PRODUCT MANUAL

Software

LoggerNet

Datalogger Support Software

Version 4.9

Revision: 10/2024

RELIABLE MONITORING SINCE 1974

Campbell SCIENTIFIC®

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.

This software can be installed as a trial version or as a fully licensed copy. All terms and conditions contained herein apply to both versions of software unless explicitly stated.

TRIAL VERSION: Campbell Scientific distributes a trial version of this software free of charge to enable users to work with Campbell Scientific data acquisition equipment. You may use the trial version of this software for 30 days on a single computer. After that period has ended, to continue using this product you must purchase a fully licensed version.

This trial may be freely copied. However, you are prohibited from charging in any way for any such copies and from distributing the software and/or the documentation with any other products (commercial or otherwise) without prior written permission from Campbell Scientific.

LICENSE FOR USE: Campbell Scientific grants you a non-exclusive license to use this software in accordance with the following:

1. The purchase of this software allows you to install and use a single instance of the software on one physical computer or one virtual machine only.
2. This software cannot be loaded on a network server for the purposes of distribution or for access to the software by multiple operators. If the software can be used from any computer other than the computer on which it is installed, you must license a copy of the software for each additional computer from which the software may be accessed.

3. If this copy of the software is an upgrade from a previous version, you must possess a valid license for the earlier version of software. You may continue to use the earlier copy of software only if the upgrade copy and earlier version are installed and used on the same computer. The earlier version of software may not be installed and used on a separate computer or transferred to another party.
4. This software package is licensed as a single product. Its component parts may not be separated for use on more than one computer.
5. You may make one (1) backup copy of this software onto media similar to the original distribution, to protect your investment in the software in case of damage or loss. This backup copy can be used only to replace an unusable copy of the original installation media.

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

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 LoggerNet manual: www.campbellsci.com/old-manuals.

1. System Requirements

LoggerNet runs on Windows 10 and Windows 11. LoggerNet 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 LoggerNet to run.

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

2. Installation, Operation, and Backup Procedures

2.1 Installation

If you are installing LoggerNet from a download, run the executable file, LoggerNet_*version*.exe, to begin the installation.

If you are installing from a CD, place the installation disk in your computer's CD drive. If autorun is enabled for the drive, the LoggerNet installation will start automatically. If the installation does not start automatically, use the **Browse** button to access the CD drive and select the autorun.exe file from the disk.

The first screen displayed by the installation is a Welcome screen. Click **Next** to proceed to the licensing agreement. After reading the licensing agreement, select the "**I Accept**..." option and select **Next** to proceed to the User Information screen. At the bottom of the User Information screen is a field for entering the key for the software. The key is found in your email to download the software or on the back of the CD case in which LoggerNet is shipped. Use the drop-down list box for the first part of the key to select the software being installed: LGRNET (LoggerNet), LGNADM (LoggerNet Admin), or LGNRMT (LoggerNet Remote). Note that you must select the correct LoggerNet version for your key or you cannot proceed further in the installation. After entering the key, select **Next** and continue through the remaining screens, following the on-screen prompts to complete the installation.

Items are added to your computer's Start menu under **Campbell Scientific** that start the Toolbar and some other selected utilities. At the end of installation you also have the option to add a desktop shortcut to LoggerNet.

By default, the installation copies the LoggerNet program files to the C:\Program Files (x86)\CampbellSci\LoggerNet directory. Many operating system configurations will require the user name under which the software is installed to have administrative privileges to the computer. After the software is installed, administrative privileges are not required by the user to run the software.

In addition to placing files in the Program Files directory of your computer, the installation also creates working directories for the LoggerNet server and the individual LoggerNet applications under C:\CampbellSci. LoggerNet Directory Structure (p. 3) provides more detail on the directories that are created.

If you are installing the trial version of LoggerNet, you will have 30 days to use this fully functional trial version. Each time you run LoggerNet, you will be advised as to how many days are remaining on your trial version. At the end of the 30 days, the trial version of LoggerNet will no longer function.

If you choose to purchase LoggerNet, you will need to uninstall the trial version, run the install program (from a download or CD), and input the key. This can be done either before or after the 30-day trial period has expired.

Note that the trial version will install applications in the C:\Program Files (x86)\Campbellsci\Demo directory. When the purchased version of LoggerNet is installed, the applications will each be installed in their own directory under C:\Program Files (x86)\Campbellsci.

2.2 Upgrade Notes

When you upgrade an existing installation, LoggerNet will continue to use the network map, data collection schedules, data file locations, etc., of the existing installation. Essentially, you will be able to "pick up where you left off" the last time you used LoggerNet.

2.3 LoggerNet Operation and Backup Procedures

This section describes some of the concepts and procedures recommended for routine operation and security of the LoggerNet 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.

2.3.1 LoggerNet Directory Structure

2.3.1.1 Program Directory

As described in the installation procedures, all of the files for program execution are stored in the C:\Program Files (x86)\Campbellsci\LoggerNet 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. LoggerNet and its applications rely on registry entries to run correctly; therefore, any restoration of the program should be done by reinstalling the software.

2.3.1.2 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 LoggerNet 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. The **RTMCMediaLib** directory contains all of the media files that can be used by RTMC to provide graphics and sound for your RTMC projects. Any custom graphics or sounds that you create and wish to use in your project should be stored in one of these directories.

LoggerNet directory – The ASCII data files that are saved to disk as a result of data collection from the dataloggers are stored to the LoggerNet 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. See Appendix C, Software Organization (p. C-1), for additional information.)

2.3.2 Backing up LoggerNet

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.

LoggerNet provides several ways to back up the Network Map and data files. A manual backup can be performed at any time by selecting Manual Backup (p. 5) from the Setup Screen Backup menu. You can set up LoggerNet to perform scheduled backups on a user-defined interval by selecting Scheduled Backup (p. 6) from the Setup Screen Backup menu. Finally, a backup can be performed automatically by setting up a task in LoggerNet Task Master. This can be useful to perform a backup every time a certain event type occurs. (The Task Master can also be used to perform scheduled backups, but that function is performed more easily using the Scheduled Backup menu item described above.) See Example #2 under Task Master (p. 323) for additional information on using the Task Master to perform an automatic backup.

A backup file created using any of the above methods can be used to restore the LoggerNet network by selecting Restore Network (p. 7) from the Setup Screen Backup menu. Note that this process DOES NOT append to the existing network. The existing network will be overwritten when the restore is performed.

2.3.2.1 Manual Backup

This option is available by selecting Backup | Manual Backup from the Setup Screen.

This function saves a copy of the network map to a file, which can then be used to restore the network if necessary. The settings for all the devices in the network will be saved. The *.ini files for each of the applications will be also included in the backup. Ini files store settings such as window size and position, configuration of the data displays, etc. Along with the device map and *.ini files, there is an additional option:

> Include Cache Table Files in the Backup - If this check box is selected, the LoggerNet Server Data Cache (p. 437) will be included in the backup.

After selecting **Next**, you will be provided with another dialog box that can be used to add files to the backup file. By default, user files, such as the *.dat files that are created when you collect data from a datalogger, the program files created by Edlog or CRBasic, and the files created when running Split, are not included in the backup. They can be added to the backup from this window if desired. When **Next** is selected again, you are prompted for a file name for the backup. After supplying a file name, you can choose to execute the backup or save the backup configuration file for later use. The backup configuration file can be used to run a backup of the network as a task using LNBackup.exe. See Example #2 under Task Master (p. 323) for additional information on this utility.

2.3.2.2 Scheduled Backup

This option is available by selecting Backup | Scheduled Backup from the Setup Screen (Standard View).

This function saves a copy of the network map to a file on a specified interval. The backup file can then be used to restore the network if necessary. The settings for all the devices in the network will be saved. The *.ini files for each of the applications will be also included in the backup. Ini files store settings such as window size and position, configuration of the data displays, etc.

The Scheduled Backup dialog box has the following settings:

Scheduled Backup Enabled - Select this check box to enable the backup schedule that has been set up. Clearing the check box suspends scheduled backups.

Backup Base Date and Time - Enter a date and a time that the first backup should occur.

Backup Interval - Enter the interval on which backups should be performed. The backup interval is relative to the **Backup Base Date** and **Time** entries. For instance, if the Base Time is set at 12:15 and the interval is set for 1 day, backups will be performed at 12:15 each day. The format for this field is 00 d(ays) 00 h(ours).

> **NOTE:** The actual interval used by the LoggerNet server cannot be less than one hour. Therefore, entering a value of 0 for the Backup Interval will result in backups being performed once an hour.

> **NOTE:** If an interval of time greater than the specified Backup Interval has elapsed since the last backup, an immediate backup will occur when a new schedule is applied. Entering 0 as a Backup Interval will always cause an immediate backup when the Apply button is pressed. Therefore, temporarily entering 0 in the Backup Interval can be useful to quickly verify that your backup configuration is working.

Additional Backup Files – Indicates any additional files that will be included in the backup. By default, user files, such as the *.dat files that are created when you collect data from a datalogger, the program files created by Edlog or CRBasic, and the files created when running Split, are not included in the backup. They can be added to the backup here if desired. Press the **Add File(s)** button to add individual files to this list. Press the **Add Directory** button to add entire directories to this list. Select file or directory names in the list, and then press the **Remove Selected** button to remove them from this list.

Include LoggerNet Cache – If this check box is selected, the LoggerNet Server Data Cache (p. 437) will be included in the backup.

Backup file – Specifies the path and base file name for backup files generated by the scheduled backup. When an automated backup is generated, LoggerNet will search for the last period in this name and will insert the current date (formatted as yyyymmdd-HHMMSS) before it to make the file name unique.

File count – Specifies how many backup files will be maintained. Once this number is reached, the oldest backup file will be deleted each time a new backup file is created.

2.3.2.3 Restore Network

This option is available by selecting Backup | Restore Network from the Setup Screen.

This option regenerates the network map from a backup file. Select the name of the backup file to restore from the first dialog box, and press **Next**,

Select whether or not network communications will be enabled, when the network is restored. Clear this checkbox if the network has scheduled collection enabled, but you do not want it to start as soon as the network is restored. Communications can later be enabled with the Setup Screen > Tools Menu > LoggerNet Server Settings > Datalogger Network Communications Enabled checkbox.

Press **Restore**.

> **NOTE:** This backup will replace the existing network (it does not add to the existing network) and any existing files, including data files, program files, or Split files if they were included in the original backup.

2.3.3 Loss of Computer Power

The LoggerNet 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 LoggerNet 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 collection schedules, 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 LoggerNet 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.

2.3.4 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 LoggerNet server after a program crash or it crashes as soon as it starts, make sure that the LoggerNet 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.

2.4 Installing/Running LoggerNet as a Service

If you have LoggerNet Admin, you can install and run LoggerNet as a service. The advantage of running a software application as a service is that the software will run even when no user is logged in to the computer system. Some users may desire to run LoggerNet as a service, so that in the event of a loss of computer system power LoggerNet will resume data collection and scheduled task activities when power is restored to the computer and it "boots up".

LoggerNet Admin includes LoggerNet Service Manager, which is used to install and control LoggerNet when running as a service. The LoggerNet Service Manager is opened from the Windows Start menu. The LoggerNet Service Status box at the top of the LoggerNet Service

Manager window indicates whether or not LoggerNet is installed as a service and whether or not the service is running. When you first open the LoggerNet Service Manager, if LoggerNet has not been installed as a service you will need to select the **Install** button. Note that LoggerNet must be closed to install it as a service. If it is opened, you will be prompted to close it before you can continue.

Once LoggerNet is installed as a service, you can use LoggerNet Service Manager to Start the service (or Stop it if it is enabled). The LoggerNet service can be uninstalled by selecting the **Uninstall** button.

One caveat to running LoggerNet as a service is if LoggerNet uses system resources (such as a COM port that is continuously open waiting for datalogger call-back attempts) these resources will be unavailable to other applications until the service is stopped.

Note that when running LoggerNet as a service, tasks being run by the Task Master cannot interact with the desktop. Therefore, any tasks set up in the Task Master should not require any user interaction.

If LoggerNet is installed as a service, the service must be running for LoggerNet to run (either automatically or manually). You will be prompted to start the service if you try to launch LoggerNet manually when the service is installed but not running.

The first time LoggerNet is installed as a service, a LoggerNet user account is created. The LoggerNet service is run under this LoggerNet user account. This allows the LoggerNet Service to always run in the same environment with known user rights. You will be asked to enter and confirm a password for the LoggerNet user account. The password you enter can later be used to log in to the computer as the LoggerNet user. The password can be changed by pressing the **Options** button and then pressing **Change Password**.

> **NOTE:** If the machine has a preexisting LoggerNet user account created from LoggerNet 3.4, it will be necessary to stop, uninstall the service, then reinstall the service in order to setup the password.

> **NOTE:** The LoggerNet user account will not show up in your list of users when logging on to your computer. It can be viewed from the Windows Control Panel.

2.4.1 Issues with Running LoggerNet as a Service

AVAILABLE IN LOGGERNET ADMIN ONLY

By default, the LoggerNet service is run under the LoggerNet user account. This may cause some issues with write access and network drives. The issues, along with their solutions, are described below:

Write Access

The LoggerNet user has write access only to the CampbellSci directories. Therefore, if a task requires something to be written or done in a different directory or on the desktop, the LoggerNet user does not have sufficient access and the process will end in an error. This can be solved by giving the LoggerNet user write access to the necessary directories.

Giving the LoggerNet user write access

This is the process for giving the LoggerNet user write access to a designated directory in Windows 10. The process in other operating systems is similar.

- Right-click on the directory in Windows Explorer and choose Properties.
- Go to the Security tab of the Properties dialog box and select **Edit**. In the Permissions for directoryname dialog box, press **Add**. This will open the Select Users, Computers, Service Accounts, or Groups dialog box.
- From the Select Users, Computers, or Groups dialog box press the **Locations** button. This will open the Locations dialog box.
- In the Locations dialog box, select the computer name and press **OK**.
- From the Select Users or Groups dialog box press the **Advanced** button. Then press the **Find Now** button. Select LoggerNet in the list of names that appears at the bottom of the dialog box and press **OK**. Note that <COMPUTER-NAME>\LoggerNet has been added to the Object Names on the Select Users or Groups dialog box. Press the **OK** button to close the Select Users or Groups dialog box.
- The LoggerNet user should now be highlighted on the Permission for directoryname dialog box. Select the Full Control Allow check box to give LoggerNet full permissions for the directory. Press the **Apply** button to apply the changes. Press **OK** to close the dialog box.
- The LoggerNet user should now have full access to the designated directory.

Network Drives

Network drive mappings are associated with individual user accounts. Therefore, they cannot be used when running LoggerNet as a service. To use network drives when running LoggerNet as a service, you must use the full UNC path (e.g., \\computer_name\directory\filename). Note that the ability to write to the network drives will be governed by Windows security. It will be necessary to configure the LoggerNet service to run under an account that has network privileges. This requires changes to the properties of the LoggerNet service in Windows. Contact your network administrator for assistance.

2.5 Windows Firewall

Microsoft automatically enables a firewall application on each individual PC. This was done to protect PCs from invasion by outside, unauthorized programs that may try to connect via a socket using TCP/IP.

However, LoggerNet is a client-server application that uses TCP/IP as the link between clients and the server. This means that windows such as Setup, Connect, Status Monitor, RTMC, etc., get their access to devices and data only through the LoggerNet server. While, in the basic LoggerNet installation most of these clients typically access the server on the same PC (which will already be "behind" the PC's local firewall), the server is capable of distributing the same information to similar clients connected via TCP/IP from anywhere in the world, revealing the true power of the client-server design. If, for example, you want others to use RTMC or Data Filer to get data from your LoggerNet PC, you can enable this remote connection by selecting the **Allow Remote Connections** check box on the dialog box opened from LoggerNet's **Tools** > **Options** menu item. This causes the server to open a socket on a specific port (default is port 6789) to listen for requests for data from its clients. This is usually quite safe since: 1) no other application should be trying to use this port, and 2) the server will only respond to LoggerNet-specific messages on this port (it will not run viruses or other unauthorized bits of code). If you allow remote connections, however, the firewall in Windows may put up a window telling you that it has blocked LoggerNet Server and asking if you wish to allow LoggerNet Server from Campbell Scientific to communicate and which networks you wish to allow it to communicate on. At this point, if you select the network(s) and click **Allow Access**, Windows will make an exception for LoggerNet and you should not have to unblock it each time you start it. You can reverse this decision in the future from the Windows Control Panel.

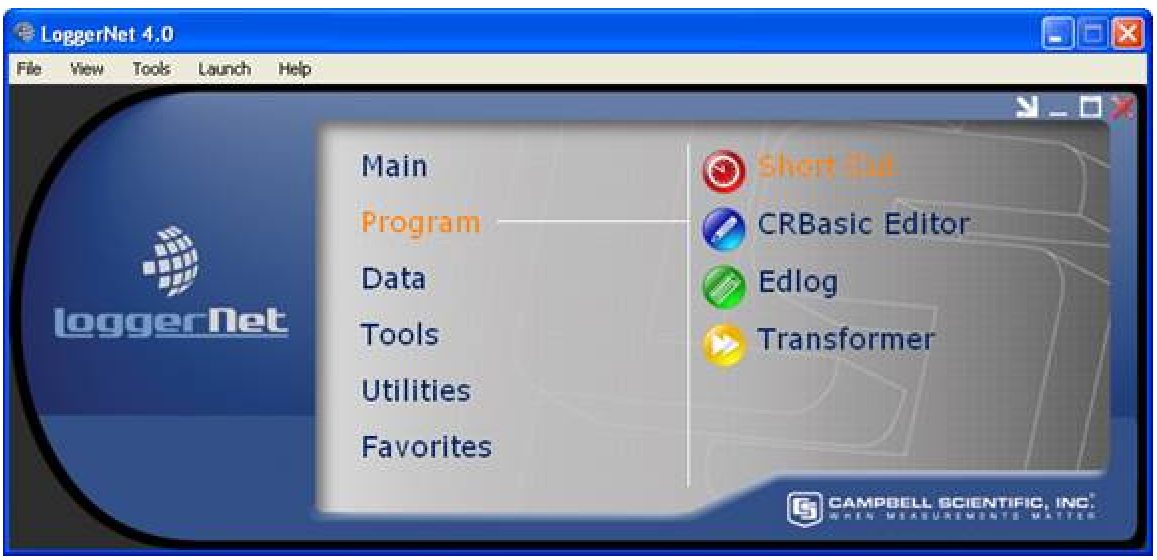
3. Introduction

LoggerNet is Campbell Scientific's full-featured software that supports both our CRBasic- and Edlog-programmed dataloggers. Applications in LoggerNet can be used to set up a datalogger communications network, automatically collect data on a schedule, create datalogger programs, and send these programs to a datalogger. There are tools to manually collect data, view real-time data, view and create reports from data files, and manage peripheral storage devices.

The toolbar is the "heart" of the LoggerNet software. This window provides an easy way to navigate between the applications. As you hover over a category in the list on the left, applications related to that category will be shown on the right. Selecting an application in the right-hand list will launch the application.

Select the Setup, Connect, Status Monitor, or Task Master options above to be taken to the manual section associated with that application. For help on other applications, refer to their respective help files. For additional information on the Toolbar, see LoggerNet Toolbar (p. 20).

By default, the LoggerNet menus are not shown on the toolbar. Press the arrow button in the upper right corner to view the LoggerNet menus as shown below. The arrow button will change direction and can then be used to hide the menus. The menus can also be displayed by pressing and releasing the Alt key.



The Favorites category can be configured to display as many or as few applications as desired by selecting the **Tools | Options** menu item and then using the Favorites tab.

The first step when using LoggerNet is to set up the communication link to your datalogger network. LoggerNet offers an EZSetup Wizard to help you accomplish this task. The first time LoggerNet is opened after installation, the EZSetup Wizard is opened. The Wizard can also be opened from the EZ View of the Setup Screen.

After the communication link to your datalogger network is set up, use one of the programming tools to create a datalogger program and download that program to the datalogger. You can then set up a data collection schedule or collect data manually. Collected data can be further processed using Split or any spreadsheet program.

Client-Server Architecture

The LoggerNet software is written using advanced "client-server" architecture. The server software engine runs in the background handling all of the datalogger communications. The server also takes care of storing the data and providing information to manage the datalogger network. In turn, the client applications connect to the server to access the information collected from the dataloggers.

One significant benefit of the software design is that some of the client applications (RTMC, for instance) can be run on any computer that connects to the main computer by a TCP/IP network connection. Some examples of these networks are Local Area Network (LAN), Wide Area Network (WAN), or the Internet. If you have LoggerNet Admin or LoggerNet Remote, any of the client applications can log on to a remote LoggerNet server. Another benefit is the efficiency that is gained, since several client applications can simultaneously request and receive information from the software server.

LoggerNet is an ideal solution for users wanting a reliable data collection system that is also flexible enough to meet a variety of needs.

LoggerNet Software Suites

Campbell Scientific offers three LoggerNet software products: LoggerNet, LoggerNet Admin, and LoggerNet remote.

LoggerNet is the base product. With it you can communicate with your dataloggers, program them, display data, collect data, and analyze that data once collected. LoggerNet Admin adds capabilities that allow you more control over your datalogger network. You can set up security to restrict certain communication from taking place with the datalogger, prevent changes being made to the network, install and run LoggerNet as a service, and export data to other applications. These functions can be done remotely or on a local machine. LoggerNet Remote includes most of the client applications, but does not include the LoggerNet server. You can use LoggerNet Remote to connect to and manage an existing LoggerNet installation from a remote location.

LoggerNet Admin and LoggerNet Remote also have the ability to launch more than one of the same client screens. In LoggerNet, you can open only one client window at a time. In LoggerNet Admin/LoggerNet Remote, if Launch Multiple Clients is selected on the Toolbar's Option menu, you can open two or more of the same window. For instance, you can open one Connect Screen and connect to datalogger A, and open a second Connect Screen and connect to datalogger B.

Much of this documentation applies to all three of the LoggerNet software products. If a function does not exist, we've tried to point that out (e.g., you may see a note that reads AVAILABLE IN LOGGERNET ADMIN ONLY).

3.1 What's New in LoggerNet?

LoggerNet 4.9

Added support for the CR1000Xe datalogger.

LoggerNet 4.8

One of the main efforts in the development of LoggerNet 4.8 has been to add the CR350 Series datalogger as a unique device. Previously, the CR350 was included as part of the CR300 Series. 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.

> **NOTE:** Starting with this version of LoggerNet, RTMC will no longer be updated if a version of RTMC Pro earlier than 5.0 is installed on the computer. This is to maintain full compatibility with the installed RTMC Pro package.

LoggerNet 4.7

One of the main efforts in the development of LoggerNet 4.7 has been support for the new CR350 Series datalogger.

An Allowed Neighbors tab has also been added to the PakBusPort device. This allows you to specify a list of PakBus addresses that the port will accept as neighbors.

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.

In LoggerNet Admin, you can now add, delete, and edit stations while viewing a subnet. Previously, you had to change your view to the whole network before making station changes.

A Software Updater has been added to LoggerNet and Device Configuration Utility that can automatically notify you and update your software when a new release is available.

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.

RTMC and RTMC Run-Time have been updated to version 5.0.

> **NOTE:** If you are using an older version of RTMC Pro and plan to continue creating RTMC projects, we recommend that you upgrade to RTMC Pro 5.0.

LoggerNet 4.6

One of the main efforts in the development of LoggerNet 4.6 has been support for the new GRANITE Data Logger Modules.

In addition, a Datalogger Network Communications Enabled check box has been added to LoggerNet Settings (p. 45) to enable/disable communications for the entire network. This option has also been added to Restore Network (p. 7).

> **NOTE:** The way custom data collection is performed changed in this version. The changes should be invisible to the user. However, custom collection will not work with the 4.6 LoggerNet server and a Connect screen from an older version of LoggerNet. Therefore, we recommend that LoggerNet Admin and LoggerNet Remote users update their clients and servers to the latest version.

LoggerNet 4.5

One of the main efforts in the development of LoggerNet 4.5 has been support for the new CR1000X 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 LoggerNet, Windows XP and Windows Vista are no longer supported.

LoggerNet 4.4

The main effort in the development of LoggerNet 4.4 has been support for the new CR300 Series datalogger. Support was also added for the CDM-A108 and CDM-A116.

In addition, a basic security username/password was added to control access to the network when Allow Remote Connections is enabled.

A **UDP Search** button was added to the IPPort in the Setup screen to initiate a UDP discovery to search for PakBus dataloggers in the network. Also in the Setup screen, the ComPort now has an **Install USB Driver** button to allow you to install the USB drivers for our dataloggers and peripherals that require them.

The ability to display line numbers has been added to the CRBasic Editor.

LoggerNet 4.3

The main effort in the development of LoggerNet 4.3 has been support for the new CR6 Series datalogger. Support was also added for the CRS451 Series Water Level Recording Sensor and the CRVW Series Vibrating Wire Recording Interface.

LoggerNet 4.2

LoggerNet now supports IPv6 addresses. IPv6 addresses are written as eight two-byte address blocks separated by colons and surrounded by brackets (e.g., [2620:24:8080:8600:85a1:fcf2:2172:11bf]). Prior to LoggerNet 4.2, only IPv4 addresses were supported. IPv4 addresses are written in dotted decimal notation (e.g., 192.168.11.197). Leading zeroes are stripped for both IPv4 and IPv6 addresses. Note that while LoggerNet now supports IPv6 addresses and they can be used to specify servers, CR1000/CR3000/CR800 series dataloggers will not support IPv6 until a future OS release. Check the OS Revision History on our website to determine when IPv6 support is added to the OS. (Starting in LoggerNet 4.2.1, IPv6 connections are disabled by default. They can be enabled from LoggerNet's Tools | Options menu item.)

LoggerNet 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 LoggerNet 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.

The ability to set up subnets of the network map has been added to LoggerNet Admin. The Setup Screen's View | Configure Subnets menu item is used to configure the subnets. Within each subnet, you can also specify groups of dataloggers. The datalogger groups create folders than can be collapsed or expanded when viewing the subnet. Once subnets have been configured, you can choose to view a subnet rather that the entire network in the Setup Screen, Connect Screen, and Status Monitor.

You can now set up defaults for the Setup Screen's Schedule, Data Files, Clock, and File Retrieval tabs that will be used when new stations are added to the network. There is also the ability to copy these defaults to existing stations.

The ability to use 24:00 (rather than the default of 00:00) for the timestamp at midnight has been added. (This is accessed from the button next to the Output Format field on the datalogger's

Data Files tab in the Setup Screen. It is also available in the Connect Screen's Custom Collection options.)

PakBus Encryption is now supported for communication between LoggerNet 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 LoggerNet's Setup Screen. AES-128 encryption is used.

Two new root devices, SerialPortPool and TerminalServerPool, have been added to allow for modem pooling. (Pooled Devices may be useful in cases where LoggerNet is used to call, by phone, multiple remote dataloggers and there is more than one modem and phone line available to make the connections.)

You can now access a datalogger's Settings Editor from the Connect Screen either by right-clicking on the datalogger or from the Datalogger menu. You can also manually set the datalogger's clock from the Connect Screen either by double-clicking in the Station Date/Time field or from the Datalogger menu. Boolean values displayed in the Connect Screen's Numeric Display now have an LED icon next to them to allow for easy toggle.

You can now view additional statistics in the Status Monitor for table-based dataloggers including watchdog errors, skipped scans, and battery errors. (Note that there is a Poll for Statistics check box on the datalogger's Schedule tab in the Setup Screen that must be enabled to poll for these statistics.)

The Task Master has been integrated into the LoggerNet server. This allows for remote administration of the Task Master. (See Select Server in the Task Master (p. 323) help for conditions that must be met for remote administration of the Task Master.)

> **NOTE:** Integrating the Task Master into the server involved extensive changes. When upgrading to LoggerNet 4.2 from a previous version, an attempt will be made to import all previously-configured tasks. However, imports have only been tested back to LoggerNet 3.4.1. After upgrading (from any previous version of LoggerNet), you should verify that all of your tasks have imported correctly.

Calendar-based scheduling has been added to the Task Master. This allows for non-interval task execution (including data collection). See Example #3 in the Task Master (p. 323) help for an example of calendar-based data collection.

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

LoggerNet 4.1

One of the main efforts in the development of LoggerNet 4.1 has been the ability to use LNDB databases with View Pro. The ability to lock the timestamp column on the left of the data file has also been added to View Pro. This keeps the timestamp visible as you scroll through columns of data.

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 CRBasic Editor now 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.

In the Troubleshooter and the Setup Screen (Standard View), you can now click on a potential problem to bring up a menu that allows you to go the Setup Screen or Status Monitor to fix the potential problem, bring up help describing the problem, or in some cases fix the problem directly.

Campbell Scientific's new wireless sensors have been added to the Network Planner.

You can now view statistics (average, minimum, maximum, and number of data points) for each data value being displayed in a Connect Screen graph. All of the Connect Screen's displays (table monitor, numeric displays, and graphs) can now show the units that have been assigned to the data values.

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 LoggerNet has been added to the LoggerNet Toolbar's Help menu.

Miscellaneous other changes and bug fixes have also been implemented in this version.

> **NOTE:** Beginning with this version of LoggerNet, Windows 2000 is no longer supported.

LoggerNet 4.0

LoggerNet 4.0 introduces a new look and feel to the Toolbar. Applications are divided into categories to make navigating the Toolbar easier. You can also organize a Favorites category for the applications that you use most often.

A new file viewing application, View Pro, is introduced in LoggerNet 4.0. View Pro allows you to have multiple data files open at one time. Multiple graphs can be created from the same file or from multiple files. There is no limit to the number of traces per graph. Data can be graphed in a variety of formats including a Line Graph, X-Y Plot, Histogram, Rainflow Histogram, or FFT (2D or 3D).

Another new application, the Network Planner, is included in LoggerNet 4.0. The Network Planner is a graphical application that assists the user in designing PakBus datalogger networks. It allows the development of a model of the PakBus network, proposes and verifies valid connections between devices, and allows integration of the model directly into LoggerNet 4.0.

A File Retrieval tab has been added to the Setup Screen for the CR1000, CR3000, and CR800 Series dataloggers. This tab provides an easy way to retrieve files from a datalogger on a specified interval. A Notes tab has been added to all devices to allow the user to keep notes about the device for future reference. This is purely for the user's convenience. (The information in a datalogger's Notes tab is displayed in the Connect Screen, when that datalogger is selected.)

A new menu item has been added to the Setup Screen to enable a Stations Only view. When this is enabled, only stations will be shown in the Network Map. This can be especially helpful, when working with a large network.

The Connect Screen has been reorganized with most of the buttons now residing on a toolbar at the top of the window. A Table Monitor has been added in the middle of the window that can be used to monitor the values for one entire table.

Notes On Updating From Earlier Versions of LoggerNet

You should be able to update from any 2.x or 3.x version of LoggerNet by simply installing over the existing LoggerNet. Our installation program should be able to find the program and working directories from the Windows registry. You may, alternatively, wish to uninstall the previous version before installing this newest version of LoggerNet. If so, you can enter the same working directory as before (default was c:\campbellsci\loggernet) to ensure the network is brought forward.

LoggerNet 4.x can readily use the network map from LoggerNet 2.x or 3.x. However, network maps are not backwards compatible. If you upgrade your existing version, once LoggerNet 4.x is opened, the network map will no longer be compatible with LoggerNet 2.x or 3.x. For this reason the upgrade installation will automatically make a copy of the <*WorkingDirectory*>\LoggerNet\sys directory and all of its contents. The copy will reside in <*WorkingDirectory*>\LoggerNet\NetworkMapBackup\<*version*>\sys. If it then becomes necessary to revert back to a previous version of LoggerNet, you will need to remove the <*WorkingDirectory*>\LoggerNet\sys directory and replace it with the <*WorkingDirectory*>\LoggerNet\NetworkMapBackup\<*version*>\sys directory.

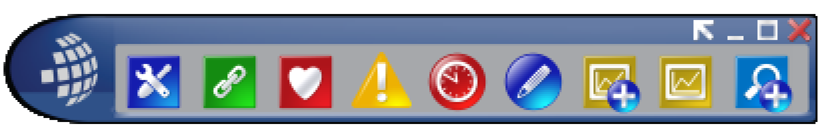
If you are installing LoggerNet and LoggerNet Remote on the same computer (which then becomes LoggerNet Admin), LoggerNet must be installed before installing LoggerNet Remote. LoggerNet Remote can also be installed independently on a computer if you will be accessing a LoggerNet server running on a different computer across a network.

3.2 LoggerNet Toolbar

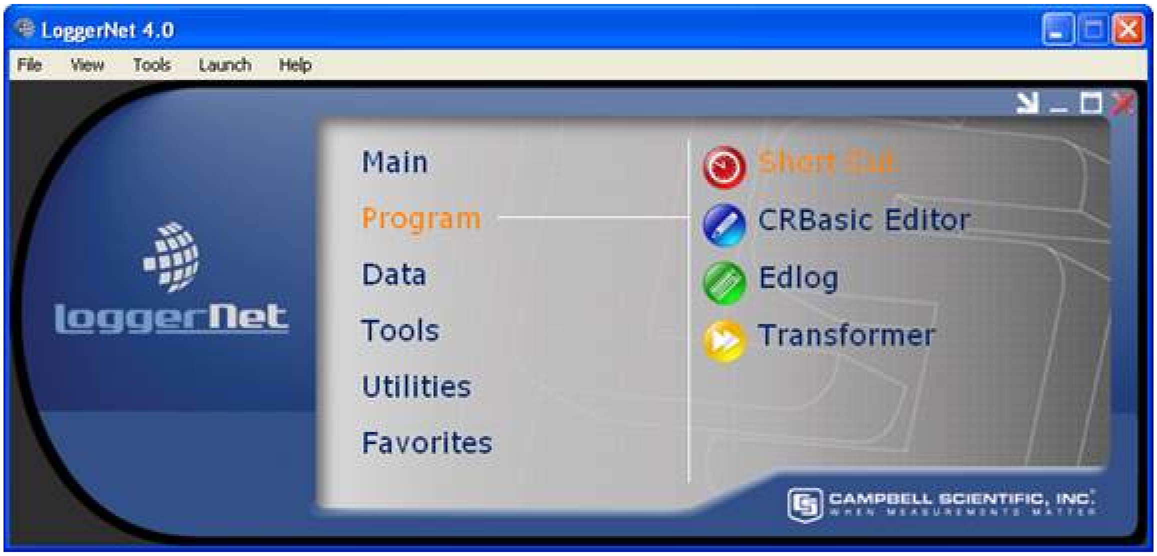
The Toolbar is the heart of the LoggerNet software. When the Toolbar is running, the LoggerNet server is running, and therefore, all scheduled communication with the devices in the network will be carried out.

The Toolbar is used to navigate to all of the applications in the software. As you hover over a category in the list on the left, applications related to that category will be shown on the right. Selecting an application in the right-hand list will launch the application.

If you prefer a smaller version of the toolbar, you can select Favorites View from the View menu. This will switch to a small view of the toolbar containing only icons for applications in the Favorites category.



By default, the LoggerNet menus are not shown on the toolbar. Press the arrow button in the upper right corner to view the LoggerNet menus as shown below. The arrow button will change direction and can then be used to hide the menus.



LoggerNet's menu items are described below.

3.2.1 File Menu

Exit - Closes the LoggerNet application.

3.2.2 View Menu

Full View – This option is only available when in Favorites View and brings up the full view of the Toolbar.

Favorites View – This option is only available when in Full View. It switches from a full view of the Toolbar to a smaller view which shows icons for only the Favorites category.

Hide Main Menu – Hides LoggerNet's main menu. The main menu can be displayed again by pressing the arrow key in the upper right-hand corner of the Toolbar or by pressing and releasing the Alt key.

3.2.3 Tools Menu

Options - This option bring up the LoggerNet Options (p. 24) dialog box. From this dialog box you can specify various options such as whether the toolbar always stays on top, whether legacy applications (Edlog and Transformer) are shown on the toolbar, the behavior of the system tray icon, language, whether to minimize LoggerNet clients when the toolbar is minimized, whether remote connections are allowed, and whether IPv6 connections are allowed. You are also able to specify which applications are included in the Favorites Category.

3.2.4 Launch Menu

Provides a drop-down list of all the categories on the LoggerNet toolbar. Hovering over a category will display a list of applications related to that category. When an application is selected from this list, it will be started.

The following applications are available:

Main

Setup - The Setup Screen (EZ View vs. Standard View) (p. 31) is used to set up your datalogger network.

Connect -The Connect Screen (p. 81) is used to maintain datalogger stations. Programs can be sent to or retrieved from the datalogger, the datalogger clock can be set, and data can be collected manually. Tools are also available to view or graph data.

Status Monitor -The Status Monitor (p. 159) provides information on the communication link and most recent data collection attempts with the dataloggers in the network.

Task Master - The Task Master (p. 323) is used to set up a Task that can be run on a defined schedule or based upon a data collection event from a datalogger. A Task can be data collection from another datalogger, or anything that can be executed in a computing environment (i.e., a command line operation, a program executable, a batch file, or a script).

Program

Short Cut - The Short Cut (SCWIN) program generator guides you through four steps for generating or editing a program.

CRBasic Editor -The CRBasic Editor is used to create programs for the CR1000X, CR6-series, CR300-series, CR350-series, CR1000, CR3000, CR800-series, CR200-series, GRANITE 6, GRANITE 9, GRANITE 10, CR5000, CR9000, and CR9000X dataloggers.

Edlog - Edlog is used to create programs for the CR7, 21X, CR500, CR510, CR510TD, CR510PB, CR10, CR10T, CR10X, CR10XTD, CR10XPB, CR23X, CR23XTD, and CR23XPB.

Transformer - The Transformer is used to convert Edlog programs to CRBasic programs. Support is included for CR10X to CR1000, CR10X to CR800, CR510 to CR1000, and CR23X to CR3000 programs.

Data

RTMC Development - RTMC is used to create graphical displays for real-time data collected from dataloggers.

RTMC Run-Time - The run-time engine for the RTMC program. Use RTMC Development to develop the data displays for this run-time program.

View Pro - View Pro is used to view data files saved to disk.

Split - Split is used to post process and create reports from data files.

CardConvert - CardConvert is used to collect binary data from PC and CF cards and convert the data to ASCII. Binary files stored on some other media may also be converted.

Tools

Troubleshooter -The TroubleShooter (p. 182) application is used to help discover the cause of communication problems in a datalogger network. From the application you can launch a Comm Test (p.

175), search for PakBus devices in the network, view the Status Table for PakBus dataloggers, open the LogTool (p. 172) client, or open PakBus Graph (p. 176).

Network Planner - The Network Planner is a graphical application that assists you in designing PakBus datalogger networks. You interact with a drawing canvas on which you place your stations and add peripheral devices to those stations. You then create links between stations and specify the nature of those links. Finally, you specify the activities that will take place between various devices within the network. As you do these things, the Network Planner automatically species many individual device settings such as PakBus address, neighbor lists, verify intervals, network parameters, etc.

PakBus Graph - PakBus Graph (p. 176) is an application that graphically displays the PakBus devices in the datalogger network of which the LoggerNet server is aware.

LogTool - LogTool (p. 172) is an application that helps you view the communication messages that are passed between the LoggerNet server and the datalogger network.

Security Manager - **(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE)** - The Security Manager is used to set up security access rights for the LoggerNet clients.

Utilities

Device Config Utility - The Device Configuration Utility (DevConfig) assists in the set up of Campbell Scientific dataloggers and peripherals.

Hole Monitor **(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE)** –Hole Monitor is used to monitor the hole collection activity for dataloggers configured for data collection via data advise in a LoggerNet network. A hole is any discontinuity of data in the communication server's data cache. Holes can occur if the server is unable to collect data from a datalogger because of communication failure, or if packets sent to the server from the datalogger are out of order because of a marginal communications link.

> NOTE: The Hole Monitor does not support the CR10X, CR23X, or CR510 mixed array dataloggers (default operating system). In addition, the CR5000, CR9000, and CR9000X are not supported.

CoraScript - CoraScript is a command line scripting utility for managing the dataloggers in a LoggerNet network. Refer to the LoggerNet manual and the CoraScript help file for additional information. (The CoraScript help file, cora_cmd.chm, can be found in the C:\Program Files (x86)\Campbellsci\LoggerNet directory.)

Data Filer **(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE)** – Data Filer is used to retrieve data from the LoggerNet data cache and save the data to a file. It provides a means for a user to manually retrieve and store ASCII data on a remote PC, which can then be used for further analysis.

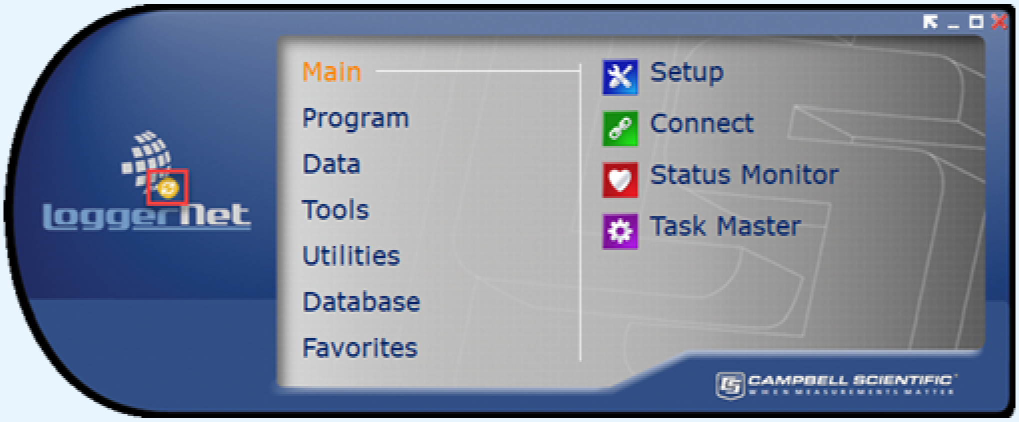
Data Export **(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE)** –Data Export is used to output data stored in the LoggerNet server's data cache to another application. When Data Export acknowledges that a client is available and "listening" on a specified port, the data is transmitted.

3.2.5 Help Menu

LoggerNet Help - Opens the LoggerNet on-line help file.

Check for Updates - Opens the Campbell Scientific Software Updater to check for LoggerNet updates.

NOTE: LoggerNet will automatically check for updates periodically, if Automatic Update Checks are turned on in the Software Updater. If one is available, it will be indicated by the icon on the toolbar highlighted below. The icon will be yellow when updates are available, green when there are no updates available.



Give Feedback on LoggerNet - Opens a form on our website which allows you to provide feedback on LoggerNet to Campbell Scientific.

About LoggerNet - Displays version and copyright information for LoggerNet.

NOTE: All LoggerNet applications will have a similar help menu structure.

3.2.6 LoggerNet Options

The following options can be set by selecting **Tools | Options** from LoggerNet's main menu:

General Tab

View

Stay on Top - This option forces the Toolbar to appear on top of all other applications.

Show Legacy Applications – When this option is selected, legacy applications (Edlog and Transformer) will be shown on the toolbar.

Tray Icon - This option places a LoggerNet icon in the system tray.

Tray Icon Only When Minimized – When this option is selected, the system tray icon will only be visible when LoggerNet is minimized.

Hide Taskbar Icon When Minimized – When this option is selected, the taskbar icon will not be visible when LoggerNet is minimized. Only the system tray icon will be visible.

Language - Certain LoggerNet clients can display the user interface component text (for buttons, dialog boxes, etc.) in an alternate language, if a separate LoggerNet language package has been installed. If a language package is installed on your machine, you will see the language in the list for the Languages menu. When a new language is chosen, the Toolbar will immediately reflect the change. Opened clients will not reflect the change until they are closed and reopened.

Applications that support alternate languages are Setup, Connect, Status Monitor, Task Master, Short Cut, CRBasic Editor, View Pro, CardConvert, TroubleShooter, Network Planner, PakBus Graph, LogTool, the Device Configuration Utility, and Data Export.

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

Behavior

Minimize Clients when Toolbar Minimizes - When this option is selected, minimizing the LoggerNet toolbar also minimizes all LoggerNet clients.

Allow Remote Connections - If client applications running on a remote device need to connect to the LoggerNet server (such as RTMC or LoggerNet Mobile Connect, or when running LoggerNet Admin or LoggerNet Remote), Allow Remote Connections must be enabled. This is done by selecting the Allow Remote Connections menu item. When it is enabled, a check will appear beside the menu item. To disable remote connections, select the menu item a second time.

Though this may be a desirable feature, enabling remote access also makes your LoggerNet network configuration vulnerable to changes by other parties on the network.

Therefore, if using standard LoggerNet (not LoggerNet Admin), we strongly recommend that you select Remote Connection Security Enabled and input (and confirm) a password. (The username cannot be changed.) The username and password will then be required before other parties on the network can connect to the LoggerNet server.

If using LoggerNet Admin, we strongly recommend that the Security Manager be used to set up account access with password protection to limit the ability to make changes to the datalogger network. (The Security Manager can be opened by pressing the **Security Manager** button under Allow Remote Connections.)

Launch Multiple Clients **(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE)** - This option allows you to open multiple instances of a single LoggerNet client (e.g., open one or more Connect Screens and connect to multiple dataloggers). This option is enabled by selecting it from the options menu. When enabled, a check mark will appear to the left of the menu option. It is disabled by selecting it a second time. This feature is available only in LoggerNet Admin/LoggerNet Remote.

NOTE: When launching multiple instances of a client connected to the same server, all instances will use the same *.ini file (initialization file). This *.ini file is saved any time any instance is closed. This means that each new instance will come up with the configuration that was saved the last time any instance was closed. In the Connect screen, for example, this affects such things as the size and position of the window, whether clock updates are paused, whether the data displays are paused, and the station last connected to. Therefore, after closing a Connect screen with paused data displays, when a new Connect screen is launched, it will come up with the data displays paused.

Allow IPv6 Connections - This option enables IPv6 connections. IPv6 addresses are written as eight two-byte address blocks separated by colons and surrounded by brackets (e.g., [2620:24:8080:8600:85a1:fcf2:2172:11bf]). When this option is not enabled, only IPv4 connections are allowed. IPv4 addresses are written in dotted decimal notation (e.g., 192.168.11.197). Leading zeroes are stripped for both IPv4 and IPv6 addresses. This option is enabled by selecting it from the options menu. When enabled, a check mark will appear to the left of the menu option. It is disabled by selecting it a second time.

Favorites Tab

The Favorites tab allows you to specify which applications will be available in the Favorites category. The Available Applications column shows all applications that are available in LoggerNet. (Press the + sign next to a category to show the applications in that category.) An application can be added to the Favorites category by selecting it in the Available Applications column and pressing the right arrow key. An application can be removed from the Favorites category by selecting it in the Favorites column and pressing the left arrow key.

The applications will appear on the Toolbar's Favorites category in the same order as they appear in the Favorites column. The up and down arrow keys can be used to rearrange the order of applications in the Favorites column. To move an application up in the Favorites column, select the application and press the up arrow until the application is in the desired location. Use the down arrow key in a similar manner to move the application down in the Favorites column.

3.3 Command Line Arguments

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

Currently, there are threet command line arguments:

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

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

where "C:\Program Files (x86)\CampbellSci\LoggerNet\ToolBar.exe" is the directory and filename for the LoggerNet Toolbar (which is the LoggerNet server) 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".

/M Launches LoggerNet in a minimized state. Usage:

"C:\Program Files
(x86)\CampbellSci\LoggerNet\ToolBar.exe" /m

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

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

NOTE: If you are running LoggerNet Admin, which requires that you log in to a particular server with each client, you must specify this alternate port number when entering the Server Address in the login window (e.g., LocalHost**:6700,** or 192.168.7.123**:6700, or** [2620:24:8080:8600:85a1:fcf2:2172:11bf]:6700).

NOTE: If LoggerNet is being run as a service (available in LoggerNet Admin/LoggerNet Remote), you must specify an alternate working directory or IP port using the LoggerNet Service Manager.

3.4 Accessing the Help System

There are several ways to access theLoggerNet help system:

- The help file can be opened by choosing **LoggerNet Help** from the Help menu.
- At any time you can press **F1** to bring up help relevant to where you are in the program.
- If the help file is opened, selecting the **Content** tab on the help system's toolbar will return you to a contents page.
- If the help file is opened, choosing the **Index** tab from the help system's toolbar will bring up an index. Keywords can be typed in to search for a topic.
- If a highlighted link takes you to another topic, you can return to the original topic by selecting the **Back** button from the help system toolbar.
- Additionally, CRBasic and Edlog program editors have context sensitive help that can be displayed by right-clicking an instruction or parameter.
- Popup hints are available for many of the on-screen controls. Let the mouse pointer hover over the control, text box or other screen feature; the hint will appear automatically and remain visible for a few seconds. These hints will often explain the purpose of a control or a suggested action. For text boxes where some of the text is hidden, the full text will appear in the hint.

3.5 Campbell Scientific Technical Support

If you should have trouble with the operation of your datalogger hardware or software, contact your local Campbell Scientific Product Dealer or Campbell Scientific directly. Contact information is included with the hardware items shipped with your system.

Access Our Website

www.campbellsci.com

Campbell Scientific regularly announces new products, upgrade opportunities, and other technical information on its website. Also available are software demo packages, tutorials, on-line manuals, and application notes. A support request form is available at this site, or you can email your questions directly to support@campbellsci.com.

4. Getting Started

Step 1: Communication Link Setup

The first step in setting up LoggerNet is to configure the communication link for the devices in your datalogger network. To do this select Setup from the Main category of the Toolbar. By default this will bring up the Setup Screen (EZ View) (p. 31) which uses the EZSetup Wizard to setup a network. To use the Wizard, press the **Add** button, and follow the directions as you move through each window. A help button is provided on each screen if you need additional information.

If you wish to set up the datalogger network yourself rather than using the wizard, press the **Std View** icon at the top right of the Setup Screen to switch to the Setup Screen (Standard View) (p. 33). You will need to add all of the devices in the communication link to your datalogger to the network map on the left side of the window. Add the devices in the order they actually appear in the link. Communications ports (e.g., serial COM ports, Internet ports), are added by pressing the **Add Root** button and selecting a device. The other devices (e.g., phone modems, RF devices, dataloggers) are added by pressing the **Add** button. As each new device is added, the contents of the Add box will change, displaying only those devices which are valid connections to the highlighted device. You can also right click within the network map or on a selected device to display the Add box.

As an example, a phone modem connection to a datalogger would be set up following these steps:

- Click **Add Root** and select a COM port.
- From the Add box, select Communications and then a Phone Base. This device represents the internal or external modem attached to the LoggerNet server computer.
- Now select the Phone Remote from the Add box. This represents the phone modem attached to the datalogger in the field.
- The last device to add is the datalogger. Select a datalogger from the Add box. Press **Close** to return to the main Setup Screen. Press **Apply** on the Setup Screen to enable the changes. LoggerNet will review the changes and display potential network conflicts in a hint box at the bottom of the Setup Screen.

Each tab for each device will need to be set up with the appropriate configuration information. For additional information on the settings for each tab, select your device from see the online help.

> **NOTE:** The order that devices appear in the device map can be changed by dragging and dropping a device, or branch of devices, to a new location.

Step 2: Connecting to the Datalogger

The next step is to test the communication link that has been set up for the datalogger. If you used the EZSetup Wizard to set up communication to your datalogger, you were given the option to test the communication link near the end of the Wizard. If you did not test the link at that time, or if you are setting up the datalogger from the Standard View of the Setup Screen, select the **Connect** option from the Toolbar's Main category to open the Connect Screen. The datalogger(s) that you configured in the Setup Screen will be listed on the left side of the window under the **Stations** field. Click on the datalogger to highlight it, and press the **Connect** button. When a successful connection is made, this button will change to read **Disconnect**.

Once connected, you can confirm that communication with the datalogger has been successful by monitoring the **Station Date/Time** field.

If you cannot connect to the datalogger or errors are returned, check the following:

- The proper cables are being used, and they are in good working order.
- All devices have adequate power.
- No other application is trying to access one of the devices in the communication link.

LoggerNet has a TroubleShooter (p. 182) which may be helpful in assessing communication problems. Also refer to the troubleshooting section of the LoggerNet manual.

For more discussion on the Connect Screen, see The Connect Screen (p. 81).

Step 3: Programming

A datalogger program can be written using Edlog, CRBasic, or Short Cut for Windows (SCWin). Use the respective program for Edlog-programmed Dataloggers or CRBasic-programmed Dataloggers, or use SCWin to generate programs for either type of datalogger. Refer to the CRBasic, Edlog, or SCWin help files for additional information on using these programs.

After a program is created, return to the Connect Screen to transfer the program to the datalogger using the **Send Program** button. To verify that the datalogger program is operating as expected, you can look at the data by launching one of the three Numeric Displays. For additional information on using a Numeric Display, see Numeric Display (p. 94).

Step 4: Collecting Data

Data can be collected from the datalogger by manually initiating a data collection or by setting up a schedule for automatic data collection.

Manual Collection

To collect data manually from a datalogger, open the Connect Screen and connect to the datalogger. You can collect all data that has not yet been collected to a file by pressing the **Collect Now** button. If you want data other than what has not yet been collected, choose the **Custom** button. Custom collect options are different based on the type of datalogger. For information on these options, see the online help.

> **NOTE:** Manual data collection for storage modules is best accomplished using SMS. SMS operates only over a direct COM port link using either an SC532 or a datalogger. (SMS can be downloaded from our website, www.campbellsci.com/downloads. For more information on SMS, refer to the SMS help file.) If your only access to the Storage Module is over a telecommunication link, the **Collect Now** button on the Status window can be used to collect the data.

Scheduled Collection

To set up an automatic schedule for data collection, return to the Setup Screen and highlight the datalogger by clicking it. All datalogger types have a Schedule tab that is used to define the time interval for automatic data collection. The Scheduled Collection Enabled check box on this tab must be selected for scheduled data collection to occur. Each datalogger also has one or more tabs that are used to define what data is collected from the datalogger and where it is stored (Final Storage Area or Data Files). Complete the information on these tabs. For additional information, refer to the online help.

After Data Collection…

Once you've got your data, you can use View Pro to view the data in tabular format or using graphical displays. For more information on View Pro, refer to the View Pro Help file.

Need to do calculations on your data? Or process it into other formats? Or perform quality checking? You can use Split to post-process data files choosing where in the file to start or stop, convert values into new values, and output the results as new files or reports complete with headers. For more information on Split, refer to the Split help file.

Use RTMC to view data as it's collected, display alarms, and set values in the datalogger. For more information on RTMC, refer to the RTMC Help file.

5. Setting up Datalogger Networks

The EZ and Standard Views of the Setup Screen provide ways to create and maintain the communications link and data collection schedules for a network of dataloggers. The EZ View uses the EZSetup Wizard which walks you through the setup step-by-step. In the Standard View, you add devices and configure their settings on your own. Either method will result in a network map with all of the devices and communications links to reach the datalogger stations.

The ***Network Planner*** is a graphical application that assists the user in designing a PakBus datalogger network.

The ***Device Configuration Utility***, or ***DevConfig***, is a stand-alone tool that can be used to configure settings in the dataloggers themselves, as well as in communication devices such as RF401A radios or NL201s.

5.1 Setup Screen (EZ View vs. Standard View)

By default, the Toolbar's Setup option brings up the EZ View of the Setup Screen. The Setup Screen (EZ View) (p. 31) uses the EZSetup Wizard to setup your datalogger network.The wizard is designed to walk you step-by-step through the setup and configuration of your datalogger network.

If you prefer to setup your network without using the EZSetup Wizard, you may press the Std View icon at the top right of the Setup Screen. This will bring up the Setup Screen (Standard View) (p. 33) which allows you to setup your network manually.

Pressing the EZ View icon at the top right of the Standard View of the Setup Screen allows you to switch back to the EZ View.

The Toolbar's Setup option will bring up the EZ View or the Standard View of the Setup Screen depending on which was last used.

5.2 Setup Screen (EZ View)

The EZ View of the Setup Screen uses the EZSetup Wizard to walk you through the setup and configuration of your datalogger network, which is the first step in using LoggerNet with your dataloggers.

As you work through the steps for the Wizard, each screen has fields that are completed with the pertinent information about your station. After you have entered information for each field, press the **Next** button to move on to the next step. Once you have moved to the next step, you may return to any previous step(s) by pressing the **Previous** button.

Help is provided for each step of the Wizard. This is accessed by pressing the **Help** button.

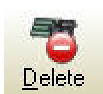
After you have progressed through all steps of the Wizard, press the **Finish** button to save the configuration. To add another datalogger, press the **Add** button to open the Wizard again. If you want to change something in the datalogger setup, select that datalogger from the main window, and press the **Edit** button. The Wizard will open, and you can move through the steps using the **Next** button, or you can navigate quickly to a particular page in the Wizard by selecting it from the **Progress** column.

A datalogger can also be renamed or deleted by highlighting the station and pressing the appropriate button (**Rename** or **Delete**).

> **NOTE:** Alternately, the Setup Screen (Standard View) (p. 33) can be used to set up the datalogger network.

Toolbar Buttons

Opens the EZSetup Wizard to adds a station to the network.



Deletes the highlighted station.

Opens the EZSetup Wizard to edit the highlighted station.

Allows you to change the name of the highlighted station.

Opens the Standard View of the Setup Screen.

Menu Items

File

Select Server - Allows you to select the LoggerNet server to which the Setup Screen should connect **(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE)**.

Exit - Closes the Setup Screen.

View

EZ (Simplified) – Opens the EZ View of the Setup Screen.

Standard – Opens the Standard View of the Setup Screen.

Show Hints - When this option is enabled, hints for some fields will be displayed when the mouse is used to hover over the field. A check mark appears beside the menu item when enabled.

Network

The Network menu contains items with the same functionality as many of the toolbar buttons: **Add**, **Delete**, **Rename**, and **Edit.**

Manual Backup - Saves a copy of the network to a file, which can then be used to restore the network if necessary. The settings for all the devices in the network will be saved. The *.ini files for each of the applications will be also included in the backup. Ini files store settings such as window size and position, configuration of the data displays, etc.

When this menu item is chosen, the subsequent dialog box offers an additional option:

Include Cache Table Files in the Backup - If this check box is selected, LoggerNet Server Data Cache (p. 437) will be included in the backup.

After selecting Next, you will be provided with another dialog box that can be used to add files to the backup file. By default, user files, such as the *.dat files that are created when you collect data from a datalogger, the program files created by Edlog or CRBasic, and the files created when running Split, are not included in the backup. They can be added to the backup from this window if desired. When Next is selected again, you are prompted for a file name for the backup. After supplying a file name, you can choose to execute the backup or save the backup configuration file for later use.

A network backup file can also be created from a task, by LNBackup.exe. See Example #2 under Task Master (p. 323) for additional information on this utility.

Restore Network - Restores the network map from a backup file. Select the name of the backup file to restore from the first dialog box, and press Next.

Select whether or not network communications will be enabled, when the network is restored. Clear this checkbox if the network has scheduled collection enabled, but you do not want it to start as soon as the network is restored. Communications can later be enabled with the Setup Screen > Tools Menu > LoggerNet Server Settings > Datalogger Network Communications Enabled checkbox.

Press the **Restore** button.

> **NOTE:** This backup will replace the existing network (it does not add to the existing network) and any existing files, including data files, program files, or Split files if they were included in the original backup.

Help

Setup Screen Help – Opens LoggerNet's online help system to Setup Screen specific help.

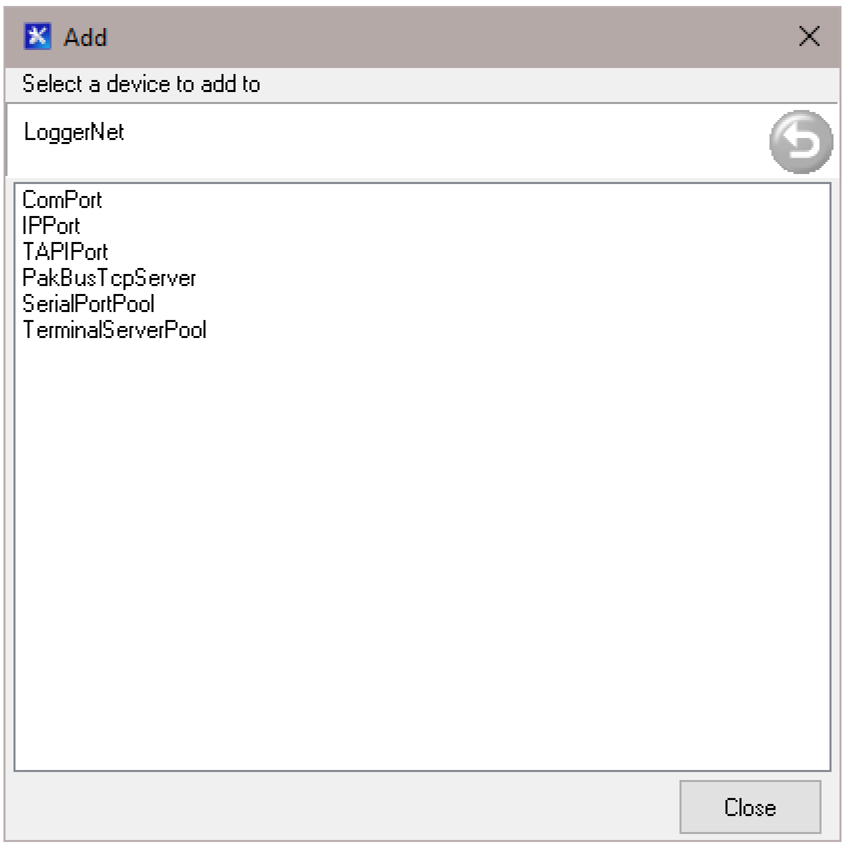
About Setup Screen – Provides copyright and version number information for the Setup Screen.

5.3 Setup Screen (Standard View)

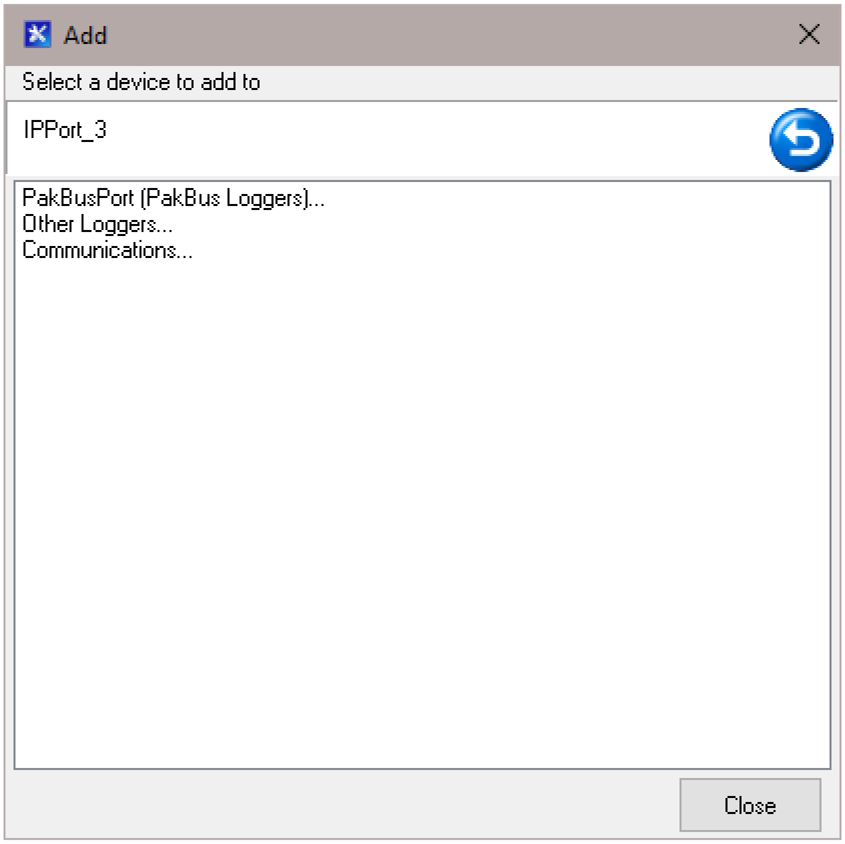
The Standard View of the Setup Screen is used to manually configure the datalogger network that will be monitored using LoggerNet. The datalogger network can consist of one datalogger connected directly to a computer, or many dataloggers connected by more sophisticated means of communication such as phone modems, RF modems, or TCP/IP.

A network is configured by adding each device to the network map in the order that it appears in the actual communication link. For instance, if you are communicating with a CR6 using an IPPort, your network map would consist of an IPPort, a PakBus port, and the CR6:

To add a communication port, press the **Add Root** button and select a device.

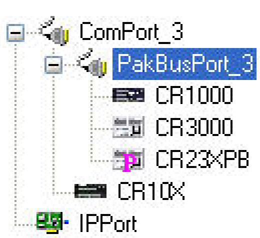


Once a device is added to the network map, the devices shown in the Add box will change, displaying only those devices which are valid connections for the highlighted device in the network map. Continue selecting devices from the Add box until the communication link to the datalogger is represented.

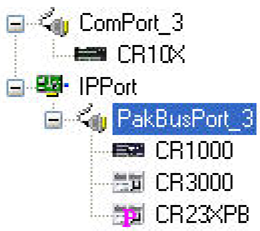


To add a device to an existing device in the network map, highlight the device and press **Add**. Devices can also be added by highlighting an existing device in the network map and pressing the right mouse button. A list of valid devices that can be added will be displayed.

Devices (or device branches) can be moved by clicking on a device (or the top device in a branch) and dragging and dropping it to a new location. For example, in the network map below, the PakBusPort and all of the dataloggers connected to the PakBusPort can be moved from the ComPort to the IPPort by clicking on the PakBusPort and dragging it to the IPPort.



This results in the network map shown below.



A device (and all of its settings) can be copied by selecting the device and pressing Ctrl-C, then pressing Ctrl-V at the position you wish the device to be copied to. Device branches can be copied by selecting the top device in the branch and pressing Shift-Ctrl-C, then pressing Ctrl-V at the position you wish the branch to be copied to. Note that when pasting a root device, you must have a root device selected in your network. The root device being pasted will be inserted in your network below the selected root device. Also note that when using the **Undo** button (or Ctrl-Z) to undo a copy, each setting for each device will be undone separately. Therefore, it may require pressing the **Undo** button several times to actually undo the copy. Thus, it may be easier to use the **Delete** button to remove the device(s).

Ctrl-C (or Shift-Ctrl-C) and Ctrl-V can be used to copy a device (or device branch) to a different LoggerNet network map. Use Ctrl-C (or Shift-Ctrl-C) to copy the device (or device branch) as described above. Open a basic text editor, press Ctrl-V, and then save the file in plain text. Open the text file on the computer where you want to copy the device (or device branch) to, select the entire contents of the file, and then press Ctrl-C. On the LoggerNet network map, select the position you want the device (or device branch) to be copied to and press Ctrl-V.

Applying Changes, Undo, and Redo

The device map is not saved or entered in LoggerNet until you click the **Apply** button at the bottom of the screen. You can build a complete network and set up the configurations for all of the devices without applying. However, it is a good idea to build the network map in stages and periodically apply changes. If there is a problem with the computer, any changes that have been applied have been saved and will not have to be entered again.

Changing the network map or any of the device settings enables the **Undo** button. Clicking the **Undo** button will roll back each change in reverse order to the originally saved network and settings. If you undo a change and really wanted to keep it, you can click the **Redo** button and restore the change.

Once the changes to the network map and device settings have been applied, they can no longer be rolled back or restored using the **Undo** or **Redo** button.

Clicking the **Cancel** button before changes are applied will undo all of the changes to the network map and settings, and restore the saved configuration.

Renaming Network Devices

The names of all of the devices can be changed as desired. Rename a device by selecting the device and either clicking again with the left mouse button on the selected device or clicking the **Rename** button. The name of the selected device will change to a text edit box and the new device name can be entered. Valid names consist of letters, numbers and the underscore (_). The device name must be unique in the network and the first character must be a letter.

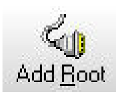
Device names can reflect a location, layout, or physical location of network devices. Think carefully when naming the devices since these names are used throughout LoggerNet to refer to the devices.

> **NOTE:** Alternately, the Setup Screen (EZ View) (p. 31) can be used to set up the datalogger network.

Toolbar Buttons

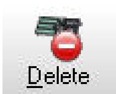
Allows you to choose to view the entire network or a subnet configured using View | Configure Subnets. AVAILABLE IN LOGGERNET ADMIN ONLY

Allows you to choose to view all devices or stations only. Note that when the Stations Only view is enabled, you will not be able to Add, Delete, or Rename devices.



Adds a communication port to the device map.

Adds a datalogger, modem, or storage device to the device map.



Deletes the highlighted device. If the device has other devices attached to it, they will also be deleted.

Allows you to change the name of the highlighted device.

Reverses any changes that were made since the **Apply** button was pressed.

Reverses the last undo operation.

Opens the EZ View of the Setup Screen.

Menu Items

File

Select Server - Allows you to select the LoggerNet server to which the Setup Screen should connect **(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE)**.

Exit - Closes the Setup Screen.

View

EZ (Simplified) – Opens the EZ View of the Setup Screen.

Standard – Opens the Standard View of the Setup Screen.

Configure Subnets – Opens the Datalogger Subnets dialog box that allows you to configure subnets of your network. You can then use the **Subnet** button to choose to view your entire network or a subnet. **(AVAILABLE ONLY IN LOGGERNET ADMIN)**.

Show Hints - When this option is enabled, pop-up hints are displayed when you hover over buttons on the Setup Screen's toolbar. A check mark appears beside the menu item when enabled.

Backup

Scheduled Backup - Saves a copy of the network to a file on a specified interval. The backup file can then be used to restore the network if necessary. The settings for all the devices in the network will be saved. The *.ini files for each of the applications will be also included in the backup. Ini files store settings such as window size and position, configuration of the data displays, etc.

When this menu item is chosen, the subsequent dialog box allows you to specify the backup base date and time, the backup interval, any additional files to be backed up, whether the backup includes the LoggerNet cache, the name of the backup file, and the number of backup files that will be kept before the oldest is overwritten.

Manual Backup - Saves a copy of the network to a file, which can then be used to restore the network if necessary. The settings for all the devices in the network will be saved. The *.ini files for each of the applications will be also included in the backup. Ini files store settings such as window size and position, configuration of the data displays, etc.

When this menu item is chosen, the subsequent dialog box offers an additional option:

Include Cache Table Files in the Backup - If this check box is selected, the LoggerNet Server Data Cache (p. 437) will be included in the backup.

After selecting Next, you will be provided with another dialog box that can be used to add files to the backup file. By default, user files, such as the *.dat files that are created when you collect data from a datalogger, the program files created by Edlog or CRBasic, and the files created when running Split, are not included in the backup. They can be added to the backup from this window if desired. When Next is selected again, you are prompted for a file name for the backup. After supplying a file name, you can choose to execute the backup or save the backup configuration file for later use.

A network backup file can also be created from a task, by LNBackup.exe. See Example #2 under Task Master (p. 323) for additional information on this utility.

Restore Network - Restores the network map from a backup file. Select the name of the backup file to restore from the first dialog box, and press Next. Select whether or not network communications will be enabled, when the network is restored. Press the Restore button. Note that this backup will replace the existing network (it does not add to the existing network), and any existing files, including data files, program files, or Split files if they were included in the original backup.

Tools

LoggerNet Server Settings

LoggerNet Settings: Used to select what time option will be used for the LoggerNet server. Options are to use GMT, or use local server time with or without adjustment for daylight saving time. Also used to set the maximum size for data files and to enable/disable communications for the entire datalogger network.

PakBus Settings: LoggerNet can be configured so that all PakBus ports in the network map are part of one PakBus network, or so that each PakBus port in the network map will be an independent PakBus network. This menu item is used to set up this aspect of LoggerNet and assign either a global PakBus port ID for LoggerNet or IDs for each PakBus port. Valid PakBus IDs are 1 through 4094, though typically numbers greater than 3999 are used for PakBus ports. This is because, when a neighbor filter is set up, a PakBus datalogger will answer a Hello message from any device with an ID greater than 3999, but will ignore devices with IDs less than 4000 that are not in their neighbor list.

LoggerNet Defaults: This option is used to set default values for the Schedule, Data Files, File Retrieval, and Clock tabs that will be used when new devices are added to the network. The dialog box can also be used to copy the specified defaults to existing stations.

Copy Device Settings- Opens a dialog box that allows you to copy settings from one device in your network to other device(s) in your network.

TroubleShooter - Opens the TroubleShooter (p. 182) application. This application is used to help discover the cause of communication problems in a datalogger network.

Help

Setup Screen Help – Opens LoggerNet’s online help system to Setup Screen specific help.

About Setup Screen – Provides copyright and version number information for the Setup Screen.

Network Analysis

In addition to the TroubleShooter, the Setup Screen has two problem boxes that display potential problems with the network.

The first is a device-specific problem box that is shown at the bottom of the Setup Screen. If no problems are detected for a device, the text in the box will read "No problems found with settings for the selected device." However, if some potential device conflict or other problem is found, it will be indicated here. Some instances for which a message will be displayed are: if communication is disabled for a device, if a device has the same PakBus ID as another device in the network, or if a COM port is placed in the device map twice. Clicking on a potential problem will bring up a menu that allows you to fix the potential problem or get additional help.

The second problem box provides a list of problems found with all devices in the network map. To display this information, press the **Check** button. Any potential problems that exist with the network will be listed, including the use of conflicting communication ports, duplicate device IDs, and disabled scheduled data collection. Clicking on a potential problem in the list will bring up a menu that allows you to fix the potential problem or get additional help.

5.3.1 Device Settings

When you highlight any device on the network shown on the left side of the Setup Screen, configuration tabs appear on the right side with the relevant settings. These settings are different for different devices and are described in detail below. Some of the tabs have a **Standard** section and an **Advanced** section. The **Standard** section contains information that must be reviewed to ensure it matches the settings for the device. The **Advanced** section contains settings that can be left at the default for most applications.

All devices have a **Notes** tab which is only for the user's convenience. It may be used to keep notes about the device for future reference.

As with changes to the network map, the changes made to the device settings are not used until they have been applied.

For information on the settings for a specific device, see the LoggerNet help file.

5.3.2 Scheduled Data Collection

5.3.2.1 Data Collection Scheduling Considerations

One of the goals in datalogging applications is to retrieve the data from the datalogger memory to a computer so that it can be analyzed further. LoggerNet can be used to retrieve the data from the datalogger manually, on demand, or you can set up an automatic data collection schedule. When the schedule is enabled and LoggerNet is up and running, the LoggerNet server will initiate calls to the datalogger on the defined schedule and collect its data. Remember that what data is collected, and how it is stored, is configured on the datalogger **Data Files** tab. When the data is collected, it is stored to a file (unless the **No Output File** option is chosen when setting up what data to collect), and it is also stored in the LoggerNet data cache. LoggerNet client applications, such as the Numeric and Graphical Displays or RTMC, retrieve their information from this data cache.

In some cases, LoggerNet client applications display only data that has been collected. In other cases, LoggerNet initiates the retrieval of data to display. These cases are described below:

Numeric Displays and Graphs (opened from the Connect Screen)

When connected, data from table data dataloggers is updated based on the **Update Interval**. (This is referred to as real time monitoring.) Note that data can be updated no faster than the data values are being generated by the datalogger. When not connected, data from table data dataloggers is updated only as often as data collection is performed. (This is referred to as passive monitoring.) Therefore, for input locations or public variables to be updated when not connected, they must be included for scheduled collection.

RTMC

In RTMC, data displays will be updated no more frequently than data is being collected from the datalogger, either manually or on a schedule.

5.3.2.2 Intervals

One of the most significant considerations for setting up data collection is all of the intervals associated with reading, storing, and retrieving data. The intervals and their significance for data handling are described below.

Datalogger Program Intervals

There are two types of intervals written into the datalogger program which affect the availability and collection of data:

- Program Execution Interval – The execution, or scan, interval determines how often the datalogger carries out the instructions in the datalogger program. It is specified in seconds and determines the fundamental rate at which data is available. In typical programs the sensor readings are taken at this rate and the values are stored in corresponding Input Locations or variables. This execution interval is the fastest that data measurements can be updated and data stored. (Depending on how the program is written, sensor readings may occur at specified intervals and not on every program execution.)
- Table Storage Interval – Most data tables are set up to store data records at regular intervals. The data record consists of a record number and time stamp, followed by the output processing (i.e., sample, average, min, max, etc.) of the variable values. This interval must be a multiple of the program execution interval or storage intervals will be skipped. For example, if the program execution interval is 5 seconds and the table interval is set to 3 seconds, there will only be an entry in the table every 15 seconds. The interval specified determines the fastest rate the server can collect new data that is stored to the datalogger final storage memory.

Data Collection Setting Intervals

The collection interval at which the LoggerNet server requests new data from the datalogger is set up on the **Schedule** tab for that datalogger in the Setup Screen. If the collection interval is faster than the rate at which data is being stored to a data table by the datalogger program, data will not be collected every call, but only at intervals when new data is available.

If data collection is enabled for Public tables, the current values will be collected every time a scheduled collection occurs, whether or not the values have been updated. Therefore if your scan interval were one minute in the datalogger program, and the data collection interval were 30 seconds, you would get two identical records in the Public table for each one minute scan.

Communications Path Considerations

When setting up data collection intervals for the dataloggers you should consider the communications path to the datalogger, and its affect on how often you can retrieve data.

For example, RF networks with repeaters also add time delays since each radio must be contacted and then pass the message on to the next RF radio and so on. Each of these operations takes time so the time schedule for RF networks with repeaters should allow enough time for the link to be established with the datalogger and collect the data.

Consideration should also be given to other operations such as clock sets, program downloads, or manual data requests to the datalogger. These require significantly more time and can affect RF network responsiveness.

5.3.2.3 Setting Up Scheduled Data Collection

The data to be collected and the output file locations and format are specified on the datalogger **Data Files** tab. The **Schedule** tab is used to define the interval on which the LoggerNet server will check the datalogger for new data. If new data exists, it will be stored in the data files and the LoggerNet data cache.

To set up a data collection schedule for a datalogger, first ensure that your device map has been configured with all of the devices listed as they actually exist. Next, determine which tables should be collected from the datalogger each time a data collection attempt is made. If no tables are selected on the **Data Files** tab, no data will be collected from the datalogger.

You should check the directory path and the data file options to make sure the files are where you want them and in the right format. Note that each table must be configured separately (that is, selected for collection, file name provided, file format specified, etc.).

> **NOTE:** If no table names appear on the **Data Files** tab, click the **Get Table Definitions** button.

The data collection schedule should be set up next. Set the base date and time to when you would like the first data collection attempt to occur and set the interval at which subsequent data collection attempts should occur. Make sure that communications are enabled for all devices in the communications path, and that scheduled collection is enabled. If the initial date and time is set to a time that has already passed, data collection will begin immediately.

The Status Monitor (p. 159) screen can be used to ensure that data collection is occurring on the defined schedule. Some issues will also be identified by the TroubleShooter (p. 182). If data is not being collected, check the following:

- The **Scheduled Collection Enabled** box on the **Schedule** tab for the datalogger must be selected. This turns the schedule "on". You can temporarily disable data collection by clearing this check box and applying the change.
- The tables from which you desire data should be enabled for collection in the **Data Files** tab of the Setup Screen.
- All devices in the communications path to the datalogger must have the **Communications Enabled** check box on the **Hardware** tab selected.
- Unsuccessful attempts to communicate with the datalogger may exhaust the number of Primary Retries specified so that the Secondary Retry interval is in effect. Check the date and time listed for the next data collection in the Status Monitor.
- Look at the collection state data for the datalogger in the Status Monitor. This is displayed as one of four states.
  - Normal collection
  - Primary retry
  - Secondary retry
  - Collection disabled
- Check the Status Monitor window and ensure the **Pause Schedule** check box is cleared.
- Ensure the table definitions have been retrieved from the datalogger and are current.

5.3.3 Setting the Clock

The datalogger **Clock** tab can be used to define a schedule at which an automatic clock check will be performed. The datalogger clock will be set if it varies from the LoggerNet server clock more than the amount of time specified in the **Allowed Clock Deviation** field.

Because it is important to maintain accurate time stamping of your data, there are a few things to take into consideration when setting up a clock check schedule.

Your datalogger clock should deviate no more than ±1 minute per month. Typically, this drift is less than what will be experienced with a personal computer. Therefore, if your computer clock is not synchronized with an atomic clock or other accurate time keeping device, the datalogger clock may be more accurate than the computer clock.

Another point to consider is how the clock checks may affect the timestamp for your data. Let's say, for instance, that you have a data collection schedule of one minute with a clock set if the two clocks deviate more than two minutes. Over time, the clocks may drift sufficiently that the datalogger clock is set. If the datalogger clock is 12:02:00, and the LoggerNet computer clock is 12:04:15 the datalogger clock will be set to 12:04:15. Therefore, there will be no data for the timestamps 12:03 and 12:04. Conversely, if the datalogger clock is a few minutes faster than the LoggerNet computer clock, the result would be duplicate timestamps that contained different data.

When a record is stored in a data table, the timestamp is not stored along with the record. Instead, when data is retrieved from the datalogger, the datalogger uses the timestamp of the last record stored and the table interval to calculate the timestamp for any previous records. This calculated timestamp is then stored by the server as part of the data record along with the other data values when data is collected. Because of this timestamping method, if the datalogger clock is changed such that it passes an output interval a discontinuity could occur in the records that could cause the timestamps to be incorrect.

In table-based dataloggers the record number can be used along with the timestamp to assure that records are in order, and no data has been missed.

Changing the computer system clock while the display screens are running will terminate the connection for most of the screens. This can also affect LoggerNet operations or even crash the program.

5.3.4 Sending a Program to the Datalogger from Setup

In most instances, you will send a new program to the datalogger from the Connect Screen. Once the program is sent from Connect, you can view measurements in the Table Monitor to ensure the program is working correctly. However, for convenience when configuring a datalogger for communication and data collection, a program can also be sent from the Setup Screen.

Select the **Program** tab for the highlighted datalogger. If a program exists in the datalogger that LoggerNet has knowledge of, it will be displayed in the **Current Program** fields. To send a new program, press the **Send** button.

The **Associate** button can be used to associate a TDF file with a datalogger. TDF stands for Table Definitions File. When a program is compiled for a datalogger, a *program_name*.TDF file is created along with the original program file. This file contains the table definitions (table size, variable names, data types, etc.) for that program. Associating the TDF file with a datalogger can be useful if communication is taking place over a slow or unreliable communications link where the attempt to receive table definitions back from the datalogger fails.

5.3.5 Setup Tools Menu

5.3.5.1 LoggerNet Server Settings

LoggerNet Settings

LoggerNet Clock Settings

This box is used to select a time option for the LoggerNet server.

- **Use local time without correction for daylight saving time** - The LoggerNet server will use the clock from the computer on which the LoggerNet server is running. However, if the computer's regional settings are set to adjust for daylight saving time, when the time changes to daylight saving time, the LoggerNet server's time is not affected.
- **Use local time with correction for daylight saving time**- The LoggerNet server will use the clock from the computer on which the LoggerNet server is running. If the computer's regional settings are set to adjust for daylight saving time, when the time changes to daylight saving time, the LoggerNet server's time will be adjusted
- **Use Greenwich Mean Time (GMT)** - The LoggerNet server will use Greenwich Mean Time, regardless of whether daylight saving time is applicable.

Data File Settings

This box is used to set a maximum size, in bytes, for data files. When the maximum file size is reached, the current file will be archived (with an incrementing number and a .backup extension) and a new file will be created. Entering a value of 0 or less indicates that no maximum data file size should be enforced.

Network Communications

The **Datalogger Network Communications Enabled** checkbox is used to enable/disable communications for the entire datalogger network. When this checkbox is cleared, you will not be able to communicate with your stations and scheduled collection will not occur.

LoggerNet PakBus Settings

This dialog box is used to set up the PakBus ID for one or more PakBus ports in LoggerNet. LoggerNet can be configured so that each PakBus port in the network map will be an independent PakBus network (where communication between PakBus devices exists only in that network), or so that all PakBus ports in the network map are part of one PakBus network. To link all of the PakBus networks, select the **Bridge PakBus Ports** check box. The PakBus ID for LoggerNet is then entered into the **PakBus ID for Global PakBus Router** field.

If the PakBus ports are set up as separate networks, the PakBus ID for each PakBusPort is entered in the PakBus ID/PakBus Port table.

Valid PakBus IDs are 1 through 4094, though typically numbers greater than 3999 are used for PakBus ports. This is because, when a neighbor filter is set up, a PakBus datalogger will answer a Hello message from any device with an ID greater than 3999, but will ignore devices with IDs less than 4000 that are not in their neighbor list.

> **NOTE:** The default address for LoggerNet is 4094. It is good practice to set the PakBus address for all PCs that connect to a datalogger network to different PakBus values.

LoggerNet Defaults

This screen is used to set default values for the Schedule, Data Files, Clock, and File Retrieval tabs that will be used when new stations are added to the network.

You can also use the **Copy Defaults to Existing Stations** button to apply these defaults to existing stations. You will be asked to select the default settings to apply and the stations to apply them to.

If you have changed settings on the LoggerNet Defaults screen, you can press the **Restore Original Defaults** button to restore the settings to the original LoggerNet defaults.

> **NOTE:** Not all of the default settings apply to all station types.

Schedule Tab

Scheduled Collection Enabled - Select this check box to enable the data collection schedule that has been set up for the device. Clearing the check box suspends the schedule.

Base Date/Time - Enter a date and a time that the first data collection attempt for the device should occur. If the date and time reflected by these fields has already passed, data collection will be attempted immediately when the schedule is enabled.

Collection Interval - Enter the interval on which data should be collected from the device. The collection interval is relative to the **Base Date** and **Time** entries. For instance, if the Base Time is set at 12:15 and the interval is set for 1 hour, data collection will be attempted at 12:15, 1:15, 2:15, etc. The format for this field is 00 d(ays) 00 h(ours) 00 m(inutes) 00 s(econds) 000 m(illi)s(econds).

Primary Retry Interval - If a scheduled data collection attempt fails, data will be collected on the interval specified by this field. When a call is successful, data collection will resume on the normal schedule. The format for this field is 00 d(ays) 00 h(ours) 00 m(inutes) 00 s(econds) 000 m(illi)s (econds).

Number of Primary Retries - Enter a value for the number of data collection attempts that should be made using the **Primary Retry Interval**. Once the number of primary retries is exhausted, data collection will be attempted on the **Secondary Retry Interval** if it is enabled. Otherwise, data collection attempts will continue to be made on the original calling interval.

Secondary Retry Interval - This retry interval is optional. It can be enabled by selecting the **Secondary Retry Interval Enabled** check box. It is the interval that will be used for data collection attempts once the primary retries have been exhausted. The format for this field is 00 d(ays) 00 h (ours) 00 m(inutes) 00 s(econds) 000 m(illi)s(econds).

Secondary Retry Interval Enabled – Enables the **Secondary Retry Interval**. If this field is disabled, data collection attempts continue on the original calling interval until a successful call is made.

Collect Ports and Flags - Enable this check box to retrieve the status of the datalogger's ports and flags during scheduled data collection. When this information is retrieved it is not saved to a data file, but it is made available for real-time display by other applications such as RTMC.

Stay On Collect Schedule - By default, when LoggerNet has missed a scheduled collection because of some condition (i.e., LoggerNet was closed, scheduled collection was disabled, the schedule was paused from the Status Monitor, etc.), once the condition that prevented collection is no longer true, if an entire collection interval has elapsed since the last collection attempt, LoggerNet will immediately try to perform a collection. In some cases, this may not be the desired behavior. Selecting the Stay On Collect Schedule check box will cause LoggerNet to always wait until the next even **Collection Interval** to perform a collection.

Poll for statistics - The Status Monitor (p. 159) displays information about datalogger data collection and communication status. There are some potential useful statistics (columns) that are available for some dataloggers that are not available for other datalogger types. Sometimes there are statistics obtained automatically as part of data collection for some dataloggers but can be only obtained with additional communication commands for other dataloggers. In this latter case, these statistics are not retrieved by default as users with slow or expensive communication may not wish to incur the additional cost or time associated with the extra commands. In cases where the user does want to retrieve the additional statistics, the Poll for Statistics setting can be enabled to request that the statistics are retrieved. The statistics will be retrieved during scheduled or manual data collection.

When Poll for Statistics is enabled, the Status Monitor can show the following statistics even if the Status Table is not being collected:

	Status Table Values						
Server Statistic (displayed in Status Monitor)	CR1000X CR6 Series CR300 Series CR350 Series GRANITE Data Logger Modules CR1000 CR800 Series CR3000	CR200	CR10X PB CR23X PB CR510 PB	CR5000	CR9000X	CRS451 Series	CRVW Series
Watch Dog Err	WatchDogErrors	WatchDogCnt	Watch Dog	WatchDogErrors		WatchDogCnt	WatchdogErrors
Prog Overrun	SkippedScan	SkipScan	Overruns	SkippedScan	Skipped Scan	SkipScan	Skipped Scan
Low Volt Stopped	Low12VCount						Low10_5V_Count
Low 5V	Low5VCount						
Lith Batt Volt	LithiumBattery*		LithBat	LithiumBattery	Battery	LithiumVoltage	RTC_Battery

NOTE: *For CR300-series dataloggers, a voltage will not be displayed. If the internal lithium battery supplied sufficient power to maintain the clock while external power was absent, the field will display “OK, ON POWER UP.” If the internal lithium battery is missing or failed to supply enough power while external power was absent, the field will display “FAIL, ON

> POWER UP." The LithiumBattery field is only updated on power up, that is, when external power is first applied.

> **NOTE:** If the above Server Statistic columns are not currently being displayed in the Status Monitor, you can add them by selecting Edit | Select Columns from the Status Monitor menu.

Reschedule On Data - When this box is selected, each time that data is received for a station, the data collection schedule will be reset using the current system time as the base time.

In bandwidth constrained networks, particularly those involving RF-TD protocols, One Way Data and Data Advise are the primary means of collecting data from network stations. When these mechanisms are used, users will typically not enable scheduled collection. This has the disadvantage of not providing the information needed by the LoggerNet Status Monitor and Troubleshooter applications to help the user recognize when data collection for a station has fallen seriously behind. By using the Reschedule on Data setting, the data collection schedule can be enabled for a station with the associated interval set to be greater than or equal to the longest expected interval at which data will be sent by the station. Since the schedule is restarted each time that new data is received, scheduled collection will not take place as long as the flow of data continues from that station. If, however, communication is interrupted from the station and no data is received, the server will start a collection attempt when the schedule fires.

Collect Via Data Advise (AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE FOR CR510TD, CR10XTD, CR23XTD, AND CR10T DATALOGGERS IN AN RF TD NETWORK) - When this option is enabled, an agreement is established between the LoggerNet server and the datalogger. As part of the Data Advise agreement, LoggerNet reports the tables that are marked for collection and the datalogger stores that information in memory. When the datalogger receives a communication packet of any kind, it checks for new records in the tables marked for collection. If a new record exists, it is sent by the datalogger to the LoggerNet server.

Data Advise is used within RF telemetry networks to increase the speed of data collection. The RF polling process using the TD-RF ("Time-division polling") PROM or OS can take advantage of the Data Advise agreement to collect data very quickly by broadcasting a communication packet to all dataloggers in the RFBase-TD network concurrently. This broadcast packet triggers all dataloggers to check for and send any new records at once. The records are simultaneously stored in the individual RF remote modems (RFRemote-TD) until retrieved through the RF polling process (initiated by the RFBase-TD).

One Way Data Hole Collection - A discontinuity in collected data is referred to as a hole. One Way Data hole collection does not apply to holes collected during manual data collection or scheduled data collection since the default behavior of those collection methods will always retrieve the records needed from the datalogger and store them in the data cache in sequential order. However, there are settings in LoggerNet and other data collection methods that can produce missing records in the data cache.

One Way Data and Data Advise are two collection methods that can produce holes. These collection methods both rely on the datalogger to send records to LoggerNet. Since the transmission of these records is unacknowledged, there is a possibility that the data will be lost. If LoggerNet doesn't receive a record for any reason, a hole is created. If this check box is selected to enable One Way Data hole collection, LoggerNet will attempt to contact the datalogger and request the missing records. Otherwise, LoggerNet will not attempt to collect records missing from the data cache.

Please note that LoggerNet puts records from One Way Data hole collection in the .dat files as they are received. If there are holes in the data that are retrieved later, the records will not be in sequential order in the .dat file created by LoggerNet.

> **NOTE:** One Way Data hole collection will not occur at a time when doing so would force the communication link to be dialed.

When the Server's Table Definitions are Invalid - Choose the option for the action that should be taken in the LoggerNet Server Data Cache (p. 437) (memory) during scheduled data collection (or upon the arrival of a One Way Data record) when LoggerNet determines that the table definitions it has stored for the datalogger and the table definitions actually in the datalogger do not match.

Automatically Reset Changed Tables - LoggerNet will reset (delete and recreate) any tables that have changed. Unchanged tables will not be reset. Scheduled data collection will continue without action from the user.

Stop Collection Until Manually Updated - Scheduled data collection will be halted until the user manually updates the table definitions (Setup Screen, Data Files tab, **Get Table Definitions** button). At that time, the user will be prompted to Merge or Reset the table definitions.

Background information on table definitions:

The LoggerNet server keeps track of the table definitions for a datalogger in its data cache. If the server recognizes that the table definitions for one or more tables in a datalogger are different than that for the datalogger in the server, it will use one of two methods to resolve the differences: Merge or Reset.

When LoggerNet attempts to **Merge** the changes in the tables into the data cache, it will delete from the data cache and then recreate any tables whose header information is different from the information stored in the table definitions. Tables which have not changed, or whose header information has not changed, will not be deleted and recreated.

When LoggerNet attempts to **Reset** any changed tables, all changed tables will be deleted and recreated, regardless of whether or not the change actually affected the structure of the data fields of the table.

Note that if merging a table affects the structure of the data fields, when data is stored to disk the existing file will be saved with a *.backup extension and a new file will be created. If the change does not affect the structure of the data fields (for instance, only the number of records for a table has changed), the data will continue to be stored in the existing file.

Data Files Tab

> **NOTE: Table Output Format, Table—Collect Mode**, and **Use Reported Station Name** apply only to table-based dataloggers. **Final Storage Output Format** and **Final Storage—Collect Mode** apply only to array-based dataloggers.

Output File Name - Enter the default file name. The setting can contain these predefined symbols that will be expanded by the LoggerNet server at the time the file is opened or created:

%a	LoggerNet working directory. (By default, C:\Campbellsci\LoggerNet.)
%w	LoggerNet server working directory. (By default, C:\Campbellsci\LoggerNet\sys\bin.)
%s	Name of the station.
%n	Name of the table or final storage area.
%%	Substitutes a percent character in the path.

File Output Option - **Append to End of File** adds new data to the end of the existing data file. **Overwrite Existing File** replaces the existing file with a newly created file. **Create Unique File Name** leaves any existing files intact and creates a new file with a unique file name based on the timestamp of the first record that will be written to the file. For example, CR1000_Table1_2008_09_03_1415.dat. **No Output File** results in no data file being written to disk.

> **NOTE:** When appending to a data file, LoggerNet will check to make sure that the existing data file header matches the new data that is to be appended to the file. If the header does not match, LoggerNet may back up the existing data file and then create a new data file.

Table Output Format - The format in which the data file is saved to the computer.

ASCII, Table (no header) - Data is stored in a comma separated format. No header information is included in the file. When this option is selected the browse button to the right of the field becomes available. It launches a dialog box from which to specify if a timestamp and record number should be included, if strings should be surrounded by quotation marks, and whether midnight is specified as 2400 or 0000.

ASCII, Short Header - Data is stored in a comma separated format. Header information for each of the columns is included.

ASCII, Long Header - Data is stored in a comma separated format. Header information for each of the columns is included, along with field names and units of measure if they are available. When this option is selected, the browse button to the right of the field is available. Pressing this button opens a window from which you can specify whether timestamps and record numbers are included with each record and whether midnight is specified as 2400 or 0000.

Binary, 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. When this option is selected, the browse button to the right of the field is available. Pressing this button opens a window from which you can specify whether timestamps and record numbers are included with each record.

Array Compatible CSV - Data is stored in a user-defined comma separated format. When this option is selected, the browse button to the right of the field is available. Pressing this button opens a window in which you customize the data string for the output file. This option can be used to produce output files from table data dataloggers that are similar to those created by mixed array dataloggers.

CSIXML - Data is stored in XML format with Campbell Scientific defined elements and attributes. For additional information, refer to the LoggerNet manual. When this option is selected, the browse button to the right of the field is available. Pressing this button opens a window from which you can specify whether timestamps and record numbers are included with each record and whether midnight is specified as 2400 or 0000.

> NOTE: The station name contained in the data file header information is the name of the station in LoggerNet's device map. It is not the station name returned by the StationName instruction in CRBasic-programmed dataloggers.

Table - Collect Mode - This option determines how data will be collected on the first and subsequent collections.

Data Logged Since Last Collection - If **Collect All On First Collection** is enabled with this option, all data will be collected on the first collection and any new data will be collected on subsequent collections. If **Collect All On First Collection** is disabled, only the number of arrays specified in the **Records to Collect On First Collection** will be collected and any new data will be collected on subsequent collections.

Most Recently Logged Records - During each data collection, the number of records specified in the **Records to Collect** field will be collected.

Collect At Most - During each data collection, LoggerNet will collect the most recently logged records up to the number of records specified in the **Records to Collect** field. In contrast to the **Most Recently Logged Records** mode, the **Collect At Most** mode will not duplicate records previously collected. Therefore, fewer records than the number specified may be collected.

Use Reported Station Name - Enabling this check box will cause the station name from the Status Table to be used in the header of the data files. If this check box is not enabled, the network map station name will be used.

> NOTE: This check box affects only the header of the data files. It has no effect on the filenames.

Final Storage Output Format - The format in which the data file is saved to the computer. All data is retrieved in binary format and then converted by LoggerNet, if necessary, when saved to disk.

ASCII, Comma Separated stores data in a standard format with data points separated by commas. All leading and trailing zeros, unnecessary decimal points, and plus signs are stripped. Each line of data is separated with a carriage return line feed.

Data points in a **ASCII, Printable** format are preceded with a 2-digit ID and a + or - sign. The ID and fixed spacing of the data points make specific data easy to locate on a printed output.

The **Binary** option uses patterns of 0s and 1s to represent each value in the data stream. This format saves space, but must be converted using Split or some other editor before it can be read.

The default format, ASCII, Comma Separated, is most easily imported into other applications. Split can be used to convert Binary or ASCII, Printable data to a comma separated file.

Final Storage - Collect Mode - This option determines how data will be collected on the first and subsequent collections.

Data Logged Since Last Collection - If **Collect All On First Collection** is enabled with this option, all data will be collected on the first collection and any new data will be collected on subsequent collections. If **Collect All On First Collection** is disabled, only the number of arrays specified in the **Arrays to Collect On First Collection** will be collected and any new data will be collected on subsequent collections.

Most Recently Logged Arrays - During each data collection, the number of arrays specified in the **Arrays to Collect** field will be collected.

Clock Tab

Automated Clock Check - A schedule can be set up to compare the LoggerNet server's clock with the datalogger's clock, and automatically set the datalogger's clock if it varies by a certain amount.

Enabled - This check box is used to turn the clock check schedule on or off.

Initial Date/Initial Time - These fields are used to specify when the first scheduled clock check should occur. If the time reflected by these fields has already occurred, a clock check will be performed during the next data collection attempt with the datalogger.

Interval - Enter an interval for how often a clock check should be performed.

Allowed Clock Deviation - Enter the amount of time, in seconds, that the datalogger's clock can differ from the LoggerNet server's clock before the datalogger's clock is corrected. If 0 is entered, the clock will be checked but not set. The Last Clk Chk and Last Clk Diff statistics can be viewed in the Status monitor to determine the time of the last clock check and the amount of deviation when this value is set to 0.

> **NOTE:** A datalogger will not be contacted by LoggerNet only for a clock check. If a clock check interval occurs outside of a scheduled data collection interval, the clock check will be executed the next time data collection is attempted.

WARNING: The Automated Clock Check option should be used with caution since changing the datalogger's clock close to a time when the datalogger should be storing data can result in data with missing or duplicate timestamps. Therefore, the Allowed Clock Deviation should not be set so small that it results in frequent datalogger clock adjustments. Also the clock check interval and/or the scheduled data collection interval should not cause clock checks to occur close to a time when the datalogger should be storing data. (Note that the scheduled data collection interval is important because, as stated above, the datalogger will not be contacted specifically for a clock check. Once a clock check interval passes, the clock check will be performed during the next data collection attempt.)

Also note that when setting the Allowed Clock Deviation, you should take into account the accuracy of the datalogger clock relative to the accuracy of the computer's clock.

WARNING: If the LoggerNet server computer automatically adjusts for daylight saving time, then your datalogger's clock will be adjusted accordingly if the Automated Clock Check is enabled.

File Retrieval Tab

NOTE: The File Retrieval Tab applies only to the CR1000X, CR6 Series, CR300 Series, CR350 Series, GRANITE Data Logger Modules, CR1000, CR3000, CR800 Series, CR10XPB, CR23XPB, and CR510PB.

Retrieval Mode - This option determines the schedule for file retrieval.

Disabled - No files are retrieved.

Follow Scheduled Data Collection - Files will be retrieved on the same schedule that has been set up for scheduled data collection on the Schedule tab. An attempt to retrieve the files will be made at the scheduled time, only if scheduled collection is enabled. However, regardless of whether or not scheduled collection is enabled, an attempt to retrieve the files will take place when a manual poll is performed using a **Collect Now** button, when a call-back occurs, or when a task calls the station.

New Schedule - Files will be retrieved based on the **Base Date** and **Time** and **Collection Interval** defined below. Only the new schedule will trigger file retrieval. Attempts to retrieve files will be made following the new schedule, whether or not scheduled collection is enabled.

Base Date/Time - Enter a date and a time that the first file retrieval attempt for the device should occur. If the date and time reflected by these fields has already passed, retrieval will be attempted immediately when the schedule is enabled.

Retrieval Interval - Enter the interval on which files should be retrieved from the device. The retrieval interval is relative to the **Base Date** and **Time** entries. For instance, if the Base Time is set at 12:15 and the interval is set for 1 hour, file retrieval will be attempted at 12:15, 1:15, 2:15, etc. The format for this field is 00 d(ays) 00 h(ours) 00 m(inutes) 00 s(econds) 000 m(illi)s(econds).

Delete Files After Retrieval - When this box is selected, the files will be deleted from the datalogger after they are retrieved.

Add New - When this button is pressed, a new pattern is added to the list of files to be retrieved. The user must then designate the **File Pattern**, **Output Directory**, **Max Files**, **Force Retrieval**, and **Record If Skipped** fields for this pattern.

Delete - When this button is pressed, the selected pattern is deleted from the list of files to be retrieved.

Edit File Pattern - Specifies a file pattern that will select the files that will be retrieved. Select an option from the drop-down list or type it in directly. This can be an exact filename or it can contain the wildcard characters "*" or "?". The asterisk is able to replace zero or more characters while the question mark replaces exactly one character. The file pattern can also have a prefix indicating the drive from which to retrieve the files. Note that the file pattern is not case sensitive.

Examples

*.jpg retrieves all the jpg files from all available drives

CPU:*.jpg retrieves all of the jpg files from the CPU drive

CRD:test8.txt retrieves the file test8.txt from the CRD drive

test8.txt retrieves any file named test8.txt from all available drives

Output Directory - Enter the directory to store the retrieved files. It can be entered into the field directly, or you can press the browse button to the right of the field to select a path from the Explorer window. The setting can contain these predefined symbols that will be expanded by the LoggerNet server at the time the file is opened or created:

%a	LoggerNet working directory. (By default, C:\Campbellsci\LoggerNet.)
%w	LoggerNet server working directory. (By default, C:\Campbellsci\LoggerNet\sys\bin.)
%s	Name of the station.
%%	Substitutes a percent character in the path.

Max Files - Specifies the maximum number of files that can be retrieved on each retrieval. The newest files will be retrieved.

Force Retrieval - When this box is selected, a file that matches the file pattern will be retrieved regardless of the file's timestamp or whether the file has already been retrieved.

Record If Skipped - When this box is selected, the names and dates of any files that are not retrieved because of the **Max Files** parameter will be recorded and they will not be retrieved later. If this box is not selected, the skipped files can be retrieved in a later attempt.

5.3.5.2 Copy Device Settings

This dialog box allows you to copy setting(s) from the device currently selected in the network map to other devices in the network. The list on the left side shows the settings that can be copied. The list on the right side shows the available devices in the datalogger network. Select the setting(s) you wish to copy and the device(s) you wish to copy the settings to. You can press the **Select All** (or **Clear All**) button under the Selected Settings list to select all (or clear all) of the possible settings. You can press the **Select All** (or **Clear All**) button under the Copy Settings to list to select all (or clear all) of the possible devices. After selecting the desired settings and stations, press **OK** to copy the settings.

5.3.5.3 Troubleshooter

This menu item opens the Troubleshooter application. This application is used to help discover the cause of communication problems in a datalogger network. From the application you can launch a communication test, search for PakBus devices in the network, view the Status Table for PakBus dataloggers, or open the LogTool Client. Refer to TroubleShooter (p. 182) for more information.

5.3.6 Selecting a Remote Server

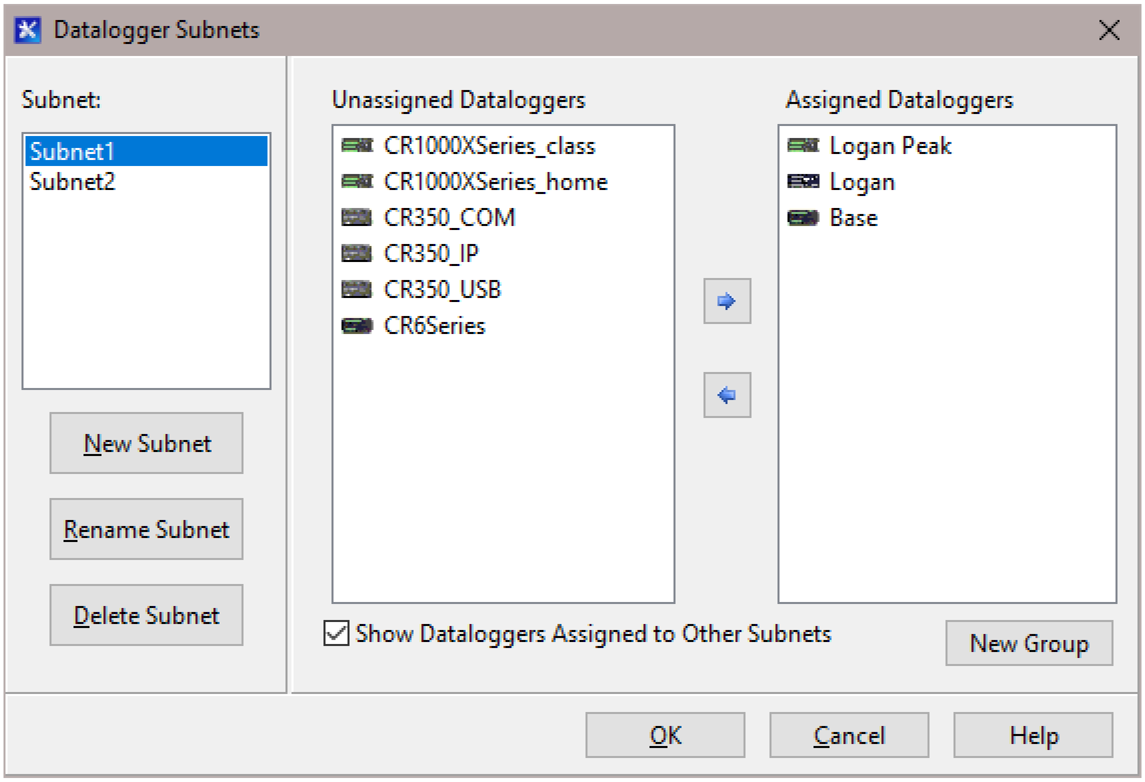
If you are using LoggerNet Admin or LoggerNet Remote, you can connect to a LoggerNet server running on a different computer over a TCP/IP connection. This functionality allows you to add dataloggers, modify the data collection schedule, or make other changes to the datalogger network.

Choose **File** > **Select Server** from the Setup Screen toolbar to open a dialog box in which to type the IP address of the server. If security has been enabled on that server you will need to enter a user name and password as well. This option is not available if you are running a standard version of LoggerNet.

5.3.7 Selecting a View

The **Display** button on the Setup Screen toolbar can be used to determine what is shown in your network map. You can choose to view only your dataloggers by selecting **Stations Only**. Selecting **All Devices** will show your entire network including root devices, communication devices, etc.

In LoggerNet Admin, you can use the **View** > **Configure Subnets** menu item to configure subnets of the dataloggers in your network map. You can then use the **Subnet button** to view a subnet rather that the entire network in the Setup Screen, Connect Screen, or Status Monitor.



Press the **New Subnet** button to add a new subnet. You will be asked to enter a name for the subnet. All of the dataloggers that are not assigned to a subnet will be shown in the Unassigned Dataloggers column. Select a datalogger and press the right arrow key to add it to the current subnet. It will be moved to the Assigned Dataloggers Column. (You can also add a datalogger to the subnet by dragging and dropping it from the Unassigned Dataloggers column to the Assigned Dataloggers column.) You can remove a datalogger from the current subnet by selecting the datalogger in the Assigned Dataloggers column and pressing the left arrow key or by dragging and dropping it from the Assigned Dataloggers column to the Unassigned Dataloggers column. Dataloggers in the Assigned Dataloggers column can be rearranged by dragging and dropping a datalogger to a new location in the list.

The **New Group** button can be pressed to add groups to your subnet. Groups are a way to group dataloggers together within a subnet. Note that groups only show up, when viewing the network map in “Stations Only” view.

To add another subnet, press the **Add Subnet** button again. By default, only the dataloggers that are not a part of another subnet will be shown in the Unassigned Dataloggers column. Select the Show Dataloggers Assigned to other Subnets check box to show all dataloggers.

A subnet can be renamed by selecting it in the Subnet list, pressing **Rename Subnet**, and typing in a new name. A subnet can be deleted by selecting it in the Subnet list and pressing **Delete Subnet**.

Once subnets have been set up, you can use the **Subnet** button in the Setup Screen, Connect Screen, Status Monitor, and Troubleshooter to select whether to view the entire network or a subnet.

> **NOTE:** When viewing a subnet of the network map with Display set to All Devices, you will only see each network branch down to the datalogger(s) included in the subnet. If there are devices (communication devices, other dataloggers, storage modules, etc.) in the branch below the bottom datalogger included in the subnet, they will not be shown when viewing the subnet. This means you will not be able to view a storage module in a subnet.

> **NOTE:** Beginning in LoggerNet 4.7, you can now edit stations while viewing a subnet. (Previously, you had to change your view to the whole network before making station changes.) Adding a device while in a subnet view will add the device to the network and the subnet. Deleting a device while in a subnet view will delete the device from the network and remove it from every subnet it is a member of. Attempting to add or delete devices while in Stations Only view will prompt you to change to All Devices view, but will leave you in the subnet.

5.4 Network Planner

The Network Planner is a graphical application that assists you in designing PakBus datalogger networks. You interact with a drawing canvas on which you place stations and add peripheral devices to those stations. You then create links between stations and specify the nature of those links. Finally, you specify the activities that will take place between various devices within the network. As you do these things, the Network Planner automatically specifies many individual device settings such as PakBus address, neighbor lists, verify intervals, network parameters, etc.

5.4.1 Getting Started

A Network Planner model is made up of stations, links between these stations, and activities that occur over these links.

The first step in developing a Network Planner model is to create the stations. A station is created by adding a root device to the Drawing Canvas (p. 59). Examples of root devices include PakBus dataloggers, the LoggerNet server, as well as radio and phone modems. Once the root device is added to the model, any peripherals that connect to the root device are added. After adding the stations to your network, you must identify links ( ) between stations and the activities ( ) that occur over those links.

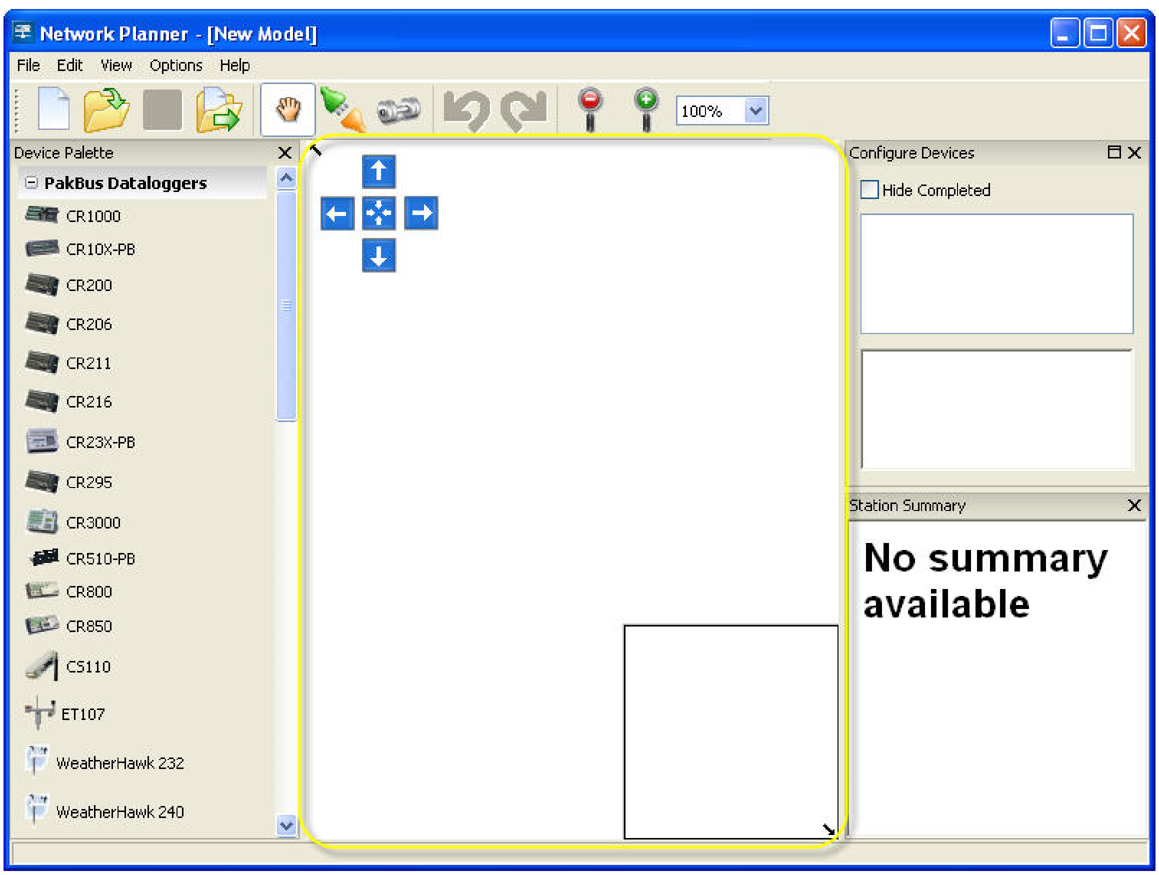
After the stations, peripherals, links, and activities devices and links have been added, all items on the Configure Devices (p. 71) list will need to be completed in order to configure the devices in the system. Check-boxes are provided that allow items to be checked off as an indication that they have been completed.

At anytime, details about a station can be viewed and edited by selecting the station on the Drawing Canvas and viewing the Station Summary (p. 70).

Each of these steps in creating a Network Planner model is described in detail below. Refer to the online help for a step-by-step example of creating a simple Network Planner model.

5.4.2 Drawing Canvas

The Drawing Canvas is the large section in the center of the Network Planner window that is highlighted in the figure below. As stations are added to the network, they are placed on the drawing canvas. As links are added between stations, they are indicated on the drawing canvas as lines between the stations.



5.4.2.1 Adding a Background Image

A background image can be associated with the model by using the **Options | Change Background Image** menu item. If you select this menu item, a file selection dialog will be shown and you will be allowed to select a bitmap image file for the background. Alternatively, the background image can be pasted from the clipboard using the **Options | Paste the Background Image** menu item. This item will only be enabled if there is an image object on the clipboard. You can clear the background image by selecting the **Options | Clear the Background Image** menu item.

Any source for images files can be used for the background image. One example is Google Earth™. You can save a Google Earth™ image to your clipboard by choosing **Edit | Copy Image** from the Google Earth™ menu. Then select **Options | Paste the Background Image** from the Network Planner menu to copy the background image from the clipboard to the Network Planner background.

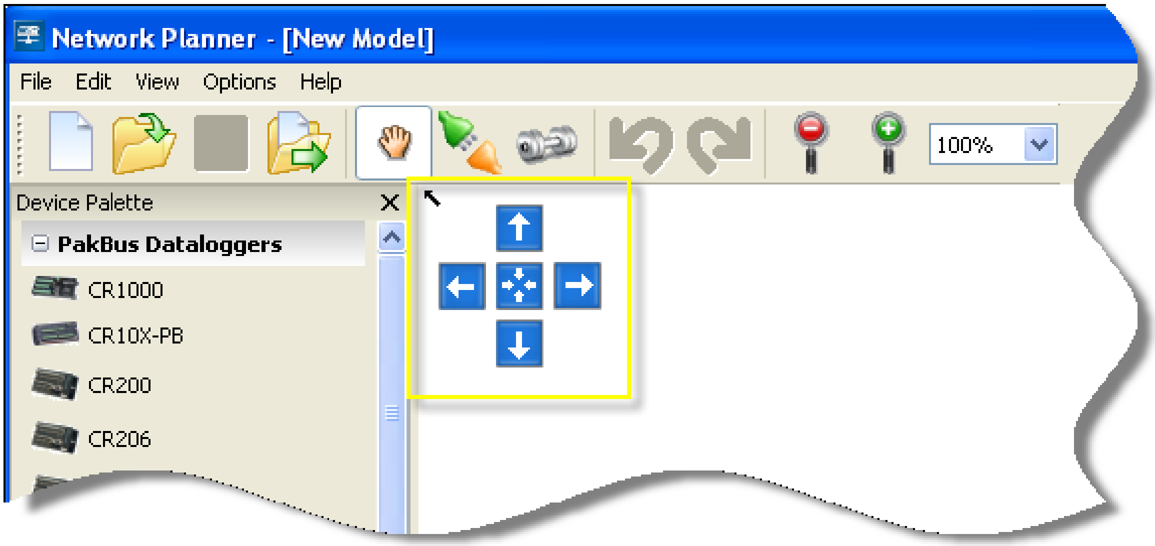
Background images offer significant value in that they allow you to see the layout of your network related to the geography that you are trying to cover. A good image can show landmarks and/or topographical locations that guide placement of things like radio repeaters. That said, the Network Planner does not derive any intelligence from the background image. It is present strictly to satisfy user aesthetics.

5.4.2.2 Scrolling the Drawing Canvas

The drawing canvas is designed so that it has no real boundaries. Rather than using traditional scroll bars, the Network Planner provides two means of changing the canvas viewing window: the scroll buttons and the model overview. In addition to these features, when stations are selected in the Station Device List, the canvas will be scrolled so that the selected station is visible. Finally, you can scroll the canvas directly using the mouse.

Using the Scroll Buttons

Navigation buttons that can be used to scroll the model canvas are highlighted in the image below.



The following buttons and features are associated with this control:

- **Arrow Buttons**—Pressing any arrow button will scroll the drawing canvas in the direction of the arrow.
- **Center Button**—Pressing the center button will reposition the drawing canvas at the graph origin.
- **Collapse Button**—The small black arrow in the upper left hand corner of the scroll button area allows you to hide the navigation buttons. When this is done, the black arrow will remain but point in the opposite direction and, if clicked on, will expand the scroll button area.

Using the Model Overview

The model overview is a small square area in the lower right hand corner of the canvas. If a background bitmap is associated with the model, a miniature version of that bitmap will appear in the overview. The positions of stations on the canvas are represented with red dots. A black rectangle is used within this area to show the current viewing area. If you click the left mouse button while the pointer is anywhere within this area, the canvas will be scrolled so that the center of the canvas is positioned relative to the mouse position within the overview area. If you depress the left mouse button and drag the mouse pointer, the canvas will be scrolled as you move the mouse. If you click on the small black arrow on the lower left hand corner of the overview area, most of the area will be hidden although the arrow will remain with a reversed direction. Pressing the arrow will make the model overview area visible again.

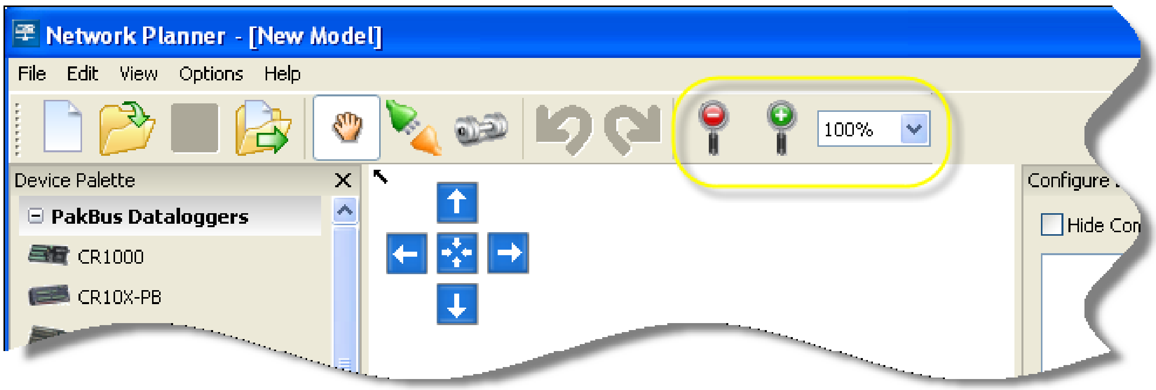
Using the Station Device List

If the station list is shown and you select a station in this list, that station will be selected in the drawing canvas. If the station is not currently visible at that time, the canvas will be scrolled until the station is shown on the canvas.

Using the Mouse

While the Hand Tool is selected by pressing the hand icon on the toolbar, you can scroll the drawing canvas by holding down the left mouse button while the mouse pointer is over a blank area of the canvas and dragging the mouse in the desired direction.

5.4.2.3 Changing the Canvas Scale



The scale of the drawing canvas can be changed using the toolbar controls highlighted in the above figure. The red button with the minus symbol decreases the zoom factor and the green button with the plus symbol increases the zoom factor. The combo-box to the right of these buttons allows the zoom factor to be selected directly. The Network Planner expresses zoom factors as percentages and supports 25%, 50%, 100%, 125%, 150%, 200%, 400%, 800%, 1600%, 3200%, and 6400%.

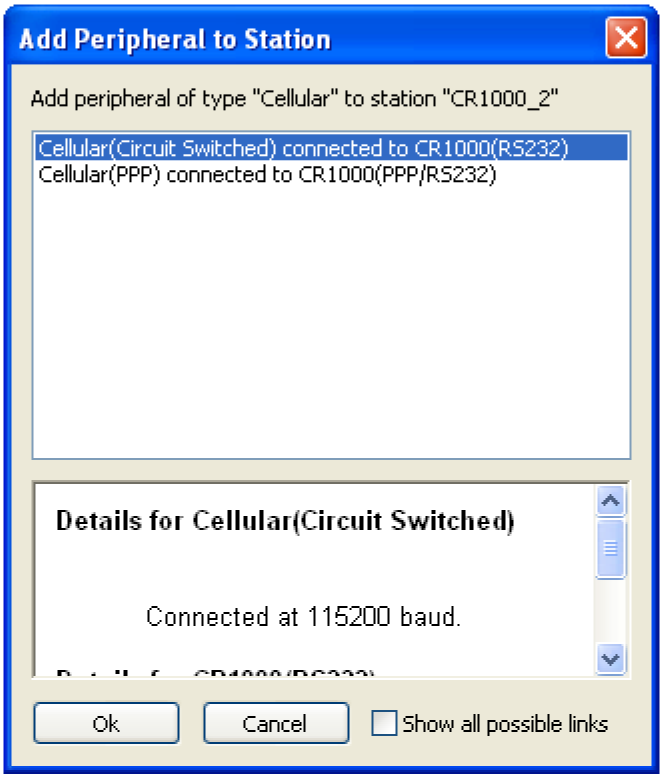
5.4.3 Adding Stations to the Network

A station can be added to the canvas by first clicking on the appropriate device type in the Device Palette. The mouse cursor will then change to a finger pointing to a plus sign surrounded by a square drawn with dashed lines. While the canvas is in this mode, a station will be created when you click on an empty part of the canvas. When this is done, a new icon for that station will appear on the canvas. The appearance of that icon will either be a small image of that station's root device or a balloon shape depending upon the state of the **View | Show Device Images** menu item. The name of the station will be shown under that station. A default, unique name will be generated when the station is first created and can be changed by pressing the F2 key while the station is selected. A station is selected by clicking on it with the left mouse pointer. The selected station will be highlighted in the **Configure Devices** list and will be shown in the **Station Device Summary**.

You can delete a station by pressing the Delete key with the station selected.

5.4.4 Adding Peripherals to a Station

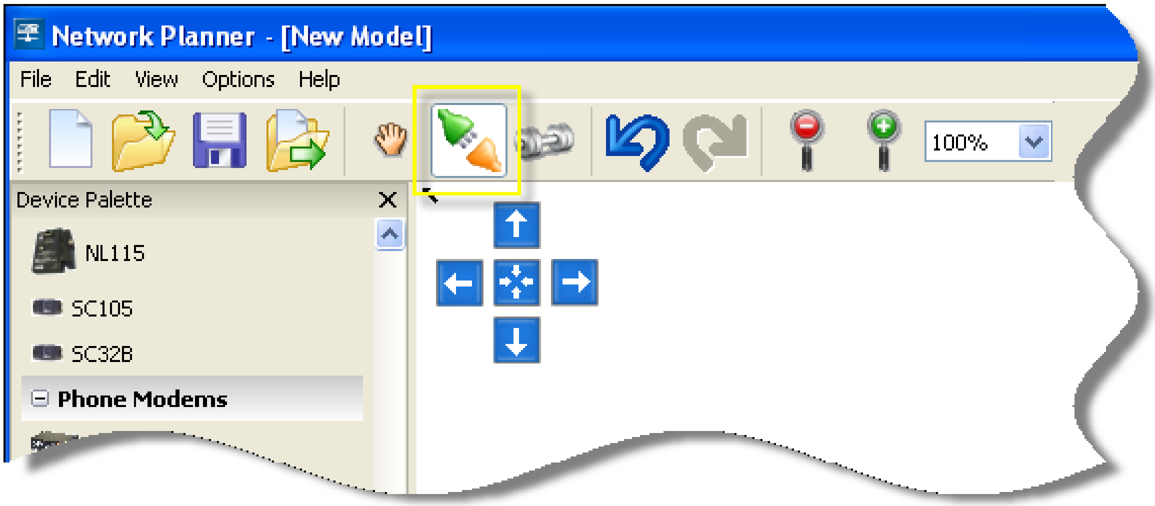
The process of adding a peripheral device to a station is similar to that of creating a station in the first place. You select the appropriate device type for the peripheral in the Device Palette and then position the mouse cursor over the top of the station icon. If a compatible interface for connecting to any of the devices in that station is available, the dashed square around the mouse cursor will disappear and small "+" icon will appear in the center of that station's icon. If you then click the left mouse button, the dialog box shown below will appear.



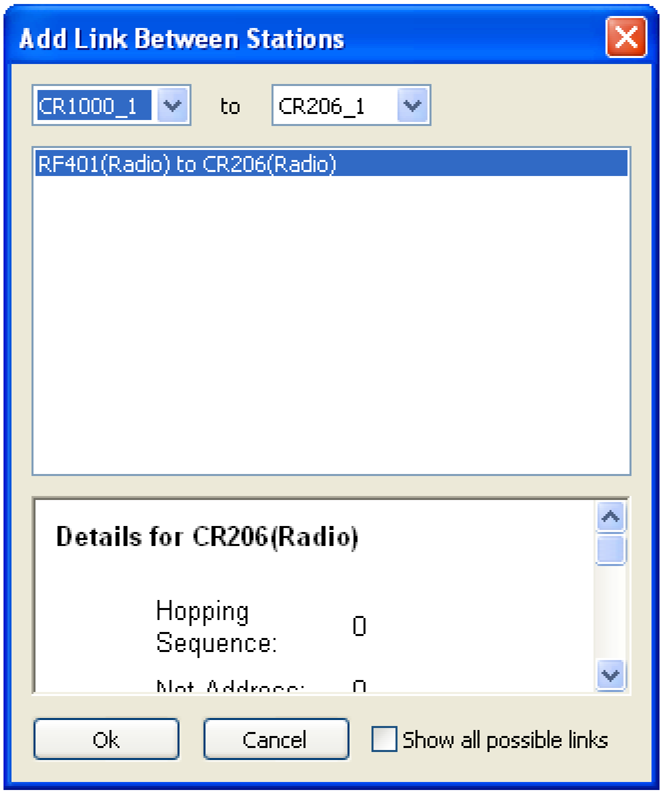
You must select the communication interface for the new peripheral from the dialog list box. These interfaces are prioritized by the Network Planner, such that the links considered the best are listed at the top. Note that the Network Planner simplifies the task of selecting a link by hiding, by default, all of the links except those that have the highest priority. You still have the option of seeing all available choices by clicking on the **Show all possible links** check box. The peripheral will not be added to the station until an appropriate link type is selected and **OK** is pressed.

5.4.5 Station Links

You can create links between stations by clicking on the Link Tool icon highlighted below.



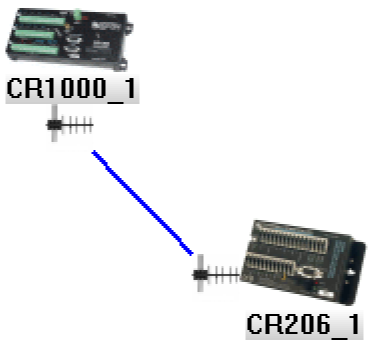
When the canvas is operating in this mode, the mouse cursor changes from a hand to a jagged line. While in this mode, you can click on a station icon to indicate the first device in the link. If that device can support a new link, a small green "+" icon will appear in the center of that station icon and, as you move the mouse, a green rubber-band line will follow the mouse cursor. At this point, when you move the mouse cursor over the icon for another station that has an interface that can be linked to the first station, another green "+" icon will appear in the middle of that station. If the mouse cursor is hovering over a station that does not have any compatible interface, a red "x" icon will appear in the center of that station icon. You can complete the link by clicking the left mouse button again, when the cursor is over another station with a small green "+" icon indicating a compatible interface. At this point, the dialog shown below will appear.



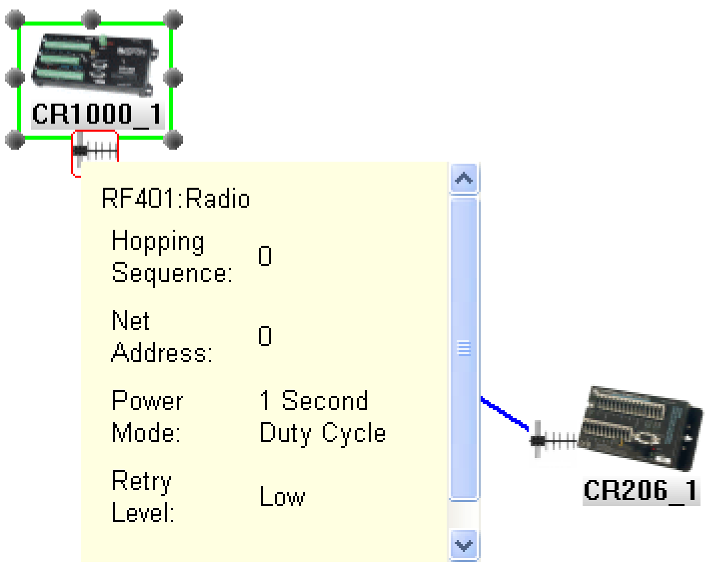
Press **OK** to add the designated link.

As with adding peripherals to stations, you must select the appropriate kind of link. The dialog presents possible combinations between all the devices in both stations. The list of link types is prioritized by the Network Planner, such that the links considered the best are listed at the top. Note that the Network Planner simplifies the task of selecting a link by hiding, by default, all of the links except those that have the highest priority. You still have the option of seeing all available choices by clicking on the **Show all possible links** check box. The link will not be added until an appropriate link type is selected and **OK** is pressed.

Station links are shown as lines between "connection points" on or around the station icons. The end points of the link lines are small icons that represent the nature of the link. An example of this is shown in the figure below of an RF401 based link.



In this instance, for radio based links, the icon is a small yagi antenna. If you hover the mouse cursor over the top of this connection point icon with the station selected, a tool-tip type window will appear that gives specific details about that link.



5.4.6 Activities

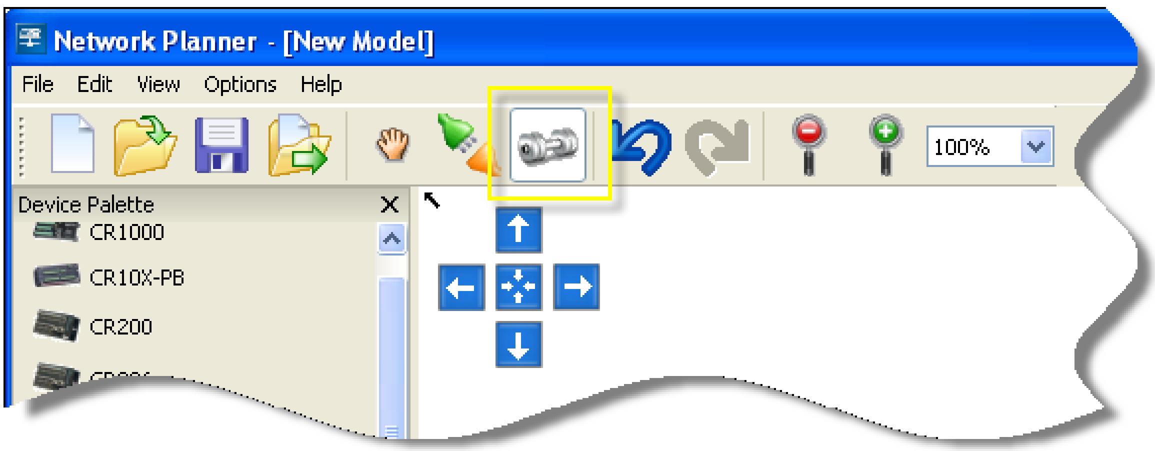
The Network Planner provides a means for you to specify that certain kinds of activities take place between station devices on a regular interval. These activities include the following:

- Scheduled polling from LoggerNet to a datalogger
- Initiation of call-back data polling from a datalogger to LoggerNet (this is a specialization of the set variable transaction)

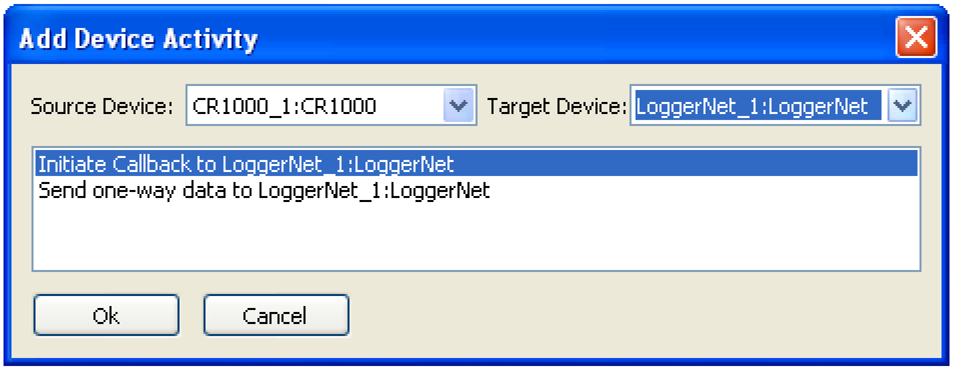
- Transmission of one-way data messages from a datalogger to either LoggerNet or to another datalogger
- Set or get variable transactions from one datalogger to another datalogger.

Aside from providing a means of documentation, activities in the Network Planner model serve an important role in specifying the intervals at which data will be expected to be transmitted over network links. The Network Planner will use these intervals to determine appropriate values for neighbor verification interval settings

Activities can be added to the model by using the Activity Tool icon highlighted below, or by choosing **Add Activity** from the context menu that results from right-clicking on a station.



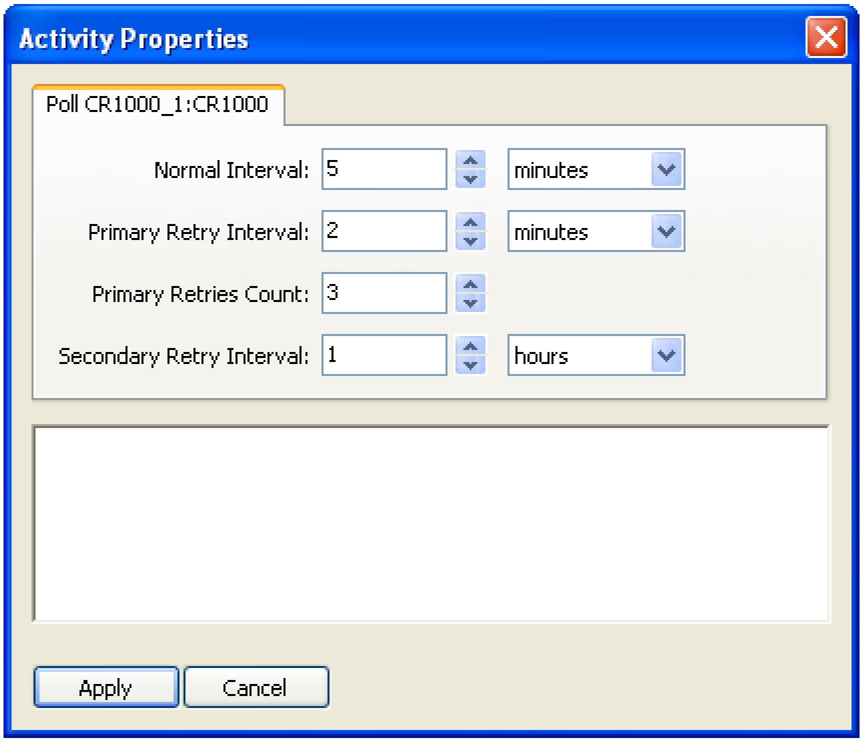
When the Add Activity mode is selected, the mouse cursor will change to indicate a linking mode and you will be expected to click on the station that contains the device that should originate the activity. When the mouse cursor hovers over a station in this mode and that station has a device that is capable of originating an activity, a small "green plus" symbol will appear in the middle of the station's icon. When the mouse cursor hovers over a station that does not contain devices capable of originating activities, a small "red cross" symbol will appear in the middle of that station's icon. You can specify the origin of the activity by clicking the left mouse button while the mouse cursor is over a station. At this point, a rubber band line will follow the mouse cursor and you will be expected to click on a second station to indicate the target of the activity. As with selecting a source station, the icon for the target station will show the "green plus" symbol if the station contains a device that is able to receive activities from the source station. Once the target station has been selected, the dialog shown below will appear. This dialog will also be shown when you add an activity from a station's context menu.



The purpose of this dialog is to allow you to specify the devices that will act as the source and target of the intended activity and also to select the type of activity that will take place. (Note that if the add activity icon was used to add the activity, the source and target devices will already be designated in the dialog box.)

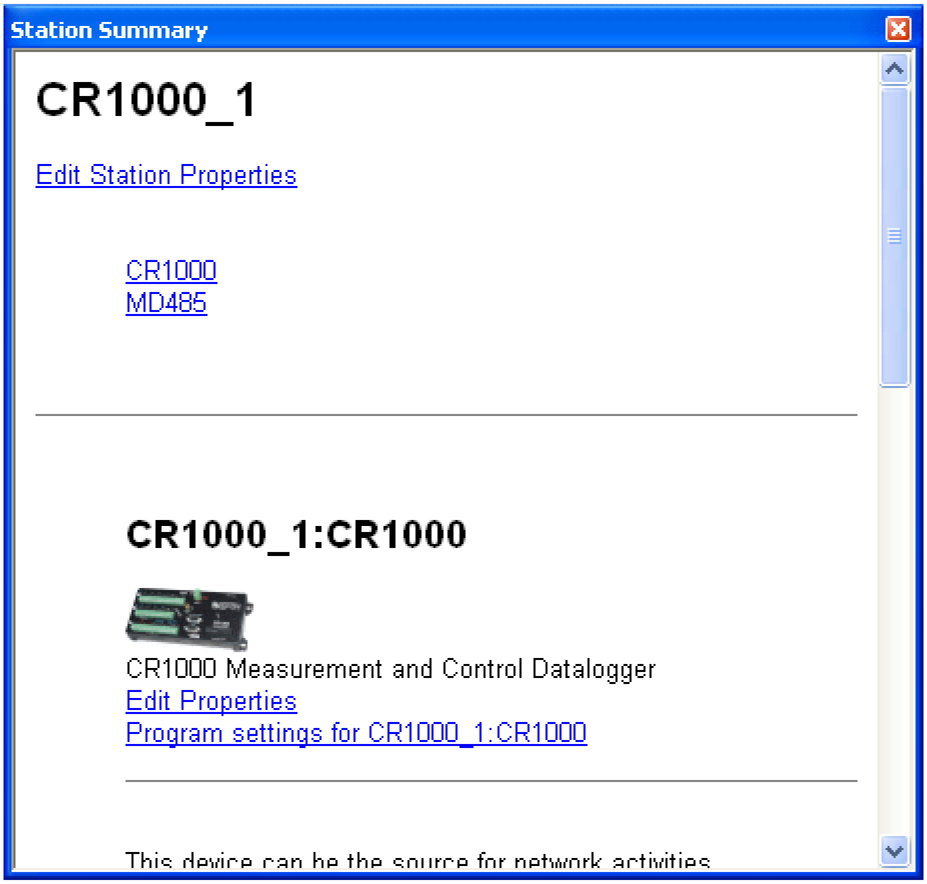
- **Source Device** - Specifies the device that will initiate the activity. This choice box will be populated with the list of all devices in the model that are capable of originating activities. These devices will be identified by their station names and device names (generally the device type) separated by a colon character.
- **Target Device** - Specifies the device that will be targeted by the activity. This choice box will be populated with the list of all devices in the model that are capable of receiving an activity. Like the source devices, devices in this list will be identified by their station names and devices names separated by a colon character.
- **Activity Types List** - This list box will be populated with the set of activities that are compatible with the two selected devices serving as source and target. If a given type of activity has already been specified between two devices, that activity will not be present in this list.

Once the source device, target device, and activity type has been chosen and **OK** has been pressed, another dialog box will appear which requires you to specify the properties for that activity. At the minimum, the properties for an activity will specify the time interval for that activity. An example of such a property sheet is shown below. After specifying the properties, press **Apply** to add the activity to the model.



Activities between station devices are not represented on the drawing canvas. To do so could create a web of lines that could easily be mistaken for station links and could also render even a moderately complex model display indecipherable. The activities associated with a station can, however, be seen in the station summary.

5.4.7 Station Summary

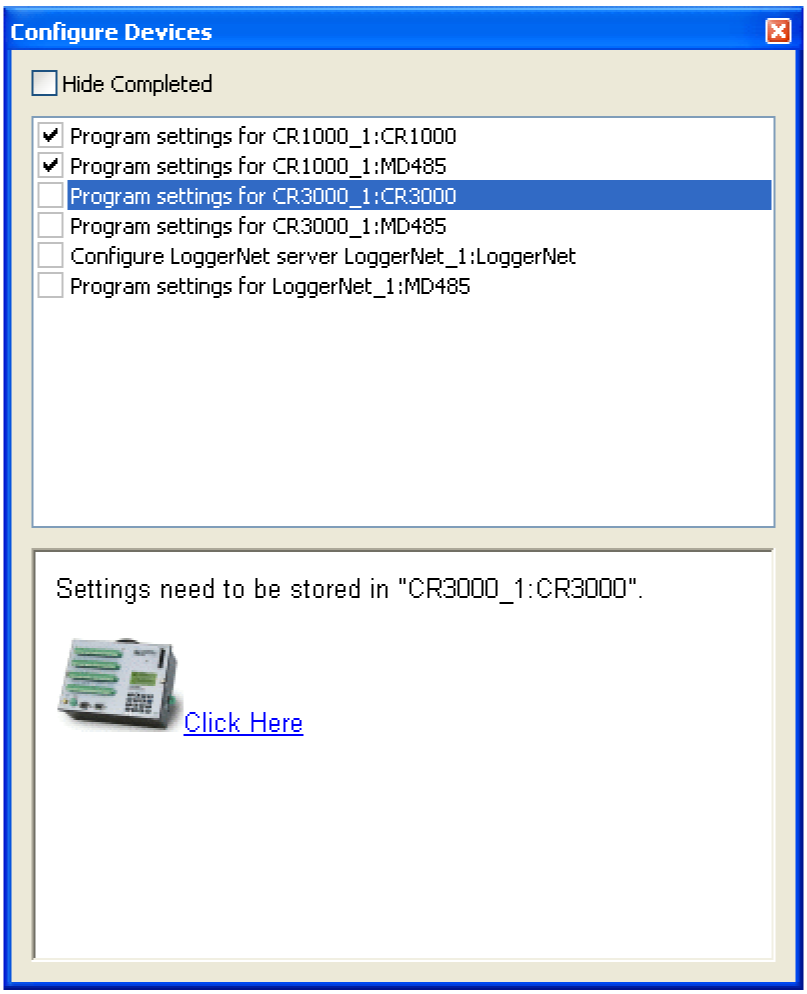


The station summary is a view that displays details about the station that is currently selected on the drawing canvas. By default, it is shown in the lower right-corner of the Network Planner window. This view provides the following features:

- An **Edit Station Properties** link that, if clicked, will present you with a dialog that contains property sheets for all of the devices and/or links in the station. (This link will not appear if there is only one device in the station and there are no links with properties that can be edited.)
- A list of links to the parts of the report that describe each of the devices in the station. (The list of links will not appear if there is only one device in the station.)
- A description of each device in the station . This description includes the following:
  - The same image that is used as an icon for the device.
  - The "official" description of the device.

- A link to edit the properties for the device.
- A link to delete the device (does not appear for the station root device)
- An optional link to perform the Configure Devices list item for the device.
- A description of any activities for which this device is a source. This includes a link to delete those activities.
- PakBus information for the device including its PakBus address(es), the device's role (router or leaf node), and the routes used to reach other PakBus devices.
- List of links for which the device is an end-point.
- Any device specific details that may be applicable. For example, on the COM220 and the RF310, a table will be shown that specifies the jumper settings for the device.

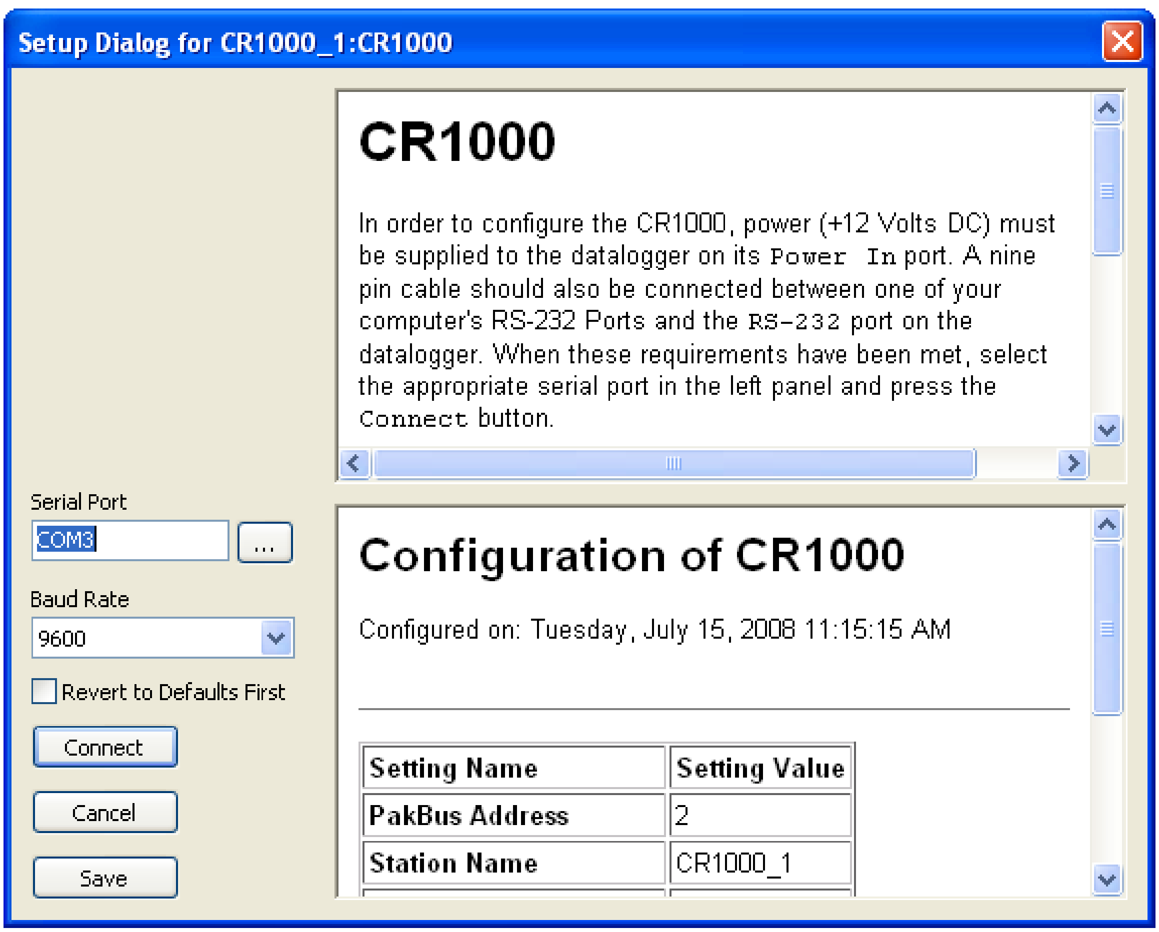
5.4.8 Configure Devices



The Configure Devices panel lists tasks that need to be completed before the network can be deployed. These tasks include configuring any LoggerNet servers and writing settings to devices . The Configure Devices panel is divided into two sections. The list box at the top lists all of the Configure Devices items and provides check-boxes that allow these items to be checked off to indicate when an item has been completed. The bottom portion of the panel displays a more detailed description of the selected item. If the item has been completed, the description will include the date and time when it was completed. Unless the selected item must be performed manually (setting a dip-switch on a COM220, for instance), the item description will also contain a link that can be followed to initiate the action associated with that item. When the selected station on the drawing canvas changes, the selected item in the Configure Devices list box will also change to the item, if any, associated with the root device of that station.

5.4.8.1 Configuring Using the Device Configuration Protocol

The majority of device types that are supported by the Network Planner can be set up using the same protocol as is used by the Device Configuration Utility. The figure below shows a screen shot of the dialog that supports this action.

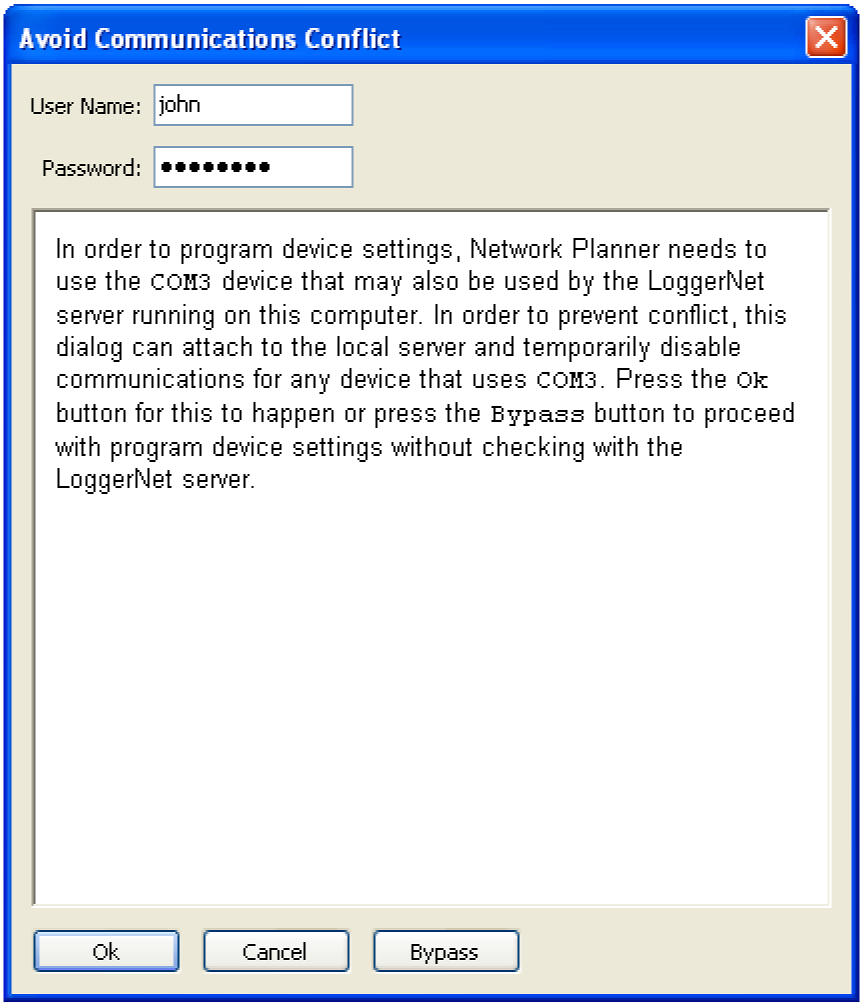


This dialog has the following controls:

- Serial Port - Specifies the PC serial port that should be used for communication with the device.
- Baud rate - Specifies the baud rate at which communication should take place with the device.
- Revert to Defaults - First Specifies whether all of the device settings should be automatically set to their factory default values before writing the settings generated by the Network Planner. Typically, the Network Planner does not generate every possible setting for a device but, rather, generates the minimal set needed to express the model parameters.
- Connect - When clicked, this button will disable the dialog controls and initiate communication with the device in order to transmit the settings. In order to accomplish this, the Network Planner uses its own private LoggerNet server.
- Save - This button allows you to save the generated settings for the device into an XML file that can be read by the Device Configuration Utility as well as the PakBus Graph LoggerNet client . This presents an alternate means of loading settings into the device.
- Connect Instructions - The HTML control in the upper right hand corner of the dialog shows the instructions to connect to the device. These instructions are obtained from the Device Configuration library and will thus always be the same as is shown in the Device Configuration Utility.
- Generated Settings Summary - The HTML control in the lower right hand corner of the dialog shows a summary of the settings that have been generated for the device. The format of this summary is exactly the same as that can be seen when connecting to the device from within the Device Configuration Utility or from within PakBus Graph .

Avoiding Conflicts with the LoggerNet Server

If the Network Planner was started by the LoggerNet tool bar and a local LoggerNet server is reported to be running, the dialog box shown below will appear.



This dialog allows you to specify a user name and password for an administrative account on the local server and, if you click on the OK button, will perform the following actions:

- The dialog will connect to the local server and scan its network map for any serial port devices that use the same serial port name as that specified in the device configuration dialog.
- For any local LoggerNet device that matches the criteria listed above, the dialog will attempt to override the communications enabled setting for that device to a value of false.
- The dialog will wait for the Line State statistics for all matching devices to report an off-line state.
- The dialog will close once all of the Line State statistics report an off-line state. At this point, communications will be initiated with the device. The settings override(s) explained above will be released once configuration of the device has been completed.

- If you click **Cancel** at any time while the dialog is waiting for the local LoggerNet devices, the dialog will be closed and all overrides canceled.

If you click **Bypass** rather than **OK**, no communication with the local LoggerNet server will take place. If **Bypass** is used and LoggerNet is actively using the selected serial port, the Network Planner 's attempt to use that serial port will fail. Thus, **Bypass** should only be used if you are certain that conflicts do not exist.

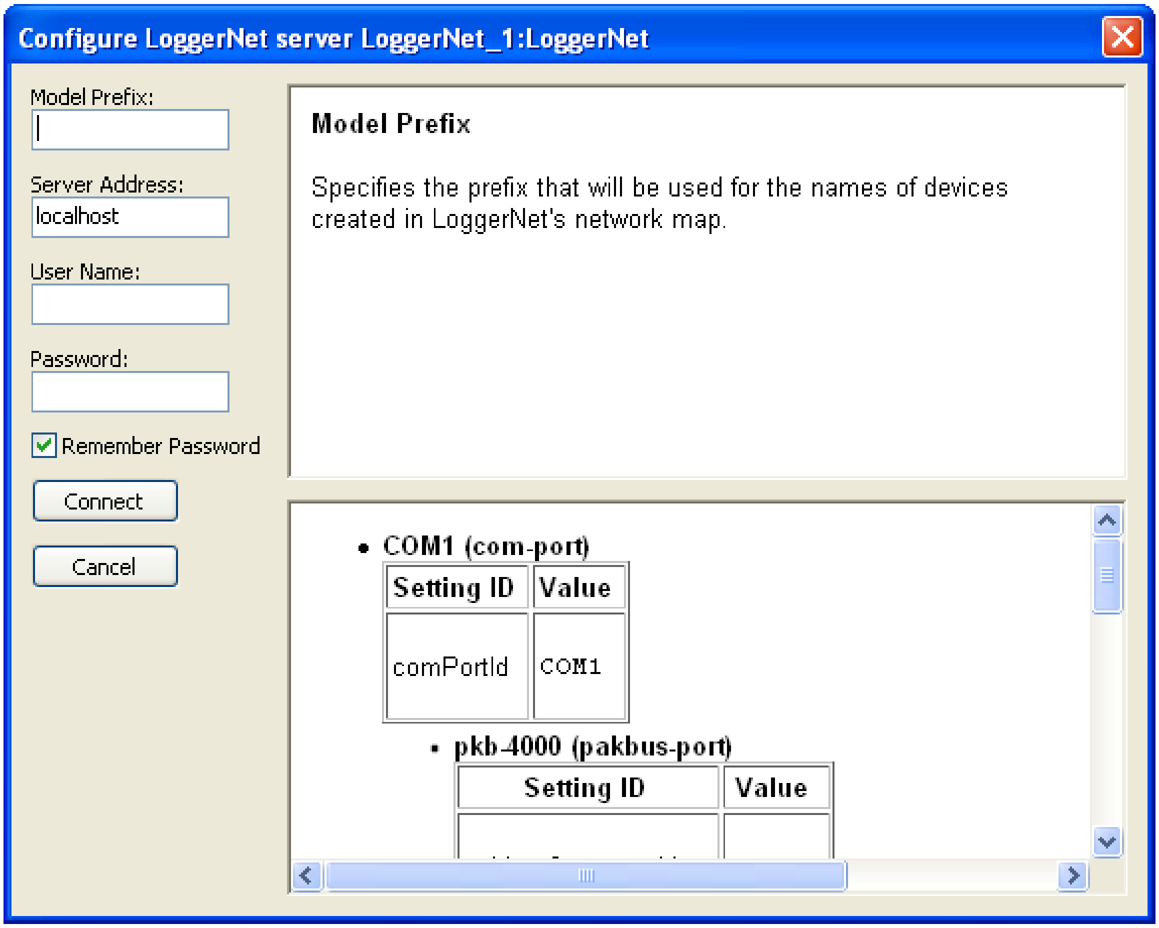
This dialog will not be shown if the Network Planner is launched outside of the LoggerNet tool bar or if there is no local server reported to be running.

Settings Generated

The settings for the device will be generated from information contained in the Network Planner model including device links and activities between devices. The Network Planner will typically not generate the value for every conceivable setting for the device but will rather only generate the value for settings for which it has relevant information. Since each device type has its own unique set of settings and behaviors, the settings generated will be specific to each device type.

5.4.8.2 Configuring a LoggerNet Server

The LoggerNet device type is configured using the LoggerNet server protocol rather than using the device configuration protocol. The dialog that is shown appears when you start the action of configuring a LoggerNet Server.



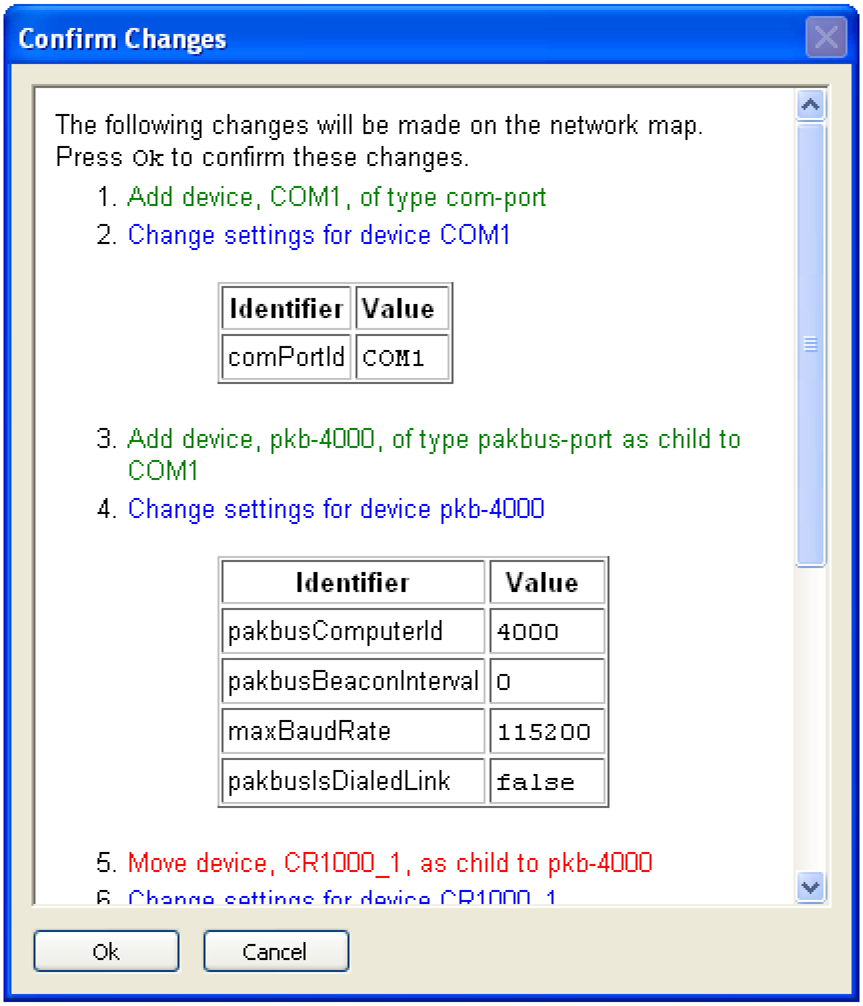
This dialog has the following controls:

- **Model Prefix** - This field allows you to enter a string that will be placed at the beginning of the name for every device in LoggerNet's network map that the Network Planner generates. If specified, this value will appear in the names followed by an underscore character. This feature allows you to use several different Network Planner models with the same LoggerNet server by keeping the generated devices separate.
- **Server Address** - This field specifies the IP address or domain name of the computer that is running the LoggerNet server. In order to configure LoggerNet on a remote computer, that instance of LoggerNet must allow remote connections and must also be reachable from the computer hosting the Network Planner.
- **User Name & Password** - These fields allow you to specify the account that should be used when connecting to the LoggerNet server. The account used should have at least Network Manager privileges. If security is not configured on the server, these values will be ignored.
- **Remember Password** - This check box can be checked (the default) if the user name and password specified should be saved as part of the information for the LoggerNet device in

the model.

- **Connect** - When this button is clicked, the Network Planner will initiate the connection to the LoggerNet server.
- **Cancel** - If this button is clicked, the configuration dialog will be dismissed and the configuration attempt canceled.
- **Help Window** - The HTML window in the upper right corner of the dialog shows context sensitive help about the control that has the current keyboard focus.
- **Generated Devices and Settings** - The HTML window in the lower right corner of the dialog shows a summary of the devices that will be created in LoggerNet's network map as well as the settings associated with those devices. This summary is presented in an indented list form where the level of indentation depends upon device links. Since the dialog has not yet communicated with the server, this summary shows only the expected structure and does not reflect any devices in the actual network map.

If you click **Connect** in the Configure LoggerNet dialog, the Network Planner will attempt to connect to the specified server address and will log in using the specified user name and password. Once attached, the dialog will attempt to reconcile the structure and settings that it has generated with the structure and settings currently in LoggerNet's network map. The result of this reconciliation is a set of changes that will be made to the current LoggerNet network map. An example of this set of changes is shown below.



The colors of items in this dialog indicate the impact they may have on the operation of devices that are already in the LoggerNet network map.

These are coded as follows:

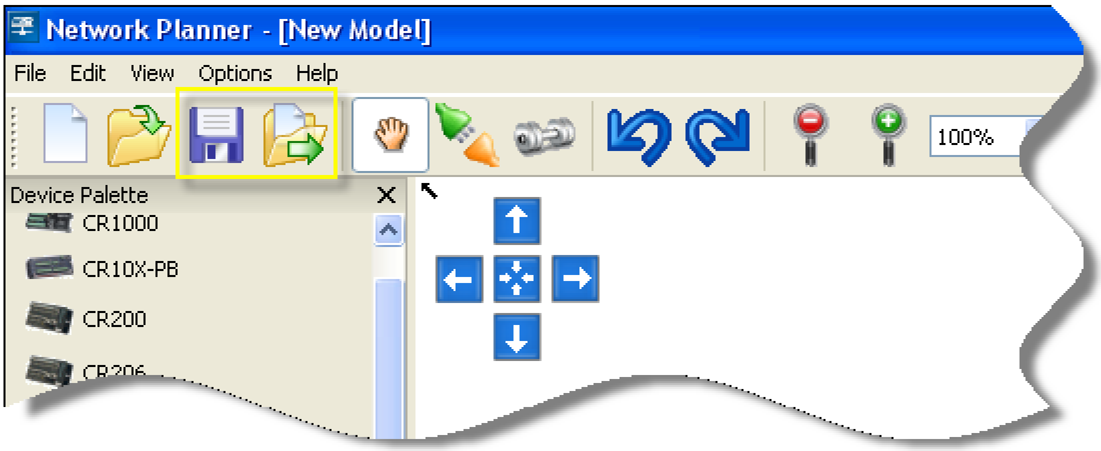
- **Green** - The change is merely additive (adding new devices, for instance) and is unlikely to have any noticeable impact on the workings of existing devices.
- **Blue** - The change involves making changes to settings of existing devices.
- **Red** - The change will alter the structure of the network map and will relocate existing devices.

If it becomes necessary to delete an existing device in order to reconcile the LoggerNet network map with the generated configuration, an error will be reported. In order to avoid loss of data, the Network Planner will not delete any existing devices from the network map.

5.4.9 Saving Your Work

The Network Planner model can be saved to a file by one of several interactions:

- By clicking on the Save or Save As tool bar button highlighted below.



- By selecting the **Save** or **Save As** item from the File menu.
- By using the Ctrl-S or Ctrl-Shift-S keyboard shortcut.

The Network Planner will store all of the information about the model in an NWP file. By default these files will be written to the C:\CampbellSci\ NetworkPlanner directory but you can select any other directory. Along with model information, the model file will also store the background image if there was any associated with the model. Screen layout and zoom options will not be stored in the model file.

The structure of these files is such that they can be easily transferred to another computer. If a Network Planner model is created on one computer and then used on another computer, some machine specific properties, such as IP addresses and serial port identifiers may have to be adjusted to account for differences between the two computers.

5.4.10 Arranging Screen Components

The Network Planner user interface has been designed so that you can exercise a great deal of control over the placement and sizes of the various user interface components. User interface components including the toolbar, the Device Palette, the Station Summary, and the Configure Devices panel can be detached from the application frame, re-anchored to the frame in a different location, resized, or hidden altogether.

Once a component is detached from the application frame, it will be shown in a separate window that can be moved to any location on the your desktop. This feature is particularly valuable where more than one monitor is available since it allows the drawing canvas to occupy one monitor and the other components to be arranged on the other thus maximizing the space available for viewing the network. It should be noted, however, that a user interface component, once detached from the application frame, will "float" over the frame when the windows overlap.

A detached component can be re-attached, or "docked", to the frame by dragging that component's title bar over the top of the frame borders or the borders of other docked components. When the component is dragged over one of these boundaries, a blue rectangle will be drawn showing the boundaries of that component if the mouse button is released in that position. When the component is docked thus, the size of the drawing canvas may be reduced in order to accommodate the component. A docked component can be resized by dragging on either its right or bottom borders. Doing this will also affect the size of the drawing canvas.

The Network Planner will "remember" the layout of components between sessions by storing the layout information in the application's INI file. This layout will remain constant as different models are loaded.

The View menu can be used to show or hide various user interface components including the toolbar. It also includes an entry, **Restore Default View**, that can be used to restore the application layout to the "standard" layout.

5.5 Device Configuration Utility

The Device Configuration is used to set up PakBus information in PakBus-capable dataloggers and to configure peripheral communication devices. Refer to Device Configuration Utility (p. 335) for information.

6. Real-Time Tools

LoggerNet’s real-time tools are used to manage your stations in the datalogger network. Tools are provided for sending new programs, setting the clock, toggling ports and flags, collecting data, and displaying data numerically and graphically.

6.1 The Connect Screen

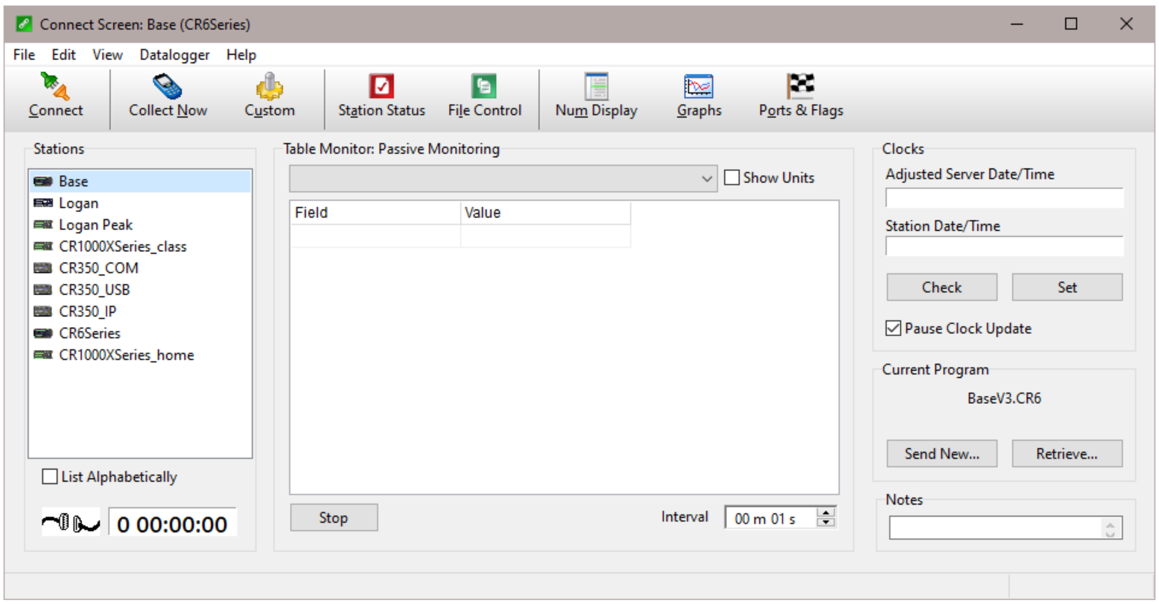
The Connect Screen provides a real-time connection to a datalogger in the datalogger network. Tools are provided for transferring programs to the datalogger, manually setting the datalogger's clock, viewing and collecting data, and communicating with the datalogger in terminal mode.

These tasks can be accomplished while actively connected to a station in the datalogger network, or, if you are not actively connected, LoggerNet will make the connection, perform the desired action, and then end communication with the station.

> **NOTE:** If you have LoggerNet Admin or LoggerNet Remote installed, you have the ability to set up security so that access to certain functions in LoggerNet is limited. If security is enabled, you may be unable to access certain functions in the Connect Screen.

6.1.1 Connecting to the Datalogger - or Not

The Connect Screen works with and displays data from only one datalogger at a time. (LoggerNet Admin and LoggerNet Remote allow you to launch multiple Connect Screens. Therefore, you can connect to more than one datalogger at a time.) The name of the selected datalogger and the datalogger type appear in the title bar at the top of the window. For ease of use in large networks of many dataloggers you can list the dataloggers in alphabetical order by selecting the **List Alphabetically** check box. All the station names will be listed alphabetically to make it easier to find a specific station. A red exclamation point next to a station name indicates that the station communication state is critical (that is, communication with this device has failed).



As noted above, you can work with a datalogger station while actively connected to it or when you are in a disconnected state. Even when not actively connected, you can choose to collect data, check or set the clock, etc. When a button is pushed, LoggerNet will attempt to contact the datalogger, performed the desired action, and then terminate communication.

If you want to perform only one task, such as collecting new data, it may be more efficient not to actively connect to the datalogger. LoggerNet will merely contact the datalogger, collect the data, and end communication. If you were actively connected, LoggerNet would also update the clock displays during this process, which, when collecting large amounts of data over a slow communication link, could affect the speed of data collection. If you want to perform multiple tasks — e.g., send a new program and view measurements on a Numeric Display to ensure the program is running correctly — then it is usually more efficient to establish an active connection, perform the tasks, and then terminate the connection yourself. Otherwise, LoggerNet must establish communication with the datalogger twice. Over remote communication links, this connect/disconnect/connect sequence will increase the time to complete the tasks.

When you select the **Connect** button the animated graphic will indicate an active connection state. It will show that LoggerNet is trying to establish the connection; the two connectors join together when the connection is made. You can also connect to the datalogger by double clicking the datalogger name or selecting **Connect** from the File menu.

> **NOTE:** When you connect to a station, LoggerNet checks for Status Table errors. If the station has Status Table errors (skipped scans, skipped records, and so forth), a yellow exclamation point will be added to the **Station Status** button. Once you click on the **Station Status** button, this indicator will be removed.

Once the datalogger connection is established, an elapsed time for the connection will be shown on the bottom left of the window. This counter will continue as long as the datalogger connection is maintained. The user should be aware of the how long the Connect Screen is connected to the datalogger. A manually initiated connection to a datalogger takes priority over other communication in the network map. Other devices that share any part of the communication path will not be contacted, even for scheduled data collection.

> **NOTE:** Once a connection is made to a datalogger with the Connect Screen, this connection takes precedence and will prevent communication with other devices sharing the same serial port or base modem.

To disconnect from the datalogger, click the button that now reads **Disconnect**. To work with another datalogger you must disconnect from the first one (unless you have installed LoggerNet Admin or LoggerNet Remote). Double clicking another datalogger will disconnect from the first datalogger and connect to the new one without prompting.

If LoggerNet fails to make a connection to the datalogger, it will time out and display an error message that it could not connect. It will immediately attempt the connection again and will continue trying until the user clicks Cancel.

6.1.2 Data Collection

The **Collect Now** and **Custom** buttons of the Connect Screen allows you initiate a manual data collection from the datalogger. LoggerNet keeps track of two separate data collection pointers for each datalogger: (1) the pointer for scheduled data collection and manual data collection from the Connect Screen's **Collect Now** button, and (2) the pointer for manual data collection from the Custom Collection window.

The data for each of these two pointers is, by default, stored in two separate data files on the computer. The default directory for scheduled data collection/manual data collection is C:\Campbellsci\LoggerNet. The default directory for data collection via the Custom Collection window is C:\Campbellsci\LoggerNet\Data. Because different pointers are kept in LoggerNet for each of these collection options, if you select Collect Now from the main Connect Screen, and your Setup option is set to collect only new data, and then you do a Custom Collection and also choose to collect only new data, the new data collected using the Custom Collection window is the new data since the last time you collected using this window. Similarly, new data collected using **Collect Now** from the main Connect Screen is the new data since the last time you chose Collect Now, or since the last scheduled data collection.

> **NOTE:** When collecting data from a remote LoggerNet server using LoggerNet Remote or LoggerNet Admin, the file will be saved on the computer running the server if a Collect Now or scheduled data collection is performed, and it will be stored on the machine from which the collection is initiated when a custom data collection is performed.

6.1.2.1 Collect Now

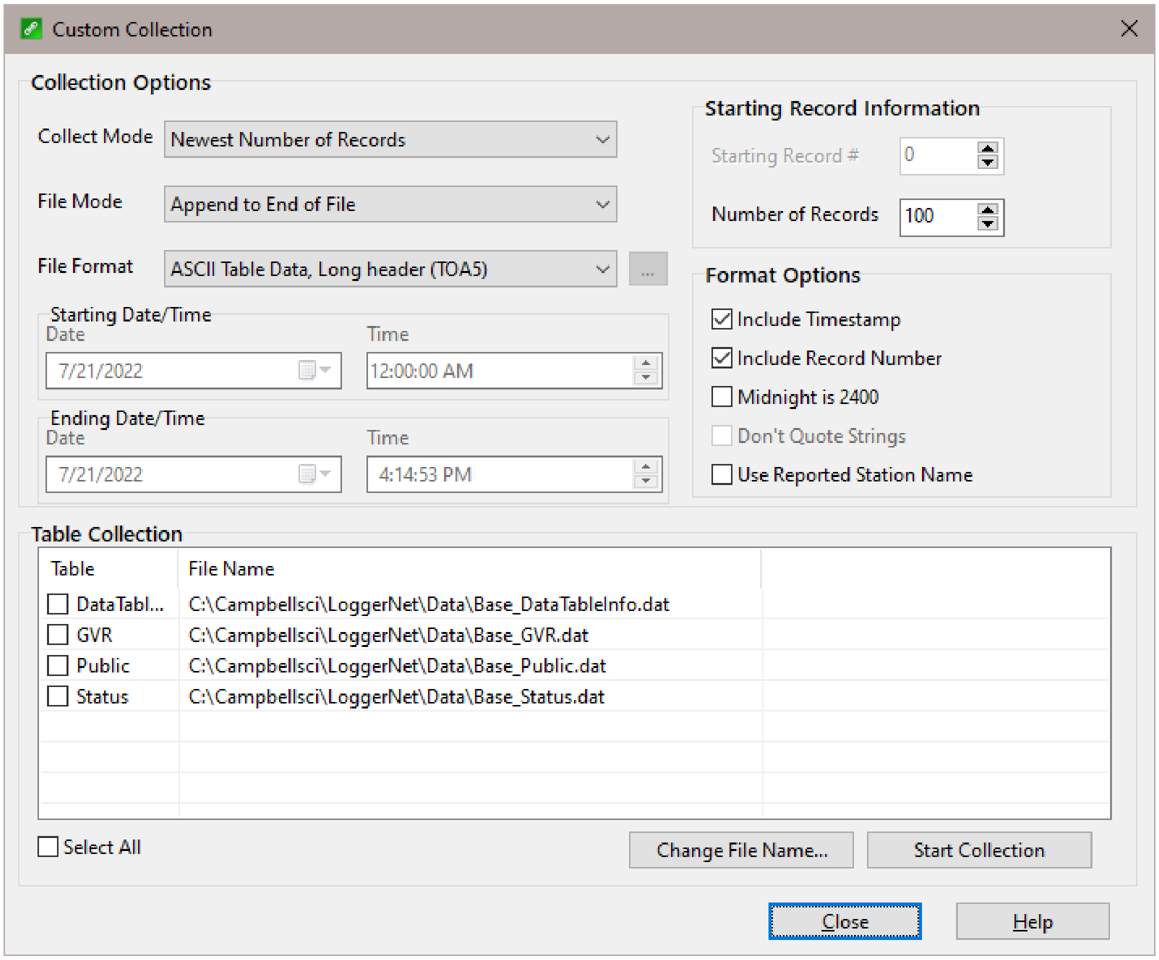
The Collect Now function is the equivalent of doing a scheduled data collection for the datalogger without waiting for the scheduled time. Clicking the **Collect Now** button will initiate a call to the datalogger and any available data will be collected and stored as specified on the Setup Screen. This function is often used to manually update data collection to see the latest data in a Numeric Display or a Graph; or before data files are copied for processing.

Once you have started data collection with **Collect Now**, you can stop it by clicking the **Cancel** button on the animated screen. This might be necessary if you started a data collection that is bringing in more data than you really wanted, especially over a slow communications link.

6.1.2.2 Custom Collection

Clicking the **Custom** button brings up the following dialog box. After making your selections, press **Start Collection** to collect the data.

> **NOTE:** While retrieving data from the datalogger using Custom Collection, scheduled data collection will be suspended. The default data file names for custom collection are separate from the files for scheduled collection data and by default are placed in a Data directory under the LoggerNet directory.



Collect Mode

Newest Number of Records - Retrieves a specific number of records from the selected tables by backing up the number of records entered in the **Number of Records** field and retrieving all data forward.

Specific Records - Allows you to specify a beginning record number and the number of records to collect after that record. The range of records to retrieve is specified by completing the **Starting Record #** and **Number of Records** fields.

Data Since Last Collection - Retrieves all uncollected records from the selected tables.

All the Data - Retrieves all records from the selected tables.

Data from Selected Date and Time - Allows you to specify a time frame for data collection. When this option is selected, the **Starting Date/Time** and **Ending Date/Time** fields will be enabled. (CR5000, CR9000, CR9000X, and CR200-series dataloggers do not support this collection option.)

File Mode

Append to End of File adds new data to the end of the existing data file.

Overwrite Existing File replaces the existing file with a newly created file.

Create New File leaves any existing files intact and creates a new file with a unique file name based on the time of data collection. For example, CR1000_Table1_2008_09_04_12_14_13.dat.

> **NOTE:** When appending to a data file, LoggerNet will check to make sure that the existing data file header matches the new data that is to be appended to the file. If the header does not match, LoggerNet may back up the existing data file and then create a new data file.

File Format

> **NOTE:** For most file formats, the browse button to the right of the field is enabled. Pressing this button opens a window which allows you to customize the data string for the output file.

ASCII Table Data, No Header - Data is stored in a comma separated format. No header information is included in the file.

TOACI1 - Data is stored in a comma separated format. Header information for each of the columns is included.

TOA5 - Data is stored in a comma separated format. Header information for each of the columns is included, along with field names and units of measure if they are available.

TOB1 (binary) - 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. When this option is selected, the browse button to the right of the field is enabled. Pressing this button opens a window in which you customize the data string for the output file. This option can be used to produce output files from table data dataloggers that are similar to those created by mixed array dataloggers.

CSIXML - Data is stored in XML format with Campbell Scientific defined elements and attributes. For additional information, refer to the LoggerNet manual.

Starting Date/Time / Ending Date/Time

The fields in this group box are enabled only when the Collect Mode is **Data from Selected Date and Time**. To complete the **Starting or Ending Date** field, type in a date directly or click the arrow to the right of the field to display a calendar from which to choose a date. To complete the **Starting or Ending Time** field, type in the time directly or use the arrows to the right of the field to increase or decrease the highlighted time value.

Starting Record Information

The fields in this group box are enabled only when the Collect Mode is **Newest Number of Records** or **Specific Records**.

The **Starting Record #** field is used to enter the first record to be collected when the **Specific Records** option is chosen.

The **Number of Records** field allows you to specify a fixed number of records to be collected from a table. When the **Newest Number of Records** option is chosen, LoggerNet counts back the number of records specified from the most recent record and collects all data forward. When the **Specific Records** option is chosen, data collection begins with the **Starting Record #** and ends when the specified number of records has been collected.

Format Options

Select the **Include Timestamp** check box to have timestamps included in your data. If the check box is not selected, timestamps will not be included.

Select the **Include Record Number** check box to have record numbers included in your data. If the check box is not selected, record numbers will not be included.

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.

When the **Don't Quote Strings** check box is selected, strings in the data will not be surrounded by quotation marks. If the check box is not selected, strings will be surrounded by quotation marks. (Note this option is only available for the ASCII Table Data, No Header Output Format.)

Enabling the **Use Reported Station Name** check box will cause the station name from the Status Table to be used in the header of the data files. If this check box is not enabled, the network map station name will be used. (Note that this check box affects only the header of the data files. It has no effect on the filenames.)

Table Collection

To mark the tables that will be collected, enable the **Select All** check box or choose the check boxes for tables individually from the list. Default names are provided in the **File Name** field. A file name can be changed by placing your cursor within the desired field and pressing the **Change File Name** button.

6.1.3 Ports and Flags

The Ports and Flags window shows the current state of the ports and flags for 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, LoggerNet 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.

6.1.4 Datalogger Clock

If a connection to a datalogger is established, the datalogger's clock is checked continuously and displayed along with the computer's clock as updates are received. The clock update can be paused by selecting the check box next to **Pause Clock Update**. In some situations it is desirable to pause the clock update to minimize data traffic over the communications link.

You can set the clock by clicking the **Set** button. LoggerNet attempts to set the datalogger clock as closely as possible to the computer clock. A slight difference in the clocks might exist after the clock is set because of the communications time delay. Over some communication links it is impossible to match the computer clock exactly. LoggerNet uses advanced compensation to get the best possible synchronization between the computer and the datalogger clocks.

Double-clicking in the **Station Date/Time** field makes it an editable field and allows you to manually edit the station clock. The date/time should be entered in the same form as it is displayed. When you press Enter on your keyboard, the datalogger clock will be updated.

If you are not connected to the datalogger, or if you are connected but the clock update is paused, you can press the **Check** button and LoggerNet will check both the datalogger clock and the computer clock and display the results on the screen.

An automatic scheduled clock check can be set up in the Setup Screen. Note that setting the clock may affect the time stamps assigned to your data. see Setting the Clock (p. 43) for more information.

6.1.5 Program Management

The Current Program section on the Connect Screen is used to send programs to or retrieve programs from dataloggers in the network. Edlog is used to create programs for the CR7, 21X, and the CR10(X), CR510, and CR23X-series dataloggers (CRXX, CRXX-TD, and CRXX-PB). The CRBasic Editor is used to create programs 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. Short Cut can be used to create programs for any datalogger type. After a program is created in one of the program editors, the Connect Screen is used to transfer it to the datalogger. (Program editors are discussed in Creating and Editing Datalogger Programs (p. 195).

NOTE: Programs for the CR7, 21X, and the CR10(X), CR510, and CR23X-series dataloggers must be compiled in the editor to create the *.dld file that is downloaded to the datalogger. The CR200-series datalogger also requires a precompiled file (*.bin), which can be done in the editor or when the program is sent using LoggerNet. CR6-series and CRX000 dataloggers compile their program on-board.

6.1.5.1 Sending a Datalogger Program

To transfer a program, press **Send New**. A standard file select dialog box will come up so you can choose the file to send. The **Files of Type** selector at the bottom of the dialog can be used to filter the files that are displayed. This filter is set to the appropriate file type for each datalogger automatically (*.dld for CR7, 21X, CR10(X), CR510, and CR23X dataloggers, and *.CR# for the CR1000X-series, CR6-series, CR300-series, CR1000, CR3000, CR800-series, CR200-series, GRANITE 6, GRANITE 9, GRANITE 10, CR5000, CR9000, and CR9000X dataloggers).

After selecting a datalogger program file a warning will appear to remind you that data may be lost when the new program is sent. (For mixed-array dataloggers data is not lost if the memory configuration does not change; sending a new program to table-based dataloggers always clears all data memory.) If there is any data in the datalogger that has not been collected, click **Cancel** to stop the program send, and collect the needed data.

If **OK** is selected at the warning, the progress bar will come up with the program transfer progress. Once the program has been sent, the text changes to Compiling Program. When the datalogger finishes compiling the program the progress box will close and a Compile Results box will open. For CRBasic-programmed dataloggers (excluding the CR200 series), this box will have a **Details** button that can be pressed to bring up information about files and tables stored in the datalogger.

NOTE: If a program downloaded to the datalogger sets or changes the active security code, make sure to change the security setting for the datalogger in Setup Screen. Otherwise, access to the datalogger may be limited or completely denied.

6.1.5.2 CR200-Series Programs

Programs for the CR200-series dataloggers must be precompiled before being sent to the datalogger. The compiled file is a binary image file with a *.bin extension. Unlike the other dataloggers, CR200-series dataloggers do not have an on-board compiler. Consequently, the *.bin file must be generated with a version of the precompiler that matches the operating system in the datalogger, either during program creation in the CRBasic Editor, or when the file is downloaded to the datalogger by LoggerNet. If a *.bin file is downloaded to a CR200, and the version of that binary file does not match the datalogger's OS, the download will fail and an error will be returned.

The LoggerNet installation includes all of the compilers for the CR200 that were available at the date of release. When sending a program, if you choose to send the *.CR2 file, LoggerNet will first check the OS version of the datalogger and then attempt to compile the *.CR2 file with the matching compiler. If you choose to send the *.bin file, LoggerNet will not check the CR200's OS or precompile the file, it will just send the *.bin file. In this instance, it is up to you to ensure that the *.bin file was created with the correct precompiler.

6.1.5.3 Retrieving Datalogger Programs

The program running in the datalogger can be retrieved and saved to a file by pressing **Retrieve**. You will be prompted for a name and directory in which to store the retrieved file. Files for the CRBasic dataloggers (e.g., CR1) can be opened directly in the CRBasic Editor. A retrieved *.dld file can be imported into Edlog for editing by using Edlog's Document DLD feature (Creating and Editing Datalogger Programs (p. 195)). The binary image files (*.bin) for the CR200-series dataloggers cannot be retrieved.

This feature is useful if the original file has been corrupted, lost, or erased. Note that programs may not be reliably retrieved over noisy or slow communications links.

6.1.6 Program Association

A table-based datalogger maintains final storage table information internally — this is referred to as the datalogger's table definitions. The LoggerNet server uses this information for the Add dialog box when you select values to view on a Numeric or Graphical display.

Mixed-array dataloggers do not store final storage information internally. However, this information is contained in the *.dld file as commented text. When you download a program to a datalogger, LoggerNet uses the input location and final storage information from this file. If you communicate with a datalogger that already has a program in it, you can use the **Datalogger** > **Associate Program** option to select a *.dld file from which LoggerNet should get this information. A program file can also be associated with a datalogger from the Setup Screen's **Program** tab.

Programs created with Edlog version 2.0 and greater include both the input location information and the final storage information in the *.dld file. Previous versions of Edlog stored only the input location information in the *.dld. If final storage information is not available for viewing in LoggerNet after associating the file, you may need to recompile the program file with a version of Edlog that stores this information in the *.dld file.

> **NOTE:** If you are using Edlog Version 2.0 or greater and labels are still not available for use, check Edlog's **Options** > **DLD** File Labels menu item and ensure that labels are being stored in the file when the program is compiled.

For CR1000X-series, CR6-series, CR300-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, and CR800-series dataloggers, you can use the Associate Program option (either from the Connect Screen's **Datalogger** menu or from the Setup Screen's **Program** tab) to associate a TDF file with a datalogger. TDF stands for Table Definitions File. When a program is compiled for a CR1000X-series, CR6-series, CR300-series, GRANITE 6, GRANITE 9, GRANITE 10, CR1000, CR3000, or CR800-series datalogger a program_name.TDF file is created along with the original program file. This file contains the table definitions for that program. Associating the TDF file with a datalogger can be useful if communication is taking place over a slow or unreliable communications link where the attempt to receive table definitions back from the datalogger fails.

6.1.7 Data Displays

The Connect Screen allows you to view real-time data using the Table Monitor, Numerics, or Graphs.

6.1.7.1 Real Time Monitoring vs. Passive Monitoring

The data values displayed on the Connect Screen's data displays (Table Monitor, Numeric Displays, and Graphs) are updated differently based on whether or not you are actively connected to your datalogger. The differences are described below:

Real Time Monitoring (Actively connected to the datalogger)

Table Data Dataloggers

Public variables and final storage data are updated based on the Update Interval. Note that data can be updated no faster than the data values are being generated by the datalogger.

Mixed Array Dataloggers

Input locations are updated based on the Update Interval. Note that data can be updated no faster than the data values are being generated by the datalogger.

Final storage data is retrieved only when data collection from the datalogger occurs (initiated manually from the Connect Screen or based on a schedule). Therefore, the final storage information on the data displays will be updated only as often as data collection is performed.

Passive Monitoring (Not connected to the datalogger)

Table Data Dataloggers

Public variables and final storage data are retrieved only when data collection from the datalogger occurs (initiated manually from the Connect Screen or based on a schedule). Therefore, information on the data displays will be updated only as often as data collection is performed for these dataloggers. Public variables will be updated only if the Public table is included for scheduled collection on the Setup Screen.

Mixed Array Dataloggers

Input locations are not updated.

Final storage data is retrieved only when data collection from the datalogger occurs (initiated manually from the Connect Screen or based on a schedule). Therefore, the final storage information on the data displays will be updated only as often as data collection is performed.

6.1.7.2 Table Monitor

The Table Monitor in the center of the Connect Screen can be used to monitor the values for one entire table of a table data datalogger.

To begin, select the table you wish to monitor from the drop-down list. The fields of the specified table will be displayed in the Field/Value grid.

If desired, select the Show Units check box to display any units that have been assigned to data values in the table.

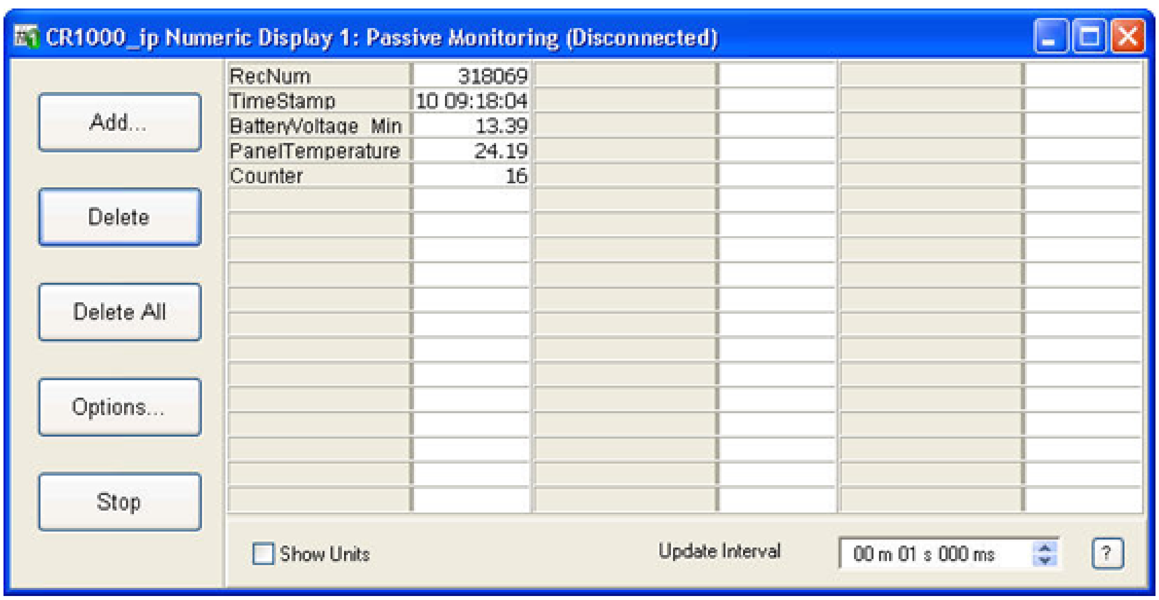
When data is being monitored, you can press the **Stop** button to stop the monitoring of data. The text on the button will change to **Start** and it can be pressed to start monitoring data again.

The **Interval** determines how often data in the table monitor will be updated. This interval controls how often the table monitor is updated, only when you are actively connected to a station (by pressing the **Connect** button). This is referred to as real time monitoring. When you are not actively connected to a station, the table monitor will be updated only when data is collected (on a schedule or by pressing the **Collect Now** or **Custom** buttons). This is referred to as passive monitoring. For more information on real time monitoring and passive monitoring, click Real Time Monitoring vs. Passive Monitoring (p. 93).

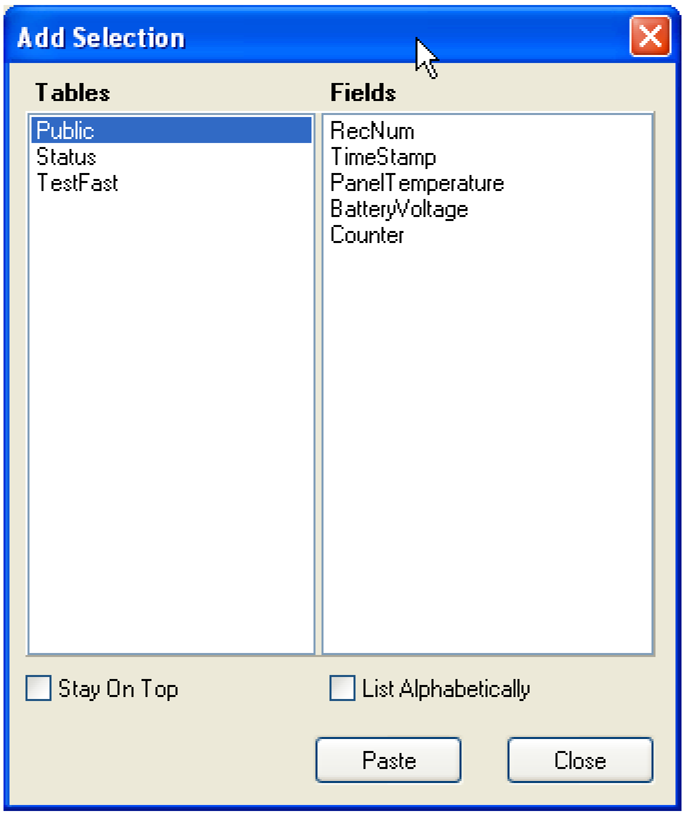
Double-clicking on a value in the Table Monitor will display the value in a separate dialog box and allow you to change the value (if it is editable).

6.1.7.3 Numeric Display

Real-time data and final storage data can be viewed on the Numeric Displays. There are three displays that can be set up with different data points. By default, each display has 3 columns of 18 cells.



Press the **Add** button to bring up the Add Selection dialog box which displays a list of data values that can be viewed on the Numeric Display. Tables and other collection areas are displayed on the left side of the dialog box. When an item is selected in the left hand column, the input locations, variables, or data table fields associated with that item are displayed on the right. These values can be dragged from the list and dropped on to a cell in the Numeric Display, or you can highlight a value, highlight the cell in which to place it, and then press **Paste**. An existing data value on the Numeric Display can be dragged and dropped to a different cell if desired. Additionally, for dataloggers with a mixed-array operating system, you can press the Input Location IDs button and type in a range of values to be displayed. Selecting the **Stay on Top** check box will keep the Add Selection dialog box on top of all other opened windows, until it is closed. By default, Fields are listed in the order they are declared in the CRBasic program. Selecting the **List Alphabetically** check box will cause the fields to be listed in alphabetical order.



Highlight cells and press the **Delete** button to remove data values from the display.

Press the **Delete All** button to remove all data values from the display.

The **Options** button can be pressed to configure various display, alarm, and setup options for each cell.

If desired, select the Show Units check box to display any units that have been assigned to data values being displayed.

The **Start** button must be pressed to begin monitoring the data values. The name of this button then changes to **Stop**, and it can be pressed to stop monitoring the data values.

The arrow to the right of the buttons can be pressed to hide the buttons and allow more room for the data grid. The arrow button will change direction and can then be used to restore the buttons.

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

> **NOTE:** The timestamps on input locations for mixed-array dataloggers are a combination of the time retrieved from the datalogger and LoggerNet's date. With even a slight difference between the LoggerNet clock and the datalogger clock, this can result in data that is incorrectly timestamped for data points close to midnight.

> **NOTE:** Timestamp and RecNum are special field names used by the Numeric Display to display the timestamp and the record number. If TimeStamp or RecNum is used as a variable name in your program, that variable cannot be displayed on a Numeric Display.

If your mouse pointer is left over a cell for a few seconds, the table name and field name for the data value will be displayed.

The values of input locations (or variables) and the state of ports and flags can be changed using the Numeric Display. To edit an input location value, double-click the displayed value, enter the new value, and press the **Enter** key. 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.

When a value is enabled for editing, if the **ESC** key is pressed, the change will be canceled.

Right Click Menu Options

Press the right mouse button when your cursor is within a cell to display a short cut menu. This menu contains short cuts for the **Add**, **Delete, Delete all**, and **Options** buttons, as well as a **Rename** menu item that enables a label for editing, a **Select All** menu item that highlights all of the cells in the Numeric Display, a **View/Modify Value menu** item that displays the value in a separate dialog box and allows you to change the value (if it is editable), and a **Help** menu item that brings up help for the Numeric Display.

Numeric Display Options—Display

Options can be set for one cell or multiple cells by selecting the cell(s) and then clicking **Options**.

Data Display

Numerical Format Options

Select **Automatic** to have the number of decimal places determined automatically by the data value. Select **Specify Decimal Places** to specify how many decimal places will be shown in the **Decimal Places** field. Select **Show as Timestamp** to display the data value as a timestamp.

Decimal Places

Determines the number of decimal places that will be displayed when Format is set to **Specify Decimal Places**. The maximum number of decimal places is seven. Type in a number or use the arrow keys to the right of the value.

> **NOTE:** This option will only affect the display of a floating point number.

Boolean Options

True Text

Specifies the string that will be displayed when a Boolean data value is true. Type in a string value.

False Text

Specifies the string that will be displayed when a Boolean data value is false. Type in a string value.

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

Timestamp Options

Show Dates(s)

Check this box to show both the date and time for a timestamp. Clear the check box to display only the time.

Show Milliseconds

Check this box to show milliseconds for a timestamp. Clear the check box to omit milliseconds on the timestamp display.

Color Options

Cell Color

Defines the color to be used for the background of the data value name. Press the button to the right of the color square to define a new color

Text Color

Defines the color to be used for the data value name. Press the button to the right of the color square to define a new color.

Data Cell Justification

Field Name

Indicates whether the field name will be left-justified, centered, or right-justified.

Data Value

Indicates whether the data value will be left-justified, centered, or right-justified.

Numeric Display Options—Alarms

Options can be set for one cell or multiple cells by selecting the cell(s) and then clicking **Options**.

Enable Alarms

Determines whether alarms are enabled. Select the check box to enable alarms. Clear the check box to disable alarms.

High Alarm

Select Color

Displays a list of colors from which to select a color that the data cell will turn if a high alarm condition is triggered.

Trigger Value

Sets the value that will trigger the alarm. For a high alarm to be triggered, the value must be equal to or greater than this value.

Enable Sound

Indicates whether a sound file will be executed when an alarm is triggered. Select the check box to execute a sound file. Clear the check box to disable execution. You can select a sound file by pressing the **...** button and browsing to the desired file.

Low Alarm

Select Color

Displays a list of colors from which to select a color that the data cell will turn if a low alarm condition is triggered.

Trigger Value

Sets the value that will trigger the alarm. For a low alarm to be triggered, the value must be equal to or less than this value.

Enable Sound

Indicates whether a sound file will be executed when an alarm is triggered. Select the check box to execute a sound file. Clear the check box to disable execution. You can select a sound file by pressing the **...** button and browsing to the desired file.

Sound Alarm Interval

Determines the frequency at which the audible alarm will be repeated during an alarm condition. Use the arrows to the right of the field, or type in a number directly, to specify this value.

> **NOTE:** An alarm can also be set for a Port or a Flag for those dataloggers that represent ports and flags as floating point numbers (CR200 and CR9000). The value to use when setting the alarm for a True (or High/On) state is 1; the value to use when setting the alarm for a False (or Low/Off) state is 0.

> **NOTE:** An alarm is triggered based on the value retrieved from the datalogger. If the Number of Decimal Places field from the Numeric Display Options Display tab is set so that incoming values are rounded up or down, the display may show what appears to be the data value meeting the alarm condition when, in fact, the condition is not yet met. The alarm is not triggered until the actual condition is met.

Sound files come in a variety of formats, sample rates, etc. Not all varieties are playable on all versions of Windows. Therefore, you should test your sound file on all systems you intend to support.

Numeric Display Options—Setup

Grid Configuration

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. The maximum number of total cells is 300.

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. The maximum number of total cells is 300.

Restore Defaults

Restores the display to the default configuration of 18 rows and 3 columns.

Display Configuration

Save Config

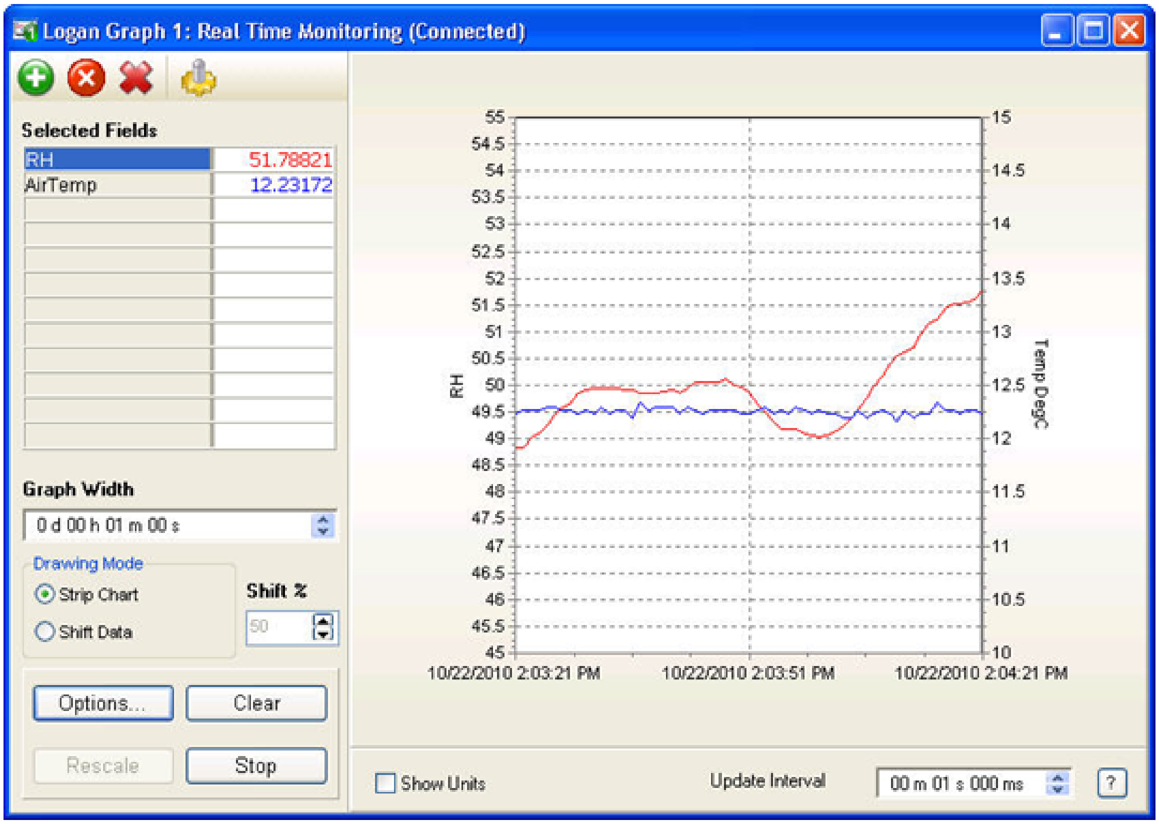
Saves the numeric display configuration to a file that can be loaded in the future.

Load Config

Loads a previously saved numeric display configuration file.

6.1.7.4 Graphs

Real-time data and final storage data can be plotted on LoggerNet's graphs. There are three displays that can be set up with up to 12 different data points each. The data values will be graphed on the y-axis against their timestamps on the x-axis.



NOTE: Timestamp and RecNum are special field names used by a Graph to display the timestamp and the record number. If TimeStamp or RecNum is used as a variable name in your program, that variable cannot be displayed on a Graph.

NOTE: The timestamps on input locations for mixed-array dataloggers are a combination of the time retrieved from the datalogger and LoggerNet's date. With even a slight difference between the LoggerNet clock and the datalogger clock, this can result in data that is incorrectly timestamped for data points close to midnight. This incorrect timestamp can cause the graphical display to stop, because the timestamp of subsequent data is older than the timestamp of the last data plotted. If you see this occurring, adjust the datalogger clock to be equal to or slightly ahead of the PC clock.

> **NOTE:** Some settings (Selected Fields, Graph Width, right-click menu) may be changed while a graph is running. However, when graphing very fast data, the response time for changes to take effect will always be better, if the graph is stopped before attempting to change the settings.

Selected Fields - To select the fields to graph, press the **Add** ( ) button. All tables will be displayed. Highlight any table to show the available fields in that table. Select fields to graph by highlighting a field and then pressing paste or dragging and dropping fields onto the Selected Fields table. Fields can be removed by highlighting the field name and then pressing the **Delete** ( ) button. All fields can be removed be pressing the **Delete All** ( ) button. Pressing the **Trace Options** ( ) button with a trace selected will bring up the Trace Options dialog box that allows you to set the color and appearance of that trace.

Graph Width - Set the amount of time to be displayed on the graph in hours, minutes, seconds and milliseconds. Type in the numbers directly or use the arrow keys to increase or decrease graph width. The minimum graph width is 1 millisecond.

Drawing Mode - Choose the Drawing Mode for the graph. The choices are Strip Chart or Shift Data. In Strip Chart mode, the data will stream across the graph. After the graph is filled, the oldest points will fall off the left edge of the graph as new points are added to the right edge. If Shift Data is chosen, the data will be positioned in a static location. Once the graph is filled, the data on the graph will be shifted over. The size of this shift and, therefore, the amount of data that will be removed from the graph is determined by the percentage specified in the Shift % field.

Options - Brings up the Options dialog box for the graph. This dialog box can be used to set options for scaling, data display, visual display, and NAN values. A graph configuration can also be saved or loaded from the dialog box.

Rescale - Press this button to scale all of the data points so they fit within the graph axis. This button is available only if Powers of 10 scaling is being used for the graph.

Clear - Press this button to clear the data contained in the graph and continue graphing with a fresh screen.

Start/Stop Button - Press the **Start** button to begin the graphing of data for the selected field(s). When data is being graphed, the text on this button will change to **Stop**, and it can be pressed to stop the retrieving and graphing of data.

Show Units – Select this check box to display any units that have been assigned to data values being displayed.

Update Interval – Specifies how often data in the graph will be updated. Note that data can be updated no faster than the data values are being generated by the datalogger.

The arrow to the right of the **Add**/**Delete**/**Delete All**/**Trace Options** buttons can be pressed to hide the settings pane and allow more room for the graph. The arrow button will change direction and can then be used to restore the settings pane.

Right Click Menu Options

Pressing the right mouse button will bring up a short cut menu relative to where your mouse pointer is on the graphical display.

Right Click Within the Graph - Displays a short cut menu with items for saving, printing, and formatting the graph:

Save As - Allows you to save a picture of the current graph in a BMP or WMF format.

Copy - Save a copy of the current graph image to the Window's clipboard. This copy can then be pasted into another application.

Options - Opens the Graph Display Options dialog box.

Clear - Erases the existing traces on the graph.

Rescale – Scales the data values so they are all displayed within the graph boundaries. This option is available only when Powers of 10 Scaling is chosen for the graph.

Start/Stop - Starts the graphing of data when a graph is currently stopped. Stops the retrieving and graphing of data, when a graph is currently running.

Print Preview - Displays a preview of the printed page with the ability to set the paper orientation, page margins, and other print properties.

Print - Brings up the standard windows Print dialog box so that the graph can be printed.

View Statistics - Displays the average value, minimum, maximum, and number of data points for each data value being displayed. (Note that these values are for the default graph view, i.e. not zoomed or panned.)

Right Click on a Table Cell - Displays a short cut menu with options specific to traces.

Add - Brings up the Add Selection dialog box from which you can add a trace to the graph.

Delete - Removes the trace from the graph.

Rename - Sets the name of the field to a state in which it can be edited.

Do Not Plot - Stops the trace from being plotted on the graph. A check mark appears beside the Do Not Plot menu item for a trace that will not be plotted. Record numbers and timestamps are not plotted.

Delete All - Resets all settings for the traces on the graph. This will remove all traces from the graph.

Select All - Selects all traces on the graph. This allows options for all traces to be set at once.

Show Symbol – Determines whether symbols are displayed for this data value. (Symbols are set from the Trace Options dialog box.)

View/Modify Value - Displays the value in a separate dialog box and allows you to change the value (if it is editable).

Trace Options - Displays a dialog box that lets you set the color and appearance of the trace.

Help - Brings up help for the Graph.

Additional Capabilities

Vertical Line Marker

Double-clicking a point on the graph will bring up a vertical line marker on the screen that moves with the cursor. As the vertical bar is moved across the graph, the Selected Fields box will display the data value(s) corresponding to the timestamp of the vertical marker. (If a vertical line marker is brought up while graphing is underway, the graph will be automatically paused. It will be necessary to push **Start** to resume graphing.)

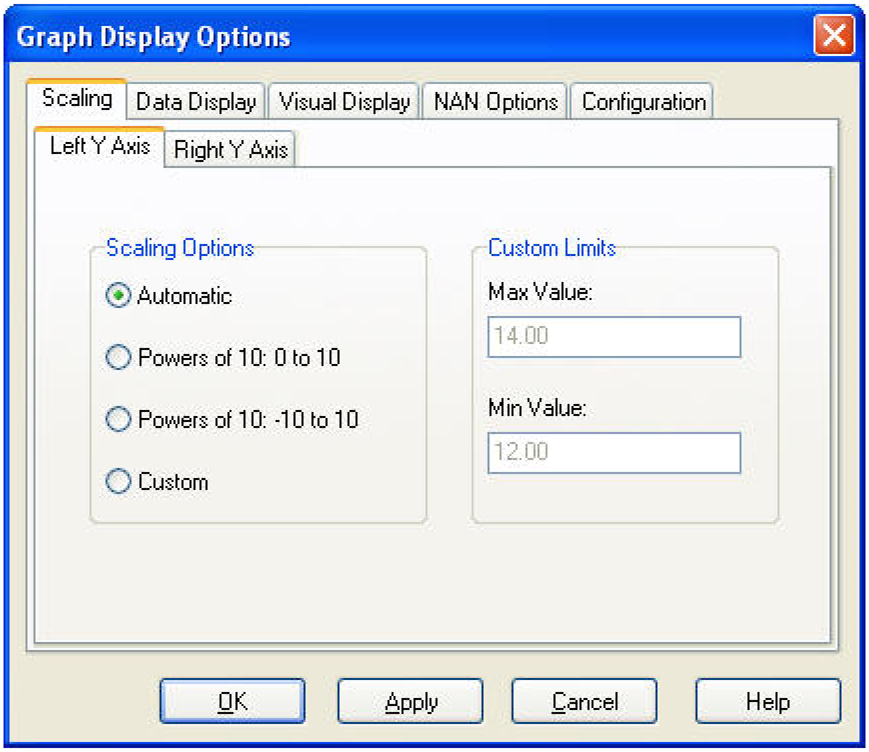
Zoom and Scroll

An area of the graph can be zoomed in on by using the mouse pointer to draw a box around the area to be viewed. (Place the mouse cursor in the area for the upper left of the box, press the mouse button, and hold and drag the mouse pointer to the desired bottom right corner of the box.) To return to normal view, press the **Undo Zoom** button in the upper right corner of the graph. (You may also press the mouse button, hold and drag the mouse pointer up and to the left to return to normal view.)

If you have stopped a graph and zoom in to a region, you can use the right mouse button to drag the screen and thus scroll to other locations of the graph at the current zoom level.

Graph Display Options—Scaling

The Scaling tab of the Graphing Options dialog box is used to set scaling options for the axes of the graph. Choose the tab corresponding to the axis for which you wish to set the scale (Left Y-Axis or Right Y-Axis).



Once the graph options are set, press **Apply** or **OK** to make the changes. (If **Apply** is pressed, the dialog box remains open. If **OK** is pressed, the dialog box is closed.) Press **Cancel** to remove any changes that have not yet been applied.

Scaling Options

Automatic Scaling - LoggerNet will automatically adjust the Y axis scale so that all traces are displayed on the graph.

Powers of 10: 0 to 10 - Data values will be scaled so that they fit on a graph ranging from 0 to 10. Negative values will not be displayed. Each trace is scaled based on its maximum value. If that value is greater than 10, all of the points in the series are repeatedly divided by 10 until the maximum value is less than or equal to 10. If the maximum value is greater than 0 but less than or equal to 1, all of the points in the series are repeatedly multiplied by 10 until the maximum value is greater than 1. Note that scaling occurs when the **Apply** button is pressed on the dialog box. Rescaling does not automatically occur if the maximum value goes out of the range. However, you may press the **Rescale** button to recalculate the scale at any time.

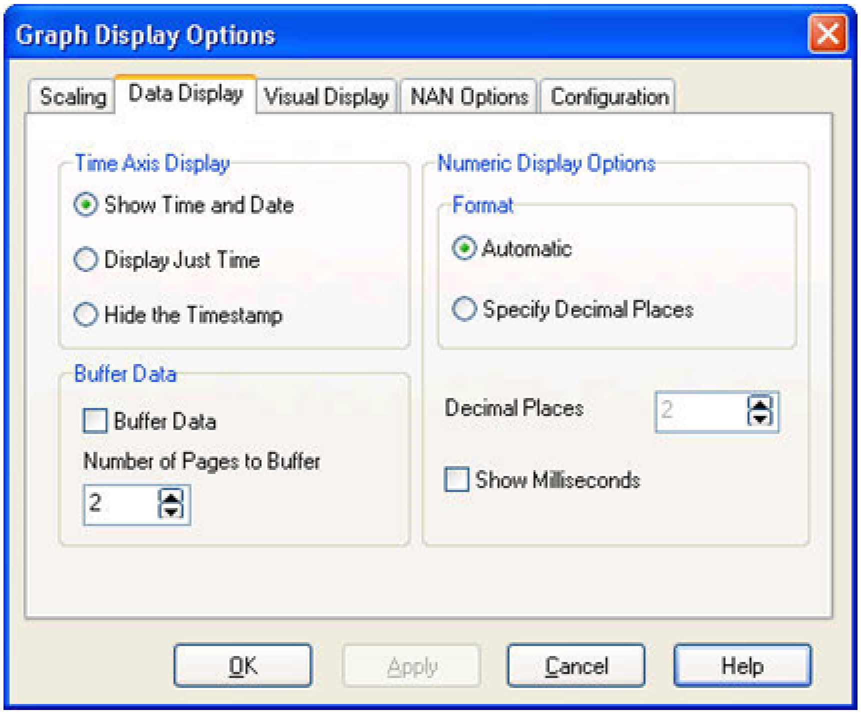
Powers of 10: -10 to 10 - Data values will be scaled so that they fit on a graph ranging from -10 to 10. Each trace is scaled based on its maximum and minimum values. If the maximum value is greater than 10 or if the minimum value is less than -10, all of the points in the series are repeatedly divided by 10 until the maximum value is less than or equal to 10 and the minimum value is greater than or equal to -10. If all of the points in the series are greater than or equal to -1 and less than or equal to 1, all of the points in the series are repeatedly multiplied by 10 until the at least one value is outside that range. Note that scaling occurs when the **Apply** button is pressed on the dialog box. Rescaling does not automatically occur if the values go out of the range. However, you may press the **Rescale** button to recalculate the scale at any time.

Custom - Lets the user specify an upper and lower value for the Y axis. The graph will not display traces that fall outside of the specified limits.

Custom Limits - These fields (**Max Value** and **Min Value**) are used to enter an upper and lower value for the Y axis. These fields are disabled if **Custom** is not selected as the Scaling Option.

Graph Display Options—Data Display

The Data Display tab of the Graphing Options dialog box is used to set options determining how data is displayed in the graph.



Once the graph options are set, press **Apply** or **OK** to make the changes. (If **Apply** is pressed, the dialog box remains open. If **OK** is pressed, the dialog box is closed.) Press **Cancel** to remove any changes that have not yet been applied.

Time Axis Display - These options affect the format of the timestamp for the bottom axis of the graph. You can choose to **Show Time and Date**, **Display Just Time**, or **Hide the Timestamp**.

Numeric Display Options - Select **Automatic** to have the number of decimal places determined automatically by the data value. Select **Specify Decimal Places** to specify how many decimal places will be shown in the **Decimal Places** field. Check the **Show Milliseconds** check box to show milliseconds for the timestamp. Clear the check box to omit milliseconds on the timestamp display.

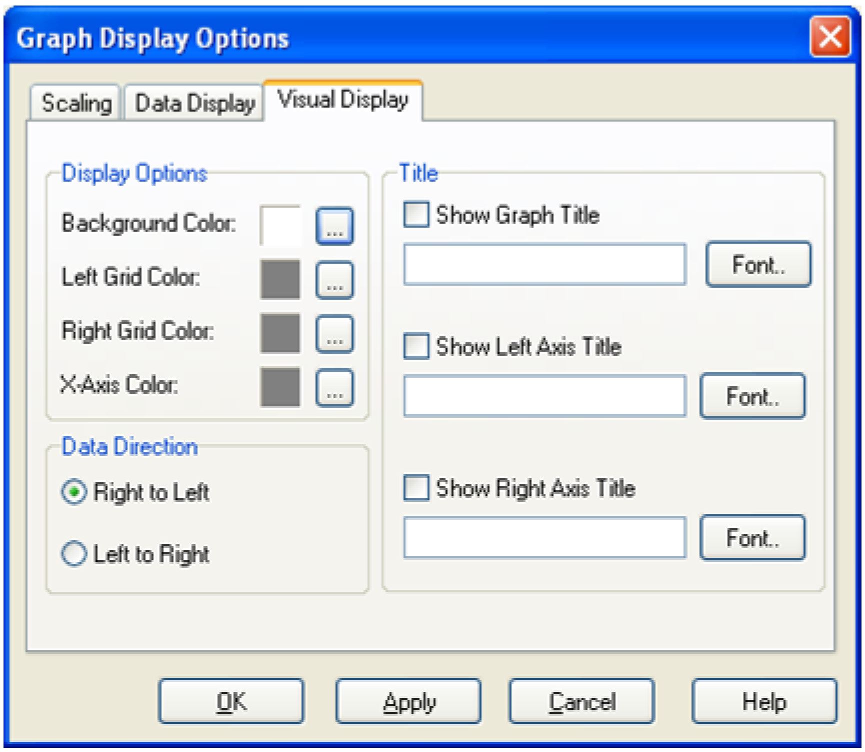
Buffer Data - Determines whether data coming into the graph is buffered. Select the **Buffer Data** check box to buffer data. Clear the check box to prevent data from being buffered.

The amount of data to be buffered is specified in the **Number of Pages to Buffer** field. Each page will contain the amount of data specified by the **Graph Width**.

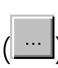
When the **Buffer Data** option is on, a stopped graph can be scrolled backward in time for the amount of pages specified. The paused graph will have arrows which appear to facilitate moving backward or forward one page at a time, moving to the earliest page, or moving to the latest page. You can also manually navigate through the buffered data by right-clicking and dragging to move backward or forward in time.

Graph Display Options—Visual Display

The Visual Display tab of the Graphing Options dialog box is used to set colors, titles, and data direction for the graph.



Once the graph options are set, press **Apply** or **OK** to make the changes. (If **Apply** is pressed, the dialog box remains open. If **OK** is pressed, the dialog box is closed.) Press **Cancel** to remove any changes that have not yet been applied.

Display Options - Press the browse button () to the right of each field to display a color palette from which to choose a custom color for the graph element. The grid colors are the grid lines for the left and right axes.

Data Direction - Choose whether the traces are displayed on the graph from Right to Left (newest data is displayed on the right side of the graph) or Left to Right (newest data is displayed on the left side of the graph).

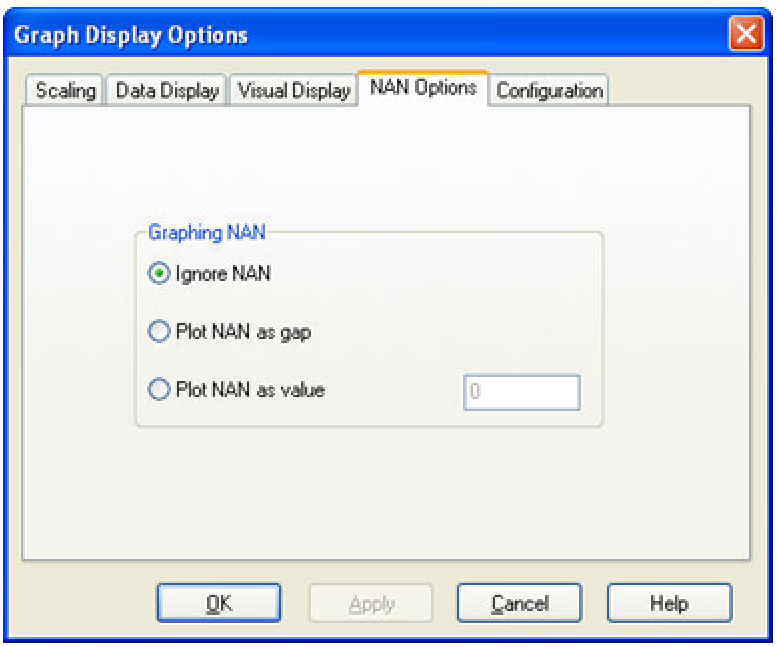
Title - Enter a title that will be centered over the top edge of the graph and for each axis.

Show Title - Select this check box to display the specified title. Clear the check box to remove it.

Font - Press this button to display a font dialog box from which you can select a font for the specified title.

Graph Display Options—NAN Options

The NAN Options tab of the Graph Display Options dialog box is used to specify how NAN (Not-A-Number) values will be represented in the graph.



Once the graph options are set, press **Apply** or **OK** to make the changes. (If **Apply** is pressed, the dialog box remains open. If **OK** is pressed, the dialog box is closed.) Press **Cancel** to remove any changes that have not yet been applied.

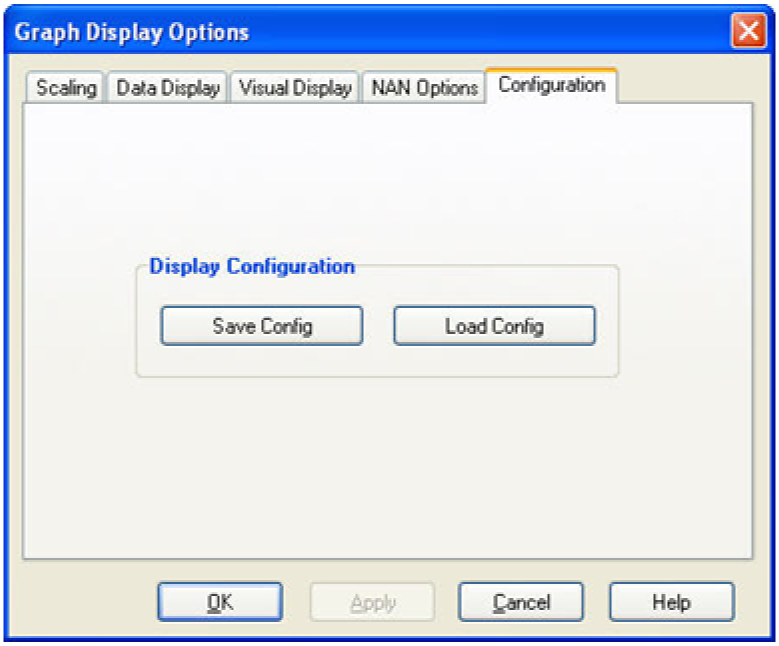
Ignore NAN - If this option is selected, the NAN is not represented on the graph. There will be one continuous line in which the data points adjoining the NAN values will be bridged

Plot NAN as gap - When this option is selected any NAN value in the data will be represented by a discontinuity. This means that the data points on either side of the NAN will not be connected by a line. There will be breaks in the line for each NAN in the data.

Plot NAN as value - With this option is selected, each NAN value in the data will be represented by the specified value.

Graph Display Options—Configuration

The Configuration tab of the Graph Display Options dialog box is used to Save or Load graph configurations.



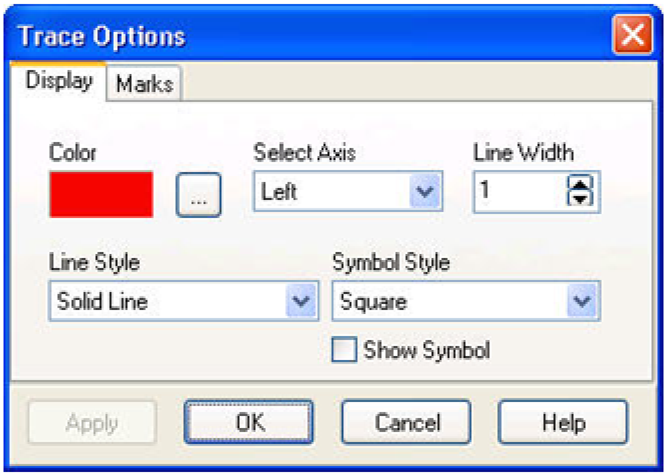
Once the graph options are set, press **Apply** or **OK** to make the changes. (If **Apply** is pressed, the dialog box remains open. If **OK** is pressed, the dialog box is closed.) Press **Cancel** to remove any changes that have not yet been applied.

Save Config - Saves the graph configuration to a file that can be loaded in the future.

Load Config - Loads a previously saved graph configuration file.

Trace Options—Display

Trace Options allows you to control how the trace will look on the graph.



Color - Displays the color currently selected for the trace. Press the button to the right of the color square to display a color palette from which to select a new color.

Select Axis - Use the drop-down list box to choose whether to plot the trace on the left or right axis.

Line Width - Use the arrows to the right of the field to increase or decrease the width of the line that will represent the data value.

Line Style - Use the drop-down list box to select the style for the line that will represent the data value.

Symbol Style - Use the drop-down list box to select the symbol that will represent each data point.

Show Symbol – Determines whether symbols are displayed for this data value. Select the check box to have symbols displayed. If the check box is cleared, no symbols will be displayed.

Trace Options—Marks

Marks are the labels for the data points on the graph. The items on this tab affect how the marks appear on the graph.

Show Marks - When selected, the labels for the data points are displayed on the graph. When cleared, they are not displayed and all other items on this tab are disabled.

Draw Every - Determines how often the data points should be labeled. When set to 1, every data point will be labeled. When set to 2, every second data point will be labeled, and so on.

Round Frame - When selected, the frame that surrounds the data label will have rounded corners. When cleared, the frame will have angled corners.

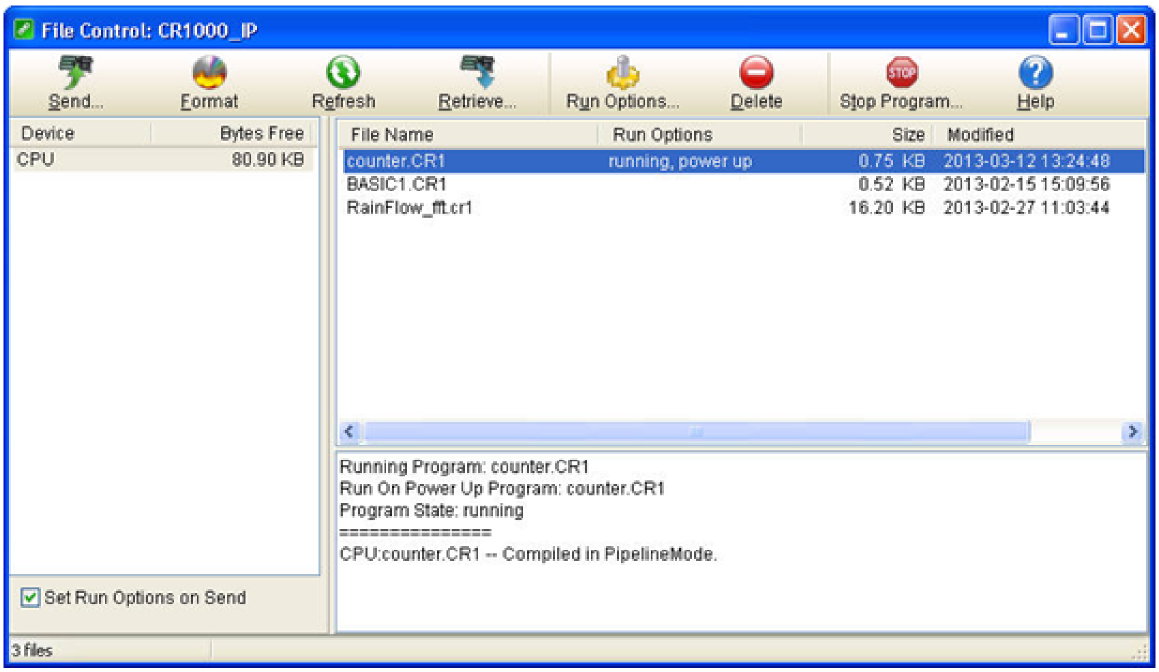
Transparent - When selected, a frame will not be displayed for the data labels.

Color - Defines the color of the background for framed data labels. Press the button to the right of the color square to define a new color.

Data - Determines the type of label to be displayed for data point. **Value** displays the numeric value for the data point. **Timestamp** displays the time for the data point. **Value and Timestamp** displays the time and the numeric value for the data point.

6.1.8 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.

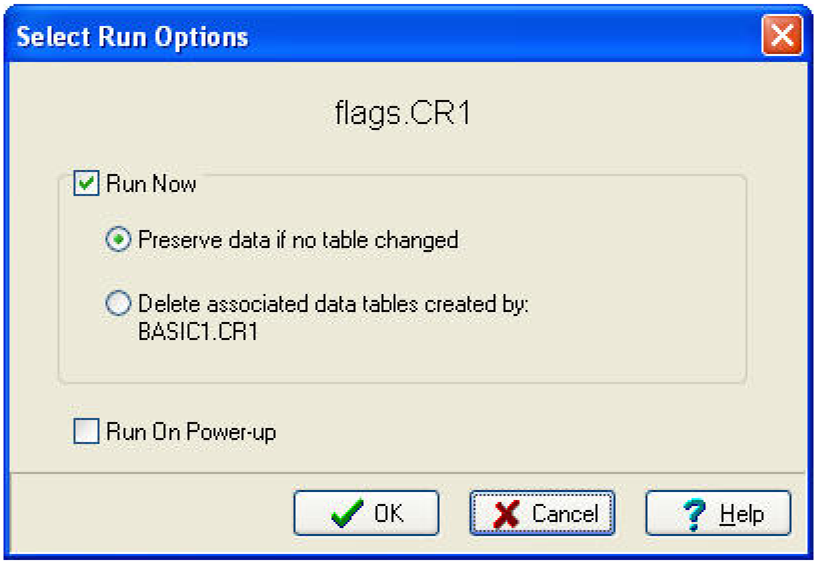
> **NOTE:** File Control should not be used to retrieve data from a CF card created using the CardOut instruction. Using File Control to retrieve the data file can result in a corrupted data file.

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.

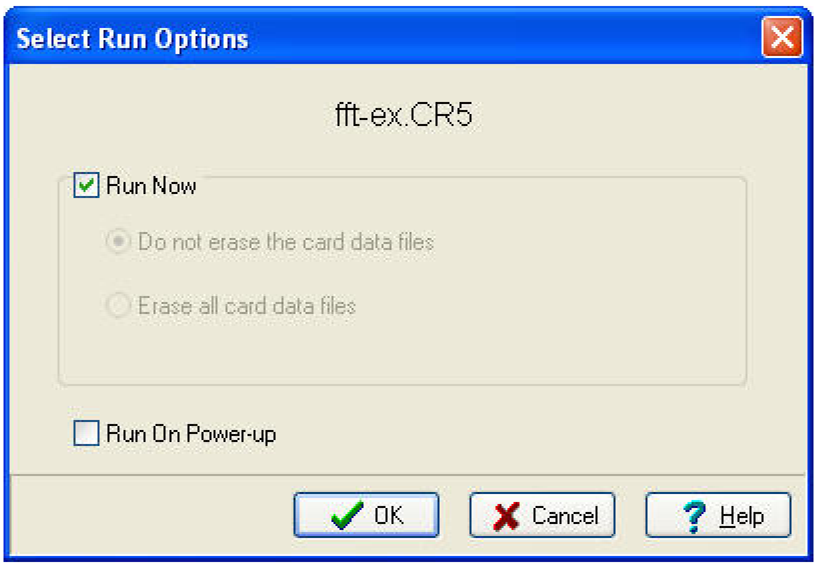
When using the **Preserve data if no table changed** option, existing data and data table structures are retained unless one of the following occurs:

- 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.

6.1.9 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.

6.1.9.1 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 .

6.1.9.2 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.

6.1.9.3 Peripheral Device Test

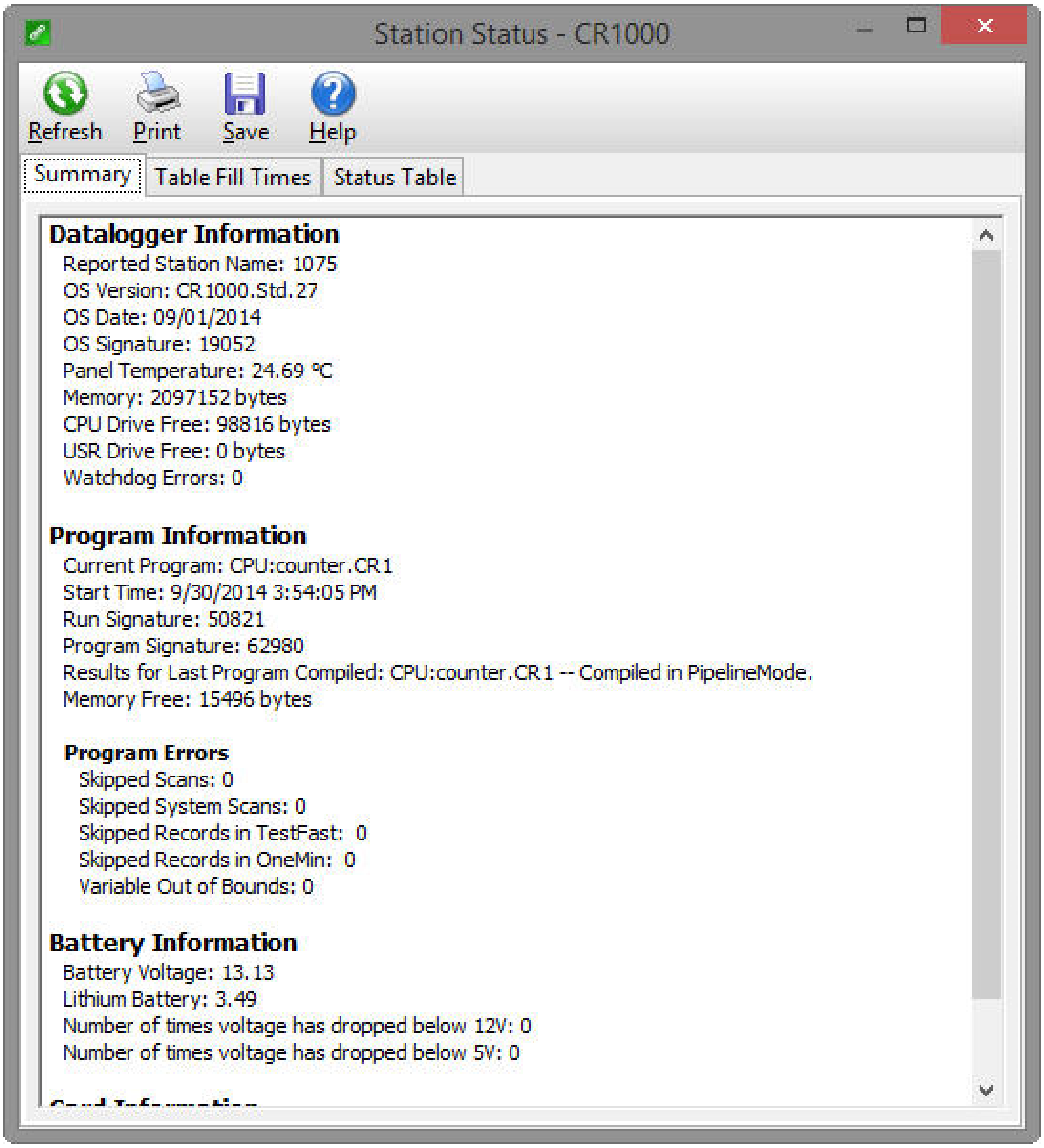
The Terminal Emulator can also be used to test a peripheral device attached to the COM port such as a phone modem. Commands can be sent to test initialization strings or to manually dial a modem to connect to the datalogger.

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.

6.1.10 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/modif**y 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 LoggerNet 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.)

6.1.11 Calibration Wizard

The Calibration wizard, opened from the Connect Screen’s Datalogger menu, is used to assist in the calibration of one or more variables in the datalogger program. This calibration tool is available only for the CR1000X-series, CR6 series, CR300-series, GRANITE 6, GRANITE 9, GRANITE

10, CR1000, CR3000, CR800-series, CR5000, and CR9000X dataloggers that support the FieldCal instruction.

The program running in the datalogger must contain one or more FieldCal or FieldCalStrain instructions for the variables you wish to calibrate. One function of this instruction is to write a programname.cal file to datalogger memory that contains information on the variables to be calibrated and the most recent calibration values. The Calibration wizard looks for a *.cal file with a name that matches the program currently running in the datalogger. If a matching file is found, the calibration wizard will use the information from that file to walk you through each step of the calibration. Calibration options offered in the FieldCal instruction are zeroing, offset, two-point multiplier and offset, and two-point multiplier only. Calibration options offered in the FieldCalStrain instruction are zeroing, 1/4 bridge strain shunt, bending 1/2 bridge strain shunt, and bending full bridge strain shunt.

More information on calibration and zeroing is found in Calibration and Zeroing (p. 483). Also, refer to the datalogger's CRBasic help file for additional information on the FieldCal and FieldCalStrain instructions, and to the LoggerNet help file if help is needed while using the Calibration wizard.

6.2 RTMC Real Time Monitoring and Control Software

RTMC is an application used to graphically display data from a LoggerNet datalogger network or RTDAQ datalogger. Available components include status bars, digital displays, charts, gauges, tables, and alarms. In addition, RTMC can be used to remotely control a datalogger and its peripherals by toggling ports and flags with switch components, and it can be used to set input locations or variables in a datalogger with value setter components.

> **NOTE:** Starting with LoggerNet 4.8, RTMC will no longer be updated if a version of RTMC Pro earlier than 5.0 is installed on the computer. This is to maintain full compatibility with the installed RTMC Pro package.

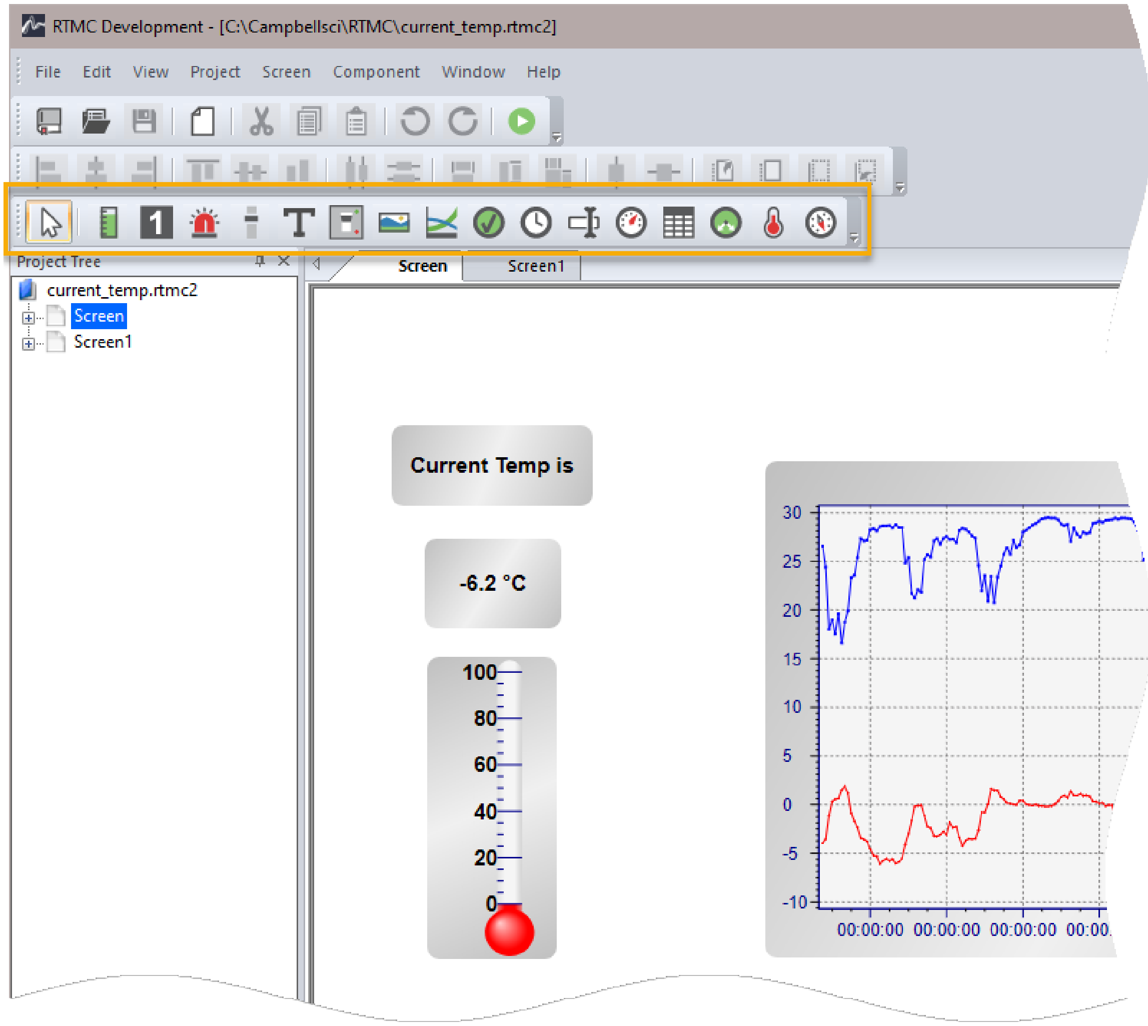
RTMC's main window is divided into a Workspace and the Project Tree. The Workspace, on the right hand side, is where the components will be placed for the display. As each component is added to the Workspace, it will show up in the Project Tree.

Adding Components to the Workspace

Click an item on the Component Toolbox (see below), and then click within the Workspace to add the component to the display. When an item is selected from the Component Toolbox, the cursor will change to an arrow and the component name to indicate the selection (

). If you

decide not to insert the component, press the Pointer ( ) to return the cursor to its normal state.

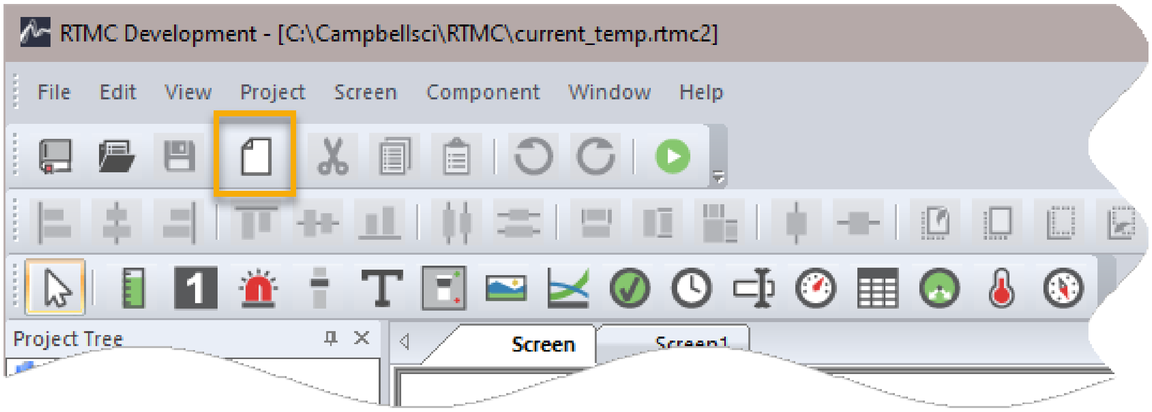


For more information, see Component Toolbox (p. 125).

Adding Tabbed Screens

When a new project is started in RTMC, it has only one screen on which to place components. However, more screens can be added. Each screen will appear in the project tree and as a tabbed page in the workspace. A new screen can be added using the **Project | Add New Screen** menu or by selecting an icon from the Standard Toolbar. Navigate between the screens using the tabs at the top of the workspace.

The Add Screen icon is highlighted below.



The order in which the screens appear can be changed by selecting **Project | Change Screen Order** from the menu. A dialog box with all the screen names listed is displayed. Highlight a screen name and use the up or down arrow buttons to change the order of the screens.

Execution Modes

Development Mode - The development mode is used when creating projects to be displayed by RTMC. The development mode is launched when RTMC Development is selected from the LoggerNet or RTDAQ toolbar.

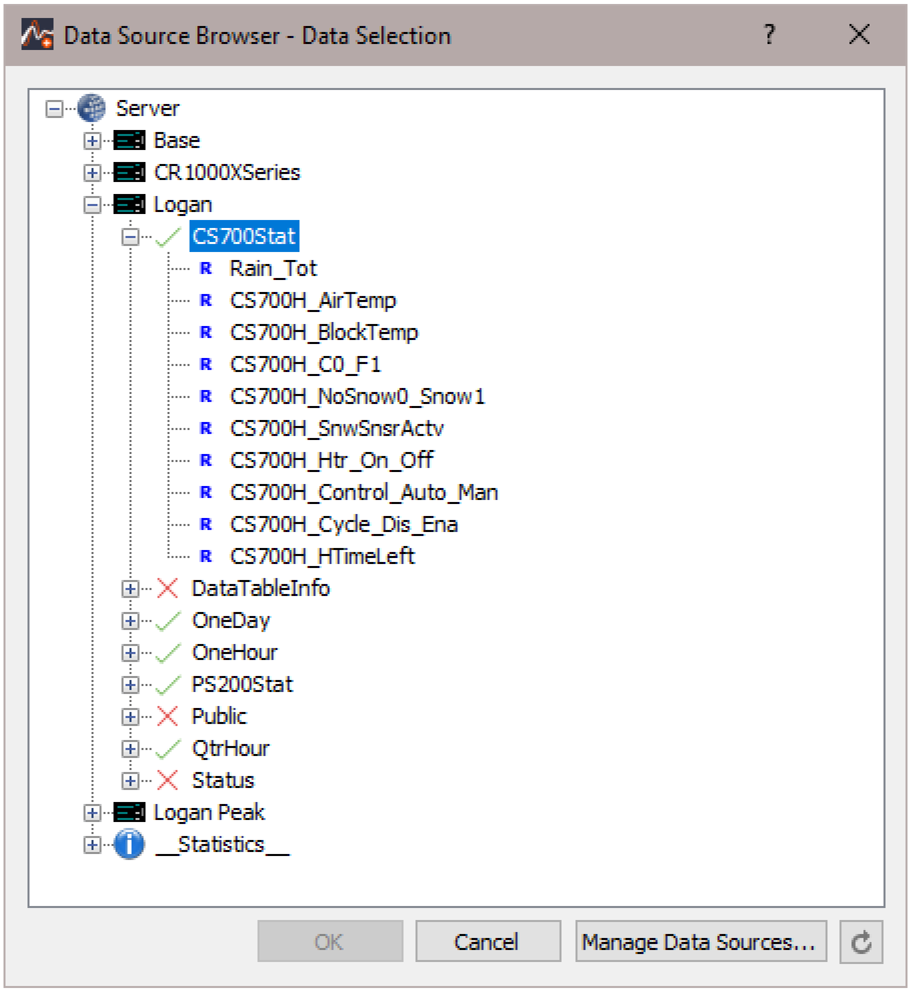
Run-time Mode - The run-time mode is used to display a completed project. The run-time mode is launched when RTMC Run-time is selected from the LoggerNet or RTDAQ toolbar. In LoggerNet, it can also be started by choosing **All Apps | Campbell Scientific | RTMC Run-time** from the Windows Start menu or by Saving and Running the project within the development mode (File | Save and Run Project or the icon.)

> **NOTE:** In RTDAQ, you must be connected to a datalogger for the RTMC Run-time icon on the RTDAQ toolbar to be enabled. Also, the option to Save and Run Project and the associated button are not available when running the development mode of RTMC from RTDAQ.

Data Displayed by RTMC

LoggerNet - RTMC displays data from LoggerNet's binary data cache, not from a data file. The data cache is updated during scheduled or manual data collection in LoggerNet. In RTMC, when associating a data value from a table with a component, a table represented with a red X indicates the table is not being collected in LoggerNet. This is because scheduled collection is not enabled for the datalogger, the table is not included for collection, or the schedule has been paused from the Status Monitor.

For instance, in the example below, the Public, Status, and DataTableInfo tables are not being collected from the Logan datalogger.

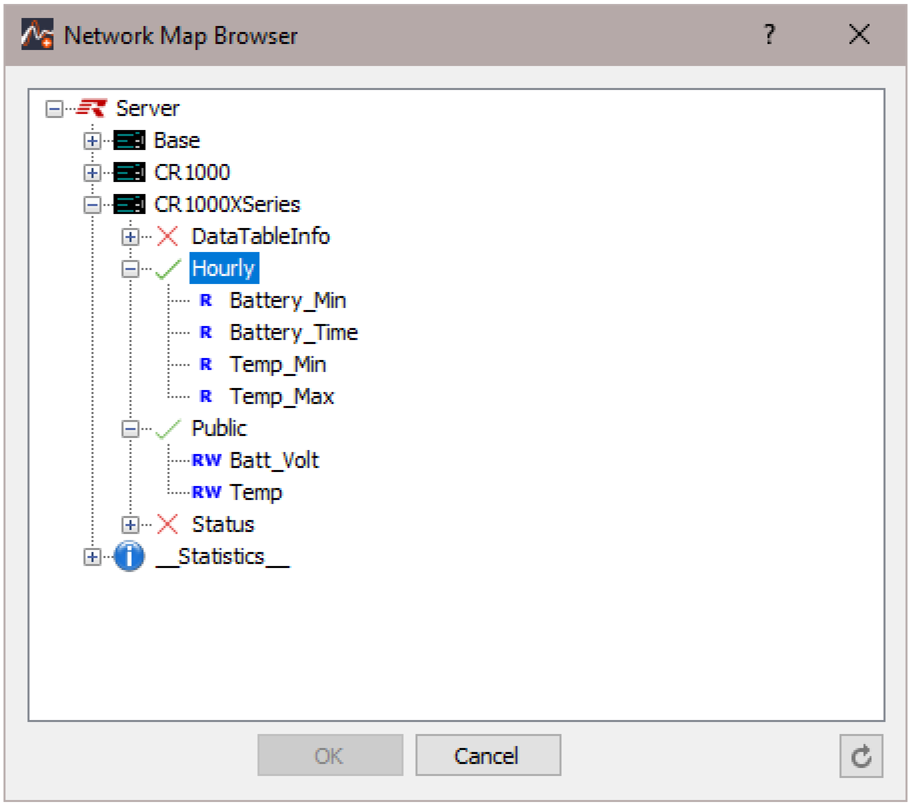


In this instance, if a data value from the Public, Status, or DataTableInfo table is associated with a component, that component will likely not be updated when RTMC is run. Scheduled collection can be enabled and tables can be included for data collection in LoggerNet's Setup window.

Data displays will be updated no more frequently than data is being collected from the datalogger.

Input locations, public variables, ports, and flags will show only the current value in the datalogger -- the datalogger does not keep historical records for these values. A "snapshot" of these values will be updated only when data is collected.

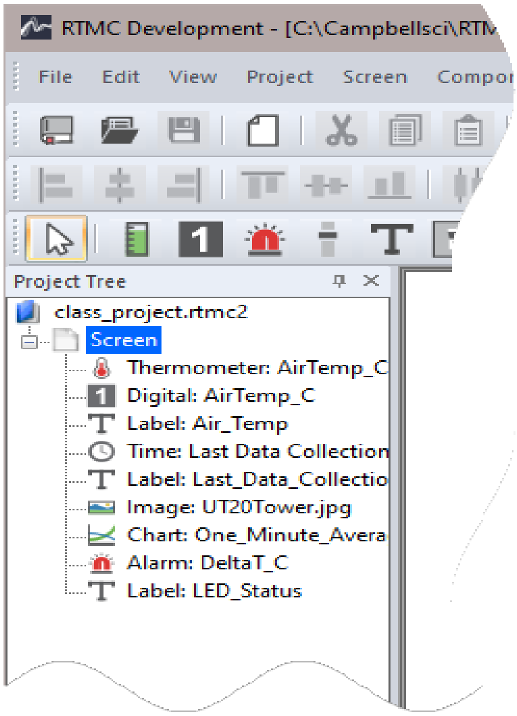
RTDAQ – In RTDAQ, RTMC projects are limited to a single station. This station must be specified when a new project is opened. Thus when components are added to a project, only the specified station's tables and fields will be shown in the Data Source Browser. When associating a data value from a table with a component, the station's table will be represented with a green check mark when RTDAQ is connected to the station and monitoring the table through a real-time window (field monitor, table monitor, graph, etc.). A table will be represented with a red X when RTDAQ is not monitoring the table.



When RTDAQ is connected to the station, RTMC will perform a manual poll to update the project data every second. Thus, RTMC data displays will only be updated when RTDAQ is connected to the station.

6.2.1 Project Tree

The Project Tree lists the project, each screen, and all of the components for each screen in a tree-type structure. As an example, look at the following Project Tree:



Note that when you first add a new component to a screen, it has a default name that reflects the component type; e.g., "Thermometer". You can provide a meaningful name for the component by selecting the component in the Project Tree, pressing F2, and typing in a new name (alternately, you can select the component in the Project Tree, right-click, and select **Rename Component**). RTMC does not allow duplicate component names. If a second thermometer is added it will be named "Thermometer1". When a component is selected in the Project Tree, it will also be selected in the Workspace.

To move between the different screens of a project, select the screen name in the Project Tree or the tab above the workspace.

6.2.2 Workspace

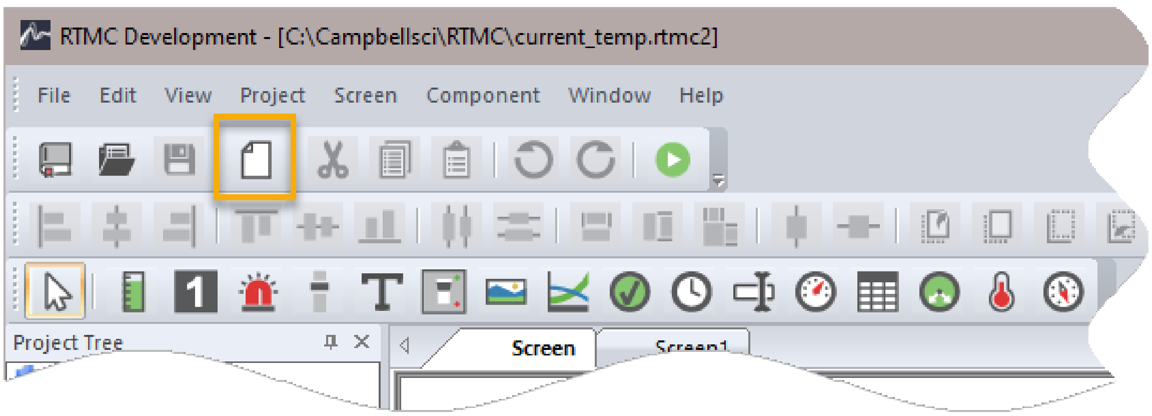
The Workspace is the large section on the right side of the RTMC window. Your finished project will reflect the layout of the Workspace when it is run.

The Workspace Configuration (p. 139) dialog box can be used to determine the size of the Workspace and the Run-time screen.

To place a component in the Workspace, select an item from the Component Toolbox. Your cursor will look similar to  . With this cursor, click in the Workspace where you want the component placed. The component will be dropped onto the Workspace and the Properties window for that component will be displayed.

Once the properties are set and you have pressed **OK**, you can move or resize the component as desired. To move a component, click on it and hold the mouse button while dragging it to a new location. To resize a component, click somewhere on its outer border and drag the image larger or smaller. Note that if you resize an object, you may need to adjust some of its properties so that scale values, bars that depict the data, etc., are the correct size and position for the new image. The aspect ratio of a component can be locked or unlocked prior to resizing it by right-clicking on the component and enabling/disabling the Lock Aspect Ratio option.

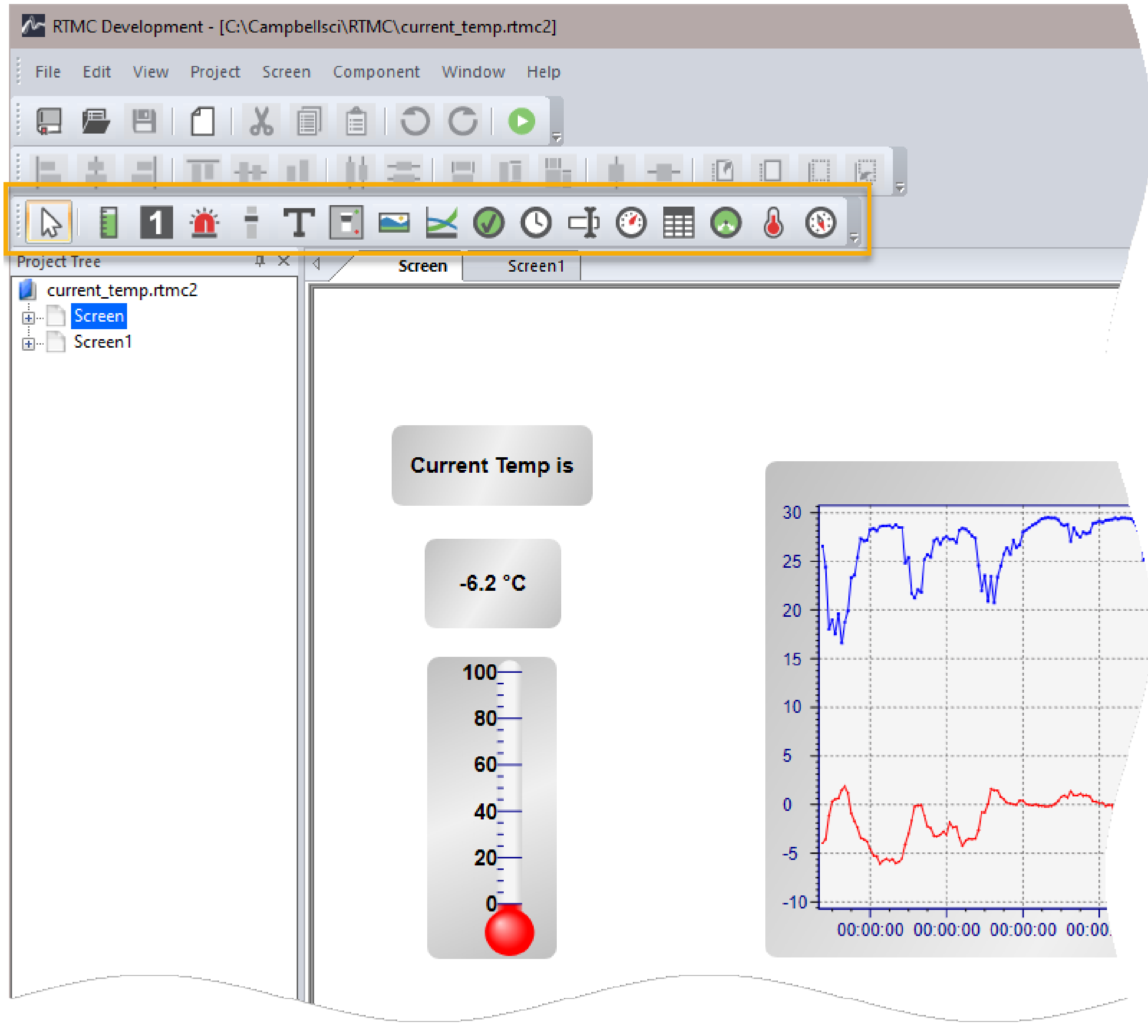
One RTMC project can contain multiple tabbed windows. A tabbed window is created by adding a new screen (**Project | Add New Screen** or press the icon on the Standard Toolbar). To navigate between screens in development mode, select a screen in the Project Tree.



For more information on manipulating the components on your Workspace, see Component Selection and Manipulation (p. 129).

6.2.3 Component Toolbox

Components are the objects that are used to display data and create the graphical layout of the project. To add a component to the Workspace, click on an item in the Component Toolbox, then click anywhere in the Workspace. The component's Properties window is automatically displayed when the component is first placed in the Workspace. The Properties window is used to set colors, scale values, define text, etc., and to assign the data value to be displayed by the component.



After a component's properties have been set, select **OK** to apply the changes and close the Properties window. Once closed, the Properties window can be displayed by double-clicking the component. If you make changes to a component's properties but then decide to reject those changes, press the **Cancel** button to return the properties to the last applied state. If **Cancel** is selected when a component is first placed in the Workspace (and **OK** has not been pressed), the component will be removed from the screen.

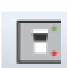
> **NOTE:** In LoggerNet, RTMC displays data from the data cache (and not from a collected data file). Data must be collected from the datalogger for the data cache to be updated, and therefore, for the RTMC displays to be updated. Typically this is done by setting up a schedule in LoggerNet's Setup window. Data displays will be updated no more frequently than data is being collected from the datalogger.

NOTE: In RTDAQ, when RTDAQ is connected to the datalogger, RTMC performs a manual poll every second to update project data. Therefore, RTDAQ must be connected to the datalogger for the RTMC displays to be updated.

NOTE:  An exclamation point in a red circle at the upper right of a component indicates a problem with the linked data value. Hover your mouse pointer over the box for a few seconds to display a hint about the error RTMC has detected.

NOTE:  A yellow N in a red circle at the upper right of a component indicates the value of the field referenced in the Select Data field of the component is currently a NaN.

The following components are available in RTMC.

Button Icon	Component	Description
	Pointer	Returns the cursor to a normal selection tool.
	**Status Bar**	Depicts the selected data value as a single vertical bar.
	**Digital**	Depicts the selected data value as a numeric value, text string or Boolean.
	**Alarm**	Provides visual and/or audible notification that a data value has exceeded a defined limit. An audible alarm can be silenced by right-clicking the component with your mouse and selecting **Acknowledge Alarm**.
	**Slider**	Depicts the selected data value as a single horizontal bar. The data value can also be set to a new value by dragging the slider.
	**Label**	Displays a text string that can be used to label other components.
	**Switch**	Indicates the state of a port, flag, input location, or public variable. A 0 is considered Off (false); any non-zero number is considered On (True). In run-time mode, right-click a switch to change its state. The option to change the state of a switch with a single-click or a double-click can be enabled in the Properties window.
	Image	Allows you to place a static image on the display.
	**Chart**	Displays one or more time domain series on a chart. The time stamp on the X axis reflects the data timestamp.
	**Comm Status**	Provides a visual and/or audible alarm when scheduled

Alarm	collection is disabled in the Setup Screen, the schedule is paused from the Status Monitor, or communication has failed a sufficient number of times to put the datalogger into a Primary or Secondary Retry mode (the retry mode used is based on the Sensitivity property for the component). An audible alarm can be silenced by right-clicking the component with your mouse. Note that if RTMC is launched from RTDAQ, a Comm Status Alarm will only be triggered when an RTDAQ real-time display is started or stopped. This is because RTDAQ uses scheduled collection to temporary collect areas in order to update the real-time displays.
Time	Depending upon the option chosen, displays the server time, server time at last data collection, station time, station time of last record stored, PC time, or a time stored in the data tables (such as time of maximum or minimum).
Set Point	Like the Digital component, depicts the selected data value as a numeric value, text string or Boolean. However, in run-time mode, a data value can also be set to a new value by double-clicking the component and entering a new value in the resulting dialog box.
Gauge	Displays the selected data value on a gauge.
Table Display	Displays the data from a datalogger table in a row and column format.
Value Forwarder	Reads a value in a datalogger and writes to another value in that datalogger or a different datalogger. The value that is written can be the value read, a 0 or -1, or a specified constant. Note that in RTDAQ, RTMC projects are limited to a single station. Therefore, the value read can only be forwarded to another value in the same datalogger.
Thermometer	Displays the data value on the image of a thermometer.
Compass	Provide an eight-point compass on which to display data.

NOTE: A description of each field in a component's Properties box can be displayed by pressing F1 or clicking the help button (**?**) in the top right-hand corner of the dialog box while your mouse cursor is within the box.

Expressions

Components that display data values either numerically or graphically can be processed using mathematical expressions. For instance, a temperature reading in degrees Celsius can be processed to display in degrees Fahrenheit using a mathematical expression. For a list of functions and operators that are supported, see Expressions (p. 146).

> **NOTE:** When referencing data from separate tables within the same expression, the ValueSynch f unction should be used to synchronize the data values.

6.2.4 Component Selection and Manipulation

A component in your project can be selected by clicking on the component in the Workspace or on the component's name in the Project Tree. Multiple components can be selected by holding the **Ctrl** key while clicking on the components or by using the mouse to click and drag a box around the desired components. You can select all components on the current screen using the Select All item on the Edit menu.

The last component selected will be identified by the dark blue handles and by dark blue highlighting in the Project Tree. The other selected components have handles with blue outlines and are highlighted in light blue in the Project Tree. (The last component selected is important, because many of the component manipulation commands described below are in relation to the last component selected.)

Once a component selection has been made, the selected component(s) can be manipulated using items from the Component menu as described below. All of the functions available from the Component menu can also be accessed by right-clicking on one of the selected components. Those functions which involve how the components are placed on the Workspace can also be accessed from the Layout Toolbar. (Note that icons on the Layout Toolbar will only be enabled when their functions are currently applicable based on whether no components are currently selected, one component is currently selected, or multiple components are currently selected.)

Properties - Opens the Properties dialog box for the selected component.

Delete Selection - Removes the highlighted component(s) from the Workspace.

Lock Aspect Ratio - The aspect ratio of a component is the ratio of its width to its height. When this menu item is selected, the aspect ratio of the highlighted component is locked. If the width (or height) of the component is changed, the height (or width) will automatically be changed to keep the aspect ratio constant. The lock on the aspect ratio can be temporarily overridden by holding down the Ctrl key, while dragging the boundaries of a component to resize it.

Rename - Allows you to edit the name of the selected component.

Manual Resize - Allows you to set the size and position of the selected component. Use the X and Y spin boxes to set the position of the component on the screen. Use the W and H spin boxes to set the width and height of the component.

Cut - Removes the selected component(s) from the screen and places a copy on the Windows clipboard.

Copy - Places a copy of the selected component(s) on the Windows clipboard.

Align - Allows you to align two or more components on a screen. The last component selected will be the basis on which the other components are aligned.

> **Lefts** - The left border of all selected components will be aligned.
>
> **Centers** - All selected components are positioned vertically, with their centers aligned.
>
> **Rights** - The right border of all selected components will be aligned.
>
> **Tops** - The upper border of all selected components will be aligned.
>
> **Bottoms** - The bottom border of all selected components will be aligned.
>
> **Middles** - All selected components are positioned horizontally, with their centers aligned.

Space Evenly - Used to adjust the spacing of three or more components.

> **Across** - Spaces the components horizontally across the page using the left-most and right-most components as the left and right boundaries for the spacing.
>
> **Down** - Spaces the components vertically down the page, using the top-most and the bottom-most components as the top and bottom boundaries for the spacing.

Make Same Size - Allows you to set two or more components to the same height, width, or both. The last component selected will be the basis on which the other components are sized.

Center - Used to center one or more components on the page.

> **Vertical** - Centers the component(s) vertically on the page.
>
> **Horizontal** - Centers the component(s) horizontally on the page.

Order - Lets you choose the order in which the components appear on the display (often referred to as the Z order). Select a component and choose to send it to the back, send backward (one step), bring forward (one step), or bring it to the front of all other components. This order is important when layering transparent components.

Group Selection - Allows you to group two or more components together. They can then be moved, copied, ordered, etc. as a single object.

When a component group is selected, the **Ungroup Selection** menu item will be enabled. You can undo the component grouping by selecting this menu item.

When components are grouped, the properties for each of the components will show up as an item in the Component right-click menu. These menu items can be used to modify the properties for each component.

6.2.5 Functions Available from the RTMC Menus

All of the RTMC operations are available from the menus at the top of the RTMC window. Many of the options are also available as buttons on the toolbar, or by right clicking the components or other parts of the window or project tree.

6.2.5.1 File Menu

File | New Project

Closes an open project and provides a blank Workspace in which to create a new project.

File | Open

Allows you to open a project which has been saved to disk.

File | Save

Saves the current project.

File | Save As

Allows you to save an existing project under a new file name.

File | Save and Run Project

Saves the current project, launches RTMC-RT and runs the current project.

> **NOTE:** This option is not available when RTMC is run from inside RTDAQ.

File | Exit

Closes RTMC. You will be prompted to save any unsaved changes to the active project before RTMC closes.

File | Recent Files List

Provides a list of the last five projects open in RTMC. Select a project from the list to open it.

6.2.5.2 Edit Menu

Edit | Cut

Removes the selected component from the screen and places a copy on the Windows clipboard.

Edit | Copy

Places a copy of the selected component on the Windows clipboard.

Edit | Paste

Places a copy of the Windows clipboard content on the active screen. (Note that only valid XML code can be pasted into RTMC.)

Edit | Undo

Reverses the last change that was made.

Edit | Redo

Reverses a change that was made to the Workspace using the **Undo** feature.

Edit | Select All

Selects and highlights all components on the active screen.

Edit | Clear Selection

Clears the selection of components currently highlighted on the active screen.

Edit | Preferences

Opens the Preferences (p. 133) dialog box from which you can set the visual theme for RTMC, set a new working directory for RTMC files, set a default font size for all text objects in an RTMC project, determine whether a grid is shown, and set the graphics options for the RTMC display.

In the Preferences dialog box, you can also disable the summary that is displayed when you hover over a component on the screen. The summary provides the type of component and the data value it is displaying in a hint box.

> **NOTE:** The settings in the Preferences dialog box are global for the Development environment. They are not saved within each project. Preferences are also available in RTMC Run-time. Preferences are not global between the two environments (Development and Run-time). They each save their own set of global preferences.

Preferences

> **NOTE:** The settings in the Preferences dialog box are global for the Development environment. They are not saved within each project. Preferences are also available in RTMC Run-time. Preferences are not global between the two environments (Development and Run-time). They each save their own set of global preferences.

Working Directory

Sets the working directory for RTMC files. Type in a directory or use the button to the right of the field to browse.

Show Component Summaries

Determines whether component summaries are shown when you hover over a component on the screen. Check the box to make component summaries visible. If the box is not checked, component summaries will not be displayed.

Grid Options

Show Grid

Determines whether a grid is visible on the screen. Check the box to make the grid visible. If the box is not checked, the grid will not be visible.

Width

Determines the distance between vertical grid lines.

Height

Determines the distance between horizontal grid lines.

Graphics Options

Max Frames Per Second

Determines the maximum number of times the RTMC screens will be updated per second.

Disable Animation

Determines whether animation is disabled when a data value changes. This disables the smooth transition between values on gauges, status bars, etc. When the data value changes, the component will jump to the new value. Disabling animation greatly enhances performance when dealing with fast data or large, complex projects.

Check the box to disable animation. If the box is not checked, animation will be enabled.

Edit | Customize

Opens the Customize (p. 134) dialog box which allows you to customize RTMC's toolbars and menus.

Customize

This dialog box is opened from the Edit | Customize menu item or by right-clicking within the toolbar menu bar and selecting Customize. It can be used to customize RTMC's toolbars and menus.

Commands

The Commands tab allows you to drag and drop any command available from RTMC's menus onto a toolbar. As a menu name is highlighted in the **Categories** column, its menu items will be shown in the **Commands** column. Any command can then be added to the Standard or Layout Toolbar by clicking on the command name and dragging it to the desired position on the toolbar.

With the Customize dialog box open, buttons can be removed from a toolbar by dragging and dropping them away from any toolbar. Buttons can be moved from one toolbar to another by dragging and dropping them to the desired position.

Toolbars

The Toolbars tab allows you to show or hide the toolbars, reset toolbar buttons to their default configuration, and show text labels on the toolbars. Clear the box next to the Layout Toolbar, Standard Toolbar, or Toolbox to hide the toolbar. (Note that the Menu Bar cannot be hidden.) Press the **Reset** button to reset only the highlighted toolbar. Press the **Reset All** button to reset the Layout Toolbar, Menu Bar, Standard Toolbar, and ToolBox to their default button configurations. Select the box next to Show text Labels to add text labels to each of the buttons on the highlighted toolbar.

Keyboard

The Keyboard tab allows you to create new keyboard shortcuts for any commands available from RTMC's menus. As each menu name is chosen under **Category**, its menu items will be shown under **Commands**. When a command is highlighted in the **Commands** column, its current keyboard shortcut(s) will be shown in **Current Keys**. To add a new keyboard shortcut, type the key sequence in the **Press New Shortcut Key** field. Then press the **Assign** button to assign this shortcut to the currently highlighted command. (Note that if the key sequence entered is already assigned, this will be indicated in the **Assigned to** field. You will need to remove this assignment before assigning the key sequence to a different command.) To remove any current keyboard shortcuts, highlight the shortcut under Current Keys and then press the **Remove** button. Pressing the **Reset All** button will reset all keyboard shortcuts to their default values.

> **NOTE:** There is only one Accelerator available in RTMC. Therefore, Default is the only available option in the **Set Accelerator for** field.

Menu

The Menu tab allows you to configure the appearance of an open menu.

The **Menu animations** drop-down box determines how a menu is opened when a menu name is pressed. Choose *None* for no animation, *Unfold* to have the menu unfold down and to the right, *Slide* to have the menu slide down, *Fade* to have the menu fade in, or *Default* to use the default animation.

The **Menu shadows** check box determines whether shadows are shown around an open menu. Select the check box to display shadows. Clear the check box to disable shadows.

> **NOTE:** Only one menu configuration is available for RTMC. Therefore, Default Menu is the only option available under **Show Menus** for.

> **NOTE:** No user-configurable context menus are available in RTMC. Therefore, the right half of the Menu tab is always disabled.

Options

The first two check boxes determine whether screen tips are shown when you hover over icons on RTMC's toolbars and whether shortcut keys are included in the screen tips. Select the **Show Screen Tips on toolbars** check box to show screen tips. Clear the check box to disable screen tips. If screen tips are being displayed, select the **Show shortcut keys in Screen Tips** check box to include keyboard shortcuts in the screen tips. Select the **Large Icons** check box to increase the size of the toolbar icons.

Right-Click Menu

Right-clicking on one of RTMC's menus or buttons with the Edit | Customize dialog box open, brings up a menu with the following options:

Reset to Default - Resets a button to its default configuration.

Copy Button Image - Copies a button image to the Windows clipboard.

Delete - Deletes the menu or button. (A menu or button that has been deleted can be restored by resetting the menu bar or the appropriate toolbar from the Toolbars tab of the Edit | Customize dialog box.)

Button Appearance - Opens a dialog box from which you can choose to display an **Image**, **Text**, or **Image and Text**. Note that you can specify the text to be displayed, but RTMC does not allow you to select a user-defined image. Also note that for a menu, Text Only is the only option available. Therefore, this dialog box only allows you to change the text that is displayed.

Image - Designates that a button will display only an image.

Text - Designates that a button will display only text. The text that is displayed can be changed from the Button Appearance menu item.

Image and Text - Designates that a button will display an image and text. The text that is displayed can be changed from the **Button Appearance** menu item.

Start Group - Inserts a divider bar before the menu or button.

6.2.5.3 View Menu

All of the View menu items are toggles. When a check mark appears to the left of the menu item, it is enabled. When the check mark is absent, the option is disabled. These options are toggles—if an option is off (unchecked), select it once to turn it on (checked) and vice versa.

View | Full Screen Mode

When selected, the RTMC Workspace expands to fill the entire computer screen. This provides more space to work with in designing your RTMC project. In this mode, you must use the right-click menus to add components and perform other functions available from RTMC's toolbars.

Press the Esc key to exit this mode.

View | Show Project Tree

When selected, the Project Tree is displayed on the left side of the RTMC. The Project Tree can also be repositioned on the screen by grabbing its upper margin and dragging it to a new location or by pressing the Maximize button at the top right of its margin. By default, the Project Tree display is enabled.

View | Show Toolbox

When selected, the Component Toolbox is displayed at the top of the RTMC screen. The Component Toolbox can also be repositioned on the screen by grabbing its left margin and dragging it to a new location. By default, the Component Toolbox display is enabled.

View | Show Layout Toolbar

When selected, the Layout Toolbar is displayed at the top of the RTMC screen. By default, the Layout Toolbar display is enabled.

The Layout Toolbar gives quick access to the Align, Space Evenly, Make Same Size, Center, and Order menu items on RTMC's Component menu. Icons on the Layout Toolbar will only be enabled when their functions are currently applicable based on whether no components are currently selected, one component is currently selected, or multiple components are currently selected.

View | Show Tabs

When selected, tabs are displayed at the top of the RTMC workspace to allow the user to switch between screens. By default, tabs are shown when RTMC is first opened. When tabs are not shown, you can switch between screens by selecting a screen from the Project Tree.

View | Show Standard Toolbar

When selected, the Standard Toolbar is displayed at the top of the RTMC screen. By default, the Standard Toolbar display is enabled.

View | Show Status Bar

When selected, the Status bar is displayed at the bottom of the Workspace. The workspace size is shown on the Status bar, along with a hint line and the name of the server to which RTMC is connected. By default, the Status Bar display is enabled.

View | Show Grid

When selected, a grid is displayed for the workspace background, to help align components on the screen. By default, the grid is not enabled when RTMC is first opened.

6.2.5.4 Project Menu

Project Menu options work with the whole project or workspace.

Project | Configure Workspace

The workspace size, in pixels, is listed at the bottom of the Workspace on the hint line. The Workspace Configuration dialog box can be brought up by double-clicking on the workspace size or choosing the Project | Configure Workspace menu item. From this Workspace Configuration dialog box, size options for both the development Workspace and the Run-time screen can be set:

Development Sizing Options

Size - Choose a size from the drop-down list for the development Workspace. If you choose Custom, you will be asked to specify the Width and Height in Millimeters, Inches, or Pixels

Orientation - Select Portrait or Landscape orientation.

Run-time Options

Auto Size - The size of an RTMC screen is set automatically by dragging the boundaries of the RTMC screen to a new size. The components will resize to match the new screen size. The aspect ratios of the components will be maintained, only if **Lock Aspect Ratio** is selected.

Fixed Size - The screen size is fixed. It is determined by the size set in the development mode.

Workspace Configuration

The workspace size, in pixels, is listed at the bottom of the Workspace on the hint line. The Workspace Configuration dialog box can be brought up by double-clicking on the workspace size or choosing the Project | Configure Workspace menu item. From this Workspace Configuration dialog box, size options for both the development Workspace and the RTMC Run-time screen can be set:

Development Sizing Options

Size

Choose a size from the drop-down list for the development Workspace. If you choose Custom, you will be asked to specify the Width and Height in Millimeters, Inches, or Pixels.

Orientation

Select Portrait or Landscape orientation.

Run-time Options

Auto Size

The size of an RTMC screen is set automatically by dragging the boundaries of the RTMC screen to a new size. The components will resize to match the new screen size.

Lock Aspect Ratio

If Lock Aspect Ratio is selected, the aspect ratios of the components will be maintained, if they are resized to match a new screen size.

Fixed Size

The screen size is fixed. It is determined by the size set in development mode.

Project | Change Server Connection

Allows you to specify the computer to which RTMC should connect when the project is run. The resulting dialog box has the following fields:

> **Source Name** - Designates the name that will be used for the LoggerNet server in the Data Source Browser and on the status bar.

Server Address - This is the hostname or TCP/IP address of the computer running LoggerNet (or PC400) software, to which you are trying to connect This must be the valid name of an existing computer or a TCP/IP address in the form XXX.XXX.XXX.XXX for an IPv4 address or [XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX] for an IPv6 address. If the software server resides on the same computer as RTMC, you can simply type in LocalHost for the server address. By default, LoggerNet's port number is 6789. If this default port number is used, it does not need to be specified in RTMC. Otherwise, it is specified after server address, separated by a colon. An example would be 192.168.4.32:3000, where 3000 is LoggerNet's port number.

Username - Your username on the software server.

Password - Your password for the software server.

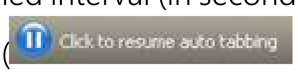
The **Username** and **Password** fields are required only if your server administrator has set up security on your system.

Each time you start RTMC, you will be prompted to enter this information. However, you can save the login information by selecting the **Remember username and password** check box, or you can select the **Automatically login to this server** check box to skip this window and use the information from the last session.

> **NOTE:** The option to change server connection is not available when running RTMC from RTDAQ. The server connection is always to LocalHost.

> **NOTE:** If LoggerNet Security is enabled and your project will be published to the Web using CSI Web Server, you must select the Remember username and password checkbox. You will not be prompted for the LoggerNet Security password when viewing your project from the web. Therefore, if the username and password have not been saved with the project, you will not be able to perform any functions that require the username and password.

Project | Configure Auto Tabbing

When multiple screens have been set up in RTMC, this option will enable automatic switching from tab to tab in run-time mode, based on a defined interval (in seconds). If a screen is interacted with, a button appears below the menu (

) that can be used to resume tabbing.

Project | Add New Screen

Adds a new display screen to the open project.

Project | Change Screen Order

Allows you to change the order in which the tabbed screens appear in the Project Tree and on the tabs in development mode and on the tabs in run-time mode.

6.2.5.5 Screen Menu

Screen Menu options work with the tabbed screens in the project. The Screen Menu is also available by right clicking any blank area of the workspace.

Screen | Screen Properties

Allows you to change the background image and color for the active screen.

Screen Properties

Background Color

Displays a list of colors from which to select a color to be used for the screen background.

Background Image

Show Image

Determines whether a background image is visible for the screen. Check the box to make the background image visible. If the box is not checked, the background image will not be visible.

Image

Lists the name of the image that will be used for the background. The image is selected by pressing the button to the right of the field or by typing in a new file name directly. Valid image types are GIF, BMP, JPG, PNG, and TIF.

RTMC comes with a library of images. If you have created custom images that you want to use, copy them into C:\Campbellsci\Lib\RTMCMediaLib directory.

Screen | Delete Screen

Removes the active screen from the open project.

Screen | Rename Screen

Highlights the active screen name and prepares it for editing.

Screen | Duplicate Screen

Creates a duplicate of the active screen. Once the duplicate screen is created, it can be modified as needed. This allows you to easily create multiple similar screens, for example, screens that display the same information for different stations.

Screen | Paste

Places a copy of the Windows clipboard content on the active screen. (Note that only valid XML code can be pasted into RTMC.)

Screen | Insert New

Provides a drop-down list of components that can be added to the screen.

6.2.5.6 Component Menu

The Component Menu is used to set the component properties, placement and alignment. The Component Menu is also available by right clicking any of the components in the workspace. Many of the Component Menu's items (Align, Space Evenly, Make Same Width, Center, and Order) are also available from the Layout Toolbar.

Component | Properties

Opens the Properties dialog box for the selected component.

Component | Delete Selection

Removes the highlighted component from the Workspace.

Component | Lock Aspect Ratio

The aspect ratio of a component is the ratio of its width to its height. When this menu item is selected, the aspect ratio of the highlighted component is locked. If the width (or height) of the component is changed, the height (or width) will automatically be changed to keep the aspect ratio constant. The lock on the aspect ratio can be temporarily overridden by holding down the Ctrl key, while dragging the boundaries of a component to resize it.

When enabled, a check mark appears beside this item in the Component menu. Select the menu item to toggle the check mark off or on. By default, Lock Aspect Ratio is off for all components.

Component | Rename Component

Highlights the selected component name and prepares it for editing.

Component | Manual Resize

Allows the user to set the size and position of the selected component. Use the **X** and **Y** spin boxes to set the position of the component on the screen. Use the **W** and **H** spin boxes to set the width and height of the component. Use the **Angle** spin box to set the rotation of the component.

Component | Cut

Removes the selected component from the screen and places a copy on the Windows clipboard.

Component | Copy

Places a copy of the selected component on the Windows clipboard.

Component | Paste

Places a copy of the Windows clipboard content on the active screen. (Note that only valid XML code can be pasted into RTMC.)

Component | Align

Allows you to align two or more components on a screen. Select each component by holding the **Ctrl** key and clicking the components with the primary mouse button, and then choose one of the alignment options from the menu. (Alternately, you may select components by clicking on their names in the Project Tree with the **Ctrl** key held down or by using the mouse to click and drag a box around the desired components.) The last component selected will be the basis on which the other components are aligned. The last component selected will be identified by the dark blue handles and by dark blue highlighting in the Project Tree. The other selected components have handles with blue outlines and are highlighted in light blue in the Project Tree.

Lefts - The left border of all selected components will be aligned.

Centers - All selected components are positioned vertically, with their centers aligned.

Rights - The right border of all selected components will be aligned.

Tops - The upper border of all selected components will be aligned.

Bottoms - The bottom border of all selected components will be aligned.

Middles - All selected components are positioned horizontally, with their centers aligned.

Note that these functions are also available from the Layout Toolbar.

Component | Space Evenly

Used to adjust the spacing of three or more components. Select each component by holding the **Ctrl** key and clicking the components with the primary mouse button, and then choose one of the spacing options from the menu. (Alternately, you may select components by clicking on their names in the Project Tree with the **Ctrl** key held down or by using the mouse to click and drag a box around the desired components.) **Across** will space the components horizontally across the page using the left-most and right-most components as the left and right boundaries for the spacing. **Down** will space the components vertically down the page, using the top-most and the bottom-most components as the top and bottom boundaries for the spacing.

Note that these functions are also available from the Layout Toolbar.

Component | Make Same Size

Allows you to set two or more components to the same height, width, or both. Select each component by holding the **Ctrl** key and clicking the component with the primary mouse button, and then choose one of the options from the menu. (Alternately, you may select components by clicking on their names in the Project Tree with the **Ctrl** key held down or by using the mouse to click and drag a box around the desired components.) The last component selected will be the basis on which the other components are sized. The last component selected will be identified by the dark blue handles and by dark blue highlighting in the Project Tree. The other selected components have handles with blue outlines and are highlighted in light blue in the Project Tree.

Note that these functions are also available from the Layout Toolbar.

Component | Center

Used to center one or more components on the page. **Vertical** will center the component(s) vertically on the page. **Horizontal** will center the component(s) horizontally on the page.

Note that these functions are also available from the Layout Toolbar.

Component | Order

Allows you to choose the order in which the components appear on the display (often referred to as the Z order). Select a component and choose to send it to the back, send backward (one step), bring forward (one step), or bring it to the front of all other components. This order is important when layering transparent components.

Note that these functions are also available from the Layout Toolbar.

Component | Group Selection

Allows you to group components together. They can then be moved, copied, ordered, etc. as a single object. Select the components to be grouped by holding the **Ctrl** key and clicking the components with the primary mouse button. Then choose the Group Selection item from the Component menu or the Component right-click menu. You must have at least two components selected for this menu item to be enabled.

When a component group is selected, the Ungroup Selection menu item will be enabled. You can undo the component grouping by selecting this menu item.

When components are grouped, the properties for each of the components will show up in the submenu displayed when Subcomponent Properties is selected from the Component menu or the Component right-click menu. These menu items can be used to modify the properties for each component individually.

6.2.5.7 Window Menu

If there are multiple screens in the project, the Window menu will allow you to change between the screens using the menu.

6.2.5.8 Help Menu

Help | RTMC Development Help

This menu item provides access to help for all of the features of RTMC.

Help | Keyboard Map

This menu item opens the Keyboard Map (p. 145) dialog box from which you can print RTMC's keyboard shortcuts or copy them to the Windows clipboard.

Keyboard Map

This dialog box is opened from the Help | Keyboard Map menu item. It allows you to print RTMC's keyboard shortcuts or copy them to the Windows clipboard.

Use the Category drop-down list to select one of RTMC's menus or All Commands. The menu items for that category will be displayed with their keyboard shortcut(s). If a description is available for a command that will also be displayed. You can print the current category by pressing the print button (). You can copy a single command to the Windows clipboard by selecting the command and then pressing the copy button ().

> **NOTE:** There is only one Accelerator available in RTMC. Therefore, Default is the only available option in the **Show Accelerator** for field.

> **NOTE:** Keyboard shortcuts cannot be changed from this dialog box. The Customize (p. 134) dialog box must be used to make changes.

6.2.6 Expressions

Components that display data values either numerically or graphically can be processed using expressions. These expressions can include simple mathematical expressions, functions to manipulate strings, or more complex functions that deal with the state of a data value over time.

For instance, a temperature reading in degrees Celsius can be processed to display in degrees Fahrenheit using a simple mathematical expression. This is done by first selecting the data value in the **Select Data** field, and then entering the mathematical expression after the defined data value. Using the above example, if the data value is defined as "Server:CR5000.TempData.Temp1" ("*Source*:*datalogger*.*table*.*variable*"), you would enter

    "Server:CR5000.TempData.Temp1" * 1.8 + 32

to convert the temperature reading from degrees Celsius to degrees Fahrenheit.

Strings

As shown above, double quotes are used in RTMC to enclose the name of a data value (or source, datalogger, or table depending on the component). Therefore, when defining a literal string, a dollar sign is used as a prefix. This indicates to RTMC that you are defining a literal string rather than a data value. For example, to search for the position of the sequence abc in the data value mystring, you would use the following expression:

    InStr(1, "Server:CR1000.hourly.mystring", $"abc")

Statistical Functions and Start Options

Expressions can also use Statistical Functions, some of which involve the state of a data value over a period of time. For instance, you can return the maximum value of a data value over the past 24 hours using the expression:

    MaxRunOverTime("Server:CR1000.QtrHour.Temp",Timestamp
    ("Server:CR1000.QtrHour.Temp"),nsecPerDay)

When RTMC-RT is launched it begins processing with the newest record by default. Therefore, using the above expression, a component will not immediately display the maximum value over the past 24 hours. Rather, it will display the maximum value since RTMC-RT was launched. The 24-hour maximum will only be displayed after it has been running for 24 hours. In order to get a 24 hour maximum immediately, you can use a "Start Option Function" to cause RTMC to begin processing data at an earlier point. For example,

StartRelativeToNewest(nsecPerDay,ordercollected); MaxRunOverTime ("Server:CR1000.QtrHour.Temp",Timestamp("Server:CR1000.QtrHour.Temp"),nsecPerDay)

would begin displaying a 24 hour maximum immediately, provided that the data is available in the communications server's data cache.

Aliases

If a data value is used multiple times in an expression, the expression can be simplified by declaring an alias for the data value at the first of the expression, in the form:

Alias(alias_name, data_value)

For example,

StartAtOffsetFromNewest(5,OrderCollected);IIF(ABS(("Server:CR1000.MyTable.Value"-ValueAtTime("Server:CR1000.MyTable.Value",TimeStamp ("Server:CR1000.MyTable.Value"),30*nsecPerSec,0))>10 AND ABS(ValueAtTime ("Server:CR1000.MyTable.Value",TimeStamp("Server:CR1000.MyTable.Value"),30*nsecPerSec,0)-ValueAtTime("Server:CR1000.MyTable.Value",TimeStamp ("Server:CR1000.MyTable.Value"),60*nsecPerSec,0)))>10,1,0)

can be replaced by:

Alias(X,"Server:CR1000.MyTable.Value");StartAtOffsetFromNewest(5,OrderCollected);IIF((ABS(X-ValueAtTime(X,TimeStamp(X),30*nsecPerSec,0))>10 AND ABS(ValueAtTime(X,TimeStamp (X),30*nsecPerSec,0)-ValueAtTime(X,TimeStamp(X),60*nsecPerSec,0)))>10,1,0)

Synchronizing Variables

The ValueSynch function can be used to synchronize data values coming from multiple data sources so that you can display the results of a calculation on those data values in a single component. The Value Synch function takes the form:

ValueSynch(synchronized_name, data_value)

Where synchronized_name is the name of a new variable that will be used in a calculation at the end of the expression and data_value is the name used within RTMC to access the data value, i.e., *Source*:*datalogger.table.variable.*

For example, if you wish to display the average air temperature of two stations on a chart, the following expression can be used to synchronize the timestamps of the stations and then calculate the average air temperature:

ValueSynch(air_temp_1,"Server:CR1000_1.SECOND.air_temp");ValueSynch(air_temp_2,"Server:CR1000_2.SECOND.air_temp"); (air_temp_1 + air_temp_2) / 2

NOTE: Timestamps are truncated to seconds prior to synchronization. Therefore, synchronizing sub-second data is not recommended as the results will be unpredictable.

NOTE: If the timestamps of the stations are not the same (for example, if one datalogger is a few minutes behind the other), the component will display the exclamation point indicating no data, until the data sources have common timestamps and, therefore, can be synchronized.

NOTE: RTMC will buffer up to 100,000 points of a data value while waiting for a common timestamp from the other datalogger(s). Once the buffer reaches 100,000 data points the oldest data value will be removed from the buffer, each time a new data value is collected.

All of the functions available in RTMC are described below. For details on a function, see the online help.

NOTE: Spaces must be used to delimit the predefined constants and functions. Operators allow but do not require spaces.

NOTE: An expression can include data values from multiple dataloggers.

NOTE: An expression must include a data value from a LoggerNet server.

6.2.6.1 Operators

	Description
( )	Prioritizes an expression
*	Multiply by
/	Divide by
^	Raised to the power of
+	Add
-	Subtract/Unary negation
=	Equal
< >	Not equal

>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

6.2.6.2 Order of Precedence

- Anything inside parentheses ( )
- Exponentiation ^
- Negation (unary) -
- Multiplication *, division /
- Modulo (remainder) MOD
- Addition +, subtraction -

When consecutive operators have the same priority, the expression evaluates from left to right. This means that an expression such as **a-b-c** is evaluated as **(a-b)-c**.

6.2.6.3 Predefined Constants

The following constants are defined for convenience within numeric expressions:

Constant	Description
e	2.718282
PI	3.141593
True	-1
False	0
NOPLOT	NAN
NAN	NAN (not a number)
INF	INF (non-finite number)

6.2.6.4 Predefined Time Constants

These predefined time constants can be useful as a parameter for the Statistical Functions, where the interval parameter must be specified in nanoseconds.

Constant	Description
nsecPerUSec	Number of nanoseconds in a microsecond
nsecPerMSec	Number of nanoseconds in a millisecond
nsecPerSec	Number of nanoseconds in a second

nsecPerMin	Number of nanoseconds in a minute
nsecPerHour	Number of nanoseconds in an hour
nsecPerDay	Number of nanoseconds in a day
nsecPerWeek	Number of nanoseconds in a week

6.2.6.5 Predefined Reset Options

These predefined reset options are used as a parameter for the Statistical Functions with a reset parameter.

Constant	Description
RESET_HOURLY	Reset whenever there is a change in the hour in the value's timestamp
RESET_DAILY	Reset whenever there is a change in the day in the value's timestamp
RESET_WEEKLY	Reset whenever the day of the week is less than the day of the week of the newest timestamp stored (Sunday marks the beginning of the week) or when the difference between the current timestamp and the newest timestamp stored exceeds seven days.
RESET_YEARLY	Reset whenever there is a change in the year in the value's timestamp
RESET_CUSTOM	Reset whenever the doReset parameter is set to a non-zero value.

6.2.6.6 Math Functions

See the RTMC online help for examples of using each of these functions.

Function	Description
ABS	Returns the absolute value of a number.
ACOS	Returns the arc cosine of a number.
ASIN	Returns the arc sine of a number.
ATN	Returns the arc tangent of a number.
ATN2(y,x)	Returns the arctangent of y/x.
CEILING	Rounds a number up to an integer value.
COS	Returns the cosine of a number.
COSH	Returns the hyperbolic cosine of a number.
CSGN	Changes the sign of a number by multiplying by -1.0.
EXP	Returns e raised to a power.
FIX	Returns the integer portion of a number. If the number is a negative, the first negative

	integer greater than or equal to the number is returned.
FLOOR	Rounds a number down to an integer value.
FRAC	Returns the fraction part of a number.
FormatFloat	Converts a floating point value into a string.
FormatFloatL	Converts a floating point value into a string and applies any rules associated with the locale of the computer running RTMC.
INT	Returns the integer portion of a number. If the number is a negative, the first negative integer less than or equal to the number is returned.
IsFinite	Determines if a value is finite.
LN	Returns the natural log of a number. (Note that LN or LOG may be used to perform the same function.)
LOG	Returns the natural log of a number. (Note that LN or LOG may be used to perform the same function.)
LOG10	Returns the logarithm base 10 of a number.
MOD	Performs a modulo divide of two numbers.
PWR	Raises a constant to a specified exponent.
RND	Generates a random number.
ROUND	Rounds a number to a higher or lower number.
SGN	Used to find the sign value of a number (-1, 0, or 1).
SIN	Returns the sine of an angle.
SINH	Returns the hyperbolic sine of a number.
SQR	Returns the square root of a number.
TAN	Returns the tangent of an angle.
TANH	Returns the hyperbolic tangent of a number.

6.2.6.7 Logical Functions

See the RTMC online help for examples of using each of these functions.

Function	Description
AND	Performs a logical conjunction on two numbers.

EQV	Performs a logical equivalence on two numbers.
IIF(x,y,z)	Evaluates an expression and returns one value if true, a different value if false.
IMP	Performs a logical implication on two numbers.
NOT	Performs a logical negation on a number.
OR	Performs a logical disjunction on two numbers.
SelectSwitch	Iterates through the set of predicates and values in the order in which these are specified in its arguments list. It will return the value associated with the first predicate that specifies a non-zero integer value. If no asserting predicate can be found, the function will return the default_value
XOR	Performs a logical exclusion on two numbers.

6.2.6.8 String Functions

See the RTMC online help for examples of using each of these functions.

Function	Description
Hex	Returns a hexadecimal string representation of an expression.
HexToDec	Converts a hexadecimal string to a float or integer.
InStr	Finds the location of a string within a string.
InStrRev	Finds the location of a string within a string. (Differs from InStr in that it searches from the end of the string rather than from the start of the string.)
Left	Returns a substring that is a defined number of characters from the left side of the original string.
Len	Returns the number of bytes in a string.
LTrim	Returns a copy of a string with no leading spaces.
Mid	Returns a substring that is within a string.
Replace	Used to search a string for a substring, and replace that substring with a different string.
Right	Returns a substring that is a defined number of characters from the right side of the original string.
RTrim	Returns a copy of a string with no trailing spaces.
Space	Returns a string value that is filled with a defined number of spaces
StrComp	Compares two strings by comparing the characters in one

	string to the characters in another.
StrReverse	Returns a copy of a string with the characters in reverse order.
Trim	Returns a copy of a string with no leading or trailing spaces.

6.2.6.9 Conversion Functions

See the RTMC online help for examples of using each of these functions.

Function	Description
ToDate	Converts a value to a date.
ToFloat	Converts a value to a floating point number.
ToInt	Converts a value to an integer.

6.2.6.10 Time Functions

See the RTMC online help for examples of using each of these functions.

Function	Description
FormatTime	Produces a string that formats a timestamp in the manner specified.
SystemTime	Returns the current computer time.
SystemTimeGMT	Returns the current GMT (Greenwich Mean Time) system time.
Timestamp	Returns the timestamp associated with the record from which a value is derived.
SetTimestamp	Returns the value specified and sets its timestamp to the timestamp specified.

6.2.6.11 Start Option Functions

See the RTMC online help for examples of using each of these functions.

> **NOTE:** The majority of Start Option Functions are most useful when used with Statistical Functions. This is because when RTMC-RT is launched it begins processing with the newest record by default. Therefore, a Start Option Function must be used to cause RTMC to begin processing data at an earlier point that provides the history required by some Statistical Functions. For example using MaxRunOverTime to display the maximum value over the past 24 hours will initially display the maximum value since RTMC-RT was launched. The 24-hour maximum will only be displayed after it has been running for 24 hours. In order to get a 24-hour maximum immediately, a Start Option Function must be used.

NOTE: StartAfterNewest and StartAtNewest can be useful with trend components (Time Series Chart, XY Chart, Scope, Table Display, Wind Rose) if you wish to prevent RTMC from backfilling a component with historical data.

Function	Description
StartAfterNewest	No records are processed until a new record has been collected.
StartAtNewest	Attempts to start processing at the newest record in the table.
StartAtOffsetFromNewest	Attempts to start processing with the record at the specified offset back from the newest record in the table. For example, an offset of one starts processing at the newest record, an offset of two starts processing one record back from the newest record, and so forth.
StartAtRecord	Attempts to start processing at the specified file mark and record number. If the specified record cannot be located, it starts processing at the oldest record since the specified record. If the specified record is beyond what is available, it starts processing at the oldest available record.
StartAtTime	Attempts to start processing at the specified timestamp. If the specified timestamp cannot be located, it starts processing at the oldest record since the specified timestamp. If the specified timestamp is beyond what is available, it starts processing at the first record that is subsequently written to the data table (i.e., you will have no backfill).
StartRelativeToNewest	Attempts to start processing with the first record whose timestamp is greater than or equal to the newest record's timestamp minus the specified interval.

6.2.6.12 Statistical Functions

See the RTMC online help for examples of using each of these functions.

NOTE: Start Option Functions can be useful when using Statistical Functions. This is because when RTMC-RT is launched it begins processing with the newest record by default. A Start Option Function can be used to cause RTMC to begin processing data at an earlier point that provides the history required by some Statistical Functions. For example using MaxRunOverTime to display the maximum value over the past 24 hours will initially display the maximum value since RTMC-RT was launched. The 24-hour maximum will only be displayed after it has been running for 24 hours. In order to get a 24 hour maximum immediately, a Start Option Function must be used.

Function	Description
AvgRun	Returns a running average of up to the last specified number of values.
AvgRunOverTime	Returns the running average of the specified value over time.
AvgRunOverTimeWithReset	Returns the running average of the specified value since the function was reset.
AvgSpa	Returns the average of the specified values.
Last	Stores the specified value and returns the previous value.
MaxRun	Returns the maximum of all values that it has considered.
MaxRunOverTime	Returns the maximum of all values whose timestamps are greater than the newest timestamp minus the specified interval.
MaxRunOverTimeWithReset	Returns the maximum of all values since the function was reset.
MaxSpa	Returns the maximum of the specified values.
MedianRun	Returns the median value of up to the last specified number of values.
MedianRunOverTime	Returns the median value in the set of values whose timestamps are greater than the newest timestamp minus the specified interval.
MinRun	Returns the minimum of all values that it has considered.
MinRunOverTime	Returns the minimum of all values whose timestamps are greater than the newest timestamp minus the specified interval.
MinRunOverTimeWithReset	Returns the minimum of all values since the function was reset.
MinSpa	Returns the minimum of the specified values.
StdDev	Returns the standard deviation of up to the last specified number of values.
StdDevOverTime	Returns the standard deviation of the specified value over time.
StdDevOverTimeWithReset	Returns the standard deviation of the specified value since the function was reset.
Total	Returns the total of all values that it has considered.
TotalOverTime	Returns the total of all values whose timestamps are greater than the newest timestamp minus the specified interval.
TotalOverTimeWithReset	Returns the total of all values since the function was reset.
ValueAtTime	Returns the oldest value in a set of values from a specified time

interval.

6.2.7 Troubleshooting

Red Exclamation Point

An exclamation point in a red circle at the upper right of a component indicates a problem with the linked data value. Hover your mouse pointer over the box for a few seconds to display a hint about the error RTMC has detected.

Blue Exclamation Point

A blue exclamation point on a graph indicates that more than 100,000 data points were requested. In this case, RTMC graphs just the first 100,000 data points and then stops. This issue can be fixed by decreasing the display and/or scrolling range for the graph or by using data from a slower data table.

Yelllow N

A yellow N in a red circle at the upper right of a component indicates the value of the field referenced in the Select Data field of the component is currently a NaN.

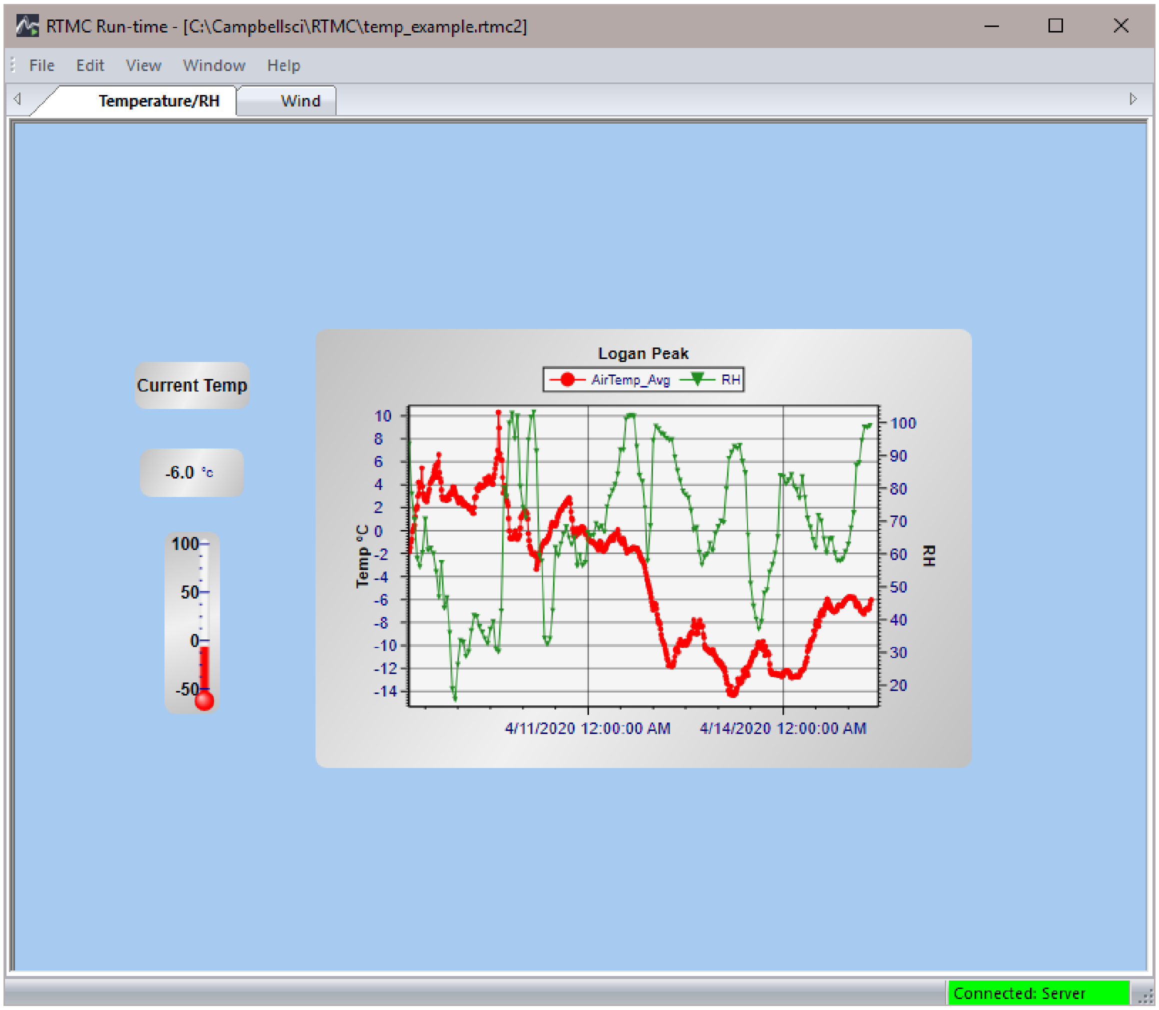
Non-English Characters

If you are having trouble displaying non-English characters, the issue may be resolved by correcting your Windows Region and Language settings. Use the following steps:

1. Open the Windows Control Panel.
2. Select Region and Language.
3. On the Format tab, select the Format for your location.
4. On the Location tab, select your location.
5. On the Administrative tab, press Change system locale. In the resulting pop-up window, select your current system locale.

6.2.8 RTMC Run-time

RTMC Run-time allows you to run the real-time graphic display screen that was created in RTMC. In LoggerNet, you can test the operation of the display screen from the RTMC window by clicking the Save and Run button ( ) on the toolbar. This will start the project window with RTMC Run-time as shown in the window below.



In RTDAQ, you must launch RTMC Run-time from the RTDAQ toolbar, and then open the project you have developed in RTMC. When the run-time display screen is started, the display components will have an exclamation point in a red circle at the upper right until data is received from LoggerNet or RTDAQ. In LoggerNet, if data is not displayed, check to see that the data is being collected. This can be done in the LoggerNet Setup screen. Click on the appropriate station and then choose the **Schedule** tab. Also check the **Data Files** tab to verify the desired table is enabled for scheduled collection. In RTDAQ, check to see that RTDAQ is connected to the datalogger.

Once a project file has been created, the display screen can be run without starting RTMC. From the Windows Start Menu select **RTMC Run-time**. In the run-time window select **File | Open Project** to select the RTMC project screen to run. If **Remember Username and Password** was not selected in RTMC, it will be necessary to enter them each time the project is run in RTMC Run-time.

In run-time mode, you can print an image of the RTMC display screen by selecting **File | Print Screen**. A new form to be run is selected under **File | Open Project**.

A copy of RTMC Run-time comes with LoggerNet and RTDAQ. When RTMC Pro is installed on a computer, RTMC Run-time will be updated to match the RTMC Pro version. For LoggerNet, if you want to run RTMC Pro projects on remote computers, additional copies of RTMC Run-time can be purchased separately. One copy is required for each computer on which RTMC Run-time will be used. As noted above, when running RTMC Run-time on a remote computer, the host computer must have Remote Connections enabled (LoggerNet Toolbar, **Tools | Options | Allow Remote Connections**).

7. Network Status and Resolving Communication Problems

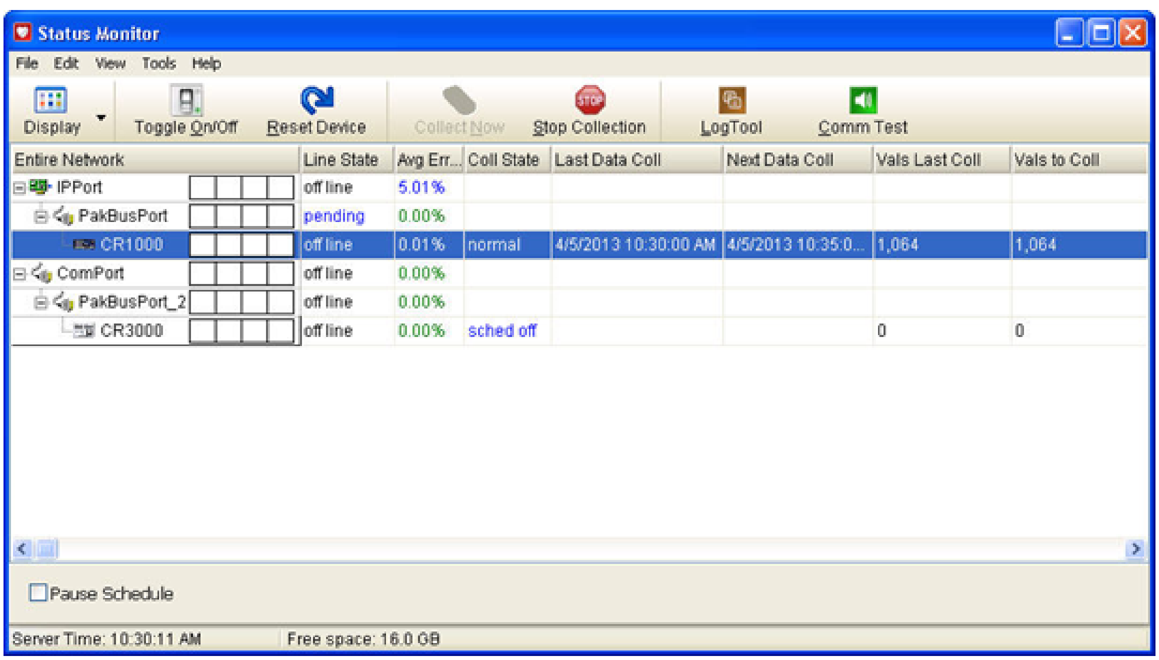
LoggerNet provides several tools for monitoring the status of a datalogger network and troubleshooting communication problems within that network.

The Status Monitor screen provides a way to monitor communications statistics. Statistics are displayed for data collection attempts and communication failures. PakBus Graph provides a visual representation of the devices in a PakBus network and lets you edit PakBus device setting. The Log Tools utility provides a way to read communication logs more easily. The Troubleshooter highlights potential problems in a communication network and provides access to a Communications Test and other troubleshooting tools. The LoggerNet Server Monitor is used to monitor the communication log for a remote instance of LoggerNet or when LoggerNet is being run as a service.

Note that a Troubleshooting Guide (p. 382) is also provided.

7.1 Status Monitor

The Status Monitor window provides information on the communication link and most recent data collection attempts with the dataloggers in the network.



The datalogger network is depicted on the left side of the Status window. The communication status of each device can quickly be determined by the color of the icon associated with it. A green N indicates that communication is Normal; blue M indicates that communication is Marginal; red C indicates that communication is Critical; and gray U indicates that the communication state is unknown.

The graphical bar to the right of the device name () is the communication history. It provides a quick summary of communication with that device over the past 24 hours (it is divided into four 6 hour periods). A green bar means there have been no retries and no failures for the period. A blue bar indicates there have been retries but no failures. A red bar indicates there have been failures. If a bar is gray, there have been no attempts to contact the device. The height of the bar (as a percentage of the height of the box) is determined by the following equation:

$$Height = 25 + 75 * \frac{(failures + retries)}{attempts}$$

where

- failures = number of communication failures
- retries = number of communication retries
- attempts = total number of communication attempts

The Communication History can be displayed in a larger view by right clicking within the history for a device and selecting **Comm History** from the short cut menu. Each segment of time in this larger view represents 10 minute intervals. Clicking on a segment will display the time period for the interval and the number of Attempts (A), Retries (R), and Failures (F) during that period. This information is provided by a pop-up above the history and by a line below the Communication History. (The pop-up can be turned off by clearing the Show Annotations check box.) The right and left arrow keys can be used to move right or left one segment at a time. You can zoom in on a particular interval by clicking and dragging your mouse cursor from the upper left of an area to the lower right of an area. (To return to normal view, click and drag from the right to the left.)

Information on the Status window can be sorted by column. Click on a column heading to sort the devices. For instance, if the Network Map column is clicked once, the devices will be sorted alphabetically by name. Clicking the column a second time will sort the devices in reverse alphabetical order, and a third time will restore the Network map to its original order.

Sorting on a column in the Status window can be an effective means to locate trouble spots, particularly with a large network. For example, sorting on the Avg Err% column can bring all those stations with high average error rates to the top, in order from highest average error rate to lowest average error rate. Sorting on Collection State can group the stations according to secondary retry mode, primary retry mode, normal collection schedule, and schedule off. Sorting on the Holes statistic can indicate the dataloggers having collection difficulties. Once you have located a station with difficulties, you can right-click on the station and select TroubleShooter (p. 182) to launch the diagnostic tool for the station.

For further explanation of the statistics available for display by the Status Monitor, see Select Columns (p. 163).

Toolbar Buttons and Check Boxes

Allows you to choose to view the entire network or a subnet configured using Setup Screen | View | Configure Subnets. AVAILABLE IN LOGGERNET ADMIN ONLY

Allows you to choose to view all devices or stations only.

Toggles the schedule for the selected datalogger on or off.

Resets the collection state and error statistics for the selected device.

Initiates data collection for the selected datalogger.

Stops a data collection attempt in progress.

Opens the LogTool (p. 172) application for analyzing communication messages between the LoggerNet server and datalogger network.

Opens a Comm Test (p. 175) window.

Pause Schedule - Select this check box to pause data collection schedules for all devices in the network.

> **NOTE:** Selecting **Pause Schedule** affects scheduled data collection from all dataloggers, as well as data collection via data advise in an RF-TD datalogger network and the collection of holes.

Server Time - Indicates the current time of the LoggerNet server.

Free Space - Indicates the amount of free space remaining on the LoggerNet server's hard drive.

Menu Items

The menu items provide another way to access the buttons on the Status window. There are, however, additional options that are described below.

File | Select Server - Allows you to select the LoggerNet server to which the Status window should connect (AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE).

Edit | Select Columns- The information displayed on the Status window can be customized. Select this menu item to display a dialog box that can be used to add or remove columns from the display. The resulting dialog box can also be accessed by right-clicking within the Status window. For additional information, see Select Columns (p. 163).

Edit | Continuous Sorting - When this option is enabled and a column is selected as the sort option for the stations displayed in the status window, the list of stations will be sorted continuously based on that column. This option is enabled by selecting it from the menu. A check box will appear to the left of the menu item when enabled. It is disabled by selecting the menu item a second time (AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE).

View Menu - The View menu is used to turn on or off some of the graphical components of the Status window. An item can be toggled on or off by selecting the menu item. If a check appears beside the menu item name, the associated image will be displayed.

> **Comm Status Images** - The Comm Status Images are the icons that show up to the left of the device name (), indicating the state of communication.
>
> **Trouble Indicator** - The Trouble Indicator is an exclamation point in a yellow circle that appears to the left of the device name (). If a communication problem occurs with a device, this indicator will appear before the text Network Map in the Network Map column and beside the device experiencing the problem.
>
> **Communication History** - The Communication History  is a chart that is placed to the right of the device name. This graph shows the state of the communication link with the device over the previous 24 hour period. The colors of bar for the chart indicate whether communication was Normal, Marginal, Critical, or unknown.
>
> **Collection Image** - When this option is enabled and datalogger is being dialed or is on-line for data collection, a green right-pointing triangle will be displayed to the left of the station name in the network map. If data collection with a station is pending (it is in the queue while data collection is occurring with another station), a blue right-pointing triangle will be displayed.

Tools | Terminal Emulator - Opens the Terminal Emulator (p. 115), which allows you to communicate with a device in remote terminal mode.

Tools | Pool Statistics - Opens a new window displaying overall error rate, availability, percent used, and current target for all pooled devices. For more information see Pool Statistics (p. 168).

Tools | Pool Devices- Opens a new window that offers information about each pool (root device) and each pooled device that has been assigned to it. For more information see Pool Devices (p. 169).

Tools | State of Operations - Opens a new window displaying a list of operations currently being tracked by the LoggerNet server. For more information see State of Operations (p. 170).

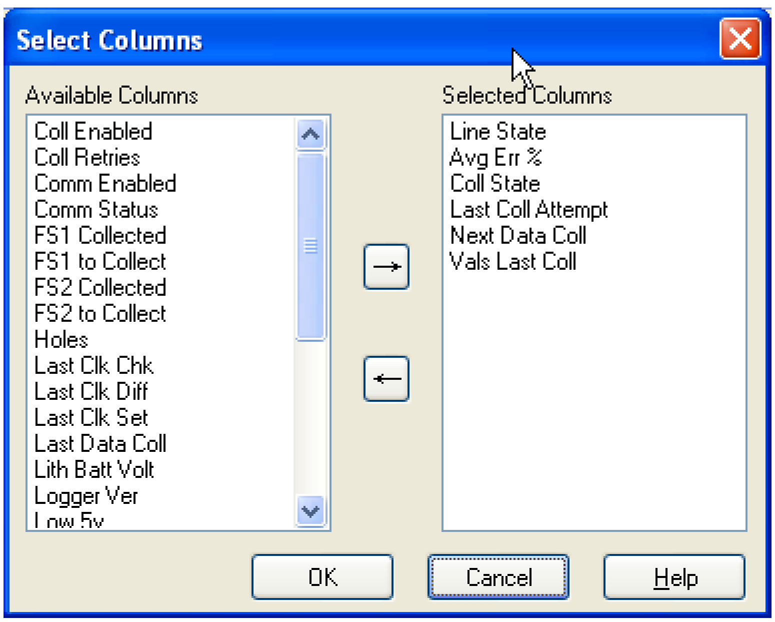
7.1.1 Select Columns

Columns of data can be added to the Status window that provide specific information about the communication link, data collection attempts, and clock checks.

When **Select Columns** is chosen from the Edit menu, the Select Columns dialog box appears. The **Available Columns** field will list all of the available statistics that are not already displayed on the Status window. To choose a statistic for display, highlight it and press the right arrow button or, alternately, double-click the column name. This moves the column heading into the **Selected Columns** field. When the dialog box is closed, the columns will be added to the Status window.

Columns can be removed from the display by moving them back into the **Available Columns** field using the left arrow button or, alternately, by double-clicking the column name in the **Selected Columns** list.

Columns will be displayed in the Status Monitor in the same order that they are listed in the **Selected Columns** list. You can reorder the columns by clicking on a column name in the **Selected Columns** list and dragging it to a new position in the list.



Note that some statistics are obtained automatically as part of data collection for some dataloggers but can be only obtained with additional communication commands for other dataloggers. In this latter case, these statistics are not retrieved by default as users with slow or expensive communication may not wish to incur the additional cost or time associated with the extra commands. In cases where the user does want to retrieve the additional statistics, the Poll for Statistics setting (on the datalogger's Schedule tab in the Setup Screen) can be enabled to request that the statistics are retrieved. The statistics will be retrieved during scheduled or manual data collection. These statistics are shown in the table below. The table also shows how the LoggerNet server maps these server statistics to the Status Table of each datalogger.

	Status Table Values						
Server Statistic (displayed in Status Monitor)	CR1000X CR6 Series CR300 Series CR350 Series GRANITE Data Logger Modules CR1000 CR800 Series CR3000	CR200	CR10XPB CR23XPB CR510PB	CR5000	CR9000X	CRS451 Series	CRVW Series
WatchDog Err	WatchDogErrors	WatchDogCnt	WatchDog	WatchDogErrors		WatchDogCnt	WatchdogErrors
Prog Overrun	SkippedScan	SkipScan	Overruns	SkippedScan	SkippedScan	SkipScan	Skipped Scan
Low Volt Stopped	Low12VCount						Low10_5V_Count
Low 5V	Low5VCount						
Lith Batt Volt	LithiumBattery*		LithBat	LithiumBattery	Battery	LithiumVoltage	RTC_Battery

NOTE: *For CR300-series dataloggers, a voltage will not be displayed. If the internal lithium battery supplied sufficient power to maintain the clock while external power was absent, the field will display "OK, ON POWER UP." If the internal lithium battery is missing or failed to supply enough power while external power was absent, the field will display "FAIL, ON POWER UP." The LithiumBattery field is only updated on power up, that is, when external power is first applied.

Column Descriptions

Avg Err % - A running average of the number of communication failures and retries over a period of time. If the average error rate is 5% or less, the text will be green. If it is greater than 5% but less than 20%, the text will be blue. If the average error rate is 20% or greater, the text will be red.

Coll Enabled - Indicates whether or not scheduled data collection is enabled for the device.

Coll Retries - The total number of data collection attempts that have occurred under the primary and secondary retry collection states (Coll State).

Coll State - This column provides information on the mode of data collection that the device is in.

Normal - Data is being collected on schedule.

Primary - Data collection will be attempted on the primary retry schedule because of failures of normal data collection.

Secondary - Data collection will be attempted on the secondary retry schedule because of failures of the primary retry schedule.

Schedule Off - The data collection schedule has been toggled off.

Comm Disabled - Communication has been disabled, by the user or by LoggerNet, for the device, a parent device, or the entire network.

Invalid Table Defs - Data collection has been disabled by LoggerNet because the table definitions do not match what LoggerNet expects. Go to the Setup Screen | Data tab and refresh the table definitions by selecting **Get Table Definitions**.

Network Paused - LoggerNet has paused communication in the network.

Unreachable - The device cannot be reached through the network.

The color of the text for the Collection State statistic will vary, depending upon the state of the device. The text will be green when collection state is normal. If the schedule is turned off the text will be blue. All other states will be reflected by red.

Comm Enabled - Indicates whether or not communication is enabled for the device.

Comm Status - A device has four communication states: **Normal** (green text), **Marginal** (blue text), **Critical** (red text), or **Unknown** (gray text). The current status of the device will be reflected in this column.

FS1 Collected - The number of final storage locations collected from final storage area 1 during the last data collection attempt (relevant only to array-based dataloggers).

FS1 to Collect - The number of final storage locations that were available for collection from final storage area 1 during the last data collection attempt (relevant only to array-based dataloggers).

FS2 Collected - The number of final storage locations collected from final storage area 2 during the last data collection attempt (relevant only to array-based dataloggers).

FS2 to Collect - The number of final storage locations that were available for collection from final storage area 2 during the last data collection attempt (relevant only to array-based dataloggers).

Holes - A "hole" is a discontinuity in record numbers for dataloggers that support data advise. This statistic is the number of values that are in holes.

Last Clk Chk - The date and time of the last clock check for the device.

Last Clk Diff - The amount of time the datalogger clock deviated from the LoggerNet server's clock when the last clock set was performed. This value will be reflected as positive (datalogger clock was greater than computer clock) or negative (datalogger clock was less than the computer clock).

Last Clk Set - The date and time that the datalogger's clock was last set to match the LoggerNet server's clock.

Last Col Attempt - The date and time that data collection was last attempted with the datalogger.

Last Data Collection - The date and time of the last successful data collection with the datalogger.

When scheduled data collection is enabled and the time for the Last Collection Attempt has surpassed the Last Data Collection, the text for both of these fields will be blue if polling is active or red if polling is not active.

Line State - The state of communication with the device in the network.

> **Off-line** - No communication is taking place with the device.
>
> **On-line** - Communication for all parent devices for the device has been successful and the parent devices have been put into a transparent state. Therefore, an On-Line state for a device indicates that the communication link is ready for the device, but it may not necessarily be on-line yet.
>
> **Transparent** - The device is part of an active communication link. For instance, when a datalogger is called using a phone modem, the phone modem will become transparent once communication is established with the datalogger.
>
> **Undialing** - LoggerNet is in the process of shutting down communication with the device.
>
> **Comm Disabled** - Communication has been disabled either by the user or by LoggerNet.
>
> **Unreachable** - The device cannot be reached through the network.
>
> **Pending** - The device has requested the link from its parent but that request is still pending.
>
> **Targeted** - The device has requested the link from its parent and its parent (and grandparents) are being dialed to open the link.
>
> **Waiting** - The device is a TCP comm port that is waiting for an incoming callback.
>
> **Not-applicable** – In its current configuration, the device will not communicate directly with the server. This value will appear in association with an array-based datalogger connected to the server through an RF95T.

The color of the text for the Line State statistic will vary, depending upon the state of the device. The text will be black when the device is Off-line. The text will be green when device is On-line. The text will be blue if a transaction is Pending or Targeted. The text will be gray when a device is Transparent. The text will be red if Communication is Disabled or a device is Unreachable.

Link Time Remaining – The time remaining, in milliseconds, until the Maximum Time On-Line is reached and the device is automatically disconnected. The value is only updated every 10 seconds. A value of 4294967295 indicates that the device is not connected or the Maximum Time On-Line is not set.

Lithium Batt Volt - The voltage level of the datalogger's lithium SRAM back-up battery.

> **NOTE:** For CR300-series dataloggers, a voltage will not be displayed. If the internal lithium battery supplied sufficient power to maintain the clock while external power was absent, the field will display "OK, ON POWER UP." If the internal lithium battery is missing or failed to supply enough power while external power was absent, the field will display "FAIL, ON POWER UP." The LithiumBattery field is only updated on power up, that is, when external power is first applied.

Logger Ver - The version code for the datalogger. (This option is available only for array-based dataloggers).

Low 5 V - The number of times the datalogger's 5V supply has dropped below 5V.

Low Volt Stopped - The number of times the datalogger program has been halted because the datalogger's 12 V power source has dropped below the minimum power requirement.

Mem Code - The amount of memory, in kilobytes, in the datalogger. (This option is available only for array-based dataloggers).

Next Data Coll - The date and time of the next scheduled data collection for the device.

Polling Active - Either **True** or **False**; indicates whether or not LoggerNet is currently querying the datalogger for data.

Prog Overrun - The number of times an execution interval has been skipped in the datalogger, because it was busy with another task.

RFTD Blacklisted - Indicates that a station has been blacklisted by the RF Base following a failed communication attempt. The RF Base will not forward communication attempts originating from LoggerNet to a blacklisted station. This includes clock checks, getting table definitions, data collection, program sends, etc. However, the RF Base will continue to request communications with the blacklisted station on its regular RF Polling Interval. Once the station has responded to one of these regular RF polling broadcasts, it will be removed from the blacklist. At that point, communication attempts originating from LoggerNet will be allowed again.

Table Defs State - Indicates the status of the datalogger's table definitions, as known by the LoggerNet server.

None - No table definitions have been retrieved from the datalogger (red text).

Current - The table definitions from the datalogger match what LoggerNet has stored as the table definitions for the datalogger (black text).

Suspect - A collection attempt has returned an invalid table definitions code. LoggerNet will attempt to verify the table definitions for the datalogger (blue text).

Getting Table Defs - LoggerNet is in the process of retrieving the table definitions from the datalogger (black text).

Invalid Table Defs - The table definitions from the datalogger do not match what LoggerNet has stored as the table definitions for the datalogger. Table definitions will need to be updated before data collection can occur (red text).

Total Attempts - The total number of times LoggerNet has attempted to communicate with the device.

Total Failures - The total number of times the primary and secondary retries for a device have been exhausted before the communication attempt was successful.

Total Retries - The total number of times LoggerNet has attempted to communicate with a device after the original attempt failed.

Uncoll Holes - The number of values in holes that cannot be collected from the datalogger (most often, because the data has been overwritten by newer data).

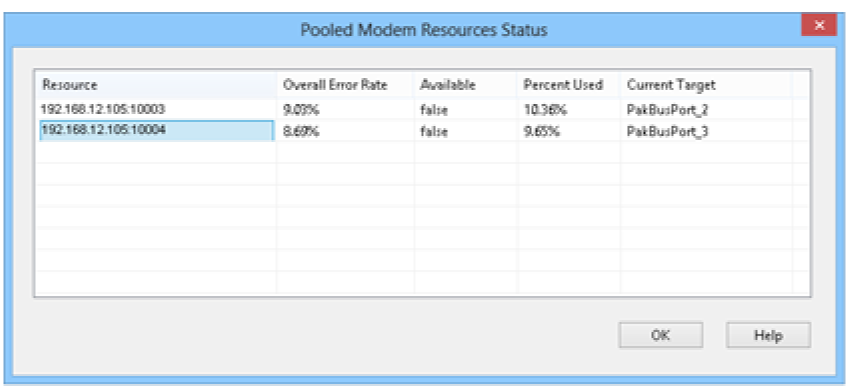
Vals Last Coll - The total number of data values collected as a result of the last scheduled or “Collect Now” data collection attempt plus any data values received as a result of One-Way Data or Data Advise transactions. The number of values will not be reset until data is collected either through a scheduled collection or by pressing the Collect Now button.

Vals to Coll - The number of values that were available for collection during the last data collection. (Note that if data is received from the datalogger through either One-Way Data or Data Advise, the number of values received will be added to both Vals Last Coll and Vals to Coll. These values will not be cleared until data is collected either through a scheduled collection or by pressing the Collect Now button.)

Watchdog Err - The number of times the datalogger's processor has been reset.

> **NOTE:** Recurring watchdog errors could indicate a problem with datalogger hardware or with the program in the datalogger.

7.1.2 Pool Statistics



This dialog shows all of the pooled devices. For each pooled device (resource), the following information is given:

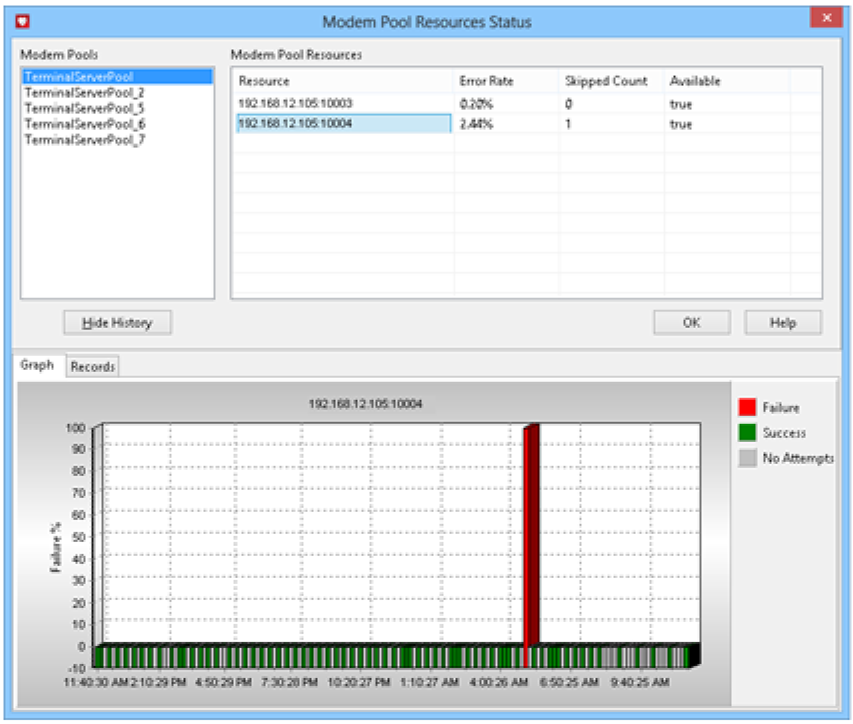
Overall Error Rate - This represents the error rate based on all dialing attempts this device has made.

Available - This indicates if the resource is currently available or if it is in use.

Percent Used -This gives an indication of how much this resource has been used. This can assist in organizing pools and devices to minimize wait time or increase calling rates.

Current Target – This shows the target device of the current call.

7.1.3 Pool Devices



This dialog offers information about each pool (root device) and each pooled device that has been assigned to it. For example, given that a modem pool root device (SerialPortPool or TerminalServerPool) for a particular station has three serial ports assigned to it, the specific information for each of the serial port/modems, based on when it has been used to call that station, can be viewed by selecting the Modem Pool and in turn each Modem Pool Resource assigned to it. The following information can be displayed:

Error Rate – This represents the error rate specific to the selected Modem Pool use of the selected pooled device.

Skipped Count – The number of times this pooled device has been skipped when using the selected Modem Pool.

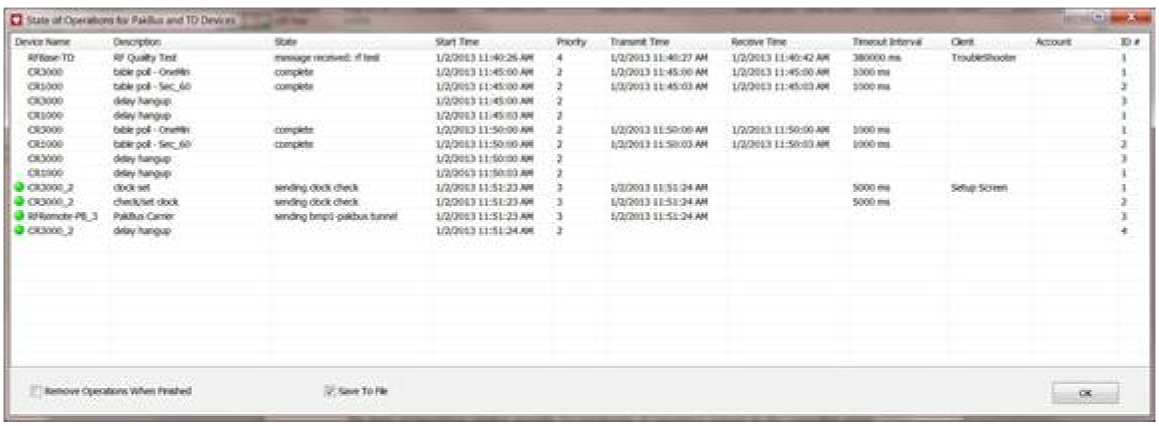
Available – Indicates if the selected resource is available or is in use.

The Graph can be used to view the history of attempts to use the selected resource (pooled device) by the selected Modem Pool.

The Records tab is used to view events associated with the use of the pooled resources.

7.1.4 State of Operations

The State of Operations window provides for monitoring of operations queued by the LoggerNet server for BMP1 and BMP5 devices in the network map (i.e., CR10X-TD, CR23X-TD, CR510-TD, CR10X-PB, CR510-PB, CR23X-PB, CR200X Series, CR800 Series, CR3000, CR1000, CR1000X, CR6 Series,CR300 Series, CR350 Series, GRANITE 6, GRANITE 9, GRANITE 10, RFBase-TD, and RFRemote-TD/PB).



Column Descriptions

Device Name – Indicates the name of the device associated with the operation.

Description – Describes the type of operation.

State – Indicates the current state of an active operation, or the most recent state of a completed operation. Currently active operations are identified by a green circle displayed to the left of the Device Name.

Start Time – Indicates the time the operation started.

Priority – Indicates the priority for this operation as a value between 0 and 4.

No priority used

Low priority

Normal priority

High priority

Top priority

Normal priority operations (scheduled collections, etc.) will have a priority of 2. Client sponsored operations will typically have priority of 3 or 4. Operations such as automatic hole collection will typically have a value of 1.

Transmit Time – Indicates the time the operation last transmitted to the device.

Receive Time – Indicates the time the operation last received information from the device.

Timeout Interval – Indicates the time out interval, in milliseconds, for any datalogger transaction associated with this operation.

Client – If a client application initiated the operation, the name of the client application is indicated. Otherwise, this field will be empty.

Account - Indicates the logon name or username associated with a client application. If no username is provided by the client, this field will be empty.

ID # - Indicates the server's identifier for the operation.

Remove Operations When Finished

When this check box is selected (default), operations that are no longer active will be deleted from the displayed list. If this box is not selected, the last state of the operation before completion will continue to be displayed in the list. The displayed list is limited to a maximum of one thousand lines. After reaching the limit, the oldest lines are deleted as new lines are added.

Save to File

When this check box is selected, the information provided by the server for each listed operation is saved to a comma delimited text file (C:\Campbellsci\LoggerNet\Operations.log). The information logged provides a record of when the operation was added, changed (updated), and deleted by the server. For each line in the file, the information is ordered as follows: Event, Start Time, Device Name, Description, State, Priority, Transmit Time, Receive Time, Timeout Interval, Client, Account, and ID#. Entries in the file are limited to twelve thousand lines. After reaching the limit, the oldest four thousand lines are deleted. Note that the file will only be written to, when the State of Operations window is open.

Added, 1/2/2013 8:45:00 AM, CR3000, table poll - OneMin, , 2, , , , , , 1

Added, 1/2/2013 8:45:00 AM, CR1000, table poll - Sec_60, , 2, , , , , , 2

Changed, 1/2/2013 8:45:00 AM, CR3000, table poll - OneMin, requesting focus, 2, , , 1000 ms, , , 1

Changed, 1/2/2013 8:45:00 AM, CR1000, table poll - Sec_60, requesting focus, 2, , , 1000 ms, , , 2

Added, 1/2/2013 8:45:00 AM, CR3000, delay hangup, , 2, , , , , , 3

Changed, 1/2/2013 8:45:00 AM, CR3000, table poll - OneMin, requesting focus, 2, 1/2/2013 8:45:00 AM, , 1000 ms, , , 1

Changed, 1/2/2013 8:45:00 AM, CR3000, table poll - OneMin, requesting focus, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, , , , 1

Changed, 1/2/2013 8:45:00 AM, CR3000, table poll - OneMin, collecting holes between 28827 and 28830, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, 1000 ms, , , 1

Changed, 1/2/2013 8:45:00 AM, CR3000, table poll - OneMin, complete, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, 1000 ms, , , 1

Deleted, 1/2/2013 8:45:00 AM, CR3000, table poll - OneMin, complete, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, 1000 ms, , , 1

Added, 1/2/2013 8:45:00 AM, CR1000, delay hangup, , 2, , , , , , 1

Changed, 1/2/2013 8:45:00 AM, CR1000, table poll - Sec_60, getting newest record, 2, 1/2/2013 8:45:00 AM, , 1000 ms, , , 2

Changed, 1/2/2013 8:45:00 AM, CR1000, table poll - Sec_60, getting newest record, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, , , , 2

Changed, 1/2/2013 8:45:00 AM, CR1000, table poll - Sec_60, collecting holes between 29 and 32, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, 1000 ms, , , 2

Changed, 1/2/2013 8:45:00 AM, CR1000, table poll - Sec_60, complete, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, 1000 ms, , , 2

Deleted, 1/2/2013 8:45:00 AM, CR1000, table poll - Sec_60, complete, 2, 1/2/2013 8:45:00 AM, 1/2/2013 8:45:00 AM, 1000 ms, , , 2

Deleted, 1/2/2013 8:45:00 AM, CR1000, delay hangup, , 2, , , , , , 1

Deleted, 1/2/2013 8:45:00 AM, CR3000, delay hangup, , 2, , , , , , 3

7.2 LogTool

The LogTool utility allows you to view the communication packets transferred between the computer running the datalogger support software and other devices in the network. These logs can be used for troubleshooting communication problems. Four logs are available:

Transaction Log (tran$.log) - This log includes information on the transactions that occur between the datalogger support software and devices in the datalogger network. Examples of these types of events are clock checks/sets, program downloads, and data collection.

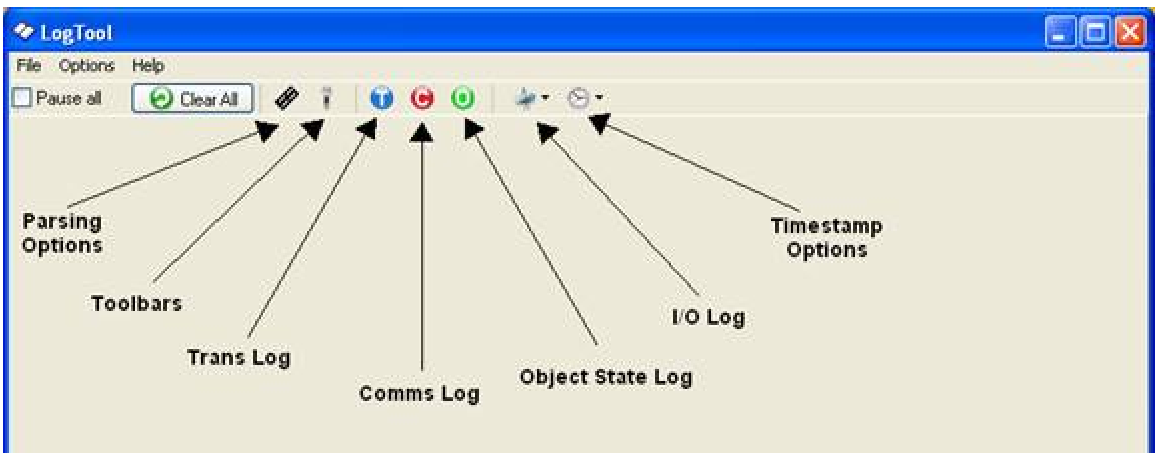
Communication Log (comms$.log) - This log records information on the quality of communications in the datalogger network. Three types of messages are recorded: status messages (shown in black), warning messages (shown in blue), and failure messages (shown in red).

Object State Log (state$.log) - This log is used for troubleshooting an object in the datalogger network. The information in this log conveys the state of an object at a given time.

Low Level I/O Log (io$SerialPort_1) - This log displays low level incoming and outgoing communications for a root device (i.e., serial port).

These logs can be saved to a file by selecting **Options | Log File Settings** from the LogTool menu. (For additional information, see Log File Settings (p. 174).) The log messages currently being displayed can be saved to a text file by selecting **File | Save Log Windows to File**. The log messages currently being displayed can be printed by selecting **File | Print Log Windows**.

When the LogTool is first opened, two logs are displayed: the transaction log and the communication log. The Object State log and the Low Level I/O log can be displayed by toggling their associated button on the toolbar. Refer to the diagram below for information on these and other functions available from the toolbar. Note that most of these functions are also available from the View menu.



Parsing Options - Toggles the display of the Message Parsing Window (p. 174).

Toolbars - Toggles the display of an individual tool bar for each of the logs. You can pause the display of messages for a tool bar by selecting the **Pause** check box. You can clear all messages for a log by pressing the button. For the Communication Log, you can also choose what type of messages to show (failure, warning, and/or status) by pressing the button.

Trans Log - Toggles the display of the Transaction log.

Comms Log - Toggles the display of the Communication log.

Object State Log - Toggles the display of the Object State log.

I/O Log - Opens the Low Level I/O log for a specific COM port in a new window.

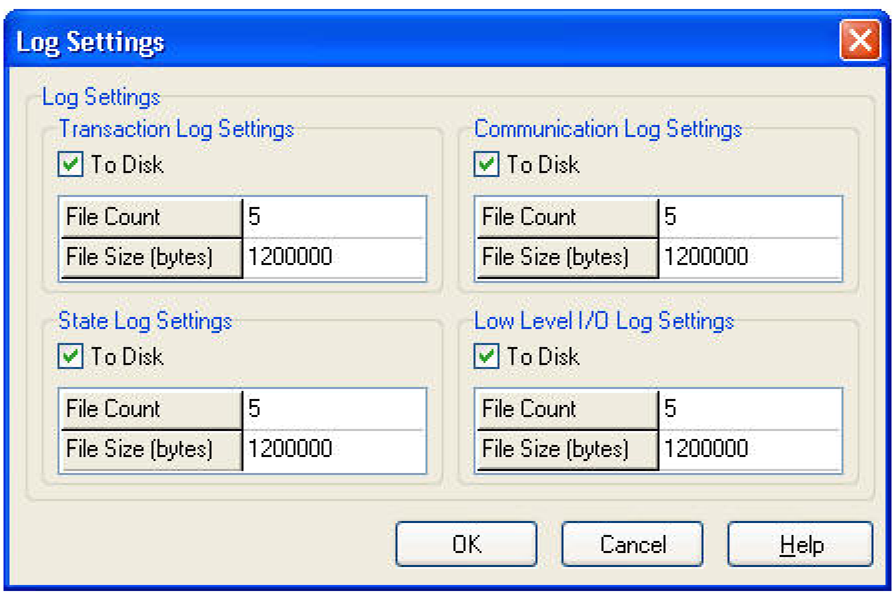
TimeStamp Options - Allows you to select the format for the time stamp in the logs. If none of the options are enabled (an option is enabled if a check mark appears to the left of the option name), only the time is displayed (hh:mm:ss AM/PM). If **Date** is selected, a date (MM/DD/YY) will be added to the time stamp. If **Military** is selected, the time stamp will be displayed in 24 hour format instead of 12 hour format. If **ms Resolution** is chosen, the time stamp will also include milliseconds.

7.2.1 Message Parsing Window

When you are trying to determine the cause of a communication problem with a datalogger, it can be overwhelming trying to interpret all of the information coming from all of the devices in the network. It is possible to apply a filter so that messages for only the device(s) you are interested in are shown in the LogTool displays.

The left side of the LogTool window displays a list of all devices in the network. By default, this list is set to **Show All Msgs**. When this option is selected, all communication messages for all devices in the network will be displayed in the logs. You can look at the logs for one or more devices in the network by choosing **Select by Device** from the drop-down list box, and then selecting the check box that appears to the left of each device for which you wish to view logs. This option makes available all devices in the network, including COM ports, modems, RF devices, etc. If you want to view logs for datalogger stations only, choose the **Select by Station** option from the list box and check the box for one or more dataloggers. Both the **Select by Device** and **Select by Station** options have a **List Alphabetically** check box. This can make it easier to find the desired device or station.

7.2.2 Log File Settings

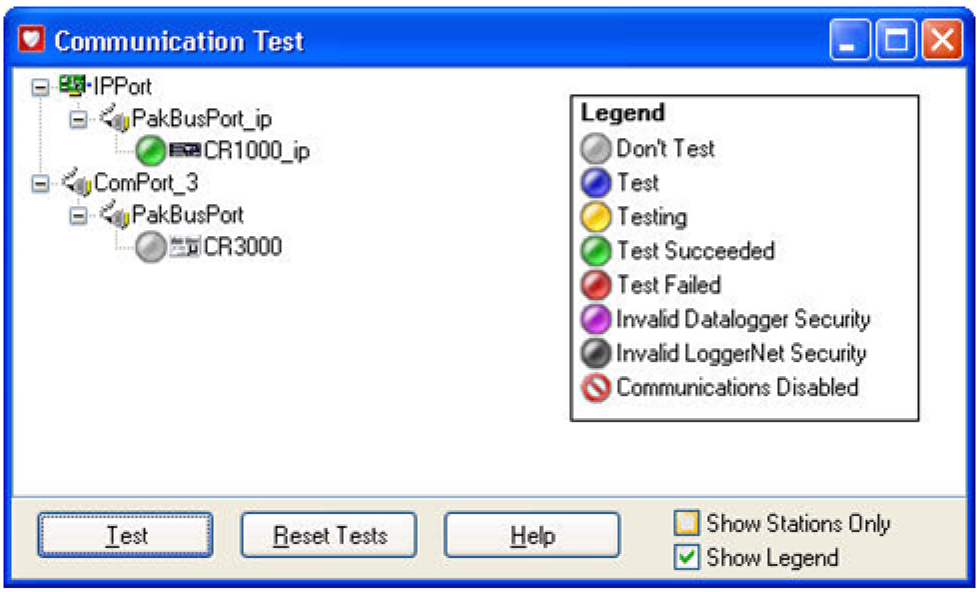


The Log File Settings option is used to configure the four log files that contain information about communication in the datalogger network. The four logs are the **Transaction Log** (tran$.log), Communication Log (comms$.log), **Object State Log** (state$.log), and **Low Level I/O Log** (io$SerialPort_1).

The log files can be saved to disk by selecting the **To Disk** check box. If the log files are being saved to disk they will have filenames as noted above. A $ sign identifies the active file. Once the files reach a specified size (**File Size**), they are saved to disk with sequential numbers beginning with 0 (i.e., tran0.log, tran1.log, etc.). The number of logs saved is determined by the **File Count** field.

> **NOTE:** Log files have the potential of consuming a significant amount of hard disk space. For instance, if you accept the default file count of 5 and have 5 communication ports in your network map, you could have twenty-five 1.2 M files for each log type that is enabled.

7.3 Comm Test



This option allows you to test the communication link to the datalogger stations. The color of the circle to the left of each datalogger station indicates the quality of communication. The legend displayed on the right side of the window provides a key to the color codes. The legend can be removed from the window by clearing the **Show Legend** check box.

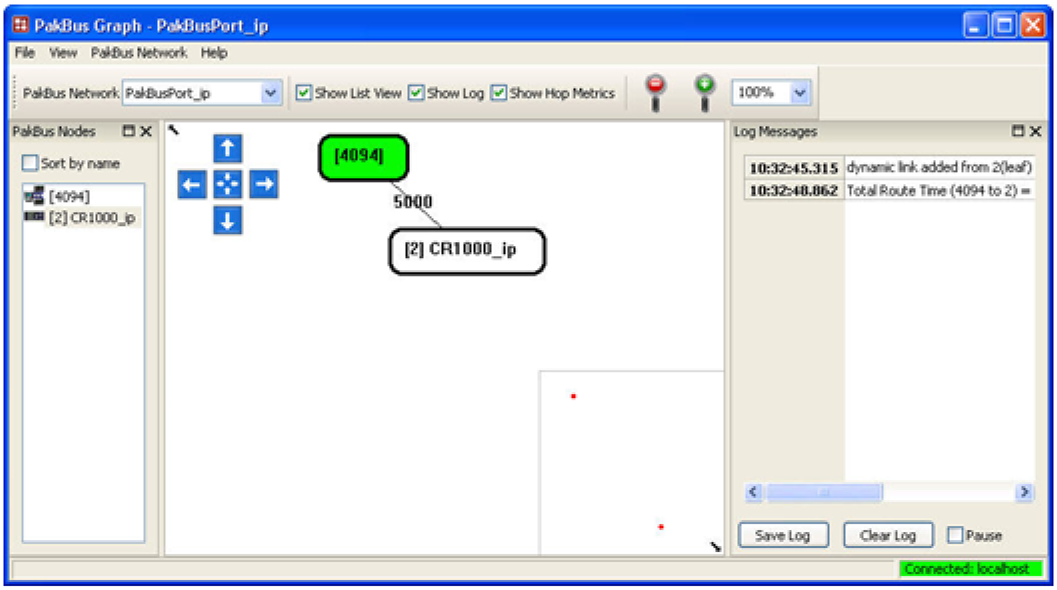
When you first open Comm Test, the state of the devices is unknown, so the circles for each device will appear grey.To initiate the test, click on one or more of the datalogger stations to select them (the circles will appear blue), and press the **Test** button. The LoggerNet server will attempt to contact the selected device(s) and perform a simple clock check. While a test is in progress, the circle for a device will appear yellow. Once the test is performed, the resulting circle will be green (clock check successful) or red (clock check failed).

Press **Reset Tests** to clear the test results before running the test again.

By default, the Communication Test window shows all devices in the network. You can display only the stations by selecting the **Show Stations Only** check box.

7.4 PakBus Graph

PakBus Graph is a utility that graphically depicts the connections in a PakBus datalogger network. In PakBus graph, the PC400 software server is, by default, assigned PakBus address 4093, the RTDAQ software server address 4090, and the LoggerNet software server address 4094. Each of the PakBus dataloggers that have been configured will be shown by name and address.



7.4.1 Static and Dynamic Links

There are two types of links to PakBus dataloggers that the server recognizes: static links and dynamic links. Static links (depicted using red lines) are the communication links to dataloggers that have been set up in the software, but which have not been confirmed by communicating with the datalogger(s). You will see these dataloggers listed in the software's network map. Dynamic links (black lines) are communication links to dataloggers that have been confirmed. You may also see links to leaf node dataloggers that have not been set up in the software, but which the server has "learned about" by activity within the PakBus network.

7.4.2 Connecting to the Software Server

When PakBus Graph is opened in LoggerNet Admin or LoggerNet Remote, you will be prompted for the address of the server with which to connect and a user name and password:

Server Address - This is the TCP/IP address of the computer running LoggerNet software, to which you are trying to connect. This must be the valid name of an existing computer or a TCP/IP address (in the form

###.###.###.### consisting of the IP network number, ###.###.###, and the host number, ###). If the software server resides on the same computer as PakBus Graph, you can simply type in LocalHost for the server name.

User Name - Your user name on the LoggerNet server.

Password - Your password for the LoggerNet server.

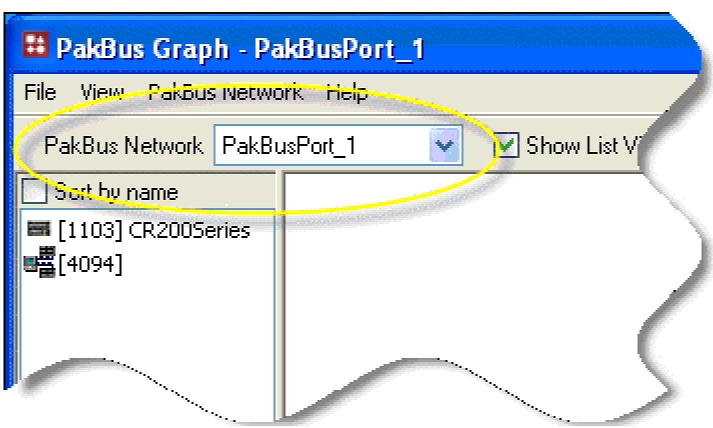
> NOTE: The **User Name** and **Password** fields are required only if security has been set up on the software server to which you are trying to connect.

Each time you start PakBus Graph, you will be prompted to enter this information. However, you can save the login information by selecting the **Remember User Name and Password** check box, or you can select the **Automatically log in with this information** check box to skip this window and use the information from the last session.

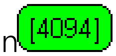
> NOTE: You do not have the option of selecting a server when PakBus Graph is opened in RTDAQ, PC400, or the standard version of LoggerNet. The connection is always to LocalHost.

7.4.3 Selecting the PakBus Network to View

By default, each of the software server's PakBus networks (comprised of the dataloggers attached to a specific PakBus Port) is independent from the other networks. Each network is displayed individually in PakBus Graph. To select the network to view, select the PakBus Port from the **PakBus Network** list box.



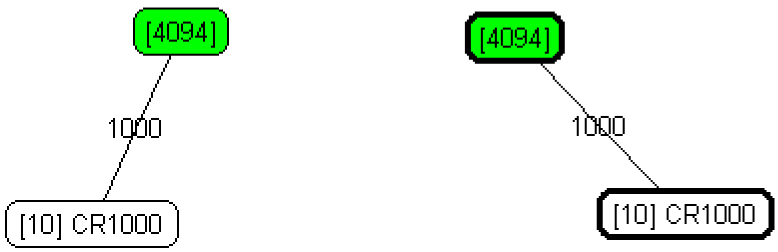
The software server can also be set up with "bridged" PakBus ports. In this case, every datalogger attached to every PakBus port is part of the same network. If the PakBus ports have been bridged, the resulting single port will be named "__global__" and it will be the only selection in the **PakBus Network** list.

Software servers are identified in PakBus Graph by the color green

. Other devices remain colorless unless they have been selected with the mouse cursor. When selected, they are colored cyan

.

The network map will change as new devices are discovered or known devices fall out of communication with the network. When the network map is redrawn, devices may be displayed in a new position on the screen. To make the network easier to read, you can reposition the devices on the screen by dragging them to a new location. Once a device is moved, it is "locked" into position and the outside border will be shown with a heavier line than an "unlocked" device. The locked position of a device will be maintained between PakBus Graph sessions. You can unlock a device by right-clicking it and choosing **Unlock Position** or you can unlock all devices in the network by selecting **View | Unlock All Positions** from the menu.



Unlocked Devices **Locked Devices**

7.4.4 PakBus Graph Configuration

PakBus Graph can be set up to show one or all of the following informational objects. By default, they are all displayed when PakBus Graph is opened. (If changes have been made to this default view, you can select **View | Restore Default View** to return to the default view.)

Select the **Show List View** check box to display a list of the PakBus devices on the left side of the graph window. Clear the check box to remove the list. By default, devices are displayed in this list in numerical order. You can sort the list by name in alphabetical order by selecting the **Sort by Name** check box.

Select the **Show Log** check box to display a log of PakBus communication messages on the right side of the graph window. The time will be displayed indicating when each message was sent to or received from the datalogger.

When the **Show Hop Metrics** check box is selected, the graph will include the time, in milliseconds, that communication takes to each device. You can also click a device, and then click a second device, to retrieve the communication time between the two devices. The results are displayed in the Log Messages portion of the window.

7.4.5 Viewing/Changing Settings in a PakBus Datalogger

If you right-click a device in PakBus graph, you will be presented with a floating menu. From this menu, select **Edit Settings** to display a list of the PakBus settings for the datalogger. Some of these settings are read-only, but other settings can be changed. Click within the cell for a setting, enter a new value and then press return to make a change.

> **NOTE:** Pressing F5 with the Settings window open will refresh the Settings table.

7.4.6 Right-Click Functionality

There are several options available from the floating menu that is displayed when you right-click a device (not all devices will have all settings):

Edit Settings - This option shows the PakBus settings of a device (see above).

Ping Node - This option will send a packet to the selected device to determine if it is reachable in the PakBus network. The results of the ping will be displayed in the Log Messages. Each ping message will include the size of the packet sent, and the time of response from the pinged device. The last message recorded will include summary information from the ping.

Verify Routing Table - This option will request the routing table from the highlighted PakBus device.

Reset Node - This option will reset the routing table in a PakBus device.

Change PakBus Address - By default, the PakBus address of the software server is 4093 (PC400), 4090 (RTDAQ), or 4094 (LoggerNet). This option lets you change this default.

Search for Neighbors - When this option is selected, the software server will broadcast a Hello Request every 5 seconds to search for PakBus neighbors with which it can communicate. During this time, the PakBus port is kept on-line.

> **NOTE:** PakBus Graph will continue searching for neighbors until the **Stop** button is pressed or the Search for Neighbors window is closed. This is because there is no way to predict how long it may take to find neighbors.

Broadcast Reset - This option will reset the routing table in the selected PakBus device, as well as any neighbors of the selected device that are acting as routers.

Unlock Position - This option will unlock a device that has been locked into position in PakBus Graph by dragging it to a new position on the screen. All devices can be unlocked by selecting **View | Unlock All Positions** from the menu.

7.4.7 Discovering Probable Routes

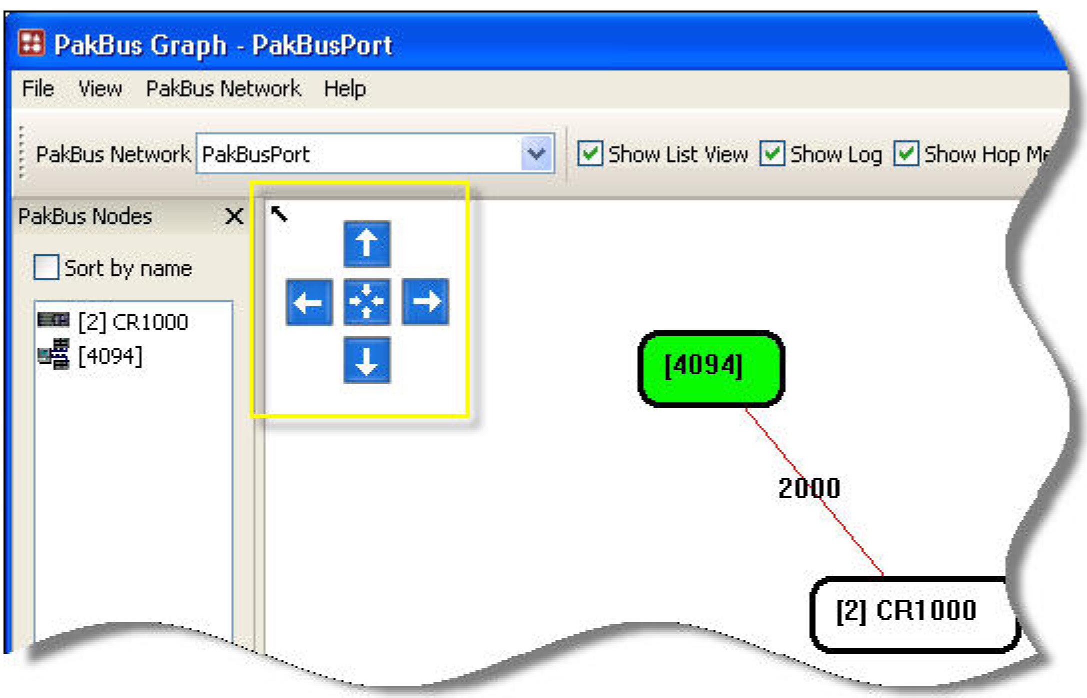
You can view the probable route between two devices by sequentially clicking on the two devices in Pakbus Graph. The probable communication route will be highlighted in cyan. If the **Show Hop Metrics** check box is selected, the graph will include the time, in milliseconds, that communication takes between the two devices. The results are also displayed in the Log Messages portion of the window.

7.4.8 Scrolling

The PakBus Graph network map window is designed so that it has no real boundaries. Rather than using traditional scroll bars, PakBus Graph provides two means of changing the viewing window: the scroll buttons and the model overview. In addition to these features, you can scroll the viewing window directly using the mouse.

Using the Scroll Buttons

Navigation buttons that can be used to scroll the network map window are highlighted in the image below.



The following buttons and features are associated with this control:

Arrow Buttons—Pressing any arrow button will scroll the window in the direction of the arrow.

Center Button—Pressing the center button will reposition the window at the graph origin.

Collapse Button—The small black arrow in the upper left hand corner of the scroll button area allows you to hide the navigation buttons. When this is done, the black arrow will remain but point in the opposite direction and, if clicked on, will expand the scroll button area.

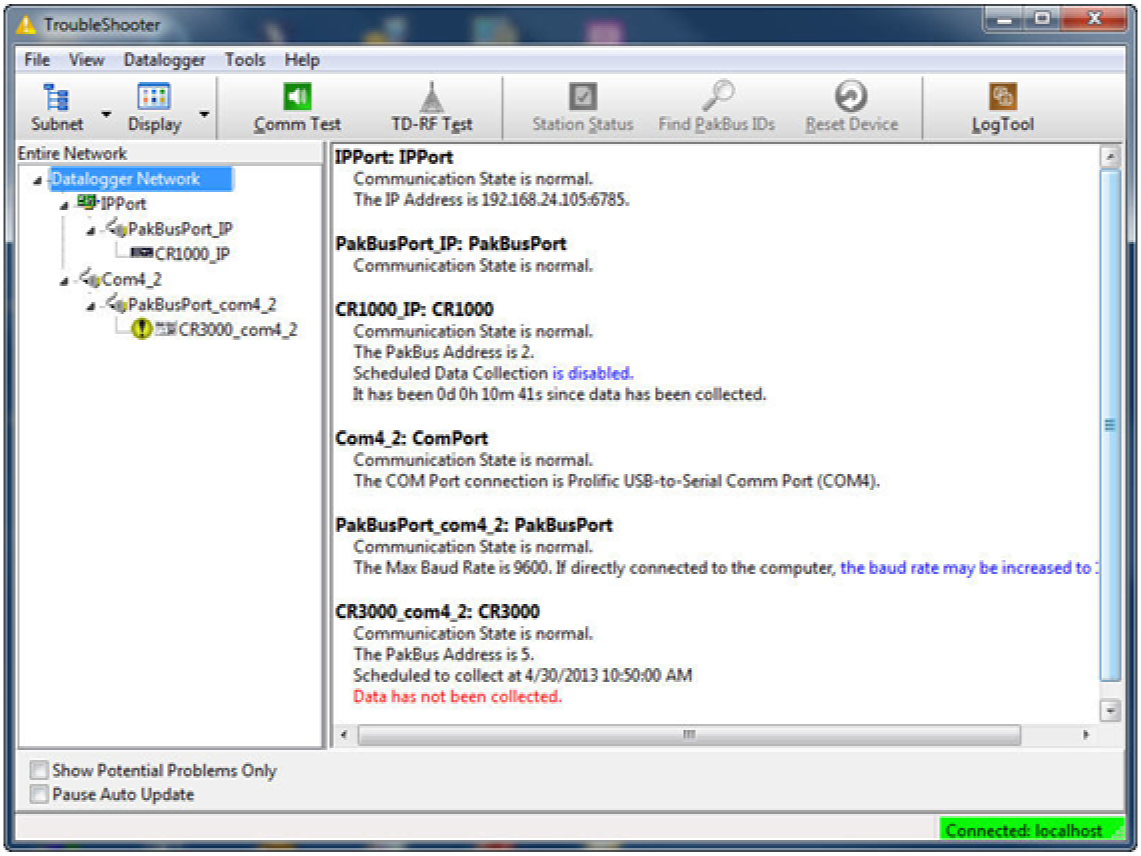
Using the Model Overview

The model overview is a small square area in the lower right hand corner of the window. The positions of devices in the window are represented with red dots. A black rectangle is used within this area to show the current viewing area. If you click the left mouse button while the pointer is anywhere within this area, the window will be scrolled so that the center of the window is positioned relative to the mouse position within the overview area. If you depress the left mouse button and drag the mouse pointer, the window will be scrolled as you move the mouse. If you click on the small black arrow on the lower left hand corner of the overview area, most of the area will be hidden although the arrow will remain with a reversed direction. Pressing the arrow will make the model overview area visible again.

Using the Mouse

You can scroll the network map window by holding down the left mouse button while the mouse pointer is over a blank area of the window and dragging the mouse in the desired direction.

7.5 TroubleShooter



The TroubleShooter is a tool that can be used to help assess communication problems in a datalogger network. When the TroubleShooter is opened, all communication ports, peripheral communication devices, and dataloggers that have been set up in LoggerNet will be displayed on the left side of the window.

To review the current known state of devices in the network, use your mouse to highlight the Datalogger Network (to see all devices) or highlight only one device (to see the status of all devices required to reach that device). On the right side of the window, status information is displayed. Note that the information is not updated until communication is attempted with the datalogger (either manually from the Connect window or automatically based on scheduled data collection).

The information will be different for each device, but may include the type of device, the device name in the network map, the state of communication with the device, whether or not scheduled data collection is enabled, whether or not table definitions are valid, modem type, phone number, device address, and the error rate.

Much of the status information is self-explanatory. Some information which may require further definition follows.

Communication State - Communication State will be listed as Unknown until communication is attempted with a device. Once a communication attempt has been initiated, it will be listed as Normal (communication is successful), Marginal (there are some errors in communication, but none that cannot be recovered from), or Critical (communication failed).

Scheduled Data Collection- This is the automatic data collection schedule. The state is either enabled, disabled, or paused. Scheduled data collection is paused from the Status window (Pause schedule check box).

Table Definitions- Table-based dataloggers return information on all the data tables in the datalogger to the LoggerNet server. The state can be Valid or Out of Date. If table definitions are out of date, they must be updated before data can be collected from the datalogger (LoggerNet attempts to do this automatically. Table definitions can be updated manually from the Setup window's Data File tab for the datalogger.)

Avg Error Rate- A running average, expressed in percent, of the number of communication failures and retries over a period of time.

Buttons

Subnet - Allows you to choose to view the entire network or a subnet configured using Setup Screen | View | Configure Subnets. AVAILABLE IN LOGGERNET ADMIN ONLY

Display - Allows you to choose to view all devices or stations only.

Comm Test - Pressing the **Comm Test** button will open the Communication Test window. From this window you can test the communication link for one or more dataloggers in the network map. For additional information, see Comm Test (p. 175).

TD-RF Test - This option opens a window from which you can perform a communications test on a table data RF modem link (RFBase-TD, RFRemote-TD, or RFRemote-PB). This test is not applicable for RF400 radios or non-TD based RF modems. See Tools | TD-RF Test (p. 185) for additional information.

Station Status - Highlight a datalogger from the list on the left and press the **Station Status** button to display the station status information from the datalogger. For additional information on Station Status, see Station Status (p. 116).

Find PakBus IDs - This option is used to find PakBus devices attached to a PakBus port within the communication network. Highlight a PakBusPort from the list on the left and press the **Find PakBus IDs** button. LoggerNet will initiate a search for all PakBus devices on that particular PakBus Port. (It may take a few moments to return a response.) If the PakBus ports are bridged, the IDs for all PakBus devices found will be returned.

Reset Device - Resets the statistics and collection schedule for the selected device.

Log Tool - Pressing the **Log Tool** button opens the Log Tool application. For additional information on Log Tool, see LogTool (p. 172).

Check Boxes

Show Potential Problems Only - This check box is used to filter out all messages but ones that may point to a communication problem.

Show Stations Only - This check box is used to show status information for only the dataloggers in the TroubleShooter window.

Pause Auto Updates - This check box is used to pause automatic updates to the status information. This option is generally only useful when viewing an entire large network. This is because with a large network, not all status information is visible on the screen, and a scroll bar will be used to scroll through the information. Automatic updates always end with the scroll bar at the bottom of the screen, which can cause difficulty in reading the information at the top of the screen.

Menu Items

Many of the menu items provide an alternate way of accessing the functions that can be launched by the buttons or check boxes on the main window of the TroubleShooter. Other menu items are as follows:

File | Save Analysis to File or **Print Analysis** - Saves or prints the information shown on the right side of the TroubleShooter window. If the Datalogger Network is selected, information for all devices will be printed or saved. If only one device is selected, only information for that device is printed or saved.

View | Customize Network Checks - Use this option to select the items for which warnings are to be given. When the **Select All** check box is marked, all warnings will be given. For only specified warnings to be given, clear the **Select All** check box and then manually select those warnings you wish to see.

View | Show Hints - When this option is selected, popup hints will be displayed for a button if the mouse cursor hovers over the button for a few seconds. When the option is not selected, no popup hints will be displayed.

Datalogger | Update Table Definitions - This option is used only for table-based dataloggers.

Table definitions provide information on the number of tables in the datalogger, the size of the tables, and the data values they contain. The LoggerNet server uses this information when creating the data cache for the datalogger. If the table definitions in the datalogger are different than the information that LoggerNet has stored, data collection may be disabled. When this menu item is selected, LoggerNet will update its information with table definition information from the highlighted datalogger.

Datalogger | Reset to Factory Defaults (Hardware Reset) - This option is used only for mixed array dataloggers with a default operating system.

This option should be used with extreme caution. It should be performed across a direct COM port connection. It will perform a full hardware reset of the datalogger. All memory will be erased including any data and programs. The datalogger will be returned to its factory defaults. The communication link will be lost and the datalogger will do a full memory test. Communication should not be attempted with the datalogger until after the memory test is complete. The time required for this memory test is as follows:

Datalogger	Time
CR10X	35 seconds
CR10X-2M	8 minutes
CR510	35 seconds
CR510-2M	8 minutes
CR23X	8 minutes
CR23X-4M	16 minutes

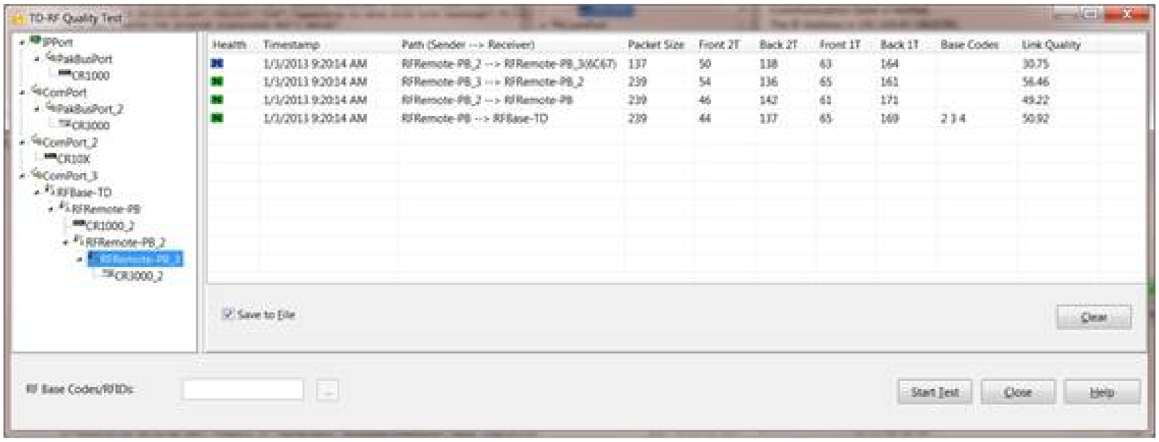
Tools | Terminal Emulator - Opens LoggerNet's Terminal Emulator (p. 115).

7.5.1 Tools | TD-RF Test

This selection launches the TD-RF Quality Test window from which one can execute RF Link Quality Test (p. 188) and/or activate the Advanced Features (p. 191) of the TD-RF modems. The features and functions accessible through the TD-RF Quality Test window are applicable only to TD-RF devices in the network (RFBase-TD, RFRemote-TD, and RFRemote-PB).

The left-hand pane of the TD-RF Quality Test window displays the network map as configured in LoggerNet, and provides a means of selecting a device or an RF path to be tested. The branches of the network map can be expanded or collapsed by clicking the small triangle to the left of the parent device. The devices in the network map can be individually selected, but cannot be otherwise manipulated.

The right-hand pane of the TD-RF Quality Test window displays the results of successful RF Link Quality tests as well as events activating or deactivating the Advanced Features of the TD-RF modems. The most recent entries are added to the top of the display. A maximum of one thousand entries are retained in the display, after which the oldest entries are deleted as new entries are added.



Column Descriptions

Health- Applicable only to an RF Link Quality Test (p. 188), one of three icons will appear in this column providing a graphical indicator of the relative health of the associated RF link

 Normal

 Marginal

 Critical

Timestamp – The server time for when the test response or command acknowledgement was received.

Path (Sender --> Receiver)– Indicates the relative function of the devices involved in the associated test or command.

Packet Size– Applicable only to an RF Link Quality Test (p. 188), this is the size in bytes of the test packet received by the RF modem. See TD-RF Quality Report (p. 189).

Front 2T – Applicable only to an RF Link Quality Test (p. 188), this is the minimum transition point of all T2 transitions within the tolerance window. See TD-RF Quality Report (p. 189).

Back 2T– Applicable only to an RF Link Quality Test (p. 188), this is the maximum transition point of all T2 transitions within the tolerance window. See TD-RF Quality Report (p. 189).

Front 1T – Applicable only to an RF Link Quality Test (p. 188), this is the minimum transition point of all T1 transition within the tolerance window. See TD-RF Quality Report (p. 189).

Back 1T– Applicable only to an RF Link Quality Test (p. 188), this is the maximum transition point of all T1 transitions within the tolerance window. See TD-RF Quality Report (p. 189).

Base Codes – The codes sent to the Base modem to activate the Advanced Features (p. 191) or define an RF path for the RF Link Quality Test (p. 188).

Link Quality – Applicable only to an RF Link Quality Test (p. 188), the Link Quality indicator provides an intuitive means of quickly determining the relative merit of RF Links. If tracked over time, one can easily identify variances and trends in link quality.

The numerical value for Link Quality is derived from the detailed information contained in the TD-RF Quality Report (p. 189) and has a theoretical range of 0 to 102, with greater values equating to greater quality. In practice, the maximum value can only be achieved in a laboratory environment, and the minimum value will never be returned as it is indicative of an inability to decode the test pattern; the RF Link Quality test would timeout. In the real world, typical values for a viable link will fall into mid-range.

The following equation is used to derive the Link Quality value:

Q = [102 – (StdDev(F1T,B1T,F2T,B2T))] * (TestPktSize / (237+NumRptrs)

Where:
Q = Link Quality
StdDev() = Population standard deviation
F1T = Front 1T
B1T = Back 1T
F2T = Front 2T
B2T = Back 2T
TestPktSize = Number of bytes reported in the test packet
NumRptrs = Number of RF repeaters in the complete RF path

Save to File– When this checkbox is checked, all entries in the right-hand pane are also written to a text file in LoggerNet's working directory (C:\Campbellsci\LoggerNet\Logs\RFTestResults.log). Entries in the file are limited to maximum of ten thousand records. After reaching the limit, the oldest two thousand records are deleted.

"1/3/2013 9:20:14 AM" "RFRemote-PB" "RFBase-TD" 239 44 137 65 169 50.92
"1/3/2013 9:20:14 AM" "RFRemote-PB_2" "RFRemote-PB" 239 46 142 61 171 49.22
"1/3/2013 9:20:14 AM" "RFRemote-PB_3" "RFRemote-PB_2" 239 54 136 65 161 56.46
"1/3/2013 9:20:14 AM" "RFRemote-PB_2" "RFRemote-PB_3(6C67)" 137 50 138 63 164 30.75

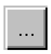
Clear– Clicking this button clears the contents of the right-hand pane.

Start Test– The action taken when this button is activated depends on the device selected in the network map.

With an RFRemote-PB or RFRemote-TD selected: An RF Link Quality Test (p. 188) will execute on the branch of the network map comprising the selected RFRemote-TD/PB and the associated RFBase-TD.

With an RFBase-TD selected: The **RF Base Codes/RFIDs** field is enabled and the entries in this field will determine the action taken. See **RF Base Codes/RFIDs**.

RF Base Codes/RFIDs: - When enabled by selecting an RFBase-TD in the network map, the **RF Base codes** or **RFIDs** entered in this field will determine the action taken when the **Start Test** button is activated.

> **RF Base codes:** Entering a supported command string of numeric values, separated by a space or comma, will activate the associated Advanced Features (p. 191) in the RFBase-TD or specified RFRemote-TD/PB.
>
> **RFIDs:** Entering one or more RFIDs, separated by a space or comma, defining an RF path will initiate an RF Link Quality Test (p. 188) on the specified path. This feature allows one the option of testing an RF path that, with the exception of the selected RFBase-TD, may or may not be composed of devices configured in the network map.
>
>  When enabled by selecting an RFBase-TD in the network map, the Browse button will open the **Advanced RF Commands** window for selecting and executing Advanced Features (p. 191) in the RFBase-TD or specified RFRemote-TD/PB.

7.5.1.1 RF Link Quality Test

Fundamentally, an RF Link Quality test provides an indication of how well the specific data patterns of an RF test packet have been decoded by each of the TD-RF modems in the specified RF path. This information, in turn, is indicative of the relative quality (i.e. Signal to Noise Ratio) of the RF signal received by the associated FM transceivers.

Before executing an RF Link Quality test, one must first specify the RF path to be tested. This can be done in one of two ways. If all the devices in the RF path to be tested are configured in the network map, simply select the RFRemote-TD/PB that represents the end of the path. (Right-clicking the end of path RFRemote-TD/PB will simultaneously select the RF path and launch a popup window for selecting **Start Test**.) If one or more of the RFRemote-TD/PB devices in the path to be tested is not configured in the network map, one must specify the RF path by first selecting the RFBase-TD that is configured in the network map and then manually entering the RFIDs of each TD-RF modem along the path in the **RF Base Codes/RFIDs** field. (see **RF Base Codes/RFIDs**)

Once the RF path has been designated, the RF Link Quality test is initiated by clicking the **Start Test** button.

LoggerNet responds by sending an RF Test command packet, containing a definition of the RF path to be tested, to the RFBase-TD. The RFBase-TD executes the RF Link Quality test by sending an RF Test Packet across the network to the RFRemote-TD/PB at the end of the RF path. As the end of path modem receives the RF Test Packet, it internally logs a **TD-RF Quality Report** detailing information about the size of the packet and its ability to decode the data contained in the packet. (For details, see TD-RF Quality Report (p. 189).) The end of path modem will then send the RF Test Packet back across the network toward the RFBase-TD. As the RF Test Packet makes the return trip across the network, each RFRemote-TD/PB modem in the path as well as the RFBase-TD will log a **TD-RF Quality Report** based on the received packet. The RFBase-TD then sends a command across the network to collect the **TD-RF Quality Report** from each of the modems involved in the test. The RFBase-TD sends the collected reports to LoggerNet in the RF Test response packet.

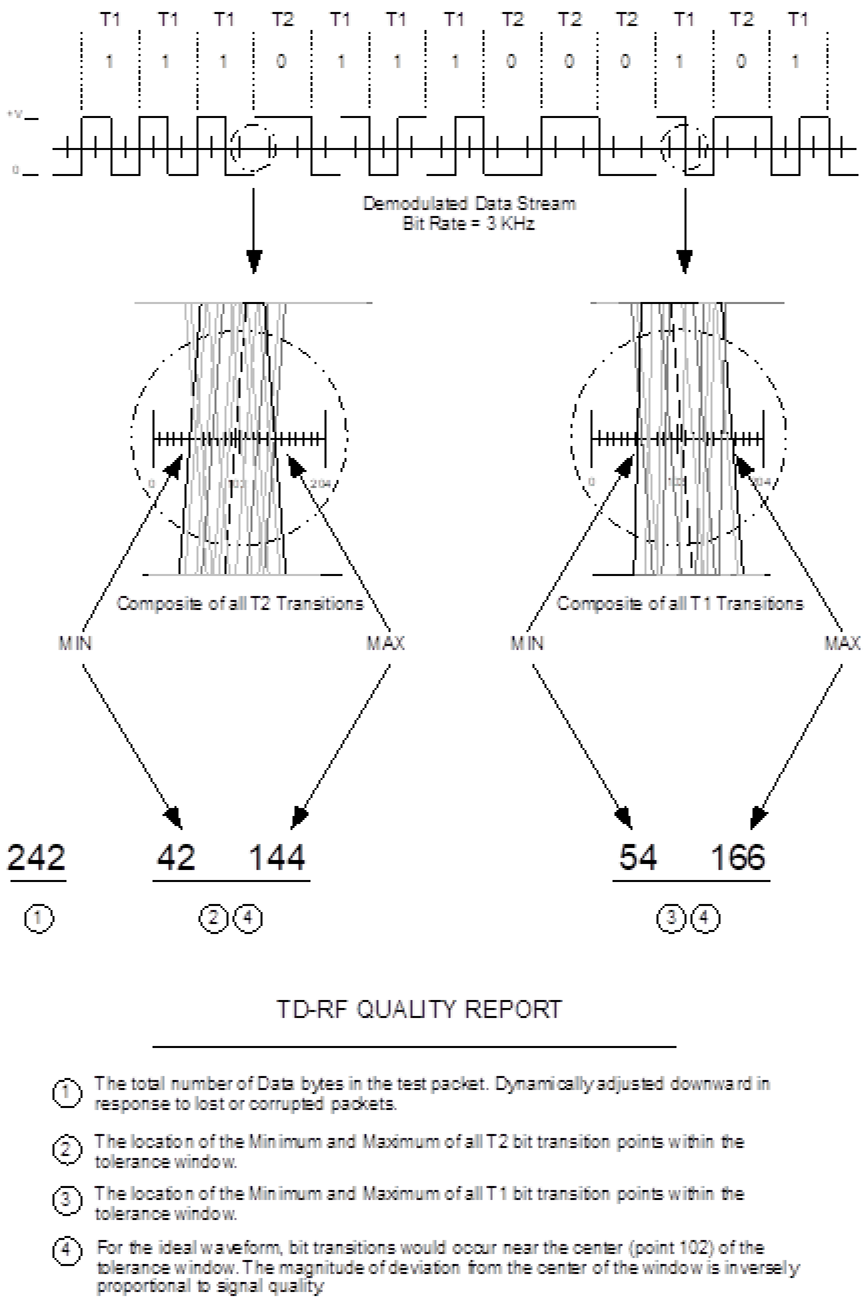
The **TD-RF Quality Reports** are displayed in the right-hand pane of the TD-RF Quality Test window in the order (top-down) they were collected; starting with the end of path modem and ending with the RFBase-TD.

7.5.1.2 TD-RF Quality Report

An RF modem primarily functions as an interface bridging two distinctly different modes of data transport; the wired medium of dataloggers and PCs, and the wireless medium of RF transceivers. The RF modem accomplishes this by converting the serial data stream from the wired medium into a waveform of proper amplitude and frequency for driving the FM modulator circuits of the RF transceiver, and vice versa.

In a process known as line coding, the RF modem encodes the binary data from the wired data stream onto a 3 KHz waveform. The line coding utilized in CSI's TD-RF modems is called Miller Encoding. This encoding scheme employs a method of differentiating the binary "1s" and "0s" of the data stream based on the timing of transitions in the waveform from one level to another within the bit period (approximately 333 microseconds for a 3 KHz bit rate). A binary "1" is represented by a level transition occurring in the middle of the bit period. A binary "0" is represented by either there being no transition occurring within the bit period or, in the case of consecutive "0"s, the transition occurs at the end of the bit period.

In order to properly decode the encoded data from a received signal, one must precisely detect when the level transitions are occurring in relation to the middle and end of the bit period; the T1 and T2 transition points respectively. The OS in the RF modem does this by establishing a detection window centered about the time a transition is expected to occur. The detection window is 204 units wide, so the optimal transition timing would occur in the center of the window; unit 102. In the real world, the optimal transition timing is not likely to occur that often so there will always be some deviation to where the transitions occurs within the detection window. The more noisy the received signal, the greater the deviations. If the transition occurs outside the detection window, the proper bit will not be detected and a data error will occur. The demodulated data stream and the associated detection windows for the T1 and T2 transition points are illustrated in the graphic below.



The information recorded in the TD-RF Quality Report includes the location of the maximum and minimum transition point for the T1 and T2 detection windows and the size, in bytes, of the received test packet.

As has been mentioned, the magnitude of the T1 and T2 transition point deviations from the center of the detection window is directly related to the level of noise in the received signal. The level of noise relative to the level of the desired information in the received signal is known as the Signal to Noise Ratio (SNR) and is a prime metric of received RF signal quality; the greater the level of desired information relative to the level of noise in the received signal, the greater SNR and the greater the quality of the received signal.

The test packet size is significant as an indicator of lost packets. The over the air (OTA) communications protocol utilized by the RF modems requires that a modem acknowledge the reception of an RF Test Packet. If the sending modem does not receive an acknowledgment, it will resend the packet. This is known as a 'retry'. After executing a number of unacknowledged retries, the sending modem will decrease the number of bytes in the packet by approximately half before attempting additional retries. Therefore a decrease in the size of the test packet is another indication of less than optimal link quality.

The test packet size, as calculated by the TD-RF Quality Report, includes the packet header. The packet header contains the RFIDs of each modem in the RF path being tested. Therefore the packet size will be a minimum of 237 for an RFBase-TD and a single RFRemote-TD/PB, and will increase by one for each additional RFRemote-TD/PB in the path.

7.5.1.3 Advanced Features

The Advanced Features of the TD-RF modems include functionality that may be activated for network diagnostics and/or making adjustments to the Time Division Polling process.

WARNING: The misapplication of some Advance Features could severely degrade the performance of the TD-RF Network. It is recommended that these features be activated only at the direction of a Campbell Scientific Applications Engineer.

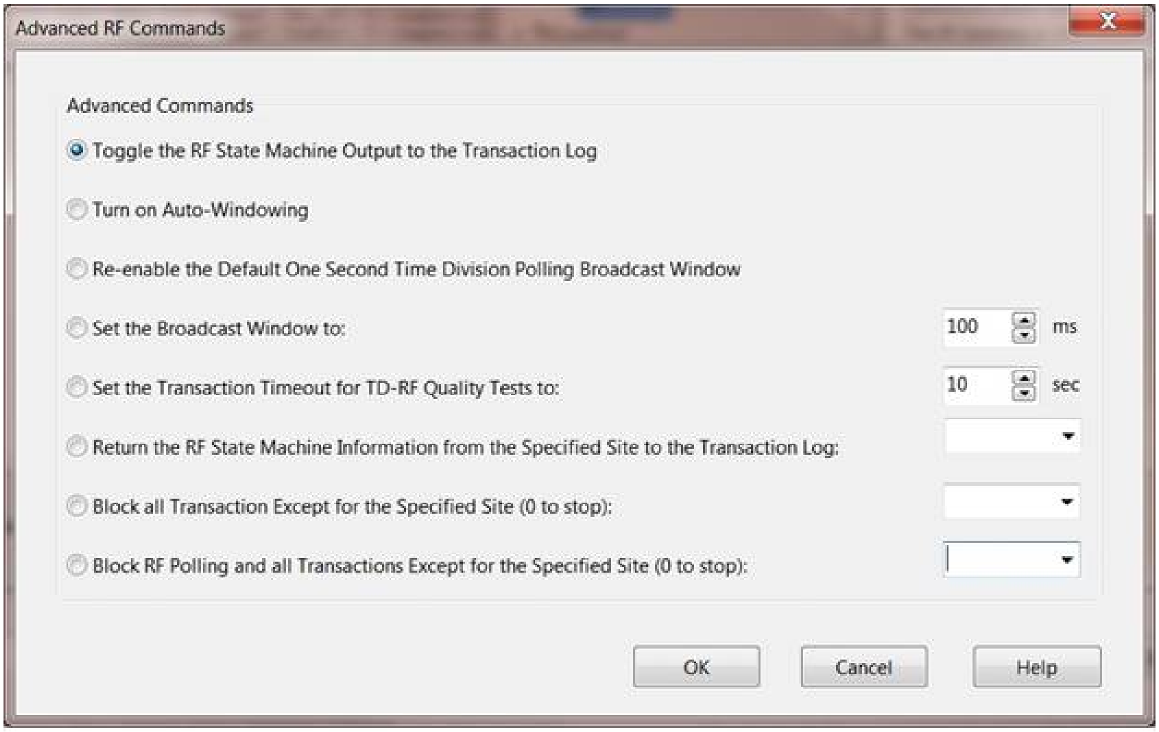
It should be noted that the activation of the Advanced Features is not persistent; when the modem is reset (i.e. power removed), the functionality will be discontinued. Additionally, while all of the Advanced Features are available in TD-RF versions of the RF500M (-PB and -AL OS options), some features are not available in early versions of the TD-RF PROMs utilized in the RF95 and RF310M modems.

Activation of the Advanced Features is accomplished by one of two methods:

Activation Method 1 – In the TD-RF Quality test window, select the RFBase-TD in the network map, enter the appropriate code string in the **RF Base Codes/RFIDs** field and click the Start Test button.

NOTE: Enter the Code string with a 'space' separating each value.

Activation Method 2 – In the TD-RF Quality test window, select the RFBase-TD in the network map and click the browse button to the right of the **RF Base Codes/RFIDs** to open the Advanced RF Commands window. Select the feature to be activated using the radio buttons, enter/select any ancillary information that may be required and click the OK button. The **RF Base Codes/RFIDs** field will be populated with the appropriate code string and the feature activated.



Advanced Feature Functionality:

Toggle the RF State Machine Output to the Transaction Log:– Activating this feature will cause the RFBase-TD's state machine output to be written to LoggerNet's Transaction log. The state machine output will continue to be written to the Transaction log until the feature is deactivated by repeating the activation process.

Code string – 0
Availability – PROM version 6873-14 and later; RF500M (-PB and -AL OS options)

Turn on Auto-Windowing – Activating this feature will enable the RFBase-TD to dynamically adjust the amount of time allotted to each RFRemote-TD/PB for responding to the Time Division Polling broadcast.

Code string – 0 1
Availability – PROM version 6873-16 and later; RF500M (-PB and -AL OS options)

Re-enable the Default One Second Time Division Polling Broadcast Window – Sets amount of time that the RFBase-TD allocates to each RFRemote-TD/PB for responding to the Time Division Polling broadcast to one second.

Code string – 0 2
Availability – PROM version 6873-16 and later; RF500M (-PB and -AL OS options)

Set the Broadcast Window to: - Sets the amount of time that the RFBase-TD allocates to each RFRemote-TD/PB for responding to the Time Division Polling broadcast to a value of between 100 milliseconds and 2.55 seconds in steps of 10 milliseconds resolution.

Code string – 0 3 X : Where X is a value of 1 to 255 (10ms resolution)
Availability – PROM version 6873-16 and later; RF500M (-PB and -AL OS options)

Set the Transaction Timeout for TD-RF Quality Test to: - Sets the maximum amount of time (Lifetime) that the RFBase-TD allows for the RF Link Quality Test (p. 188) to be completed. If the TD-RF Quality Report (p. 189)s have not been returned to the RFBase-TD before the Lifetime has elapsed, the RFBase-TD will report to LoggerNet that the RF Link Failed. The Timeout can be set to a value of between 5 and 255 seconds.

Code string – 0 4 X : Where X is a value of 5 to 255
Availability – PROM version 6873-18 and later; RF500M (-PB and -AL OS options)

> **NOTE:** The transaction Lifetime reverts to a default value of 120 seconds ***only*** when the RFBase-TD modem is reset.

Return RF State Machine Information from the Specified Site to the Transaction Log: - Activating this feature will cause the selected RFRemote-TD/PB to send approximately 2 KB of its most recent state machine output to the RFBase-TD. The information is subsequently written to LoggerNet's Transaction log.

Code string – RFID … RFID 0 : Where RFID identifies the RFRemote-TD/PB addresses in the RF Path to the selected RFRemote-TD/PB, separated by spaces.
Availability – PROM version 6873-18 and later; RF500M (-PB and -AL OS options)

Block all Transactions Except for the Specified Site – When this feature is activated, the RFBase-TD will block all LoggerNet initiated transactions to any RFRemote-TD/PB in the network other than the one specified. Time Division Polling will continue for all sites. This feature is automatically deactivated after 30 minutes. To deactivate the feature manually, re-initiate the feature with 0 as the RFID of the specified site.

Code string – 0 5 RFID : Where RFID is the address of the specified RFRemote-TD/PB
Code string – 0 5 0 : Code to deactivate
Availability – PROM version 6873-18 and later; RF500M (-PB and -AL OS options)

> **NOTE:** If LoggerNet attempts to communicate with a blocked site (clock check/set, data collection, program send, etc.) the RFBase-TD will respond with a Delivery Fault message. This, in turn, will cause LoggerNet to immediately 'fail' the transaction and flush any pending transactions to this site from the queue. Failure messages and/or indicators will appear in the Status Monitor, Troubleshooter, and Log Tool screens for the associated devices.

Block RF Polling and all Transactions Except for the Specified Site – When this feature is activated, the RFBase-TD will block all LoggerNet initiated transactions to any RFRemote-TD/PB in the network other than the one specified. **Time Division Polling will be suspended for all sites**. This feature is automatically deactivated after 30 minutes. To deactivate the feature manually, re-initiate the feature with 0 as the RFID of the specified site.

Code string – 0 6 RFID : Where RFID is the address of the specified RFRemote-TD/PB
Code string – 0 6 0 : Code to deactivate
Availability – PROM version 6873-18 and later; RF500M (-PB and -AL OS options)

NOTE: If LoggerNet attempts to communicate with a blocked site (clock check/set, data collection, program send, etc.) the RFBase-TD will respond with a Delivery Fault message. This, in turn, will cause LoggerNet to immediately 'fail' the transaction and flush any pending transactions to this site from the queue. Failure messages and/or indicators will appear in the Status Monitor, Troubleshooter, and Log Tool screens for the associated devices.

7.6 LoggerNet Server Monitor

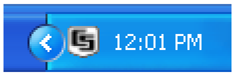
The LoggerNet Server Monitor is a utility that is used to monitor the status of LoggerNet when it is being run as a service or being run on a remote computer. Once started it can be minimized, where it resides in the Window's system tray. An icon, which changes color depending upon network status, provides a visual indication of communication with the dataloggers in the network and whether or not the LoggerNet server is running. Multiple instances of the Server Monitor can be started to monitor multiple LoggerNet servers running on remote computers.

The Server Monitor is started from the Window's Start menu, LoggerNet Server Monitor. When first opened, a Login dialog box appears. This dialog is used to specify the name or address of the computer running the LoggerNet server that you want to monitor. If the server is running on your local computer, use the default name of LocalHost. Otherwise, enter the valid name or IP address of the remote computer running LoggerNet. If security is enabled, you will need to type in a user name and password.

NOTE: The LoggerNet Server Monitor can be run by a user assigned any one of the five levels of security.

Once you are connected to a LoggerNet server, the main window of the Server Monitor will be displayed. This window shows messages related to the activity of the LoggerNet server that are written to the Comm.log. By default, only Warning and Fault (failure) messages are displayed. However, you can choose to monitor Status messages as well by selecting the **Options** > **Show All**menu item. When the Show All option is enabled, a check mark will appear to the left of the menu item. This option is a toggle. Select it once to enable it; select it a second time to disable it.

When the LoggerNet Server Monitor is minimized, a Campbell Scientific icon appears in the Windows systems tray:



The icon has three states:

	(white background/black text) The LoggerNet server is running, and no Warnings or Faults have been encountered.
	(white background/blue text) The LoggerNet server is running, but Warnings or Faults have occurred. There have likely been communication problems with a device in the datalogger network.
	(white background/red text) The LoggerNet server is no longer running.

If Warnings or Faults have been encountered, you can reset the state of the icon by right-clicking it and choosing **Reset**, or by opening the Server Monitor and choosing **Options** > **Reset**. If you select **Options** > **Clear – Reset**, the messages will be cleared from the message window as well.

> **NOTE:** To increase or decrease the font of the message display, select the Font + or Font – option from the Font menu.

8. Creating and Editing Datalogger Programs

Dataloggers must be programmed before they can make measurements. LoggerNet 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.

8.1 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.

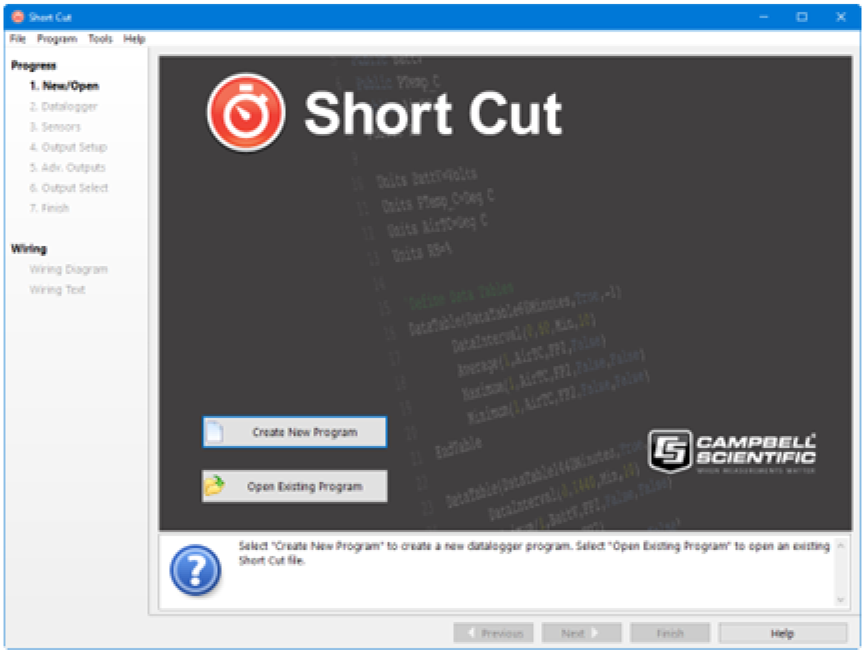
Short Cut cannot be used to edit existing Edlog, CRBasic, or Short Cut for DOS programs. Program editing and more complex datalogger programming functions should be accomplished using our Edlog or CRBasic Editor programming tools.

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.

8.1.1 Creating a Program Using Short Cut

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

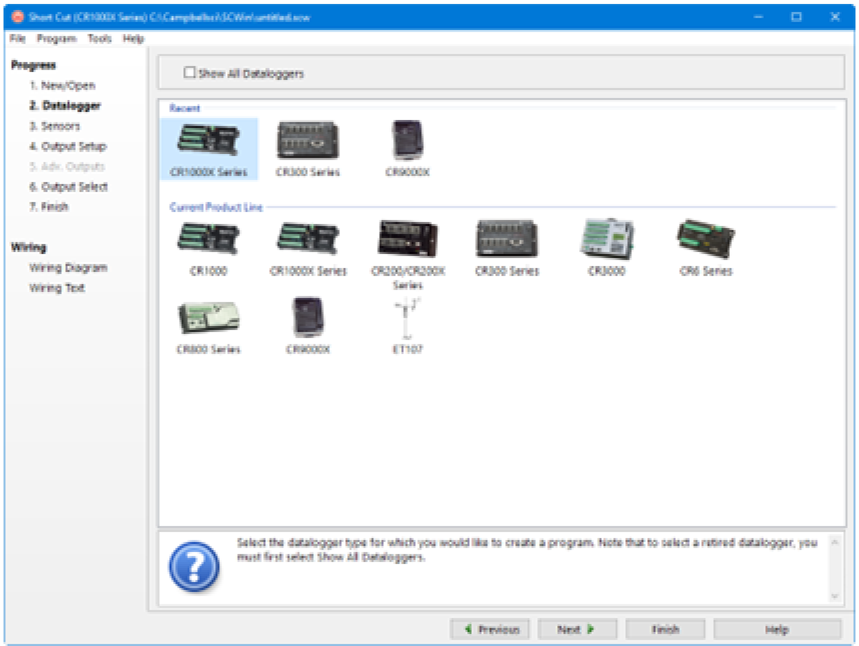


8.1.1.1 Step 1 – Create a New File or Open Existing File

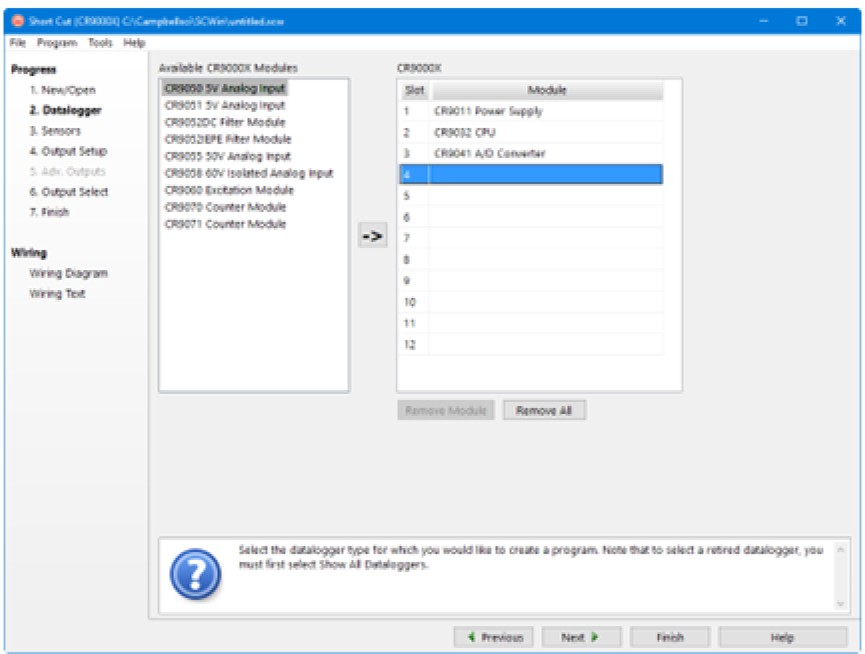
To begin creating a new program, press **Create New Program**. To open an existing program, press **Open Existing Program** and select a file from the resulting browser window.

8.1.1.2 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**.

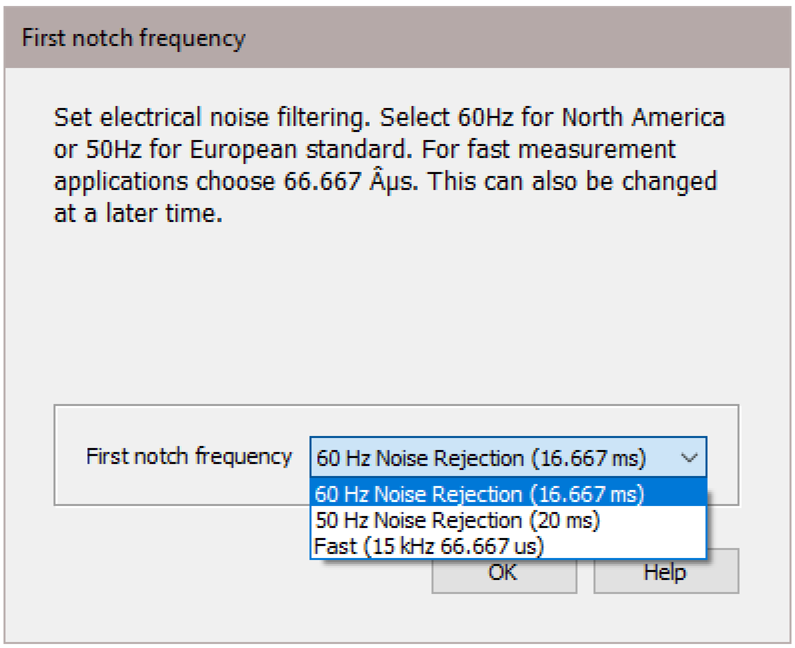


From this screen, you indicate which CR9000X modules are inserted into which CR9000X slots. To add a module, select the module by clicking on it in the **Available CR9000X Modules** list, select the **Slot** by clicking on the slot number, then press the arrow key.

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.

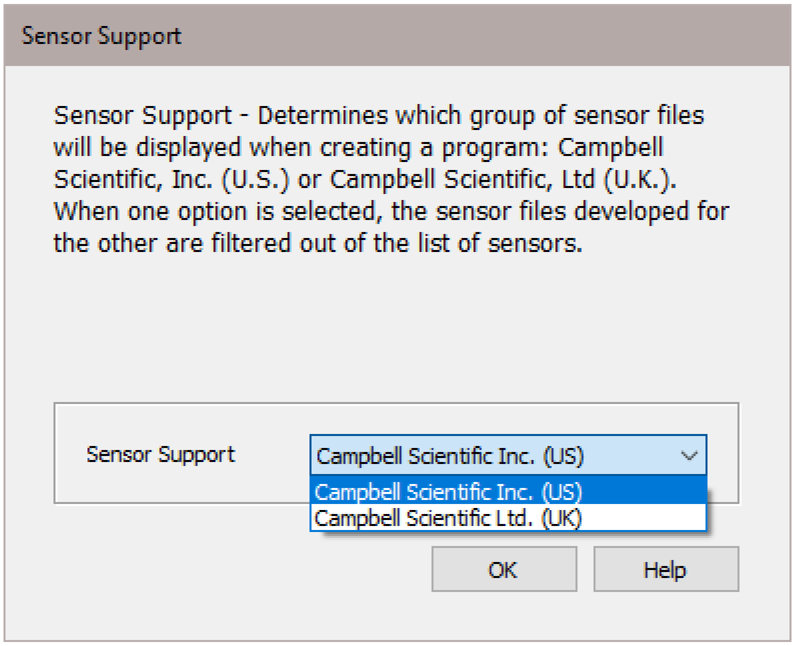
The next dialog box that is displayed is used to select the **First notch frequency** to apply to the measurements in the program. This can be used to filter out AC signals that might affect the accuracy of your measurements (such as noise from fluorescent lighting or a generator). Typically 60 Hz rejection 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.

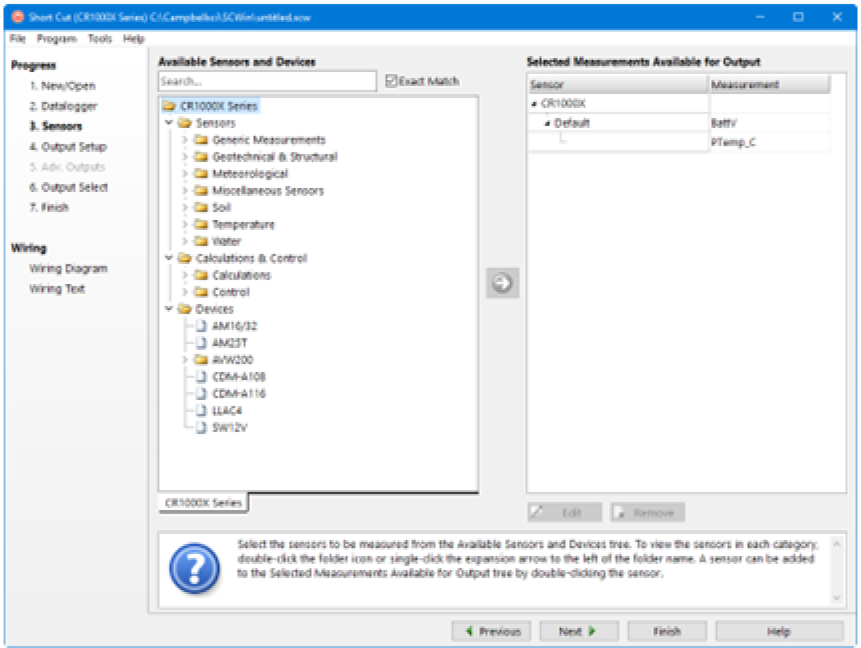
> **NOTE:** The **First notch frequency** and the **Sensor Support** settings are persistent settings for each datalogger model. The first time you create a program for a particular datalogger model, you will be presented with these two dialog boxes. The state of these settings is saved between Short 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".

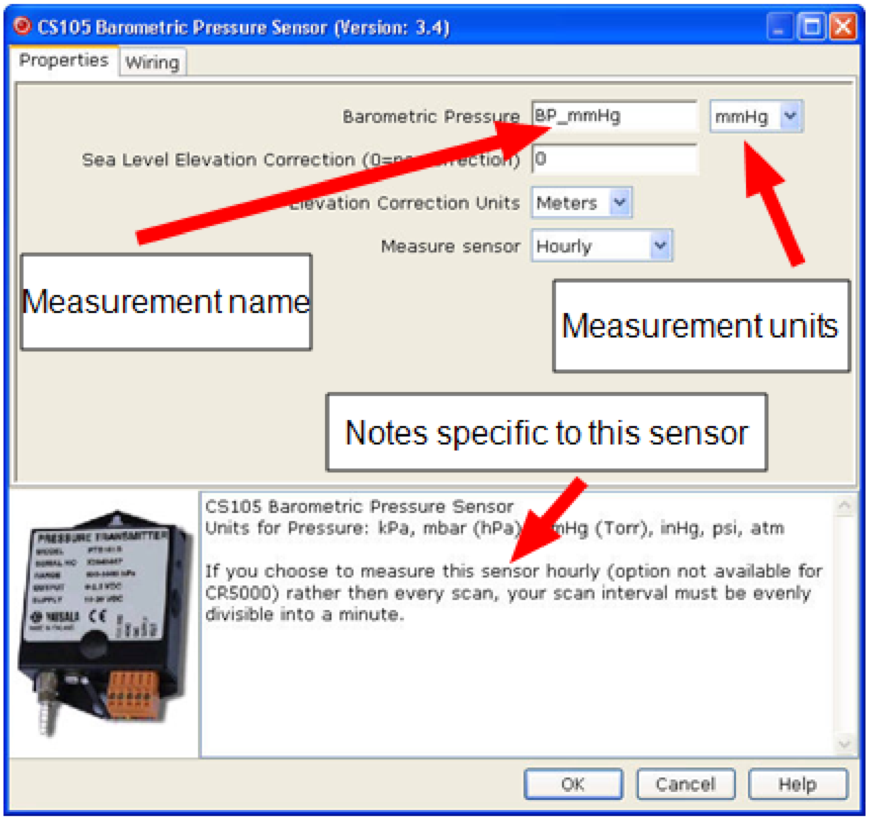
8.1.1.3 Step 3 – Choose Sensors to Monitor

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



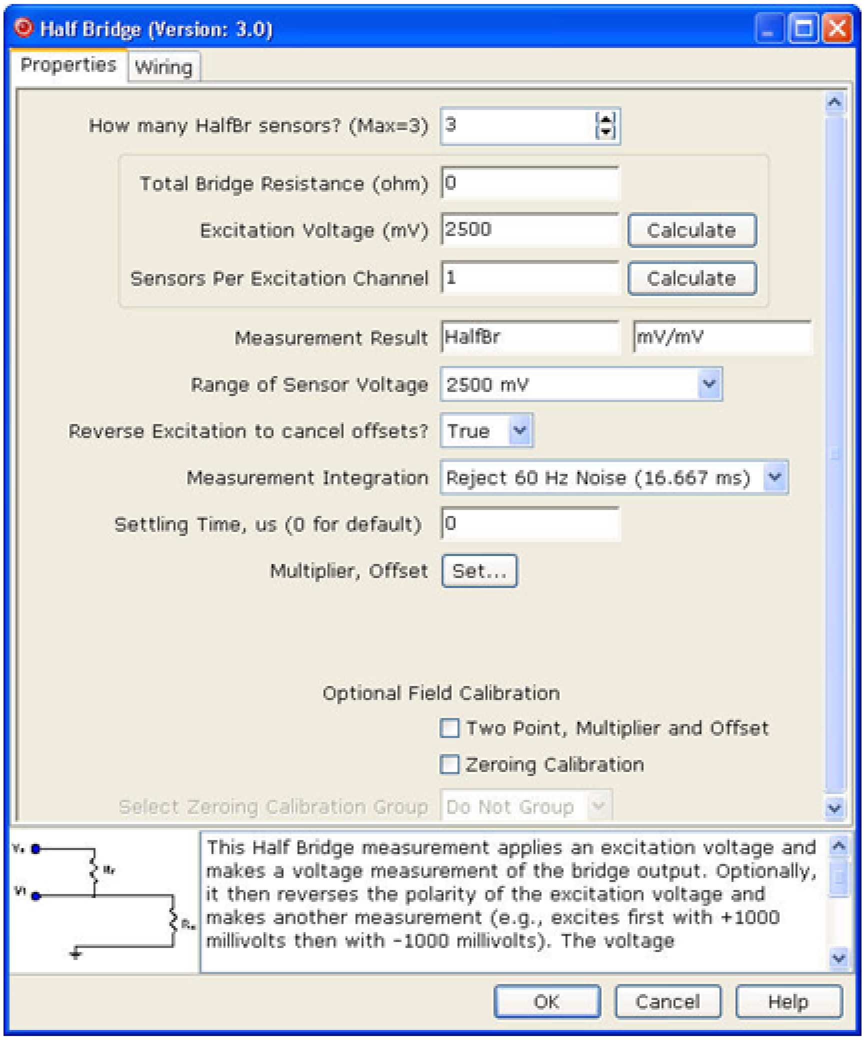
Some major groups have subgroups. Double-clicking the Meteorological group folder shows several subgroups of meteorological sensors. Double-click a subgroup to show the available sensors. Refer to the documentation for your sensors for the name of the sensors you have. If your sensor is not shown, you 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.

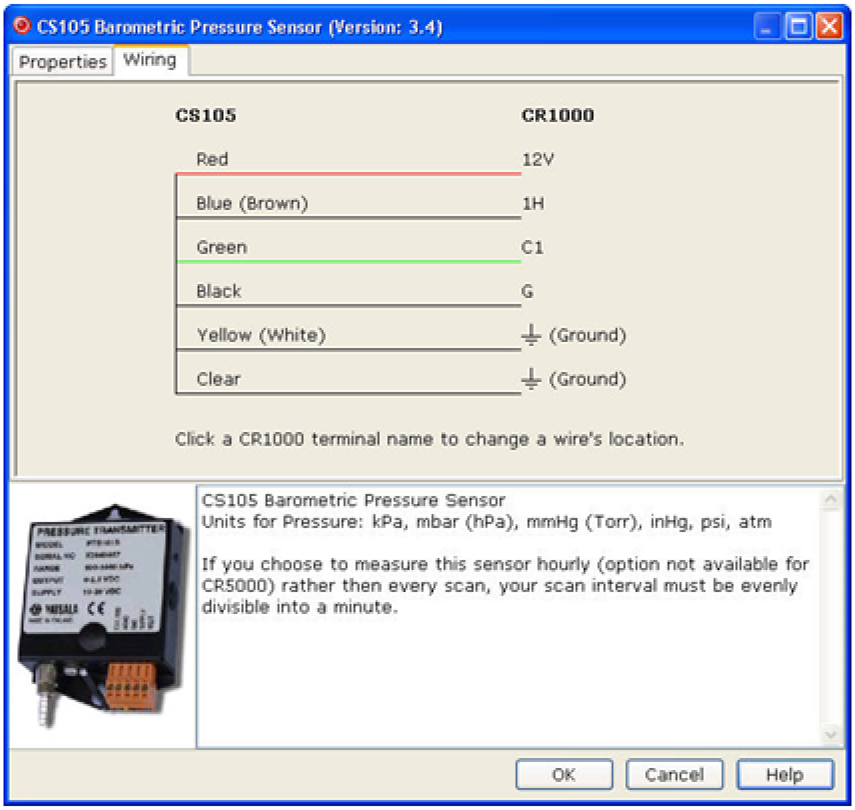


Note that this sensor not only offers a custom name field and units, but also allows you to correct for sea level, a common practice in measuring atmospheric pressure. In the middle of the screen, look over the notes (or refer to the Help for this sensor), for this sensor may require other sensors or 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.



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).



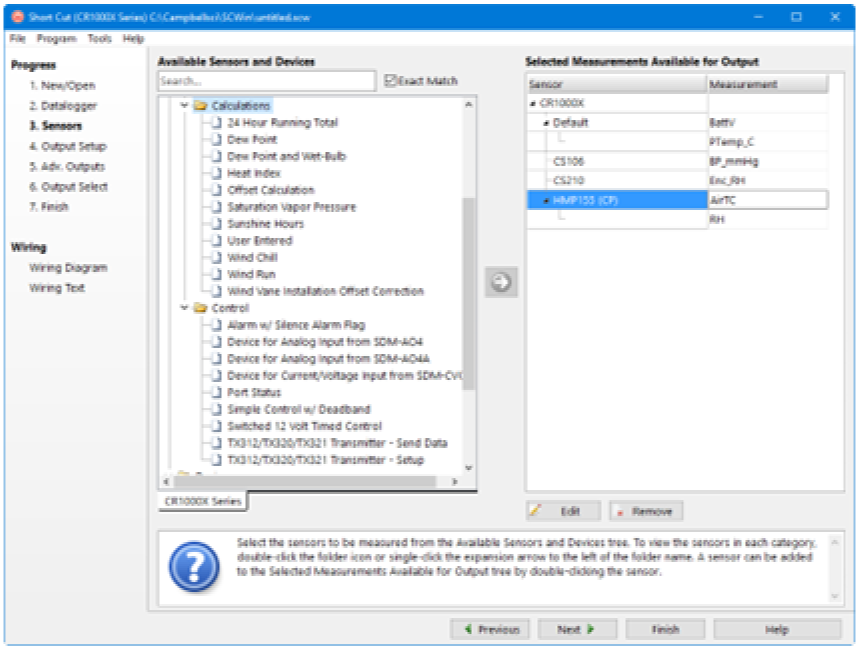
Each wire's caption/color is shown on the left side of the wire. The location where the wire will be connected to the device is shown on the right side (under the device). You can change a caption/color by clicking on the caption/color label. A wiring location can also be changed by clicking on the wiring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

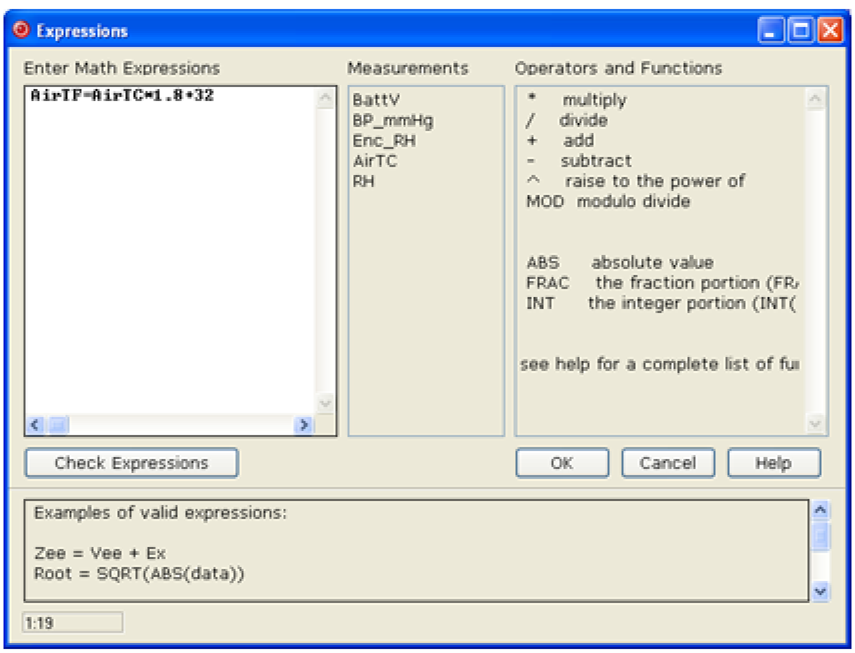
At any time, you may choose a measurement label on the right side of the Sensors screen and edit it or remove it.

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



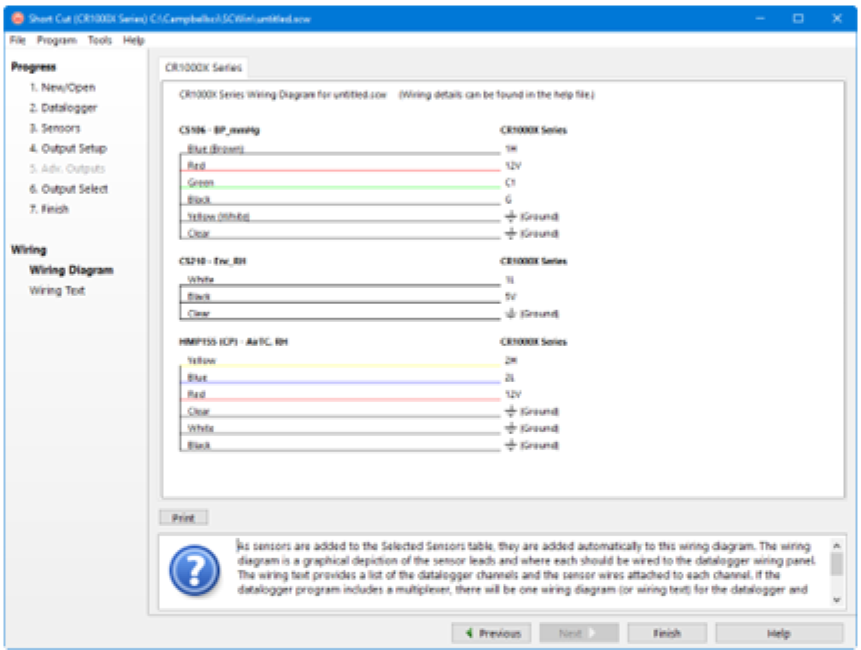
Some of these calculations may require additional sensors, or sensor measurements stored in particular units. See the help for each calculation to determine the necessary inputs. Note that there is also a User Entered calculation available in the Calculations folder. With it you can enter your own custom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

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

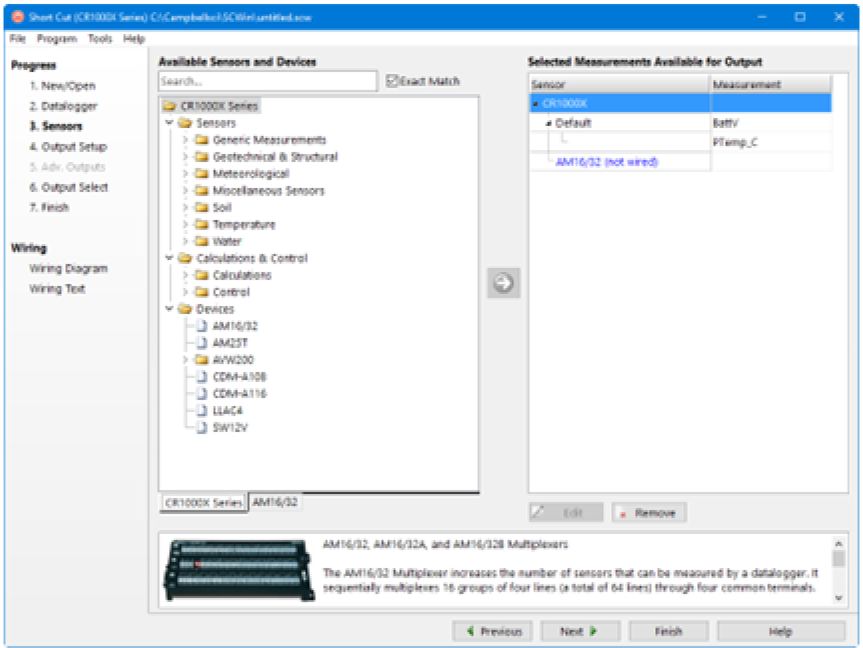


Refer to the online help for complete information on creating User Calculation.

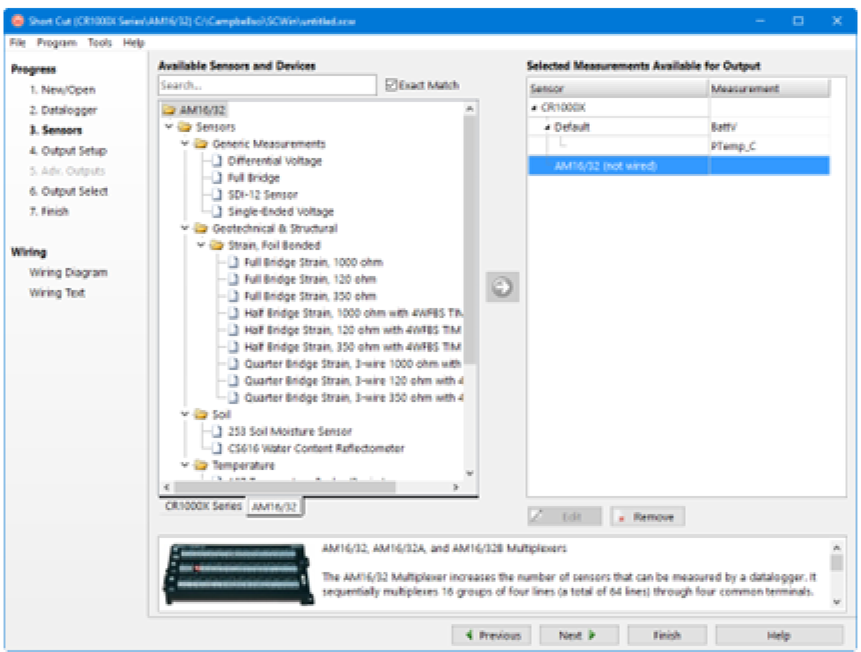
Short Cut provides you with a wiring diagram by clicking on **Wiring Diagram** on the left side of the Sensors window. In the example below, Short Cut was told to measure a CS106 Barometric Pressure sensor, a CS210 enclosure relative humidity sensor, and an HMP155 Air Temperature and Relative Humidity 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:



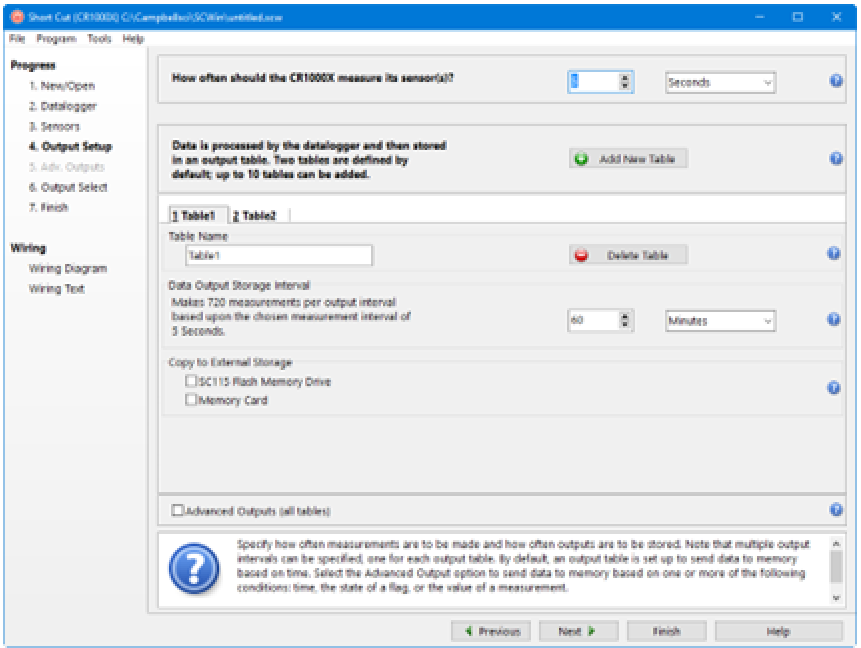
You can then add sensors to that device just as you would to the main datalogger.

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**.

8.1.1.4 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.

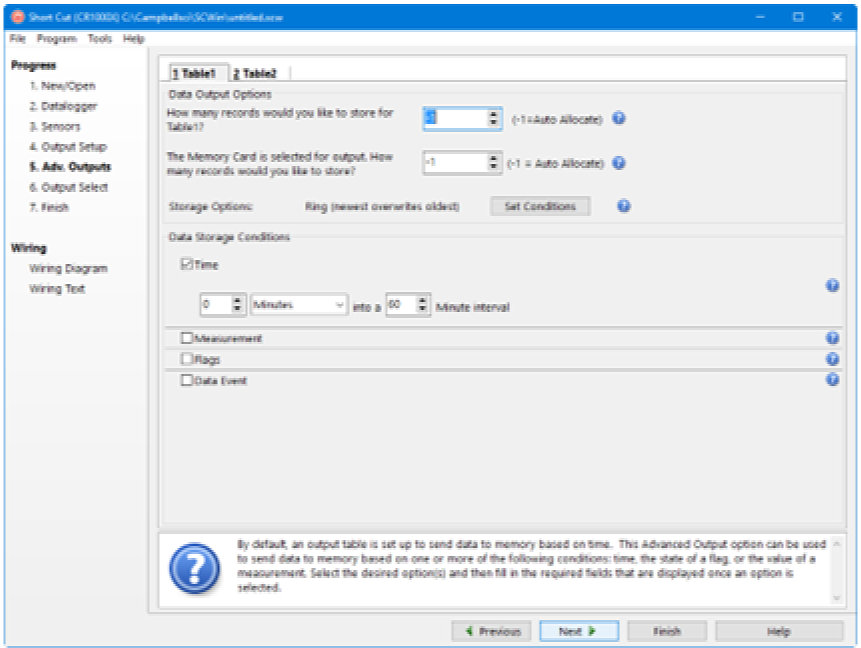
Two tables are defined by default. Additional tables can be added by pressing the A**dd Table** button. Short Cut limits the number of output tables to 10. An output table can be removed by clicking on the table to make it the active table and pressing the **Delete Table** button.

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.)

8.1.1.5 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 O**utput 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).

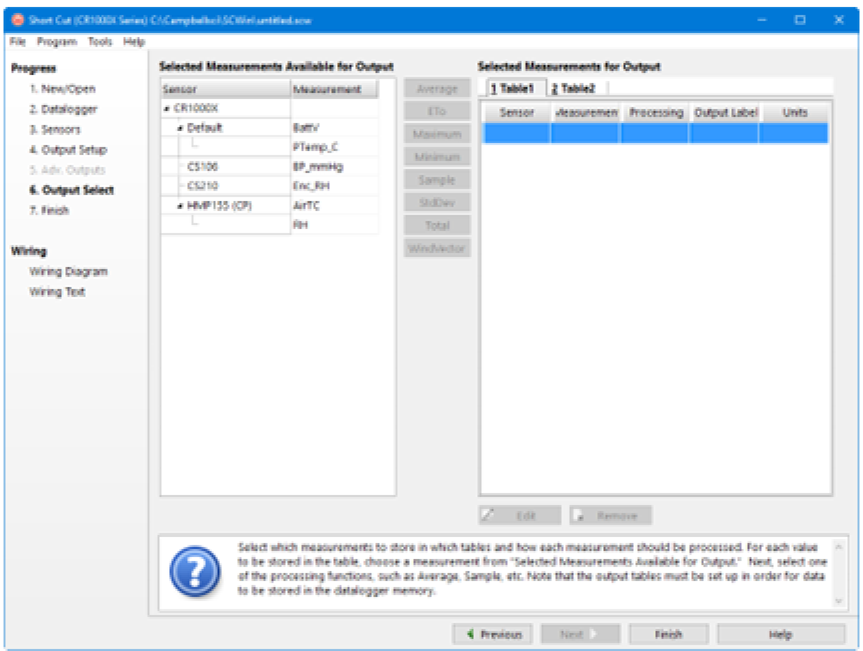
  - The **Records Before** field is used to enter the number of records that should be stored prior to the condition being met (the datalogger will keep this number of records in memory). The **Trigger** is the variable that will be monitored for the specified condition. Use the drop-down list box to select 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.

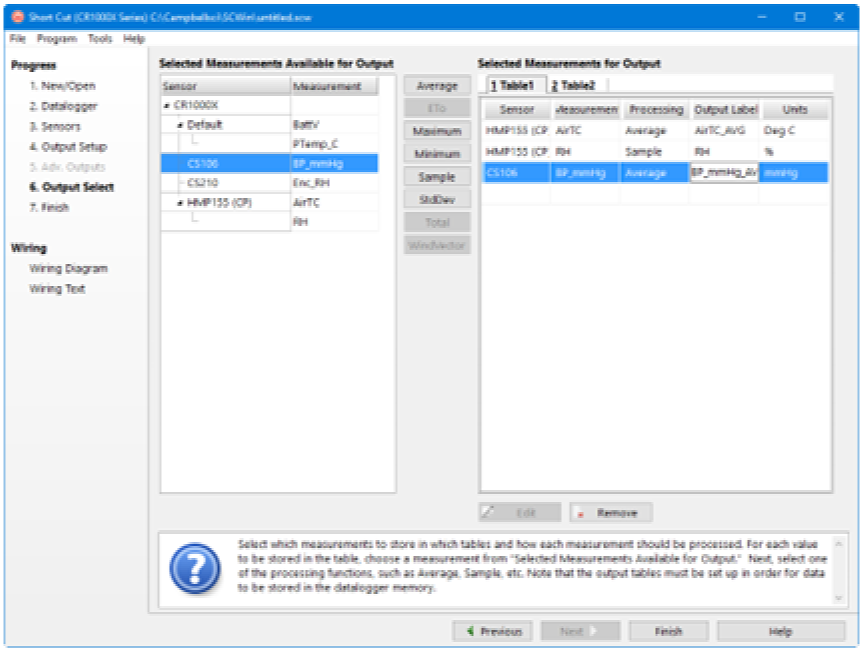
8.1.1.6 Step 6 – Select Outputs

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.

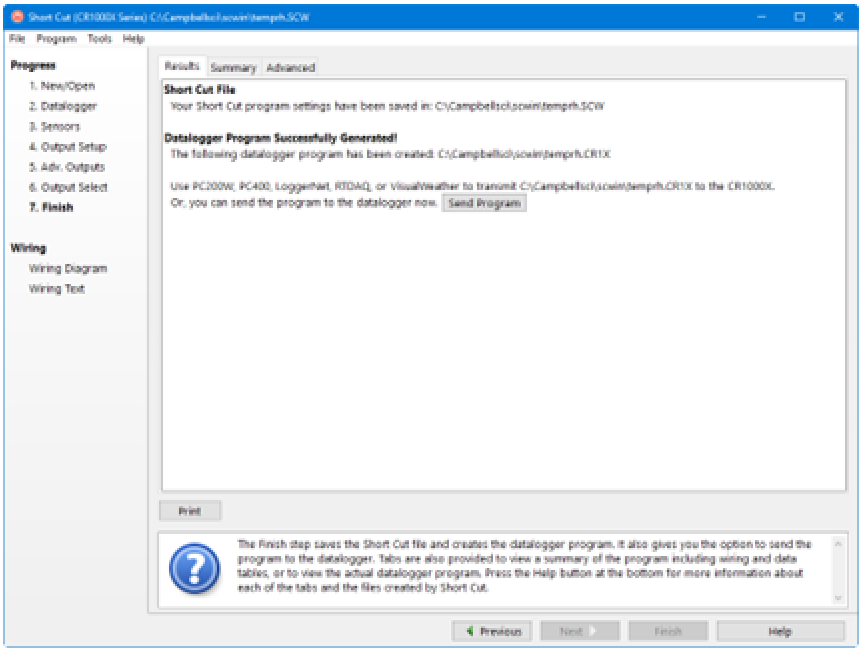
An **Output Label** can be changed by selecting it and making the desired modifications.

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**.

8.1.1.7 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, P*rogramName*.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.)

8.1.2 Short Cut Settings

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

8.1.2.1 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.

8.1.2.2 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.

8.1.2.3 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.

8.1.2.4 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.

8.1.2.5 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.

8.1.2.6 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.

8.1.2.7 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.

8.1.2.8 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.

8.1.2.9 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. 218).

8.1.3 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.

8.1.4 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

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.

8.1.5 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.

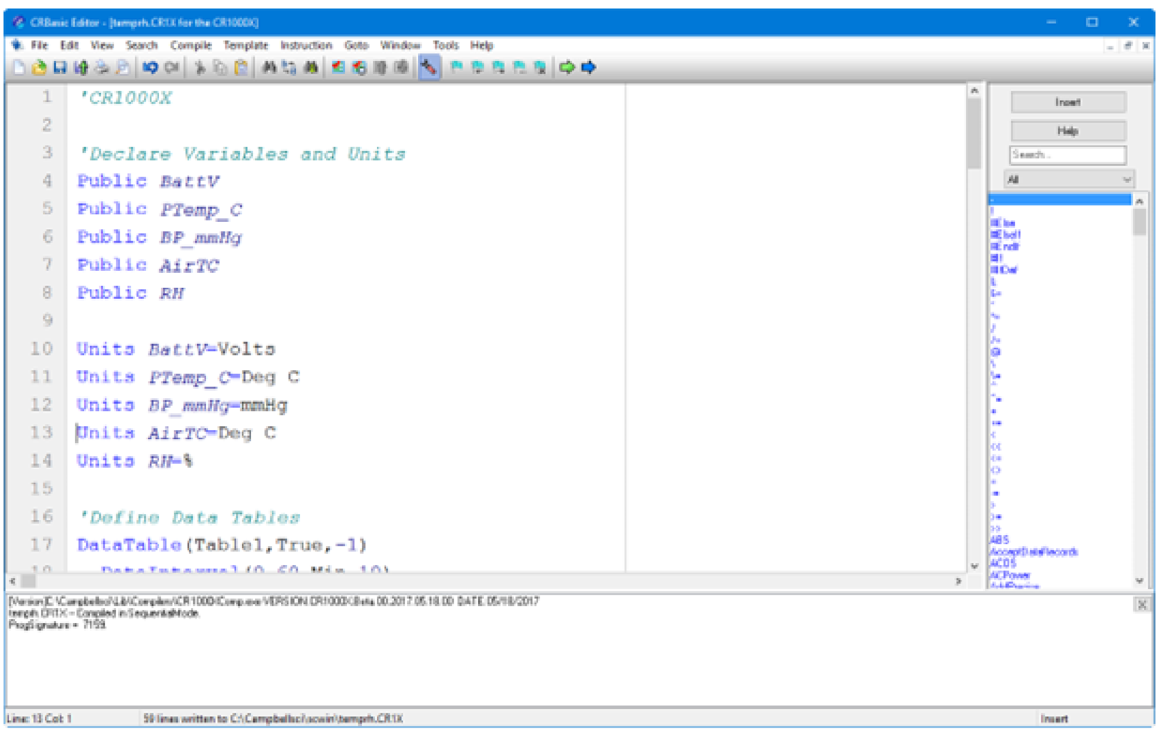
The resulting dialog box will allow the user to make changes to the chosen sensor file and then save it with a new name. (See Short Cut's Online Help for additional information on changes that can be made.) By default, custom sensor files will be created in C:\CampbellSci\SCWin\SENSORS, which is a different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

8.2 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.



8.2.1 Inserting Instructions

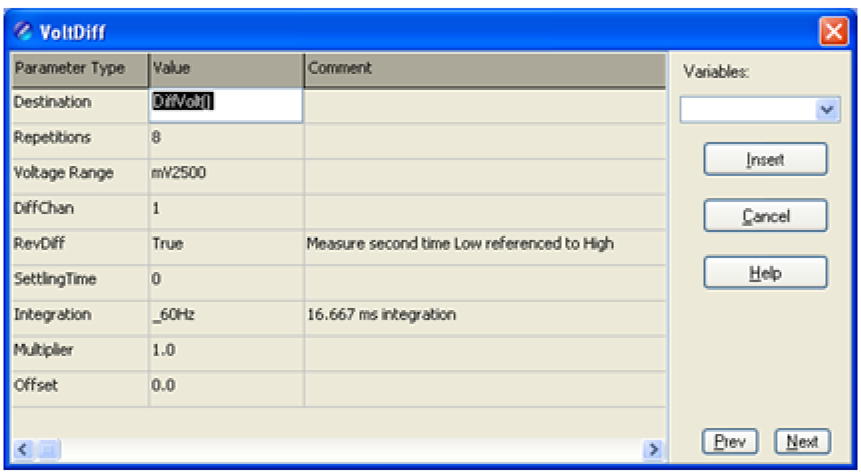
An instruction can be easily inserted into the program by highlighting it in the Instruction Panel list and pressing **Insert** or by double-clicking the instruction name. If an instruction has one or more parameters, an instruction dialog box will be displayed to facilitate editing the parameters. Complete 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.

8.2.1.1 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.

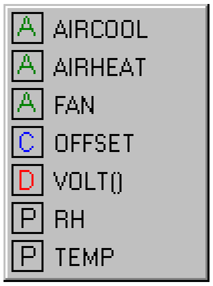
Below is an example of the **Parameter** dialog box for the differential voltage instruction (*VoltDiff*).



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.

8.2.1.2 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.

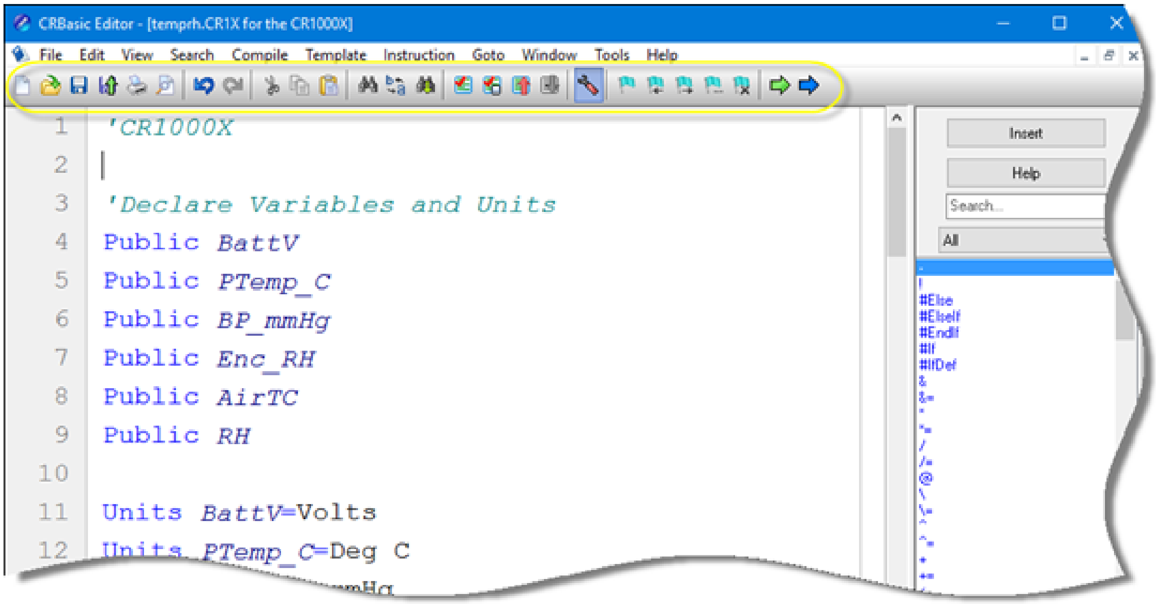
Right-click another type of parameter to bring up help for that parameter.

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 chose 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 indention 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.

8.2.2 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.

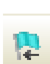
Save and Compile – Saves and then compiles the opened file.

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

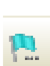
Next Error – Moves the cursor to the part of the program where the next error was identified.

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.

8.2.2.1 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 Erro**r button or choose P**revious 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.

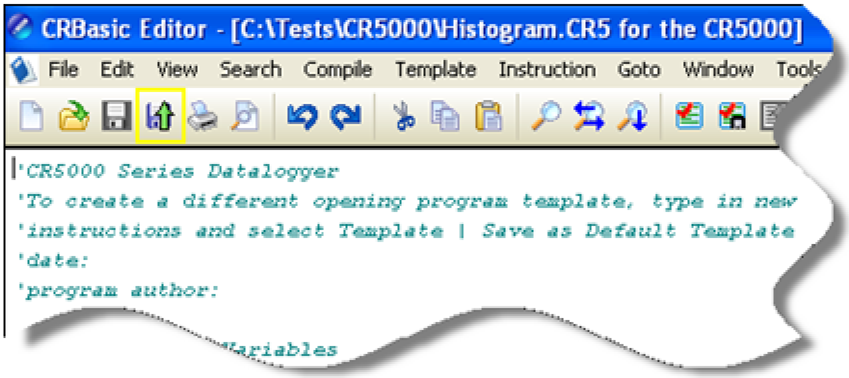
The error window can be closed by selecting the **Close Message Window** menu item from the **View** menu, or by clicking the **X** in the upper right corner of the message window.

> **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.

8.2.2.2 Compile, Save, and Send

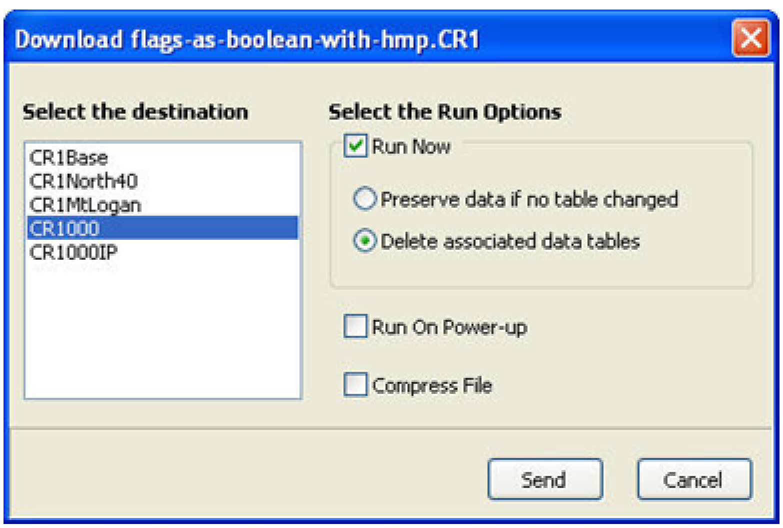
The CRBasic Editor allows you to send a program to a datalogger that has already been defined on the network map in LoggerNet, PC400, or RTDAQ. This only works if LoggerNet, PC400, or RTDAQ is running at the time you attempt to send the program.

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.

8.2.2.3 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. 234)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.

8.2.2.4 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.

8.2.2.5 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**.

8.2.2.6 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.)

8.2.2.7 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.

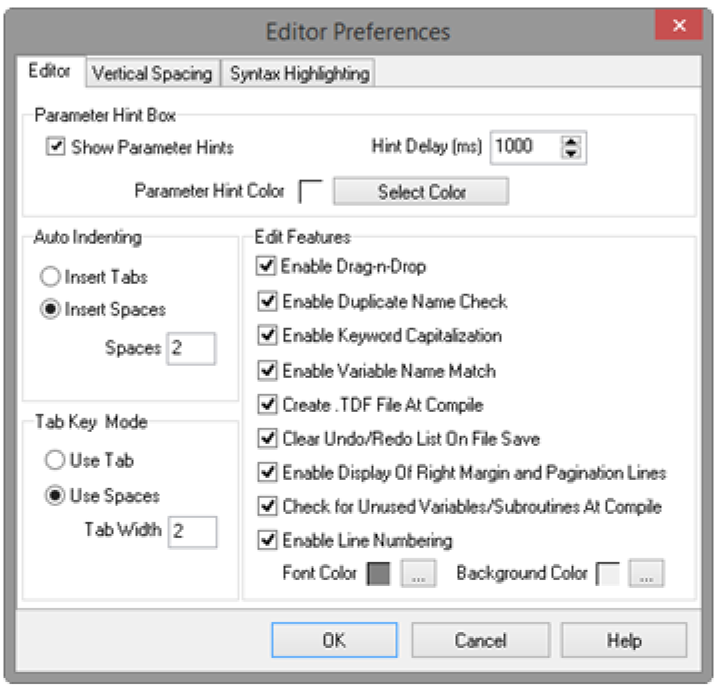
8.2.2.8 CRBasic Editor View Menu

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

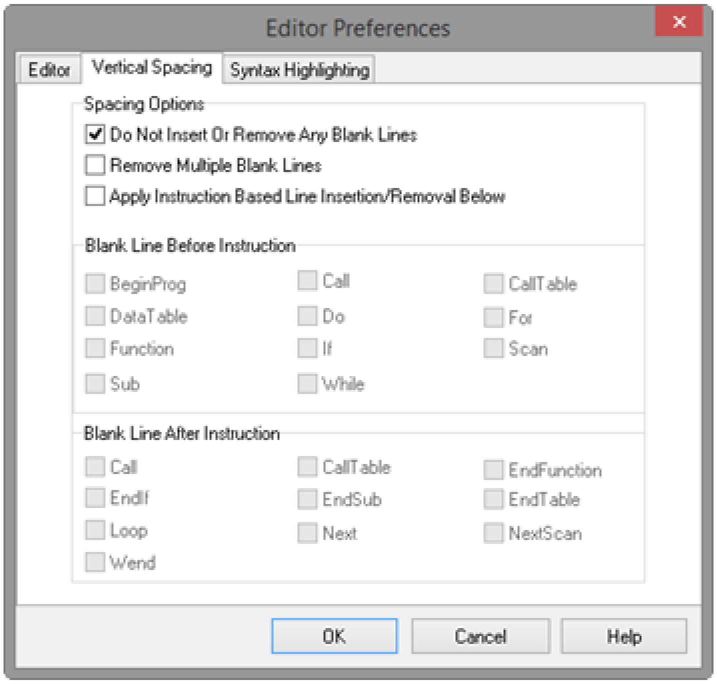
Editor Preferences

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

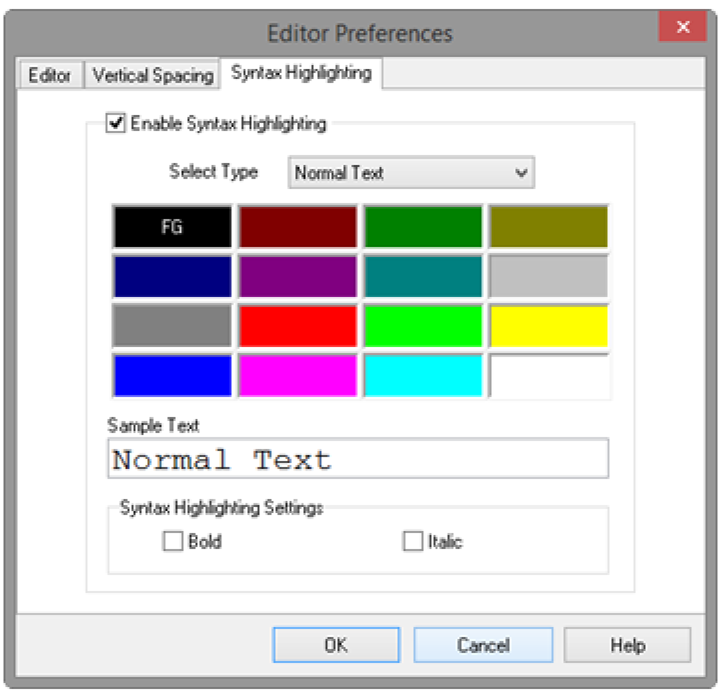
The **Editor** tab allows the user to toggle on or off the pop-up hints for parameters in instructions, set the amount of time the cursor must hover over the instruction before the pop-up hint appears, and the background color of the pop-up hint. This is also used to choose whether CRBasic automatic 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.



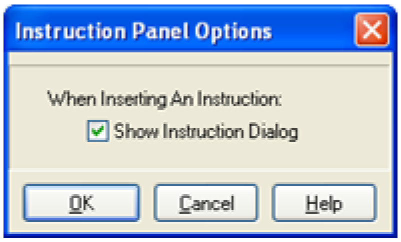
The **Syntax Highlighting** tab sets up the appearance of different text elements in the program using different font styles and colors. You can customize the appearance of the text by giving normal text, keywords, comments, operators, numbers, strings, and parentheses each a different font style and 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 C**ompile** > **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.

8.2.2.9 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.)

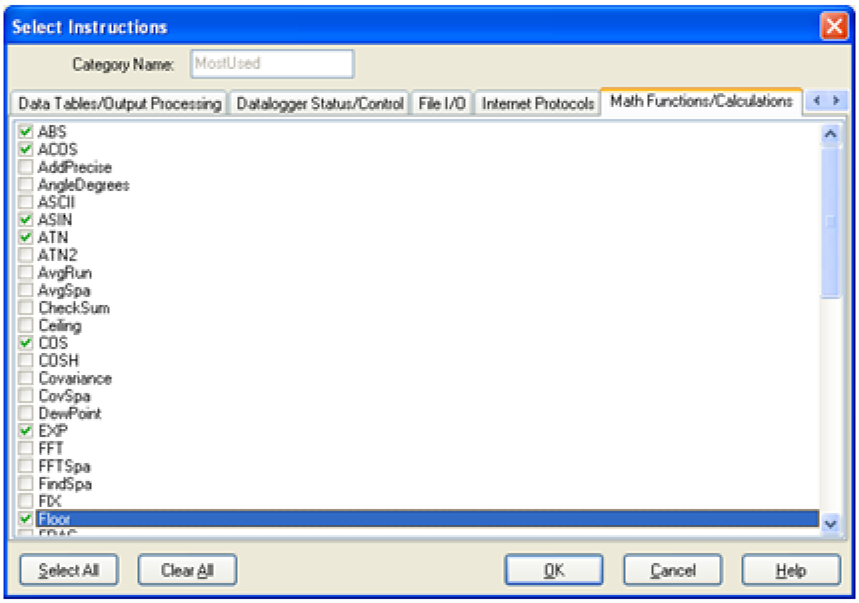
To create a new list, first select the Add New Category button and provide a name for the user-created category. Next, ensure the category name is selected and click the Edit Category button to bring up the Select Instructions dialog (shown below). Instructions that should be included in the new list 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 (C**ompile** > **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

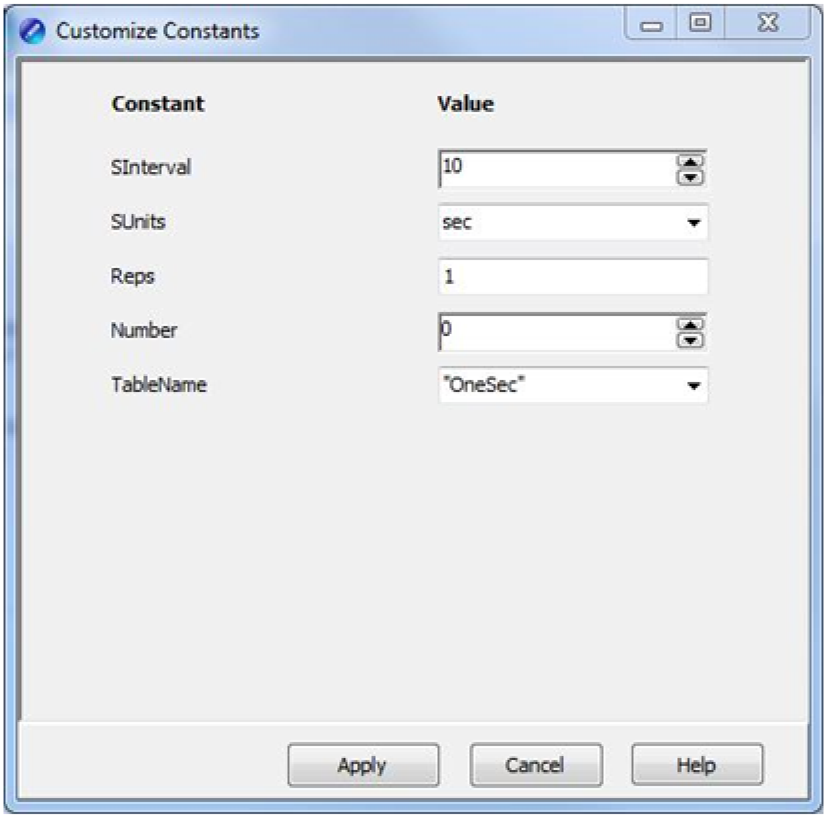
Within these two comments, define the constants. Following each constant, use the keywords noted below formatted as a comment, to set up edit boxes, spin boxes, or list boxes for the constant values. The fields are edit boxes by default. If a maximum/minimum are defined for a constant, the field will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

8.2.3 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.

8.2.3.1 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.

Declarations	*Make a list of what to measure and calculate.*
Declare constants	*Within this list, include the fixed constants used,*
Declare Public variables	*Indicate the values that the user is able to view while the program is running,*
Dimension variables	*the number of each measurement that will be made,*
Define Aliases	*and specific names for any of the measurements.*
Define data tables	*Describe, in detail, tables of data that will be saved from the experiment.*
Process/store trigger	*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?*
Table size	*Set the size of the table in RAM.*
Other on-line storage devices	*Should the data also be sent to the external storage?*
Processing of Data	*What data are to be output (current value, average, maximum, minimum, etc.).*
Define Subroutines	*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.*
Program	*The program section defines the action of datalogging.*
Set scan interval	*The scan sets the interval for a series of measurements.*
Measurements	*Enter the measurements to make.*
Processing	*Enter any additional processing with the measurements.*
Call Data Table(s)	*The Data Table must be called to process output data.*
Initiate controls	*Check measurements and Initiate controls if necessary.*
NextScan	*Loop back (and wait if necessary) for the next scan.*
End Program	

8.2.3.2 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.

8.2.3.3 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)
```

8.2.3.4 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)
```

8.2.3.5 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, IntRH, IntTemp, RainTot, _
RainInt, Solar
```

8.2.3.6 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.

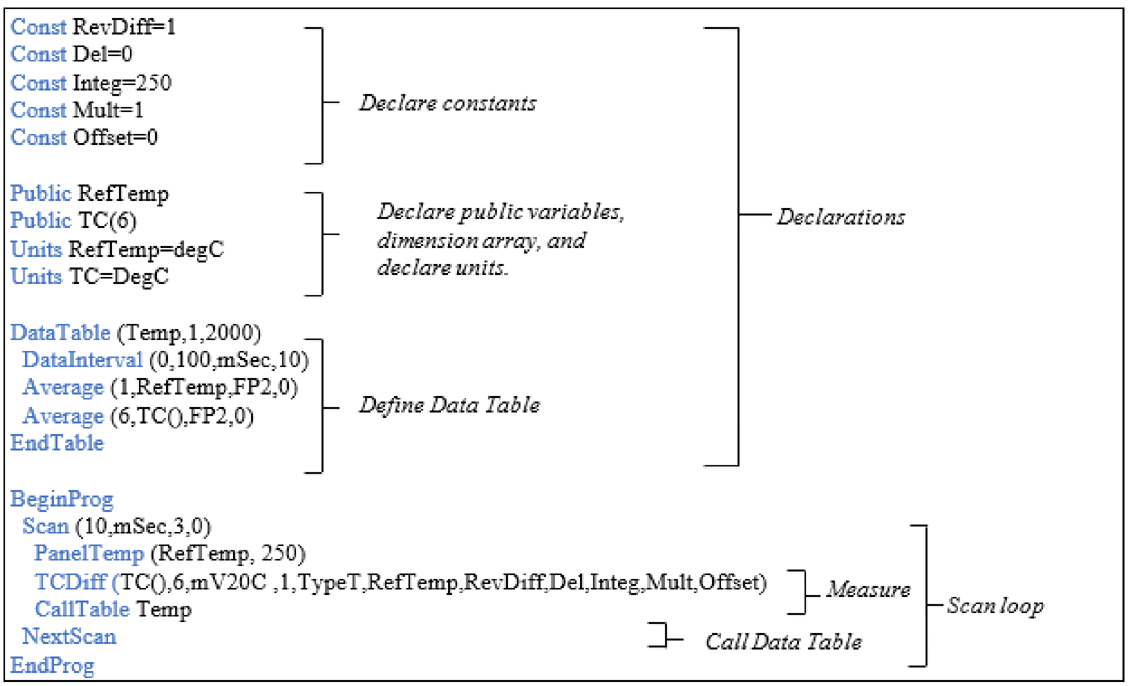
```
'CR1000X
```

```
'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.

8.2.3.7 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.



8.2.3.8 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	StnName 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)

8.2.3.9 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(TC(),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.

8.2.3.10 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.

8.2.3.11 Logical Expression Evaluation

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	≠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>=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>=5 and Z=2 then Y=0

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

If X>=5 or Z=2 then Y=0

will set Y=0 if either X>=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>=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<=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.

8.2.3.12 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.

8.2.3.13 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.

<table>
<tr><th colspan="3">Rules for Names</th></tr>
<tr><th>Name for</th><th>Maximum Length (number of characters)</th><th>Allowed characters</th></tr>
<tr><td>Variable or Array</td><td>39</td><td rowspan="5">Letters A-Z, upper or lower case, underscore "_", and numbers 0-12. The name must start with a letter. CRBasic is not case sensitive.</td></tr>
<tr><td>Constant</td><td>39</td></tr>
<tr><td>Alias</td><td>39</td></tr>
<tr><td>Data Table Name</td><td>20</td></tr>
<tr><td>Field name</td><td>39</td></tr>
</table>

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)().

8.2.3.14 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.

9. Working with Data Files

After data has been collected from the datalogger, you need a way to analyze that data. LoggerNet provides two tools to do this.

View Pro is a file viewer that provides a way to look at the collected data. It will open data files (*.DAT) saved in a variety of formats including files from mixed-array and table-based dataloggers. It can also be used to view data from a LoggerNet database table created with LNDB. View can also open other CSI file types (*.DLD, *.CSI, *.PTI, *.FSL, *.LOG, *.CR2, *.CR5, *.CR1, *.CR1X, *.CR3, *.CR300, *.CR6, *.CR8, *.CR9). Once a data file or database table is opened, data values can be graphed in several different formats including Line Graphs, Histograms, XY Plots, FFTs, and Rainflow Histograms.

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. Split is discussed in Split (p. 502).

Data stored on a compact flash, microSD, or PCMCIA card must be converted prior to analyzing it on your computer. CardConvert is a utility used to retrieve binary data from a compact flash or PCMCIA card, convert it to an ASCII or binary file, and save it to disk.

9.1 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. (TOACI1, 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 TOACI1 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. 258) 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.

9.1.1 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.

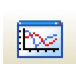

Tile Vertically. Rearranges all open, non-minimized data file windows as non-overlapping vertical tiles. This makes them all visible at the same time.

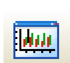

Tile Horizontally. Rearranges all open, non-minimized data file windows as non-overlapping horizontal tiles. This makes them all visible at the same time.


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.

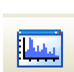

Print Preview. Displays how the currently selected data file will appear when it is printed.

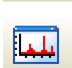

Print. Brings up a dialog box that allows you to print the currently selected data file.


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. 252) 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.

9.1.1.1 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.

9.1.2 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. 254) 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. 254) 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.)

9.1.2.1 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.

9.1.2.2 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 our 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.

9.1.3 Opening Files

9.1.3.1 Data Files

File | Open

To open a data file, press the  icon or select **File | Open** from the menu. TOACI1, 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. 258) 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.

9.1.3.2 Other Files

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.

File | Open as Hex

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

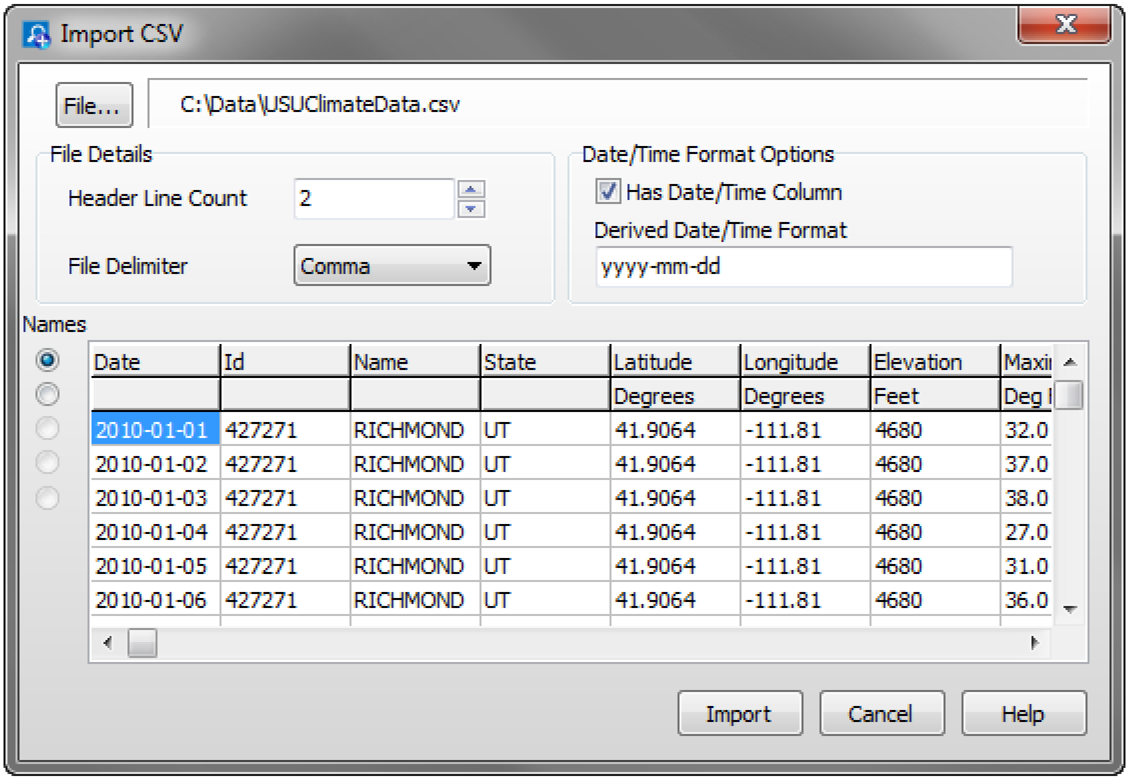
Import CSV

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.



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.

9.1.4 Viewing Database Tables

9.1.4.1 File | View LoggerNet Database Table

This option allows you to view data from a LogggerNet database table. The first time the option is selected you will need to select a database (see Selecting a Database (p. 264)) and then select a database table (see Selecting a Table (p. 268)).

9.1.4.2 Selecting a Database

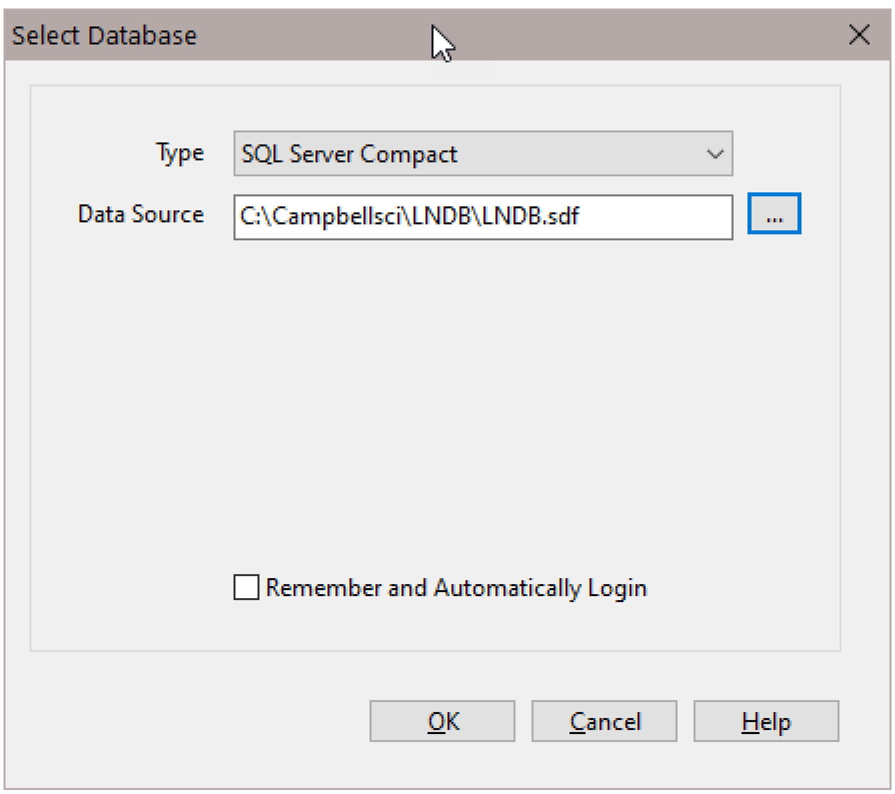
The Select Database dialog box comes up the first time you select **File | View LoggerNet Database Table**. (After a database has been selected, this menu item will bring up the Selecting a Table (p. 268) dialog box. To view a table from a different database, you will need to press the Change Database 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. 269) or Installing Oracle ODBC Drivers (p. 278).

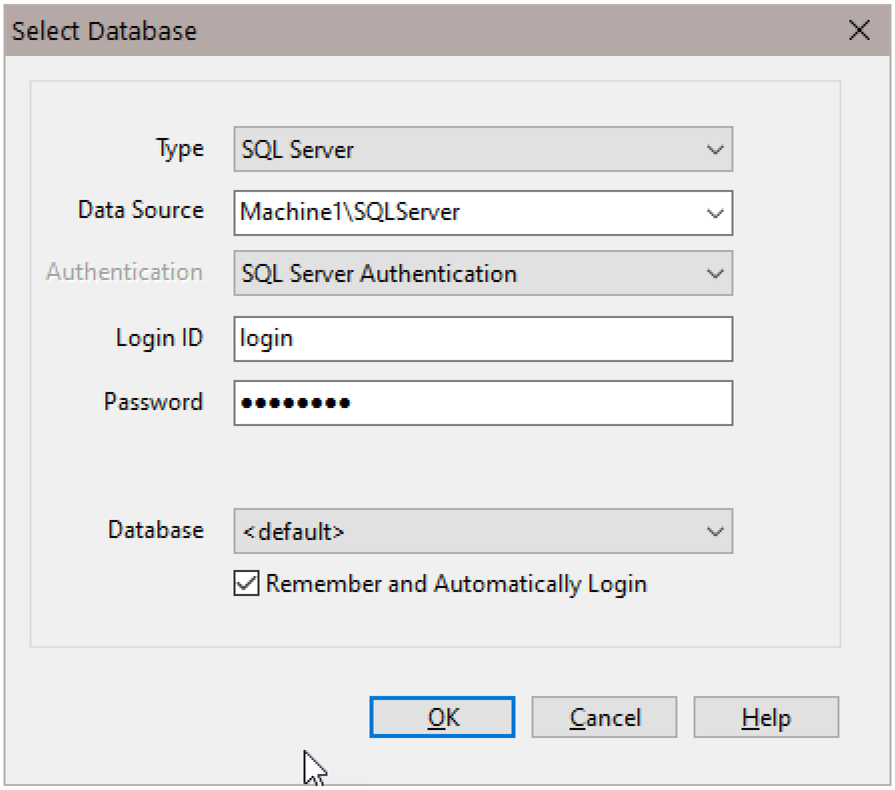
The information to enter changes depending on the database type as described below:

SQL Server Compact



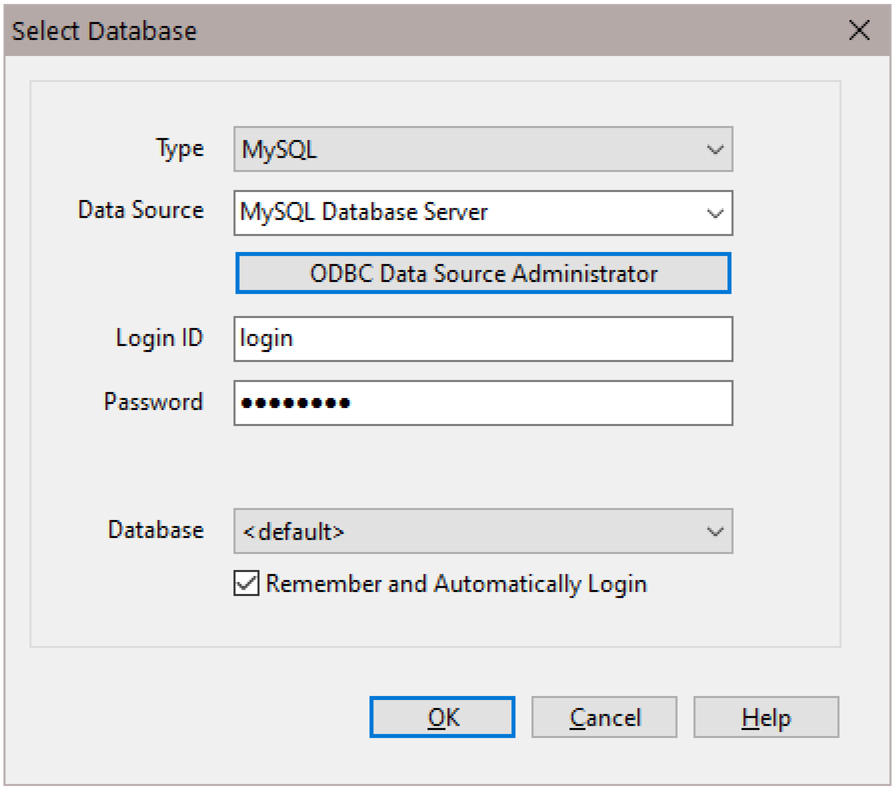
SQL Server Compact is an embedded database that just requires the selection of a filename. Press the Browse button to the right of the Data Source field to browse to the desired database.

SQL Server



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.

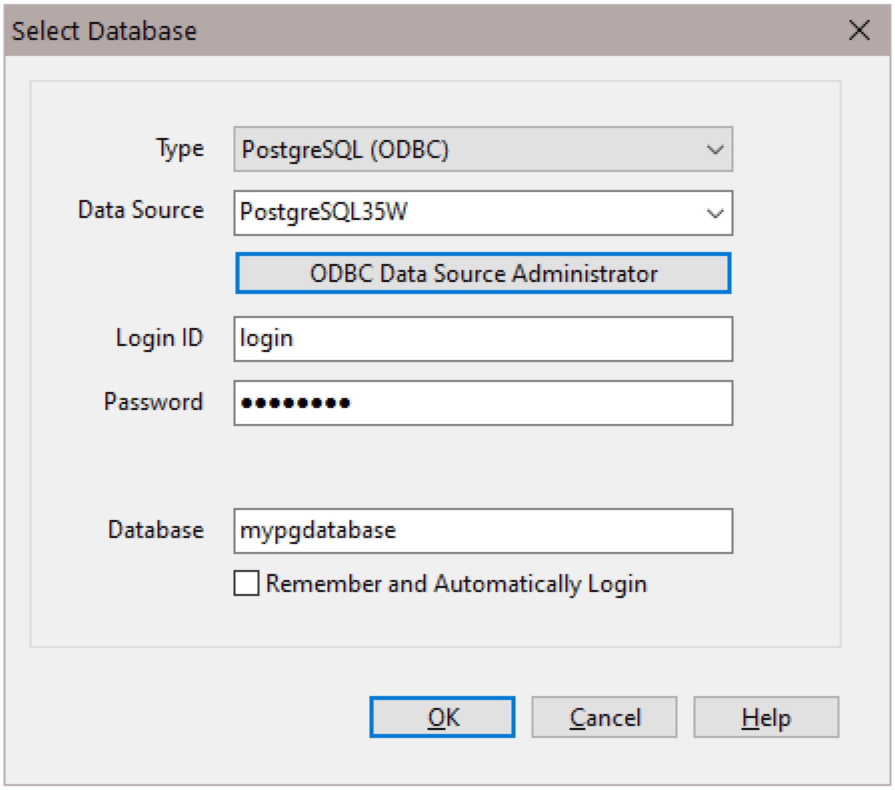
MySQL



The 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)

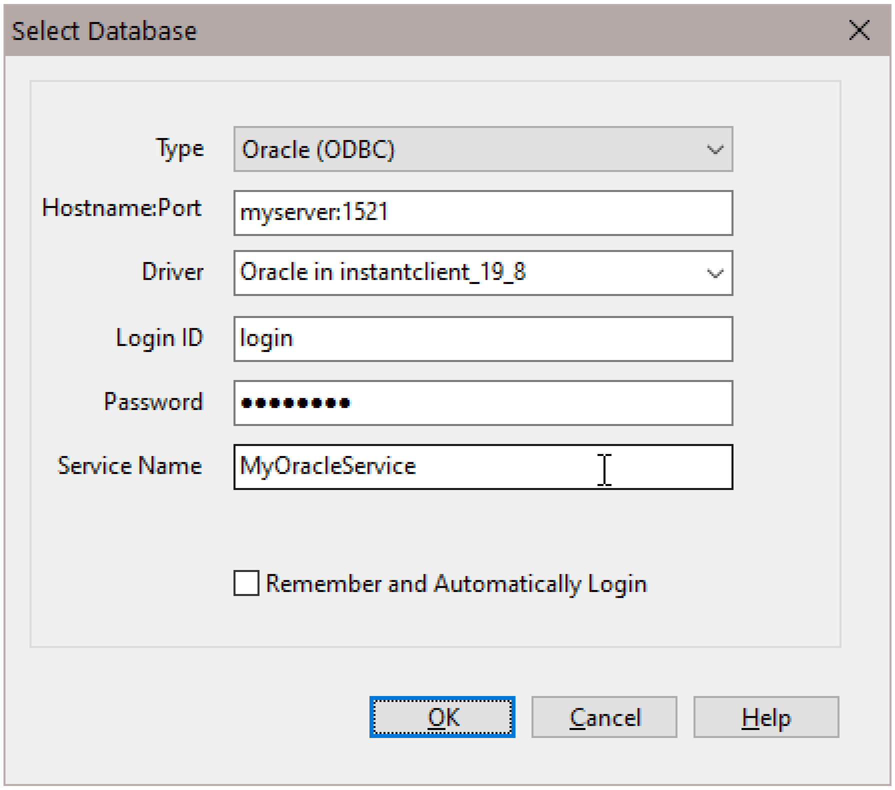
> **NOTE:** To use a PostgreSQL database, ODBC drivers are required. See Installing PostgreSQL ODBC Drivers (p. 269).



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. 278).



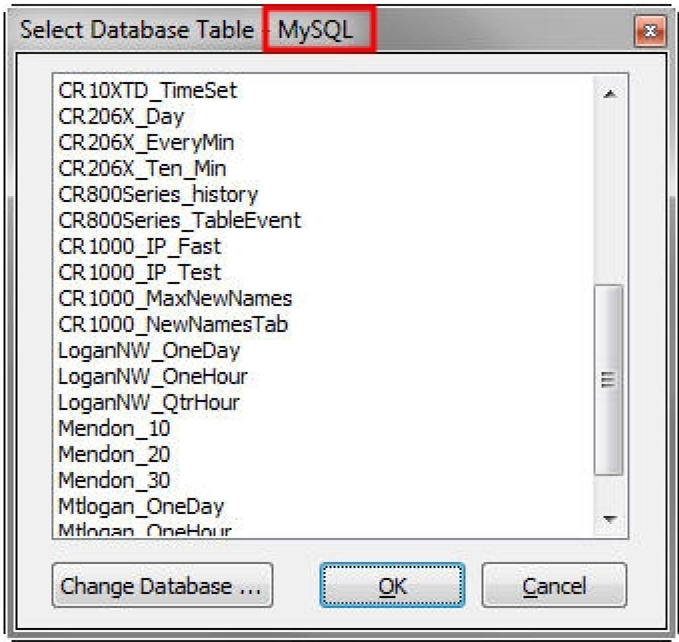
In **Hostname:Port**, enter the IP address or DNS name of the host (server) and port if not the default. (The default Oracle port is 1521.) Choose the driver from the **Driver** dropdown list ("Oracle in instantclient_19_8" in the driver installation example). Enter the **Login ID**, **Password**, and **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.

9.1.4.3 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.

9.1.4.4 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.

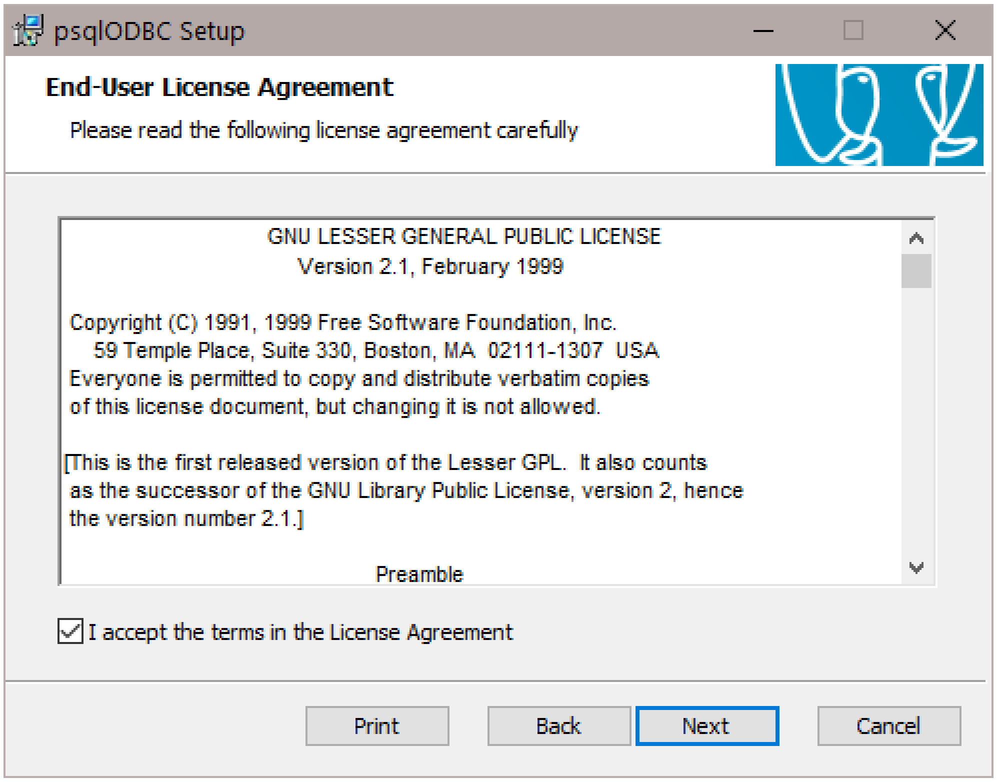
> NOTE: View Pro is different from the LNDB Manager (and LNDB in general), which can connect to the PostgreSQL database without an ODBC driver.

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.)

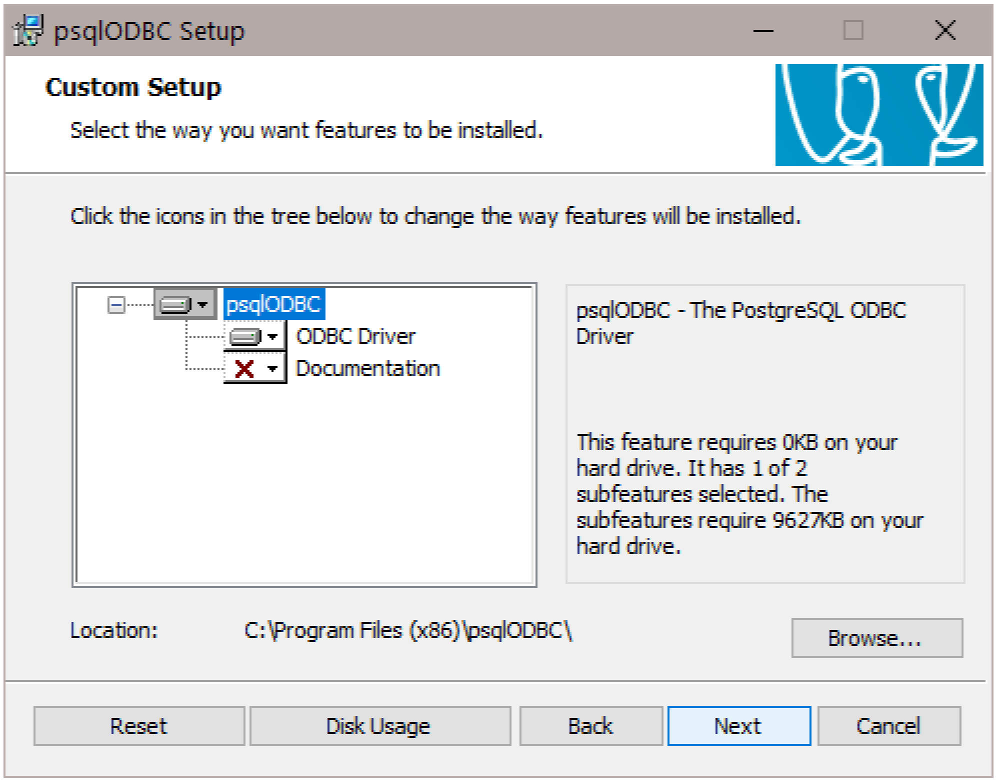
From the home page:

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, psqlodbc_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.)

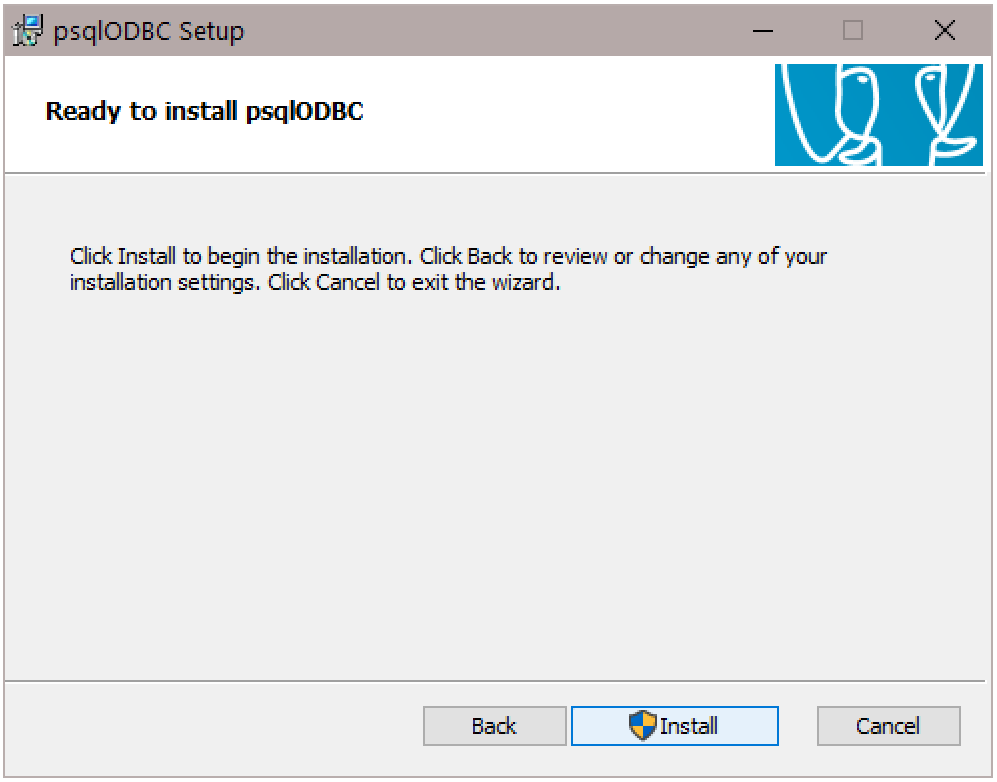
5. Click the appropriate .zip file to download the file.
6. Right-click the .zip file to extract the .msi file from it.
7. Double-click the .msi file to run the driver installation process.
8. Click **Next** to begin the installation.
9. Accept the license agreement and click **Next**.



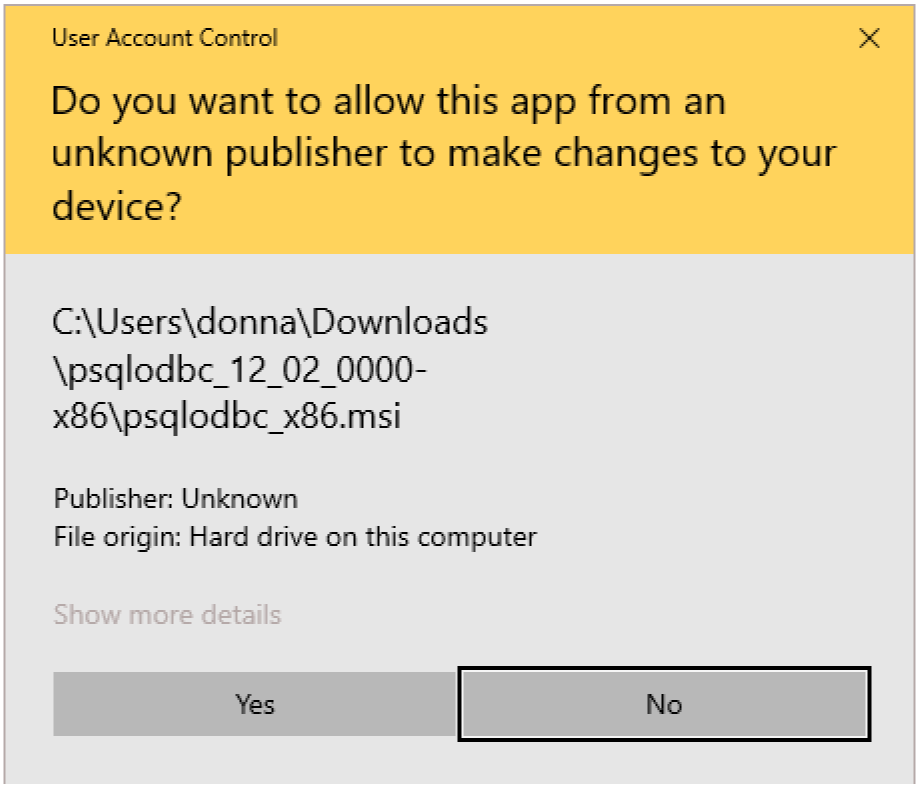
10. Click **Next** to accept the default setup.



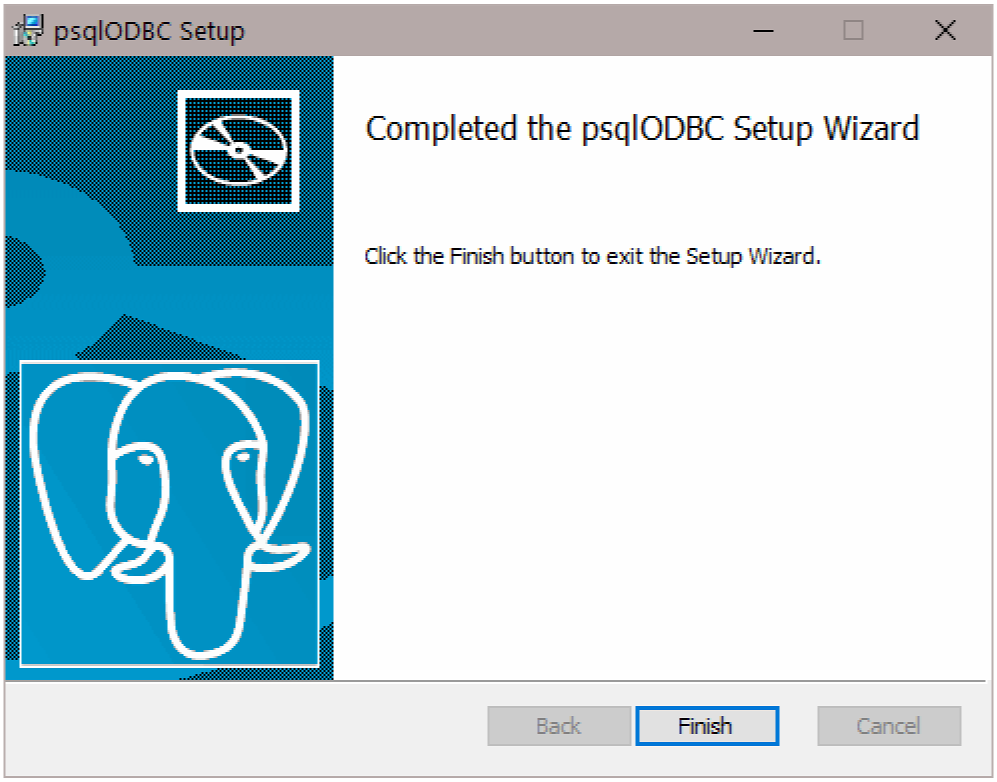
11. Click **Install**.



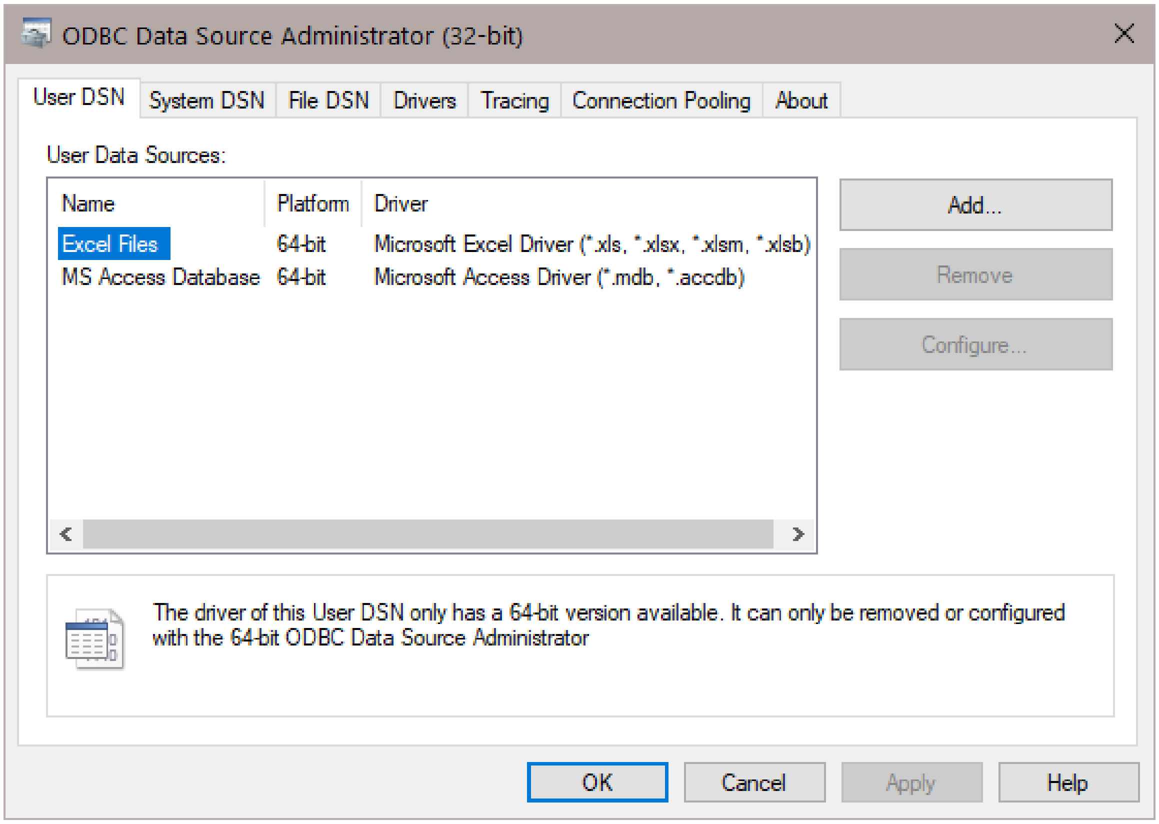
12. You may see a Windows warning similar to this. Click **Yes** to continue.



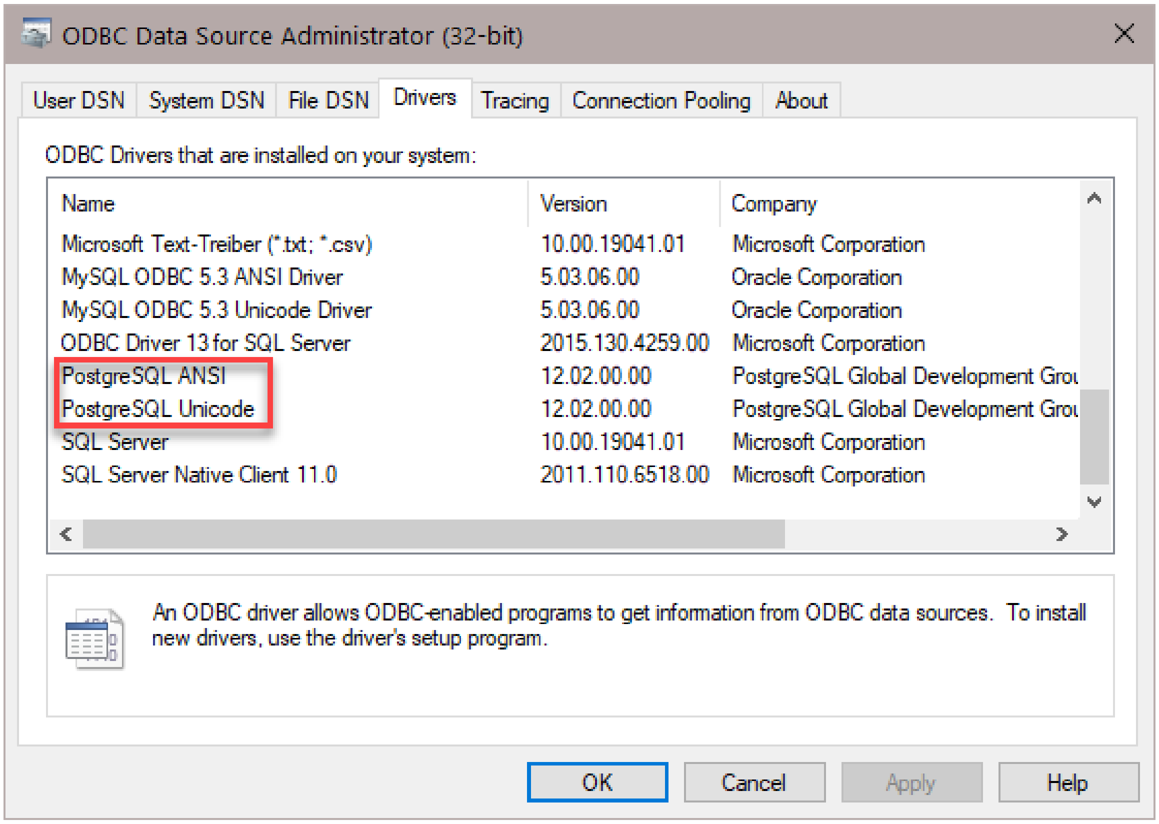
13. Click **Finish** to close the installation wizard.



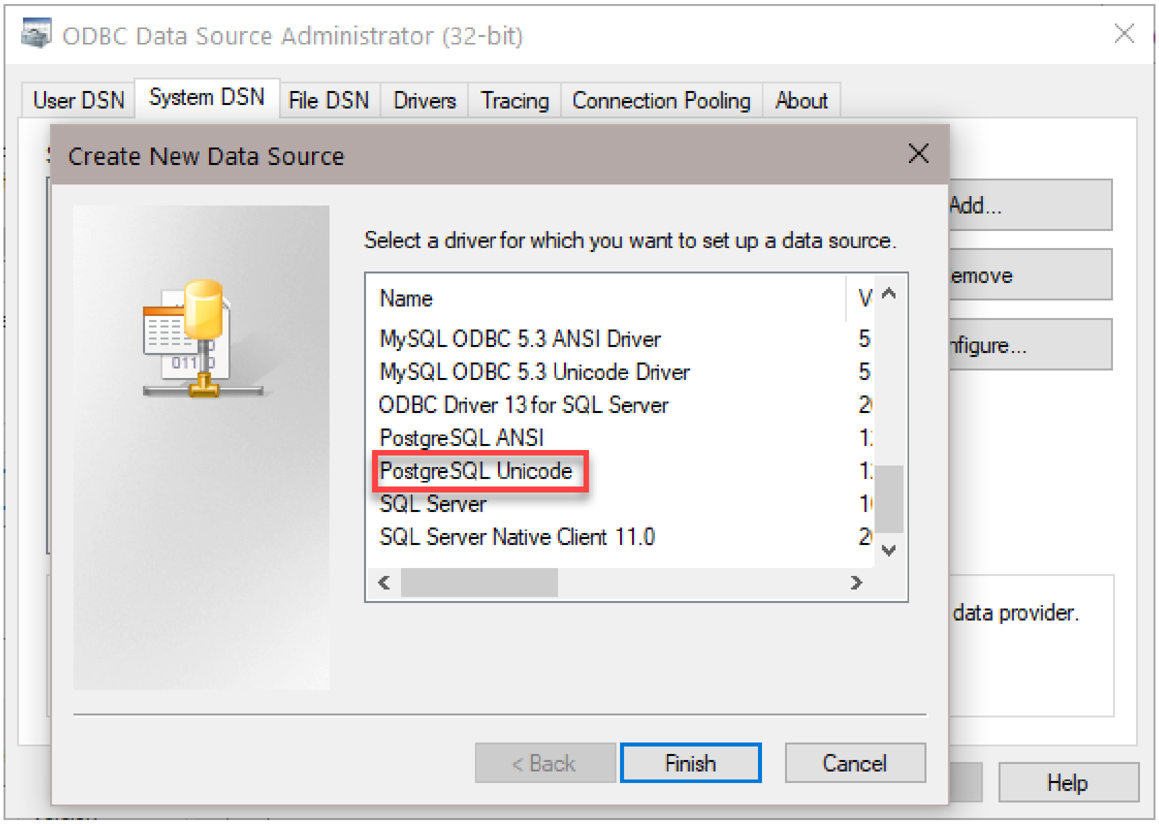
14. Once the installation is complete, run the 32-bit (Windows/Microsoft) ODBC Data Source Administrator. (Type ODBC in the Windows search bar. From the choices, make sure to select the 32-bit version.)



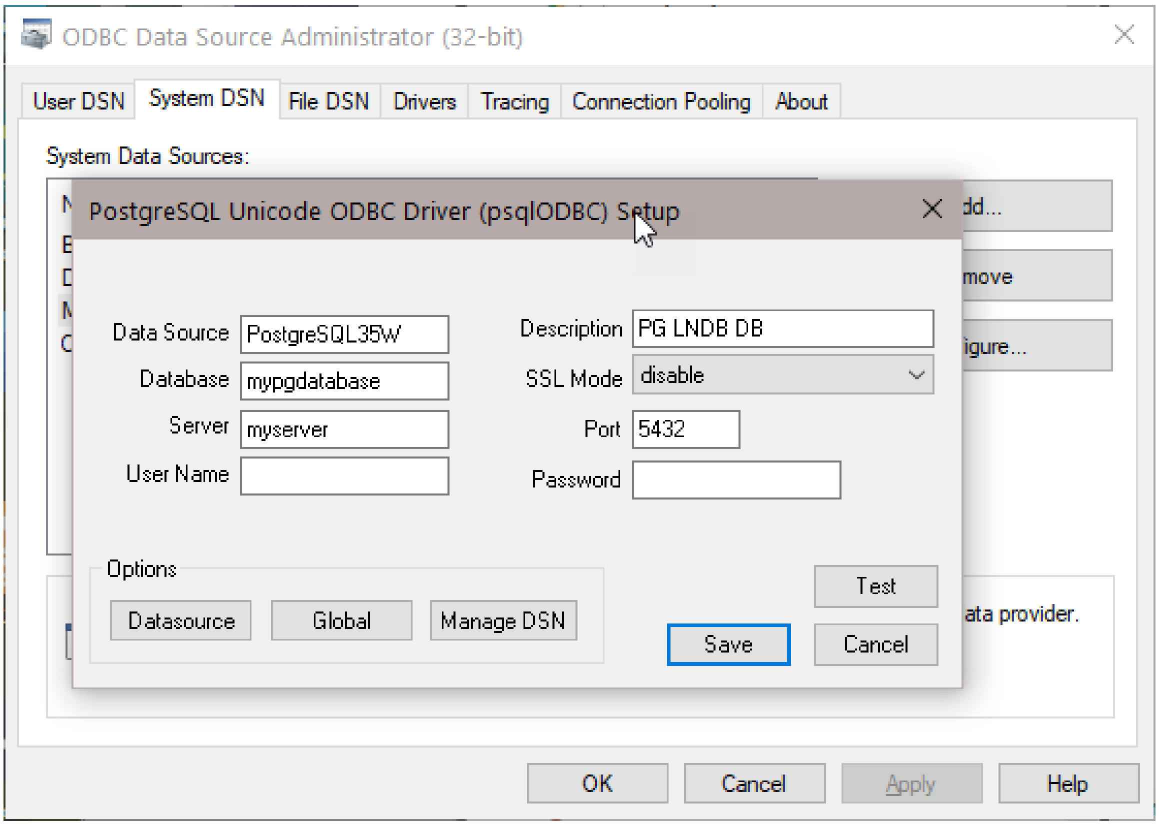
15. Click the **Drivers** tab and verify the drivers highlighted here:



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 **Sever** 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. 264) and Selecting a Table (p. 268) for how to use the data source in View Pro.

9.1.4.5 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.

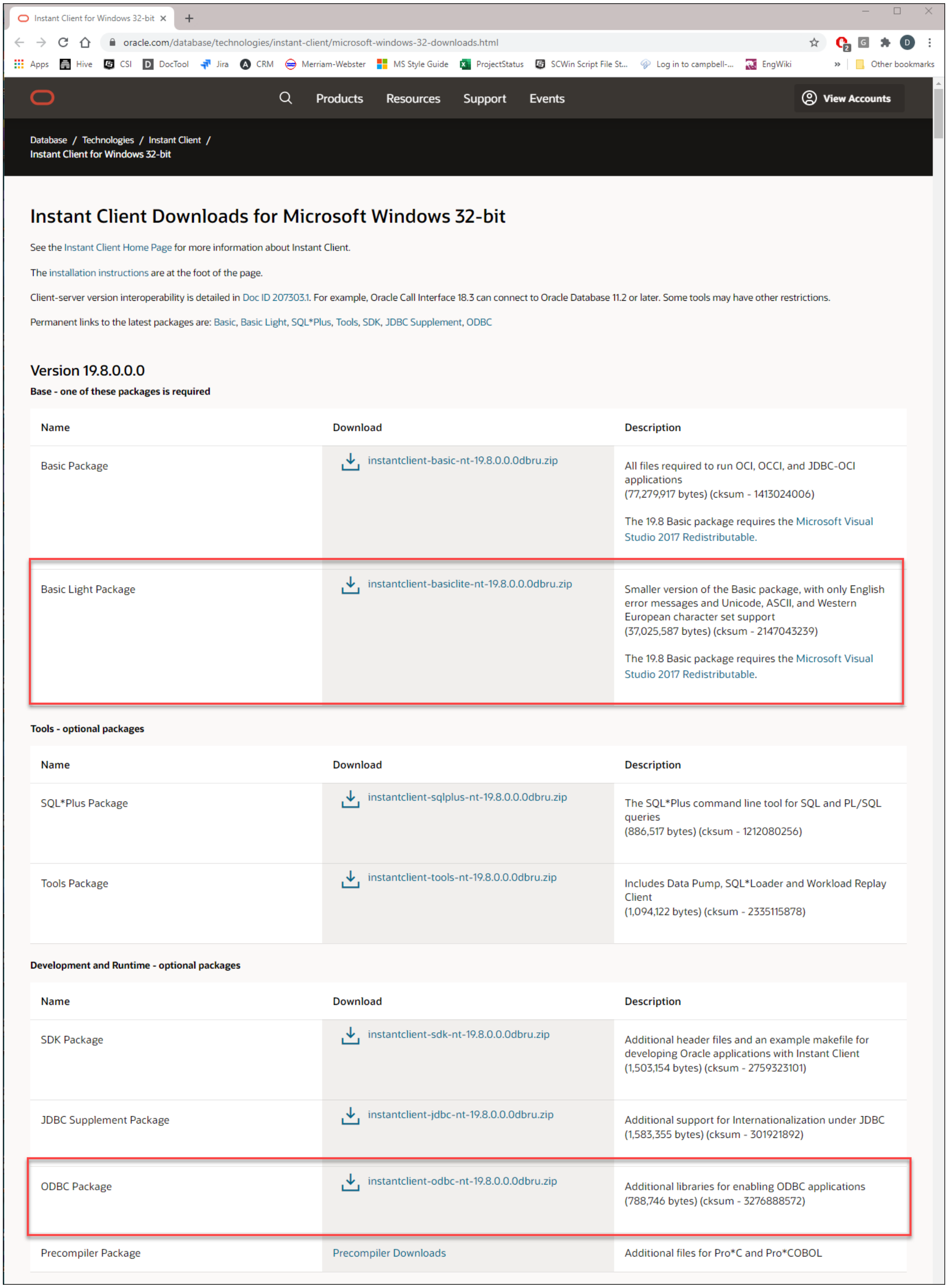
Oracle provides multiple official ODBC drivers for use with Oracle databases. View Pro can work with any of these drivers, but it must be a 32-bit version. Also, the driver must have a version number at least as large as the version of the Oracle database server being used (12c, 18c, 19c etc.) Consult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


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	instantclient-basic-nt-19.8.0.0.0dbru.zip	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	instantclient-basiclite-nt-19.8.0.0.0dbru.zip	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.

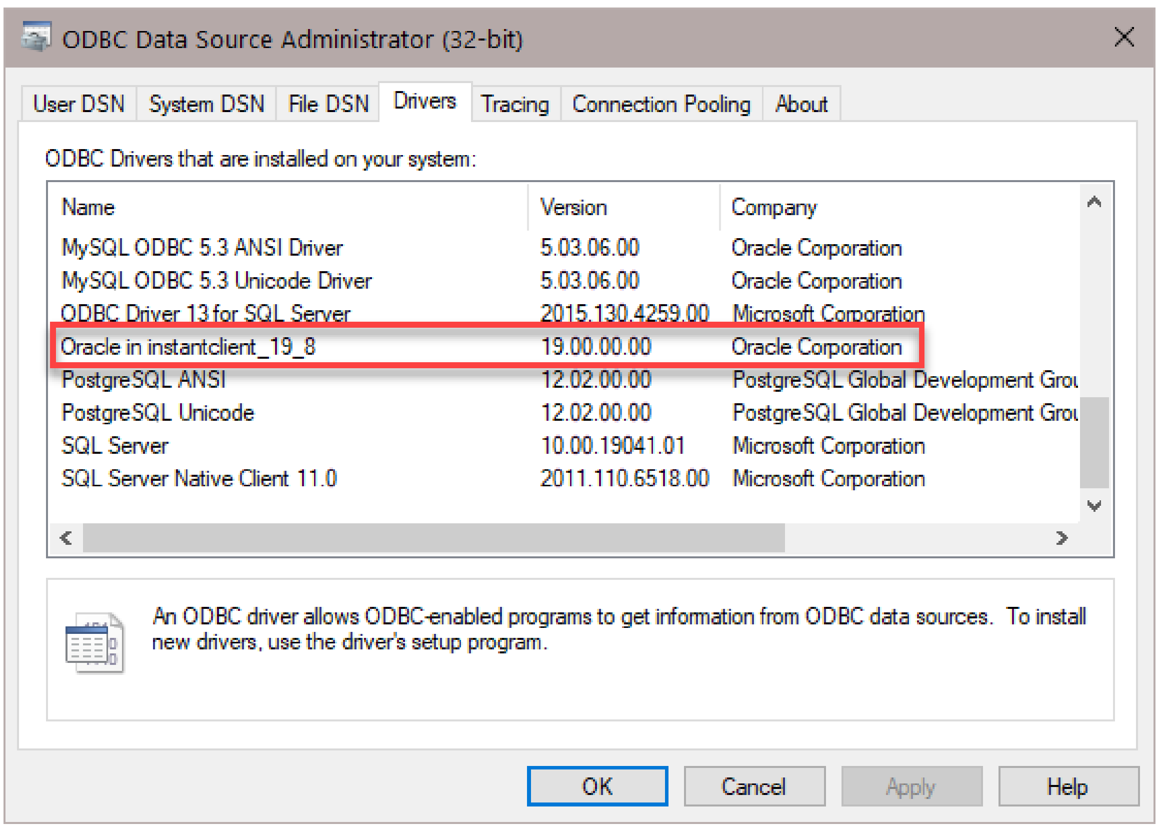
Tools - optional packages

Name	Download	Description
SQL*Plus Package	instantclient-sqlplus-nt-19.8.0.0.0dbru.zip	The SQL*Plus command line tool for SQL and PL/SQL queries (886,517 bytes) (cksum - 1212080256)
Tools Package	instantclient-tools-nt-19.8.0.0.0dbru.zip	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	instantclient-sdk-nt-19.8.0.0.0dbru.zip	Additional header files and an example makefile for developing Oracle applications with Instant Client (1,503,154 bytes) (cksum - 2759323101)
JDBC Supplement Package	instantclient-jdbc-nt-19.8.0.0.0dbru.zip	Additional support for Internationalization under JDBC (1,583,355 bytes) (cksum - 301921892)
ODBC Package	instantclient-odbc-nt-19.8.0.0.0dbru.zip	Additional libraries for enabling ODBC applications (788,746 bytes) (cksum - 3276888572)
Precompiler Package	Precompiler Downloads	Additional files for Pro*C and Pro*COBOL

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.



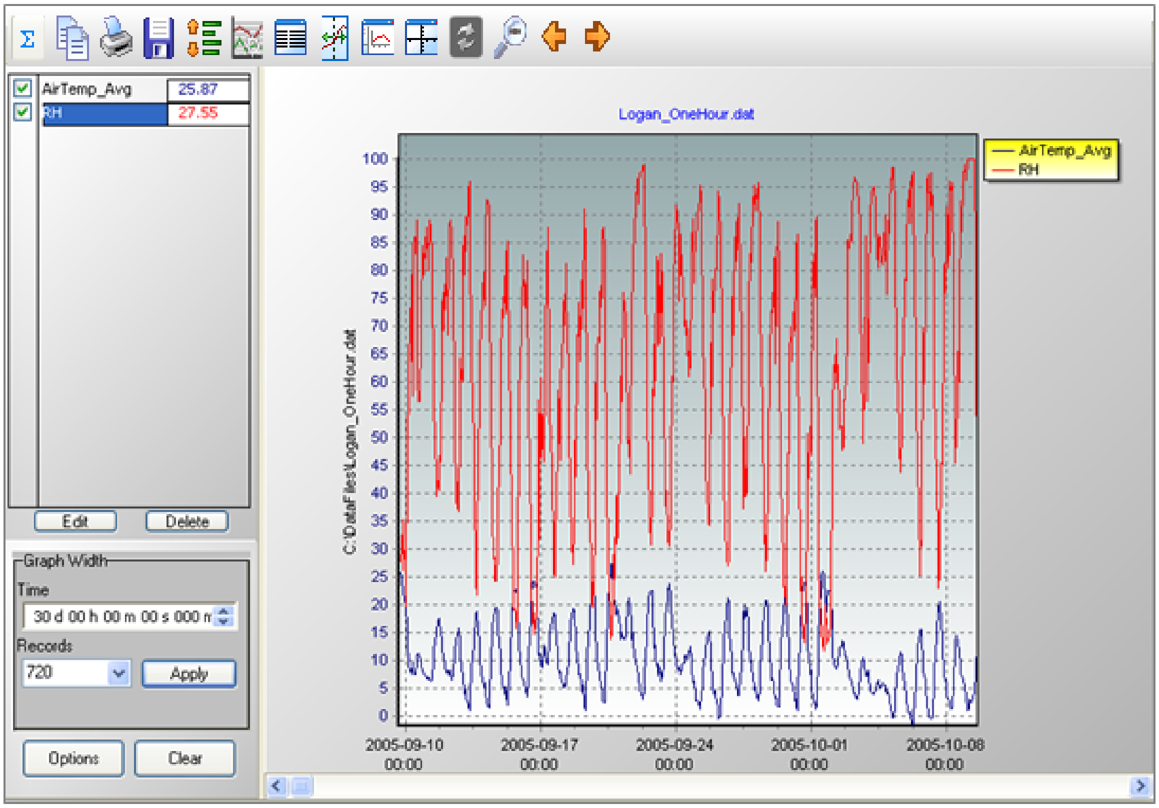
For testing or troubleshooting, a DSN can be configured in the System DSN tab, but this is not required.

See Selecting a Database (p. 264) and Selecting a Table (p. 268) for information on setting up the database in View Pro.

9.1.5 Data View Options

9.1.5.1 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. 286) 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.

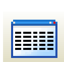
Lock Scrolling. Locks and unlocks the scroll bar at the bottom of the graph.

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

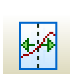
When locked, the graph will not scroll. However, moving the scroll bar on either the graph or the data file will still scroll the data file.

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)

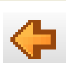
When using independent y-axes, the scale shown on each y-axis will apply only to the last selected trace assigned to that axis. (A trace is selected by clicking on its name in the list above the Edit and Delete buttons.)

Refer to Common vs. Independent Y-Axes (p. 286) 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.

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.

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.

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.

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.

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

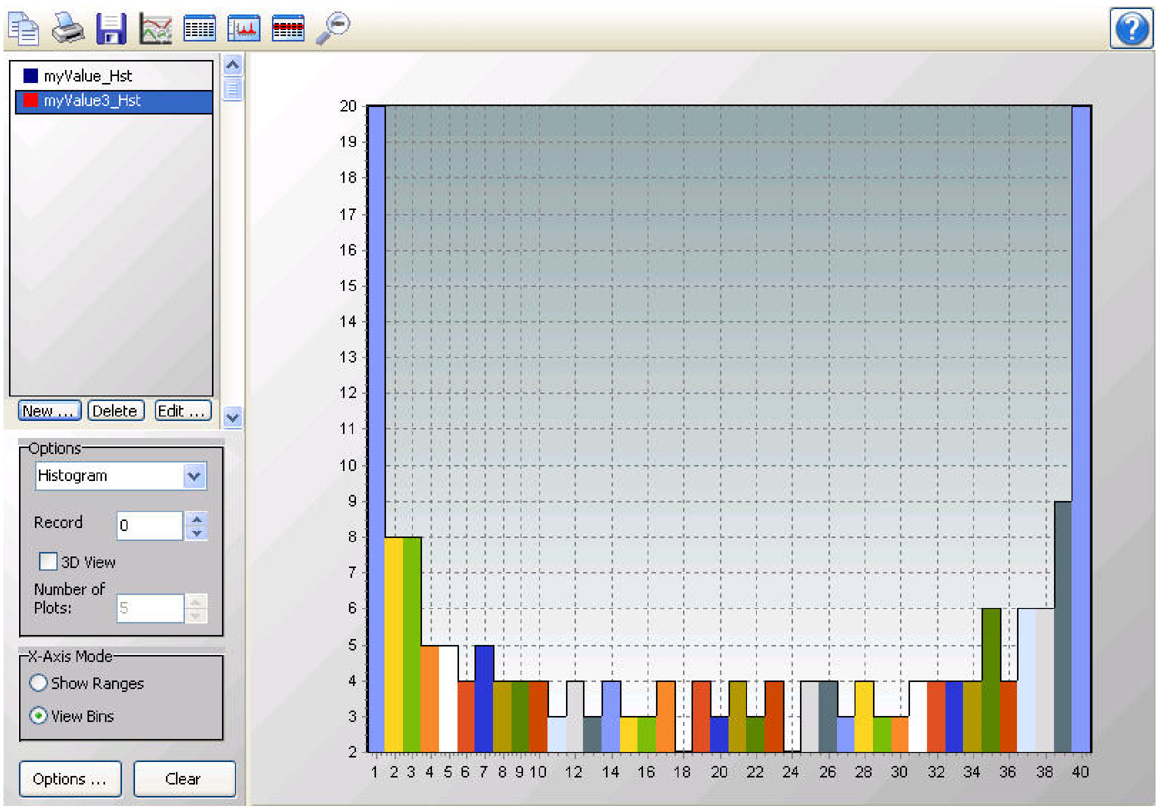
Press the button to open a dialog box from which to choose a color for the frames around the marks.

Style

Determines the data contained in each mark. Choose Y-Value, X-Value, or X and Y Value.

9.1.5.2 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 TOACI1 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. 252) 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

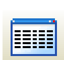
The Histogram includes the following toolbar icons:


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.

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.

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.

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.

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

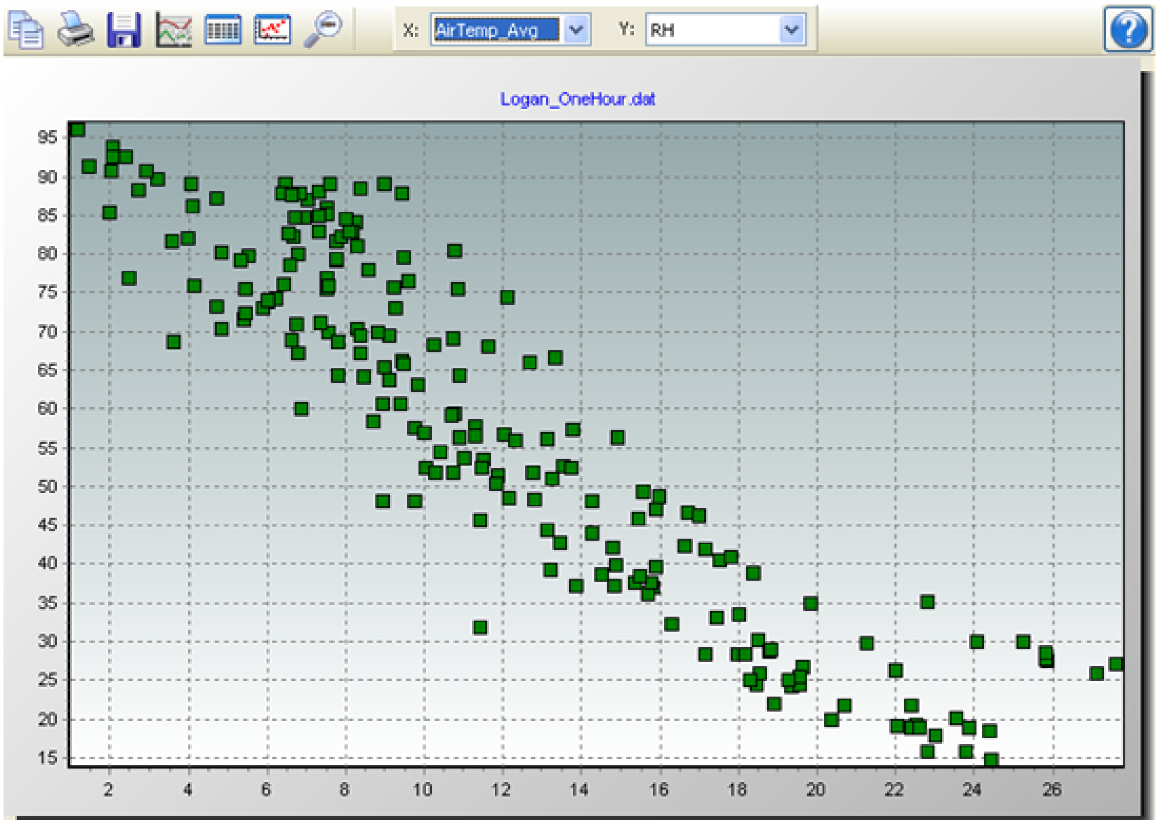
The minimum value on the X Axis scale. This field is disabled when the Scaling Option is set to Automatic Scaling.

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.

9.1.5.3 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

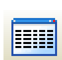
The XY Plot includes the following toolbar icons:

 **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.

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.

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.

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.

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.

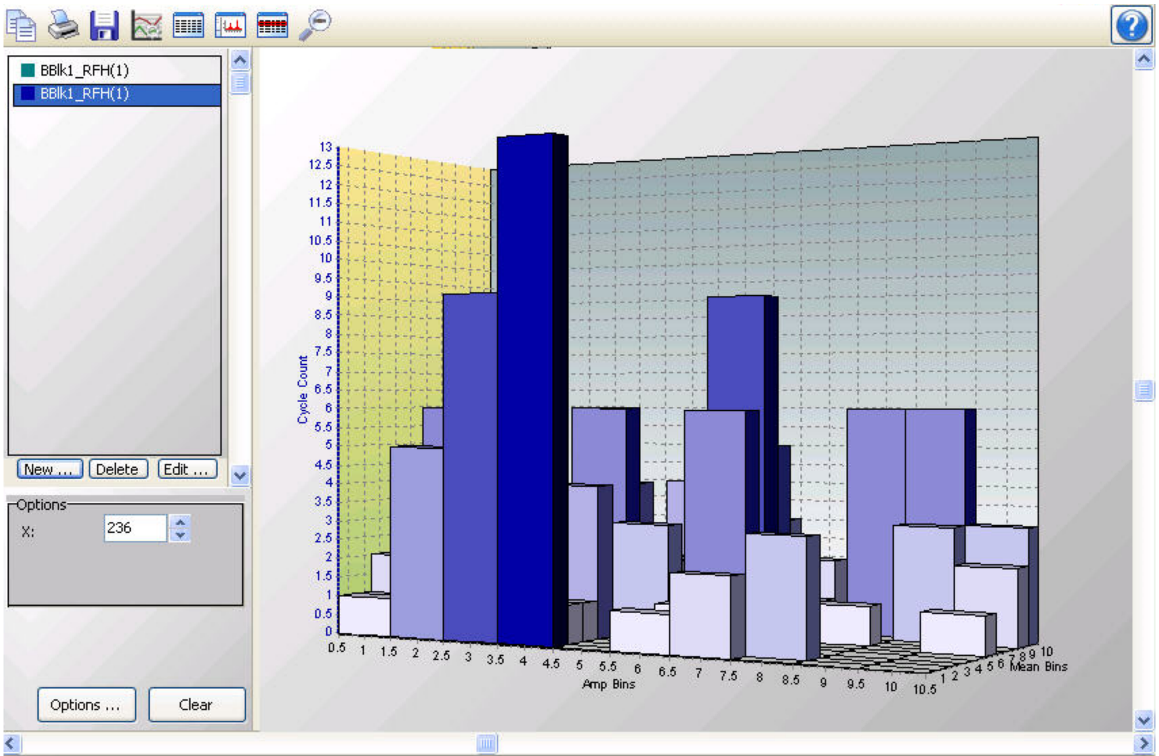
Style

Determines the data contained in each mark. Choose Y-Value, X-Value, or X and Y Value.

9.1.5.4 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. 252) 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.

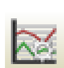
Rainflow Histogram Toolbar Icons

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.

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.

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.

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

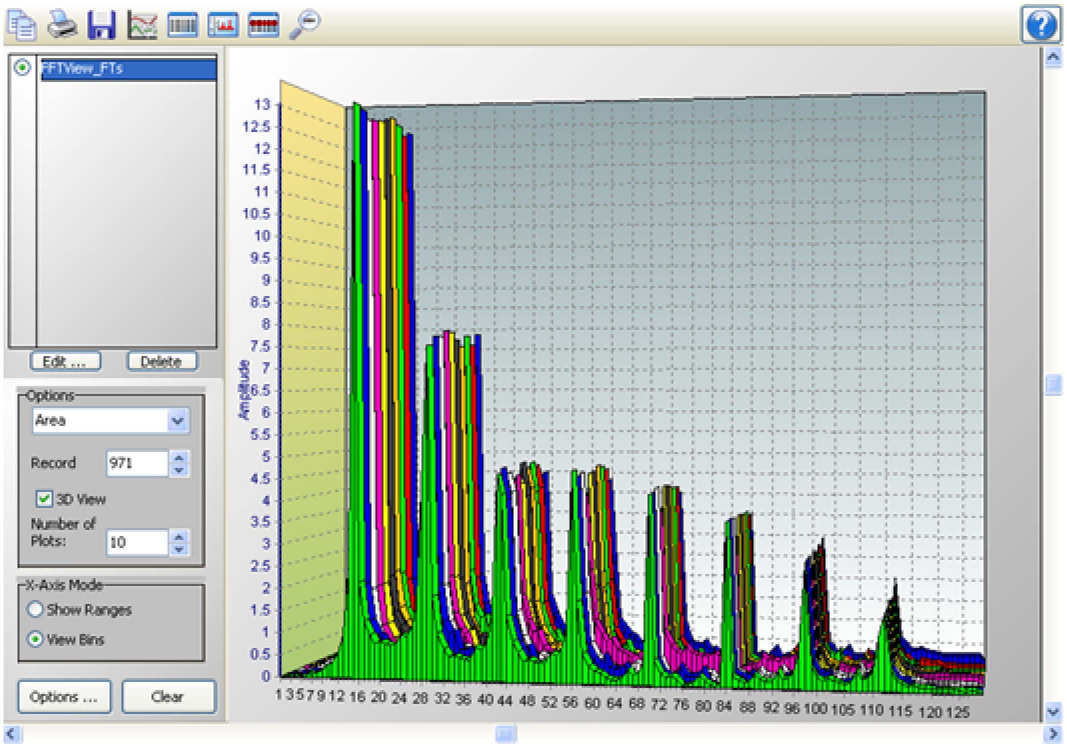
Determines whether a title is shown for the Z Axis. Select the check box to show a title. Type in the desired title and press the Font button to choose the font, style, size, effects, and color of the title.

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.

9.1.5.5 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. 252) 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 FF T 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.

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.

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.

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.

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.

9.1.6 Printing

9.1.6.1 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.

9.1.6.2 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:

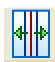
 Displays the first page of the data file.

 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).

9.2 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.

9.2.1 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.

9.2.2 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.

9.2.2.1 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 H**our/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 (p. 410).

The file format is reflected in the default filename by the prefix of TOA5, TOB1, CSV, or CSIXML added to the table name.

9.2.2.2 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.

9.2.2.3 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.

9.2.2.4 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.

9.2.3 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 C**ancel 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).

9.2.3.1 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.

9.2.4 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.

9.2.5 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.

10. Automating Tasks with Task Master

The Task Master is an application that is used to set up a Task that can be run on a defined schedule or based upon a data collection event from a datalogger. A Task can be data collection from another datalogger, FTP of a collected file, or anything that can be executed in a computing environment, such as a command line operation, a program executable, a batch file, or a script.

It can be used to launch third party software utilities such as a command line FTP client to send data to the Internet or a phone dialer to call or text a phone upon an alarm condition.

Also note that when running LoggerNet as a service, tasks being run by the Task Master cannot interact with the desktop. Therefore, any tasks set up in the Task Master should not require any user interaction.

10.1 Task Master

The Task Master is used to set up a Task that can be run on a defined schedule or based upon a data collection event from a datalogger. A Task can be data collection from another datalogger, FTP of a collected file, or anything that can be executed in a computing environment (i.e., a command line operation, a program executable, a batch file, or a script).

> **NOTE:** When running LoggerNet as a service, tasks being run by the Task Master cannot interact with the desktop. Therefore, any tasks set-up in the Task Master should not require any user interaction.

> **NOTE:** By default, when running LoggerNet as a service, the service is run under the LoggerNet user account. The LoggerNet user has write access only to the CampbellSci directories. Therefore, if a task requires something to be written or done in a different directory or on the desktop, the LoggerNet user does not have sufficient access and the process will end in an error. This can be solved by giving the LoggerNet user write access to the necessary directories. Click Issues with Running LoggerNet as a Service (p. 9) for more information.

NOTE: The Task Master was integrated into the LoggerNet server in LoggerNet 4.2. Integrating the Task Master into the server involved extensive changes. When upgrading to LoggerNet 4.2 from a previous version, an attempt will be made to import all previously-configured tasks. However, imports have only been tested back to LoggerNet 3.4.1. After upgrading (from any previous version of LoggerNet), you should verify that all of your tasks have imported correctly.

The Task Master is opened by selecting Task Master from the Toolbar's Main category or from LoggerNet's **Launch | Main | Task Master** menu item.

When the Task Master is opened, there are two tabs. The **Setup** tab is used to define the tasks to be run. The **Status** tab is used to view the last time a task was run, the next time it will be run, and to run a task outside of its defined schedule.

10.1.1 Setup Tab

10.1.1.1 Add Scheduled

To execute a task on a defined schedule, choose the **Add Scheduled** button. A Task item will be added to the list of stations and tasks. The schedule can be set up on an interval or based on the calendar. Select which kind of schedule to set by selecting the Interval or Calendar radio button. Then define the schedule for the task by completing the fields as described below.

Interval

Base Date - The first date on which the task should be executed. If the date entered has already occurred, the task will be run on the next occurrence of the event interval and base time.

Base Time - The time on which each execution of the task should be based.

Schedule Interval- How often execution of the task should occur.

For example, to execute a task at 12:30 p.m. daily, set the **Base Time** at 12:30 p.m. and the **Schedule Interval** at 1 day.

Calendar

Set the Hours of the Day, Minutes of the Hour, Days of Month, Days of Week, and Months on which the task should be executed. The task will run when ALL of the specified settings are met. If a setting is left blank, it will always apply.

For example:

To execute a task on the first day of every month at 8:00 a.m., set the **Hours of the Day** to 8, the **Minutes of the Hour** to 00, the **Days of the Month** to 1, and leave the other settings blank.

To execute a task every Tuesday at 6:15 a.m., set the **Hours of the Day** to 6, the **Minutes of the Hour** to 15, the **Days of the Week** to 3-Tuesday, and leave the other settings blank.

To execute a task on the first Monday of every month at midnight, set the **Hours of the Day** to 00, the **Minutes of the Hour** to 00, the **Days of the Month** to 1, 2, 3, 4, 5, 6, 7, the **Days of the Week** to 2-Monday, and leave the other settings blank.

To execute a task on the fifth day of every quarter at midnight, set the **Hours of the Day** to 00, the **Minutes of the Hour** to 00, the **Days of the Month** to 5, the **Months** to 1-January, 4-April, 7-July, 10-October, and leave the other settings blank.

After specifying the desired schedule, press the **View Schedule** button to bring up a calendar that shows the current defined schedule. Verify this is the desired schedule. (You can zoom in on an area of the schedule by dragging your mouse from top-left to bottom-right.)

10.1.1.2 Add After

To execute a task based on a data collection event, select the station from the Task and Stations list and press the **Add After** button. A Task item will be added to the list of stations and tasks. Define the condition under which the task should be executed using the **Station Event Type** drop-down list box.

> **NOTE:** Many of the tasks will be triggered by scheduled collection or by using a **Collect Now** button, or by another task that calls the associated station. (**Collect Now** buttons are found on the Status Monitor and on the Connect Screen.)

> **NOTE:** Using Custom Collect from the Connect Screen will not trigger any tasks.

- **After Any Scheduled Call** - After a scheduled data collection attempt, regardless of whether or not the call is successful. The task is triggered only by a scheduled collection.
- **After Successful Call** - After a successful scheduled data collection, after using a **Collect Now** button successfully, after a call-back, or after another task that calls the associated station successfully.
- **After Failed Scheduled Call** - After a scheduled data collection fails, after using a **Collect Now** button unsuccessfully, or by another task that calls the associated station unsuccessfully.
- **After Any Call** - After a scheduled collection, after using a **Collect Now** button, after a call-back, or after another task that calls the associated station. The task is triggered regardless of whether or not the data collection was successful

- **After Some, Not All, Data** - After some, but not all data is collected during a scheduled collection, when using a **Collect Now** button, or when another task calls the associated station.
- **On Call-Back** - When a call-back attempt is detected. This option does not wait for data collection to finish. The task is triggered by a call-back from a datalogger executing a P97 instruction or a SendVariable instruction.
- **After Call-Back** - When a call-back attempt is completed and data is collected. Even if data collection fails part way through the call, the task will be run. The task is triggered by a call-back from a datalogger executing a P97 instruction or a SendVariable instruction.
- **After Failed Retry** - Whenever a retry fails. This can be the failure of a primary retry, a secondary retry, or a retry using a **Collect Now** button. (Scheduled collection failures and "Collect Now" failures both increment the same retry counter.)
- **After Primary Retries Failed** - After the specified number of primary retries has been exhausted. The task is triggered by a failure of primary retries from scheduled collection or from using a **Collect Now** button. (Scheduled collection failures and "Collect Now" failures both increment the same retry counter.)
- **After One Way Data** - After data is received from a datalogger executing a SendData instruction or when data is collected via Data Advise.
- **After File Closed** - After any data file being written to is closed. The task is triggered by a scheduled collection or by using a **Collect Now button**. A datalogger call-back, One Way Data, Data Advise, or another task that calls the associated station, which causes a file to be written to and closed, will also trigger the task.

  The option %f can be used in the command line options to represent the just closed data file. This condition is especially useful when performing post-processing on a data file that has been created using Create Unique File Name as the File Output Option. In this case the user does not know the file name ahead of time. Therefore, the %f option can be used to insert the file name in the command line options.

  > **NOTE:** %f returns an unquoted string. Therefore, if there are spaces in the path or filename, you will need to add quotes around the %f.

- **After Any Data Collected** - After data is collected by any means, and the call is terminated for any reason (success or failure). The task is triggered by a scheduled collection, by using a **Collect Now** button, a datalogger call-back, or another task that calls the associated station which causes any data to be collected.

- **After File Retrieved** - After a file is retrieved based on the datalogger's File Retrieval tab in the Setup Screen.

  When the station's File Retrieval Mode (on the Setup Screen's File Retrieval tab) is set to Follow Scheduled Data Collection, the task is triggered by a scheduled collection, by using a **Collect Now** button, by a datalogger call-back, or by another task that calls the associated station and causes a file retrieval. An attempt to retrieve the file(s) will be made at the scheduled time, only if scheduled collection is enabled. However, when a manual poll/Collect Now is performed, an attempt to retrieve the file(s) will be made regardless of whether scheduled collection is enabled or not.

  When the station's File Retrieval Mode is set to New Schedule, only the new schedule will trigger file retrieval and thus, the task. Attempts to retrieve the file(s) will be made following the new schedule, whether scheduled collection is enabled or not.

An "Add After" task can also be added to an existing task. In this case, the only fields applicable are the What Task Does information.

10.1.1.3 What Task Does

This tab describes the task that will be performed. Select a tab to set up the action(s) that should be performed by the task. If multiple check boxes (**Execute File, Call Station, FTP Settings**) are selected, the actions will be launched at the same time.

Execute File - Select this check box to execute a file or command when a task event is triggered. Use the browse button to the right of the **File Name** field to select the file, or type in the name and path directly. If command line options should be passed to the executable, enter those into the **Command Line Options** field. A **Start In** Directory for the executable can be typed in directly, or you can browse for it. Select the**Run Minimized** check box to have the file executed in a minimized state. When minimized, it will appear as a Windows taskbar item but will not open on your desktop.

The **File Name, Command Line Options,** and **Start In** fields can contain these predefined symbols that will be expanded by the LoggerNet server:

%a	LoggerNet working directory. (By default, C:\Campbellsci\LoggerNet.)
%w	LoggerNet server working directory. (By default, C:\Campbellsci\LoggerNet\sys\bin.)
%b	LoggerNet program directory. (By default C:\Program Files (x86)\Campbellsci\LoggerNet for 64-bit computers and C;\Program Files\Campbellsci\LoggerNet for 32-bit computers.)
%s	Name of the station that triggers the task.

%f	Name of the data file when the task is triggered by the closing of a data file (that is, by the **After File Closed** condition). See **After File Closed** above for more information. (Note that %f returns an unquoted string. Therefore, if there are spaces in the path or filename, you will need to add quotes around the %f.)
%%	Substitutes a percent character in the path.

NOTE: Always enter the full path when specifying the file to execute. Otherwise, the file may not be found or may not run as expected.

NOTE: The Task Master can process only one command in a command line. If multiple commands are required, place the entire command sequence in a batch file and execute the batch file from the Task Master. (If you are running Windows as a restricted user or if you are running LoggerNet as a service, you must explicitly set all file paths in the batch file.)

NOTE: The Task Master may not be able to process certain characters on a command line, such as the redirect (<) character. Use the Window's cmd function with the /c option to process the command (e.g., cmd /c ""cora_cmd <run.script"). The cmd function will carry out the command in the subsequent string and then terminate. You can also put the command line in a batch file and then start the .BAT file from the Task Master.

Call Station - Select this check box to trigger a call to a station when a task event is triggered. Data will be collected according to settings in the Setup Screen for the datalogger. Use the drop-down list box to select the station that will be called.

FTP Settings - This check box is only available when configuring an **Add After** task with the **Station Event Type** set to **After File Closed** or **After File Retrieved**. Select the **Enable FTP** check box to transfer the file to a designated FTP directory with the following settings:

- **Host Address** - The FTP server to which the file will be sent.
- **User ID** - The username on the FTP server.
- **Password** - The user’s password on the FTP server
- R**emote Folder** - Selects the folder on the FTP server to which the file will be transferred. The file will be saved to this folder under the FTP server’s FTP root directory. Press the button to browse to the desired directory.
- **FTP Protocol** - Use the check box to select the FTP Protocol to use. The options are **FTP (File Transfer Protocol)**, **SFTP (SSH FTP)**, and **FTPS (FTP over TLS**).
- **FTP Queue Size** - If an FTP fails for some reason, the file can be queued up to be sent in the future when the issue that caused the failure has been resolved. An FTP will only be

attempted each time the task is triggered.

The FTP Queue Size determines how many files the Task Master will keep in the queue to attempt to FTP again.

- **Extended Passive Mode** - Select the **EPSV Enabled** checkbox to enable the EPSV (extended passive mode) command instead of the PASV (passive) command for FTP.

  PASV is limited to IPv4, while EPSV works with any network protocol. However, since EPSV is an extension of PASV, it is not necessarily supported by all FTP servers. Therefore, this checkbox must be checked if the FTP server is using IPv6. Otherwise, it should be checked only if you are certain the FTP server supports EPSV.

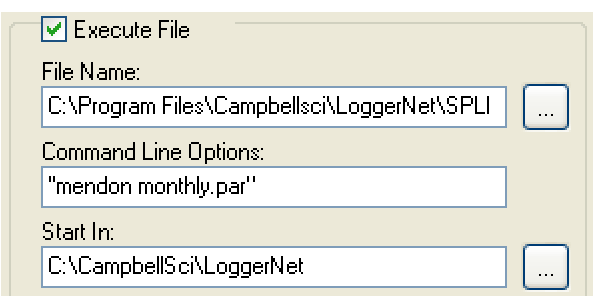
Any data file associated with the designated station will be transferred, whenever that file is closed. (Therefore, a table's File Output Option on the Setup Screen's Data Files tab must be set to anything but "No Output File" in order for the table's collected data to be transferred.) If more than one file is closed (i.e., multiple tables are collected and written), all of the files are transferred. If a failure occurs, the failure information will be written to the log file described below.

> **NOTE:** When the Task Master's Pause All Tasks option is selected, no tasks will be triggered. Therefore, any files that would have been FTP'd, if tasks were not paused, will not be added to the FTP Queue.

> **NOTE:** It is not ideal to set up an FTP task with a datalogger on a fast scheduled collection interval, because of the time it can take an FTP transaction to occur.

Example #1:

The following configuration will run Splitr.exe and process the parameter file named mendon monthly.par. If your parameter file name includes spaces (as with the example shown below), you will need to put quotes around the entire string or an error will be returned.

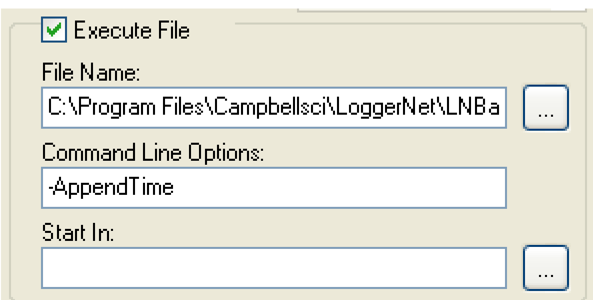


The **Start In** directory indicates the directory in which the Split parameter file is found.

NOTE: When running Splitr.exe as a task, caution should be used when using the **Run Minimized** option or when running LoggerNet as a service. Splitr.exe will remain running after the first execution of the task. At that point, Splitr.exe will not run again, until you close it through the Windows Task Manager. The only way to successfully run Splitr.exe in a minimized state or when running LoggerNet as a service is to use the /R or /Q option immediately after the PAR file name (with no space) to close Splitr.exe after the PAR file is run. With the /R option, Splitr.exe will remain open if an error occurs and will not run again, until the error is acknowledged. With the /Q option, you will get no indication if an error occurs. You will need to check Split's log files for information on any errors that have been generated. The /Q option should always be used when LoggerNet is running as a service.

Example #2:

The following configuration will run LNBackup.exe to perform an automated backup. The -AppendTime command line option creates a unique filename based on date and time each time the task is run. If the -AppendTime command line option is omitted, the backup file will be overwritten each time the task is run.



If LoggerNet security is enabled, the command line options must also include the username and password as shown below:

-user="username" -password="password"

If you have used a Command Line Arguments (p. 26) to change LoggerNet's default port number, the command line options must also include the server address and port number as shown below:

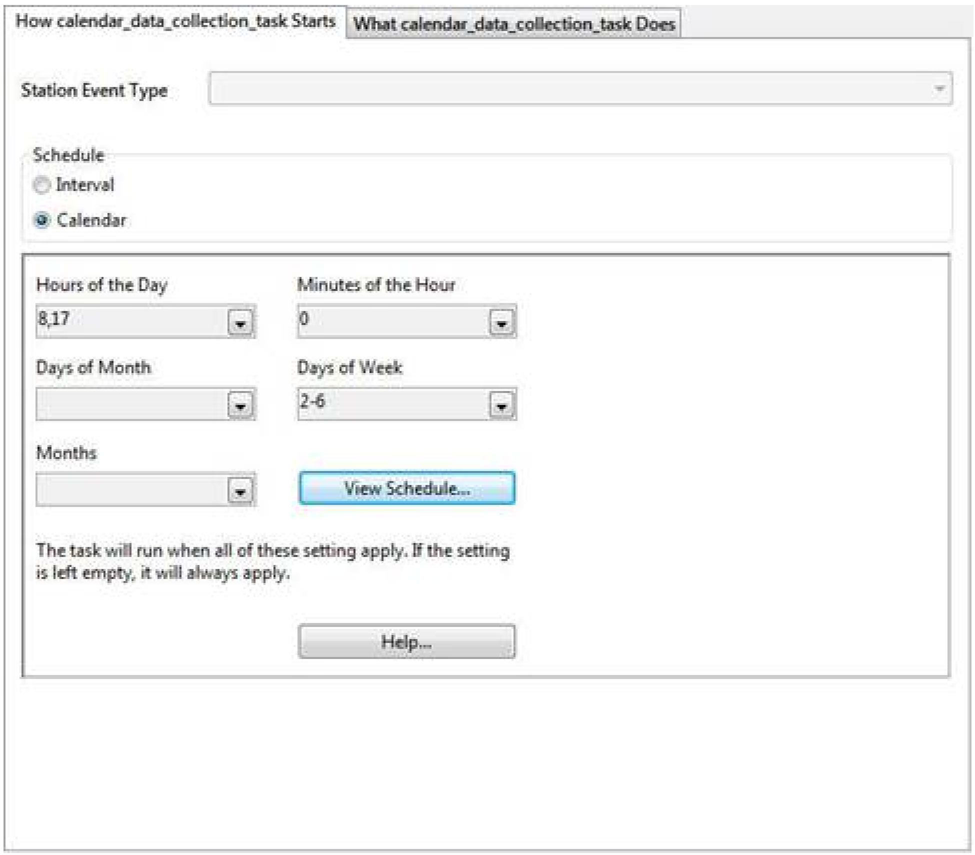
-server=server_address:port (e.g., LocalHost:6700, 192.168.7.123:6700, or [2620:24:8080:8600:85a1:fcf2:2172:11bf]:6700)

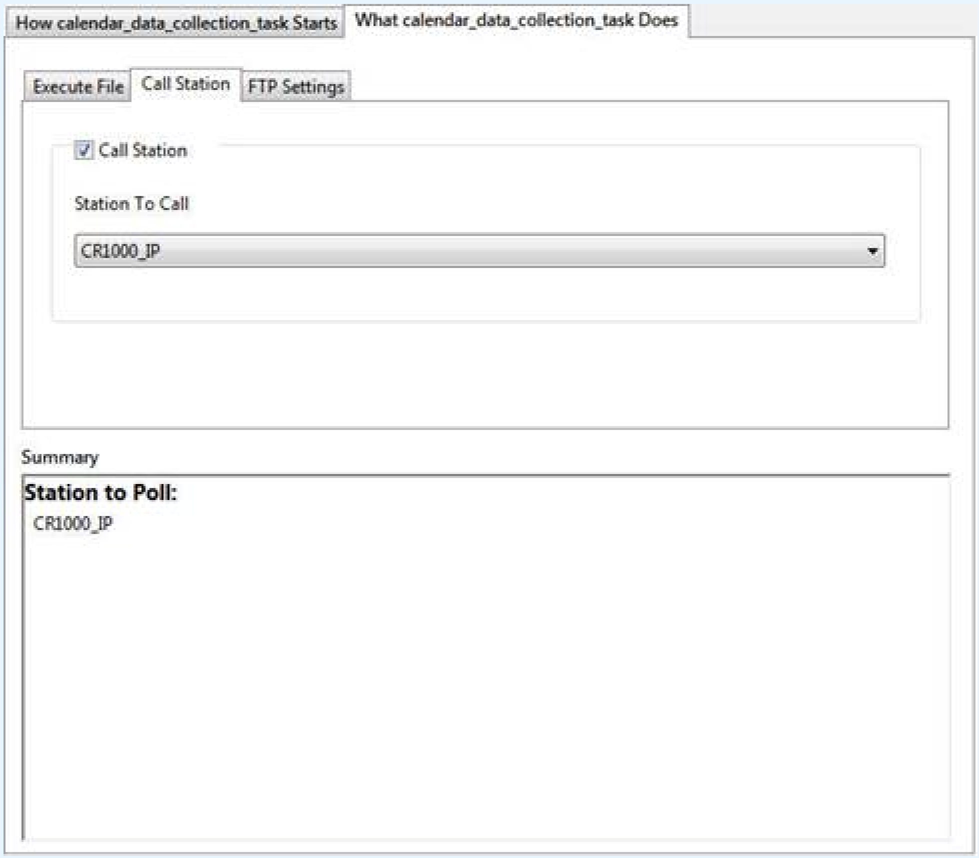
NOTE: The files contained in the backup will be based on a saved backup configuration file. To save a backup configuration, choose Network | Manual Backup from the Setup Screen's menu. Proceed through the Backup wizard. At the last step, choose Save Configuration. The configuration will be saved to C:\CampbellSci\LoggerNet\Backup.Configuration.

NOTE: Automated backups on a specified interval can be performed using the Setup Screen's Network | Scheduled Backup menu item.

Example #3:

The following configuration will set up a task (calendar_data_collection_task) to collect data from the datalogger that is named CR1000_IP in the network map. Data will be collected at 8:00 a.m. and 5:00 p.m. Monday-Friday. Note that the Data Files tab in the Setup Screen must be configured to collect the desired tables.

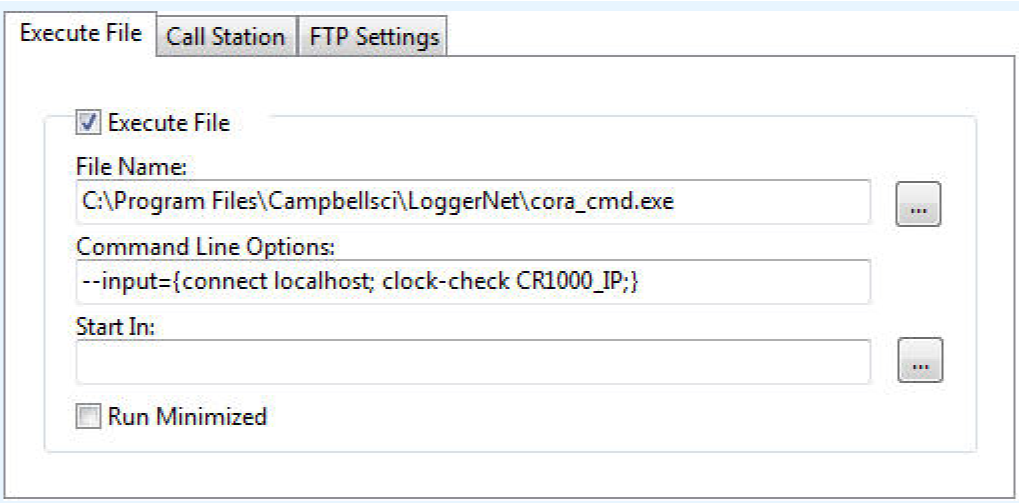




NOTE:

Example #4:

The following configuration will set up a task to perform a clock check on the datalogger that is named CR1000_IP in the network map.



NOTE:

10.1.2 Status Tab

The Status tab for the Task Master provides information on the type of task, the last time a task was run, the next time it will be run (if it is a scheduled task), and the outcome the last time the task was run.

By default, all columns are shown on the Status tab. You can select which columns are shown by choosing Edit | Select Status Columns from the Task Master menu or by right-clicking on the column headings on the Status tab and selecting Select Status Columns.

Task Name - The name that was given to the task when it was set up.

Action - Indicates whether the task will Call Station, Execute File, FTP File, or perform Multiple actions.

Event Type - This column indicates what type of event will trigger the task. It is only applicable to **Add After** tasks. The event types are those listed above (**Station Event Types**).

Event Trigger – For a **Scheduled Interval** task, the schedule for the task will be listed in the format DD HH:MM:SS followed by the word "Interval". For a **Scheduled Calendar** task, the word “Calendar” will be displayed. For an **Add After** task, the device which the task is dependent upon will be listed.

Last Time Run - The last time that the task was run by LoggerNet.

Next Time to Run - The next time that the task is scheduled to be run. If the task is not a scheduled task (interval or calendar), this field is not applicable and will be left blank. If **Pause Tasks** is selected, this field will read Paused.

Pending Actions - Species the number of actions that are currently pending for this task. This value will be zero if there are no actions currently pending. An increasing number may indicate that you are attempting to run the task faster than is possible.

Last File Run Started - The last time the attempt to execute the file was started.

Last File Run Finished - The last time the attempt to execute the file was finished.

Last File Run Outcome - The outcome of the last attempt to execute the file. This can have one of the following values:

- Failed
- Started
- Does Not Exist
- Timed Out

Last File Run Exit Code – Windows system exit code of the last attempt to execute the file.

Last Poll Started - The last time that polling of the specified station was started.

Last Poll Finished - The last time that polling of the specified station was finished.

Last Poll Outcome - The outcome of the last poll. This can have one of the following values:

- Not Polled
- Success
- Security Failure
- Communication Failure
- Communications Disabled
- Bad Table Definitions
- Task Disabled
- Datalogger is Locked
- File Write Failure
- Datalogger is not Valid

Last FTP Started - The last time the attempt to FTP a file was started.

Last FTP Finished - The last time the attempt to FTP a file was finished.

Last FTP Outcome –Outcome of the Last FTP attempt. This will have a value defined under Exit Codes here:http://curl.haxx.se/docs/manpage.html

Run Selected Task - Press this button to immediately execute the highlighted task.

Pause All Tasks- Select this check box to temporarily stop the execution of all tasks.

10.1.3 Remote Administration of the Task Master

(AVAILABLE ONLY IN LOGGERNET ADMIN/LOGGERNET REMOTE) The **File | Select Server** option allows you to select the LoggerNet server to which the Task master should connect. You also specify the username and password to be used.

For remote administration of the Task Master, the following conditions must be met:

- LoggerNet security must be enabled in the Security Manager, and the user must have Full Administrator credentials.
- Allow Remote Task Management must be enabled from the Security Manager's Edit menu.
- Allow Remote Connections must be enabled in LoggerNet Options (p. 24).

10.1.4 Task Master Logs

Messages about the activity of tasks are saved in LoggerNet's transaction log files. They show information about when a task is set up and when a task is executed. The Task messages are interspersed with other server messages in the tran.log files.

For FTP tasks, an additional log file, FTPLastTx.log, will provide details about the last FTP task that was run. It will be overwritten with each FTP event.

By default, the log files are stored in C:\Campbellsci\LoggerNet\Logs.

11. Utilities Installed with LoggerNet

Along with LoggerNet's server, clients and program editors, we also install several utilities. These are launched either from the Utilities category of the LoggerNet toolbar or from a command line calling the executable itself. These utilities include **Device Configuration Utility**, an application that uses a serial or an IP port to configure Campbell Scientific dataloggers and communications devices, **CoraScript**, a utility to configure and run LoggerNet from a command line, **File Format Convert**, an application that is used to convert data files from one format to another, and **Toa_to_tob1**, a command line utility to convert TOA5 files to the TOB1 format.

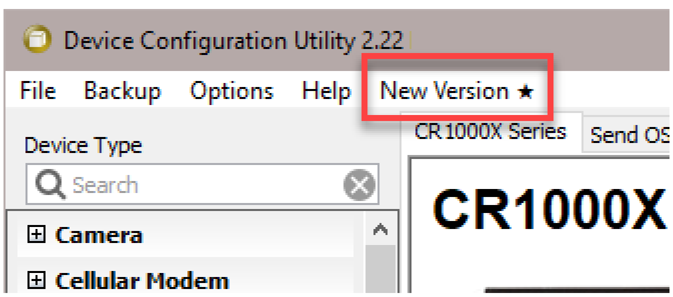
11.1 Device Configuration Utility

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:

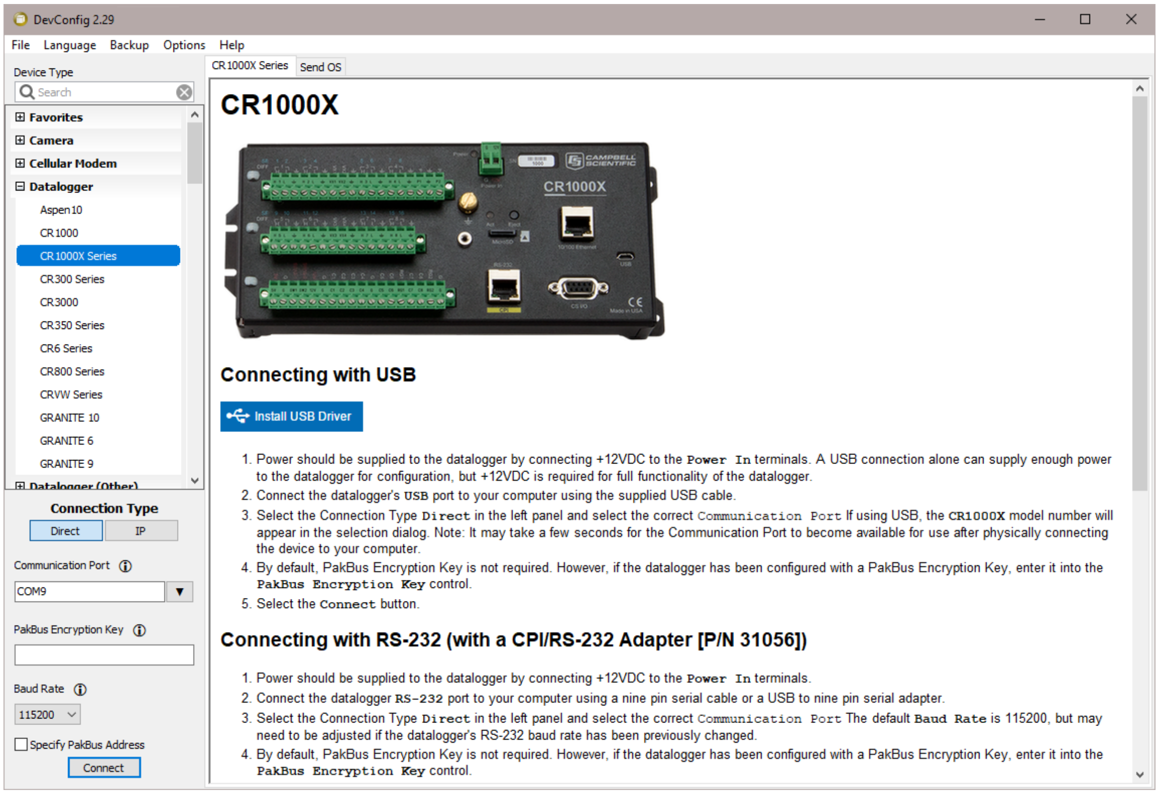
- To keep the process as simple as possible, DevConfig supports only serial and IP connections between the PC and devices.
- DevConfig cannot only send operating systems to supported device types, but can also set datalogger clocks and send program files to dataloggers.
- DevConfig allows you to determine 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.

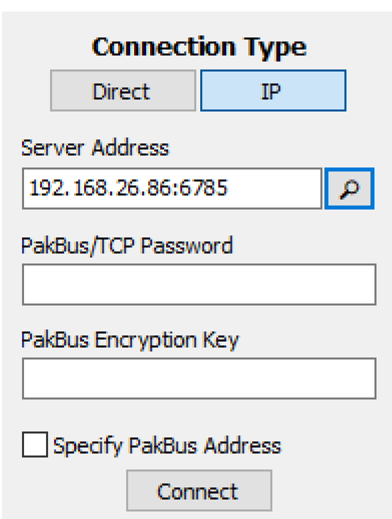
- DevConfig provides a reporting facility where a summary of the current configuration of a device can be shown on the screen and printed. This configuration can also be saved to a file and used to restore the settings in the same or a replacement device.
- Some devices may not support the configuration 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.



11.1.1 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.

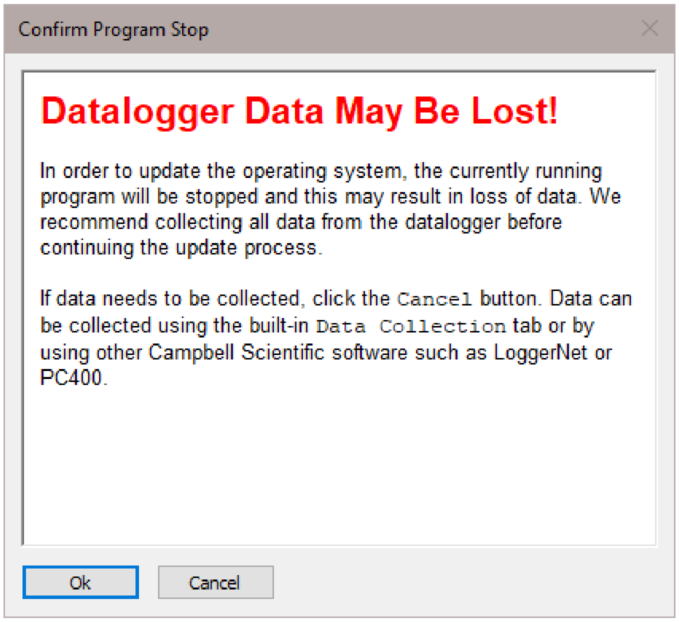
When the user presses the **Connect** button, the device type, serial port, and baud rate selector controls become disabled and, if DevConfig is able to connect to the device, the button will change from "Connect" to "Disconnect". The tabs on the right side of the window will be replaced with tabs that represent the various operations that are available for that device in a connected state. These operations can vary from device to device.

11.1.2 Downloading an Operating System

> **WARNING:** Downloading an operating system may result in data loss. We recommend always collecting data before sending an operating system.

11.1.2.1 Manage OS Tab

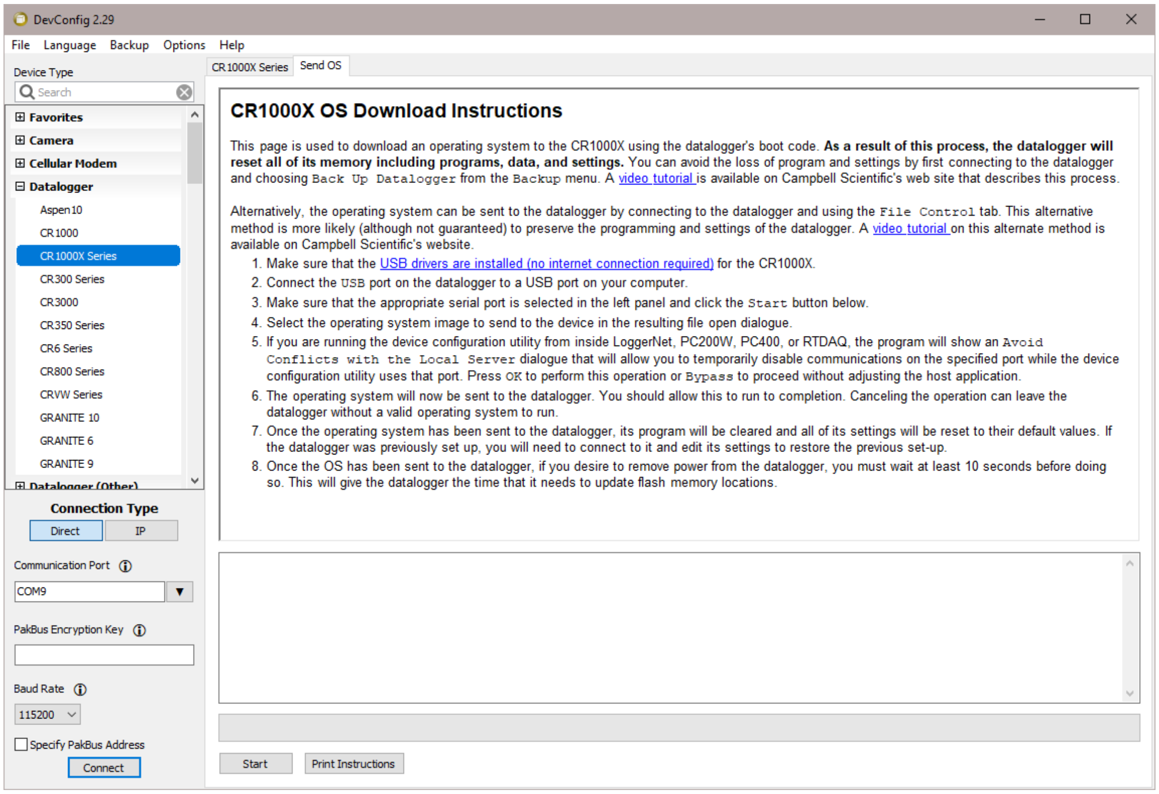
Our new data loggers have a **Manage OS** tab in DevConfig. By default, when connecting to a data logger, DevConfig checks for an updated operating system. If one is found, DevConfig automatically navigates to the **Manage OS** tab. This tab displays the changes in the new version of the operating system. 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. 345)).



11.1.2.2 Send OS Tab

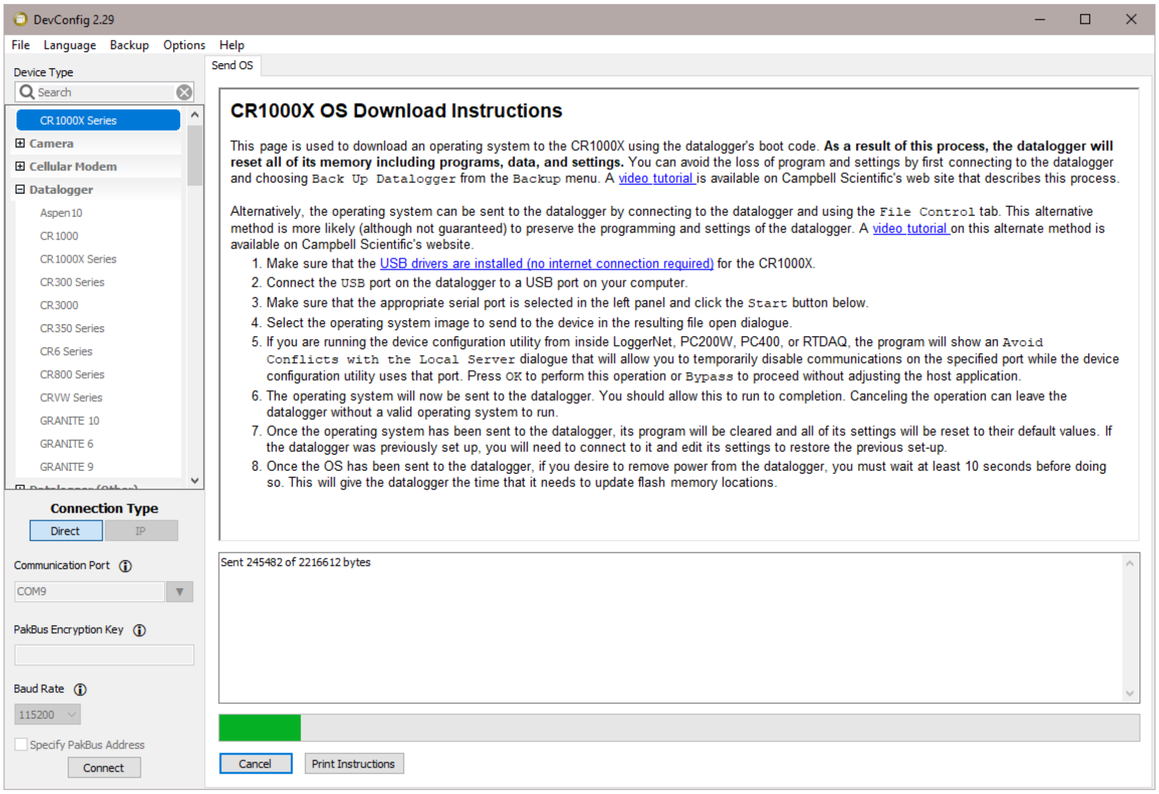
DevConfig can send operating systems from the **Send OS** tab to all Campbell Scientific devices with flash replaceable operating systems. When sending an operating system to a data logger from the **Send OS** tab, data and programs will be lost, and all settings will be returned to their default values. Therefore, before sending an operating system from the **Send OS** tab, collect your data and back up you data logger (see Backing up and Restoring a Datalogger (p. 345)).

An example for the CR1000X is shown below:

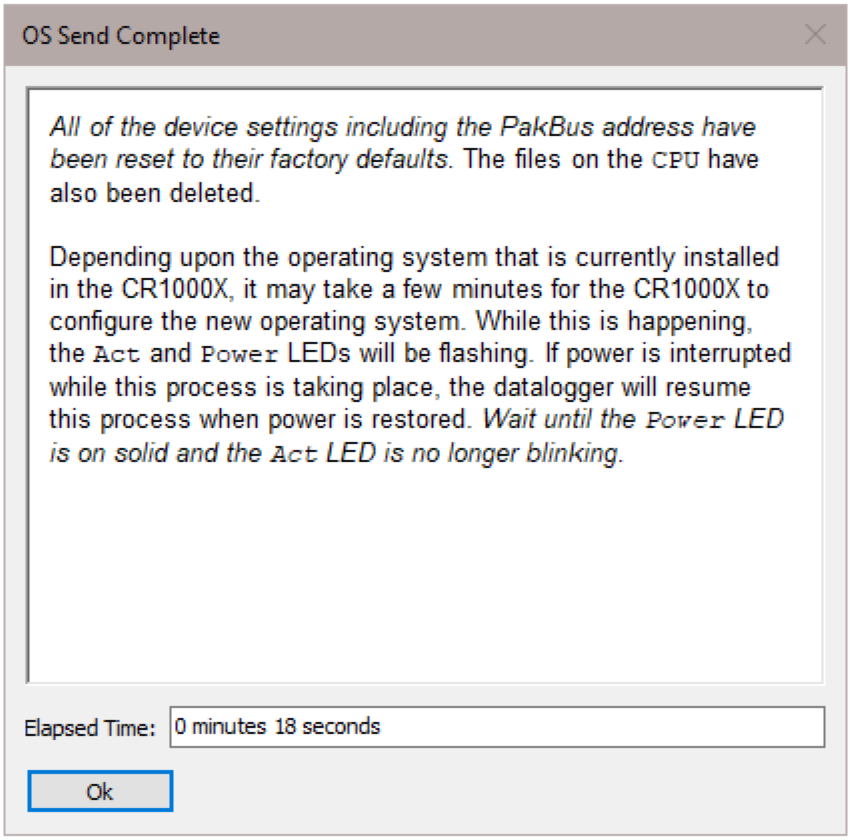


The text at right describes any interface devices or cabling required to connect the PC to the device. Screens for other devices vary only in the text on the right side. This screen differs from other screens that are available in DevConfig in that it can be accessed from either a connected or disconnected 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.

11.1.3 Security Check

A Security Check is provided through ***Device Configuration Utility***, starting with version 2.29. This check helps you identify areas where security can be improved.

All suggestions shown in ***Device Configuration Utility*** are optional and no changes will be made unless you make them. For example, ***Device Configuration Utility*** uses a simple set of criteria to suggest a strong password. If you have your own criteria, you can use it. Because every deployment can be different, ***Device Configuration Utility*** will provide you with the information you need to ensure your data logger security is optimized for your application.

In general, green, blue, and red icons indicate password strength.

- Green: strong password
- Blue: weak password
- Red: no password set

A strong password has the following:

- Eight or more characters
- One upper case letter
- One lower case letter
- One digit
- One special character

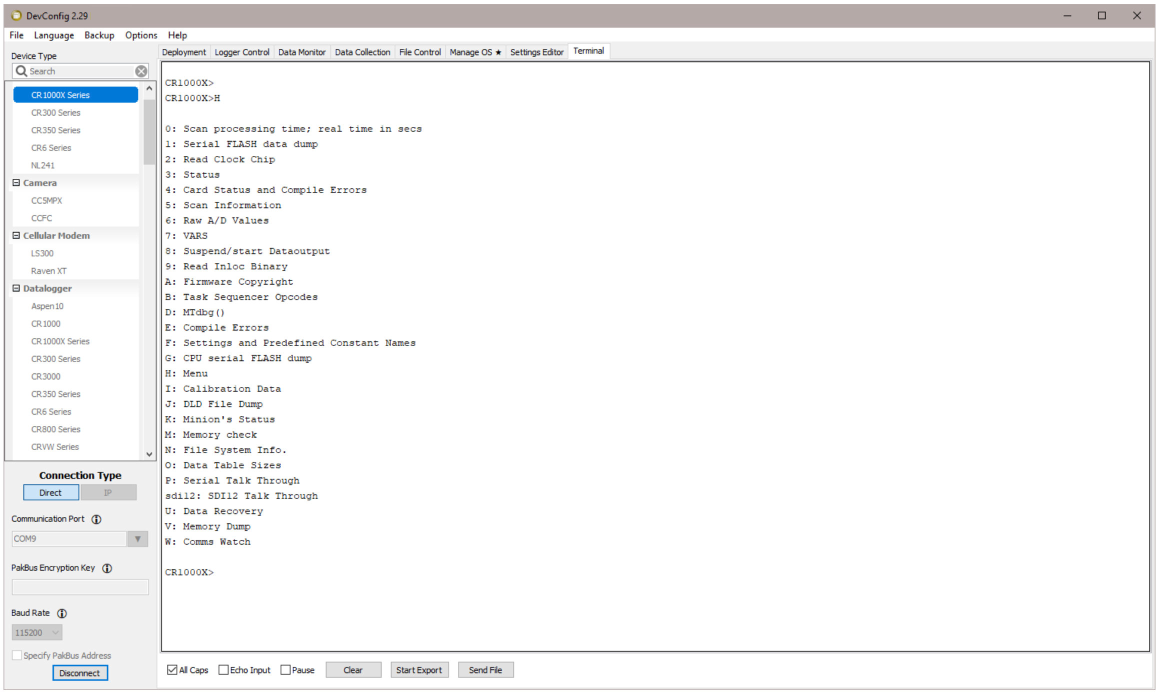
The green, blue, and red icons may also show the potential severity of a security vulnerability.

- Green: good, no action needed
- Blue: advisory information
- Red: action recommended

For more information on the Security Check in ***Device Configuration Utility*** and data logger security in general, see Data logger security.

11.1.4 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.

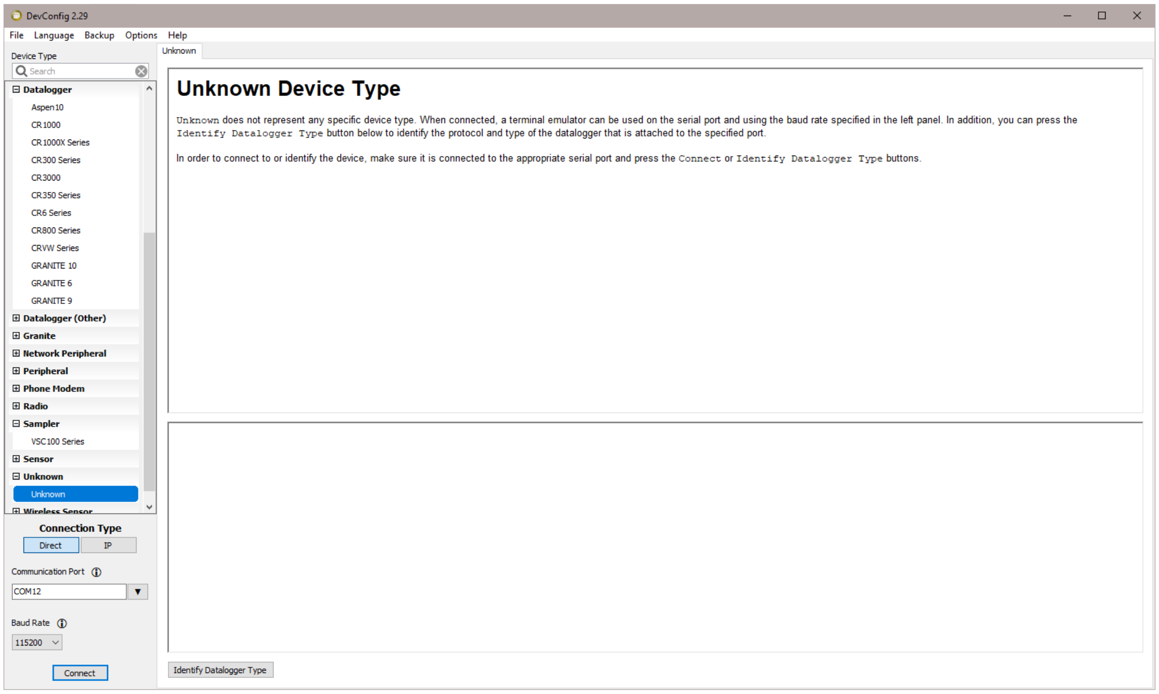


The default for the **Terminal** tab is to only show characters that are returned from the device. However, if the **Echo Input** check box is enabled, the screen will also display the characters actually typed by the user.

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.

11.1.5 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.

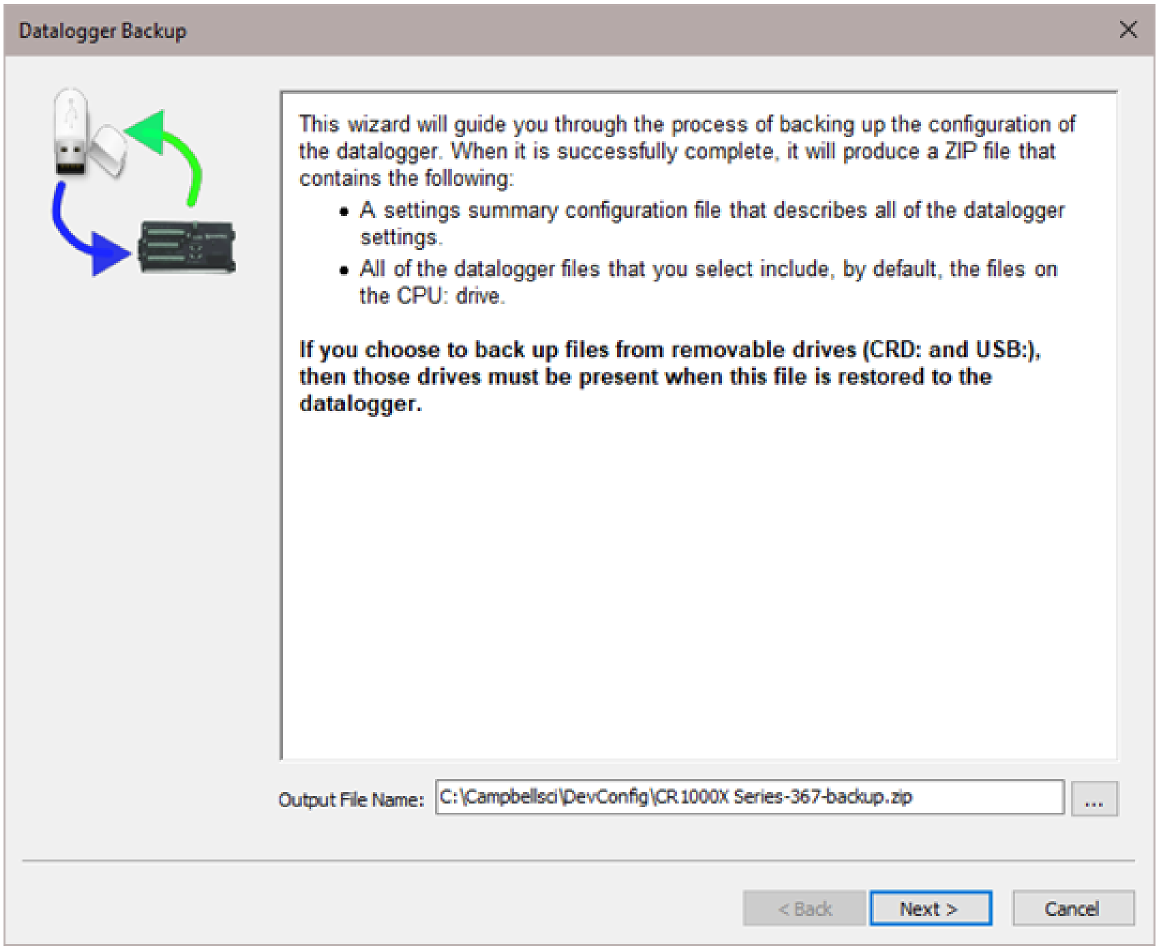
When you click on **Identify Datalogger Type**, DevConfig will attempt to identify the type of device that is connected on the specified serial port. It will attempt to communicate using each of the datalogger protocols (mixed-array, table-data, and PakBus) in turn. If it fails to get any answer to any 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.

11.1.6 Off-line Mode

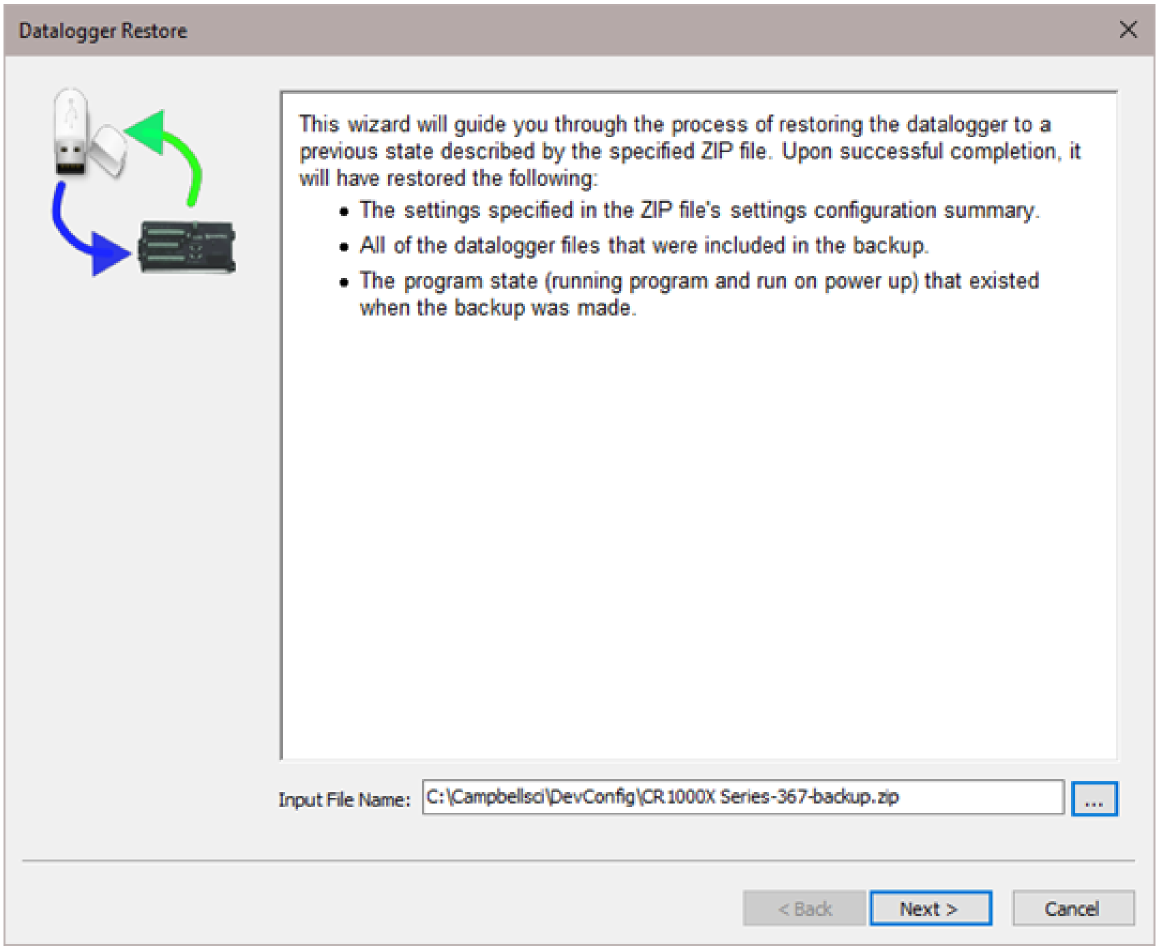
Many devices in DevConfig have an off-line mode available that allows you to browse the device's settings without actually being connected to a device. You can select a device, and then select **Off-line Mode** from DevConfig's **File** menu. You will be able to see the device's settings and associated 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.

11.1.7 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.



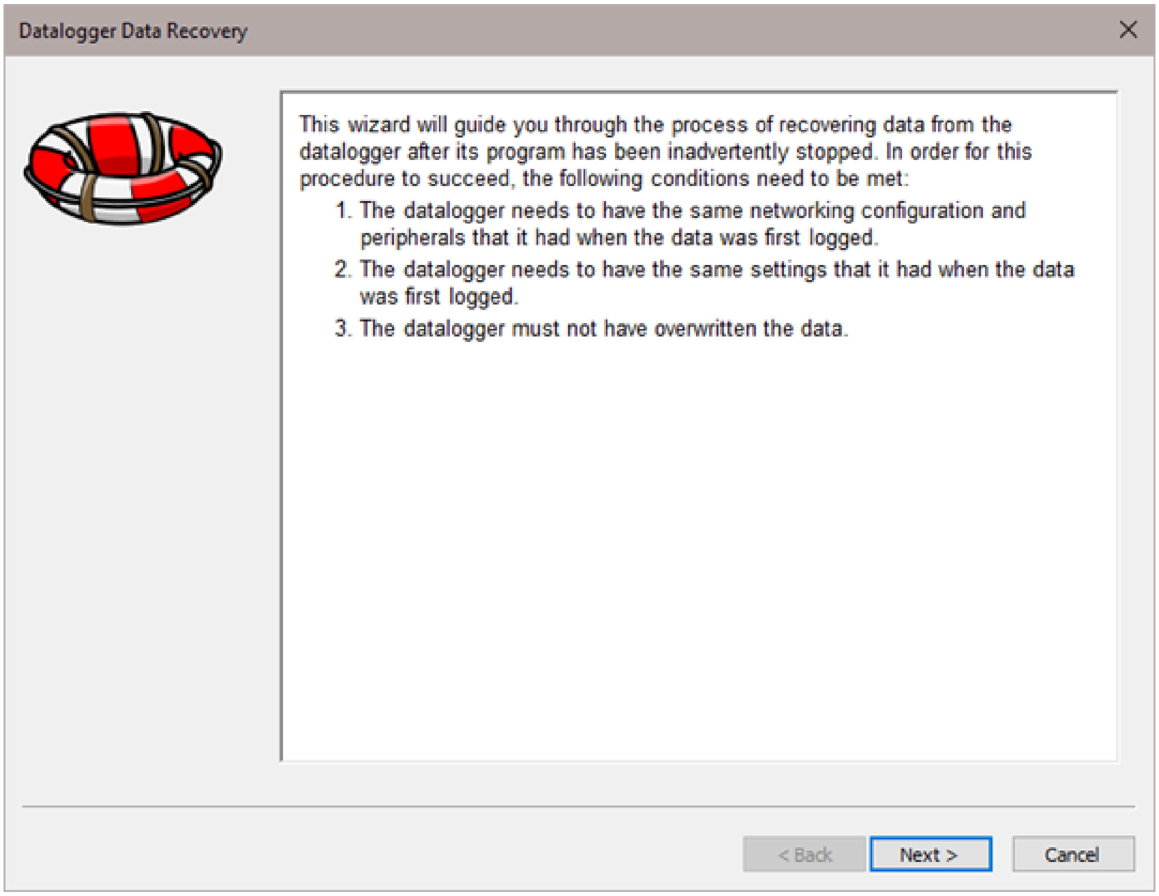
11.1.8 Data Recovery

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.



11.2 CoraScript

CoraScript is a command line interpreter that reads its commands as text from its standard input device and writes the results of those commands as text to its standard output device. This style of input and output makes it possible to externally control the LoggerNet server operation using input and output redirection. It also makes it possible to string together commands in scripts that can be executed from the command line.

The CoraScript command interpreter is started by executing the program file Cora_Cmd.exe in a command prompt environment. CoraScript is also available through the LoggerNet Toolbar's Utilities category. When the script processor starts up it will output a response:

CoraScript 1,1,1,30

The numbers indicate the version number of the current CoraScript.

CoraScript is a batch processing interpreter. It treats its input (from the standard input device) as a sequence of commands that are processed serially from the first to the last. As a command is processed, the results are written to the standard output device. A command is defined as the text up to a semicolon (;). The semicolon tells CoraScript that the command is complete and ready to execute.

The flexibility of the commands available within CoraScript and the independence from user interface considerations make CoraScript a valuable tool for testing, troubleshooting, and automating LoggerNet server operations.

> **NOTE:** Because the commands available in CoraScript operate directly on the LoggerNet server and not through a user interface, there are no confirmation prompts for critical operations. Care should be exercised in using the commands to avoid interrupting normal server operations.

There is an extensive on-line help file available for CoraScript. To bring up the help file, type "help;" on the command line. (Make sure to include the semicolon ';' at the end and leave off the quotes.) Read through the directions and try some examples.

11.2.1 Useful CoraScript Operations

The following sections provide an overview of some common and very useful commands available with CoraScript. Some rules about formatting input and interpreting the responses:

- Always end the command with the semicolon (;) character. CoraScript uses the semicolon to mark the end of the command input and will not process anything until it is detected.
- Command parameters are often set using a combination of the parameter name and the value in this format: --name=login. Be sure to read the help for the command you are using.
- A response preceded by a plus sign (+) indicates that the command was successfully processed.
- A response preceded by a minus sign (–) indicates that the command failed and will usually be accompanied by a reason for the failure.

11.2.1.1 Connecting to the LoggerNet Server

Before you can execute any commands a connection to the LoggerNet server must be established. The connect command sets up the server context in which subsequent commands will operate until the end of the script is reached or another connect command is processed. The following segment shows a sample use of the connect command:

connect	# the name of the command
localhost	# specifies the server's host address

--name="bilbo"	# specifies the logon name (optional)
--password={baggins}	# specifies the password (optional)
;	# marks the end of the command

This command would normally appear on one line as follows:

connect localhost --name="bilbo" --password={baggins} ;

For a more detailed explanation of the interpretation of the symbols and syntax refer to the CoraScript help.

11.2.1.2 Checking and Setting Device Settings

The current value of any of the configuration settings for a device is available using the *get-device-setting* command. Devices are referred to by the name as shown in the Setup Screen network map and settings are referenced by number. The setting numbers and their meanings are described in the CoraScript help.

To set the configuration setting for any device, use the *set-device-setting* command. As with *get-device-setting* the device is referred to by name and the setting by number.

11.2.1.3 Creating and Using a Network Backup File

This command is used to create a backup image of LoggerNet's working directory. The backup file will contain the exact images of LoggerNet's configuration files and, when restored, will restore LoggerNet to the exact state that existed when the backup file was created. A file created using this command can be restored using the restore-snapshot command.

> **NOTE:** You may want to consider using LoggerNet's Scheduled Backup/Manual Backup/Restore Network options available from the Setup Screen as an alternative to using CoraScript for network backups.

To create a backup file, connect to the LoggerNet server (LoggerNet must be running), and type in the following command:

```
create-backup-file;
```

If executed successfully, you will see something similar to the line below:

```
+create-backup-file, C:\CampbellSci\Loggernet\2005-3-1_14-15-01.snapshot;
```

Where the directory is the path in which the backup file is stored, and the file name reflects the date and time the snapshot was created.

Create-backup-file has three options. You can specify the path and filename instead of using the default, include additional files in the backup image that would not otherwise be saved, and specify whether or not the data cache will be stored with the image. By default, the data cache will not be saved, so it may be a good idea to include at least this option if your intent is to fully restore LoggerNet to the exact state it was in when the backup was created. In this instance, the command would be:

+create-backup-file include-tables="true";

To restore LoggerNet from a snapshot file created using the create-backup-file command, use restore-snapshot. For instance, to restore from the backup file created above:

restore-snapshot 2005-3-1_14-15-01.snapshot;

By default, when the network is restored, LoggerNet will first delete all files from the LoggerNet working directory. However, you can override this default by using the clear="false" option after the filename.

Refer to the CoraScript on-line help for more information on these two commands and their associated options.

11.2.1.4 Hole Management

There are several commands available with CoraScript to manage the hole collection process. These functions are not available through the standard user interface applications. See the CoraScript help for details about these commands.

List-holes

Purge-holes

Delete-holes

11.2.1.5 Scripting CoraScript Commands

To automate network processes, scripts can be created with other scripting language tools that would call the CoraScript interpreter, and send commands to the LoggerNet server. This provides an alternate means of controlling data collection, hole collection and maintenance functions such as clock check and set.

11.3 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 C**onvert/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.

11.3.1 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 TOACI1, 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.

11.4 Toa_to_tob1

This utility is used to convert TOA5 (ASCII Table Data) files to TOB1 (Binary Table Data) format. By default, it is located in C:\Program Files (x86)\Campbellsci\LoggerNet.

The utility is executed from a DOS command prompt as follows:

toa_to_tob1 *input_filename output_filename*

where *input_filename* is the name of the TOA5 data file

and *output_filename* is the name of the TOB1 file after conversion.

Note that if the utility does not reside in the same directory as the data files, the entire directory paths must be used. Also note that the utility will overwrite any existing file with the same name as output_filename. So, use caution in specifying the *output_filename*.

12. Utilities Installed with LoggerNet Admin and LoggerNet Remote

The LoggerNet Admin and LoggerNet Remote packages include several additional utilities or client applications that can be useful for the management of larger networks. The main difference between these two packages is that LoggerNet Admin includes the LoggerNet server and client applications/utilities, while LoggerNet Remote includes only the clients/utilities. LoggerNet Remote was developed specifically to provide remote management capabilities for an existing LoggerNet network.

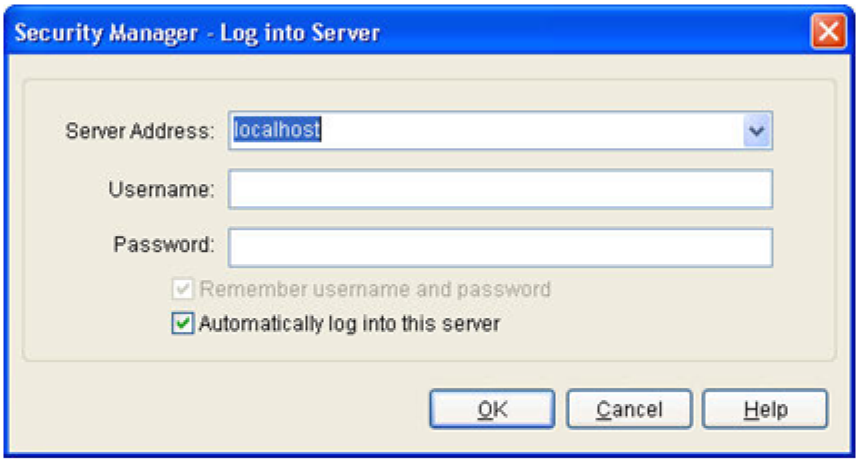
The utilities installed with LoggerNet Admin and LoggerNet Remote are Hole Monitor, Data Filer, Data Export, LN Server Monitor, LoggerNet Service Manager, and Security Manager. The LoggerNet Service Manager, which allows LoggerNet to be run as a service, is discussed in the installation notes (Installing/Running LoggerNet as a Service (p. 8)). The LN Server Monitor, which monitors the status of a remote LoggerNet server or a LoggerNet server being run as a service, is discussed in LoggerNet Server Monitor (p. 194). The remaining utilities are discussed below.

12.1 Security Manager

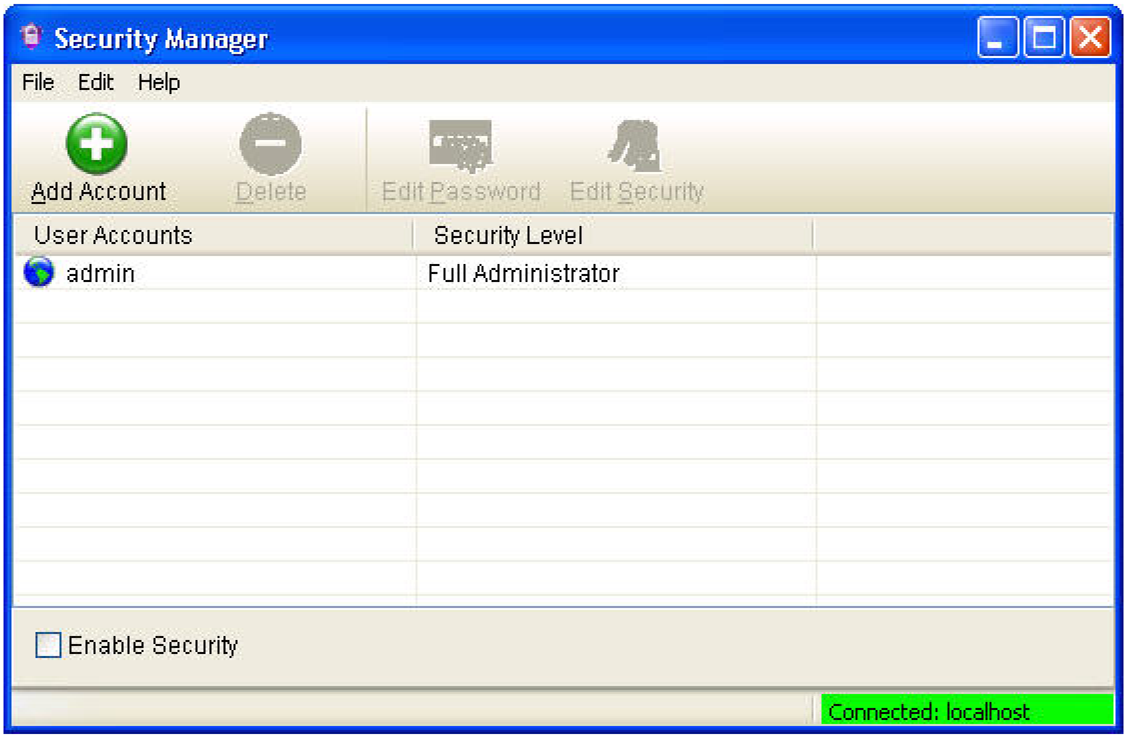
The Security Manager allows you to set up security for the LoggerNet Server, restricting access to certain functions. A User Account is set up for each individual using the system. The User Account is given a user name and password, and assigned one of the five levels of security. The user must then enter the name and password when opening a LoggerNet client. Any options that are not available to that user based on security settings will be disabled in the application.

12.1.1 Initial Configuration of Security Manager

When the Security Manager is opened for the first time, a login screen is displayed.



Enter the IP address or alias for the LoggerNet server (e.g., LocalHost), leave the User Name and Password fields blank, and press **OK**. A wizard is launched to help you set up an Administrator Account, which will be used for managing the security for the LoggerNet network. Follow the instructions on screen to set up the account. Once the setup is complete, the Security Manager will display its main window, and from here, you can begin setting up user accounts.



When setting up new accounts, one of five levels can be assigned to each user. Multiple accounts with Full Administrator rights can be set up, if desired. Only users with Full Administrator rights can open and make changes in the Security Manager (regardless of whether or not security is enabled).

Once the security accounts have been set up, select the **Enable Security** check box to turn on security for the LoggerNet server.

12.1.2 Managing User Accounts

12.1.2.1 Adding an Account

An account is set up for a new user by selecting the Add Account button from the Security Manager's main window. A New Account dialog box is opened. The fields for this box are:

- **Account Name** - Enter the name to be used for the account. This name will be typed in each time the user connects to the LoggerNet server using a client application.
- **Password** - Enter the password for the account. Passwords are case sensitive. As you type, each character will be represented on the screen with an asterisk.
- **Confirm Password** - Enter the password for the account a second time.

- **Security Level** - Use the list box to select one of five security levels for the user:
  - **Read Only** – The user can view data values and status information but has no other rights.
  - **Operator** – The user can view data values, check the clock, and collect data. He cannot make changes to the datalogger program, datalogger settings, or the server settings.
  - **Station Manager** – The user can view data values, check and set the clock, and collect data. The user can send programs to the datalogger and change datalogger settings. The user cannot make changes to the server.
  - **Network Administrator** – The user has full access rights in all LoggerNet clients, except the Security Manager.
  - **Full Administrator** – The user has full access rights in all LoggerNet clients, including the Security Manager.

If an option in the LoggerNet user interface is not applicable for the security level of the user logged in to LoggerNet, that option will be disabled. The following table provides an overview of the functions available to each level of security.

Security Manager Access Table

	Read Only	Operator	Station Manager	Network Admin	Full Admin
General					
Change servers	X	X	X	X	X
Update Table Defs			X	X	X
Open Terminal Emulator			X	X	X
Setup Screen					
Make changes to LoggerNet				X	X
Make changes to datalogger network				X	X
Change device settings			X	X	X
Back-up network					X

Security Manager Access Table					
	Read Only	Operator	Station Manager	Network Admin	Full Admin
Restore Network					X
Connect Screen					
Connect to datalogger		X	X	X	X
Manual data collection		X	X	X	X
Custom data collection		X	X	X	X
Send program			X	X	X
Retrieve program		X	X	X	X
Check clock		X	X	X	X
Set clock			X	X	X
View/configure numeric display or graph	X	X	X	X	X
Change values displayed on data displays		X	X	X	X
View/configure ports/flags	X	X	X	X	X
Toggle ports/flags		X	X	X	X
View Status Table		X	X	X	X
Change Status Table		X	X	X	X
Reset data tables via Station Status/Table Fill Times window			X	X	X
File Control		X	X	X	X
Associate Program			X	X	X
Status Monitor					

Security Manager Access Table					
	Read Only	Operator	Station Manager	Network Admin	Full Admin
View Status	X	X	X	X	X
Toggle schedule			X	X	X
Reset device			X	X	X
Access to Log Tool	X	X	X	X	X
Perform Comm Test		X	X	X	X
Modify Statistics Viewed	X	X	X	X	X
Add/Modify View	X	X	X	X	X
Pause scheduled collection				X	X
Collect Data		X	X	X	X
PakBus Graph					
View Networks	X	X	X	X	X
Retrieve network settings			X	X	X
Retrieve device settings			X	X	X
Change device settings			X	X	X
Tasks					
Open Task Master	X	X	X	X	X
Add/Modify tasks					X
Pause Tasks					X
Run Tasks on Demand					X
RTMC Runtime					
Run	X	X	X	X	X

Security Manager Access Table					
	Read Only	Operator	Station Manager	Network Admin	Full Admin
Edit Values		X	X	X	X
LoggerNet Service Manager					
Install/uninstall service	X	X	X	X	X
Stop/start service	X	X	X	X	X
Troubleshooter					
Open TroubleShooter	X	X	X	X	X
Retrieve Datalogger Status		X	X	X	X
Find ID		X	X	X	X
Com Test		X	X	X	X
RF Test		X	X	X	X
Other Clients					
Hole Monitor (full access)			X	X	X
Data Export (full access)	X	X	X	X	X
Security Manager					X

12.1.2.2 Deleting an Account

To delete an account, highlight it and press the Delete button. When you are logged in to the Security Manager under the Administrator Account, you cannot delete that account. To delete it, log in under a different account with Full Administrator rights.

12.1.2.3 Editing a Password

To edit the password for an account, highlight that account, press **Edit Password** and enter the new information in the resulting dialog box.

12.1.2.4 Changing Security Levels

To change the level of security for an account, highlight that account and press **Edit Security**. Select the new security level from the list box.

12.1.2.5 Special Access

Users who have a security level of Read Only or Operator can be granted Station Manager access to selected datalogger stations. To do this, highlight the user and select **Edit** > **Advanced**. From the resulting dialog box, select one or more stations to grant access to by moving them from Stations Available field into the Selected Stations field.

12.1.3 Remote Task Management

The Security Manager's **Edit** > **Allow Remote Task Management** menu item must be enabled in order for the Task Master to be administered remotely. See Remote Administration of the Task Master (p. 334) for more information.

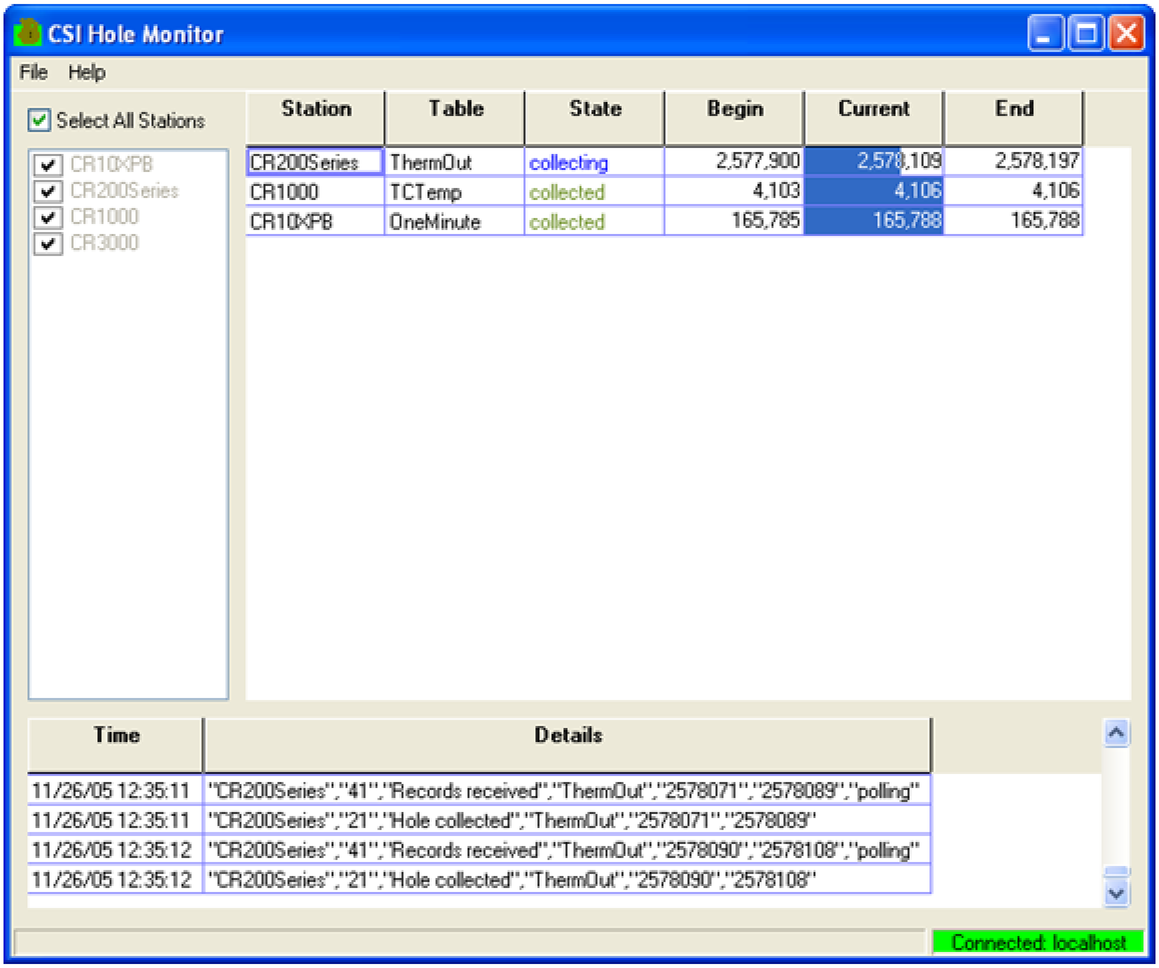
12.2 Hole Monitor

The LoggerNet Admin Hole Monitor Utility is used to monitor the hole collection activity for the dataloggers in a LoggerNet network.

A hole is any discontinuity of data in the LoggerNet server's data cache for a datalogger. Holes can occur if the server is unable to collect data from a datalogger because of communication failure, or if packets sent to the server from the datalogger are out of order because of a marginal communications link. If the data can be retrieved from the datalogger, then it is a collectable hole. If the data has been overwritten by the datalogger, then it is an uncollectable hole.

12.2.1 Hole Collection Activity

The main window for the Hole Monitor is shown below.



The list of stations in the LoggerNet datalogger network is displayed on the left side of the Hole Monitor Utility's main window. You can monitor hole collection for all stations by enabling the Select All Stations check box above this list, or you can monitor a subset of these stations by clearing this check box and selecting the check box to the left of each datalogger that you want to monitor. If a hole is detected in the data for a datalogger that is being monitored, an informational record for the hole will be displayed on the right side of the window. The fields in the record are:

- **Station** - The name of the datalogger for which a hole has been detected.
- **Table** - The name of the table in the datalogger that has the hole.
- **State** - The current state of the hole. The state options are:
  - **detected** – This state, printed with black text, indicates that the hole has been detected but attempts have not yet been made to collect it.

- **collecting** – This state, printed with blue text, indicates that LoggerNet has made attempts to collect the hole since it was detected by the hole monitor application.
- **collected** – This state, printed with green text, indicates that the server has succeeded in collecting the hole. Once the hole is collected, the information will be kept in the grid for a period of about fifteen seconds and then it will be removed.
- **lost** – This state, printed in red text, indicates that a hole could not be collected because the data records no longer exist in datalogger. This can happen when the datalogger overwrites its oldest records before the LoggerNet server was able to collect those records.

  Quite often, a portion of a hole will become uncollectable while a portion remains collectable. When this occurs, a new informational record will be created for the “lost” hole and the range for the collectable hole will be adjusted. The informational record for a lost hole will be displayed for approximately fifteen seconds and then deleted.

- **Begin** - The beginning record for the hole that has been identified.
- **Current** - The record currently being collected for the hole that has been identified.
- **End** - The last record in the hole that has been identified.

12.2.2 Message Log

The messages displayed at the bottom of the screen are messages that are recorded in the LoggerNet server’s transaction log (tran$.log usually found in C:\Campbellsci\LoggerNet\Logs) regarding hole activity. The time of the message is displayed on the left of the screen and the message itself is displayed on the right. The types of messages are:

Message	Description
hole detected	The server has detected a new range of records that need to be collected.
hole collected	The server has succeeded in collecting a range of records.
hole lost	The server has determined that the range of record numbers are no longer collectable.
hole collect started	The server has started a data collection operation that will involve a range of records.
hole collect failed	An automated hole collection attempt has failed.
records received	Records have been received from the datalogger.

12.3 Data Filer

Data Filer is a LoggerNet application that is used to retrieve data from the LoggerNet data cache, and save the data to a file. It provides a means for a user to manually retrieve and store ASCII data on a remote PC, which can then be used for further analysis.

12.3.1 Data Filer Requirements

Data Filer is an application that is capable of accessing the data in the LoggerNet data cache and storing that data to a file. The Data Filer can run on the same computer as the LoggerNet software, but more commonly it connects to a LoggerNet server computer over a TCP/IP connection. The LoggerNet server must be configured to allow remote connections; this is set up during the installation of the LoggerNet software (see Allowing Remote Connections to the LoggerNet Server (p. 380) for additional information on allowing remote connections).

Because Data Filer retrieves data from LoggerNet's data cache (and not the datalogger directly), LoggerNet must first collect the data from the datalogger before it is available for use by the Data Filer. Data collection in LoggerNet can be performed manually by a user or automatically by setting up a data collection schedule.

For information on collecting data from a datalogger, refer to Real-Time Tools (p. 80). For a description of LoggerNet's data cache, refer to Software Organization (p. 437).

12.3.2 Using the Data Filer

12.3.2.1 Connecting to a Computer Running the LoggerNet Server Software

When Data Filer is first opened, it will prompt you for the Server Address, Username, and Password of the LoggerNet server:

- **Server Address** – The name of the computer on which the LoggerNet software is running. This must be the valid name of an existing computer or a TCP/IP address (in the form ###.###.###.### consisting of the IP network number, ###.###.###, and the host number, ###). If the LoggerNet server resides on the same computer as the Data Filer application, you can simply type in LocalHost for the server name.
- **Username** – Your user name on the LoggerNet server.
- **Password** – Your password for the LoggerNet server.

> **NOTE:** The **Usernam**e and **Password** fields are required only if security has been set up on the LoggerNet server to which you are trying to connect.

Each time you start the Data Filer, you will be prompted to enter this information. However, the **Automatically log in to this server** check box can be selected to skip this window and use the information from the last session.

To specify a different LoggerNet server, select the **File** > **Select Server** menu option.

Setting up the Data Filer

Once connection to the LoggerNet server has been established, a list of dataloggers set up in LoggerNet will be displayed in the **Stations** field (left side of the window). To retrieve data for a particular station, use the mouse pointer to select the datalogger then set up the **Collection Options** (explained below), select one or more tables to be collected, and press the **Start Collection** button. The retrieved data will be stored to the directory and file name shown in the **File Name** field. The directory or file name can be changed by highlighting the table and pressing the **Change File Name** button.

Tip: Quickly choose all tables for the highlighted datalogger by selecting the Select All check box.

12.3.2.2 Collection Options

Collect Mode

This option is used to specify what data will be retrieved from the LoggerNet data cache and stored on the remote computer by the Data Filer:

- **All the Data** – Retrieves all records from the selected tables.
- **Data Since Last Collection** – Retrieves all uncollected records from the selected tables.
- **Newest Number of Records** – Retrieves a specific number of records from the selected tables by backing up the number of records entered in the **Number of Records** field and retrieving all data forward.
- **Specific Records** – Allows you to specify a beginning record number and the number of records to collect after that record. The range of records to retrieve is specified by completing the **Starting Record #** and **Number of Records** fields.
- **Data from Selected Date and Time** – Allows you to specify a s**pan of time for data collection. When this option is selected, the Starting Date/Time and** Ending Date/Time fields will be enabled.

File Mode

This option is used to determine how data will be stored in relation to existing data files with the same name:

- **Append to End of File** – Adds new data to the end of the existing data file.
- **Overwrite Existing File** – Replaces the existing file with a newly created file.
- **Create New File** – Renames the existing file with a *.bak extension, and stores the new data with the specified file name. Subsequent *.bak files will be named *.bak1, *.bak2, etc. The most recently *.bak file will have the highest number.

File Format

This option is used to determine the format in which the data file will be saved:

- **TOACI1** – Data is stored in a comma separated format. Header information for each of the columns is included.
- **TOA5** – Data is stored in a comma separated format. Header information for each of the columns is included, along with field names, units of measure (if they are available), and output processing types (average, sample, total, etc.).
- **TOB1 (binary)** – 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.
- **CSV** – Data is stored in a comma separated format, without any header information. This format is easily imported into spreadsheet applications.

- **CSIXML** – Data is stored in XML format with Campbell Scientific defined elements and attributes. For additional information, refer to Campbell Scientific File Formats (p. 410).
- **ASCII Table Data, No Header** – Data is stored in a comma separated format. No header information is included in the file.

Starting Record Information

The Starting Record Information is applicable if the Collect Mode is **Newest Number of Records** or **Specific Records**.

- For **Newest Number of Records**, enter a value into the **Number of Records** field. Data collection will include the number of records specified, prior to and including the last record stored (i.e., back up X number of records from the last record stored, and collect all records from there).
- For **Specific Records**, enter values into the **Starting Record #** and **Number of Records** fields. The **Starting Record** is the first record that will be collected from the datalogger; data collection will continue until the number of records specified have been received.

Record Information

The Record Information is applicable if File Format is TOA5, TOB1, CSIXML, or ASCII Table Data, No Header.

Select the **Include Timestamp** check box to have timestamps included in your data. If the check box is not selected, timestamps will not be included.

Select the **Include Record Number** check box to have record numbers included in your data. If the check box is not selected, record numbers will not be included.

Starting Date/Time and Ending Date/Time

The **Starting Date/Time** and **Ending Date/Time fields** are used when the **Collect Mode** is **Data from Selected Date and Time**. The two fields are used to specify a range of records to collect, based on the records' time stamps.

To complete a date field, type in a date directly or click the arrow to the right of the field to display a calendar from which to choose a date. To complete a time field, type in the time directly or use the arrows to the right of the field to increase or decrease the highlighted time value.

12.3.3 The Collected Data

After data is collected for one or more tables, a Summary window will show the table name in the datalogger, the number of records stored, the first and last timestamps of the collected data, and the first and last record numbers of the collected data.

The stored file can be viewed by pressing the **View Data** File button. The Data Filer uses LoggerNet's View Pro utility to display the ASCII file.

12.3.4 Determing the Data Available in the Data Cache

When a datalogger is selected in the **Stations** list, you can press the **View Data Info** button to display a Data Information table that indicates the number of records and range of record numbers in the LoggerNet data cache for each table in the datalogger. These are the records that are available for collection and storage by the Data Filer. The Data Information table is retrieved from LoggerNet when the window is opened. It can be updated by pressing the **Refresh** button.

- **Table Name** – The name of the data storage table in the datalogger.
- **# of Records** – The number of records in LoggerNet's data cache for the table. By default, the size of the data cache for each datalogger table is set to two times the size of the table in the datalogger. Once a datalogger table in the data cache has reached its defined size, the oldest record is deleted from the data cache when the newest one is written.
- **Earliest Timestamp** – The time stamp of the first record in the data cache.
- L**atest Timestamp** – The time stamp of the last record in the data cache.
- **Earliest Record #** – The record number of the first record in the data cache.

Latest Record # – The record number of the last record in the data cache.

> **NOTE:** Because the data cache is updated based on data collection from the datalogger, there could be additional records stored in the datalogger's memory which have not yet been retrieved to the data cache.

12.3.5 Record Number Anomalies

Under certain circumstances it may appear there is a problem with the number of records and their record numbers reflected by the Data Information table. It is possible for the oldest record to have a record number higher than the newest record. This is due to a combination of events.

Tables in dataloggers are configured as ring memory. Eventually, they will fill and the oldest records will be overwritten with newer ones. The LoggerNet data cache, too, is configured as ring memory, but sized to hold twice the number of records that can be stored in the datalogger (default size). When the datalogger compiles its program, it starts with record number 0; therefore, if something causes the datalogger to recompile its program (such as sending a program to the datalogger or using a keyboard display to alter the program slightly) all of its tables will start with record number 0 again. Therefore, the record numbers reflected in the Data Information table may appear to be incorrect.

As an example, if the datalogger's internal table size were 100 records, LoggerNet's cache would be sized at 200 records. If both had rung around and LoggerNet's cache now held record numbers 201–400 and someone re-sent the same program to the datalogger, LoggerNet would not clear its data cache, but would continue to store the new records. These record numbers, however, would start at 0. After a short while as the new records were put into the data cache and old ones overwritten, the earliest record in the data cache might be 251 while the newest record number might be 50. In the data file, however, data would appear in correct sequence ordered by date/time stamps.

12.3.6 Communication Status

A box in the lower right corner of the Data Filer's window provides an indication of the Data Filer's connection with the LoggerNet server. When the Data Filer is in communication with the LoggerNet server, the box will appear green and the IP address or computer name (e.g., LocalHost) will be displayed. If communication with the server is lost (for instance, if LoggerNet is closed), the box will appear red with the text "No Connection". If communication is lost but the Data Filer is attempting to reconnect, the box will appear blue with the text "Attempting Connection".

12.4 Data Export

The Data Export client provides a way to export the data collected by the LoggerNet communications server to another computer program. In this role the Data Export application acts as both a client and a server. It is a client to the LoggerNet server and gets data from the LoggerNet data cache. It works as a server to provide data to customer supplied client applications.

12.4.1 Functional Overview

The data to export is specified by selecting tables from each of the stations in the datalogger network. When a data table is selected for export, every record that the server collects from the datalogger for that table is sent out by Data Export. If a table is selected but for some reason there is no data being collected by the server, no data will be sent.

Once the data tables to be exported are specified, the user selects an output socket port and the export utility will begin “listening” for a request from a remote application to send data. When the connection to the application is established, data export is initiated.

The options that determine the operation of Data Export are set from the dialog box opened from Data Export’s **Edit** > **Options** menu item. There are five options as described below:

- **Listening Port Number** – The Listening Port Number is the port number that Data Export will monitor for a request for data. The default port number is 1200; this can be changed to any valid four-digit port number.
- **Starting Options** – There are two options for choosing what should be the first record exported when data export is first started. If the Get All Data option is chosen, Data Export will attempt to export all available data from the data cache for the specified datalogger tables. When Start with Newest Record is chosen, export will begin with the most recent record. In this instance, no historical data will be exported from the data cache. Note that this setting applies only to the first time data export is initiated for a table. Subsequent data export sessions will begin exporting after the last known exported record.
- **Collection Options** –Holes are discontinuities in data that is being collected via data advise. There are two options for Data Export behavior when a hole is encountered. If Wait for Any Holes is selected Data Export will wait until data holes are filled or become uncollectible before exporting the next record. Thus, the utility attempts to export data in record order. If Collected Order is selected, Data Export will export records as they are collected by the LoggerNet server. With this option, it is possible that data records are exported out of order.
- **Data Format** – This is the format in which the data should be exported. If the RTMS Format option is selected, the data is formatted to be received by an RTMS compatible computer. RTMS (real-time monitoring software) is a format developed by CSI for communication between OS/2 operating systems and table-based dataloggers. If Standard Format is selected, the data is formatted as an ASCII comma separated record format that includes header information. The protocols for both formats are described in later in this section.

- **Resending Options** – If a message is exported and there is no response or an incorrect response is received from the remote application, Data Export will resend the data. The Resending options are used to set the time interval on which the data will be resent (Time to Wait before Resending the Data) and the number of times Data Export will attempt to send the data and receive a valid response (Number of times to Resend the Data).

You can run multiple instances of the Data Export application by specifying a different initialization directory for each instance. This is done by adding the directory information to the command line of the shortcut that starts the application. An example of this command line would be:

C:\Program Files (x86)\Campbellsci\LoggerNet\SocketDataExport.exe directory *pathname*

where "directory" is a keyword indicating that the next parameter *pathname* is a valid directory path on the computer file system. Each instance of Data Export started in this manner will save it's setting in a separate *.ini file. This initialization file is saved to the directory specified by the *pathname* command line argument.

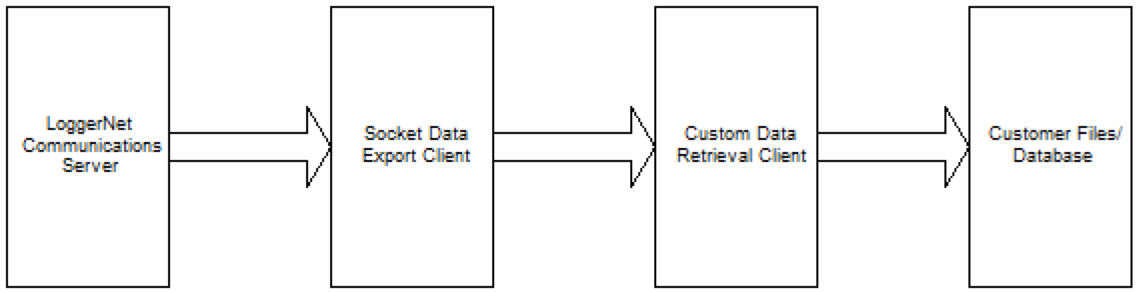
For example a shortcut with the following as the command line in the "Target" window would start Data Export using the initialization file stored in the directory "c:\Campbellsci\LoggerNet\SD1".

C:\Program Files (x86)\Campbellsci\LoggerNet\SocketDataExport.exe directory c:\Campbellsci\LoggerNet\SD1

12.4.2 Theory of Operation

The Data Export client is used in conjunction with TCP/IP Berkeley sockets network transfer protocol to transfer datalogger records from one computer (or process) to another. In this role the Data Export is acting as both a client and a server. The Data Export Client attaches to the LoggerNet communication server and gets the selected data from the data cache. It then makes this data available for retrieval on a TCP/IP socket. The computer program that retrieves the data (the custom data retrieval client application) must connect to the provided socket. The Data Export application acts as a server for the custom data retrieval client.

The most typical use for the Data Export functionality is a situation where the customer has a database or file system that is already integrated with data management procedures. The custom data retrieval client gets the data from the socket provided by the Data Export and writes it to the customer's database or file.



The LoggerNet server has the responsibility to see that every collectable record is collected from the network of dataloggers. The collected data is stored in the data cache of the server. When the Data Export client is first initialized it sets up the socket and then waits for a data retrieval client to connect. Once the data retrieval client connects, the Data Export client gets the records for the selected tables from the server data cache, and sends them one at a time to the custom data retrieval client.

To ensure that all of these records are transferred to the client, Data Export uses an acknowledgment scheme. The basic idea behind the protocol is that as each record is sent to the client, the client will report the Station Name, Table Name, and Record Number back to the server after it has secured that record. The server uses the acknowledgment to mark the progress of the transfer. When the session is broken, or if the Data Export doesn't receive the acknowledgment, the unsent records remain in the LoggerNet server's data cache. The Data Export maintains transfer progress information on disk so that if the server goes down or there is another problem with the transfer, it can recover and continue to transfer all collectable records.

The record acknowledgment allows Data Export to ensure that every record it intended to send was successfully received by the client. This capability, coupled with reasonable algorithms that make sure the LoggerNet server receives every record logged by the datalogger, allows for reliable data collection.

12.4.3 Custom Data Retrieval Client

Because there are so many different types of database applications and data handling processes in use, the data retrieval client must be created either by the customer or on contract with Campbell Scientific to the custom specifications of the user's process.

The custom data retrieval client is a software application that connects to the socket provided by the Data Export application. It can be programmed to run on any computer platform that is configured to support TCP/IP as long as there is a computer network connection available to the host computer where the Data Export application is running.

When a connection is established, the Data Export will send one data record as soon as it is available. The first data record sent depends on the Data Export option settings.

When the data retrieval client receives the record, it must parse the data and return the acknowledgment message to the Data Export. The acknowledgment message consists of the name of the datalogger, the name of the table, and the record number of the record received.

If the acknowledgment message is not returned within 60 seconds or if the message is incorrect, the Data Export will re-send the same record again. It will continue sending the same record at 60-second intervals until either the connection is broken or a valid acknowledgment for that record is received.

The custom data retrieval client is programmed to write to the database or file system defined for the user's data handling process.

12.4.4 Custom Client/Data Export Interface Description

This section details the interface for writing a custom data retrieval client that will get data from the Data Export application. The programming concepts presented assume a familiarity with programming software applications to connect with a TCP/IP socket.

The Data Export application functions as a server providing a TCP/IP socket connection for one remote client. Once the Data Export has connected to the LoggerNet communication server the TCP/IP socket is established and the Data Export application starts “listening” for a client to attach. As soon as a data retrieval client connection is detected the Data Export application sends the first record out over the socket connection. Upon receiving the record the data retrieval client needs to send back an acknowledgment message consisting of the datalogger name, table name and the record number of the received record.

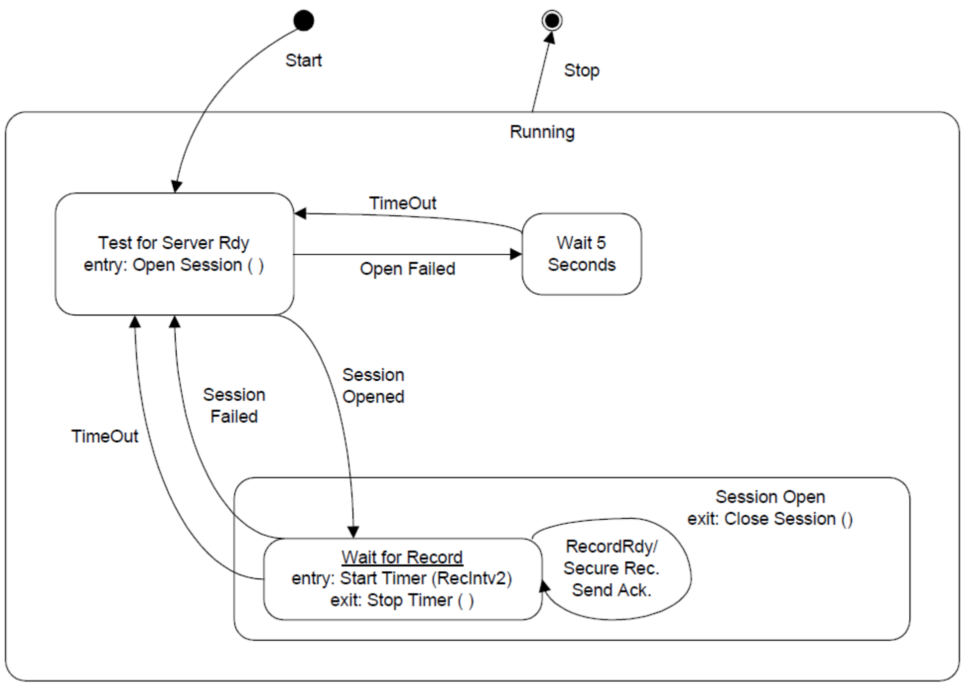
If the Data Export application loses its connection with the LoggerNet communication server, it will need to be re-connected before any records can be obtained and sent out.

There are two record formats used to send the record data:

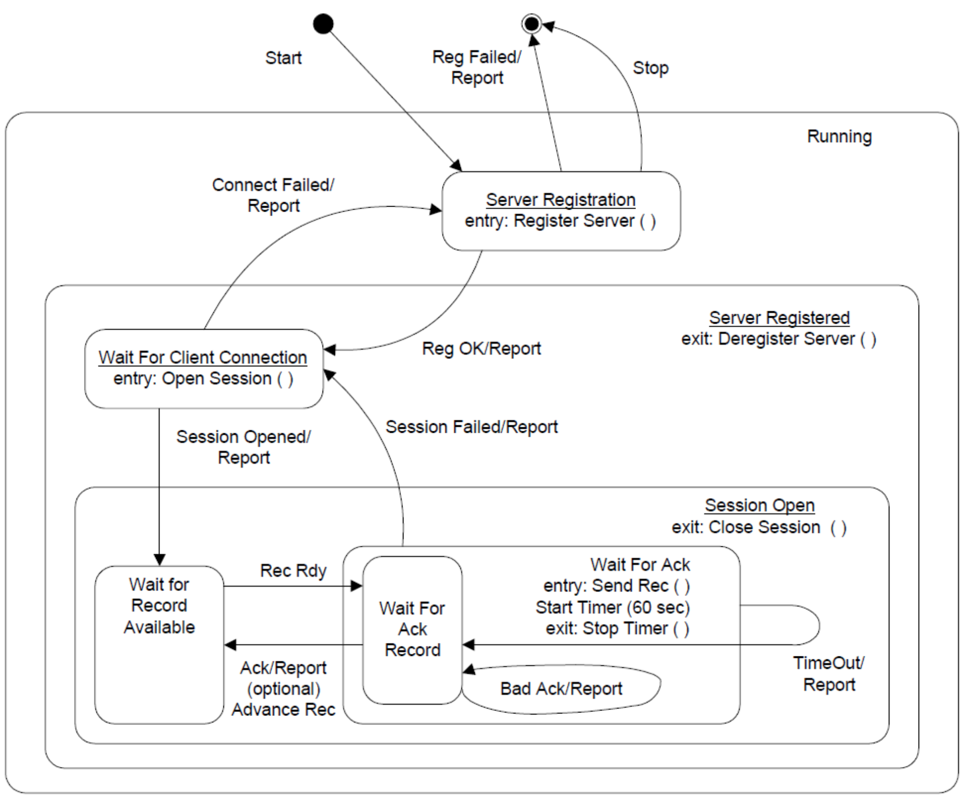
- **RTMS format** – this format is provided for backward compatibility for customers who had the RTMS system and developed data handling procedures that use the format provided by the RTMS socket export.
- **Standard format** – this format provides an easily interpreted data string containing the data along with format information for each data field.

Details on these formats are provided at the end of this section.

The following illustrations show the state diagrams for the custom client/Data Export interface. (The diagramming notation is by Booch who claims to have adopted it from Harel).



Client State Diagram



Data Export Server State Diagram

Key concepts from the state diagrams are shown in the following tables with key words from the diagrams. In these definitions the "server" refers to the Data Export data server and the "client" is the custom data retrieval client application.

Client State Diagram:

Key Word	Description
Test For Server Rdy	With Socket APIs, usually there will be a function used to open the socket. In this state, the client program should attempt to open the socket. If **Open Failed**, the client should wait 5 seconds and try again.
Wait For Record	In this state the client is waiting for the next data record from the server. When a record is received it should be "secured" (saved to disk or database), then an acknowledgment should be sent back to the server. Once the server has processed the acknowledgment it will not send that record again. The client should use a watchdog timer while waiting for a data record. If the client is in the **Wait For Record** state for longer than expected (**RecIntv2**) then it should assume that the server has died and close the session. This watchdog operation may be difficult to implement, but it seems that some implementations of sockets do not properly report a broken socket and so the watchdog is necessary for reliability.
Rec Intv 2	This is an amount of time greater than 2 times the expected interval between data records. It is just longer than the longest period between records the client would expect to receive from the server. If the client goes longer than this interval without receiving a new record then it should close and reopen the socket, thus allowing the server to recover if it has broken socket connection.
Secure Rec	The secure record action is taken when a **Record Rdy** event occurs while the client is in the **Wait For Record** state. Before the client sends an acknowledgment to the server it should "secure" the data record sufficiently so that if a power failure or crash occurs the data will be safe.
Send Ack	The send acknowledgment action is done in response to the **Record Rdy** event, after the record is secured. In **Send Ack** the client forms an acknowledgment record from information taken from the data record and sends it to the server.
Stop	It is important to note that the Stop event could occur at any time. If it occurs while in the Session Open state then the socket should be closed (**Close Session**) before program termination.

Server State Diagram:

Key Word	Description
Wait For Record Available	In this state the server is waiting for the next record to become available from the server's data record source.
Wait For Ack	In this state the server is waiting for the client to acknowledge that it has secured the record. If an acknowledgment for the wrong record comes in, the server will just continue to wait. After waiting for a minute, the server will re-issue the data record and wait again.
Advance Rec	The advance record action is executed after the server receives a valid acknowledgment record from the client while in the **Wait For Ack** Record state. This is the point at which the server recognizes that the client has secured a record and the server relinquishes responsibility for the well being of that record. The server moves on to the next record.
Stop	Note that in the **Session Open** and **Server Registered** states there are "exit" actions that need to be executed on the Stop event.

Communications between the client and server are conducted using ASCII records where each record is terminated by a carriage return – line feed (CRLF) pair. Record length varies quite a bit. For each datalogger record there is exactly one ASCII record. Because of the Block Mode Protocol used to communicate with dataloggers, the maximum size datalogger record is limited to something less than 1024 field values. Assuming 6 characters per value, 13 characters per field name, and 6 characters per field type designation, a single ASCII record could come out to be a little longer than 25K characters.

Typical datalogger programming will produce record sizes of about 150 characters. It would not be unusual to see records that contain one or two hundred values which would come out to a length of 2 to 3K characters in ASCII.

To express the format of ASCII records used for communications between the client and server, we will use Extended Backus Naur – Formalism (EBNF), a notation used to express syntax. This notation was adopted from Wirth [3], and extended here by adding a repetition count preceding some brackets. EBNF is summarized in the following table where A, B and C are syntactic entities of the language being described. Where one of these entities is a literal string it is enclosed in quotes.

Expression Means

A = BC	The construct A consists of B followed by C.
A = B | C	A consists of B or C.

A = [B]	A consists of B or nothing.
A = {B}	A consists of any number of B's including none.
( )	Brackets used to group sections of an expression.

12.4.5 RTMS Format Description

The EBNF description of RTMS syntax is as follows:

Record = (DataRecord | AckRecord) CRLF.
DataRecord = StationName "," TableName " (" FieldSpecs ") VALUES (" FieldValues ")".
AckRecord = StationName "," TableName "," RecordNumber.
FieldSpecs = FieldName " " FieldType {"," FieldName " " FieldType}.
FieldValues = FieldValue {"," FieldValue}.
StationName = Label.
TableName = Label.
FieldName = Label.
Label = Letter { Letter | Digit }.
FieldType = ("TIMESTAMP" | Decimal | "FLOAT" | "INTEGER" | VarChar).
Decimal = "DECIMAL(" Digit [Digit] "," Digit [Digit] ")".
VarChar = "VARCHAR(" Digit { Digit } ")".
FieldValue = (TimeStamp | RecordNumber | Number | String).
TimeStamp = "'" Year "–" Month "–" Day " " Hour ":" Minute ":" Second "'".
Year = 4(Digit).
Month = 2(Digit).
Day = 2(Digit).
Hour = 2(Digit).
Minute = 2(Digit).
Second = 2(Digit) ["." { Digit }].
RecordNumber = 10{ Digit }.
Number = { Digit } ["."] { Digit }.
String = "'" { Character } "'".

A typical data record might look something like this:

Lgr,Sec15 (TMSTAMP TIMESTAMP,RECNBR DECIMAL(10,0),Battery_V FLOAT,Temp FLOAT)
VALUES ('1993-12-08 15:02:00',123456,13.5,72.123)

Only without the tabs and carriage return in the middle. One with strings might look like this.

PC1,StatMsg (TMSTAMP TIMESTAMP,RECNBR DECIMAL(10,0),SrcStn

VARCHAR(256),AbtStn VARCHAR(256),Hop DECIMAL(3,0),Message VARCHAR(256)) VALUES
('1993-12-08 15:02:02.25',13355,'PC1','StatMsg',0,'DBSelect End Pipe Queue Dump')

The acknowledgment records to be sent back to the server for the two records shown above would be:

Lgr,Sec15,123456

and

PC1,StatMsg,13355

12.4.6 Standard Format Description

The following is an EBNF syntax of a new record format that we have developed that we believe is more digestible than the pseudo-SQL syntax that is in the original protocol:

outputRec = recordHeader { "," fieldName "," fieldType "," fieldValue } "\r\n".
recordHeader = stationName "," tableName "," timeStamp "," recNo.
FieldName = string.
FieldType = ("TIMESTAMP" | decimalType | "FLOAT" | "INTEGER" | varCharType).
FieldValue = string.
StationName = string.
TableName = string.
TimeStamp = "\"" year "–" month "–" day " " hour ":" minute ":" second "\"".
RecNo = "\"" digit {digit} "\"".
Year = 4(digit).
Month = 2(digit). ; 0 < month <= 12
day = 2(digit). ; 0 < day <= 31
hour = 2(digit). ; 0 <= hour < 60
minute = 2(digit). ; 0 <= minute < 60
second = 2(digit) ["."] {digit}. ; 0.0 <= second < 60.0
string = "\"" {ascii_character} "\"".
DecimalType = "DECIMAL(" digit [digit] "," digit [digit] ")".
VarCharType = "VARCHAR(" digit {digit} ")".

Within a string, quotation marks and back slash characters will be quoted with a backslash character.

The sample record from the original protocol would have the following format under this new syntax:

"Lgr","Sec15","1993-12-08 15:02:00","123456","Battery_V","FLOAT", "13.5","Temp","FLOAT","72.123" CRLF

The acknowledgment message is the same as for the RTMS format. The acknowledgment for the above record would be:

Lgr,Sec15,123456

13. Optional Client Applications Available for LoggerNet

Several client applications are available that are compatible with LoggerNet. Many of these allow remote access to the data in the LoggerNet data cache, or provide a way to post process that data.

Client applications include RTMC-RT, RTMC Pro, and LNDB.

For the client applications that allow remote access to the LoggerNet data cache, LoggerNet must be configured to allow connection from remote clients.

13.1 Allowing Remote Connections to the LoggerNet Server

LoggerNet is a 32-bit client/server application, and therefore, the server can run on one computer while a client application can be run on a separate computer attached to the same network. Campbell Scientific offers client applications for LoggerNet that take advantage of this remote access capability. If Allow Remote Connections is enabled, you can run LoggerNet on one computer, and use a remote client application to display data remotely on a different computer or save a copy of the data on the remote computer. If remote connections are denied, data access from a remote computer is not possible.

Remote Connections is enabled from the LoggerNet Toolbar's **Tools** > **Options** menu item. Select the **Allow Remote Connections** check box to allow remote connections. Conversely, when the check box is not selected, Remote Connections will be denied. After the change is made, LoggerNet must be restarted for the change to take affect.

Though this may be a desirable feature, enabling Allow Remote Connections also makes your LoggerNet network configuration vulnerable to changes by other parties on the network.

Therefore, if using standard LoggerNet (not LoggerNet Admin), we strongly recommend that you select **Remote Connection Security Enabled** and input (and confirm) a password. (The username cannot be changed.) The username and password will then be required before other parties on the network can connect to the LoggerNet server.

If using LoggerNet Admin, we strongly recommend that the Security Manager be used to set up account access with password protection to limit the ability to make changes to the datalogger network.

13.2 RTMC Run-Time

RTMC Run-Time is an application that allows you to remotely run real-time graphic display screens that have been created in the RTMC Development version. RTMC Run-Time is discussed in Real-Time Tools (p. 80).

13.3 RTMC Pro

RTMC Pro is an enhanced version of the RTMC Development application that ships with LoggerNet and is talked about in RTMC Real Time Monitoring and Control Software (p. 119) RTMC Pro contains more graphical components than RTMC. For example, more alarms (multi-state), alarm events (email, FTP, run/open), switches (lever, rocker, rotary), charts (XY and scope), gauges (rotary, compass), and layout components (group box, bevel, panel) are available. For components that exist in both versions, more properties have been exposed in RTMC Pro resulting in more design control. RTMC Pro also includes run/open button, hotspot, snapshot, and alarm log capabilities.

RTMC Pro is not covered in this manual. RTMC Pro comes with a separate user's manual. Product literature can be downloaded from our website at www.campbellsci.com.

13.4 LNDB

LNDB is an application that enables you to easily move data from a LoggerNet data cache into a database such as Microsoft SQL Server or MySQL. The two main components of LNDB are LNDB Manager and LNDB Engine. LNDB Manager is used to set up a database and select the datalogger data tables that will be stored in the database. It also provides tools to monitor the LNDB Engine and to review the database data. LNDB Engine runs as a service and sends the selected data from the LoggerNet data cache to the database. Additionally, LNDB includes utilities for importing and exporting data, and generating simple reports from your database data.

LNDB is not covered in this manual. LNDB comes with a separate user's manual. Product literature can be downloaded from our website at www.campbellsci.com.

14. Troubleshooting Guide

This section is provided as an aid to solving some of the common problems that might be encountered using the LoggerNet software. This list is not comprehensive but should provide some insight and ability to correct simple errors without a call to Campbell Scientific technical support.

This section also includes descriptions of some of the tools such as Terminal Emulator and Data Table Monitor that can be useful in troubleshooting LoggerNet problems.

14.1 What's Changed

When things stop working the most important thing to ask yourself is: "What's changed?" A new computer, new software (especially non-Campbell Scientific software), different communication peripheral such as a new modem, new datalogger program, etc., can all interrupt communications. If you can't deduce the nature of the problem, go back to the original configuration. If communication with the datalogger starts to fail for no apparent reason, ask someone else to try it from a different computer. Try swapping out components – one at a time – from a link in the network that you know works or from spare equipment that you are certain is functioning correctly. Sometimes the smallest thing – a cable or a new PC utility program – can cause widespread havoc. If using TCP/IP or cellular telephone communications, check with the network administrator to see if anything changed that coincided with the loss of communications.

14.2 LoggerNet Server Problems

The following sections identify problems that have been observed with operation of the server. If you are experiencing problems with the server look through the following conditions to see if any of these match the problem you are having. If you find the problem listed, try the suggested remedies. If your problem is not listed or the remedies don't fix the problem, contact Campbell Scientific for technical assistance.

14.2.1 Unable to start LoggerNet

Problem: Unable to start LoggerNet because the Campbell Scientific communications server is already running.

- **Remedy**: Most Campbell Scientific software (for example, PC400, LoggerNet, RTDAQ) is built on common code for the underlying software server that communicates with data loggers. This message indicates another instance of the communications server is running. Close all other Campbell Scientific software that may be open on the computer before relaunching LoggerNet or LoggerNet Admin

14.2.2 Socket Errors

The LoggerNet Server uses TCP/IP sockets for communications. Various problems can occur with these socket connections. Some of the most common errors and remedies are listed below.

Maximum Number of Sockets Open

The Windows operating system has limits on the number of socket connections that can be held open. For most operations this should be more than enough to cover the open applications that use sockets. One situation that does cause problems is using the IPPorts to communicate with dataloggers where the socket is being opened and closed quickly. For example if you have 20 stations on IPPorts and you do normal data collection every 5 seconds, 20 new sockets are created every 5 seconds. The normal lifetime of the created socket is about 4 minutes leaving about 1000 active sockets at a time. If there are other applications that use sockets, it is possible to exceed the allowed number of sockets.

To work around this problem, either slow down the rate of data collection, or use the **Delay Hangup** setting for the IP Port (accessed from the Setup Screen) to keep the stations online.

Socket Error Messages

When you get an error message that says Socket Error and a number, check the chart below for the type of error that occurred and what to do about it. Note that these error messages can show up either in pop up error boxes or as part of the LoggerNet Communications log.

Socket Error Messages		
Socket Error Number	**Message Meaning**	**User Response to Message**
10013	*Permission Denied*. The requested socket connection has refused the connection.	This is normally a network type of issue. Check with your computer network operator.
10024	*Too many open files*. Too many open sockets for the applications running.	This can occur when you have many applications that are using sockets running at the same time.
10047	*Address family not supported by protocol family*. The socket being addressed does not support the type of connection being attempted.	This message shows up when the LoggerNet Toolbar comes up but the server did not come up because TCP/IP is not installed on the computer. Install TCP/IP and restart LoggerNet..
10055	*No buffer space available*. Cannot create more temporary sockets.	The operating system cannot create any more socket connection. See the text above about Maximum Number of Sockets Open.
10058	*Cannot send after socket shutdown*. A message was sent to a socket that has been closed.	This would be an indication that an application is not communicating well with the server. Check the application.
10060	*Connection timed out*.	Either the server has crashed and is not responding or the application did not maintain the connection to the server. Try restarting LoggerNet. This message can also be seen in connection with the NL201 Network Link Interface.

Socket Error Messages		
Socket Error Number	**Message Meaning**	**User Response to Message**
10061	*Connection refused*. The LoggerNet server or an NL201 refused to allow the socket connection.	This is normally associated with the NL201 and occurs because the last connection did not have enough time to close before a new connection is requested. Slow down the low level polling delay interval.
10065	*No route to host*. The application is trying to connect to a host address that isn't in the routing table.	This occurs with remote connections to a LoggerNet server running on another computer. The requested host name can't be found.

14.2.3 Data Collection Issues

Problem: Scheduled data collection is enabled but no data is being saved in the data files, or data is not updating on numeric or graphical display.

- **Remedy 1**: Make sure that communications are enabled for the datalogger and all the devices in the communications link.
- **Remedy 2**: For table-based dataloggers, make sure tables are included for collection and the table definitions are current. For array-based dataloggers, make sure the correct final storage areas are included for collection.
- **Remedy 3**: Check communication state on Status Monitor. An indication of Primary or Secondary retry indicates that LoggerNet is waiting for the next collect time. If the station is waiting in Secondary Retry mode, click **Collect Now** or **Reset Device**. This will return the datalogger to the normal collection state.

Problem: Invalid Table Defs indicates that LoggerNet does not have a current copy of the datalogger table definitions.

- **Remedy**: You will need to update the table definitions from the Setup Screen or the Connect Screen.

Problem: Network Paused indicates that data collection for the entire network has been suspended.

- **Remedy**: You will need to go to the Status Monitor and remove the check mark from the Pause Schedule check box.

14.3 General Communication Link Problems

Problem: Communications are not solid and difficulty is experienced sending programs to the datalogger.

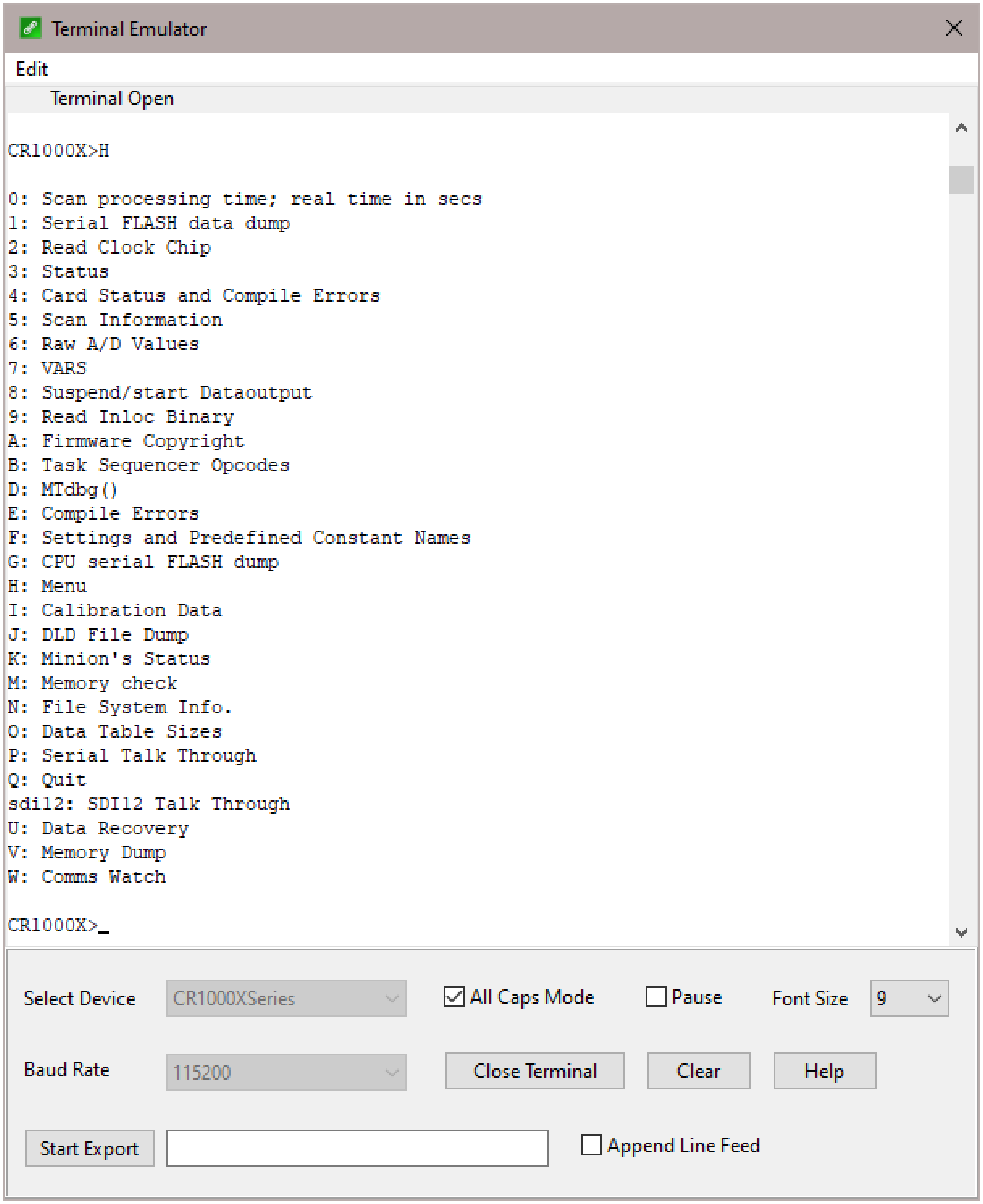
- **Remedy 1**: If there are slow serial devices in the communication path, such as older modems, the server might be overrunning the buffers. Set the maximum packet size in the Setup Screen for the datalogger to a smaller number.
- **Remedy 2**: On noisy phone links try lowering the max baud rate for the datalogger on the Setup Screen.
- **Remedy 3**: On RF networks, make sure that the extra response time is sufficient for the reply to come back, especially if there are repeaters in the network. Use the minimum value necessary to make the link function; usually less than 5 seconds.

14.4 Terminal Emulator to Test Communications

Terminal Emulator is a utility to test communications with the devices in the datalogger network. Terminal Emulator is accessed from the Datalogger menu of the Connect Screen. The operation of a ComPort and the connection to a phone modem can be tested.

The Terminal Emulator utility is available from the Connect Screen to help troubleshoot communications problems. When you choose a device with the Select Device field, the Terminal Emulator will attempt to establish communications with that device. The Terminal Emulator will use the lowest baud rate among all of the devices involved in the link. For example if choosing a COM port, the baud rate will typically be 115,200 baud and LoggerNet simply opens the port. You can also use terminal emulation to send commands to the dataloggers.

When the Terminal Emulator screen comes up as shown click the drop down arrow to the right of the Select Device box to choose the device from the list of devices in the network map. The correct baud rate for the link is automatically set. The characters you type in the window are sent as ASCII text to the selected device. The options that are available from this screen depend on the device you select.



Dataloggers

The example above shows a terminal emulation session with a datalogger. Once you have selected the datalogger, click **Open Terminal** to start communications. Array-based dataloggers require you to type in the letters 7H (2178H for CR7 and 21X) and press Enter to establish terminal emulation mode. Table-based dataloggers are ready for terminal emulation when they are first selected. Just press **Enter** a few times. Type H when the prompt comes up and a list of options will be displayed.

> **NOTE:** Use caution while in terminal emulation mode. You can change or disable operation of the datalogger with these commands.

ComPort

You can use the Terminal Emulator to perform a communications test on a ComPort. To perform a feedback test, select the ComPort and click **Open Terminal**. Then connect the Transmit and Receive lines (2 and 3) of the serial port cable using a small screw driver or paper clip. Click in the window to get the cursor and type some characters on the computer keyboard. If the characters are echoed back to the screen, the ComPort is working.

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

See Also

Terminal Emulator (p. 115) for more information on troubleshooting tools available from the Terminal Emulator.

14.5 Using Data Table Monitor

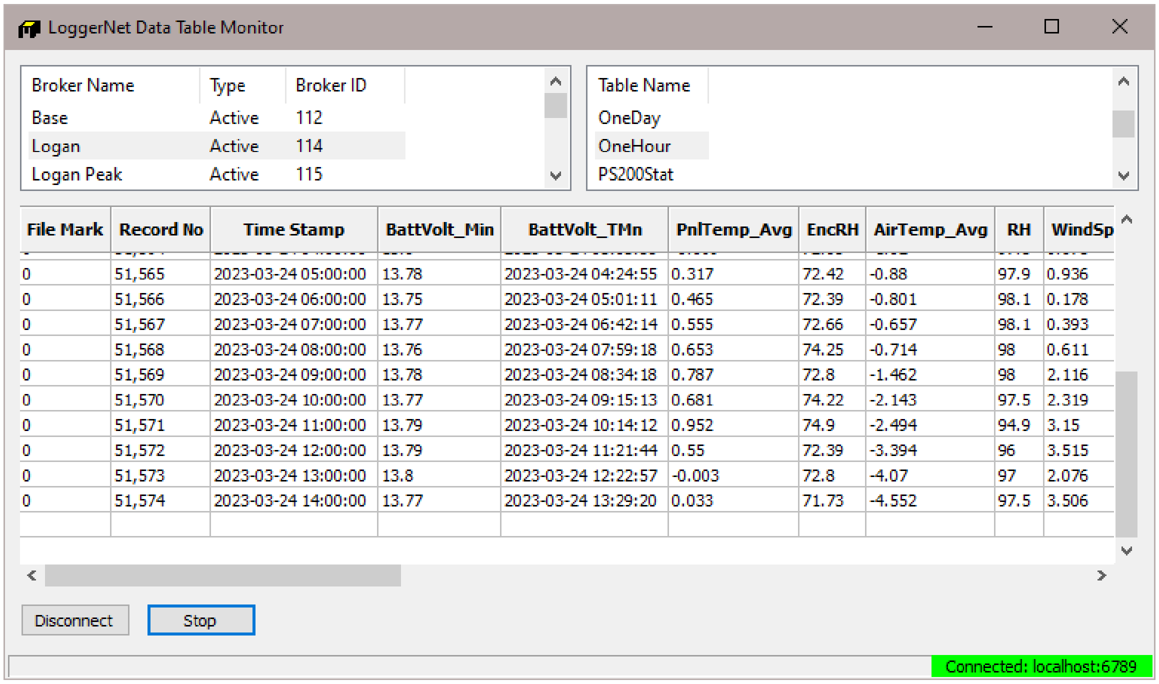
Data Table Monitor is a utility that was created to retrieve data from the LoggerNet server data cache and display it on the screen. It also has the option to export it to a file. Once the utility has been started, as new records are collected by the server, the new records will be displayed and sent to the file.

The most important use of Data Table Monitor is to see what records are being stored in the data cache and to diagnose suspected data cache problems.

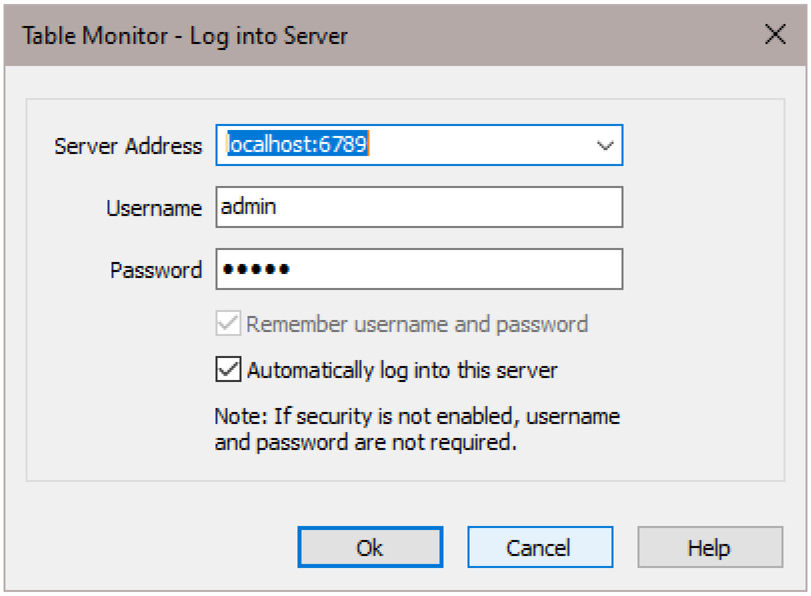
Data Table Monitor gets all the data available from the data cache that matches the export conditions. As the server collects new records from the datalogger, they are automatically displayed and sent to the data file. This continues until Data Table Monitor is closed or data export is stopped.

> **WARNING:** One caution about the data file created by Data Table Monitor—there are no limits to size or longevity. If you plan to use the export to file feature on a regular basis, make sure to either restart Data Table Monitor (which overwrites the exported file) or delete the files periodically. The data export can easily be restarted by clicking the Start button. This will delete the old file and start a new one.

To start Data Table Monitor open Windows Explorer and got to the C:\Program Files (x86)\Campbellsci\LoggerNet directory. Double click the Tablemon2.exe file. The utility will start with a screen similar to the one shown below.

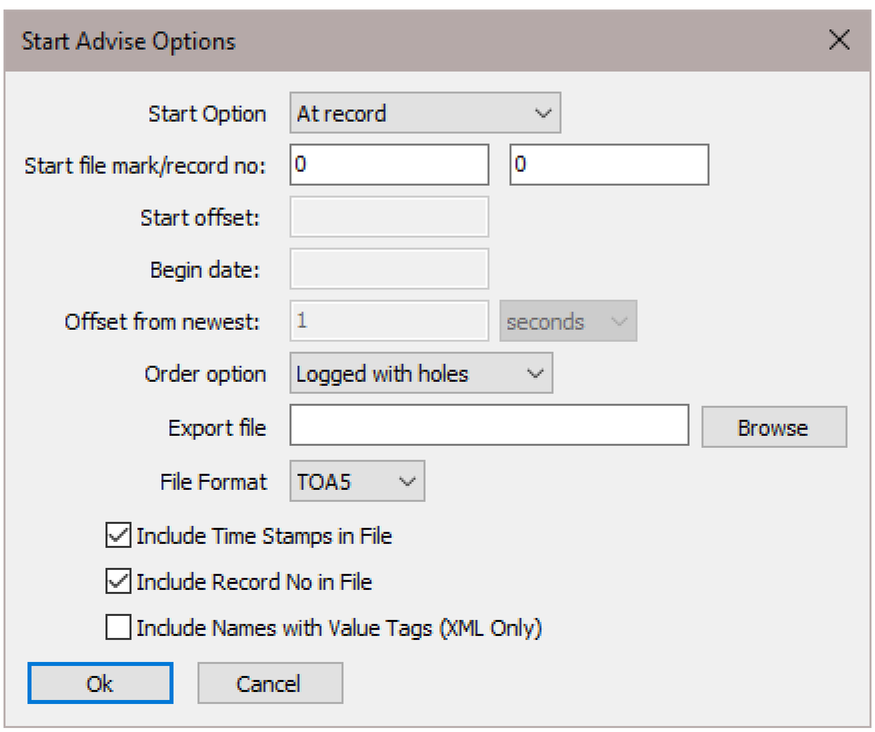


If Data Table Monitor does not automatically connect to the LoggerNet server, click the **Connect** button to connect to the server. The dialog box shown below will be displayed. If you are working on the same computer where LoggerNet is running leave the default **Server Address** as localhost. The Server Port (default is 6789) should be appended to the **Server Address** separated by a colon. The **Username** and **Password** are only used with versions of LoggerNet that support security. To connect to LoggerNet on another computer, enter the computer network name (or IP address) and port number in the **Server Address**. When you click **OK** a list of the dataloggers in the network will be shown in the upper left window.



Selecting a datalogger will list the names of the data tables or array IDs in the datalogger. Note that if data collection has not been set up and enabled in the Setup Screen, no data will be coming into the data cache. Data Table Monitor can only display and output data from the data cache. Data Table Monitor displays and outputs all the data points from an array or table.

Click the **Start** button to bring up the **Start Advise Options** dialog. This dialog gives you choices about which records to display and the data file in which to store them.



Start Option: This selects the starting point for the data to be displayed and output to the file.

- **At Record**: This option allows a selection of starting position based on the file mark and record number. An entry of 0 in both fields will get all of the data in the data cache.
- **At Time**: This option allows a selection of the starting position based on the timestamp in the data. The time and date are set in the **Begin Date** field. All of the records available after this timestamp are output.
- **At Newest**: This option will set the starting position to the last record stored in the data cache. This last record and any future records stored will be output.
- **After Newest**: This option will set the starting position to be the next record stored in the data cache. Output begins with the next record stored in the data cache. No historical records will be output.
- **Relative to Newest**: This option starts from the most recent record collected. The **Offset from Newest** specifies how much time to go back from the current write index. For example, an offset of 10 with a setting of minutes will get the last 10 minutes of data collected.
- **At Offset from Newest**: This option allows you to specify how many records back from the current write index to go. A setting of 10 in the **Start Offset** box will display the last 10 records collected.
- The **Start File Mark**, **Start Record Number**, **Start Offset**, **Begin Date**, and **Offset from Newest** edit boxes are used only with the corresponding start options above. For each option selected, the appropriate boxes are enabled.

Order Option:

- **Collected**: displays and writes the data to the file in the order it was collected by the server. This setting is useful to look at the actual data record storage in the data cache.
- **Logged With Holes**: The output will include only complete data sequences. If the Data Table Monitor comes to a hole that has not yet been filled, it will wait for the hole to fill before displaying or writing the next record to the file.
- **Logged Without Holes**: The data output will be displayed and written to file as quickly as it is collected, without waiting for holes to be filled. Any data in holes will be skipped in the output.
- **Real Time**: the most recent data is always sent out starting with the last record stored. This will not provide a complete data set.

Set the output file directory and name in the **Export File** box. The **Browse** button will bring up a Windows Save As dialog box to select the file name and directory.

File Format:

- **TOACI1** – Data is stored in a comma separated format. Header information for each of the columns is included.
- **TOA5** – Data is stored in a comma separated format. Header information for each of the columns is included, along with field names and units of measure if they are available.
- **TOB1** (binary) – 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.
- **XML** – Data is stored in XML format with Campbell Scientific defined elements and attributes. For additional information, refer to Appendix B, Campbell Scientific File Formats (p. B-1).

Once the start options have been set, click the **OK** button to start. The records are displayed in the list box on the bottom of the screen. If you have set up an output file they are also sent to the output file.

14.6 Troubleshooting PakBus Communications

Problem: LoggerNet can't communicate with in-range PakBus datalogger

(PC-RF401A-----RF401A-CR1000X-----RF401A-CR1000X)

- **Possible reason 1**: LoggerNet's **PakBus Address** for datalogger doesn't match datalogger's **PakBus Address**.

  Remedy 1: Make them match.

- **Possible reason 2**: An RF401A-series radio is set to a different **RF Hop Sequence**, **RF Network**, or **Power Mode**.

  Remedy 2: Set all radios exactly the same in the above parameters.

Problem: LoggerNet can't communicate via datalogger-router to a certain remote datalogger.

(PC-RF407-----RF407-CR1000X-----RF407-CR1000X-----CR300-R407

- **Possible reason 1**: A datalogger router has insufficient **Communication Allocation** configured.

  Remedy 1: Increase the **Communication Allocation**

- **Possible reason 2**: Last datalogger router has no means configured of discovering the remote datalogger.

  Remedy 2: Configure the datalogger router with either **Allowed Neighbors** or a **Beacon Interval**. Make sure **Allowed Neighbors** include the remote datalogger's address. Whether you set up a **Beacon Interval** or **Allowed Neighbors** make sure the port so configured matches the communications device port configuration. For example, if the selected **Allowed Neighbor** port is CSDC7, make sure that the RF407 **Active Interface** is "CSDC7."

- **Possible reason 3**: An RF407 radio is set to a different **RF Hop Sequence**, **Network ID**, or **Power Mode**.

  Remedy 3: Set all network radios exactly the same in the above parameters. Dataloggers automatically detect the RF407's port (**Active Interface**) for packet communications, however, the **Allowed Neighbor** port or **Beacon Interval** port must be configured to match the RF407's **Active Interface**, or no discovery of neighbors will take place.

- **Possible reason 4**: The two routers in the path to the CR300 have **Allowed Neighbors** and at least one of the **Allowed Neighbors** doesn't list the other as a potential neighbor.

  Remedy 4: If you have two in-range routers using **Allowed Neighbors**, in order for them to discover one another you must list each of them as a potential neighbor in the other's **Allowed Neighbors**.

Problem: Rapid spurious communications lasting a few seconds at a time between devices in RF4 network.

- **Possible reason**: Two network devices have the same **PakBus Address**.

  Remedy: Change one of the duplicate PakBus addresses. Make all addresses unique throughout the network.

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 ACSII 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. Campbell Scientific File Formats

Campbell Scientific, Inc. uses different formats for data in datalogger memory, external memory cards, datalogger communication software, and computer files. The data formats written to computer files by LoggerNet are written by default as .DAT files. The following sections will focus on the format of these computer files, discuss the data formats that exist in the datalogger and on memory cards, and describe methods for converting binary data formats.

B.1 Computer File Data Formats

The type of data file generated by LoggerNet depends on the type of datalogger from which data are being collected. Mixed-array dataloggers such as the CR10X include the option to output comma separated files, ASCII printable files, and binary files. Data from table-data dataloggers are output as ASCII table files with no header, TOACI1 files, TOA5 files, TOB1 binary files, Array Compatible CSV files, or CSIXML files. Data from memory cards (generated using the CRBasic CardOut instruction) are output as TOB2 or TOB3 binary format.

B.1.1 TOA5

c with the following features:

- Contains a text header that provides the following information:
- The file format type, the station name, the datalogger type, the serial number, the OS version, the DLD name, the DLD signature, and the table name. Note that, by default, the station name is the name given to the datalogger in the network map. If the **Use Reported Station Name** check box is selected, the station name from the Status Table will be used.
- The field name for each of the data values.See Field Name Suffixes (p. 411)for field name suffixes.
- The units for each field as determined by the datalogger program.
- The processing performed in the datalogger to produce each value in the table.
- A timestamp and record number can optionally be included as part of the record data. If the timestamp is present, it will be formatted with sub-second resolution.

- Data values are formatted as comma separated text suitable for importing into spreadsheet or database applications.
- Each TOA5 file contains data from only one table.

An example showing a TOA5 file containing the optional timestamp and record number:

```
"TOA5","CR1000","CR1000","1031","CR1000.Std.00.60","CPU:Test.CR1","4062","Test"
"TIMESTAMP","RECORD","batt_volt_Min","PTemp"
"TS","RN","Volts","C"\
"","","Min","Smp"
"2004-11-11 15:03:45",0,13.7,24.92
"2004-11-11 15:04:00",1,13.7,24.95
"2004-11-11 15:04:15",2,13.7,24.98
```

B.1.1.1 Field Name Suffixes

Each field name will have a suffix corresponding to the output instruction used as described in the table below:

Output Instruction Suffixes		
Instruction	**Description**	**Suffix**
Totalize	Totalize	Tot
Average	Average	Avg
Maximum	Maximum Time of Maximum	Max TMx
Minimum	Minimum Time of Minimum	Min TMn
SampleMaxMin	Sample at Maximum or Minimum	SMM
StdDev	Standard Deviation	Std
Moment	Moment	MMT
Sample	Sample	No Suffix
Histogram	Histogram	Hst
Histogram4D	4 Dimensional Histogram	H4D

Output Instruction Suffixes		
Instruction	Description	Suffix
FFT	FFT	FFT
Covariance	Covariance	Cov
RainFlow	RainFlow Histogram	RFH
Level Crossing	Level Crossing	LCr
WindVector	Wind Vector Average	WVc
Median	Median	Med
ET	Evapotranspiration Solar Radiation	ETsz RSo

NOTE: Not all dataloggers have all output types.

B.1.2 TOACI1

TOACI1 is a text-based file format similar to TOA5 but with less information in the header. This format has the following features:

- The file includes a header that contains the following:
  - The file format type, the station name, and the table name. Note that, by default, the station name is the name given to the datalogger in the network map. If the **Use Reported Station Name** check box is selected, the station name from the Status Table will be used.
  - The field name for each of the data values. See Field Name Suffixes (p. 411)for field name suffixes.
- Each record in the file is assigned a timestamp and record number. The record number is a logged sequence number that is assigned by the datalogger.
- The data values are formatted as comma-separated text suitable for importing with little modification into most spreadsheet and database applications.
- Each TOACI1 file contains data from only one table.

An example of a TOACI1 file with a header and data values:

```
"TOACI1","gold","one_min"
"TMSTAMP","RECNBR","temp_degf_AVG","meas1","meas2"
```

"2001-12-30 19:16:00",18002,69.05,3000,1500
"2001-12-30 19:17:00",18003,69.06,3001,1499
"2001-12-30 19:18:00",18004,69.06,3002,1498

B.1.3 TOB1

TOB1 files can be generated by LoggerNet when outputting data files to the PC. This binary file format is typically only used when it is essential to minimize the file size or when other software requires this format. It has the following structure:

ASCII header line 1
ASCII header line2
ASCII header line3
ASCII header line 4
ASCII header line5
Binary Records

An example of a TOB1 ASCII header:

"TOB1","STATION","CR9000","1000","1.00","CPU:BIG.DLD","25871","VALUES"
"SECONDS","NANOSECONDS","RECORD","Array(1)","Array(2)","Fast","my_string"
"","","RN","mVolts","mVolts","mVolts"
"","","","Smp","Smp","Smp"
"ULONG","ULONG","ULONG","IEEE4","IEEE4","FP2","ASCII(25)"

Header line one describes the file environment with the following eight fields:

- Data file type (TOB1).
- Station name (STATION). (Note that, by default, the station name is the name given to the datalogger in the network map. If the **Use Reported Station Name** check box is selected, the station name from the Status Table will be used.)
- Model name of the datalogger (CR9000).
- Serial number of the datalogger (1000).
- Operating system on the datalogger (1.00).
- Name of the program running in the datalogger (CPU:BIB.DLD).
- Signature of the program running in the datalogger (25871).
- The name of the datalogger table (VALUES).

Header line two consists of a set of comma-delimited strings identifying the names of the fields in the table of the datalogger program.

Header line three describes the units associated with each field in the record. Units are optional and are specified in the datalogger program, if included. If no units are provided in the program, then an empty string placeholder is left in this line for that specific field.

Header line four describes the processing performed in the datalogger to produce the value for each field in the record; for example, sample, average, min, max, etc. If there is no known processing for a field, that field will be assigned an empty processing string. There will be one value on this line for each field name given on header line two.

Header line five describes the data type for each field and supports the following values: IEEE4, IEEE8, FP2, ULONG, LONG, SecNano, BOOL, and ASCII(len).

> **NOTE:** BOOL is a single-byte Boolean value that represents true as 0xFF and false as 0x00. The four-byte CRBasic BOOLEAN data type will be converted to the one-byte BOOL.

Each data record following the header is a sequence of binary values. The length of each value is determined by the data type assigned to it in header line five and the length of the entire record is the sum of the individual data value lengths. There are no characters that separate records so the application that reads the TOB1 file must understand the file header so that the record length can be calculated.

The timestamp and record number for each record are an optional output in a TOB1 file. If these elements are present, a “SECONDS”, “NANOSECONDS”, and “RECORD” column will be generated as names in the field list of header line two.

B.1.4 Array Compatible CSV

This file format can be used to product output files from table data dataloggers that are similar to those created by mixed array dataloggers. The file format has the following features:

- Commas separate all values.
- The user determines whether to include an Array ID. If an Array ID is to be included, it is specified by the user.
- The user specifies the format of the timestamp.

An example of an Array Compatible CSV file:

101,2009,105,1051,27,13.39,24.04,23.99
101,2009,105,1052,28,13.39,24.04,23.98
101,2009,105,1053,29,13.39,24.04,23.98
101,2009,105,1054,30,13.39,24.04,24

101,2009,105,1055,31,13.39,24.04,23.98
101,2009,105,1056,32,13.39,24.04,23.98

B.1.5 CSIXML

CSIXML is an XML (eXtensible Markup Language) based file format designed to provide the following features:

- Contains data for a single table.
- Data records can be appended without having to reformat the entire file.
- The file meta-data can be verified for appending data without having to read the entire file.
- Lends itself readily to XSL transformations to produce various other CSI text based formats as well as customer specific formats.
- Simple to encode and to decode table records.
- Can handle both interval driven and event driven data without significant structural complexity.
- XSD (XML Schema) files can be generated readily for a specific table file using XSL transforms.

B.1.5.1 A Short Introduction to XML

First and foremost, CSIXML is a well-formed XML document. This means that all CSIXML files will conform to the syntax rules for XML. Well formed XML documents possess a tree structure that consists of elements and element attributes. The document is expected to have a single root element which can contain any number of sub-elements which in turn can contain any number of sub-elements and/or other content. Every element must have a name and can optionally have a set of attributes which are a collection of name/value pairs where the name is unique.

Most XML files will begin with a sequence that identifies the file as XML and can also specify the character encoding of the file (if no character encoding is specified, the file is assumed to use the UTF-8 unicode character encoding). The following example shows this sequence as it will appear in CSIXML data files:

<?xml version="1.0" standalone="yes"?>

XML is derived from SGML (Standard Generalized Markup Language) and shares much of the same syntax rules as SGML. HTML (HyperText Markup Language) is also derived from SGML and, as a result, also has a significant resemblance to XML. XML elements are represented using tags. A tag begins with the less than character (<) followed by the name of the element. If that element has attributes, these will be expected to follow the element name with a name="value" syntax. At least one white space character is expected to separate the attributes and the element name. Other than this rule, XML parsers ignore the presence of whitespace within the tag. If an element is empty (contains no child elements), the tag can end with a slash character (/) and the greater than character (>). If the element is not empty (the element does have child elements), the tag is expected to end with a greater than character (>) and the child elements or element content will be expected to follow. In this case, the end of the element is marked by another less-than character (<) followed by a slash character (/), the element name, and a greater than character (>).

The following example shows how an element with content may appear:

<v n=“pi”>3.14159</v>

The following example shows how an empty element may appear:

<v n=“emptyString”/>

Because XML reserves special characters for its mark-up language, pre-defined entities are recognized by all XML parsers. These entities include the following:

Pre-Defined XML Entities	
<	less than sign (<)
>	greater than sign (>)
&	ampersand (&)
"	a double quote (“)
'	apostrophe or single quote (‘)

In addition to these pre-defined entities, arbitrary unicode characters can be represented by using the sequence &xxx; where xxx is the decimal unicode code value for the desired character.

For more details regarding XML documents and their contents, you can visit the W3C consortium web page at www.w3.org/XML/. In addition, they offer an excellent tutorial at www.w3schools.com/xml/default.asp.

B.1.5.2 File Syntax

Our formal description of the file format will not be the character-by-character description generally given in EBNF formats but will instead describe the general XML Schema (see www.w3.org/TR/xmlschema-0/ for details).

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="csixml" type="csixmlType"/>

  <xsd:complexType name="csixmlType">
```

```
    <xsd:sequence>
      <xsd:element
          name="head"
          type="headType"
          minOccurs="1"
          maxOccurs="1"/>
      <xsd:element
          name="data"
          type="dataType"
          minOccurs="1"
          maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="version" fixed="1.0"/>
</xsd:complexType>

<xsd:complexType name="headType">
  <xsd:sequence>
    <xsd:element
        name="environment"
        type="environmentType"
        minOccurs="1"
        maxOccurs="1"/>
    <xsd:element
        name="fields"
        type="fieldsType"
        minOccurs="1"
        maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="environmentType">
  <xsd:sequence>
    <xsd:element
        name="station-name"
        type="xsd:string"
        minOccurs="1"
        maxOccurs="1"/>
    <xsd:element
        name="table-name"
        type="xsd:string"
        minOccurs="1"
        maxOccurs="1"/>
    <xsd:element
        name="model"
        type="xsd:string"
        minOccurs="0"
        maxOccurs="1"/>
    <xsd:element
        name="serial-no"
        type="xsd:unsignedInt"
```

```
      minOccurs="0"
      maxOccurs="1"/>
    <xsd:element
      name="os-version"
      type="xsd:string"
      minOccurs="0"
      maxOccurs="1"/>
    <xsd:element
      name="dld-name"
      type="xsd:string"
      minOccurs="0"/>
    <xsd:element
      name="dld-sig"
      type="xsd:unsignedShort"
      minOccurs="0"
      maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="fieldsType">
  <xsd:element
     name="field"
     type="fieldType"
     minOccurs="1"
     maxOccurs="unbounded"/>
</xsd:complexType>

<xsd:simpleType name="fieldDataType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="xsd:string"/>
    <xsd:enumeration value="xsd:long"/>
    <xsd:enumeration value="xsd:unsignedLong"/>
    <xsd:enumeration value="xsd:int"/>
    <xsd:enumeration value="xsd:unsignedInt"/>
    <xsd:enumeration value="xsd:short"/>
    <xsd:enumeration value="xsd:unsignedShort"/>
    <xsd:enumeration value="xsd:byte"/>
    <xsd:enumeration value="xsd:unsignedByte"/>
    <xsd:enumeration value="xsd:float"/>
    <xsd:enumeration value="xsd:double"/>
    <xsd:enumeration value="xsd:boolean"/>
    <xsd:enumeration value="xsd:dateTime"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="fieldType">
  <xsd:attribute
     name="name"
     use="required"
     type="xsd:string"/>
  <xsd:attribute
```

```
      name="type"
      use="required"
      type="fieldDataType"/>
   <xsd:attribute
      name="units"
      use="optional"
      type="xsd:string"/>
   <xsd:attribute
      name="process"
      use="optional"
      type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="dataType">
   <xsd:element
      name="r"
      type="recordType"
      minOccurs="0"
      maxOccurs="unbounded"/>
</xsd:complexType>

   <xsd:complexType name="recordType">
     <xsd:attribute
        name="no"
        type="xsd:unsignedInt"
        use="optional"/>
     <xsd:attribute
        name="time"
        type="xsd:dateTime"
        use="optional"/>

     <xsd:element
        minOccurs="1"
        maxOccurs="unbounded"
        type="valueType">
       <xsd:annotation>
        <xsd:documentation xml:lang="en">
          In order to make value elements easily addressable in
          transforms as well as describable in table specific XML
          Schema documents, value element names will begin and end
          with a unique number so that value elements will be named
          using the following sequence: { v1, v2, v3, ... vn }.
        </xsd:documentation>
       </xsd:annotation>
     </xsd:element>
   </xsd:complexType>

   <xsd:complexType name="valueType">
     <xsd:attribute name="n" type="xsd:string" use="optional"/>
     <xsd:simpleContent type="anyType"/>
   </xsd:complexType>
```

```
</xsd:schema>
```

B.1.5.3 The csixml Element

This element is the root element of the csixml format. It defines one attribute, version, that may help in the future if changes are made to the format. This element has exactly two sub-elements, **head** and **data**.

The head Element

This element contains the meta-data or descriptive data for the table. It does not have any attributes and has exactly two sub-elements**, environment** and **fields**.

The environment Element

This element contains sub-elements that provide information about the station that generated the data and the program running on it. These elements include the following:

- **station-name** – Specifies the name of the station that generated the data. This element must be present.
- **table-name** – Specifies the name of the table as given in the datalogger program. This element must be present.
- **model** – Specifies the model number of the station. This element may be omitted if the information is not available.
- **serial-no** – Specifies the serial number of the datalogger. This element may be omitted if the information is not available.
- **os-version** – Specifies the version of the operating system running in the datalogger. This element may be omitted if the information is not available.
- **dld-name** – Specifies the file name of the program that is running in the datalogger. This element may be omitted if the information is not available.
- **dld-sig** – Specifies the signature of the program running in the datalogger. This element may be omitted if the information is not available.

The fields Element

This element specifies the meta-data for all of the fields in the header file. It will contain a collection of one or more **field** elements. One for each scalar element in the file (strings are considered to be scalar elements).

The field Element

This element specifies the meta-data for a single field. It is an empty element (contains no child elements) but defines the following attributes:

- **name** This attribute is required and specifies the name of the field. If the field is part of an array, the name will include the array subscripts as a comma separated list of integers within parentheses.
- **type** This required attribute specifies the data type for the field. This data type is a string that corresponds with a subset of XML Schema data types. The following values will be used within csixml:

Field Data Types	
xsd:string	Specifies string content
xsd:long	64 bit signed integer
xsd:unsignedLong	64 bit unsigned integer
xsd:int	32 bit signed integer
xsd:unsignedInt	32 bit unsigned integer
xsd:short	16 bit signed integer
xsd:unsignedShort	16 bit unsigned integer
xsd:byter	8 bit signed intege
xsd:unsignedByte	8 bit unsigned integer
xsd:float	32 bit floating point number
xsd:double	64 bit floating point number
xsd:boolean	Boolean value
xsd:dateTime	date and time stamp

- **units** This optional attribute will specify the units string provided by the datalogger program.
- **process** This optional attribute specifies the process string given by the datalogger program based upon the processing instruction used to output data into final storage.

The data Element

This element marks the beginning of data storage in the file. It will contain a collection of zero or more **r (record)** elements.

The r (record) Element

This element describes one table record. It can have the following optional attributes:

- **no** Specifies the record number for this record. These values indicate the logged order of the data and will generally increment by one with each record logged. Records can appear out of order, however, if one-way or data advise data is used in conjunction with hole collection. Missed numbers can signify missed records (holes).
- **time** Specifies the time stamp for the record. This format will conform to the standard XSD timestamp format.
- This element will contain as many **value** sub-elements as there are **field** elements in the **fields** header element.

The v (value) Element

This element conveys one scalar value (or string) for a field. There will be one of these elements for each **field** element defined in the **fields** element of the header. The contents of these element will be the data for that field. An optional attribute, n (field Name), is supported. This attribute can be included in order to increase the human readability of the file.

B.1.5.4 File Example

The following example was generated using the Tablemon2 utility:

```
<?xml version="1.0" standalone="yes"?>
<csixml version="1.0">
  <head>
    <environment>
      <station-name>logan-nw</station-name>
      <table-name>OneDay</table-name>
    </environment>
    <fields>
      <field name="BattVolt_Min" type="xsd:float" units="Volts" process="Min" />
      <field name="BattVolt_TMn" type="xsd:dateTime" units="Volts" process="TMn" />
      <field name="PnlTemp_Max" type="xsd:float" process="Max" />
      <field name="PnlTemp_TMx" type="xsd:dateTime" process="TMx" />
      <field name="PnlTemp_Min" type="xsd:float" process="Min" />
      <field name="PnlTemp_TMn" type="xsd:dateTime" process="TMn" />
      <field name="EncRH_Max" type="xsd:float" units="%" process="Max" />
      <field name="EncRH_TMx" type="xsd:dateTime" units="%" process="TMx" />
      <field name="EncRH_Min" type="xsd:float" units="%" process="Min" />
      <field name="EncRH_TMn" type="xsd:dateTime" units="%" process="TMn" />
      <field name="AirTemp_Max" type="xsd:float" process="Max" />
      <field name="AirTemp_TMx" type="xsd:dateTime" process="TMx" />
      <field name="AirTemp_Min" type="xsd:float" process="Min" />
      <field name="AirTemp_TMn" type="xsd:dateTime" process="TMn" />
```

```
    <field name="RH_Max" type="xsd:float" units="%" process="Max" />
    <field name="RH_TMx" type="xsd:dateTime" units="%" process="TMx" />
    <field name="RH_Min" type="xsd:float" units="%" process="Min" />
    <field name="RH_TMn" type="xsd:dateTime" units="%" process="TMn" />
    <field name="WindSpd_Max" type="xsd:float" units="m/s" process="Max" />
    <field name="WindSpd_TMx" type="xsd:dateTime" units="m/s" process="TMx" />
    <field name="WindDir_SMx" type="xsd:float" process="SMM" />
    <field name="WindSpd_Min" type="xsd:float" units="m/s" process="Min" />
    <field name="WindSpd_TMn" type="xsd:dateTime" units="m/s" process="TMn" />
    <field name="WindDir_SMn" type="xsd:float" units="m/s" process="SMM" />
    <field name="SlrFDensity_Max" type="xsd:float" units="W/m" process="Max" />
    <field name="SlrFDensity_TMx" type="xsd:dateTime" units="W/m" process="TMx" />
    <field name="SlrTotalF_Tot" type="xsd:float" units="MJ/m" process="Tot" />
    <field name="BP_Max" type="xsd:float" units="mmHg" process="Max" />
    <field name="BP_TMx" type="xsd:dateTime" units="mmHg" process="TMx" />
    <field name="BP_Min" type="xsd:float" units="mmHg" process="Min" />
    <field name="BP_TMn" type="xsd:dateTime" units="mmHg" process="TMn" />
    <field name="Rain_Tot" type="xsd:float" units="mm" process="Tot" />
  </fields>
</head>
<data>
  <r no="340" time="2006-08-16T00:00:00">
    <v1>12.96</v1>
    <v2>2006-08-15T17:18:00</v2>
    <v3>31.44</v3>
    <v4>2006-08-15T17:16:07</v4>
    <v5>15.51</v5>
    <v6>2006-08-15T23:01:05</v6>
    <v7>57.54</v7>
    <v8>2006-08-15T20:18:49</v8>
    <v9>34.96</v9>
    <v10>2006-08-15T10:22:10</v10>
    <v11>31.12</v11>
    <v12>2006-08-15T17:06:55</v12>
    <v13>14.06</v13>
    <v14>2006-08-15T22:47:40</v14>
    <v15>82.7</v15>
    <v16>2006-08-15T05:00:20</v16>
    <v17>10.45</v17>
    <v18>2006-08-15T17:21:13</v18>
    <v19>10.98</v19>
    <v20>2006-08-15T00:28:05</v20>
    <v21>127.8</v21>
    <v22>0</v22>
    <v23>2006-08-15T01:39:45</v23>
    <v24>0</v24>
    <v25>1108</v25>
    <v26>2006-08-15T10:13:37</v26>
    <v27>24.76</v27>
    <v28>766.1</v28>
```

```
    <v29>2006-08-15T07:15:00</v29>
    <v30>761.7</v30>
    <v31>2006-08-15T18:30:00</v31>
    <v32>0.254</v32>
  </r>
  <r no="341" time="2006-08-17T00:00:00">
    <v1>12.97</v1>
    <v2>2006-08-16T16:09:52</v2>
    <v3>31.16</v3>
    <v4>2006-08-16T15:49:27</v4>
    <v5>10.43</v5>
    <v6>2006-08-16T06:13:46</v6>
    <v7>52.5</v7>
    <v8>2006-08-16T22:20:10</v8>
    <v9>32.15</v9>
    <v10>2006-08-16T10:19:42</v10>
    <v11>30.34</v11>
    <v12>2006-08-16T15:37:33</v12>
    <v13>10.06</v13>
    <v14>2006-08-16T05:25:24</v14>
    <v15>84.6</v15>
    <v16>2006-08-16T04:26:52</v16>
    <v17>13.21</v17>
    <v18>2006-08-16T17:28:53</v18>
    <v19>11.86</v19>
    <v20>2006-08-16T14:34:31</v20>
    <v21>217.4</v21>
    <v22>0</v22>
    <v23>2006-08-16T00:02:48</v23>
    <v24>0</v24>
    <v25>922</v25>
    <v26>2006-08-16T12:47:13</v26>
    <v27>26.68</v27>
    <v28>764.5</v28>
    <v29>2006-08-16T10:00:00</v29>
    <v30>761.7</v30>
    <v31>2006-08-16T18:45:00</v31>
    <v32>0</v32>
  </r>
 </data>
</csixml>
```

B.1.6 CSIJSON

CSIJSON is a file format that is relatively easy to parse in any language but more particularly so in JavaScript since it adopts the same syntax rules that are used for JavaScript object initialization. Its structure is much like CSIXML It is very easy to digest in a JavaScript or ActionScript (Flash) environment and is probably the most efficient means of handling CSI generated data in a web browser context.

The CSIJSON file format is available for some CRBasic instructions including TableFile and the WebPageBegin/WebPageEnd Format command.

B.1.6.1 A Short Introduction to JSON

Much like XML, JSON is a recursive structure with a root object (represented by an opening and closing curly brace ('{' and '}'). This root object can contain named strings, numbers, objects, and arrays. A simple object specification follows:

```
{
  "head": {
    "signature": xxxx,
    "transaction": "xxxxyyyy",
    "environment": {
    }
    "fields": [
    ]
  },
  "data": [ ]
}
```

This declaration declares an object that contains two empty sub-objects, head, and data. In a JavaScript program, a string in this format can be easily parsed using the Eval() function or the newer ParseJSON() function. Once parsed, the data contained therein can be accessed using standard JavaScript notation.

B.1.6.2 File Syntax

CSIJSON contains two subordinate objects: "head" and "data". The head object contains the station meta-data and field descriptions while the data object is an array of record objects.

The head Object

The head object contains information about the datalogger and program that is responsible for generating the data as well as information about the fields in the data.

head.signature

This numeric value is the signature calculated on the table definitions. This value can be used by the web client to determine whether the table definitions have changed while that client is monitoring or polling for data. If the web client is using the DataQuery command in the datalogger web services and specifies a tablesig value that matches this value, the server will not send the head.environment or head.fields values.

head.transaction

This optional value specifies a transaction identifier that can be sent by a web client in the DataQuery parameter. This value can help the client route responses to the correct object.

head.environment Object

The environment object contains information about the datalogger and its program and the data table.

station_name	Specifies the name of the station. This can either be the name of the “Station Name” setting or can be the name of the station device in LoggerNet’s network map.
table_name	Specifies the datalogger table name.
model	Specifies the model of the datalogger that produced this data. This object is optional.
serial_no	Specifies the serial number for the datalogger that produced this data. This object is optional.
os_version	Specifies the version of the operating system for the datalogger that produced this data. This object is optional.
dld_name	Specifies the name of the datalogger program that produced this data. This object is optional.
dld_sig	Specifies the CSI signature of the datalogger program that produced this data. This object is optional.

head.fields Array

The fields object is an array of field descriptions for the data contained in this file. The order and number of field descriptions in this array must match exactly that of the actual data.

Each field description object will contain the following objects:

name	Specifies the name of the field as assigned by the datalogger program.	
type	Specifies the expected data type for this field. These strings will include the following:	
	xsd:string	The data will be expected to be formatted and treated as string data
	xsd:long	The data will be expected to represent a signed eight byte integer.
	xsd:unsignedLong	The data will be expected to represent an unsigned eight byte integer.
	xsd:int	The data will be expected to represent a signed four byte integer.
	xsd:unsignedInt	The data will be expected to represent an unsigned four byte integer.
	xsd:short xsd:	The data will be expected to represent a signed two byte integer.
	unsignedShort	The data will be expected to represent an unsigned two byte integer.
	xsd:byte	The data will be expected to represent a signed one byte integer.
	xsd:unsignedByte	The data will be expected to represent an unsigned one byte integer.
	xsd:float	The data will be expected to represent a four byte floating point number.
	xsd:double	The data will be expected to represent an eight byte floating point number.
	xsd:timeStamp	The data will be expected to a time stamp formatted as a string.
	xsd:boolean	The data will be expected to be a boolean value (true, false, 1, or 0).
units	Specifies the units string for this field as assigned by the datalogger program.	
process	Specifies the output processing instructions parameters as specified by the datalogger program.	

the data Array

The data array is an array of record objects. Each record object will contain the following subordinate objects:

- **no** An integer that specifies the record number as assigned by the datalogger.
- **time** Specifies the time stamp assigned by the datalogger.
- **vals** An array of the data values for this record. Each element in this array must correspond with the equivalent element in the head.fields array.

B.1.6.3 File Example


```
{
  "head": {
    "signature:" 21334,
    "transaction": "xxxxyyyy",
    "environment": {
      "station_name": "jon-cr1000",
      "table_name": "one_day",
      "model": "CR1000",
      "serial_no": "1084",
      "os_version": "cr1000.std.18",
      "dld_name": "lights-web.cr1",
      "dld_sig": "31837"
    },
    "fields": [
      {
        "name": "temp_degf_Min",
        "type": "xsd:float",
        "units": "DegF",
        "processing": "Min"
      },
      {
        "name": "temp_degf_TMn",
        "type": "xsd:dateTime",
        "units": "",
        "processing": "TMn"
      },
      {
        "name": "temp_degf_Avg",
        "type": "xsd:float",
        "units": "DegF",
        "processing": "Avg"
      },
      {
        "name": "temp_degf_Max",
        "type": "xsd:float",
```

```
      "units": "DegF",
      "processing:" "Max"
     },
     {
      "name": "temp_degf_TMx",
      "type": "xsd:dateTime",
      "units": "",
      "processing": "TMx"
     }
    ]
   }
  "data": [
   {
    "no": 43,
    "time": "2010-01-20T00:00:00",
    "vals": [
     69.62625, "2010-01-19T07:53:40", 73.69058, 78.82542,
     "2010-01-19T17:41:05"
    ]
   },
   {
    "no": 44,
    "time": "2010-01021T00:00:00",
    "vals": [
         70.85629, "2010-01-20T08:14:40", 74.24667, 77.28296,
         "2010-01-20T17:41:51"
    ]
   },
   {
    "no": 45,
    "time": "2010-01-22T00:00:00",
    "vals": [
     70.90952, "2010-01-21T07:17:08", 74.41795, 78.02577,
     "2010-01-21T17:39:01"
    ]
   }
  ]
}
```


B.2 Datalogger Data Formats

B.2.1 TOB2 or TOB3

TOB2 and TOB3 files are created when data are retrieved from external PC cards on dataloggers such as the CR9000, CR5000, and CR1000. The TOB2 file format has been replaced in new dataloggers by the TOB3 file format. TOB3 format is similar to TOB2 in most respects but differs from TOB2 in the following ways:

- Frame headers in TOB3 are twelve bytes long rather than eight bytes long. The additional four bytes contain an unsigned integer with the least significant byte written first to identify the record number for the first record in the frame.
- In the TOB3 format, the offset field in the major frame footer no longer represents the number of frames back to the last minor frame. This information is used in TOB2 to help accelerate searching for data but is not considered to be necessary in TOB3 because of the presence of the record number in the frame header.

The TOB2 or TOB3 binary file format has the following structure with each header line terminated with a carriage return and line feed (CRLF):

<table>
<tr><td colspan="3">ASCII Header Line 1</td></tr>
<tr><td colspan="3">ASCII Header Line 2</td></tr>
<tr><td colspan="3">ASCII Header Line 3</td></tr>
<tr><td colspan="3">ASCII Header Line 4</td></tr>
<tr><td colspan="3">ASCII Header Line 5</td></tr>
<tr><td colspan="3">ASCII Header Line 6</td></tr>
<tr><td>Frame Header</td><td>Frame Body n</td><td>Frame Footer</td></tr>
</table>

Header line one describes the file environment with the following fields:

- Data file type (TOB2 or TOB3).
- Station name.
- Model name of the datalogger.
- Serial number of the datalogger.
- Operating system on the datalogger.
- Name of the program running in the datalogger.

- Signature of the program running in the datalogger.
- The time that the file was created.

Header line two contains:

- The name of the table as declared in the datalogger program.
- The non-timestamped record interval.
- The data frame size.
- The intended table size.
- The validation stamp.
- The frame time resolution.

Header line three describes the names for each field in a table record as determined by the datalogger program.

Header line four describes the units associated with each field in the record. Units are optional and are specified in the datalogger program, if they are included. If no units are provided in the program, then an empty string placeholder is placed in this line for that specific field.

Header line five describes the processing performed in the datalogger to produce the value for each field in the record; for example, sample, average, min, max, etc. If there is no known processing for a field, that field will be assigned an empty processing string. There will be one value on this line for each field name given on header line three.

Header line six defines the data types for each field in the record and supports the following values: IEEE4, FP2, ULONG, LONG, SecNano, and ASCII(len).

TOB2 frame headers are eight bytes long and hold the timestamp for the first record in the frame. TOB3 frame headers are twelve bytes long and contain the same timestamp information but also add a four-byte unsigned integer that represents the beginning record number for that frame.

The frame data begins immediately following the frame header and consists of zero or more data records. Each record contains one data point for each of the field names identified in header line three. The data type and implied size of these data points are identified by the data types list given by header line six.

The frame footer makes up the last four bytes of the frame.

B.3 Binary Data Value Types

When data is written in datalogger memory or in binary data files each value must be assigned a particular data type. These data types describe the format of the data.

B.3.1 FP2 (2 Byte Low Resolution Format)

A two-byte floating-point number format created by Campbell Scientific, Inc. and used to store low-resolution values. Basically, this format consists of a single sign bit, a two-bit negative decimal exponent, and a 13-bit mantissa.

B.3.2 FP4 (4 Byte High Resolution Format)

A four-byte floating-point number format created by Campbell Scientific, Inc. and used for input location values as well as high-resolution final storage values. This format consists of a single sign bit, a seven-bit base-two exponent, and a 24-bit mantissa.

B.3.3 IEEE4

A standard four-byte floating-point number format used for certain values within a record. This format consists of a single sign bit, an eight-bit binary exponent, and a 23-bit mantissa.

B.3.4 IEEE8

A standard eight-byte floating-point number format used for certain values within a record. This format consists of a single sign bit, an 11-bit exponent, and a 52-bit mantissa.

B.4 Converting Binary File Formats

Campbell Scientific dataloggers not only use the previously mentioned binary file formats but users may also choose to use and access these binary formats on the PC. Binary files may be output as data files to the PC by LoggerNet to save hard disk space or to accommodate a user that is only interested in using binary files in an application. In addition, PC cards that are written by the datalogger will contain binary files that can be accessed directly by the PC.

Binary files cannot be interpreted through mere visual inspection. Therefore, binary file processing tools are available to read and convert these binary data files to ASCII text. These conversion tools are Split, View Pro, CardConvert, File Format Convert, and TOB32.EXE. Refer to Working with Data Files (p. 249) for complete information on View Pro and CardConvert. Refer to Utilities Installed with LoggerNet (p. 335) for complete information on File Format Convert.

B.4.1 CardConvert

The CardConvert program can convert TOB1, TOB2, and TOB3 binary files to TOA5, Array Compatible CSV, or CSIXML file format. It can also be used to convert TOB2 or TOB3 binary files to TOB1 file format.

B.4.2 File Format Convert

File Format Convert can convert TOA5, TOACI1, TOB2, TOB3, TOB1, or CSIXML files to TOA5, TOACI1, TOB1, CSIXML, or CSV file format.

B.4.3 View Pro

View Pro converts TOB1, TOB2 and TOB3 data files and generates a new TOA5 file. Once the TOA5 file is generated, View Pro can display the converted data. In that sense, View Pro is a software tool that combines the conversion of binary files to TOA5 with the ability to view the data once the file has been converted.

B.4.4 Split

Split has the capability of reading TOB1, TOB2, and TOB3 files and displaying data from those files in ASCII format. The output parameters are user specified and Split generates a file containing the converted ASCII format values.

B.4.5 TOB32.EXE

The TOB32.EXE command line utility is installed by default in the LoggerNet program directory at C:\Program Files (x86)\Campbellsci\Loggernet\tob32.exe. The output is similar to CardConvert. Command line switches are used to determine the new file format that will be created. Some of the basic switches available are listed below:

-h or -? | Help

-a | ASCII (TOA5) Generates CSI Table Oriented ASCII version 5 format files

-b | Binary (TOB1) Generates CSI Table Oriented Binary version 1 format files

Some examples using these switches include:

tob32.exe -a mydata.dat (converts mydata.dat to TOA5 format)

tob32.exe -b mydata.dat (converts mydata.dat to TOB1 format)

NOTE: TOB32.EXE is now obsolete and has been replaced by CSIDFT_convert.exe. It is included only for backwards compatibility. CSIDFT_convert.exe should be used for any new functionality as TOB32.EXE does not support all current data types.

B.4.6 csidft_convert.exe

The csidft_convert.exe command line utility is installed by default in the LoggerNet program directory at C:\Program Files (x86)\Campbellsci\LoggerNet\csidft_convert.exe. It takes as input the name of a data file in one of Campbell Scientific's standard formats and will create a second file in another specified format.

It has the following syntax:

csidft_convert input_file_name output_file_name output-format

where:

input_file_name = the file name of your input file

output_file_name = the output file name to be created

output-format = one of the following options: toaci1, toa5, tob1, csixml, custom-csv, no-header

When converting an array-based file, you must include the following parameters:

--fsl = fsl_file (This is the *.FSL file for the input file.)

--array = array_id (The array id of the array in the input file to be converted.)

All output formats other than toaci1 have an additional optional parameter:

--format-options = format-options (This is an integer value as described below for the different output formats. In each case, add the numbers together for all desired options and input this number in the --format-options parameter.)

NOTE: The format-options parameter does not need to be used to include timestamp and record number. They are included by default

TOA5

Include Timestamp	1
Include Record Number	2
Midnight is 2400	4

TOB1

Include Timestamp	1
Include Record Number	2

CSIXML

Include Timestamp	1
Include Record Number	2
Include Field Names in each row	4
Midnight is 2400	8

Custom-CSV

Include Seconds	1
Include Hour/Minutes	2
Include Julian Day	4
Include Year	8
Midnight is 2400	16
Include Array ID	256
Array ID	Desired array ID (between 1 and 1023) multiplied by 65,536

No-Header

Include Timestamp	1
Include Record Number	2
Surround strings by quotation marks	4
Midnight is 2400	8

Examples

The following example converts myinput.dat to TOA5 format and stores it in myoutput.dat:

```
csidft_convert.exe myinput.dat myoutput.dat toa5
```

The following example converts myinput.dat to custom-csv format and stores it in myoutput.dat with an array ID of 15 and data values included for seconds, hour/minutes, Julian day, and year:

```
csidft_convert.exe myinput.dat myoutput.dat custom-csv --format-options=983311
```

The following example looks for array ID 20 in myinput.dat, converts it to TOA5 format using myfsl.fsl, and stores it in myoutput.dat:

```
csidft_convert.exe myinput.dat myoutput.dat ToA5--array-id=20 --fsl=myfsl.fsl
```

NOTE: If the utility does not reside in the same directory as the data files, the entire directory paths must be used. Also, note that the utility will overwrite any existing file with the same name as output_file_name. Therefore, use caution in specifying the output_file_name.

NOTE: TOB1 requires that a size be specified for each field containing a string. Therefore, when converting a file containing strings to TOB1, this utility will estimate the length of each field containing strings. It will do this by taking the length of the string in the first record and multiplying it by 10 with a minimum string size of 64. If a string in the field is too large to fit into this estimated string size, it will be truncated.

B.5 RTMS Format Description

Refer to RTMS Format Description (p. 378).

Appendix C. Software Organization

C.1 LoggerNet/Client Architecture

The LoggerNet communication server provides the interface to all of the dataloggers and the support for the different communications mediums. It runs in the background and provides an attachment for the clients that provide the user interface. The server handles all communications with the dataloggers.

The LoggerNet server handles connections from all the user interface screens simultaneously, allowing many different views and ways to access the data collected from the dataloggers. In addition to running on the same computer with LoggerNet, some client applications can be run on other computers connected to the LoggerNet computer over a local area network (LAN).

LoggerNet can automatically collect data from the dataloggers on a schedule as well as on request from the user. It can automatically check and update the clocks in the dataloggers and handle administration support functions.

C.2 LoggerNet Server Data Cache

The LoggerNet server data cache is a set of files kept on the hard disk of the computer where the server is running. These data files are in binary format and can only be used or interpreted by the

LoggerNet server. The data cache files are stored in addition to the output data files.

C.2.1 Organization

The data cache is set up to emulate the way data is stored in the datalogger. When a new datalogger station is defined for the network and communication is established with the station, the server requests the table definitions from table data dataloggers. For array based dataloggers the array definitions are contained in a final storage label file that is associated with a datalogger. This table or array information is used to set up equivalent tables and data arrays for data storage in the data cache. The size of the areas set up in the data cache is dependent on the size of final storage in the datalogger.

Datalogger tables that hold only one record, such as the Input Locations table and the Status table, would have only two records assigned in the data cache.

The storage in the data cache is designed to operate with "ring memory" just like the datalogger. This means that records will be stored in the data cache area for that table until it has reached the maximum number of records, the next data record will replace the oldest record in the storage table, and so on.

C.2.2 Operation

Normal data collection from the datalogger is done with polling based on the scheduled collection interval set up by the user. This is the most efficient means of data collection for networks with rapid direct communications links. When it is time for a scheduled data collection the server sends a data poll request to the datalogger to get all of the data stored in the selected tables since the last poll. The tables to be collected are specified by the user in the Setup Screen.

As each record is written to the data cache, the server adds a filemark number to the record as it is stored. This filemark number is used to identify discontinuities in the data. The filemark number starts out as zero when the table for the data cache is created or re-initialized. This number is incremented each time a discontinuity is seen in the data records. Such a discontinuity can occur when there is a gap in the record numbers because the data table filled and overwrote the requested data. This also can occur if the record number rolls over from the maximum to start back at zero or an identical program is loaded into the datalogger without going through the server.

Data can also be collected from the datalogger using a manual poll operation. This is achieved by selecting **Collect Now** from the Connect Screen. When a manual poll is done the data from the datalogger is saved in the output data file and is also put into the data cache.

C.2.3 Retrieving Data from the Cache

Once the data has been stored in the data cache it is retrieved by the applications such as the graphical and numerical displays that request the data by datalogger, table or array, and data field. The data can be requested by a query where the request specifies the starting and ending timestamp or record number along with the data to retrieve.

C.2.4 Updating Table Definitions

When the table definitions are obtained from the datalogger they are kept in the server and used to identify the data available in the data cache. Every time new data is collected from a datalogger, a table definition signature is sent that should match the signature stored in the server. If this table definition signature doesn't match it indicates that the table definitions in the datalogger have changed.

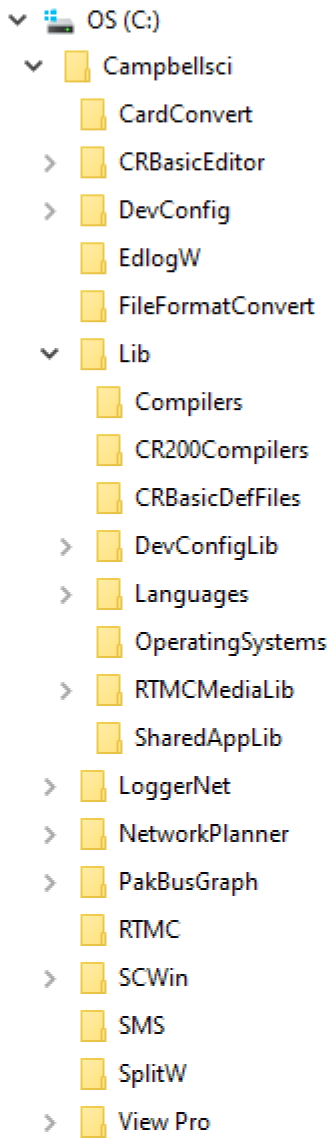
There are a number of things that could cause datalogger table definitions to change. A new program may have been downloaded to the datalogger, or the keyboard display may have been used to manually make changes to the datalogger program.

> **NOTE:** If the datalogger program is re-compiled without changing table definitions, the record numbers will reset to zero causing the server to assume the datalogger record numbers have wrapped around. This will result in the re-collection of all of the data in the datalogger.

When a change in table definitions is detected, the server stops data collection and indicates in the Collection State of Status Monitor that the table definitions have been changed. Data collection cannot be restarted until either a new datalogger program is loaded into the datalogger by the server, or updated table definitions are received from the datalogger. Either of these actions causes the data in the data cache for that datalogger to be removed and new data cache tables set up based on the new table definitions for that datalogger. LoggerNet will save the existing output data file with a modified name and create a new output data file.

C.3 Directory Organization

The default installation of the LoggerNet software creates folders and installs software in two directories: the C:\CampbellSci working directory and C:\Program Files (x86)\Campbellsci\LoggerNet program directory.



C.3.1 Working Directory

When LoggerNet is installed, several directories are created under the C:\CampbellSci directory. A LoggerNet directory is created, as well as a directory for all of LoggerNet's client applications (such as CRBasic Editor, SCWin, SplitW, and View Pro).

The C:\CampbellSci\LoggerNet directory is used for storing data files and system status information files from LoggerNet. System files are stored in the Sys directory. The C:\CampbellSci\LoggerNet\Sys\Bin subdirectory contains the configuration file for your datalogger network (CsiLgrNet.xml). This subdirectory has a C:\CampbellSci\LoggerNet\Sys\Bin\Data subdirectory that is used to store the data cache for the devices in the network. The C:\CampbellSci\LoggerNet\Sys\Inifiles subdirectory contains all configuration files for LoggerNet applications which do not have a C:\CampbellSci subdirectory (such as the Setup Screen or Status Monitor).

There is a C:\CampbellSci\LoggerNet\Data directory that is used to store data files from Custom Data Collection, which is launched from the **Custom** button on the Connect Screen. Note that regular data collection (triggered by choosing **Collect Now** from the Connect Screen) and scheduled data collection both store data to the same file identified in the Setup Screen for the datalogger, which, by default, is C:\CampbellSci\LoggerNet. Only Custom Data Collection stores data to the file in the C:\CampbellSci\LoggerNet\Data directory.

The client application directories are used for storing system related files for each application, and they can also be used for storing user files for that application (such as the *.SCW program file for Short Cut for Windows). The client applications are each given their own directories so that if more than one Campbell Scientific software product is installed on the system, common applications will be shared among these software applications. For instance, you may have PC400 and LoggerNet installed on the same computer. Both of these applications include the CRBasic Editor. By sharing directories among CSI applications, you have only one "instance" of CRBasic running on your machine, and it will look the same regardless of whether it is started from LoggerNet or PC400.

> **NOTE:** Changing or removing any of the files in the C:\CampbellSci\LoggerNet\SYS\bin directory can cause loss of data, a system crash or destruction of the network map. There are no user editable files in this directory.

System administrators concerned about security and system integrity should use Windows and its directory access tools to control access to the working directories.

C.3.2 Program Directory

Most files necessary for running LoggerNet and its applications are stored in theC:\Program Files (x86)\Campbellsci\LoggerNet subdirectory. Applications, such as RTMC and SCWIN, have their own subdirectories under C:\Program Files (x86)\Campbellsci.

No other files are saved to these program file subdirectories in order to protect the integrity of the LoggerNet communication server and the clients.

Appendix D. Log Files

D.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.

D.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.

D.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.

D.1.3 Log File Message Formats

D.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.

D.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)	

Transaction Log Messages				
Code	**Message Text**	**Message Parameters**	**Message Meaning**	**User Response to Message**
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.	
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.

Transaction Log Messages				
Code	**Message Text**	**Message Parameters**	**Message Meaning**	**User Response to Message**
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 appears 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.	Codes 1 and 3 are rare. If ever seen contact an technical support engineer at Campbell Scientific. 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. 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.

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 **Collect Now** 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	The packet message type code: 0 Packet Delivery Fault Notification 1 Status/Warning/Fault Notification 2 Network Description Transaction 3 Clock Check/Set Transaction 4 Program Down-load Transaction 5 Program Up-load Transaction 7 Data Advise Command Transaction 8 Data Advise Notification Packet 9 Hole Collection Command Transaction 10 Control Command (Set Variable) Transaction 11 User I/O Transaction (Terminal Mode) 12 Memory Image Down-load	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	The packet message type code: 0 Packet Delivery Fault Notification 1 Status/Warning/Fault Notification 2 Network Description Transaction 3 Clock Check/Set Transaction 4 Program Down-load Transaction 5 Program Up-load Transaction 7 Data Advise Command Transaction 8 Data Advise Notification Packet 9 Hole Collection Command Transaction 10 Control Command (Set Variable) Transaction 11 User I/O Transaction (Terminal Mode) 12 Memory Image Down-load	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.	

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		
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.	Usernames 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 retrieval 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"
```

D.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.

D.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"
```

D.2 CQR Log (RF Link)

The CQR log contains information about the quality of communication each time an RFBase is dialed. (Note that information on an RFBase-TD link is not contained in the CQR log). The CQR log is written to the <*working directory*>\Logs directory. By default, this is C:\Campbellsci\LoggerNet\Logs.

Each time an RF link is shutdown, an entry will be written to the CQR Log. The first line in each entry is the timestamp and the name of the datalogger being communicated with. The remaining lines are the RF Link Quality Accumulators (RLQA) for each modem in the link. The RLQA are representative of the active period of the link. The line for each modem will contain three numbers:

xxxx yyyy zzzz

where

xxxx = Number of communication failures
yyyy = Noise Level Indicator
zzzz = Noise Level Indicator

A communication failure occurs when a signature of a block of data does not match its original signature. These blocks are subsequently retransmitted. The noise level indicators should be 102 (±70) at 3.0K baud rate or 124 (±70) at 2.4K baud.

Example CQR log entry

"10/14/2010 12:10:35",CR10XTD
0002 0128 0055
0000 0129 0063

The first line is the timestamp and name of the datalogger being communicated with. The next line is the RLQA for the EOL (End of Link) modem. This is the remote modem connected to the datalogger. The last line is the RLQA for the SOL (Start of Link) modem. This is the base modem. (This entry is for a link that contains no repeaters. A link with repeaters would show an additional line for each repeater between the EOL line and the SOL line.) The 0002 indicates that two interruptions occurred on the EOL modem while the link was active. All noise level indicators are within acceptable bounds in this example.

Appendix E. Calibration and Zeroing

E.1 Calibration Essentials

E.1.1 Definition of Calibration

Calibration, in general, refers to actions taken on a measurement system to increase its accuracy. This is usually done by matching the system's outputs to known "control" values in order to increase confidence in the measurement of future unknowns.

Campbell Scientific's approach to calibration uses a datalogger's measurement and computational capability to calculate the multipliers and/or offsets to be used by a measurement instruction to provide more accurate readings. The process of calibration uses the datalogger to assist the operator in intelligently and automatically setting the multiplier and offset to be used in a measurement instruction in order to obtain more accurate output data.

Calibration is periodically necessary when there has been sensor drift or other variation in sensor outputs. When a calibration instruction is part of the datalogger program, it is quick and easy to use a software Wizard to change the measurement configuration at run-time. This saves time over previously used methods, such as re-writing the CRBasic program or interfering with measurements to obtain calibration constants manually. With this method, changes to multipliers and offsets can be made quickly and automatically without rewriting datalogger programs or interfering with sensor measurements.

E.1.2 Basic Calibration Process

When calibrating with a Campbell Scientific datalogger, known and measured values are given as inputs. The outputs of the calibration then become the new values for the multiplier and offset variables in the CRBasic program. If needed, these calculated multipliers and offsets can be permanently stored and automatically reloaded upon program restart (such as when a power-cycle occurs on the datalogger). The datalogger makes use of a calibration file (*.cal) to store these calibration values and load them as desired. This can be done at datalogger power-up or at other times designated within the datalogger program.

To evaluate calibration histories, a final storage output table can be configured to store the results of calibrations that have been performed, and the date and time at which those calibrations were performed. This data is separate from the calibration file and forms a permanent history of calibration constants used within the program.

E.2 Writing Calibration Programs with the CRBasic Editor

E.2.1 The FieldCal Instruction

If you wish to make measurements that will be calibrated as discussed above, you should use the `FieldCal()` instruction within the CRBasic program. When the program is running in the datalogger, you can use the LoggerNet Calibration Wizard to perform the actual calibrations (in real-time) on the sensors that were previously designated for calibration. You can also perform a manual calibration against a running program using the LoggerNet Numeric Display (see Using the Calibration Wizard with Running Programs (p. 491)) or from a keyboard display connected directly to the datalogger.

The `FieldCal()` instruction works together with other related CRBasic instructions to complete the calibration task. These instructions are shown in the following table.

The FieldCal Instruction Family

Instruction	Description
FieldCal	This is the main calibration instruction. The CRBasic program should contain one `FieldCal()` instruction per measurement requiring calibration. This instruction is placed after the measurement instruction to which it applies.
LoadFieldCal	(optional) This instruction loads values into program variables from the calibration file (*.cal), if it exists. It will also indicate whether the attempt to load those values was successful or not by returning a Boolean (true/false) result.

The FieldCal Instruction Family	
Instruction	**Description**
SampleFieldCal	(optional) This is a table output instruction. It writes the latest calibration values for all calibrated measurements to a data table (separate from the *.cal file).
NewFieldCal	(optional) This is a Boolean system value indicating when a calibration has succeeded. During one scan cycle after a calibration has occurred this value will be true. Its value is then set to false until another calibration occurs. The value of this variable cannot be set within a CRBasic Program, but only evaluated. The main purpose for this variable is to be used together with the `SampleFieldCal()` instruction to output one table record per calibration to a specified table.

To set up a measurement for calibration in CRBasic, first insert the instruction(s) that make the measurement, using variables for the multiplier and/or offset. Then add a `FieldCal()` instruction after the measurement instruction and refer back to the measurement being calibrated using the variable containing the measured output. Provide the `FieldCal()` instruction with the variables holding the multiplier and the offset of the measured sensor. If you need to retrieve a calibration value into multipliers and offsets upon program start or under other conditions, use the `LoadFieldCal()` instruction. To store calibration values to a data table (in addition to the values stored in the *.cal file), use the `SampleFieldCal()` table output instruction with the `NewFieldCal()` system variable as the trigger.

For more information about how to use these instructions, refer to the `FieldCal()` instruction topic of your datalogger manual, or use the online help topic for `FieldCal()` within the CRBasic Editor.

E.2.2 Calibration File Details

It is important to understand the purpose and function of the calibration file created by a CRBasic program when using the `FieldCal()` instruction. The calibration file has the same name as the program that creates and uses it, except that it ends with a .cal extension. For example, myProg.CR1 would generate a calibration file called myProg.CAL. The calibration file is located on the same datalogger storage device as the program that creates it (e.g., CPU, CRD, USR). The calibration file is created when the program runs and doesn't find an existing calibration file that it can use, and it is updated upon each successful calibration. The calibration file contains information about the latest calibrations performed during program execution and information that the LoggerNet Calibration Wizard needs to step users through the calibration process.

E.3 Four Kinds of Calibration

The `FieldCal()` instruction family can perform four basic kinds of calibrations: Zeroing Calibration, Offset Calibration, Two-point Multiplier/Offset Calibration (Linear Fit), and Two-point Multiplier Only Calibration. These calibration types are described below.

E.3.1 Zeroing

Zeroing is the act of placing a sensor into a state where the output condition is known to be zero and changing the measurement’s offset variable so that the sensor output reads as zero. By measuring the output of the sensor in this specialized condition (the zero condition), the offset variable will be changed to ensure that the known zero condition results in a measurement value of zero. Note that this process only changes the offset variable that is shared between the measurement instruction and the `FieldCal()` instruction. The multiplier is unaffected.

A simple example of zeroing would be taking off all items from a scale designed to measure the mass of objects. With nothing on the scale, this is the condition in which the scale should give a “zero” reading for its output. The calibration is triggered and the offset is adjusted to ensure the scale gives a zero reading for that condition.

To perform a zeroing calibration, use an argument of 0 (the number zero) for the calibration type in the `FieldCal()` instruction of your CRBasic program. The Calibration Wizard can be used to calculate and apply the proper offset while the program is running in the datalogger, or code can be configured within the CRBasic program to trigger the zeroing event based on flags or other user-defined conditions that occur while the program runs.

E.3.2 Offset Calibration

Offset Calibration is the act of placing a sensor into a state where the output condition is known to be a certain value and then changing the measurement’s offset variable so that the sensor output reads as exactly that value. It is similar to a zeroing calibration, except that the known value is a non-zero value. By measuring the output of the sensor in this specialized condition (the known offset condition), the offset variable will be changed to ensure that this condition results in a measurement value that matches it. Note that this process only changes the offset variable that is shared between the measurement instruction and `the FieldCal()` instruction. The multiplier variable is unaffected.

A simple example of offset calibration would be placing an object of known weight (such as 10 lbs.) on a scale designed to measure the mass of objects. With a known weight on the scale, this is the condition in which the scale should give a known reading for its output. First the calibration is triggered, then the user informs the datalogger about the value of the known weight, and finally the offset is adjusted to ensure that the scale gives a properly matched reading for that condition.

To perform an offset calibration, use an argument of 1 (the number one) for the calibration type in the `FieldCal()` instruction of your CRBasic program. The Calibration Wizard can be used to calculate and apply the proper offset while the program is running in the datalogger, or code can be configured within the CRBasic program to trigger the offset event based on flags or other user-defined conditions that occur while the program runs.

E.3.3 Two-Point Multiplier and Offset Calibration

Two-point multiplier and offset calibration uses a linear fit technique against two different known value conditions of the sensor's measurement. The sensor is placed into the first condition, and the known value for that condition is provided to the datalogger program. One or more measurements of that first condition are stored, and then the datalogger informs the user that the second known condition should now be applied to the sensor. The second condition is applied and its known value is then provided to the datalogger. The datalogger then measures the second condition. When the measurement of the second point condition is complete, a linear fit of the two points is calculated. The results are a slope value (m value, or multiplier), and a y-intercept (b value or offset). Thus the simple form y=mx+b is a representation of the linear fit, where m is the new multiplier value used and b is the new offset used.

A simple example of a two-point multiplier and offset calibration would be placing two objects of known weight (such as 5 lbs. and 15 lbs.) on a conventional scale at two different times. With the first known weight on the scale (5 lbs), this is the first condition in which the scale should give a known reading for its output. The calibration is triggered, the datalogger is informed of the value of the known weight, and the measurement is read. The datalogger then notifies the user that it is ready to measure the second point. The second known weight is placed on the scale (15 lbs), and this is the second condition in which the scale should give a known reading for its output. The second point of calibration is triggered, the datalogger is informed of the value of the second known weight, and the measurement is read. At this point the new multiplier and offset are calculated by the datalogger and the variables are adjusted accordingly to ensure that the scaling gives a properly matched reading for those two conditions. For future measurements (unknowns), a linear response will be used based on the line defined by those two points.

To perform a two-point multiplier and offset calibration, use an argument of 2 (the number two) for the calibration type in the `FieldCal()` instruction of your CRBasic program. The Calibration Wizard can be used to calculate and apply the two different known conditions while the program is running in the datalogger.

E.3.4 Two-Point Multiplier Only Calibration

Two-point multiplier only calibration uses a linear fit technique against two different known value conditions of the sensor's measurement, but only the slope value (multiplier) is calculated and changed. The offset is unaffected by this calibration. The sensor is placed into the first condition, and the known value for that condition is provided to the datalogger program. One or more measurements in that first condition occur, and then the datalogger informs the user that the second known condition should be applied to the sensor. When that condition is applied, the second known value is provided to the datalogger, and the datalogger measures the second condition. After completing the measurement of the second point condition, a best fit of the two points is calculated, resulting in a slope value (m value or multiplier) with the offset assumed to be zero. Thus the simple form y=mx is a representation of the fit, where m is the new multiplier value.

To perform a two-point multiplier only calibration, use an argument of 3 (the number three) for the calibration type in the `FieldCal()` instruction of your CRBasic program. The Calibration Wizard can be used to calculate and apply the two different known conditions while the program is running in the datalogger.

E.4 Performing a Manual Calibration

E.4.1 How to Use the Mode Variable for Calibration Status and Control

To perform a manual calibration (without use of the LoggerNet Calibration Wizard) on a `FieldCal()` enabled program running in a datalogger, it is necessary to understand the function of the mode variable that is used as an argument of the `FieldCal()` instruction.

In a CRBasic calibration program, a mode variable is declared and associated with a particular `FieldCal()` instruction, thereby associating it with the measurement to be calibrated.

Most values of the mode variable represent the status of the calibration for that affected measurement. A few values of the mode variable are set by the user of the datalogger to instruct the program to proceed with calibrations.

The following values of the mode variable give the status of the calibration:

0	No calibration has been performed since program start
2	Calibration in progress OR first stage of two-point calibration in progress

3	Waiting for second stage of two-point calibration to begin
5	Second stage of two-point calibration in progress
6	Calibration complete
-1	Calibration error
-2	Measurement range error
-3	Error with the value used as the Reps argument in the `FieldCal()` instruction.

The following values of the mode variable are used to initiate a calibration process:

1	Start the calibration, OR start the first point of a two point calibration
4	Start the second point of a two point calibration

> **NOTE:** For Zeroing and Offset calibration, a mode value of 4 is never used. The entire calibration process is initiated with the mode value being set to 1.

By properly changing the known value variables and the mode variables in a calibrating program, a manual calibration can be performed on a sensor. Steps for doing this are given below.

E.4.2 Using the Mode Variable for Manual Single-Point Calibration

These steps demonstrate how to perform manual single-point calibrations (Zeroing or Offset calibrations):

1. Ensure the status (value of the mode variable) is 0 or 6 before you start.
    a. A number greater than 0 that is not 6 indicates that a calibration is in progress or that the last calibration did not complete properly.
    b. A number less than 0 indicates that the calibration process encountered an error. Resolve the error before proceeding with the calibration then set the mode value to zero for a fresh start.
2. Place the sensor into the zeroing or offset condition.
3. Indicate the known offset value (if applicable) by changing the “known value” variable to that value.
4. Set the mode variable to 1 to initiate the calibration process.

5. Note that the datalogger automatically sets the mode variable to 2 during the calibration process.
6. Note that the mode variable is set to 6 automatically by the datalogger when the calibration process completes.

E.4.3 Using the Mode Variable for Manual Two-Point Calibration

These steps demonstrate how to perform manual two-point calibrations (Multiplier/Offset or Multiplier Only):

1. Ensure the status (value of the mode variable) is 0 or 6 before you start.
   a. A number greater than 0 that is not 6 indicates that a calibration is in progress or that the last calibration did not complete properly.
   b. A number less than 0 indicates that the calibration process encountered an error. Resolve the error before proceeding with the calibration then set the mode value to zero for a fresh start.
2. Place the sensor into the first known point condition.
3. Indicate the known value of the first point by changing the “known value” variable to that value.
4. Set the mode variable to 1 to initiate the first part of the calibration process.
5. Note that the datalogger automatically sets the mode variable to 2 during the first point calibration process.
6. Note that the mode variable is automatically set to 3 when the first point is completed.
   a. The datalogger is waiting for the user to place the system into the second point condition.
7. Place the sensor into the second known point condition.
8. Indicate the known value of the second point by changing the “known value” variable to that value.
9. Set the mode variable to 4 to initiate the second part of the calibration.
10. Note that the datalogger sets the mode variable to 5 during the second point calibration process.

11. Note that the mode variable is set to 6 by the datalogger when the calibration process completes successfully.

E.5 Using the Calibration Wizard with Running Programs

The LoggerNet Calibration Wizard provides an easy to use interface which steps the user through the calibration process described above. By using simple screens to gather information, the proper changes to the mode variable and known measurements are performed automatically by the Wizard. The user only needs to set the sensors to the known value(s), and provide a few required inputs to the Wizard. This greatly simplifies the user's interaction with the sensors and datalogger program.

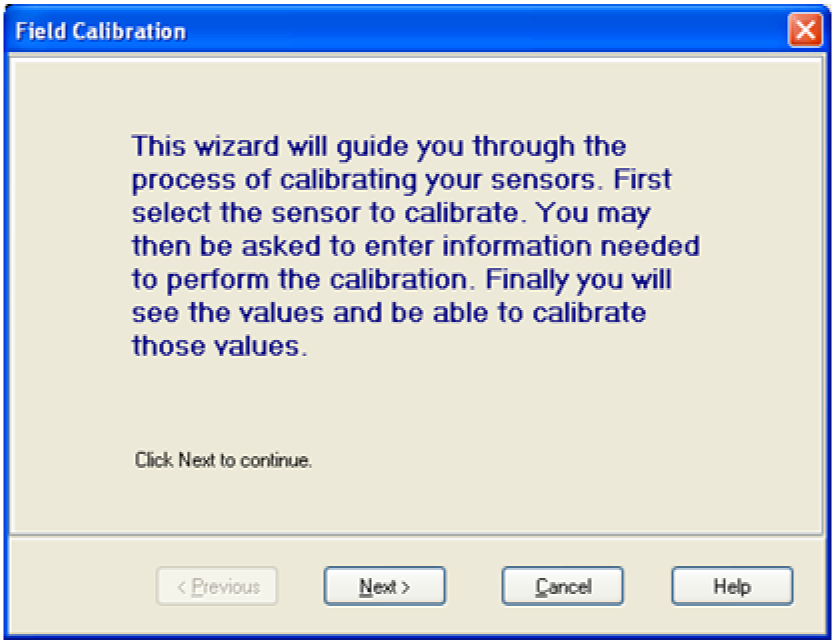
E.5.1 Calibration Wizard Basic Operation

When a program with one or more `FieldCal()` instructions is running on a datalogger, and LoggerNet is connected to that datalogger, you can select **Datalogger**> **Calibration Wizard** from the Connect Screen's menu to start the Wizard. A list of measurements referenced from one or more `FieldCal()` instructions used within the program is then displayed to the user. The user selects a measurement for calibration and moves forward by entering known values, if necessary, and triggering calibration steps.

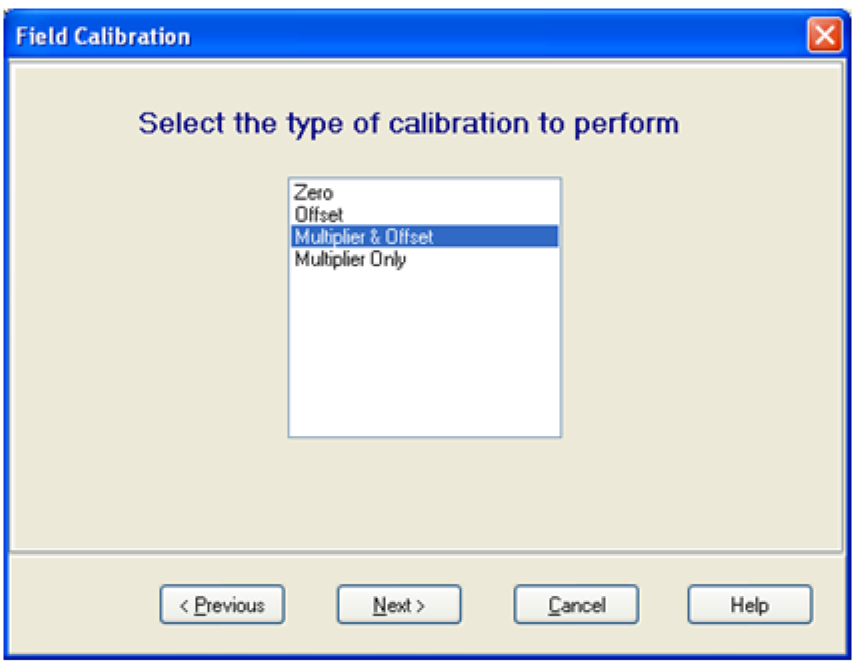
E.5.2 Using the Wizard to Perform Two-Point Multiplier and Offset Calibrations

To perform a two-point calibration using the Wizard, run a program in your datalogger utilizing a two-point multiplier/offset in the `FieldCal()` instruction. Connect to your datalogger and choose **Calibration Wizard** from the Connect Screen's **Datalogger** menu.

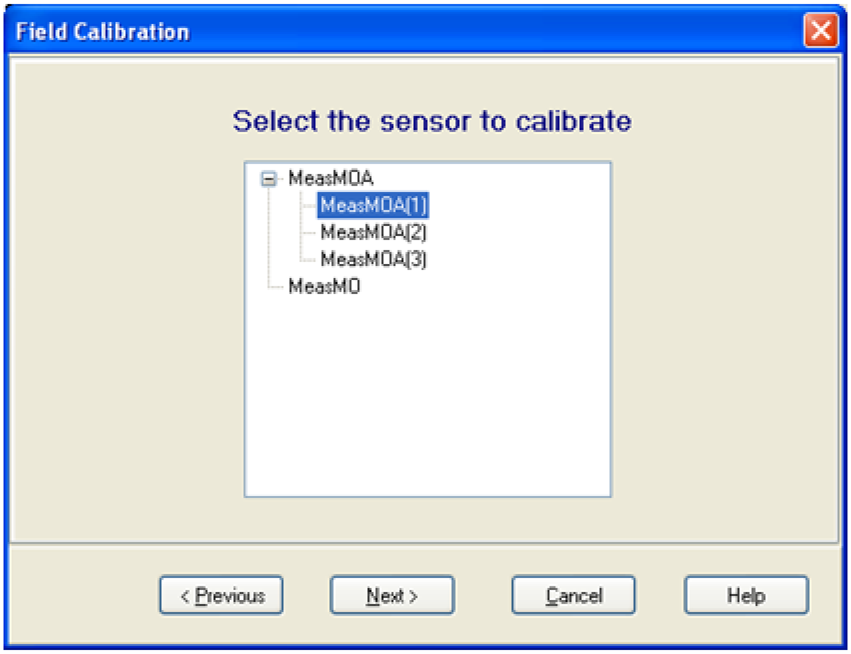
The Introduction screen for the Wizard will appear. Review the instructions and press **Next**.



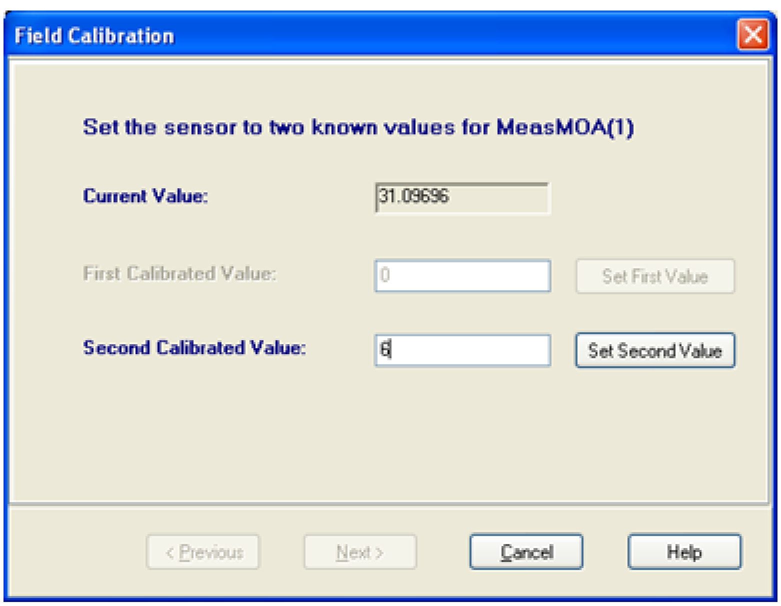
Now select the kind of calibration you wish to perform, which in this case is **Multiplier and Offset**, and press **Next**.



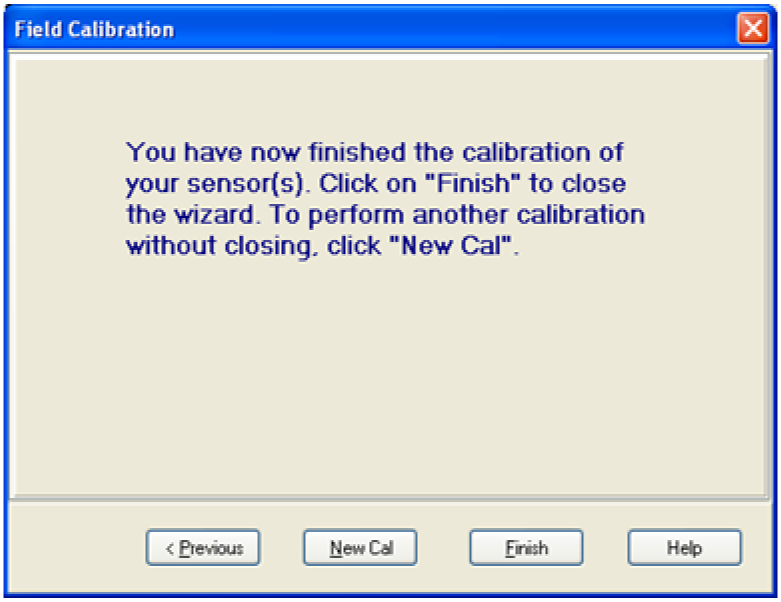
Now select which sensor it is that you wish to calibrate and press **Next**. You can select an entire array, or any single element of that array, as well as scalar (single-valued) variables. Any items that have been aliased (i.e., given an alternate name using the *Alias* instruction in the CRBasic program) will show by the alias name, including aliased elements of an array.



The currently measured value for the sensor will be displayed in the next screen. Now place the sensor into the first known condition, and enter that known value into the **First calibrated value** box. Press **Set First Value**. Wait for the calibration process to measure the first value. The word *Calibrating* will be visible in the **Current Value** box until that process is complete. Now place the sensor into the second known condition, and then enter the corresponding known value into the **Second calibrated value** box. Press **Set Second Value**. The calibration process measures the second point value. At that point the datalogger calculates the new multiplier and offset and applies them within the running program. These values are also written to the calibration file.



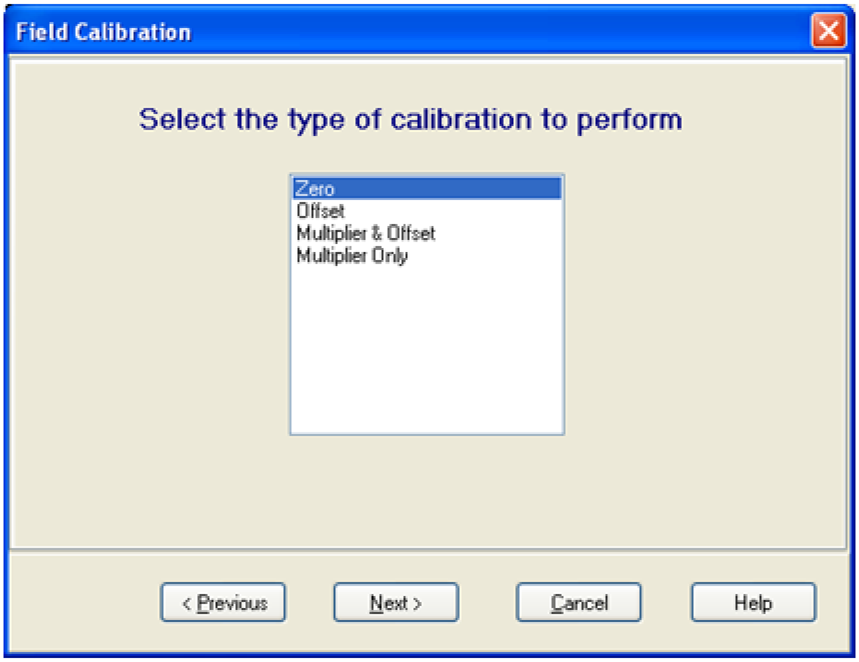
After the multiplier and offset have been calculated and set, the ending screen of the Wizard appears. You can conclude the calibration, or return to the starting point to perform more calibrations of the same or different sensors.



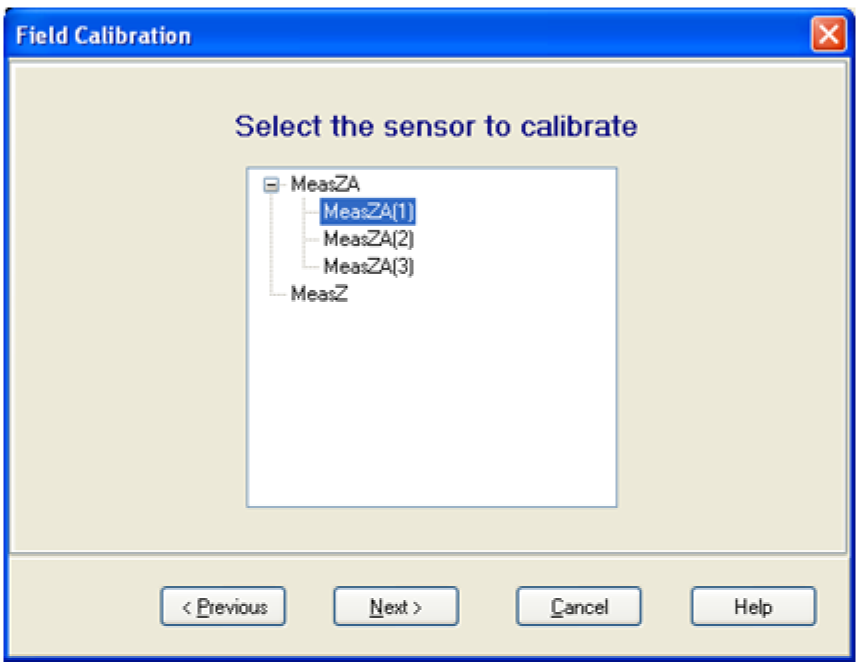
NOTE: The steps for performing a two-point slope only (multiplier only) calibration in the Wizard are nearly identical to those shown above for a two-point multiplier and offset calibration.

E.5.3 Using the Wizard to Perform Zeroing Calibration

In the Wizard, select **Zero** for the type of calibration.

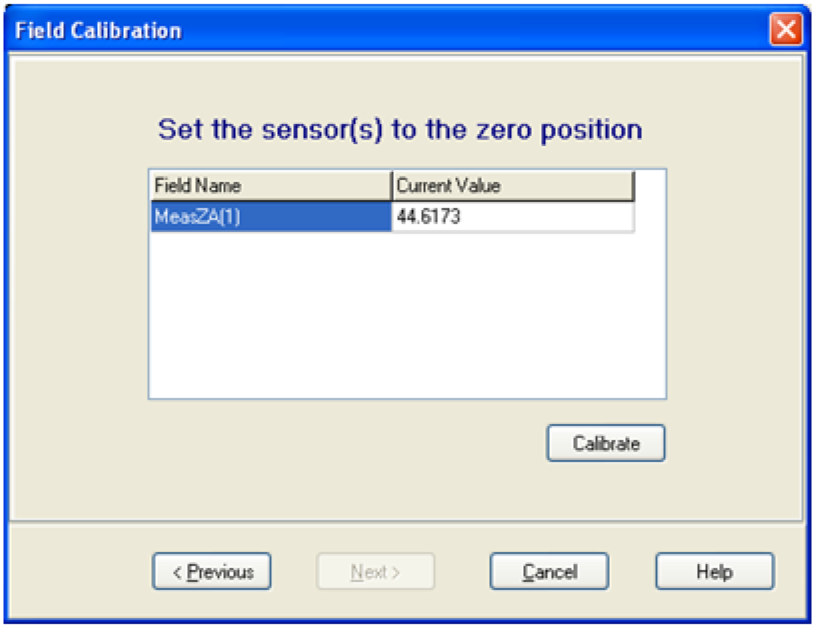


Select the sensor you wish to calibrate, press **Next**.



If the variable is configured as an array of sensors, you may select an individual item to calibrate only that element, or you can select the entire array.

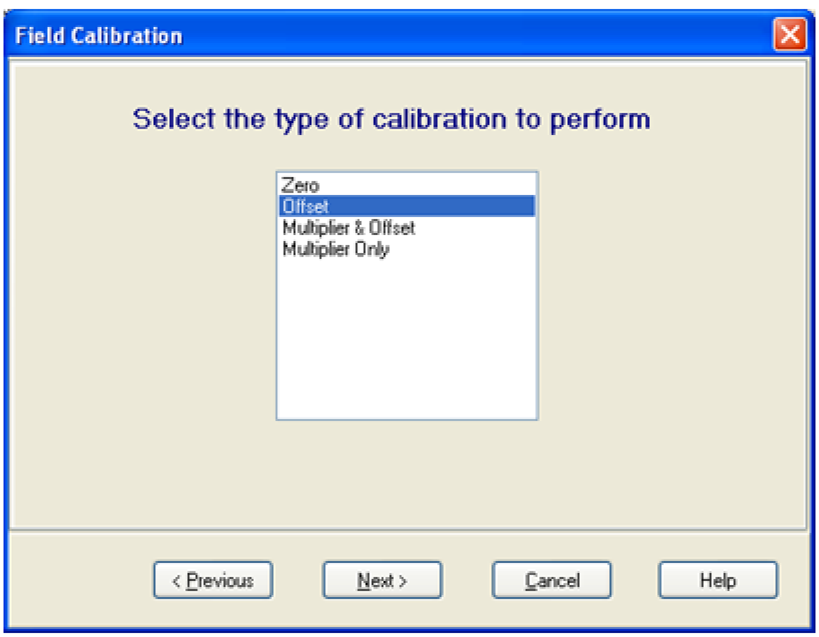
Now you can monitor the reading on the sensor to be calibrated. Set the sensor to the zero condition, and press **Calibrate**.



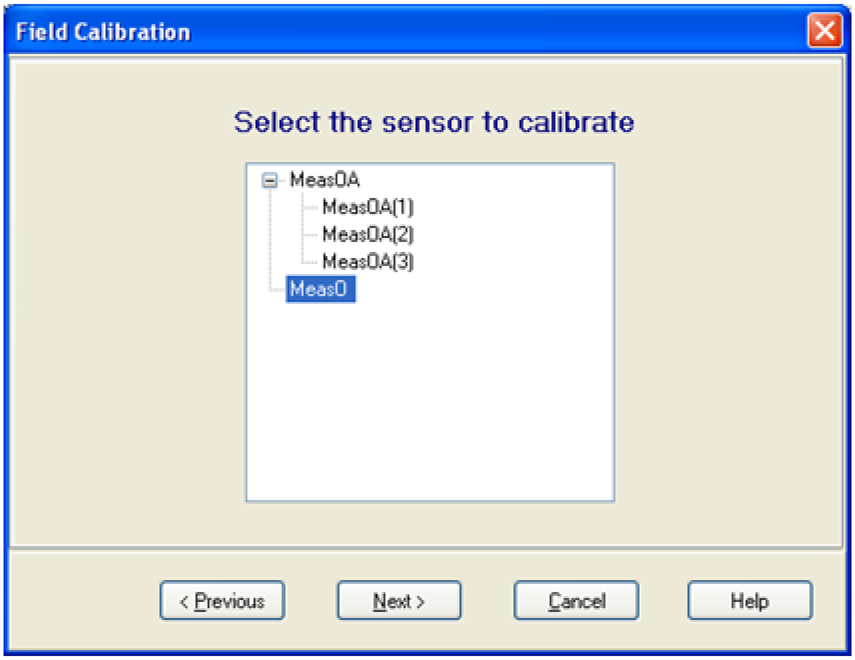
The **Current Value** box will be yellow during the calibration process. When it finishes, you will see the new value of the sensor after the application of the zeroing offset. Press **Finish** to end the calibration.

E.5.4 Using the Wizard to Perform Offset Calibration

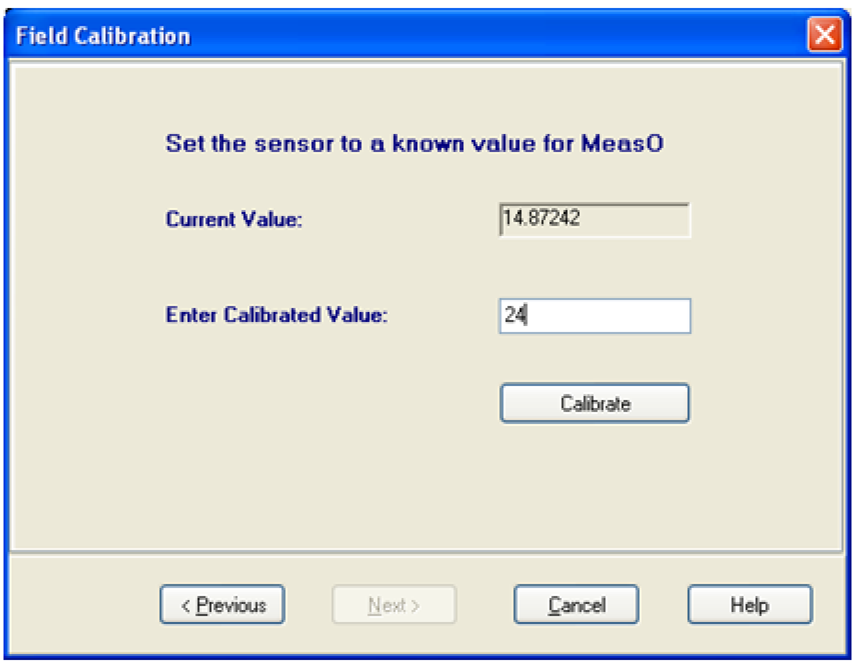
In the Wizard, select **Offset** for the type of calibration.



Select the sensor you wish to calibration, press **Next**.



Now you can view the current reading on the sensor to be calibrated. Set the sensor to the known value ("calibrate to" value). Enter that value into the **Enter Calibrated Value** box. Press **Calibrate**. The current value will show **Calibrating** until the process is complete. You will then have the opportunity to press **Finish**, or press **Previous** to return and calibrate more sensors.



E.6 Strain and Shunt Calibration

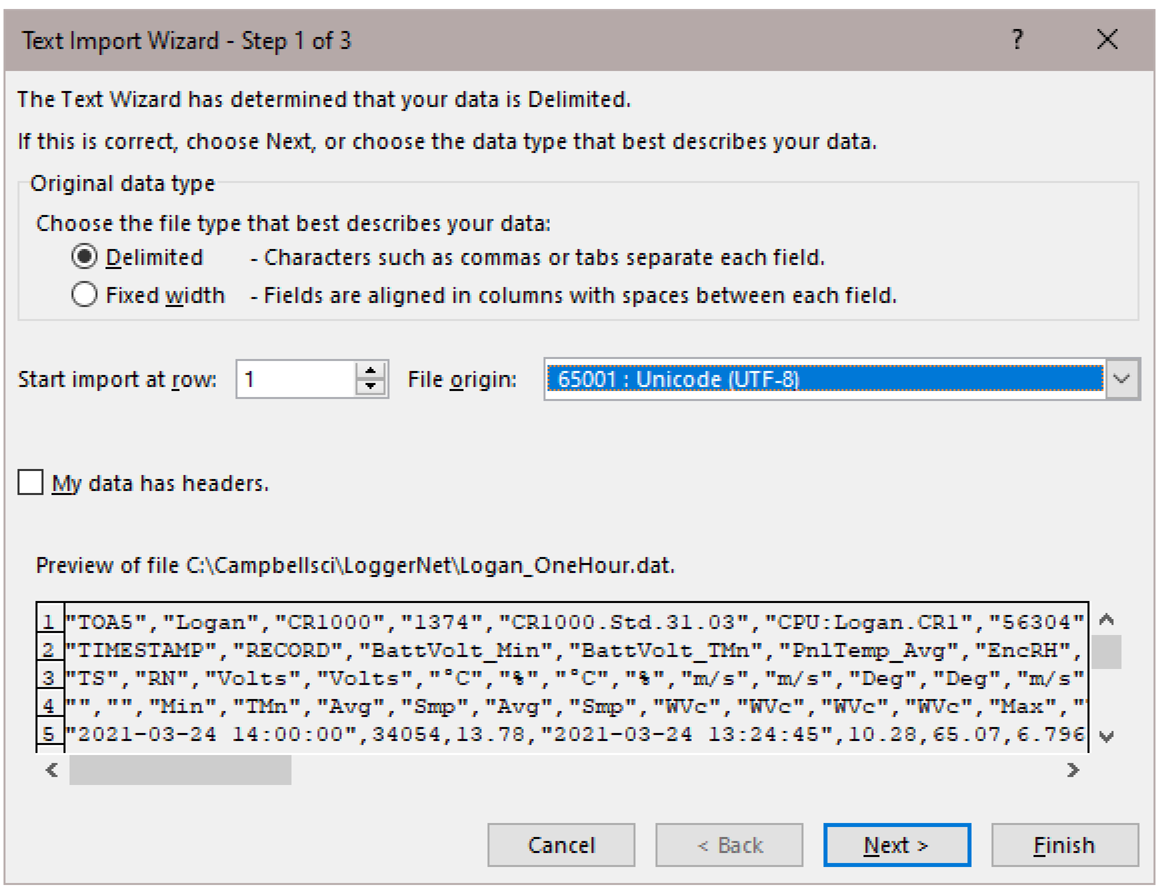
In addition to the `FieldCal()` instruction that performs calibrations on measurements, there is a specialized instruction for performing calibrations on strain bridge measurements (strain or shunt calibration). This instruction is called `FieldCalStrain()`. It functions in a similar manner to the `FieldCal()` instruction, but has additional arguments to meet the specialized needs for shunt and strain gauge calibrations. It uses the same calibration file (*.cal) as well as the other utility calibration instructions (`SampleFieldCal()`, `NewFieldCal()`, `LoadFieldCal()`). For more information about how to use these instructions, refer to the `FieldCalStrain()` instruction section of your datalogger manual, or refer to the online help topic for `FieldCalStrain()` within the CRBasic Editor. The Calibration Wizard also assists users through the process of calibrating sensors in a program using the `FieldCalStrain()` instruction. For more information, refer to the online Help topic provided within the Wizard.

Appendix F. Importing Files into Excel

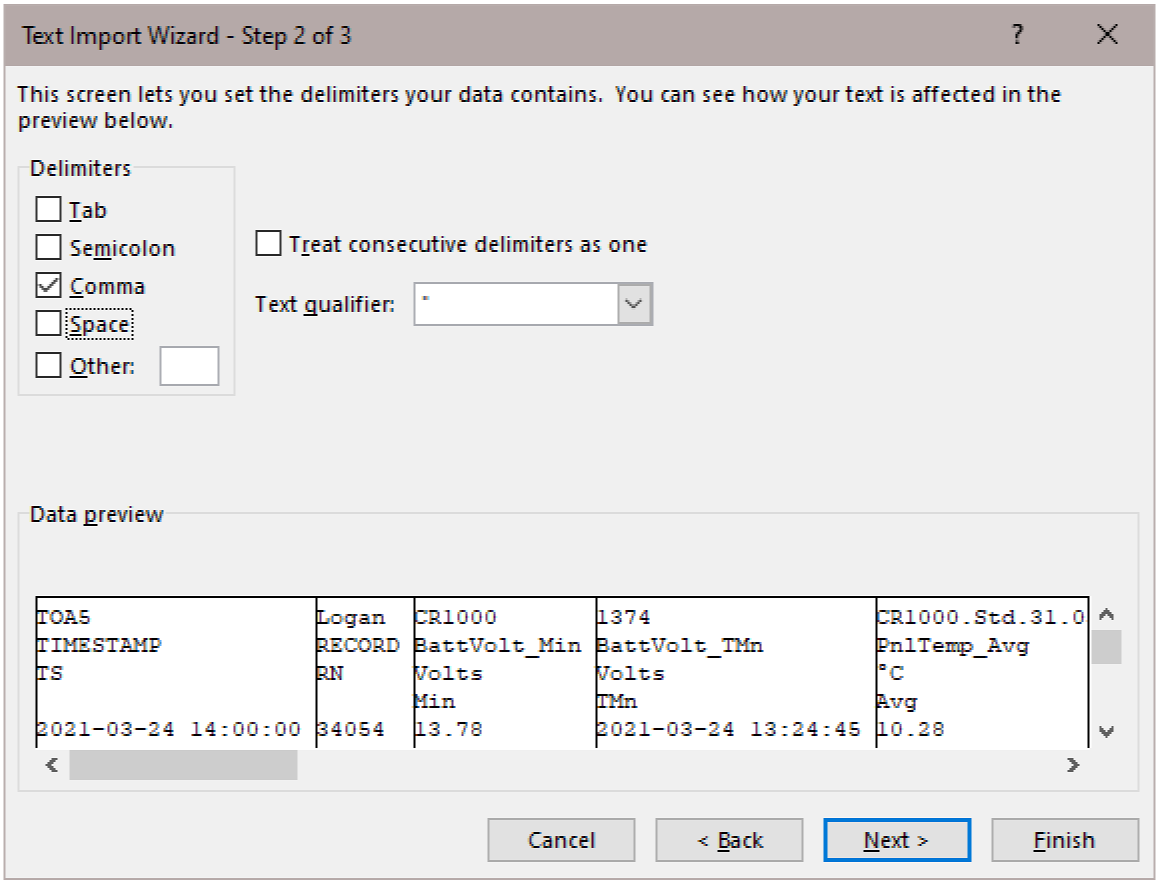
Data files saved by LoggerNet can be imported into a spreadsheet program for analysis or manipulation. Instructions are given below for importing a comma separated file into Microsoft Excel.

From the Excel menu, select **File** | **Open**. Browse for the *.dat file that you want to import. Excel will recognize the file as not being in an xls format, and will invoke the Text Import Wizard. The Text Import Wizard consists of three steps, each having its own window.

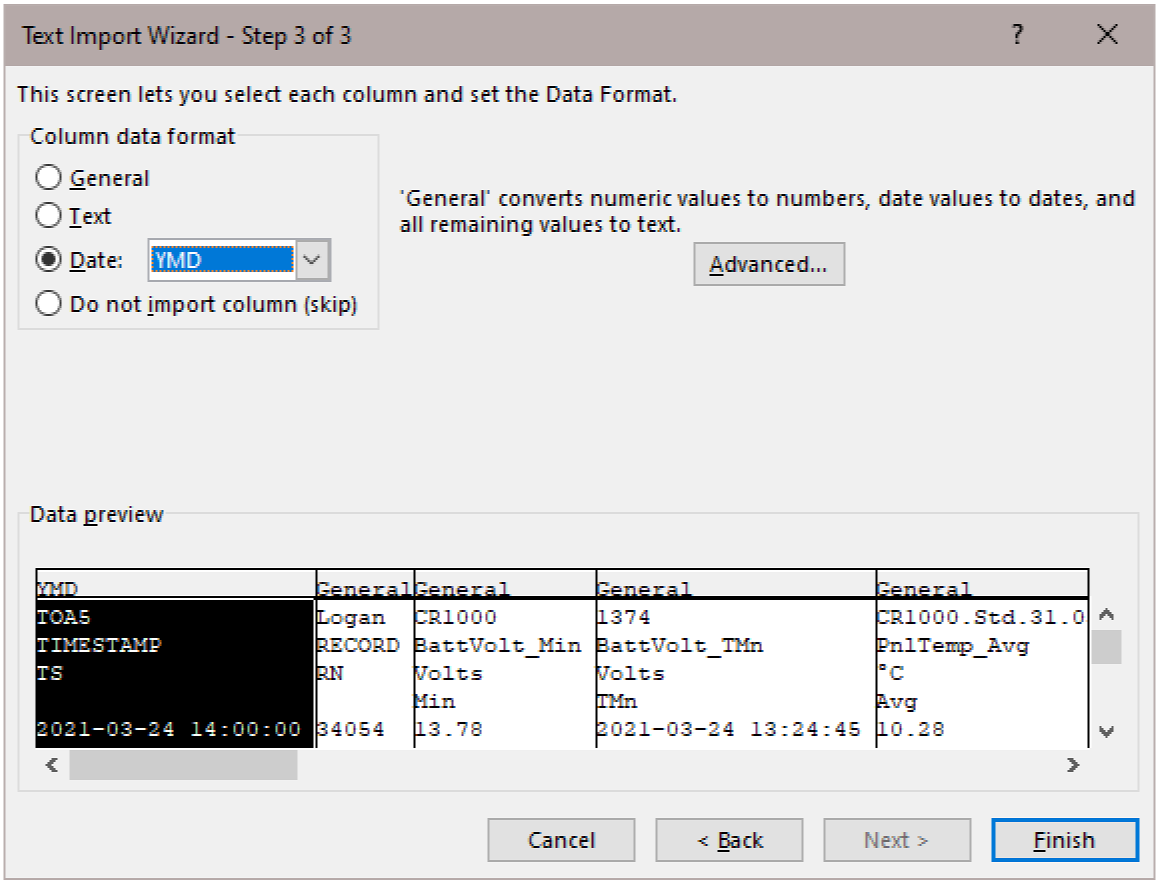
1. Select the **Delimited** option from the **Original Data Type** group box. Using the arrow buttons to the right of the **Start import at Row** field, select the number of the first row of data to be imported. If your data file has headers included, you can import those or start the import at the first row of data. Click **Next**.



2. From the **Delimiters** group box, select **Comma**. The Comma option directs Excel to place each data value, which is separated by a comma, into a separate column. Click **Next**.



3. A quick look at the columns of data is provided in the **Data preview** group box. Highlight the column with the year/month/day and from the **Column data format** group box, select the **Date** option. From the drop down list box to the right of this option select the **YMD** format. To complete the import, click **Finish**.


Text Import Wizard - Step 3 of 3
This screen lets you select each column and set the Data Format.
Column data format
General
Text
Date: YMD
Do not import column (skip)
'General' converts numeric values to numbers, date values to dates, and all remaining values to text.
Advanced...
Data preview
YMD General General General General
TOA5 Logan CR1000 1374 CR1000.Std.31.0
TIMESTAMP RECORD BattVolt_Min BattVolt_TMn PnlTemp_Avg
TS RN Volts Volts °C
Min TMn Avg
2021-03-24 14:00:00 34054 13.78 2021-03-24 13:24:45 10.28
Cancel
< Back
Next >
Finish

Appendix G. 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 (TOACI1 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.

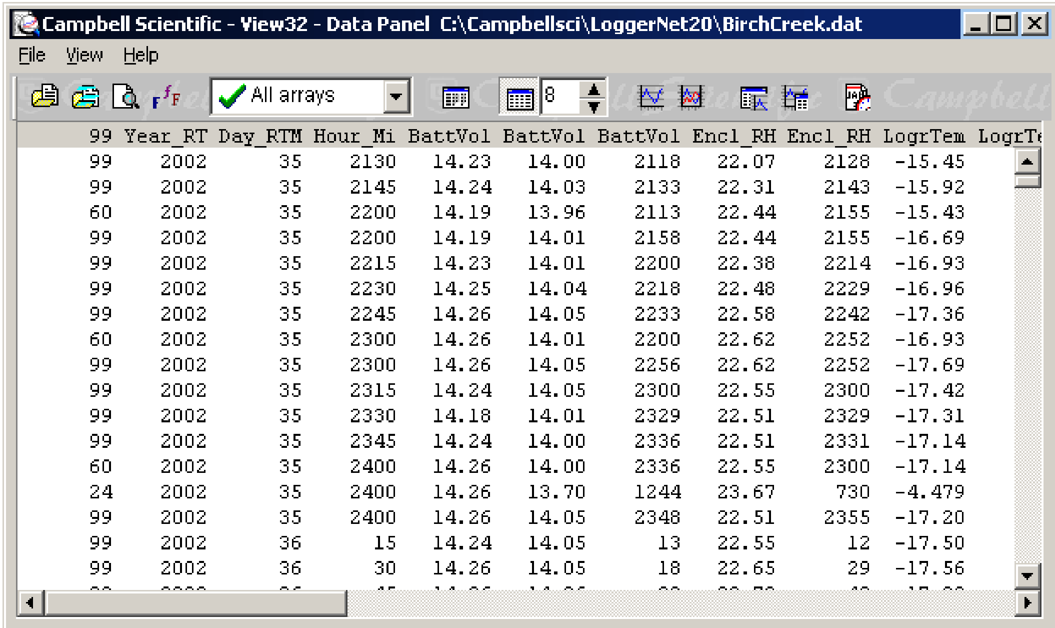
Split lends itself to experimentation. The processed data are displayed on the screen, giving immediate feedback as to the effect of changes or new entries to the parameter file. Split does not modify the original Input File.

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.

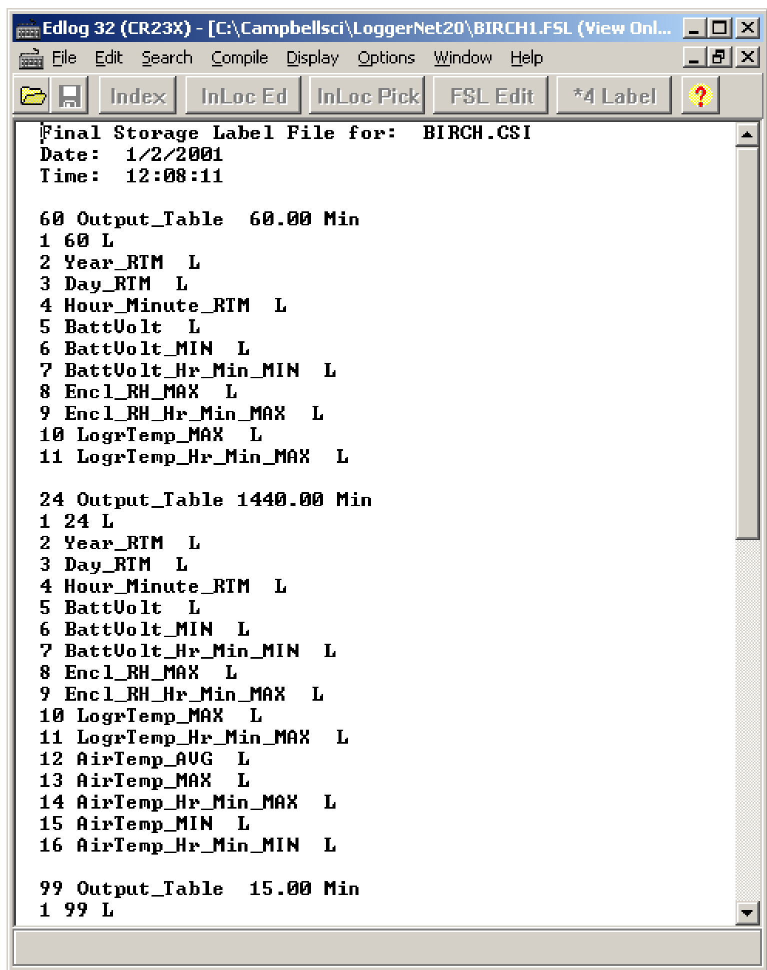

Campbell Scientific - View32 - Data Panel C:\Campbellsci\LoggerNet20\BirchCreek.dat

File View Help

All arrays

99	Year_RT	Day_RTM	Hour_Mi	BattVol	BattVol	BattVol	Encl_RH	Encl_RH	LogrTem	LogrTe
99	2002	35	2130	14.23	14.00	2118	22.07	2128	-15.45	
99	2002	35	2145	14.24	14.03	2133	22.31	2143	-15.92	
60	2002	35	2200	14.19	13.96	2113	22.44	2155	-15.43	
99	2002	35	2200	14.19	14.01	2158	22.44	2155	-16.69	
99	2002	35	2215	14.23	14.01	2200	22.38	2214	-16.93	
99	2002	35	2230	14.25	14.04	2218	22.48	2229	-16.96	
99	2002	35	2245	14.26	14.05	2233	22.58	2242	-17.36	
60	2002	35	2300	14.26	14.01	2200	22.62	2252	-16.93	
99	2002	35	2300	14.26	14.05	2256	22.62	2252	-17.69	
99	2002	35	2315	14.24	14.05	2300	22.55	2300	-17.42	
99	2002	35	2330	14.18	14.01	2329	22.51	2329	-17.31	
99	2002	35	2345	14.24	14.00	2336	22.51	2331	-17.14	
60	2002	35	2400	14.26	14.00	2336	22.55	2300	-17.14	
24	2002	35	2400	14.26	13.70	1244	23.67	730	-4.479	
99	2002	35	2400	14.26	14.05	2348	22.51	2355	-17.20	
99	2002	36	15	14.24	14.05	13	22.55	12	-17.50	
99	2002	36	30	14.26	14.05	18	22.65	29	-17.56	

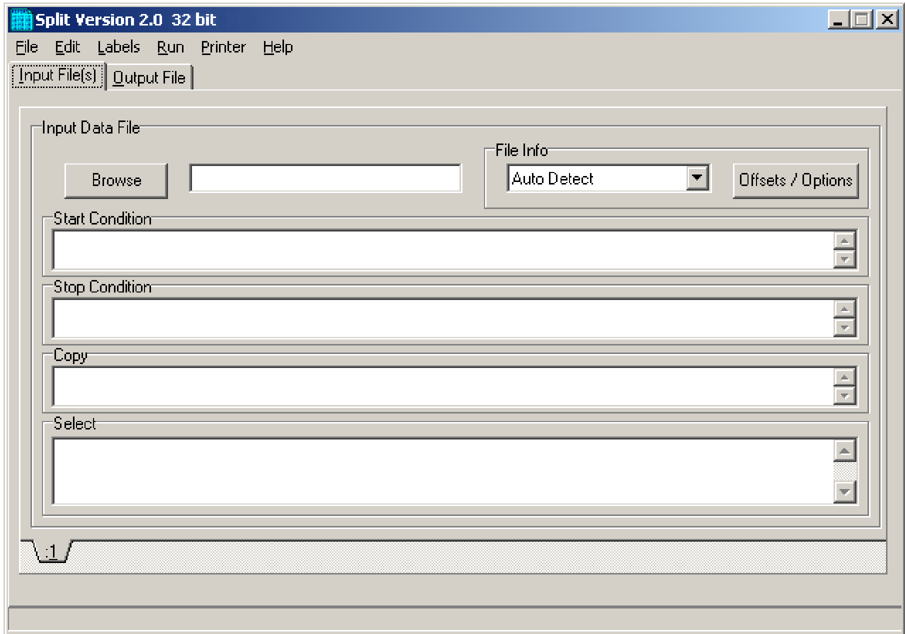
When Edlog compiled Birch.dld, it also created the Final Storage Label file, Birch.fsl that lists the final storage locations for each data element.


Edlog 32 (CR23X) - [C:\Campbellsci\LoggerNet20\BIRCH1.FSL (View Onl...

File Edit Search Compile Display Options Window Help

Index InLoc Ed InLoc Pick FSL Edit *4 Label

```
Final Storage Label File for:  BIRCH.CSI
Date:  1/2/2001
Time:  12:08:11

60 Output_Table  60.00 Min
1 60 L
2 Year_RTM  L
3 Day_RTM  L
4 Hour_Minute_RTM  L
5 BattVolt  L
6 BattVolt_MIN  L
7 BattVolt_Hr_Min_MIN  L
8 Encl_RH_MAX  L
9 Encl_RH_Hr_Min_MAX  L
10 LogrTemp_MAX  L
11 LogrTemp_Hr_Min_MAX  L

24 Output_Table 1440.00 Min
1 24 L
2 Year_RTM  L
3 Day_RTM  L
4 Hour_Minute_RTM  L
5 BattVolt  L
6 BattVolt_MIN  L
7 BattVolt_Hr_Min_MIN  L
8 Encl_RH_MAX  L
9 Encl_RH_Hr_Min_MAX  L
10 LogrTemp_MAX  L
11 LogrTemp_Hr_Min_MAX  L
12 AirTemp_AVG  L
13 AirTemp_MAX  L
14 AirTemp_Hr_Min_MAX  L
15 AirTemp_MIN  L
16 AirTemp_Hr_Min_MIN  L

99 Output_Table  15.00 Min
1 99 L
```



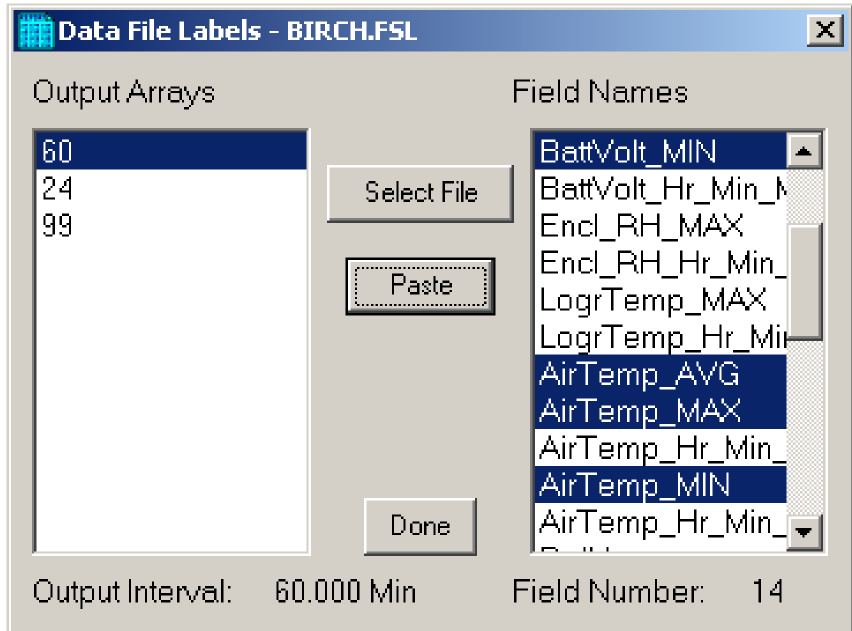
When you start Split a blank template similar to the one above is shown. This template is used to enter the parameters that will define what data from the input file to include in the output file. The parameters entered on this template can be saved as a parameter file (*.PAR) and reused for other data.

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.

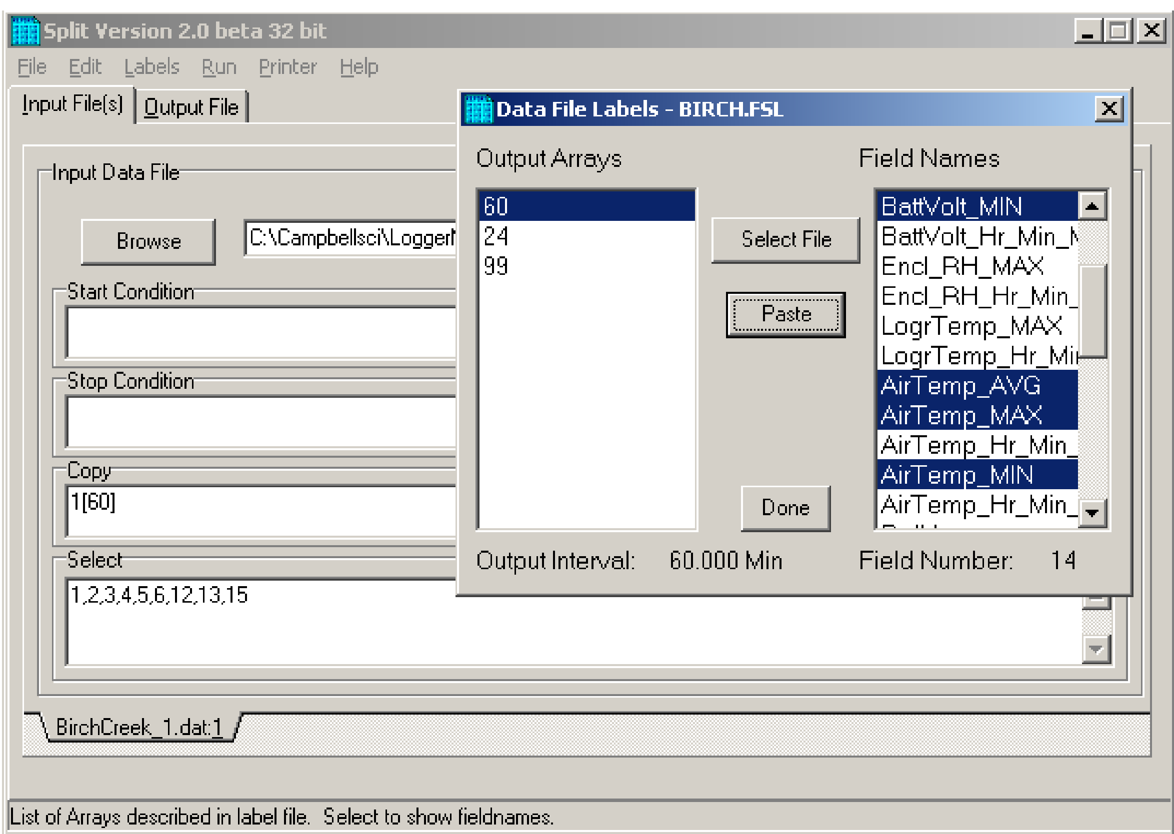
The name of the Input Data File can be typed in or the **Browse** button can be used to select from available files. In this example BirchCreek.dat will be selected as the input data file.

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.

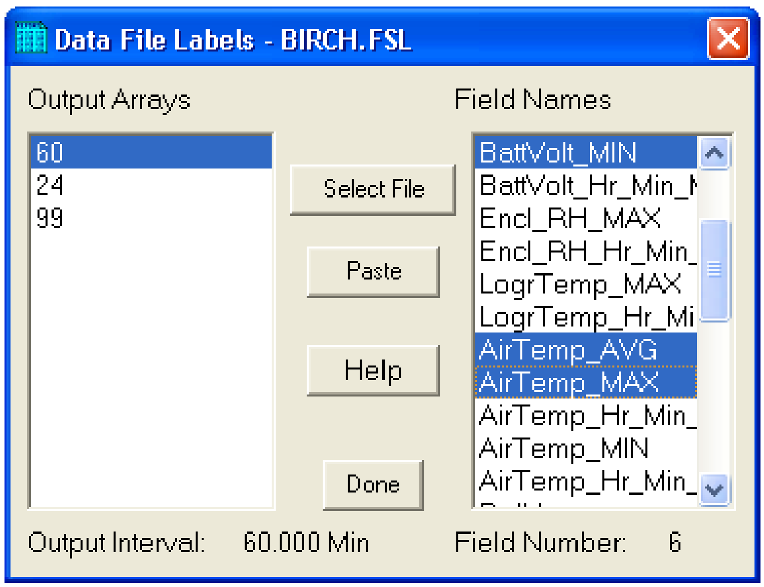


Note that the cursor in the **INPUT FILE(S)** screen must be in valid paste area (Copy or Select). If the cursor is in the File name box or in Start/Stop condition, you will get the error message “Cannot Paste There”.

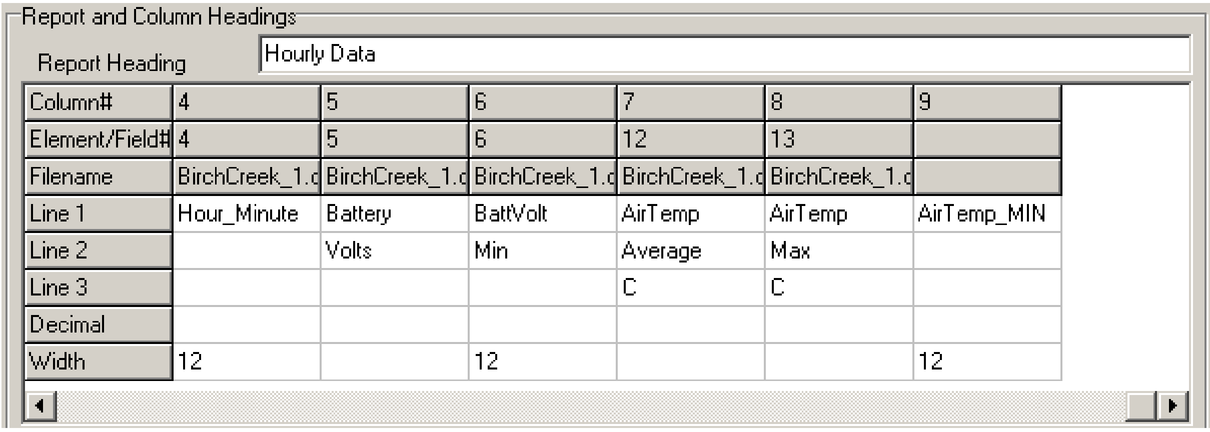
The Paste operation copied the numbers of each of the fields into the Select box. Notice also that it pasted the Array ID into the copy condition: 1[60] tells Split that in order to copy a line of data, the first value in that line must be 60. Split uses the Array ID to discriminate between the hourly 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

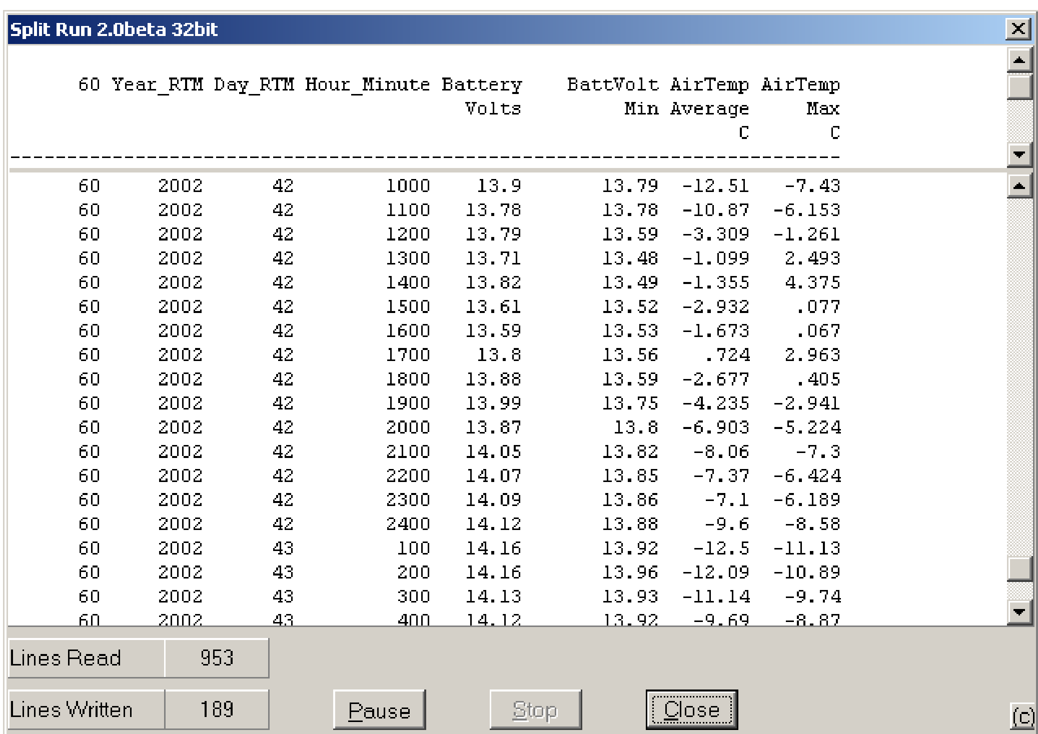
Column#	4	5	6	7	8	9
Element/Field#	4	5	6	12	13	
Filename	BirchCreek_1.d	BirchCreek_1.d	BirchCreek_1.d	BirchCreek_1.d	BirchCreek_1.d	
Line 1	Hour_Minute	Battery	BattVolt	AirTemp	AirTemp	AirTemp_MIN
Line 2		Volts	Min	Average	Max	
Line 3				C	C	
Decimal						
Width	12		12			12

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 Volts	BattVolt Min	AirTemp Average C	AirTemp Max 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 (c)

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 (TOACI1 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

```
"TOACI1","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

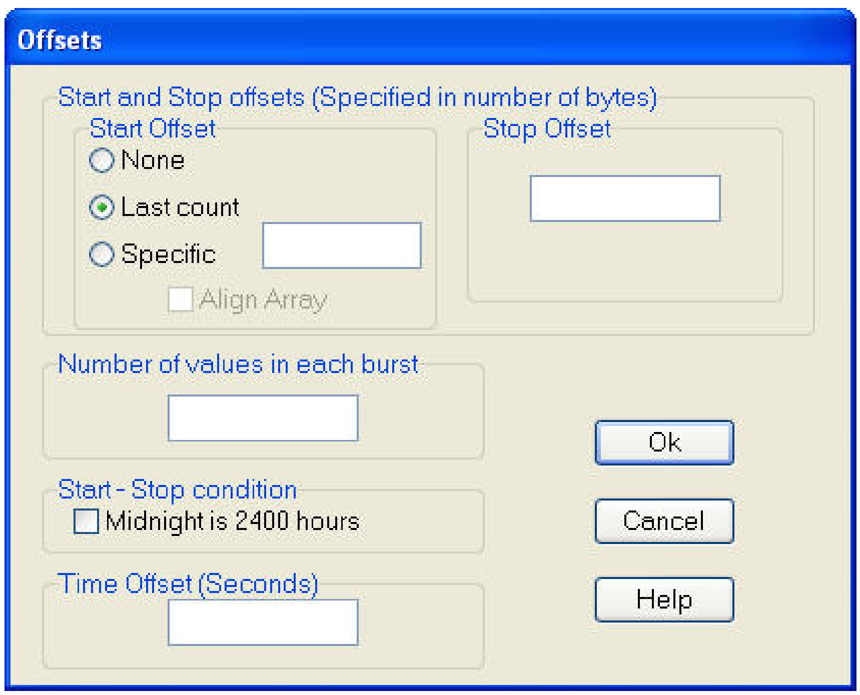
Start Offset

None

Select his 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.



CAUTION

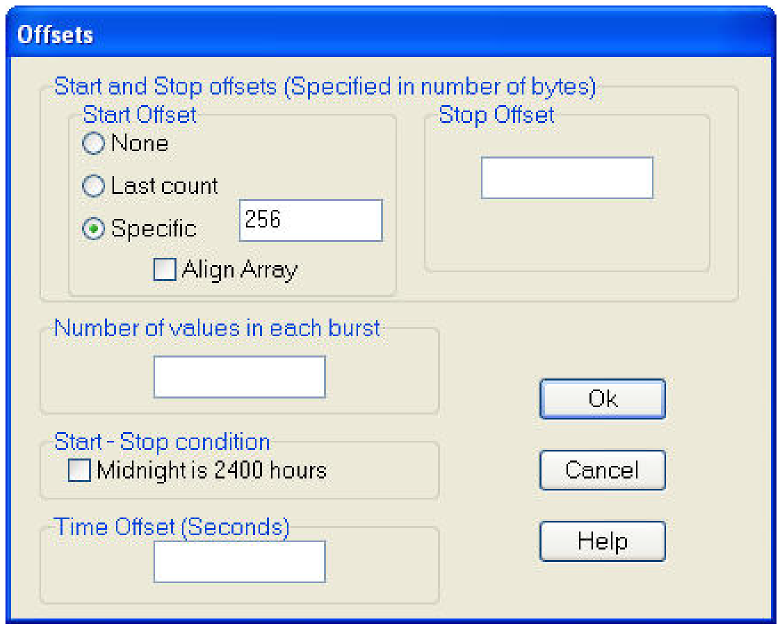
When using the Last Count option, if the Start and Stop Conditions are specified, they must exist in the newly appended data or Split will never begin execution.

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.



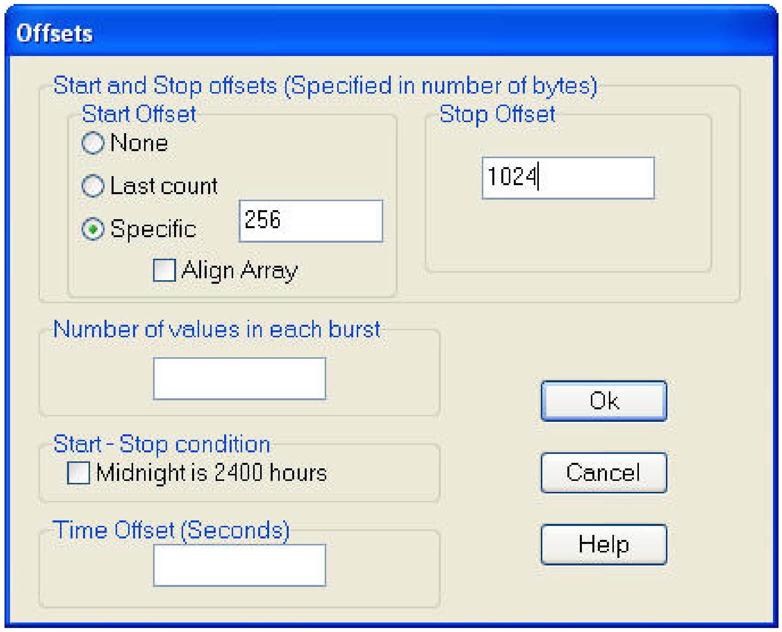
Align Array

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.



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;1) would be "2008-10-09 11:25".

This may 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_i 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, hrmn, 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	Data from arrays output during the first test.
200	
100,12.5,9.89,30.1 102,56.2,50.,100.5,210.6	Second test.
200	
100,13.1,10.1,33.1 101,94.1,69	Third test.
200	

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_i[val_i].

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			
Output Option	**Screen Display***	**PRN File**	**RPT File or Printer Output**
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=6/3,2,3,6,7,7∗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

TABLE 6. Split Operators and Math Functions

OPERATORS	**OPERATOR PRECEDENCE ORDER** (3 = high, 1 = low)
^ = raise to the power x Mod y = Modulo divide of x by y ∗ / = multiplication, division + – = addition, subtraction	3 2 2 1
EXAMPLES OF SYNTAX FOR MATHEMATICAL OPERATORS	
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
MATH FUNCTIONS	
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^x)
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

Abs(x) returns the absolute, or positive value of element x.
Examples:
Abs(4) = 7.89
Abs(4∗5) = 434.74

Arctan(x) returns the arc tangent of element x in degrees.
Examples:
Arctan(7) = 89.788
Arctan(7/6) = 87.365

Cos(x) returns the cosine of element x in degrees.
Examples:
Cos(5) = .57215
Cos(5–6) = .73551

Exp(x) returns the exponential base e to the power of element x.
Example:
Exp(4) = .00037

Frac(x) returns the fractional value of the element x.
Examples:
Frac(4) = –.89
Frac(6+7) = .95

Int(x) returns the integer portion of the element x.
Examples:
Int(7) = 270
Int(5∗6) = 685

Ln(x) returns the natural log of element x.
Examples:
Ln(6) = 2.5217
Ln(7/6∗5/1) = 2.4337

Sin(x) 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:
SpaAvg(1..7) = 258.74
SpaAvg(1,4,7) = 122.54

SpaMax(x..y) returns the maximum value of elements x through y.
Examples:
SpaMax(1..7) = 1200
SpaMax(1,2,5) = 176

SpaMin(x..y) returns the minimum value of elements x through y.
Examples:
SpaMin(1..7) = –7.89
SpaMin(1,2,5) = 55.1

SpaSd(x..y) returns the standard deviation of elements x through y.
Examples:
SpaSd(1..7) = 394.57
SpaSd(5,2,1) = 49.607

Sqrt(x) returns the square root of element x.
Examples:
Sqrt(3) = 34.641
Sqrt(3^ 2.0) = 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)
NOTE: 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 24th 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

Day	**Time**	**Airtemp deg F**	**RH**	**Mean Wind Speed mph**	**Mean Wind Direction**	**Std Dev of Direction**
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^{th} 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^{th} 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^{th} 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^{th} 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^{th} value is specified, a running total will be output every n^{th} 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^{th} 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^{th} 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^{th} 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^{th} value is specified then it outputs the sample on a maximum every n^{th} 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^{th} value is specified then it outputs the sample on a minimum every n^{th} 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^{th} 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	
Crlf	= Insert carriage return line feed in Output File.
Date(“format”S;H;D;Y)	= 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.
Edate(“format”S;H;D;Y)	= The same as the Date function except that the output text is not quoted. EDate can be used to create monthly time series summaries.
“Label”	= Insert Comment in Output file. (Label is anything within the quote marks.)
Line	= Number of lines written to Output file.
smpl(.pa;n)	= Page break such that n is the number of lines per page for the printer or the .RPT file.
PCdate or PCEdate	= Used in a report header to print the current date.
WDQ(n)	= Outputs the wind direction using an alphabetical abbreviation, based on 8 quadrants.
WDQS(n)	= 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.

String Entered	Output
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

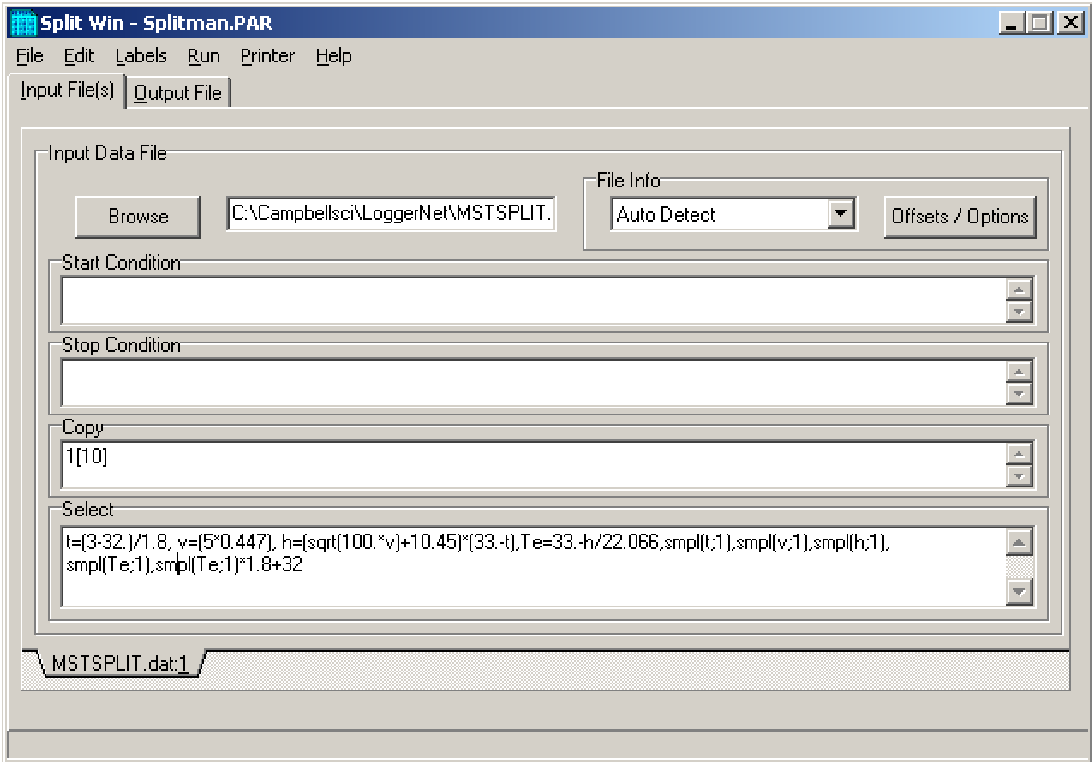
The Date function can be used to produce a monthly summary of daily time series data by using Date( ) for the interval in the time series function. This will trigger time series output for the first day of each month. The syntax is avg(7;date(3;2)), where you want to take a monthly average of element 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)).

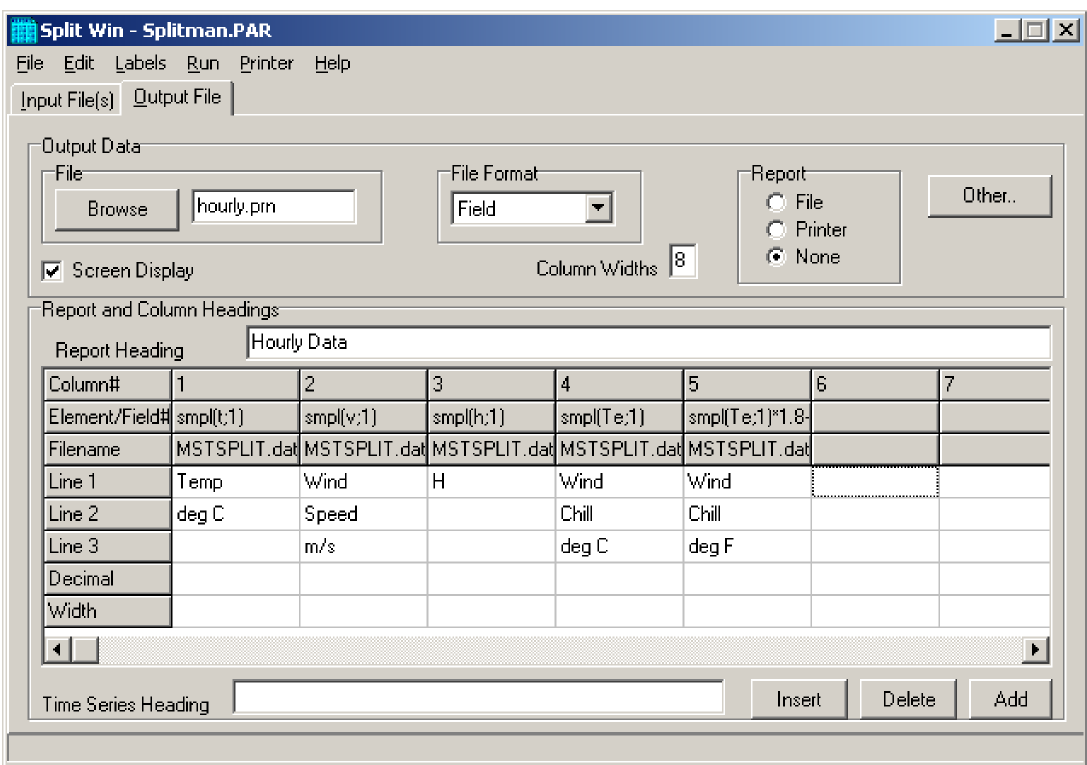
When producing a monthly summary and outputting the month along with the data, you might want to set up the value for the month as "month –1", to correctly reflect the month that the data actually represents.

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.



The following screen shows the output file setup including the column headings and the units.



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:

$$Te = 33-(h/22.066)$$

where

Te = 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).[1] 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

[1] "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 x 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	**INVALID EXPRESSION**
Arctan (2/3)	Arctan ((2/3))
Arctan (2/3, 3/4, 4/5)	Arctan ((2/3, 3/4), 4/5)
Arctan (COS(2))	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:

$$e_i[day]:e_i[hrmn]:e_i[seconds]$$

Referring to TABLE 1, to identify the day of year for a mixed-array data file, type:

2[189]::

for hrmn type:

:3[1200]:

and seconds are expressed as:

::4[5]

A single colon is assumed to be between day and hrmn (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 hrmn 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 hrmn 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.

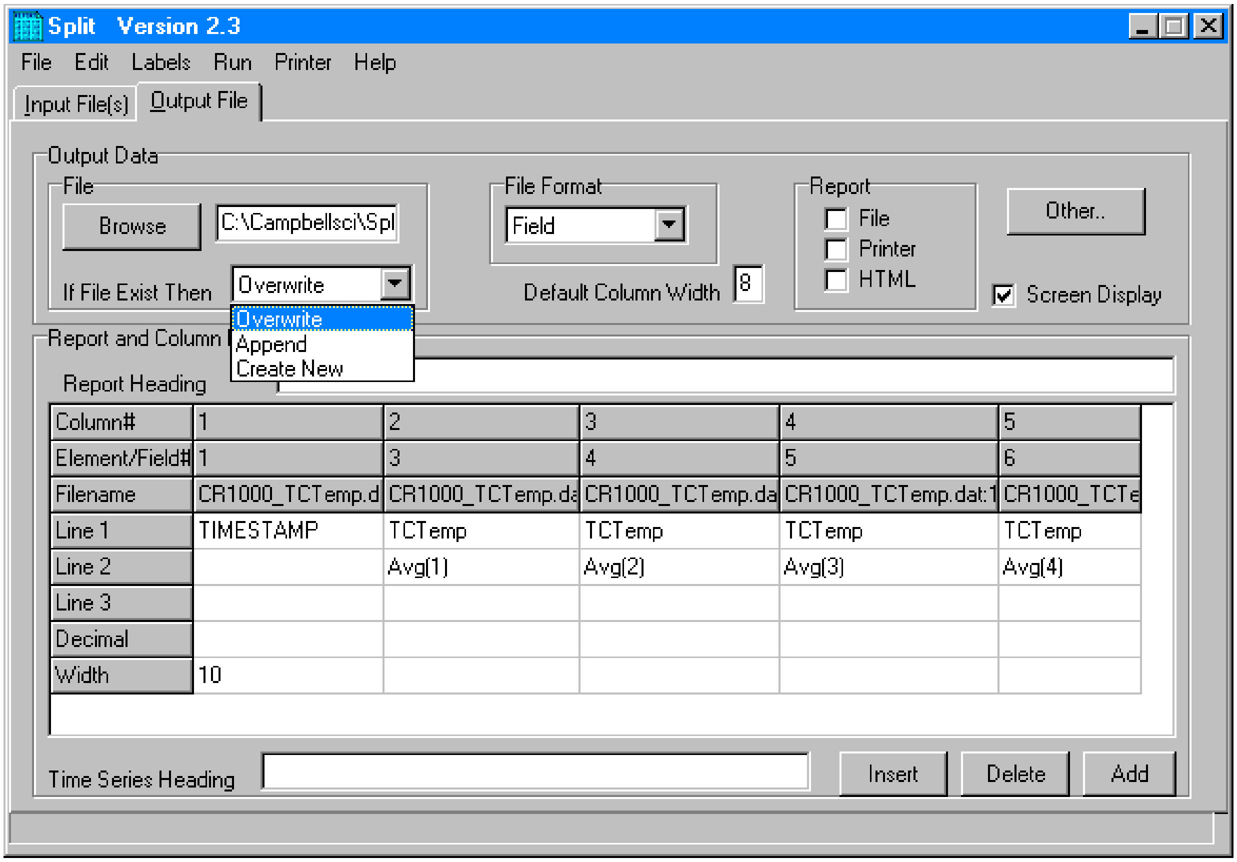
Split will assign this file an extension of .PRN if an extension is not specified by the user. Whenever an Output file name is entered, regardless of extension, an Output file is created only when the **RUN | GO** menu option is selected.

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.

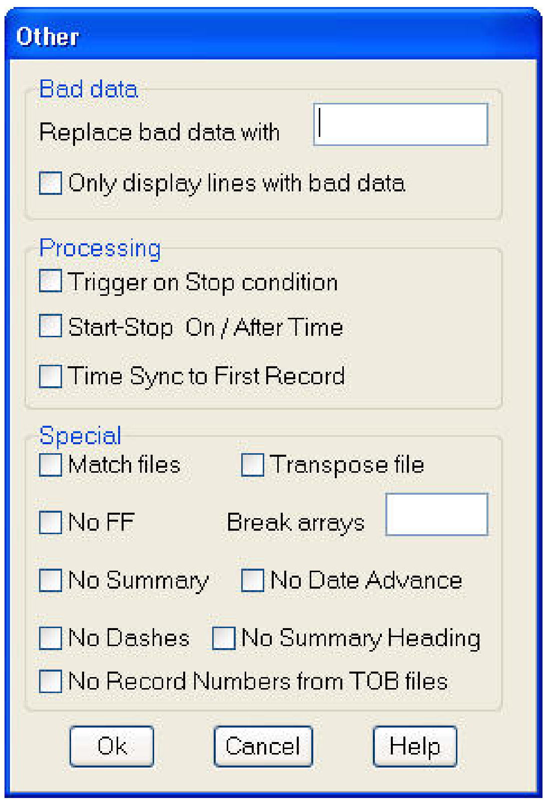
Screen Display
The Screen Display field controls writing the processed data to the screen. To write to the screen, check the box. For faster execution, clear the box to omit writing to screen. The data will then be written to the file only.

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.



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 9th 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 1st to Sept 9th. 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 **Output File** tab.

/R Creates a formatted RPT file. This is the same as checking the File box for the Report type on the **Output File** tab.

/W Creates a simple HTML file. This is the same as checking the HTML box for the Report type on the **Output File** 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 **Output File** 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 **Output File** tab.

/O Turns the screen display off when Split is processing the PAR file. This is the same as clearing the **Screen Display** check box on the **Output File** 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 **Output File** 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 **Other** button of the **Output File** tab.

/M Compares two input files and creates an output file with a complete data set comprised of both files. This is the same at 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.

/0 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
Website: www.campbellsci.com.au

Brazil

Location: São Paulo, SP Brazil
Phone: 11.3732.3399
Email: vendas@campbellsci.com.br
Website: www.campbellsci.com.br

Canada

Location: Edmonton, AB Canada
Phone: 780.454.2505
Email: dataloggers@campbellsci.ca
Website: www.campbellsci.ca

China

Location: Beijing, P. R. China
Phone: 86.10.6561.0080
Email: info@campbellsci.com.cn
Website: www.campbellsci.com.cn

Costa Rica

Location: San Pedro, Costa Rica
Phone: 506.2280.1564
Email: info@campbellsci.cc
Website: www.campbellsci.cc

France

Location: Montrouge, France
Phone: 0033.0.1.56.45.15.20
Email: info@campbellsci.fr
Website: www.campbellsci.fr

Germany

Location: Bremen, Germany
Phone: 49.0.421.460974.0
Email: info@campbellsci.de
Website: www.campbellsci.de

India

Location: New Delhi, DL India
Phone: 91.11.46500481.482
Email: info@campbellsci.in
Website: www.campbellsci.in

Japan

Location: Kawagishi, Toda City, Japan
Phone: 048.400.5001
Email: jp-info@campbellsci.com
Website: www.campbellsci.co.jp

South Africa

Location: Stellenbosch, South Africa
Phone: 27.21.8809960
Email: sales@campbellsci.co.za
Website: www.campbellsci.co.za

Spain

Location: Barcelona, Spain
Phone: 34.93.2323938
Email: info@campbellsci.es
Website: www.campbellsci.es

Thailand

Location: Bangkok, Thailand
Phone: 66.2.719.3399
Email: info@campbellsci.asia
Website: www.campbellsci.asia

UK

Location: Shepshed, Loughborough, UK
Phone: 44.0.1509.601141
Email: sales@campbellsci.co.uk
Website: www.campbellsci.co.uk

USA

Location: Logan, UT USA
Phone: 435.227.9120
Email: info@campbellsci.com
Website: www.campbellsci.com