

Quick Start Guide

FTSolution for APT

FTSolution for APT - Hardware Debugger

FTSolution for APT Simulator

FTVersionTrak for APT

FasTrak
SoftWorks, Inc.

www.fast-soft.com

262.238.8088

FTSolution for APT, FTSolution for APT Hardware Debugger, FTSolution for APT Simulator, FTVersionTrak for APT, and Foundations: FTSolution for APT are trademarks of FasTrak SoftWorks, Inc. Siemens APT and APT are registered trademarks of Siemens Industrial Automation, Inc. 2500 Series is a registered trademark of Control Technology, Inc. All other trademarks and trade names are property of their respective owners. All rights reserved.

Contents

Introduction 6

 Compilation Differences 7

Installing FTSolution for APT 9

Importing an Existing APT Program14

 Importing .1 Archive Files.....15

Using the Solution Explorer16

 Displaying and Editing Project Components16

 Displaying Multiple Project Components.....17

PLC Type Configuration18

Displaying and Editing Recipe Templates 20

Displaying and Editing Recipes21

Displaying and Editing Declarations 22

Displaying and Editing Continuous Function Charts 23

 Displaying and Editing Block Settings in Continuous Function Charts 24

 Displaying and Editing Math Statements in Continuous Function Charts 25

 Code Regions 25

 Flyby Windows Containing Statements Within a Collapsed Code Region..... 26

 New Instructions that Separate Blocks 26

 New Text Editor Features 27

 Syntax Coloring 27

 Hover Text..... 27

 Incremental Parsing 28

 Statement Completion..... 28

Displaying and Editing Sequential Function Charts..... 29

 SFC Magnification..... 30

 SFC Zoom Mode..... 30

 Displaying and Editing Math Statements in Sequential Function Charts31

 Code Regions 31

 Flyby Windows Containing Statements Within a Collapsed Code Region..... 32

 New Instructions that Separate Steps and Transitions 33

 New Text Editor Features 33

 Syntax Coloring 33

 Hover Text..... 34

 Incremental Parsing 34

 Statement Completion..... 34

Displaying and Editing Devices..... 35

Displaying and Editing Subroutines..... 36

 Displaying and Editing Math Statements in Subroutines..... 37

 Code Regions 37

 Flyby Windows Containing Statements Within a Collapsed Code Region..... 38

New Instructions in Subroutines.....	39
New Text Editor Features	39
Syntax Coloring	39
Hover Text.....	40
Incremental Parsing	40
Statement Completion.....	40
Compiling an FTSolution for APT Project	41
Compilation Report	43
Viewing the Objects Modified by Incremental Compile.....	44
Preconditions for Incremental Compile and Download.....	45
Preconditions for Incremental Download in PROGRAM mode.....	45
Preconditions for Incremental Download in RUN mode.....	46
Finding References to an Object	47
Communication Setup.....	48
Downloading a Compiled FTSolution for APT Project	51
Debug Mode	53
Warning About Debug Mode.....	53
Starting Debug Mode.....	53
Monitoring Variables in Debug Mode.....	55
Modifying I/O in a Watch Window	57
Forcing I/O.....	57
Clearing Forced I/O	58
Tracing SFC Program Flow.....	58
Logging.....	61
Adding an I/O Module for a 505 Series / CTI Controller	62
Defining I/O Points for 505 I/O	63
Profibus Configuration	64
Adding GSD files to the Hardware Directory	65
Assigning Slaves to the Profibus Configuration	66
Configuring and Addressing Profibus I/O Modules	67
Defining I/O Points in Profibus I/O	69
Merging a .2bf File.....	69
Viewing and Editing I/O Symbolic Names.....	70
FTSolution for APT Hardware Debugger.....	71
Creating an .AHD file in FTSolution for APT	71
Opening an .AHD file in FTSolution for APT	72
Opening an .AHD file in Hardware Debugger	72
FTSolution for APT Simulator (505 Simulator)	73
Setting Up 505 Simulator as an External Tool.....	73
Starting 505 Simulator	76
Selecting PLC Type	77
Setting Up Communications with 505 Simulator in FTSolution for APT.....	77

Connecting to 505 Simulator 78
 Performing an Incremental Download to 505 Simulator in Run Mode..... 79
FTVersionTrak for APT 87
 FTVersionTrak Toolbar..... 88
 Binding to a Repository 89
 Setting FTVersionTrak Options 90
 Using Solution Explorer with FTVersionTrak 90
 Example of Compare: Comparing at the Project Level 91
 Example of History: Historical Information for a Single File 91
PCS Tagging – Marking Objects for Translation 92
 Tag Translation and Compile Options..... 96
 Location of Install.tag and Append.tag Reports 96

Introduction

Thank you for purchasing FTSolution for APT® by FasTrak SoftWorks, Inc. FTSolution for APT is part of a family of products and services provided by FasTrak SoftWorks, Inc. to meet the needs of APT language programmers. This family includes:

- **FTSolution for APT**
Windows-based APT language programming software. FTSolution for APT allows you to create and maintain APT-based PLC programs with a modern Microsoft Windows interface.
- **FTSolution for APT Hardware Debugger**
Sophisticated PLC hardware debugger utilizing ladder language-based tools. FTSolution for APT - Hardware Debugger allows users to download compiled FTSolution for APT programs into Siemens 505/CTI 2500 Series processors, utilize sophisticated ladder language-based debugging tools, and limit the ability to modify existing programs.
- **FTSolution for APT Simulator**
PLC Simulator for most APT-supporting Siemens 505 and CTI 2500 Series processors. FTSolution for APT Simulator saves time and hardware expense by running and testing APT projects without the use of a PLC.
- **FTVersionTrak for APT**
Version control for FTSolution for APT. FTVersionTrak for APT safeguards your FTSolution project files with version management, detailed program comparison, and automatic backup.

Contact your FasTrak SoftWorks, Inc. Sales Representative at **262-238-8088** or sales@fast-soft.com for assistance in purchasing or upgrading these products. For assistance installing and using FTSolution for APT contact Technical Support at **262.238.8088, Option 3** or techsupport@fast-soft.com.

Compilation Differences

The following list describes intended differences related to compilation that exist between APT and FTSolution for APT.

- FTSolution for APT's advanced compiler

The new compiler tests programs more thoroughly than the original APT compiler, and may detect previously unidentified errors. Depending upon their severity, you may ignore these errors or edit your programs to eliminate them.

- Programs that contain specific user-defined I/O modules

In APT, programs can contain user-defined I/O modules with a mix of analog input (WX) and digital output (Y) that total 8. This consists of user-defined I/O modules configured in the following ways:

- WX7 and Y1
- WX6 and Y2
- WX 5 and Y3
- WX4 and Y4
- WX3 and Y5
- WX2 and Y6
- WX1 and Y7

APT compiles and downloads programs with I/O modules configured this way without a warning or an error message. FTSolution for APT will allow import of this condition, but will flag an error on compilation.

- SCL Verbs

In APT, SCL verbs can be used as identifiers. (For example, the statement: **AUTO AUTO** is allowed in APT). FTSolution for APT does not allow SCL verbs to be used as identifiers.

- Literals of real numbers

In APT, literal real numbers having the form of 1E6 are allowed. FTSolution for APT rejects literal real numbers in the form of 1E6 . A decimal point (".") before the E (1.E6), which distinguishes the real literal from an identifier, is required in FTSolution for APT.

- ASM Semantic Checking

No ASM semantic checking is implemented in FTSolution for APT. For this reason, only use already semantically-valid ASM when compiling in FTSolution for APT.

- User-assigned Y address in a Boolean declaration

APT cannot compile a program when a user-assigned Y address is used in a Boolean declaration. In FTSolution for APT, Y addresses in a Boolean declaration are allowed.

- PLC Type Checking on Download

In FTSolution for APT, an error will occur during download if the PLC type set in the program's configuration file does not match the actual PLC to which FTSolution for APT is connected (see *PLC Type Configuration* on page 18 for additional information.)

- PLC Type Checking on Compile

In FTSolution for APT, if an imported APT file has configured memory beyond that of a 555, FTSolution for APT will generate an error on compile. This typically occurs when using a CTI 2500 to emulate a Simatic 555. To prevent an error on compile, a CTI 2500 PLC type must be specified in the program's configuration file (see *PLC Type Configuration* on page 18 for additional information.)

Note: There is no internationalization in FTSolution for APT. FTSolution for APT supports only United States English Regional and Language options. For example, floating-point numbers must follow US conventions: 1.5 must be used instead of 1,5. String literals must also follow US conventions: double quotes must be used to indicate a string (" ") instead of double brackets (<< >>). You may want to change your Regional and Language Options in Windows to reflect United States English while using FTSolution for APT.

Installing FTSolution for APT

Previous versions of FTSolution for APT must be uninstalled before v1.x.x can be installed. Uninstalling previous versions of FTSolution for APT removes the application only. Any APT programs that you imported, edited, or wrote will not be deleted. **When using v1.x.x, please re-import and compile any APT programs you have previously used with FTSolution for APT. Version 1.x.x requires re-importing in order to experience full functionality.**

Note for Windows XP users: FTSolution for APT requires that Windows XP Service Pack 3 is installed first.

To uninstall the previous version of FTSolution for APT:

Windows 8

1. Go to the **Start** menu by pressing the Windows key.
2. Right-click **FTSolution for APT**.
3. Click **Uninstall**.
4. In the list of installed applications, click **FTSolution for APT**.
5. Click **Uninstall**.
6. Follow the prompts. A progress bar shows you how long it will take to safely remove FTSolution for APT. When it is finished, you will see the list again.

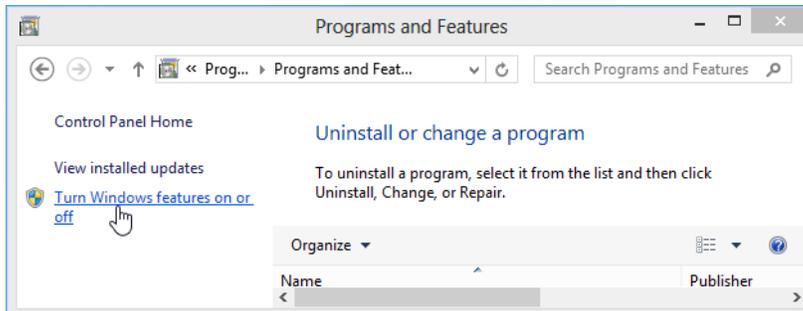
Windows XP, Windows Vista, and Windows 7

1. On the **Start** menu, click **Control Panel** and then under **Programs** do one of the following:
 - a. Windows 7 and Vista users: Click **Uninstall a Program**.
 - b. Windows XP users: Click **Add or Remove Programs**.
2. On the list of installed programs, click **FTSolution for APT**, then do one of the following:
 - a. Windows 7 and Vista users: Click **Uninstall**.
 - b. Windows XP users: Click the **Remove** or **Change/Remove** tab (to the right of the program).
3. Follow the prompts. A progress bar shows you how long it will take to safely remove FTSolution for APT. When it is finished, you will see the list again.

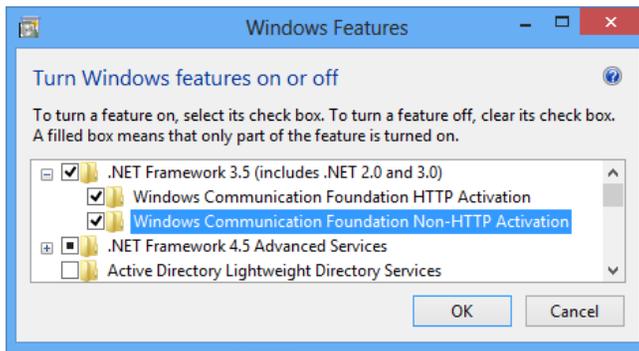
Download Windows 8 Prerequisites:

For Windows 8 users only: Windows 8 users must download additional components before downloading and installing FTSolution for APT with FTVersionTrak.

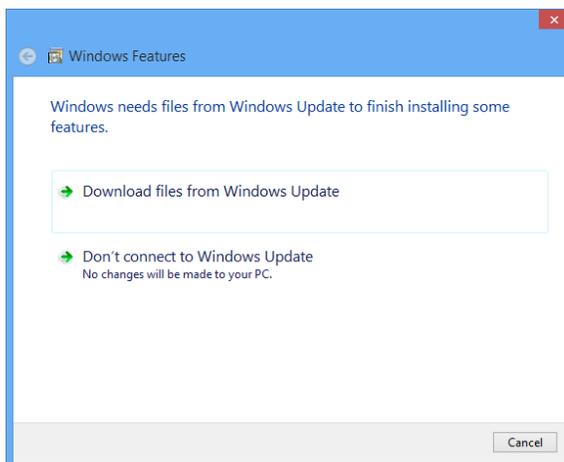
1. Go to Control Panel.
2. Click **Program and Features**. (If Control Panel is set to Category View, click **Programs** first, and then click **Program and Features**).
3. In the left panel, click **Turn Windows Features On and Off**.



4. In the Windows Features dialog box, expand **.NET Framework 3.5**.
5. Select the **Windows Communication Foundation HTTP Activation** and the **Windows Communication Foundation Non-HTTP Activation** check boxes.



6. Click **OK**.
7. Click **Download files from Windows Update**.



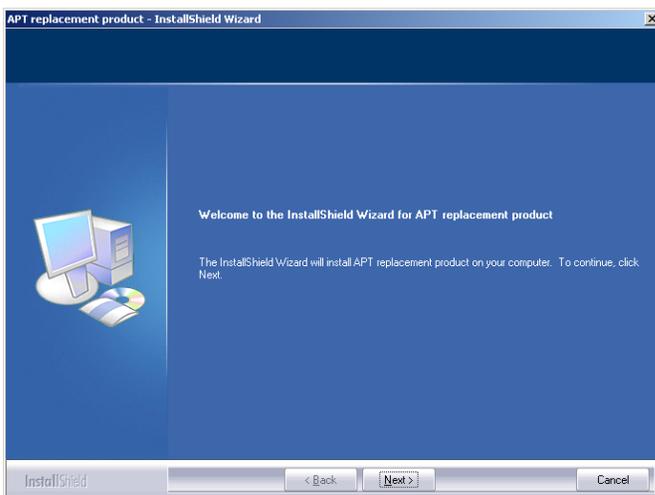
When these files have downloaded, proceed with the following instructions.

Download and Install FTSolution for APT and FTVersionTrak:

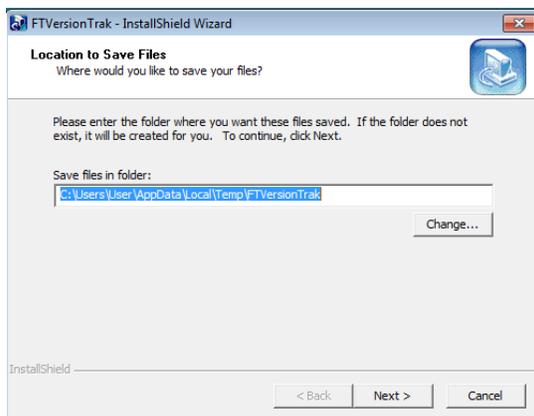
1. Access the FasTrak website: **www.fast-soft.com**.
2. Login (or register) to the FasTrak website.
3. Select **Software Upgrades** from the **Downloads** menu.
4. Click the link to **FTSolution for APT** or **FTSolution for APT Full Version** to download the self-extracting installer to your hard drive. *NOTE:* The full version includes all of the prerequisites (such as .NET framework) necessary to install the product. The regular version does not contain

any prerequisites; if needed they are downloaded on demand during the installation. Either one can be used to install FTSolution for APT. The full version is a larger file and can take longer to download. If you have already installed a previous version of FTSolution for APT, the regular version is recommended.

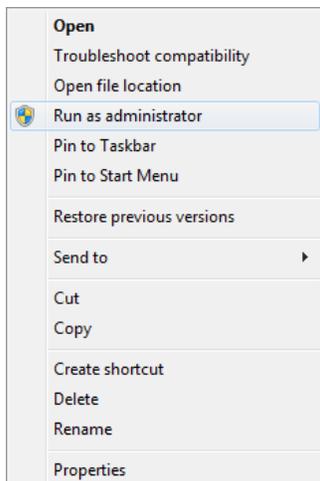
5. If you have purchased FTVersionTrak for APT and it was not previously installed, click the link to **FTVersionTrak** to download the self-extracting installer to your hard drive.
6. Run the FTSolution for APT installer after it downloads to your hard drive.
7. When the installer window appears, follow the prompts to complete the installation of FTSolution for APT.



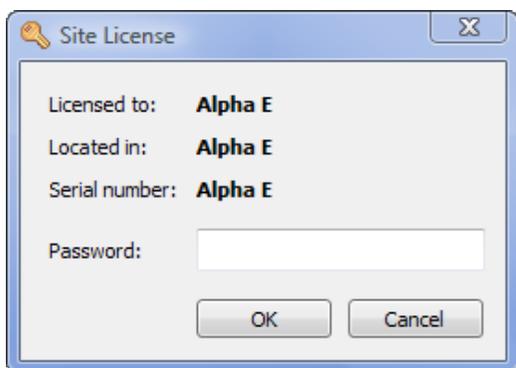
8. When the installation of FTSolution for APT is complete, run the FTVersionTrak installer.
9. When the installer window appears, follow the prompts to complete the installation of FTVersionTrak.



10. When FTVersionTrak has completed installing, start FTSolution for APT using the **Run As Administrator** option:
 - a. Right-click the APT replacement desktop icon.
 - b. Select **Run as Administrator** from the shortcut menu.



- When FTSolution for APT starts, make sure your key is attached or enter your site license password in the **Site License** dialog box. This password is case-sensitive. Please contact FasTrak SoftWorks, Inc. if you have questions about your site license.

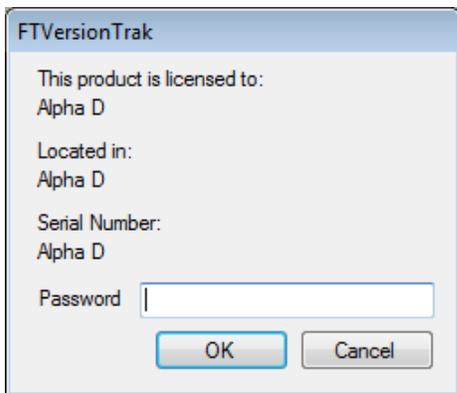


- If applicable, Start FTVersionTrak from within FTSolution for APT by clicking the **FTVersionTrak** toolbar button on the FTVersionTrak toolbar.

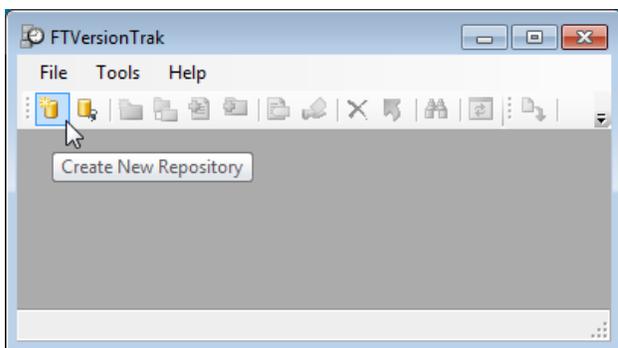


(If the FTVersionTrak toolbar is not displayed in FTSolution for APT, click the **View** menu, point to **Toolbars**, and select **FTVersionTrak Toolbar**).

- When FTVersionTrak starts, make sure your key is attached or enter your site license password in the **Site License** dialog box. This password is case-sensitive. Please contact FasTrak SoftWorks, Inc. if you have questions about your site license.



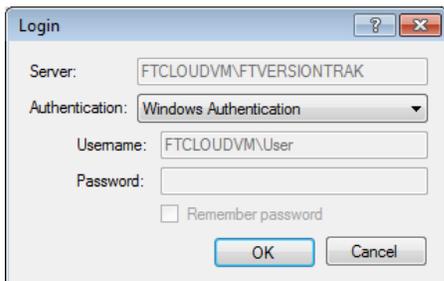
- In FTVersionTrak, click the **Create New Repository** toolbar button.



- In the **Create New Repository** dialog box, in the **Repository Name** box, type a name for the repository.



- If a Login dialog box is displayed, click **OK**.



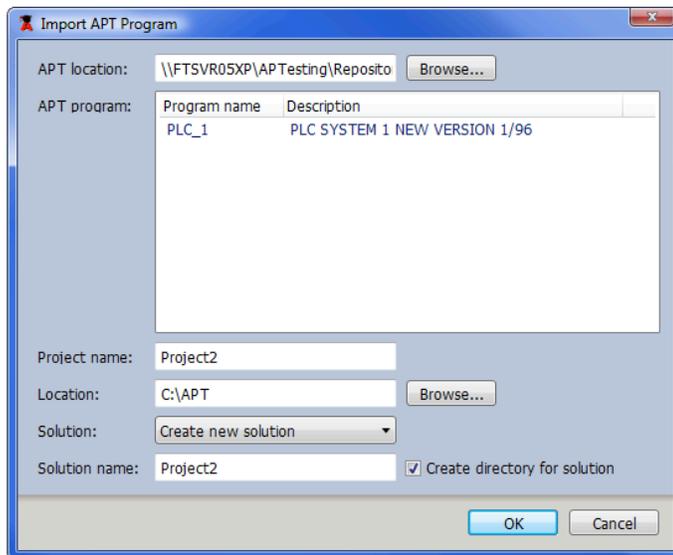
You can now use FTVersionTrak and FTSolution for APT.

Importing an Existing APT Program

When using v.1.x.x, re-import and compile any APT programs you have previously used with FTSolution for APT. Re-importing is required in order to experience full functionality.

Use the following procedure to import APT programs from an APT repository.

1. Click **Import APT Program** on the **File** menu. The Import APT Program dialog box appears.



2. Do one of the following:
 - In the **APT location** box, type the location of the APT repository that contains the APT program you want to import.
 - Click **Browse** to browse to the APT repository that contains the APT program you want to import.

A list of APT programs appears in the **APT program** box after a valid repository is selected.

3. In the **APT program** box, click the APT program you want to import. The **Project name** and **Solution name** boxes are automatically configured to match the name of the APT program you are importing.
4. Click **OK**.

Importing .1 Archive Files

FTSolution for APT must import programs from an APT database. If your programs are stored as .1 archive files, they must first be restored to an APT database, and then imported from the database.

Use the following procedure to restore your .1 archived files to an APT database:

1. Determine the location of your .1 archive file. You will need this path in Step 4.
2. Launch APT.
3. From the main program menu in APT, press **8** or select **Restore** from the F3 context menu.
4. Set the Path to the location of the .1 archive, determined in Step 1.
5. Enter the name of the archive file that you wish to restore, and press **ENTER**. If a message indicates that the program already exists in your database, you do not need to restore the program to import it into FTSolution for APT.
6. If APT and FTSolution for APT are installed on different computers, copy APT's PROGRAM and DATABASE folders to the APT replacement computer. FTSolution for APT imports programs from these folders.

Using the Solution Explorer

FTSolution for APT stores your PLC programs as Projects. These Projects are stored within Solutions. Each Solution may contain one Project. But Solutions can hold multiple Projects. This allows related PLC programs to be grouped within the same Solution.

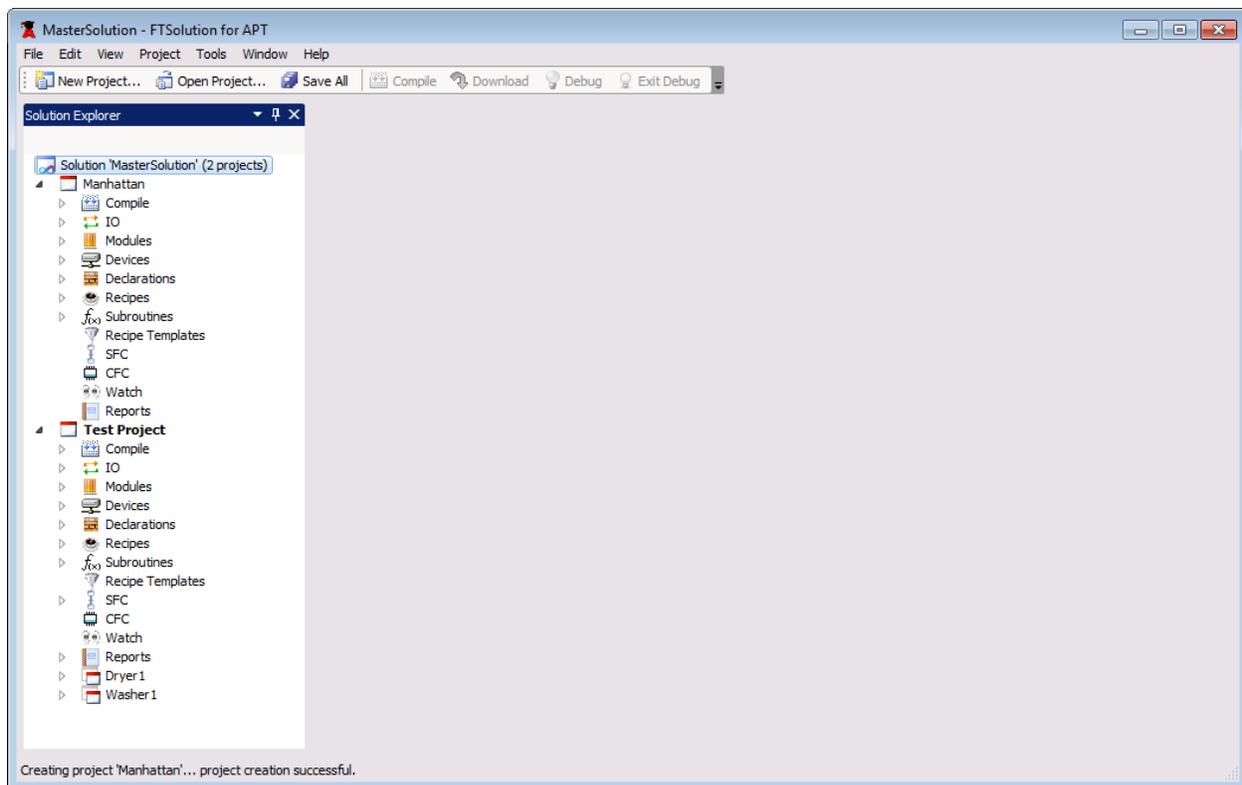
You access the Projects in a Solution with the Solution Explorer. This Windows tree-control provides direct access to every component of your PLC program Project.

You can expand and contract the nodes in the Solution Explorer to display and hide lower nodes.

Open Solutions and Programs through items in the **File** menu, or the **Open Project** button.

Selecting Projects from the Solution Explorer is similar to choosing PLC programs from the APT Program Directory Level screen. But the Solution Explorer also provides access to all PLC program components, similar to the APT Program Content Level screen.

In the following illustration, the Solution Explorer appears along the right edge. Solution **MasterSolution** contains two Projects, **Manhattan** and **Test Project**.



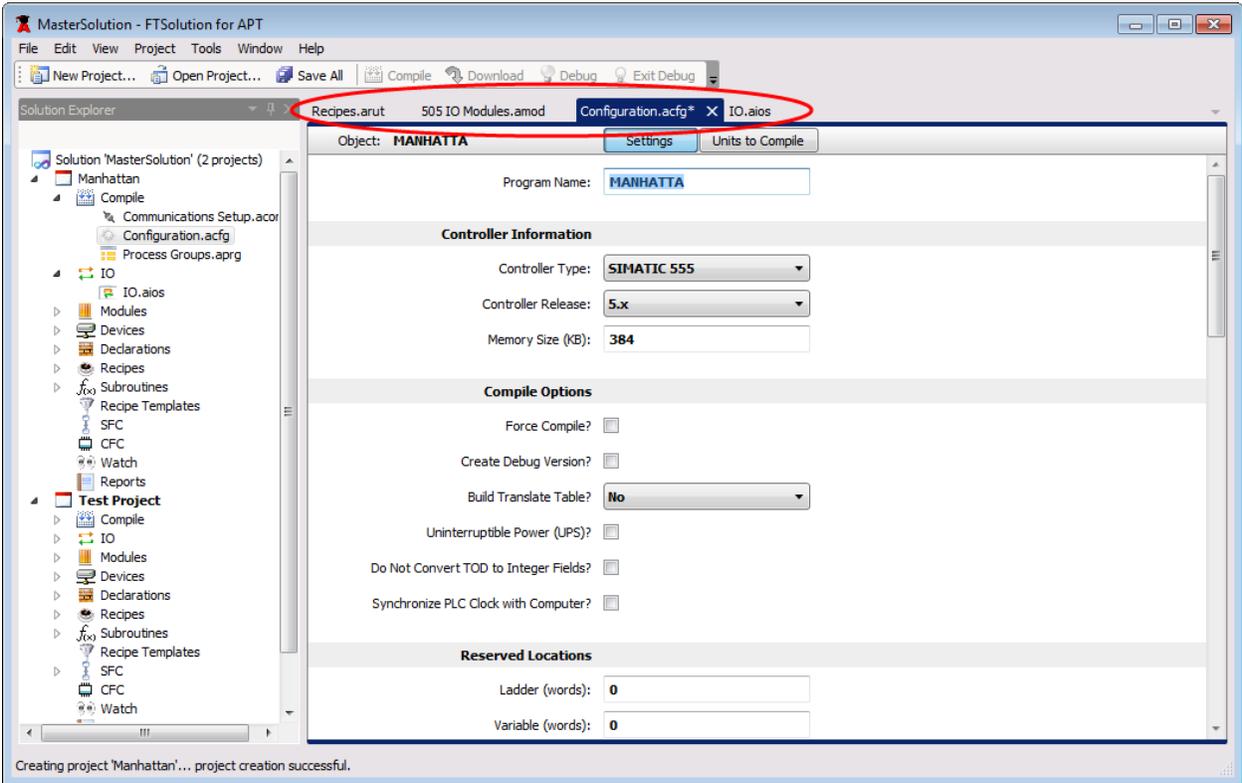
Displaying and Editing Project Components

Any component within a Project can be displayed in its own editor window.

To edit a Project component, use the up and down arrow keys to highlight the item in the Solution Explorer, and press **Enter**. Or double-click an item to select it.

Displaying Multiple Project Components

Multiple Project components can be opened at the same time, and displayed in their own editor windows. In the following illustration, the editor windows of four components are open. The red circle near the upper left corner indicates the four tabs that provide access to the four editor windows. Click on a tab, or press **Ctrl-Tab** repeatedly, to bring the desired editor window to the foreground.

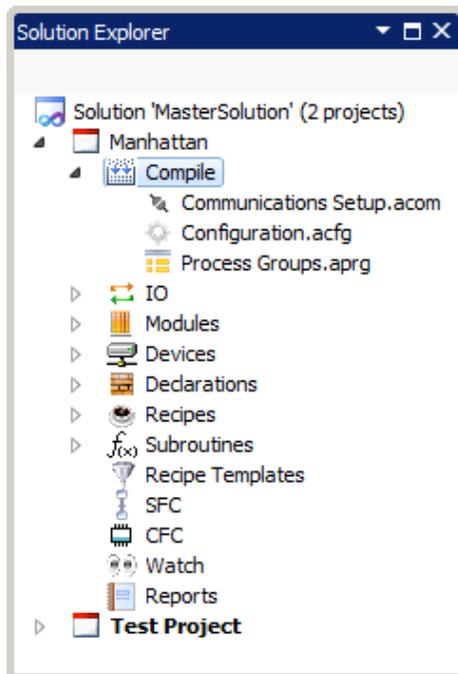


PLC Type Configuration

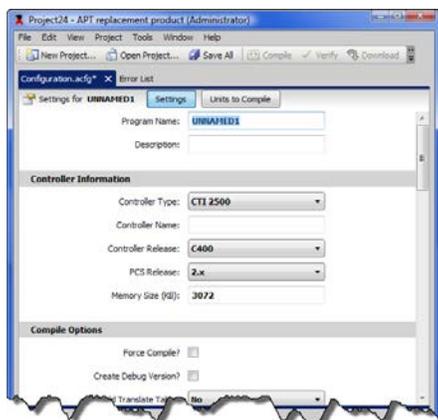
PLC type is one of the settings specified in the compile configuration file (**Configuration.acfg**). During an import, the PLC type is automatically configured. For a successful download, the configured PLC Type **must** match the target controller type.

Use the following procedure to specify the PLC type.

1. View an FTSolution for APT project in the Solution Explorer.
2. Double-click **Compile** in the Solution Explorer. You can also double-click **Configuration.acfg**.



3. In the **Controller Type** box, select the type of controller you are using.
4. In the **Controller Release** box, select the release of the controller.
5. Enter other settings as required.



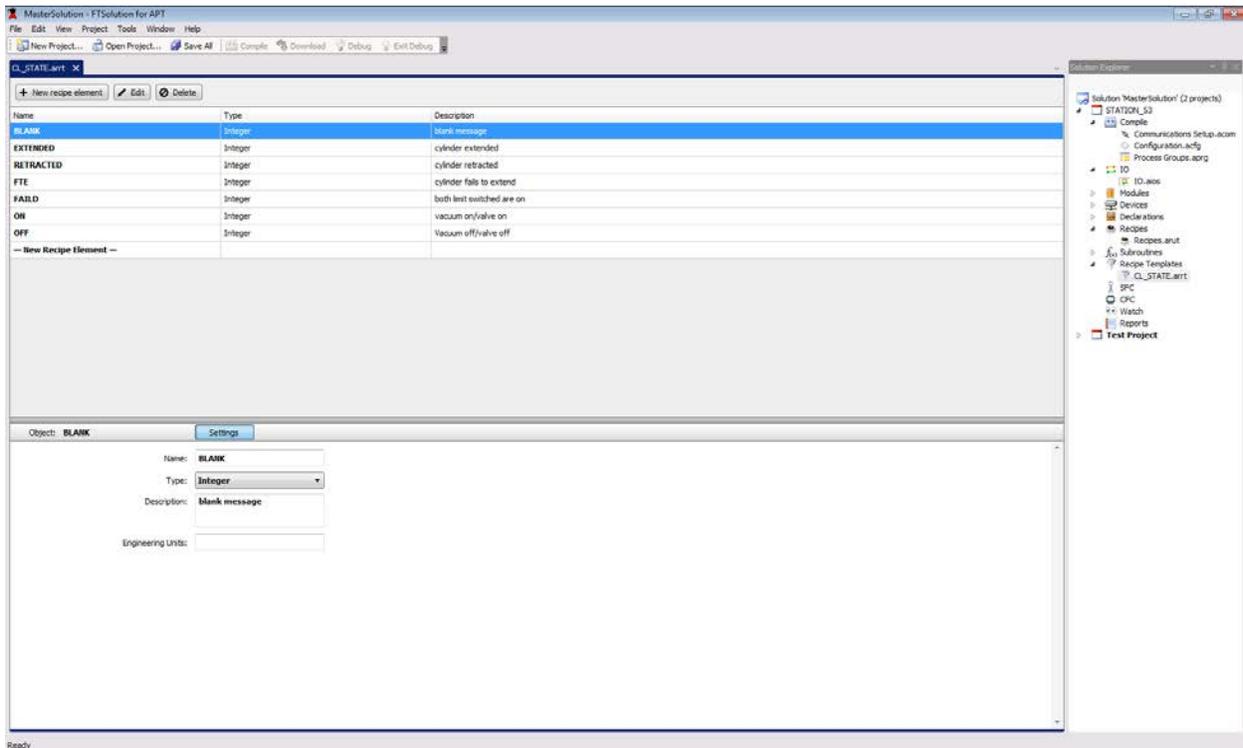
6. Save the **Configuration.acfg** file by clicking **Save Configuration.acfg** on the **File** menu.

Note: The original APT did not directly support the CTI 2500 family of PLCs. In order for the CTI 2500 Series® to work with the original APT, the CTI 2500 PLCs had to emulate the Simatic 505 series. This emulation required the use of a software patch for APT and a DIP switch setting on the CTI 2500 Series PLCs.

As FTSolution for APT directly supports the CTI 2500 Series, FTSolution for APT ignores the setting of the DIP switch. However, in FTSolution for APT, **you must specify a PLC type of CTI 2500 if you are connecting to a CTI 2500 processor.** A mismatched PLC type will cause an error and prevent the program from downloading.

Displaying and Editing Recipe Templates

In the following illustration, **Recipe Templates** in the **STATION_53** Project is selected. Selecting **Recipe Templates** from the Solution Explorer along the right edge of the window opens the Recipe Templates editor on the left side.



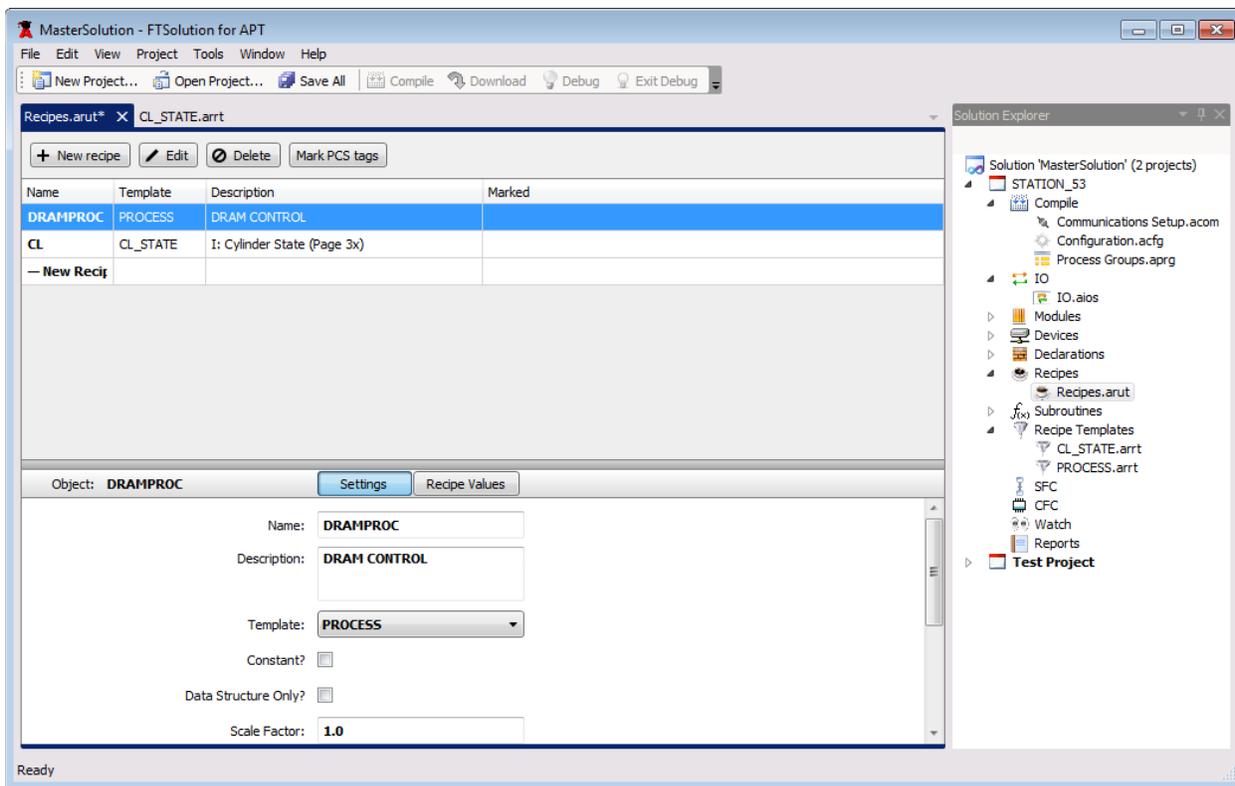
The upper portion of this editor lists all the **Recipe Templates** of the **STATION_53** Project. The lower portion of this editor displays the **Settings** of the Recipe Template that is highlighted in the upper portion.

Highlight a Recipe Template in the upper portion, to display its Settings in the lower portion.

Settings are validated as they are edited.

Displaying and Editing Recipes

In the following illustration, **Recipes** in the **STATION_53** Project is selected. Selecting **Recipes** from the Solution Explorer along the right edge of the window opens the Recipes editor on the left side.



The upper portion of this editor lists all the **Recipes** of the **STATION_53** Project. The lower portion of this editor displays the **Settings** or **Recipe Values** of the Recipe that is highlighted in the upper portion.

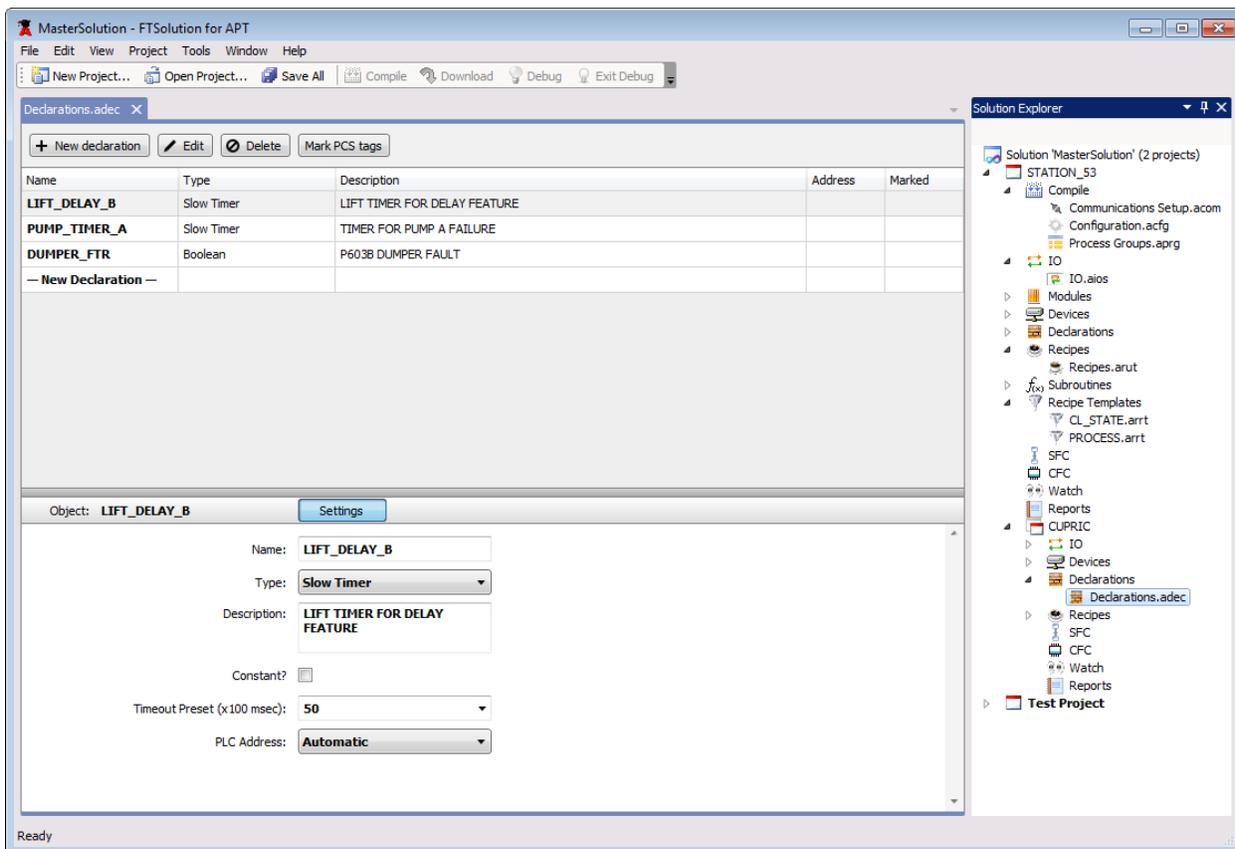
Press **F10**, or click the **Settings** and **Recipe Values** buttons, to toggle between the Settings and Recipe Values editors.

Highlight a Recipe in the upper portion, to display its Settings or Recipe Values in the lower portion.

Settings and Recipe Values are validated as they are edited.

Displaying and Editing Declarations

In the following illustration, **Declarations** of the **CUPRIC** item in the **POWERLINE** Project is selected. Selecting **Declarations** from the Solution Explorer along the right side of the window opens the Declarations editor on the left side.



The upper portion of this editor lists all the **Declarations** of the **CUPRIC** item. The lower portion of this editor displays the **Settings** of the Declaration that is highlighted in the upper portion.

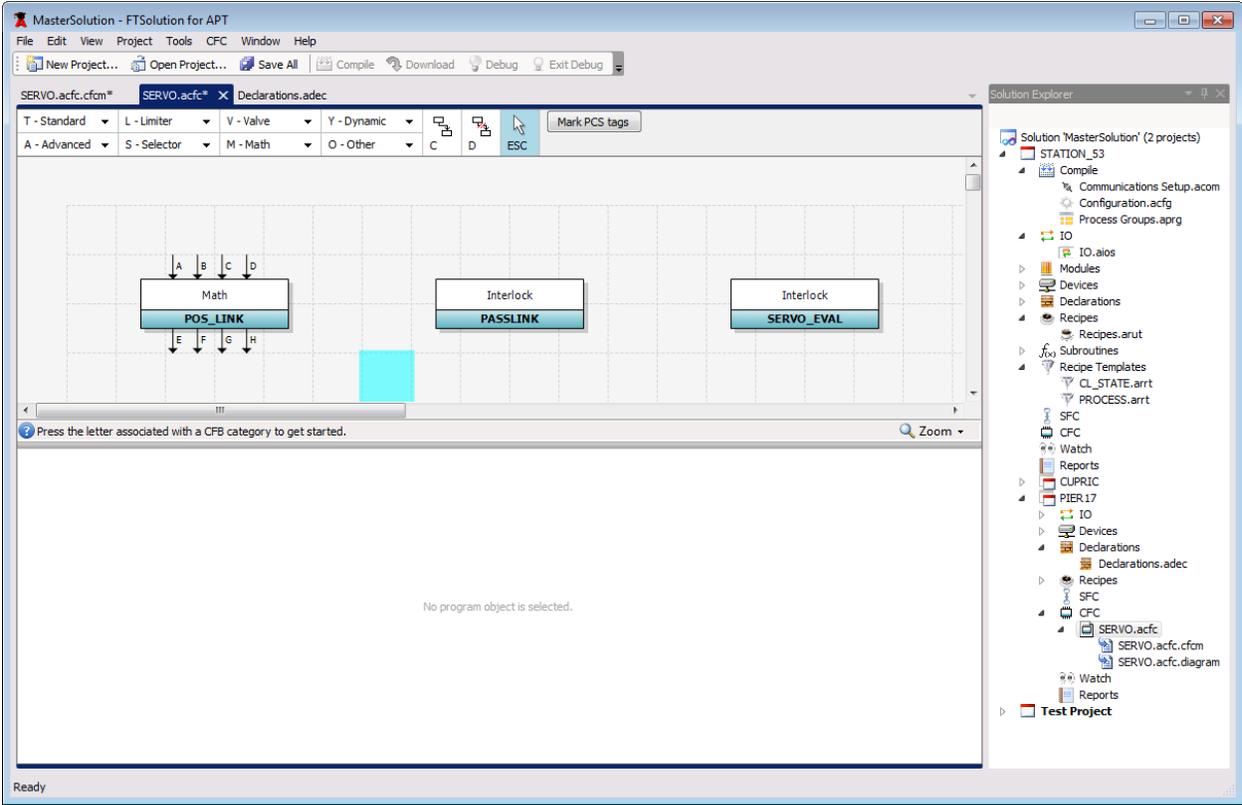
Highlight a Declaration in the upper portion to display its Settings in the lower portion.

Settings are validated as they are edited.

Displaying and Editing Continuous Function Charts

To display and edit a Continuous Function Chart, use the up and down arrow keys in the Solution Explorer to highlight a CFC, and press **Enter**. Or double click a CFC to select it.

In the following illustration, the **SERVO** CFC of the **PIER17** item was selected in the Solution Explorer. The CFC diagram appears in the upper portion of the edit window. This editor supports mouse operations, and the same keystrokes available in APT.

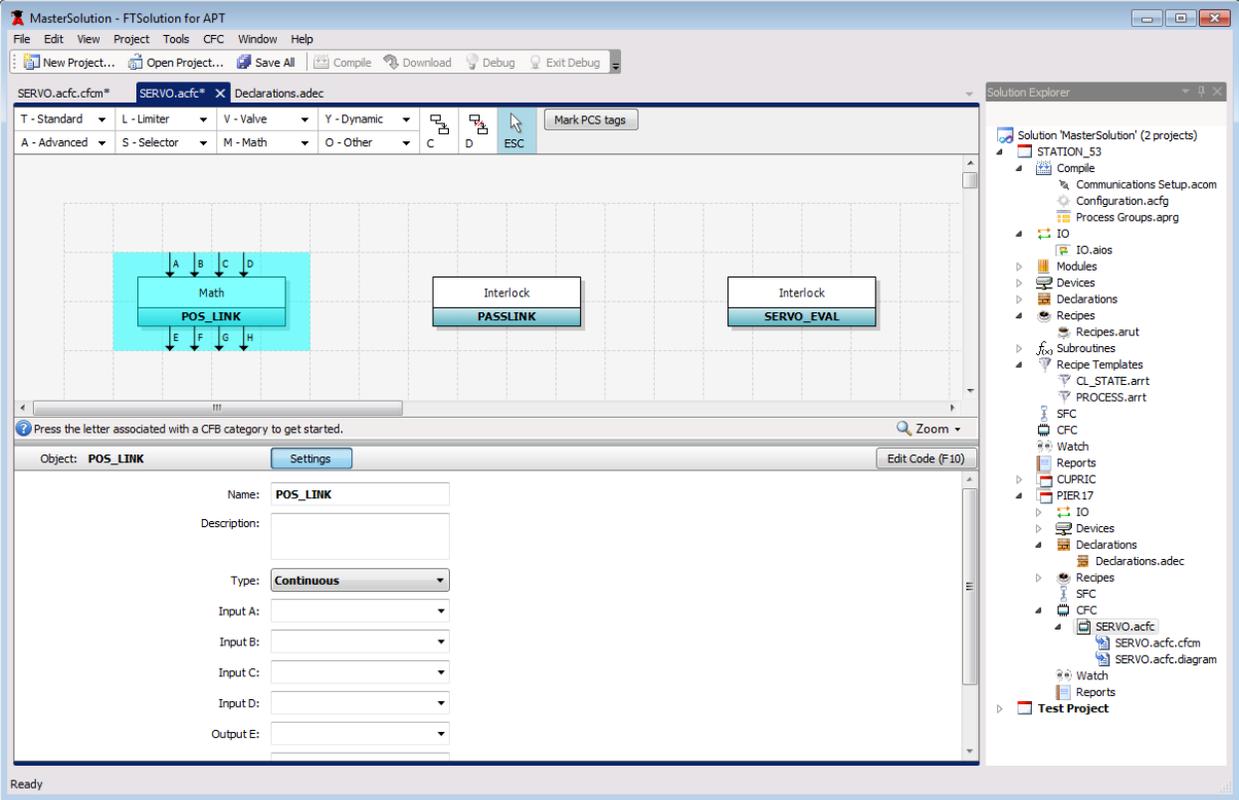


Displaying and Editing Block Settings in Continuous Function Charts

Use the arrow keys and press **Enter**, or mouse pointer to select a CFC block in the upper portion to edit its **Settings** in the lower portion.

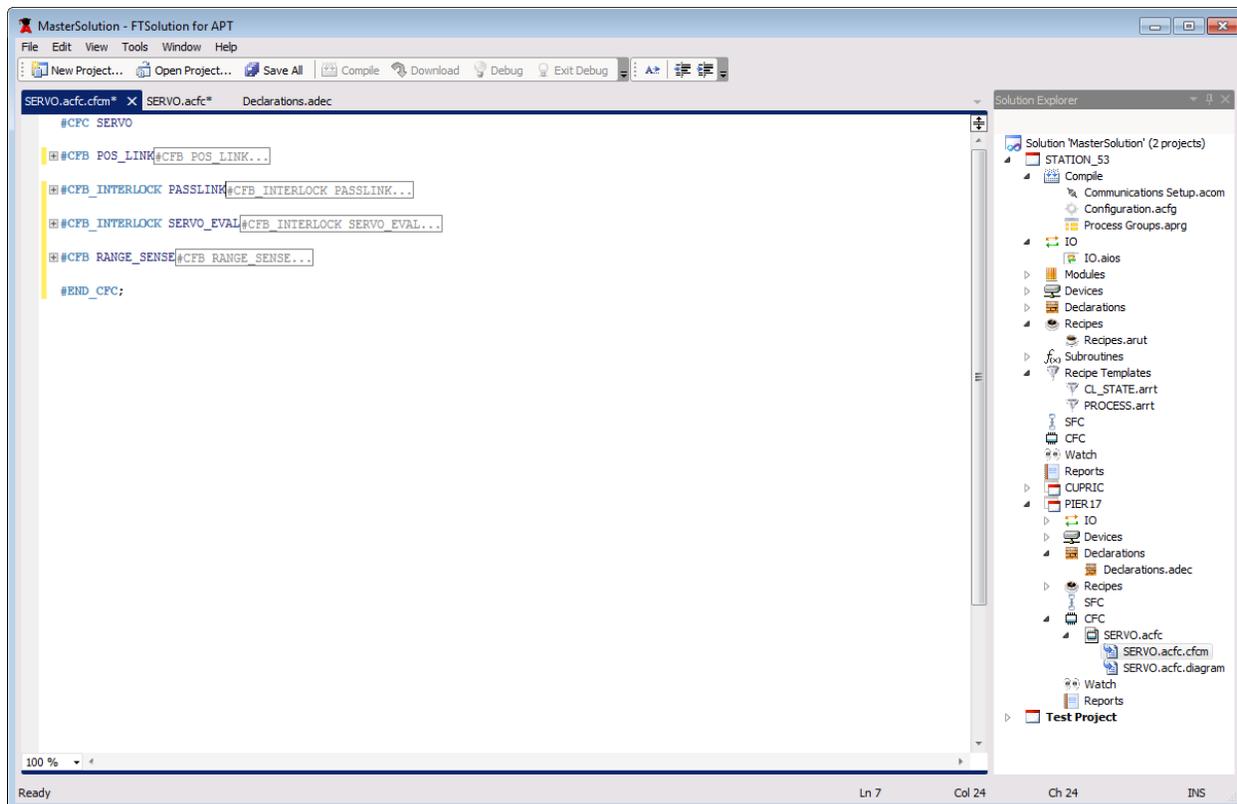
In the following illustration, the **POS_LINK** Math block is selected, and its Settings appear.

Settings are validated as they are entered.



Displaying and Editing Math Statements in Continuous Function Charts

Press **F10** or click the **Edit Code (F10)** button in the Settings portion of the editor above to open the CFC Math Language Text Editor, as illustrated.



Code Regions

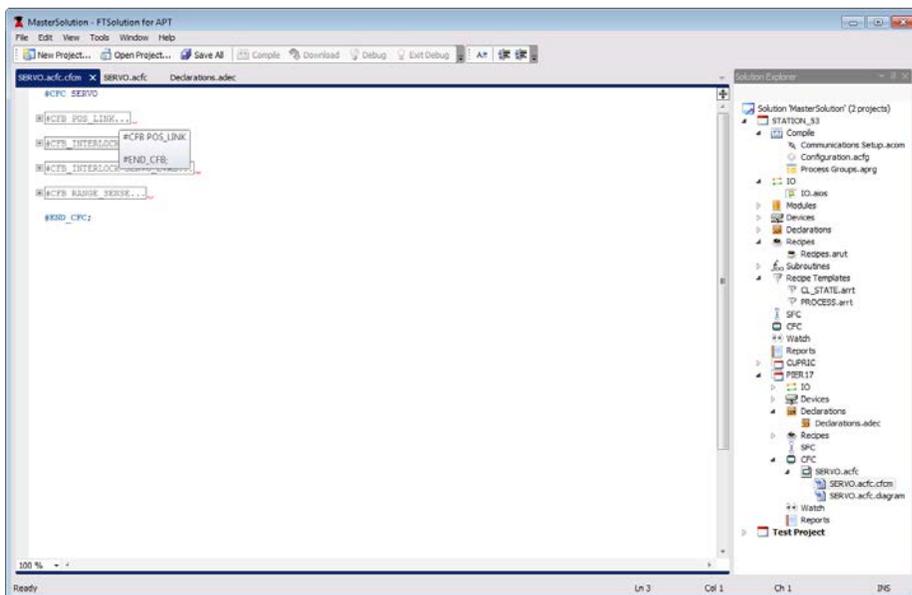
The text editor in APT displays the math statements for the selected CFC block only. However, the Math Statement Text Editor in FTSolution for APT provides access to the math statements for every block in the CFC.

This editor uses Code Regions to expand and collapse the statements for each block.

Click a + plus sign to expand the statements for that block, and click the – minus sign to collapse the statements. Or locate the cursor within a code region and press **Ctrl+MM** to toggle its expansion and collapse.

Flyby Windows Containing Statements Within a Collapsed Code Region

You can view the statements within a collapsed code region by hovering the mouse pointer over it. As illustrated, a flyby window containing the statements in the collapsed region appears. These flyby windows allow you to examine the statements in a collapsed region without expanding it.



New Instructions that Separate Blocks

The text editor in APT displays the math statements for the selected CFC block only. However, the Math Statement Text Editor in FTSolution for APT provides access to the math statements for every block in the CFC.

Because the math statements of all blocks can be accessed through the same CFC Math Language Text Editor, each block must be separated from the other blocks.

To separate math statement blocks, each block must start with the **#CFB *block_name*** instruction and end with the **#END_CFB** instruction, as illustrated.

These new instructions are required to separate the math statements of all blocks from each other.

```
#CFB 05_MAINT
{
  Date   : 12/31/2011
  Author : T Thomas
  Description: Operate station 5 independently.
}
PRAGMA("RLL");
BEGIN
IF (device_num = 295) AND switch_start_pb THEN
  295sv.MOPEN := TRUE;
ENDIF;
IF (device_num = 295) AND switch_stop_pb THEN
  295sv.MOPEN := FALSE;
ENDIF;
#END_CFB;
```

New Text Editor Features

The CFC Math Language Text Editor provides features that increase the speed and accuracy of writing programs.

Syntax Coloring

The text editor uses **Syntax Coloring**, in which keywords, variables, operators, and comments are displayed with different colors, as illustrated. This visually distinct method of displaying statements improves program readability.

```

{***** Orientation by 12 step operation / target input ***** }
{----- Start positioner 4 -----}
IF (sel_pos_no = 4) AND
    EDGE(ps_begin_tmr.ENABL) AND { Start button }
    NOT(x_s04_motion) { Stalled }
THEN
    position_cod[4] := destination_code; { Input Target }
    s04_begin := TRUE;
ENDIF;

```

Hover Text

The text editor provides **Hover Text**, in descriptive tooltips that appear when the mouse pointer hovers over keywords and variables. These tooltips contain information regarding the item over which the mouse pointer hovers. In the following illustration, the tooltip appeared when the mouse hovered over the variable `x_s04_ind_dn`.

```

{ Refine index complete
s04_comp := x_s04_ind_dn AND NOT(s04_start)
           A Math language identifier NOT(ps04_f
IF x_s04_motion OR ( s04_ftr AND alarm_reset

```

Incremental Parsing

APT identifies errors only after exiting the text editor and compiling. However, FTSolution for APT editor identifies errors immediately with **Incremental Parsing**, in which each line is validated as it is typed. Errors are underlined in red and described in the **Error List** window. Select the **View \ Error List** menu item to display the Error List window. In the following illustration, the red underline in the second TIMER SLOW line identifies the “:=” syntax error.

```
PRAGMA("RLL");
TIMER SLOW : jg1ac1at_timer := 15,15,FALSE,FALSE;
TIMER SLOW : jg1ac2at_timer ::= 15,15,FALSE,FALSE;
BEGIN
```

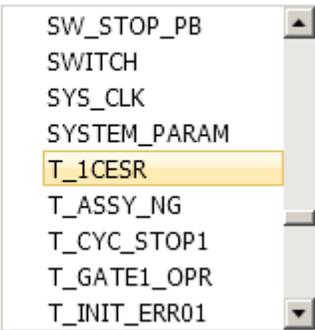
Statement Completion

The text editor utilizes **Statement Completion**, in which lists of keywords and variables are offered, based upon the characters entered. In the following illustration, the list of keywords and variables appeared, when “t_” was typed into the line. The **T_1CESR** item is highlighted in the list because it is the closest match to what was typed. Continue typing the remainder of the keyword or instruction or select an item from the list to complete the entry.

```
resetA_timer.ENABL := t_ ;
resetA_timer.RESET :=
resetC_timer.ENABL :=
resetB_timer.RESET :=

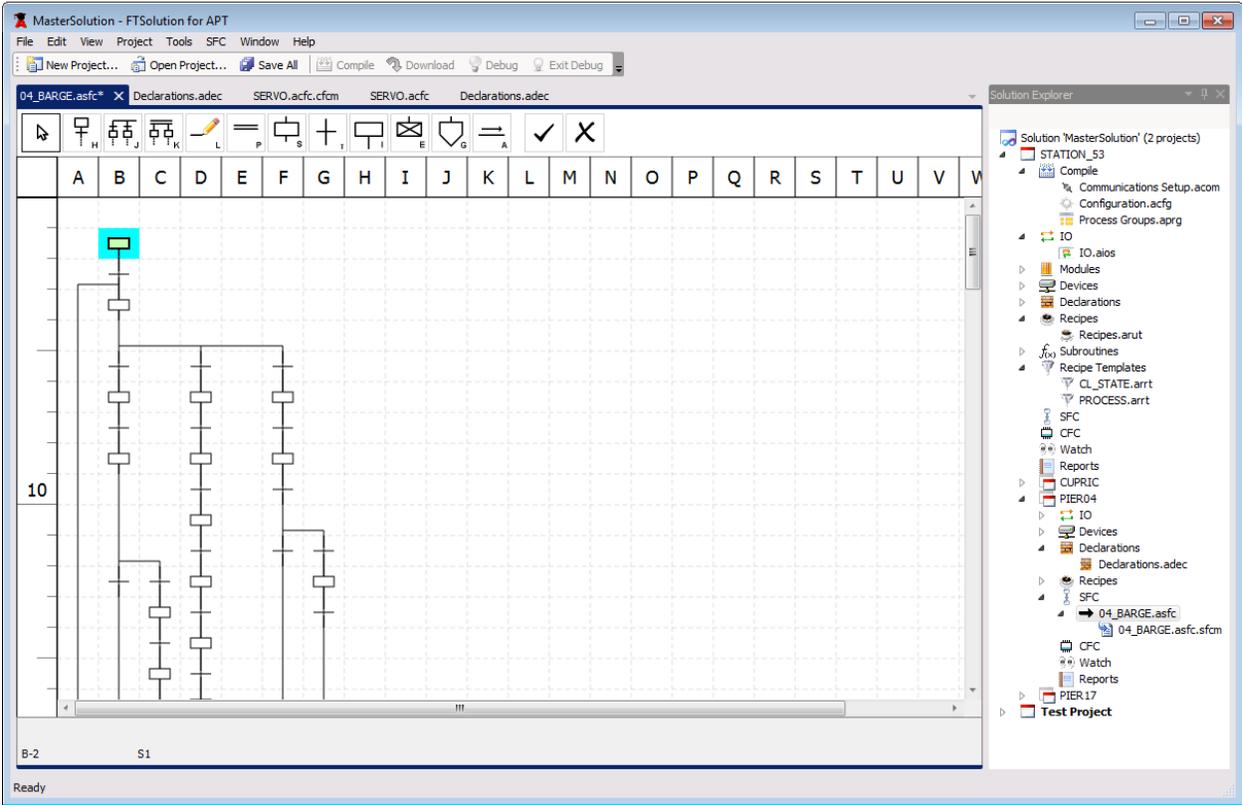
s04_reset := alarm_res
y_s04_reset := s04_res

{===== Check cycle ===== }
```



Displaying and Editing Sequential Function Charts

To display and edit a Sequential Function Chart, use the up and down arrow keys in the Solution Explorer to highlight the desired SFC, and press **Enter**. Or double-click an SFC to select it. In the illustration, the **04_BARGE** SFC of the **PIERO4** item was selected in the Solution Explorer.



The SFC editor supports the same keystrokes that are available in APT. It also supports mouse operations. For example, attach an instruction to the mouse pointer by clicking the instruction toolbar icon at the top of the editor. Then click and drag the mouse pointer in the SFC grid to add that instruction.

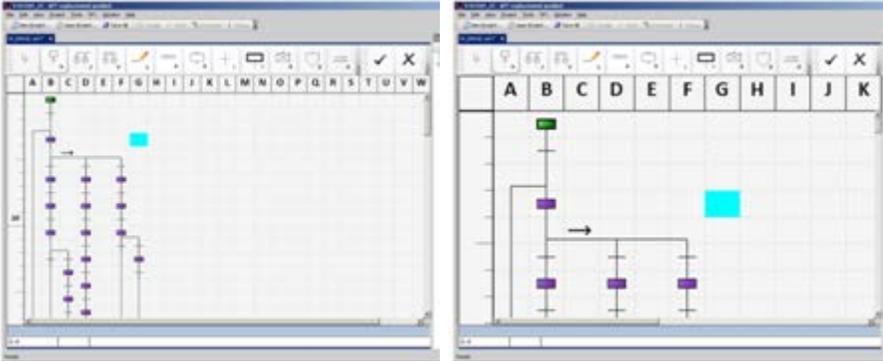
The **Pointer** toolbar icon  detaches any instruction from the mouse pointer.

The **Check Mark** toolbar icon  is equivalent to pressing **Enter**.

The **X** toolbar icon  is equivalent to pressing **Escape**.

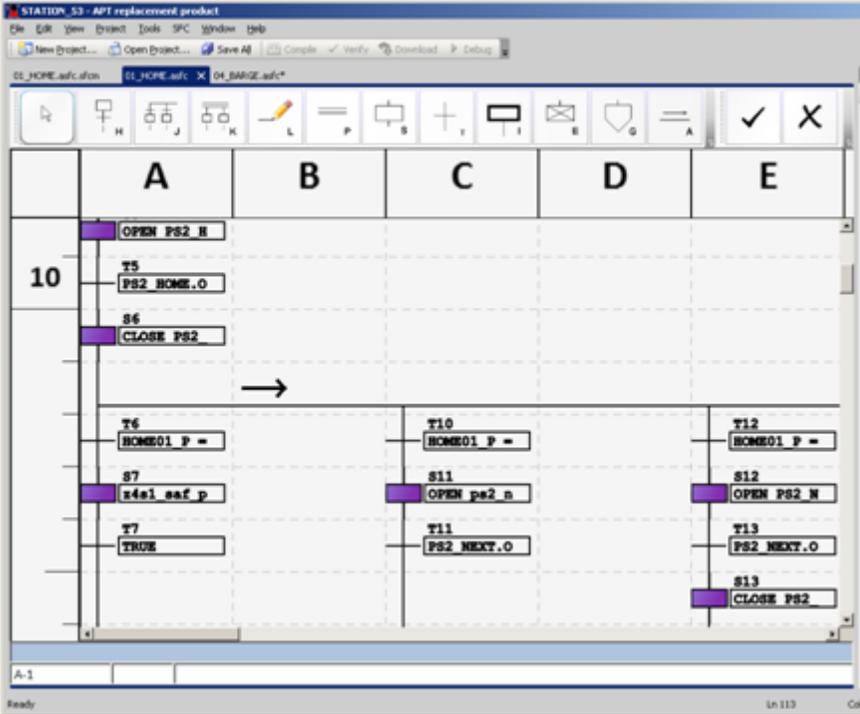
SFC Magnification

Press **Ctrl** and scroll the mouse wheel to magnify the contents of the SFC grid. In the following illustration, the right screen shot is a magnification of the SFC in the left screen shot.



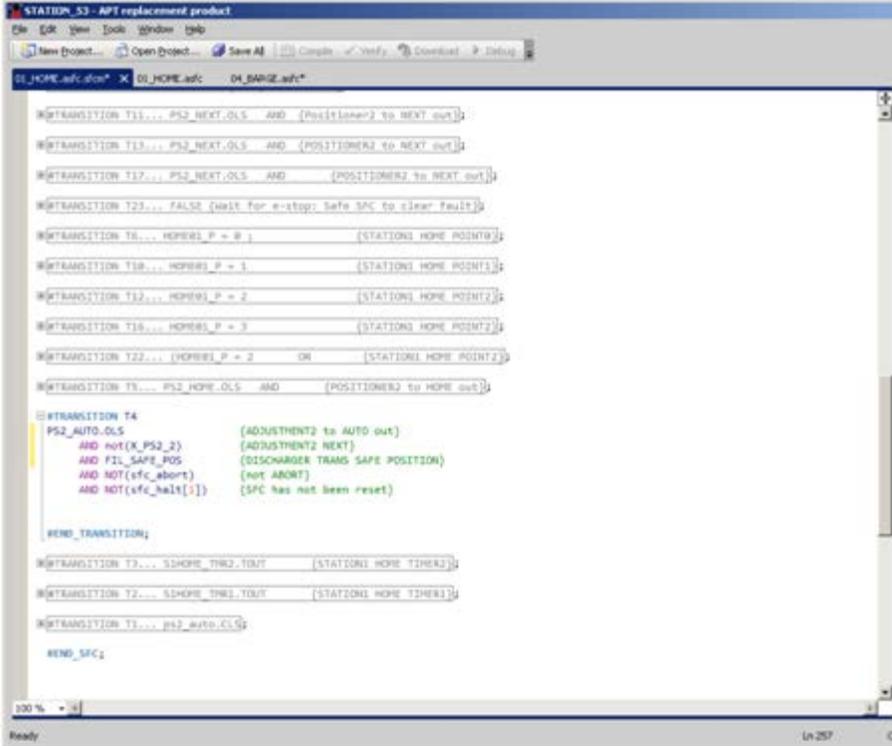
SFC Zoom Mode

Press **Alt-Z** to enable the SFC zoom mode. This displays Comments next to each step and transition, as illustrated. Comments are the first characters in the statements of each step and transition. Set the Comment width by pressing **Alt-W** and entering a value from 0 – 68.



Displaying and Editing Math Statements in Sequential Function Charts

Use the arrow keys or mouse to place the SFC cursor on a step or transition, and press **F10** to display its math block in the SFC Math Language Text Editor, as illustrated.



Code Regions

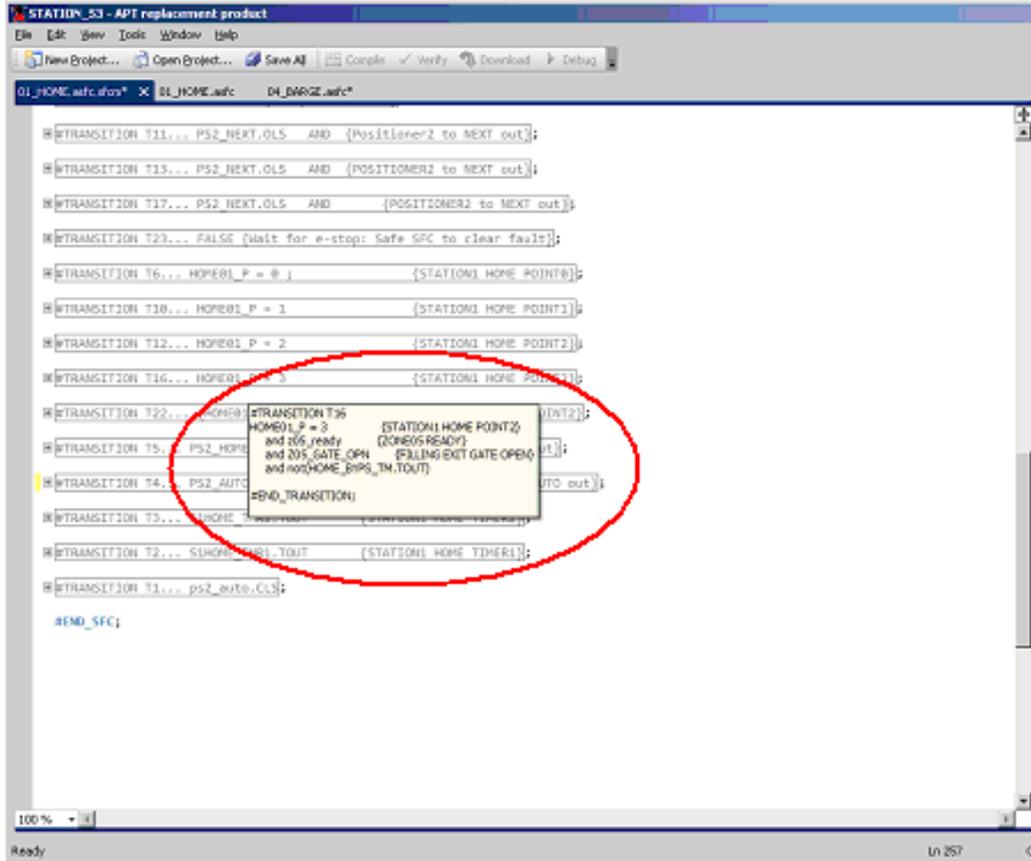
The text editor in APT displays the math statements for the selected SFC step or transition only. However, the Math Statement Text Editor in FTSolution for APT provides access to the math statements for every step and transition in the SFC.

This editor uses Code Regions to expand and collapse the statements for each step and transition.

Left click a + plus sign to expand the statements for that step or transition, and click the – minus sign to collapse the statements. Or locate the cursor within a code region and press **Ctrl+MM** to toggle its expansion and collapse.

Flyby Windows Containing Statements Within a Collapsed Code Region

You can view the statements within a collapsed code region by hovering the mouse pointer over it. As illustrated, a flyby window containing the statements in the collapsed region appears. These flyby windows allow you to examine the statements in a collapsed region without expanding it.



New Instructions that Separate Steps and Transitions

The text editor in APT displays the math statements for the selected SFC step or transition only. However, the Math Statement Text Editor in FTSolution for APT provides access to the math statements for every step and transition in the SFC.

Because the math statements of all steps and transitions can be accessed through the same SFC Math Language Text Editor, each step and transition must be separated from the other steps and transitions.

To separate math statements, step statements must start with the **#STEP *step_name*** instruction and end with the **#END_STEP** instruction. Similarly, transition statements must start with the **#TRANSITION *transition_name*** instruction and end with the **#END_TRANSITION** instruction, as illustrated.

These new instructions are required to separate the math statements of all steps and transitions from each other.

```

#STEP S2
y_aps5_3 := TRUE;           {LOCATOR2 E_STOP}
delay s23plate_tmr1;       {CUTTER1 HOME TIMER1}

#END_STEP;

#TRANSITION T21
S2_NEXT.OLS AND           {COMPRESSOR2 to NEXT out}
NOT(sfc_halt[1])          {SFC has not been reset}

#END_TRANSITION;

```

New Text Editor Features

The SFC Math Language Text Editor provides features that increase the speed and accuracy of writing programs.

Syntax Coloring

The text editor uses **Syntax Coloring**, in which keywords, variables, operators, and comments are displayed with different colors, as illustrated. This visually distinct method of displaying statements improves program readability.

```

{***** Orientation by 12 step operation / target input ***** }
{===== Start positioner 4 =====}
IF (sel_pos_no = 4) AND
    EDGE(ps_begin_tmr.ENABL) AND           { Start button }
    NOT(x_s04_motion)                     { Stalled }
THEN
    position_cod[4] := destination_code;   { Input Target }
    s04_begin := TRUE;
ENDIF;

```

Hover Text

The text editor provides **Hover Text**, in descriptive tooltips that appear when the mouse pointer hovers over keywords and variables. These tooltips contain information regarding the item over which the mouse pointer hovers. In the following illustration, the tooltip appeared when the mouse hovered over the variable `x_s04_ind_dn`.

```
{ Refine index complete
s04_comp := x_s04_ind_dn AND NOT(s04_start)
          A Math language identifier NOT(ps04_f
IF x_s04_motion OR ( s04_ftr AND alarm_reset
```

Incremental Parsing

APT identifies errors only after exiting the text editor and compiling. However, the FTSolution for APT editor identifies errors immediately with **Incremental Parsing**, in which each line is validated as it is typed. Errors are underlined in red and described in the **Error List** window. Select the **View \ Error List** menu item to display the Error List window. In the following illustration, the red underline in the second `TIMER SLOW` line identifies the “`:=`” syntax error.

```
PRAGMA("RLL");
TIMER SLOW : jg1ac1at_timer := 15,15,FALSE,FALSE;
TIMER SLOW : jg1ac2at_timer ::= 15,15,FALSE,FALSE;
BEGIN
```

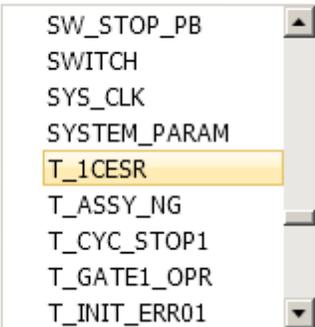
Statement Completion

The text editor utilizes **Statement Completion**, in which lists of keywords and variables are offered, based upon the characters entered. In the following illustration, the list of keywords and variables appeared, when “`t_`” was typed into the line. The `T_1CESR` item is highlighted in the list because it is the closest match to what was typed. Continue typing the remainder of the keyword or instruction or select an item from the list to complete the entry.

```
resetA_timer.ENABL := t_ ;
resetA_timer.RESET :=
resetC_timer.ENABL :=
resetB_timer.RESET :=

s04_reset := alarm_res
y_s04_reset := s04_res

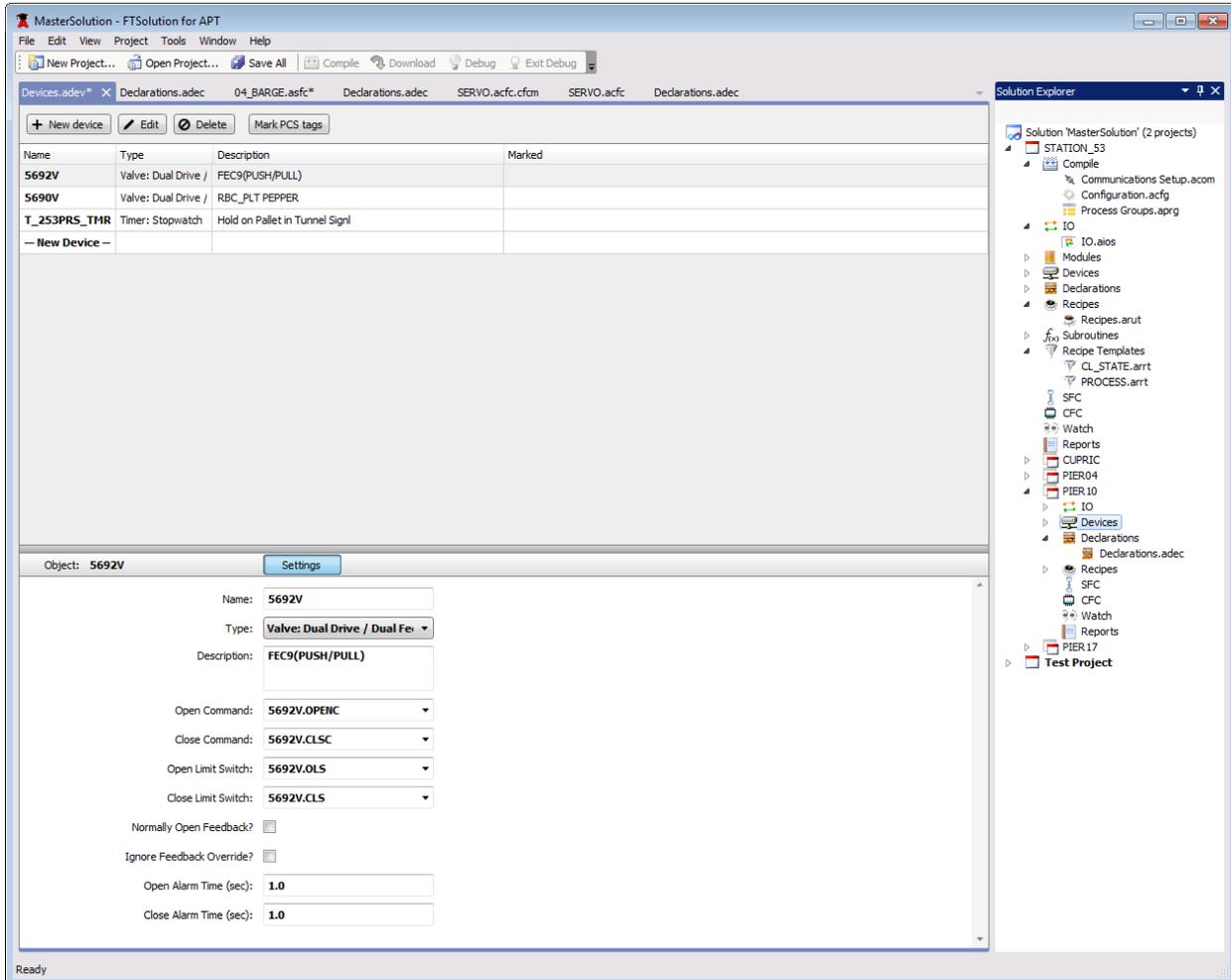
{===== Check cycle ===== }
```



- SW_STOP_PB
- SWITCH
- SYS_CLK
- SYSTEM_PARAM
- T_1CESR
- T_ASSY_NG
- T_CYC_STOP1
- T_GATE1_OPR
- T_INIT_ERR01

Displaying and Editing Devices

In the following illustration, **Devices** of the **PIER10** item in the **STATION_53** Project is selected. Selecting **Devices** from the Solution Explorer along the right edge of the window opens the Devices editor on the left side.



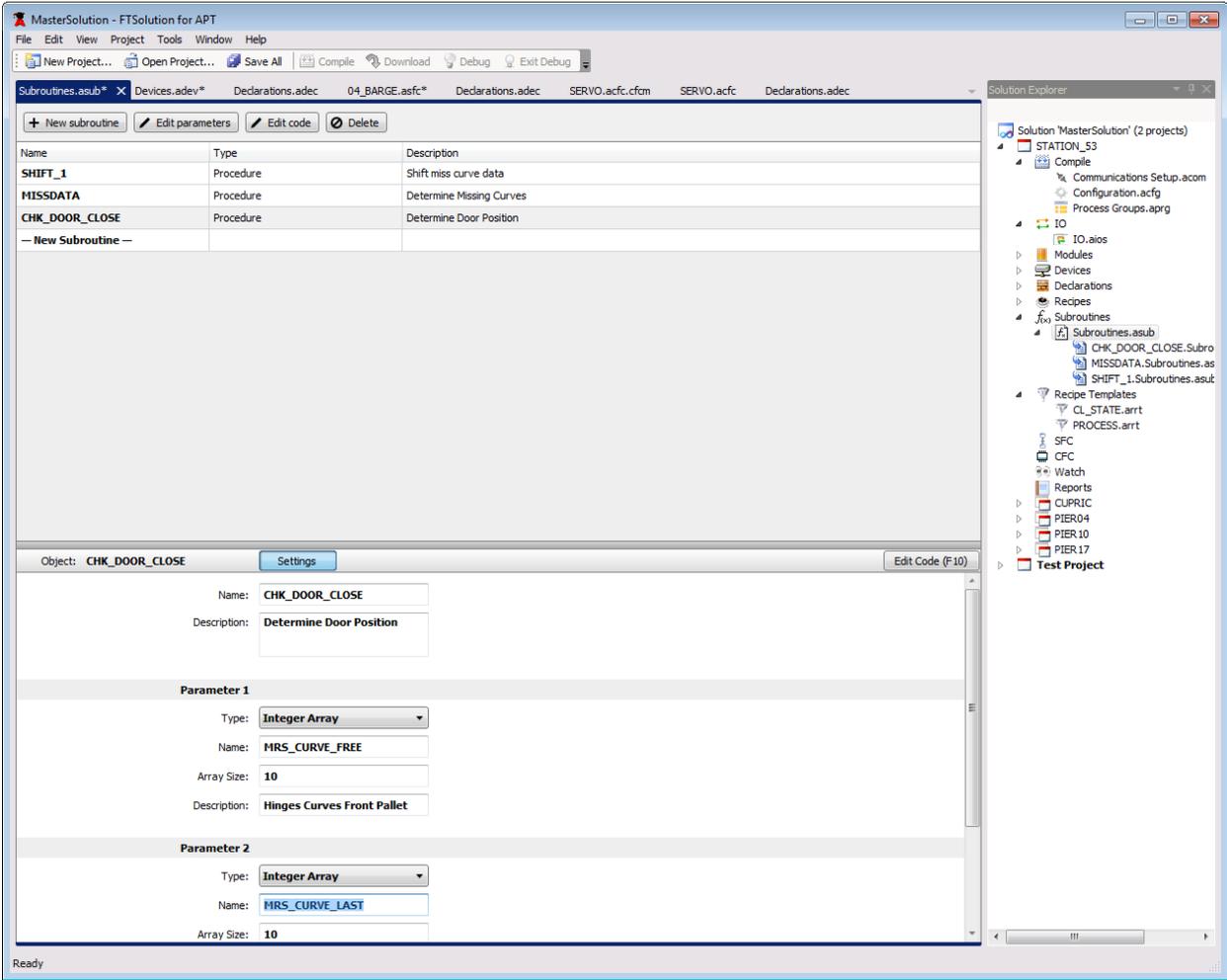
The upper portion of this editor lists all the **Devices** of the **PIER10** item. The lower portion of this editor displays the **Settings** of the Device that is highlighted in the upper portion.

Highlight a Device in the upper portion to display its Settings in the lower portion.

Settings are validated as they are edited.

Displaying and Editing Subroutines

In the following illustration, **Subroutines** in the **STATION_53** Project is selected. Selecting **Subroutines** from the Solution Explorer along the right edge of the window opens the Subroutine editor on the left side.



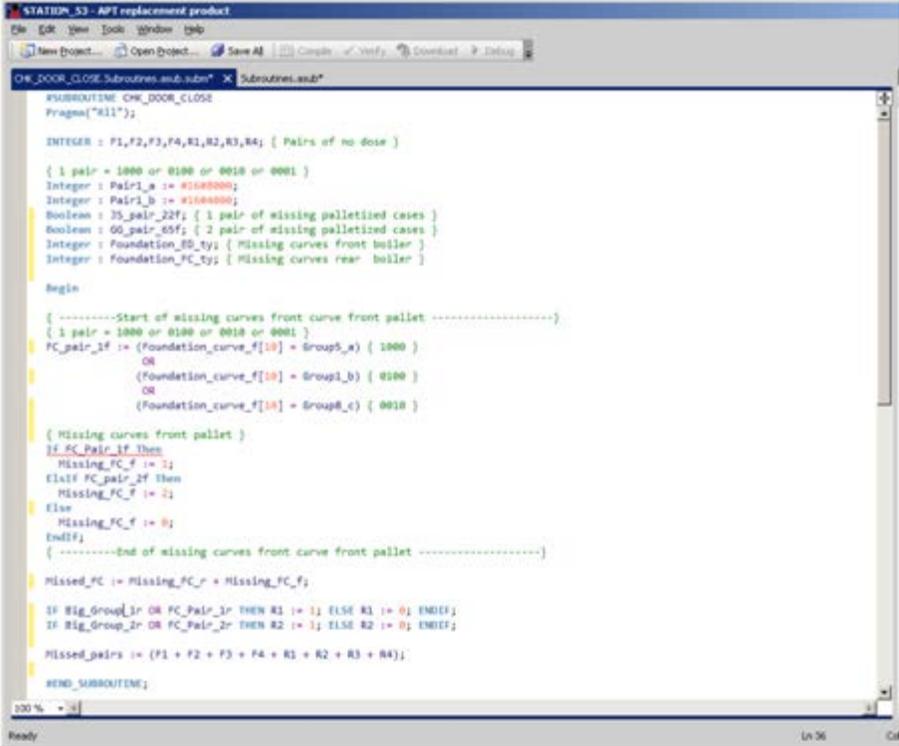
The upper portion of this editor lists all the **Subroutines** of the **STATION_53** Project. The lower portion of this editor displays the **Settings** of the Subroutine that is highlighted in the upper portion.

Highlight a Subroutine in the upper portion to display its Settings in the lower portion.

Settings are validated as they are edited.

Displaying and Editing Math Statements in Subroutines

Press **F10** or click the **Edit Code (F10)** button to display the math statements in the Subroutine Math Language Text Editor, as illustrated.



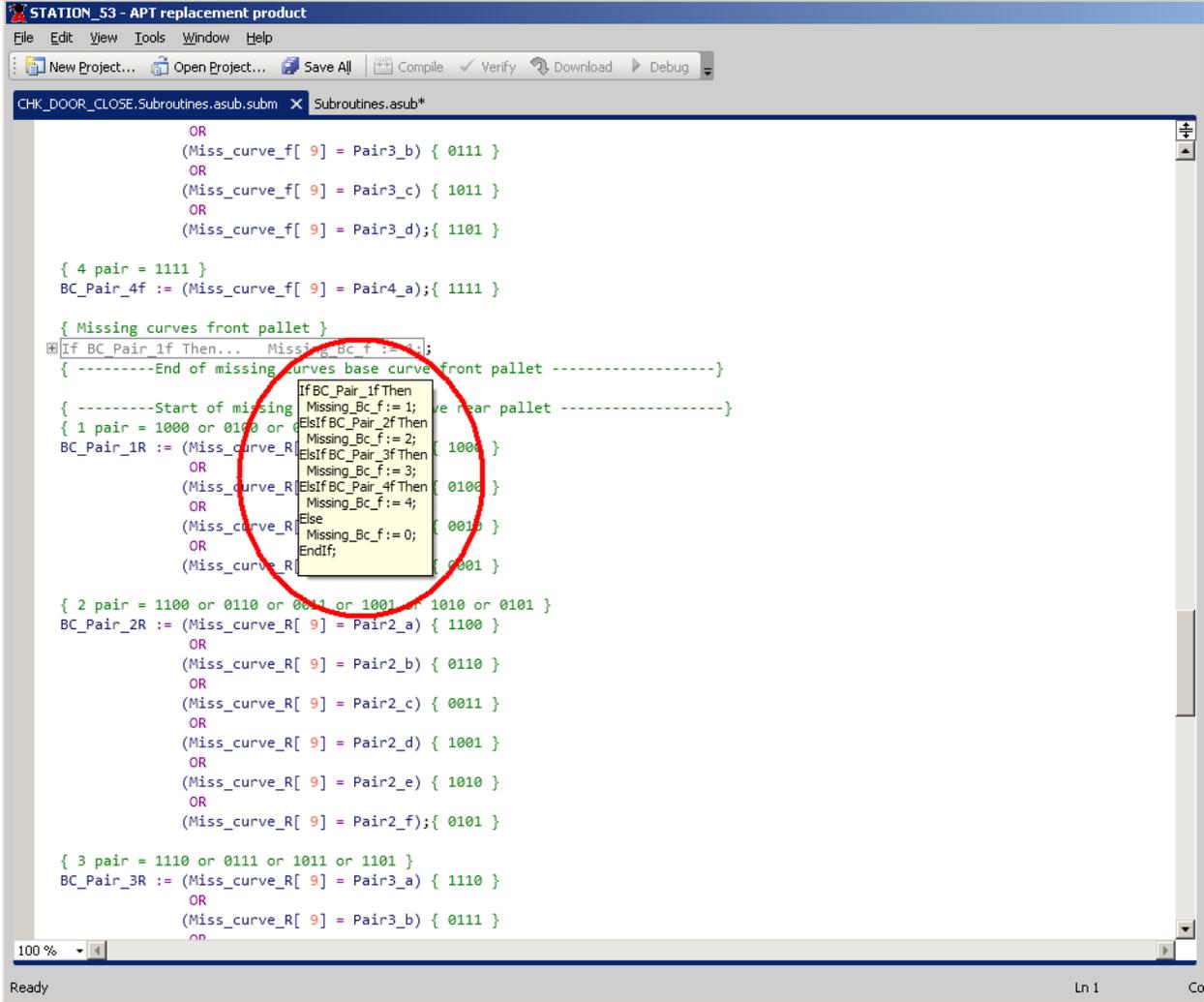
Code Regions

The Subroutine Math Text Editor in FTSolution for APT groups statements into Code Regions that can be expanded and collapsed.

Left click a + plus sign to expand the statements in that region, and click the – minus sign to collapse the statements. Or locate the cursor within a code region and press **Ctrl+MM** to toggle its expansion and collapse.

Flyby Windows Containing Statements Within a Collapsed Code Region

You can view the statements within a collapsed code region by hovering the mouse pointer over it. As illustrated, a flyby window containing the statements in the collapsed region appears. These flyby windows allow you to examine the statements in a collapsed region without expanding it.



New Instructions in Subroutines

The Subroutine Math Language Text Editor identifies the start of each Subroutine with the **#SUBROUTINE *subroutine_name*** instruction and identifies the end with the **#END_SUBROUTINE** instruction, as illustrated.

These new instructions are required at the start and end of each Subroutine.

```
#SUBROUTINE CHK_DOOR_CLOSE
Pragma("R11");

INTEGER : F1,F2,F3,F4,R1,R2,R3,R4; { Pairs of no dose }

{ 1 pair = 1000 or 0100 or 0010 or 0001 }
Integer : Pair1_a := #16#8000;
Boolean : FC_pair_4f; { 4 pair of missing shoes }
Integer : Regulator_FC_f; { Weight from scale }
Missed_FC := Missing_GG_q + Missing_QQ_g;
AB_Pair_4f := (Miss_curve_f[ 9] = Pair4_a);{ 1111 }

#END_SUBROUTINE;
```

New Text Editor Features

The Subroutine Math Language Text Editor provides features that increase the speed and accuracy of writing programs.

Syntax Coloring

The text editor uses **Syntax Coloring**, in which keywords, variables, operators, and comments are displayed with different colors, as illustrated. This visually distinct method of displaying statements improves program readability.

```
{***** Orientation by 12 step operation / target input ***** }
{===== Start positioner 4 ==}
```

IF	(sel_pos_no = 4)	AND	
	EDGE(ps_begin_tmr.ENABL)	AND	{ Start button }
	NOT(x_s04_motion)		{ Stalled }
THEN			
	position_cod[4] := destination_code;		{ Input Target }
	s04_begin := TRUE;		
ENDIF;			

Hover Text

The text editor provides **Hover Text**, in descriptive tooltips that appear when the mouse pointer hovers over keywords and variables. These tooltips contain information regarding the item over which the mouse pointer hovers. In the following illustration, the tooltip appeared when the mouse hovered over the variable `x_s04_ind_dn`.

```
{ Refine index complete
s04_comp := x_s04_ind_dn AND NOT(s04_start)
          A Math language identifier NOT(ps04_f
IF x_s04_motion OR ( s04_ftr AND alarm_reset
```

Incremental Parsing

APT identifies errors only after exiting the text editor and compiling. However, the FTSolution for APT editor identifies errors immediately with **Incremental Parsing**, in which each line is validated as it is typed. Errors are underlined in red, and described in the **Error List** window. Select the **View \ Error List** menu item to display the Error List window. In the following illustration, the red underline in the second `TIMER SLOW` line identifies the “:=” syntax error.

```
PRAGMA("RLL");
TIMER SLOW : jglac1at_timer := 15,15,FALSE,FALSE;
TIMER SLOW : jglac2at_timer ::= 15,15,FALSE,FALSE;
BEGIN
```

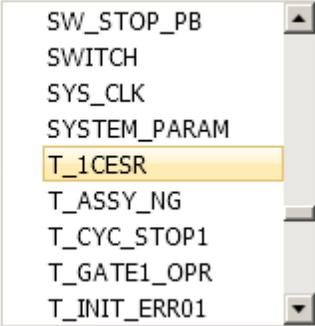
Statement Completion

The text editor utilizes **Statement Completion**, in which lists of keywords and variables are offered, based upon the characters entered. In the following illustration, the list of keywords and variables appeared, when “`t_`” was typed into the line. The `T_1CESSR` item is highlighted in the list because it is the closest match to what was typed. Continue typing the remainder of the keyword or instruction, or select an item from the list to complete the entry.

```
resetA_timer.ENABL := t_ ;
resetA_timer.RESET :=
resetC_timer.ENABL :=
resetB_timer.RESET :=

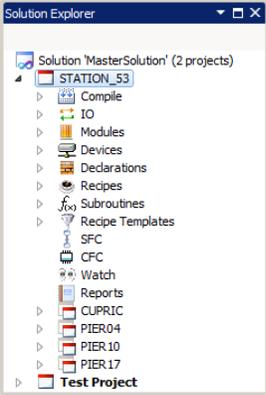
s04_reset := alarm_res
y_s04_reset := s04_res

{===== Check cycle ===== }
```



Compiling an FTSolution for APT Project

The option to compile a project is available when a project (or its compile configuration file) is selected in the Solution Explorer.

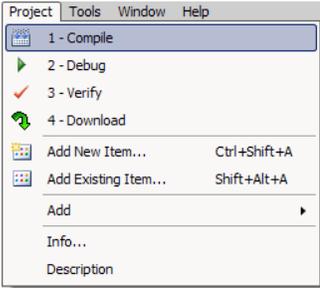


1. When a project is selected in the Solution Explorer, a compile of the project can be initiated in several ways:

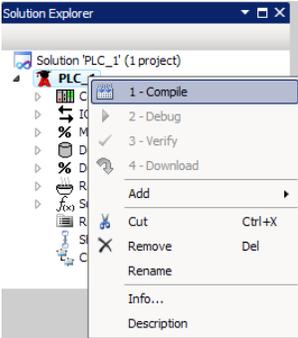
- By pressing the **1** key.
- By clicking **Compile** on the standard toolbar.



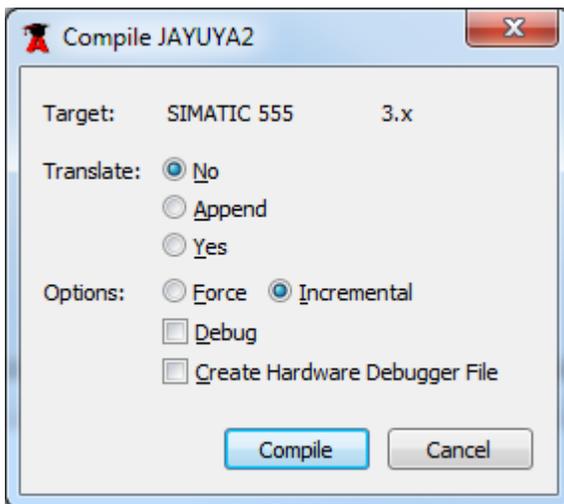
- By clicking **Compile** on the **Project** menu.



- By right-clicking a project in the Solution Explorer and then clicking **Compile** from the shortcut menu.



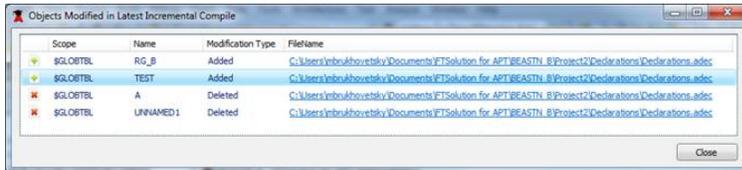
2. After a compile is initiated, the Compile dialog box is displayed.



3. The **Target** represents the controller specified in the compile configuration file.
4. Under **Translate**, specify the translate option you want to use.
 - When you select **No**, existing marked tags are not modified.
 - When you select **Append**, existing marked tags are placed in the *install.tag* file. Tags marked since the last translate build and tags that were unmarked and re-marked are also placed in a separate file, *append.tag*.
 - When you select **Yes**, all marked tags are placed in the *install.tag* file.
5. Under **Options**, specify your compile options.
 - When you select **Force**, a force compile will be performed and any previous force compile and incremental compile files will be removed.
 - When you select **Incremental**, an incremental compile will be performed. The Debug option will be disabled. FTSolution will save the incremental compile file with a name suffix following the pattern "*projectname-inc-n*" where "*inc*" indicates it is an incremental compile and "*n*" indicates the number of incremental compiles that have been successfully performed since the last force compile. See the section *Preconditions for Incremental Compile and Download* for more information on incremental compile and download.
 - When you select **Debug**, you will compile for Debug mode.
 - When you select **Create Hardware Debugger File**, the compile will create an .ahd file.
6. Click **Compile** to start compiling the project.

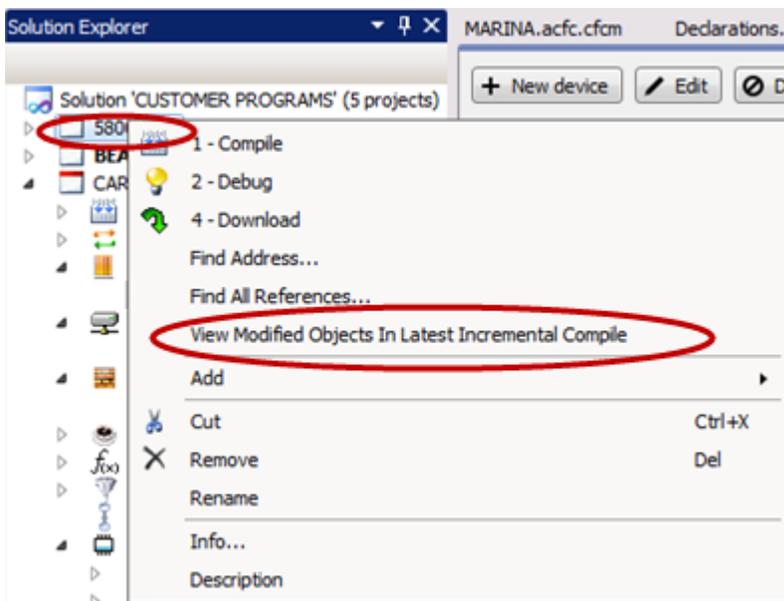
Viewing the Objects Modified by Incremental Compile

Once FTSolution has completed an incremental compile you can view the objects that were modified by the incremental compile. The Modification Type column indicates whether the object was been added, deleted, or modified. In addition, the FileName column includes a hyperlink that will open the file that references (or referenced) the selected object.



In order to view modified objects, select a project that has been incrementally compiled and then do one of the following:

- Right-click the project in the Solution Explorer, and then click **View Modified Objects in Latest Incremental Compile** from the shortcut menu.
- Click **View Modified Objects in Latest Incremental Compile** from the **Project** menu.



Preconditions for Incremental Compile and Download

When user selects an incremental compile, it is allowed to execute under specific conditions. If any of the following conditions are true, the force compile is required:

- No object file exists.
- A change to the configuration is made in any of the following areas:
 - *Report by Exception time slice* has been edited.
 - *Network Communication time slice (575)* has been edited.
 - Reserved locations (whether the numeric value or U-Memory file field)
 - Memory size
 - Controller Type
 - Controller release
 - PCS release
 - Application Id (575)
 - Application Dependencies
 - Number of RCC Cards (56x)
 - *Uninterruptable power supply* option is selected/deselected
 - *Do not convert Time Of Day to integer fields* option is selected/deselected
 - Address allocator runs out of memory to contain a particular type of data
- An item is marked or unmarked for PCS tagging when *Build translate table* option is set to Yes
- For an item, the PCS tag is renamed when *Build translate table* option is set to Yes
- A unit is deleted from the program, whether it had been included in the *Units to Compile* or not
- New I/O assignments occupy currently allocated overflow CR addresses as Y data types
- A change to the configuration is made in one of the following areas:
 - A unit is excluded from *Units to Compile*
 - *Build translate table* option is changed from No/Append to Yes

Preconditions for Incremental Download in PROGRAM mode

Typically the incremental download will take place as long as the user has first successfully executed an incremental compile, however, there are certain conditions that can entirely prevent an incremental download from taking place. If the incremental download is prevented, a full download is the user's only option, which requires that the PLC is placed in PROGRAM mode.

An incremental download may not occur if any of the following conditions are not true:

- The program has been successfully compiled incrementally
- The resulting object file exists.

Preconditions for Incremental Download in RUN mode

APT is sending a series of task codes to the PLC to accomplish the program rewrite while the PLC is in Run mode. APT relies on the Task Code specification of the 505 Series PLCs to perform incremental download.

Ideally users would want to execute an incremental download while the controller is in RUN mode, however, there are certain conditions that prevent this from occurring.

An incremental download may not occur if any of the following conditions are not true:

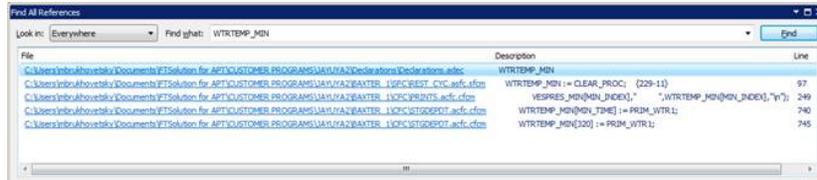
- The program has been successfully compiled incrementally,
- The resulting object file exists.

If any of the following conditions are true, the PLC must be placed in PROGRAM mode when executing the incremental download:

- A re-sequencing of SFC steps has occurred.
- 575 PLC only, any of the following time line configuration settings were changed:
 - Total scan time
 - Loop time slice
 - Analog alarm time slice
 - Cyclic SF program time slice
 - Priority SF program time slice
 - Normal SF program time slice
 - Ladder SF sub time slice
 - Normal communications time slice
 - Priority communication time slice
 - Ladder SF sub zero
 - Cyclic RLL scan time
- In any PLC, any of the following were changed:
 - The constants area (K) for program objects \$DP_MAKEDATE, \$DP_PROGNAME, \$DP_VERDATE, or \$DP_VERSION.
 - The memory configuration
 - The first network in PLC
 - The last network in PLC is changed or has different RLL address location.

Finding References to an Object

FTSolution includes the ability to find all the locations where an object is referenced through the Find All References tool. The File column includes a hyperlink to the file that references the selected object. In addition, the Description column contains either the name of the object or the line of code that includes the reference to the object (from the code behind file). Finally, the Line column contains the specific line number that includes the reference to the object (if the reference is from a code behind file).



To perform a Find all references:

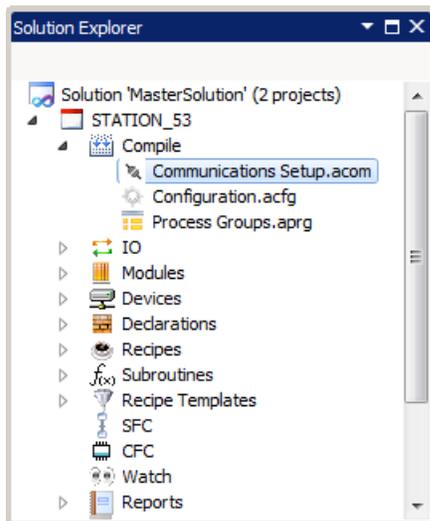
- Select a project and then do one of the following:
 - Right-click the project or the compile object in the Solution Explorer, and then click **Find All References** from the shortcut menu.
 - Click **Find All References** from the **Project** menu.
- Select the scope of your search from the **Look in** drop down list. The look in list options include:
 - Everywhere:** the search is done on program and unit level.
 - Program:** the search is performed on program level only.
 - Unit name:** the search is done only within given unit scope.
- Enter the object name in the **Find What** box and click **Find**.

Communication Setup

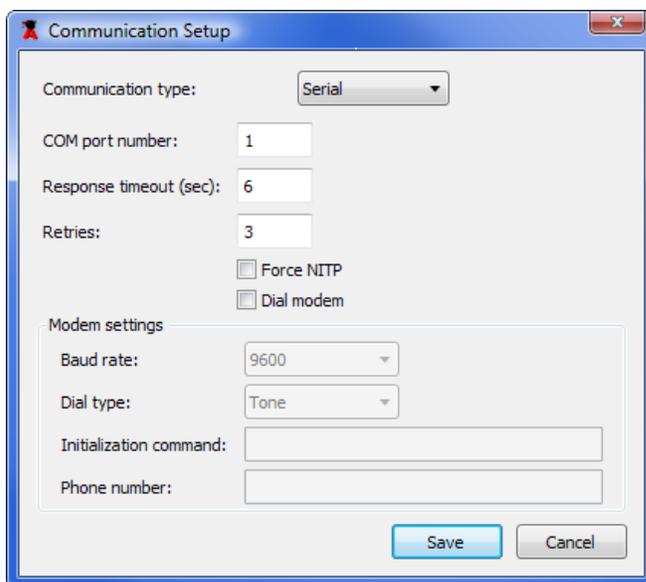
Every FTSolution for APT project has a file that defines its communication settings. This file, named **Communications Setup.acom**, is accessible from the Solution Explorer. Communication settings define how FTSolution for APT will communicate to a PLC.

To specify the communication settings:

1. View a project in the Solution Explorer.
2. Expand **Compile** so that the **Communications Setup.acom** file is visible.

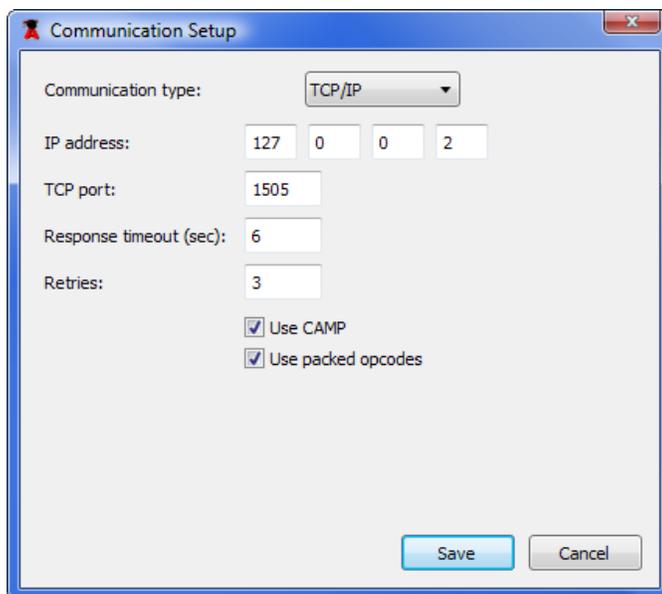


3. Double-click **Communications Setup.acom**. The Communication Setup dialog box is displayed.



4. In **Communication Type**, select **Serial** to connect to a PLC using serial communication; select **TCP/IP** to connect to a PLC/Ethernet adapter using TCP/IP.

5. If the **Communication Type** is **Serial**, specify the following settings:
 - In the **Com Port Number** box, enter the communication port you want to use.
 - In the **Response Time Out** box, enter the response time out you want to use. The response time out specifies the amount of time, in seconds, the software waits for a response from the PLC before returning a time out error.
 - In the **Retries** box, enter the number of times the software will try to re-establish communications with the PLC after a time-out error.
 - To force the use of NITP, select the **Force NITP** check box. (This specifies ASCII communication for use with Simatic 545, 555, and 575 PLCs.)
 - When the selected form of serial communications is through a modem, select the **Dial Modem** check box, and then specify the following modem settings:
 - In the **Baud Rate** box, select the rate of communications between the computer and modem.
 - In the **Dial Type** box, select the type of dialing you want to use. Select Pulse dialing only if this is the only type of dialing supported by your phone line.
 - In the **Initialization Command** box, enter any initialization commands required by your modem. Consult your modem manual for a list of appropriate commands.
 - In the **Phone Number** box, enter the phone number to be dialed.
6. If the **Communication Type** is **TCP/IP**, specify the following settings:

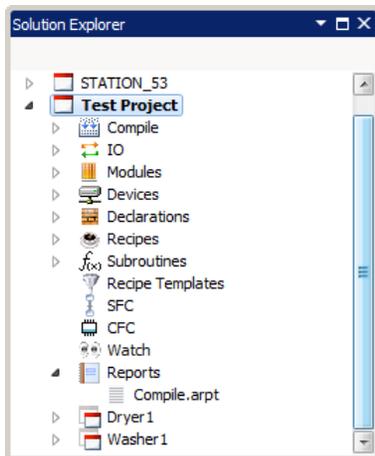


- In the **IP Address** box, enter the IP address of the PLC/Ethernet adapter to which you want to connect.
- In the **TCP Port** box, enter the TCP port used by the PLC/Ethernet adapter to which you are connecting.
- In the **Response Time Out** box, enter the response time out you want to use. The response time out specifies the amount of time, in seconds, the software waits for a response from the PLC before returning a time out error.

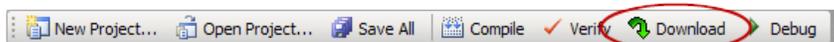
- In the **Retries** box, enter the number of times the software will try to re-establish communications with the PLC after a time-out error.
 - If the PLC/Ethernet adapter to which you are connecting supports CAMP protocol, select the **Use CAMP** check box.
 - If the PLC/Ethernet adapter to which you are connecting supports Packed Opcodes, select the **Use Packed Opcodes** check box.
7. Click **Save** to save the communication settings.

Downloading a Compiled FTSolution for APT Project

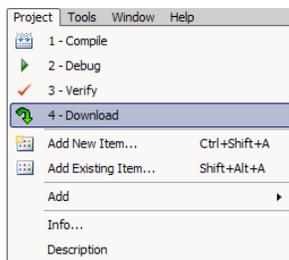
The option to download a project is only available when a project (or its compile configuration file) is selected in the Solution Explorer and the project has already been compiled.



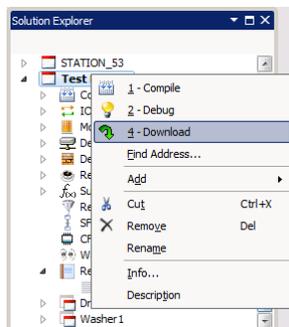
1. If a project is currently selected and it has already been compiled, initiate a download by doing one of the following:
 - Pressing the **4** key.
 - Clicking **Download** on the standard toolbar.



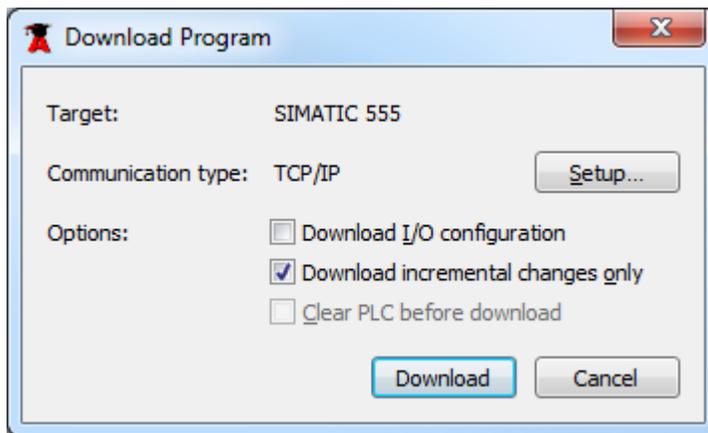
- Clicking **Download** on the **Project** menu.



- Right-clicking a project in the Solution Explorer and then clicking **Download** from the shortcut menu.



2. After a download has been initiated, the Download Program dialog box is displayed.



- The **Target** reflects the controller type defined in the compiler configuration file.
- The **Communication type** displays the currently selected communication type.

Note: If communication settings have not been configured, **Not configured** is displayed for the **Communication type**, and the **Download** button cannot be selected. To set or modify the current communication settings, click **Setup**. (Detailed information about communication setup is provided in the previous section)

- To download the I/O configuration, select the **Download I/O configuration** check box.
- To download an incremental compile, select the **Download incremental changes only** check box.
- To clear the PLC before downloading, select the **Clear PLC before download** check box. The Clear PLC before download option will be disabled when Download incremental changes only is selected.

3. Click **Download** to begin the download.

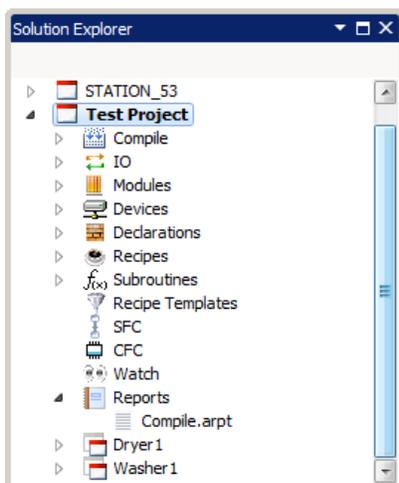
Debug Mode

Warning About Debug Mode

Debug mode enables transitioning from Run to Program mode. When using Debug Mode, be aware of the consequences when you transition from Run to Program Mode and outputs are cleared.

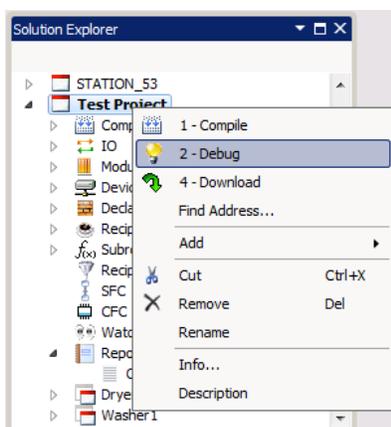
Starting Debug Mode

Once an FTSolution for APT solution has been compiled with the Debug Mode option enabled and downloaded to a PLC, Debug Mode can be activated whenever a project or the Compile object is selected in the Solution Explorer. In the following figure, the project named **flipper1** is selected in the Solution Explorer.

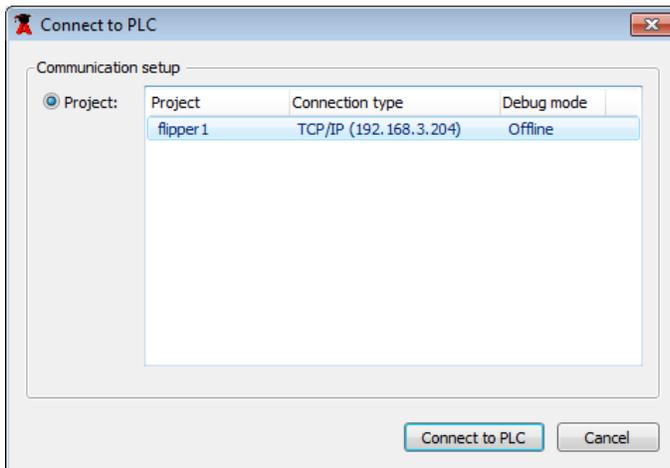


If a project or Compile object is selected, activate Debug Mode by doing one of the following:

- Right-clicking the project or the compile object in the Solution Explorer, and then clicking **2-Debug** from the shortcut menu.
- Clicking the **Debug Mode** toolbar button on the Standard toolbar.
- Clicking **2-Debug** from the **Project** menu.



After activating debug mode, the Connect to PLC dialog box appears.



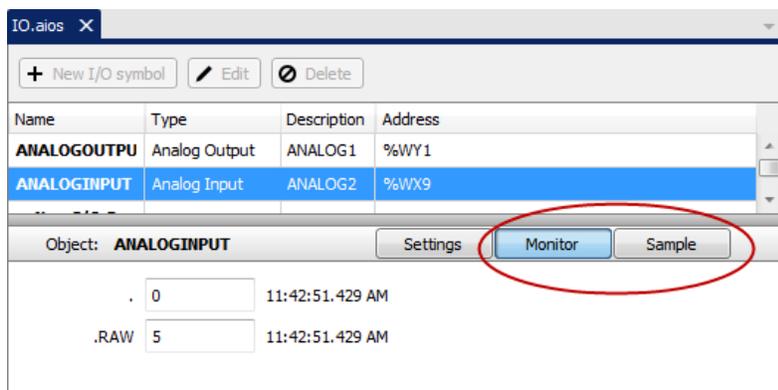
In the Connect to PLC dialog box, select the project you want to debug and then click **Connect to PLC**.

Monitoring Variables in Debug Mode

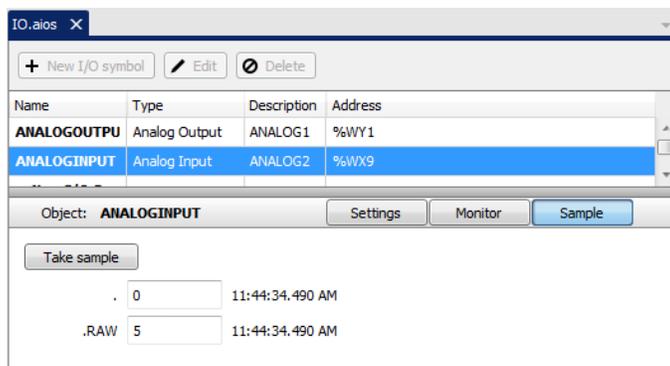
Watch windows in FTSolution for APT are analogous to the charts used to monitor variables in APT.

Built-In Watches

When Debug Mode is active, FTSolution for APT provides built-in Watch window capabilities. While in Debug Mode, selecting an object in a table editor or the CFC editor causes a **Monitor** tab and a **Sample** tab to be displayed in the Settings Editor.



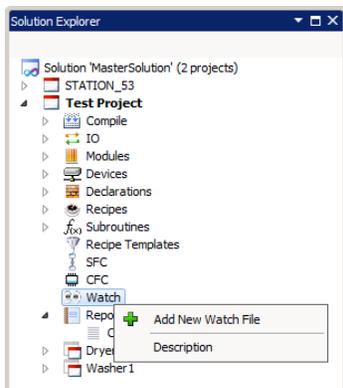
- Select the **Monitor** tab to display continuously updating data about the object's settings.
- Select the **Sample** tab to display a single instance of data about the object's settings. Click the **Take Sample** button on the **Sample** tab to update the data.



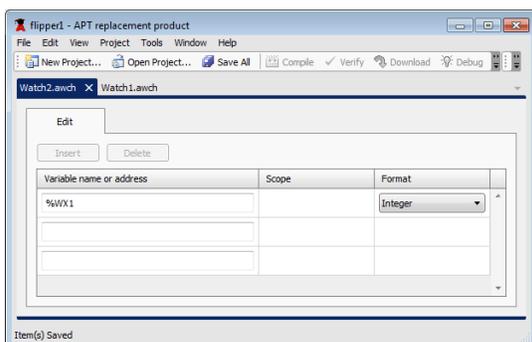
Custom Watches

To create a custom watch in FTSolution for APT, do the following:

1. Right-click the Watch object in Solution Explorer and select **Add New Watch File** from the shortcut menu.



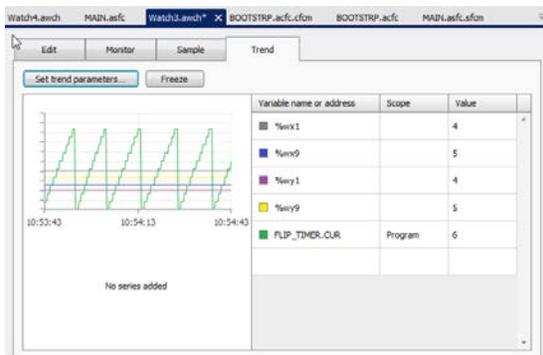
2. In the Add New Item dialog box, enter a name for the Watch file and click **Add**. The Watch file should now be visible in the Solution Explorer.
3. Double-click the Watch file in Solution Explorer.
4. Enter the variables or addresses you want to add to this Watch window.



5. Save the Watch window.

In Debug mode, a custom Watch window display a **Monitor** tab, a **Sample** tab, and a **Trend** tab. Clicking the **Trend** tab displays a graphical representation of the data monitored in the Watch window.

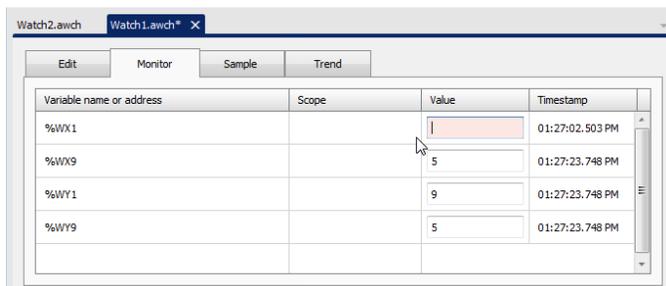
The following figure is an example of the information in the Trend tab.



Modifying I/O in a Watch Window

To modify the value of an input/output point or a variable:

1. Add the input/output point or variable to a Watch window. Or, you can open a built-in Watch by selecting an object's editor or a CFC editor.
2. In the Watch window, select the input/output point or variable you want to modify, select the **Monitor** tab, and enter the value you want to use in the **Value** box.

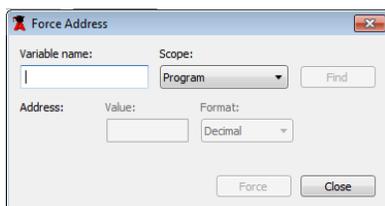


Forcing I/O

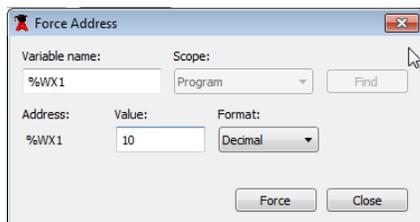
1. To force the value of input/output point or a variable, click **Forces** on the **Debug** menu.
2. In the Forces dialog box, click **Set Force**.



3. In the Force Address dialog box, enter the input/output point or variable you want to force in the **Variable name** box, and then click **Find**.



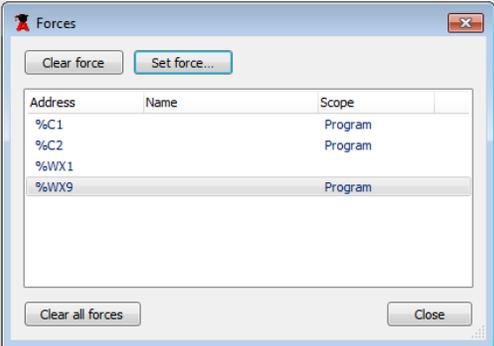
4. In the **Value** box, enter the value you want to use.



5. Click **Force**. This input/output point or variable is now forced. A list of all forced items is displayed in the Forces dialog box.

Clearing Forced I/O

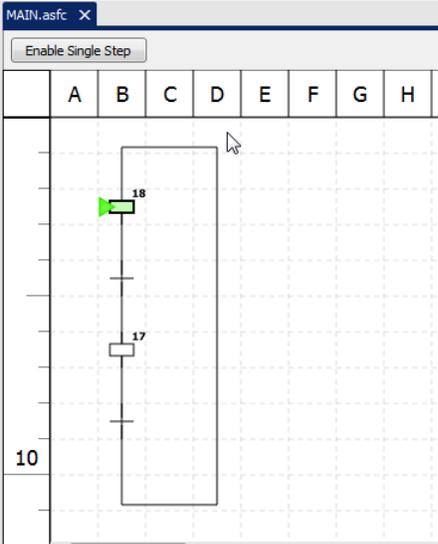
- 1. Click **Forces** on the **Debug** menu. A list of all forced items appears in the Forces dialog box.



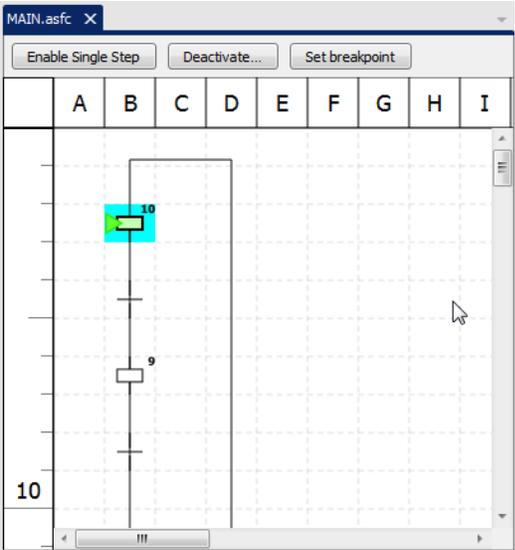
- 2. To clear all forced items, click **Clear all forces**.
- 3. To clear a single force, select the force you want to clear and then click **Clear Force**.

Tracing SFC Program Flow

In Debug Mode, SFC program flow is automatically displayed in the SFC editor. The current SFC step is indicated by a green arrow.

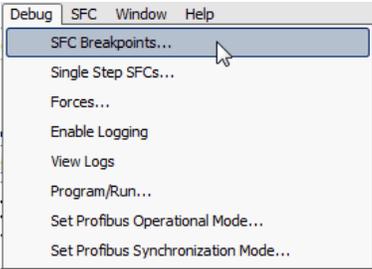


Additional operations, such as the ability to activate and deactivate a step, set and remove break points, and enable and disable Single Step Mode, are accessible using the buttons along the top of the SFC editor. The buttons are displayed whenever a step in the SFC editor is selected.

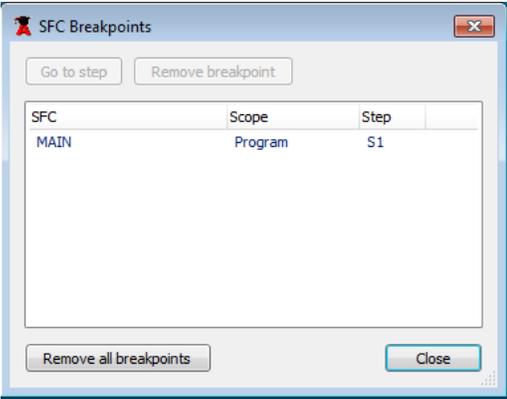


Additional debug operations related to SFC can be accessed using the **Debug** menu.

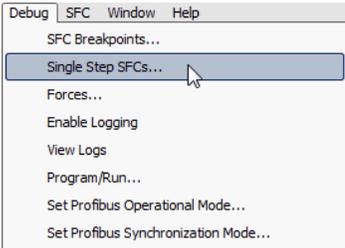
- To view a list of SFC breakpoints, click **SFC Breakpoints** on the **Debug** menu.



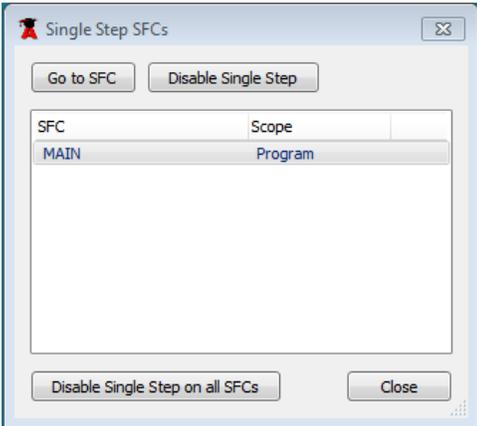
A SFC Breakpoints dialog box is displayed.



- To view single step SFCs, click **Single Step SFCs** on the **Debug** menu.

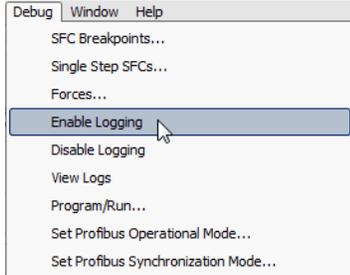


A Single Step SFCs dialog box is displayed.



Logging

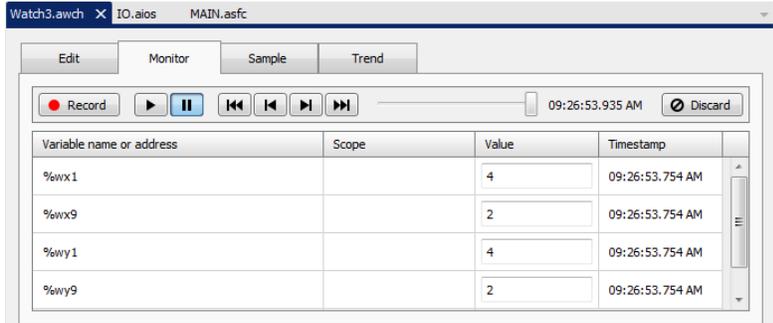
Logging is enabled by clicking **Enable Logging** on the **Debug** menu.



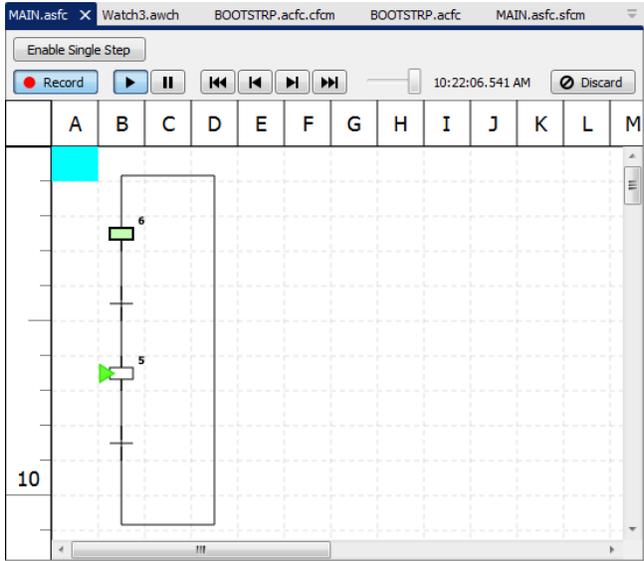
Once Logging is enabled, a Logging Timeline Control bar is accessible from table editors, CFC and SFC editors, and Watch windows.



The Logging Timeline Control bar is used to start or stop recording, begin playback, pause playback data, and jump to a step. The following figure shows the Logging Timeline Control bar in a custom Watch window.



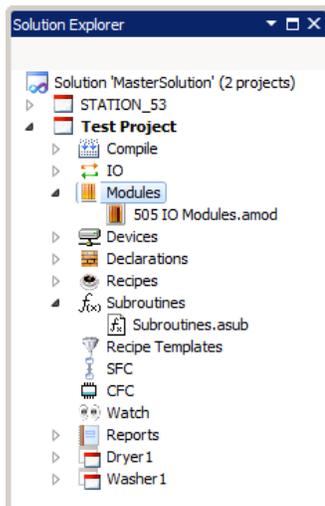
The following figure shows the Logging Timeline Control bar in a SFC editor.



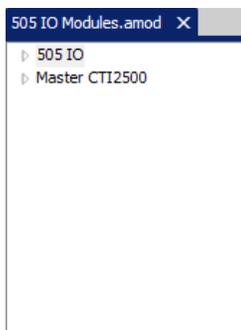
Adding an I/O Module for a 505 Series / CTI Controller

Use the following procedure to define an I/O module for a 505 Series or CTI controller: (Note: A controller type must be defined in FTSolution for APT solution order to define an I/O module).

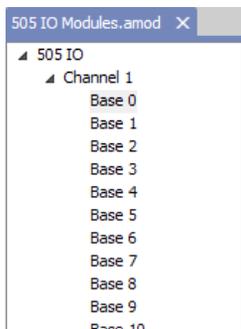
1. View a project in Solution Explorer.
2. Expand % **Modules**.



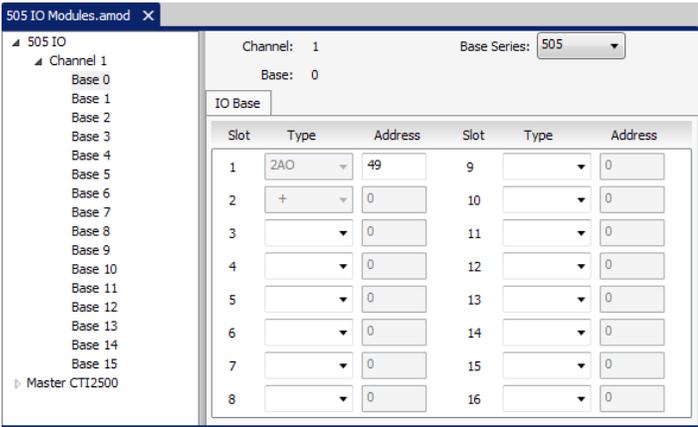
3. Double-click a modules table definition file (the default table definition file is named *505 IO Modules.amod*). A .amod window is displayed.



4. In the .amod window, expand **505 IO**.
5. Expand a channel.
6. Select a base.



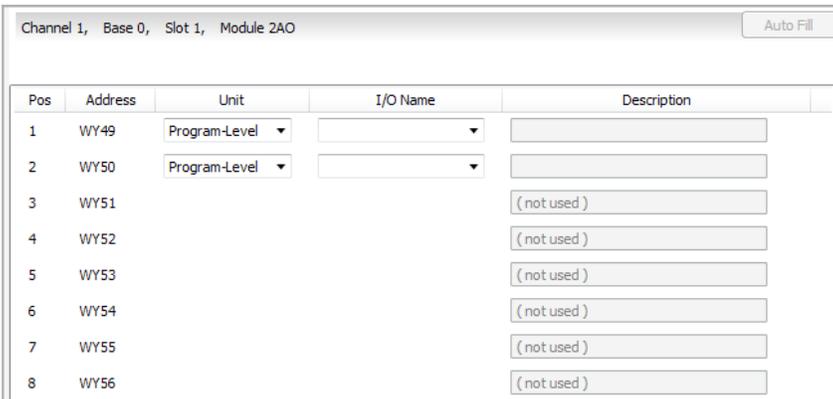
- 7. In a slot you want to define or edit, select or type a module name in the **Type** box. For example, click **2AO** in slot 1 to add an analog output module.



- 8. Press **TAB** to move to the **Address** box. The address of the first available I/O eight-point boundary is automatically calculated and displayed in the **Address** box. You can accept the default address or enter a different address in the **Address** box.
- 9. Verify that the configured I/O matches the physical I/O. You can press **ALT+S** to change the width of the slot if necessary, **ALT+I** to search for I/O symbol names, **ALT+M** to move the module, and **ALT+D** to delete the module.

Defining I/O Points for 505 I/O

The I/O point editor is displayed below the module editor. When a slot is selected in the I/O module editor, its associated points are displayed in the I/O point editor.



To define or edit an I/O point, select the appropriate point and then enter the following information.

- In the **Unit** box, select the unit you want to use, or select **Program-Level**.
- In the **I/O Name** box, select or enter the I/O symbolic name you want to use.
- In the **Description** box, type a description.

Save the I/O configuration, and the associated I/O points, by clicking **Save** on the **File** menu.

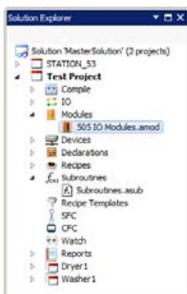
Profibus Configuration

You can use FTSolution for APT to create and manage Profibus networks and devices. A compatible PLC serves as the host of a Profibus master module. (Note: A controller type must be defined in FTSolution for APT solution in order to configure a Profibus network).

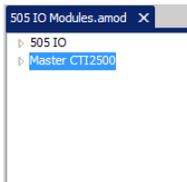
If you are importing an APT program that contains Profibus configuration, the Profibus configuration will be automatically generated as part of the import process. You will not be able to edit the Profibus configuration within FTSolution for APT, however, without first adding the required .gsd files to the hardware directory and then performing a merge using the associated .2bf file. (Detailed information about performing a merge can be found on page 78).

Use the following procedure to open Profibus configuration screens within FTSolution for APT:

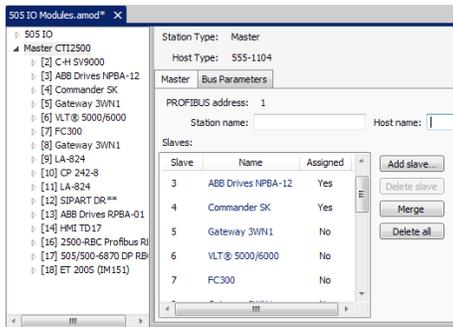
1. View a project in Solution Explorer.
2. Expand % **Modules**.



3. Open a modules table definition file (the default table definition file is named *505 IO Modules.amod*). A .amod window is displayed.



4. In the .amod window, select the PLC type that is acting as the master module. For example, if you have configured a CTI 2500 controller, select **Master CTI2500**. At this point you can add .gsd files to the Hardware Directory and then add slaves to your Profibus network.
 - The master module has a PROFIBUS address of 1.
 - You can enter a **Station name** and **Host name**. These names can help identify this Profibus network when used with other Profibus networks.

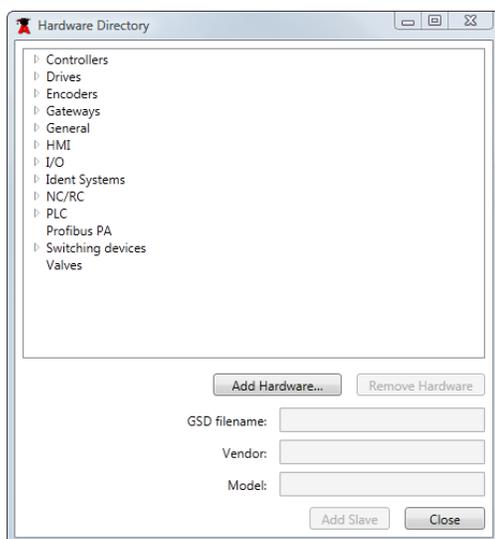


Adding GSD files to the Hardware Directory

The Hardware directory displays and organizes the Profibus slave devices available for configuration. Each device in the Hardware Directory is a representation of a device master file (.gsd) that has been added to the directory. The information contained in the .gsd determines how and where the device is displayed in the directory. Once the .gsd associated with a device has been added to the hardware directory, that device may be assigned to the Profibus network as a slave.

When a master module is selected in the 505 IO Modules.amod window, use the following procedure to add .gsd files to the Hardware Directory within FTSolution for APT.

1. Click **Add slave**. The Hardware Directory window appears.

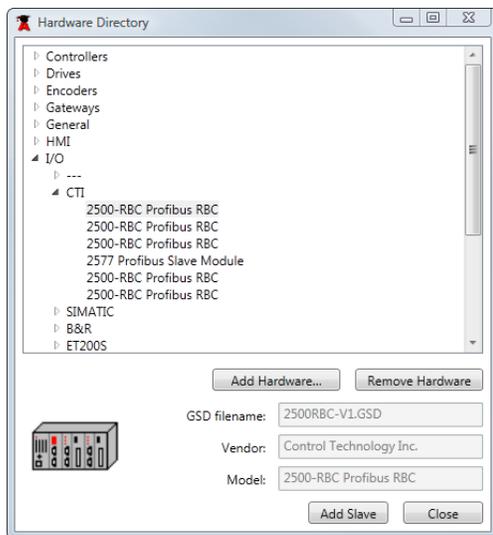


2. Click **Add Hardware**.
3. **Browse** to the location that contains the .gsd file(s) of the devices you want to add and select them.
4. Click **Open** to add the .gsd file(s) to the Hardware Directory. When adding one or more .gsd files to the Hardware Directory, the files are scanned for unsupported or unrecognized information. If any unsupported information is found, a warning message appears, allowing you to view an event log for more details about the problem. In most cases, the device is still added to the Hardware Directory, with any unsupported features ignored. Contact your device manufacturer for more information.

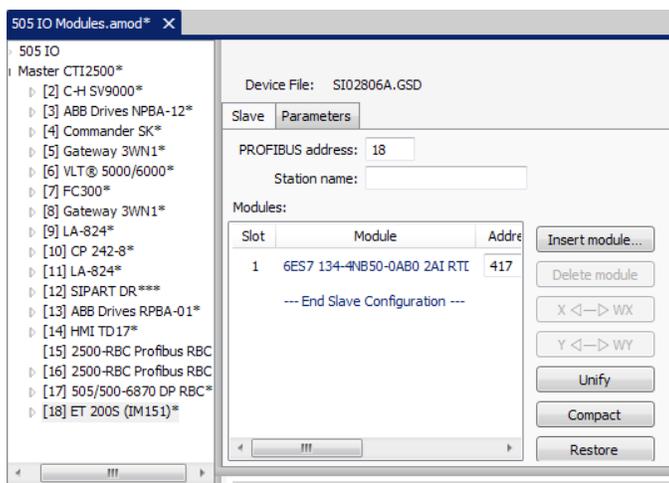
Assigning Slaves to the Profibus Configuration

When a master module is selected in the 505 IO Modules.amod window, use the following procedure to assign slaves to the Profibus configuration:

1. Click **Add Slave**. The Hardware Directory appears.



2. Browse the Hardware Directory to find and select the slave device you want to add. When you select a device, information about the device is displayed in the **GSD filename**, **Vendor**, and **Model** boxes.
3. Click **Add Slave** to add the selected device to the Profibus Configuration.

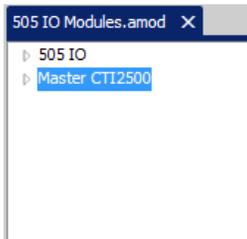


- The first available PROFIBUS address is automatically entered in the **PROFIBUS address** box. If you want, you can type a **Station name** to help identify this device.
4. To configure parameters for the device, select the device you just added under the Master module, click the **Parameters** tab, and then enter the necessary information. Not all devices will have configurable parameters.

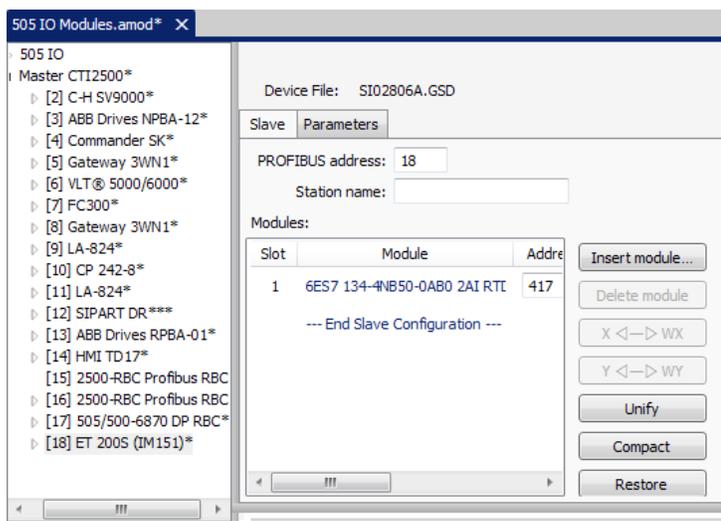
Once a slave is added, you need to configure the modules installed in the slave, and specify the addresses used by the slave's modules.

Configuring and Addressing Profibus I/O Modules

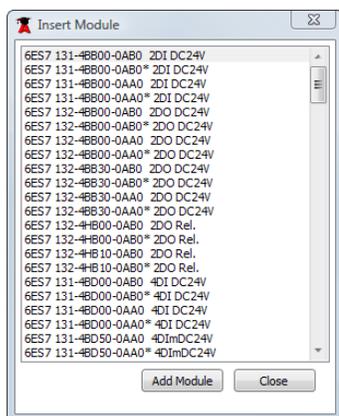
1. Expand the master module in the 505 IO Modules.amod window.



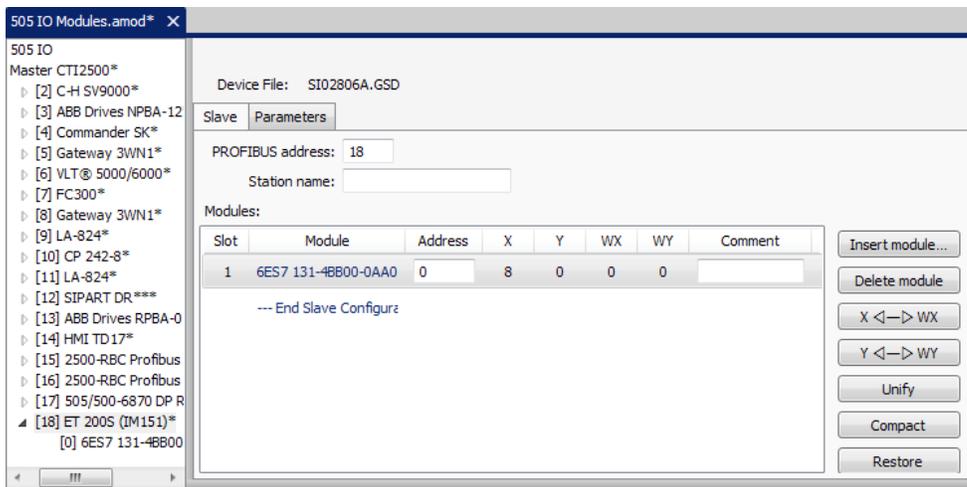
2. Select the slave device you want to configure.



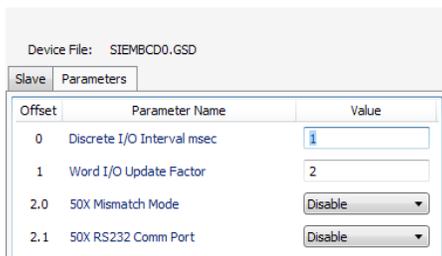
3. Click the **Slave** tab.
4. Click **Insert Module** to add a module for the selected slave device. An Insert Module dialog box appears, which displays modules for the slave.



5. Select the module you want to add, and then click **Add Module**. The Module is added to the slave.
6. For each slot, enter the address you want to use in the **Address** box.

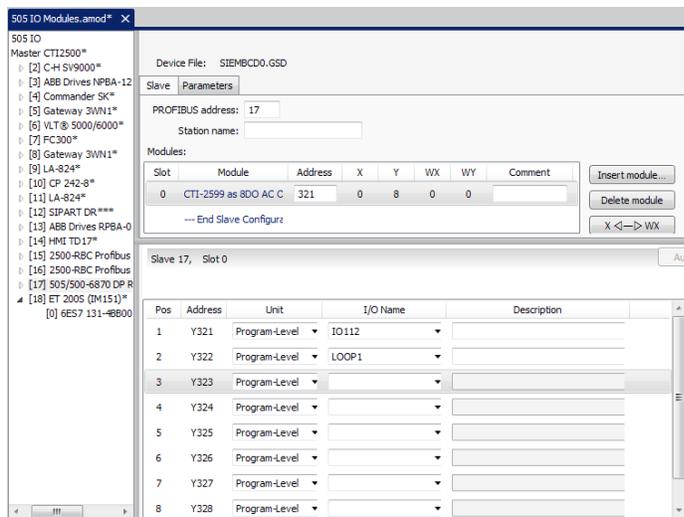


- Click **X <-> WX** to toggle the image register type for the selected module between discrete and word inputs. To ensure that discrete are toggled to words, the discrete must be on a 16-bit boundary.
 - Click **Y <-> WY** to toggle the image register type for the selected module between discrete and word outputs. To ensure that discrete are toggled to words, the discrete must be on a 16-bit boundary.
 - Click **Unify** to move all modules for the selected slave into the first module. Unification is only allowed if all modules are either discrete or words. Only the first address is retained.
 - Click **Compact** to move all discrete image register types to word image register types for every module of the selected slave. Then, all modules are unified into the first module. To ensure that discrete are toggled to words, the discrete must be on a 16-bit boundary. Only the first address is retained.
 - Click **Restore** to restore modules for the selected slave to their initial configuration, including the image register types. Zero addresses are assumed for all but the first address, which is kept.
 - If you have added a Smart Connect slave device, click **Smart connect** to configure it.
 - You can press **ALT+I** to search for I/O symbol names and **ALT+D** to delete the module.
7. Optional: Enter a comment in the **Comment** box.
 8. To configure parameters for the module, click the **Parameters** tab, and then enter the necessary information. Not all modules will have configurable parameters.



Defining I/O Points in Profibus I/O

Defining an I/O point consists of entering an I/O symbolic name and information for each address you want to access. To begin defining an I/O point go to the I/O Module editor and select a module to display the I/O Point Editor.



To define or edit an I/O point, select the appropriate point and then enter the following information.

- In the **Unit** box, select the unit you want to use, or select **Program-Level**.
- In the **I/O Name** box, select or enter the I/O symbolic name you want to use.
- In the **Description** box, type a description.

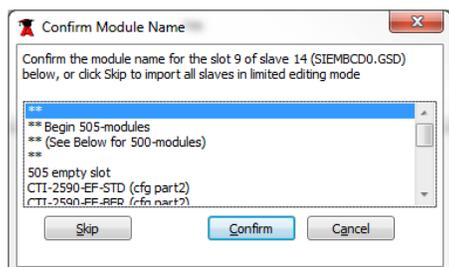
Save the I/O configuration, and the associated I/O points, by clicking **Save** on the **File** menu.

Merging a .2bf File

After importing a program with Profibus configuration, the Profibus I/O editor will be in limited editing mode. The Profibus configuration cannot be changed in this mode. To enable full editing, the original .2bf file must be merged after the required .gsd files are added to the hardware directory.

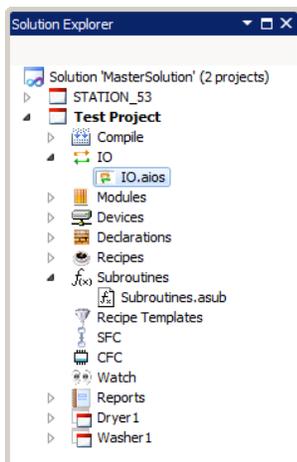
To merge a .2bf file:

1. Import an APT program.
2. Expand **%Modules** in the Solution Explorer and open the 505 IO Modules.amod file.
3. Select the PLC that is acting as the master module.
4. Click **Merge**.
5. During the merge, you may be asked to confirm the name of modules. Select the appropriate module in the Confirm Module Name dialog box.

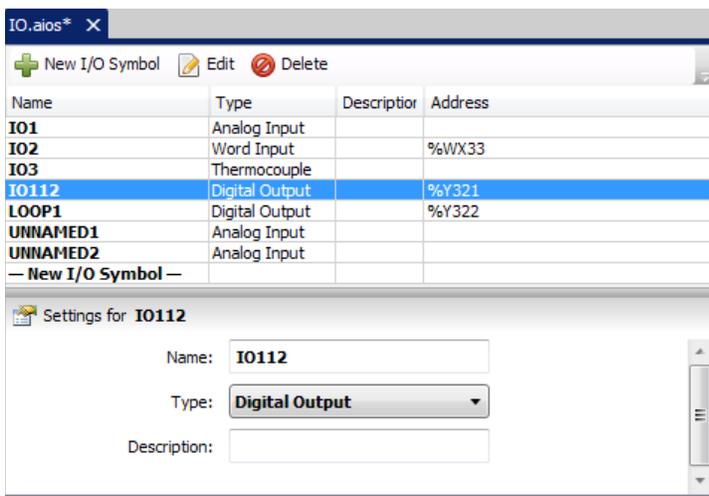


Viewing and Editing I/O Symbolic Names

You can use FTSolution for APT to view and edit the I/O symbolic names used in your FTSolution for APT project. When a project is displayed in the Solution Explorer, you can open the default IO table definition file (named IO.aios), by expanding **IO** and then opening **IO.aios**.



You can then add new I/O symbols, edit existing symbols, or delete symbols that are not needed.

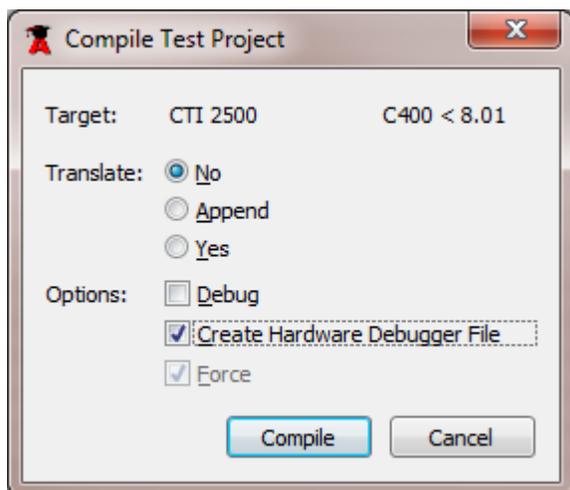


FTSolution for APT Hardware Debugger

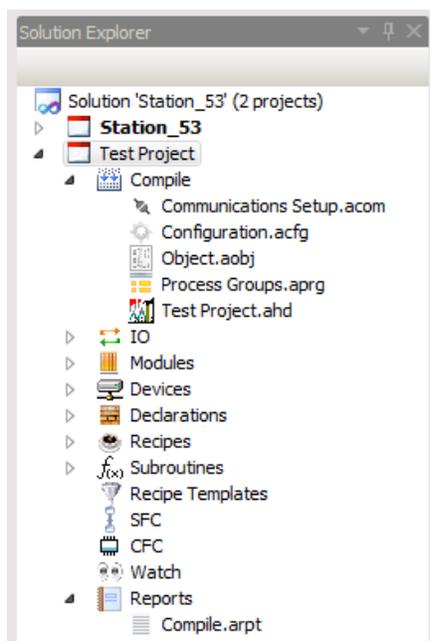
FTSolution for APT Hardware Debugger is a sophisticated PLC hardware debugger utilizing ladder language-based tools. FTSolution for APT Hardware Debugger allows users to download compiled FTSolution for APT programs into Siemens 505/CTI 2500 Series processors, utilize sophisticated ladder language-based debugging tools, and limit the ability to modify existing programs.

Creating an .AHD file in FTSolution for APT

To use FTSolution for APT Hardware Debugger you must create a Hardware Debugger file (.ahd file) from within FTSolution for APT. To create an .ahd file for use within Hardware Debugger you must compile your project. During a compile select the option **Create Hardware Debugger file**.

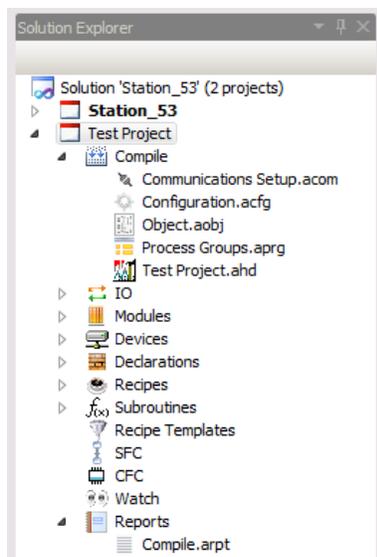


The .ahd file will automatically be placed within the Compile folder of the applicable Solution and Project. For example, the Test Project.ahd file picture below for **Test Project** is located within the folder `\Station_53 \Test Project\Compile`.



Opening an .AHD file in FTSolution for APT

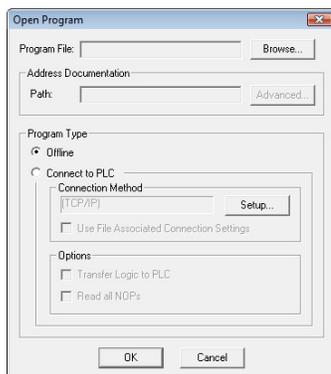
To open the .ahd file within FTSolution for APT open the **Compile** item within the Solution Explorer and double click the .ahd file. The Hardware Debugger will launch. In the example below the .ahd file is **Test Project.ahd**.



Opening an .AHD file in Hardware Debugger

You can also open the file within Hardware Debugger.

1. Select the **File/Open** menu click the  toolbar icon, or press [Ctrl-O]. The Open Program dialog appears.
2. Click **Browse** and select the *.AHD file.
3. Select the **Offline** radio button or enter information into **Connect to PLC**. Click **Ok**.



WARNING: DO NOT use Hardware Debugger to modify an FTSolution for APT-created .AHD file. An FTSolution for APT-created .AHD file contains unique elements which must be downloaded to a PLC to correctly run an APT language program. Saving an .AHD file with Hardware Debugger omits these critical changes.

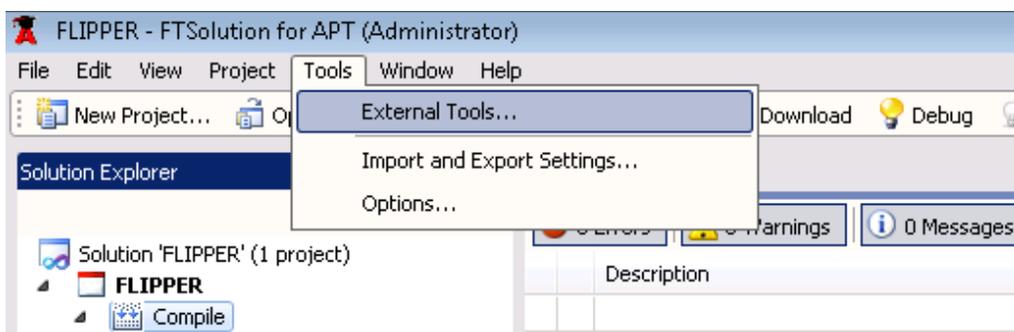
FTSolution for APT Simulator (505 Simulator)

FTSolution for APT Simulator (505 Simulator) is designed to run and test FTSolution for APT projects without the use of a PLC.

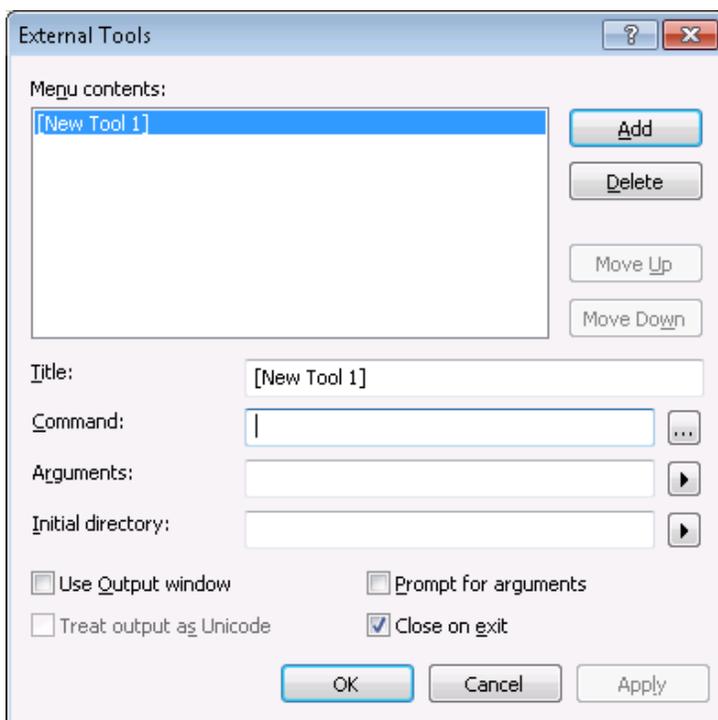
Setting Up 505 Simulator as an External Tool

505 Simulator can be started from inside FTSolution for APT by adding it to the list of External Tools. This is a one-time setup on each FTSolution for APT installation. To add 505 Simulator as an External Tool:

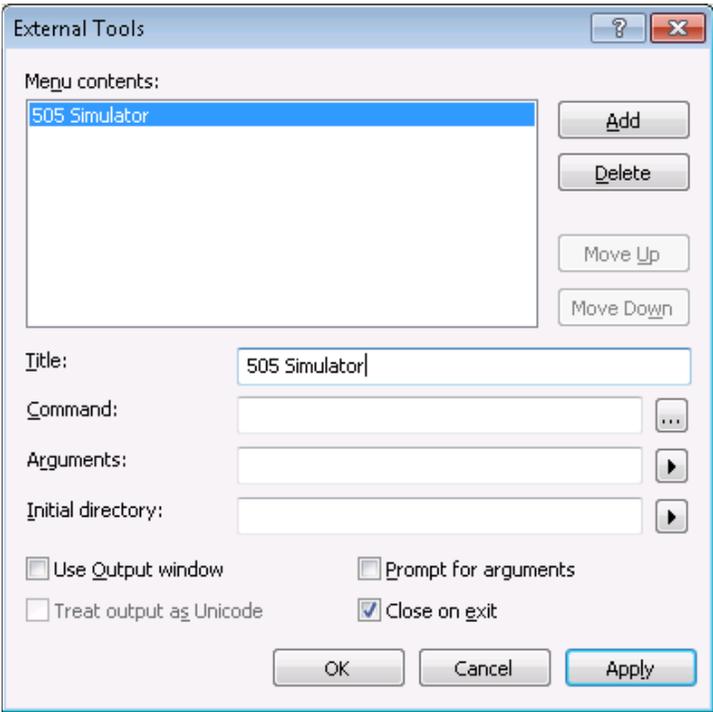
1. In FTSolution for APT, open the Tools menu and click on **External Tools...**



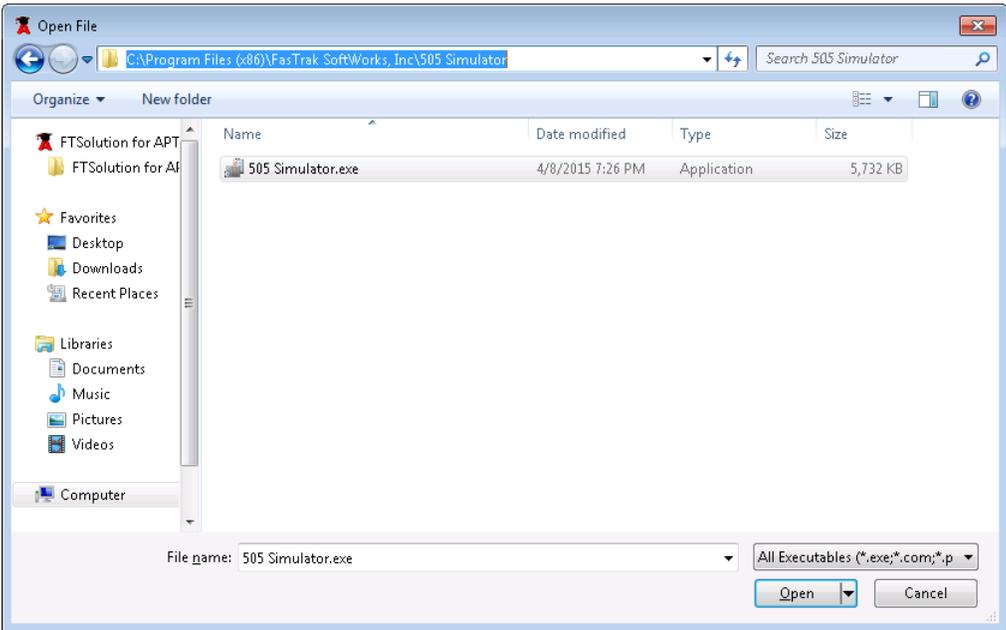
2. The External Tools window will be shown as follows with a default entry in the Title field.



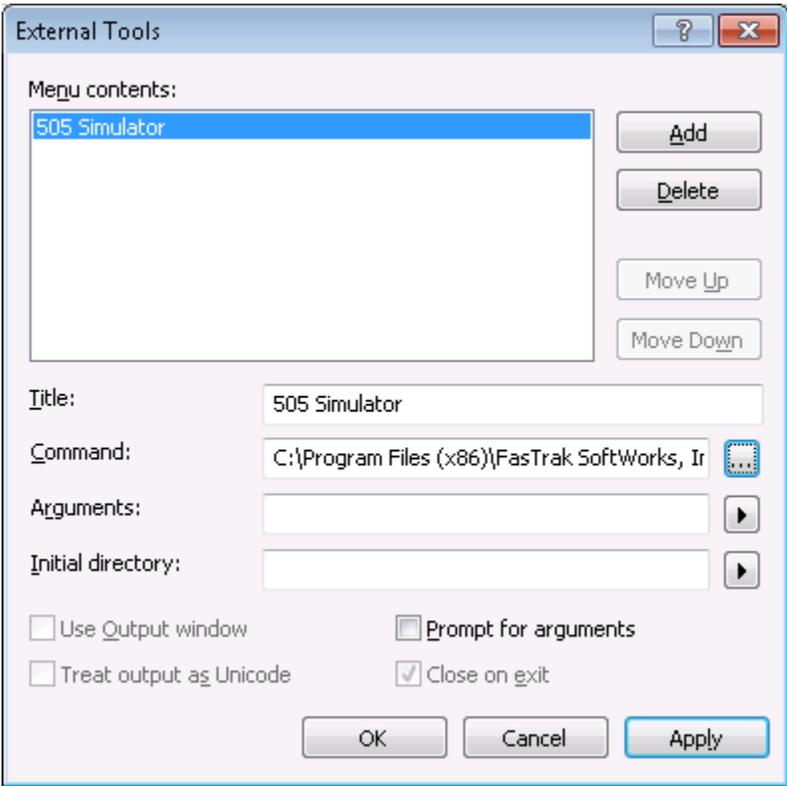
3. Change the Title to “505 Simulator”



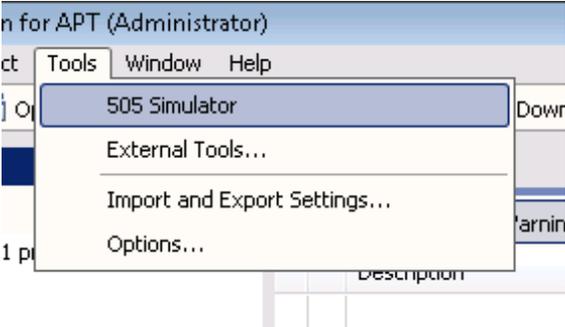
- 4. In the **Command** field, either type in the full file path to the 505 Simulator executable or click the browse button  to the right of the field to search for the executable. The 505 Simulator executable is typically located at C:\Program Files\FasTrak SoftWorks, Inc\505 Simulator.
- 5. If browse option is used, go to the installed location of the 505 Simulator then select the executable and click **Open**.



6. In the External Tools window, click **Apply** and then **OK** to close the window.

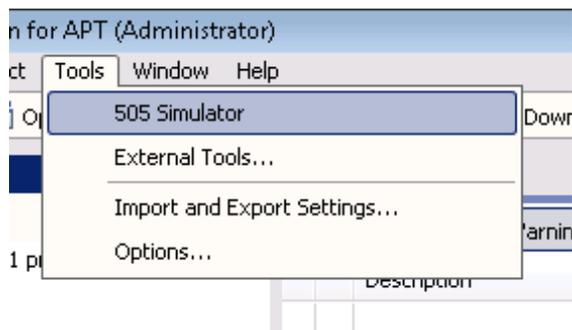


7. Open the **Tools** menu and click **505 Simulator** to launch the simulator from FTSolution for APT.

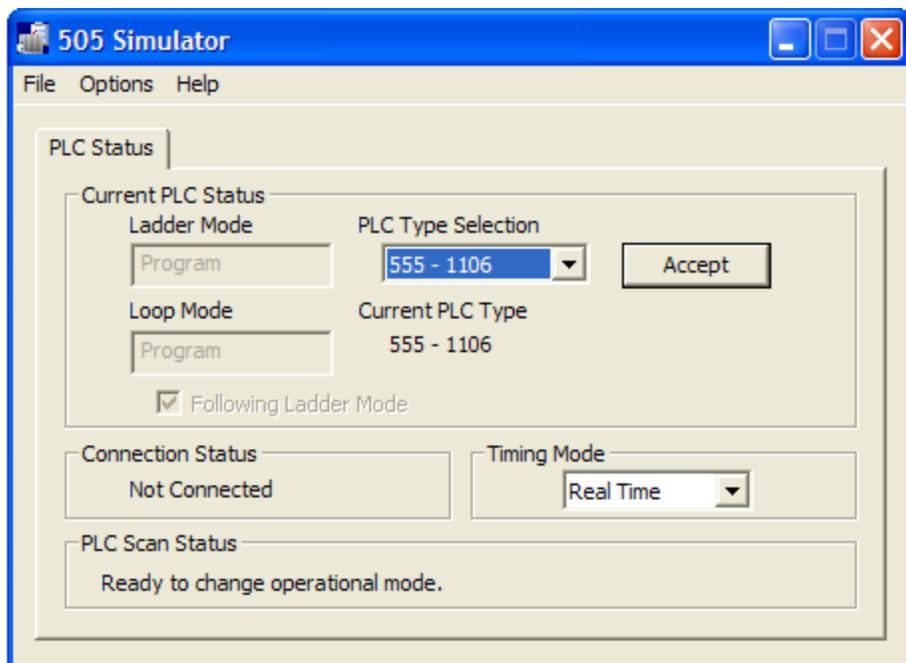


Starting 505 Simulator

To start the 505 Simulator click the **Start** button, click **All Programs**, click **FasTrak SoftWorks**, and then click **505 Simulator**. Alternatively, 505 Simulator can be started from inside FTSolution for APT (see Setting up 505 Simulator as an External Tool). To start 505 Simulator from FTSolution for APT, go to the Tools menu and click **505 Simulator**.

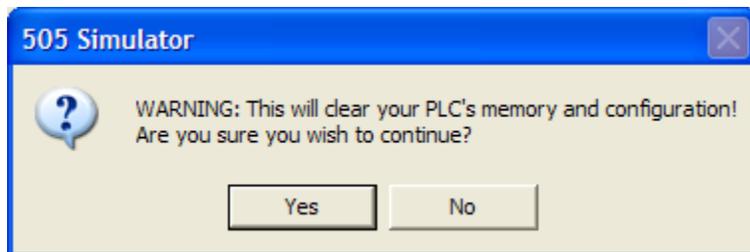


After starting, the 505 Simulator Window is displayed:



Selecting PLC Type

1. In the **PLC Type** Selection box, select the PLC type you want to simulate.
2. Click **Accept**. The following warning appears:

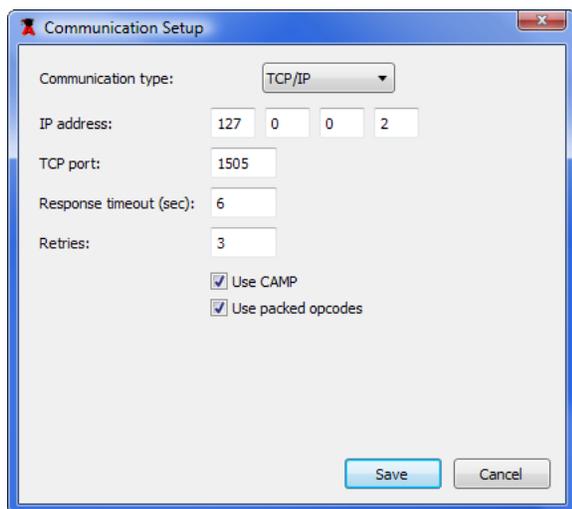


3. Click **Yes**. The simulator is now simulating the PLC type you selected.

Setting Up Communications with 505 Simulator in FTSolution for APT

505 Simulator uses the TCP/IP communication method to communicate with FTSolution for APT. Do the following to create a new TCP/IP connection in FTSolution for APT to connect to the 505 Simulator:

In FTSolution for APT, open the project and click **Communications Setup.acom** from the Solution Explorer. The Communication Setup dialog box appears. Select **TCP/IP** from the **Communication Type** list. The TCP/IP setup information appears.



In the **IP Address** box, enter **127.0.0.1**. In the **TCP port** box, enter **1505** in the TCP Port box. The TCP Port must be set to 1505 to enable connection with the simulator. The other communication settings can be changed as needed; refer to the FTSolution for APT online help for additional information.

Click **Save**.

Connecting to 505 Simulator

Prerequisites

In order to connect to the 505 Simulator, both FTSolution for APT and 505 Simulator must be currently running. In addition, within FTSolution for APT, a TCP/IP communication method that uses an IP address of 127.0.0.1 must be created.

Connecting

Once these prerequisites have been met, connecting to the simulator is identical to going online with an actual PLC. See *Starting Debug Mode* on page 53 for additional information on how to enter run mode.

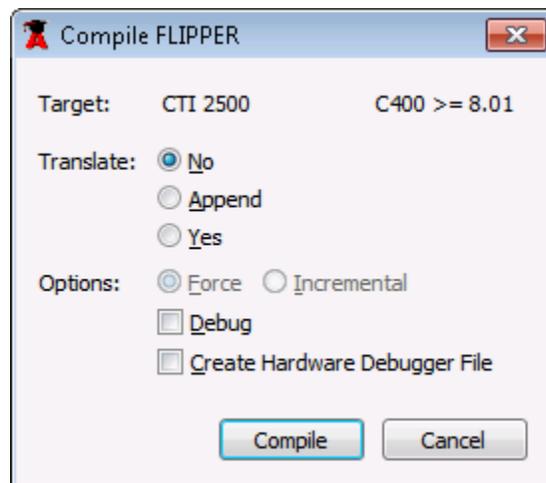
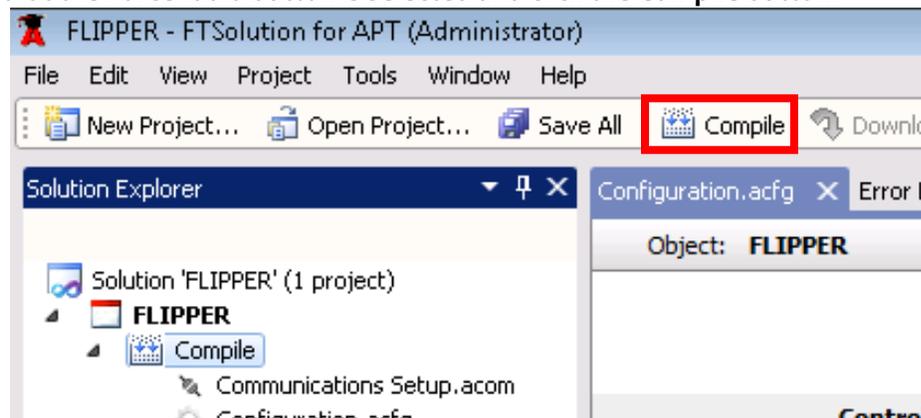
Note: 505 Simulator must be running before attempting to connect to it.

Performing an Incremental Download to 505 Simulator in Run Mode

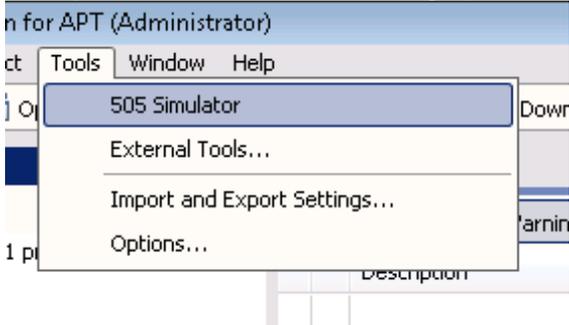
Users who wish to perform an incremental compile may benefit from attempting the incremental compile using the 505 Simulator. The 505 Simulator can be used to determine if a program can be incrementally downloaded to a running PLC without the need to reload and restart the PLC. This can avoid costly and unwanted production stoppages.

To attempt an incremental download to 505 Simulator in run mode:

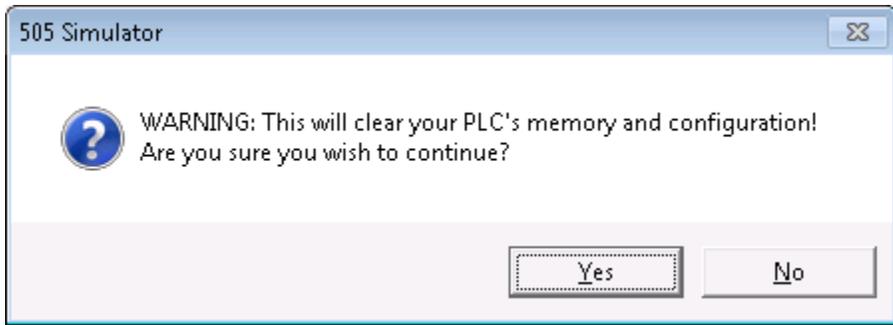
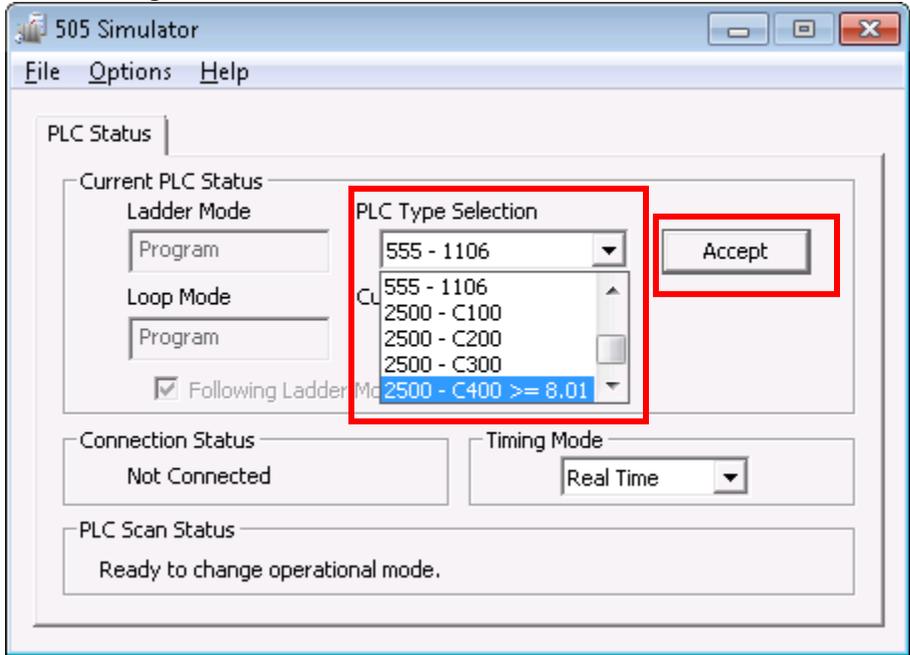
1. Begin the process by force compiling your project. Click the **Compile** option from the toolbar with the project node or the compile node in the project. In the compile options window, ensure that the Force radio button is selected and click the **Compile** button.



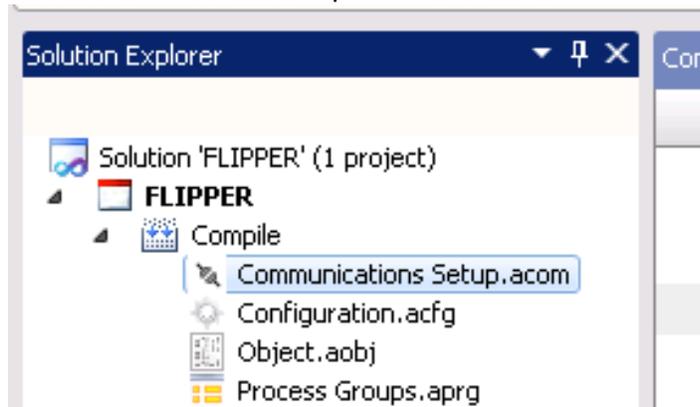
- 2. Launch simulator by opening the **Tools** menu and selecting **505 Simulator**
Note: If this option is not shown see *Setting Up 505 Simulator as an External Tool*



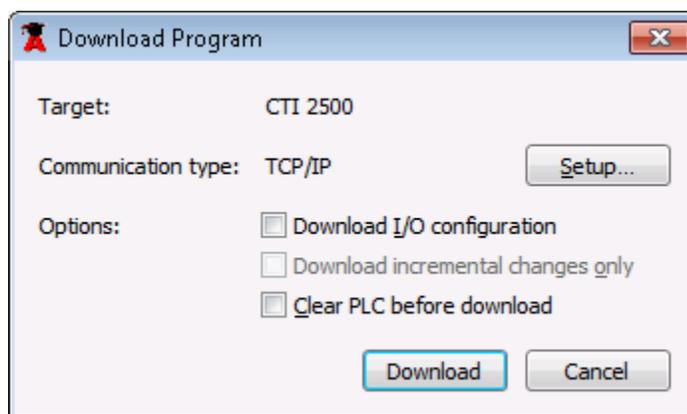
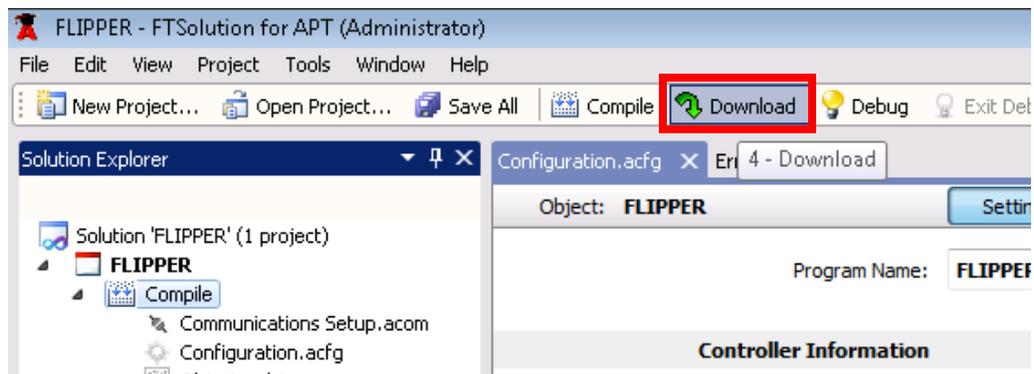
- 3. In the 505 Simulator window, change the PLC Type Selection to match type set in the FTSolution for APT project and then click the **Accept** button. Click **Yes** in the Warning dialog to confirm the change.



- In FTSolution for APT, open the Communications Setup. Change the Communication Type to TCP/IP and set the IP Address to 127.0.0.1. Click Save.

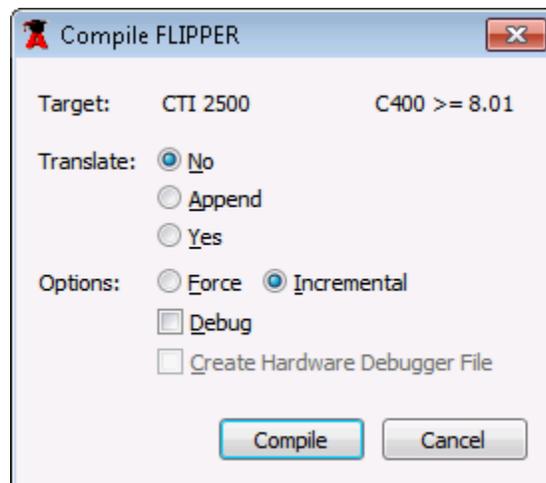
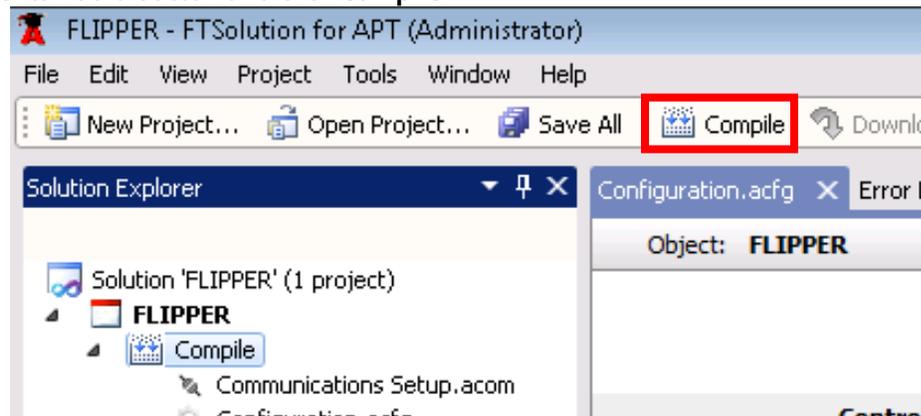


- Select **Download** from the toolbar with the project node or the compile node in the project. Click **Download** in the Download Program options window.



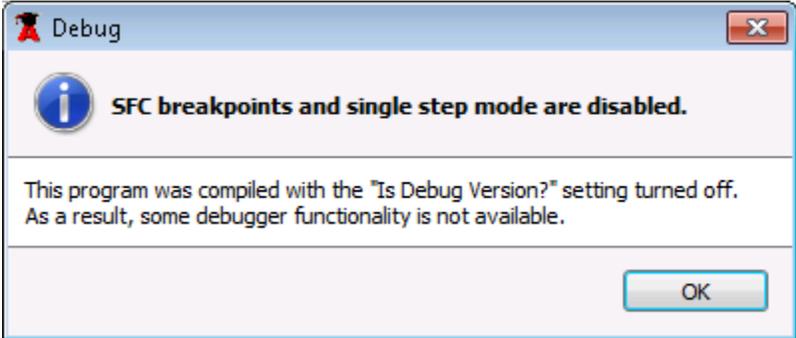
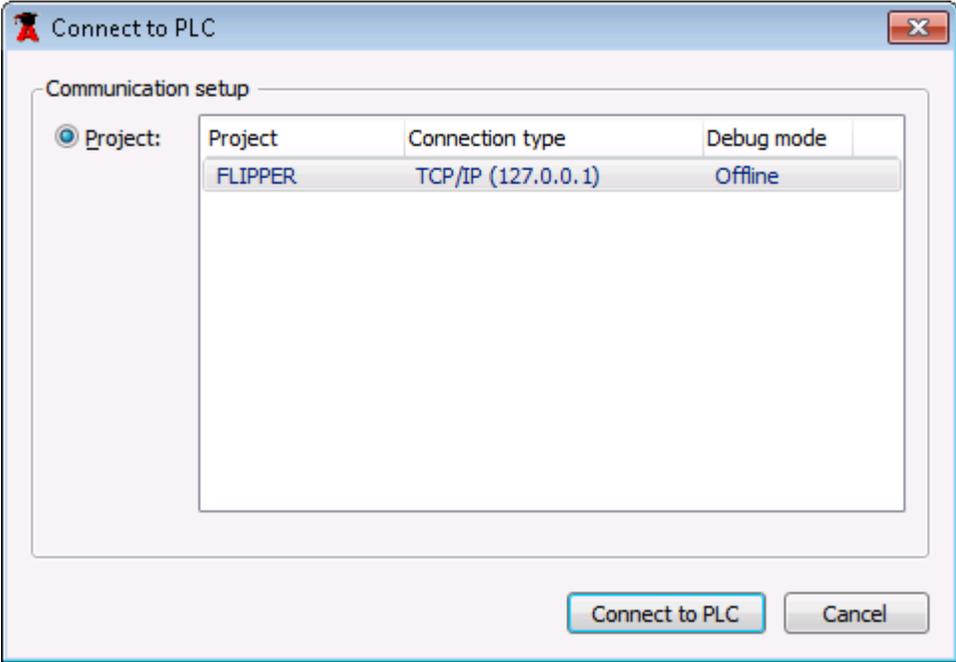
- Make the desired changes to your program.

7. After you have made your program changes, click the **Compile** from the toolbar with the project node or the compile node in the project. In the compile options window, select the Incremental radio button and click **Compile**.

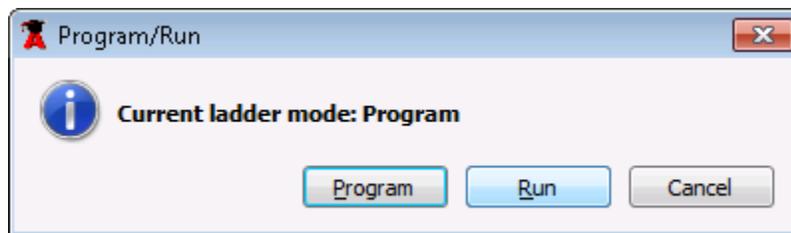
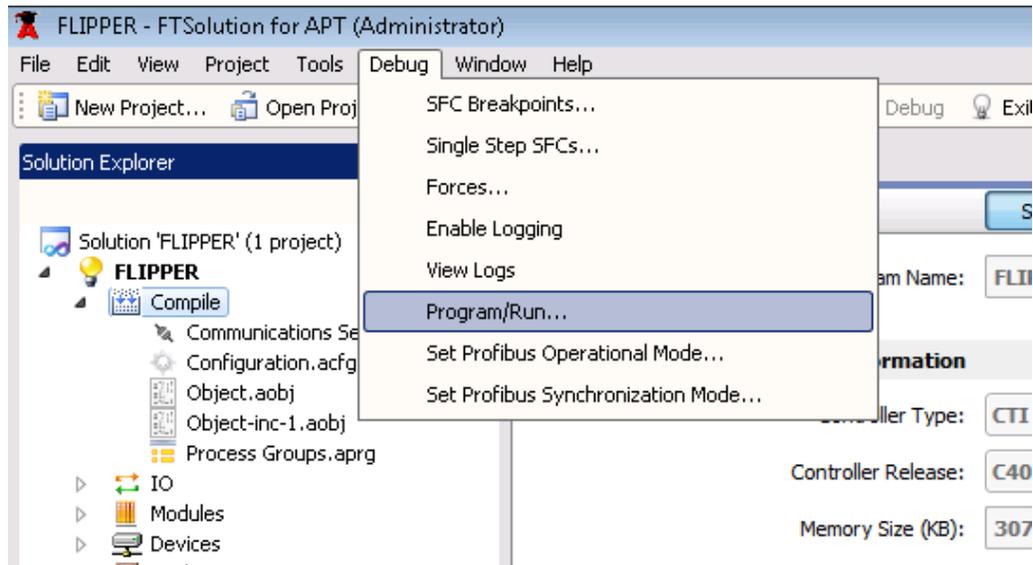


- 8. Click **Debug** from the toolbar. In the Connect to PLC window, click Connect to PLC button. A Debug prompt will be shown. Click **OK** to continue.

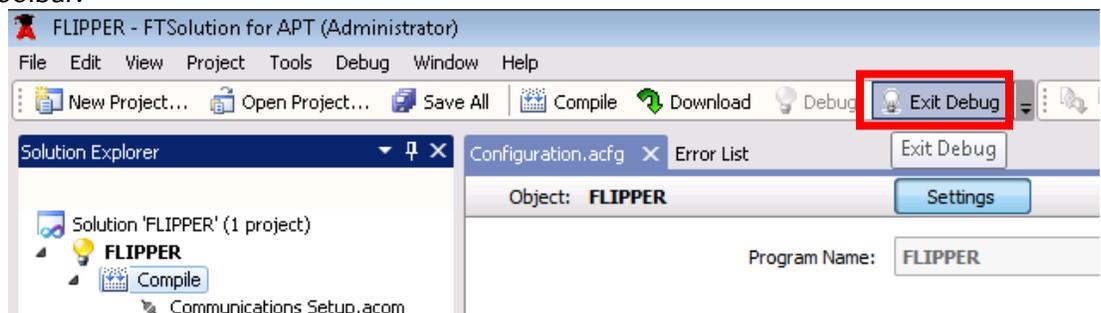
Note: The Debug prompt is shown because the Debug option was not selected in the compile options. This is ok because full debug functionality is not necessary to put the simulator into run mode.



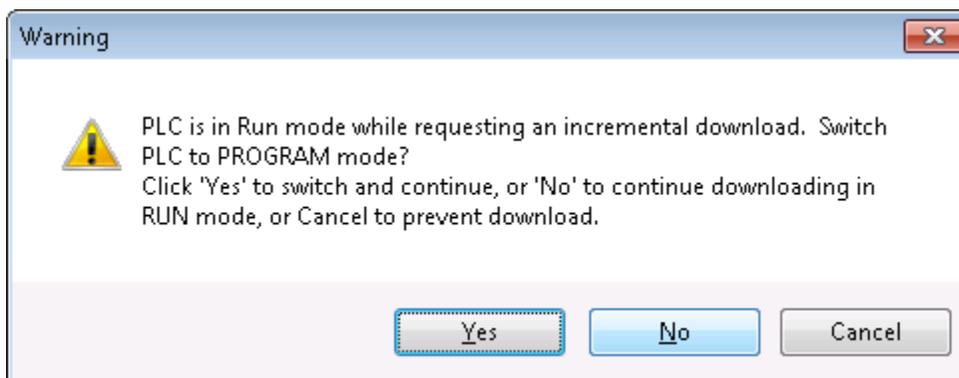
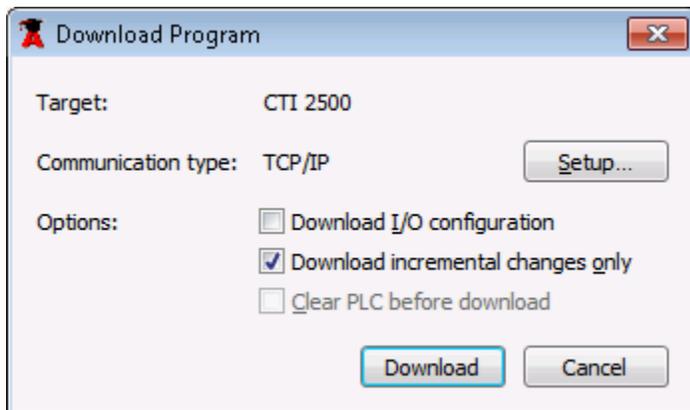
- Open the **Debug** menu and select **Program/Run...**. In the Program/Run window, click the **Run** button.



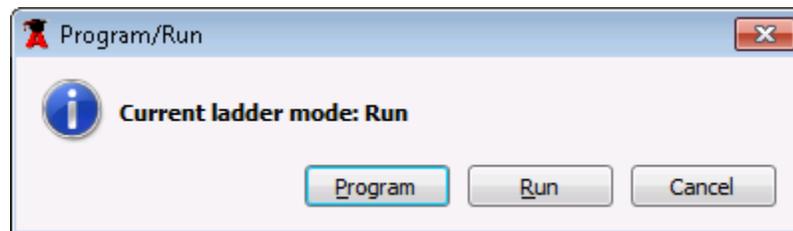
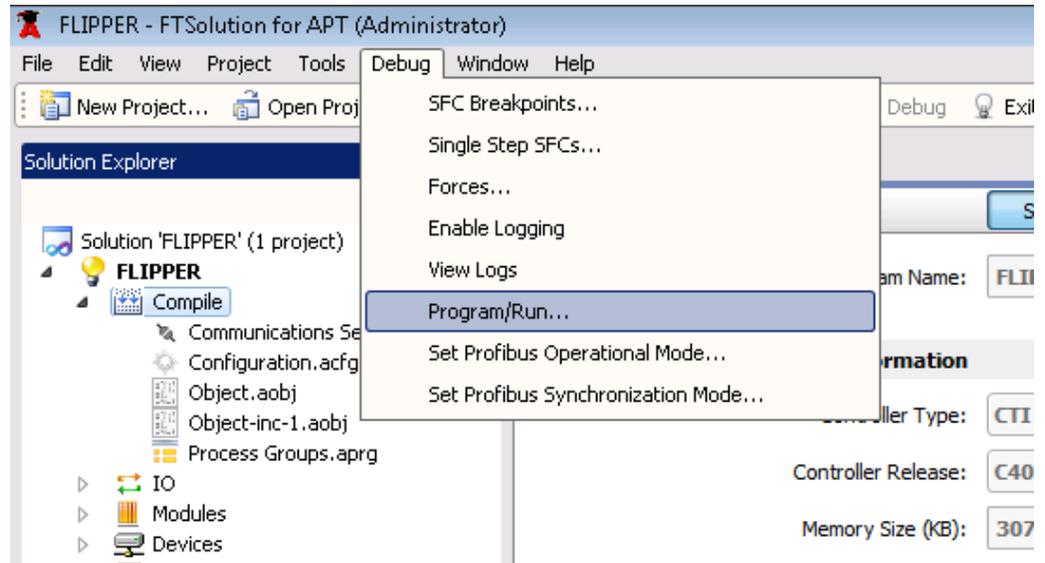
- A Download cannot proceed while the project is in debug mode. Click the **Exit Debug** button in the toolbar.



11. Select **Download** from the toolbar with the project node or the compile node in the project to download incrementally. In the Download Program window, ensure that **Download incremental changes only** is checked and click **Download**. In the Warning prompt, click **No** to continue downloading to the PLC in Run mode



- At the end of this process, it is recommended that you confirm the Simulator is still in Run mode. To do so, click **Debug** from the toolbar. In the Connect to PLC window, click **Connect to PLC button**. In the Debug prompt click **OK** to continue. Open the **Debug** menu and select **Program/Run....** Observe in the Program/Run window the Current ladder mode status is Run.

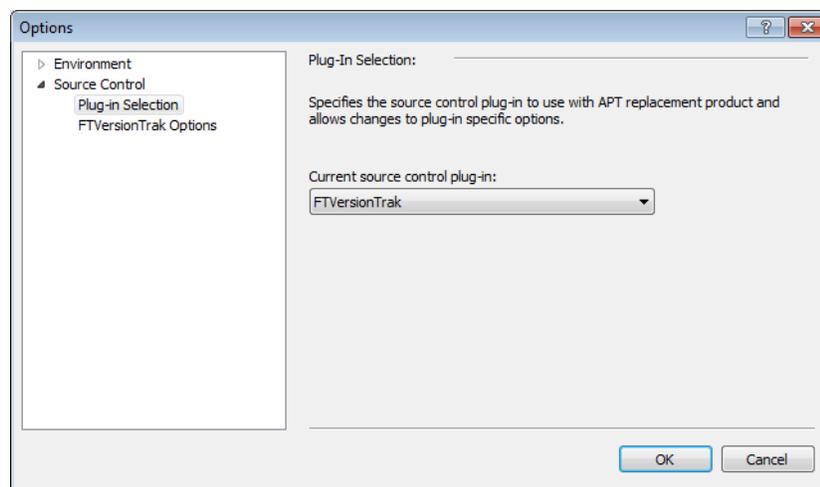


FTVersionTrak for APT

FTSolution for APT provides for integration with source control, or version control, software through the use of FTVersionTrak for APT. Contact your FasTrak SoftWorks, Inc. Sales Representative at **262-238-8088** or **sales@fast-soft.com** to purchase FTVersionTrak for APT. To use the functionality provided by FTVersionTrak, make sure that FTVersionTrak is listed as the designated source control plug-in for FTSolution for APT.

To specify the source control plug-in:

1. Start FTSolution for APT.
2. Click **Options** on the **Tools** menu.
3. In the Options dialog box, select **Source Control**.
4. Select **FTVersionTrak** as the **Current source control plug-in** and click **OK**.



FTVersionTrak Toolbar

Most of the source control-related operations you will need to perform in FTSolution for APT can be accomplished using the FTVersionTrak toolbar. If the toolbar is not displayed, click the **View** menu, point to **Toolbars**, and select **FTVersionTrak Toolbar**.



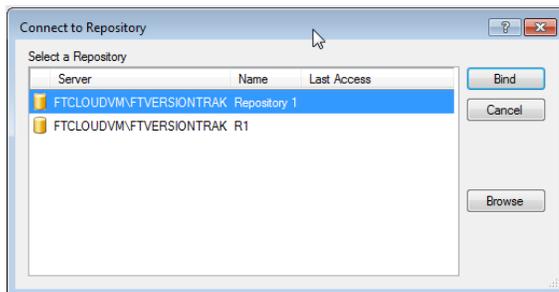
- 
Get Latest Version. Use this button to retrieve the latest repository version of the selected items in the Solution Explorer.
- 
Get. Use this button to retrieve a user-specified repository version of the selected item in the Solution Explorer.
- 
View Repository Properties. Use this button to view the repository properties of the selected item in the Solution Explorer.
- 
Check Out. Use this button to check out the selected items in the Solution Explorer.
- 
Undo Check Out. Use this button to undo a check out.
- 
Check In. Use this button to check in the selected items in the Solution Explorer.
- 
History. Use this button to view the history of the selected items in the Solution Explorer.
- 
Compare. Use this button to compare the item currently selected in the Solution Explorer.
- 
Refresh. Use this button to refresh the display.
- 
Disconnect. Use this button to disconnect from the repository.
- 
Launch FTVersionTrak. Use this button to launch FTVersionTrak.
- 
Bind. Use this button to bind an FTSolution for APT project to a repository.

Binding to a Repository

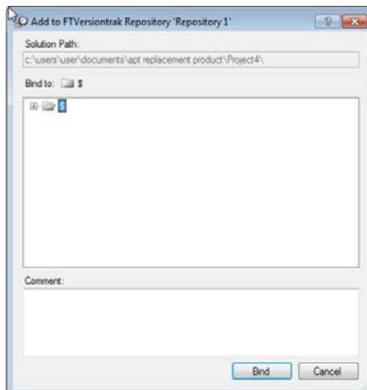
The first step in using FTVersionTrak with FTSolution for APT is to bind an APT project to an FTVersionTrak repository. Binding a project creates a link between the working files of a solution and a repository in the FTVersionTrak database. Instructions for creating a repository were provided earlier in this document, in the procedure for installing FTVersionTrak.

To bind an FTSolution for APT project to a repository:

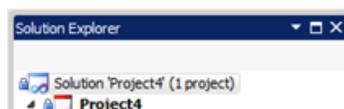
1. Open the solution you want to bind.
2. Click the **Bind** toolbar button on the FTVersionTrak toolbar.
3. In the Connect to Repository dialog box, select a repository and click **Bind**. You will bind the FTSolution for APT project to this repository.



4. In the Add to FTVersionTrak Repository dialog box, click **Bind** to bind the FTSolution for APT project to the root folder of the repository. If you want, you have the ability to bind the project to another folder on the repository.



Additional icons are displayed in the Solution Explorer of an APT replacement project that has been successfully bound to a repository. These icons are used to indicate the status of the files under FTVersionTrak control. The following figure shows icons that represent the checked-in and checked-out status of the project, solution, and files.



 **Checked-in.**

 **Checked-out.**

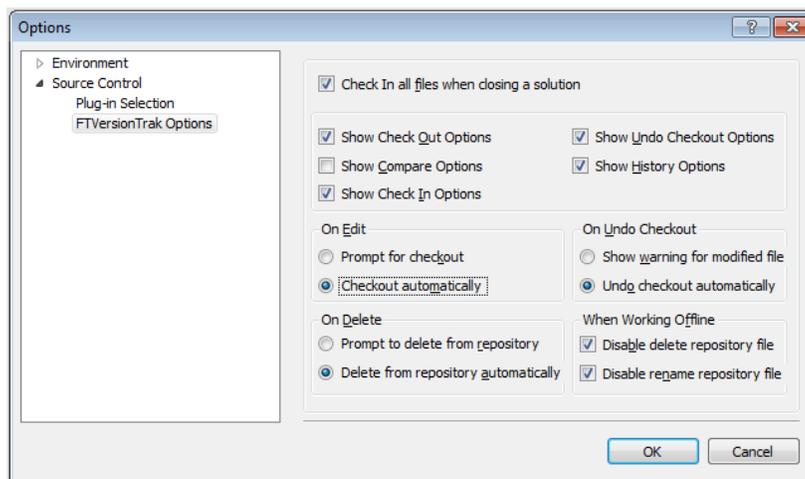
Additional icons are used to represent other statuses.

Setting FTVersionTrak Options

There are many customizable settings that determine how FTVersionTrak operates within FTSolution for APT. Examples of these settings include whether check-ins and check-outs are performed automatically, what occurs when files are deleted, and what restrictions are in place when working offline.

To specify FTVersionTrak options:

1. Click **Options** on the **Tools** menu in FTSolution for APT.
2. In the Options dialog box, in the left pane, expand **Source Control**.
3. Select **FTVersionTrak Options**.



4. Select the options you want to use.

Using Solution Explorer with FTVersionTrak

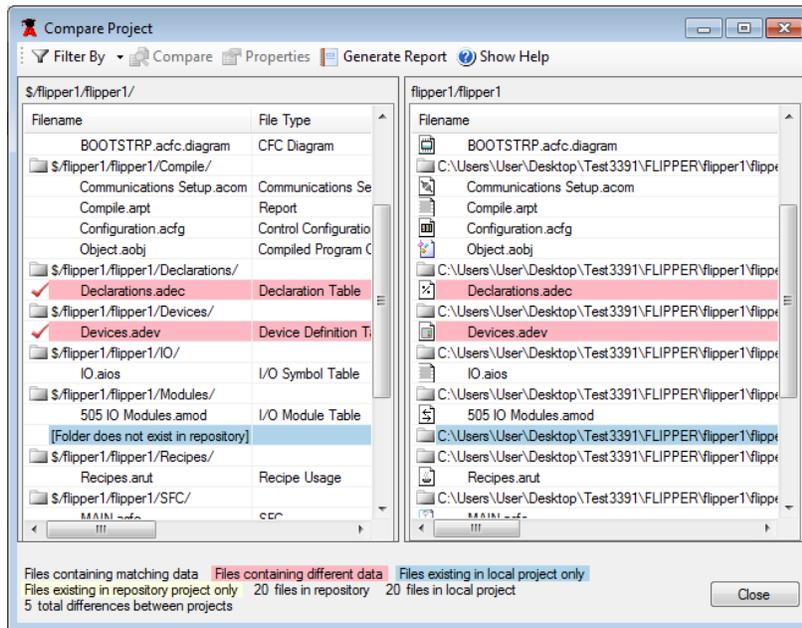
The operation of many FTVersionTrak operations will vary depending on what is selected in the Solution Explorer. When a project, solution, or unit is selected in the Solution Explorer, the history and compare commands will show information about the entire project, solution, or unit. When an individual file is selected, the history and compare commands will show detailed information about that specific file.

Similarly, the Check-In and Check-Out commands will also work on a project, solution, object, and unit level. For example, when a solution is selected, all of the files contained in that solution will be checked in or checked out at once. If only a single file is selected in Solution Explorer, only that file will be checked in or out.

Example of Compare: Comparing at the Project Level

If a project or solution is selected in the Solution Explorer, clicking the **Compare** toolbar button provides a way to compare the current version of a project or solution with the most recent version in the repository.

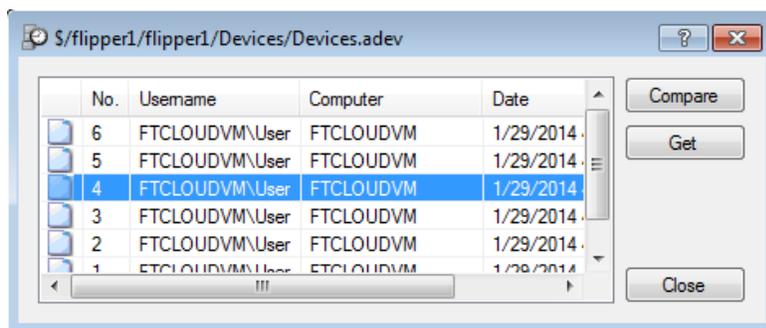
The following figure shows the result of a compare project done on a project named **flipper1**. In this comparison, the **Declarations.adec** and **Devices.adev** files do not match the latest repository versions.



Example of History: Historical Information for a Single File

When a single file is selected in the Solution Explorer, such as **Devices.adev**, clicking the **History** toolbar button provides detailed historical information for that file.

The following illustration shows the historical information for the **Devices.adev** file.

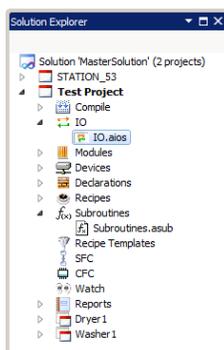


From this dialog box, it is possible to compare versions of **Devices.adev**. For instance, selecting **No. 4** and then clicking **Compare** will compare the current version with the **No. 4** version in the repository. It is also possible to perform a get operation from this dialog box. Selecting **No. 4** and then clicking **Get** will retrieve the **No. 4** version from the repository and place it in the working folder.

PCS Tagging – Marking Objects for Translation

FTSolution for APT supports the ability to mark simple objects (such as declarations) as well as complex objects (such as CFBs, recipes, or declarations with more than one extension). When marking a complex object, you are able to edit the object and mark the individual extensions that require translation. You are also able to specify an object's attributes and specify the process group to which a tag belongs.

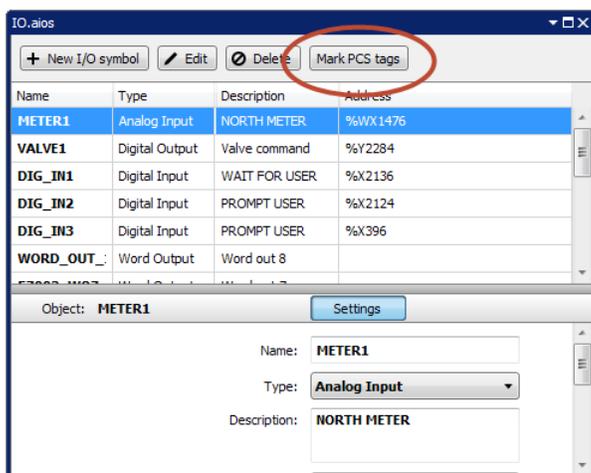
To mark an object for translation, select the item in Solution Explorer and then type the letter **K**. In the following example illustration, the file **IO.aios** is selected in Solution Explorer.



The following items may contain objects that can be marked for translation:

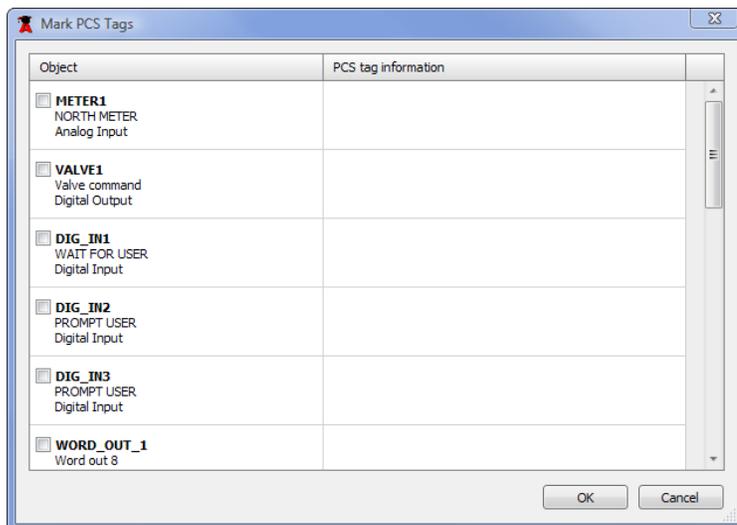
- .aios files (**IO.aios** is the default)
- .adev files (**Devices.adev** is the default)
- .adec files (**Declarations.adec** is the default)
- .arut files (**Recipes.arut** is the default)
- .acfc files (only CFCs files that certain CFC blocks can be marked for translation)

You can also click the **Mark PCS tags** button to mark an object for translation. The **Mark PCS tags** button appears on several table file definition and configuration screens, as well as the graphical editor for CFCs. The following illustration shows the **Mark PCS tags** button in the **IO.aios** configuration screen.

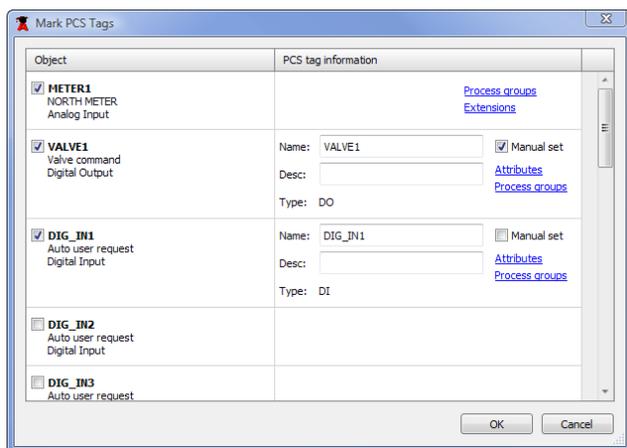


Note: To mark an item that is part of a unit, open the unit in Solution Explorer, select the item, and then type the letter **K**. Or, click the **Mark PCS tags** button within the appropriate Unit configuration screen.

After pressing the letter **K** (or clicking the **Mark PCS tags button**), the **Mark PCS Tags** dialog box is displayed. In this dialog box, select the check boxes of the objects you wish to mark for translation. The **Mark PCS Tags** dialog box is equivalent to the Tag Translate editor and sub-editor in APT.

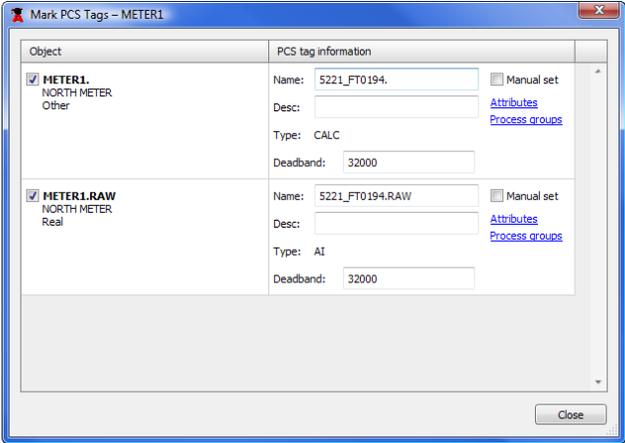


After selecting the check box of an object, you can enter additional information in the **PCS tag information** column. The information you are able to enter will vary depending on the type of object selected. For example, the following illustration shows the **Mark PCS Tags** dialog box for objects in the **IO.aios** file.

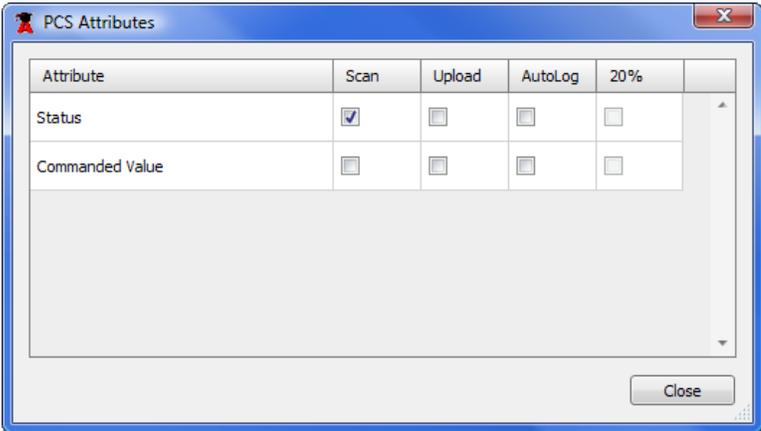


In the **Mark PCS tags** dialog box, the objects named **METER1**, **VALVE1**, and **DIG_IN1** are all marked for translation. For the **VALVE1** and **DIG_IN1** objects, you can type a new name for the PCS tag in the **Name** box and a description for the PCS tag in the **Desc** box.

The first object, **METER1**, is a complex object. Click the **Extensions** link to mark the individual extensions associated with the complex object. The following illustration shows the extensions associated with the **METER1** object.



To specify the attributes of an object or an object’s extension, click the **Attributes** link. After clicking the **Attributes** link, select the check boxes of the attributes you want to use in the **PCS Attributes** dialog box. The PCS Attributes dialog box is equivalent to the tag attributes sub-editor in APT.



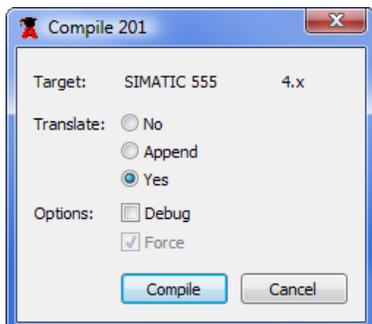
The attributes you can specify consist of the following:

- Scan
- Upload
- Autolog
- 20% offset

By default, certain attributes may already be selected, depending on the type of object.

Tag Translation and Compile Options

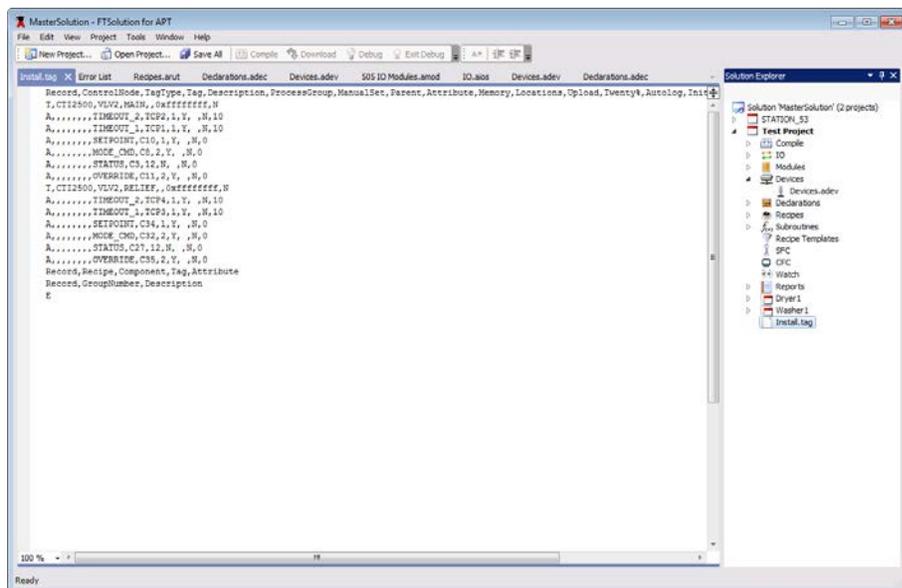
The compile options under **Translate** in the Compile dialog box have a direct effect on PCS tagging. Under **Translate**, specify the translate option you want to use:



- When you select **No**, existing marked tags are not modified.
- When you select **Append**, existing marked tags are placed in the *install.tag* file. Tags marked since the last translate build and tags that were unmarked and re-marked are also placed in a separate file, *append.tag*.
- When you select **Yes**, all marked tags are placed in the *install.tag* file.

Location of Install.tag and Append.tag Reports

Once created, the *install.tag* and *append.tag* files appear in the Solution Explorer within FTSolution for APT. The *install.tag* file is displayed in the following illustration.



Using File Explorer, the *install.tag* and *append.tag* files are also accessible from the Reports folder within the project's folder.