

---

# Modicon Simulator

Version 1.05

By FasTrak SoftWorks, Inc.

Throughout this document, Modicon Simulator will be referred to as the simulator. PLC WorkShop for Modicon – 32 Bit will be referred to as Modicon WorkShop, PLC WorkShop, or WorkShop. Modicon Simulator and PLC WorkShop are trademarks of FasTrak SoftWorks, Inc. Modicon is a trademark of Schneider Electric, Inc. Windows is a registered trademark of Microsoft Corporation. Other product names mentioned herein are used for identification purposes only and may be trademarks of their respective companies.

**Contacting FasTrak SoftWorks, Inc.**

FasTrak SoftWorks, Inc.	262-238-8088
Technical Support	262-238-8088
Sales	262-238-8088
Web	<a href="http://www.fast-soft.com">http://www.fast-soft.com</a>
Email	<a href="mailto:support@fast-soft.com">support@fast-soft.com</a>
Mail	Attention: Technical Service FasTrak SoftWorks, Inc. PO Box 240065 Milwaukee, WI 53224-9003

# LICENSE TERMS AND CONDITIONS

## FasTrak SoftWorks, Inc.

Licensor is the owner of all rights, including the copyright, in and to that certain set of executable computer programs identified in the Registration Form, including design and structure thereof (the "Software"), together with all manuals and other written or printed technical material provided with the Software to explain its operation and to aid in its use (the "Documentation"). Licensee wishes to have the right to use the Software, and Licensor is willing to grant such a right to Licensee on the terms and conditions set forth herein.

1. **GRANT OF LICENSE.** In consideration of Licensee's payment of the license fee referred to below and Licensee's agreement to abide by the terms and conditions stated herein, FasTrak SoftWorks, Inc. (referred to as "Licensor") grants Licensee a nonexclusive right to use and display one (1) copy of the Software with respect to one microcomputer at a time for so long as Licensee complies with the terms hereof. Licensor retains the right to terminate this Agreement and Licensee's rights at any time, by written notice to Licensee, in the event Licensee violates any of the provisions hereof.

Licensor reserves all rights in and to the Software and Documentation not expressly granted to Licensee herein. Licensee agrees to pay Licensor the license fee specified by Licensor as of the date hereof, payable in full upon deliver of a copy of the Software and Documentation to Licensee. Licensee acknowledges that the license fee payable hereunder is consideration solely for the right to use the Software, and payment thereof will not entitle Licensee to support, assistance, training, maintenance or other services, or the enhancements or modifications to the Software which may subsequently be developed by Licensor, except as otherwise expressly provided in this Agreement.

2. **LICENSEE'S AGREEMENTS.** Licensee agrees to comply with the terms and conditions set forth in this Agreement, specifically including but not limited to the following:

- (a) Licensee will take all reasonable steps to protect the Software from theft or use contrary to the terms of this Agreement.
- (b) Licensee agrees to pay Licensor additional license fees as specified by Licensor if and to the extent Licensee intends to use or does use the Software in any way beyond the scope of this Agreement.
- (c) Licensee agrees not to modify the Software and not to disassemble, decompile, or otherwise reverse engineer of the Software.
- (d) Licensee agrees to either destroy or return the original and all existing copies of the Software to Licensor within five (5) days after receiving notice of Licensor's termination of this Agreement.
- (e) Licensee agrees not to disclose the Software or Documentation or any part thereof or any information relating thereto to any other party, it being understood that the same contains and/or represents confidential information which is proprietary to Licensor.

3. **OWNERSHIP OF SOFTWARE.** Licensee shall be deemed to own only the magnetic or other physical media on which the copy of the Software provided to Licensee is originally or subsequently recorded or fixed, as well as any boards, key-locks, or cables provided for use with the Software, but an express condition of this Agreement is that Licensor shall at all times retain ownership of the Software recorded on the original diskette copy and all subsequent copies of the Software, regardless of the form or media in or on which the original or other copies may initially or subsequently exist. This Agreement does not constitute a sale of any copy of the Software to Licensee.

4. **POSSESSION AND COPYING.** Licensee agrees that the Software will only be displayed or read into or used on one (1) computer at a time, at the location designated for notices to Licensee under paragraph 13, below. Licensee may change the computer on which Licensee uses the Software to another computer at such location. Licensee agrees not to make copies of the Software other than for its own use, all of which copies shall be kept in the possession or direct control of Licensee. Licensee agrees to place a label on the outside of all copies showing the program name, version number, if applicable, and Licensor's copyright and trademark notices in the same form as they appear on the original licensed copy.

5. **TRANSFER OR REPRODUCTION.** Licensee is not licensed to copy, rent, lease, transfer, network, reproduce, display or otherwise distribute the Software except as specifically provided in this Agreement. Licensee understands that unauthorized reproduction of copies of the Software and/or unauthorized transfer of any copy of the Software is a violation of law and will subject Licensee to suit for damages, injunctive relief and attorney's fees. Licensee further understands that it is responsible for the acts of its agents and employees. Licensee may not transfer any copy of the Software to another person or entity, on either a permanent or temporary basis, unless Licensee obtains the prior written approval of Licensor which will ordinarily be subject to payment of Licensor's then current license transfer fee. Such approval will not unreasonably be withheld if Licensee advises Licensor in writing of the name and address of the proposed transferee, such transferee is suitable in Licensor's sole judgement, and such transferee agrees in writing to be bound by the terms and conditions of this Agreement. If the transfer is approved, Licensee must deliver all copies of the Software, including the original copy to the transferee.

6. **ENHANCEMENTS AND UPDATES.** Licensor may from time to time release updates of the Software incorporating changes intended to improve the operation and/or reliability of the Software. Such updates will be provided to Licensee at no charge (except shipping charges and media costs) for a period of twelve (12) months from the date hereof, and Licensee agrees to install all updates designated by Licensor as mandatory. Licensor may also offer enhanced versions of the Software from time to time incorporating changes intended to provide new or enhanced features and/or capabilities, at such license fees as Licensor may from time to time establish.

7. **LIMITED WARRANTY AND DISCLAIMER OF LIABILITY.** LICENSOR HAS NO CONTROL OVER THE CONDITIONS UNDER WHICH LICENSEE USES THE SOFTWARE. THEREFORE, LICENSOR CANNOT AND DOES NOT WARRANT THE PERFORMANCE OR RESULTS THAT MAY BE OBTAINED BY ITS USE. HOWEVER, LICENSOR PROVIDES THE FOLLOWING LIMITED WARRANTY:

Licensor warrants, for a period of twelve (12) months only, that the Software shall be free from significant programming errors. Licensor further warrants that it has full power and authority to grant the rights granted by this Agreement with respect to the Software and that the use by Licensee of the Software and Documentation will not infringe the rights of others. In the event Licensee believes that it has discovered one or more significant programming errors, Licensee shall immediately notify Licensor of such fact in writing. If such notice is received by Licensor within twelve (12) months from the date hereof, Licensor shall, within a reasonable time, subject to the demands of Licensor's other customers and subject to delays beyond Licensor's control (including but not limited to labor trouble, illness, delays in shipment of materials, and bad weather), at Licensor's expense, correct the programming errors. In the event Licensor is unable to correct the programming error within a reasonable time, Licensee may elect to terminate this Agreement and receive a refund of the licensee fee paid hereunder. For purposes hereof, a programming error is "significant" only if, as a result thereof, the software does not substantially perform the functions described in the Documentation. Licensor does not warrant that the operation of the Software will be uninterrupted or error free. EXCEPT FOR THE ABOVE EXPRESS WARRANTY, LICENSOR MAKES AND LICENSEE RECEIVES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, AND LICENSOR SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

THE REMEDY PROVIDED HEREIN IS EXCLUSIVE. UNDER NO CIRCUMSTANCES WILL LICENSOR BE RESPONSIBLE FOR DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES OR LOST PROFITS. LICENSEE ACKNOWLEDGES THAT THE LICENSE FEE HEREUNDER IS NOT ADEQUATE FOR LICENSOR TO ASSUME OBLIGATIONS TO LICENSEE GREATER THAN THE EXPRESS REMEDY PROVIDED ABOVE.

8. **GOVERNING LAW.** The validity and performance of this Agreement shall be governed by Wisconsin law, except as to copyright and trademark matters which are governed by United States laws and international treaties. This Agreement is deemed entered into in Wisconsin. All lawsuits arising out of this Agreement shall be brought in a court of general jurisdiction in Milwaukee, Wisconsin. Licensor shall be entitled to recover its costs and expenses (including attorney's fees) incurred in enforcing its rights under this Agreement.

9. **WAIVER.** The failure of Licensor to enforce any of the provisions hereof shall not be construed to be a waiver of the right to enforce such provisions at a later time or to enforce any of the other provisions hereof.

10. **EFFECT OF TERMINATION.** The expiration or termination of this Agreement shall not affect the obligations of Licensee which by their character are of continuing nature.

11. **INTEGRATION.** This Agreement sets forth the entire understanding and agreement of the parties shall be bound by any conditions, definitions, warranties or representations with respect to any of the terms or conditions hereof other than as expressly provided in this Agreement. This Agreement may only be modified by a writing signed by the party to be charged.

12. **BINDING EFFECT.** This Agreement shall be binding upon and shall inure to the benefit of the parties hereto and their respective successors and assigns, subject to the limitations on the transfer of Licensee's rights to the Software provided in paragraph 5, above.

13. **NOTICES.** All notices shall be in writing and shall be hand delivered or sent by U.S. mail, first class, postage prepaid, if to Licensor at its address first above written, and to Licensee at the address indicated in the Registration Form. A party may change its address for notices at any time by notice to the other party in the manner provided herein, but each party may have only one address for notices at a time.

14. **REGISTRATION FORM.** The Registration Form is a part of this Agreement and is incorporated herein by reference. This Agreement will not take effect, and Licensee will have no rights whatsoever with respect of the Software, unless and until the Registration Form is duly executed and returned to the Licensor and is accepted by Licensor.

# Table Of Contents

<b>Overview</b> .....	1
Setting Screen Resolution .....	1
Demo Mode Restrictions.....	2
Starting Modicon Simulator .....	3
<b>Setup</b> .....	<b>5</b>
Selecting PLC Type .....	5
Changing PLC Type.....	6
Setting Up Communications .....	7
Connecting to Modicon Simulator.....	10
<b>Programming Features</b> .....	<b>11</b>
Overview of Simulated Scan.....	11
Unsupported Features .....	11
<b>Using the Simulator PLC Status Window</b> .....	<b>13</b>
Current PLC Status .....	13
Connection Status.....	13
PLC Scan Status .....	14
Timing Mode .....	14
Real Time Mode .....	14
Sweep Mode .....	14
Timing Limitations.....	15
Saving PLC State.....	16
Startup Restore .....	16
<b>Using the Simulator Debugger Toolbar</b> .....	<b>17</b>
Step Mode .....	17
Current Execution Location.....	18
Step to Next Location.....	18
Single Scan Step Mode .....	19
Single Network / Single Instruction Step Mode .....	19
Breakpoint Editor .....	21
Viewing Breakpoints .....	21
Adding a Breakpoint.....	22
Disabling an Individual Breakpoint.....	23

Re-Enabling an Individual Breakpoint .....	24
Removing a Breakpoint .....	25
Editing a Previously Entered Breakpoint .....	25
Editing Ladder with Valid Breakpoints.....	26
Breakpoint Color / Icon Scheme .....	26
I/O Simulator Control .....	27
<b>Using the I/O Simulator .....</b>	<b>29</b>
Creating an I/O Batch .....	31
Managing Recipe Properties .....	33
Creating Cyclic Recipes.....	34
Modifying Batch Properties.....	36
Reordering a Recipe's Batches .....	37
Saving Recipe Files.....	38
Opening Recipe Files .....	38
Downloading Recipes .....	38
Executing Recipes .....	39
Time-Based Batch Execution .....	39
Zero Trigger Batches .....	39
Viewing the Error Log.....	39
Data Window Fill .....	40
I/O Simulator Window Fill .....	41
<b>Index .....</b>	<b>45</b>

# OVERVIEW

Use Modicon Simulator to program, run, and test Modicon ladder logic and simulated I/O without using a PLC.

Used in conjunction with Modicon WorkShop and data acquisition software such as ControlShop, Modicon Simulator has been designed to assist programmers in testing and debugging Modicon PLC programs before they go live, eliminating the need for expensive hardware. The simulator may also be used to test and debug operator interfaces.

Designed to simulate logic from any Modicon PLC, Modicon Simulator offers preset configuration for all Modicon family PLCs.

Modicon Simulator responds in a similar manner to the PLC type selected. Programs may be loaded, saved, edited, and executed while online with the simulator. In addition, Modicon Simulator allows specific simulator settings that have been created to be saved.

Powerful testing and debugging features include:

- Single Scan Step Mode
- Single Network Step Mode
- Single Instruction Step Mode
- Use of Conditional and Unconditional Breakpoints
- I/O Simulation

## Setting Screen Resolution

To achieve optimum usability of the Modicon Simulator, screen resolution must be set to at least 800 x 600 pixels. For best results, a resolution of at least 1024 x 768 is recommended.

## Demo Mode Restrictions

If using this software without a license, Modicon Simulator will start up in **Restricted Demo Mode**, which allows a 30-minute temporary license and restricted memory size, and will attempt to load the *DemoFile.FTS* file. If it is unable to load this file, it will load the default maximum memory configuration, leaving all internal memory blank. If memory configuration is changed, Modicon Simulator will automatically switch to **Full Demo Mode**, which allows a 5-minute temporary license upon entering Run Mode and full control of memory configuration.

Upon expiration of the temporary restricted license, you will be prompted with an expiration notice, Modicon Simulator will be forced to complete the current scan, and the program execution will halt.

When using this software in demo mode, certain restrictions apply:


You will be unable to:

- Load \*.FTS (simulator state) files
- Save \*.FTS (simulator state) files
- Use Startup Restore



## Starting Modicon Simulator

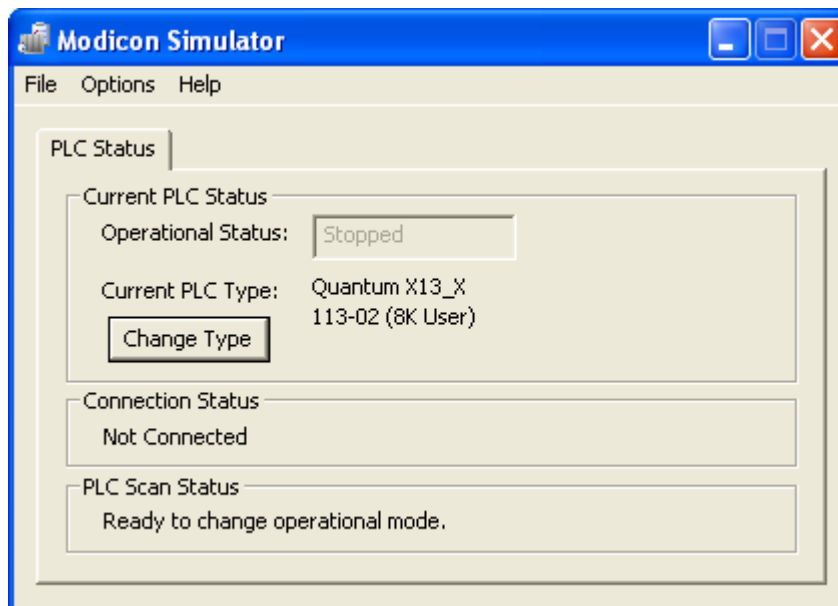
To run the Modicon Simulator:

Within Modicon WorkShop, select the **View \ Modicon Simulator** menu item or click the  toolbar icon.

OR

Select **Start \ Programs \ FasTrak SoftWorks \ Modicon Simulator**.

In either case, the **Modicon Simulator Window** appears.



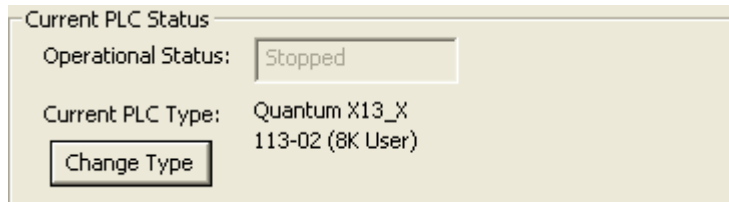


# SETUP

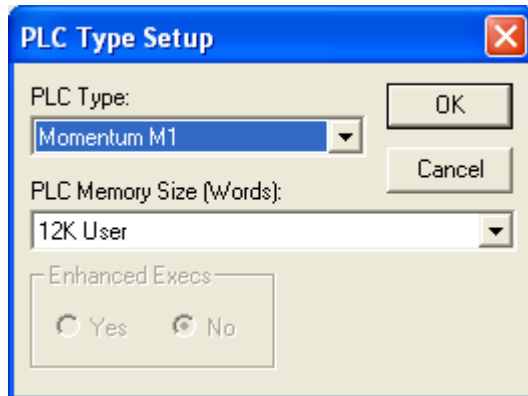
## Selecting PLC Type

The simulator will start up with the same default memory configuration as a new offline Momentum M1 program in Modicon WorkShop. To select another PLC type:

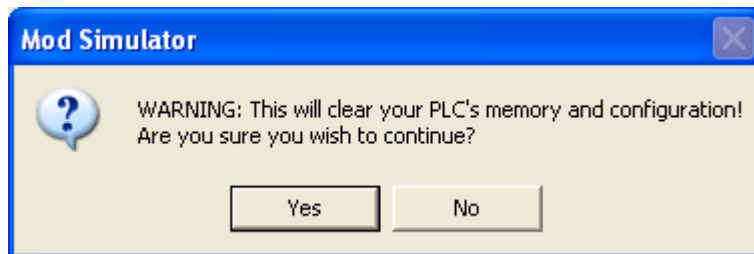
1. Click the **Change Type** button within the **Current PLC Status** group box.



2. The **PLC Type Setup** dialog appears. Select the desired PLC Type and memory configuration.



3. Click **OK**. The following warning will appear:



4. Click **Yes**. The **Current PLC Type** field will update to the new PLC type.

## Changing PLC Type

Attempting to change the PLC type by clicking the **Change Type** button while a client (i.e., Modicon WorkShop) is connected to the simulator, will result in the following warning message to prevent errors in the connected client:

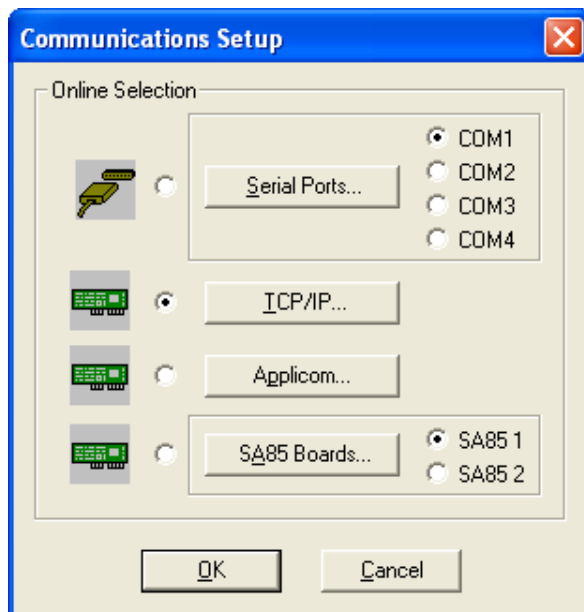


To change the PLC type, any attached clients must first be disconnected from the simulator.

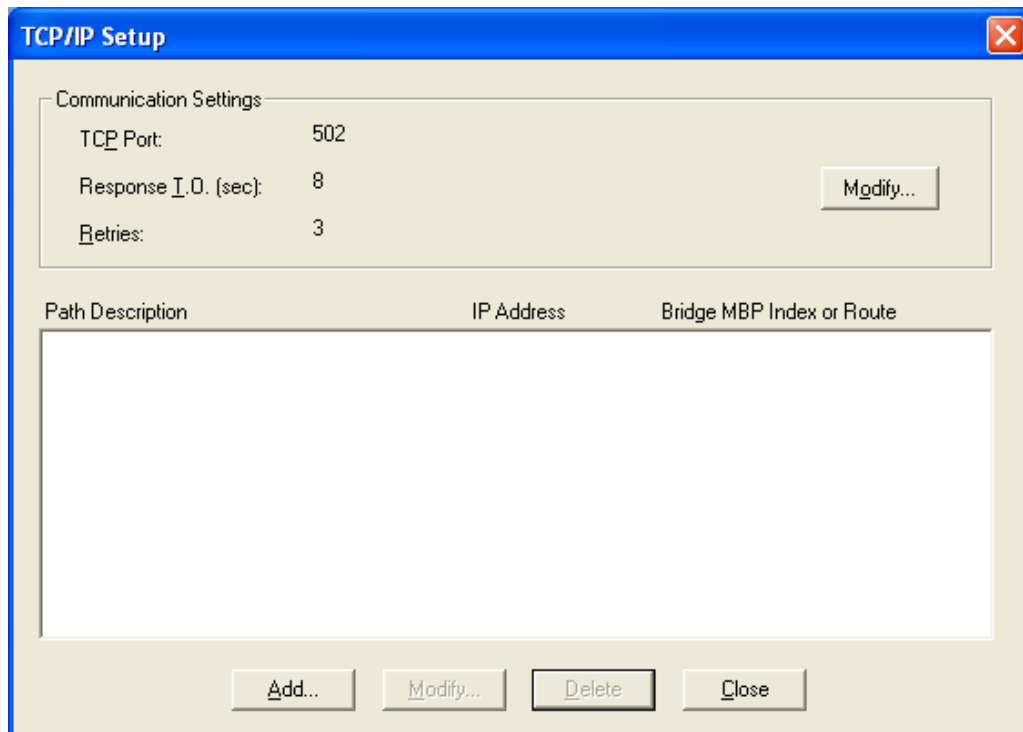
## Setting Up Communications

Modicon Simulator uses the TCP/IP communication method for connection with Modicon WorkShop. To create a new TCP/IP connection in WorkShop to connect to the simulator:

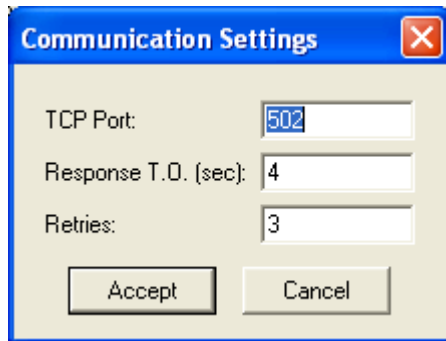
1. Select the **File \ Communications Setup** menu option. The **Communications Setup** dialog appears.



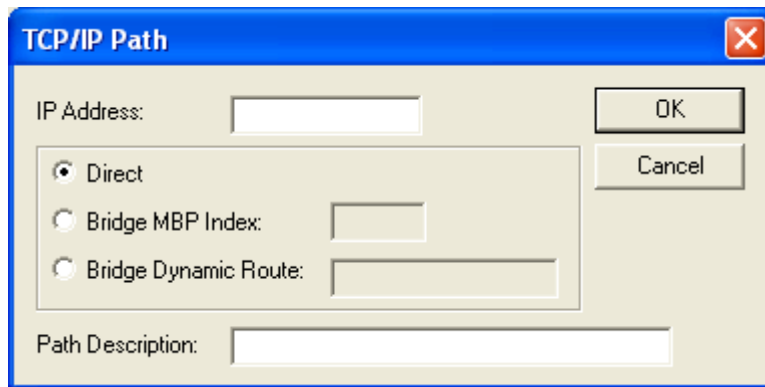
2. Select the **TCP/IP** button. The **TCP/IP Setup** dialog appears.



3. Click the **Modify** button. The **Communication Settings** dialog appears.

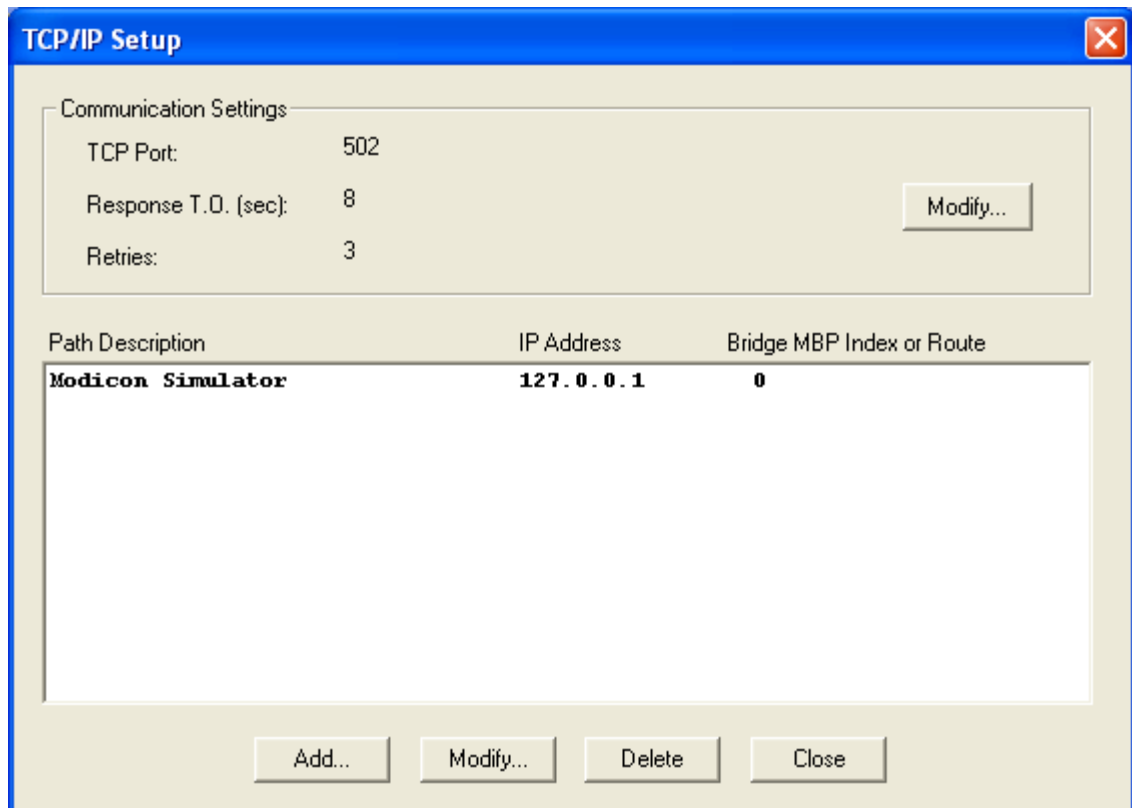


4. In the **TCP Port** text box, enter **502**. Click **Accept** to return to the **TCP/IP Setup** dialog.
5. Click the **Add** button at the bottom of the **TCP/IP Setup** dialog. The **TCP/IP Path** dialog appears.



5. Enter an IP address of **127.0.0.1**, which is the localhost (loopback address).


6. Enter a name for this connection (such as **Modicon Simulator**) in the **Path Description** text box. Click **OK** when finished to return to the **TCP/IP Setup** dialog. The new path appears in the path list.

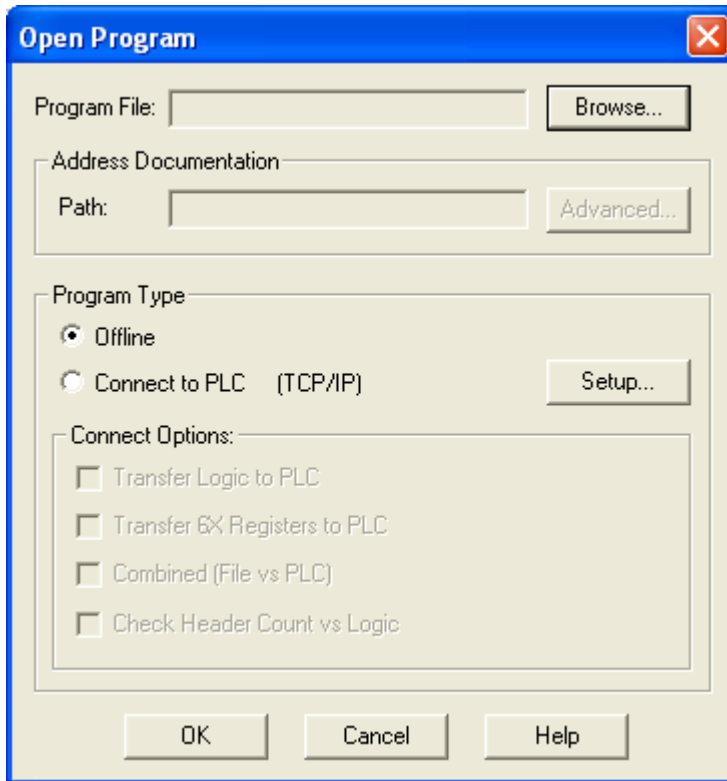


5. Click **Close** in the **TCP/IP Setup** dialog to return to the **Communication Setup** dialog. Click **OK** to exit and return to the **PLC WorkShop** main window.

## Connecting to Modicon Simulator

After Modicon WorkShop and Modicon Simulator are both running, and TCP/IP communications have been set up using a localhost IP Address of 127.0.0.1, connecting to the simulator is identical to going online to a PLC.

1. Within Modicon WorkShop, select the **File \ Open** menu item or click the  toolbar icon. The **Open Program** dialog appears.



---

**NOTE:** Modicon Simulator must be running before attempting to connect to it.

---

2. If desired, select a program to open online by clicking the **Browse** button and browsing for the desired \*.FMD file.
3. Select the **Connect to PLC** radio button. Verify that **(TCP/IP)** appears to the right of **Connect to PLC**. If it does not, click the **Setup** button and select the **TCP/IP** radio button.
4. If you are opening a program, verify that the **Transfer Logic to PLC** check box has been selected and click **OK**. PLC WorkShop is now connected to Modicon Simulator.



# PROGRAMMING FEATURES

In order to get the most benefit from Modicon Simulator, programmers must be aware of how the simulator will compile and scan programs compared to an actual PLC and which programming features are not supported.

## Overview of Simulated Scan

Because Modicon Simulator runs on a PC in a Windows environment, it will execute its scan somewhat different than an actual Modicon PLC, to allow some of the unique features of the simulator to operate correctly. Communication is no longer restricted from executing simultaneously with logic execution, meaning that it is always accessible, even when the scan would not normally be processing communications.

Unless the PLC is in **sweep mode**, all timing in the Modicon Simulator is governed by the PC's system timer, via the Windows API (**real time**). This timer will be updated at the beginning of every scan to reflect the amount of time that has progressed since it last reached the top of the scan. If a break is encountered during the scan, the simulator will record the current timestamp and discontinue advancing the time progressed until the execution is released from its breakpoint. With this timing mechanism, users should be aware that the speed of their PC will affect the amount of scans that will be completed within a certain unit of time.

---

**NOTE:** If program timing is critical to a process, it is recommended that it be validated on a Modicon PLC before going live.

---

## Unsupported Features

Although the following features are not supported, PLC programs containing them can be simulated. Modicon Simulator will store any applicable edits that are made in programs that are connected to it.

- Profibus I/O is not supported, but can be simulated using the I/O Simulator. Profibus I/O can also be configured and will be stored by Modicon Simulator for file saving purposes.
- Configuration Extensions and Loadables are not supported, but can be configured and will be stored by Modicon Simulator for file saving purposes.
- 6x registers are not supported.



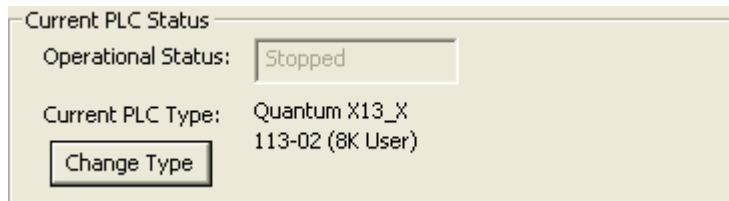
# USING THE SIMULATOR PLC STATUS WINDOW

The PLC Status Window provides information about the operational state of the simulator, connection status, PLC scan status, and timing mode.

## Current PLC Status

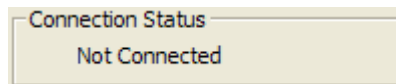
The Current PLC Status section of the PLC Status Window has the following functions:

- Indicates status of **Stopped** and **Running**.
- Allows PLC type to be selected. See *Selecting PLC Type*, pg. 5, for information on selecting or changing PLC type.



## Connection Status

The Connection Status field will display *Client Connected* when a connection has been established or *Not Connected* when there are no clients communicating to Modicon Simulator.



---

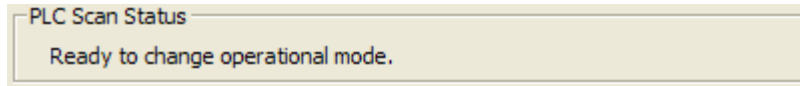
**NOTE:** Connection to more than 10 clients at a time is not recommended.

---

## PLC Scan Status

The PLC Scan Status field indicates when the following operations, which change the state of the simulator, may be performed.

- Changing state from Run to Program



---

**NOTE:** If an attempt is made to perform one of these operations while the simulator is running it will force the scan to finish in order to perform the operation.

---

## Timing Mode

### Real Time Mode

Modicon Simulator will use the PC's system timer via use of the Windows API. This timer will be updated at the beginning of every scan to reflect the amount of time that has progressed since it last reached the top of the scan. If a break is encountered during the scan, the simulator will record the current timestamp and discontinue advancing the time progressed until the execution is released from its breakpoint. Real time mode is recommended for users who wish to view their process timers progressing with respect to real time. With this timing setting, users should be aware that the speed of the PC will affect the amount of scans that will be completed within a certain amount of time.

### Sweep Mode

Sweep mode is a debugging feature included in PLC WorkShop for controlling monitoring scan time and number of scans. See the PLC WorkShop documentation for more information about sweep mode.

## Timing Limitations

- Since debugging allows the scan to be halted at any time, communications with the simulator are always active instead of being restricted to a time slice.
- Because the actual execution time on a Modicon PLC cannot be measured, Modicon Simulator will not update or set the scan-overflow indicator or scan watchdog timer.

## Saving PLC State

Simulator state may be saved. The file containing the simulator state will be saved with an \*.FTS file type.

---

**NOTE:** If the simulator is currently in the middle of a scan, it will be unable to save until the scan finished. A warning that continuing the save operation will force the current scan cycle to finish will display and options to continue or cancel the save will be provided.

---

To save Modicon Simulator state:

1. Select the **File \ Save** menu item within the **Modicon Simulator** Window.
2. If the state is being saved for the first time, a **Save As** dialog will appear and prompt for a file name.

## Startup Restore

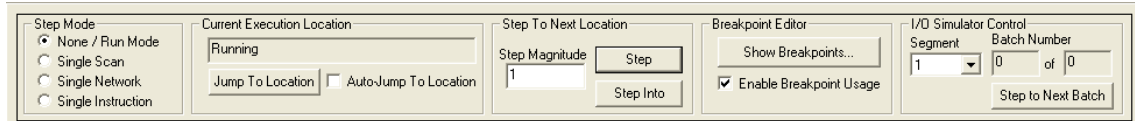
When Modicon Simulator is closed, it forces the current PLC scan to complete and then initiate the save sequence, retaining its state, if the Startup Restore option has been selected.

To set the **Startup Restore** feature, select the **Options \ Startup Restore** menu item within the Modicon Simulator Window.

When the simulator is started again, it will attempt to load its previous state, if it exists. This state will be saved to a file named *LastState.FTS* in the execution directory when exiting.

# USING THE SIMULATOR DEBUGGER TOOLBAR

The Simulator Debugger toolbar provides the ability to control the programming process offering features such as step debugging and use of breakpoints. This toolbar will automatically open within Modicon WorkShop when WorkShop is connected to the simulator.



This toolbar can be resized as well as docked to the top or bottom of the WorkShop window or free floating.

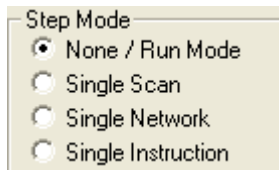
---

**NOTE:** To hide the Simulator Debugger toolbar, deselect the **View \ Toolbars \ Simulator Debugger** menu item.

---

## Step Mode

Step debugging allows each single scan, network, or instruction to be examined.

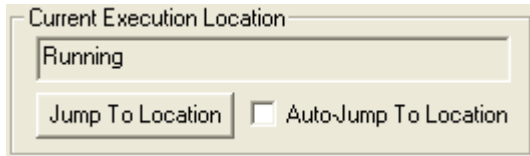


The following options for step debugging may be set:

Step Mode	The simulator will...
<b>None / Run Mode</b>	... execute each scan continuously, not stopping unless an enabled breakpoint is reached.
<b>Single Scan</b>	... stop execution between every scan. It will stop during a scan if an enabled breakpoint is reached.
<b>Single Network</b>	... stop execution between every network. It will stop during a scan if an enabled breakpoint is reached.
<b>Single Instruction</b>	... stop execution at each instruction in ladder as they are executed in order from top row to bottom row across each column from left to right.

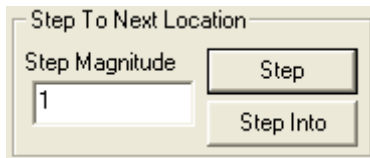
## Current Execution Location

If Modicon Simulator is not at a breakpoint location, the **Current Execution Location** field displays *Running*. If **Step Mode** is set to something other than **None / Run Mode**, this field displays the ladder execution location of the current step. If a breakpoint is set, this field displays the location in which the breakpoint occurred.



Press the **Jump To Location** button to view the program logic listed in the **Current Execution Location**. To always view the program logic from the current execution location, select the **Auto-Jump To Location** check box.

## Step to Next Location



### Stepping

Stepping allows program execution to be viewed one step at a time. See pg. 19-20 to learn how the different step modes affect stepping.

### Step Magnitude

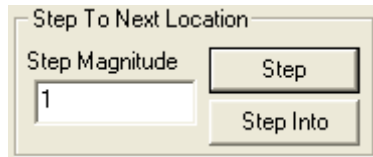
A number may be entered in the **Step Magnitude** field to set the number of steps taken before the program stops execution. For example:

- If **Single Scan** step mode has been selected, a step magnitude of 10 indicates that the program will scan 10 times before the execution is halted.
- If **Single Network** or **Single Instruction** has been selected, a step magnitude of 10 indicates that the scan will stop after the next 10 networks and/or instructions have been executed.



## Single Scan Step Mode

While in **Single Scan** step mode, click the **Step** button to step an entire PLC scan. If a breakpoint occurs, the scan will halt at the breakpoint location. In this mode, the **Step Into** button is unavailable.



### Step Magnitude

If **Single Scan** step mode has been selected, a step magnitude of 10 indicates that the program will scan 10 times before the execution is halted.

## Single Network / Single Instruction Step Mode

Use the following chart to understand the results of clicking the **Step** button while in **Single Network** or **Single Instruction** step mode.

Condition	Execution Location	Button	Result
No specific condition.	Ladder	Step	The execution will break at the next Ladder network in the same execution scope unless a breakpoint is hit on a different execution scope. (i.e. it will run until it reaches the next network that is not contained within a new subroutine, unless a breakpoint is hit.) If it is the last Ladder network to be executed, it will restart at the beginning.
If in <b>Network Step Mode</b> and current network contains a <b>JSR</b> or if in <b>Instruction Step Mode</b> and current execution location is at a <b>JSR</b> instruction that has power.	Ladder	Step Into	The execution will break at the first Ladder network, if in <b>Network Step Mode</b> , or at the first instruction in the first network, if in <b>Instruction Step Mode</b> , within the subroutine that is called.

Condition	Execution Location	Button	Result
If in <b>Network Step Mode</b> and current network contains a <b>JSR</b> or if in <b>Instruction Step Mode</b> and current execution location is at a <b>JSR</b> instruction but it does not have power.	Ladder	Step Into	The execution will break at the next Ladder network at the same execution scope unless a breakpoint is hit on a different execution scope. (i.e. it will run until it reaches the next network that is not contained within a new subroutine, unless a breakpoint is hit.) If it is the last Ladder network to be executed, it will break on the first network.

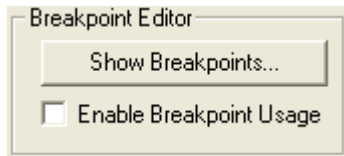
### Step Magnitude

If **Single Network** or **Single Instruction** has been selected, a step magnitude of 10 indicates that the scan will stop after the next 10 networks or instructions have been executed.

## Breakpoint Editor

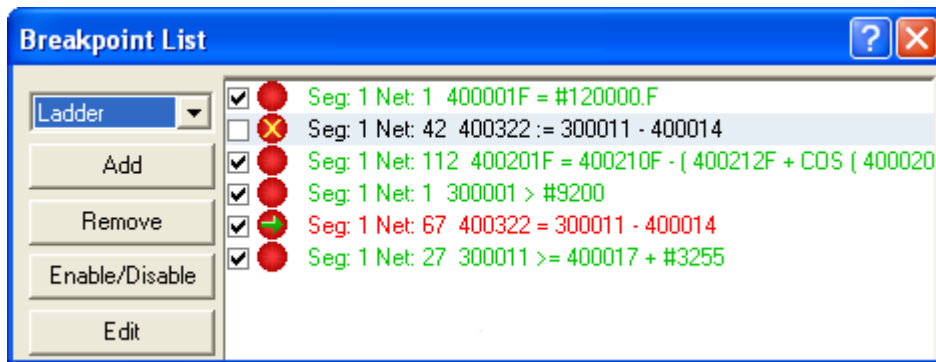
The Breakpoint Editor allows conditional or unconditional breakpoints to be set at any ladder network as well as unconditional address breakpoints that break when a specific address is modified. Conditional breakpoints allow an expression to be evaluated every time the scan reaches a breakpoint location.

Breakpoint expressions may be used for 0x, 1x, 3x, and 4x address types. Address breakpoints may also be set at any of these same address locations.



## Viewing Breakpoints

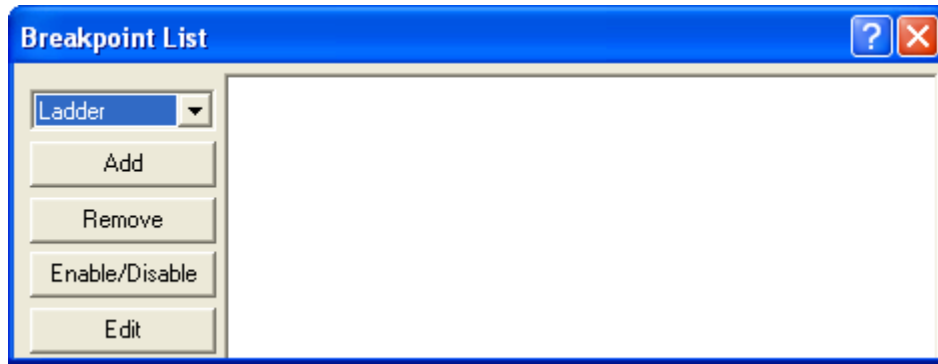
To view breakpoints that have been set, click the **Show Breakpoints** button. The **Breakpoint List** dialog appears. Breakpoints are listed according to type with separate lists for Ladder and Address. To view breakpoints of a particular type, select one of the following options from the combo box: **Ladder** or **Address**. See *Breakpoint Color/Icon Scheme*, pg. 26, for details on colors and icons used by Modicon Simulator.



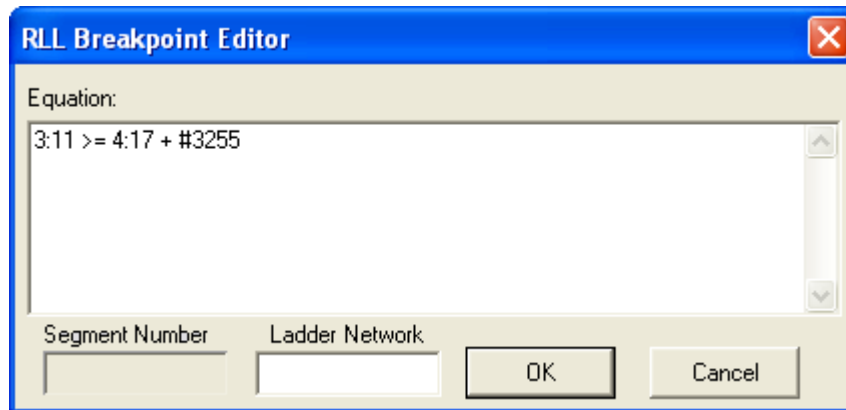
## Adding a Breakpoint

To add a breakpoint:

Access the **Breakpoint List** dialog by clicking the **Show Breakpoints** button on the **Simulator Debugger** toolbar. The **Breakpoint List** dialog appears.

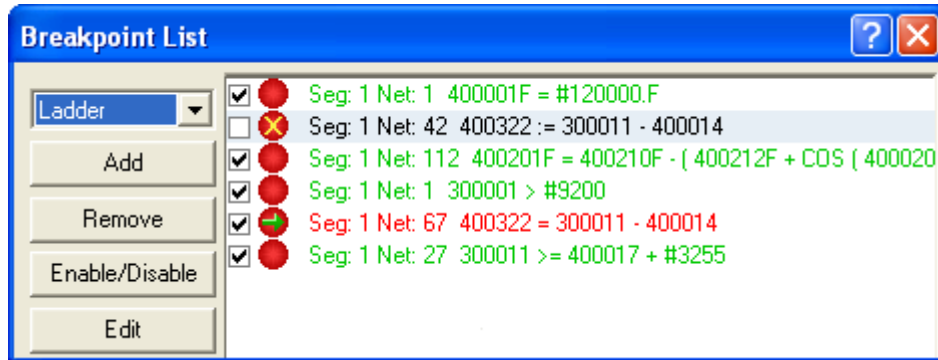


1. Select one of the following options from the combo box: **Ladder** or **Address**.
2. Click the **Add** button. A **Breakpoint Editor** dialog, similar to the following, appears.



4. For RLL breakpoints, within the **Ladder Network** fields, select the network location for the breakpoint. If **Network** mode is **By Segment**, the **Segment Number** field is active, otherwise it is disabled.
5. If the breakpoint being added will be conditional, enter an expression in the **Equation** field, which will be evaluated like a network equation, except the user is not required to assign a 4x address. If the breakpoint will be unconditional, leave the **Equation** field blank.
6. Click **OK**. If the expression is invalid, an error message will display. If the expression is parsed and found to be a valid statement, the breakpoint will be added to the **Breakpoint List** as the following example shows. Type and address are displayed followed by the expression that was entered. When the breakpoint has been added, the text will display in green font, the ● icon will

display, and a checkmark will indicate that the breakpoint is enabled. See **Breakpoint Color/Icon Scheme** for details on other colors and icons used by Modicon Simulator.




---

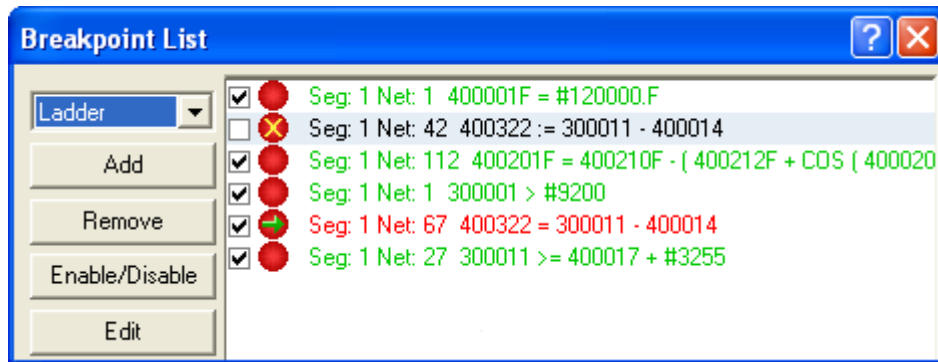
**NOTE:** Adding a breakpoint adds it to the **Breakpoint List**, but does not automatically enable it. In order for breakpoint usage to be enabled, the **Enable Breakpoint Usage** check box on the Simulator Debugger toolbar must be selected.

---

## Disabling an Individual Breakpoint

To disable a breakpoint:

1. Access the **Breakpoint List** dialog by clicking the **Show Breakpoints** button on the **Simulator Debugger** toolbar. The **Breakpoint List** dialog appears.

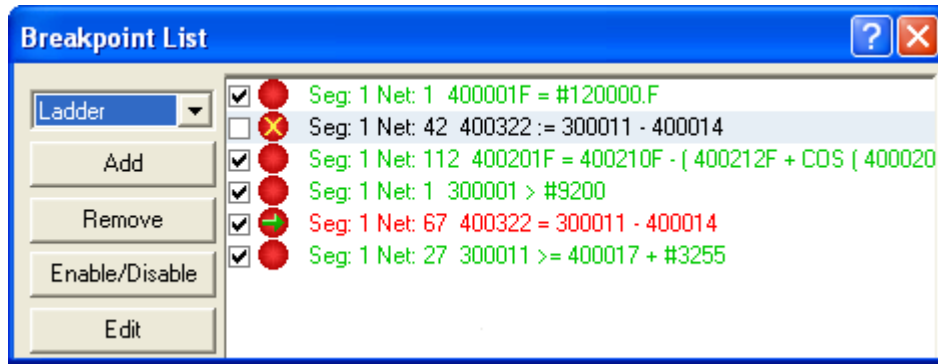


2. Deselect the checkmark associated with the breakpoint to be disabled or select the breakpoint and click the **Disable** button.

## Re-Enabling an Individual Breakpoint

To re-enable a disabled breakpoint:

1. Access the **Breakpoint List** dialog by clicking the **Show Breakpoints** button on the **Simulator Debugger** toolbar. The **Breakpoint List** dialog appears.

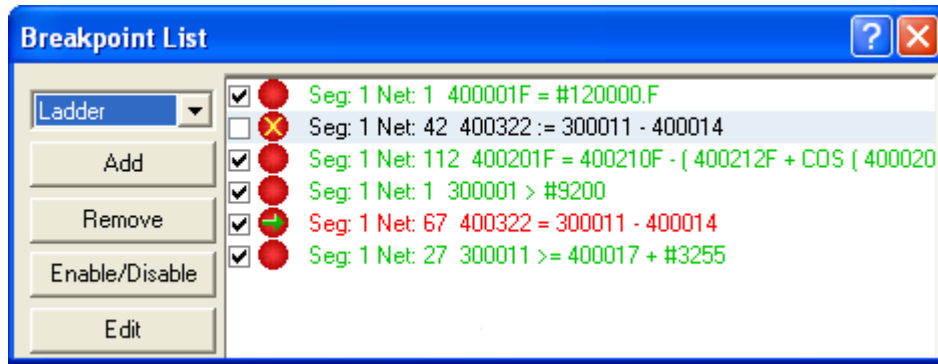


2. Select the check box preceding the breakpoint or select the breakpoint and click the **Enable** button.
3. Ensure the **Enable Breakpoint Usage** check box is selected on the **Simulator Debugger** toolbar.

## Removing a Breakpoint

To remove a breakpoint:

1. Access the **Breakpoint List** dialog by clicking the **Show Breakpoints** button on the **Simulator Debugger** toolbar. The **Breakpoint List** dialog appears.

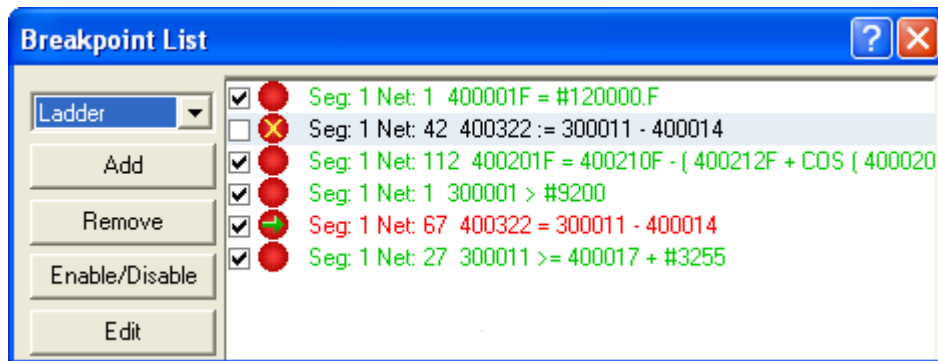


2. Select the breakpoint to be removed.
3. Click the **Remove** button. The breakpoint will be removed from the list, and any subsequent breakpoints in the list will shift to close the resulting gap.

## Editing a Previously Entered Breakpoint

To edit a breakpoint:

1. Access the **Breakpoint List** dialog by clicking the **Show Breakpoints** button on the **Simulator Debugger** toolbar. The **Breakpoint List** dialog appears.



2. Select the breakpoint to be edited and click the **Edit** button or double-click the text portion of the breakpoint. The **Breakpoint Editor** dialog will appear with the network address displayed for RLL breakpoints. If the **Network** mode is **By Segment**, the **Segment Number** is also displayed. For all types of breakpoints, the breakpoint expression will also display.
3. Make the appropriate changes and click **OK**. A new breakpoint, incorporating the edits, will be created.





4. Remove the old breakpoint. (See *Removing a Breakpoint*, pg. 25, for details.)

## Editing Ladder with Valid Breakpoints

When edits are made to the ladder logic the breakpoints will automatically adjust their location based on the changes that occur. For instance, if a breakpoint is set at ladder network 10 and a network is inserted before this location, the breakpoint network will automatically adjust itself to be set at network 11. This will allow the user to create breakpoints as edits are being made so that the user is not restricted to creating all breakpoints before entering Run Mode and adjusting them manually when additional changes are made.

## Breakpoint Color / Icon Scheme

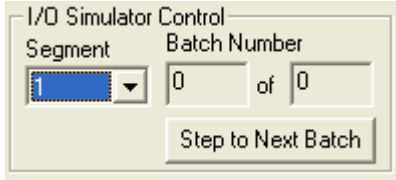
Within the **Breakpoint List**, text is displayed in a variety of colors with corresponding icons to indicate current status.

Font Color	Icon	Breakpoint Status	The breakpoint...
Green		Enabled	...will be evaluated when the desired location is reached.
Gray		Disabled	... will NOT be evaluated until it is re-enabled.
Red		Activated	...has been activated, the expression evaluated, the result is a non-zero value, and the process has been halted. The current location where the process stopped is displayed in the <b>Current Execution Location</b> group box of the <b>Simulator Debugger</b> toolbar. Click the <b>Step</b> button to continue program step execution or set the <b>Debug Mode</b> to <b>None / Run Mode</b> for continuous execution.
Yellow		Warning	...encountered an error and is disabled. This may occur when any error preventing the expression from properly evaluating has been encountered.



## I/O Simulator Control

The I/O Simulator Control box within the Simulator Debugger toolbar is used to control the execution of simulated I/O batches.



Select the a segment and click the **Step to Next Batch** button to execute a batch for which trigger conditions have been set to **Manual**.

See the following section for details on setting up batches and recipes using the I/O Simulator.



## USING THE I/O SIMULATOR

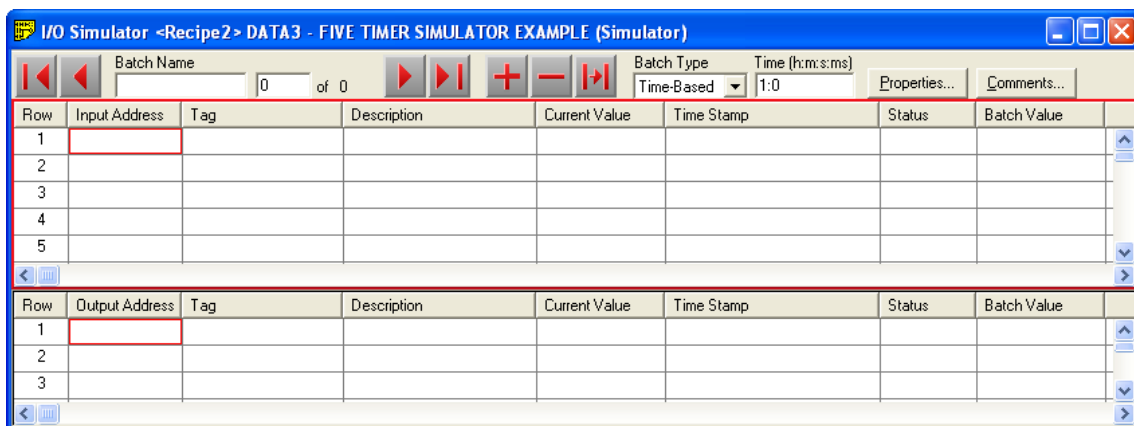
The I/O Simulator allows a sequence of automated I/O values to be created and deployed that are loaded directly by the simulator based on conditional triggers. Each I/O sequence is known as a recipe. A recipe consists of one to many data sets known as batches. A batch consists of a trigger, which executes the batch under certain conditions, and a list of addresses and values, which may correspond to input values, expected output values, or a combination of the two.

When the simulator enters **Run** mode or completes a recipe transfer while already in **Run** mode, it begins executing its recipe information for each segment that has been loaded. It will load the input values for the first batch after the conditions for its trigger are satisfied. These trigger conditions can be one of the following types:

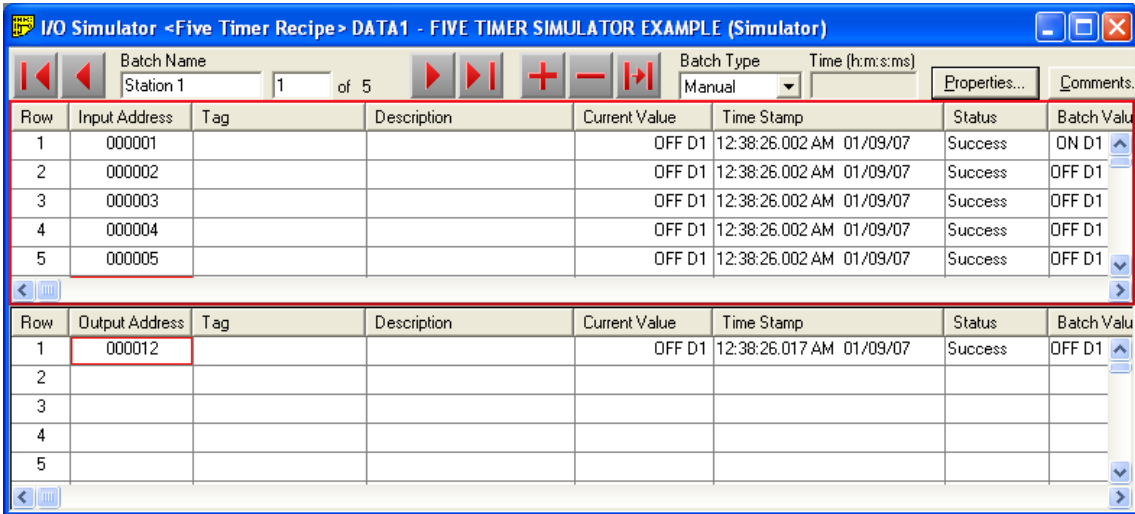
- **Manual:** Evaluated when the **Step to Next Batch** button on the Simulator Debugger toolbar is clicked
- **Scan-Based:** Evaluated after the specified number of scans has elapsed since the last batch in sequence was evaluated
- **Time-Based:** Evaluated after the specified amount of time has elapsed since the last batch in the sequence was evaluated

When a batch is evaluated, all of the input values are written to their specified addresses at the beginning of the segment execution. If no input addresses exist, then none are written. At the end of the same simulated PLC scan, the output addresses are read and compared against expected values. Any differences between the expected and actual values are logged in an error file. See *Viewing the Error Log*, pg. 39.

To access the I/O Simulator, within WorkShop select the **View \ I/O Simulator** menu item. The **I/O Simulator** Window appears.



The following I/O Simulator Window provides an example of an I/O batch that has been created:



Each batch is given a unique name of up to 32 characters. The edit field adjacent to the Batch Name field displays the number of the batch that is currently displayed in the I/O Simulator Window. In this example, *Station 1* is the first of five batches for a recipe file, which has been named *Five Timer Recipe*.

The following buttons on the I/O Simulator’s toolbar are used to navigate through the recipe’s batches:

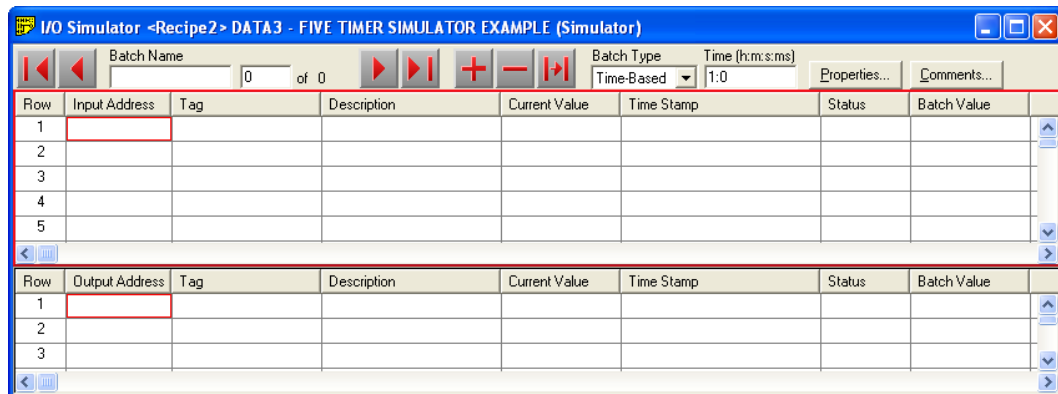
Button	When clicked...
	...the first batch will be displayed.
	...the previous batch will be displayed.
	...the next batch will be displayed.
	...the last batch will be displayed.

## Creating an I/O Batch

Batches are created by entering data in the I/O Simulator Window similar to how data is entered in WorkShop's Data Window. Addresses and values are entered that will be loaded as inputs and outputs.

To create an I/O batch:

1. Access the I/O Simulator by selecting the **View \ I/O Simulator** menu item. The **I/O Simulator** Window appears.



2. Within the **Batch Name** field, enter a name for the batch.
3. Within the top half of the **I/O Simulator** Window, enter an address in the **Input Address** field. If the address is valid, the simulator will populate the **Current Value** field with the current value for the address that has been entered and the **Batch Value** field with a value of 0.
4. Enter a value appropriate to the current format in the **Batch Value** column.
5. If desired, change the format of any row. The values will automatically convert into the new format.
6. If an expected output is desired, within the lower half of the **I/O Simulator** Window, enter an address in the **Output Address** field. If the address is valid, the simulator will populate the **Current Value** field with the current value for the address that has been entered and the **Batch Value** field with a value of 0.
7. Enter the value that is expected at the end of the scan when the batch executes into the **Batch Value** column.
8. Using the **Batch Type** combo box, select the type of condition that will trigger the batch. Choices include Manual, Scan-Based, or Time-Based. The field adjacent to the **Batch Type** field becomes enabled to accept a trigger setting that corresponds to the type of batch that has been selected. If the batch type is **Manual**, this trigger field is disabled.
9. If a **Batch Type** of **Scan-Based** or **Time-Based** has been selected, enter the appropriate trigger:

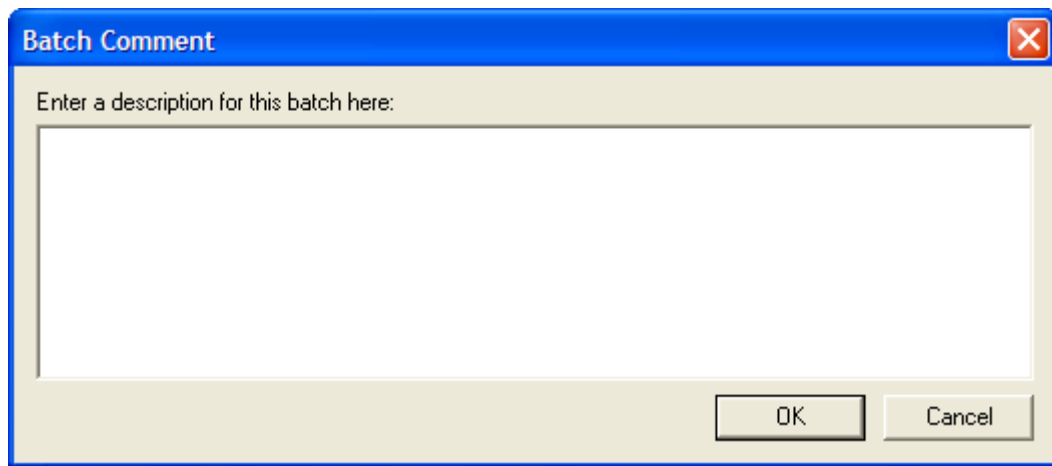
- **Scan-Based:** Enter the number of scans that will pass since the previous batch executed.
- **Time-Based:** Enter the amount of time between the previous batch's execution and the current batch's execution. This time is entered as hours, minutes, seconds, and milliseconds with each time component separated by a colon. For example, entering [0:0:2:500] means the batch will execute 2.5 seconds after the previous batch. If a [0] is entered, the batch will execute immediately following the previous batch's execution.



---

**NOTE:** Triggers are relative to the previous batch's execution. For example, if a batch has a trigger of 2 seconds, it will execute 2 seconds after the previous batch was executed. The trigger for the first batch is relative to the start of the I/O Simulator's execution.

---

10. If desired, click the **Comment** button to enter documentation. The **Batch Comment** dialog appears.



11. Enter up to 1000 characters into the text field to document information about the batch's purpose
12. Select the **I/O Simulator \ Add Batch** menu item or click the  toolbar icon, to add the new batch to the end of the list or select the **I/O Simulator \ Replace Batch** menu item or click the  toolbar icon, to replace the current batch with the new batch information that was just added.

## Managing Recipe Properties

Recipe properties and corresponding batch information may be conveniently managed and modified through use of a list control.

To manage recipe properties, with the **I/O Simulator** Window in focus within WorkShop select the **I/O Simulator \ Properties** menu item

OR

Click the **Properties** button within the **I/O Simulator** Window.

In either case, the **I/O Simulator Properties** dialog appears.

**I/O Simulator Properties**

Recipe Properties

Repeat After Last Batch (Cyclic Recipe)

Cyclic Recipe Type

Manual

Hours Minutes Seconds Milliseconds

Batch Manager

Number	Name	Type	Trigger
1	Station 1	Manual	-----
2	Station 2	Time-Based	0:0:5:0
3	Station 3	Time-Based	0:0:5:0
4	Station 4	Time-Based	0:0:10:0
5	Station 5	Time-Based	0:0:4:0

Move Up Move Down Delete View Comments...

Batch Trigger Type Batch Name

Hours Minutes Seconds Milliseconds

OK Cancel Apply

The **I/O Simulator Properties** dialog is used to create cyclic recipes, adjust batch properties, and reorder a recipe's batches.

## Creating Cyclic Recipes

A cyclic recipe is a recipe that, once run to completion, will restart itself at the first batch in the list. Similar to batches, cyclic recipes may utilize a trigger condition that will restart them. A recipe's trigger type may be **Manual**, **Scan-Based**, or **Time-Based**. For example, if the recipe cycle trigger type is set to Scan-Based and an argument of 10 is specified, the recipe execution will wait 10 scans before evaluating the trigger for the first batch. So, if the first batch had a trigger of 2 seconds, the following scenario would occur:

1. The last batch of the cyclic recipe is executed.
2. 10 simulated PLC scans are completed.
3. 2 seconds elapse.
4. The first batch of the cyclic recipe is executed.

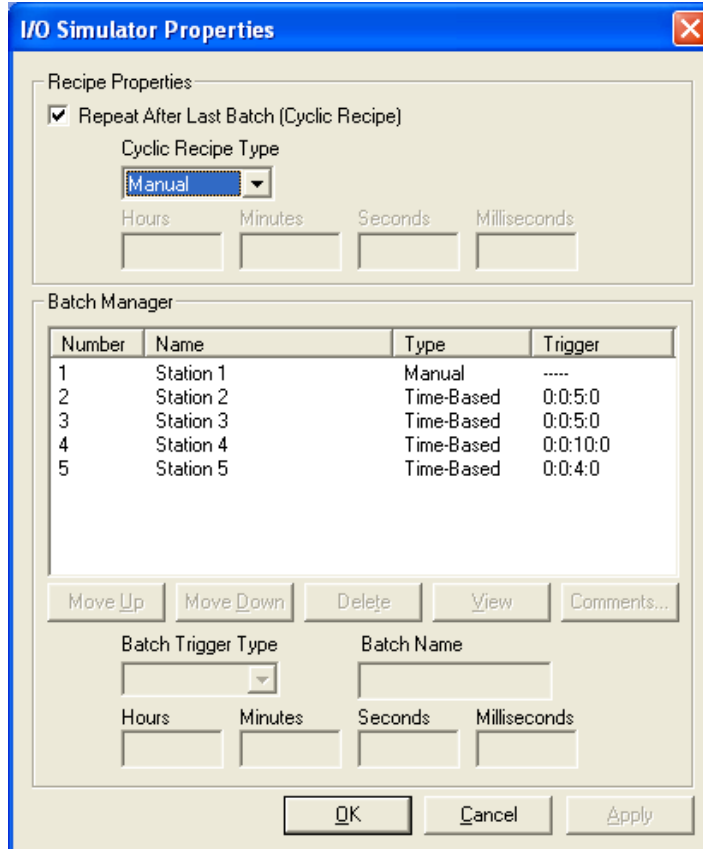
---

**NOTE:** The cyclic recipe's trigger is relative to the last batch in the sequence.

---

To create a cyclic recipe:

1. Access the **I/O Simulator Properties** dialog by selecting the **I/O Simulator \ Properties** menu item or by clicking the **Properties** button within the **I/O Simulator Window**. The **I/O Simulator Properties** dialog appears.





2. Within the **Recipe Properties** group box, select the **Repeat After Last Batch (Cyclic Recipe)** check box.
3. Using the **Cyclic Recipe Type** combo box, select the type of condition that will trigger the cyclic recipe. Choices include **Manual**, **Scan-Based**, and **Time-Based**. The field underneath the **Cyclic Recipe Type** combo box becomes enabled to accept a trigger setting that corresponds to the type of cyclic recipe that has been selected. If the cyclic recipe type is **Manual**, this trigger field is disabled.
4. If a **Cyclic Recipe Type** of **Scan-Based** or **Time-Based** has been selected, enter the appropriate trigger:
  - **Scan-Based:** Enter the number of scans that the recipe's execution will wait before the trigger for the first batch is evaluated.
  - **Time-Based:** Enter the amount of time between the execution of the last batch of the cyclic recipe and the execution of the first batch of cyclic recipe. This time is entered as hours, minutes, seconds, and milliseconds.

## Modifying Batch Properties

To modify a batch's properties:

1. Access the **I/O Simulator Properties** dialog by selecting the **I/O Simulator \ Properties** menu item or by clicking the **Properties** button within the **I/O Simulator Window**. The **I/O Simulator Properties** dialog appears.

**I/O Simulator Properties**

Recipe Properties

Repeat After Last Batch (Cyclic Recipe)

Cyclic Recipe Type

Manual

Hours Minutes Seconds Milliseconds

Batch Manager

Number	Name	Type	Trigger
1	Station 1	Manual	----
2	Station 2	Time-Based	0:0:5:0
3	Station 3	Time-Based	0:0:5:0
4	Station 4	Time-Based	0:0:10:0
5	Station 5	Time-Based	0:0:4:0

Move Up Move Down Delete View Comments...

Batch Trigger Type Batch Name

Hours Minutes Seconds Milliseconds

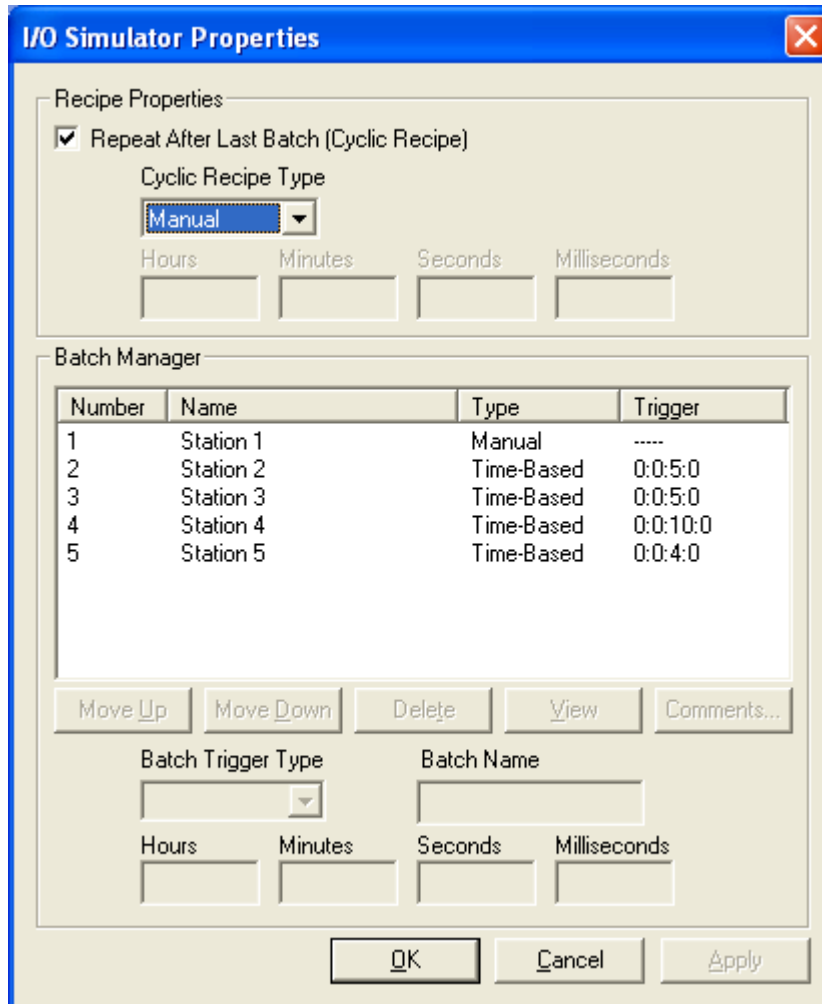
OK Cancel Apply

2. Within the **Batch Manager** group box, select a batch from the list of batches that have been created within the **I/O Simulator Window**. When an item is selected, the dialog's controls become active allowing modifications to be made.
3. Use the appropriate control to modify the batch's trigger type and value, name, and/or comments.
4. Click the **Apply** button to accept the changes and continue working in the **I/O Simulator Properties** dialog. Click the **OK** button to accept the changes and return to the **I/O Simulator Window**.

## Reordering a Recipe's Batches

To reorder a recipe's batches:

1. Access the **I/O Simulator Properties** dialog by selecting the **I/O Simulator \ Properties** menu item or by clicking the **Properties** button within the **I/O Simulator** Window. The **I/O Simulator Properties** dialog appears.



2. Within the **Batch Manager** group box, select a batch to be reordered from the list of batches that have been created within the **I/O Simulator** Window. When an item is selected, the dialog's controls become active allowing modifications to be made.

3. Click the **Move Up** button to adjust the batch's order by one position backward. Conversely, click the **Move Down** button to adjust the batch's order by one position forward. For example, if *Batch 2* is selected and the **Move Up** button is clicked, *Batch 2* becomes *Batch 1*. Conversely if *Batch 2* is selected and the **Move Down** button is clicked, *Batch 2* becomes *Batch 3*.

---

**NOTE:** The batch can also be selected, dragged, and dropped into a new location. To drag a batch to the last location, drag and drop the item below the last item in the list.

---

4. Click the **Apply** button to accept the changes and continue working in the I/O Simulator Window.

## Saving Recipe Files

After a recipe file has been created, it may be saved so that it may re-opened at a later time. The recipe will be saved with an \*.FTR file type.

To save a recipe file:

1. With the **I/O Simulator** Window in focus within WorkShop, select the **I/O Simulator \ Save Recipe** menu item.
2. If a recipe is being saved for the first time, a **Save As** dialog will appear and prompt for a file name.

## Opening Recipe Files

To open a previously created recipe file, with the **I/O Simulator** Window in focus within WorkShop, select the **I/O Simulator \ Open Recipe** menu item.

## Downloading Recipes

Before a recipe can be executed, it must be downloaded to Modicon Simulator.

To download a recipe:

1. With the **I/O Simulator** Window in focus within WorkShop, open the recipe file by selecting the **I/O Simulator \ Open Recipe** menu item.
2. Select the **I/O Simulator \ Download To Simulator** menu item. A status meter will appear displaying download progress.

## Executing Recipes

After a recipe has been downloaded, it will begin executing if the simulator is in **Run** mode and the segment into which the recipe has been loaded is scheduled to execute, or will begin executing as soon as the simulator is placed into **Run** mode and the segment into which the recipe has been loaded is scheduled to execute.

---

**NOTE:** To prevent a batch from executing immediately after entering Run mode or being downloaded, set the first batch trigger to **Manual**.

---

## Time-Based Batch Execution

The I/O Simulator only executes one batch per scan. If a batch is set with a trigger of x milliseconds, it will execute in the scan cycle following its previous batch, given that at least x milliseconds have elapsed since the last batch executed. The exception to this process is **zero trigger batches**.

## Zero Trigger Batches

Modicon Simulator allows up to 10 I/O batches to be executed in the same scan offering the ability to organize one set of data across multiple batch screens. A trigger is set for the first batch, and a trigger of 0 is set for each subsequent batch. Each batch is executed in the same scan as the previous batch if these triggers are set to 0. Subsequent batches, for which a trigger of 0 has been set, are referred to as Zero Trigger batches.

## Viewing the Error Log

As a recipe is executed, any differences between expected and actual values are logged in an error file in the install directory of Modicon Simulator. This ASCII text-based file is named *IOErrorReport.Log*.

After the recipe has executed, this file, which open in the PC's default text editor, may be manually opened in the install directory.

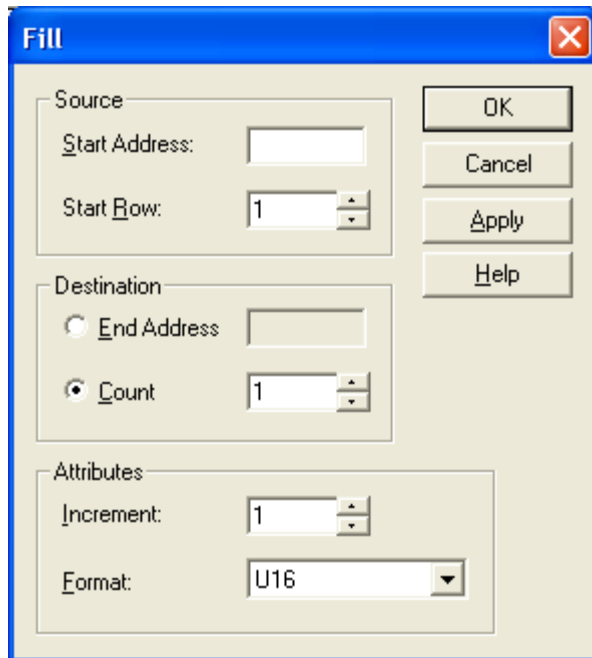
The log file may be manually cleared at any time and is cleared automatically whenever a recipe download sequence is completed.

## Data Window Fill

Use the Fill feature within the Data Window to monitor I/O values.

To fill the Data Window:

1. Access the **Data Window** by selecting the **View \ Data Window** menu item within WorkShop.
2. Select the row in which data is to be filled.
3. Select the **Data \ Fill** menu item. The **Fill** dialog appears.



- **Start Address:** Enter the first address in the series.
- **Start Row:** Enter the row number of the line that will hold the start address.
- **End Address:** To specify the range by address, select the **End Address** option button and enter the last address in the series.
- **Count:** To specify the range by number of lines/addresses, select the **Count** option button and enter the number of lines to be filled.
- **Increment:** Set the increment between addresses. For example, with an increment of 1, each address in the range is added to the log sheet. With an increment of 2, starting with an even numbered address, the even numbered addresses will be added to the log sheet.
- **Format:** Select the display format for the data. The options in this list include the size.
- **OK:** Fill the range and close the dialog.
- **Cancel:** Close the dialog without filling the range.

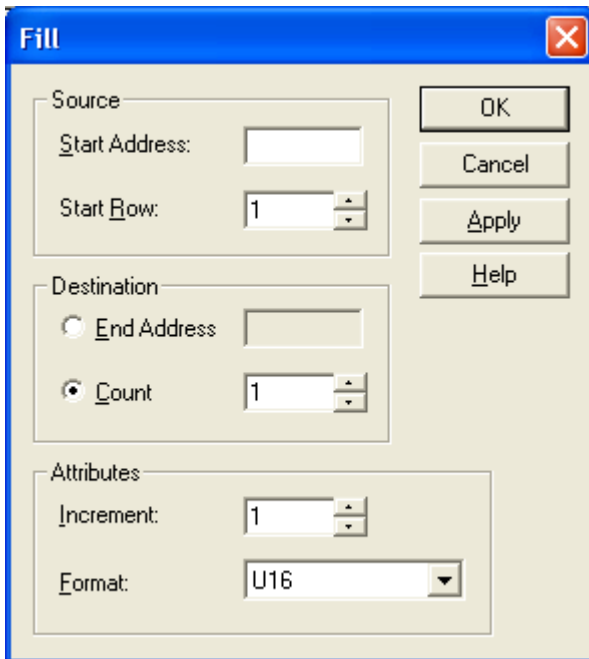
- **Apply:** Click the **Apply** button to fill the range without closing the dialog. The **Start Address** and **Start Row** fields will be updated to just past the end of the range specified previously. For example, if the start address was 000001, the start row 10, and the Count 10, after the **Apply** button is clicked, the new start address is 000011 and the new start row 20.

## I/O Simulator Window Fill

Use the Fill feature within the I/O Simulator Window to conveniently enter simulated I/O values.

To fill the I/O Simulator Window:

1. Access the **I/O Simulator Window** by selecting the **View \ I/O Simulator** menu item within WorkShop.
2. Select the row in which data is to be filled.
3. Select the **I/O Simulator \ Fill** menu item. The **Fill** dialog appears.



- **Start Address:** Enter the first address in the series.
- **Start Row:** Enter the row number of the line that will hold the start address.
- **End Address:** To specify the range by address, select the **End Address** option button and enter the last address in the series.
- **Count:** To specify the range by number of lines/addresses, select the **Count** option button and enter the number of lines to be filled.

- **Increment:** Set the increment between addresses. For example, with an increment of 1, each address in the range is added to the log sheet. With an increment of 2, starting with an even numbered address, the even numbered addresses will be added to the log sheet.
- **Format:** Select the display format for the data. The options in this list include the size.
- **OK:** Fill the range and close the dialog.
- **Cancel:** Close the dialog without filling the range.
- **Apply:** Click the **Apply** button to fill the range without closing the dialog. The **Start Address** and **Start Row** fields will be updated to just past the end of the range specified previously. For example, if the start address was 000001, the start row 10, and the Count 10, after the **Apply** button is clicked, the new start address is 000011 and the new start row 20.







# INDEX

## A

- Adding
  - Breakpoint ..... 16
- Adding..... 16

## B

- Batch..... 23, 27
- Batch Properties ..... 27
- Breakpoint
  - Adding ..... 16
  - Disabling..... 18
  - Editing ..... 19
  - Re-Enabling ..... 18
  - Removing ..... 19
  - Viewing ..... 16
- Breakpoint.....16, 18, 19
- Breakpoint Editor..... 16
- Breakpoint Status ..... 20

## C

- Communications ..... 11
- Communications Setup ..... 4
- Connecting.....6
- Connection Status ..... 11
- Current Execution Location ..... 13
- Cyclic Recipes ..... 25

## D

- Data Window ..... 30
- Debugger ..... 13
- Demo Mode Restrictions..... 1
- Disabling
  - Breakpoint ..... 18

- Disabling ..... 18
- Downloading
  - Recipes..... 29
- Downloading..... 29

## E

- Editing
  - Breakpoint ..... 19
- Editing ..... 19
- Error Log..... 29
- Executing
  - Recipes..... 29
- Executing..... 29

## F

- Features ..... 9
- Fill
  - I/O Simulator Window ..... 31
- Fill ..... 29, 30, 31

## I

- I/O Simulator ..... 21
- I/O Batch ..... 23
- I/O Simulator Control..... 20

## M

- Modicon Simulator
  - run ..... 1
  - Starting ..... 1
- Modicon Simulator..... 1

## O

- Opening
  - Recipe Files..... 29
- Opening..... 29
- Overview ..... 1

**P**

PLC Scan Status..... 11

PLC State

    Saving ..... 12

PLC State..... 12

PLC Status..... 11

PLC Status Window ..... 11

PLC Type

    Selecting..... 3

PLC Type..... 3

**R**

Recipe

    Opening..... 29

    Saving ..... 28

Recipe ..... 28, 29

Recipe Properties..... 24

Recipes

    Downloading..... 29

    Executing..... 29

Recipes..... 29

Re-Enabling ..... 18

Removing

    Breakpoint ..... 19

Removing ..... 19

Reordering..... 27

Run..... 1

**S**

Saving

    Recipe Files..... 28

Saving ..... 28

Screen Resolution ..... 1

Setup..... 4

Simulated Scan..... 9

Simulator Debugger toolbar ..... 20

Single Instruction ..... 15

Single Network ..... 15

Single Scan..... 14

Starting ..... 1

Step

    Next Location..... 14

Step ..... 14

Step Mode..... 13, 14, 15

**T**

TCP/IP..... 6

Timing Mode..... 11

**U**

Unsupported Features ..... 9

**V**

Valid Breakpoints ..... 19

Viewing

    Breakpoints..... 16

    Error Log..... 29

Viewing ..... 16, 29

