

## Release Notes

---

### SIMATIC APT

APT Software Release 1.9A

---

Siemens Energy & Automation, Inc., is pleased to present the tenth major release of its computer-aided software design package, the SIMATIC® Application Productivity Tool—APT™, Release 1.9A. This release of APT enhances the APT Release 1.9 software package, offering new features and improving existing ones. A summary of the major enhancements is listed in these *Release Notes*.

Documentation for APT Release 1.9A consists of the tenth editions of the *SIMATIC APT User Manual* and *SIMATIC APT Programming Reference Manuals*, and the eighth editions of the *SIMATIC APT MAITT User Manual* and the *SIMATIC APT Applications Manual*. APT Release 1.9A ships with manuals exclusively on CD-ROM. Printed manuals can be ordered separately. The order number for the complete printed manual set is APT-8201-T; order numbers for individual manuals appear on the title pages of the corresponding CD-ROM files.

These *Release Notes* list general guidelines and software characteristics that are not described in the manuals. Technical Advisories TA #1-15, 17-21, 23-30, 32-48, and 50-62 are superseded by these *Release Notes*. TA #16 is hardware-related, and the problem was corrected for RCC Rel. 4.52 and Rel. 3.52. TAs #22, #31, and #49 still apply to APT Release 1.9A.

#### Read Me File

Information that became available after publication of the manuals and these *Release Notes* can be found in the **README.TXT** file on the APT software CD-ROM. If you are upgrading your system from APT Rel. 1.9 Modification 1a or earlier, refer also to the “Updating Pre-Release 1.9A Programs” section in these *Release Notes*.

#### Software Installation

Refer to Appendix A of the *SIMATIC APT User Manual* for APT 1.9A software installation instructions. Appendix B contains information on installing APT on an Industrial Ethernet (H1) network. Appendix C discusses using APT with DOS, Appendix D discusses using APT with Windows® 95, and Appendix E discusses using APT with Windows NT® and Windows 2000®. Refer to Chapter 9 for configuring and using the PROFIBUS-DP network with APT.

#### For Help in Installation

If you need assistance, call the Technical Services Group in the U.S. at (800) 333-7421, or outside the U.S. at +49 911 895 7000.

**2807402-0005**

publication date: 04/01

## Summary of Major Enhancements for APT Release 1.9A

---

<b>Windows NT and Windows 2000</b>	In addition to MS-DOS and Windows 95, the Windows NT and Windows 2000 operating systems are now supported. Computer to P/C communications for Windows NT and 2000 include Industrial Ethernet (H1), TIWay via the Host and UniLink adapters, and serial communication via direct connect.
<b>CP1413 and CP1613</b>	For Windows NT and Windows 2000 platforms, the CP1613 communications card is now used for Industrial Ethernet (H1) communications. The CP1413 communication card is still supported and required for H1 communications for MS-DOS and Windows 95 platforms.
<b>S5 Download</b>	The download functionality for S5 controllers has been improved by adding a checksum for Functions Blocks (FBs). This provides faster and more efficient incremental download capability.
<b>Development Tools</b>	The FSL, PDL, and PBL executables are included on the installation media. These tools facilitate the creation of custom objects and devices by knowledgeable users who are familiar with the internal architecture of the APT product.
<b>Binary Format</b>	Debug charts now support the display of words in binary format. In addition, the words may be modified in binary, hexadecimal, or integer format.
<b>Problems Fixed</b>	Several reported software problems with previous product releases have been corrected. Specifically, the issues described by Technical Advisories 059, 060, 061, and 062 have been addressed. Communication issues reported with the use of the TIWay Host and UniLink Adapters have been resolved.
<b>CD-ROM Media</b>	The APT manuals and software now ship exclusively on CD-ROM. If you require diskettes for software installation, the contents of the software CD is organized to permit generation of diskettes in either 1.2 MB or 1.44 MB format. Instructions for doing this are included in the "Diskettes" section of these <i>Release Notes</i> . The APT manuals continue to be provided on a separate CD. The manuals are best viewed using Adobe Acrobat Reader, Version 4.0 or later. If you do not have at least this version installed on your system, you can install it (on systems running Windows 95 or later) from the manuals CD.

---

## Creating Diskettes

If you require 3.5 or 5.25-inch (1.44 MB or 1.2 MB) diskettes for installation, you can copy the CD-ROM folders to diskettes. You can create diskettes either from the DOS Command Prompt or from Windows Explorer.

**DOS Command Prompt:** To generate diskettes using the DOS Command Prompt, enter the following command:

```
XCOPY <CD-ROM drive>:\DISK1 <Floppy drive>:\ /S/E/V
```

Repeat this command for each of the CD-ROM folders DISK2, DISK3, ... DISK8.

**Windows Explorer:** To generate diskettes using Windows Explorer, follow these steps:

1. Double-click on your CD-ROM drive.
2. Click on the CD-ROM's folder **DISK1**.
3. Click **Edit->Select All** on the menu bar.
4. Drag the highlighted contents from the right-hand side of the display) to your Floppy (diskette) drive.
5. Repeat steps 2, 3, and 4 for each of the folders DISK2, DISK3, ... DISK8.

For both methods, be sure to label each diskette with its DISK<sub>n</sub> folder name immediately after it is created.

## Updating Pre-Release 1.9A Programs

---

### Archive and Restore

Archive any existing programs before installing APT Rel. 1.9A on your system. All pre-Rel. 1.9A programs must be restored and recompiled under Rel. 1.9A before you can download and debug them.

---

**NOTE:** To minimize the risk of losing your APT program in the event of system or media failure (malfunctioning hard disk, diskettes with bad sectors, etc.), always create a backup that you can use to recover your program. You can back up your program by any of these methods.

- Execute an APT archival and place the archive on another hard disk or on diskettes that can be stored in a safe place. You can verify that the archival is successful by restoring the archive to another drive or system.
  - Do a backup of the entire **\APT** subdirectory.
  - Use the MS-DOS® **BACKUP** command or a commercially available backup utility to back up the **\APT\PROGRAM** and **\APT\DATABASE** subdirectories.
- 

APT Rel. 1.9A cannot restore programs earlier than Rel. 1.3a. You must first restore a Rel. 1.1 or a Rel. 1.2d archive to either Rel. 1.3a or Rel. 1.4a, then archive the Rel. 1.3a or Rel. 1.4a program, and finally use Rel. 1.9A to restore the archive. The Rel. 1.9A DOS utility **APTREST** can display the information of an APT archive, including the version of APT that was used to archive the program. Refer to Appendix H, “DOS Utilities,” of the *SIMATIC APT User Manual* for information on **APTREST**.

### Marked Tags

To ensure correct and current tag translation information, all tags that are marked for Tag Translation must be unmarked and then remarked.

### Compile

After restoring a pre-Rel. 1.9A program under Rel. 1.9A, you must compile the program. You do not have to select the **Force Compile** option. APT automatically does a forced compile of the program.

### Recipe Extensions

Starting with APT Rel. 1.4a, APT creates the following extensions, which are used with recipe code: **.INUSE**, **.STATUS**, **.DRDY**, **.RTU**, **.DSTBL**.; and with Rel. 1.6: **.SCALE**, **.RECREQ**, and **.HOLDREQ**. The function of these extensions is described in the “Recipes” chapter of the *SIMATIC APT Programming Reference (Tables) Manual*.

If you are restoring a program compiled under a release prior to Rel. 1.4, and any of these names were used as element names, you need to change the element name before compiling the program.

---

**Device Output Manipulation**

In pre-Rel. 1.4 programs, you can write code to manipulate the device outputs (.CMMMD, .OPENC, .CLSC, .STRTC, .STOPC, etc., or the symbolic name that has been configured as one of these extensions) when the device is locked. In Rel. 1.5a and higher, the program containing this code still compiles successfully, but the code is ignored. This is due to changes in how APT implements the device logic. Use the commands described in Section 6.1, “Basic Operation of Devices,” of the *SIMATIC APT Programming Reference (Tables) Manual* to manipulate devices.

**Real Fields Require Real Literals**

Fields such as loop alarm ranges require real literals. In pre-Rel. 1.4 programs, APT accepts and displays integers, but converts the values to real values by appending a .0 during compilation. APT now performs the conversions when exiting the form field.

When you compile a pre-Rel. 1.8 program for the first time, APT issues warnings in Phase 2, indicating the integers will be converted to reals. APT no longer issues warnings on subsequent compiles.

**Boolean Array of Length 10: DO10 Tag Type**

When you upgrade a program that was created under APT Rel. 1.3a or earlier, and the program has a Boolean array of length 10 mapped to a TISTAR DO10 tag type, follow these steps before doing a tag translation:

1. Unmark the Boolean array declaration.
2. Change the declaration to a DX declaration type.
3. Mark the DX declaration for translation. The DX declaration type is translated to a DO10 tag type.

## S5 Notes

---

### Sequential Function Charts

The APT Sequential Function Charts (SFCs) are fully operational for S5 controllers. S5 users must be aware that all of the SFCs are reset to the first step when a power failure occurs, regardless of whether a cold or warm restart is issued.

### Archive/Restore from Series 505 to S5

To restore a Series 505 program for an S5 controller, you can use the **Restore** option and then modify the Compiler Control File for an S5 controller. When you switch controller families, everything is retained except the module information. The I/O and their attributes are not lost, but you will lose the addresses. In order to successfully compile, you must perform the following steps:

1. Enter the new addresses for the S5 I/O.
2. Unmark and then remark all tags that are marked for Tag Translation.
3. Select the **Force Compile** option and compile.

If PRAGMA statements are present in the Series 505 program that you are restoring, the compiler ignores them for S5 controllers.

### Scaling S5 Analog Inputs and Outputs

When you select scaling option B, Z or T for an S5 analog input or scaling option B or Z for analog output, the input or output value is automatically shifted by the SCALE or UNSCALE procedure, as described in the “Math Functions and Procedures” chapter of the *SIMATIC APT Programming Reference (Graphics/Math) Manual*. In consequence, when you use Debug to view your scaled analog inputs and scaled analog outputs, their values seem to fall outside the range permitted by the scaling option. This is not an error; during Debug, the values are interpreted as if all 16 bits had resolution. However, analog input and analog output values are properly interpreted by APT and by your S5 module, both of which read the correct in-range value from the upper bits of the analog input or analog output value. See the *SIMATIC APT Programming Reference (Graphics/Math) Manual* for information about how to understand and analyze the values you see when you view scaled analog inputs and scaled analog outputs using Debug.

---

<b>Address Report</b>	<p>The APT address report identifies the peripheral I/O as WORD IN/OUT. The address report does not distinguish between inputs and outputs, because the address can be the same for an input and an output. The user must know which I/O names are inputs and which are outputs.</p> <p>An I/O address report (IO.ADR) lists the I/O points, with their addresses, in consecutive order. With your cursor at the I/O Symbolic Name Table, select <b>R</b> for reports; then select <b>Address Image</b>, which lists the name, type, address, and description for each I/O point.</p>
<b>RBE Code Included</b>	<p>If you are translating to PCS, you must select the <b>RBE code included</b> option in the Compiler Control File. When you select <b>RBE code included</b>, the RBE code uses FB 120 through FB 127 (data-handling blocks) and DBs 100, 101, 108, 110, and 111. Since they are used by the RBE code, these FBs and DBs cannot be reserved.</p> <p>If you are not including the RBE code, you do not get the data-handling blocks. If you need the data-handling blocks, you can reserve FB 120 through FB 127 and download them to the controller. You can also reserve DBs 100, 101, 108, 110, and 111.</p>
<b>Retentive Safe-State SFCs</b>	<p>If you want safe-state SFCs to be retentive in the event of power failure, you must select the <b>RETENTIVE</b> option for the safe-state SFC. A retentive safe-state SFC will execute upon returning from power failure.</p>

## General Notes

---

### Using a Microsoft Mouse

The Microsoft® Mouse software driver Rel. 9.0 causes APT to lock up and act erratically. You must use an earlier version of the driver with APT.

### Low RAM Memory for APT

Even if APT has enough memory (520K bytes) to allow you into the programming environment, you may still have memory problems. For Series 505 controllers, the Module Table may run out of memory in split-screen mode with a large number of I/O points. The CFC editor may run out of memory after entering six PID loops. In both cases, APT displays a yellow memory message. In addition, MAITT may run out of memory if you assign a large number of synonyms and run large WAIT loops.

Returning to the hierarchy may fix the memory problem because it defragments the RAM memory. Another technique to try is reducing the number of CFBs in the CFC. If the problem still occurs, exit out of APT completely. If the problem still continues, consider freeing additional DOS RAM memory. Refer to Appendix C of the *SIMATIC APT User Manual*. Consider freeing 520K bytes of memory for APT.

### Large Arrays

For Series 505 controllers, APT Rel. 1.5a and higher permits large arrays of 32767 elements for integer arrays, and 16383 elements for real arrays. Many operations treat each array element as an individual point. Depending on the speed of your computer, APT may require several minutes on a very fast computer to create the array. Archiving, restoring, copying, or moving a large array individually, or with the declaration table, unit, or program, requires several hours, even with a very fast computer.

If you do not need to provide initial values, or to translate the array elements to PCS, consider reserving a large amount of memory, creating an array of length 1, and providing a user-defined address.



## Compile/Controller Issues

---

### Escape from Compile

To terminate a compile in process, use `[ESC]`. The system acknowledges that `[ESC]` was pressed, but the response is not immediate. The compiler must finish the compile of an object in progress before it can stop.

### CAUTION

**In almost all cases, you can terminate a compile by pressing the ESCAPE key. If you are sure the compile has locked up, you must power cycle the system.**

**Performing a power cycle during program compile corrupts the APT cross-reference files.**

**In order to recover from the power cycle, you must perform a forced compile and full program download in the PROGRAM mode.**

### Database Time/Date Stamps

Each APT object has a time/date stamp that gets updated each time you edit the object. APT incrementally compiles the objects based on the objects' time/date stamps. If you restore an APT program to a computer with an internal clock time and date older than an object's time/date stamp, APT does not incrementally compile the object and can cause unexpected results. You must keep the internal clocks among computers synchronized.

### Extensions of Changed Objects

If you modify an object, some, but not all, records of extensions are deleted. For example, when you create an Integrator block with an integer output, APT creates the `.IOUT` extension. If you later change the output to a real number, the `.IOUT` extension is deleted and an `.OUT` extension is created in its place. However, records of `.IOUT` remain in the symbol table; you can see them in symbol and address reports, and in Debug. The compile fails if you use `.IOUT` in your code.

The compile can report warnings even if you do not reference the object directly in code. If the integer output, `.IOUT`, of an Integrator CFB is marked for translation, and the output is changed to a real variable, the `.IOUT` extension no longer exists. You must unmark and remark the object after the change.

### Analog Alarm and G-Memory

Do not allow an analog alarm CFB process variable to be a SIMATIC 575 G-memory location. Use a V-memory location for the process variable and copy to or from G-memory as needed.

### S-Memory Mapping Change

For Series 505 controllers, if you change the parameters of a standard type CFB, such as a PID or analog alarm block, to map from S-memory instead of V-memory and to incrementally compile, the old initial values may overwrite any new initial values. These changes should be accompanied by a force compile and a full download.

## Compile/Controller Issues (continued)

---

### Flags and Incremental Download

If you delete the only section of code that references a command for an object, e.g., LOCK, OPEN, CLOSE, etc., and then do an incremental download while the controller is in RUN mode and the command is executing, the command remains continuously active. For example, if the only DISABLE command for a CFB is deleted while the command in another CFB is still active, the flag used by APT to disable the CFB remains on continuously.

The following sequence of events causes this to occur:

- You delete the only section of code that references a command for an object. This can be a line of code within an SFC step or a CFB math block, or an entire SFC step or CFB containing the code.
- You recompile the program without selecting the **Force Compile** option, and then do an incremental download.
- The controller is in the PROGRAM or RUN mode when you do the incremental download, and the code referencing the command (the SFC step or the CFB) is active.

The incremental download deletes the code referencing the command, but does not disable the command.

If the command is used in any other section of code in the program, the flag for the command operates correctly, even if this code is not active when you download the program. Therefore, to avoid this situation, be sure that you include at least one other reference to the command in another section of the program.

---

**NOTE:** If the program contains no other code that references the command, you can use APT Debug to disable the command. For example, to turn off the LOCK command, set the device\_name **.RTL** extension to false.

---

### 575 Release 5.x Support

For the 575 Release 5.x controller selection in APT, verify that your controller version number is 5.02 or greater. Please refer to TA 575-006 (Technical Advisory) for more information.

### COPY\_BYTES and the 575

For Series 505 controllers, the COPY\_BYTES procedure to or from word I/O operates differently on the 575 than on the 545 and 555. For the 545 and 555 controllers, a single byte copy of #16#1234 to %WY1 results in a %WY value of #16#12xx where xx retains the value before the procedure. However, for the 575 controller, the same single-byte copy results in a %WY value of #16#1212.

---

**UNSCALE with Reversed Ranges**

For Series 505 controllers, the UNSCALE procedure typically has the low-range value less than the high-range value. Although TISOFT does not allow the low-range value to be higher than the high-range value, APT allows the reversed ranges. In APT, the procedure operates as desired except at the low range. If you choose to use UNSCALE in an unconventional manner, you must unscale the input in a separate conditional structure at the low-range value.

**Object Level Download**

Disable any enabled objects in the controller before incrementally downloading a recompiled version of the object. For Series 505, for example, Dynamic CFBs do not reflect a changed gain until the block is disabled and re-enabled. For S5, CFBs may or may not be disabled and re-enabled; thus, a changed gain may take effect immediately.

**Downloading G-Memory**

For Series 505 controllers, APT does not download application-specific G-memory initial values (for example, %GA5), even if the specified application is the local controller. The APT download ignores any initial value entered in a declaration or recipe value field for objects with application-specific G-memory addresses.

APT does not download VMM- or VMS-memory initial values. The APT download ignores any initial value entered in a declaration or recipe value field for objects with user-assigned VMM- or VMS-memory addresses.

**Initial Value Downloads**

When you change the initial value of an object, and incrementally compile and download, Table 1 shows the objects that are downloaded for each change.

**Table 1 Incremental Downloading**

<b>Object Changed</b>	<b>Object Downloaded</b>
Initial value of a declaration	Declaration
Initial value of an array element	Entire array
Initial value of a recipe array element	Entire recipe array only
Initial value of a recipe element	Recipe element only
Additions or deletions to template	All recipes with the same template

If you incrementally compile and change the user-defined address of an object before incrementally downloading, all addresses that were compiled for the object receive the last initial value during compile.

## Compile/Controller Issues (continued)

---

<b>PROGRAM/RUN Transitions for Timers</b>	For Series 505 controllers, any APT object that uses the RLL TMR or TMRF instruction may be delayed in execution after a controller PROGRAM to RUN transition or a HOLD to RUN transition. The TMR/TMRF instruction does not decrement during the first controller scan after the transition. The declaration timers are examples of APT objects that use the RLL TMR and TMRF instructions.
<b>Disable Event Math CFBs</b>	Disable event math CFBs as soon as the execution finishes. The CFB executes again under certain conditions if left enabled. An event math CFB that maps to RLL or STL, and that is left enabled, executes again if that CFB is re-downloaded as part of an incremental download. For Series 505 controllers, event math CFBs that map to SFPGM code, and that are left enabled, execute again after a controller PROGRAM to RUN transition.
<b>De-queuing SF Math Blocks</b>	For Series 505 controllers, when you disable a math block (CFB math or SFC step math) that is mapped to SFPGM code, the block is no longer placed on the queue for execution. If the block is already on the queue, the block cannot be taken off, and executes one more time. If your application requires that the block not execute again after being disabled, then place the entire math section inside an IF statement, where the IF condition is ( <i>block_name.ENABL</i> ).
<b>Analog Alarm and .SMODE = 1</b>	<p>When an analog alarm CFB is disabled, the <b>.SMODE</b> extension should have a value of 0. When the block is enabled, the <b>.SMODE</b> extension should have a value of 1. Occasionally, the value of the extension may start with a value of 1 after a download when the controller is placed in the RUN mode, and before the block is enabled.</p> <p>After a controller power cycle with the battery on (COLD START), the analog alarm may return in the disabled state unless code enables the block. When the block returns in the disabled state, the <b>.SMODE</b> extension has an incorrect value of 1.</p>

## MAITT Considerations

---

<b>MAITT ASSIGN CONFIG APTPROGRAM</b>	After changing to a new controller using the ASSIGN CONFIG APTPROGRAM = <i>new_controller</i> statement, APT incorrectly applies any previously defined assignments to the new controller. Use the CLEAR ALL statement before the ASSIGN CONFIG APTPROGRAM = <i>new_controller</i> statement to clear any assignments. Refer to the <i>SIMATIC APT MAITT User Manual</i> for assistance with these statements.
<b>MAITT Print Number Base Parameter</b>	You can use the B parameter in the MAITT PRINT instruction to set the number base: 2 = Base <sub>2</sub> , 16 = Base <sub>16</sub> , etc. Do not use 10 with the B parameter; this causes MAITT to generate an error. Since 10 is the default number base, leave out the B parameter for a Base <sub>10</sub> number.
<b>MAITT Memory Constraints</b>	MAITT may run out of memory if you assign a large number of synonyms and run large WAIT loops. To resolve this constraint, use automatic synonym assignments and/or break your WAIT loops into smaller, easier-to-manage WAIT loops.

## TISTAR Considerations

---

**TISOFT Support** If your PCS™ or TISTAR™ system does not support TISOFT 4.x, you cannot view APT-generated code in the controller.

**Trackball Problems** If the trackball does not work correctly with TISTAR after loading APT, you must use the MOUSESYS.SYS driver that came with TISTAR.

Instead of pointing to the MOUSESYS.SYS from APT ...	... point to the MOUSE.SYS from TISTAR
<b>Old CONFIG.SYS</b>	<b>New CONFIG.SYS</b>
⋮	⋮
device=c:\app\drivers\msmouse.sys /g:input	device=c:\msmouse.sys /g:input
⋮	⋮

**TISTAR Memory Limits** If your APT program is very large, or if you reserve lots of V-memory, APT objects may be created with V-memory addresses above 65536. TISTAR Rel. 1.x and 2.x cannot address V-memory addresses above 65536. This is not a limitation for PCS 3.x.

APT now supports 2048 I/O points for the 545 and 8192 I/O points for the 555 and 575 controllers. This does not affect PCS 3.x. When configuring a TISTAR secondary, if your TISTAR release does not support these upper limits, consider choosing a 565 controller that supports up to 8192 I/O points.

**PCS/TISTAR and APT Downloads** When downloading an APT program through a PCS or TISTAR system, PCS and TISTAR automatically turn off the scan to the associated secondary. After the download is complete, the scan is turned on again. PCS and TISTAR do not turn off the scan if the program is downloaded by another method. Turn the scan off and on manually from the NETWORK UTILITIES for TISTAR 1.x and 2.x or from Network Setup for PCS 3.x.

**Setting DOWNLOAD Attribute on Read-Only-Memory Addresses** For Series 505 controllers, an error occurs in tag translation to TISTAR 1.x or 2.x if the DOWNLOAD attribute has been set in the Mark PCS Tags editor for tags with read-only memory addresses K, WX, and X. For example, the analog input .RAW extension could produce this error if the DOWNLOAD attribute is set. Verify that the DOWNLOAD attribute for such tags is not set before compiling.

## Restricting Write Operations in Debug, Download, and MAITT

---

### Remove Debug Functionality

APT Debug and MAITT typically allow you to write or force controller variables. Download allows you to add code to the controller. You may restrict a user, such as an operator, to read functions only. Enter the command below at the DOS prompt to create a file **LIMIT.APT** in the **X:\APT\SYS** subdirectory where **X:** is the drive on which you installed APT.

```
ECHO no > x:\apt\sys\limit.apr
```

The file **LIMIT.APT** will contain the word `no`.

### Customizing the Restrictions

Instead of placing `no` in **LIMIT.APT**, you may tailor which task codes are restricted. Restriction changes occur for all task codes that start with the prefix listed in **LIMIT.APT**. A “-” before the prefix instructs the task code to be blocked. A “+” before the prefix instructs the task code to be allowed.

**Remove this sheet of paper and place it in a secure location if you do not want operators to read how to turn off these restrictions.**

S5 is a trademark, and STEP, SIMATIC, and SINEC are registered trademarks, of Siemens AG.

PCS, APT, Series 505, TISOFT, and TISTAR are trademarks of Siemens Energy & Automation, Inc.

Windows, Windows 95, Microsoft, and MS-DOS are registered trademarks of Microsoft Corporation.