505 Simulator

Version 2.02

By FasTrak SoftWorks, Inc.

Throughout this document, 505 Simulator will be referred to as the simulator. PLC WorkShop for Siemens 505 will be referred to as 505 WorkShop, PLC WorkShop, or WorkShop.
505 Simulator and PLC WorkShop are trademarks of FasTrak SoftWorks, Inc.
505 is a trademark of Siemens.
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	http://www.fast-soft.com
Email	support@fast-soft.com
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

SYSTEM REQUIREMENTS / INSTALLATION

Hardware

To install 505 Simulator on your computer, you need the following hardware:

- A personal computer with an Intel Pentium 100 MHz processor or higher.
- 32 MB or more of RAM.
- An 800 x 600 VGA monitor with at least 256 colors.
- 25 MB free space on your hard drive.
- A mouse is recommended, but not required.

505 Simulator may not function properly on systems that are not 100% Intel compatible. Certain other hardware components and peripherals can create incompatibility problems.

Software

To install 505 Simulator on your computer, you need the following software loaded on your computer:

- Windows® 98, Windows NT, Windows 2000, or Windows XP.

Installing 505 Simulator

Before you begin installation, review the preceding *Hardware* and *Software* sections.

505 Simulator is installed locally with an administrator account. Follow these steps to install the software:

1. Insert the 505 WorkShop CD in your computer's CD-ROM drive.
2. The CD should autostart. If not, click the Windows **Start** button. Then click **Run**, and type `x:\setup.exe`, where `x` is the letter for the CD-ROM drive.
3. Follow the instructions that appear on the screen.

Setting Screen Resolution

To achieve optimum usability of the 505 Simulator, screen resolution must be set to at least 800 x 600 pixels. For best results, higher resolutions are recommended.

DEMO MODE RESTRICTIONS

If using this software without a license, 505 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, 505 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, 505 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

Additional Restricted Demo Mode Restrictions

While in Restricted Demo Mode, you will also be unable to:

- Exceed a maximum of 1 kilobyte of the following memory types: Ladder, Special, Variable, Shift Register, and One Shot
- Configure more than 64 drums, 1024 Timers/Counters, and 1024 Table Move Instructions

OVERVIEW

505 Simulator is the only software of its kind available for Siemens 505 and CTI 2500 PLCs. Use 505 Simulator to program, run, and test 505 ladder logic, special functions, PID loops, Analog alarms, and simulated I/O without using a PLC.

Used in conjunction with 505 WorkShop and data acquisition software such as ControlShop, 505 Simulator has been designed to assist programmers in testing and debugging Siemens 505 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 Siemens Simatic / TI PLC, as well as the CTI 2500 line of PLCs, 505 Simulator offers preset configuration for all 545 and 555 PLCs, 1101 through 1106, and CTI 2500 PLCs, revisions C100 through C400.

NOTE: For additional information on using 505 Simulator with a 545-1101 with one kilobyte of Bit I/O or non-545 or –555 Simatic PLC types, refer to *PLC Types*, pg. 8.

505 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, 505 Simulator allows specific simulator settings that have been created to be saved.

Powerful testing and debugging features include:

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

PLC TYPES

Using 505 Simulator for a 545-1101 PLC Program

If using the simulator for a 545-1101 with one kilobyte of Bit I/O, the following steps are required:

1. Within 505 WorkShop, select the **PLC Utilities / PLC Configuration** menu item.
2. Change the setting for **Bit I/O (K)** from **1** to **2**. While programming with the simulator, do not exceed one kilobyte of I/O.
3. After testing and debugging has been completed, switch this setting back to **1**.

Using 505 Simulator for a Simatic 505 PLC Program other than a 545 or 555

If using the simulator for a non-545 or -555 PLC program, the following steps are required:

1. Within the PLC Status dialog, select a 545 or 555 PLC type in the **PLC Type Selection** combo box. Not all instructions for the target PLC program will be available.
2. After testing and debugging has been completed, convert the program by changing the PLC type while offline. For older PLC types, for which configuration may not be compatible, perform the following actions to convert the program. Create a new program with the correct configuration settings, copy logic from the 545 or 555 simulated program, and import any documentation.

STARTING 505 SIMULATOR

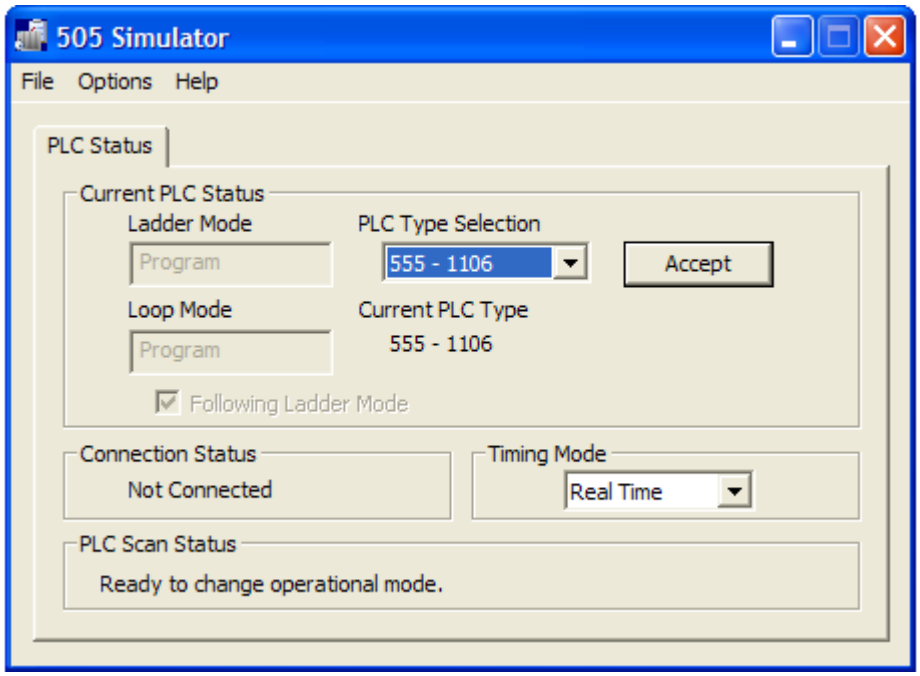
To run the 505 Simulator:

Within 505 WorkShop, select the **View / 505 Simulator** menu item or click the  toolbar icon.

OR

Select **Start / Programs / FasTrak SoftWorks / 505 Simulator.**

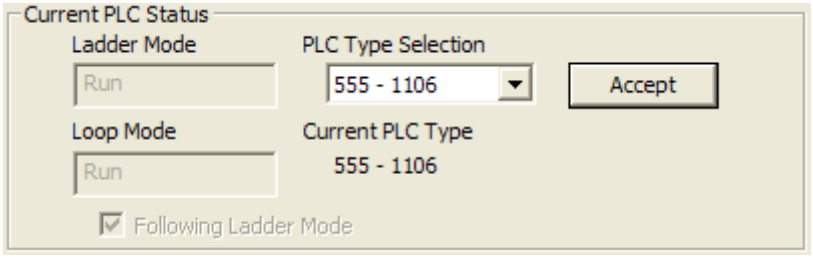
In either case, the **505 Simulator** Window appears.



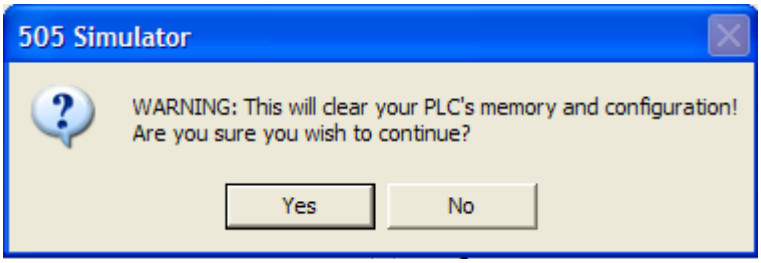
SELECTING PLC TYPE

The simulator will start up with the same default memory configuration as a new offline 555 – 1106 program in 505 WorkShop. To select another PLC type:

1. Select the desired PLC type from the **PLC Type Selection** combo box of the **PLC Status** Window.



2. Click the **Accept** button. The following warning will appear:



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

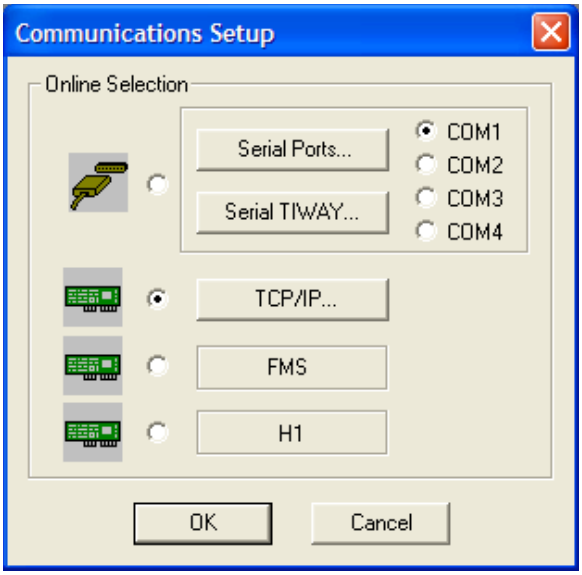
Changing PLC Type

When any client (i.e., 505 WorkShop) is connected to the simulator, the option for selecting PLC type will be grayed out and unavailable to prevent errors in the connected client. To change the PLC type, any attached clients must first be disconnected from the simulator.

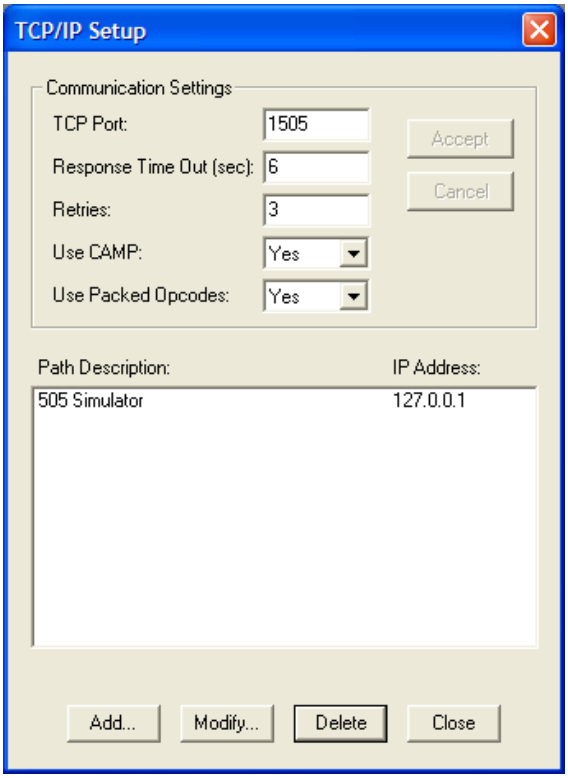
SETTING UP COMMUNICATIONS

505 Simulator uses the TCP/IP communication method for connection with 505 WorkShop. To create a new TCP/IP connection in WorkShop to connect to the simulator:

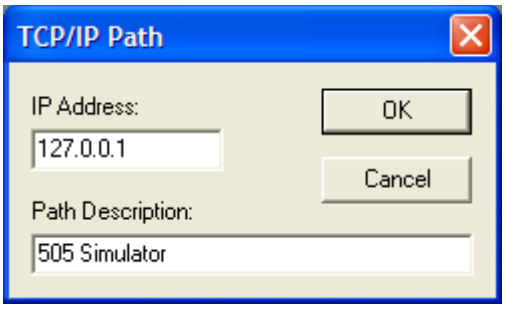
1. Select the **File/Communications Setup** menu option. The **Communications Setup** dialog appears.



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



3. Enter the communication settings (**TCP Port** must be set to **1505**) and click **Accept**.
4. Click the **Add** or **Modify** button within the IP Addresses field. The **TCP/IP Path** dialog appears.



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

NOTE: 505 Simulator supports the use of CAMP protocol and packed opcodes, which may be used to increase communication speeds.

CONNECTING TO 505 SIMULATOR

After 505 WorkShop and 505 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. Within 505 WorkShop, select the **File / Open** menu item or click the  toolbar icon.

NOTE: 505 Simulator must be running before attempting to connect to it.

PROGRAMMING FEATURES

In order to get the most benefit from 505 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 Compile

505 Simulator compiles both RLL and SF programs and subroutines before executing them.

SF Programs and Subroutines

The simulator will compile SF programs and subroutines after every instruction is validated if the SF being programmed is enabled. When programming SF programs and subroutines, it is recommended that they initially be created disabled. After they have been created and enabled, run-time edits can be made just like they are with an actual PLC.

Overview of Simulated Scan

Because 505 Simulator runs on a PC in a Windows environment, it will execute its scan somewhat different than an actual 505 PLC, which runs in a deterministic time frame.

Programs can be run in real time or simulated time. Refer to *Timing Mode*, pg. 17

NOTE: If program timing is critical to a process, it is recommended that it be validated on a 505 PLC before going live.

Simulated execution occurs in the following order:

1. I/O portion of logic in Task 15

The simulator reserves RLL Task Number 15 for programming simulated I/O. Logic can be created in this space to write to the I/O registers, which will be modified during the I/O portion of the scan. Task 15 simulated I/O logic can be saved, but it must be deleted before the program is loaded in an actual PLC.

2. Cyclic RLL (according to execution timer)
3. Main RLL
4. SF portion of Analog Scan

NOTE: Any SF programs and subroutines that are queued to run, will run to completion during a single scan and not obey the rules of time slicing. Very large, long-running SFP or SFS may take several real-time seconds to execute.

UNSUPPORTED FEATURES

Although the following features are not supported, PLC programs containing them can be simulated. 505 Simulator will store any applicable edits that are made in programs that are connected to it.

- 505 I/O is not supported, but can be simulated using the I/O Simulator. 505 I/O can also be configured and will be stored by 505 Simulator for file saving purposes.
- 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 505 Simulator for file saving purposes.
- External User Subroutines are not supported, but User Memory (U-Memory) may be programmed and will be stored by 505 Simulator for file saving purposes.
- Passwords may be entered, but will not affect access to programming features, except for initial connection, and will be stored by 505 Simulator for file saving purposes.
- Interrupt RLL is not supported, but may be programmed.

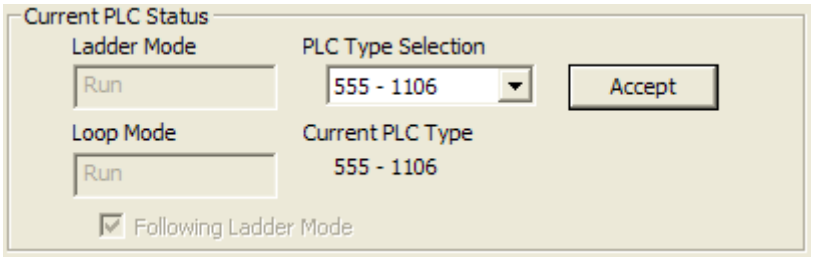
USING THE SIMULATOR PLC STATUS WINDOW

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

Current PLC Status

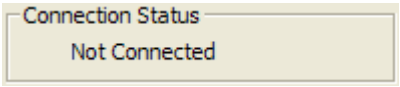
The **Current PLC Status** section of the PLC Status Window has the following functions:

- Indicates status of **Ladder Mode** and **Loop Mode**
- Checkmark in the **Following Ladder Mode** option box indicates that the **Loop Mode** is locked to the **Ladder Mode**
- Allows PLC type to be selected. Refer to *Selecting PLC Type,* pg. 10 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 505 Simulator.

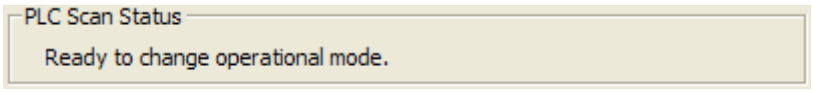


NOTE: Connection to more than 10 clients at a time is not recommended.

PLC Scan Status

The **PLC Scan Status** field indicates when the following operations, which change the state of the simulator, may be performed without error.

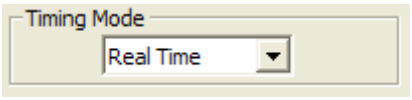
- Changing state from Run to Program
- Changing state from Edit to Program
- Changing state from Edit to Run
- Enabling a Special Functions Program, Subroutine, Loop, or Alarm
- Disabling a Loop or Alarm



NOTE: If an attempt is made to perform one of these operations while the simulator is running, an error message will display offering the choice to force the scan to finish in order to perform the operation.

Timing Mode

Simulated scans can be based on real time or simulated time. Use the **Timing Mode** combo box of the **PLC Status** dialog to select real time or simulated time.



Real Time Mode

505 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.

Simulated Time Mode

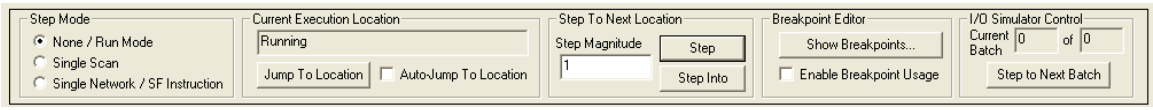
505 Simulator uses the configured fixed scan time, which has a default setting of 50ms. To change this setting, when connected with the simulator, select the **PLC Configuration / Scan Time** menu item in 505 WorkShop. Simulated Time Mode is recommended for users who wish to achieve more accurate timing relative to the execution of other parts of the process. With this time setting, the speed of the PC in correlation with the configured fixed scan time may cause timers within a program to advance much faster or much slower than real time.

Timing Limitations

- Time slicing is not supported. This means that any SF programs or subroutines queued in a given scan will run to completion during the scan.
- 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 505 PLC cannot be measured, 505 Simulator will not update or set the scan overrun indicator or scan watchdog timer.

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 505 WorkShop when WorkShop is connected to the simulator.

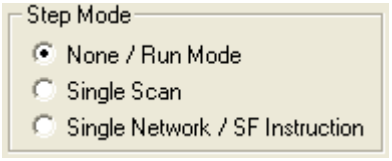


This toolbar can be resized as well as docked to the top or bottom of the WorkShop window or free floating.

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

Step Mode

Step debugging allows each individual network or SF instruction to be examined.

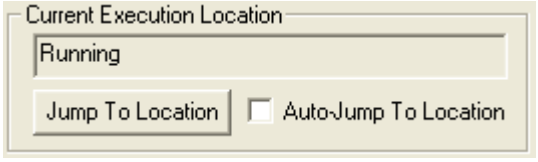


The following options for step debugging may be set:

Step Mode	The simulator will...
None / Run Mode	... execute each scan continuously, not stopping unless an enabled breakpoint is reached.
Single Scan	... stop execution between every scan. It will stop during a scan if an enabled breakpoint is reached.
Single Network / SF Instruction	... stop execution between every RLL network and every SF program or subroutine instruction.

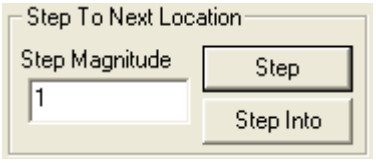
Current Execution Location

If 505 Simulator is not at a breakpoint location, the **Current Execution Location** field displays *Running*. If a breakpoint has been reached, this field displays the RLL execution location or SF Instruction and SF Line Number where the breakpoint has occurred.



Press the **Jump To Location** button to view the program logic where the breakpoint occurred. To always view the program logic when a breakpoint occurs, 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. When a program contains SF, stepping passes over SF program and subroutines. Stepping into allows program execution of the SF to be viewed.

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 / SF Instruction** has been selected, a step magnitude of 10 indicates that the scan will stop after the next 10 networks and/or SF 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.

Single Network / SF Instruction Step Mode

Use the following chart to understand the results of clicking the **Step** button while in **Single Network / SF Instruction** step mode:

Condition	Execution Location	Button	Result
No specific condition	RLL	**Step**	The execution will break at the next RLL 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 RLL network to be executed, it will restart at the beginning unless a breakpoint is hit while executing the analog scan.

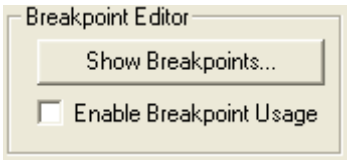
Condition	Execution Location	Button	Result
Current network contains a GTS, PGTS, or PGTSZ instruction that has power flow	RLL	**Step Into**	The execution will break at the first RLL network within the subroutine that is called.
Current network contains an inline SFPGM or SFSUB instruction for a valid, compiled SF, that has power flow	RLL	**Step Into**	The execution will break at the first SF Instruction in the SF number that is being executed.
Current network contains any instructions, which call a subroutine but do not receive power flow, or any other instructions	RLL	**Step Into**	The execution will break at the next RLL 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 RLL network to be executed, it will break on the first SF instruction that is executed from the queue.
An error occurs in the current SF Instruction and "continue on error" is not set for the SF being executed.	SF	**Step** or **Step Into**	The execution will break on the next RLL network unless another SF is executed inline before the next RLL network is executed.
Any condition other than an error	SF	**Step**	The execution will break at the next SF instruction 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 SF instruction that is not contained within a new subroutine, unless a breakpoint is hit.) If it is the last SF instruction to be executed, it will break at the next SF instruction/network, depending on where the SF was called from, unless another SF is executed inline before the next RLL network is executed.
Current Instruction is a CALL instruction that references a valid	SF	**Step**	The execution will break at the first SF Instruction in the SFS number that is being executed.

SFS			
Current Instruction is anything other than a valid CALL instruction	SF	**Step Into**	The execution will break at the next SF instruction or RLL network that it reaches.

Breakpoint Editor

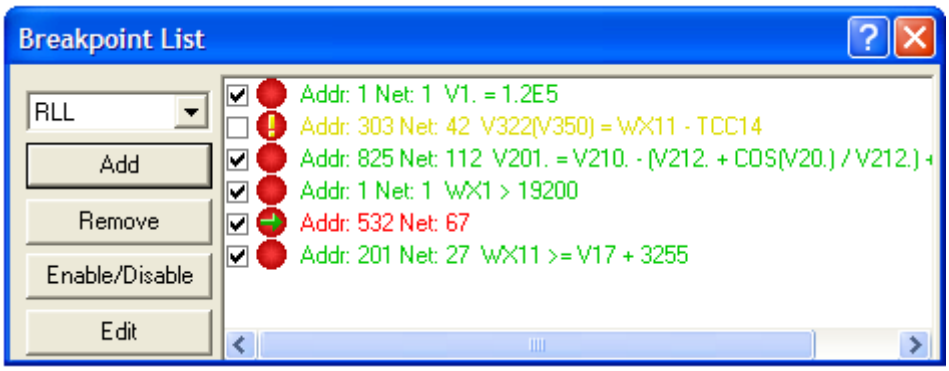
The **Breakpoint Editor** allows conditional or unconditional breakpoints to be set at any RLL network address or SF program / subroutine number and line number. Conditional breakpoints function exactly like a Special Function "IF" instruction, allowing an expression to be evaluated every time the scan reaches a breakpoint location.

Conditional breakpoint expressions may be used for the following address types: V, K, TCC, TCP, DSP, DCP, DCC, PSWA/W, PSIR/B, WX/WY, C, X/Y, and Constants. Address breakpoints may also be set at any of these same address locations excluding Constants.



Viewing Breakpoints

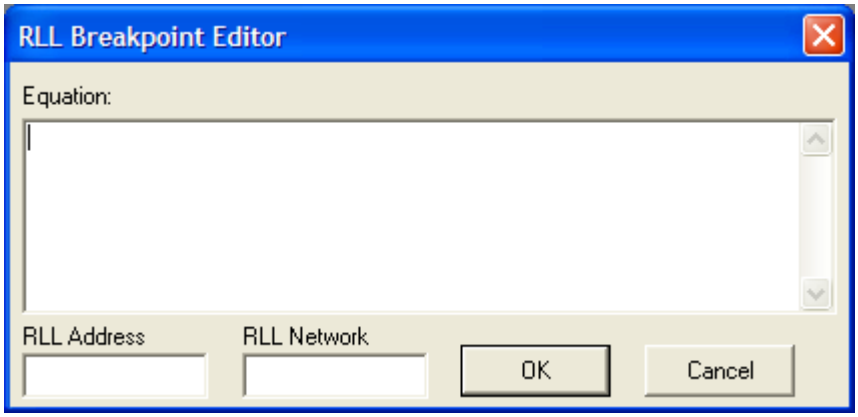
To view breakpoints that have been set, click the **Show Breakpoints** button. The **Breakpoint List** dialog appears. Breakpoints are listed according to type with separate lists for RLL, SF, and Address. To view breakpoints of a particular type, select one of the following options from the combo box: RLL, SF Program, SF Subroutine, or Address. Refer to *Breakpoint Color/Icon Scheme,* pg. 26 for details on colors and icons used by 505 Simulator.



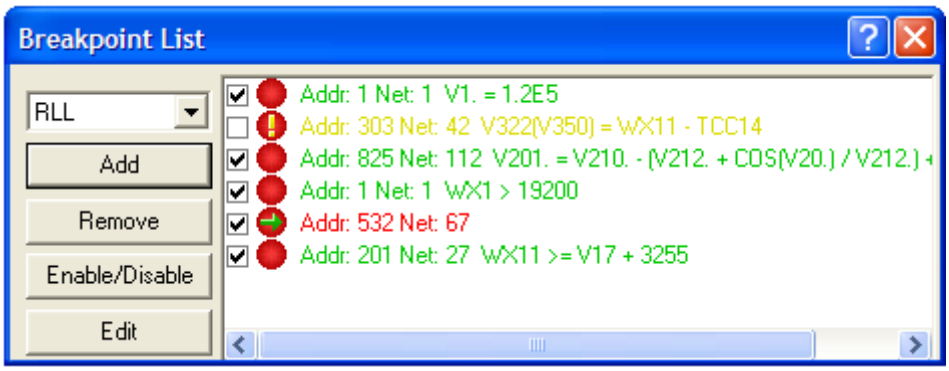
Adding A Breakpoint

To add a breakpoint:

1. Access the **Breakpoint List** dialog by clicking the **Show Breakpoints** button on the Simulator Debugger toolbar.
2. Select one of the following options from the combo box: RLL, SF Program, SF Subroutine, or Address.
3. Click the **Add** button. A **Breakpoint Editor** dialog, similar to the following, appears.



4. For RLL breakpoints, within the **RLL Address** or **RLL Network** fields, select the specific address or network location for the breakpoint. If 505 WorkShop is in **Address** mode, only the RLL Address field will be available. If 505 WorkShop is in **Network** mode, both the RLL Address and RLL Network fields will be available. When one of these two fields is edited, the other one become unavailable until any changes are deleted. For SF breakpoints, the dialog offers fields for program or subroutine numbers and line numbers instead of network address. Within the **Program Number** or **Subroutine Number** and **Line Number** fields, select the specific location for the breakpoint.
5. If the breakpoint being added will be conditional, enter an expression in the **Equation** field, which will be evaluated like an "IF" statement. 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 or purple font, the  icon or  icon will display, and a checkmark will indicate that the breakpoint is enabled. Refer to *Breakpoint Color/Icon Scheme,* pg. 26 for details on other colors and icons used by 505 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:

From the **Breakpoint list**, deselect the check box 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. From the **Breakpoint List**, select the check box associated with the breakpoint or select the breakpoint and click the **Enable** button.
2. Ensure the **Enable Breakpoint Usage** check box is selected on the Simulator Debugger toolbar.

Removing a Breakpoint

To remove a breakpoint:

1. From the **Breakpoint List**, select the breakpoint to be removed.
2. 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. From the **Breakpoint List**, 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 and SF number and line number displayed for SF breakpoints. For all types of breakpoints, the breakpoint expression will also display.
2. Make the appropriate changes and click **OK**. A new breakpoint, incorporating the edits, will be created.
3. Remove the old breakpoint by following the instructions in the preceding *Removing a Breakpoint* section.

Editing Ladder / SF with Valid Breakpoints

When RLL or SF logic is edited, any associated breakpoints will automatically adjust their location. For example, if a breakpoint is set at ladder address 10, and a network is inserted before this location with a length of 5, the breakpoint address will automatically adjust to a new setting of location 15.

Invalid Breakpoint Locations

If a breakpoint is placed on one of the following instructions, used as a flag to control the state or location where a program is compiled, the breakpoint will be enabled BUT WILL NOT BE EVALUATED.

- END
- NOP
- TASK
- TEXT

NOTE: Breakpoints cannot be used for networks containing these instructions.

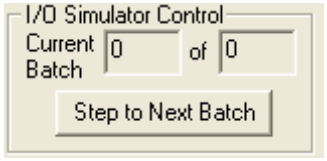
Breakpoint Color/Icon Scheme

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

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 **Current Execution Location** group box of the Simulator **Debugger** toolbar. Click the **Step** button to continue program step execution or set the **Debug Mode** to **None / Run Mode** for continuous execution.
Yellow		Warning	**…has been disabled upon encountering a Special Function error. Hover the mouse cursor over the breakpoint to see the SF error.**
Purple		Pending/Enabled	…has been entered with no frame of reference. This occurs when an RLL breakpoint is entered when in **Edit** or **Program Mode** or when a Special Function breakpoint is added to a disabled SFP or SFS. **Pending / Enabled** breakpoints can be disabled to enter a **Pending / Disabled** state. Re-enabling the breakpoint will cause it to enter a **Pending / Enabled** state unless: ▪ **RLL Pending / Enabled** breakpoints will become **Enabled** when the simulator transfers to **Run Mode**, and **SF Pending / Enabled** breakpoints will become **Enabled** when the SFP or SFS is enabled. ▪ **RLL Pending / Disabled** breakpoints will become **Disabled** when the simulator transfers to **Run Mode**, and **SF Pending / Disabled** breakpoints will become **Disabled** when the SFP / SFS is enabled.
		Pending/Disabled	…has been disabled from a **Pending / Enabled** state.

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.



Click the **Step to Next Batch** button to execute a batch for which trigger conditions have been set to **Manual**.

For details on setting up batches and recipes, refer to *Using the I/O Simulator*, pg. 28.

USING THE I/O SIMULATOR

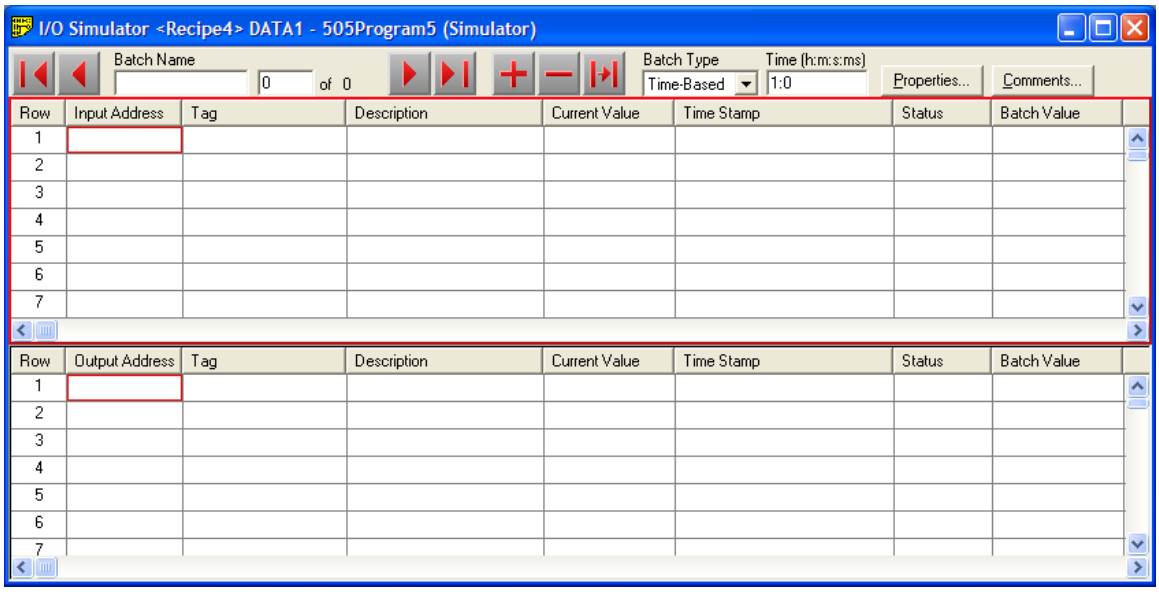
The I/O Simulator allows a sequence of automated I/O values to be created and deployed that are loaded directly by the simulator based on conditional triggers. Each I/O sequence is known as a **recipe**. A recipe consists of one to many data sets known as **batches**. A batch consists of a trigger, which 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** or **Edit** mode, it begins executing its recipe information. 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:

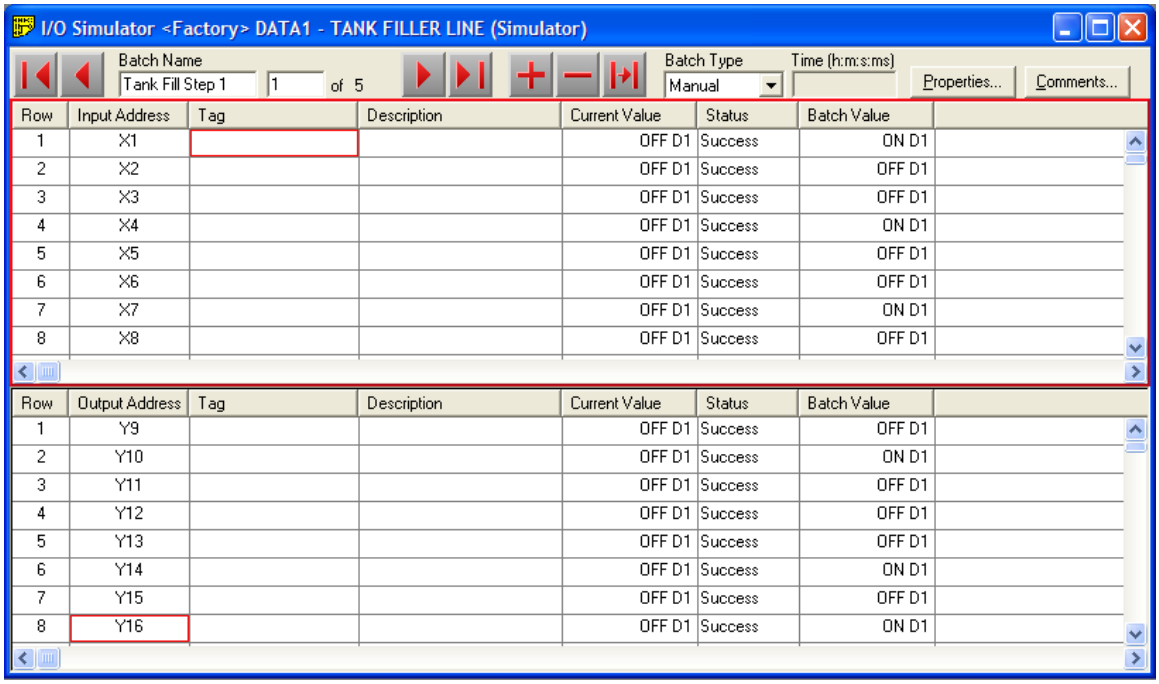
- **Manual:** Evaluated when the **Step to Next Batch** button on the Simulator Debugger toolbar is clicked
- **Scan-Based:** Evaluated after the specified number of scans has elapsed since the last batch in sequence was evaluated
- **Time-Based:** Evaluated after the specified amount of time has elapsed 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 scan. 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. Refer to *Viewing the Error Log*, pg. 35.

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, *Tank Fill Step 1* is the first of five batches for a recipe file, which has been named factory.

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. To automatically enter data using the fill feature, refer to *Using I/O Configuration Fill*, pg. 36. Addresses and values are entered that will be loaded as inputs and outputs.

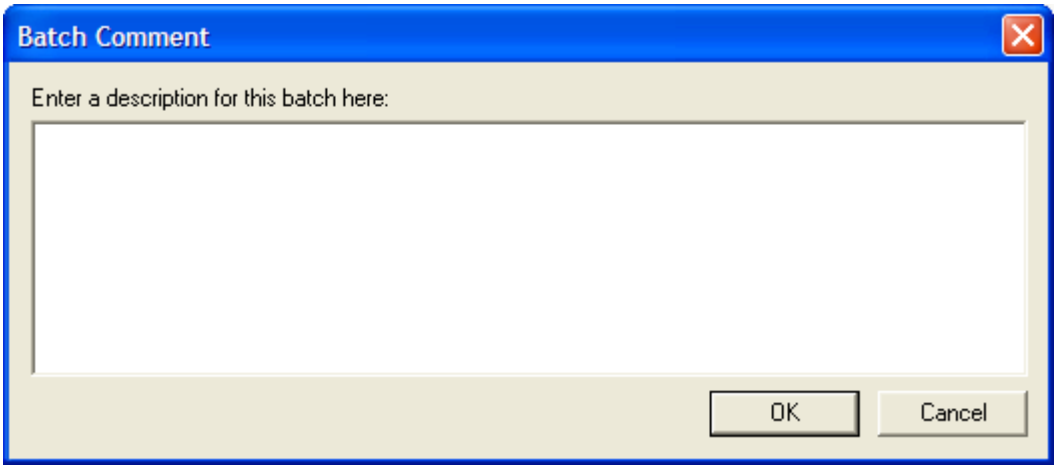
To create an I/O batch:

1. Within the **Batch Name** field, enter a name for the batch.
2. 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`.
3. Enter a value appropriate to the current format in the **Batch Value** column.
4. If desired, change the format of any row. The values will automatically convert into the new format.
5. 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`.
6. Enter the value that is expected at the end of the scan when the batch executes into the **Batch Value** column.
7. 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.
8. 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.

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



10. Enter up to 1000 characters into the text field to document information about the batch's purpose

11. Select the **I/O Simulator / Add Batch** menu item or click the toolbar button, 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 button, to replace the current batch with the new batch information that was just added.

Managing Recipe Properties

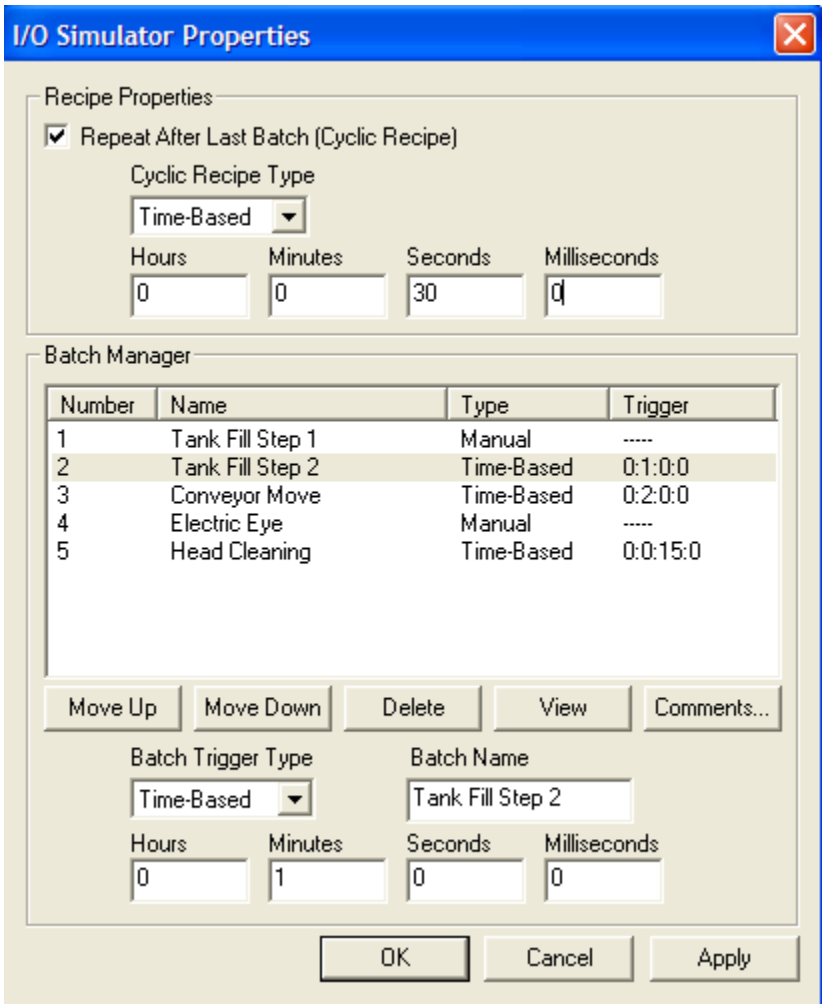
Recipe properties and corresponding batch information may be conveniently managed and modified through use of a list control.

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.



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

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. Within the **Recipe Properties** group box, select the **Repeat After Last Batch (Cyclic Recipe)** check box.
2. 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.
3. 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.

To modify a batch's properties:

1. 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.
2. Use the appropriate control to modify the batch's trigger type and value, name, and/or comments.

3. 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.

To reorder a recipe's batches:

1. 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.
2. 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.

3. 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.

Saving Recipe Files

After a recipe file has been created, it may be saved so that it may reopened 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 505 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 or will begin executing as soon as the simulator is placed into **Run** mode.

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. Exception: zero trigger batches.

Zero Trigger Batches

505 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 505 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.

USING I/O CONFIGURATION FILL

505 and Profibus I/O points may be automatically entered into the Data or I/O Simulator Windows through use of the Fill feature.

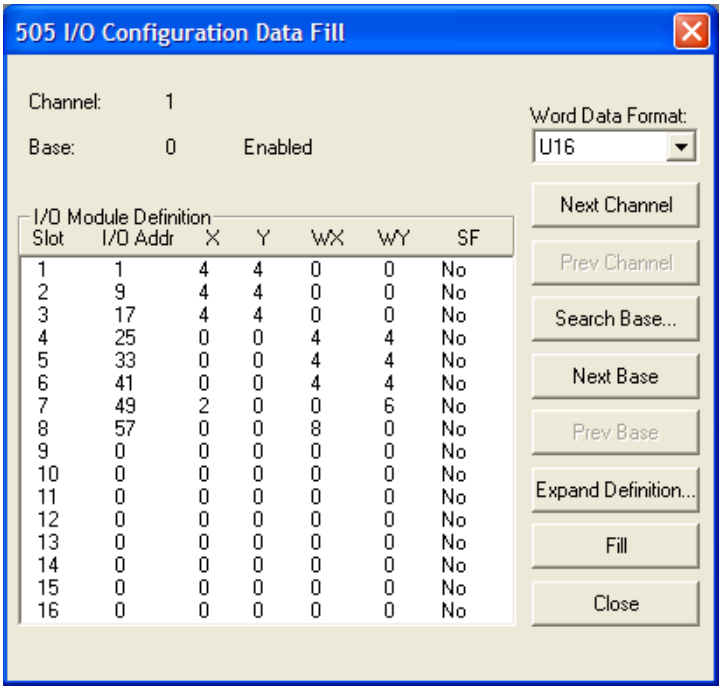
505 I/O Fill

Data Window

Use the 505 I/O fill feature within the Data Window to monitor 505 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 / 505 I/O Fill** menu item. The **505 I/O Configuration Data Fill** dialog appears, containing a list of slots in the current Base and controls to navigate through the additional bases.



4. Select a slot.
5. Click the **Fill** button. The corresponding data is automatically inserted into the **Data Window** in the row that has been selected. When a **WX** or **WY** address is inserted, it will automatically change its display format to the format that is currently selected in the **Word Data Format** combo box of the **505 I/O Configuration Data Fill** dialog.

- If the **Insert** keyboard key is selected (OVR displays in the **Status Line** within WorkShop) existing addresses in the Data Window will be overwritten when new data is entered.
- If the **Insert** keyboard key is toggled off, existing addresses in the Data Window will be moved down to make room for the data that has been inserted.

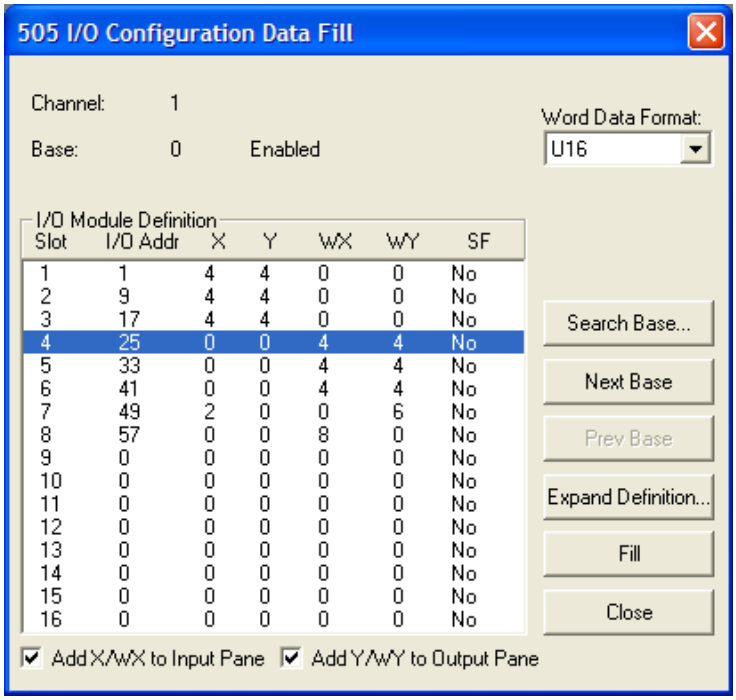
NOTE: When used with a 565 PLC type, the 505 I/O Configuration Data Fill dialog allows access to multiple channels.

I/O Simulator Window

Use the 505 I/O 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 / 505 I/O Fill** menu item. The **505 I/O Configuration Data Fill** dialog appears, containing a list of slots in the current Base and controls to navigate through the additional bases.



4. Select a slot.

5. Click the **Fill** button. If the **Add X/WX to Input Pane** check box is selected, **X/WX** addresses are automatically inserted into the **Input** pane of the I/O Simulator Window. If the **Add Y/WY to Output Pane** checkbox is selected, **Y/WY** addresses are automatically inserted into the **Output** pane of the I/O Simulator window. When a **WX** or **WY** address is inserted, it will automatically change its display format to the format that is currently selected in the **Word Data Format** combo box of the **505 I/O Configuration Data Fill** dialog.

- If the **Insert** keyboard key is selected (OVR displays in the **Status Line** within WorkShop) existing addresses in the Data Window will be overwritten when new data is entered.
- If the **Insert** keyboard key is toggled off, existing addresses in the Data Window will be moved down to make room for the data that has been inserted.

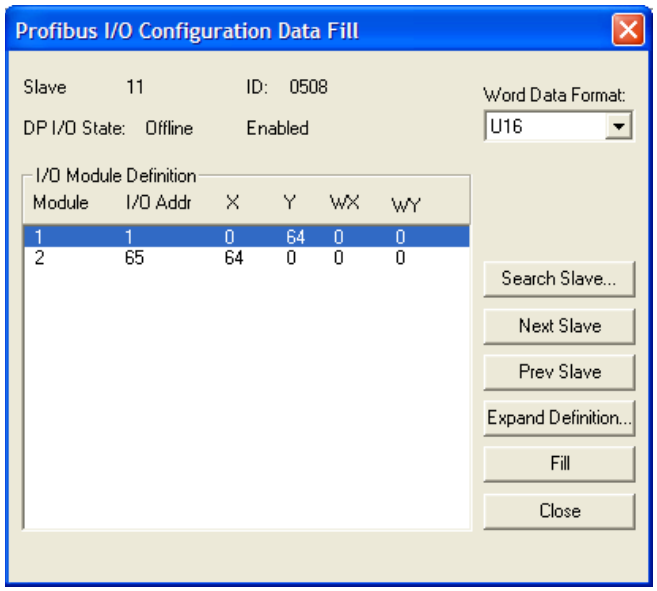
Profibus I/O Fill

Data Window

Use the Profibus I/O fill feature within the Data Window to monitor Profibus 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 / Profibus I/O Fill** menu item. All configured slaves are read, and the configuration is updated. The **Profibus I/O Configuration Data Fill** dialog appears, containing a list of configured modules in the first configured slave detected and controls to navigate through additional slaves.



4. Select a configured module.
5. Click the **Fill** button. The corresponding data is automatically inserted into the **Data Window** in the row that has been selected. When a **WX** or **WY** address is inserted, it will automatically change its display format to the format that is currently selected in the **Word Data Format** combo box of the **Profibus I/O Configuration Data Fill** dialog.

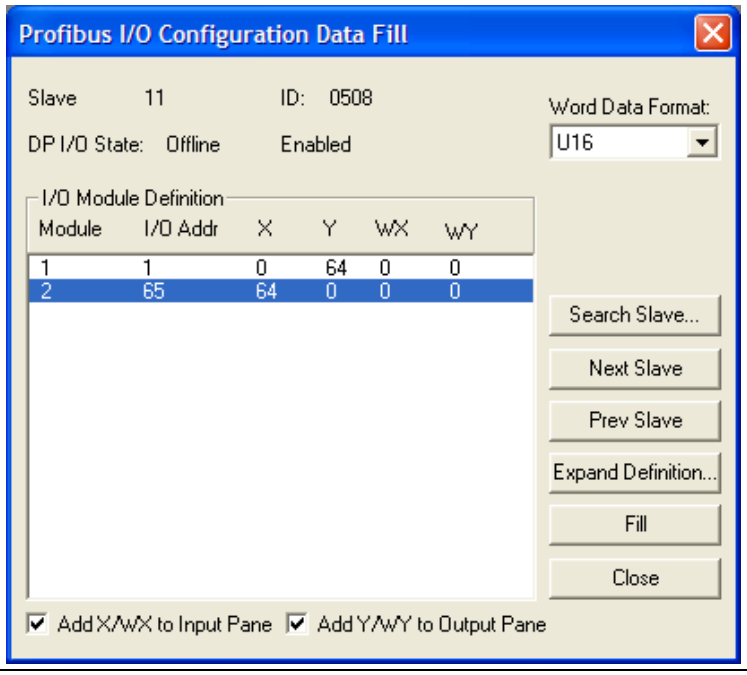
- If the **Insert** keyboard key is selected (OVR displays in the **Status Line** within WorkShop) existing addresses in the Data Window will be overwritten when new data is entered.
- If the **Insert** keyboard key is toggled off, existing addresses in the Data Window will be moved down to make room for the data that has been inserted.

I/O Simulator Window

Use the Profibus I/O 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.
2. Select the row in which data is to be filled.
3. Select the **I/O Simulator / Fill / Profibus I/O Fill** menu item. All configured slaves are read, and the configuration is updated. The **Profibus I/O Configuration Data Fill** dialog appears, containing a list of configured modules in the first configured slave detected and controls to navigate through additional slaves.



4. Select a configured module.
5. Click the **Fill** button. If the **Add X/WX to Input Pane** check box is selected, **X/WX** addresses are automatically inserted into the **Input** pane of the I/O Simulator Window. If the **Add Y/WY to Output Pane** checkbox is selected, **Y/WY** addresses are automatically inserted into the **Output** pane of the I/O Simulator window. When a **WX** or **WY** address is inserted, it will automatically change its display format to the format that is currently selected in the **Word Data Format** combo box of the **Profibus I/O Configuration Data Fill** dialog.

- If the **Insert** keyboard key is selected (OVR displays in the **Status Line** within WorkShop) existing addresses in the Data Window will be overwritten when new data is entered.
- If the **Insert** keyboard key is toggled off, existing addresses in the Data Window will be moved down to make room for the data that has been inserted.

SAVING PLC STATE

505 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 finishes. 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 505 Simulator state:

1. Select the **File / Save** menu item within the **505 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 505 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 **505 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.