop-down list.

### **Record**

Indicates which record of the histogram is being viewed. The arrow buttons can be used to scroll through records of the histogram.

### **3D View**

Determines whether the histogram is viewed in 2D or 3D mode. Select the checkbox to view the histogram in 3D. Clear the checkbox to view the histogram in 2D.

### **Number of Plots**

This field is only enabled for **3D View**. Sets the number of plots (histogram records) to be viewed.

### **X-Axis Mode**

Determines how the labels on the X-Axis are displayed. Select **Show Ranges** to have ranges of data values shown on the X-Axis. Select **View Bins** to have bin numbers shown on the X-Axis.


### **Options**

Brings up the Options dialog box for the Histogram. This dialog box can be used to set the title, margins, and chart colors.

### **Clear**

Press this button to clear all histogram records contained in the Histogram.

## Zoom Feature

You can zoom in on a particular area of a Histogram by holding the left mouse button and dragging the mouse pointer from top-left to bottom-right (or bottom-left to top-right) over the area to be zoomed. Pressing the Undo Zoom toolbar icon  or dragging the mouse pointer from bottom-right to top-left (or top-right to bottom-left) will undo the zoom.

In **3D View**, you can also zoom in and out by using the Page Down and Page Up buttons on your keyboard.

## Rotating the Histogram

In **3D View**, you can rotate the Histogram by using the scroll bars at the bottom and right of the Histogram.

## Right-Click Menus

Right-clicking on the graphical display area will bring up a menu from which you can choose **Export** to save the Histogram in a choice of formats, **Copy to Clipboard** to place the Histogram on the clipboard, **Print** to print the Histogram, or **Options** to bring up the Histogram's Options dialog box.

Right-clicking on a histogram record in the list above the New, Edit and Delete buttons brings up a menu from which you can choose **Edit Selection** to bring up the Selection Properties dialog box, **Delete Selection** to delete the selection from the Histogram, or **Selection Summary** to see information about the histogram record, the data file, and the datalogger and program that generated the data file.

## Histogram Toolbar Icons

The Histogram includes the following toolbar icons:



**Copy to Clipboard.** Places the Histogram graphic on the Windows clipboard. It can then be pasted into other applications.



**Print.** Prints the Histogram. Print options can be set before printing begins.



**Export.** Allows the Histogram to be exported in a choice of text or graphical formats.



**Histogram Options.** Opens a dialog box from which you can set properties for the Histogram including scaling, colors, margins, titles, etc. This dialog box can also be brought up by pressing the Options button.



**Show Table.** Brings the main View Pro window in front of other windows, making the data file(s) visible.



**Show/Hide Gradient.** A toggle button that turns on and off the gradient background of the Histogram. It may be useful to hide the gradient, when printing the Histogram.



**Modify Selection.** Brings up the Histogram Setup dialog box from which you can change the options for the selection.



**Undo Zoom.** Returns the Histogram to its original state after zooming.

## 6.5.2.1 Histogram Options

### Title

#### Show Graph Title

Determines whether a title is shown for the graph. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

### Chart Colors

#### Back Wall Color

Determines the color for the back wall of the graph. Press the ... button to select a color.

#### Use Gradient

Select the check box give the graph's plot area a gradient. Clear the check box to clear the gradient. Press the Edit Gradient button to the right of the field (...) to display a Gradient Editor that can be used to select a default or customized gradient.

### Margins

Sets the left, top, bottom, and right margins of the graph as percentages of the available space.

## 6.5.2.2 Histogram Selection Properties—Display

### Selection

#### Name

Determines the name that will be used to identify the selection in the list on the left side of the Histogram which shows the histogram records being viewed.

#### Color

Opens a dialog box from which to choose a color to be used to display the data if **Use Selection Color** is selected. Note that this color will be shown next to this histogram record's name in the list of histogram records being graphed. This color will also be used to highlight the histogram record in the data file.

#### Color Each Plot

In 3D View, uses a different color for each plot (histogram record) being displayed.

### **Color Each Bin**

Uses a different color for each bin of the histogram record.

### **Use Selection Color**

The specified color is used for all data points.

### **Marks**

#### **Show Marks**

Determines whether marks are shown on the graph to display the value of data points. Select the check box to show marks.

#### **Round Frame**

Sets the shape of the frame around each mark. Select the check box to make the frame round. Clear the check box to make the frame square.

#### **Transparent**

Determines whether a frame is shown around each mark. Select the check box to display a frame around each mark. Clear the check box to display only text.

#### **Draw Every**

Determines whether marks are shown for all or only some data points. The default value is one, which will show marks for every data point. Entering a value of  $n$  will cause marks to be shown only for every  $n$ th data point.

### **Color**

Press the button to open a dialog box from which to choose a color for the frames around the marks.

### **Style**

Determines the data contained in each mark. Choose Y-Value, X-Value, or X and Y Value.

## **6.5.2.3 Histogram Selection Properties—Y Axis**

### **Scaling Option**

#### **Automatic Scaling**

Configures the Y Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

#### **Use Custom Limits**

Allows the user to set the maximum and minimum values in the Custom Limits fields.

## Logarithmic Scale

Configures the Y Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

### Custom Limits

#### Max Value

The maximum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

#### Min Value

The minimum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

### Scientific Notation

Determines whether values on the Y Axis are shown in scientific notation. Select the check box to display values in scientific notation.

### Axis Title

Determines whether a title is shown for the Y Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## 6.5.2.4 Histogram Selection Properties—X Axis

### Scaling Option

#### Automatic Scaling

Configures the X Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

#### Use Custom Limits

Allows the user to set the maximum and minimum values in the Custom Limits fields.

### Logarithmic Scale

Configures the X Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

### Custom Limits

#### Max Value

The maximum value on the X Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.



## Min Value

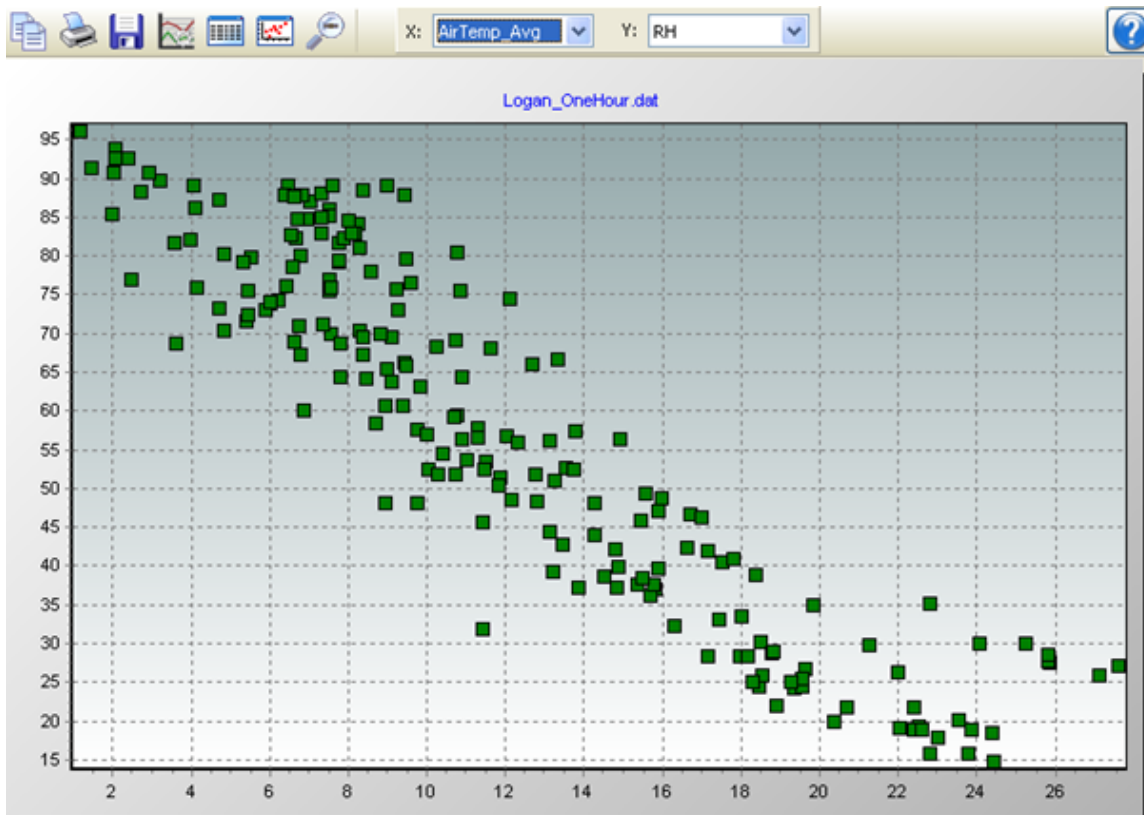
The minimum value on the X Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

## Axis Title

Determines whether a title is shown for the X Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## 6.5.3 XY Plot

From the XY Plot screen, you can graph a data value on the y-axis against a different data value on the x-axis. The user specifies what will be used for both the X axis data value and the Y axis data value. Each Y axis data value is plotted against the X axis data value with the identical timestamp.



Multiple XY Plot screens can be created.

### Selecting Data to be Plotted

To select the data values to be plotted, click the column headings in the data file with a single mouse click. The selected columns will be highlighted and will be added to the XY Plot in both

the X and Y drop-down lists. Select from the X and Y drop-down lists the values to be used for the X axis and Y axis, respectively.

### **New**

Adds another series to the XY Plot. The X and Y values for the new series can then be selected from the X and Y drop-down list boxes.


### **Delete**

Deletes the selected series from the XY Plot.

### **Edit**

Brings up a dialog box to set properties for the selected series. This dialog box can be used to set properties for symbols and marks.

### **Zoom Feature**

You can zoom in on a particular area of an XY Plot by holding the left mouse button and dragging the mouse pointer from top-left to bottom-right (or bottom-left to top-right) over the area to be zoomed. Pressing the Undo Zoom toolbar icon  or dragging the mouse pointer from bottom-right to top-left (or top-right to bottom-left) will undo the zoom.

### **Right-Click Menu**

Right-clicking on the graphical display area will bring up a menu from which you can choose **Export** to save the XY Plot in a choice of formats, **Copy to Clipboard** to place the XY Plot on the clipboard, **Print** to print the XY Plot, or **Options** to bring up the XY Plot's Options dialog box.

Right-clicking on a series in the list above the New, Edit and Delete buttons brings up a menu from which you can choose **Edit Series** to bring up the Series options dialog box or **Delete Series** to delete the series from the XY Plot.

### **XY Plot Toolbar Icons**

The XY Plot includes the following toolbar icons:



**Copy to Clipboard.** Places the XY Plot graphic on the Windows clipboard. It can then be pasted into other applications.



**Print.** Prints the XY Plot. Print options can be set before printing begins.



**Export.** Allows the XY Plot to be exported in a choice of text or graphical formats.



**Graph Options.** Opens a dialog box from which you can set properties for the XY Plot including colors, margins, titles, scaling, etc.



**Show Table.** Brings the main View Pro window in front of other windows, making the data file(s) visible.



**Show/Hide Gradient.** A toggle button that turns on and off the gradient background of the XY Plot. It may be useful to hide the gradient, when printing the XY Plot.



**Undo Zoom.** Returns the XY Plot to its original state after zooming.

### 6.5.3.1 XY Plot Options—Visual Display

#### Chart Colors

##### Back Wall Color

Determines the color for the back wall of the graph. Press the ... button to select a color.

##### Use Gradient

Select the check box give the graph's plot area a gradient. Clear the check box to clear the gradient. Press the Edit Gradient button to the right of the field (...) to display a Gradient Editor that can be used to select a default or customized gradient.

#### Margins

Sets the left, top, bottom, and right margins of the graph as percentages of the available space.

#### Titles

##### Show Graph Title

Determines whether a title is shown for the graph. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

### 6.5.3.2 XY Plot Options—Y Axis

#### Scaling Option

##### Automatic Scaling

Configures the Y Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

##### Use Custom Limits

Allows the user to set the maximum and minimum values in the Custom Limits fields.

##### Logarithmic Scale

Configures the Y Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

Note that this option is not available if the Y Axis data value has any negative values.

## **Custom Limits**

### **Max Value**

The maximum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

### **Min Value**

The minimum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

### **Axis Title**

Determines whether a title is shown for the Y Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## **6.5.3.3 XY Plot Options—X Axis**

### **Scaling Option**

#### **Automatic Scaling**

Configures the X Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

#### **Use Custom Limits**

Allows the user to set the maximum and minimum values in the Custom Limits fields.

#### **Logarithmic Scale**

Configures the X Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

Note that this option is not available if the X Axis data value has any negative values.

## **Custom Limits**

### **Max Value**

The maximum value on the X Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

### **Min Value**

The minimum value on the X Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

## Axis Title

Determines whether a title is shown for the X Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## 6.5.3.4 XY Plot Series Options

### Symbol

#### Color

Opens a dialog box from which to choose a color for the symbols used to display the data for the plot.

#### Style

Sets the style of the symbol that is used to represent each data point for this trace. The available styles are Rectangle, Circle, Triangle, Right Triangle, Down Triangle, Left Triangle, Cross, Diagonal Cross, Star, Diamond, Small Dot, or Nothing.

### Marks

#### Show Marks

Determines whether marks are shown on the plot to display the value of data points. Select the check box to show marks.

#### Round Frame

Sets the shape of the frame around each mark. Select the check box to make the frame round. Clear the check box to make the frame square.

#### Transparent

Determines whether a frame is shown around each mark. Select the check box to display a frame around each mark. Clear the check box to display only text.

#### Draw Every

Determines whether marks are shown for all or only some data points. The default value is one, which will show marks for every data point. Entering a value of  $n$  will cause marks to be shown only for every  $n$ th data point.

#### Color

Press the button to open a dialog box from which to choose a color for the frames around the marks.

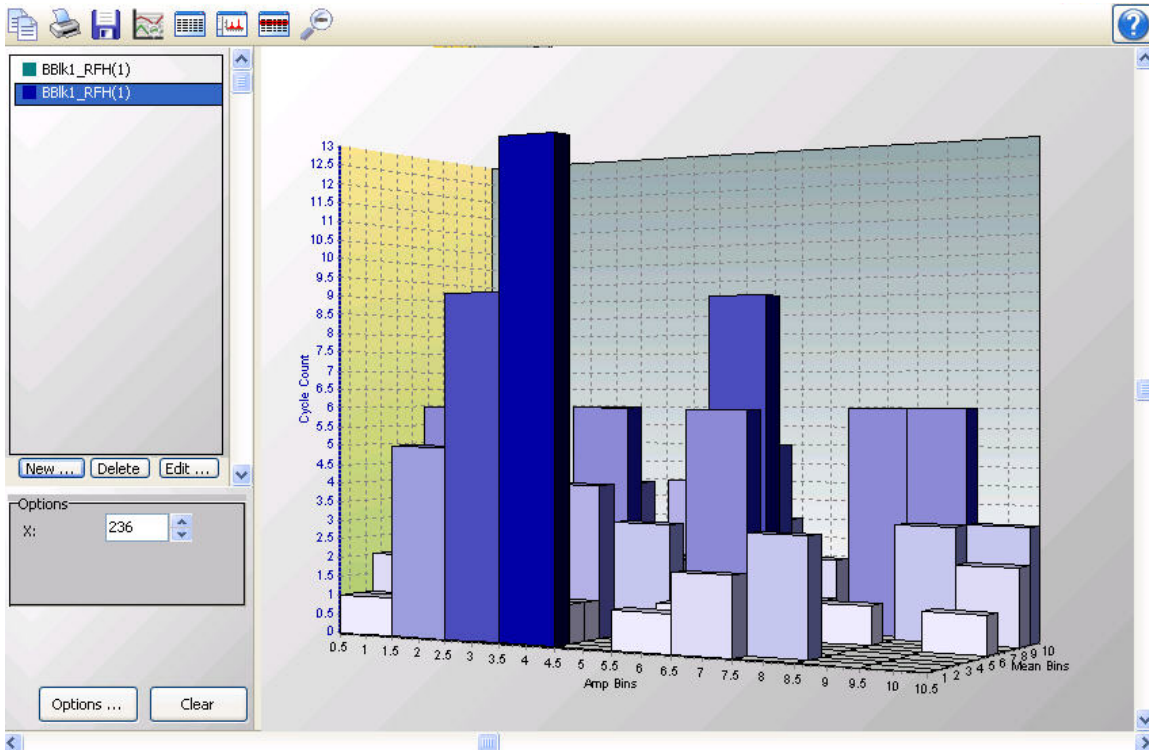
#### Style

Determines the data contained in each mark. Choose Y-Value, X-Value, or X and Y Value.

## 6.5.4 Rainflow Histogram

From the Rainflow Histogram screen, you can view rainflow histogram data. The Rainflow Histogram button on the toolbar will be enabled if there is at least one valid rainflow histogram in the currently selected data file.

A rainflow histogram is a 3D representation based on the rainflow counting algorithm of Endo and Matsuishi which was first published in 1968. These diagrams can be used to monitor fatigue levels of structures under stress such as components of a large bridge.



**NOTE:** View Pro does not create rainflow histogram data from time series information. It only displays rainflow histogram data contained in a \*.DAT file. Rainflow histogram data in a \*.DAT file is created by using the CRBasic Rainflow instruction in a CRBasic program Data Table.


Multiple Rainflow Histogram screens can be created.

### Selecting Data to be Viewed

#### From a Rainflow Histogram Screen

When a Rainflow Histogram screen is first opened with no rainflow histogram records selected in the data file, the Rainflow Histogram Setup dialog box will open which allows you to set up the Rainflow Histogram. The first option is a drop-down list that shows the available rainflow histograms in the currently selected data file. Select the rainflow histogram that you would like to

view. The second option allows you to choose which record of the rainflow histogram you would like to view initially. Type in a number directly or use the arrow keys to the right of the box to change the value.

This dialog box can also be opened from a button, , on the Rainflow Histogram toolbar. This allows you to change the options for the rainflow histogram record that is selected in the list on the left side of the Rainflow Histogram screen.

Additional rainflow histogram records can be added by pressing the New button. (These additional records can be from either the same rainflow histogram or a different rainflow histogram in your data file.) You can then choose which rainflow histogram record is being displayed by selecting it in the list.

### From the Data Grid

You can also select rainflow histogram records directly from a data file to be displayed on a Rainflow Histogram screen. Clicking on any data value in a rainflow histogram record will select that rainflow histogram record. Rainflow histogram records can be selected before the Rainflow Histogram screen is opened with the [Selected Graph](#) (p. 87) set to None. When the Rainflow Histogram screen is opened, all selected rainflow histogram records will be listed on the left side of the Rainflow Histogram screen. A rainflow histogram record can then be displayed by clicking on it in the list. Once the Rainflow Histogram screen is opened, additional rainflow histogram records can be added to the Rainflow Histogram screen by selecting them in the data file as described above.

**NOTE:** All rainflow histogram records from the same rainflow histogram will have the same default name in the list. They can be distinguished by the colored boxes next to their names. Each box is the same color with which that rainflow histogram record is highlighted in the data file. It is also the color with which that rainflow histogram record is displayed. The color associated with a histogram record can be changed from the Selection Properties dialog box. (The Selection Properties dialog box is opened by clicking on the rainflow histogram record in the list and then pressing the Edit button.)

### New

Brings up the Rainflow Histogram Setup dialog box to allow you to add a new rainflow histogram record to the display.

### Delete

Deletes the selected rainflow histogram record from the Rainflow Histogram.

## Edit

Brings up a dialog box to set properties for the selected rainflow histogram record. This dialog box can be used to set properties for Display (name, color, marks) and Axes (scaling, limits, and title).

## Options

### X

Indicates which record of the rainflow histogram is being viewed. The arrow buttons can be used to scroll through records of the rainflow histogram.


## Options

Brings up the Options dialog box for the Rainflow Histogram. This dialog box can be used to set options for chart colors, margins, and the chart title.

## Clear

Press this button to clear all rainflow histogram records contained in the Rainflow Histogram.

## Zoom Feature

You can zoom in on a particular area of a Rainflow Histogram by holding the left mouse button and dragging the mouse pointer from top-left to bottom-right (or bottom-left to top-right) over the area to be zoomed. Pressing the Undo Zoom toolbar icon  or dragging the mouse pointer from bottom-right to top-left (or top-right to bottom-left) will undo the zoom.

You can also zoom in and out by using the Page Down and Page Up buttons on your keyboard.

## Rotating the Rainflow Histogram

You can rotate the Rainflow Histogram by using the scroll bars at the bottom and right of the Rainflow Histogram.

## Right-Click Menus

Right-clicking on the graphical display area will bring up a menu from which you can choose **Export** to save the Rainflow Histogram in a choice of formats, **Copy to Clipboard** to place the Rainflow Histogram on the clipboard, **Print** to print the Rainflow Histogram, or **Options** to bring up the Rainflow Histogram's Options dialog box.

Right-clicking on a rainflow histogram record in the list above the New, Edit and Delete buttons brings up a menu from which you can choose **Edit Selection** to bring up the Selection Properties dialog box, **Delete Selection** to delete the selection from the graph, or **Selection Summary** to see information about the rainflow histogram record, the data file, and the datalogger and program that generated the data file.



## Rainflow Histogram Toolbar Icons

The Rainflow Histogram includes the following toolbar icons:



**Copy to Clipboard.** Places the Rainflow Histogram graphic on the Windows clipboard. It can then be pasted into other applications.



**Print.** Prints the Rainflow Histogram. Print options can be set before printing begins.



**Export.** Allows the Rainflow Histogram to be exported in a choice of text or graphical formats.



**Graph Options.** Opens a dialog box from which you can set properties for the Rainflow Histogram including scaling, colors, margins, titles, etc. This dialog box can also be brought up by pressing the Options button.



**Show Table.** Brings the main View Pro window in front of other windows, making the data file(s) visible.



**Show/Hide Gradient.** A toggle button that turns on and off the gradient background of the Rainflow Histogram. It may be useful to hide the gradient, when printing the Rainflow Histogram.



**Modify Selection.** Brings up the Rainflow Histogram Setup dialog box from which you can change the options for the selection.



**Undo Zoom.** Returns the Rainflow Histogram to its original state after zooming.

### 6.5.4.1 Rainflow Histogram Options

#### Chart Colors

##### Back Wall Color

Determines the color for the back wall of the graph. Press the ... button to select a color.

##### Use Gradient

Select the check box give the graph's plot area a gradient. Clear the check box to clear the gradient. Press the Edit Gradient button to the right of the field (...) to display a Gradient Editor that can be used to select a default or customized gradient.

##### Margins

Sets the left, top, bottom, and right margins of the graph as percentages of the available space.

##### Titles

##### Show Graph Title

Determines whether a title is shown for the graph. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## **6.5.4.2 Rainflow Histogram Selection Properties-Display**

### **Selection**

#### **Name**

Determines the name that will be used to identify the selection in the list on the left side of the Rainflow Histogram which shows the rainflow histogram records being viewed.

#### **Color**

Opens a dialog box from which to choose a color to be used to display this rainflow histogram record. Note that this color will be shown next to this rainflow histogram record's name in the list of rainflow histogram records being graphed. This color will also be used to highlight the rainflow histogram record in the data file.

### **Marks**

#### **Show Marks**

Determines whether marks are shown on the graph to display the value of data points. Select the check box to show marks.

#### **Round Frame**

Sets the shape of the frame around each mark. Select the check box to make the frame round. Clear the check box to make the frame square.

#### **Transparent**

Determines whether a frame is shown around each mark. Select the check box to display a frame around each mark. Clear the check box to display only text.

#### **Draw Every**

Determines whether marks are shown for all or only some data points. The default value is one, which will show marks for every data point. Entering a value of  $n$  will cause marks to be shown only for every  $n$ th data point.

#### **Color**

Press the button to open a dialog box from which to choose a color for the frames around the marks.

#### **Style**

Determines the data contained in each mark. Choose Z-Value, Y-Value, or Y and Z Value.

### 6.5.4.3 Rainflow Histogram Selection Properties-Axes

**NOTE:** For a Rainflow Histogram, the Z Axis is vertical, the Y Axis is horizontal, and the X Axis is depth.

#### **Z Axis Scaling Option**

##### **Automatic Scaling**

Configures the Z Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

##### **Use Custom Limits**

Allows the user to set the maximum and minimum values in the Custom Limits fields.

##### **Logarithmic Scale**

Configures the Z Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

#### **Z Axis Custom Limits**

##### **Max Value**

The maximum value on the Z Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

##### **Min Value**

The minimum value on the Z Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

#### **Z Axis**

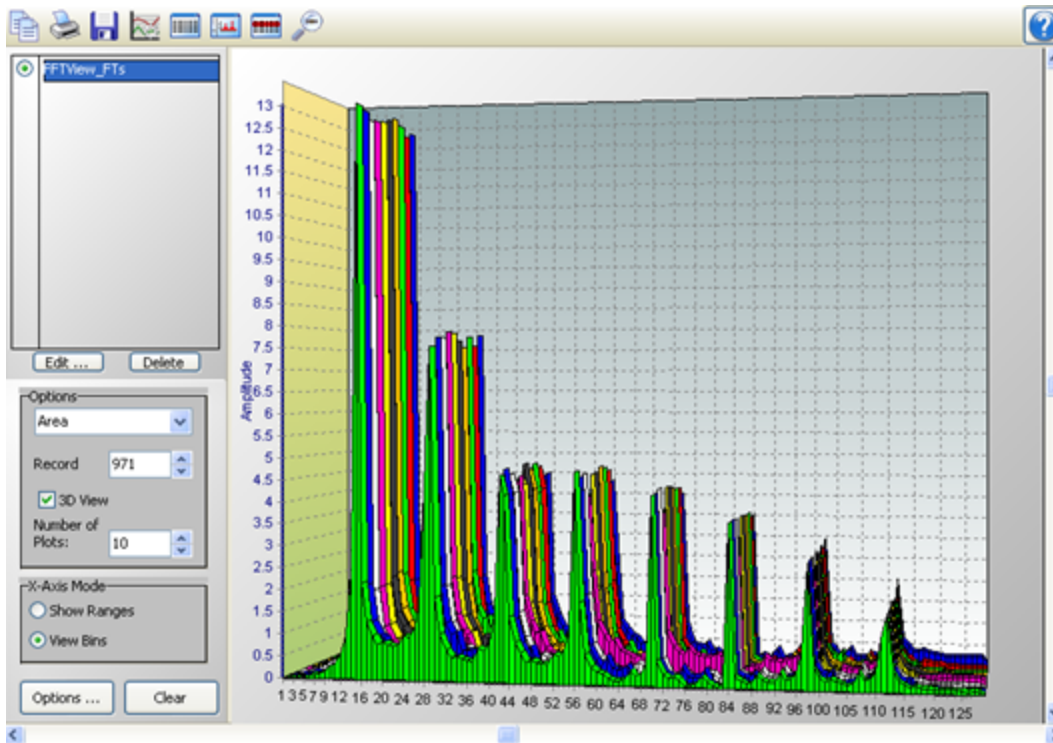
Determines whether a title is shown for the Z Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

#### **Y Axis**

Determines whether a title is shown for the Y Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

## 6.5.5 FFT

From the FFT screen, you can view FFT data. The FFT button on the toolbar will be enabled if there is at least one valid FFT in the currently selected data file.



**NOTE:** View Pro does not create FFT data from time series information. It only displays FFT data contained in a \*.DAT file. FFT data in a \*.DAT file is created by using the CRBasic FFT instruction in a CRBasic program Data Table.


View Pro cannot display FFTs from a TOACI1 data file.

Multiple FFT screens can be created.

## Selecting Data to be Viewed

### From an FFT Screen

When an FFT screen is opened, a Fast Fourier Transform Setup dialog box will open which allows you to set up the FFT. The first option is a drop-down list that shows the available FFTs in the currently selected data file. Select the FFT that you would like to view. The second option allows you to choose which record of the FFT you would like to view initially. Type in a number directly or use the arrow keys to the right of the box to change the value.

This dialog box can also be opened from a button, , on the FFT toolbar. This allows you to change the options for the FFT record that is selected in the list on the left side of the FFT screen.

Additional FFT records can be added by pressing the New button. (These additional records can be from either the same FFT or a different FFT in your data file.) You can then choose which FFT record is being displayed by selecting it in the list.

## From the Data Grid

You can also select FFT records directly from a data file to be displayed on an FFT screen. Clicking on any data value in an FFT record will select that FFT record. FFT records can be selected before the FFT screen is opened with the [Selected Graph](#) (p. 87) set to None. When the FFT screen is opened, all selected FFT records will be listed on the left side of the FFT screen. An FFT record can then be displayed by clicking on it in the list. Once the FFT screen is opened, additional FFT records can be added to the FFT screen by selecting them in the data file as described above.

**NOTE:** All FFT records from the same FFT will have the same default name in the list. They can be distinguished by the colored boxes next to their names. Each box is the same color with which that FFT record is highlighted in the data file. It is also the color with which that FFT record is displayed if the "Use Selection Color" option is chosen in the Selection Properties dialog box. The color associated with an FFT record can also be changed from this dialog box. (The Selection Properties dialog box is opened by clicking on the FFT record in the list and then pressing the Edit button.)

### New

Brings up the Fast Fourier Transform Setup dialog box to allow you to add a new FFT record to the display.

### Delete

Deletes the selected FFT record from the FFT.

### Edit

Brings up a dialog box to set properties for the selected FFT record. This dialog box can be used to set properties for Display (name, color, marks), Y Axis (scaling, limits, and title), and X Axis (scaling, limits, title).

### Options

Determines the graph type. Select Area, Histogram, Line, or Bar from the drop-down list.

### Record

Indicates which record of the FFT is being viewed. The arrow buttons can be used to scroll through records of the FFT.

### 3D View

Determines whether the FFT is viewed in 2D or 3D mode. Select the checkbox to view the FFT in 3D. Clear the checkbox to view the FFT in 2D.

### Number of Plots

This field is only enabled for **3D View**. Sets the number of plots (FFT records) to be viewed.

## X-Axis Mode

Determines how the labels on the X-Axis are displayed. Select **Show Ranges** to have ranges of data values shown on the X-Axis. Select **View Bins** to have bin numbers shown on the X-Axis.


## Options

Brings up the Options dialog box for the FFT. This dialog box can be used to set the title, margins, and chart colors.

## Clear

Press this button to clear all FFT records contained in the FFT.

## Zoom Feature

You can zoom in on a particular area of an FFT by holding the left mouse button and dragging the mouse pointer from top-left to bottom-right (or bottom-left to top-right) over the area to be zoomed. Pressing the Undo Zoom toolbar icon  or dragging the mouse pointer from bottom-right to top-left (or top-right to bottom-left) will undo the zoom.

In **3D View**, you can also zoom in and out by using the Page Down and Page Up buttons on your keyboard.

## Rotating the FFT

In **3D View**, you can rotate the FFT by using the scroll bars at the bottom and right of the FFT.

## Right-Click Menus

Right-clicking on the graphical display area will bring up a menu from which you can choose **Export** to save the FFT in a choice of formats, **Copy to Clipboard** to place the FFT on the clipboard, **Print** to print the FFT, or **Options** to bring up the FFT's Options dialog box.

Right-clicking on an FFT record in the list above the New, Edit and Delete buttons brings up a menu from which you can choose **Edit Selection** to bring up the Selection Properties dialog box, **Delete Selection** to delete the selection from the graph, or **Selection Summary** to see information about the FFT record, the data file, and the datalogger and program that generated the data file.

## FFT Toolbar Icons

The FFT includes the following toolbar icons:



**Copy to Clipboard.** Places the FFT graphic on the Windows clipboard. It can then be pasted into other applications.



**Print.** Prints the FFT. Print options can be set before printing begins.



**Export.** Allows the FFT to be exported in a choice of text or graphical formats.



**FFT Options.** Opens a dialog box from which you can set properties for the FFT including scaling, colors, margins, titles, etc. This dialog box can also be brought up



by pressing the Options button.



**Show Table.** Brings the main View Pro window in front of other windows, making the data file(s) visible.



**Show/Hide Gradient.** A toggle button that turns on and off the gradient background of the FFT. It may be useful to hide the gradient, when printing the FFT.



**Modify Selection.** Brings up the Fast Fourier Transform Setup dialog box from which you can change the options for the selection.

**Undo Zoom.** Returns the FFT to its original state after zooming.

## 6.5.5.1 FFT Options

### Title

#### Show Graph Title

Determines whether a title is shown for the graph. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

### Chart Colors

#### Back Wall Color

Determines the color for the back wall of the graph. Press the ... button to select a color.

#### Use Gradient

Select the check box give the graph's plot area a gradient. Clear the check box to clear the gradient. Press the Edit Gradient button to the right of the field (...) to display a Gradient Editor that can be used to select a default or customized gradient.

### Margins

Sets the left, top, bottom, and right margins of the graph as percentages of the available space.

## 6.5.5.2 FFT Selection Properties-Display

### Selection

#### Name

Determines the name that will be used to identify the selection in the list on the left side of the FFT which shows the FFT records being viewed.

#### Color

Opens a dialog box from which to choose a color to be used to display the data if **Use Selection Color** is selected. Note that this color will be shown next to this FFT record's name in the list of

FFT records being graphed. This color will also be used to highlight the FFT record in the data file.

### **Color Each Plot**

In 3D View, uses a different color for each plot (FFT record) of the trace.

### **Color Each Bin**

Uses a different color for each bin of the FFT record.

### **Use Selection Color**

The specified color is used for all data points.

### **Marks**

#### **Show Marks**

Determines whether marks are shown on the graph to display the value of data points. Select the check box to show marks.

#### **Round Frame**

Sets the shape of the frame around each mark. Select the check box to make the frame round. Clear the check box to make the frame square.

#### **Transparent**

Determines whether a frame is shown around each mark. Select the check box to display a frame around each mark. Clear the check box to display only text.

#### **Draw Every**

Determines whether marks are shown for all or only some data points. The default value is one, which will show marks for every data point. Entering a value of  $n$  will cause marks to be shown only for every  $n$ th data point.

#### **Color**

Press the button to open a dialog box from which to choose a color for the frames around the marks.

#### **Style**

Determines the data contained in each mark. Choose Y-Value, X-Value, or X and Y Value.



### 6.5.5.3 FFT Selection Properties-Y Axis

#### Scaling Option

##### Automatic Scaling

Configures the Y Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

##### Use Custom Limits

Allows the user to set the maximum and minimum values in the Custom Limits fields.

##### Logarithmic Scale

Configures the Y Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

Note that this option is not available if the FFT has any negative values.

#### Custom Limits

##### Max Value

The maximum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

##### Min Value

The minimum value on the Y Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

#### Scientific Notation

Determines whether values on the Y Axis are shown in scientific notation. Select the check box to display values in scientific notation.

#### Axis Title

Determines whether a title is shown for the Y Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

### 6.5.5.4 FFT Selection Properties-X Axis

#### Scaling Option

##### Automatic Scaling

Configures the X Axis to a linear scale. The maximum and minimum values will be automatically chosen based on the maximum and minimum data values in the graph.

## Use Custom Limits

Allows the user to set the maximum and minimum values in the Custom Limits fields.

## Logarithmic Scale

Configures the X Axis to a logarithmic scale. **Logarithmic Scale** can be used with either **Automatic Scaling** or **Use Custom Limits**.

## Custom Limits

### Max Value

The maximum value on the X Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

### Min Value

The minimum value on the X Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

## Axis Title

Determines whether a title is shown for the X Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.


# 6.6 Printing


6.6.1 Printing Options ..... 151


6.6.2 Print Preview ..... 152

## 6.6.1 Printing Options


### Printing Text

To print numerical data, press the print button  or select File | Print from the menu. A dialog box will appear allowing you to choose the printer, print range, number of copies, etc. After setting the properties, press OK to print the data.

To preview your data before printing, press the print preview button  or select File | Print Preview from the menu. From Print Preview you can browse among the pages that will be printed and change the paper orientation if desired. You can zoom in on a particular area of the previewed page by left-clicking the page. You can zoom out by left-clicking with the Shift button pressed. You can pan across a page by right-clicking and dragging the page. To return to normal











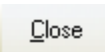

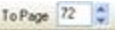
view, choose the Page Width or the Full Page icon. Simply press the print button  on the toolbar to print one or more pages. See the online help for details of the Print Preview options.

## Printing Graphs

With a graph window opened, click the print button  to preview the printed page and set various printing options. Then select the Print button to print the graph. You can also right-click the graph to bring up a menu from which you can select Print.

## 6.6.2 Print Preview

The Print Preview window allows you to review what the data will look like on the printed page and provides the option to print the file. The Print Preview window has the following toolbar icons:

	Displays the first page of the data file.
	Displays the previous page of the data file.
	Displays the next page of the data file.
	Displays the last page of the data file.
	Prints the specified pages of the data file.
	Adjusts the magnification of the page so that its entire width is visible in the preview.
	Adjusts the magnification of the page so that the entire page is visible in the preview.
	Sets the printing orientation of the page to portrait (8.5" wide by 11" tall).
	Sets the printing orientation of the page to landscape (11" wide by 8.5" tall).
	Brings up this Help topic.
	Closes the Print Preview window.
	Specifies the pages of the data file that will be printed.
	

The status bar at the bottom of the Print Preview window indicates the page that is currently being viewed (i.e., Page 1 of 4). It also gives directions for zooming and panning (zoom in by clicking the left mouse button, zoom out by holding down the shift key while clicking the left mouse button, pan by moving the mouse while holding down the right mouse button).

# 7. Utilities

---

CardConvert is a utility that is used to quickly read and convert binary datalogger data that is retrieved from a compact flash, microSD, or PCMCIA card. The binary data can also be a file already saved to the user's computer. The binary data is converted to a TOA5 format and saved on the user's computer.

The Device Configuration Utility, or DevConfig, is a utility for setting up datalogger and communication peripherals for use in a datalogger network.

File Format Convert is an application that is used to convert data files from one format to another.

The utilities in PC400 are opened by going to the Window's Start menu. CardConvert and DevConfig can be opened from buttons on PC400's toolbar.

## 7.1 CardConvert

CardConvert is a utility that is used to quickly read and convert binary datalogger data that is retrieved from a compact flash, microSD, or PCMCIA card. The converted data is saved on the user's PC.

### 7.1.1 Input/Output File Settings

The file settings are used to specify the directory where the binary data is stored, and the directory in which the converted file(s) should be saved.

Press the **Select Card Drive** button to bring up dialog box that helps you browse for the drive assigned to the card reader. Note that you can also select a directory on your hard drive in which binary data files have been copied. When a card drive or directory is selected, any binary files found with a \*.dat extension will be displayed in the Source Filename column in CardConvert.

By default, the converted data files will be saved to the same drive or directory as the source files. To change the destination, press the **Change Output Dir** button. Once again you will be provided with a dialog box that helps you to browse for the desired drive or directory. When the drive or directory is selected, the path and the filename that will be used for the converted files will show up in the Destination Filename column.

The default filename for a converted file is comprised of the table name in the datalogger program, along with a prefix that reflects the file format, and a \*.dat extension. For instance, the default name for a table called MyData stored in TOA5 format would be TOA5\_MyData.dat.

The destination directory or filename for a converted file can be changed on an individual file basis. Click on the row for the file that you wish to change. It will be highlighted. Select **Options > Change Output File** from the CardConvert menu, and browse for or type in a new path and/or filename. You can apply a directory path change to all files by selecting **Options > Apply Directory to All**.

You do not have to convert all files that are found in the selected directory. Select one or more files for conversion by selecting or clearing the check box beside the individual file name. If a box is checked the file will be converted; if a box is cleared the file will not be converted. To quickly select or clear all check boxes, choose **Options > Check All or Clear Check All** from the CardConvert menu.

The list of files displayed for a particular drive or directory can be updated by selecting **Options > Rebuild File Lists** from the menu. Any new files that have been stored since you last selected the drive (or since the last rebuild), will be added to the list.

Tip: Right-click within the file list to display a shortcut menu containing the items on the Options menu.

## 7.1.2 Destination File Options

The Destination File Options determine whether the data will be stored on the PC in ASCII or binary format, how filemarks will be processed, and what should happen when existing files with the same name are found.

### 7.1.2.1 File Format

The File Format is used to specify the format in which the data file should be saved. Select the desired option from the list box:

- ASCII Table Data (TOA5) – Data is stored in an ASCII comma separated format. Header information for each of the data values is included, along with field names and units of measure if they are available.
- Binary Table Data (TOB1) – Data is stored in a binary format. Though this format saves disk storage space, it must be converted before it is usable in other programs.
- Array Compatible CSV – Data is stored in a user-defined comma separated format. This option can be used to produce output files from table data dataloggers that are similar to those created by mixed array dataloggers. When this option is chosen, the **Array CSV Options** button becomes available, so that you can customize the data string for the CSV file.

If an array ID is desired, select the **Include Array ID** check box and enter a value into the field. The value can range from 1 to 1023. The array ID will be the first value in the array of data.

Select the appropriate timestamp options for the type of timestamp to write to the file. Each time element will be output as a separate data value in the array and the data values will be separated by a comma. Selecting **Year** will output the year represented by four digits, YYYY (e.g., 2006). The **Day** will be represented as a Julian Day. The **Hour/Minutes** will be represented by four digits (hhmm). When **Midnight is 2400** is selected, the timestamp will reflect midnight as the current date with 2400 for the Hour/Minutes. Otherwise, the timestamp will reflect midnight as the next day's date, with the Hours/Minutes as 0000.

The **Max and Min Timestamp Options** is used to determine the type of timestamp that will be used for Maximum and Minimum outputs that include a timestamp along with the value. You can choose to output No Timestamp, a timestamp that includes Hours/Minutes/Seconds (produces two values, hhmm and seconds), a timestamp that includes Hours/Minutes only, or a timestamp that includes Seconds only.

- CSIXML – Data is stored in XML format with Campbell Scientific defined elements and attributes. For more information, refer to Campbell Scientific File Formats in the LoggerNet manual.

The file format is reflected in the default filename by the prefix of TOA5, TOB1, CSV, or CSIXML added to the table name.

## 7.1.2.2 File Processing

**Use Filemarks** – CRBasic dataloggers have a FileMark instruction that allows you to store a filemark along with the data. These filemarks are ignored by the LoggerNet or PC400 data collection process. However, in CardConvert you can convert the file with the **Use Filemarks** option selected, and the file will be stored as multiple files, based upon the filemarks. Each file created will be given a numeric suffix prior to the \*.dat extension. The first file is stored with a \_1 at the end of the root file name (e.g., TOA5\_Mytable\_1.dat). The number is incremented by one with each new file saved. If a file with the same name is found, the number will be incremented to the next available number.

**Use Removemarks** – When a compact flash card is removed from a CR1000 or CR3000 datalogger, a special mark is inserted in the last record. The Removemark is similar in nature to the Filemark. In CardConvert, you can split a file into multiple files, separated at the Removemarks, by converting the file with the **Use Removemarks** option selected. As with the **Use Filemarks** option, the first file stored uses a \_1 at the end of the root file name and the number is incremented by one with each new file saved.

**Use Filemarks** and **Use Removemarks** can be selected at the same time, to create a new file from the data table any time either of the marks is encountered.

**Use Time** – This option is used to store the converted data into files based on the timestamp of the data. When the **Use Time** check box is selected, the **Time Settings** button becomes available. This button opens a dialog box that is used to set a **Start Date** and **Time**, along with an **Interval**, which are used to determine the time frame for the data that goes into each file. Note that the **Start Date** and **Time** are not used to specify the actual time and date to begin processing the file; rather, they are used as a reference for the file interval. Processing always starts at the beginning of the file.

When **Use Filemarks**, **Use Removemarks**, or **Use Time** is selected, the **Create New Filenames** option is disabled. New file names will always be created.

**Convert Only New Data** – When this option is selected, only data that has been collected since CardConvert's last conversion of the specified file(s) will be converted. The first time CardConvert is used on a file, all data will be converted. On subsequent conversions, only new data will be converted. However, if CardConvert cannot tell what data is new (i.e. if data on the card has wrapped since the last conversion), all data will be converted. This option can be used with **Append to Last File** to create a continuous file with no repetition of data.

### 7.1.2.3 File Naming

**Time/Date Filenames** – When this option is selected, the date and time of the last record of data in the file will be appended to the end of the base file name. The suffix includes a four digit year, a two digit month, a two digit day of month, and a four digit hour/minute. When this option is selected, **Use Day of Year** becomes available. If this option is selected, the Julian day (day of year) will be used for the suffix instead of the year/month/day/hour/minute suffix.

**Create New Filenames** – When the **Create New Filenames** option is selected, CardConvert will add a \_01 to the filename, if a file of the same name is found (e.g., TOA5\_Mydata\_01.dat). If a \*\_01.dat file is found, the file will be named with a \_02 suffix. If the **Create New Filenames** check box is cleared and a file with the same name is found, you will be offered the option to Overwrite the existing file or Cancel the conversion.

The **Create New Filenames** option is disabled when the **Use Filemarks**, **Use Removemarks**, or **Use Time** option is enabled.

**Append to Last File** – When this option is selected, converted data will be appended to the end of the destination file. If the destination file does not exist when a conversion is done, a new file will be created. On subsequent conversions, converted data will be appended to the end of that file. If the header of the new data does not match that of the data in the destination file, an error will

be generated. This option is most useful with the **Convert Only New Data** option to create a continuous file with no repetition of data.

### 7.1.2.4 TOA5/TOB1 Format

These two options are available when the ASCII Table Data (TOA5) or the Binary Table Data (TOB1) output option is selected.

- **Store Record Number** – By default, the record number for each row of data is stored in the data file. This record number can be omitted from the converted file by clearing the **Store Record Number** check box.
- **Store Time Stamp** – The time stamp can be omitted from the file by clearing the **Store Time Stamp** check box.

## 7.1.3 Converting the File

Once the File and Conversion settings are selected, press the **Start Conversion** button. CardConvert will begin processing the file. When the file is being processed, the estimated number of records and a percentage of the conversion completed will be displayed at the bottom edge of the window. Note that the values reflect an *estimate* of the amount of data in a table. If the table is set to a fixed size, CardConvert returns a fairly close estimate. However, if the table is set to auto-allocate, CardConvert essentially returns an estimate that reflects the maximum number of records that can be stored based on card size (even if the table is not completely full). Because of this, you may see the progress reported as something less than 100% when the conversion is complete.

If a conversion is in progress and you wish to stop it, press the **Cancel Conversion** button.

After file conversion is complete, summary information is provided in the field below the file list. The summary provides a listing of the new files that were created, and the total number of records converted for each table (if filemarks are being processed for a table, the number of records returned is the cumulative number of records for all files).

### 7.1.3.1 Repairing/Converting Corrupted Files

If you attempt to convert a file and receive a message that the input file contained no data, you may want to consider using the Repair File option. You may also want to consider using the Repair File option if you think there is additional data on the card that is not being converted and included in the output file. With either case, it is possible that data on the card has become corrupted. The Repair File Option will attempt to scan the card for good frames of data and output that data to a new binary file.



In some instances, data on a card can become corrupted. Corruption can occur if the card is subjected to electrostatic discharge or if it is removed when data is being written to the card (e.g., the card is removed without pressing the button to stop data storage to the card). This corruption can be at the beginning of the data file or anywhere within the stored data. Using the standard conversion option, CardConvert will stop if it encounters a corrupted frame of data because it assumes it has come to the end of the data file. If corrupted frames of data are found at the beginning of the file, CardConvert will display a message indicating that no data could be found on the card. If corrupted frames of data are found within the data file, you may get some, but not all, of the data that you expect in the converted file.

CardConvert offers a repair option, which will attempt to scan the card for good frames of data and output that data to a new binary file (the original file is unchanged). To start the repair of a file, highlight the suspected corrupt file in the list of Source Filenames and right-click to display a floating menu. Select the **Repair File** option from the list. The repair process will create a new TOB3 file (the default name is Repair\_existingfilename), which can then be converted to an ASCII file using the standard CardConvert process.

When CardConvert comes to what it believes is the end of the data file during the repair process (the end of valid frames), it will stop and display a message. The message prompts the user either to continue searching the file for more good data frames or to stop the repair process.

CardConvert displays the last time stamp for data in the repaired file. If you think there should be additional data on the card, you can continue to run the repair process. If it appears that all the data has been stored to the new file, you can stop. The option to continue processing the file allows you to recover all good data on a card with more than one corrupted frame.

Note that CardConvert can repair only TOB2 or TOB3 files. TOB1 files cannot be repaired.

**NOTE:** The Repair File option should be used only if a standard conversion cannot be done.

## 7.1.4 Viewing a Converted File

Converted data files can be reviewed using the View Pro file viewing application. View Pro can be launched by pressing the **View Files** button. If a file is highlighted in the list of files, that file will be displayed when View Pro is opened. Otherwise you can select the file to view from View Pro's **File > Open** menu.

## 7.1.5 Running CardConvert From a Command Line

In order to run CardConvert from a command line without user interaction, you will first need to create a CCF file that contains the CardConvert settings to be used when running from a command line. To create the CCF file, open CardConvert and select the desired source directory

(Select Card Drive), destination directory (Change Output Dir), and Destination File Options. When CardConvert is closed, it will produce a file named "lastrun.ccf" that contains the designated settings. The file will be written to the C:\Campbellsci\CardConvert directory. You should rename this file as it will be overwritten the next time that CardConvert is closed.

When running CardConvert from a command line, you can designate the CCF file using the command line option runfile. For example,

```
"C:\Program Files (x86)\Campbellsci\CardConvert\CardConvert.exe"  
runfile="C:\Campbellsci\CardConvert\myfile.ccf"
```

The above command line will run CardConvert using the settings contained in myfile.ccf.

**NOTE:** The path to the CCF file should be specified. It will not default to the CardConvert working directory.

If there are no problems or questions encountered, CardConvert will start, convert the file(s), and then exit with no user interaction. However, if any problems or questions are encountered, CardConvert will display a dialog box as usual and then wait for a user response. Therefore, this command line option allows some automation of CardConvert but does not allow for completely unattended automation. To minimize the user interaction required, the Destination File Option "Create New Filenames" should be used as this prevents CardConvert from asking whether a file should be overwritten.

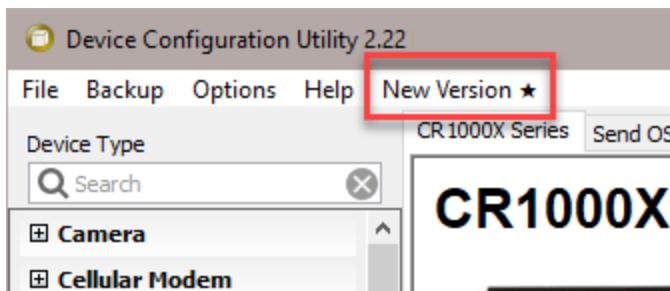
## 7.2 Device Configuration Utility

The Device Configuration Utility (DevConfig) is used to set up dataloggers and intelligent peripherals before those devices are deployed in the field and before the devices are added to networks in Campbell Scientific datalogger support software such as LoggerNet or PC400. Some key features of DevConfig include:

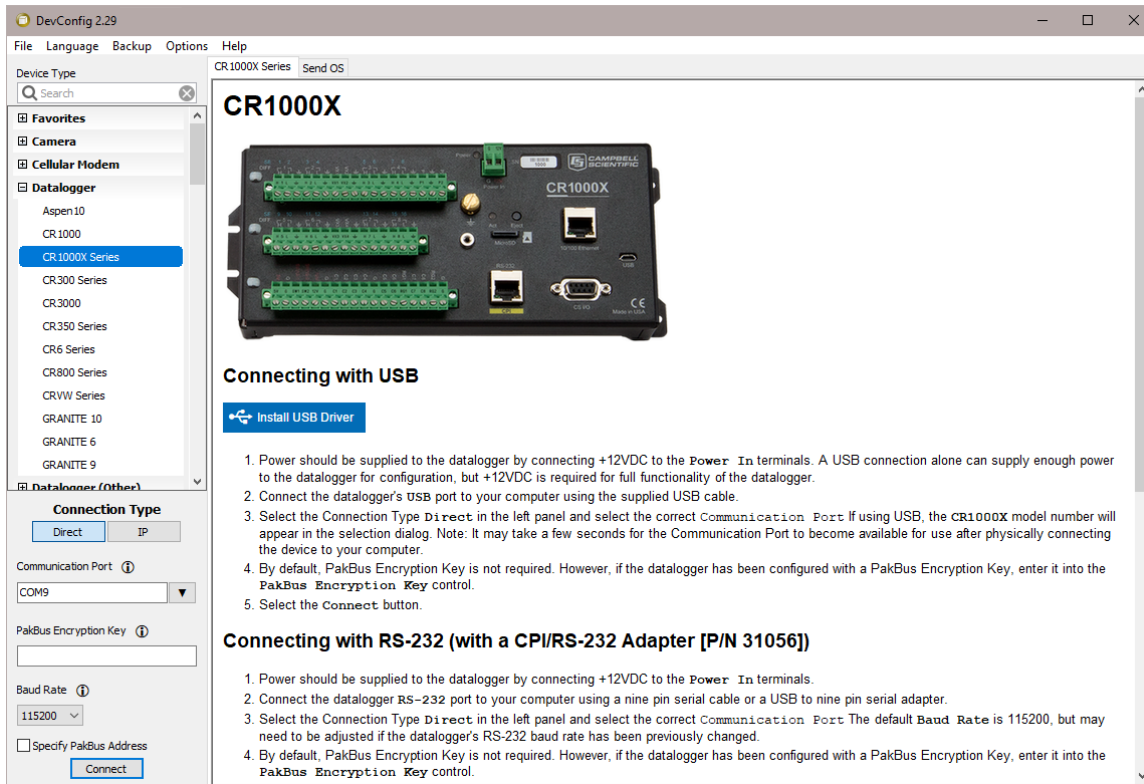
- To keep the process as simple as possible, DevConfig supports only serial and IP connections between the PC and devices.
- DevConfig cannot only send operating systems to supported device types, but can also set datalogger clocks and send program files to dataloggers.
- DevConfig allows you to determine operating system types and versions, which can be very useful in classic dataloggers, such as the CR10X, where the operating system version in the datalogger is not known.
- DevConfig provides a reporting facility where a summary of the current configuration of a device can be shown on the screen and printed. This configuration can also be saved to a

file and used to restore the settings in the same or a replacement device.

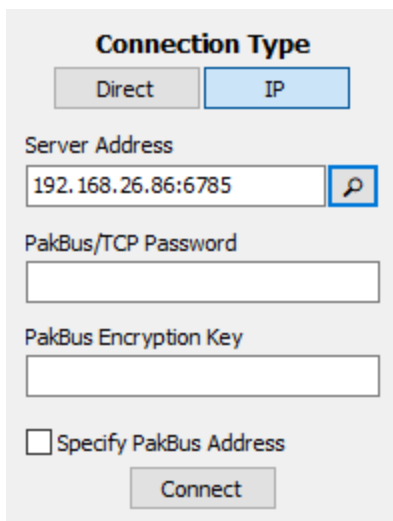
- Some devices may not support the configuration protocol in DevConfig, but do allow configurations to be edited through the terminal emulation screen.
- Help for DevConfig is shown as prompts and explanations on its main screen. Help for the appropriate settings for a particular device can also be found in the user's manual for that device.
- When DevConfig is opened, it automatically checks for a newer version. If there is a new version, it is indicated on the toolbar. Select **New Version** to see the available update. Press the new version to see the changes in the new version. Then press **Update** to download and run the update. Updates to DevConfig are also available from Campbell Scientific's website. These may be installed over the top of older versions.



## 7.2.1 Main DevConfig Screen



The DevConfig window is divided into two main sections: the device selection panel on the left side and tabs on the right side. After choosing a device on the left, you will then have a list of the serial ports (COM1, COM2, etc.) installed on your PC.



If the device supports IP communication, the **IP** button will be enabled. In order to communicate via IP, click **IP** under **Connection Type** and enter the IP address or domain name for the device in

the **Server Address** field. For some devices, you may be able to click on the **Browse** button to the right of the **Server Address** control to bring up a dialog that searches your local area network for any available devices. If the device has a PakBus/TCP Password, you will need to enter it in the **PakBus/TCP Password** field.

If the device is using PakBus encryption, you will need to enter the key in the **PakBus Encryption Key** field.

You'll be offered a choice of baud rates only if the device supports more than one baud rate in its configuration protocol. The page for each device presents instructions about how to set up the device to communicate with DevConfig. Different device types will offer one or more tabs on the right.

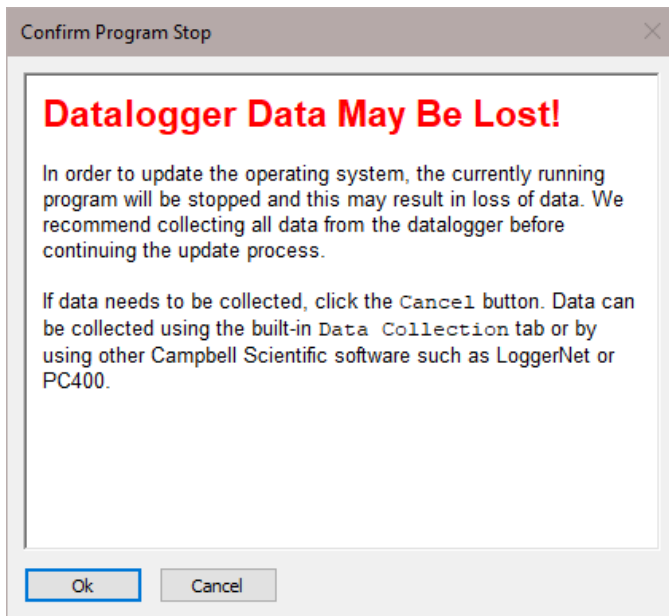
When the user presses the **Connect** button, the device type, serial port, and baud rate selector controls become disabled and, if DevConfig is able to connect to the device, the button will change from "Connect" to "Disconnect". The tabs on the right side of the window will be replaced with tabs that represent the various operations that are available for that device in a connected state. These operations can vary from device to device.

## 7.2.2 Downloading an Operating System

**WARNING:** Downloading an operating system may result in data loss. We recommend always collecting data before sending an operating system.

### 7.2.2.1 Manage OS Tab

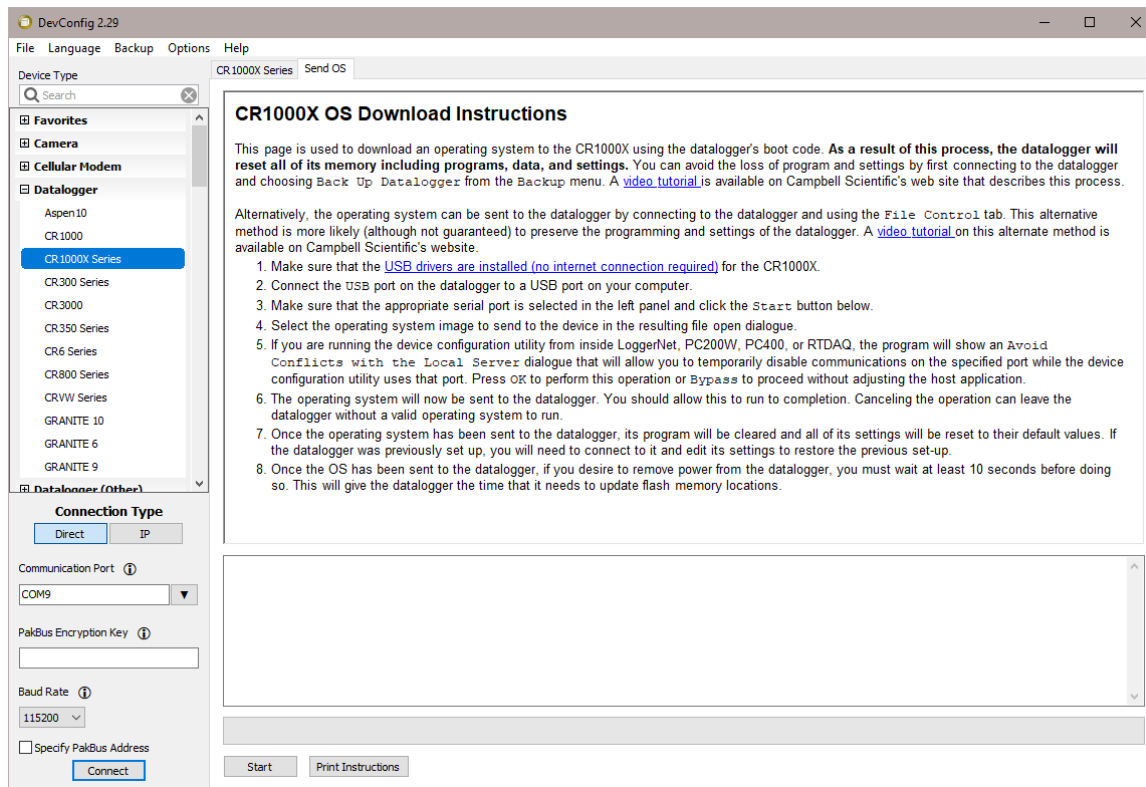
Our new data loggers have a **Manage OS** tab in DevConfig. By default, when connecting to a data logger, DevConfig checks for an updated operating system. If one is found, DevConfig automatically navigates to the **Manage OS** tab. This tab displays the changes in the new version of the operating system. Press **Update** to install the new version. A warning will appear about potential data loss. Click **OK** to let DevConfig continue with the OS update; click **Cancel** to stop the OS update. When sending an operating system from the **Manage OS** tab, an attempt is made to preserve data and settings. However, there is still a possibility that settings and data may be lost. Therefore, you will see the warning shown below. Before sending an operating system, we recommend you collect your data and back up your data logger (see [Backing up and Restoring a Datalogger](#) (p. 168)).



### 7.2.2.2 Send OS Tab

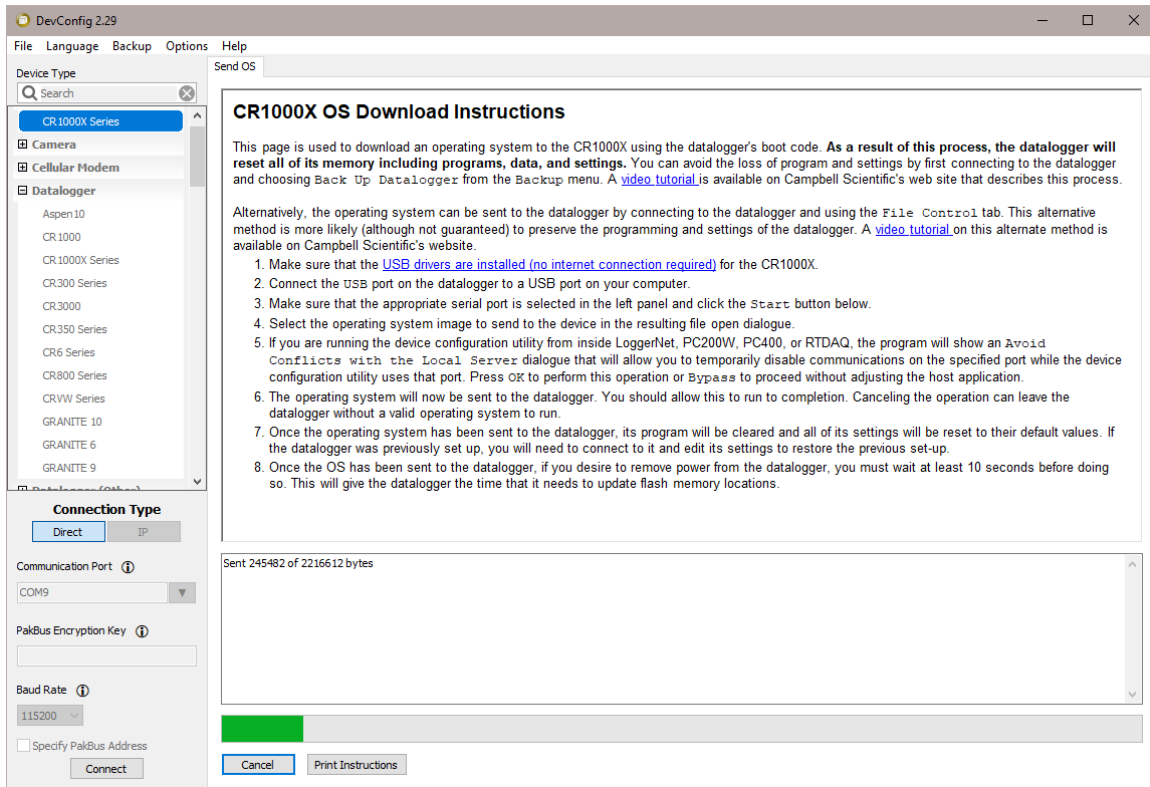
DevConfig can send operating systems from the **Send OS** tab to all Campbell Scientific devices with flash replaceable operating systems. When sending an operating system to a data logger from the **Send OS** tab, data and programs will be lost, and all settings will be returned to their default values. Therefore, before sending an operating system from the **Send OS** tab, collect your data and back up your data logger (see [Backing up and Restoring a Datalogger](#) (p. 168)).

An example for the CR1000X is shown below:



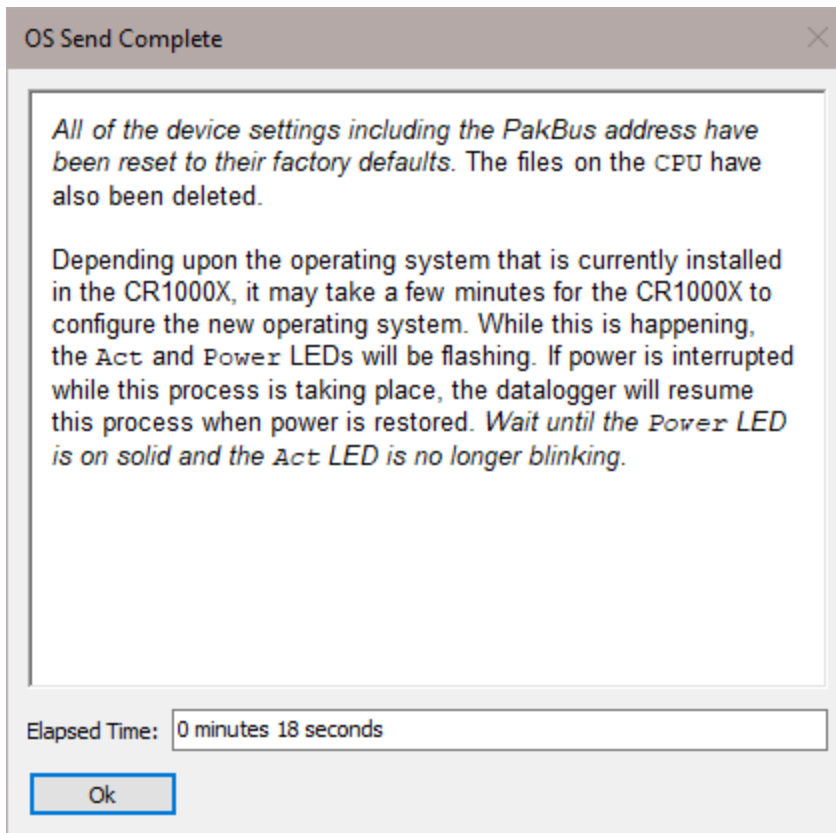
The text at right describes any interface devices or cabling required to connect the PC to the device. Screens for other devices vary only in the text on the right side. This screen differs from other screens that are available in DevConfig in that it can be accessed from either a connected or disconnected state.

When you click the **Start** button, DevConfig offers a file open dialog box to prompt you for the operating system file (usually a \*.obj file). You may be required to cycle power the device or press a special "program" button. When the device issues the appropriate prompts, DevConfig starts to send the operating system:



When the operating system has been sent to the device, a message dialog will appear similar to the one shown below:

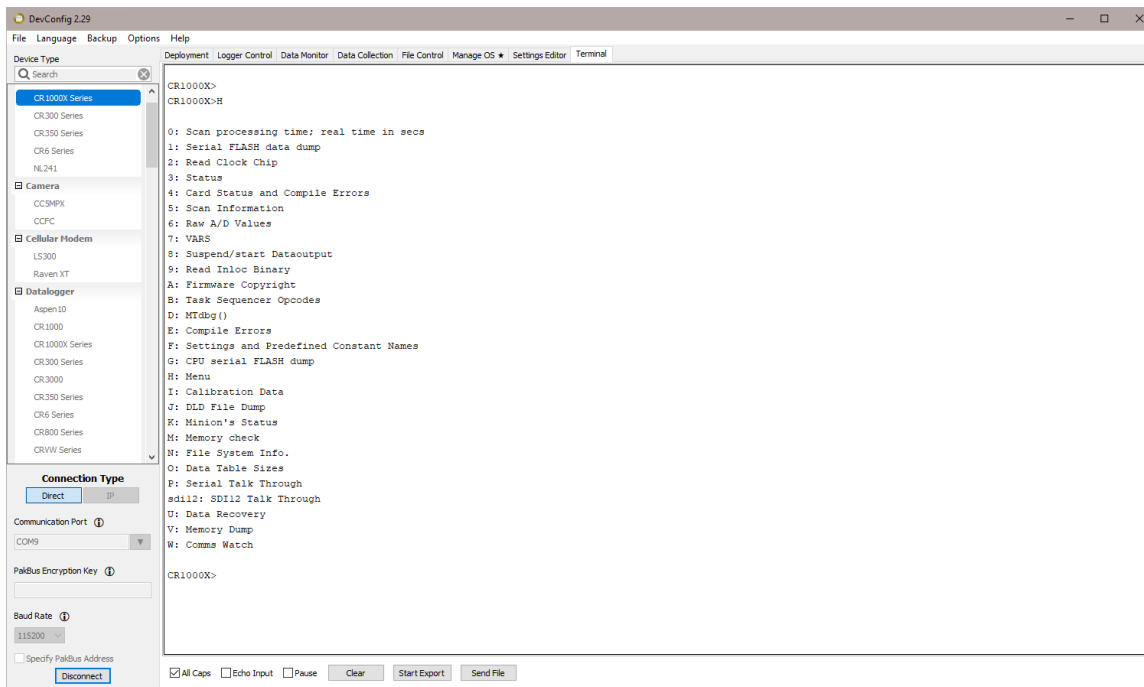




The information in the dialog helps to corroborate the signature of the operating system sent. For devices such as the CR10X (especially those with extended memory) that can take a long time to reset following an OS download, text warns you against interrupting the memory test.

## 7.2.3 Terminal Tab

The **Terminal** tab will be available when the application is connected to any device type that can be communicated with in a remote terminal mode. The **Terminal** tab offers a terminal emulator that can be useful in accessing settings or status information that are not exposed in other windows. For example, classic dataloggers with PakBus operating systems that are configured as routers contain routing tables that list the other PakBus nodes that are known to that datalogger. This routing table is only available through the \*D17 mode (see \*D descriptions in the datalogger's operators' manuals) using the keyboard/display or a terminal emulator. Another example is that the status table in mixed-array dataloggers (\*B) can also be accessed via an "S" command in terminal mode. This status information can provide important data for troubleshooting purposes.

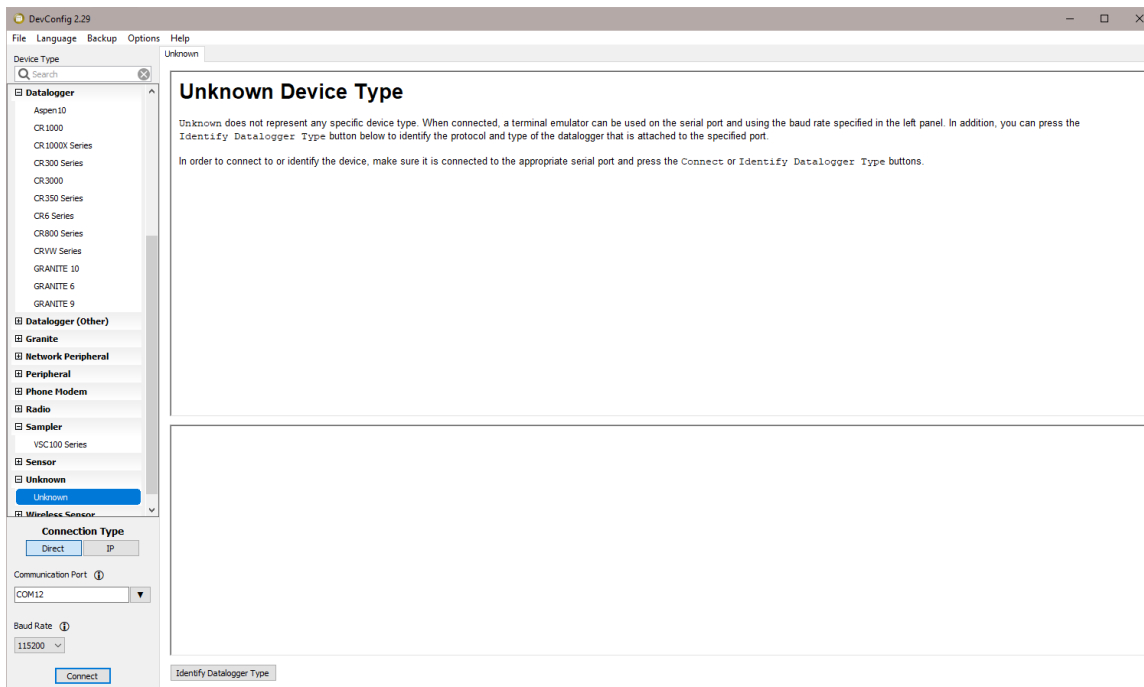


The default for the **Terminal** tab is to only show characters that are returned from the device. However, if the **Echo Input** check box is enabled, the screen will also display the characters actually typed by the user.

The **All Caps** check box controls whether the keyboard input will be forced to upper case before the characters are sent to the device. It will be disabled for some device types that require upper case input.

## 7.2.4 The Unknown Device Type

When the Unknown device type is selected, a panel will be shown in the tab control similar to that shown below:



Clicking **Connect** puts DevConfig into Terminal emulation mode on the Serial Port and at the Baud Rate selected.

When you click on **Identify Datalogger Type**, DevConfig will attempt to identify the type of device that is connected on the specified serial port. It will attempt to communicate using each of the datalogger protocols (mixed-array, table-data, and PakBus) in turn. If it fails to get any answer to any of these attempts, the baud rate will be automatically changed and the various protocols will be attempted again. When DevConfig recognizes the response from the device and the device type is one of the supported types, that device type will automatically be selected.

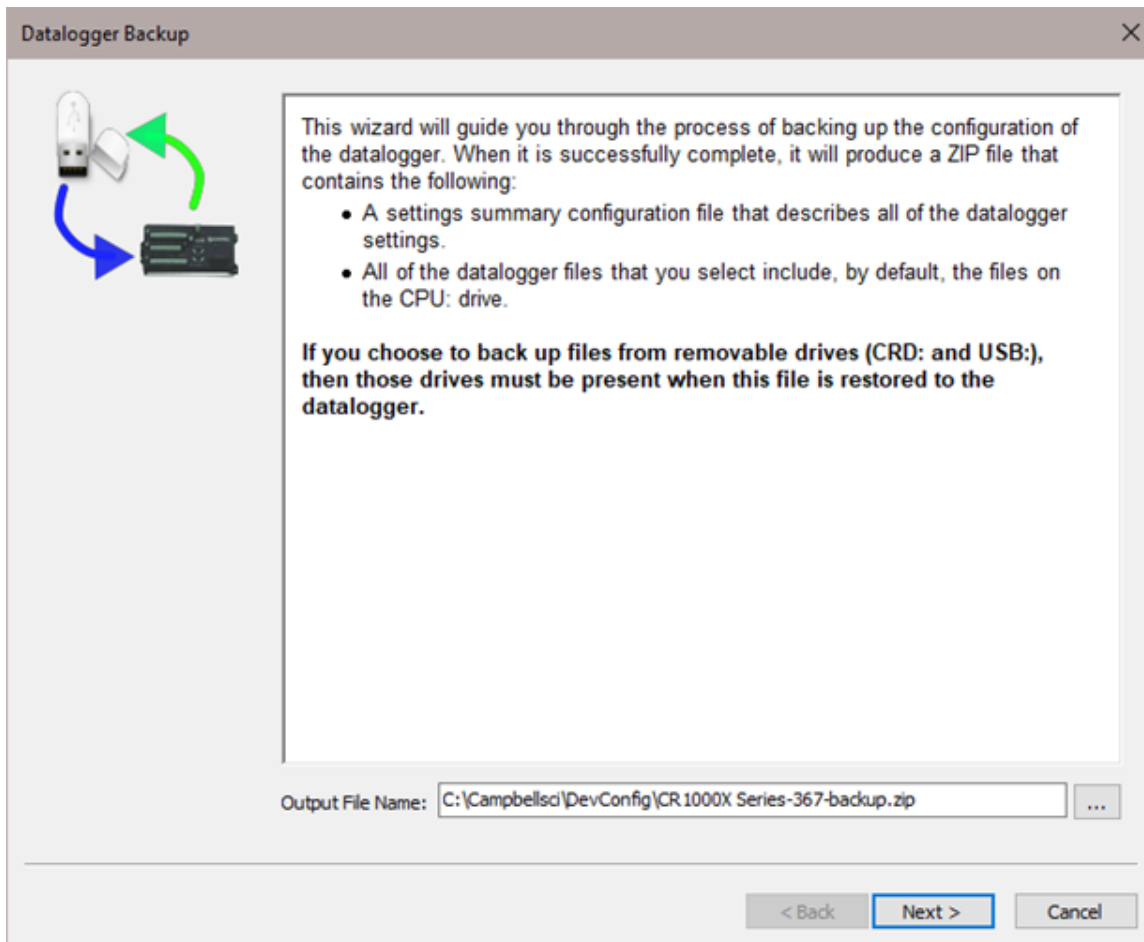
## 7.2.5 Off-line Mode

Many devices in DevConfig have an off-line mode available that allows you to browse the device's settings without actually being connected to a device. You can select a device, and then select **Off-line Mode** from DevConfig's **File** menu. You will be able to see the device's settings and associated help. You can also make changes to the settings and then press **Apply** to bring up the option to **Save** or **Print** the configuration. Saving the configuration will allow you to load it into a device at a later time.

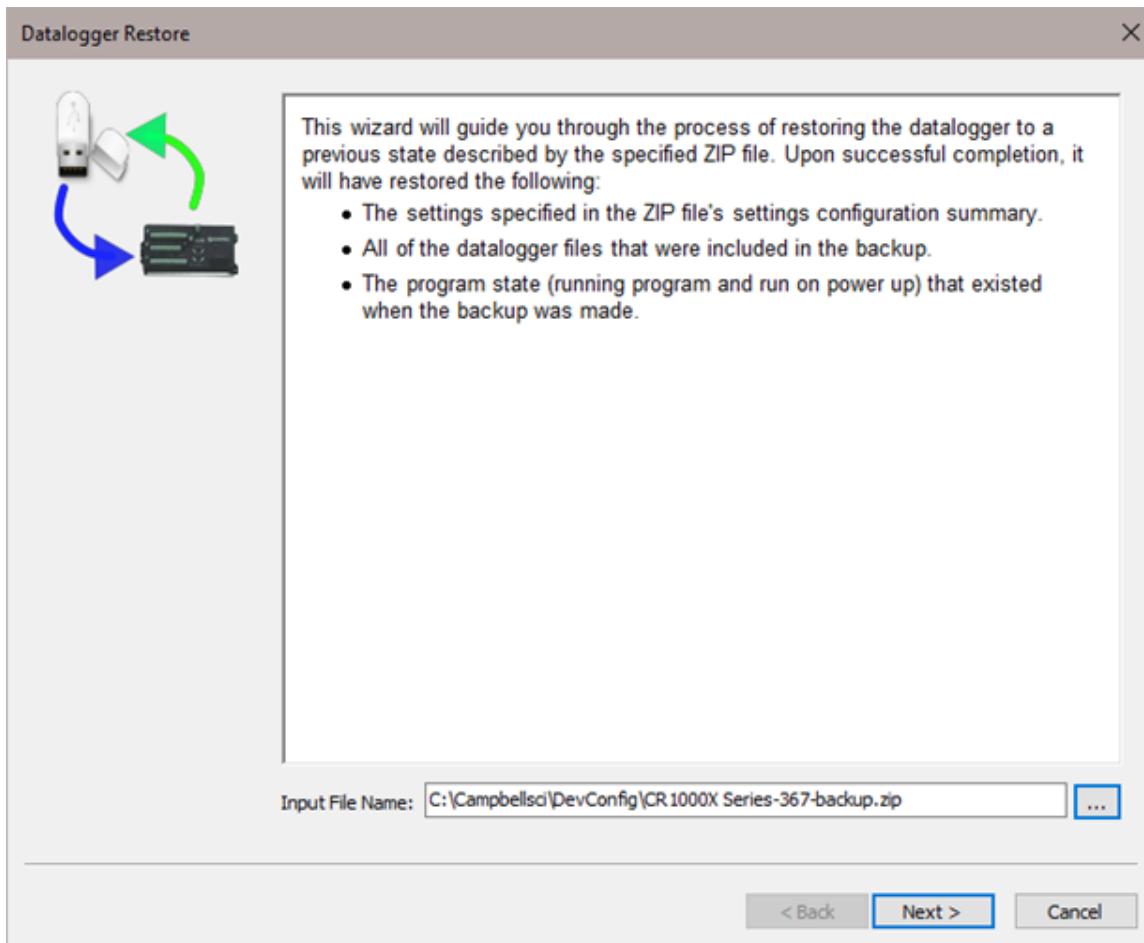
## 7.2.6 Backing up and Restoring a Datalogger

Since all settings will be restored to their default value when a new operating system is sent to a datalogger from DevConfig, it is a good idea to back up the datalogger first. This is done by

selecting **Back Up Datalogger** from the DevConfig **Backup** menu. A wizard will appear to guide you through the backup process.



After downloading the operating system (or any other time you want to restore the datalogger to its state at the time of the backup), select **Restore Datalogger** from the **Backup** menu. A wizard will again appear to guide you through the restoration process.



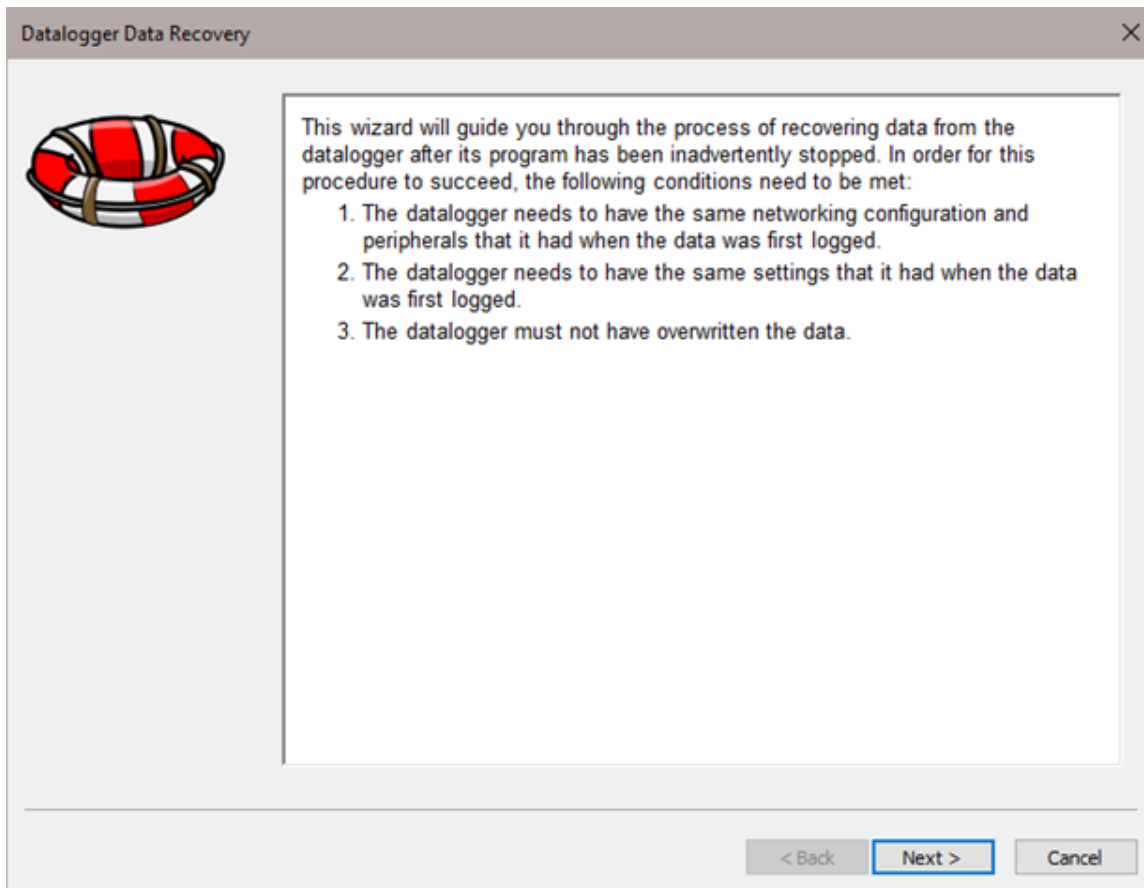
## 7.2.7 Data Recovery

DevConfig can be used to recover data from a datalogger after its program has been inadvertently stopped.

In order for the data recovery procedure to succeed, the following conditions need to be met:

- The datalogger needs to have the same networking configuration and peripherals that it had when the data was logged.
- The datalogger needs to have the same settings that it had when the data was logged.
- The datalogger must not have overwritten the data.

To recover data from a datalogger in this condition, select Data **Recovery** from the **Backup** menu. A wizard will appear to guide you through the recovery process.



## 7.3 File Format Convert

File Format Convert is not available from the LoggerNet toolbar. It can be opened from the Window's Start menu.

File Format Convert is used to convert data files from one format to another. It can also perform the following functions:

- Break large files into smaller files (also known as baling).
- Check for missing records by checking the record number and or timestamp.
- Bale based on time
- Bale based on File Marks and Remove Marks (TOB2, TOB3 files)
- Bale files when missing records are discovered.
- Fill in missing records with 'Null' or empty records.

More than one of the above functions can be performed in one pass.

In general files can be converted from:

- TOA5
- TOACI1
- TOB2
- TOB3
- TOB1
- CSIXML

To:

- TOA5
- TOACI1
- TOB1
- CSIXML
- CSV

**NOTE:** File Format Convert cannot produce TOB2 or TOB3 files, and it cannot read CSV files.

**NOTE:** Some file headers have less information than other formats. If you convert from a file with more information in the header to one with less, information will be lost. If you convert from a format with less information, some fields will be left blank.

**NOTE:** Some formats (e.g. TOB1) store string in fixed length fields and have headers that specify how big that field is. Other formats use variable length strings. If you convert from a format that uses variable lengths to a fixed length, the length is assigned to 64. If the string is longer than this, it is truncated.

### Converting a File

Press the **Open** button to browse to a file to be converted. After a file is selected, press the **Options** button and set up the options for the conversion. Then press the **Convert/Check** button to convert the file.

If a conversion is in progress and you wish to stop it, press the **Abort** button.

### Log File

If the **Write Log File** check box is checked, a "Log.Txt" file will be created in the same directory as the source data file. The log.txt file will be overwritten if it exists.

## 7.3.1 Options

### Check

**Record Numbers** – Checks for missing record numbers.

**Timestamps** – Checks for missing timestamps based on entered interval.

Both can be checked in the same pass. If a file is written, other options are available.

Files can be baled if missing records are found. See the Bale based on information below. When checking timestamps, "null" records can be written to "fill" missing records. See Missing Records information below.

### File

Check **Write File** to cause an output file to be created. The file will be created in the same directory as the source file. The base name will be the same as the source name with the new format prepended. For example, test.dat becomes TOA5\_test.dat. Use the drop-down list to select the format of the new file.

For all output options except TOAC11, the **Browse** button to the right of the field becomes available and can be pressed to set additional file output options.

### File Naming

**Date Time Filename** – When this option is selected, the date and time of the first record of data in the file will be appended to the end of the base file name. The suffix includes a four digit year, a two digit month, a two digit day of month, and a four digit hour/minute. When this option is selected, **Use Day of Year** becomes available. If this option is selected, the Julian day (day of year) will be used in the suffix instead of the month and day of the month.

**Create New Filenames** – When the Create New Filenames option is selected, File Format Convert will add a \_1 to the filename, if a file of the same name is found (e.g., TOA5\_Mydata\_1.dat). If a \*\_1.dat file is found, the file will be named with a \_2 suffix. If the Create New Filenames check box is cleared and a file with the same name is found, you will be offered the option to Overwrite the existing file or Cancel the conversion. (Note that if any of the baling options are selected, new filenames will automatically be created as described below.)

### Missing Records

If Timestamps are checked (see Check section above), then missing records can be filled. These will be Null records. A timestamp and record number will be added. Values will be "NAN". Strings will be empty, etc.



- **Just Log** – Missing records are not filled.
- **Fill Null Records** – All missing records are filled.
- **Prompt** – Shows what records are missing and lets you choose to fill or not. If you have big gaps (e.g. bad timestamp), filling can be quite slow.

### Bale based on

This allows a file to be broken into smaller files. A new file is started based on:

- **Time** – A new file is created based on interval.
- **Remove Marks** – A new file is created when a remove mark is found in the data file (TOB3 only).
- **File Marks** – A new file is created when a file mark is found in the data files (TOB3 and TOB2 only).
- **Discontinuity** – A new file is created when missing records are encountered (see Check section above). The How Many? can be used so that small gaps do not start a new file. The file will be baled only if the entered number of records (or more) are missing.

### Bale Info

Use to specify the **Start Time** and **Interval** and start time for baling based on time.

## 7.4 Split

Split is a tool that is used to post-process collected data from either mixed-array or table-based dataloggers. Split can create reports by filtering data based on time or conditions. It can generate statistics, perform calculations, reformat files, check for data quality (limit testing), and generate tables with report and column headings. It can also handle the time synchronization necessary to merge up to eight data files. For information on using Split, see [Split](#) (p. 231).

# Appendix A. Glossary

---

## A

---

### Advise

See Data Advise

### Analog Channel

A terminal on the datalogger's wiring panel where leads for analog signals are connected. The analog channels are designated single-ended (SE) or differential (DIFF) on the wiring panel. Many sensors, such as thermistor temperature probes and wind vanes, output analog signals.

### Array-based Datalogger

See Mixed-array Datalogger

### ASCII File

A computer file containing letters, numbers, and other characters using the ASCII character encoding.

### Asynchronous

The transmission of data between a transmitting and a receiving device occurs as a series of zeros and ones. For the data to be "read" correctly, the receiving device must begin reading at the proper point in the series. In asynchronous communications, this coordination is accomplished by having each character surrounded by one or more start and stop bits that designate the beginning and ending points of the information (see Synchronous). The transfer of information is not otherwise coordinated between the sender and receiver.

## **B**

---

### **Batch Files**

An ASCII text file that contains one or more DOS commands or executable file commands. When the batch file is run, the commands in the file are executed sequentially.

### **Battery**

This entry in the status table returns the datalogger battery voltage.

### **Baud**

The rate at which a communication signal travels between two devices.

### **Binary File**

A file based on software defined formatting. A binary file can only be interpreted by the software programmed to decode the formatting. This format is used for more efficient data storage than is provided by ASCII.

### **BMP (Block Mode Protocol)**

The communications protocol used by the server to communicate with table-based dataloggers and RF modems.

### **Broadcast**

Part of the radio (RF) technique of polling remote radio modem datalogger sites. A single modem sends a message (broadcast) that all affected remotes hear and respond to.

## **C**

---

### **Call-back**

When a datalogger is programmed for Call-back, it will automatically call the host computer when a specified condition is met. The computer must be set up to look for such an incoming call.

## **Call-back ID Number**

A three-digit number that is used to identify what datalogger has called the host computer. (Not available for Table-based dataloggers.)

## **Cancel**

Choosing Cancel from a dialog box will typically ignore any changes made and close the box.

## **Carrier**

An electrical signal used to convey data or other information. For example, radio and phone modems use carrier signals. Phone modems attempt to detect carrier when the call is placed. The red LED on the RF95T lights when the modem detects a carrier.

## **Child Node**

See Node. A node that is accessed through another device (parent node). For example a remote radio frequency (RF) site is accessed through and a child of the base RF232T. All nodes are child nodes of the PC.

## **Client**

A software application designed to connect to a server. Usually provides some type of user interface or data acquisition. Email programs running on individual PCs are typically client applications that connect to an email server program running on a computer at an Internet Service Provider to receive and send email messages.

## **Coaxial cable**

Special type of cable with two conductors (center conductor and outer shield conductor). Classified by size, impedance, and loss characteristics. Used to connect MD9 modems and to connect radios to antennas.

## **Collection**

see Data Collection

## COM Port

A computer's serial communications port. Cables and other interface devices are connected between the computer's COM port and the datalogger.

## Communication Server

The software (typically packaged as a DLL) that provides the communications functions within other software such as PC400 or LoggerNet.

## Control Port

Dataloggers have digital output ports that can be used to switch power to sensors such as the HMP35C relative humidity circuit or to control relays. These digital outputs are called Control Ports and are labeled C1, C2, etc., on the wiring panel. Control ports on some dataloggers can also be used as inputs to sense the digital (high or low) state of a signal, monitor pulse signals, control Synchronous Devices for Measurement (SDM), or used as data input/output connections for SDI-12 sensors.

## CoraScript

A command line interpreter client to the LoggerNet server that allows the user access to many of the capabilities of the LoggerNet server using direct commands or programmed script files.

## CRBasic

The programming language used for CR1000X-series, CR6-series, CR300-series, CR350, CR1000, CR3000, CR800-series, CR200-series, GRANITE 6, GRANITE 9, GRANITE 10, CR5000, CR9000, and CR9000X dataloggers. Short Cut or the CRBasic Editor are used to create program files for these dataloggers.

## CRBasic Datalogger

A CR1000X-series, CR6-series, CR350, CR300-series, CR1000, CR3000, CR800-series, CR200-series, GRANITE 6, GRANITE 9, GRANITE 10, CR5000, CR9000, or CR9000X datalogger. Sometimes referred to as "CRx000 dataloggers."

## **CRx000 Datalogger**

Generally, a CR1000X-series, CR1000, CR350, CR300-series, CR6-series, CR800-series, CR3000, CR200/205, CR5000, or CR9000 datalogger. More correctly referred to as "CRBasic dataloggers."

## **D**

---

### **Data Advise (Datalogger)**

A mutual agreement between the communication server and the datalogger about which tables are to be collected every time the datalogger is contacted. Based on the dataloggers table definitions.

### **Data Advise (Server)**

An agreement between a client application and the communication server to provide specified data as it is collected by the server.

### **Data Advise Notification**

The packet of data sent by the datalogger based on the Data Advise agreement.

### **Data Cache**

The storage for data collected from the datalogger by the communication server. This data is stored in binary files on the hard disk of the computer where the server is running.

### **Data Collection**

Getting a copy of the data stored in the datalogger and saving it in the communication server's data cache (compare to Data Retrieval).

### **Data Point**

A data value that is sent to Final Storage as the result of an Output Instruction. A group of data points output at the same time makes up a record in a data table

## Data Retrieval

Sending a copy of the data from the communication server's data cache to a file, network, or data display (compare to Data Collection).

## Data Storage Table, Data Table

A portion of the datalogger's Final Storage allocated for a particular output. Each time output for a given data table occurs, a new record is written to the table. The size of the table (in number of records) and when records are written to the data table are determined by the datalogger's Data Table Instruction (P84). The fields (columns) of the table are determined by the Output Processing Instructions that follow the Data Table Instruction.

## DaysFull

A field in the status table that shows the number of days before any of the tables using automatic record allocation are filled.

## DevConfig

Short for "Device Configuration Utility", a software application that provides a graphical user interface to configure settings in dataloggers and communications peripherals. Available in PC400, LoggerNet, and as a stand-alone application from the Campbell Scientific website.

## Differential Analog Input

Some sensors have two signal wires and the measurement is reflected in the voltage difference between them. This type of sensor requires two analog connections. The channels marked DIFF on the datalogger wiring panel are used to connect differential sensors.

## E

---

### Excitation Channel

Sensors utilizing electrical bridge circuits require a precise electrical voltage to be applied. The excitation channels, marked as VX1, VX2, etc., on the datalogger wiring panel, provide this required precision voltage.

## Execution Interval

See Scan Interval.

## Execution Time

The time required to execute an instruction or group of instructions. If the total execution time of a Program Table exceeds the table's Execution Interval, the Program Table will be executed less frequently than programmed. Each time this occurs, a Table Overrun occurs. Table Overruns are considered to be

## F

---

### F1

In most instances, pressing the F1 key will provide context sensitive help for the highlighted object on the screen

### Fault

Message relating to network activity where repeated problems or errors have occurred. Repeated faults usually indicate a failure of some kind.

## Final Storage

Final Storage is an area in the datalogger's memory where data is stored for collection to a PC. When you collect data from the datalogger you are collecting data from a Final Storage table.

## Flag

Memory locations where the program can store a logical high or low value. These locations, called User Flags, are typically used to signal a state to another part of the program.

## G

---

## Ground Connection

Most sensors require one or more ground connections in addition to excitation or signal inputs. Ground connections may serve any of several purposes: a reference for a single-



ended (SE) analog voltage (use analog ground if available); a power return path (do NOT use analog ground for power return); a connection for cable shield wire to help reduce electrical noise (do not use analog ground for shield wires, also known as drain wires)

## H

---

### Hole Collection

The process used by the server to collect data records missing from the data cache but possibly still in the datalogger. If Hole Collection is delayed or disabled, the memory in the datalogger can ring around and overwrite the missing data records resulting in an Uncollectable Hole.

### Holes

When using Data Advise, the communications server always gets the most recent data records, so if there are more records to be returned than can fit in one packet there can be sequences of older data available from the datalogger that have not yet been collected to the data cache. The server tracks and collects these holes only if that option is enabled. This entry in the status table shows the number of data points in missed records for the data storage tables in that station.

### Host Computer

The machine where the communication server software is running.

## I

---

### INI Files

Configuration files that are used to preserve the last known setups or states of a program or device.

### Initialization String

A string of alphanumeric characters that are sent to a device, such as a modem, to prepare that device for communications.

## Input/Output Instructions

Datalogger program instructions used to make measurements or send data automatically to other devices.

## Intermediate Storage

Datalogger memory used to temporarily store values (such as a running total and number of samples for an average calculation), typically to be used for output calculations. The datalogger uses Intermediate Storage to accumulate sensor readings until output.

## L

---

### LDEP

Logger Data Export Protocol, a protocol and client application that provides for data distribution from the communications server to a third party application through a standard TCP/IP socket. Installed with LoggerNet Admin; see the associated PDF file for more information. Requires record-specific acknowledgements for record flow control. See LDMP.

### LDMP

Logger Data Monitoring Protocol, a protocol and client application that provides for data distribution from the communications server to a third party application through a standard TCP/IP socket. Installed with LoggerNet Admin; see the associated PDF file for more information. Requires very simple acknowledgements for record flow control. See LDEP.

### Link

Communications route between two devices, for example the phone link between two phone modems.

### Log Files

Text files that are stored on the computer's hard drive that record activity. They contain information about communications between the communications server and other devices in the datalogger network. Log files are typically used for troubleshooting purposes. LoggerNet has four types of log files: Transaction, Communications Status, Object State, and Low Level I/O. Refer to the Log Files Appendix for information on these log files.

## M

---

### MD9

An MD9, or multi-drop modem, is a communications device that uses twisted pair cable for connection. Typically, the system consists of one MD9 base modem that is attached to the user's computer, with one or more remote modems at the datalogger field site. One remote modem is needed for each datalogger at the field site

### Measurements

Values stored by the datalogger in an Input Location after reading an electronic signal from a sensor and converting the raw signal into meaningful units.

### Modem

From "modulator-demodulator"; a device used to transmit and receive digital data over normally analog communications lines, such as an audio signal on telephone circuits. A modem attached to a computer performs a digital-to-analog conversion of data and transmits them to another modem that performs an analog-to-digital conversion which permits its attached computer to use the data.

## N

---

### Node

Part of the description of a datalogger network. Each node represents a device that the communications server will dial through or communicate with individually. Nodes are organized as a hierarchy with all nodes accessed by the same device (parent node) entered as child nodes. A node can be both a parent and a child node.

## O

---

### Output Interval

The output interval is the interval at which the datalogger writes data to a Final Storage data table.

## Output Processing

Writing to final storage memory a sample or summary statistic of data measurements. Output processing options include sending a sample, average, maximum, minimum, total, or wind vector of data to Final Storage. Each Output Processing data value is kept in a separate location within the datalogger. This allows multiple output processing for each measurement. For example, you can average air temperature over a 60-second interval, a one-hour interval, and a 24-hour interval. See the operator's manual or programming software for output processing options available for each datalogger model.

## P

---

### Packet

A unit of information sent between two BMP or PakBus devices that are communicating. Each packet can contain data, messages, programming, etc. Usually contains addressing and routing information.

### PakBus

A packet-based and packet-switched networking protocol used by newer dataloggers. PakBus allows for robust transmission of commands and data, dynamic routing between PakBus devices, and peer-to-peer communications (such as when one datalogger needs to control another datalogger without involving the PC).

### Parameter

Number or code which helps to specify exactly what a given datalogger instruction is to do.

### Path

The modems, or other devices that make up a link to communicate with a remote site datalogger.

### Polling

Process where a datalogger or other communications device is periodically checked for any packets it needs to send. The server polls dataloggers for most communications links. Some communications devices, such as RF232T radio bases or repeaters can also poll datalogger sites.

## **Polling Interval**

The user-specified interval that determines when to poll a given device.

## **Program Control Instructions**

Datalogger instructions that modify the sequence of execution of other instructions in the datalogger program; also used to set or clear user flags.

## **Program Signature**

A program signature is a unique value calculated by the datalogger based on program structure. Record this signature in a daily output to document when the datalogger program is changed.

## **Pulse Channel**

Some sensors output voltage pulse signals. Such sensors can be connected to Pulse Channels for measurement (labeled as P1, P2, etc., on the datalogger's wiring panel).

## **Q**

---

## **Quiescent Mode**

Often referred to as "sleep mode" – a low power state between program execution intervals.

## **R**

---

## **Real-Time Clock**

All dataloggers have an internal clock. The date and time information from this clock are used in the time stamp for stored data. The datalogger's scan interval and timer are synchronized with the clock. All CRBasic dataloggers have battery backups that maintain the clock even when 12V power is not available.

## **Record**

A group of data values output at the same time to the same data table. Records are written in response to the DataTable declaration in CRBasic dataloggers. The individual fields within

each record are determined by the Output Processing instructions following the instruction that created the data table.

### **Remote Site**

Typically where a datalogger is located at the other end of a communications link. Also can refer to the site where a radio (RF) repeater is located.

### **Repeater**

A radio (RF) site that relays packets of information to a remote site. Used to extend the range of radio transmissions. Most remote datalogger sites with radios can act as repeaters.

### **Retries**

When a transaction or communication between two devices or programs fails, the transaction or communication can often be triggered to repeat until it succeeds.

### **Retrieval**

See Data Retrieval.

### **RF**

Radio Frequency

### **RTMC**

Real Time Monitoring and Control software. A client application to the communications server that displays data from the server's data cache (only) and updates as new data is collected. RTMC is relatively easy to set up, and ships with LoggerNet.

## **S**

---

### **Scan Interval**

The periodic interval on which the datalogger program is run. The scan interval is sometimes referred to as the execution interval. For example, when a scan interval of 60 seconds is set, the datalogger will execute its program table every 60 seconds. Between scans the datalogger enters a sleep (quiescent) mode. This conserves battery power and creates

predictable measurement intervals. The scan interval is synchronized with the datalogger's real-time clock.

## **SDI-12**

SDI-12 stands for Serial Digital Interface at 1200 baud. It is an electrical interface standard and communications protocol that was originally developed by Campbell Scientific and other manufacturers for the U.S. Geological Survey for hydrologic and environmental sensors. SDI-12 was designed to be a simple interface (ground, 12 volts, and signal) that improves compatibility between dataloggers and "smart" microprocessor-based sensors. Other goals of the SDI-12 standard are: low power consumption for battery powered operation via the datalogger; low system cost; use of multiple sensors on one cable connected to one datalogger; allow up to 200 feet of cable between a sensor and a datalogger.

## **Security Code**

A code entered into the datalogger either directly with a keypad or via software or the datalogger's program to prevent unauthorized access to datalogger settings, programs, and data.

## **Server**

Also "communication server", a software application that accepts connections from client applications and provides data or other information as requested. The LoggerNet server manages all the communications and data collection for a network of dataloggers. The collected data is made available for client applications. PC400 also uses the communication server but in a more limited configuration.

## **Short Cut**

A program generator application that ships with PC400, LoggerNet, and is available as a stand-alone product from the Campbell Scientific website. Short Cut does not require knowledge of individual program instructions. Users need only know what kind of datalogger and sensors they're using and decide what output they require. Short Cut generates the program for them. (Contrast a "program generator" with the full-featured "program editors", Edlog and CRBasic Editor.)

## Signature

Number calculated to verify both sequence and validity of bytes within a packet or block of memory.

## Single-ended Analog Input

Some analog sensors have only one signal wire. (They will also have another wire that can be grounded and that is used as the reference for the signal wire.) With this type of sensor, only one analog connection is required. Hence, it needs a "single-ended" or SE analog input. The single ended channels are marked as SE on the datalogger wiring panel.

## Skipped Scan

Skipped scans occur when the actual program execution time exceeds the scan interval. This causes program scans to be skipped. When an skipped scan occurs, the SkippedScan parameter in the datalogger's status table is incremented by 1.

## Station

A datalogger site is often referred to as a station.

## Synchronous

The transmission of data between devices occurs as groups of zeros and ones. For the data to be "read" correctly, the receiving device must begin reading at the proper point in the series. In synchronous communications, this coordination is accomplished by synchronizing the transmitting and receiving devices to a common clock signal (see Asynchronous).

## T

---

### Table Definitions

List of data available from a table-based datalogger. The datalogger supplies this list on request. The tables are determined by the datalogger program. The LoggerNet server must have a current version of the table definitions to collect data from the datalogger.

## Time Stamp

The date and time when data are stored in the datalogger.



## **Transaction**

The exchange of data or information between two devices or programs. For example, setting the clock in a datalogger requires a transaction between the server and the datalogger.

## **U**

---

### **Uncollectable Hole**

Occurs when a hole in the data cache cannot be collected from the datalogger before the data table wraps around and the records are overwritten.

## **W**

---

### **WatchdogErrors**

An entry in the status table that shows the number of watchdog errors that have occurred. The watchdog checks the processor state and resets it if necessary. If an error occurs, the watchdog error counter is incremented.

### **Wiring Panel**

The set of terminals and underlying circuits that enable connections of sensors, control and power supply wiring to the datalogger itself.

# Appendix B. Log Files

---

B.1 Event Logging .....	191
B.1.1 Log Categories .....	191
B.1.2 Enabling Log Files .....	192
B.1.3 Log File Message Formats .....	192

## B.1 Event Logging

As LoggerNet performs its work, it will create records of various kinds of events. The logs can be very useful for troubleshooting problems and monitoring the operation of the datalogger network. You can monitor these logs using LogTool launched from the Tools category of the LoggerNet toolbar. They can also be saved to disk and opened in a text editor. Most users will not need to understand these logs, but if you request technical assistance, a Campbell Scientific applications engineer may ask you to send them one or more of the logs.

### B.1.1 Log Categories

The LoggerNet server logs events in four different kinds of logs as follows:

- **Transaction Status (TranX.log)** – This log file documents the state of the various transactions that occur between the LoggerNet server and devices in the datalogger network. This is the most readable of the logs and contains event messages that are meaningful to most users. Examples of these events are:
  - Datalogger clock check/set
  - Datalogger program downloads
  - Data collection

The format and type of records in this log are strictly defined to make it possible for a software program to parse the log records.

- **Communications Status (CommsX.log)** – This log file documents the quality of communications in the datalogger network.
- **Object State (StateX.log)** – This log file documents the state of an object. This is primarily for troubleshooting by software developers and the messages are relatively free in form.

- **Low Level I/O (IOXSerial Port\_1.log)** – A low level log file is associated with each root device in the datalogger network to record incoming and outgoing communications. While the entire network can be monitored from a single messaging session of the transaction, communications status, or object state logs, monitoring of the low-level log is performed on a session with the root device for that log.

## B.1.2 Enabling Log Files

Use LogTool (**Options > Log File Settings**) to enable logging of events to files. If enabled, the server will write log records to text files in the log file directory using the following file names (depending on the log type):

- Transaction Log – TranX.log
- Communications Log – CommsX.log
- Object State Log – StateX.log
- Low Level Log – IOXSerial Port\_1.log

where “X” is “\$” for the currently active file and 0, 1, 2, etc. for archived files.

The server stores the most recent log records in a file that has a \$ character in the place of the version number. When this file grows to the point that it will exceed the threshold set by the File Size setting for that log, the server will rename the log file by replacing the dollar sign with a new version number. At the same time that the server rolls over to a new log file, the File Count parameter for that log will also be evaluated. If there are more saved files for that log than are allowed by the File Count parameter, the server will delete the oldest of these files until the count is less than or equal to the File Count.

## B.1.3 Log File Message Formats

B.1.3.1 General File Format Information .....	192
B.1.3.2 Transaction Log Format .....	193
B.1.3.3 Communications Status Log Format .....	226
B.1.3.4 Object State Log Format .....	229

### B.1.3.1 General File Format Information

The communications status, transaction, and object state logs all share the same basic file format. Each record in a log file ends with a carriage return and line feed. A single record will consist of

two or more fields where each field is surrounded by quotation marks and separated by commas. The two fields that will be present in all records are:

- **Timestamp** – The server time when the record was generated. It will have the following format:

YYYY-MM-DD HH:MM:SS.mmm

where “YYYY” is the 4-digit year, “MM” is the month number, “DD” is the day of the month, “HH” is the hour in the day (24 hour format), “MM” is the minutes into the hour, “SS” is the seconds into the minute, and “mmm” is the milliseconds into the second.

- **Device Name** – The name of the device associated with the message. If the message is associated with the LoggerNet server, this will be an empty string.

### **B.1.3.2 Transaction Log Format**

Each record in the transaction log includes at least two fields in addition to the timestamp and device name:

- **Message Type Code** – Identifies the type of event that has occurred. This is a number that corresponds to the description immediately following. If this log is being read by a software program, a number is very easy to use for comparison when looking for specific message types.
- **Message Type Description** – Text that describes the message type code.

The following table is a list of the different messages that can appear in the transaction log, some of the optional parameters and what the message means. Where appropriate, a suggested response to the message is provided.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
1	Network device added	Device Name	A new device was added to the network map.	
2	Network branch deleted	Device Name	A branch of the network map was deleted (this may consist of a single device)	
3	Network branch moved	Device Name	A branch of the network map was moved from one parent device to another (not supported in LoggerNet 1.1)	
5	Network logon succeeded	Logon Name		
6	Network logon failed	Logon Name	A client application failed to attach to the server	If unsuccessful logon messages occur frequently, use a network monitor to determine who is trying to connect. If security is enabled this message will appear for someone trying to connect with the wrong user name or password.
7	Security session opened		The security configuration utility has attached to the server.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
8			When the server started up it could not read the security settings file.	This is a normal message on server startup if security has not been set up. If security should be set the file needs to be removed and security re-configured.
9	Modem default database read failed		When the server started up it could not read the default modem file wmodem.ini.	This file should exist in the working directory on the server computer (c:\campbellsci\loggernet\sys\bin). May indicate a permissions or configuration problem on the computer.
10	Modem custom database read failed		When the server started up it could not read the user customized modem settings file wmodem.cust.	If the user has not set up custom modem configurations, this file will not exist.
11	Clock check started		A clock check has been initiated. This clock check is not sent out to the station until the transaction is sent.	
12	Clock set	Device time before set; Server time;	The device clock has been set.	
13		Datalogger time	The datalogger clock has been checked.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
14	Clock check failed	Reason code: 3. Communication failure 4. Invalid datalogger security clearance 5. Invalid transaction number specified (already in use) 6. Communications are disabled for this device 7. The transaction was aborted by client request 8. The device is busy with another transaction	The clock check/set failed for the reason specified in the reason code.	Check the connections of the communication path to the datalogger, make sure the datalogger is connected and has power, check the security setting in the datalogger and in Setup, check that communications are enabled in Setup for all the devices in the path.
15	Starting BMP data advise transaction		A start data advise operation has been initiated. Data advise is not in place until the datalogger responds.	
16	Stopping BMP data advise transaction		A stop data advise operation has been initiated.	
17	BMP data advise transaction started		The message from the datalogger confirming the start of data advise has been received.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
18	BMP data advise transaction stopped		The message from the datalogger confirming the suspension of data advise has been received.	
19	BMP data advise transaction failed		The attempt to start or stop a data advise with the datalogger has failed or the operation has timed out waiting for a response.	Check communications with the datalogger by trying to check the clock. If that fails follow the steps for message 14.
20	Hole detected	Table name; Beginning record number; Ending record number	A hole or missed records has been detected in the data coming from the	The server will automatically try to collect the data if hole collection is enabled.
21	Hole collected	Table name; Beginning record number; Ending record number	The missing records specified have been collected from the datalogger.	
22	Hole lost	Table name; Beginning record number; Ending record number	The missing records have been overwritten in the datalogger.	



Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
23	Hole collect start	Table name; Beginning record number; Ending record number	The hole collect request has been started. This message won't go to the datalogger until the BMP1 message is sent. (see message 104)	
24	Hole collect response received		The datalogger has returned the response to the hole collect request. This will contain either the data or state that the hole is lost.	
25	Hole collect failed		The hole collection request either timed out or a communication failure occurred.	Check communications with the datalogger by trying to check the clock. If that fails follow the steps for message 14.
26	Data polling started		Data collection by polling started.	
27	Data polling complete		Data collection by polling completed	
28	Data polling failed		Data collection by polling failed due to communication failure or a timeout.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
29	Directed data query start		A user initiated query has been started.	
30	Directed data query continue		The requested data in the directed query could not fit in one block and the next part is being requested.	
31	Directed data query complete		The user requested data has been received by the server.	
32	Directed data query failed		The directed query request failed.	
33	Getting logger table definitions		The server is getting the table definitions from the datalogger.	Getting the datalogger table definitions will erase any data in the data cache.
34	Received logger table definitions		The server has received the datalogger table definitions.	
35	Failed to get logger table definitions		The request to get table definitions has failed.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
36	Logger table definitions have changed		The server has detected a change in the table definitions in the datalogger.	A change in table definitions indicates that the datalogger program may have changed. Before updating table definitions make sure the needed data in the data cache has been saved to a file if desired.
37	Updating BMP1 network description		The network description in the RF base is being updated to reflect changes in collection schedule or stations to collect.	
38	BMP1 network description update complete		The RF base has acknowledged the network description update.	
39	BMP1 network description update failed		The network description update to the RF base has either timed out or communication has failed.	Check the connections from the PC to the RF base.
40	Datalogger message	Severity (S for Status, W for Warning, F for Fault); Message text.	This is a message that has been generated by the datalogger (or in some cases the RF base on behalf of the datalogger).	Datalogger warning and fault messages should be investigated using the datalogger operators manual or contacting an applications engineer at Campbell Scientific.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
41	Records received	Table name; Beginning record number; Ending record number	Datalogger records have been received and stored in the data cache.	
42	A datalogger transaction has timed out	Time out period in milliseconds	The server has waited longer than the allotted time for the expected response to a transaction.	Determine the reason for the timeout. This is usually due to a problem with the communications path between the PC and the datalogger.
43	Terminal emulation transaction started		Terminal emulation message has been sent to the datalogger.	
	Terminal emulation transaction complete		Terminal emulation response message has been received from the datalogger.	
45	Terminal emulation transaction failed		The expected terminal emulation response from the datalogger was not received.	
46	Set variable started		The message to set an input location, flag or port has been sent to the datalogger.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
47	Set variable complete		The datalogger has acknowledged the set of an input location, flag or port.	
48	Set variable failed		The datalogger failed to acknowledge the set variable message.	
49	Table resized		The size of the table storage area in the data cache has been changed.	If the table is made smaller the oldest data will be lost.
50	Program file send start		The server is sending a program to the datalogger. The actual program segments will appear as BMP1 message type 4.	
51	Program file send status		The datalogger has received the program segment.	
52	Program file send complete		The datalogger has compiled the program.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
53	Program file send failed		The datalogger did not acknowledge the receipt of the program, the program did not compile, or communications failed with the datalogger.	If the program did not compile check the error messages. Otherwise, check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.
54	Program file receive start		The server is requesting the datalogger program. The actual program segments will appear as BMP1 message type 5.	
55	Program file receive status		A program segment has been received.	
56	Program file receive complete		The datalogger program has been received from the datalogger.	
57	Program file receive failed		The datalogger failed to send the program or communications with the datalogger failed.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
58	Collection schedule: normal		This is an advisory message that the normal data collection schedule is active.	
59	Collection schedule: primary retry		A normal data collection has failed and data collection will be attempted at the primary retry interval.	Determine the reason for communication failure. Temporary communication problems may cause the collection state to change between normal and primary.
60	Collection schedule: secondary retry		The number of primary retries specified has passed and data collection will be attempted at the secondary retry interval.	
61	Collection schedule suspended		The scheduled data collection has been turned off or suspended because communication is disabled or table definitions have changed.	
62	Primary retry collection attempt failed		Data collection on the primary data collection interval failed.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
63	Secondary retry collection attempt failed		Data collection on the secondary data collection interval failed.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.
64	Device restore from file succeeded		On server startup a device previously entered in the network map has been restored.	
65	Device restore from file failed		On server startup a device in the network map could not be restored.	This is an indication that the configuration file has been corrupted. Check the network map and the computer file system.
66	Device save to file succeeded		The update to the device configuration file was successful.	
67	Device save to file failed		The update to the device configuration file failed.	This may be due to a problem with directory permissions or a corrupted directory.



Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
68	Packet delivery failed	Fault code: 1. Incompatible BMP1 device or malformed packet 2. Routing failure {unrecognized station number} 3. Temporarily out of resources 4. Link failure	This is a message from the RF base indicating that a BMP1 message didn't make it to the data logger.	<p>Codes 1 and 3 are rare. If ever seen contact an technical support engineer at Campbell Scientific.</p> <p>Code 2 indicates that the RF base has lost the network map and doesn't know how to route the message. The server automatically resends the network map.</p> <p>Code 4 is an indication that the RF base was not able to communicate with the RF modem attached to the datalogger. These will happen occasionally as part of normal operations. Frequent occurrences indicate that the radio, antenna, connectors and RF link be reviewed.</p>

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
69	Unexpected change in datalogger table definitions		As part of data collection the server has detected a change in the datalogger's table definitions.	A change in table definitions indicates that the datalogger program may have changed. This will suspend data collection and warnings will be shown in the Status Monitor. Data Collection can only be restored by updating table definitions. Before updating table definitions make sure the needed data in the data cache has been saved to a file if desired.
70	A device setting value has changed	Setting Identifier; Client's logon name; New value of the setting	A client has changed one of the device configuration settings.	
71	A LgrNet setting value has changed	Setting Identifier; Client's logon name;	A client has changed one of the server configuration settings.	
72	Client defined message	Client defined message	These messages are placed in the transaction log by client applications. The message should indicate which client entered the message.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
73	Socket listen failed		Indicates an error in the computer system that prevents the server from listening for client connections on a socket.	This is a rare error and results in a problem with the computer operating system. If rebooting the computer does not clear the error, contact an application engineer
74	Device renamed		The name of a device in the network was changed.	
75	Logger locked		This message indicates the start of a transaction such as terminal emulation that will tie up the datalogger preventing other operations.	
76	Logger unlocked		The transaction blocking datalogger access has completed.	
77	Null program sent		The server has sent a null program to get an older datalogger (CR7X or 21X) out of keyboard emulation mode.	
78	Server started	The server version	The server has been started.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
79	Server shut down		The server is being shut down	If a new "server started" message is seen without the shut down message before it, this is an indication that the server or the PC crashed without exiting properly.
80	Collect area initialized	Collect area name	A data cache collect area has been created.	
82	Collect area removed		A data cache collect area has been removed	
83	LgrNet restore failed		On server startup the network description file, csilgrnet.dnd, could not be read.	The network setup and configuration will have to be restored from a backup or re-entered. Try to determine what corrupted or removed the network description file.
84	Security manager restore failed		On server startup the security manager database could not be restored.	There is a problem with the computer or operating system. If rebooting the machine does not get it working get help from someone who can troubleshoot computer problems.
85	Data restore failed		On server startup the data broker data storage area could not be created.	This is a computer problem. The files are either not present or are corrupted. See notes for message 83.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
86	Manual poll transaction started	Client logon name	The listed client is starting a manual poll operation according to the scheduled collection settings. A manual poll is initiated from the <b>Collect Now</b> button on the Connect Screen.	
87	Manual poll transaction complete		The manual poll operation has received the data from the datalogger.	
88	Manual poll aborted		The manual poll operation was stopped or failed to complete due to communications failure or a timeout.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.
89	Selective manual poll begun	Collect area name	A user specified poll has been started for one of the datalogger collect areas.	
90	Selective manual poll complete	Collect area name	The user specified manual poll has completed.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
91	Selective manual poll aborted	Collect area name	The user specified manual poll failed.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.
92	Polling started on collect area	Collect area name	Data has been requested for the specified collect area. This message is always associated with another message indicating whether this is scheduled, manual or selective manual polling.	Collect areas can be table for table mode dataloggers, final storage areas, ports and flags, or input locations.
93	Collect area poll data	Collect area name	Data has been received from an array based datalogger for the specified collect area.	
94	Collect area polling complete	Collect area name	Data collection for the specified collect area has successfully completed.	
95	Collect area polling failed	Collect area name	Data collection for the specified collect area failed.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
96	Scheduled polling begun		Scheduled data collection has started.	
97	Scheduled polling succeeded		Scheduled data collection has completed.	
98	Scheduled polling failed		Scheduled data collection failed.	Check communications with the datalogger by trying to check the clock. If that fails, follow the steps for message 14.
99	Collect area first poll		This message is posted either the first time data is collected for a collect area, or holes were lost for the datalogger.	If this is not the first poll for the collect area, this message indicates that data that had been stored in the datalogger was lost before it could be collected.
100	Table mount failed	Table name; Operating system information regarding the failure	The server was not able to create a data collection area from the stored table configuration file or new table definitions. This could be the result of trying to create table files that are too large for the computer system.	Check the computer operating system integrity. Verify that the LoggerNet system configuration files exist and the directory has not been corrupted.

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
101	Add record failed	Table name; Beginning record number; End record number; A reason for the failure	The server was not able to write data records to the data storage area.	This indicates a problem writing to files on the computer hard disk. Verify write permissions are set and that there is sufficient space left on the disk.
102	Collect area skipped warning	Collect area name	The specified collect area was skipped because the associated table has not been initialized by the server yet.	During system startup this is a normal message. If it occurs at other times contact an application engineer.
103	Collect area skipped error	Collect area name	The specified collect area was skipped because the server could not initialize the associated table.	See message 100



Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
104	BMP1 packet sent	<p>The packet message type code:</p> <p>0 Packet Delivery Fault Notification</p> <p>1 Status/Warning/Fault Notification</p> <p>2 Network Description Transaction</p> <p>3 Clock Check/Set Transaction</p> <p>4 Program Down-load Transaction</p> <p>5 Program Up-load Transaction</p> <p>7 Data Advise Command Transaction</p> <p>8 Data Advise Notification Packet</p> <p>9 Hole Collection Command Transaction</p> <p>10 Control Command (Set Variable) Transaction</p> <p>11 User I/O Transaction (Terminal Mode)</p> <p>12 Memory Image Down-load</p>	The specified BMP1 packet was sent to the serial communication interface. The number specifies the type of message that was sent.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
		Transaction 13 Memory Image Up- load Transaction 14 Get Table Definitions Transaction 15 RF Test Transaction 16 Communication Status Notification		

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
105	BMP1 packet received	<p>The packet message type code:</p> <p>0 Packet Delivery Fault Notification</p> <p>1 Status/Warning/Fault Notification</p> <p>2 Network Description Transaction</p> <p>3 Clock Check/Set Transaction</p> <p>4 Program Down-load Transaction</p> <p>5 Program Up-load Transaction</p> <p>7 Data Advise Command Transaction</p> <p>8 Data Advise Notification Packet</p> <p>9 Hole Collection Command Transaction</p> <p>10 Control Command (Set Variable) Transaction</p> <p>11 User I/O Transaction (Terminal Mode)</p> <p>12 Memory Image Down-load</p>	<p>The specified BMP1 packet was received over the serial communications link. The number indicates the type of message received. The specified BMP1 packet was received over the serial communications link. The number indicates the type of message received.</p>	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
		Transaction 13 Memory Image Upload Transaction 14 Get Table Definitions Transaction 15 RF Test Transaction 16 Communication Status Notification		
106	Data file output failed		Data collected from a datalogger could not be written to the data output file.	Check that there is space available on the hard disk and that write permissions allow the server to write the data output files.
107	Max time on-line exceeded	The amount of time the device was connected, in milliseconds	A client kept the communication link on-line longer than the specified max time on-line.	
108	Table reset	The name of the table that was reset; The account name of the logged in client	The name of a table was changed at the request of a client. On CR5000 and CR9000 loggers this is a reset for the table in the datalogger and on the PC.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
109	Collect schedule reset	The account name of the logged in client	The collection schedule was reset by the indicated client.	
110	Collect area setting changed	The name of the collection area; The setting identifier for the setting that was changed; The new value of the setting; The account name of the logged in client.	One of the settings for the specified collect area was changed. The identifiers for the setting can be found in CoraScript help.	
111	PakBus route added		A new PakBus route has been added to the routing table.	
112	PakBus route lost		A PakBus route has been lost and will be removed from the routing table.	
113	PakBus station added		A new PakBus station was added to the network.	
114	Call-back begin		A device has called in to the server starting the call-back response.	
116	Call-back stopped		A datalogger that called in to the server with call-back is hanging up.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
117	Client logged off	The login name of the client; The reason the session was closed.	A client application has closed or lost the connection to the server.	
118	Table size reduced during creation	The size of the table in the data cache was reduced because there was not enough computer disk space to create it, or the file would have exceeded the 2 Gbyte size limit.	Reduce the size of the tables in the datalogger program or get more hard disk storage space for the computer.	
119	Security enabled	Account name used to enable security.	Security has been enabled on the LoggerNet server.	Username and passwords will now be required for communication with the LoggerNet server.
120	Security disabled	Account name used to disable security.	Security has been disabled on the LoggerNet server.	
121	Security account added	Account name used to add new account; Name of the account that was added.	A new security account has been added.	
122	Security account changed	Account name used to change account; Name of the account that was changed.	A change has been made to the attributes of a security account.	
123	Security account deleted	Account name used to delete account; Name of the account that was deleted.	A security account has been deleted.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
124	Security interface locked	Account name used by the client that started the transaction that locked the interface.	The security interface is locked because an account is currently making changes to the interface.	
125	Security interface unlocked	Account name used by the client that started the transaction that unlocked the interface.	The security interface is unlocked because pending changes were applied or canceled.	
126	Network lock started	Account name used by the client that started the transaction that locked the network; Client that started the transaction that locked the network.	The network is locked because a client is currently making changes to the interface.	Some functionality will be disabled until the network lock is stopped. To unlock, determine why the client transaction locked the network. For instance, there may be unapplied changes in the Setup Screen. Apply or cancel the changes to unlock the network.
127	Network lock stopped		The network is unlocked because pending changes were applied or canceled.	
128	Set value command received	Name of the table specified; Name of the field specified.	A device has requested to set a value in one of its tables.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
129	Column renamed	Name of the table; Original column name; New column name; Reason why column was renamed.	The name of a column has been changed due to an incompatibility with a previous field in the table that had the same name.	
130	Last primary retry failed	Number of retries that were made.	The last primary retry attempt failed.	Check the connections of the communication path to the datalogger, make sure the datalogger is connected and has power, check the security setting in the datalogger and in Setup, check that communications are enabled in Setup for all the devices in the path.
131	Working directory snapshot	Name of the file that was created.	The server created a backup.	
132	Working directory snapshot restored	Name of the file from which the network was restored.	The network was restored from a backup file.	
133	File receive started	Name of the file being received.	The server has begun a file retrieval from the datalogger.	
134	File receive completed	Name of the file received.	The server completed a file retrieval from the datalogger.	



Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
135	File receive failed	Name of the file received; Reason for the failure.	The server failed to retrieve a file.	Check the connections of the communication path to the datalogger, make sure the datalogger is connected and has power, check the security setting in the datalogger and in Setup, check that communications are enabled in Setup for all the devices in the path.
136	File send started	Name of the file being sent.	The server has begun to send a file to the datalogger.	
137	File send completed	Name of the file sent.	The server has completed a file send to the datalogger.	
138	File send failed	Name of the file sent; Reason for the failure.	The server failed to send a file to the datalogger.	Check the connections of the communication path to the datalogger, make sure the datalogger is connected and has power, check the security setting in the datalogger and in Setup, check that communications are enabled in Setup for all the devices in the path.
139	Collect area poll stopped due to table interval		Polling on a collect area was aborted because the table interval has not expired.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
140	Device setting override	Setting identifier; Name of the user's account overriding the setting; Value of the setting.	One of the device settings has been overridden.	
141	Device setting override stopped		The device setting override has been stopped.	
142	Collect area setting overridden	Name of the collect area; Setting identifier; Name of the user overriding the setting; Value of the setting.	One of the device collect area settings has been overridden.	
143	Device collect area setting override stopped		The device collect area setting override has been stopped.	
144	Data file opened	Collect area name; File name.	Collect area data file has been opened by the server.	
145	Data file closed	Collect area name; File name.	Collect area data file has been closed by the server.	
146	Datalogger query started	Table name; Query Mode; Client Logon Name	A datalogger query has been started by a client.	
147	Datalogger query temp table created	Table name; Temporary table name.	A temporary cache table has been created for a datalogger query.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
148	Datalogger query records received	Table name; Begin record number; End record number.	Records have been received from the datalogger for a datalogger query transaction.	
149	Datalogger query complete	Table name.	All of the data for a datalogger query transaction has been collected from datalogger.	
150	Datalogger query closed	Table name.	Client has closed a datalogger query transaction.	
151	Existing data file renamed	Collect area name; File Name; Reason for renaming.	Server has renamed an existing data file as a result of attempting to append data in an incompatible format.	Existing data file will be renamed with a .backup extension. New data will be stored to the specified file name.
153	Program/TDF file associate start	User account name.	Client has begun a program file association transaction.	
154	Program/TDF file associate complete		Program file associate transaction has successfully concluded.	

Transaction Log Messages				
Code	Message Text	Message Parameters	Message Meaning	User Response to Message
155	Program/TDF file associate failed	Reason for the failure.	Program file associate transaction has failed.	
156	File control started	File control command; First argument (optional); Second Argument (optional); User name (optional).	A file control operation has begun with a PakBus datalogger.	
157	File control complete	File control command; First argument (optional); Second Argument (optional); User name (optional).	A file control operation with a PakBus datalogger has successfully completed.	
158	File control failed	File control command; First argument (optional); Second Argument (optional); User name (optional).	A file control operation with a PakBus datalogger has failed.	Check the connections of the communication path to the datalogger, make sure the datalogger is connected and has power, check the security setting in the datalogger and in Setup, check that communications are enabled in Setup for all the devices in the path.

## Transaction Log Example

```
"2009-04-15 16:41:05.367","CR1000","11","Clock check started"
"2009-04-15 16:41:05.429","CR1000","13","Clock checked","2009-04-15 16:41:33.44","2009-04-15 16:41:05.421","-28"
"2009-04-15 16:41:06.367","CR1000","86","Manual poll started","admin"
"2009-04-15 16:41:06.367","CR1000","92","Collect area poll started","TestFast"
"2009-04-15 16:41:06.382","CR1000","41","Records received","TestFast","21007","21007","polling"
"2009-04-15 16:41:06.382","CR1000","20","Hole detected","TestFast","20769","21006"
"2009-04-15 16:41:06.429","CR1000","11","Clock check started"
"2009-04-15 16:41:06.492","CR1000","41","Records received","TestFast","20769","20799","polling"
"2009-04-15 16:41:06.507","CR1000","144","data file opened","TestFast","C:\Campbellsci\LoggerNet\CR1000_TestFast.dat"
"2009-04-15 16:41:06.507","CR1000","21","Hole collected","TestFast","20769","20799"
"2009-04-15 16:41:06.507","CR1000","41","Records received","TestFast","20800","20864","polling"
"2009-04-15 16:41:06.507","CR1000","21","Hole collected","TestFast","20800","20864"
"2009-04-15 16:41:06.523","CR1000","13","Clock checked","2009-04-15 16:41:34.55","2009-04-15 16:41:06.516","-28"
"2009-04-15 16:41:06.601","CR1000","41","Records received","TestFast","20865","20899","polling"
"2009-04-15 16:41:06.601","CR1000","21","Hole collected","TestFast","20865","20899"
"2009-04-15 16:41:06.601","CR1000","41","Records received","TestFast","20900","20960","polling"
"2009-04-15 16:41:06.601","CR1000","21","Hole collected","TestFast","20900","20960"
"2009-04-15 16:41:06.648","CR1000","41","Records received","TestFast","20961","20999","polling"
"2009-04-15 16:41:06.648","CR1000","21","Hole collected","TestFast","20961","20999"
"2009-04-15 16:41:06.648","CR1000","41","Records received","TestFast","21000","21006","polling"
"2009-04-15 16:41:06.648","CR1000","21","Hole collected","TestFast","21000","21006"
"2009-04-15 16:41:06.679","CR1000","145","data file closed","TestFast","C:\Campbellsci\LoggerNet\CR1000_TestFast.dat"
"2009-04-15 16:41:06.679","CR1000","94","Collect area poll complete","TestFast","956","956"
"2009-04-15 16:41:06.695","CR1000","87","Manual poll complete"
"2009-04-15 16:41:07.429","CR1000","11","Clock check started"
"2009-04-15 16:41:07.445","CR1000","13","Clock checked","2009-04-15 16:41:35.46","2009-04-15 16:41:07.438","-28"
"2009-04-15 16:41:08.429","CR1000","11","Clock check started"
```

### B.1.3.3 Communications Status Log Format

Each record in the communications status log includes two fields in addition to the timestamp and device name:

- **Severity** – A single character code that indicates the type of message. The following values are legal:
  - "S" (Status) Indicates that the identified operation has successfully completed.
  - "W" (Warning) Indicates that the server has attempted to retry the operation with the identified device.
  - "F" (Fault) Indicates that the identified operation has failed and that the server has stopped retrying.
- **Description** – text providing more details about the event.

## Communications Status Log Example

```
"2009-04-15 16:41:05.367","IPPort","S","Device dialed"
"2009-04-15 16:41:05.382","PakBusPort_ip","S","sending message","src: 4094","dest: 2","proto: PakCtrl","type: 0x09","tran: 214"
"2009-04-15 16:41:05.398","PakBusPort_ip","S","received message","src: 2","dest: 4094","proto: PakCtrl","type: 0x89","tran: 214"
"2009-04-15 16:41:05.398","CR1000","S","PakCtrl message received","89"
"2009-04-15 16:41:05.413","PakBusPort_ip","S","sending message","src: 4094","dest: 2","proto: BMP5","type: 0x17","tran: 213"
"2009-04-15 16:41:05.429","PakBusPort_ip","S","received message","src: 2","dest: 4094","proto: BMP5","type: 0x97","tran: 213"
"2009-04-15 16:41:05.429","CR1000","S","BMP5 message received","type: 0x97","check/set clock"
"2009-04-15 16:41:06.367","PakBusPort_ip","S","sending message","src: 4094","dest: 2","proto: BMP5","type: 0x09","tran: 217"
"2009-04-15 16:41:06.382","PakBusPort_ip","S","received message","src: 2","dest: 4094","proto: BMP5","type: 0x89","tran: 217"
"2009-04-15 16:41:06.382","CR1000","S","BMP5 message received","type: 0x89","table poll","CR1000.TestFast"
"2009-04-15 16:41:06.382","PakBusPort_ip","S","sending message","src: 4094","dest: 2","proto: BMP5","type: 0x09","tran: 218"
"2009-04-15 16:41:06.492","PakBusPort_ip","S","received message","src: 2","dest: 4094","proto: BMP5","type: 0x89","tran: 218"
"2009-04-15 16:41:06.492","CR1000","S","BMP5 message received","type: 0x89","table poll","CR1000.TestFast"
"2009-04-15 16:41:06.523","PakBusPort_ip","S","sending message","src: 4094","dest: 2","proto: BMP5","type: 0x17","tran: 219"
"2009-04-15 16:41:06.523","PakBusPort_ip","S","received message","src: 2","dest: 4094","proto: BMP5","type: 0x97","tran: 219"
```

Communication Status Log Messages		
Message Text	Message Meaning	User Response to Message
Serial packet X exchanged	The low level serial BMP1 communication framing packet was sent and the response received from the device. (CR10X-TD table based type devices)	
Classic;;Cmd	The listed command was sent to an array based datalogger.	For a list of the commands and their meanings see the datalogger operator's manual.
BMP1 packet received	A BMP1 packet was received from the device. (CR10X-TD type devices only)	
RPC packet exchanged	A BMP3 packet was exchanged. (CR5000, CR9000 dataloggers only)	
Datalogger did not respond to end command	The computer tried to terminate the connection but the datalogger did not acknowledge the shutdown.	This is an indication that there is a communications problem between the computer and the datalogger. Check the cables and connectors and make sure the datalogger has power.

Communication Status Log Messages		
Message Text	Message Meaning	User Response to Message
PakBus framing error	LoggerNet received data from the link that cannot be verified to be part of a PakBus packet.	Some possible causes: the datalogger program or its settings are configured to write data on the port on which you are attempting to connect, there is a mismatch in baud rates between your computer serial port and the datalogger, the link is dropping sequences of characters from transmission because the CPU on the computer is heavily loaded or because of faulty USB drivers. Possible solutions: change the port you are using to communicate, try using a slower baud rate to communicate with the datalogger, use Windows Task Manager to determine whether there are processes that are loading down your CPU, check to see if there is an updated driver for your USB/RS232 adapter.
Invalid low level signature	The packet received from the device got corrupted and the packet signature doesn't match the packet contents.	Check to find out where in the communications link noise or signal corruption is causing the data to be disrupted.
Provider opened	The serial communications port has been initialized.	
Device dialed	The communications link has been initialized to transfer data packets.	

Communication Status Log Messages		
Message Text	Message Meaning	User Response to Message
Provider closed	The serial communications port has been closed.	
Unable to Locate Serial synch byte	The low level communications synchronization byte was not received after the computer sent out a serial packet.	This indicates that the device is either not responding or responding with an invalid communications protocol. This message would appear if trying to talk to an array based datalogger that is set up as a table based datalogger in the network map.

### B.1.3.4 Object State Log Format

The object state log includes two fields in addition to the timestamp and device name:

- **Object Name** – The name of the object from which the message is being generated. Typically this will be the name of an object method.
- **Description** – Any extra information associated with the event.



## Object State Log Example

"2009-04-15 16:41:05.351","CR1000","starting BMP5 operation","manage comm resource"  
"2009-04-15 16:41:05.367","CR1000","starting BMP5 operation","check/set clock"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Request Transaction Focus","check/set clock","213"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Transaction focus start","PakCtrl::Hello","2","214"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Dev::sesBegin","01100C90"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Dev::cmdAdd","MyPort::serial\_framing\_command","3"  
"2009-04-15 16:41:05.367","IPPort","Dev::reqDevice","Requesting device: PakBusPort\_ip"  
"2009-04-15 16:41:05.367","IPPort","Dev::cmdFinished","Callback Command"  
"2009-04-15 16:41:05.367","IPPort","Dev::sesEnd","016E83B0"  
"2009-04-15 16:41:05.367","IPPort","DevHelpers::HangupDelaySession","Hangup delay: 10"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Dev::reqDevResp","IPPort","PakBusPort\_ip","success"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Dev::sesBegin","016E83B0"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Dev","Going on-line"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Dev::onNextCommand","Executing command","MyPort::serial\_framing\_command","3"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","Csi::PakBus::SerialPortBase::link\_type","watch dog timeout set at 40000"  
"2009-04-15 16:41:05.367","PakBusPort\_ip","send\_ring","remote: 2","retries: 0"  
"2009-04-15 16:41:05.382","IPPort","DevHelpers::HangupDelaySession","post completion"  
"2009-04-15 16:41:05.382","IPPort","Dev::sesEnd","0166DCA8"  
"2009-04-15 16:41:05.382","IPPort","Dev","Hangup delay complete received, no sessions left"  
"2009-04-15 16:41:05.382","PakBusPort\_ip","arm transaction watchdog","PakCtrl::Hello","2","7250","37350"  
"2009-04-15 16:41:05.382","CR1000","Bmp5::Datalogger","delay\_hangup created"  
"2009-04-15 16:41:05.382","PakBusPort\_ip","Csi::PakBus::SerialPortBase::link\_type","watch dog timeout set at 40000"  
"2009-04-15 16:41:05.382","CR1000","starting BMP5 operation","delay hangup"  
"2009-04-15 16:41:05.382","CR1000","Bmp5::OpDelayHangup","transaction started","216"  
"2009-04-15 16:41:05.413","PakBusPort\_ip","PakBusTran closing","PakCtrl::Hello","2","214"  
"2009-04-15 16:41:05.413","PakBusPort\_ip","Csi::PakBus::Router","entering close\_transaction"  
"2009-04-15 16:41:05.413","PakBusPort\_ip","Release Transaction Focus","PakCtrl::Hello","2","214"  
"2009-04-15 16:41:05.413","PakBusPort\_ip","Transaction focus start","check/set clock","213"  
"2009-04-15 16:41:05.413","PakBusPort\_ip","arm transaction watchdog","check/set clock","11250","37355"  
"2009-04-15 16:41:05.413","PakBusPort\_ip","Csi::PakBus::SerialPortBase::link\_type","watch dog timeout set at 40000"  
"2009-04-15 16:41:05.413","PakBusPort\_ip","Csi::PakBus::Router","leaving close\_transaction"  
"2009-04-15 16:41:05.445","PakBusPort\_ip","PakBusTran release focus","check/set clock","37355"  
"2009-04-15 16:41:05.445","PakBusPort\_ip","Release Transaction Focus","check/set clock","213"  
"2009-04-15 16:41:05.460","PakBusPort\_ip","PakBusTran closing","check/set clock","213"  
"2009-04-15 16:41:05.460","PakBusPort\_ip","Csi::PakBus::Router","entering close\_transaction"  
"2009-04-15 16:41:05.460","PakBusPort\_ip","Csi::PakBus::Router","leaving close\_transaction"  
"2009-04-15 16:41:06.367","CR1000","starting BMP5 operation","table poll","CR1000.TestFast"  
"2009-04-15 16:41:06.367","PakBusPort\_ip","Request Transaction Focus","table poll","CR1000.TestFast","217"  
"2009-04-15 16:41:06.367","PakBusPort\_ip","Transaction focus start","table poll","CR1000.TestFast","217"  
"2009-04-15 16:41:06.367","PakBusPort\_ip","arm transaction watchdog","table poll","CR1000.TestFast","7250","37361"  
"2009-04-15 16:41:06.367","PakBusPort\_ip","Csi::PakBus::SerialPortBase::link\_type","watch dog timeout set at 40000"  
"2009-04-15 16:41:06.382","CR1000","Bmp5::OpTablePoll::on\_bmp5\_message - check newest","table poll","CR1000.TestFast"  
"2009-04-15 16:41:06.382","PakBusPort\_ip","Release Transaction Focus","table poll","CR1000.TestFast","218"  
"2009-04-15 16:41:06.382","CR1000","Bmp5::OpTablePoll::on\_check\_complete","table poll","CR1000.TestFast"  
"2009-04-15 16:41:06.382","PakBusPort\_ip","Request Transaction Focus","table poll","CR1000.TestFast","218"  
"2009-04-15 16:41:06.382","PakBusPort\_ip","Transaction focus start","table poll","CR1000.TestFast","218"  
"2009-04-15 16:41:06.382","PakBusPort\_ip","arm transaction watchdog","table poll","CR1000.TestFast","7250","37365"

# Appendix C. Split

---

The following PDF pages describe the functionality of Split.

# Functional Overview

Split is a tool to analyze data collected from Campbell Scientific dataloggers. Its name comes from its function of splitting out specific data from a larger data file. Originally, Split could only process mixed-array files, and it was used to “split” the different arrays – typically different time intervals – of a file into separate files (e.g., for hourly versus daily data).

In addition to splitting out mixed-array data, Split can filter output data based on time or conditions, calculate statistics and new values, reformat files, or check data quality (limit testing). Split can generate tables with report and column headings, as well as time synchronize and merge up to eight data files.

Input Files (maximum of eight) are read by Split, specific operations are performed on the data, and the results are output to a new Output File or a printer. Split creates a parameter file (*filename.PAR*) that saves all of your settings such as which data files are read, what operations are performed on the data set, and where the final results will be saved. The parameter file may be saved and used again.

Input Files must be formatted in Printable ASCII, Comma Separated ASCII, Field Formatted ASCII, Final Storage (Binary) Format, Table Oriented ASCII (TOASCII or TOA5), Table Oriented Binary (TOB), or Raw A/D data (such as the results of a burst measurement).

Split can be used to convert a file of one format to a different format. For example, a Table Oriented ASCII file can be converted to the Comma Separated ASCII format used in mixed-array datalogger data files. This is useful to convert table-based data files to work with applications that were written to work with mixed-array files.

Output files generated by Split can be Field Formatted (default), Comma Separated ASCII, or Printable ASCII. Split can also create reports in ASCII as well as html formats, or send them directly to a printer.

Split lends itself to experimentation. The processed data are displayed on the screen, giving immediate feedback as to the effect of changes or new entries to the parameter file. Split does not modify the original Input File.

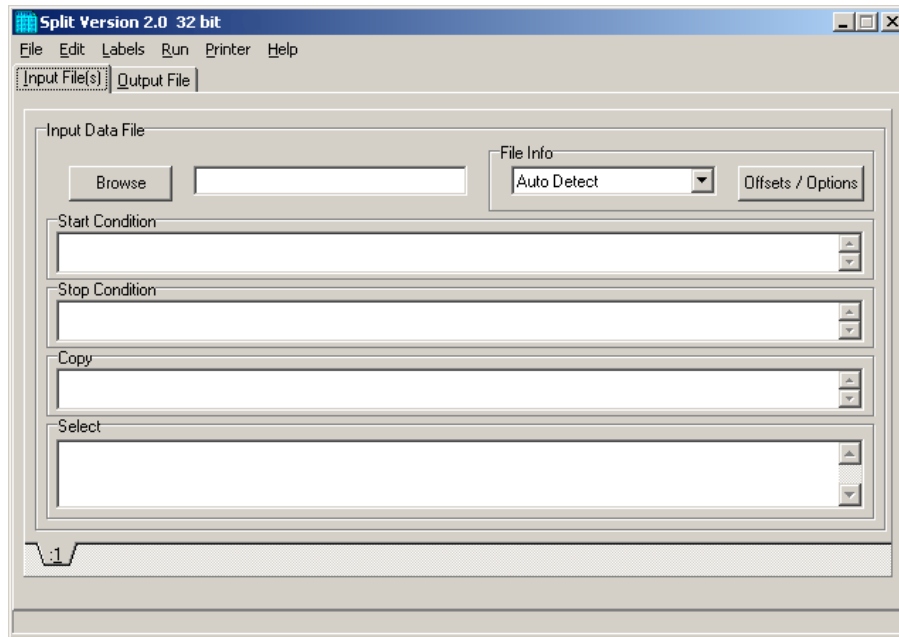
## Getting Started

The most common use of Split is to separate array data collected on a particular interval from a data file containing data output at several different intervals.

In the following example, hourly data are split from a data set that contains 15 minute, hourly and daily data. The data was collected from BirchCreek, a CR10X datalogger. The CR10X was loaded with a program created by Edlog named Birch.dld.

The 15 minute data, array 99, the hourly data, array 60, and the daily data, array 24, are intermixed in the data file.





When you start Split a blank template similar to the one above is shown. This template is used to enter the parameters that will define what data from the input file to include in the output file. The parameters entered on this template can be saved as a parameter file (\*.PAR) and reused for other data.

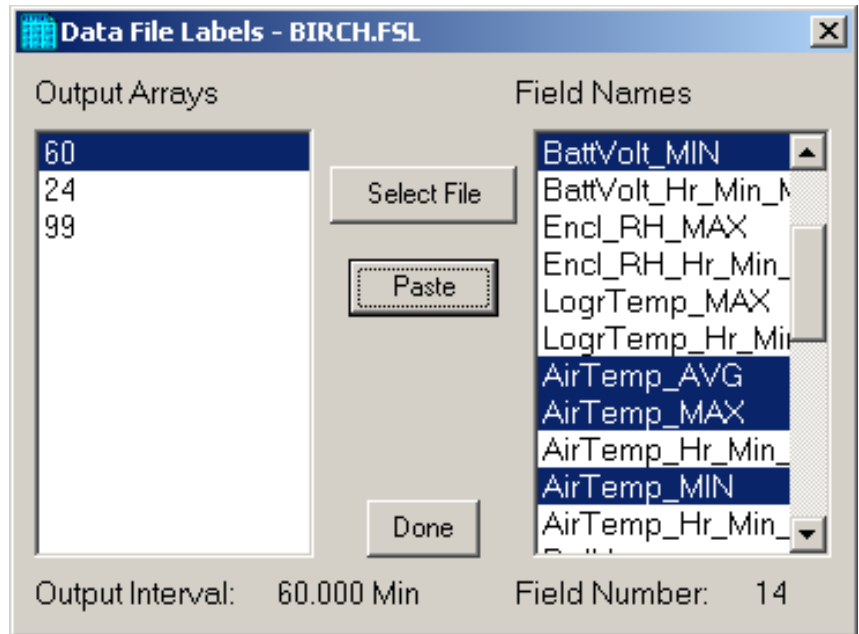
On the **INPUT FILE** tab you only need to specify the input file name, copy condition, and the data to select. Split allows start and stop conditions to be specified but if they are left blank, the entire file will be read.

The name of the Input Data File can be typed in or the **Browse** button can be used to select from available files. In this example BirchCreek.dat will be selected as the input data file.

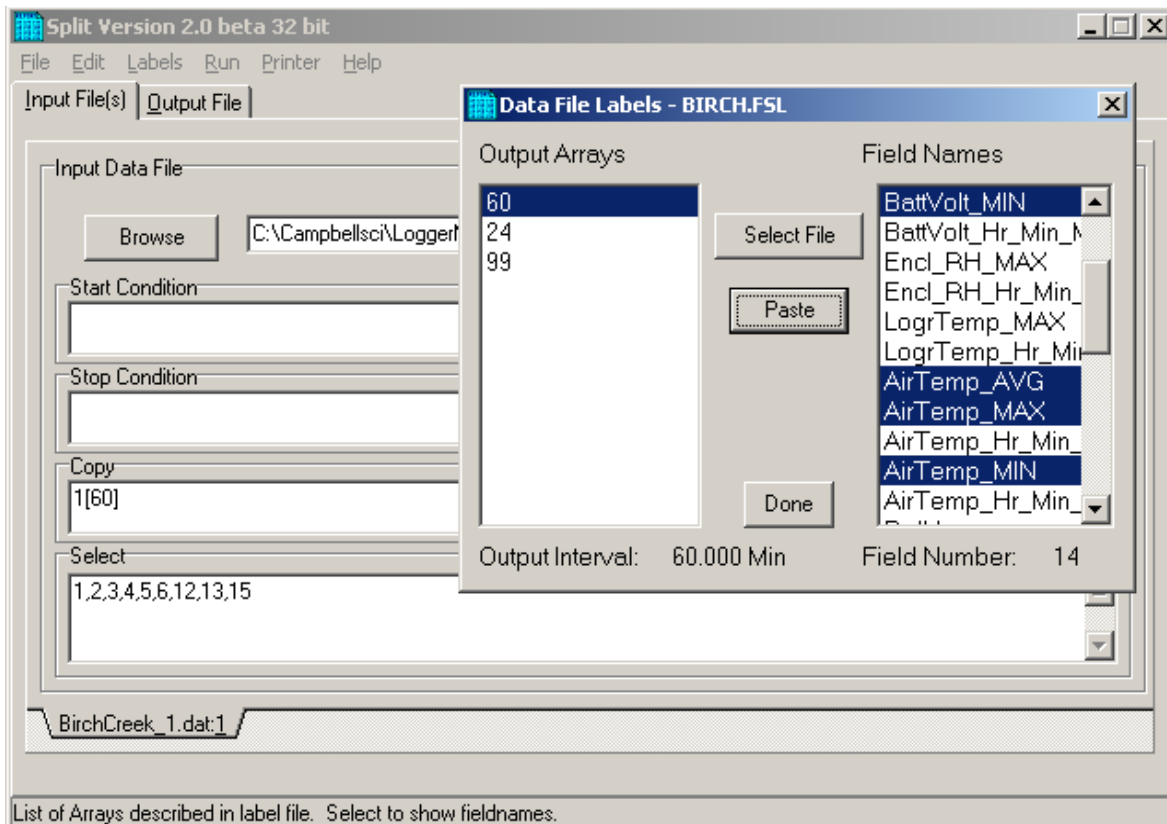
Selecting the data to copy is simplified by the use of the Birch.fsl file. From the toolbar menu, click **Labels | Use Data Labels**. From the Data File Labels pop-up, Select File is used to find Birch.fsl. When one of the Output Arrays is highlighted, the Field Names of the data in that array are displayed.

#### NOTE

In this example, a mixed array data file is processed and the Use Data Labels feature uses an FSL file. When processing a table-based datalogger file, change the file type to “Table-based data file to use for labels” and select the table-based DAT file. Split will use the header information from this file for its labels.



In this example we want the hourly data (note the Output Interval at the bottom of the Data File Label window), so click array 60. To paste the desired values from this array into the Select box, select the field names while holding down the <ctrl> key. All of the values could be selected by clicking the first one and holding the mouse button down, and dragging to the end. Once the values you want have been selected click Paste.

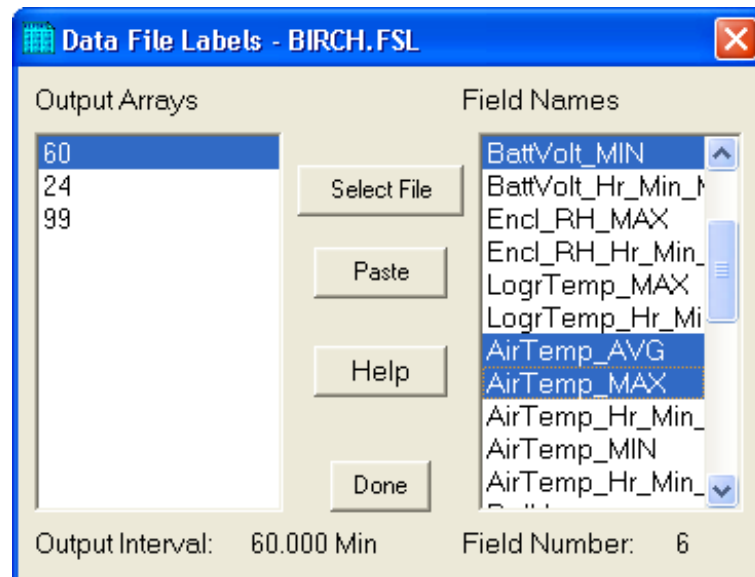


Note that the cursor in the **INPUT FILE(S)** screen must be in valid paste area (Copy or Select). If the cursor is in the File name box or in Start/Stop condition, you will get the error message “Cannot Paste There”.

The Paste operation copied the numbers of each of the fields into the Select box. Notice also that it pasted the Array ID into the copy condition: 1[60] tells Split that in order to copy a line of data, the first value in that line must be 60. Split uses the Array ID to discriminate between the hourly and daily data.

Now specify the Output File name. (Without one specified, Split will run and display results but no output file will be created.) Click the **OUTPUT FILE** tab. Type in “hourly” for the name of the output file. By default, Split will use the file extension “PRN”, creating the output file: hourly.prn. Depending upon the option chosen in the “If File Exists then” list box, an existing PRN file may be overwritten, appended to, or saved under a new name.

The Labels option from the toolbar can also assist in labeling the output values. Once again, choose **LABELS | USE FINAL STORAGE LABELS** and select array 60 and all the field names. This time move the cursor to Line 1 of the first column of labels on the **OUTPUT FILE** tab and press **Paste**. The labels from the final storage file will be pasted into each of the columns. Split will automatically break a label name into multiple rows at the “\_” in a label name.



Maximum column heading width is one less than the number entered in the Default Column Width field. However, entering a number in the Width row for the column will set the column width for an individual column. Any FSL labels that are too long for Split column headings will be shown in red. They should be edited before running Split. To edit one of the labels, press the <Enter> key or use a mouse to copy, cut, and paste. A Report Heading can also be entered using the same editing technique.

Report and Column Headings

Report Heading: Hourly Data

Column#	4	5	6	7	8	9
Element/Field#	4	5	6	12	13	
Filename	BirchCreek_1.d	BirchCreek_1.d	BirchCreek_1.d	BirchCreek_1.d	BirchCreek_1.d	
Line 1	Hour_Minute	Battery	BattVolt	AirTemp	AirTemp	AirTemp_MIN
Line 2		Volts	Min	Average	Max	
Line 3				C	C	
Decimal						
Width	12		12			12

For table based data files the timestamp is normally the first column and is a quoted text string ("2002-02-26 10:30:00"). To display these timestamps in the output you will need to change the column width for the first column to at least 24. If the column width is too small to accommodate the value output, the string will be highlighted in red and preceded by an asterisk, with the words "Bad Data" in the lower right corner when the file is processed.

To run Split, select **RUN | GO**. The hourly data will be split out and stored in hourly.prn. The results are displayed on the screen as shown below.

#### NOTE

When Split is running on large files, the line counters will update only every 1000 lines.

Close the Run window. If you wish to save this parameter file for future reports, choose **FILE | SAVE**. The file will be saved with a .PAR extension.

Split Run 2.0beta 32bit

60	Year_RTM	Day_RTM	Hour_Minute	Battery	BattVolt	AirTemp	AirTemp
				Volts	Min	Average	Max
						C	C
60	2002	42	1000	13.9	13.79	-12.51	-7.43
60	2002	42	1100	13.78	13.78	-10.87	-6.153
60	2002	42	1200	13.79	13.59	-3.309	-1.261
60	2002	42	1300	13.71	13.48	-1.099	2.493
60	2002	42	1400	13.82	13.49	-1.355	4.375
60	2002	42	1500	13.61	13.52	-2.932	.077
60	2002	42	1600	13.59	13.53	-1.673	.067
60	2002	42	1700	13.8	13.56	.724	2.963
60	2002	42	1800	13.88	13.59	-2.677	.405
60	2002	42	1900	13.99	13.75	-4.235	-2.941
60	2002	42	2000	13.87	13.8	-6.903	-5.224
60	2002	42	2100	14.05	13.82	-8.06	-7.3
60	2002	42	2200	14.07	13.85	-7.37	-6.424
60	2002	42	2300	14.09	13.86	-7.1	-6.189
60	2002	42	2400	14.12	13.88	-9.6	-8.58
60	2002	43	100	14.16	13.92	-12.5	-11.13
60	2002	43	200	14.16	13.96	-12.09	-10.89
60	2002	43	300	14.13	13.93	-11.14	-9.74
60	2002	43	400	14.12	13.92	-9.69	-8.87

Lines Read: 953

Lines Written: 189

Pause Stop Close



# Split Parameter File Entries

## Input Files

The name of the Input File is entered in the space to the right of the **Browse** button. The default directory is the working directory for Split (if the default installation directories were chosen, this will be c:\campbellsci\splitw). If the input file is not in the default directory, use the **Browse** button to find the input file.

In LoggerNet, mixed array datalogger files are stored in a simple comma separated ASCII format; tabled-based datalogger files are stored in TOA5 (a comma separated format with headers). Split can process Input files from other software, but they must be formatted in Comma Separated ASCII, Final Storage (Binary) Format, Field Formatted ASCII (Split default output format), Printable ASCII, Table Oriented ASCII (TOACII or TOA5) or Raw A/D data (refer to special Burst Mode instruction in your Campbell Scientific datalogger manual).

Files stored in Table Oriented Binary (TOB) format are converted to Table Oriented ASCII files when Split uses them. The converter runs in the background when you run Split to create the output file. You cannot use the Data Label browser to select the columns of data from a binary file. If you want to use the Data Label browser you can open the file first using View, which converts the binary file to ASCII and saves it under a new name, prior to processing it with Split.

Split's default output file, a field-separated ASCII format with a \*.PRN file extension, can be processed a second time if desired.

TABLE 1 provides an example of Comma Separated, Field Formatted, Printable ASCII, and Table Oriented ASCII input file types. The data in the various formats are identical. Each line of data represents an "Output Array", starting with an Output Array ID (in this case 115). Each data point in the Output Array is referred to as an "element". The element number is given in the Printable ASCII format, and implied in the other formats. Data presented in TABLE 1 is used for example purposes in the following sections.

TABLE 1. Comma Separated, Field Formatted, Printable ASCII, and Table Oriented ASCII Input File Format Types						
COMMA SEPARATED						
115,189,1200,89.6,55.3,25.36,270						
115,189,1300,91.3,61.5,27.25,255.4						
115,189,1400,92.7,67.7,15.15,220.1						
115,189,1500,94.1,69,20.35,260.6						
FIELD FORMATTED						
115	189	1200	89.6	55.3	25.36	270
115	189	1300	91.3	61.5	27.25	255.4
115	189	1400	92.7	67.7	15.15	220.1
115	189	1500	94.1	69	20.35	260.6

#### PRINTABLE ASCII

```
01+0115 02+0189 03+1200 04+089.6 05+055.3 06+25.36 07+270.0
01+0115 02+0189 03+1300 04+091.3 05+061.5 06+27.25 07+255.4
01+0115 02+0189 03+1400 04+092.7 05+067.7 06+15.15 07+220.1
01+0115 02+0189 03+1500 04+094.1 05+069.0 06+20.35 07+260.6
```

Element 1 = Output Array ID# (115)  
Element 2 = Julian day (189)  
Element 3 = hour, minute  
Element 4 = average temperature in deg. F  
Element 5 = average soil temperature in deg. F  
Element 6 = average wind speed in mph  
Element 7 = wind direction in degrees

#### TABLE ORIENTED ASCII

```
"TOAC11","CR10T","15Minute"
"TMSTAMP","RECNBR","TCTempF_MAX","BattVolt_MIN"
"2002-02-26 10:30:00",0,73.97,13.99
"2002-02-26 10:45:00",1,74.03,13.98
"2002-02-26 11:00:00",2,74.53,13.98
"2002-02-26 11:15:00",3,74.82,13.98
"2002-02-26 11:30:00",4,75.23,13.98
```

Element 1 = Timestamp  
Element 2 = Record Number  
Element 3 = temperature in degrees F  
Element 4 = minimum battery voltage

A maximum of eight input files may be processed by Split at one time. Additional input files are added using the **EDIT | ADD DATA FILE** menu option. Split looks for a file extension of .DAT if no extension is specified. If the Input File does not exist, an error message is displayed when **RUN | GO** is selected from the menu options.

For instance, to process two files named TEST.DAT and TEST\_1.DAT the user would select TEST.DAT and TEST\_1.DAT as Input Files. Two blank input file templates will be generated. To change from one template to the other, click the appropriate tab on the bottom of the screen. Both templates must be completed before Split will process the data. To merge different output arrays from the same input file into one array, open the data file once for each different array.

## File Info

In most instances, Split automatically recognizes the type of data file it is reading when using Auto Detect in the File Info field. However, there are two exceptions for which you should choose the appropriate option manually:

- **Reading Raw A/D Data from Burst Measurements**

To read this type of data and convert it to ASCII, select **Burst Format** in the File Info box. Once Burst Format is selected, the Number of Values in Each Burst window in the Offset Menu will become accessible. Enter the number of elements in each Burst. This number does not include the array ID number or calibration data.

- **Reading Data in Final Storage (Binary) Format**

If the data is in binary format and Start and Stop Offsets are used, Final Storage (Binary) Format must be selected in the **File Info** field. This tells Split that the file must be decoded as Final Storage before counting the bytes. If Offsets are not used, Auto Detect may be chosen and the file will be processed correctly.

## File Offset/Options

### Start Offset

#### None

Select this check box to start reading the input file from the beginning.

#### Last Count

Each time Split runs a parameter file, it keeps track of the number of bytes it read from the input file and saves this information in the parameter file. Split can then start where it last left off. This is done by clicking the **Offsets** button and selecting the **Last Count** option. This feature may be used to process only the new data from a file in which new data are being appended periodically to the data file.

The screenshot shows the 'Offsets' dialog box with a blue title bar. Inside, there are several sections: 'Start and Stop offsets (Specified in number of bytes)' with 'Start Offset' (radio buttons for None, Last count, Specific) and 'Stop Offset' (text box); 'Number of values in each burst' (text box); 'Start - Stop condition' (checkbox for 'Midnight is 2400 hours'); and 'Time Offset (Seconds)' (text box). There is also an 'Align Array' checkbox. On the right side, there are 'Ok', 'Cancel', and 'Help' buttons.

---

### CAUTION

When using the Last Count option, if the Start and Stop Conditions are specified, they must exist in the newly appended data or Split will never begin execution.

Because Last Count keeps track of the number of bytes in the file, if you delete data from the beginning of a file, Last Count will not work properly.

---

### **Specific**

By selecting the **Specific** option and entering a number, Split will “seek” that position in the file. This option saves time by starting (or stopping) part way through a large data file. The number specifies the number of bytes into the file to seek before processing data. A positive or negative number can be entered. If the number is positive, Split will start reading from the beginning of a file; if the number is negative, Split will start reading from the end of a file. All characters, including spaces, carriage returns, and line feeds, are counted.

In the following figure, Split will skip the first 256 bytes of data before it begins processing the data in Input File.

The screenshot shows the 'Offsets' dialog box. Under 'Start and Stop offsets (Specified in number of bytes)', the 'Start Offset' section has three radio buttons: 'None', 'Last count', and 'Specific' (which is selected). Next to 'Specific' is a text box containing the number '256'. Below this is an 'Align Array' checkbox. To the right is a 'Stop Offset' text box. Below these are three more sections: 'Number of values in each burst' with an empty text box, 'Start - Stop condition' with a checkbox for 'Midnight is 2400 hours', and 'Time Offset (Seconds)' with an empty text box. On the right side of the dialog are three buttons: 'Ok', 'Cancel', and 'Help'.

### **Align Array**

When using a specific start offset, the number of bytes specified may cause Split to seek to the middle of a row. Selecting the **Align Array** check box will cause Split to begin processing at the beginning of the next row.

### **Stop Offset**

This number specifies the number of bytes from the beginning of the file that Split should stop processing the data file.

In the following figure, Split will skip the first 256 bytes of data before beginning and stop execution on byte 1024.

**Offsets**

Start and Stop offsets (Specified in number of bytes)

Start Offset

☐ None

☐ Last count

☒ Specific 256

☐ Align Array

Stop Offset 1024

Number of values in each burst

Start - Stop condition

☐ Midnight is 2400 hours

Time Offset (Seconds)

Ok

Cancel

Help

### ***Number of Values in Each Burst***

When processing a burst data file, enter the total number of values recorded for each Burst (this is the number of burst scans multiplied by the number of channels per scan). This number does not include the array ID or calibration data.

To break the results into a column for each channel, enter the number of channels for the Break Arrays value (**Output File Tab, Other** button).

### ***Midnight is 2400 hours***

When programming mixed-array dataloggers, the Real Time instruction (P77) has two different options for the midnight time stamp: midnight = 2400 of the day just ending or midnight = 0000 of the day just beginning.

When processing mixed-array data files using time synchronization, select this check box if the time stamp is midnight at 2400 of the day just ending. This will ensure that Split processes the data file correctly.

### ***Time Offset***

This field specifies a time offset, in seconds, that should be applied to each item on the Select line that uses the Date or Edate function to output a date. The offset can be positive or negative. Each input file can have its own offset (or no offset) for its Select line.

For example, with an input timestamp of “2008-10-09 10:25” and an offset of 3600, the timestamp output by Date(“yyyy-mm-dd hh:nn”;1;1;1) would be “2008-10-09 11:25”.

This may be useful when adjusting for different time zones.

---

**NOTE** The offset will not be applied to Date and Edate functions with only two parameters. (The two-parameter mode is backwards compatible with the original Date and Edate functions used in older versions of Split.)

---

## Start Condition

A starting point may be specified to begin processing data. If the Start Condition field is left blank, Split will start processing data at the beginning of the data file. The starting point can be any element within the array or a combination of elements within an array.

---

**NOTE** The font for Start Condition, Stop Condition, Copy, and Select can be changed from the Options Menu.

---

The syntax can be expressed as:

$$e_i[val_i]$$

where  $e_i$  = the position number of the element within the array

$val_i$  = the value of that element.

For example, the data in TABLE 1 contains seven elements per Output Array, representing hourly data. Assume that this data file contains one month of hourly data. To start processing data at 1500 hours on the first day, the Start Condition is expressed as 3[1500], where 3 means the third element within the array and 1500 is the value of that third element.

The element must match this start value exactly to trigger the start condition. However, when starting based on time, you can enable the “Start-Stop On/After Time” function to trigger the start of processing when the exact time is found or at the first instance of data after that time has occurred. This option is found on the **Output** tab, **Other** button.

---

**NOTE** Table data files contain the time and date as a single quoted string at the beginning of each data record. Split handles the dates as long as you include a colon separator as a placeholder for each of the fields in the timestamp. 1[Year]:1[Day of Year]:1[Time of Day]:1[Seconds]

See the examples below:

:1[60]:: Day of Year 60

1[2002]:1[60]:1[1250]: Year 2002, Day of Year 60, Time of Day 12:50

::1[1445]:1[30] Time of Day 14:45, Seconds 30

---

Logical “and” and “or” statements can be used when specifying the Start Condition. A logical “and” statement means that all conditions must be true for the statement to be true. Up to three conditions can be connected with “and” statements. If too many “and” statements are used, an error message will be displayed when you run Split.

The logical “or” statement means that if *any* of the conditions are true, then the statement is true. Split allows up to six conditions to be connected with “or” statements. Additionally, each “or” statement can contain up to three “and” conditions. As with the “and” statements, if the maximum number of valid statements is exceeded, an error message will be displayed.

These rules for logical statements also apply to the Stop and Copy Conditions.

An example of a simple logical “and” statement follows:

2[189]and3[1200]

Element two (the Julian day) must equal 189, and element three (the time in hours/minutes) must equal 1200.

If the following “and” statement was used:

2[189]and3[1200]and4[92]and5[67]

an error would be returned because the maximum number of allowable “and” statements has been exceeded.

A range can be specified for val<sub>i</sub> by putting “..” between the lower and upper limit. For example:

2[189]and7[200..275]

In this example two conditions must be satisfied to start processing data. First, the day of year must be 189, and second, element 7 must be between 200 to 275 degrees, inclusive.

### **Starting Relative to PC Time**

Split has the ability to start relative to the current PC TIME (computer time). This feature allows a .PAR file to be run on new data files without changing the Start Conditions, provided the Input Data File is collected at a fixed interval and Split is run at a fixed interval. For example, the same PAR file could be run every day to display the last 48 hours of data without changing the start conditions. For example, using a table based data file:

Start Condition = 1:1[-1]:1[1200]:1:

In this instance, Split will begin processing data when the date for both files is one less than the current date (1:1[-1]:1[1200]:1:) and the time is 1200 (1:1[-1]:1[1200]:1:).

As an expanded example, assume that LoggerNet is used to append data to an archive file. SplitR is executed using a desktop shortcut. In this case the frequency of data collection and data reduction is the same. Time values in the data file (day, hr, mn, sec.) are different each time the data are collected, but by telling Split where to Start reading relative to the PC clock, the Start Conditions do not need to be changed. To accommodate variations in the data collection and reduction frequencies, an interval in minutes or seconds may be specified as shown in the examples below.

**2[-0]:3[-60,5]** tells Split to start at a timestamp in the data that is between 55 and 65 minutes prior to the current PC time (the closest 5 minute interval of the current day that is less than the PC time minus 60 minutes). If you are processing data stored at the top of the hour and the PC time is 1404, Split calculates 1304 and looks for hour 1300 to start reading.

**2[-3]:3[-120,60]** tells Split to find the closest 60 minute interval that is less than the PC time minus 3 days and 2 hours. If the PC time is the day of year 159, hour 0017, Split will start reading on data output at 2200 hours on day 155.

**2[-3]:3[-120]:4[20,5]** tells Split to find the closest 5 second interval that is less than the PC time minus 3 days, 2 hours and 20 seconds. If the PC time is 27 seconds after noon on day 30, Split will begin reading on data output at 1000 hours and 05 seconds on day 27.

Split can also begin processing a file on a particular month and day. Use the syntax **:E[Month%Day]::**, where E is the element that contains the Julian Day, and Month and Day are either constants or a value related to PC time. For example:

**:2[-1%1]::** tells Split to begin processing on the first day of the previous month.

**:2[-0%15]::** tells Split to begin processing on the fifteenth day of the current month.

**:2[5%1]::** tells Split to begin processing on May 1.

This function can be used in both the Start and Stop conditions. It provides a simple way to create a monthly report. For additional information, refer also to *Using Time Synchronization While Starting Relative to PC Time* (p. 37).

---

**CAUTION**

Split will not start reading if the exact specified starting time cannot be found, unless you enable the “Start-Stop On/After Time” feature. The interval (5 minutes, 60 minutes, and 5 seconds in the examples above) must be evenly divisible into 60 minutes.

---

---

**NOTE**

- If the start time is a certain number of days prior to the PC time, the file will be processed beginning at midnight of the day specified.
  - To specify a start time in minutes from the current PC time, you must also specify a day parameter of [-0]. Otherwise, processing will begin at the first instance in the data file that the minutes parameter equals the current minutes.
- 

## Stop Condition

The Stop Condition specifies when to stop processing data. This feature allows segments of data to be removed from large data files. For instance, if a data file contains one month of data and just one day is desired, the start and stop values allow the user to get just that day’s data.



The Stop Condition is expressed with the same syntax as the Start Condition. If the Stop Condition parameter is left blank, Split will execute until the end of the file. As with the Start Condition, logical “and” and “or” statements can be used when specifying the Stop Condition (*Start Condition (p. 12)*), as well as stopping based on PC time.

The array or record containing the Stop Condition is not included in the output file. If the stop value is not found, Split will display a dialog box that gives the option to select a new file and continue processing the data. This feature is useful when data are contained in more than one data file.

The “Start-Stop On/After Time” function can be used with a Stop Condition. This will stop processing of the file when the exact time is found or at the first instance of data after that time has occurred. This option is found on the **Output** tab, **Other** button.

The C and F commands alter the meaning of the Stop Condition.

### **“C” Option: Formatting Event Tests Containing Conditional Output Arrays**

The C option is used to combine data from two or more conditional arrays onto one Split output line. A conditional array is one that is only output when a defined event occurs.

Assume that two or more conditional Output Arrays with unique Output Array IDs compose a test period, followed by an unconditional Output Array that defines the end of a test. The unconditional “end of test” Output Array is at the end of each test, but the conditional Output Arrays may or may not be present. The data file is comprised of several of these tests.

As an example, let’s look at a vehicle test application. The start of the test is when the vehicle is turned on, and the end of the test is when the vehicle is turned off. The conditional output arrays could be:

- monitoring the engine temperature and outputting data to a unique array when the temperature exceeds a limit
- outputting data to a unique array when the brakes are applied
- outputting data when engine RPM exceeds a limit

The unconditional array data (the stop condition) would be output to a unique array when the engine is turned off. By processing the data with Split using the C option, the data collected during each test could be merged on to one line, with blanks inserted if a set of data didn’t exist (e.g., if the engine temperature never exceeded the defined limit).

- An Input File must be set up for each array ID in the test. The first Input File is configured on the **Input File** tab that appears when you open Split. Additional Input Files are added by choosing **Edit | Add Data File** from the Split menu. The same data file will be used as the Input File for each array.
- Type in the array ID in the Copy field of the **Input File** tab for each array. The array ID is the first element of a data file, so the line should read 1[123], where 123 is the actual array ID you want to process.

- In the Select field, type in the number for each element (data value) you want to be output in the report.
- In the Stop Condition field, type in a “C,” followed by the ID of your stop condition array. If your “end of test” array was array ID 200, the Stop Condition field would read: C,1[200]. This should be typed into the Stop Condition fields of each array, including the “end of test” array.

Set up the Output File as you would for any Split process. If you are including column headings, the arrays and elements will appear in the order they are listed on the Input File tabs. That is, the first column will be Input File number 1, element number 1; the next column is Input File number 1, element number 2... Input File number 2, element number 1 follows in the column immediately after the last element of Input File number 1.

Consider TABLE 2 below:

TABLE 2. Example of Event Driven Test Data Set	
100,12.1,10.,32.6 101,92.7,67.7 102,56.1,48.7,98.,220.1 200	Data from arrays output during the first test.
100,12.5,9.89,30.1 102,56.2,50.,100.5,210.6 200	Second test.
100,13.1,10.1,33.1 101,94.1,69 200	Third test.

This table contains four different output arrays: 100, 101, 102, and 200. During the first test, data was output from all three conditional arrays (100, 101, and 102), with 200 signaling the end of the test. During the second test, data was output from arrays 100 and 102. During the third test, data was output from arrays 100 and 101.

To process these files using the C option, the parameter file would be set up as follows (assuming the name of our data file is Data\_1.DAT):

First Input File = Data\_1.DAT:1  
 Stop condition = C,1[200]  
 Copy = 1[100]  
 Select = 1,2,3,4

Second Input File = Data\_1.DAT:2  
 Stop condition = C,1[200]  
 Copy = 1[101]  
 Select = 1,2,3

Third Input File = Data\_1.DAT:3  
 Stop condition = C,1[200]

Copy = 1[102]  
Select = 1,2,3,4,5

Fourth (“end of test”) Input File = Data\_1.DAT:4  
Stop condition = C,1[200]  
Copy = 1[200]  
Select = (leave blank)

**NOTE**

The *.(number)* after the data file name is inserted automatically by Split.

TABLE 3. Processed Data File Using Option C											
100	12.1	10	32.6	101	92.7	67.7	102	56.1	48.7	98	220.1
100	12.5	9.89	30.1				102	56.2	50	100.5	210.6
100	13.1	10.1	33.1	101	94.1	69					

When Split is run, the resulting data file will look similar to TABLE 3. Each line of data represents one test. Notice that blanks were inserted if the data set (conditional array) did not exist.

**Trigger on Stop Condition (F Option) Output of Time Series**

The Trigger on Stop Condition, or F option, changes the function of the Stop Condition when one or more Time Series functions (*Time Series Functions, Details, and Examples* (p. 23)) are contained in the Select field. When a Stop Condition is met, the time series data is calculated and written to the output file. However, instead of stopping at this point, processing resumes and time series data is output the next time the Stop Condition is met. This continues until the end of file or until the user stops Split manually.

The Trigger on Stop Condition is enabled by clicking **Other...** on the **Output** tab and checking the box next to the Trigger on Stop Condition field. When the Trigger on Stop Condition is enabled, the function affects all files being processed that have a Stop Condition specified. If multiple files are being processed but it is desired that the function affect one or more—but not all—of the files, the F option is used in the Stop Condition field of the files that you want processed using the function. The syntax for the F option is: F,e<sub>i</sub>[val<sub>i</sub>].

A typical application for the Trigger on Stop Condition is to reduce days of hourly data into daily summaries. A logical element to use for the Stop Condition is time (hrmn). Assuming the third element of the hourly Output Array is hrmn, and midnight is output as 0, the Stop Condition is entered as 3[0] (or F,3[0] if the F option is used). The Time Series processing is performed over a day defined by midnight to midnight.

If only hourly Output Arrays were contained in the Input File, the Copy line could be left blank. If other Output Arrays are present which need not be included in the Time Series processing, a logical Copy condition would be the Output Array ID of the hourly output.

The Trigger on Stop Condition functions the same for multiple Input files as it does for a single Input File. If the option is enabled on several Input Files, and

the Stop Conditions do not occur at the same point in each file, when a file's Stop Condition is met, its time series data are output and blanks are output for data selected from the other Input Files.

Say, for example, that you were interested in the average value of the first data point (element 2) for each test, in the data set listed in TABLE 2. The Input File template would look like that shown in TABLE 4.

TABLE 4. Input File Entries to Process the First Data Point for each Test
First Input File = DATA_1.DAT:1 Stop Condition = F,1[200] Select = AVG(2)

Copy

The Copy Condition tells Split which arrays should be used for the output data. After the Start Condition is satisfied, and before the Stop Condition is met, the Copy condition must be satisfied before any data will be processed according to Select line instructions. If the Copy condition is left blank, all arrays are processed between the Start and Stop values. Syntax for the Copy condition is similar to the Start and Stop values mentioned above. Logical “and” and “or” statements (see *Start Condition* (p. 12)) can be used when specifying the Copy condition.

For example, referring to TABLE 1, if only those hours during day 189 when the temperature was above 90 and the soil temperature was below 62 is desired, or, during day 189 when the average wind speed was below 21 while the wind direction was between 255 to 265 is desired, the Copy condition would be:

1[189]and4[90..150]and5[0..61.99]or1[189]and6[0..20.99]and7[255..265]

Only Output Arrays with hours 1300 and 1500, TABLE 1, conform to the above Copy conditions.

NOTE

The Copy Condition is used almost exclusively for mixed-array dataloggers, except when time-syncing two or more data files. See *Time Synchronization* (p. 35), for additional information.

Time Ranges

When specifying a Copy condition, a range of time values can be specified instead of a single time. If the element being tested falls within the range, the Copy condition is satisfied and the data is processed. A range is indicated by entering two periods between the first and last values of the range.

### *Examples:*

#### **Table-based**

With an entry of **1:1:1[600..1200]:1** in the Copy condition, Split will only process the data file when the time is between 6:00 a.m. and 12:00 p.m.

(Since the timestamp for table-based dataloggers is all one string, each portion of the timestamp (year, day, hour/minute, seconds) will use the same element number. Colons are used to separate each portion. The format is 1[year]:1[day]:1[hhmm]:1[seconds] (the number 1 was used since, typically, the timestamp is the first element in the data string). In this format, hhmm is the four-digit hour/minute.)

#### **Array-based**

With an entry of **1[30] and 2:3:4[600..1200]:** in the Copy condition, Split will only process the data file when the time is between 6:00 a.m. and 12:00 p.m.

(This assumes 2 is the year element, 3 is the day element, and 4 is the hour/minute element.)

---

#### **NOTE**

---

Time ranges cannot be used with the time-sync function.

---

## **Select**

The Select line specifies which elements of an Output Array are selected for processing and/or output to the specified Output File. The Select line becomes operable only after the Start Condition and Copy condition are met, and before the Stop Condition is satisfied. If the Select line is left blank, all elements in output arrays meeting the Start Condition and Copy conditions are output to the Output File.

Processing is accomplished through arithmetic operators, math functions, spatial functions, and time series functions.

## **Ranges**

Element numbers may be entered individually (e.g., 2,3,4,5,6,7), or, in groups (e.g., 2..7) if sequential. Range limits (lower to upper boundary conditions) may be placed on elements or groups of elements specified in the Select or Copy lines. For example, 3[3.7..5],4..7[5..10] implies that element 3 is selected only if it is between 3.7 and 5, inclusive, and elements 4,5,6, and 7 must be between 5 and 10, inclusive.

If range limits are used in the Select condition, when Split is run, any data which are outside of the specified range will be highlighted according to the options chosen for the output file. TABLE 5 summarizes what each option produces on the screen and in the output file if out of range data are encountered. This type of range testing is a quick way to identify data problems.

**TABLE 5. Effects of Out of Range Values for Given Output Options**

<b>Output Option</b>	<b>Screen Display*</b>	<b>PRN File</b>	<b>RPT File or Printer Output</b>
Report = None; No other options defined (default)	bad values displayed in red and preceded by asterisk; the text “bad data” highlighted in a red box at bottom right of screen	blanks inserted for bad values	N/A
Report = File or Printer; no other options defined	bad values displayed in red and preceded by asterisk; the text “bad data” highlighted in a red box at bottom right of screen	blanks inserted for bad values	bad values preceded by asterisk
Report = None; replacement text (abc) in “Replace bad data with” field	bad values displayed in red and preceded by asterisk; the text “bad data” highlighted in a red box at bottom right of screen	abc inserted in place of bad values	N/A
Report = File or Printer; comment in “Replace bad data with” field	bad values displayed in red and preceded by asterisk; the text “bad data” highlighted in a red box at bottom right of screen	comment inserted in place of bad values	bad values preceded by asterisk
Report = None; “Display only bad data” option enabled	only lines with bad data are displayed; bad values displayed in red and preceded by asterisk; the text “bad data” highlighted in a red box at bottom right of screen	only lines with bad data output; blanks inserted for bad values	N/A
Report = File or Printer; “Display only bad data” option enabled	only lines with bad data are displayed; bad values displayed in red and preceded by asterisk; the text “bad data” highlighted in a red box at bottom right of screen	only lines with bad data output; blanks inserted for bad values	only lines with bad data output; bad values preceded by asterisk
*The Screen Display box must be checked; if not, no data will be displayed on the Split Run screen.			

**NOTE**

In this instance, out of range data refers to data outside of the specified output range. It is not to be confused with out of range data generated by the logger.

**Variables**

Variables can be assigned names in the Select line. For example,  $x = 4 - 5 * (6 * 3.0)$  means that x is equal to element 6, times the number 3, times element 5, subtracted from element 4. A numeric value is distinguished from an array element by the inclusion of a decimal point. Variables must be declared before they can be used in the Select line. A variable name must start with an alpha character, can include numbers and must not exceed eight characters. Variable names can start with the same character but they must not start with another complete variable name (e.g., the variable XY is not valid if there is also the variable X). A comma must follow each variable statement, as with all parameters in the Select line. Once the variables have been declared they can be used later in the Select line (i.e.,  $x = 4 - 5 * (6 * 3.0)$ ,  $y = 6/3, 2, 3, 6, 7 * x, 6 + y$ ).

**NOTE**


---

Variables can be defined in the **first four Input File's Select lines** only, but may be used in subsequent Input File's Select lines.

---

Illegal operations (e.g., logarithm of a negative number) will cause Split to store blanks for the Output. It is possible to get a run time error (error 0/1) if the floating point math exceeds the limits of the PC.

**Numerical Limitations**

The greatest number that can be output is determined by the field width (**Output File** tab). If the width is eleven or greater, the maximum number is 99,999,999; for widths from eight through ten the maximum is 99,999; for widths less than eight the maximum is 9999. If a column is not large enough for a value, it will be stored as a 9,999, 99,999 or 99,999,999 based on the column width. In some instances, such as when a column is not large enough for the date function, you will see the text "bad data" on the Split Runtime window.

**Mathematical Functions, Details, and Examples**

<b>TABLE 6. Split Operators and Math Functions</b>	
<b>OPERATORS</b>	<b>OPERATOR PRECEDENCE ORDER</b> (3 = high, 1 = low)
^ = raise to the power	3
x Mod y = Modulo divide of x by y	2
* / = multiplication, division	2
+ - = addition, subtraction	1
<b>EXAMPLES OF SYNTAX FOR MATHEMATICAL OPERATORS</b>	
3*5	multiply element 3 by element 5
3/5	divide element 3 by element 5
(3..5)/(8..10)	same as 3/8, 4/9, 5/10
3+5	add element 3 to element 5
3-5	subtract element 5 from element 3
(3,9,5)-(8,7,10)	same as 3-8, 9-7, 5-10
3*2.0	multiply element 3 by a fixed number 2
2^3.0	raise element 2 to the third power
<b>MATH FUNCTIONS</b>	
Abs(x)	= Absolute value of x
Arctan(x)	= Arc tangent of x (in degrees)
Cos(x)	= Cosine of x (in degrees)
Exp(x)	= Natural Exponent function (e <sup>x</sup> )
Frac(x)	= Fractional portion of x
Int(x)	= Integer portion of x
Ln(x)	= Natural logarithm of x

Sin(x)	= Sine of x (in degrees)
SpaAvg(x..y)	= Spatial average of elements x through y
SpaMax(x..y)	= Spatial maximum of elements x through y
SpaMin(x..y)	= Spatial minimum of elements x through y
SpaSd(x..y)	= Spatial standard deviation of elements x through y
Sqrt(x)	= Square root of x

The following array of ASCII data will be used for all Mathematical function examples.

0105 0176 1200 -07.89 55.10 12.45 270.5

<b>Abs(x)</b>	returns the absolute, or positive value of element x. Examples: Abs(4) = 7.89 Abs(4*5) = 434.74
<b>Arctan(x)</b>	returns the arc tangent of element x in degrees. Examples: Arctan(7) = 89.788 Arctan(7/6) = 87.365
<b>Cos(x)</b>	returns the cosine of element x in degrees. Examples: Cos(5) = .57215 Cos(5-6) = .73551
<b>Exp(x)</b>	returns the exponential base e to the power of element x. Example: Exp(4) = .00037
<b>Frac(x)</b>	returns the fractional value of the element x. Examples: Frac(4) = -.89 Frac(6+7) = .95
<b>Int(x)</b>	returns the integer portion of the element x. Examples: Int(7) = 270 Int(5*6) = 685
<b>Ln(x)</b>	returns the natural log of element x. Examples: Ln(6) = 2.5217 Ln(7/6*5/1) = 2.4337
<b>Sin(x)</b>	returns the sine of element x in degrees. Examples: Sin(7) = -.99996 Sin(7-2+5) = .50603



Spatial functions, included under Mathematical functions, operate on a per Output Array basis. The average, maximum, minimum, and standard deviation of a specified group of elements within an array are calculated.

**SpaAvg(x..y)** returns the spatial average of elements x through y.

Examples:

$\text{SpaAvg}(1..7) = 258.74$

$\text{SpaAvg}(1,4,7) = 122.54$

**SpaMax(x..y)** returns the maximum value of elements x through y.

Examples:

$\text{SpaMax}(1..7) = 1200$

$\text{SpaMax}(1,2,5) = 176$

**SpaMin(x..y)** returns the minimum value of elements x through y.

Examples:

$\text{SpaMin}(1..7) = -7.89$

$\text{SpaMin}(1,2,5) = 55.1$

**SpaSd(x..y)** returns the standard deviation of elements x through y.

Examples:

$\text{SpaSd}(1..7) = 394.57$

$\text{SpaSd}(5,2,1) = 49.607$

**Sqrt(x)** returns the square root of element x.

Examples:

$\text{Sqrt}(3) = 34.641$

$\text{Sqrt}(3^2) = 1200$

## Time Series Functions, Details, and Examples

TABLE 7. Time Series Functions	
TIME SERIES FUNCTIONS	
Avg(x;n)	= Average
Blanks(x;n)	= Number of blanks in element
Count(x;n)	= Number of data points in element
Max(x;n)	= Maximum
Min(x;n)	= Minimum
RunTotal(x;n)	= Running total
Sd(x;n)	= Standard deviation
Smpl(x;n)	= Sample raw value
SmplMax(x;y;n)	= Sample (y) on a maximum (x)
SmplMin(x;y;n)	= Sample (y) on a minimum (x)
Total(x;n)	= Totalize
WAvg(x;n)	= Unit vector mean wind direction (in degrees)
<b>NOTE:</b> x can be an element or a valid expression. n is optional and is the number of arrays to include in the function. Date and Edate can be used for the “n” in the Time Series functions to produce monthly output (see TABLE 8, Special Functions).	

Time Series functions are used to perform vertical processing on selected elements, such as calculating the average of an element over a specified range of data. Time Series results are output in three instances:

1. when a Trigger on Stop Condition (F option) is met
2. at the end of a data file (or within a range specified by Start and Stop Conditions)
3. when an interval count is met

When the Trigger on Stop Condition (or F option) is used, any time series data defined in the Select line is output each time the Stop Condition is met. Refer to *Trigger on Stop Condition (F Option) Output of Time Series (p. 17)*, for more information on the Trigger on Stop Condition.

Results which are output at the end of a file or a range of data are referred to as Final Summaries. A typical select line that would produce a Final Summary is:

1,2,3,4,Avg(4)

This line would output values for elements 1 through 4 each time an array was output. Additionally, an average value for element 4 would be calculated for the entire file and output as the last line of data in the output file.

1,2,3,4,Avg(4;24)

This line would output values for elements 1 through 4 each time an array was output, and an average value for element 4 would be calculated every 24<sup>th</sup> array and output as an additional column in the file. An additional summary would occur for an Interval Count if the count was not evenly divisible into the number of output arrays present in the Input File. The summary, in this case, is calculated from an incomplete interval count.

The date( ) function can be used for the interval in a time series function to produce monthly output. Refer to the Monthly summary example in *Special Functions, Details, and Examples (p. 28)*.

#### NOTE

---

When Date and Edate are used within other functions they must be used with the older format Date(doy;y) and Edate(doy;y) instead of using the extended date functions. For example AVG(1;Date(2;2002.0)). The decimal is needed to indicate a fixed number. Numbers without the decimal are interpreted as element IDs.

---

The interval count in a Time Series Function is optional and does not require a decimal point. To determine the interval, Split counts the number of arrays which meet the specified conditions (Stop, Start, and Copy). If the time synchronize function is enabled, the Time Series functions remain synchronized to the starting time even if a complete array is missing from the input data. When elements are missing, the Time Series calculations are based on the actual number of elements found.

Semicolons are used in Time Series functions to separate the elements or expressions from the count which determines the interval. SmplMax and SmplMin require two elements separated by a semicolon. The first is checked

for a maximum or minimum, while the second is sampled on the maximum or minimum.

The following set of weather data from Mt. Logan in northern Utah gives a total of seven elements each hour. This Field Formatted output, with title and column headers, was generated by Split. These data are used in the following examples of Time Series functions.

---

**Mt. Logan Weather Data**

<b>Day</b>	<b>Time</b>	<b>Airtemp deg F</b>	<b>RH</b>	<b>Mean Wind Speed mph</b>	<b>Mean Wind Direction</b>	<b>Std Dev of Direction</b>
178	100	58.56	17.42	5.855	338.3	6.562
178	200	57.48	17.65	8.27	344.8	7.51
178	300	56.85	17.76	7.75	330.8	5.065
178	400	56.55	18.89	7.6	319.7	10.93
178	500	56.57	19.6	10.41	307.3	4.23
178	600	55.33	23.32	8.99	317.7	6.258
178	700	55.95	24.79	9.52	322.3	4.609
178	800	58.12	23.98	6.588	315.6	9.43
178	900	59.79	23.46	5.458	312	15.32
178	1000	61.09	24.12	4.622	299.3	18.3
178	1100	61.34	25.03	5.926	303	17.26
178	1200	60.61	27.46	6.815	309.7	18.71
178	1300	61.01	25.44	8.35	310.2	18.37
178	1400	60.93	25.48	10.92	317.5	12.68
178	1500	62.3	23.79	8.43	310.6	19.21
178	1600	63.75	24.31	8.88	321.4	15.22
178	1700	66.15	22.45	7.97	341	17.77
178	1800	67.33	23.06	6.758	344.1	20.74
178	1900	66.59	24.75	7.08	341.8	16.09
178	2000	64.52	26.03	8.76	337.2	14.91
178	2100	59.84	27.45	11.81	305.4	12.36
178	2200	56.19	35.46	15.62	316.7	19.01
178	2300	55.48	38.8	17.12	338.7	11.41
179	0	55.22	37.13	11.86	351.6	8.22

---

**Avg(x;n)**

returns the average of element x over a full data set or every n<sup>th</sup> value.

Examples:

Avg(3) = 59.898 (average daily temp)

Avg(3;4) = 57.36 (average 4 hour temp)

56.493 (average 4 hour temp)

60.708 (average 4 hour temp)

61.998 (average 4 hour temp)

66.148 (average 4 hour temp)

56.683 (average 4 hour temp)

**Blanks(x;n)**

returns the number of blanks or bad data in element x over a full data set or every nth value. Refer to TABLE 9 for definition of blank or bad data. Example:

Blanks(3) = 0 (no holes in data set).

**Count(x;n)** returns the number of data points (non blanks) in element x over a full data set or every n<sup>th</sup> value.  
Example:  
Count(1) = 24 (24 data points in data set).

---

**NOTE**

Blanks and Count are functions designed for checking the integrity of the data file. A common use for these two functions is “100.\*BLANKS(x;n)/BLANKS(x;n)+COUNT(x;n)” which gives the percentage of holes (bad data) in the file.

---

**Max(x;n)** returns the maximum value of element x over a full data set or every n<sup>th</sup> value.  
Examples:  
Max(5) = 17.12 (max WS for day)  
Max(5;12) = 10.41 (max WS for 12 hours)  
17.12 (max WS for 12 hours)

**Min(x;n)** returns the minimum value of element x over a full data set or every n<sup>th</sup> value.  
Examples:  
Min(7) = 4.23 (min std. dev. of WS for day)  
Min(3;8) = 55.33 (min temp for 8 hours)  
59.79 (min temp for 8 hours)  
55.22 (min temp for 8 hours)

**RunTotal(x;n)** returns a running total of element x for every line in the data set. If an n<sup>th</sup> value is specified, a running total will be output every n<sup>th</sup> value.

Example: RunTotal(5) =  
5.85  
14.12  
21.87  
29.47  
39.88  
48.87  
:  
:  
:  
166.76  
182.38  
199.50  
211.36  
211.36

Running total of hourly average wind speed provides up-to-the-hour wind run for that day. Because an n<sup>th</sup> value was not specified, the Final Summary output, which is daily wind, is the same as the “total” output.

**Sd(x;n)** returns the standard deviation of element x over a full data set or every n<sup>th</sup> value.  
Examples:  
Sd(3) = 3.6593 (std. dev. temp for day)  
Sd(3;8) = 1.011 (Sd temp for 8 hours)  
1.1182 (Sd temp for 8 hours)  
4.965 (Sd temp for 8 hours)

**Smpl(x;n)** returns a sample of element x every n<sup>th</sup> value.  
 Examples:  
 Smpl(4;8) = 23.98 (RH every 8 hours)  
               24.31 (RH every 8 hours)  
               37.13 (RH every 8 hours)

**SmplMax(x;y;n)** looks for a maximum value in element x and samples element y when the maximum is found. If an n<sup>th</sup> value is specified then it outputs the sample on a maximum every n<sup>th</sup> value, otherwise it outputs the sample on a maximum at the end of file.  
 Examples:  
 SmplMax(5;(3)) = 55.48 (on max wind speed sample temperature)  
 SmplMax(5;(3,6);8) = 56.57 307.3  
 60.93 317.5  
 55.48 338.7  
 (on max wind speed sample temperature and wind direction every 8 hours)

**SmplMin(x;y;n)** looks for a minimum value in element x and samples element y when the minimum is found. If an n<sup>th</sup> value is specified then it outputs the sample on a minimum every n<sup>th</sup> value, otherwise it outputs the sample on a minimum at the end of file. Examples:  
 SmplMin(3;5) = 11.86 (on min temp sample wind speed)  
 SmplMin(3; (5,6);8) = 8.99 317.7  
 5.458 312  
 11.86 351.6  
 (on min temperature sample wind speed and wind direction every 8 hours)

**Total(x;n)** returns the total of element x over a data set or every n<sup>th</sup> value.  
 Examples: Total(5) = 211.36 (daily wind run)

**WAvg(x;n)** Returns the unit vector mean wind direction in degrees of element x (wind direction in degrees) over a full data set or every nth value.  
 Example:  
 WAvg(6) =  
           323.14 (mean wind direction for the day)  
 WAvg(6;4) =  
           333.41 (mean wind direction for 4 hours)  
           315.73 (mean wind direction for 4 hours)  
           306 (mean wind direction for 4 hours)  
           314.92 (mean wind direction for 4 hours)  
           341.03 (mean wind direction for 4 hours)  
           328.09 (mean wind direction for 4 hours)

## Special Functions, Details, and Examples

TABLE 8. Split SPECIAL FUNCTIONS	
<b>Crlf</b>	= Insert carriage return line feed in Output File.
<b>Date("format"S;H;D;Y)</b>	= Convert day of year and time to a timestamp with calendar date and time, where format uses Windows conventions to specify output format. S=seconds, H = HoursMinutes, D = Day, Y = year. The output timestamp is quoted text. Date can be used to create monthly time series summaries.
<b>Edate("format"S;H;D;Y)</b>	= The same as the Date function except that the output text is not quoted. EDate can be used to create monthly time series summaries.
<b>"Label"</b>	= Insert Comment in Output file. (Label is anything within the quote marks.)
<b>Line</b>	= Number of lines written to Output file.
<b>smpl(.pa;n)</b>	= Page break such that n is the number of lines per page for the printer or the .RPT file.
<b>PCdate or PCedate</b>	= Used in a report header to print the current date.
<b>WDQ(n)</b>	= Outputs the wind direction using an alphabetical abbreviation, based on 8 quadrants.
<b>WDQS(n)</b>	= Outputs the wind direction using an alphabetical abbreviation, based on 16 quadrants.

The Mt. Logan data set is used for the Special Function examples. These functions are helpful in converting time fields to formatted timestamps and formatting the output. Since one of the main differences between mixed-array data files and table based data files is the time format, these functions can be used to convert between file types.

### NOTE

If you are processing the data file in multiple passes including formatting of the date and time fields, you should put the date processing in the final pass. Split cannot read all of the timestamp formats that it can produce. For example, the quoted timestamp in table based data files has a specific structure. Any changes to the structure will make the timestamp unreadable for Split.

**Crlf** returns a carriage return and line feed where the Crlf is placed in the parameter file.  
 Examples:  
 Smpl("Max Temp";24),Max(3;24),  
 Smpl(Crlf;24),Smpl("Max RH";24),Max(4;24)  
 = Max Temp 67.33  
 Max RH 38.8  
 The Crlf is placed after the maximum temperature 67.33 so that the maximum RH is on the next line.

**NOTE**

---

A carriage return/line feed is recognized by Split as an element, and may throw the column headers off in the output file.

---

**“Label”**

returns a comment in the output file. This is a useful formatting function when labels are desired on the same line as the data. The label includes anything within the quote marks, the quote marks are not output but must be in the parameter file. The label cannot exceed the width of the output column (default is eight characters). A maximum of thirty (30) labels are allowed per Select line.

Make sure that the column widths are big enough for the label to fit. Otherwise the output will indicate Bad Data.

Examples:

“Max Temp” =

Max Temp (outputs Max Temp  
Max Temp 24 times)

.  
.  
.

Max Temp

Smpl(“8 hour “;8),Smpl(“Max Temp”;8), Max(3;8) = 8 hour  
Max Temp 58.56

8 hour Max Temp 63.75

8 hour Max Temp 67.33

This example samples the labels called “8 hour” and “Max Temp” and looks for a Maximum temp for every 8 hour interval.

**Line**

numbers each line written to the report file or printer. This differs from the Count function in that Count looks at how many lines were read.

Examples:

Line, 4, 5 =

1	17.42	5.855
2	17.65	8.27
3	17.76	7.75
4	18.89	7.6
5	19.6	10.41
6	23.32	8.99
7	24.79	9.52

.  
.  
.

19	24.75	7.08
20	26.03	8.76
21	27.45	11.81
22	35.46	15.62
23	38.8	17.12
24	37.13	11.86

Smpl (Line;8), Smpl (4;8), Smpl (5;8)

1	23.98	6.588
2	24.31	8.88
3	37.13	11.86

**smpl(.PA,n)** Outputs the data to the printer or .RPT file with **n** lines per page.

Examples:

2, 3, Smpl (.PA;12) =

100	58.56
200	57.48
.	.
.	.
.	.
1100	61.34
1200	60.61
1300	61.01
1400	60.93
.	.
.	.
.	.
2300	55.48
0	55.22

**WDQ(n)** Outputs the wind direction using an alphabetical abbreviation, based on 8 quadrants (N, S, E, W, NE, NW, SE, SW). **n** is an element containing wind direction. For example, if **n** = 182, S would be returned in the output file.

**WDQS(n)** Outputs the wind direction using an alphabetical abbreviation, based on 16 quadrants (N, S, E, W, NE, NW, SE, SW, NNE, ENE, ESE, SSE, SSW, WSW, WNW, NNW). **n** is an element containing wind direction. For example, if **n** = 111, ESE would be returned in the output file.

**Date("format"; S; H; D; Y)** Converts a datalogger's time stamp to a different format and encloses it in double-quotes (edate will produce a date without quotes). "Format" is a string which identifies how the date should be output. The "format" string is similar to the date format used by Windows. See the online help in Split to get a complete list of the format parameters.

S is the element number that contains seconds; H is the element number that contains hours/minutes; D is the element number that contains day; and Y is the element number that contains the year. A constant can be used in place of any of the element numbers (the constant must be a valid value for the type of date field and include a decimal point; e.g., 2000.0 for the year). If only three elements are specified, these will be assumed to be hour/minute, day, and year.



When using the Date function for a table-based datalogger (e.g., a time stamp in the format “2002-02-03 21:16:00”), if the time stamp is the first element in the array, a 1 is used for all of the time stamp elements (S; H; D; Y).

If “serial” is entered for the “format” string, a serial date will be output. Other special functions are “hourarray” and “dayofyear”. Both of these are used when processing data from table-based dataloggers so that the timestamps are similar to that of mixed array dataloggers. Hourarray changes a 0000 hourly timestamp to 2400, and dayofyear produces a Julian Day.

In older versions of Split, the date( ) and edate( ) functions were limited to converting the Julian day to a MM-DD format, with a syntax of date(doy;y) where doy = the element number for the day of the year; y = the element number for the year. This older format is still supported.

## NOTE

Split will mark the date as Bad Data if the time and date resulting from the conversion will not fit in the specified column width. The on-screen display and the report file will precede the date with asterisks. In the .PRN output file, Split uses the Bad Data string.

When Date and Edate are used within other functions they must be used with the older format Date(doy;y) and Edate(doy;y) instead of using the extended date functions as shown in the table. For example AVG(1;Date(2;2002.0)). The decimal is needed to indicate a fixed number. Numbers without the decimal are interpreted as element IDs.

## Date Format Examples

Assume that in a mixed array data file, element 2 is Year, element 3 is Day of Year, element 4 is Hour/Minute, and element 5 is Seconds.

<u>String Entered</u>	<u>Output</u>
date(“mm/dd/yy, h:nn”;5;4;3;2)	“02/25/02, 4:10”
edate(“mm/dd/yy, hh:nn”;5;4;3;2)	02/25/02, 04:10
edate(“dddd, mmmm d, yyyy”;5;4;3;2)	Monday, February 25, 2002
edate(“‘Date:’ mmm d, yyyy”;5;4;3;2)	Date: Feb 25 02

If a time element is missing from a mixed array data file, use a valid constant instead.

If processing a table-based data file, use a 1 for all time elements (assuming the time stamp is the first element in the data file). For the examples above:

date(“mm/dd/yy, h:nn”;1;1;1;1)	“02/25/02, 4:10”
--------------------------------	------------------

edate("mm/dd/yy, hh:nn";1;1;1;1) 02/25/02, 04:10

edate("yyyy", "dayofyear", "hhnn";1;1;1;1) 2002, 56, 0410

Notice that this last example essentially creates an array-type of timestamp.

#### NOTE

When processing a data file from a mixed array datalogger, if the time stamp uses midnight as 2400 with "today's" date, the date function will convert that time stamp to 0000 hours with "tomorrow's" date. The "No Date Advance" function can be used to stop the date from rolling forward (**Other** button, **No Date Advance** check box).

**edate("format"; S; H; D; Y)** edate( ) functions identically to date( ) above, except that the time stamp is not surrounded by quotes.

#### Monthly Summary Example

The Date function can be used to produce a monthly summary of daily time series data by using Date( ) for the interval in the time series function. This will trigger time series output for the first day of each month. The syntax is avg(7;date(3;2)), where you want to take a monthly average of element 7, and the day of year is contained in element 3 and the year in element 2. If you have data recorded on a once per minute or once per hour basis, it must first be processed into a 24 hour summary for this function to produce the output expected.

#### NOTE

When Date and Edate are used within other functions they must be used with the older format Date(doy;y) and Edate(doy;y) instead of using the extended date functions. For example AVG(1;Date(3;2)). When used with table based data files the format would be AVG(1;Date(1;1)).

When producing a monthly summary and outputting the month along with the data, you might want to set up the value for the month as "month -1", to correctly reflect the month that the data actually represents.

#### Split Functions Example

The following is a parameter file that operates on the Mt. Logan data with several of the Split features being utilized. This first screen shows the input file and the select criteria that were programmed. This example does calculations based on temperature and wind speed to determine the wind chill.

**Split Win - Splitman.PAR**

File Edit Labels Run Printer Help

Input File(s) Output File

Input Data File

Browse C:\Campbellsci\LoggerNet\MSTSPLIT. File Info Auto Detect Offsets / Options

Start Condition

Stop Condition

Copy 1[10]

Select

$t = (3 - 32.) / 1.8, v = (5 * 0.447), h = (\text{sqrt}(100. * v) + 10.45) * (33. - t), Te = 33. - h / 22.066, \text{smpl}(t;1), \text{smpl}(v;1), \text{smpl}(h;1), \text{smpl}(Te;1), \text{smpl}(Te;1) * 1.8 + 32$

MSTSPLIT.dat:1

The following screen shows the output file setup including the column headings and the units.

**Split Win - Splitman.PAR**

File Edit Labels Run Printer Help

Input File(s) Output File

Output Data

File Browse hourly.prn File Format Field Report File Printer None Other..

☒ Screen Display Column Widths 8

Report and Column Headings

Report Heading Hourly Data

Column#	1	2	3	4	5	6	7
Element/Field#	smpl(t;1)	smpl(v;1)	smpl(h;1)	smpl(Te;1)	smpl(Te;1)*1.8		
Filename	MSTSPLIT.dat	MSTSPLIT.dat	MSTSPLIT.dat	MSTSPLIT.dat	MSTSPLIT.dat		
Line 1	Temp	Wind	H	Wind	Wind		
Line 2	deg C	Speed		Chill	Chill		
Line 3		m/s		deg C	deg F		
Decimal							
Width							

Time Series Heading Insert Delete Add

This .PAR file produces a wind chill summary of the Mt. Logan Peak data set. The formula for calculating wind chill is given as follows:

$$T_e = 33 - (h/22.066)$$

where

$T_e$  = Wind Chill equivalent temperature, degrees C

$$h = ((100V)^{0.5} + 10.45 - V)(33 - T)$$

where

$h$  = Kcal  $m^{-2} hr^{-1}$  wind chill index

$v$  = wind speed in meters/second

$T$  = temperature in degrees C

Note that at wind speeds between 0 to 4 mph (0 to 1.8 m/s), the wind chill should be ignored because this formula results in wind chill temperatures that are greater than the ambient temperature. The National Weather Service includes wind chill in reports only when temperatures drop below 35°F (1.7° C).<sup>1</sup> The formula is for example purposes and is not endorsed by Campbell Scientific as a standard.

When this .PAR file is executed, the following output is displayed on the screen.

Wind Chill Report from Mt. Logan					
Temp deg C	Wind Speed m/s	H	Wind Chill deg C	Wind Chill deg F	
14.756	2.6172	438.06	13.148	55.666	
14.156	3.6967	489.58	10.813	51.463	
13.806	3.4643	491.34	10.733	51.319	
13.639	3.3972	493.4	10.64	51.151	
13.65	4.6533	529.57	9.0005	48.201	
12.961	4.0185	530.58	8.9547	48.118	
13.306	4.2554	528.27	9.0596	48.307	
14.511	2.9448	456.04	12.333	54.199	
15.439	2.4397	414.97	14.194	57.55	
16.161	2.066	383.21	15.633	60.14	
16.3	2.6489	402.08	14.778	58.601	
15.894	3.0463	425.2	13.731	56.715	
16.117	3.7325	439.59	13.078	55.541	
16.072	4.8812	468.26	11.779	53.202	
16.833	3.7682	421.85	13.882	56.988	
17.639	3.9694	405.59	14.619	58.314	
18.972	3.5626	361.39	16.622	61.92	
19.628	3.0208	331.76	17.965	64.337	
19.217	3.1648	345.62	17.337	63.207	
18.067	3.9157	393.08	15.186	59.335	
15.467	5.2791	493.51	10.635	51.142	
13.439	6.9821	584.71	6.5016	43.703	
13.044	7.6526	607.86	5.4526	41.815	
12.9	5.3014	566.29	7.3368	45.206	

#### Reference

<sup>1</sup> "Wind Chill Errors", Edwin Kessler, Bulletin of the American Meteorology Society, Vol. 74, No. 9, September 1993, pp 1743-1744.

## Summary of Select Line Syntax Rules

- A fixed numeric value must include a decimal point “.” or be in scientific notation. There are some exceptions to this as noted below.
- Scientific notation has the format “mantissa E power of ten” (e.g.,  $3E5 = 3 \times 10^5$ ).
- Element numbers are entered without a decimal point.
- Commas separate Select line parameters (e.g., 2,3,(3+4)/3.2,6).
- Two decimal points are used to select consecutive elements between starting and ending elements (e.g., 3..6, refers to the elements 3,4,5, and 6).
- A set is a group of two or more elements and/or expressions separated by commas and enclosed by parentheses. No member of a set can include parentheses. Therefore, a set cannot include a set or a function as one of its members. For example:

### VALID EXPRESSION

Arctan (2/3)  
Arctan (2/3, 3/4, 4/5)  
Arctan (COS(2))

### INVALID EXPRESSION

Arctan ((2/3))  
Arctan ((2/3, 3/4), 4/5)  
Arctan (COS(2), COS(3))

- A single expression can operate on a set of elements. For example, the expression (3..6,8)/2.0 is the same as 3/2.0, 4/2.0, 5/2.0, 6/2.0, 8/2.0; (3..6)/(2..5) is the same as 3/2, 4/3, 5/4, 6/5.
- The element or expression that is the argument of a math or Time Series function, must be enclosed in parentheses. A range of elements can be specified, resulting in as many outputs as elements (e.g., Avg(3..5,7) will output 4 averages).
- Square brackets are used to enclose an allowable range for a value (e.g., 3[3.6..12] ) to indicate that the allowable range for element 3 is from 3.6 to 12. Whole numbers within brackets do not require a decimal point. TABLE 5 explains how values outside the specified range are treated.
- The interval in a Time Series function is optional and does not require a decimal point.
- Semicolons are used in Time Series functions to separate the elements or expressions from the number that determines the interval. Sample on maximum and sample on minimum require two elements or expressions also separated by a semicolon.

## Time Synchronization

The time synchronize function is useful when data is missing from a file or if several files of data need to be merged together. The files are synchronized according to time; any missing data in the file (or files) will be replaced with blank data.

This function synchronizes according to day, hrmn (hour-minute), and/or seconds. The syntax used to identify the time elements for array data is:

$e_i[\text{day}]:e_i[\text{hrmn}]:e_i[\text{seconds}]$

Referring to TABLE 1, to identify the day of year for a mixed-array data file, type:

2[189]::

for hrnm type:

:3[1200]:

and seconds are expressed as:

::4[5]

A single colon is assumed to be between day and hrnm (e.g., 2[189]: means day, :3[1200] means hours, and 2[189]:3[1200] means day and hour-minute). When the time synchronize function is used, a time interval must be specified in the Copy line of the first data file. For example, 4[60] in the Copy line will create a synchronized file containing the data from the input files that occurred every 60 minutes. If no time interval is specified in the Copy line then the time specified in the Start Condition becomes simply a starting time with no time synchronization.

Typically, the starting time specified must actually be found in the input file before the Start Condition is satisfied (e.g., if the input file starts at 1100 hrs and 1000 hrs is entered for the starting time, with no day specified, Split will skip over arrays until it reaches 1000 hrs the next day). However, the Start-Stop On/After Time function can be enabled (**Output** tab, **Other** button) to trigger the start of processing when the exact time is found or at the first instance of data after that time has occurred.

#### *Table-based dataloggers*

Because the time stamp for a table-based datalogger is all one string, and therefore read by Split as one element, the syntax is somewhat different. All elements in the time stamp are specified by a 1 (if the time stamp is the first item in each row of data).

The 1s in the string identify the position of the time stamp in the line of data. Each colon represents a portion of the time stamp. The format is 1[year]:1[day]:1[hour/minute]:1[seconds]. The colons in the time stamp must be present or the function will not work correctly.

#### **NOTE**

---

Time synchronization can only be done for data from a single year. It will not work over a year boundary.

---

Time elements can be identified without specifying a starting time (e.g., 2:3). If you are working with only one file, Split will begin processing that file at the first record in the file. If any gap in the data is found, blank data (or the “Replace Bad Data With” text) and a carriage return line feed will be inserted for each line of missing data. Note that Split will also detect a gap in data if, for instance, you specify a start time of 2[92]:3 (start at Julian day 92) and your hour/minute for day 92 starts at 9:30 a.m. The time between the start of the day (0000) and 9:30 a.m. will be considered missing data. Blanks (or the “Replace bad data with” text) and a carriage return line feed will be inserted at the beginning of the PRN file for each “missed” output interval.

If you are working with two or more files, once Split starts processing the files (based on the time of the first record of the first file), if no data exists for the other file(s), blank data will be inserted.

If multiple input files are given specific starting times, Split starts the output at the earliest specified starting time. In a PRN file, Blanks or the comment entered in the “Replace bad data with” field are inserted for values from other input files until their starting times are reached. In a RPT file only blanks are used.

---

**NOTE**

When using time synchronization with a mixed array data file, with a midnight time stamp of 2400, you will need to select the **Other** button, **Midnight at 2400 hours** check box.

---

### ***Time Synchronization and the Copy Condition***

To use the time synchronize function, time element(s) must be specified in the Start Condition. The user must also specify a time interval in the Copy condition. For instance, if the original data had 15 minute outputs and you only want hourly outputs, then an interval of 60 minutes must be specified following the element number. This is entered as (assuming hrnm is element number 3) “3[60]”. If time synchronization is specified in the Start Condition, Split looks for the interval in a time element in the Copy condition. Only one time interval is specified. This interval is the unit of time to synchronize each file.

The interval can be given tolerance limits by following the interval with a comma and the tolerance. For example, if 3 is the hrnm element, and the time interval is 60 minutes +/-2 minutes, the syntax is 3[60,2].

Table based data files need to use the same time format as described in *Start Condition* (p. 12). You can specify the interval for time synchronization on table files as ::1[60]: which will give you an output interval of 60 minutes.

If the time synchronize function is enabled and data are missing at one or more of the time intervals specified, then a blank (or the comment entered in the “Replace bad data with” field) is output to the Output File. See TABLE 5.

### ***Using Time Synchronization While Starting Relative to PC Time***

Split tries to time-sync files to the top of the hour when starting relative to PC time. If you are synchronizing files where the data output interval is not at the top of the hour, you will need to specify an interval in the Copy Condition that represents a window of time in which Split should look for the hour/minute. For instance, if your data is output 50 minutes into a 60 minute interval (and therefore, your time stamps are 50, 150, 250, 350...2350) your Start Condition and Copy Conditions for the first file might look like the following:

Start Condition

2[-1]:3[50]:

Copy Condition

1[106]and3[60,10]

Where:

- element 1 is the array ID
- element 2 is the Julian day
- element 3 is the hour/minute

The Start Condition directs Split to begin processing data when the time is one day prior to the current PC time and when the hour/minute value is equal to 50. The 1[106] in the Copy Condition specifies the array from which the data should be copied. The 3[60,10] indicates that the interval for the time stamp is 60 minutes and designates a 10 minute time window on each side of the top of the hour in which Split should look for the hour/minute data (10 minutes before the hour, 10 minutes after the hour).

The second file's Copy Condition should include only the array from which to copy the data. No interval is necessary.

## Output Files

To create an Output File, click the **OUTPUT FILE** tab. The file is created on the default drive or directory unless the file name is preceded with an alternative drive or directory. Use the **Browse** button to change directories.

Split will assign this file an extension of .PRN if an extension is not specified by the user. Whenever an Output file name is entered, regardless of extension, an Output file is created only when the **RUN | GO** menu option is selected.

If the file name you have selected already exists, you can use the "If File Exists Then" drop-down list box to determine what action Split will take. By default, each time a PAR file is run the existing output files (PRN, RPT, and HTM) are overwritten (**Overwrite** option). When **Append** is selected, the PRN file will not be overwritten — the new data will be added to the end of the existing file. However, the RPT and HTM files will be overwritten. If **Create New** is selected, Split will create all new files using the original file name and appending an \_0, \_1, and so on to each subsequent run.

In Append mode, if an HTM or RPT file is needed with all the data, you will need to run the PRN created by Split through the program a second time. If the Output File name is left blank, Split does not write data to an Output File on disk; rather, it will display the processed values on the screen if the Screen Display box is checked. If Screen Display is not enabled, no data will be displayed on the **Split RUN** screen.

---

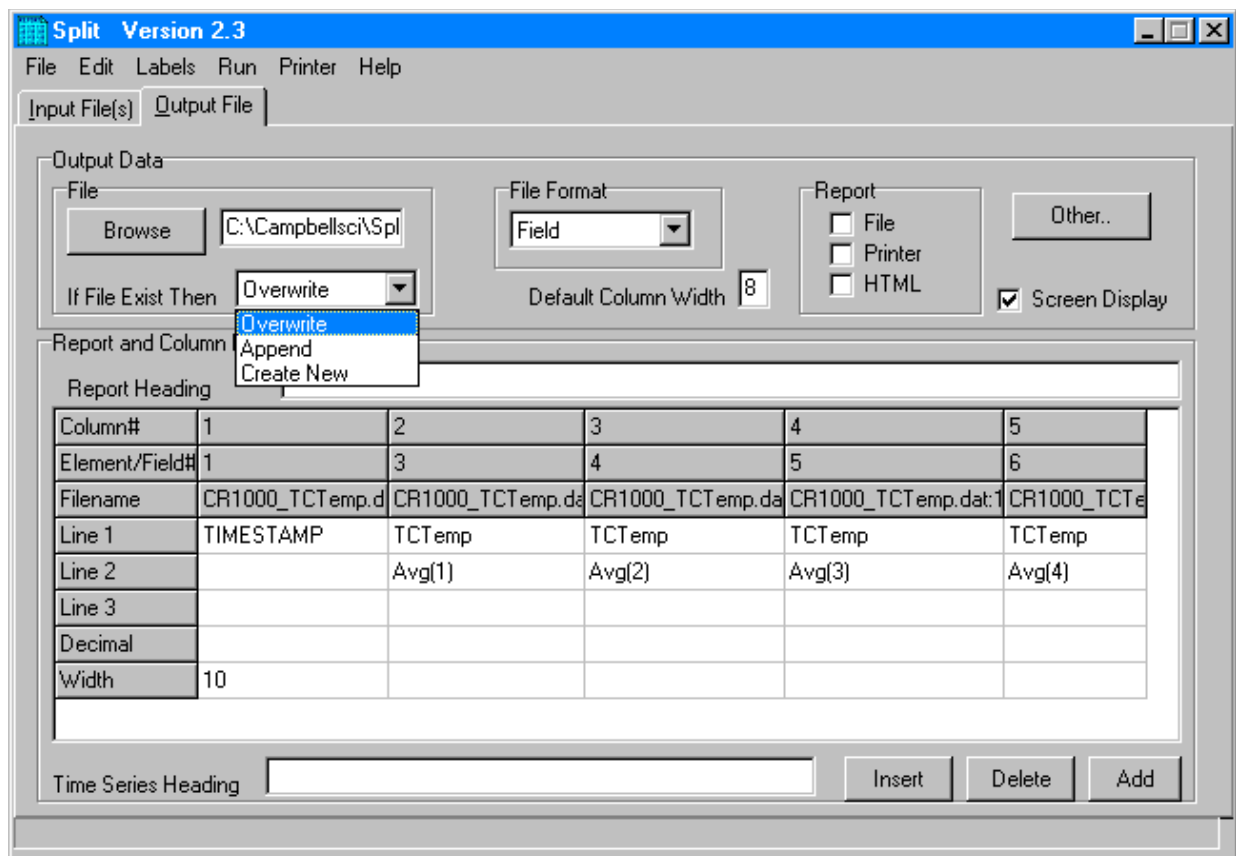
### CAUTION

The Output file name cannot be the same as the Input file name. Split will display an error message if this condition occurs.

---

Several output options may be specified to alter the default output to the file. Some are located on the main **OUTPUT FILE** screen and some are made available by pressing the **Other** button.





## Description of Output Option Commands

### File Format

There are five File Format options to choose from: No File, Field, Comma, Printable, and Custom. If No File is chosen, then only the .PRN file is saved to disk. The Field, Comma, and Printable options produce files formatted as Field Formatted, Comma Separated, and Printable ASCII, respectively. An example of each of these file types is given in TABLE 1 in the Input Files section.

The Custom file format uses the regional settings in the Windows operating system to determine the decimal symbol and the separator used with data values. In the Regional Settings for Numbers, the decimal symbol uses the character specified in the Decimal Symbol field; the separator uses the character specified in the List Separator field. These settings are typically found in **Control Panel | Regional Settings** (or Options), **Numbers** tab. This allows users who are used to the comma “,” as the decimal and the period “.” as a data separator to see the output data in that format.

### Default Column Widths

The Default Column Widths field is used to set the default width of the columns. Valid entries are 6,7,8, and 9. The initial width is 8. High Resolution Final Storage data requires a minimum column width of 8. Entering a number in the Width row for each column overrides the default settings and sets the width of individual columns. If this field is left blank, the Default Column Widths field is used.

### Screen Display

The Screen Display field controls writing the processed data to the screen. To write to the screen, check the box. For faster execution, clear the box to omit writing to screen. The data will then be written to the file only.

### Report

A report, with page and column headings, can be sent to a file or printer. There are three report options: File, Printer, HTML. One or more can be selected. A report sent to a file has the extension of .RPT. If the report is sent to a printer, the printer must be on-line. In all cases a .PRN output file is created. A basic HTML file can be created containing the formatted report data. The HTML file can be used as a display of the formatted data output in a web browser.

### NOTE

To remove page breaks in the HTML file, enable the “No FF” option.

### Other

The **Other** button provides access to the dialog box shown below.

**Other**

**Bad data**

Replace bad data with

☐ Only display lines with bad data

**Processing**

☐ Trigger on Stop condition

☐ Start-Stop On / After Time

☐ Time Sync to First Record

**Special**

☐ Match files ☐ Transpose file

☐ No FF Break arrays

☐ No Summary ☐ No Date Advance

☐ No Dashes ☐ No Summary Heading

☐ No Record Numbers from TOB files

Ok Cancel Help

It allows the following settings to be modified:

**Replace bad data with** – The text in the field, to the right of this option, is entered into the .PRN output file data set if data are blank, bad, or out of range. See TABLE 9 for definition of blank or bad data. Whatever text string the user enters in the field will be entered if a blank or question mark is in the data or if data are out of range. This option is useful when the Output file is imported into a spreadsheet program, such as Excel.

TABLE 9. Definition of Blank or Bad Data for each Data File Format	
File Format	Definition of Blank or Bad Data
Printable ASCII	????
Comma Separated ASCII	blank or any character except numeral or space
Field Formatted	blank or "" (double quotation marks)

**Only display lines with bad data** – Outputs only those arrays containing one or more Out of Range elements. If a report is generated, an asterisk precedes the Out of Range value in the .RPT file.

**Trigger on Stop condition** – Changes the meaning of Stop Condition to trigger Time Series processing output. The Stop Condition is included in the Time Series processing if it satisfies the Copy line.

If the **Trigger on Stop Condition** is selected, a Time Series output will occur each time the Stop Condition is met.

**Start-Stop On/After Time** – In most instances, Split will not start or stop processing a file unless the exact start condition is found. However, when starting or stopping based on time, you can enable Split's **Start-Stop On/After Time** option. This will trigger the start (or end) of processing when the exact time is found or at the first instance of data after that time has occurred (which meets other defined criteria in the PAR file).

**Time Sync to First Record** – This option is used with the time-sync function. It allows you to set specific times in the Start Condition, but have synchronization start at the first record in the file that meets the Start Condition. This may avoid an output file that starts with blank lines.

For example, you have table-based data file(s) containing 15 minute data. Your first data file starts on Sept 9<sup>th</sup> at 12:15 p.m. You want to time sync the files and output only the data that occurs at midnight.

You need to specify '0' for the hour/minute field in the Start Condition or the output will contain the data that occurs each day at 12:15. Therefore, you would use:

Start Condition = 1:1:1[0]:1

The Copy Condition determines the interval of your data. Therefore, to output data that occurs every 24 hours, you would use:

Copy Condition = 1:1[1]:1:1

Because you have specified a time in the Start Condition, but not the day, Split assumes the first day of the year. Therefore, by default, you will have blank lines in your output file for each day from Jan 1<sup>st</sup> to Sept 9<sup>th</sup>. Using the **Time Sync to First Record** option will avoid these blank lines.

**Match files** – This option compares two files of the same data. If good data exists in one and not the other (question marks), then Split will fill the OUTPUT file with the good data. This is used to get a more complete record from an error ridden file (e.g., one recorded at freezing temperatures by reading a tape twice and running both files through Split).

---

**CAUTION**

For the Match files option to produce a correct Output File, the differences between the two Input Files can only be question marks. Both files must have the same Start Condition or the beginning of both files must be the same.

---

**Transpose file** – Transposes the rows and columns of the input file. Only one Input File can be transposed at a time and no Select options can be specified. A maximum of 26 arrays are transposed per pass of Split.

To transpose a file containing more than 26 arrays, several passes are required. Change the Output file name and Start Condition for each pass. Split may then be used to merge the multiple files.

**No FF** – Suppresses form feeds and page breaks in RPT and HTML files. When this option is selected, a header appears on the first page only. This option is used for printing reports on continuous feed paper or for displaying HTM files in a browser.

**Break arrays** – This option breaks up the Output Array into new arrays that are #+1 elements in each new array. Split automatically assigns an array ID number equal to the first element in the first array. Only one Input File may be specified. Start, Stop, and Copy Conditions may be specified, but the Select line must be left blank.

---

**NOTE**

The Break Arrays function works only for mixed array data. It is typically used when processing data from burst measurements.

---

**No Summary** – When producing reports that include time series processing based on an interval, sometimes that interval will not divide evenly into the number of lines in the data file that is being processed. For example, you may be processing one-minute data on a five-minute interval, and the data file has 103 lines; thus, there are 3 lines of data “left over” at the end of the report. By default, the summary (average, total, maximum, etc., depending upon which time series function is being used) of the left over values is printed at the bottom of the report following the Time Series Heading. Enable the **No Summary** check box to omit the

summary of the left over values and the Time Series Heading from the report.

**No Date Advance** – When processing a data file from a mixed array datalogger, if the time stamp uses midnight as 2400 with “today’s” date, the date function will convert that time stamp to 0000 hours with “tomorrow’s” date. (This is because the algorithm used by the date function is based on Windows’ time format, and it does not support a 2400 time stamp.) For example:

Array ID	Year	Julian Day	Hour/Minute	Date Function	Data	Data
10	2002	151	2200	05/31/02 22:00	1.701	193.6
10	2002	151	2300	05/31/02 23:00	1.476	31.99
10	2002	151	2400	06/01/02 00:00	1.123	106.2

At Julian Day 151 (May 31) 2400 hours, the date function produces an output of June 1 00:00 hours. The date can be stopped from rolling forward by using the **No Date Advance** check box. The output will then be similar to:

Array ID	Year	Julian Day	Hour/Minute	Date Function	Data	Data
10	2002	151	2200	05/31/02 22:00	1.701	193.6
10	2002	151	2300	05/31/02 23:00	1.476	31.99
10	2002	151	2400	05/31/02 00:00	1.123	106.2

Caution should be used when applying the date function and enabling or disabling No Date Advance, since it is possible to produce an incorrect date. For instance, using the above example if you were to enter the following into your select line:

```
3,edate(“hh:mm”;4;3;2)
```

with the No Date Advance enabled, you would get the output:

```
151    22:00    1.701    193.6
151    23:00    1.476    31.99
151    00:00    1.123    106.2
```

If you were to enter:

```
edate(“mm/dd/yy”;4;3;2),4,6,7
```

with the No Date Advance disabled, you would get the output:

```
05/31/02    2200    1.701    193.6
05/31/02    2300    1.476    31.99
06/01/02    2400    1.123    106.2
```

**No Dashes** – When the **No Dashes** check box is selected, the dashed line that typically appears under the column headings will not be displayed. This option affects all output types (PRN, RPT, HTM, and printed page).

**No Summary Heading** – When processing data using time series functions (see No Summary, above), select this option to prevent the Time Series

Heading and Column Headings from being printed at the bottom of the report. The “left over” summary data will still be printed.

**No Record Numbers from TOB Files** – Split automatically converts TOB (binary) files to ASCII prior to being processed. When this check box is selected, the record numbers will not be included in the converted file. This will affect the element numbers used for the Start, Stop, Copy, or Select fields of the PAR (e.g., if a file has a timestamp, record number, and data value, when this check box is selected the data value would be element 2. When the check box is cleared, the data value would be element 3).

## Report Headings

A report is output to a printer or file with the extension .RPT. Headings are not included in the standard output to disk (.PRN or user named extension output file). However, a report can be labeled with a header by entering text into the Report Heading field. A report heading can have several lines, but it is limited to a total of 253 characters including backslashes and carriage returns. “\” characters break the report heading into multiple lines.

When Time Series functions are used in the Select field without an interval, they appear as a final summary at the end of the report. They can be labeled by entering a title into the Time Series Heading field at the bottom of the Output File page. Time Series interval summaries cannot be assigned individual titles directly, but you can use special functions such as “Label” and “Crlf” to create column headings and special formatting.

“PCDATE” within the Report Heading inserts the computer’s current date (Month-Day-Year). For the European format (Day-Month-Year), enter “PCEDATE”.

## Column Headings

Up to three lines per column can be entered as column headings. These headings are limited to a length of one less than the Output field width.

Column headings associated with Time Series outputs are repeated for Final Summaries if a title for the Final Summary is requested on the headings for report line.

The number of digits to report to the right of the decimal point is entered in the Decimal field and can be set independently for each column. The value output will be rounded to the specified number of digits. Leave this field blank if you do not want to round the data to a specific number of digits.

Column headings can be entered using Split’s Data Labels Function (**Labels | Use Data Labels**).

## Help Option

On-line Help is available from any location in Split. Simply select the area of Split in question and press <F1>. Split also offers a brief on-the-fly Help. Place the cursor on the area of Split in question; after a moment a brief description is displayed in the hint line of the Split window (bottom left).

# Editing Commands

Split supports the Windows Cut, Copy, and Paste commands. Text from any field in Split or other Windows applications can be Cut, Copied, or Pasted.

## Running Split From a Command Line

Existing parameter files can be executed using Splitr.exe which is a “run-time” version of the Split Report Generator. When Splitr.exe is run, the file is processed as if the user chose **Run | Go** from the Split menu. Splitr.exe can be executed by the Task Master, from a batch file, or from a Windows command line prompt or shortcut.

### Splitr Command Line Switches

Splitr has four switches that can be used to control how the executable is run.

#### ***Closing the Splitr.exe Program After Execution (/R or /Q Switch)***

Typically when Split is run, after the file is processed the user must close the Screen Display window. When Splitr.exe is run from a command line, the user must also close the Screen Display window unless the /R switch is used.

The syntax for this switch is:

SPLITR LOGAN/R

where LOGAN is the parameter file name.

The /R switch should follow immediately after the parameter file name with no space between the two. If a space is used, the following message will be displayed “There was a problem opening the input file. File could not be found or may be in use.”

The /Q switch is similar in function and syntax to /R. However, if Split encounters an error when processing the file, no message box is displayed that requires user response (the exceptions are a disk space error or an internal error with the Split executable). This option should be used with caution, since there will be no indication of a problem if a file cannot be processed.

#### ***Running Splitr in a Hidden or Minimized State (/H Switch)***

Splitr can be run in a minimized state, so that the Screen Display window does not interrupt other processes on the computer. The syntax for running Splitr minimized is:

SPLITR /H LOGAN

where LOGAN is the parameter file name.

The /H switch must be positioned after SPLITR but before the parameter file name, and a space is required between the executable name and the switch.

## **Running Multiple Copies of Splitr (/M Switch)**

Multiple copies of Splitr can be run at one time by using the /M switch. This switch must appear immediately after Splitr. For instance, a batch file containing the lines:

```
SPLITR /M Logan/R  
SPLITR /M Sinks/R
```

will open two copies of Splitr and process the two files simultaneously.

### **NOTE**

---

When using the /M switch in a batch file, the behavior may depend on your Windows version. In some cases, the files will be processed simultaneously, while in other cases, the files will be processed sequentially. It may be possible to change this behavior using the Windows "start" command.

---

## **Using Splitr.exe in Batch Files**

Batch files containing one or more Splitr command lines can be useful for automating data processing. Batch files can be executed manually or by setting them up in the Task Master.

Batch files process each command in succession, without waiting for execution of a command to be completed before proceeding to the next unless they are configured to do so. If multiple parameter files are being processed using Splitr in a batch file, there are no conflicts because only one copy of Splitr can be active at any one time (unless the /M switch is used. However, if other commands are used along with Splitr (such as opening the file in a spreadsheet, copying it to an archive directory, or appending it to an existing file) these commands might be executed before Splitr finishes processing data.

The Windows Start /w (wait) command can be added to a batch file command line to delay execution of the next command until the first command has finished. The Start command has different arguments depending upon the operating system you are using. Refer to your computer's on-line help for information on this command.

## **Processing Alternate Files**

Splitr allows the user to select different input and/or output files for an existing parameter file by entering them on the command line after the parameter file name. For example:

```
"Splitr LOGAN.PAR/R TEST.DAT TEST.PRN"
```

Replaces the Input and Output file names in LOGAN.PAR, with TEST.DAT and TEST.PRN, respectively.

A space must be used to separate command line parameters. Splitr uses as many entries as exist on the command line. However, the command line has a limit to the number of characters it can accommodate—this limit is operating system dependent. The parameters must be in the following sequence: Input file name, Output file name, Start Condition, Stop Condition, Copy Condition, and Select.

If a parameter is to be left as it is in the parameter file, then space comma space ( , ) may be entered in the command line. For instance, if the parameter file



LOGAN.PAR contained TEST1.DAT as an input file name, the following command line would leave the input file TEST1.DAT and change the output file to TEST.PRN.

“SPLITR LOGAN/R , TEST.PRN”

### ***Input/Output File Command Line Switches for Processing Alternate Files***

The one caveat of using the command line to specify an alternate input and/or output file name is that Split's default options will be used with the alternate file. For instance, by default, output files are written with field-formatted columns. If the original PAR file specified a comma-separated output, that option would be ignored and the and defaults would be used.

Command line switches can be used to control these options for the output and input files. The switch is added immediately after the input or output file name.

#### **NOTE**

---

In most instances, full path names to the Splitr executable and the input and output file names must be used. In addition, if long file names are used in the path, you may need to surround the path and file name by double quotes.

---

#### **Output File Options**

These switches are entered after the output file name; e.g., Splitr Test.par/r Input.dat **Output.prn/P**

- |         |   |
|---------|---|
| /P      | Sends the output to a printer. This is the same as checking the Printer box for the Report type on the <b>Output File</b> tab.  |
| /R      | Creates a formatted RPT file. This is the same as checking the File box for the Report type on the <b>Output File</b> tab.  |
| /W      | Creates a simple HTML file. This is the same as checking the HTML box for the Report type on the <b>Output File</b> tab.  |
| /A      | Appends the output to the end of an existing file. This is the same as selecting Append for the If File Exists option on the <b>Output File</b> tab.  |
| /L      | Creates a new output file with a different name if a file exists. This is the same as selecting Create New for the If File Exists option on the <b>Output File</b> tab.                                     |
| /O      | Turns the screen display off when Split is processing the PAR file. This is the same as clearing the <b>Screen Display</b> check box on the <b>Output File</b> tab.   |
| /6..9   | Sets the default width for all the columns in the report. This is the same as entering a value in the Default Column Width field on the <b>Output File</b> tab.   |
| /[text] | Sets the text that will be used in the place of bad data. This is the same as the text string used in the Replace Bad Data field that is found under the <b>Other</b> button of the <b>Output File</b> tab. |

- /M Compares two input files and creates an output file with a complete data set comprised of both files. This is the same as the Match Files option that is found under the **Other** button of the **Output File** tab. The two input file names are separated with a comma but no spaces. Example: Splitr Test.par/r Input1.dat,Input2.dat Output.prn/M
  
- /S Writes the output file without a form feed command after each page. This is the same as the No FF (form feed) option that is found under the **Other** button of the **Output File** tab.
  
- /G Outputs only the data marked as “bad” to the file. This is the same as the **Only Display Lines with Bad Data** check box that is found under the **Other** button of the **Output File** tab.
  
- /O Outputs the data in comma separated format. This is the same as choosing the Comma option for the File Format.
  
- /1 Outputs the data in printable ASCII format. This is the same as choosing the Printable option for the File Format.
  
- /2 Outputs the data using the Regional Settings of your Windows operating system for the decimal indicator and data value separator. This is the same as choosing the Custom option for the File Format (this is the default option for the File Format field).
  
- /F Conditionally outputs the data using the Trigger On Stop Condition. This is the same as choosing the Trigger On Stop Condition option that is found under the **Other** button of the **Output File** tab. A stop condition must also be specified. The example below does not specify a start or copy condition. These two fields are indicated by the “space-comma-space” entries. Select line entries are also shown in this example.  
  
Example: Test.par/r input1.dat Output.prn/F , 4[1450] ,  
smpl(1..6),avg(7)
  
- /T Transposes the rows and columns of a file. This is the same as choosing the Transpose File option that is found under the **Other** button of the **Output File** tab.
  
- /D Enables the No Date Advance function, which keeps the date for midnight from rolling to the next day. This is the same as choosing the **No Date Advance** check box that is found under the **Other** button of the **Output File** tab.
  
- /N Suppresses the summary information when processing time series data. This is the same as choosing the **No Summary** check box that is found under the **Other** button of the **Output File** tab.
  
- /H Removes the dashed lines from the heading of the RPT file. This is the same as choosing the **No Dashes** check box that is found under the **Other** button of the **Output File** tab.
  
- /U Removes the record number from TOB files that are processed with Split. This is the same as choosing the **No Record Numbers from**

**TOB Files** check box that is found under the **Other** button of the **Output File** tab.

/E Begins processing the file, or stops processing the file, on or after the Start or Stop condition when starting or stopping based on time (the default is to start only if the exact start condition is found). This is the same as choosing the Start -Stop On/After Time option that is found under the **Other** button of the **Output File** tab.

Example: Splitr test.par input1.dat Output.prn/E 4[1450]: 4[1456]:  
(where 1450 and 1456 are the start and stop times, respectively.  
Colons are required to indicate a time value.)

/I Suppresses the time series heading and column heading information when processing time series data. This is the same as choosing the **No Summary Heading** check box that is found under the **Other** button of the **Output File** tab.

/Bnnn Breaks a long array into multiple lines, where nnn is the number of values to place on each line. This is the same as choosing the **Break Arrays** check box that is found under the **Other** button of the **Output File** tab.

### Input File Options

These switches are entered after the input file name; e.g., Splitr Test.par/r **Input.dat/L** Output.prn

/nnn Begins processing nnn bytes into the file. If /nnn..mmm is used, then processing begins at nnn bytes into the file and stops at mmm bytes into the file. This is the same as setting a specific Start and Stop offset, which is found under the **Offsets/Options** button of the **Input File** tab.

/L Begins processing the file at the byte value where processing last stopped. If /L..mmm is used, then processing begins where it left off and stops at mmm bytes into the file. This is the same as enabling Last Count, which is found under the **Offsets/Options** button of the **Input File** tab.

/Bnnn Specifies the file type as Burst data. nnn indicates the size of the arrays. This is the same as selecting Burst Format for the File Info field on the **Input File** tab.

/F Specifies the file type as Final Storage (binary) data. This is the same as selecting Final Storage Format for the File Info field on the **Input File** tab.

/M Changes the value for midnight to 2400 instead of 0000. This is the same as selecting **Midnight is 2400 Hours** check box found under the **Offsets/Options** button of the **Input File** tab.

### Batch File Example

```
"c:\Program Files\campbellsci\LoggerNet\splitr.exe"  
c:\Campbellsci\SplitW\switch-test.par input1a.dat Output.prn/E/H/W 4[1200]: ,  
, 1..6
```

where  
PAR file: switch-test.par  
Input file: input1a.dat  
Output file: output.prn  
Other outputs: Output.HTML  
Start condition: on or after 1200  
Stop condition: end of file  
Copy condition: none  
Elements: 1 through 6

### Processing Multiple Parameter Files with One Command Line

More than one .PAR file can be executed with a single Splitr command line. Each .PAR file and its associated parameters are separated from the next .PAR file by a semicolon with one space on each side ( ; ). For example:

```
"SPLITR LOGAN/R TEST.DAT TEST.PRN ; SINKS/R TEST1.DAT  
TEST2.DAT 1[189]"
```

executes the LOGAN.PAR file on TEST.DAT and outputs the results to TEST.PRN, then executes the SINKS.PAR file on TEST1.DAT and outputs the results to TEST2.DAT. Execution of SINKS.PAR starts when the first element in TEST1.DAT is 189.

## Log Files

Split maintains a log file each time Splitr is run. The main purpose of this log file is to enable users running Splitr in command line mode to identify what happened with each execution of Splitr. The file is named splitr.log and is written to the Sys directory of the Split working directory. (By default, this is C:\Campbellsci\Splitw\sys.) The file will grow to approximately 4–5K in size and then be renamed to splitr.bak. (Any previous splitr.bak file will be overwritten. Therefore, only two log files will be retained.)

If a second instance of Splitr is started when one is already running, another log file, splitrunning.log, will be written. This file simply identifies the time that the second instance of Splitr was started and that Splitr was already running.

# Global Sales and Support Network

A worldwide network to help meet your needs



## Campbell Scientific Regional Offices

### Australia

**Location:** Garbutt, QLD Australia  
**Phone:** 61.7.4401.7700  
**Email:** [info@campbellsci.com.au](mailto:info@campbellsci.com.au)  
**Website:** [www.campbellsci.com.au](http://www.campbellsci.com.au)

### Brazil

**Location:** São Paulo, SP Brazil  
**Phone:** 11.3732.3399  
**Email:** [vendas@campbellsci.com.br](mailto:vendas@campbellsci.com.br)  
**Website:** [www.campbellsci.com.br](http://www.campbellsci.com.br)

### Canada

**Location:** Edmonton, AB Canada  
**Phone:** 780.454.2505  
**Email:** [dataloggers@campbellsci.ca](mailto:dataloggers@campbellsci.ca)  
**Website:** [www.campbellsci.ca](http://www.campbellsci.ca)

### China

**Location:** Beijing, P. R. China  
**Phone:** 86.10.6561.0080  
**Email:** [info@campbellsci.com.cn](mailto:info@campbellsci.com.cn)  
**Website:** [www.campbellsci.com.cn](http://www.campbellsci.com.cn)

### Costa Rica

**Location:** San Pedro, Costa Rica  
**Phone:** 506.2280.1564  
**Email:** [info@campbellsci.cc](mailto:info@campbellsci.cc)  
**Website:** [www.campbellsci.cc](http://www.campbellsci.cc)

### France

**Location:** Montrouge, France  
**Phone:** 0033.0.1.56.45.15.20  
**Email:** [info@campbellsci.fr](mailto:info@campbellsci.fr)  
**Website:** [www.campbellsci.fr](http://www.campbellsci.fr)

### Germany

**Location:** Bremen, Germany  
**Phone:** 49.0.421.460974.0  
**Email:** [info@campbellsci.de](mailto:info@campbellsci.de)  
**Website:** [www.campbellsci.de](http://www.campbellsci.de)

### India

**Location:** New Delhi, DL India  
**Phone:** 91.11.46500481.482  
**Email:** [info@campbellsci.in](mailto:info@campbellsci.in)  
**Website:** [www.campbellsci.in](http://www.campbellsci.in)

### Japan

**Location:** Kawagishi, Toda City, Japan  
**Phone:** 048.400.5001  
**Email:** [jp-info@campbellsci.com](mailto:jp-info@campbellsci.com)  
**Website:** [www.campbellsci.co.jp](http://www.campbellsci.co.jp)

### South Africa

**Location:** Stellenbosch, South Africa  
**Phone:** 27.21.8809960  
**Email:** [sales@campbellsci.co.za](mailto:sales@campbellsci.co.za)  
**Website:** [www.campbellsci.co.za](http://www.campbellsci.co.za)

### Spain

**Location:** Barcelona, Spain  
**Phone:** 34.93.2323938  
**Email:** [info@campbellsci.es](mailto:info@campbellsci.es)  
**Website:** [www.campbellsci.es](http://www.campbellsci.es)

### Thailand

**Location:** Bangkok, Thailand  
**Phone:** 66.2.719.3399  
**Email:** [info@campbellsci.asia](mailto:info@campbellsci.asia)  
**Website:** [www.campbellsci.asia](http://www.campbellsci.asia)

### UK

**Location:** Shephed, Loughborough, UK  
**Phone:** 44.0.1509.601141  
**Email:** [sales@campbellsci.co.uk](mailto:sales@campbellsci.co.uk)  
**Website:** [www.campbellsci.co.uk](http://www.campbellsci.co.uk)

### USA

**Location:** Logan, UT USA  
**Phone:** 435.227.9120  
**Email:** [info@campbellsci.com](mailto:info@campbellsci.com)  
**Website:** [www.campbellsci.com](http://www.campbellsci.com)