

OPTOTUTORIAL: FILE MANAGEMENT AND FTP

A Supplement to the SNAP PAC Learning Center

Form 1642-060914—September 2006

OPTO 22

43044 Business Park Drive • Temecula • CA 92590-3614

Phone: 800-321-OPTO (6786) or 951-695-3000

Fax: 800-832-OPTO (6786) or 951-695-2712

www.opto22.com

Product Support Services

800-TEK-OPTO (835-6786) or 951-695-3080

Fax: 951-695-3017

Email: support@opto22.com

Web: support.opto22.com

OptoTutorial: File Management and FTP
Form 1642-060914—September 2006

Copyright © 2006 Opto 22.
All rights reserved.
Printed in the United States of America.

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or later are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. These products include, but are not limited to, OptoTerminal-G70, OptoTerminal-G75, and Sony Ericsson GT-48; see the product data sheet for specific warranty information. Refer to Opto 22 form number 1042 for complete warranty information.

Cyrano, Opto 22 FactoryFloor, Optomux, and Pamux are registered trademarks of Opto 22. Generation 4, ioControl, ioDisplay, ioManager, ioProject, ioUtilities, mistic, Nvio, Nvio.net Web Portal, OptoConnect, OptoControl, OptoDisplay, OptoENETSniff, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC, SNAP Simple I/O, SNAP Ultimate I/O, and SNAP Wireless LAN I/O are trademarks of Opto 22.

ActiveX, JScript, Microsoft, MS-DOS, VBScript, Visual Basic, Visual C++, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Unicenter is a registered trademark of Computer Associates International, Inc. ARCNET is a registered trademark of Datapoint Corporation. Modbus is a registered trademark of Schneider Electric. Wiegand is a registered trademark of Sensor Engineering Corporation. Nokia, Nokia M2M Platform, Nokia M2M Gateway Software, and Nokia 31 GSM Connectivity Terminal are trademarks or registered trademarks of Nokia Corporation. Sony is a trademark of Sony Corporation. Ericsson is a trademark of Telefonaktiebolaget LM Ericsson.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Table of Contents

OPTO 22

Getting Started	1
Introduction to File Management and FTP	1
What is file management?	1
Understanding file management on the SNAP PAC	3
FTP	5
SNAP PAC as an FTP server	5
SNAP PAC as an FTP client	6
File management and FTP	6
Lesson Topics	7
Using This Tutorial	8
What You Need	9
Instructional approach	9
Sample files	9
FTP server	9
Installing an FTP server	9
Setting up FTP client accounts	11
 Creating and Downloading a Log File	 15
Lesson Overview	15
Concepts	16
File communication handle	16
Typical use of a communication handle	17
Setting the value of a Communication Handle	18
Size of data stored in persistent files	19
Table variables	21
LogData chart	22
Activity 1a: Creating a Log File	23

Preparation	23
Add strategy variables	25
Chart: LogData	31
Log data	41
Activity 1b: View the Data Log	41
Assign an FTP log in and password	41
An FTP session	44
Using Internet Explorer to open a file on the SNAP PAC	44
Using the Windows XP FTP client	47
Using Excel	49

Sending Files to SNAP PACs 53

Lesson 2 Overview	53
Concepts	54
How the sample chart works	54
Parsing a text file	55
Communication Handles for reading files	56
Activity 2a: Uploading and Parsing Files	56
Strategy tree	57
Chart: LEDs from recipe	59
Activity 2b: Create and Upload a File	67
Test your strategy	68

Sending Data to an FTP Server 69

Lesson 3 Overview	69
Sending data to an FTP server	70
Communication handle for an FTP server	71
Gateway	72
Activity 3a: Sending Data to an FTP Server	72
What you need	72
Create communication handle	73
Modify chart to send file to FTP server	74
Test the strategy	75
Activity 3b: Transmit a string table to an FTP server	75
Test the strategy	78

Getting Files from an FTP Server	79
Overview Lesson 4	79
Communication handle for FTP server	79
Gateway	80
Activity 4: Get a recipe file from an FTP server	80
Modify your chart	80
Test your changes	83
 Reference	 85
Resources	85
Defining a New Control Engine	89
Configuring the Gateway Address	89



Getting Started

OPTO 22

Introduction to File Management and FTP

This tutorial and its sample files are a supplement to the SNAP PAC™ Learning Center (see Opto 22 form #1638) with explanations that apply to any SNAP PAC system.

What is file management?

Imagine that you have typed a paragraph on a computer. If you turn the computer off, what happens? If you have not saved the information to a file, then the information is lost. If you have saved the information, then it has been put into a file, given a file name, and its bits of data assigned to locations on your computer's harddrive. These activities are part of the PC's file management system, which includes numerous other capabilities such as renaming, copying, overwriting, appending to, and reading files.

File management allows us to store information as well as to share it between computers. The same applies to Opto 22's SNAP PAC family of programmable automation controllers. These controllers

are designed for industrial applications, but also have a file management system allowing you to read and write data to files just as you would on any PC.

File Management on the SNAP PAC

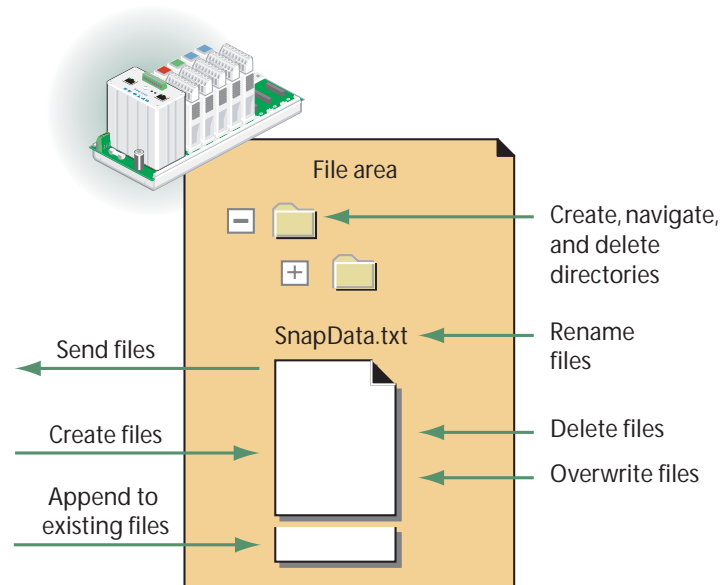


Figure 1: File management options

What this means is that the SNAP PAC has a unique ability to store and share information that is readily accessible and modeled after the file management on common PCs. This capability allows

you to connect a PC to your SNAP PAC application, allowing you to take advantage of features the PC does best.

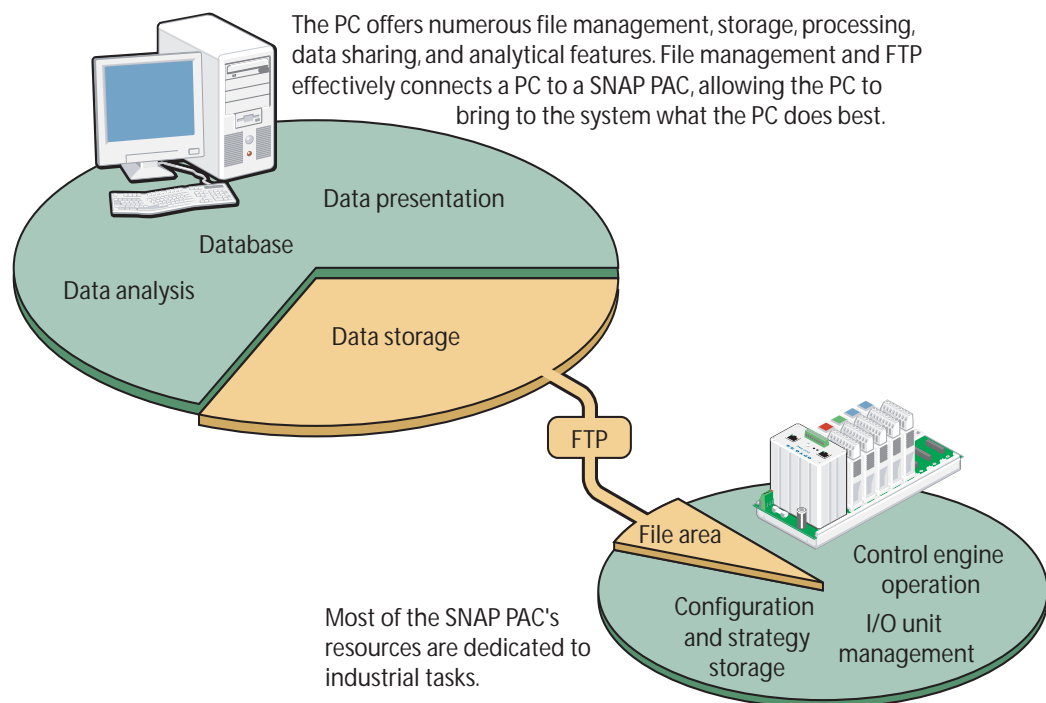


Figure 2: File management and FTP connects a SNAP PAC controller with the data management strengths and ease-of-use of a PC.

Understanding file management on the SNAP PAC

SNAP PACs (S-series and R-series) have a file management system that allows you to create directories and store files on the SNAP PAC. This file management system enables the following:

- Exchanging data between SNAP PACs and common office software programs that do not interface with ioProject software.
- Backing up data in locations where both common office applications and the SNAP PAC have access to it.

Figure 3 shows the entire memory usage of SNAP PAC R-series controller. (See appendix for similar memory for the SNAP PAC S-series.) The SNAP PAC has volatile and non-volatile memory. Most of this memory is used for the operation of the strategy, memory map (I/O), and storage of the strategy

and configuration data. The area available for storing files consists of approximately 2 MB of storage in volatile RAM and approximately 385 KB in Flash (which is non-volatile).

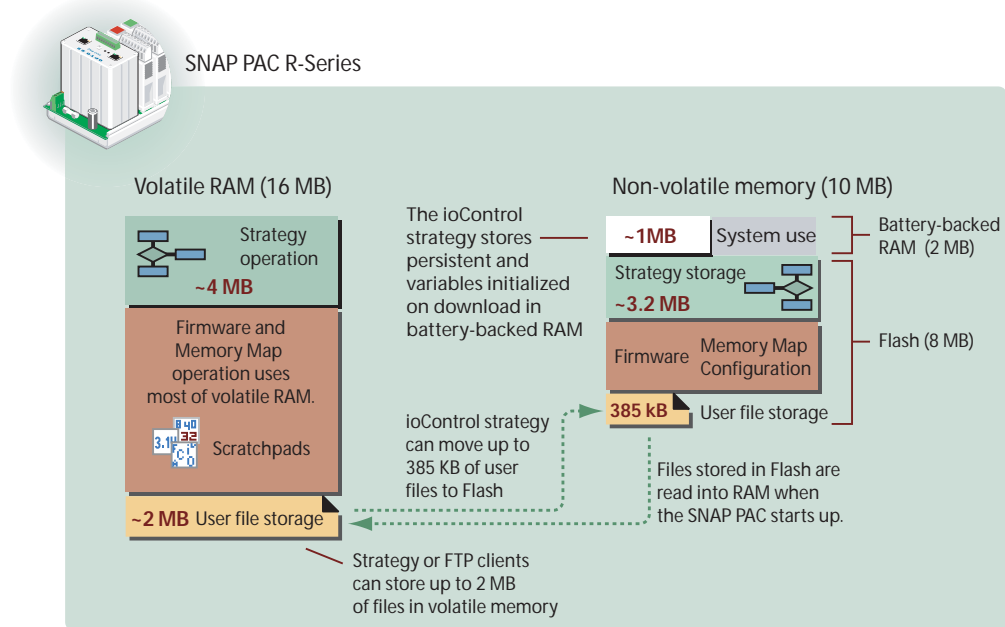


Figure 3: Memory Usage on the SNAP and File Space Available for User Files

The 2 MB of user file storage in volatile RAM can be readily written to and read by either the ioControl strategy running on the SNAP PAC or by a common office application, such as Internet Explorer or Excel.

Note there are spaces where information can be stored to non-volatile memory. One is the approximately 1 MB in battery-backed RAM used by ioControl to store variables that are configured to preserve their data across power cycles (variables initialized on download) and across downloads of new versions of the same strategy (persistent variables). This area is used by the ioControl strategy and variables here are not available via the file management system. However, any ioControl data in volatile or non-volatile memory can be written to a file where it is available to office applications.

The other non-volatile area available to the user is approximately 385 KB in Flash. Putting a file in Flash is done by first putting a file in volatile file storage and then executing a Save Files to Flash command automatically from ioControl or manually from ioManager. The dotted lines in the figure serve as a reminder that this route should be used cautiously for the following reasons:

- User Flash available may vary with versions of SNAP PAC firmware.
- An automated Save to Flash command in a repeating loop can exhaust the Flash.
- Files in Flash are copied into RAM upon startup of the SNAP PAC. Interaction with ioControl or office applications can change the file in RAM without changing the Flash file. Therefore, it's possible that the stored Flash file is out-of-date unless the RAM version is frequently resaved, a practice that is not recommended.

Saving to Flash is meant to be used sparingly, for example when you have upgraded your strategy you may wish to download and save a configuration file or record the maintenance performed on

the ioControl strategy. Note that other Flash memory is reserved for saving the strategy and memory map configuration settings. The user Flash is for any type of file you create that is smaller than 38 KB.

FTP

File Transfer Protocol (FTP) permits the exchange of files between two devices and is therefore an essential part of the SNAP PAC's file management system, allowing a PC to get files from the SNAP PAC or the SNAP PAC to get files from the PC. In FTP, one device is the FTP server and the other device is the FTP client. Figure 4 shows this distinction using the common example of PCs on a network.

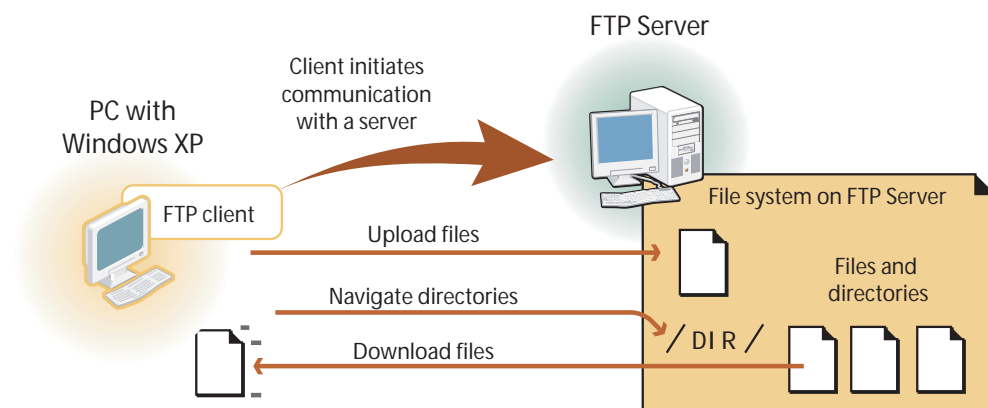


Figure 4: Distinction between an FTP client and FTP server

With the appropriate software, any network device can act as a client or as a server, the difference being that the client initiates communication while the server listens for communication and responds to requests.

Both SNAP PAC R-series controllers (provided with the Learning Center) and SNAP PAC S-series controllers can be used as an FTP client or server. Therefore, there are a variety of paths by which FTP can be used to exchange data between SNAP PACs, PCs, and servers.

SNAP PAC as an FTP server

The FTP server on the SNAP PAC provides access to the SNAP PAC's file management system. All communication between an office application and the SNAP PAC file system goes through the

onboard FTP server. Such communication consists of uploading files to and downloading files from the SNAP PAC. In FTP, these operations are known as Get and Send.

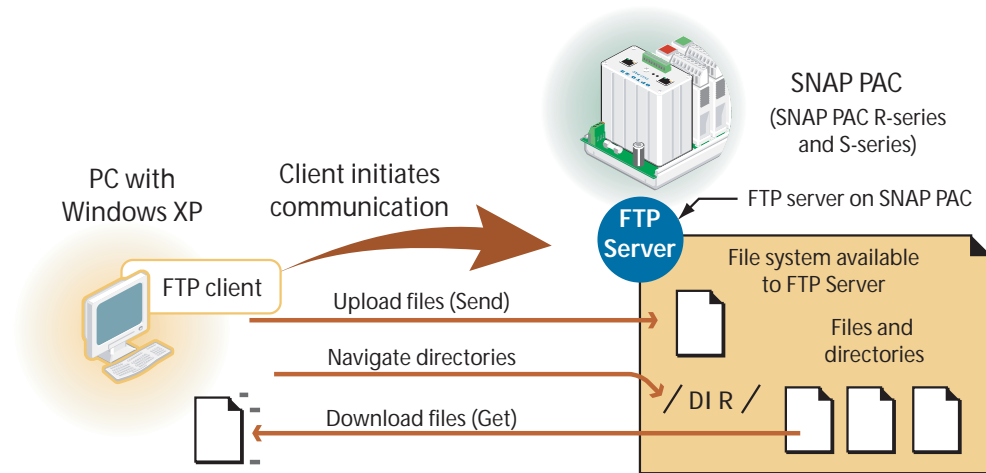


Figure 5: FTP client gaining access to the SNAP PAC's file system

SNAP PAC as an FTP client

Using the SNAP PAC as an FTP client allows the SNAP PAC to upload data to and download files from an FTP server on a PC (The SNAP PAC's FTP server is not used in this communication.) Data that is uploaded to the FTP server can come from either a stored file or directly from variables in ioControl.

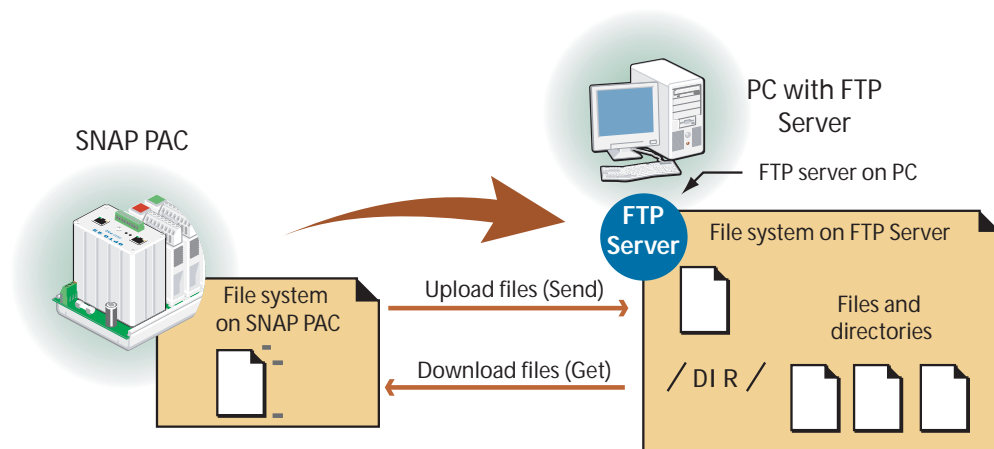


Figure 6: SNAP PAC gaining access to an FTP server

File management and FTP

The SNAP PAC combines file management with FTP client and server capabilities. These features can be used within or outside of the ioControl strategy running on your SNAP PAC. Figure 7 shows the memory use in the context of activity taking place on the controller by either the ioControl strategy

or the onboard FTP server. These numbered paths are various options that can be used themselves or combined with other options.

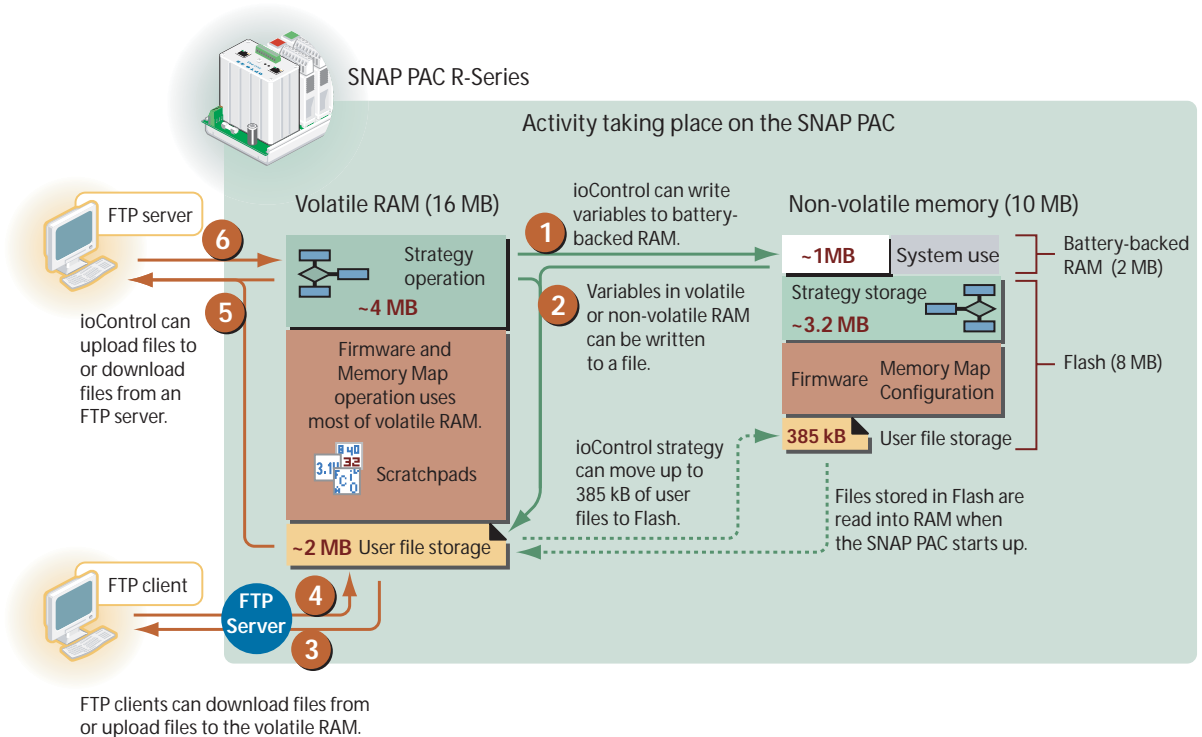


Figure 7: Routes of information using memory management and FTP

1	Route 1 shows activity taking place within an ioControl strategy. If the strategy stores information in a variable configured to be persistent or initialized upon startup, then the data in the variable is protected in battery-backed RAM.
2	The value of any variable in ioControl can be written to a file in the user file storage area. Whether the variable is in battery-backed or volatile RAM makes no difference; ioControl treats both types of variable the same.
3	Any FTP client with network access to the SNAP PAC can read a directory of files and copy a file from the SNAP PAC's file storage area.
4	Any FTP client can copy a file onto the SNAP PAC. The SNAP PAC can be programmed to detect and read the file, and then use the data contained.
5	Route 5 shows the SNAP PAC as a client uploading data either from a variable in ioControl or as a file stored in RAM. Both actions create or append to files on the FTP server.
6	Route 6 shows the reverse of 5 where the SNAP PAC as a client downloads a file that can be opened and read by ioControl.

Lesson Topics

The numbering of the routes in Figure 7 correspond to the sequence of activities in this tutorial.

The above concepts are discussed in the following lessons:

- Lesson 1: [“Creating and Downloading a Log File” on page 15](#)
(Shows how to implement routes 1, 2, and 3 in the Figure 7)
- Lesson 2: [“Sending Files to SNAP PACs” on page 53](#)
(Shows how to implement route 4 and parse a text file)
- Lesson 3: [“Sending Data to an FTP Server” on page 69](#)
(Shows how to implement route 5)
- Lesson 4: [“Getting Files from an FTP Server” on page 79](#)
(Shows how to implement route 6)

Using This Tutorial

This tutorial and its sample files are a supplement to the SNAP PAC™ Learning Center (see Opto 22 form #1638) with explanations that apply to any SNAP PAC system.

The sample ioControl strategy (**1642_FileMgtFtp.idb**) is ready for use with a SNAP PAC Learning Center. If you are using this tutorial with a different configuration, please note the following:

Point	Description	Used in Lesson	If your SNAP PAC isn't connected to the Learning Center
00	Digital Input, Emergency	2, 4	In lesson 2, this point initiates a demonstration. Substitute with any digital point or integer 32 variable. Set variable to true from Debug mode to simulate switched input.
03	Digital Input, Photo Sensor	1, 3	The value of this point is written to a log file. Substitute with any digital input, digital output, or integer 32 variable.
04	Digital Output, Outside Light	2, 4	This point is controlled by a recipe file uploaded to the controller. Substitute with any digital output or integer 32 variable.
05	Digital Output, Inside Light	2, 4	Same as point 04.
06	Digital Output, Freezer Door Status	2, 4	Same as point 04.
12	Analog Input, Store Temperature	1, 3	This point is read and written to a log file. Substitute with any analog point or float variable.

What You Need

Instructional approach

The lessons of this tutorial have two parts: The first part is an explanation of the concepts being taught; the second part is a step-by-step activity using ioControl and the various FTP client or FTP server applications.

Sample files

The sample strategy you will use with this tutorial is provided in both before and after form:

- **1642_FileMgtFtp_before.zip:** This is an ioControl strategy that you will complete during this tutorial.
- **1642_FileMgtFtp_after.zip:** This is an example of the completed ioControl strategy from this tutorial.

To prepare for the tutorial, unzip **1642_FileMgtFtp_before.zip** in a directory of your choosing. **1642_FileMgtFtp_after.zip** is provided for reference only.

FTP server

Lessons 3 and 4 use an FTP server. You have the option of using the free FTP server cited in this tutorial (see [“Installing an FTP server” on page 9](#)) or you may substitute your own. If you use your own, consider the following:

- If your FTP server and SNAP PAC are separated by a router, you will need to configure the SNAP PAC’s IP settings to include the location of the router. How to do this is described in the Appendix (see [“Configuring the Gateway Address” on page 89](#))
- The instructions in lessons 3 and 4 assume you’ve installed the FTP server below. Therefore, when following these instructions, substitute the IP address, port numbers, login name, and passwords cited in the instructions as needed with those corresponding to your FTP server.

Installing an FTP server

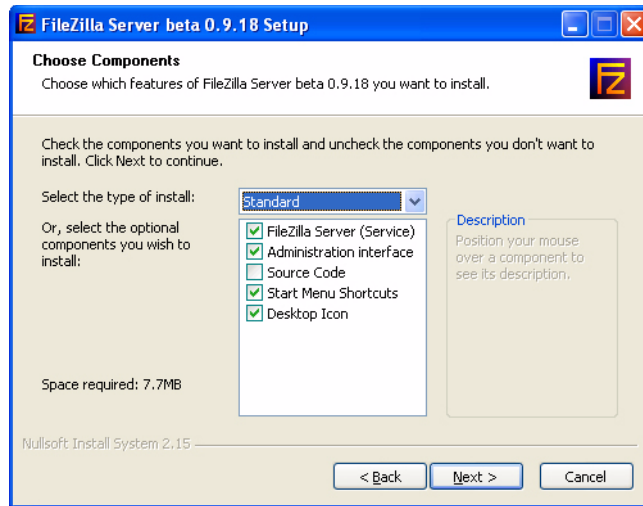
The samples shown in this tutorial use a shareware FTP server called FileZilla available from this location:

`http://sourceforge.net/projects/filezilla/`

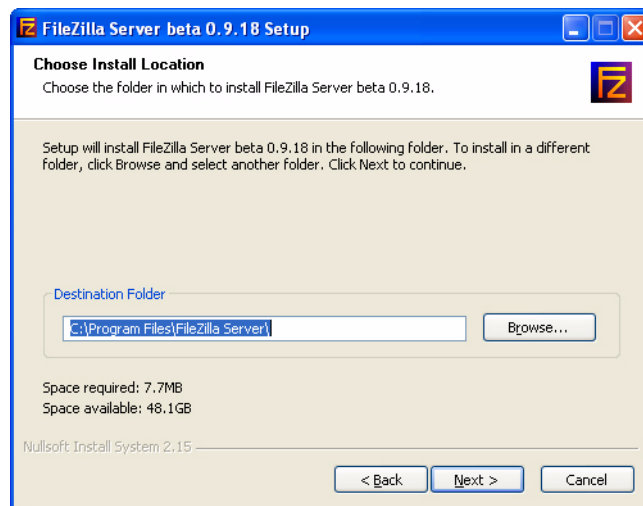
NOTE: You need only the FileZilla server, and not the complete package.

1. Run the FileZilla server setup program

2. After accepting the licensing agreement, choose standard setup.

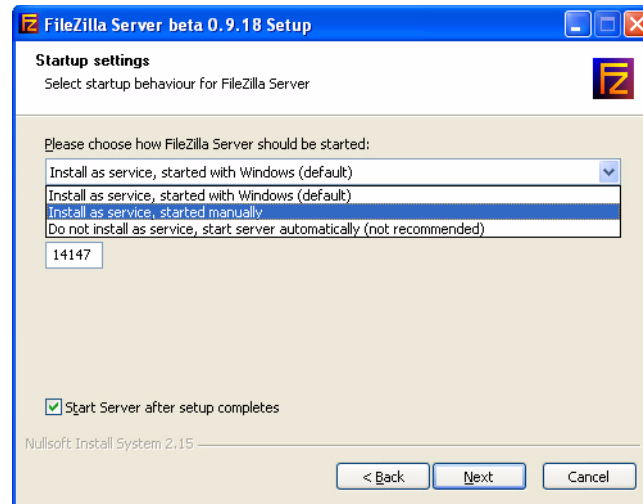


3. Choose a directory location.



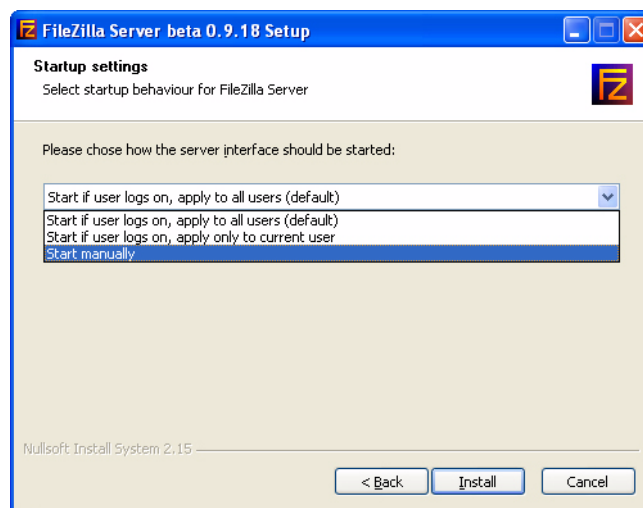
The installed location is up to you. During this tutorial you will designate a location on your computer for the FTP server to use for file sharing.

4. Choose startup settings.



Choose Install as Service, either Started with Windows or Started Manually. If you choose Started Manually, make sure you start the server before you try to access it in lessons 3 and 4.

5. Choose how the server interface is started.



The installation will run.

Setting up FTP client accounts

You will need to set up an FTP client account with the following privileges:

Name: SnapPac

Password: snap

File privileges: Read, Write, Append

Directory privileges: Create

If you are using the FileZilla FTP server, the procedure is shown below:

1. Create a directory on your PC

Use File Manager to create C:/FtpServer on your PC.

The tutorial uses this location, but you can create the directory anywhere suited to your needs.

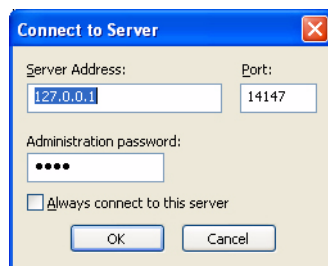
2. Start FileZilla (if not started automatically)

Start → Programs → FileZilla Server → FileZilla Server

3. Start the FileZilla server interface

Start → Programs → FileZilla Server → FileZilla Server Interface

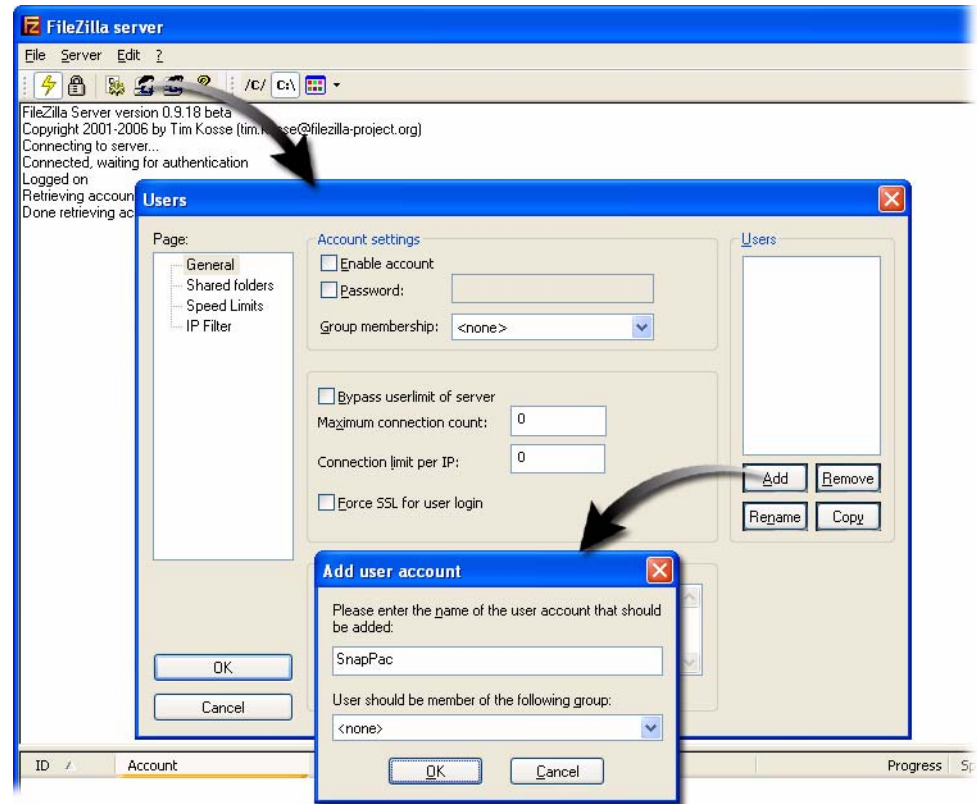
4. Connect to the server.



5. Create a user.

- a. Open the Users dialog box (click the Users button on the tool bar).

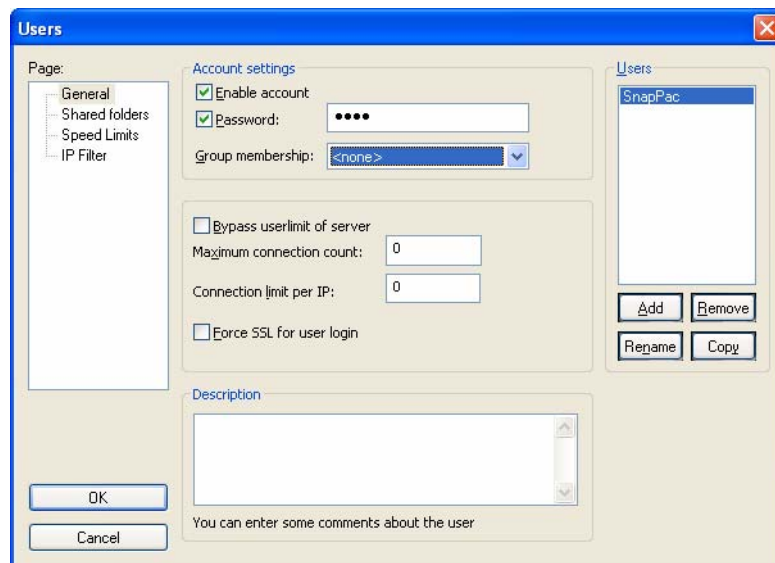
- b. In the Users dialog box, click Add.



- c. In the Add user account dialog box, type a name, such as `SnapPac` used in this tutorial.

- d. Click OK.

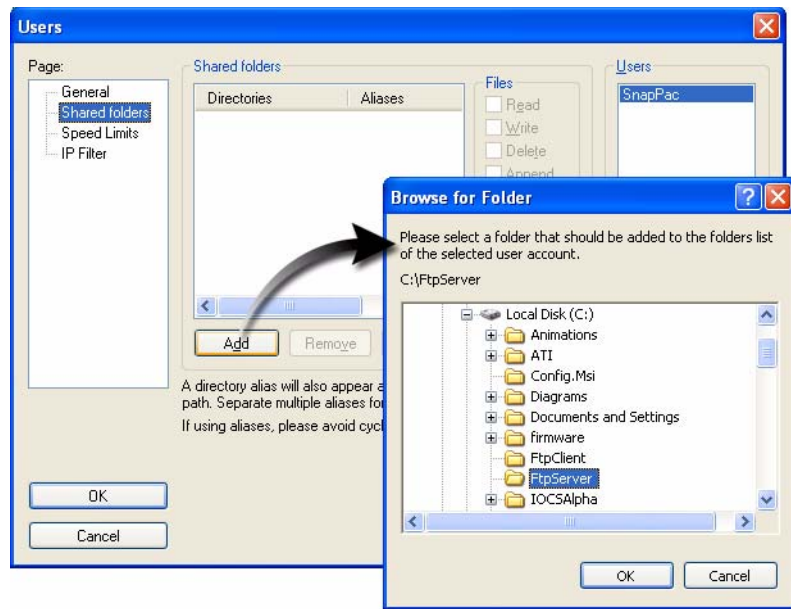
6. In the Users dialog box, type a password. This tutorial uses `snap` as a password.



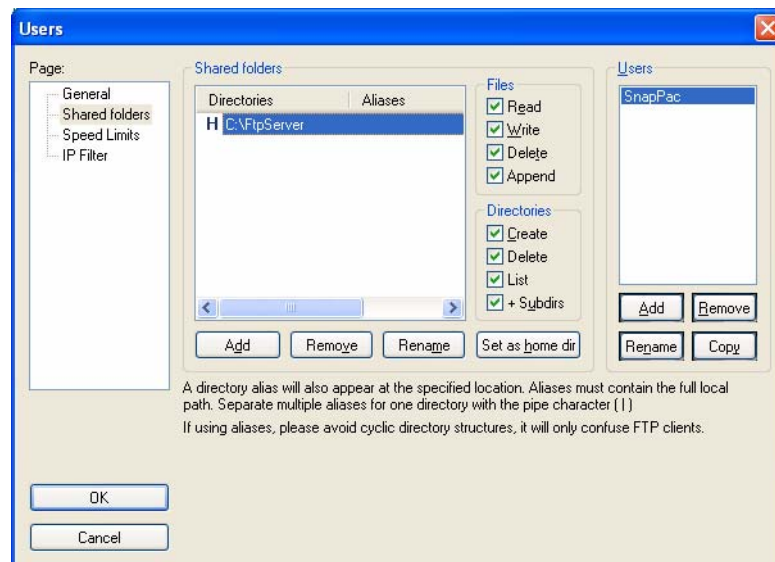
7. Create a folder for files to be uploaded and downloaded.

- a. Under the Page field, click Shared Folders.

- b. Under Shared Folders, click Add.



- c. In the Browse dialog box, select the folder you created in Step 1, for example, FtpServer.
8. Click OK.
9. Select File and Directory privileges.
- a. Select all of the options under the File and Directories as shown here.



- b. Click OK to close the Users dialog box.
- Your FTP server is ready for use in Chapters 3 and 4.

Creating and Downloading a Log File

OPTO 22

Lesson Overview

The ioControl strategy built in this lesson will demonstrate how to log data to an ASCII text file on the controller. After logging the data, you can retrieve it using any FTP client.

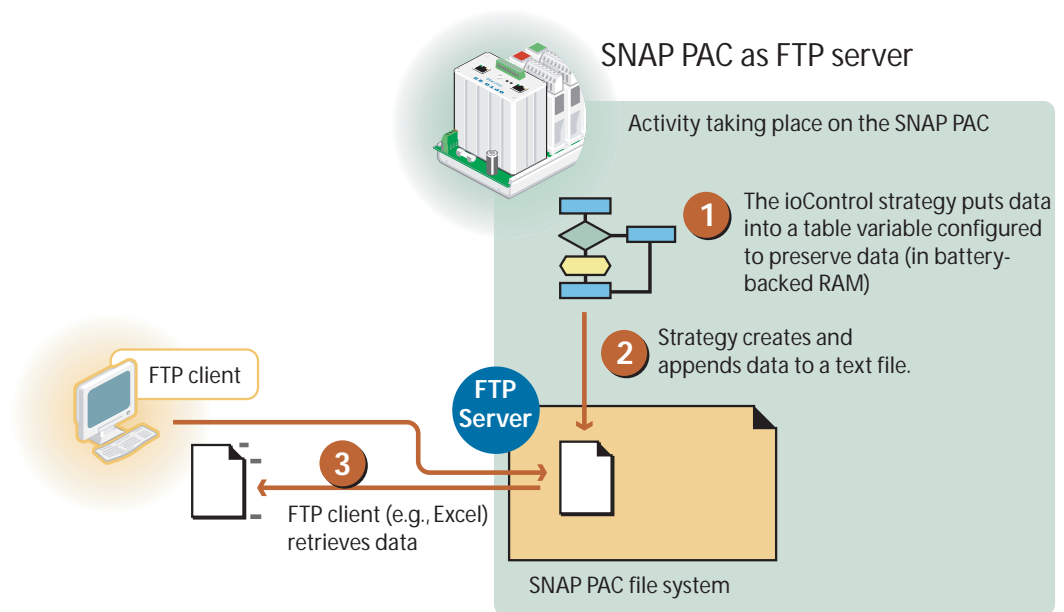


Figure 8: Functionality demonstrated in Lesson 1

1	The strategy converts numerical data to string format and puts the information into a persistent string table. An index is incremented each time a string of data is added to the table.
2	When the last element in the table has been written to, the table is appended to a file.
3	Any FTP client can get data from the SNAP PAC's file system. FTP client capabilities are built into the Windows XP operating system. This allows you to open the XP FTP client (which runs in a DOS shell), or to use programs like Excel, Internet Explorer, and even Notepad to open files from the SNAP PAC.

Concepts

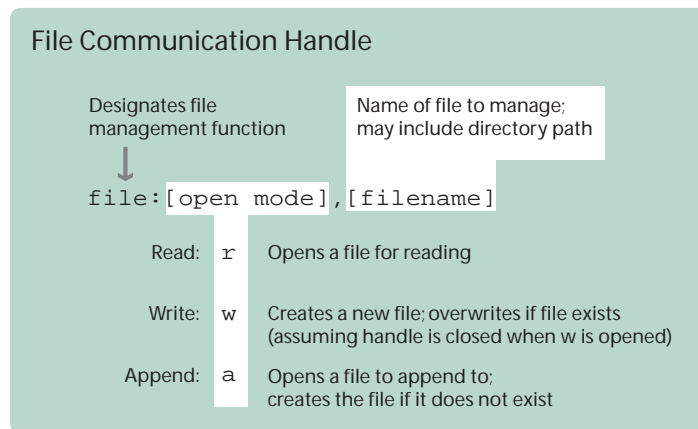
File communication handle

Communication from within ioControl is established through variables called Communication Handles. In ioControl, reading, creating, and appending to files are also forms of communication defined within Communication Handles. In this lesson, a communication handle configured to open, read, and append to files is referred to as a file Communication Handle.

A file Communication Handle is a text string that provides the communication parameters between ioControl and the file on the controller. For example, a file communication handle may contain the following information:

file: a, LogData. txt

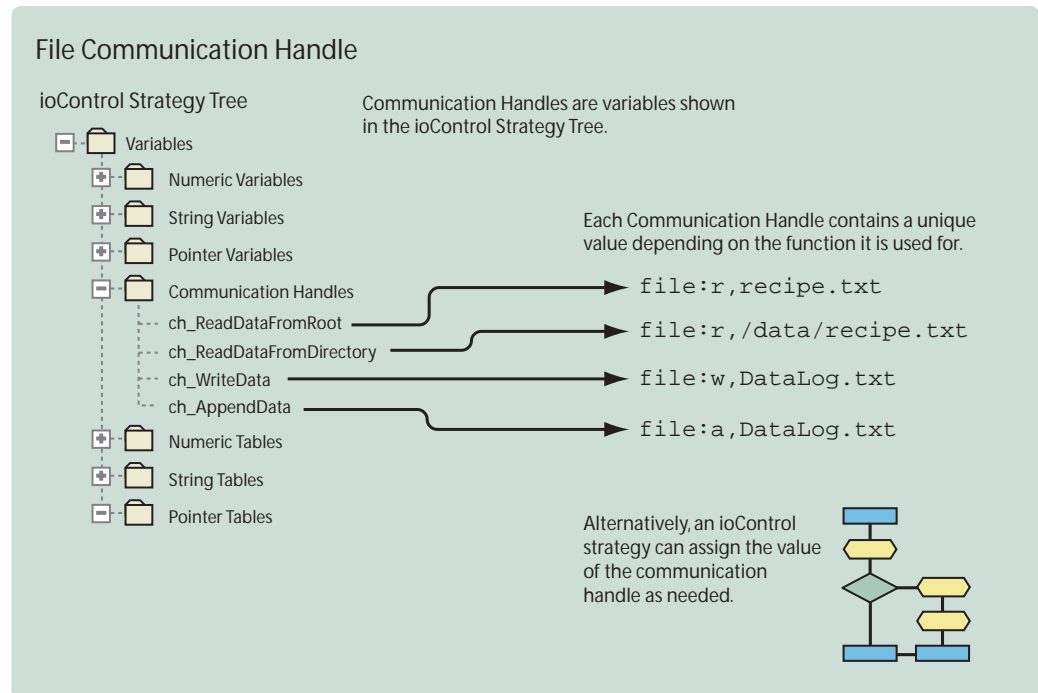
which tells ioControl to use the file management system (**file**) to append (**a**) information to a text file called **LogData. txt**. A file communication handle has the following parameters:



Note that the file communication handle contains a parameter that describes an action, as well as a file to act upon. The actions are:

- read the file (**r**),
- write the file (**w**),
- and append to the file (**a**).

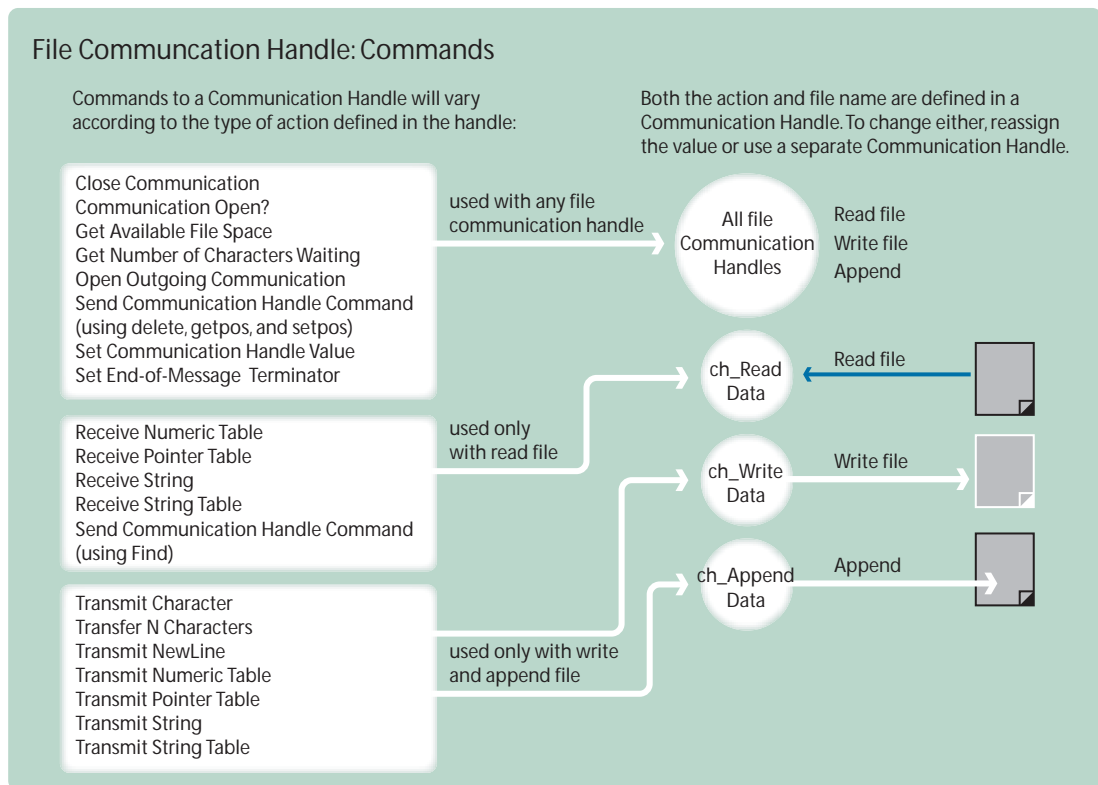
To implement these different actions to the same file, you would create unique communication handles during configure mode or use ioControl to change the value of the communication handle before implementing the desired action.



Typical use of a communication handle

When using a communication handle in your ioControl strategy, typically you would define the contents of a communication handle (set communication handle value), open the communication handle, and then send it one or more commands. After you've sent the commands, it's recommended that you close the communication handle.

The commands that you can send to a communication handle will vary depending on the action parameter selected:



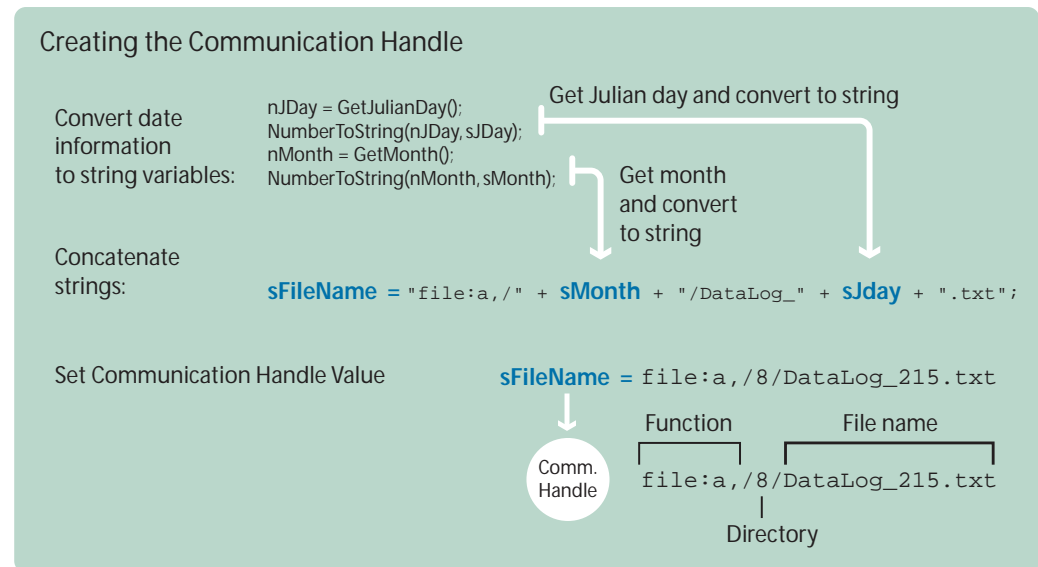
Setting the value of a Communication Handle

A Communication Handle is a variable containing communication parameters as text strings. For example, the Communication Handle `ch_AppendFile` contains the following:

file: a, data log. txt

The **file** parameter tells ioControl to use the I/O Unit's file management and is followed by an append instruction and a filename. In this lesson, the persistent variable `nRun` is stored in non-volatile RAM and is incremented each time the strategy starts. By converting this variable to a

string and inserting it into a file name, you can dynamically change the file name within a Communication Handle.

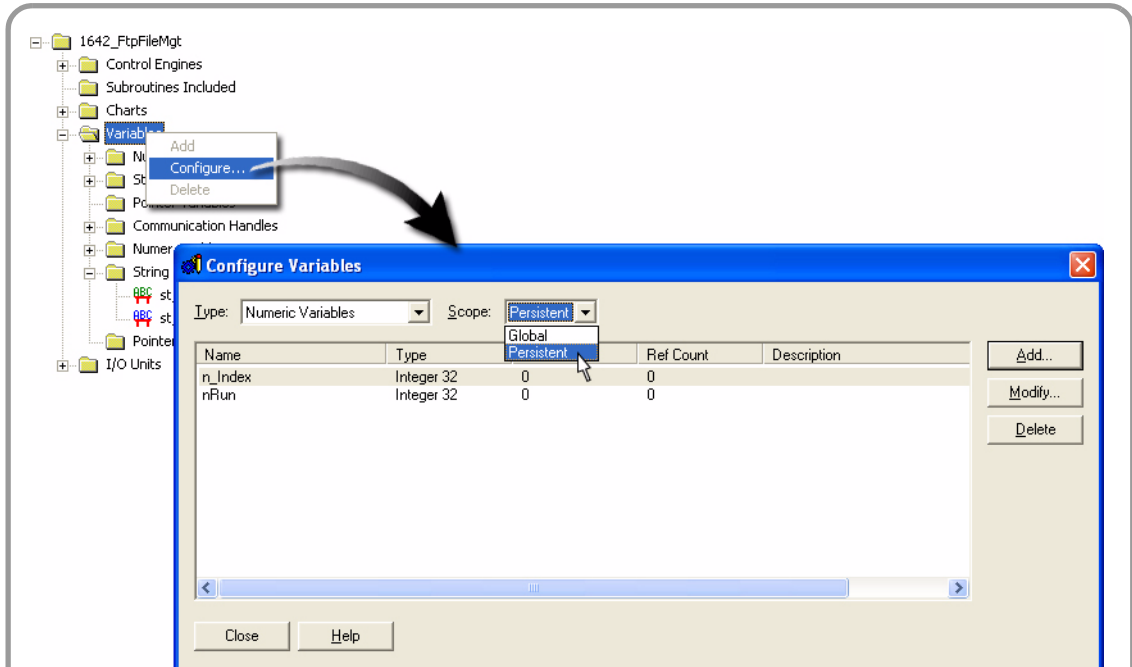


In this example, the communication handle creates a log file name that is incremented daily and stored to a directory name that incremented monthly.

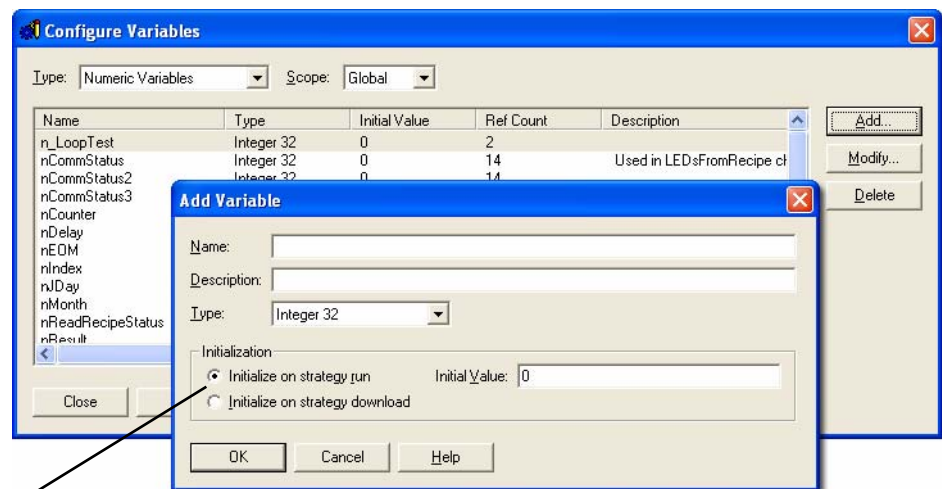
Size of data stored in persistent files

The variables you create in ioControl reside in either volatile RAM or battery-backed RAM. This is determined by configuration settings chosen when you create a variable. When creating a variable, you need to choose whether it is Global or Persistent:

- **Global variables** can be in volatile or non-volatile memory, depending on a configuration setting
 - Choose **Initialize on strategy run**, to put a variable in volatile memory
 - Choose **Initialize on strategy download** to put a variable in non-volatile, battery-backed RAM
- **Persistent variables** are stored in non-volatile memory and you can update your strategy (assuming the strategy is of the same name) without clearing the data in these variables.



Create a persistent variable using the Configure Variables dialog.



Choose Initialize on strategy run or Initialized on strategy download for global variables.

The decision, to choose either persistent, global—Initialize on download, or global—initialize on strategy run, is based on your application's needs and how much space you have available. For example, if your data doesn't have to be saved should power be lost, or you are saving the data by some other means (for example, ioDisplay, OptoOPCServer, or using FTP to send the data as shown in Lesson 3) you may not need to configure a variable as persistent or as initialized upon download.

Creating a large variable, such as a string table, in persistent memory raises the concern that there are limits to how large you would want this file to get.

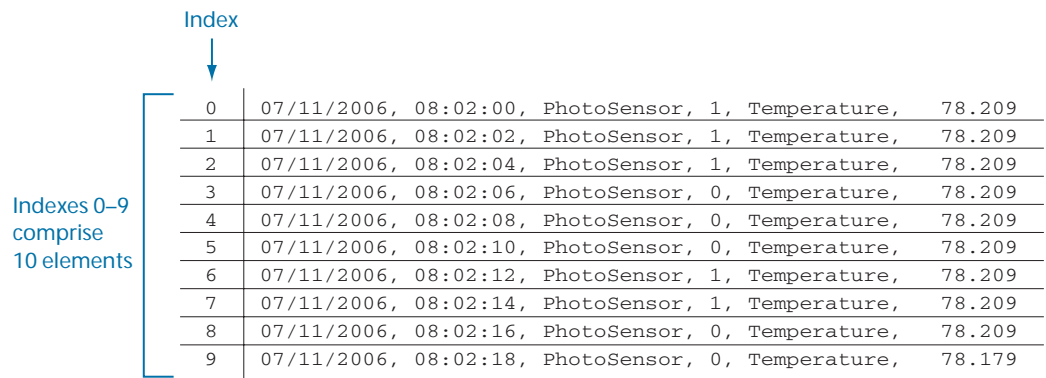
Upon downloading a strategy, ioControl determines if the strategy is within the physical memory constraints of the SNAP PAC hardware. The physical constraints are the RAM available for the ioControl strategy and the battery-backed RAM available for the persistent variables. If the size of variables requires more memory than is available, ioControl stops the download and advises you that either the strategy is too big for volatile RAM or the variables are too big for the available battery-backed RAM.

Putting data into persistent variables, such as table variables, is an easy means of keeping data in non-volatile memory, but the following considerations apply:

- A table variable cannot grow indefinitely. You have to designate a maximum length for the table (this is the maximum number of elements the table can contain). You should have a plan for retrieving the data when the table is full.
- Variables in non-volatile RAM are available to ioControl, ioDisplay, and OPC clients, but you may wish to make them available to other applications using FTP. You can use ioControl's FTP and file management features to package data in any variable into text files.

Table variables

As a table has a length representing the number of elements it can contain, commands to send table variables must cite the starting index and number of entries to send. For example, to send the first ten items, you would designate 10 for number of elements to send starting at index 0.



Index	
0	07/11/2006, 08:02:00, PhotoSensor, 1, Temperature, 78.209
1	07/11/2006, 08:02:02, PhotoSensor, 1, Temperature, 78.209
2	07/11/2006, 08:02:04, PhotoSensor, 1, Temperature, 78.209
3	07/11/2006, 08:02:06, PhotoSensor, 0, Temperature, 78.209
4	07/11/2006, 08:02:08, PhotoSensor, 0, Temperature, 78.209
5	07/11/2006, 08:02:10, PhotoSensor, 0, Temperature, 78.209
6	07/11/2006, 08:02:12, PhotoSensor, 1, Temperature, 78.209
7	07/11/2006, 08:02:14, PhotoSensor, 1, Temperature, 78.209
8	07/11/2006, 08:02:16, PhotoSensor, 0, Temperature, 78.209
9	07/11/2006, 08:02:18, PhotoSensor, 0, Temperature, 78.179

Figure 9: Sample table containing 10 elements

You can create your table to be of any size (limited by battery-backed RAM), but you want to make sure that whether your table's length is 10 or 1000, you send all elements to your file.

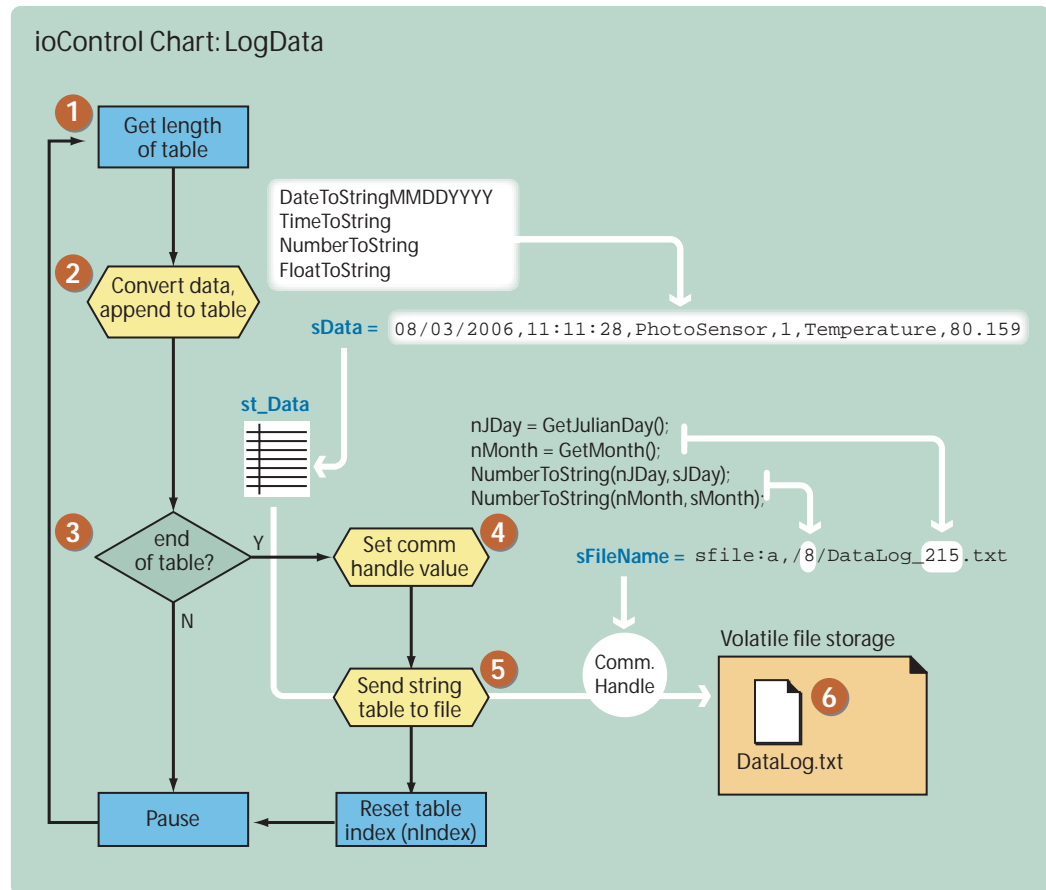
The size of a table can be measured in ioControl and the answer used to determine how many table elements to send to a file. This way, the strategy will adapt to any changes you make to the table's length.

Table variables are sent to a communication handle using these commands:

```
Transmit Numeric Table
Transmit String Table
```

LogData chart

The sample strategy (1642_FileMgtFtp.ibd) demonstrates the above concepts in the chart called LogData. This chart contains some basic functions as well as some empty blocks that serve as placeholders for functionality you will add. The diagram below describes the key functions this chart will perform when you complete lesson 1.



1	This block reads the length of the string table st_Data and puts the length in the integer variable nTableMax. nTableMax is used to detect when the table is full. Reading the length of the table, allows you some flexibility to adjust how much data you send at a time to the file. You can resize the table and the strategy will use the new size to know when the table is ready to send.
2	This block converts the data to strings and appends the data to the table. In doing this, the strategy increments a variable used as a table index.
3	The condition block detects when it's time to send the data in st_data to a file by comparing nIndex, used to write to the table, to nTableMax, based on the total number of elements in the table.
4	This script creates a string that is written to the communication handle. The script gets the current data and month and uses this information to create files and directories reflecting the month and day.
5	The Transmit String command sends the contents of sData to the communication handle. The Append action will append the string or create a new file if the file doesn't exist.

6

Once the file is written to the file area, it can be retrieved by any FTP client application.

Activity 1a: Creating a Log File

In this activity, you will do the following to the ioControl strategy 1642_FtpFileMgt.idb:

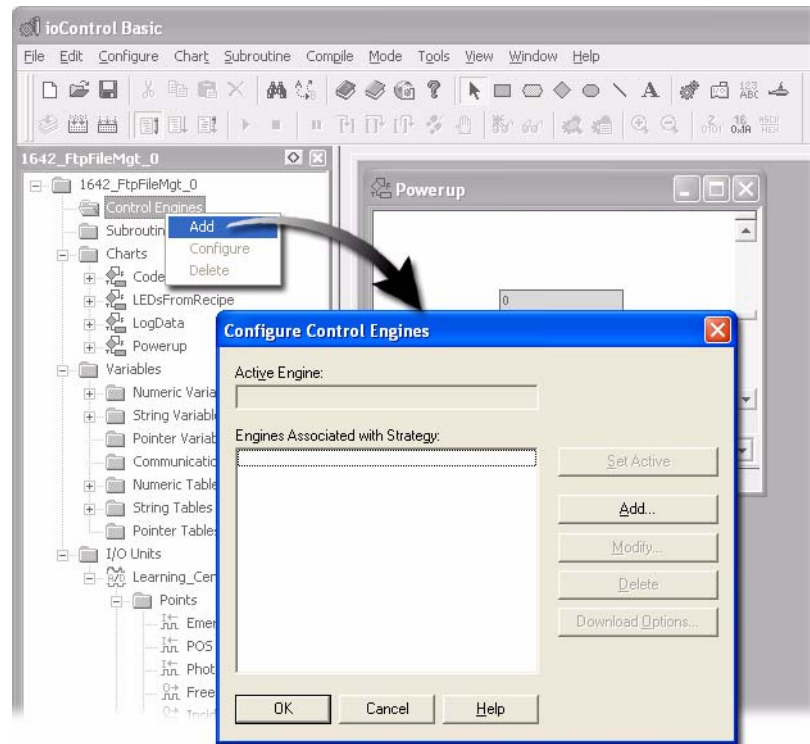
- Configure a controller
- Create persistent variables
- Configure a communication handle
- Add instructions to the LogData chart that:
 - Get length of a table
 - Convert numerical data to strings
 - Append strings to a string table
 - Write or append to a file

Preparation

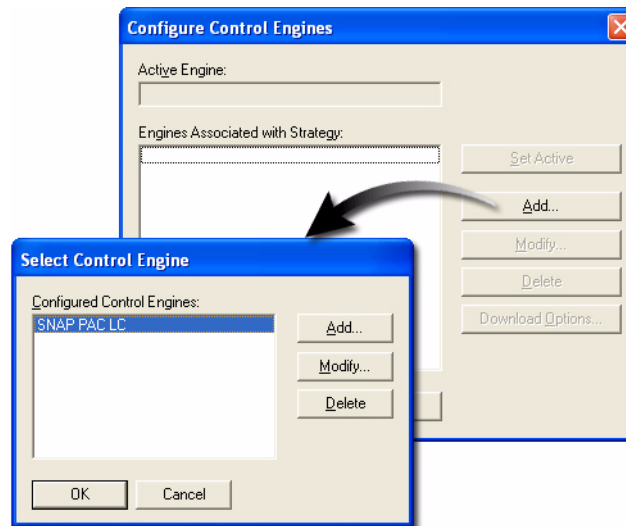
The sample ioControl strategy is ready to use with SNAP PAC Learning Centers. To use this strategy, you will need to modify the Control Engine.

1. Open Strategy 1642_FtpFileMgt.idb in ioControl.
2. Add a Control Engine
 - a. In the strategy tree, right-click the Control Engines folder.

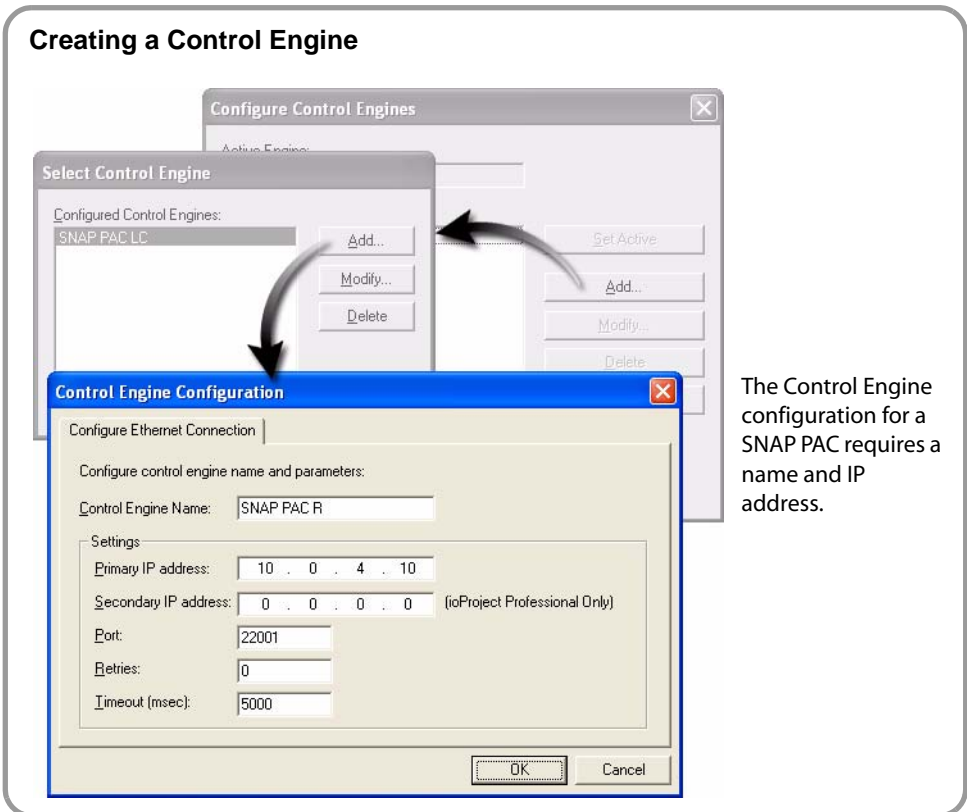
b. Click *Add*.



c. In the Configure Control Engines dialog box, click *Add*.



All control engines you have configured on your computer are shown in the Select Control Engines dialog box. If you do not have a control engine, see Creating a Control Engine below.



- d. Select your Control Engine and click **OK**.
- e. Click OK to close the Configure Control Engine dialog box.

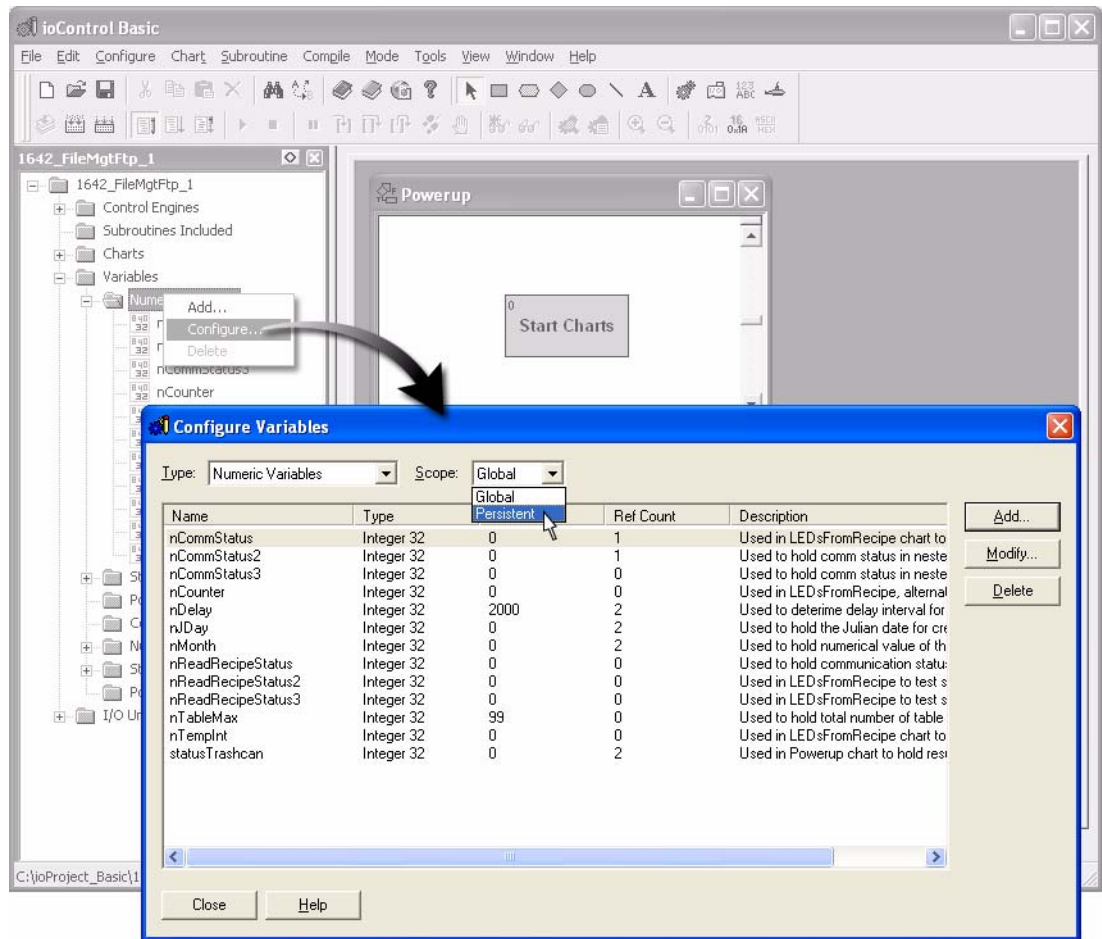
Add strategy variables

Strategy variables can be added using the Strategy tree, or added (on-the-fly) when creating instructions. There are some exceptions. Persistent variables need to be added through the strategy tree and OptoScript blocks don't offer an on-the-fly method.

1. Add a persistent variable nIndex
 - a. In the Strategy Tree, expand the Variables folder.
This shows folders indicating the variable types.
 - b. Expand the Numeric Variables folder.
This shows the variables that have already been created in the sample strategy.
2. Open the Configure Variables dialog box.
 - a. Right-click the Numeric Variables folder.
 - b. Choose Configure from the pop-up menu

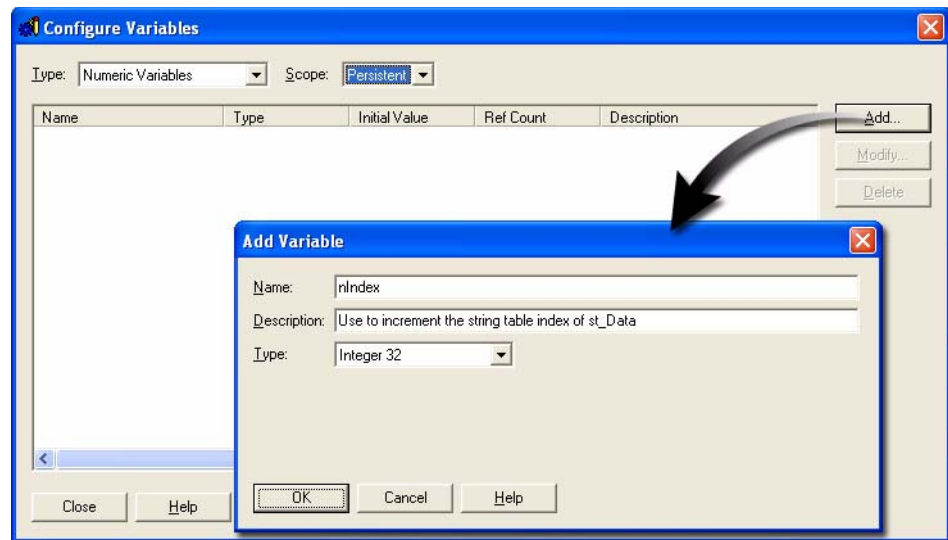
Persistent variables reside in battery-backed RAM and their values are preserved when new versions of the same strategy are downloaded. Variables initialized on download are also stored in battery-backed RAM and can be created on-the-fly. You could use either for the variables st_table and nIndex. This lesson uses persistent variables to emphasize how to find this feature.

NOTE: Make sure you choose Configure instead of Add.

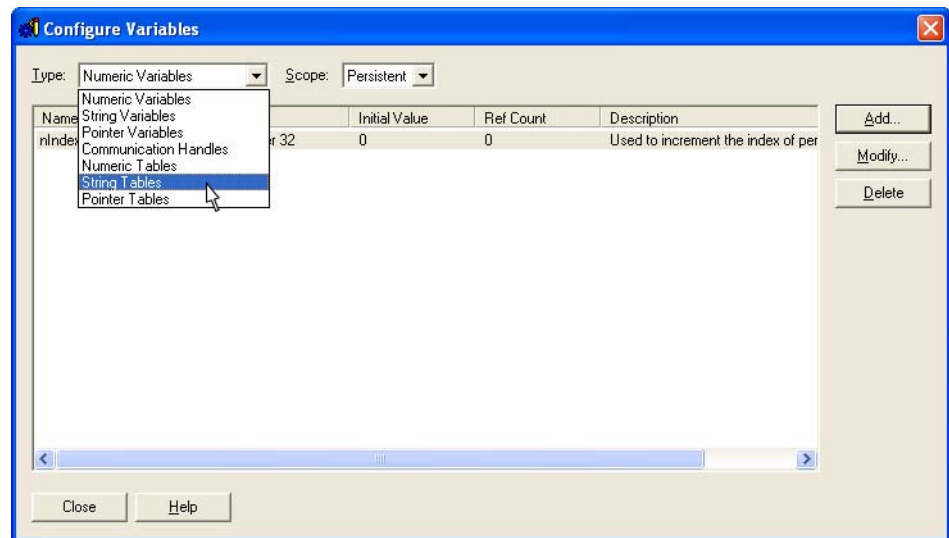


3. In the Configure Variables dialog box, choose Persistent under the Scope dropdown menu. Notice how the list of variables changes. There are no Persistent variables in this strategy.
4. Define a new persistent variable

- a. In the Configure Variables dialog box, click the Add button.

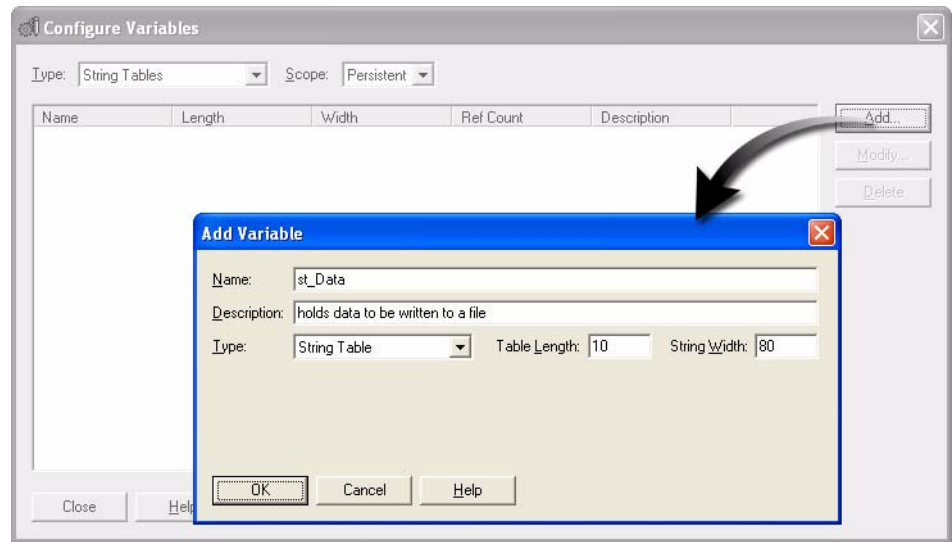


- b. In the Add Variable dialog box, provide the following information:
 Variable name: `nlIndex`
 Comment: `Used to increment the index of persistent string table st_Data`
 Type: Integer 32
- c. Click OK to close the Add Variable dialog box.
- d. Keep the Configure Variables dialog box open.
5. Add a persistent string table variable.
- a. In the Configure Variables dialog box, choose String Tables from the Type dropdown list.



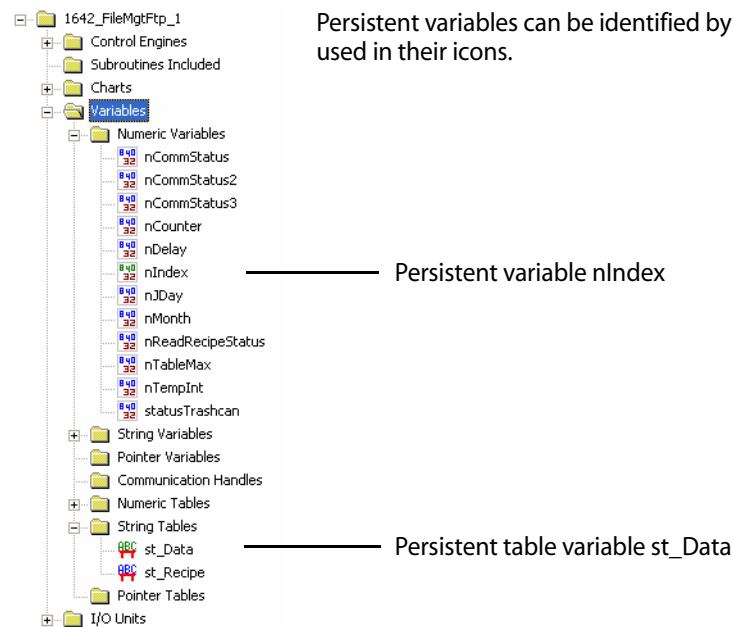
- b. Make sure Persistent is still selected.

- c. Click Add.



- d. In the Add Variable dialog box, provide the following information:
- Name: **st_Data**
 - Description: **holds data to be written to a file**
 - Type: **String table**
 - Table length: **10**
 - String width: **80**
- e. Click OK to close the Add Variable dialog box.
- f. Click Close to close the Configure Variables dialog box.
6. Examine the Persistent variables.

Persistent variables appear differently in the Strategy Tree. Expand the Numeric Variables folder and the String Table Variables folder to see the new variables.



NOTE: These persistent variables can only be configured through the Configure Variables dialog box as you have done. The Remaining variables can be created as needed when you add instructions to the LogData chart.

7. Add the string variable sData.

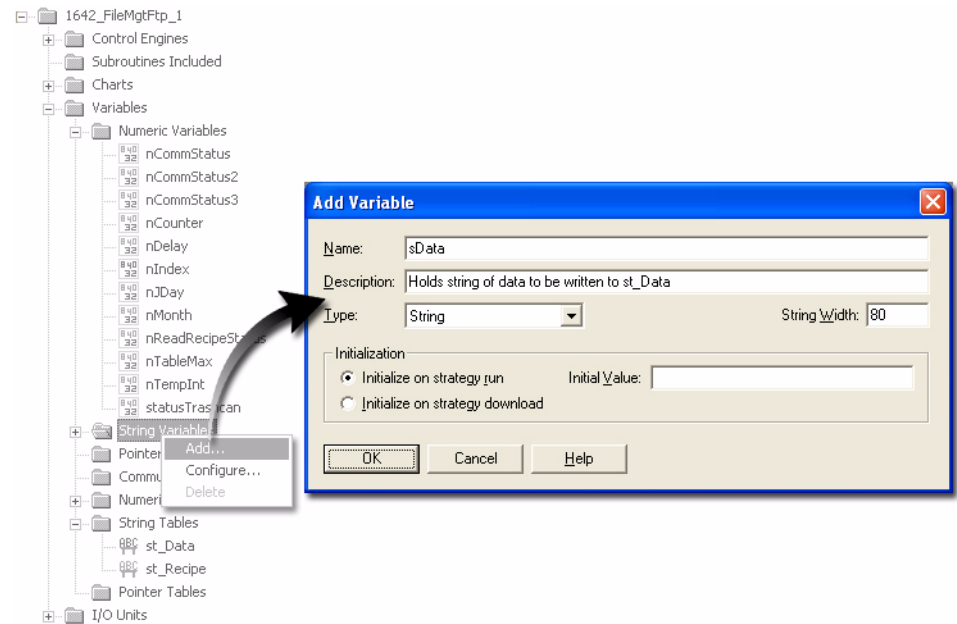
- a.** In the Strategy Tree, right-click the String Variables folder.
- b.** Choose Add
- c.** In the Add Variable dialog box, choose the following:

Name: **sData**

Description: **Holds string of data to be written to st_Data**

Type: **String**

String width: 80



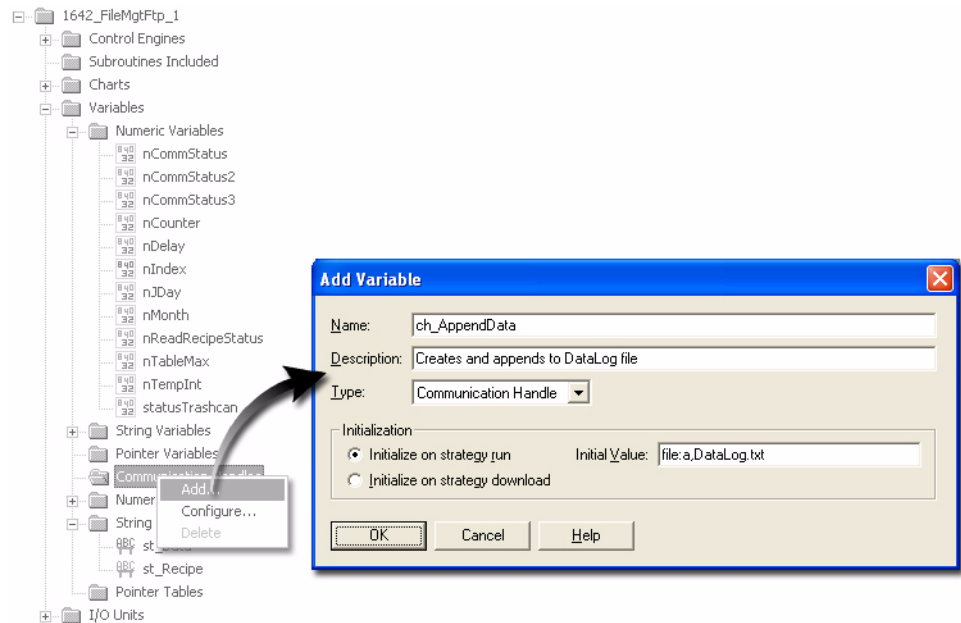
You can leave the Initial value blank.

d. Click OK to close the Add Variable dialog box.

8. Add Communication Handle `ch_AppendData`

As with any variable, Communication Handles can be assigned an initial value and their value can be reassigned using chart commands or OptoScript. Here you will assign the initial value. Later, you will use OptoScript to reassign the value.

a. Right-click the Communication Handles folder.

b. Click *Add*.**c.** In the Add Variable dialog box, type the following:Name: `ch_AppendData`Comment: `Creates and appends to the datalog file`Initial Value: `file:a, datalog.txt`

NOTE: File communication handles use lowercase letters for file:a. File names can be in upper lower case letters.

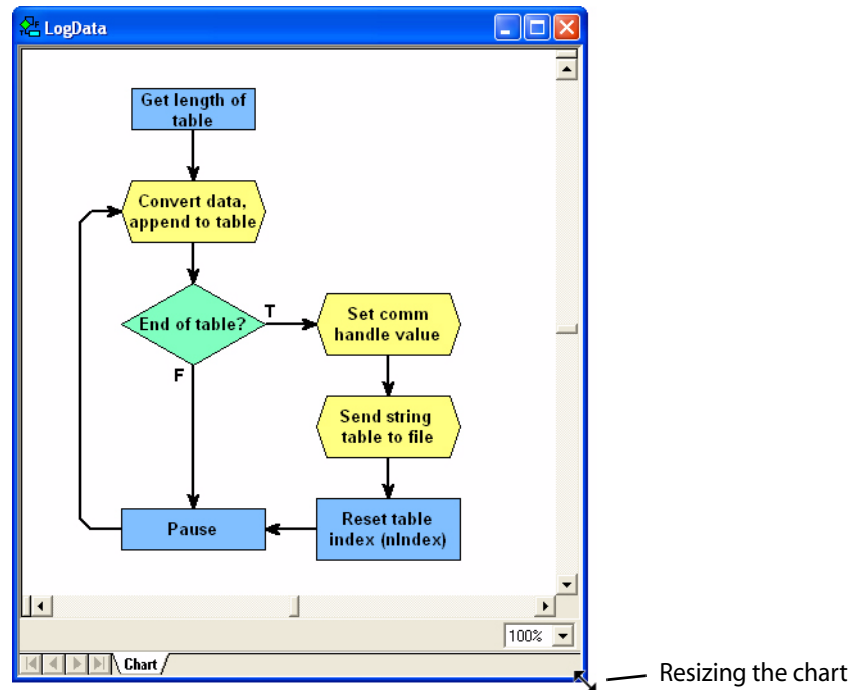
d. Click OK to close the Add Variable dialog box.

Chart: LogData

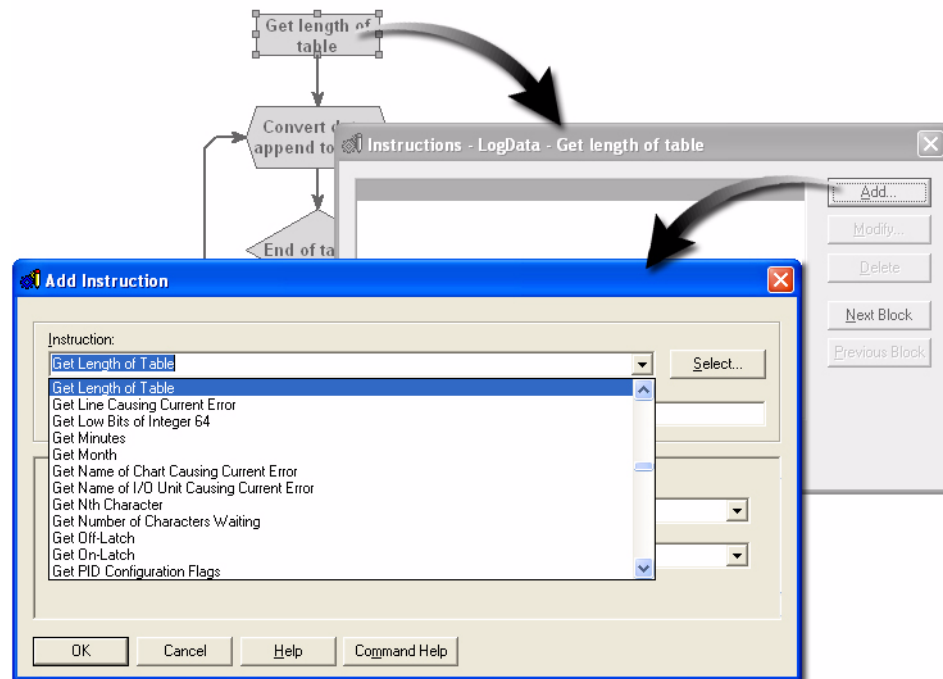
1. Open the LogData chart.

- a.** In the Strategy Tree, expand the Charts folder.
- b.** Under Charts, double-click LogData.

- c. Resize the chart if necessary.



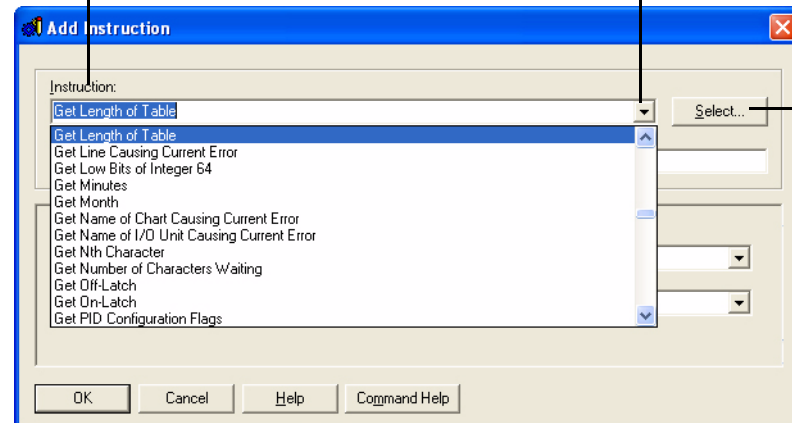
2. Get length of table.
 - a. Double-click Get Length of Table.
 - b. In the Instructions dialog box, click Add.
 - c. In the Add Instruction dialog box, select Get Length of Table for the Instruction.



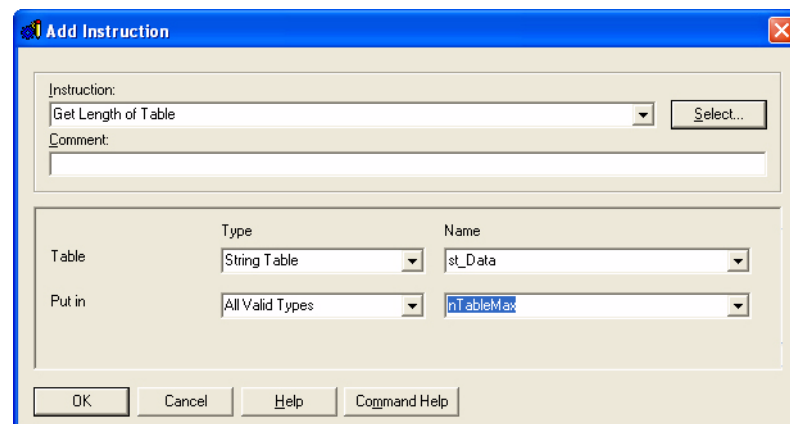
Review: Three ways to locate an instruction

Type the first few letters

Click here to see the alphabetized list



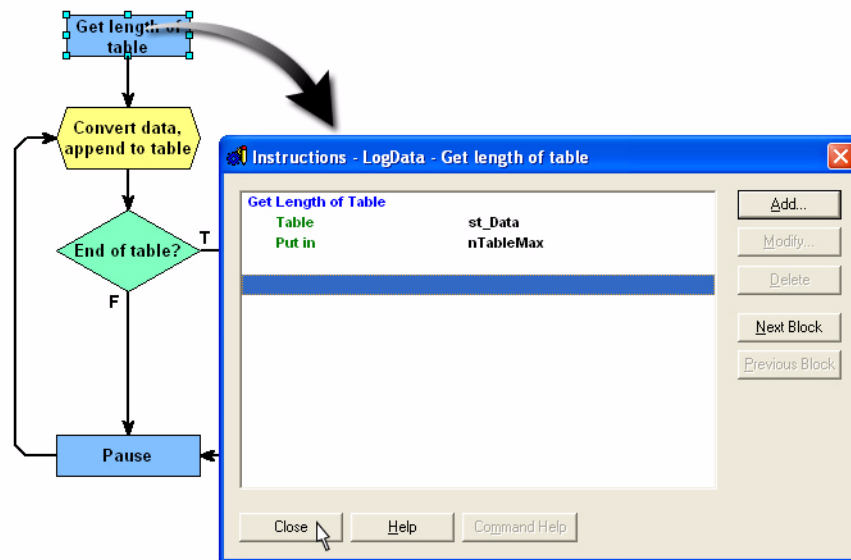
Clicked here to open a list organized by categories

d. Provide the following information:Table: **String Table**, **st_Data**Put in: **Integer32 Variable**, **nTableMax**

NOTE: The integer variable nTableMax has already been created for you.

e. Click OK to close the Add Instruction dialog box.

Your instruction should appear as shown here:



f. Click Close.

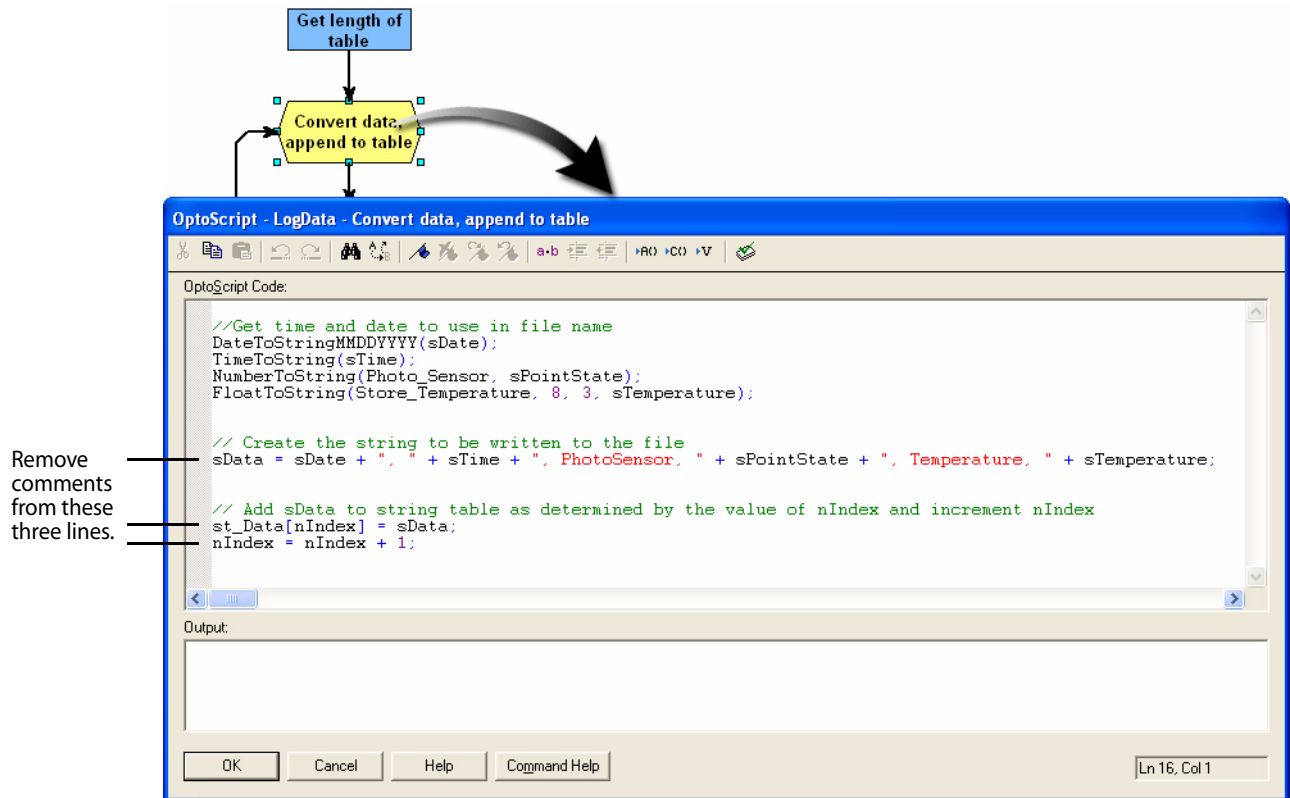
3. Convert the Data to Strings.

Before any data can be written to an ASCII file, the data must be converted to string format. ioControl provides several commands for converting data, several of which are shown in the OptoScript block entitled Convert Data, Create String.

a. Double-click the OptoScript block entitled Convert data, append to table.

The code you need has been typed for you but is commented out.

- b. Remove the comments so that your script appears as shown here:



This script block converts the date, time, digital input point 3 (Photo_Sensor) and analog input point 12 (Store_Temp) to string values.

If you have a SNAP PAC Learning Center, make no other changes to this block. If you are using this strategy with different points, see ["Modifying the Sample Strategy" on page 36](#) for how to modify these lines of OptoScript:

```
NumberToString(Photo_Sensor, sPointState);
FloatToString(Store_Temperature, 8, 3, sTemperature);
```

The following line joins the contents of variables sData, sTime, sPointState, sTemperature, sCRLF into one variable:

```
sData = sDate + ", " + sTime + ", PhotoSensor, " + sPointState + ", Temperature, " + sTemperature;
```

In addition, the optional descriptions PhotoSensor and Temperature are included. Being text, these descriptions appear in red.

- c. If you are using a non-Learning Center configuration, you may wish to edit the red text to describe your points; otherwise, make no changes to the script.
- d. Click **OK** to close the OptoScript editor.

Modifying the Sample Strategy

NOTE: If you do not have a SNAP PAC Learning Center, you will need to modify the strategy so that the sample points cited in the OptoScript match points on your system. Points 3 and 12 of the SNAP PAC Learning Center are used in this example.

OptoScript Block: Convert Data, Create String:

Replace with the name of your digital point or use any integer variable.

↓

NumberToString(PhotoSensor, sPointState);

These string variables will contain the converted integer and float.

↓

FloatToString(Store_Temperature, 8, 3, sTemperature);

↑

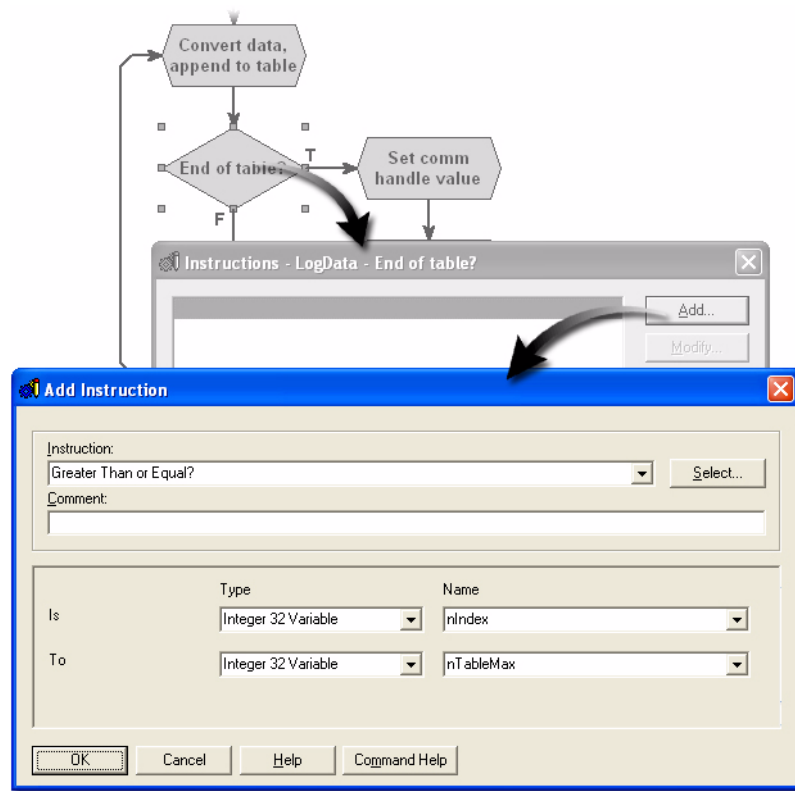
Replace with the name of your analog point or use any existing float variable.

4. Evaluate if the table is full and ready to send to a file.

The previous block increments the variable nIndex each time data is written to the table. The condition block will compare the value of nIndex to the value of nTableMax which was read from the length of the table. The table is full when these two values are equal.

- a. Open the condition block.

- b. Click Add.



- c. In the Add instruction dialog box, provide the following information:

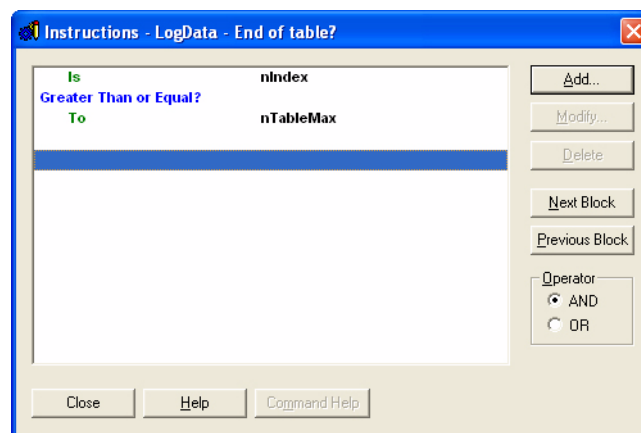
Instruction: Greater Than or Equal

Is: Integer 32 Variable, nIndex

To: Integer 32 Variable, nTableMax

- d. Click OK.

Your instruction should appear as shown here:



- e. Click Close.

5. Create file name.

a. Open the OptoScript editor for the Set comm handle value block.

This script is very similar to the previous screen. Data is converted to strings in the first section. In this case, the values for the Julian day and current month are put in a numeric variable and then converted to a string variable. Then the strings are used to build a string defining a communication handle.

```

// Convert data to strings
nJDay = GetJulianDay();
nMonth = GetMonth();
NumberToString(nJDay, sJDay);
NumberToString(nMonth, sMonth);

// Create the file name and communication handle value
sFileName = "file:a,/" + sMonth + "/DataLog_" + sJDay + ".txt";

SetCommunicationHandleValue ( sFileName , ch_AppendData );
    
```

Remove comments from these two lines

b. Remove the comments so that your code appears as shown above.

c. Click OK to close the OptoScript editor.

The script from the following script block could have been included with this block, but the script was separate to correspond to different tasks.

6. Send string table to file.

a. Open the OptoScript block Send string table to file.

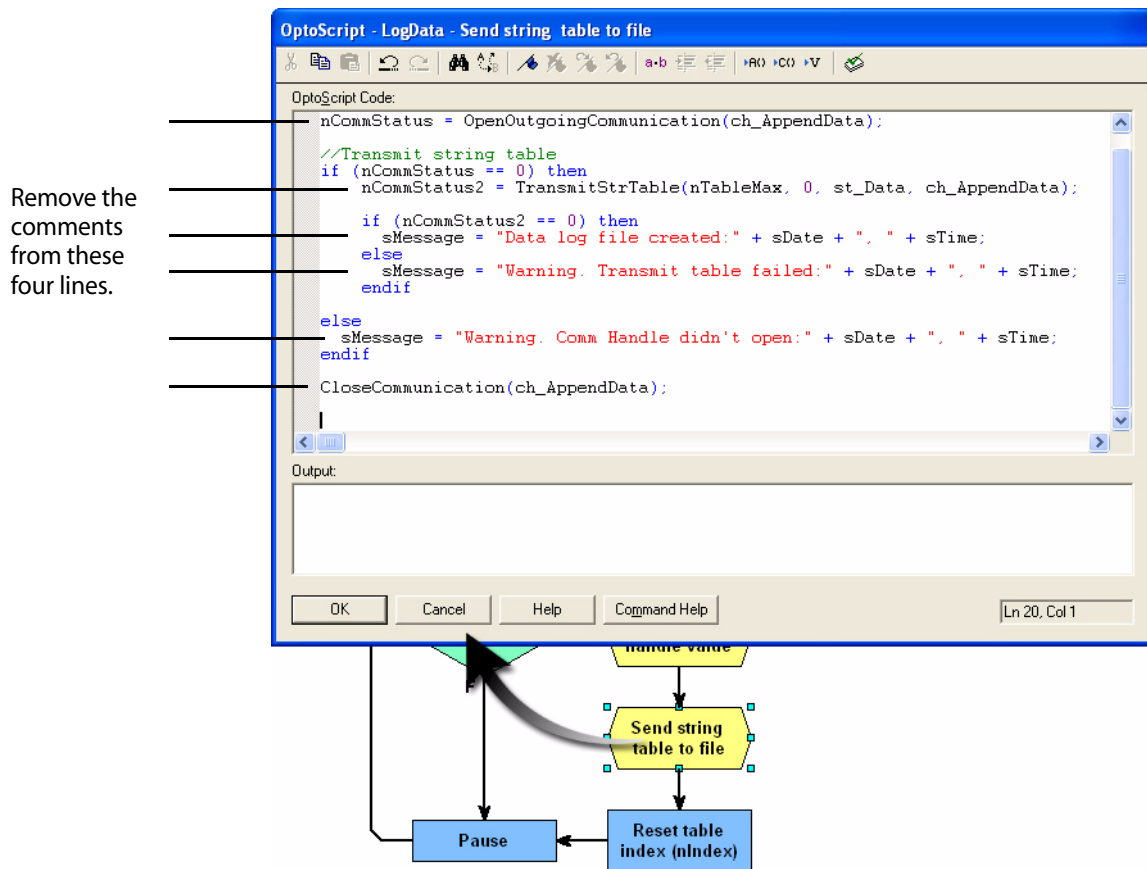
This script opens the Communication Handle defined in the previous block. Then it uses two If statements, one nested within the other, to do the following:

- Check if the Communication Handle opened by testing the value of nCommStatus. If nCommStatus is 0, then the table st_Data has been sent to the communication handle; if not, then a message is written to sMessage and the loop ends.

- If the transmit table command is successful, a message indicating success is written to the variable sMessage.

Last, the communication handle is closed.

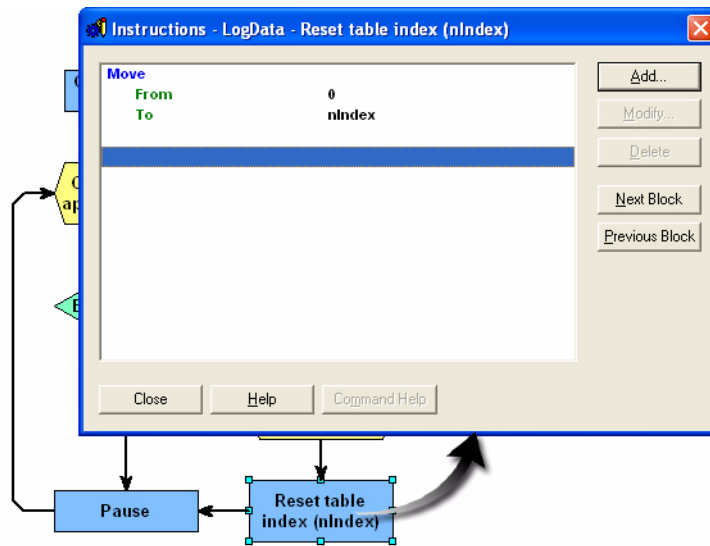
- Remove the comments to enable the script as shown.



7. Reset the table index

- Open the block Reset table index (nIndex).
- Add the following move command
 Command: Move
 From-Type: Integer 32 literal
 From-Name: 0
 To-Type: Integer 32 Variable
 To-Name: nIndex

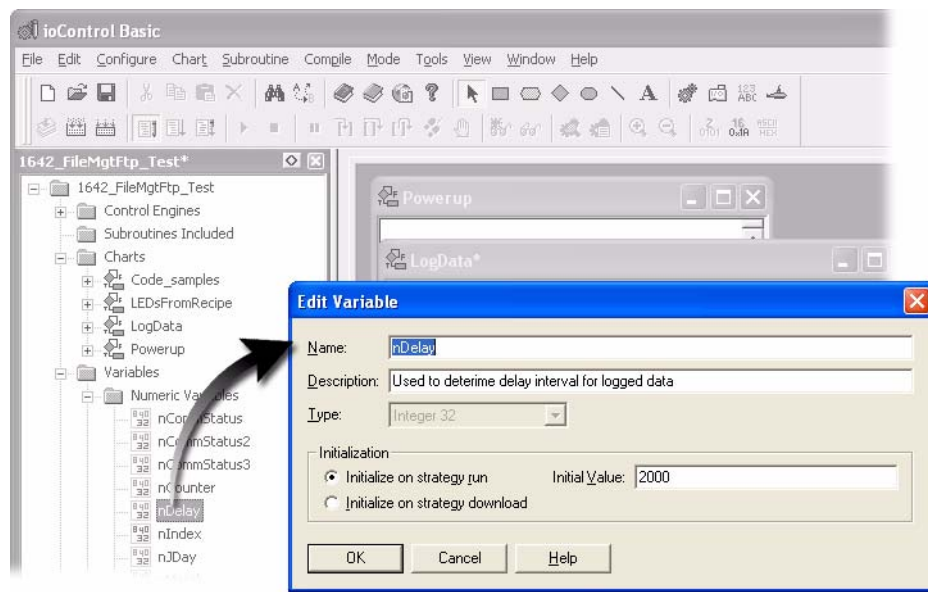
When finished, your instructions should appear as shown here.



Close the Instructions dialog box.

8. Adjust the duration of the pause.

The variable `nDelay` determines how long this chart will pause after writing a data sample to the table `st_Data`. The setting is currently 2000 milliseconds. If you wish to lengthen or shorten this pause, change the initial value for the integer variable `nDelay`.



9. Compile your strategy.

Choose **Compile** → **Compile All: 1642_FileMgtFTP**.

Log data

The strategy creates a text file and appends the date, time, temperature and point state every five seconds.

1. Make sure your SNAP PAC Learning Center is turned on and connected to the network.
2. Download and run the strategy.
 - a. Choose menu command **Mode → Debug**.
 - b. Acknowledge all download messages.
 - c. Choose **Debug → Run**.
3. Allow the strategy to run at least 30 seconds to log data (this assumes you have nDelay at a shorter interval).

While the strategy is running, flip the Photo Sensor switch and hold the temperature probe so the data will vary.

Activity 1b: View the Data Log

Assign an FTP log in and password

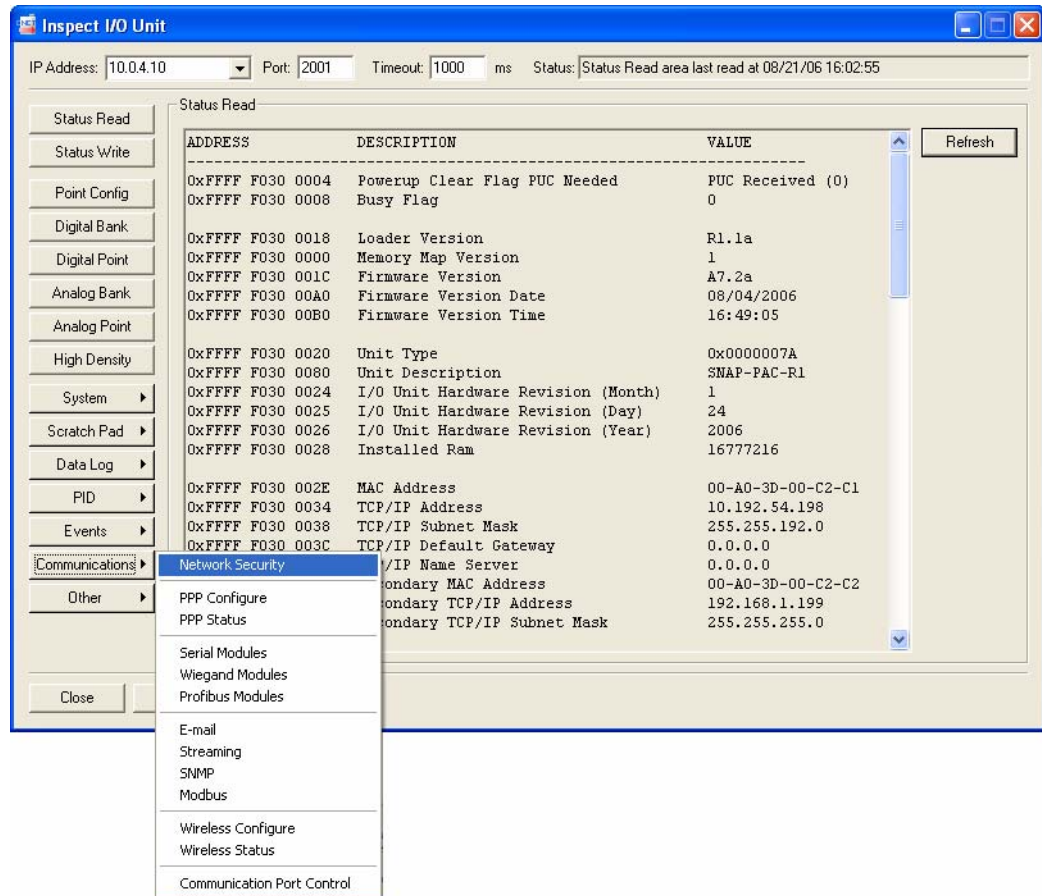
The default setting of the SNAP PAC is to allow anonymous access using any user name and any password. For example, if you log on with an FTP client, you will be prompted for the user name and password. Any user name or password will work, but you are required to supply these. Also, programs like Excel can use the FTP client but cannot answer a prompt for a user name in password. If you wish to use Excel to open the file directly from the SNAP PAC, then do not assign a user name and password.

If you wish to restrict FTP access, you can use I/O manager to assign a user name and password:

1. Open ioManager.
2. Open the Inspect dialog box.

NOTE: Do not assign a user name and password if you wish to use Excel to open the file directly from the SNAP PAC. If you set a password and wish to use Excel, you'll need to use an FTP client to download the file to where Excel can access it.

3. Choose Network Security under the Communications button.



4. Type a user name and click Apply. (The user name is case sensitive.)

The 'Inspect I/O Unit' window displays the 'Network Security' configuration. The status bar indicates 'Network Security area last read at 08/21/06 16:03:46'. The 'FTP Username' field is set to 'SnapPad'.

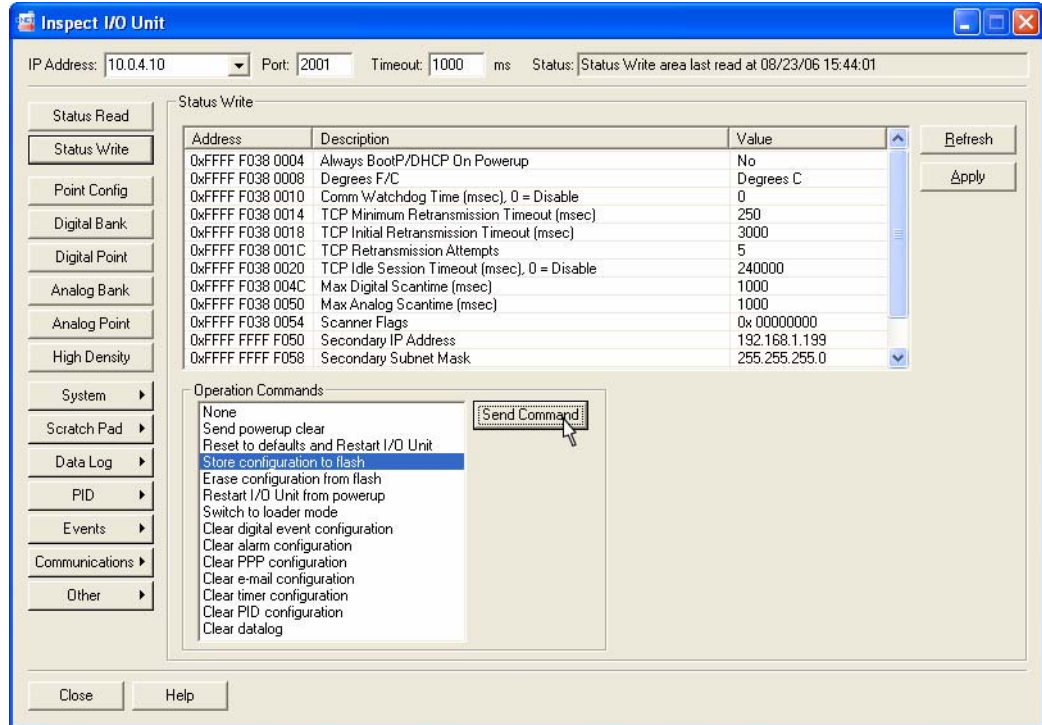
Address	Description	Value
PORTS		
0xFFFF F03A 0004	OptoMMP	2001
0xFFFF F03A 0008	Modbus	502
0xFFFF F03A 000C	SNMP	161
0xFFFF F03A 0010	FTP	21
0xFFFF F03A 0074	Control Engine	22001
FTP LOGIN		
0xFFFF F03D 0000	FTP Username	SnapPad
0xFFFF F03D 0040	FTP Password	
BROADCAST FILTER		
0xFFFF F03A 0070	Stop incoming broadcasts	No
IP FILTERS		
0xFFFF F03A 0020	Filter 0 - Address	0.0.0.0
0xFFFF F03A 0024	Filter 0 - Mask	0.0.0.0
0xFFFF F03A 0028	Filter 1 - Address	0.0.0.0
0xFFFF F03A 002C	Filter 1 - Mask	0.0.0.0
0xFFFF F03A 0030	Filter 2 - Address	0.0.0.0
0xFFFF F03A 0034	Filter 2 - Mask	0.0.0.0
0xFFFF F03A 0038	Filter 3 - Address	0.0.0.0
0xFFFF F03A 003C	Filter 3 - Mask	0.0.0.0
0xFFFF F03A 0040	Filter 4 - Address	0.0.0.0
0xFFFF F03A 0044	Filter 4 - Mask	0.0.0.0
0xFFFF F03A 0048	Filter 5 - Address	0.0.0.0
0xFFFF F03A 004C	Filter 5 - Mask	0.0.0.0
0xFFFF F03A 0050	Filter 6 - Address	0.0.0.0
0xFFFF F03A 0054	Filter 6 - Mask	0.0.0.0
0xFFFF F03A 0058	Filter 7 - Address	0.0.0.0
0xFFFF F03A 005C	Filter 7 - Mask	0.0.0.0

5. Type a password and click Apply. (The password is case sensitive.)

The 'Inspect I/O Unit' window displays the 'Network Security' configuration. The status bar indicates 'Network Security area last written and read at 08/21/06 16:04:23'. The 'FTP Password' field is set to 'snap'.

Address	Description	Value
PORTS		
0xFFFF F03A 0004	OptoMMP	2001
0xFFFF F03A 0008	Modbus	502
0xFFFF F03A 000C	SNMP	161
0xFFFF F03A 0010	FTP	21
0xFFFF F03A 0074	Control Engine	22001
FTP LOGIN		
0xFFFF F03D 0000	FTP Username	SnapPac
0xFFFF F03D 0040	FTP Password	snap
BROADCAST FILTER		
0xFFFF F03A 0070	Stop incoming broadcasts	No
IP FILTERS		
0xFFFF F03A 0020	Filter 0 - Address	0.0.0.0
0xFFFF F03A 0024	Filter 0 - Mask	0.0.0.0
0xFFFF F03A 0028	Filter 1 - Address	0.0.0.0
0xFFFF F03A 002C	Filter 1 - Mask	0.0.0.0
0xFFFF F03A 0030	Filter 2 - Address	0.0.0.0
0xFFFF F03A 0034	Filter 2 - Mask	0.0.0.0
0xFFFF F03A 0038	Filter 3 - Address	0.0.0.0
0xFFFF F03A 003C	Filter 3 - Mask	0.0.0.0
0xFFFF F03A 0040	Filter 4 - Address	0.0.0.0
0xFFFF F03A 0044	Filter 4 - Mask	0.0.0.0
0xFFFF F03A 0048	Filter 5 - Address	0.0.0.0
0xFFFF F03A 004C	Filter 5 - Mask	0.0.0.0
0xFFFF F03A 0050	Filter 6 - Address	0.0.0.0
0xFFFF F03A 0054	Filter 6 - Mask	0.0.0.0
0xFFFF F03A 0058	Filter 7 - Address	0.0.0.0
0xFFFF F03A 005C	Filter 7 - Mask	0.0.0.0

6. Write configuration to Flash.
 - a. Click the Status Write button.
 - b. On the Status Write page, under the Operation Commands, choose Store configuration to Flash.



- c. Click Send Command.
 - d. Close the Inspect dialog box.
7. Exit ioManager.

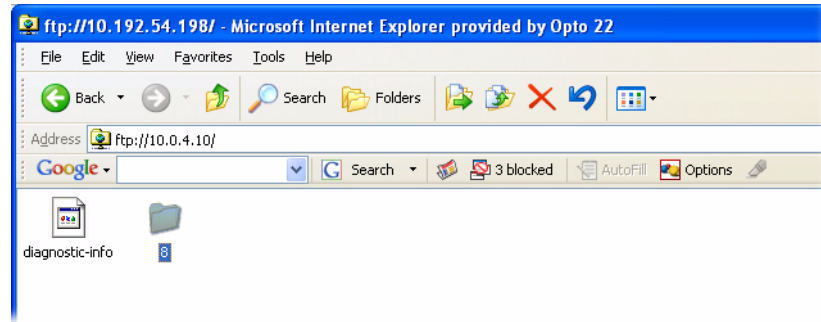
An FTP session

The data logs created on the SNAP PAC can easily be opened using any FTP client application, such as Internet Explorer or Notepad. Both of these applications use the underlying FTP application provided with Windows XP. The following shows how to open the log file in Internet Explorer and how to use the Windows XP FTP client.

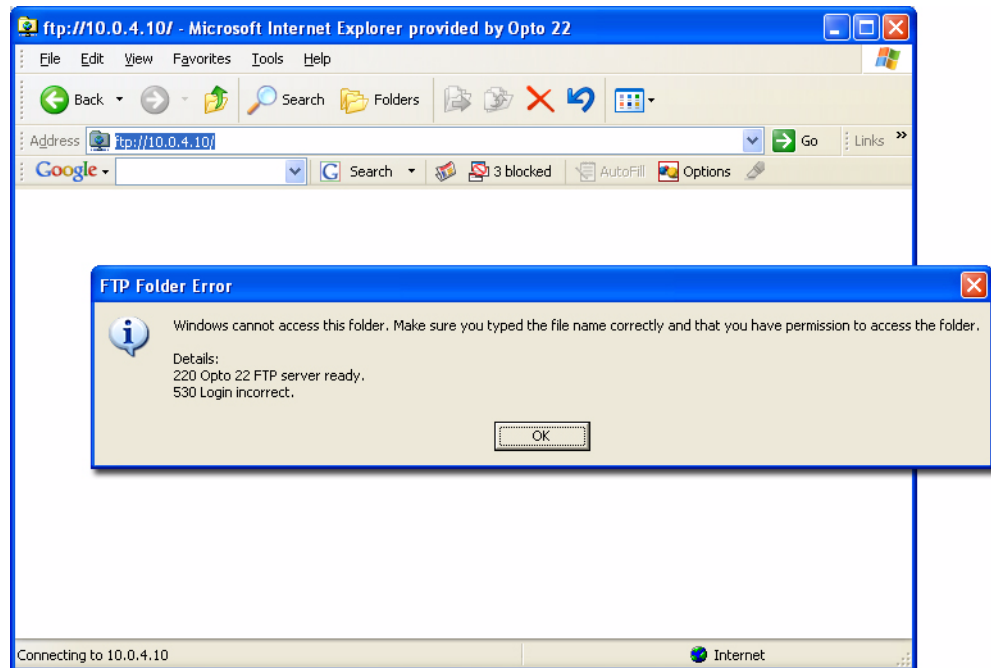
Using Internet Explorer to open a file on the SNAP PAC

1. Open Internet Explorer and in the address field type **FTP: //** followed by the IP address of your SNAP PAC, and then press Enter.

If you did not assign a password, you'll see the directory containing your log file on the SNAP PAC. Go to Step 2. (Note: The directory name will vary according to the month of the year.)

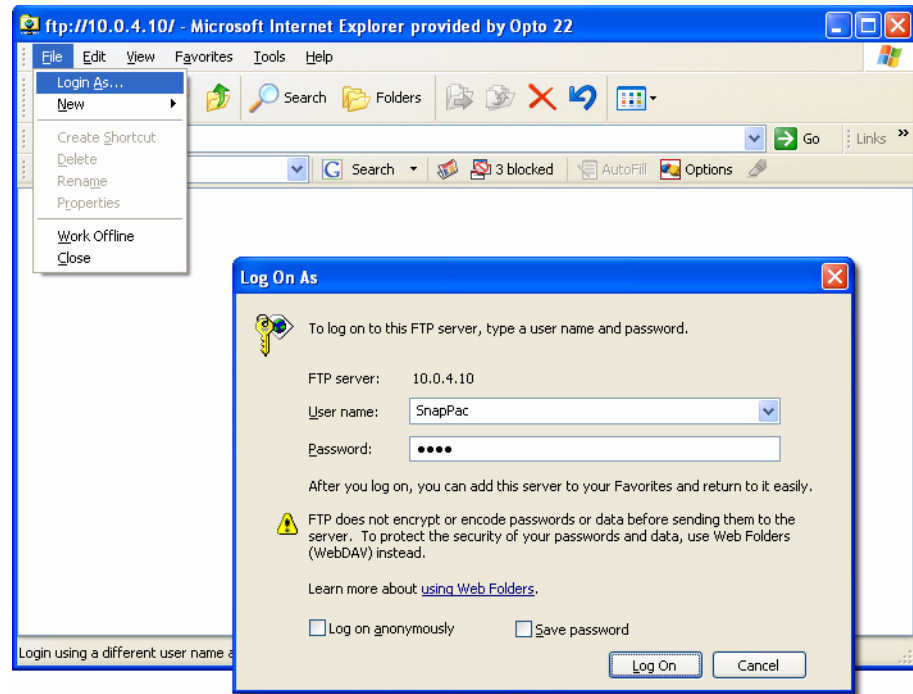


If you assigned a user name and password to the SNAP PAC, you will need to login.:



a. Click OK.

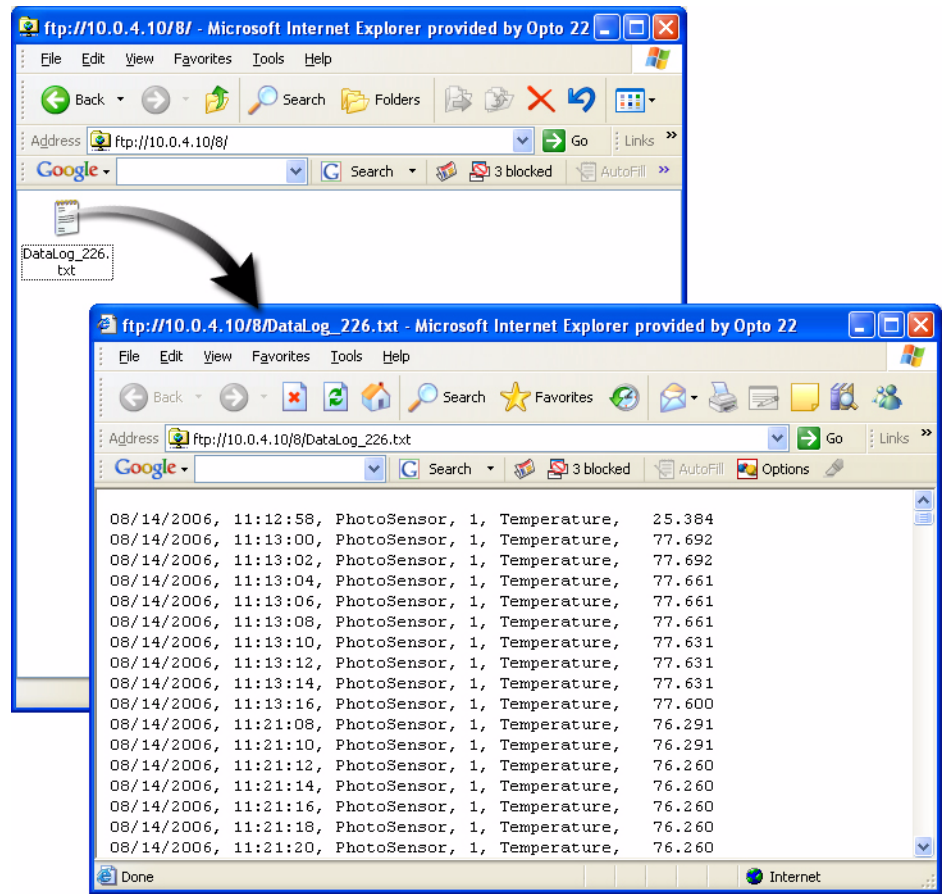
- b. Choose Login As under the File menu:



- c. Type the user name and password that you assigned to your SNAP PAC and click Log On. Internet Explore displays the files on the SNAP PAC.
2. Open DataLog
- a. Double-click the directory containing your log file.

Note: The directory name will vary according to the current month of the year.

- b. Double-click your log file.



The data obtained through Internet Explorer represents the file at the time you opened DataLog.txt. You'll need to refresh the view of the directory to open the current file.

Using the Windows XP FTP client

1. Open a DOS command.
 - a. From the Start menu, choose Run.
 - b. Type `cmd` and click OK.
2. Navigate to the directory to be used as a store for the downloaded data. The following shows how to make and move to a new directory:
 - a. Type `cd\` and press Enter to get to the root directory
 - b. Type `md [di rectory name]` to create a directory
 - c. Type `cd [di rectory name]`, e.g.,

`cd FtpClient`

This will put the cursor in a directory named FtpClient. When you run the FTP client, it will default to this directory when downloading a file.

3. Type `ftp` and press Enter to start the FTP client.

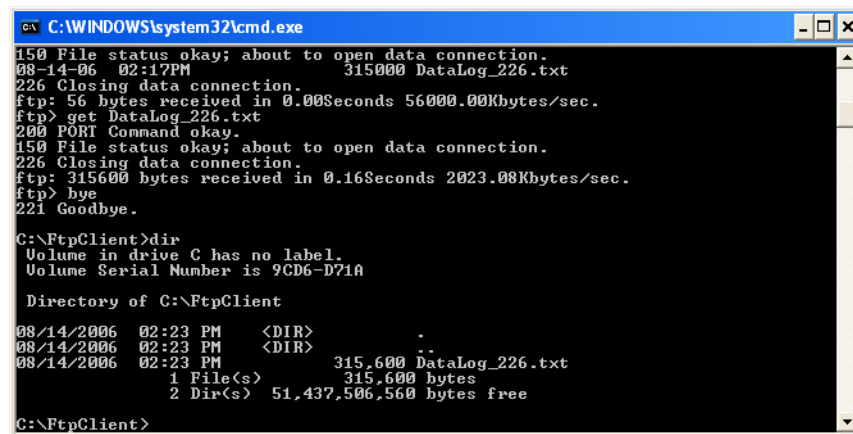
4. Connect to the SNAP PAC FTP Server.
 - a. Type `open` [IP address of Snap PAC], for example
`open 10.0.4.10` (and press enter)
 You will be prompted for a User Name
 - b. Type a user name. If you configured a user name in ioManager, then type that User Name. If you didn't configure a user name, you must still provide a user name but it can be any name, for example:
`SnapPac` (and press Enter)
 - c. Type a password. Type either the password you configured in ioManager, or any letter if you did not configure a password. You must type something, for example:
`myPassword` (and press Enter)
 If successful, you'll see a message stating "User Logged in, proceed."
5. Type `dir` and press Enter to see a directory of files on the SNAP PAC.

```

C:\>cd ftpclient
C:\FtpClient>ftp
ftp> open 10.0.4.10.
Connected to 10.0.4.10.1
220 Opto 22 FTP server ready.
User (10.0.4.10.1:(none)): snap
331 Please specify the password.
Password:
230 User logged in, proceed.
ftp> dir
200 PORT Command okay.
150 File status okay; about to open data connection.
08-14-06 11:21AM <DIR> 8
226 Closing data connection.
ftp: 42 bytes received in 0.00Seconds 42000.00Kbytes/sec.
ftp> cd 8
250 CWD requested file action okay, completed.
ftp> dir
200 PORT Command okay.
150 File status okay; about to open data connection.
08-14-06 01:48PM 262200 DataLog_226.txt
226 Closing data connection.
ftp: 56 bytes received in 0.00Seconds 56000.00Kbytes/sec.
ftp>
  
```

6. Type `CD /8` to move to the directory with your data log.
7. Type `dir` and press Enter. This displays the datalog file.
8. Download the file by typing the following:
`Get DataLog_226.txt` and press Enter.
 (Note: your file name will reflect the date and is likely to have a different 3-digit number representing the Julian date)
9. Type `bye` and press Enter to exit the FTP client.

Your file should be in the FtpClient directory.



```

C:\WINDOWS\system32\cmd.exe
150 File status okay; about to open data connection.
08-14-06 02:17PM 315000 DataLog_226.txt
226 Closing data connection.
ftp: 56 bytes received in 0.00Seconds 56000.00Kbytes/sec.
ftp> get DataLog_226.txt
200 PORT Command okay.
150 File status okay; about to open data connection.
226 Closing data connection.
ftp: 315600 bytes received in 0.16Seconds 2023.08Kbytes/sec.
ftp> bye
221 Goodbye.

C:\FtpClient>dir
Volume in drive C has no label.
Volume Serial Number is 9CD6-D71A

Directory of C:\FtpClient

08/14/2006 02:23 PM <DIR> .
08/14/2006 02:23 PM <DIR> ..
08/14/2006 02:23 PM 315,600 DataLog_226.txt
1 File(s) 315,600 bytes
2 Dir(s) 51,437,506,560 bytes free

C:\FtpClient>

```

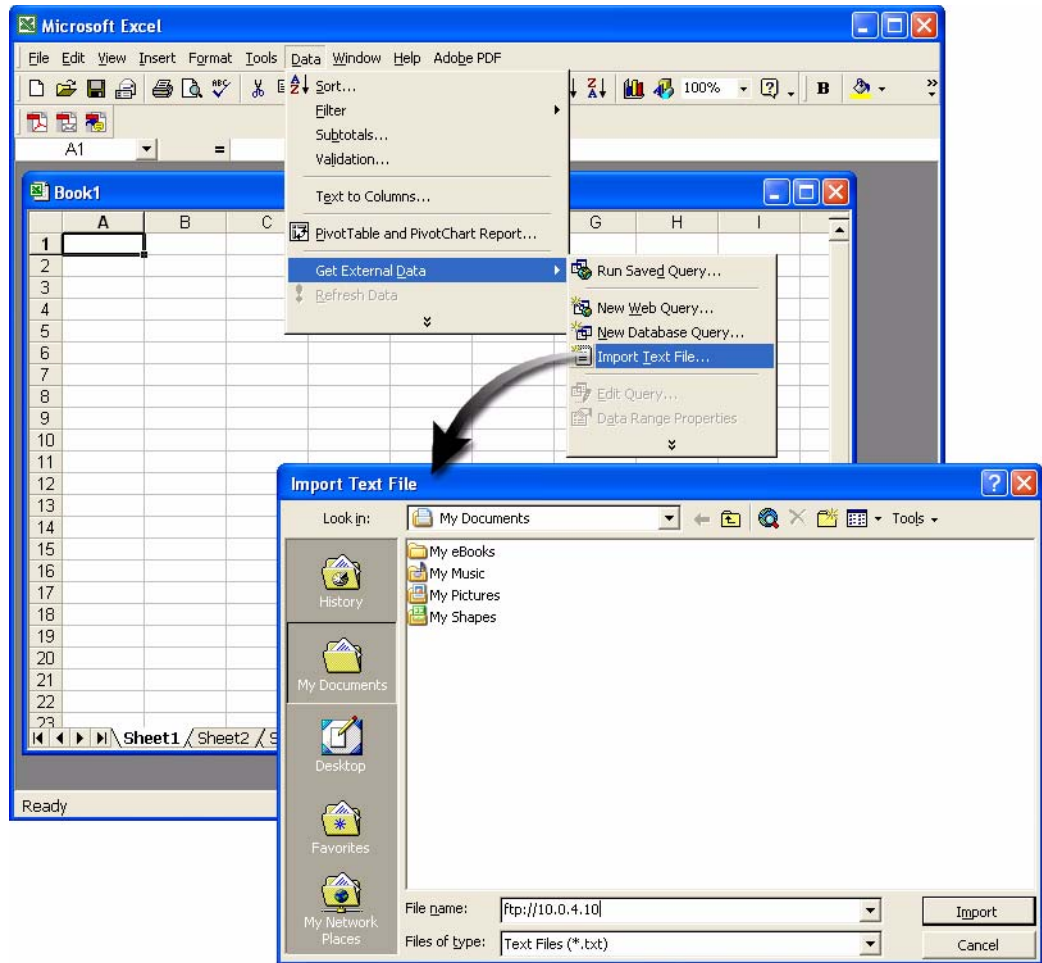
Using Excel

To open your log file in Excel, you will need to satisfy one of the following requirements:

- The SNAP PAC should not require a user name and password; Excel can get a file from an FTP server as long as a specific user name and password are not required.
- Your data log is copied to your PC.

1. Open Excel
2. Create a new book if one is not already open (you must have a new book to import a file)

3. Choose Data → Get External Data → Import Text File.

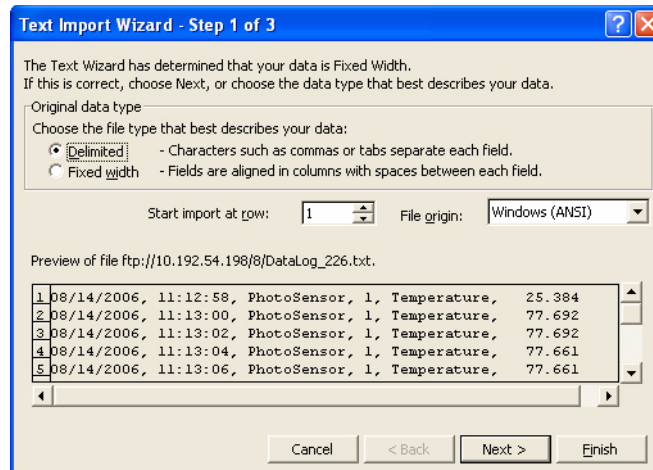


Do one of the following to open your file:

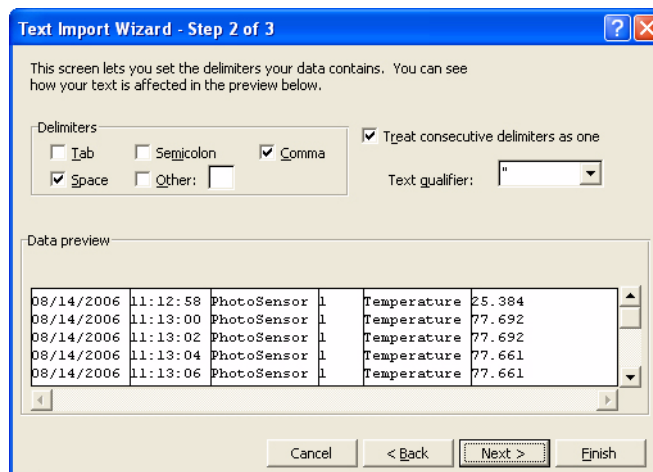
- if you have not assigned an FTP user name and password, you can open the file directly using FTP. Type the following:
ftp: //10. 0. 4. 10 and press Enter
 (Substitute **10. 0. 4. 10** with the IP address of your SNAP PAC.)
 Double-click the subdirectory.
 Double-click your data log file.
- If you have downloaded your data log to a location on your computer, such as the FtpClient directory described under Using the Windows XP FTP client, navigate to the directory where your file is stored and open the file.

After completing either of these steps, you will see the Excel text import wizard.

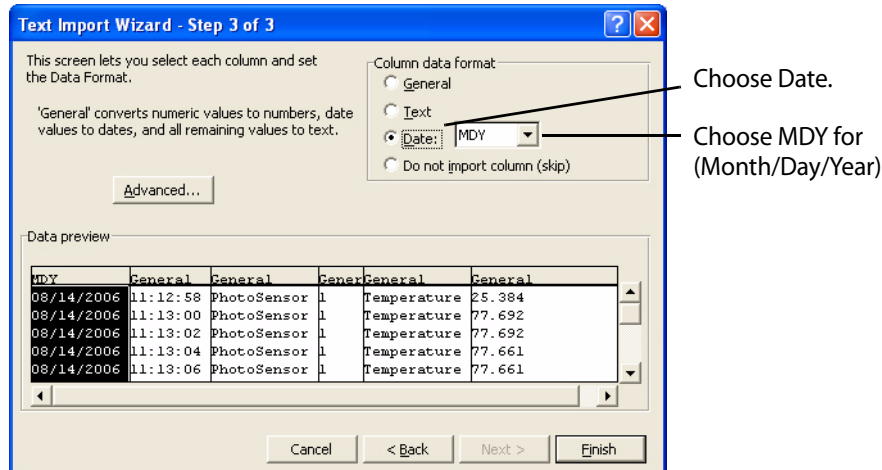
4. In the Text Import Wizard, step 1, select Delimited under Original data type and click Next:



5. In the Text Import Wizard, step 2, select the Comma and Space delimiter. Remove the Tab delimiter. Click Next.

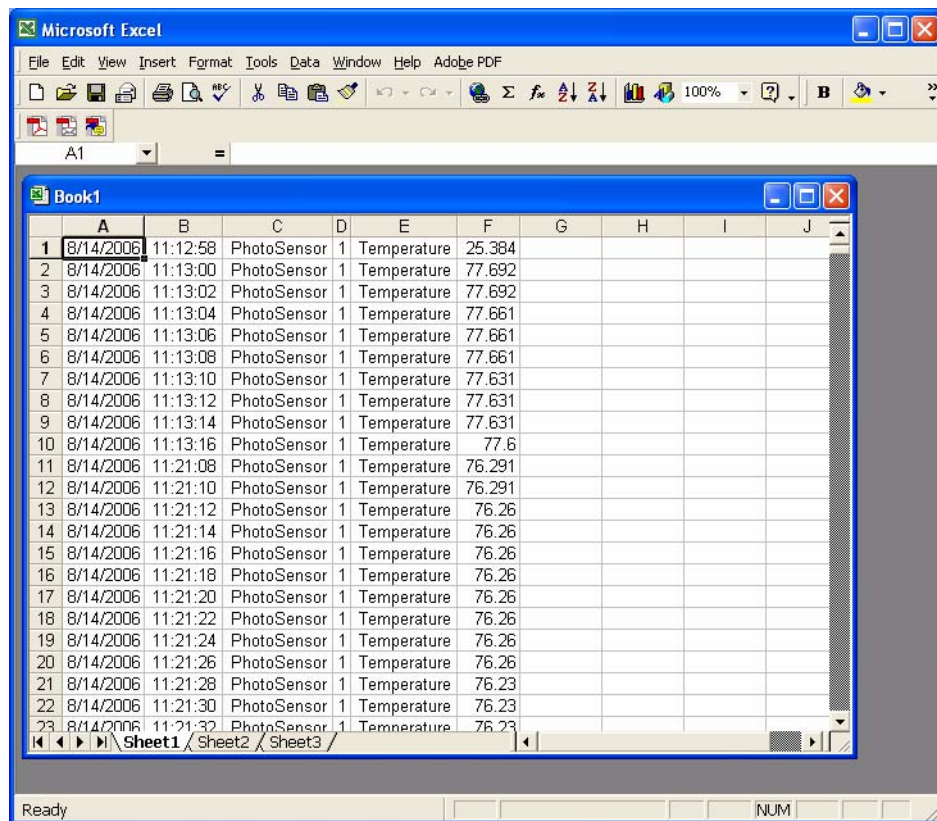


- In the Text Import Wizard, step 3, modify the data type of any fields. For example, you can select the first column, which are dates, and select the Date under Column data format



Click Finish.

- In the Import Data dialog box, choose either Existing Worksheet or New Worksheet, and click OK.



Sending Files to SNAP PACs

Lesson 2 Overview

This lesson shows how to use ioControl to open and parse a file uploaded to it from an FTP client. In the previous lesson you stored data as a file and then retrieved the data using an FTP client. As with lesson 1, this lesson uses the SNAP PAC FTP server, but this time, instead of sending data, the FTP client will send data in a file to the SNAP PAC and the SNAP PAC will parse this file for instructions.

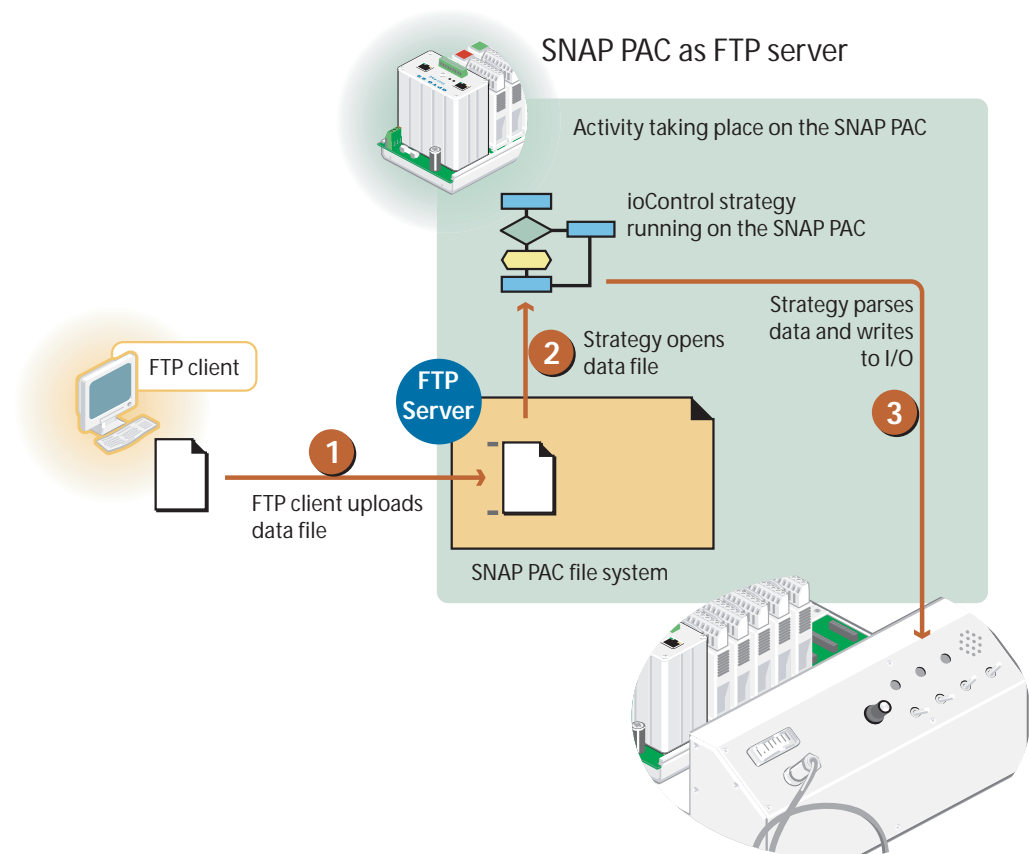


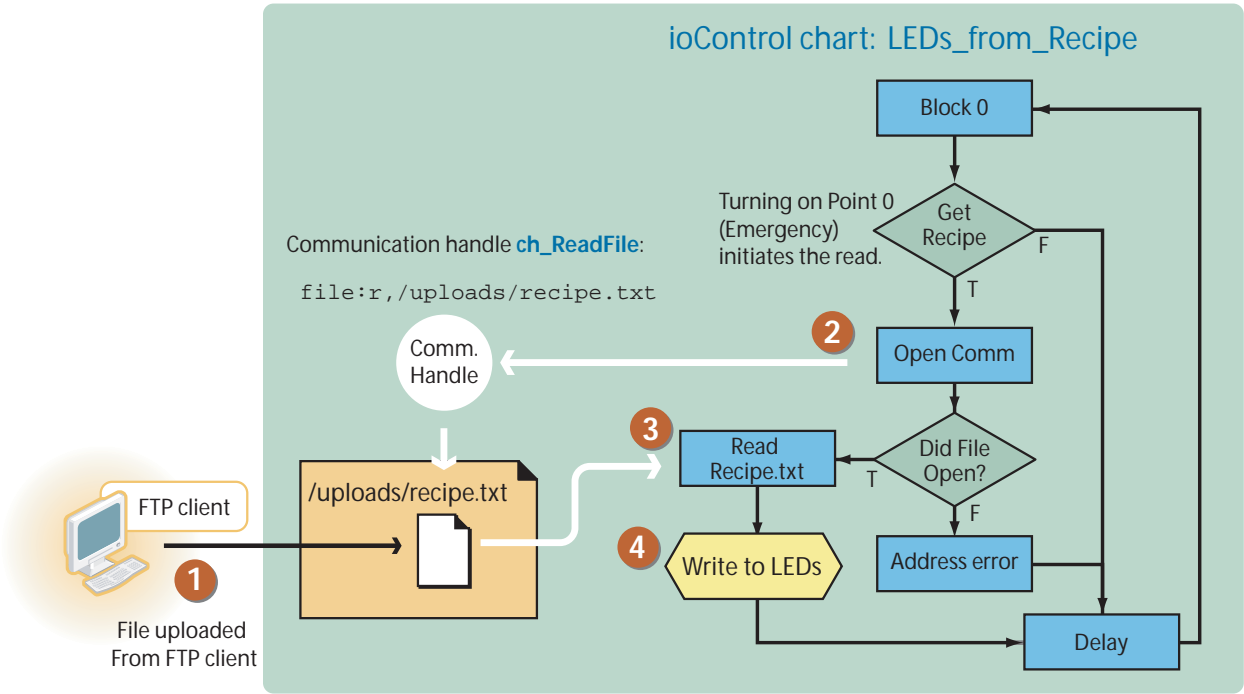
Figure 10: Path of recipe information in lesson 2

1	A comma-delimited text file is created on the workstation using Notepad (or any application capable of creating ascii text files). An FTP client uploads the file to the SNAP PAC.
2	A file communication handle defines the file to open. This communication handle expects the find recipe.txt. When successful, ioControl opens and parses the ASCII file.
3	The values contained in recipe.txt are converted to integers and then written to digital output points 5 (Outside_Light), 6 (Inside_Light), and 7 (Freezer_Door_Status).

Concepts

How the sample chart works

The trigger by which the SNAP PAC reads the file uploaded to it could work a variety of ways. The chart LEDsFromRecipe will look for the file Recipe.txt when the emergency button is pressed and help down. This is to ensure that you are watching while the file is read and its values are written to the LEDs (digital output points 4, 5, and 6). The diagram below describes the key functions this chart will perform when you complete this lesson



1	A text file containing three binary values (0 or 1), each followed by a comma delimiter, is loaded to the controller using the controller's FTP server.
2	When the Emergency button is held down, ioControl sets the value of the Communication Handle, establishes communication, and sets the end-of-message terminator to a comma.
3	The Communication Handle buffer contains the contents of the file. A receive string table command puts the contents into a string table using the end-of-message terminator to separate table elements.
4	The values in the string table are converted to integers and then moved to the digital output points.

Parsing a text file

There are a variety of methods to parse a text file. All begin with opening a communication handle configured to read a file. Once read, or opened, the file's contents reside in the communication handle buffer where the text can be moved to a string or string table, as summarized below.

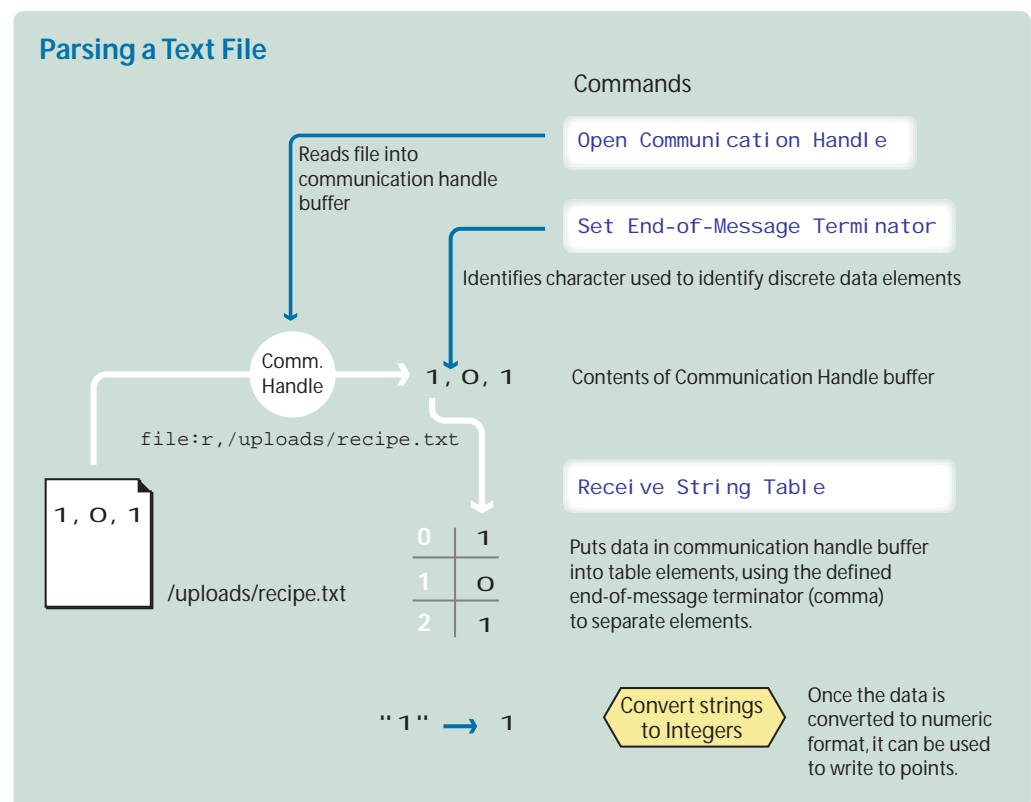


Figure 11: Steps in converting a text file to numbers used to control points

The OptoScript block entitled Convert Strings to Integers shows the efficiency of OptoScript:

```
for nCounter = 0 to 2 step 1
```

```
nt_recipe[nCounter] =  
StringToFloat(st_recipe[nCounter]);  
next
```

This script will take the first three elements of the string table and convert them to elements in a numeric table. Then the elements of the numeric table can be written to digital points, for example:

Write table element 0 to the Outside light:

```
Outside_Light = nt_recipe[0];
```

Write table element 1 to the Inside light:

```
Inside_Light = nt_recipe[1];
```

Write table element 2 to the Freezer Door status light:

```
Freezer_Door_Status = nt_recipe[2];
```

Communication Handles for reading files

A Communication Handle is used to read a file from the controller. The Communication Handle contains the `file:` and `r` parameters to indicate the read function, followed by the file name. If the file resides in a subdirectory, the subdirectories are included in the file name. For example:

```
file:r,recipe.txt
```

```
file:r,/directory/recipe.txt
```

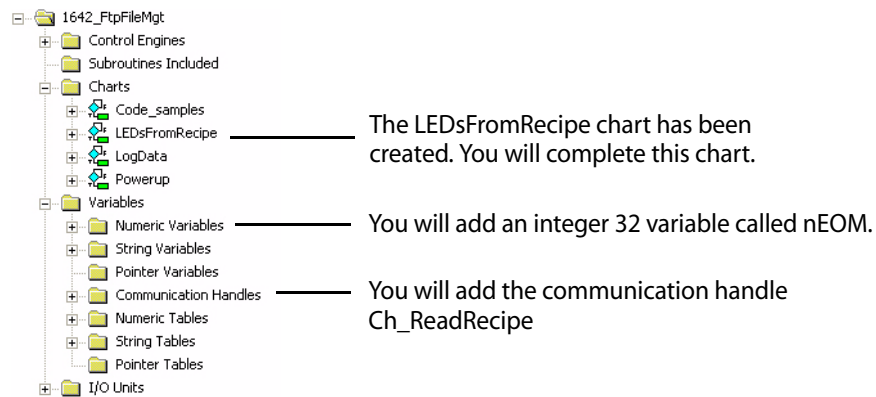
When using a read Communication Handle:

- Pay careful attention to case; the Communication Handle is case-sensitive.
- Set End-of-Message (EOM) Terminator after the Communication Handle is opened. The character is represented by an ASCII value (see the ASCII table under “String Commands” in Chapter 10 of the *ioControl User's Guide*, form #1300). Common EOMs include a comma (character 44) and a colon (character 58). The default EOM is a carriage return (character 13). In this example, where a common text editing application creates the file, the carriage return is not recommended as an end-of-message terminator, as programs like Notepad will insert two characters when you press the Enter key: a carriage return and a line feed.
- Use the correct character for directories. For example, use `/` instead of `\`.
- Close communication when finished.

Activity 2a: Uploading and Parsing Files

In this activity you will do the following:

- Add numeric variable `nEOM`
- Add a communication handle `ch_readFile`
- Revise the chart entitled `LEDs_From_Recipe`



Strategy tree

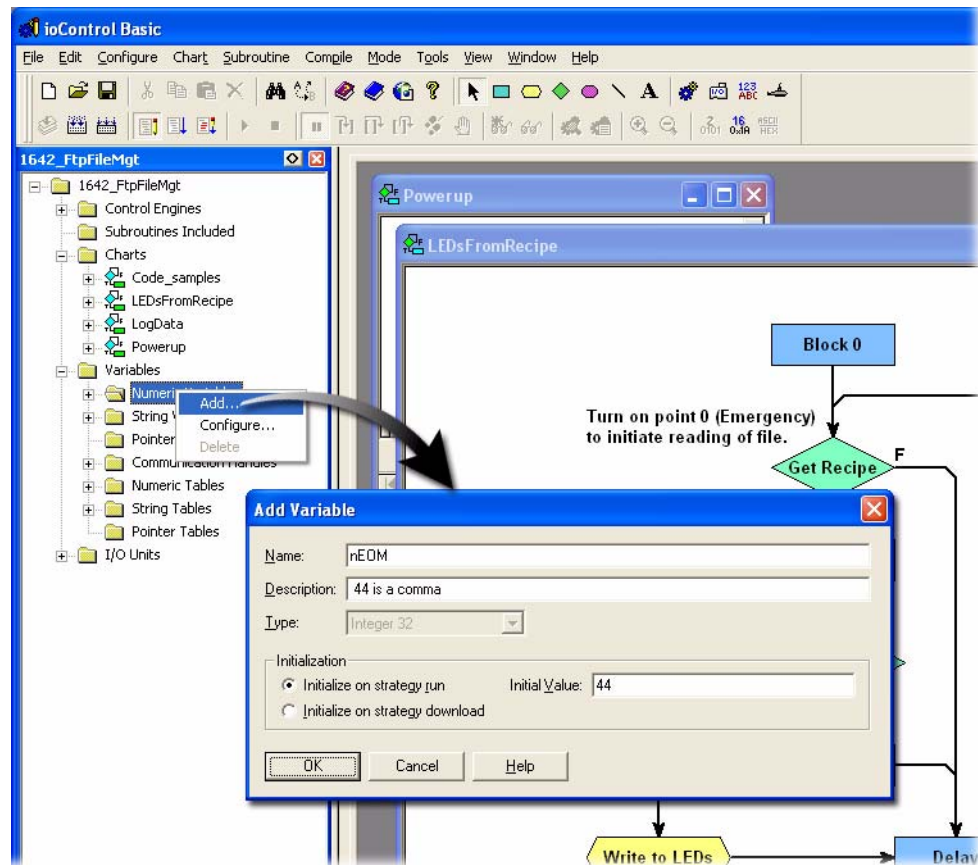
1. Make sure you are in Configure mode.
2. In the Strategy Tree, expand the variables folder.
3. Create a variable to contain an end-of-message terminator.
 - a. Right-click the Numeric Variables folder.
 - b. Choose Add.

- c. Type the following information:

Name: nEOM

Description: 44 is a comma

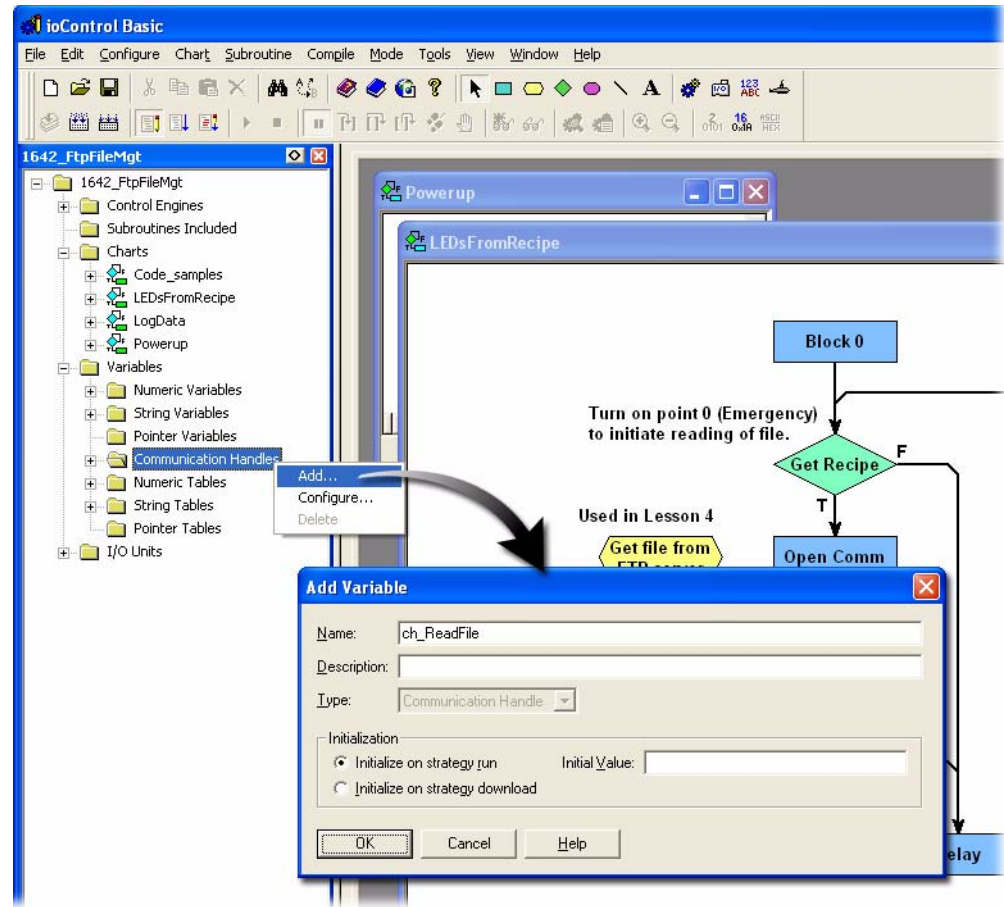
Initial Value: 44



- d. Click OK.

4. Add a communication handle.

- a. Right click the Communication Handles folder

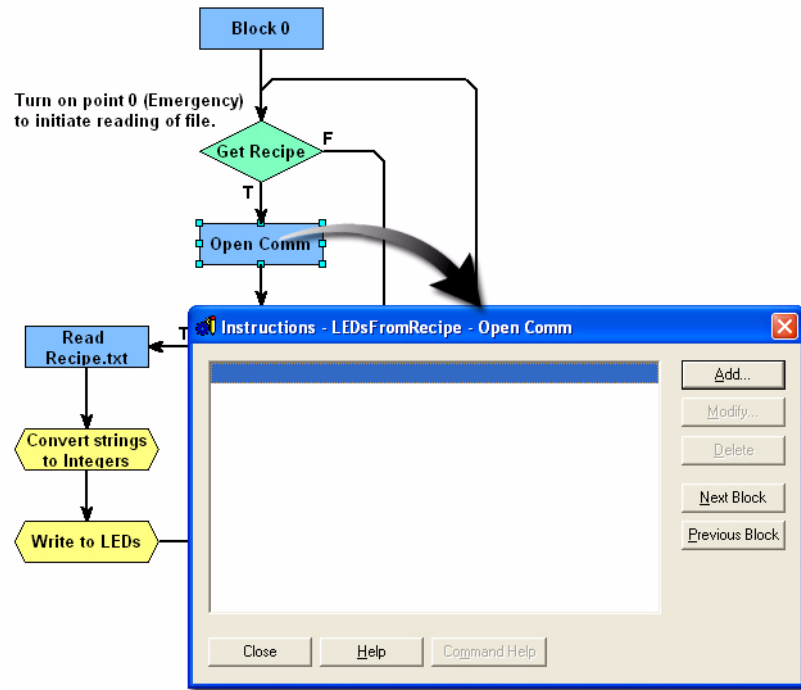


- b. Type `ch_ReadFile` in the name field.
You do not need to provide an initial value.
- c. Click OK.

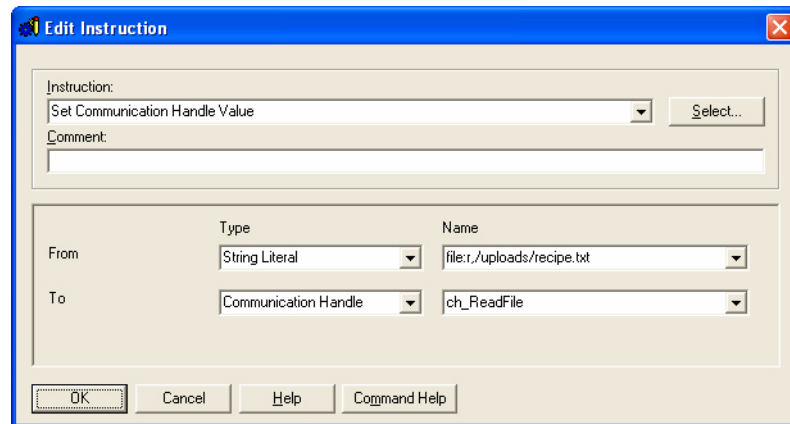
Chart: LEDs from recipe

1. Open the chart LEDsfromRecipe, if it is not already open.

2. Double-click the Open Comm block.

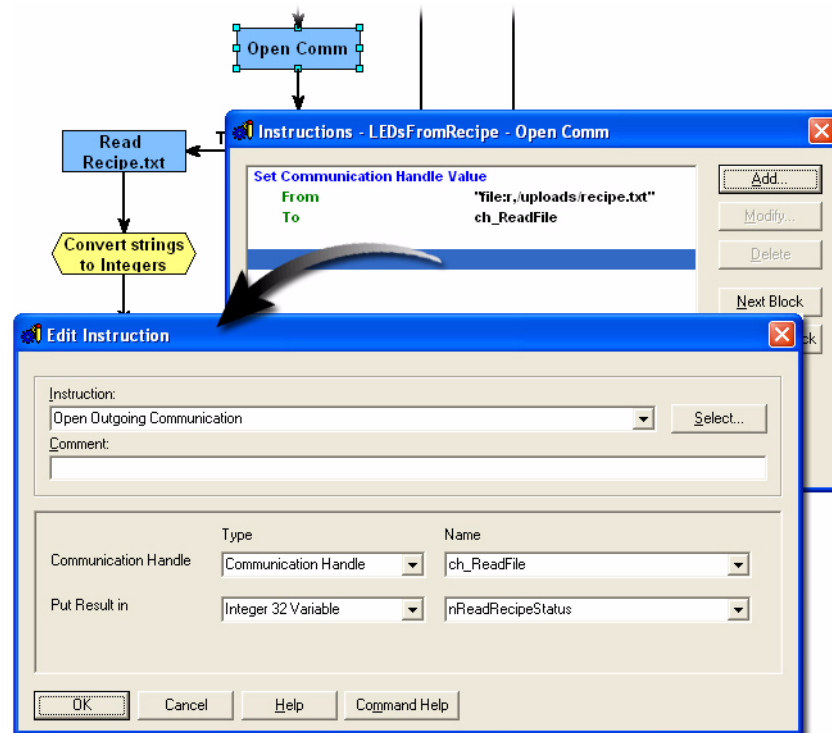


3. Add a Set Communication Handle Value command:
 - a. Click Add.
 - b. In the Edit Instruction dialog box enter the following:
 Instruction: Set Communication Handle Value
 From-Type: String Literal
 From-Name: file:r,/uploads/recipe.txt
 To-Type: Communication Handle
 To-Name: ch_ReadFile

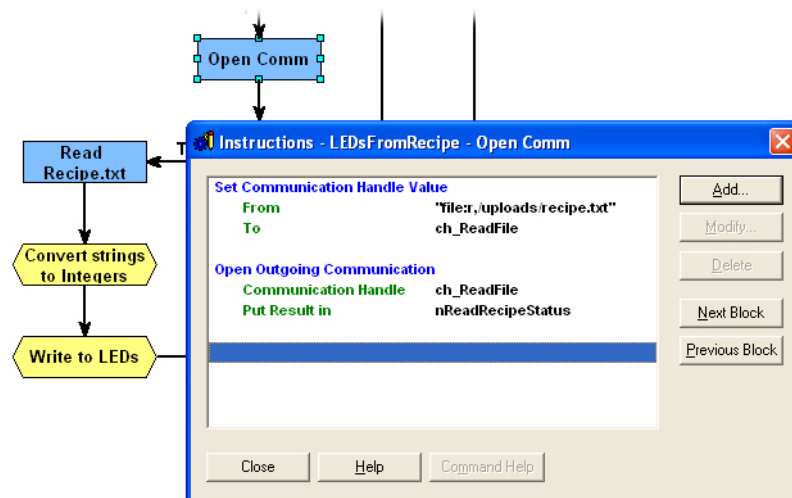


- c. Click OK.
4. Open outgoing communication.

- a. In the Instructions dialog box, double-click the space below the first instruction (or click once and click Add).



- b. In the Edit Instruction dialog box, enter the following (as shown above):
 Instruction: Open Outgoing Communication
 Communication Handle: ch_ReadFile
 Put Result in—Type: Integer 32 Variable
 Put Result in—Name: nReadRecipeStatus
- c. Click OK. Your Instructions should appear as shown here:



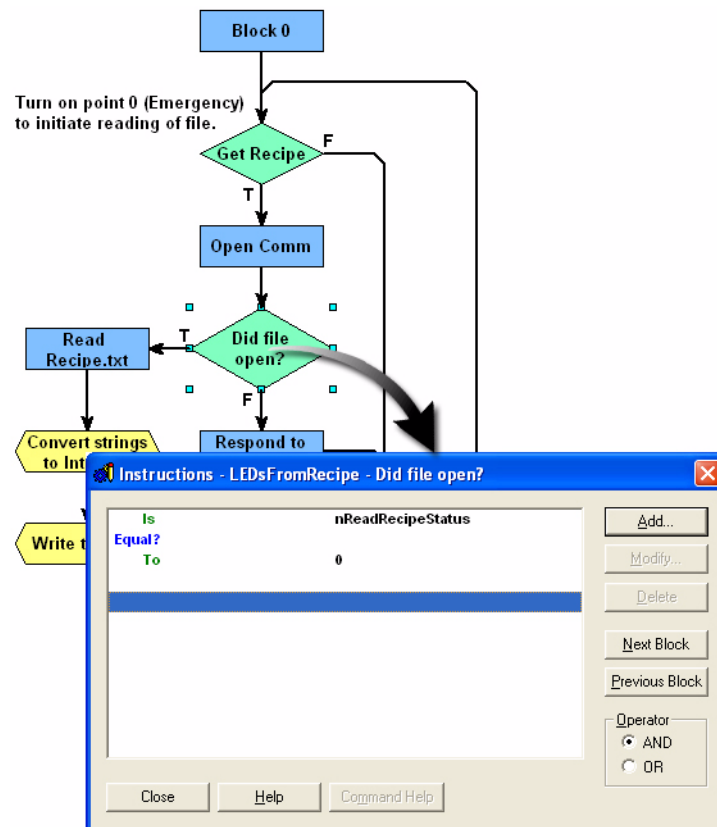
- d. Click Close.

5. Check if Recipe.txt opens.

When the Communication Handle is opened, a value is put in nReadRecipeStatus indicating success of failure. If nReadRecipeStatus is 0, then the file has been opened.

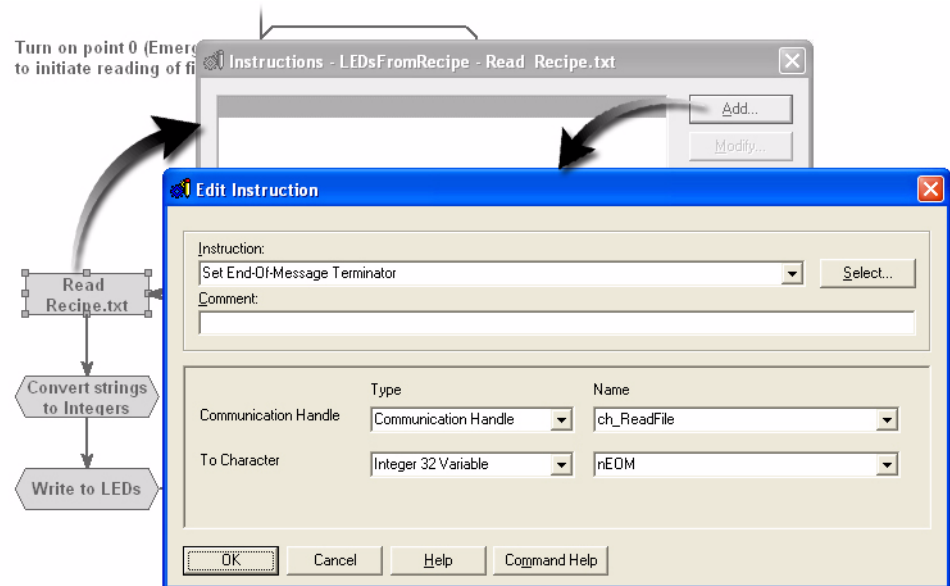
- a. Double click the condition block Did File Open?
- b. Click Add.
- c. In the Instructions dialog box, enter the following:
 Instruction: Equal?
 Is-Type: Integer 32 Variable
 Is-Name: nReadRecipeStatus
 To-Type: Integer 32 Literal
 To-Name: 0
- d. Click OK.

Your condition block should appear as shown here:



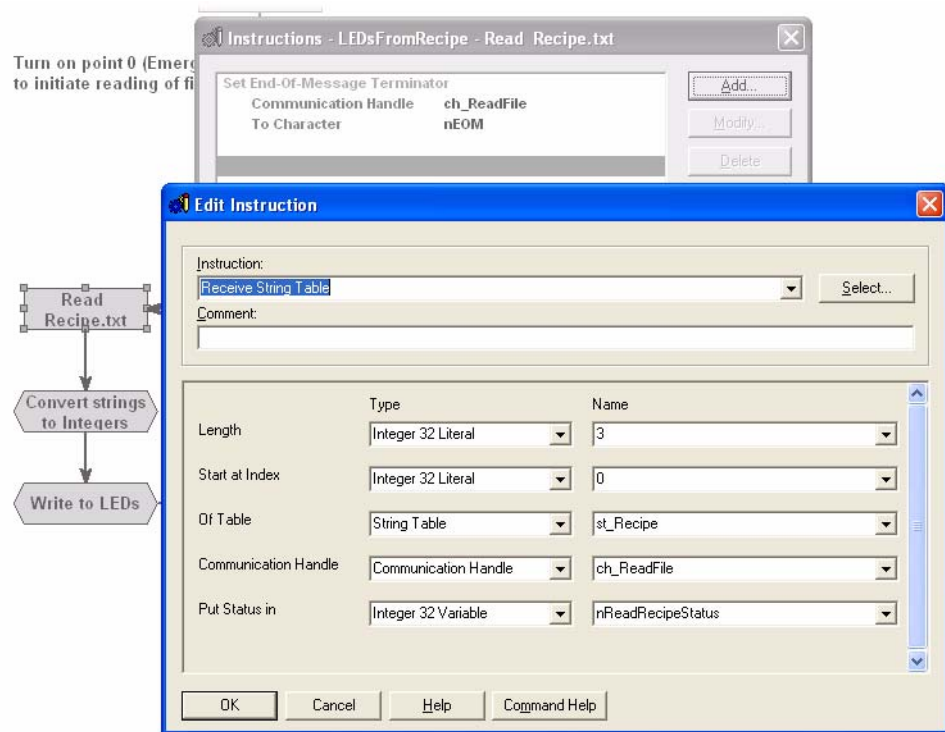
- e. Click Close.

6. Read Recipe.txt
 - a. Double-click the instruction block Read Recipe.txt
 - b. In the Instructions dialog box, click Add.



- c. In the Add Instruction dialog box, provide the following information:
 Instruction: Set End-of-Message Terminator
 Communication Handle: ch_ReadFile
 To Character: Integer 32 Variable-nEOM
 - d. Click **OK**.
 - e. Leave the Instructions dialog box open.
7. Receive string table.
 - a. In the Instructions dialog box, select below the Set End-of-Message Terminator instruction.

- b. Click Add.



- c. Select the following:

Instruction: Set End-of-Message Terminator

Length: Integer 32 Literal - 3

Start at Index: Integer 32 Literal - 0

Of Table: String Table - st_Recipe

Communication Handle: ch_ReadFile

Put Status in: nReadRecipeStatus

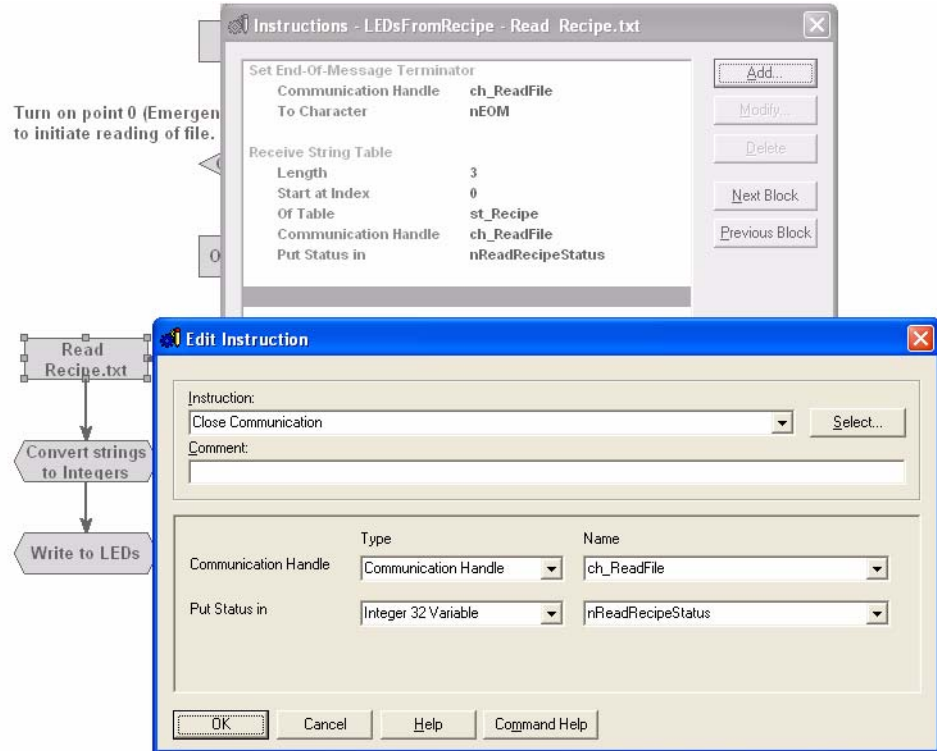
(NOTE: You may need to scroll the window to see all the options.)

- d. Click OK.

8. Close communication.

- a. In the Instructions dialog box, select below the Receive String Table instruction.

b. Click Add.



c. Select the following information:

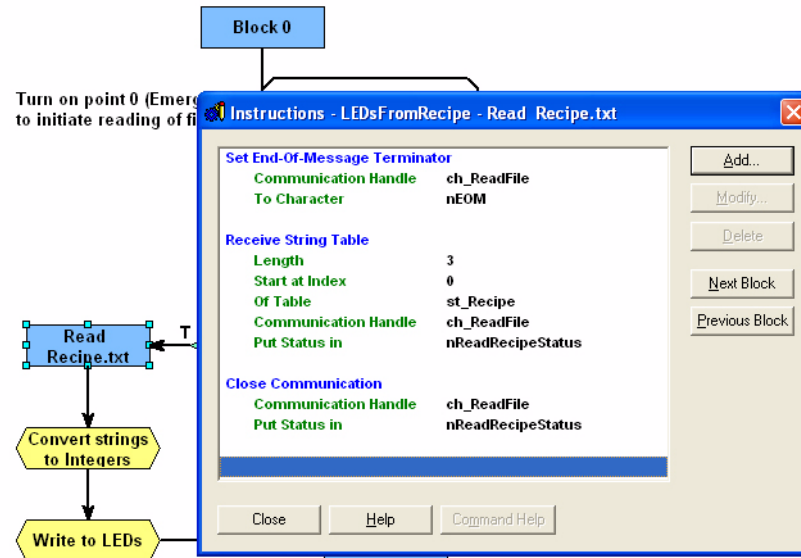
Instruction: Close Communication

Communication Handle: ch_ReadFile

Put Status in: Integer 32 Variable – nReadRecipeStatus

d. Click OK.

Your instruction block should appear as shown here:

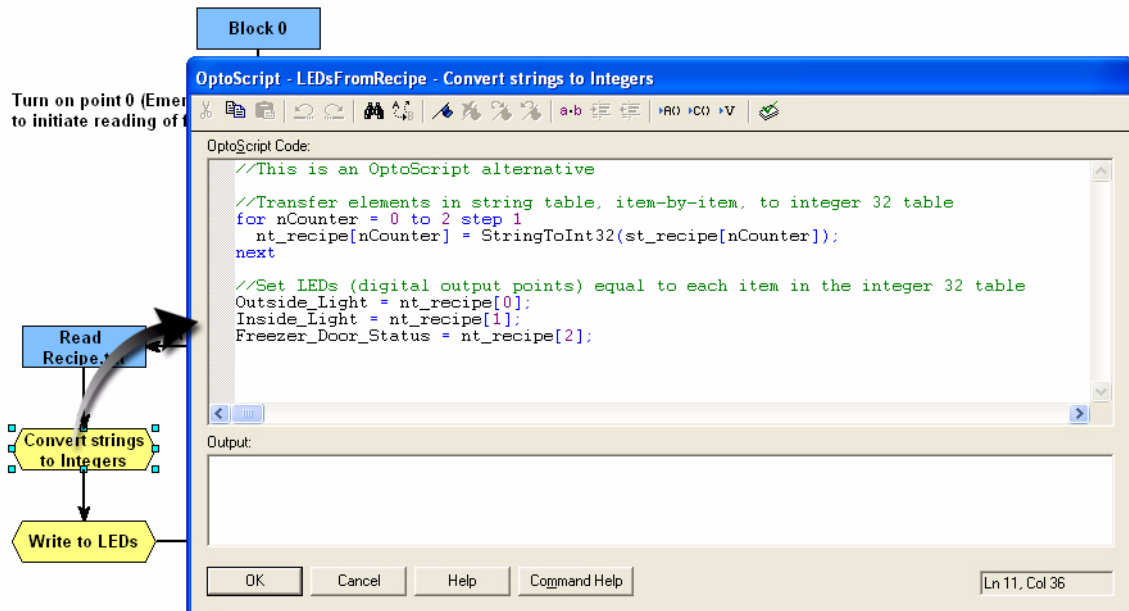


e. Click Close.

9. Convert strings to integers.

a. Double-click the OptoScript block entitled Convert strings to integers

b. Remove the comments from the OptoScript so that your code appears as shown here:

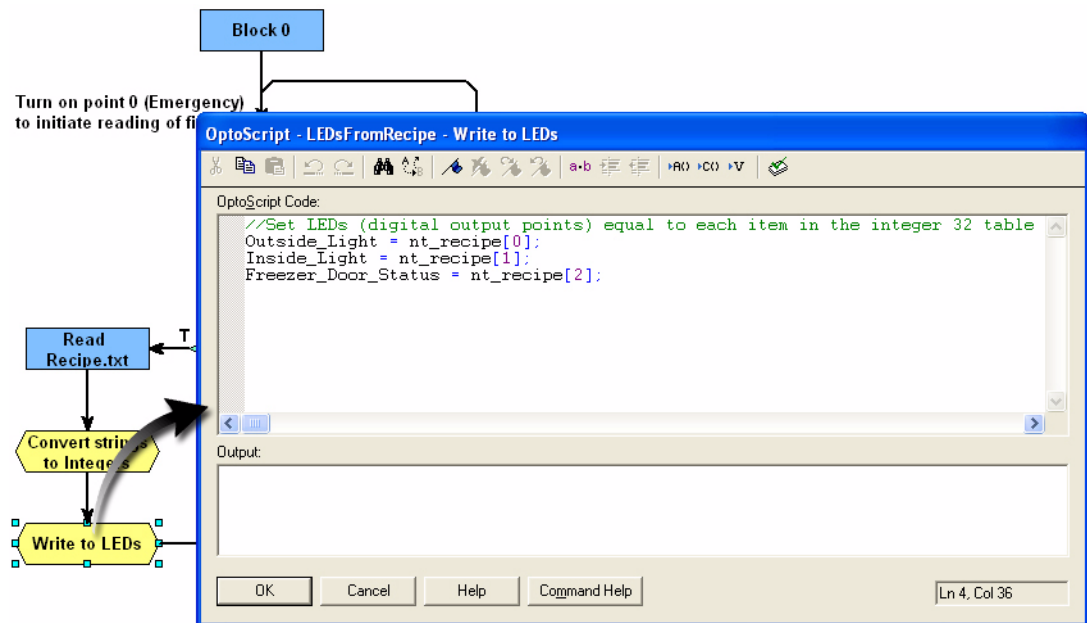


c. Click OK.

10. Write to LEDs

a. Double-click the OptoScript block entitled Write to LEDs

- b. Remove the comments from the OptoScript so that your OptoScript appears as shown below:



- c. Click OK.
11. Compile your strategy.
Choose **Compile** → **Compile All: 1642_FileMgtFTP**
12. Download and run the strategy.
 - a. Choose menu command **Mode** → **Debug**.
 - b. Acknowledge all download messages.
 - c. Choose **Debug** → **Run**.

Activity 2b: Create and Upload a File

1. Create a comma-delimited file.
 - a. Open Notepad.
 - b. Type the following in a new file:
1, 0, 1,

Make sure you include a comma after each number, and do not use spaces between numbers and commas.
 - c. Save your file as `recipe.txt` in the `FtpClient` directory.

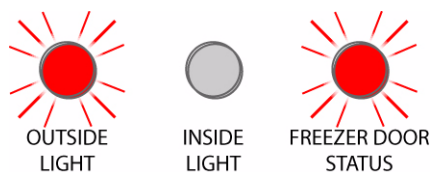
This example shows how to upload a file to the SNAP PAC using an FTP client.

2. Open a DOS command
 - a. From the Start menu, choose Run.
 - b. Type `cmd` and click OK.

3. Navigate to directory to use as a store for the data downloaded.
 - a. Type `cd \` and press Enter to get to the root directory
 - b. Type `cd [directory name]`, e.g.,
`cd FtpClient`
 This will put the cursor in a directory named FtpClient. When you run the FTP client, it default to this directory when uploading a file.
4. Type `ftp` and press Enter to start the FTP client.
5. Connect to the SNAP PAC FTP Server.
 - a. Type `open [IP address of Snap PAC]` for example
`open 10.0.4.10` (and press enter)
 You will be prompted for a User Name
 - b. Type a user name. If you configured a user name in ioManager, then type that User Name. If you didn't configure a user name, you must still provide a user name but it can be any name.
`SnapPac` (and press Enter)
 - c. Type a password. Type either the password you configured in ioManager, or any letter if you did not configure a password. You must type something.
`myPassword` (and press Enter)
 If successful, you'll see a message stating "User Logged in, proceed."
6. Upload the file Recipe.txt.
 Type the following:
`put recipe.txt /uploads/recipe.txt`
 This will create the subdirectory and copy the file.

Test your strategy

1. Hold down the emergency switch for at least a second.



Notice that your LEDs are now on or off depending on the values contained in recipe.txt.

To change the states of the points, edit recipe.txt on your computer, upload it to the controller, and then hold down the emergency switch again.

2. Return to Configure Mode.

Sending Data to an FTP Server

Lesson 3 Overview

The previous lessons showed how to move data to and from the SNAP PAC using the SNAP PAC as an FTP server. This lesson shows a variation in which the log file being written is also sent to an FTP server. In this role, the SNAP PAC is the FTP client.

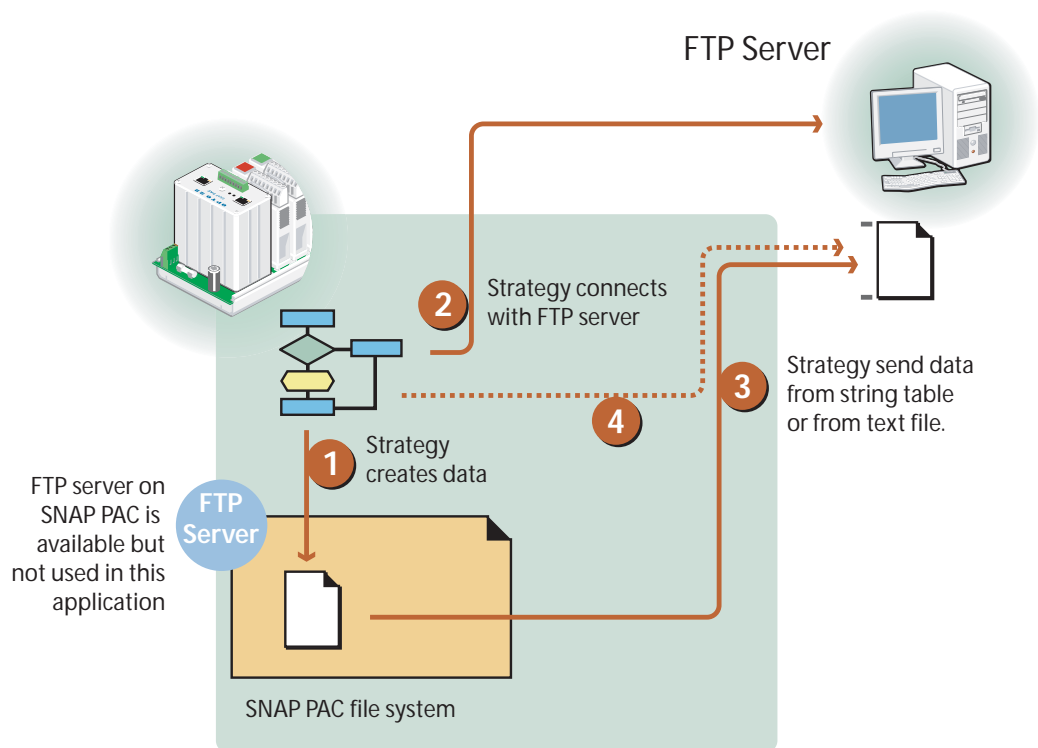


Figure 12: Paths for sending data to an FTP server

1

From Lesson 1, the LogData chart writes data to the string table st_Data and then appends this table to a datalog file.

2	Open communication with the FTP server. This is similar to opening communication through any communication handle, but this one makes a connection to an FTP server using an IP address, port number, user name and password. Once communication is open, you can implement the FTP commands supported by the SNAP PAC.
3	Using a Send command, the file on the SNAP PAC is sent to the FTP server.
4	This shows an alternative to paths 1 and 3. You can send a file, or you can transmit a string, numeric table, or string table without creating a file on the SNAP PAC. You will implement this in Activity 3b.

Using the SNAP PAC as an FTP client has some advantages:

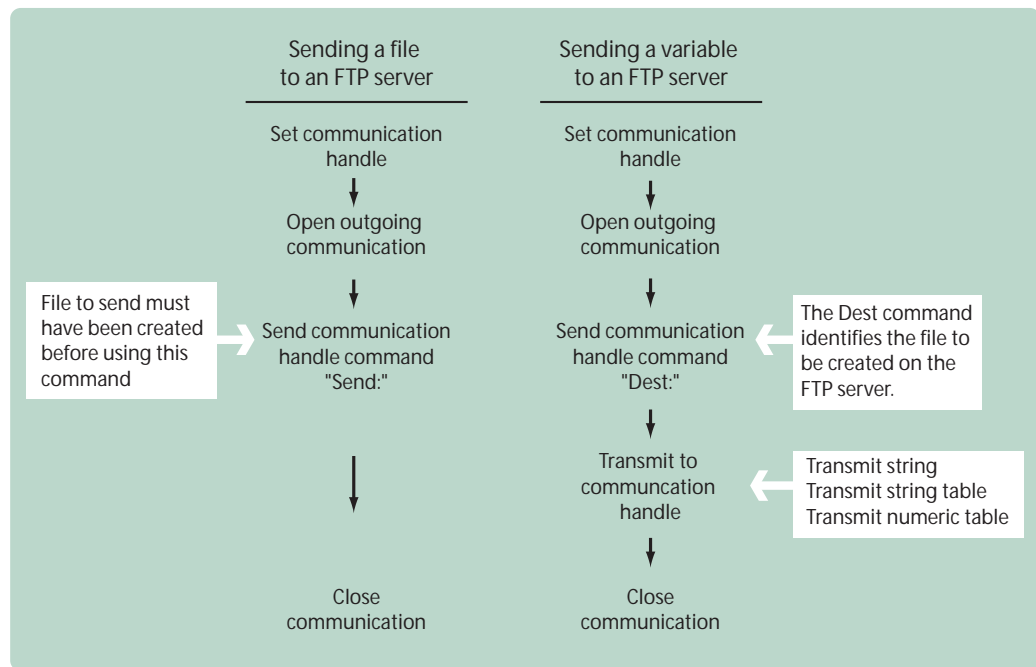
- Greater data storage space: In lesson 1 you learned that there is a 2 MB storage limit for files. Using the SNAP PAC as a client to an FTP server on a PC uses the PC's non-volatile storage space as an extension of the SNAP PAC.
- The SNAP PAC's strategy can know when its data has been collected. The SNAP PAC can send the data and can get an error message if the transmission fails. This can be anticipated with the appropriate warnings built into the strategy. For example, failure to write to the FTP server could trigger an email message.

Sending data to an FTP server

You have two options for how you package data to be sent to an FTP server:

- Write your data to a file on the SNAP PAC and then send the file to the FTP server. (Paths 1–3 in Figure 12)
- Send a variable to an FTP server. This eliminates the step of creating a text file. Path 4 in Figure 12 shows this route using a persistent variable, but this method works with both persistent and global variables.

The communication handle is identical for sending files or transmitting variables, but the commands sent once outgoing communication has been opened are different.



The Send and Get commands have the remote and local file names identified in their parameters. Therefore one command is used. The Transmit string command is not exclusively used with FTP, therefore, identifying the target file is achieved with a preceding **dest** command. In this use, the **dest** command acts as an FTP open command that establishes a blank file on the FTP server.

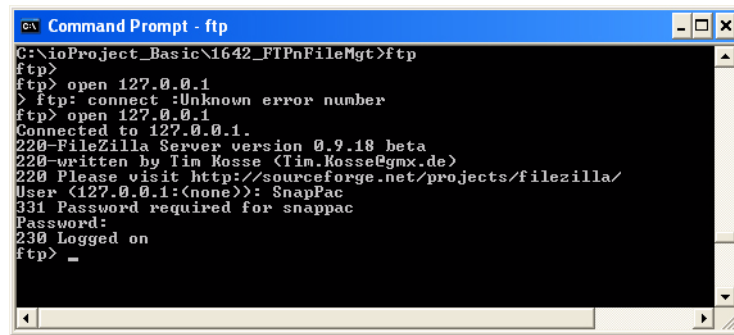
Communication handle for an FTP server

When you log on to an FTP server, you instruct your FTP client to open a server at a specified address and port. (Often this is masked by a domain name and the FTP client.)

The DOS shell client illustrates the steps needed to log on to an FTP server:

- Open the IP address
- Provide a login name requested by the server.

- Provide a password.



```
C:\ioProject_Basic\1642_FTPnFileMgt>ftp
ftp>
ftp> open 127.0.0.1
> ftp: connect :Unknown error number
ftp> open 127.0.0.1
Connected to 127.0.0.1.
220-FileZilla Server version 0.9.18 beta
220-written by Tim Kosse <Tim.Kosse@mx.de>
220 Please visit http://sourceforge.net/projects/filezilla/
User (127.0.0.1:(none)): SnapPac
331 Password required for snappac
Password:
230 Logged on
ftp> _
```

This information must be contained in an ioControl communication handle. For example:

Format:

```
ftp:<IP address>:<port>,<logon name>,<password>
```

Sample:

```
ftp:214.138.218.99:21,opto,mypassword
```

Gateway

This lesson uses an FTP server that can be running on your workstation. However, if you wish to use an existing FTP server that your computer has access to, you should make sure the SNAP PAC also has access to the server. A gateway is the IP address of the network device providing access beyond the network. If the FTP server is on a different network, e.g., on the Internet, then you need to configure the SNAP PAC's IP settings to identify the gateway. Use ioManager, Tools → Change IP Settings.

Activity 3a: Sending Data to an FTP Server

What you need

To create a communication handle to access an FTP server, you will need the following:

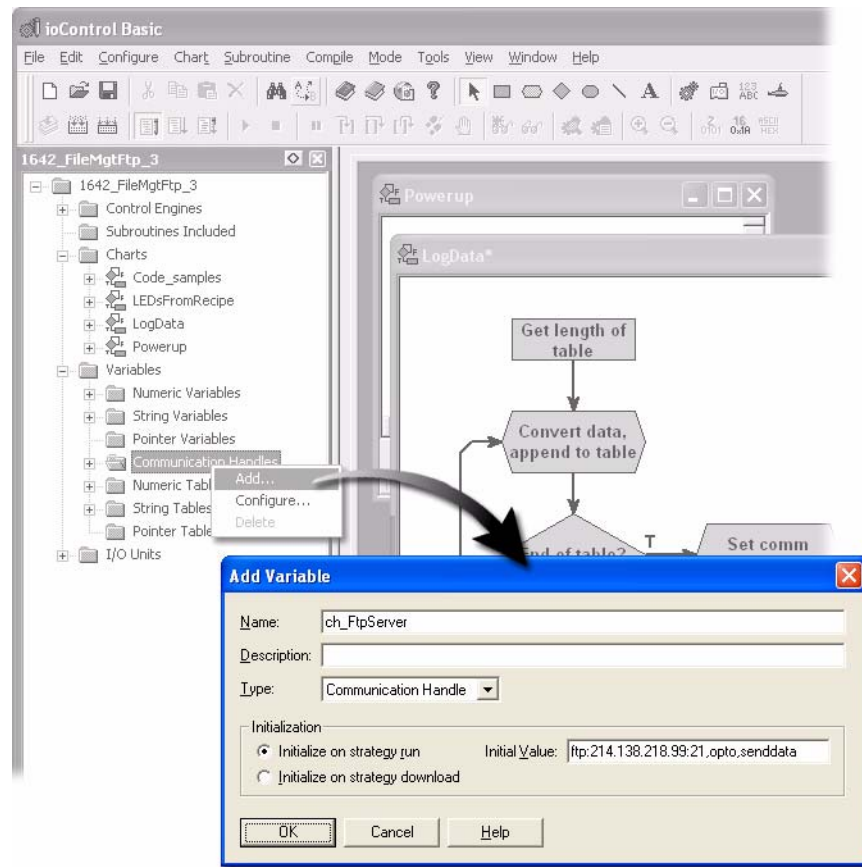
- IP address of FTP server
- If your FTP server and SNAP PAC are separated by a router, you will need to configure the SNAP PAC's IP settings to include the location of the router. How to do this is described in the Appendix (see ["Configuring the Gateway Address" on page 89](#)).
- Logon name and password of the FTP server.

If you have these, you can begin using ioControl as described below. All the examples in this lesson use a FileZilla server.

Create communication handle

NOTE: The instructions here describe the FileZilla FTP server. If you are using a different server, substitute as necessary.

1. Make sure ioControl is in Configure mode.
2. In the Strategy Tree, right-click the Communication Handles folder and click Add.
3. Type `ch_FtpServer` for the name.



4. Type the following information for the Initial Value. Note that this information will need to represent your FTP Server in place of the sample information provided here:

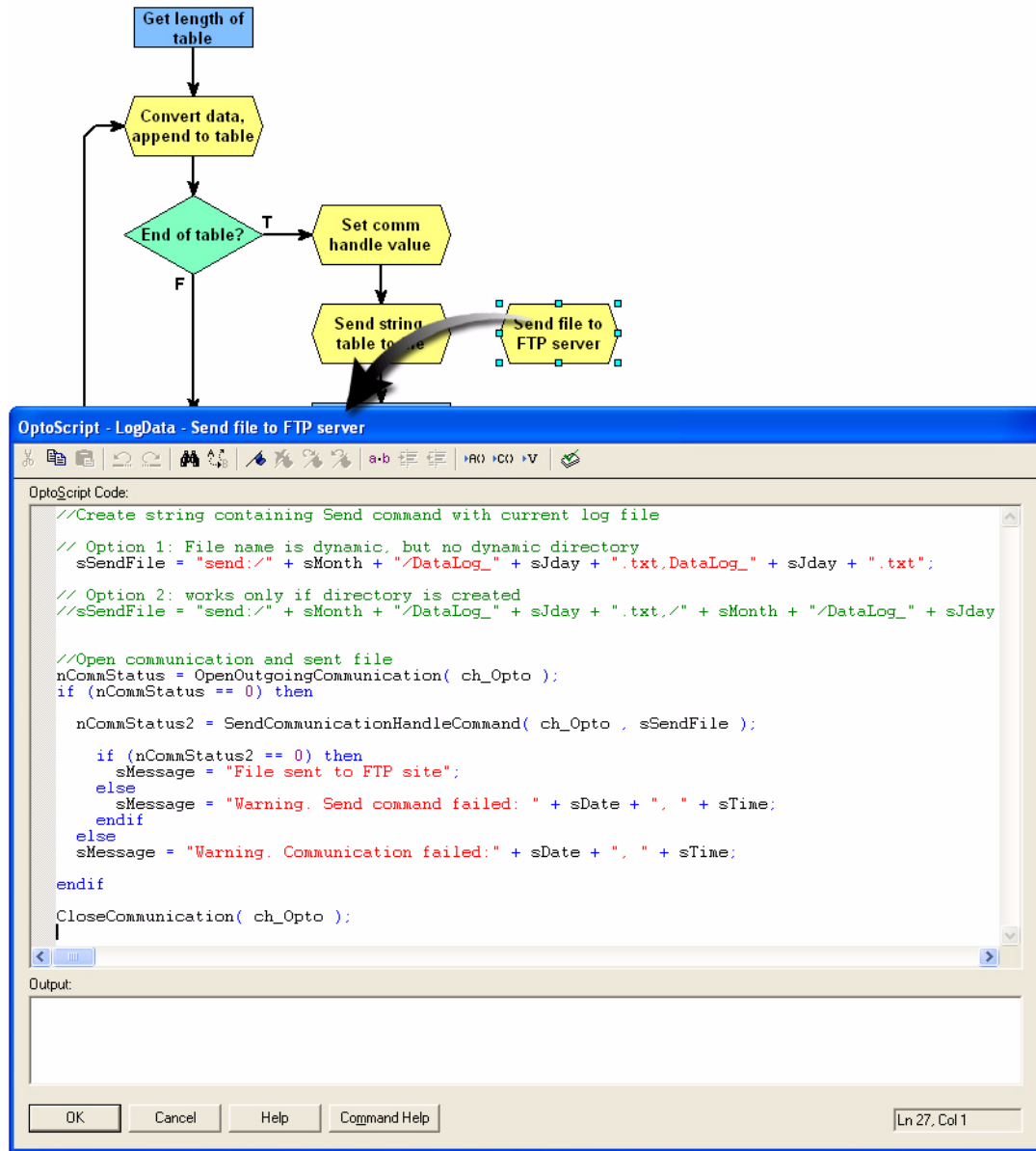
`ftp:214.138.218.99:21, SnapPac , snap`

In this example, substitute "214.138.218.99" with the IP address of your FTP server, "21" with the port (if different), "SnapPac" with your logon name, and "snap" with your password. Include no spaces.

5. Click OK to close the Add Variable dialog box.

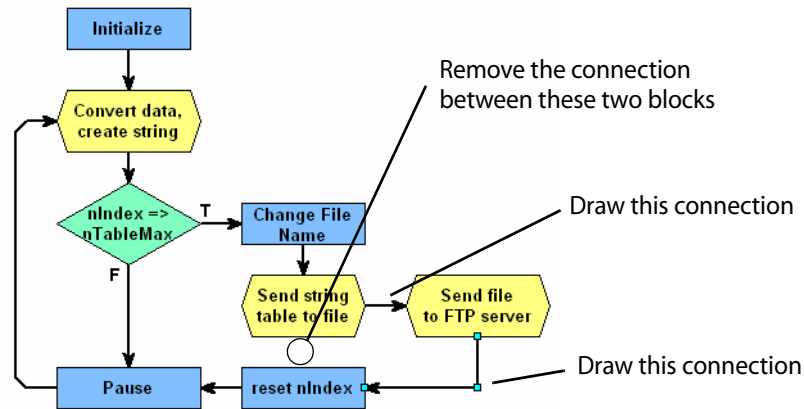
Modify chart to send file to FTP server

1. Add a new OptoScript block in the Log Data chart.
2. Name the OptoScript Send File to FTP Server
3. Open the OptoScript block and type the following code (Note: you can also copy the code from the sample chart "Code Samples"):



There are two alternatives for the destination file shown above. Option 1 puts the data log file in the FTP Server's directory. Option 2 puts the file in a directory corresponding to the directory created on the SNAP PAC. Note that using this second option requires that you create the appropriate directory on the FTP server first. The SNAP PAC doesn't create a directory on the server.

4. Test compile your code.
5. Close the OptoScript editor.
6. Redraw the connection lines as shown below:



7. Test compile you strategy.

Test the strategy

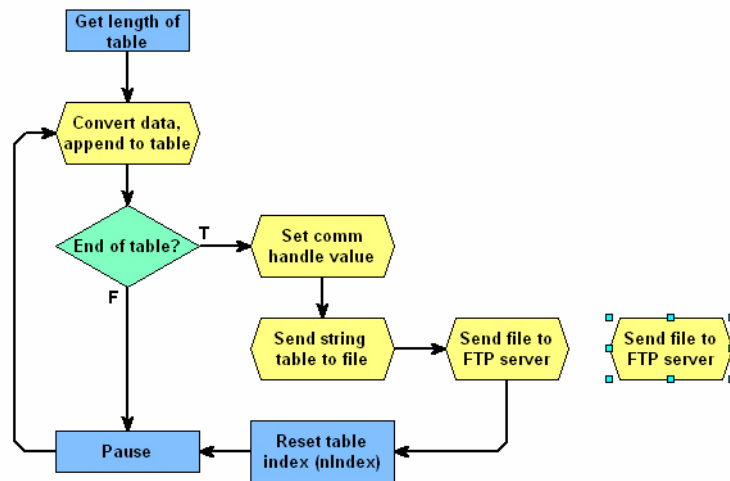
1. Make sure your FTP server is running.
2. Download and run your strategy.
3. Allow adequate time for the strategy to fill the table st_data, at which point the strategy writes to the file and send it to the FTP server.
4. You may wish to activate autostepping to see when the chart tries to send the data.
5. Use an FTP client such as Microsoft Explorer to access your FTP Server.

Activity 3b: Transmit a string table to an FTP server

If you are able to send a file to an FTP server, it is a small change to send the contents of the variable directly, without creating a file.

1. Return to Configure Mode.
2. In the LogData chart, copy the OptoScript block entitled Send file to FTP server.
 - a. Right-click Send file to FTP server.
 - b. Choose copy from the popup menu.
 - c. Right-click in the chart window
 - d. Choose Paste. (You will rename the block.)

- e. Drag the block to a position it as shown below.



- f. Rename your block to Send string table to FTP server

3. Revise the OptoScript
 - a. Open the OptoScript block and revise the code as shown here.

The flowchart illustrates the process of sending data to an FTP server. It starts with a rectangular block 'Get length of table', followed by a hexagonal block 'Convert data, append to table'. A decision diamond 'End of table?' follows. If 'T' (True), it goes to 'Set comm handle value', then 'Send string table to file', then 'Send string table to FTP server', and finally 'Send string table to FTP server'. If 'F' (False), it loops back to 'Convert data, append to table'.

The OptoScript code window shows the following code:

```

OptoScript Code:

//Open communication and sent file
nCommStatus = OpenOutgoingCommunication( ch_FtpServer );
if (nCommStatus == 0) then

  //Option 1: Construct command that opens a new or existing file on the FTP server
  sSendFile = "dest:/DataLog_" + sJday + ".txt";
  nCommStatus2 = SendCommunicationHandleCommand( ch_FtpServer , sSendFile );

  //Option 2: dynamic file name but directory must be created
  //sSendFile = "dest:/" + sMonth + "/DataLog_" + sJday + ".txt";
  //nCommStatus2 = SendCommunicationHandleCommand( ch_FtpServer , sSendFile );

  if (nCommStatus2 == 0) then
    Transmit table
    nCommStatus3 = TransmitStrTable( nTableMax , 0, st_Data, ch_FtpServer );
    if (nCommStatus3 == 0) then
      sMessage = "File sent to FTP site";
    else
      sMessage = "Tranmitting failed";
    endif
  else
    sMessage = "Destination command failed";
  endif
  else
    sMessage = "Opening communication failed";
  endif
  CloseCommunication( ch_FtpServer );

```

Choose one of these two methods.

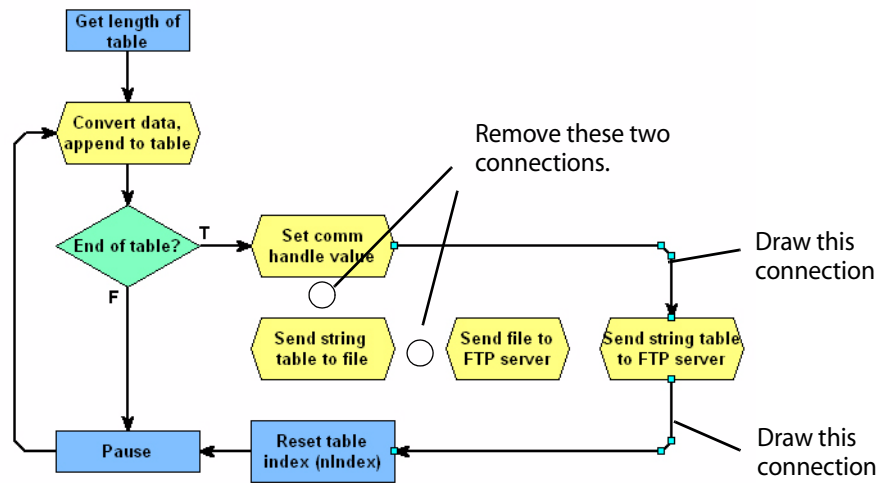
Output:

OK Cancel Help Command Help Ln 3, Col 1

Note: Two options for the file name are shown in the code, that is to put the file in a subdirectory or not. Choose either method.

- b. Test compile your script.
- c. Click OK.

4. Redraw the chart as shown here.



5. Test compile your chart.

Test the strategy

1. Download and run your strategy.
2. Wait for the string table index to reach the maximum according to table length.
3. Check for your file on the FTP server.

Getting Files from an FTP Server

OPTO 22

Overview Lesson 4

Lesson 3 showed how to send data to an FTP server. Using the SNAP PAC as an FTP client you uploaded a file to an FTP server. The same principles also allows the SNAP PAC to get a file from the FTP server. This lesson shows how to use the SNAP PAC as a client to get a file from the FTP server.

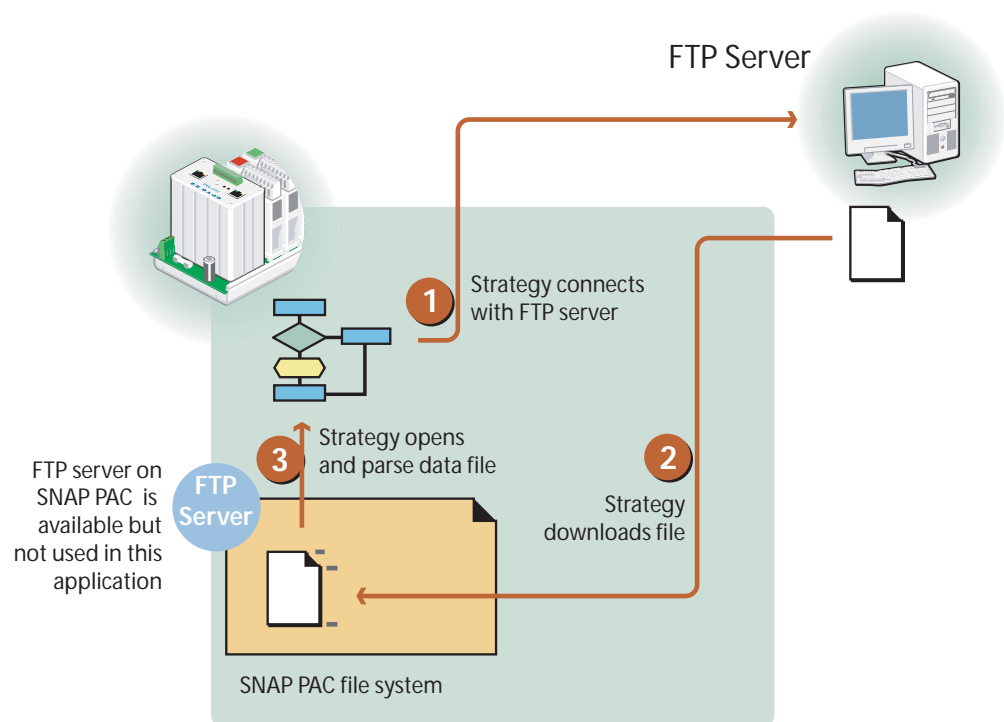


Figure 13: SNAP PAC as an FTP client getting a file from an FTP server

Communication handle for FTP server

The communication handle used to get a file is the same the one created in Lesson 3 used to upload data.

Format:

```
ftp:<IP address>:<port>,<logon name>,<password>
```

Sample:

```
ftp:214.138.218.99:21,SnapPac,myPassword
```

Gateway

A gateway is the IP address of the network device providing access beyond the network. If the FTP server is on a different network, e.g., on the Internet, then you need to configure the SNAP PAC's IP settings to identify the gateway.

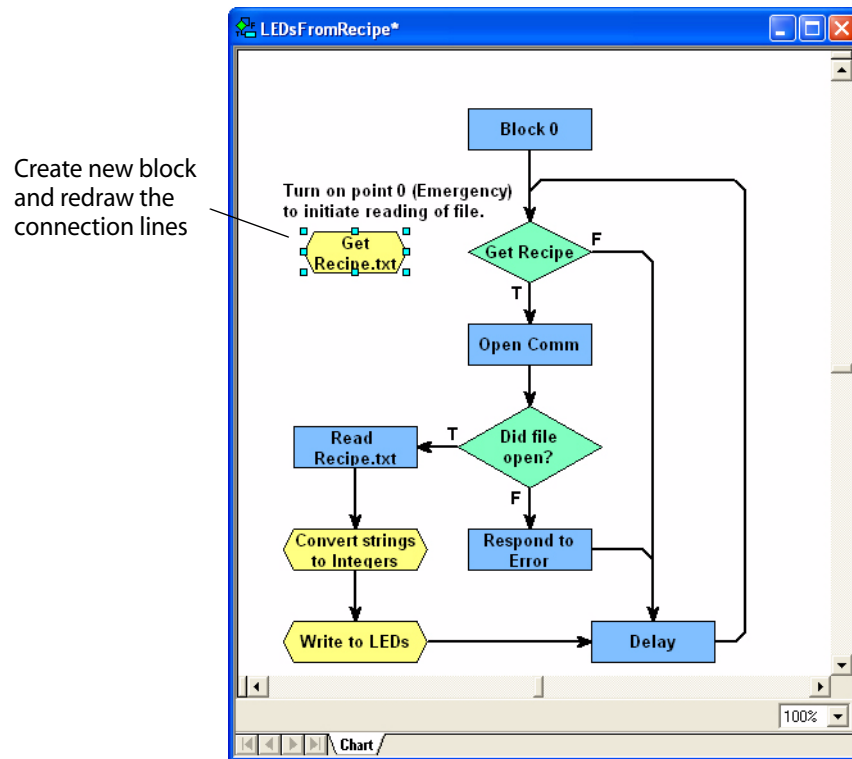
Activity 4: Get a recipe file from an FTP server

Your gateway should be configured as described in lesson 3 (see ["Configuring the Gateway Address" on page 89](#)).

Modify your chart

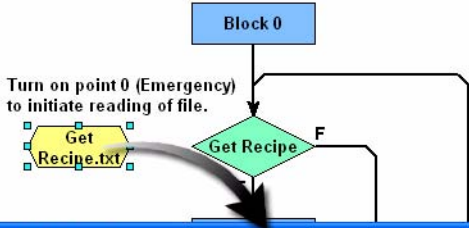
1. Return to Configure mode.
2. Open your chart entitled LEDsfromRecipe.
3. Create a new OptoScript block

4. Name your new block `Get file from FTP server`.



5. Add code to the OptoScript block.
 - a. Double-click the OptoScript block.

- b. Type the following code in the OptoScript editor.



Turn on point 0 (Emergency) to initiate reading of file.

Block 0

Get Recipe.txt

Get Recipe

F

OptoScript - LEDsFromRecipe - Get Recipe.txt

OptoScript Code:

```
//Open communication and sent file
nReadRecipeStatus = OpenOutgoingCommunication( ch_FtpServer );
if (nReadRecipeStatus == 0) then

    nReadRecipeStatus2 = SendCommunicationHandleCommand( ch_FtpServer , "get://recipe.txt./uploads/recipe.txt" );

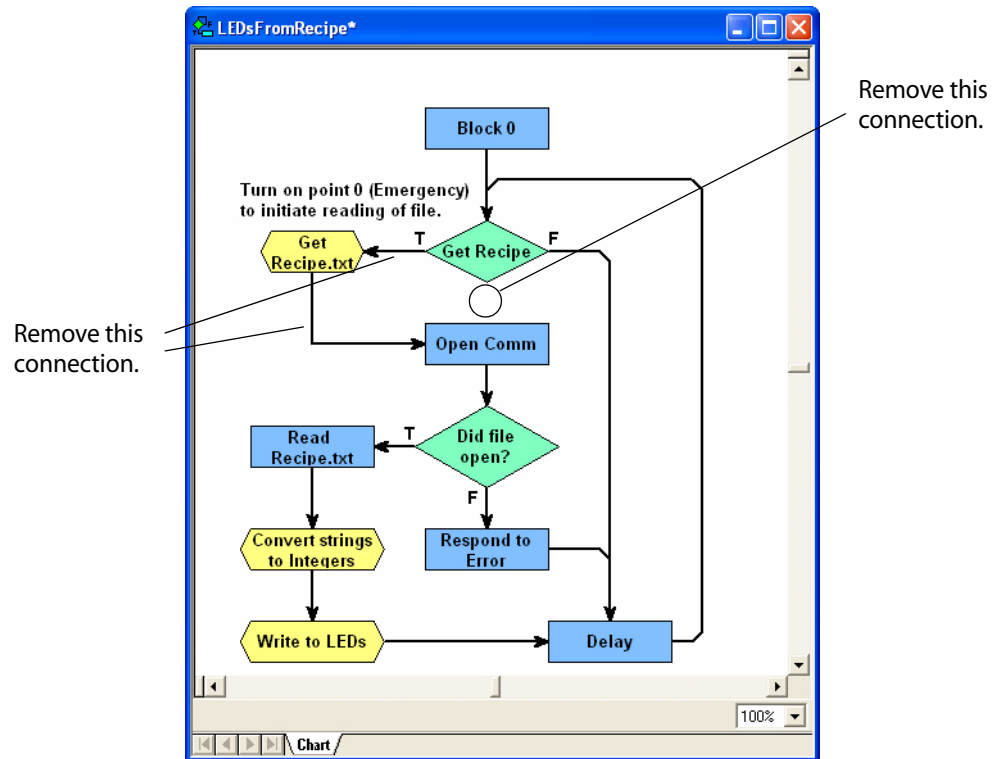
    if (nReadRecipeStatus2 == 0) then
        sRecipeMessage = "File downloaded from FTP site";
    else
        sRecipeMessage = "Warning. Send command failed: " + sDate + ", " + sTime;
    endif
else
    sRecipeMessage = "Warning. Communication failed:" + sDate + ", " + sTime;
endif

CloseCommunication( ch_FtpServer );
```

Close Help Command Help

- c. Test compile your script.
- d. Close the OptoScript editor.

6. Redraw your chart as shown here:



7. Test compile your chart.

Test your changes

- Upload a recipe file to your FTP server in the location specified within the new OptoScript.
Edit your recipe file so that will make a noticeable change to your LEDs (e.g., if your LEDs are On, use values that would turn them off, and vice versa):
To turn off all LEDs, use
0, 0, 0,
Turn on all LEDs, use
1, 1, 1,
- Download and run your strategy.
- Press and hold the Emergency button.

Reference

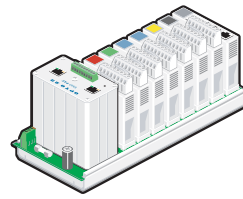
Resources

This appendix contains the following references:

- Memory usage diagram for the SNAP PAC S-series and R-series controllers
- Diagram of the various routes by which ioControl, running on either a SNAP PAC S or SNAP PAC R, can move data among volatile memory, non-volatile memory, FTP servers, and FTP clients
- Defining a Control Engine from within ioControl
- Configuring a gateway address



SNAP-PAC-S1

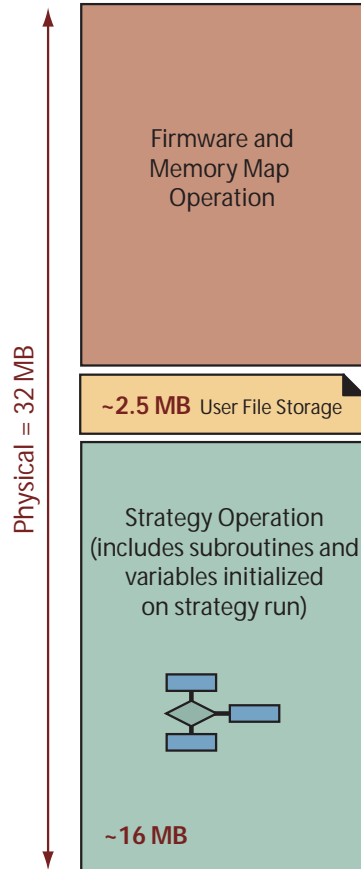


SNAP-PAC-R1
SNAP-PAC-R2

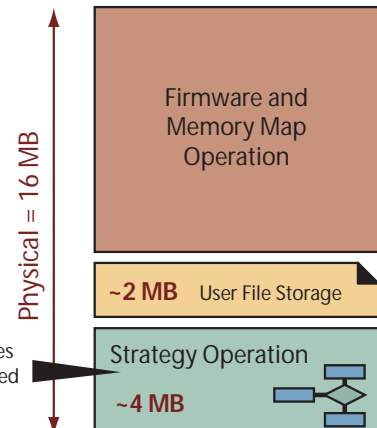
Volatile

All volatile memory is cleared after power loss.

~ **Note:** Numbers are approximations and may change with firmware releases.



Firmware and Memory Map configuration are loaded from Flash when the controller is turned on.

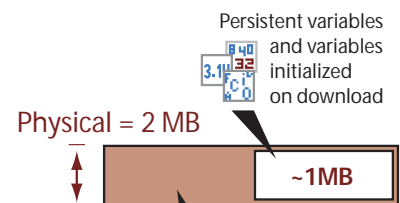
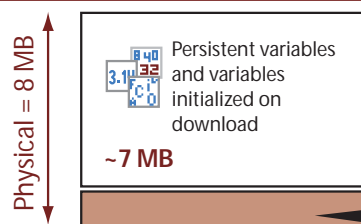


Non-volatile

(Not affected by power loss)

Battery-backed RAM

(Has no write limit)

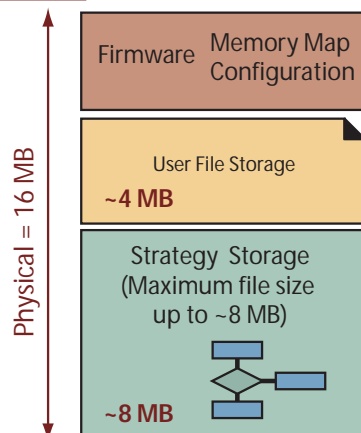


System data and flags such as Strategy Autorun

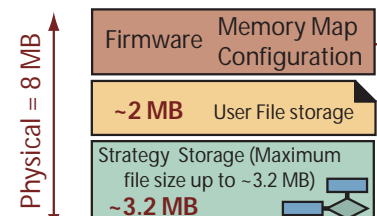
System data and flags such as Strategy Autorun

Flash

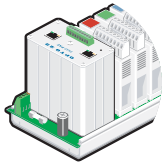
Flash memory has a finite write limit. Caution should be used when writing to Flash: Do not write to Flash in a repeating loop.



Firmware and Memory Map configuration are loaded from Flash when the controller is turned on.

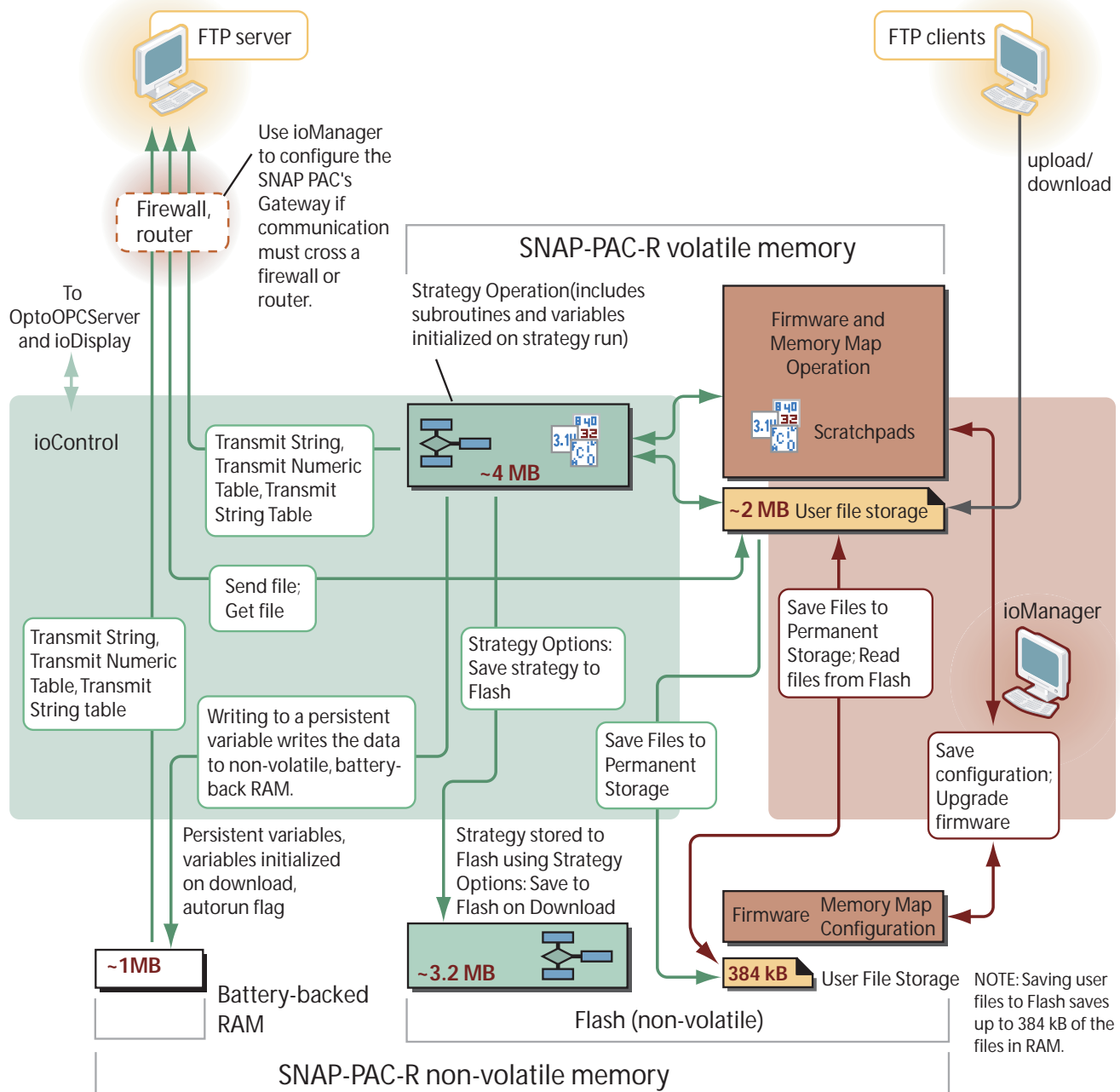


SNAP-PAC-R1 SNAP-PAC-R2



This diagram shows options for moving and storing data on a SNAP PAC R-series controller with an emphasis on file management and FTP. All data is in volatile memory, unless your ioControl strategy moves the data to non-volatile locations, such as Flash and battery-backed RAM.

- **Flash:** Writing to Flash should be done cautiously, as Flash has a finite number of times it can be written to. Therefore, writing to Flash is typically done for infrequent operations such as upgrading firmware, writing configuration settings, downloading new strategies. When data needs to be saved automatically (outside of ioDisplay and OptoOPCServer), it is best to write the data to persistent variables, upload data to an FTP server, or both.
- **Battery-backed RAM:** Variables initialized on download preserve their data until the strategy is loaded. Persistent variables will preserve their data while revised strategies of the same name are loaded and a cleared only when a different strategy is loaded.



SNAP-PAC-S1

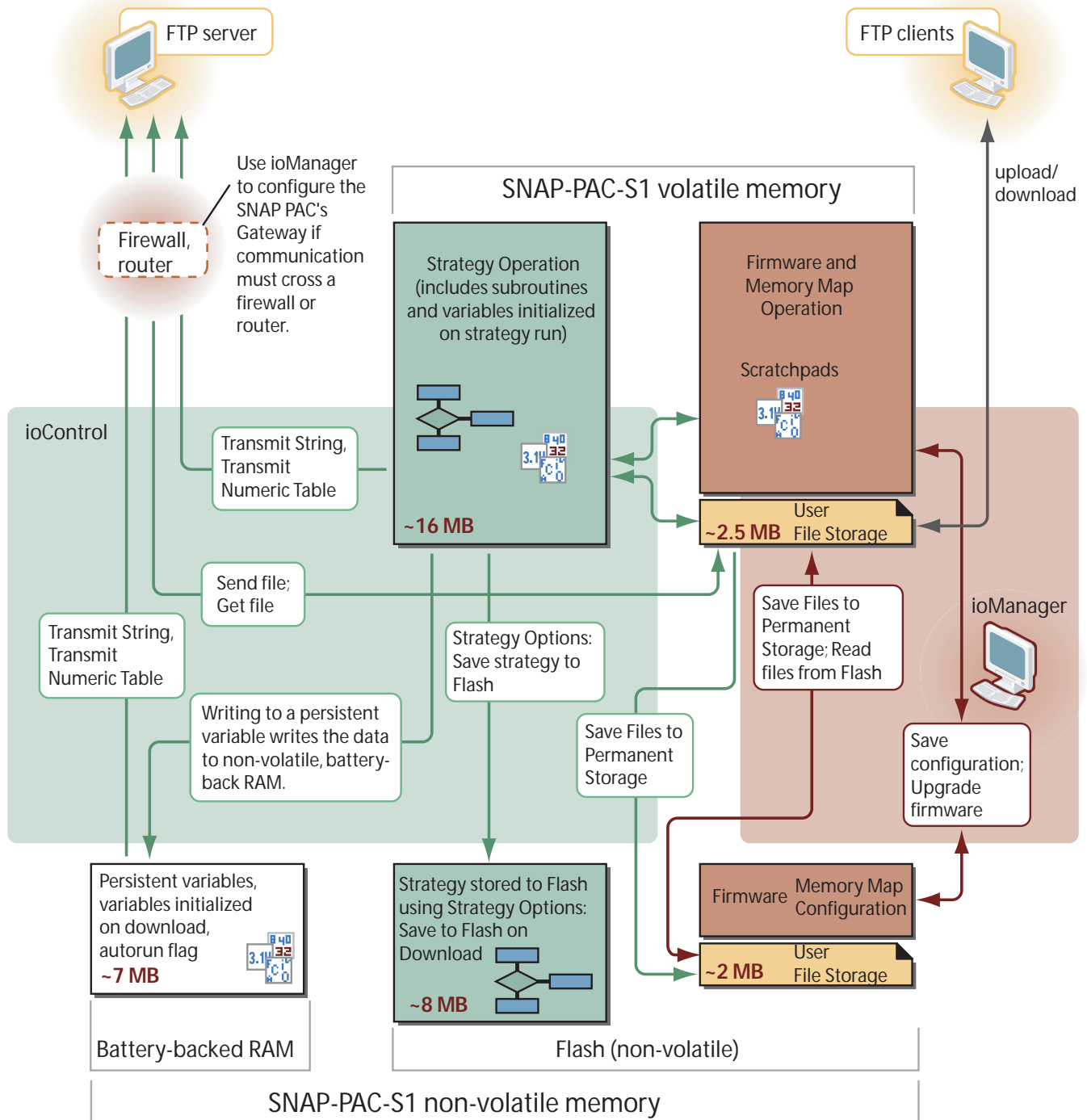


Volatile memory

Non-volatile memory

All data is in volatile memory, unless you create programming that moves the data to non-volatile locations, such as Flash and battery-backed RAM.

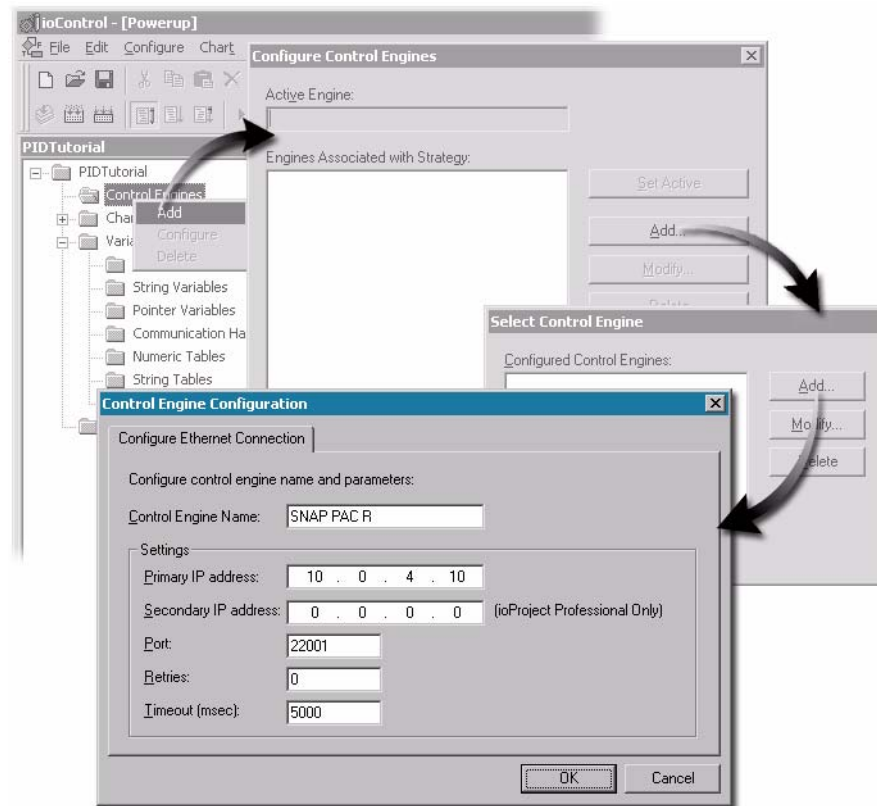
Writing to Flash should be done cautiously, as Flash has a finite number of times it can be written to. Therefore, writing to Flash is typically done for infrequent operations such as upgrading firmware, writing configuration settings, downloading new strategies. When data is needed to be saved automatically, it is best to write the data to persistent variables, upload data to an FTP server, or both



Defining a New Control Engine

This tutorial assumes that the user has completed the SNAP PAC Learning Center lessons and therefore has created a control engine. A control engine resides on the PC workstation and is a reference for ioControl to communicate with the Ultimate I/O brain. If you working on a different computer, you will need to define a new control engine.

1. Open the Control Engine Configuration dialog
 - a. In the strategy tree, right-click Control Engines and click **Add**.
 - b. In the Configure Control Engines dialog box, click **Add**.
 - c. In the Select Control Engine dialog box, click **Add**.



2. Type a Control Engine name and IP address
 - a. Type any name for the control engine, for example **SNAP_PAC_R**.
 - b. Type the IP address of your SNAP PAC, for example **10. 0. 4. 10**.
 - c. Click OK.

You can now associate your Control Engine.

Configuring the Gateway Address

Before using ioManager, make sure that ioControl is in Configure mode, as setting a gateway will interrupt communication between the SNAP PAC and your PC.

1. Start ioManager.
2. Choose **Tools** → **Change IP Settings**.
3. Type the IP address of your SNAP PAC and click Read Current Settings.
4. In the New IP Address field, retype the SNAP PAC's IP address.
5. Type the IP address of the gateway on your network in the Gateway Address field.
6. Check that you have typed the IP addresses of the SNAP PAC and gateway correctly. Note that you are reassigning the same IP address to the SNAP PAC.
7. Click Change IP Settings.

The controller will restart.

