

SIEMENS

SIMATIC APT

User Manual

Order Number: PPX:APT-8101-10
Text Assembly Number: 2601246-0010
Tenth Edition

 **DANGER**

DANGER indicates an imminently hazardous situation that, if not avoided, will result in death or serious injury.

DANGER is limited to the most extreme situations.

 **WARNING**

WARNING indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury, and/or property damage.

 **CAUTION**

CAUTION used with a safety alert symbol indicates a potentially hazardous situation that, if not avoided, could result in minor or moderate injury.

CAUTION

CAUTION used without the safety alert symbol indicates a potentially hazardous situation that, if not avoided, could result in property damage.

NOTICE

NOTICE indicates a potential situation that, if not avoided, could result in an undesirable result or state.

**Copyright 2001 by Siemens Energy & Automation, Inc.
All Rights Reserved — Printed in USA**

Reproduction, transmission, or use of this document or contents is not permitted without express consent of Siemens Energy & Automation, Inc. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Since Siemens Energy & Automation, Inc., does not possess full access to data concerning all of the uses and applications of customer's products, we do not assume responsibility either for customer product design or for any infringements of patents or rights of others which may result from our assistance.

MANUAL PUBLICATION HISTORY

SIMATIC APT User Manual

Order Manual Number: PPX: APT-8101-10

Refer to this history in all correspondence and/or discussion about this manual.

Event	Date	Description
Original Issue	11/88	Original Issue (2601246-0001)
Second Edition	03/90	Second Edition (2601246-0002)
Third Edition	07/91	Third Edition (2601246-0003)
Fourth Edition	11/92	Fourth Edition (2601246-0004)
Fifth Edition	02/94	Fifth Edition (2601246-0005)
Sixth Edition	10/94	Sixth Edition (2601246-0006)
Seventh Edition	05/95	Seventh Edition (2601246-0007)
Eighth Edition	10/96	Eighth Edition (2601246-0008)
Ninth Edition	12/98	Ninth Edition (2601246-0009)
Tenth Edition	4/01	Tenth Edition (2601246-0010)

LIST OF EFFECTIVE PAGES

Pages	Description	Pages	Description
Cover/Copyright	Tenth Edition		
History/Effective Pages	Tenth Edition		
Trademarks/Copyrights	Tenth Edition		
Preface iii — xxiii	Tenth Edition		
1-1 — 1-19	Tenth Edition		
2-1 — 2-21	Tenth Edition		
3-1 — 3-31	Tenth Edition		
4-1 — 4-14	Tenth Edition		
5-1 — 5-11	Tenth Edition		
6-1 — 6-58	Tenth Edition		
7-1 — 7-41	Tenth Edition		
8-1 — 8-31	Tenth Edition		
9-1 — 9-31	Tenth Edition		
A-1 — A-31	Tenth Edition		
B-1 — B-41	Tenth Edition		
C-1 — C-28	Tenth Edition		
D-1 — D-45	Tenth Edition		
E-1 — E-9	Tenth Edition		
F-1 — F-8	Tenth Edition		
G-1 — G-7	Tenth Edition		
H-1 — H-11	Tenth Edition		
Index-1 — Index-10	Tenth Edition		
Registration	Tenth Edition		

Trademarks and Copyrights

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

PCS™, 386/ATM™, APT™, Series 500™, Series 505™, TISOFT™, TISTAR™, TIWAY™, and UNILINK™ are trademarks of Siemens Energy & Automation, Inc.

AST® and AST Premium® are registered trademarks of Bank of America National Trust and Savings Association.

Centronics® is a registered trademark of Genicom International Holdings Corporation.

Compaq® is a registered trademark of Compaq Computer Corporation.

DeskJet®, LaserJet®, HP®, PaintJet®, and Vectra® are registered trademarks of Hewlett-Packard Company.

Deskpro® is a registered trademark of Compaq Computer Corporation.

Epson® is a registered trademark of Seiko Epson Kabushiki Kaisha Shinjuku-ku.

IBM® and AT® are registered trademarks of International Business Machines Corporation.

Logitech™ is a trademark of Logitech Corp.

Matrox® is a registered trademark of Matrox Electronic Systems, Ltd.

Norton Utilities® is a registered trademark of Symantec Corporation.

Novell™ is a trademark of Novell, Inc.

PC Tools™ is a trademark of Central Point Software, Inc.

PKZIP® and PKUNZIP® are registered trademarks of Pkware, Inc.

Pentium® is a registered trademark of the INTEL Corporation.

Quarterdeck Expanded Memory Manager 386 (QEMM®) is a registered trademark of Quarterdeck Office Systems.

Stacker™ is a trademark of Stac Electronics Company.

TIGA™ and TI855™ are trademarks of Texas Instruments.

Trackman™ is a trademark of Logitech, Inc.

UNIX® is a registered trademark of Novell, Inc.

Windows®, Windows®95, Windows NT®, Windows®2000, Microsoft®, and MS-DOS® are registered trademarks of Microsoft Corporation.

Contents

Preface	xxiii
Chapter 1 Getting Started with APT	1-1
1.1 Starting Up APT	1-2
Starting APT on a DOS Computer (Series 505 and S5)	1-2
Starting APT on an OSX Engineering Node (Series 505)	1-3
Starting APT on an OSX Engineering Node (S5)	1-4
Starting APT on a TISTAR Operator Station (Series 505)	1-5
Starting APT on a DOS Computer Connected to a TISTAR System (Series 505)	1-6
Configuring Colors	1-7
1.2 Managing APT Screens	1-8
Screen Characteristics	1-8
APT Icons	1-9
Using the Keyboard	1-10
Using the Mouse	1-11
Paging	1-12
Using a Split Screen	1-13
1.3 Understanding the APT Structure	1-14
APT Hierarchy	1-14
APT Utilities	1-14
1.4 APT Menus	1-15
Menu Windows	1-15
Using Menus	1-16
Using Completion Aids	1-17
1.5 Using On-line Help	1-18
Overview	1-18
Accessing Help	1-18
1.6 Exiting APT	1-19
Returning to the Operating System	1-19
Changing DOS Configuration	1-19
Chapter 2 Understanding the APT Hierarchy	2-1
2.1 Understanding the Directory Levels	2-2
Program Directory	2-2
Program Content	2-4
Unit Content	2-6
2.2 Accessing Levels in the Hierarchy	2-8
Moving Down the APT Hierarchy	2-8
Accessing Another Program or Unit	2-9
Moving Up the Hierarchy	2-9

2.3	Hierarchy Level Activities	2-10
	Using the Options Icon	2-10
	Adding a Program	2-12
	Adding a Unit	2-12
	Using Multi-File Command	2-13
	Archiving and Restoring	2-14
	Copying and Deleting	2-15
	Validating	2-15
2.4	Reports	2-16
	Using the Report Option	2-16
	Table Reports	2-17
	SFC and CFC Reports	2-18
	Object Code Reports	2-20
	Individual Cross Reference Reports	2-21
	Viewing Saved Reports	2-21
	Chapter 3 Using Table Definition Utilities	3-1
3.1	Getting Started with Table Definition Utilities	3-2
	Overview	3-2
3.2	Defining I/O for a Series 505 Controller	3-4
	Defining Modules	3-4
	Defining the I/O Points	3-8
3.3	Defining I/O for an S5 Controller	3-10
	Accessing the I/O Table	3-10
	Defining the I/O Points	3-12
3.4	Editing I/O, Device, Declaration, Recipe, Subroutine Tables	3-14
	Accessing a Table	3-14
	Entering Data into a Table	3-16
	Entering Data into an Array	3-19
	Entering Data into a Recipe	3-20
	Entering Subroutine Math	3-22
3.5	Building a Recipe Template	3-24
	Adding a Template	3-24
	Editing a Template	3-24
3.6	More about the Controls (F2) and Options (F3)	3-26
	Setting Parameters (Alt P)	3-26
	Selecting Completion Aids	3-27
	Using the Cross Reference (F6)	3-28
	Copying Selected Objects (Alt C)	3-30
	Moving Selected Objects (Alt M)	3-31

Chapter 4	Using the SFC Editor	4-1
4.1	Creating an SFC	4-2
	Overview	4-2
	Step/Transition Labels	4-3
	Adding an SFC	4-3
	Accessing the SFC Editor	4-3
	Understanding Drawing Icons	4-4
	Using Drawing Icons	4-5
4.2	Changing an SFC	4-7
	Overview	4-7
	Finding Objects	4-8
	Inserting Graphics	4-8
	Marking a Block	4-9
	Copying a Block to a Buffer	4-9
	Resequencing	4-9
	Using the Options Icon	4-10
	Exiting an SFC	4-11
4.3	Using the SFC Sub-Editor	4-12
	Overview	4-12
	Entering Text	4-12
	Marking a Block of Text	4-13
	Changing Text	4-14
Chapter 5	Using the CFC Editor	5-1
5.1	Creating a CFC	5-2
	Overview	5-2
	Adding a CFC	5-2
	Accessing the CFC Editor	5-2
	Understanding Block Icons	5-2
	Using Block Icons	5-4
	Linking Control Blocks	5-4
	Connecting Blocks	5-5
	Disconnecting Blocks	5-5
5.2	Changing a CFC	5-6
	Overview	5-6
	Marking a Block	5-6
	Using the Options Icons	5-7
	Exiting a CFC	5-7
5.3	Using the Form Sub-Editor	5-8
	Editing a Form	5-8
	Using the Options Icon	5-9
5.4	Using the Math Sub-Editor	5-10
	Accessing the Math Editor	5-10
	Changing Text	5-11

Chapter 6	Compiling an APT Program	6-1
6.1	Understanding the Compile Operation	6-2
	Compile Phases	6-2
	Compile Options	6-2
6.2	Compiling for Series 505 Controllers	6-4
	Understanding the Compiler Control File	6-4
	Editing the Compiler Control File	6-15
	Starting the Compile Operation	6-17
	Compile Screen	6-18
	Using the Compile Report	6-19
6.3	Compiling for S5 Controllers	6-20
	Understanding the Compiler Control File	6-20
	Operating System Defaults (928B CPU)	6-24
	Error Processing Checks (928B CPU)	6-25
	RBE Options (928B CPU)	6-26
	Operating System Defaults (948 CPU)	6-28
	RBE Options (948 CPU)	6-30
	Editing the Compiler Control File (928B and 948 CPUs)	6-34
	Completing the Compiler Control File (928B CPU)	6-36
	Completing the Compiler Control File (948 CPU)	6-37
	Starting the Compile Operation	6-38
	Using the Compile Screen	6-39
	Using the Compile Report	6-39
6.4	Object File Reports	6-41
	Overview	6-41
	Using Object Reports	6-42
	Deleting Object Code	6-43
6.5	Downloading Program to Controller	6-44
	When to Use the Incremental Download	6-45
	Quick Verification Option	6-46
	Obstacles to Incremental Download	6-46
	Overflow CRs and Incremental Download	6-47
	Incremental Download for S5 Redundant System	6-47
	Incremental Download and RUN Mode	6-48
	Downloading a Program from APT	6-50
	Downloading a Program from DOS (Series 505)	6-54
	Downloading a Program from DOS (S5)	6-56
6.6	Verifying a Downloaded Program	6-57
	Understanding Verify	6-57
	Using Verify	6-58

Chapter 7	Debugging an APT Program	7-1
7.1	Understanding the Debug Operation	7-2
	Overview	7-2
	Starting Debug	7-4
	Using Debug	7-6
	Using the Series 505 AUX Functions	7-8
	Using the S5 AUX Functions	7-10
	Debugging Any Controller Program	7-12
	Using Prog/Run Option	7-14
	Using Sort Entries Option	7-17
	Using Split Screen Option	7-17
	Exiting Debug (and Forced I/O Display for Series 505)	7-17
7.2	Monitoring Variables	7-18
	Overview	7-18
	Working with Charts	7-18
	Manually Building a Chart	7-18
	Automatically Building a Chart	7-19
	Completing the Build Monitor Chart Dialog Box	7-20
	Editing a Chart	7-22
	Listing Charts	7-23
	Changing Charts	7-23
	Using the Monitor and Sample Options	7-24
	Using Trend Option	7-26
	Using Animate Option	7-28
	Using Direct Address Option	7-29
7.3	Changing I/O and Variables	7-30
	Overview	7-30
	Using Modify Option	7-31
	Using Force Option (Series 505 Only)	7-32
	Hints: Forcing Variables	7-33
7.4	Tracing SFC Program Flow	7-34
	Overview	7-34
	Using Animate Option	7-34
	Using Breakpoint Option (Series 505 Only)	7-35
	Using Single Step Option (Series 505 Only)	7-36
	Using Step Logger Option (Series 505 Only)	7-37
7.5	Using the Activate Option	7-38
	Overview	7-38
	Activating SFC Steps	7-38
	Activating CFBs	7-40
	Activating Devices	7-41
Chapter 8	APT, OSx/TISTAR, and TISOFT/STEP 5	8-1
8.1	Translating APT Objects to OSx Tags	8-2
	For S5 Users	8-3

8.2	Tag Translation — Choosing an OSx/PCS Release	8-4
	Choosing an OSx/PCS Release	8-4
	Appending or Building the Translate Table	8-5
8.3	Tag Translation — Marking Objects	8-6
	Using the Tag Translate Editor	8-6
	Using the Tag Translate Sub-Editor	8-8
	Using the Attributes Sub-Editor	8-9
	Using the Process Groups Sub-Editor	8-10
	Unmarking an Object	8-11
8.4	Tag Translation — Compiling the Program	8-12
	Compile Options	8-12
	Compiling a Program	8-12
8.5	Tag Translation — Selecting the Translate Utilities	8-16
8.6	Translation on an OSx Engineering Node	8-17
8.7	Translation from an APT Workstation to an OSx System	8-18
8.8	Translation on a TISTAR Operator Station with APT	8-19
8.9	Translation from APT Workstation to TISTAR Operator Station with APT	8-20
8.10	Translation from APT Workstation to TISTAR Operator Station w/o APT	8-22
	Placing File TRANS.EXE on the Operator Station	8-22
	Placing File INSTALL.TAG or APPEND.TAG on the Operator Station	8-24
	Executing Translation	8-24
8.11	Translation from APT Workstation to TISTAR Model 20 System	8-25
	Transferring INSTALL.TAG to a Diskette	8-25
	Translating the Tags	8-26
8.12	Using TISOFT with APT	8-28
	Creating TISOFT Synonyms and Comments	8-28
	Examining APT Code	8-29
8.13	Using STEP 5 With APT	8-30
	Creating STEP 5 Synonyms and Comments	8-30
	Examining APT Code	8-31
Chapter 9	Using PROFIBUS with APT	9-1
9.1	Overview	9-2
9.2	Configuring the PROFIBUS Network (Version 3.3)	9-4
9.3	Exporting the Binary File to APT (COM PROFIBUS V 3.3)	9-12
9.4	Configuring the PROFIBUS Network (Version 5.0)	9-14
9.5	Exporting the Binary File to APT (COM PROFIBUS V 5.0)	9-16
9.6	Adding PROFIBUS Modules in APT	9-17

9.7	Creating a PROFIBUS Configuration File in APT	9-20
	Guidelines for PROFIBUS.CFG Report	9-22
9.8	Debugging PROFIBUS in APT	9-23
	Setting the Slave Mode	9-23
	Setting the Operational Mode	9-26
	Setting the PROFIBUS-DP Synchronization Mode	9-28
	Reading PROFIBUS-DP I/O	9-30
	Appendix A APT Installation	A-1
A.1	Before Installing APT	A-2
	Compatibility	A-2
	Minimum Hardware and Software	A-3
	Recommended Hardware and Software	A-3
	Setting Up the CONFIG.SYS and AUTOEXEC.BAT Files	A-4
	DOS Memory	A-4
	Microsoft Mouse or Compatible	A-4
A.2	Installing APT	A-5
	Overview	A-5
	Creating Diskettes	A-5
	Starting the Installation	A-6
	APT Installation	A-7
	Configuring the System	A-8
	Copying the Files	A-11
	Setting Computer to APT Mode (DOS)	A-11
	Running APT	A-12
	Loading an Example Program	A-12
	Changing between APT Mode and DOS Mode (DOS)	A-14
	Configuring the CVU/TIWAY Adapter (Series 505 Only) (DOS/Win95 Only)	A-15
	Configuring the 386/ATM (Series 505 Only) (DOS Only)	A-15
A.3	Setting DOS Environment Variables	A-16
A.4	Changing the APT Graphic Configuration	A-24
	CGI.CFG File	A-24
	Printer Support	A-25
A.5	Hardware Connections	A-26
	Port Pinouts	A-26
	Series 505 Communication Links	A-28
	S5 Communication Links	A-30
	Appendix B APT on an Industrial Ethernet Network	B-1
B.1	Overview	B-2

B.2	Planning the Network Configuration for Series 505	B-3
	Assigning Node Names	B-3
	Setting Ethernet Addresses	B-4
	Setting Logical Network Connections	B-5
	Assigning TSAPs	B-6
	Gathering Configuration Data for Each Node	B-7
B.3	Planning the Network Configuration for S5	B-9
	Assigning Node Names	B-9
	Setting Ethernet Addresses and Paths	B-10
	Setting Logical Network Connections	B-12
B.4	Connecting APT to the Industrial Ethernet Network (DOS/Windows 95)	B-14
	Beginning Hardware Configuration	B-14
	Configuring the CP1413 Card (Series 505 and S5)	B-14
	If Configuration Fails (Series 505 and S5)	B-15
	Installing a Memory Manager (Series 505 and S5)	B-16
	Verifying CP1413 Configuration (Series 505 and S5)	B-17
B.5	Connecting APT to the Industrial Ethernet Network (Windows NT/2000)	B-18
	Before You Begin	B-18
	Beginning Hardware Configuration	B-18
	Loading the SIMATIC NET Software	B-18
	Defining the Point of Access for the CP1613 Card	B-20
	Testing the CP1613 Card Installation	B-21
	Installing Service Packs	B-21
B.6	Configuring the Network	B-22
	Network Configuration (Series 505 Only)	B-22
	Creating a TF Service with the H1 Configurator (Series 505 Only)	B-24
	Creating a Matching Association with COML (Series 505 Only)	B-26
	Network Configuration (S5 Only)	B-28
	APT Configuration (DOS only)	B-30
	APT Configuration (NT/2000 Only)	B-31
	Testing the APT Configuration	B-32
	Memory Usage and the APT Drivers (DOS only)	B-33
	Manually Loading and Starting the APT H1 Server (NT/2000 Only)	B-35
	Specifying the H1 Driver for APT	B-36
	APT H1 Server Registry Values	B-37
	APT H1 Server Error Messages	B-40
	Using APT with the Industrial Ethernet Network	B-41
	Appendix C Using APT with DOS	C-1
C.1	DOS Memory Management	C-2
	Understanding the DOS Memory Types	C-2
	Using a Memory Manager	C-4
C.2	APT and DOS Memory	C-5
	Using Memory Effectively	C-5
	Configuring Memory Areas	C-5
	Effects of Hardware on Memory	C-7

C.3	Memory Configurations: Introduction	C-8
C.4	Memory Configurations: Compaq LTE Elite	C-10
	Using MS-DOS 5.0	C-10
	Using MS-DOS 6.22 with QEMM 7.04	C-12
C.5	Memory Configurations: 386/ATM Coprocessor Module	C-14
	Using MS-DOS 5.0 or Higher	C-14
C.6	Memory Configurations: TISTAR Model 70	C-16
	Using TISTAR 2.0 Operator Station	C-16
	Using TISTAR 1.3 Operator Station	C-18
C.7	Memory Configurations: Compaq Deskpro 486	C-20
	Using Compaq DOS 5.0 or Higher and a Network	C-20
C.8	Memory Configurations: HP XU 5/90C	C-24
	Using MS-DOS 6.22 and a Network	C-24
	Using MS-DOS 6.22 with QEMM 7.04 and a Network	C-26
C.9	Memory Configuration: Additional Boot Files	C-28
	Using MS-DOS 5.0 or Higher	C-28
Appendix D Using APT with Windows 95		D-1
D.1	Running APT under Windows 95	D-2
	Overview	D-2
	Selecting the Type of DOS Emulation	D-2
	Virtual DOS Environment under Windows 95	D-4
	Real Mode DOS Environment with Standard Configuration	D-6
	Real Mode DOS Environment with New Configuration	D-8
D.2	Installing APT in a Virtual DOS Environment	D-10
	Overview	D-10
	Creating a DOS Icon for APT	D-10
	Setting the Properties for the DOS Window	D-12
	Configuring Memory Management	D-15
	Installing APT	D-19
	Writing a Configuration Batch File	D-23
	Running APT in the Window	D-24
	Resource Sharing with Other DOS and Windows Programs	D-25
D.3	Installing APT in Real Mode DOS Environment (Standard Configuration)	D-26
	Overview	D-26
	Creating a DOS Icon for APT	D-26
	Setting the Properties for the DOS Window	D-27
	Configuring Memory Management	D-29
	Installing APT	D-32
	Running APT in DOS Mode	D-35

D.4	Installing APT in Real Mode DOS Environment with New Configuration	D-36
	Overview	D-36
	Creating a DOS Icon for APT	D-36
	Setting the Properties for the DOS Window	D-37
	Configuring Memory Management	D-39
	Installing APT	D-42
	Running APT in DOS Mode	D-45
	Appendix E Using APT with Windows NT/2000	E-1
E.1	Installing APT on Windows NT/2000	E-2
	Overview	E-2
	DOS Execution under Windows	E-2
	Program Information File	E-3
	System Variables	E-6
	Communications Servers	E-7
	Communication Conflicts	E-7
	Communication Reliability	E-8
	PIF for Command Line Access	E-8
	Appendix F Disk Management	F-1
F.1	Disk Management	F-2
	Multiple Drives or Drive Partitions	F-2
	APT Program Organization	F-2
	Using a Single Drive	F-3
	Using Two Drives	F-4
	Using Three Drives	F-5
	Using a Network Drive	F-5
	Using a Virtual (RAM) Drive	F-6
	Appendix G APT Computer Performance	G-1
G.1	Planning an Efficient System	G-2
	Defragmenting RAM	G-2
	Defragmenting the Hard Disk	G-2
	Disk-Caching	G-2
	Compressing the Database	G-3
G.2	Maintaining Data Integrity	G-4
	Using the APT Archive Option	G-4
	Using Other Backup Programs	G-4
	Using the DOS CHKDSK or SCANDISK Command	G-5
	Recovering from Power Failure or System Lockup	G-6
G.3	Transporting the Program	G-7
	Archiving the Program	G-7
	Compressing the Program	G-7

Appendix H	DOS Utilities	H-1
H.1	Using the Archive and Restore Utilities	H-2
	Archiving Programs	H-2
	Restoring Programs	H-3
H.2	Using the EXTRACT Utility	H-4
H.3	Using the DDS Utility	H-6
H.4	Using the SHOWLINE Utility	H-8
H.5	Using the DC Utility	H-9
H.6	Using the CHKVIDEO Utility	H-10
H.7	Additional Utilities	H-11
	Downloading a Program from DOS	H-11
	Translating Tags Outside the APT Environment	H-11
	TRANSOFF	H-11
	Recovering RAM	H-11
Index		Index-1

List of Figures

1-1	Starting APT	1-2
1-2	Data Transfer on OSx with Series 505 Controllers	1-3
1-3	Data Transfer on OSx with S5 Controllers	1-4
1-4	Data Transfer with APT as a TISTAR Operator Station	1-6
1-5	Color Configuration Dialog Window	1-7
1-6	APT Screen Layout	1-8
1-7	APT Icons	1-9
1-8	Using the Text Cursor	1-10
1-9	Using the Cross-Hair	1-11
1-10	Using the Mouse	1-11
1-11	Relative Position Icon	1-12
1-12	Using a Split Screen	1-13
1-13	APT Program Hierarchy	1-14
1-14	Using Menu Windows	1-15
1-15	Displaying the Completion Aids	1-17
1-16	On-line Help	1-18
1-17	Exit Dialog Window	1-19
2-1	APT Program Directory Level	2-2
2-2	Program Directory Controls and Options	2-3
2-3	Program Content Level	2-4
2-4	Program Content Controls and Options	2-5
2-5	Unit Content Level	2-6
2-6	Unit Content Controls and Options	2-7
2-7	Moving Down the APT Hierarchy	2-8
2-8	Selecting Another Program	2-9
2-9	Using the Multi-File Command	2-13
2-10	Device Table Report Options	2-17
2-11	SFC Report Options	2-18
2-12	CFC Report Options	2-19
3-1	Availability of Table Definition Utilities	3-2
3-2	Table Sections for All Other Table Editors	3-3
3-3	Table Sections for the Module Editor	3-3
3-4	Defining Modules	3-4
3-5	Base Slot Icon	3-6
3-6	Defining I/O Points	3-8
3-7	I/O Symbolic Name Table Object List	3-10
3-8	S5 I/O Symbolic Name Table Objects	3-13
3-9	Device Table Object List	3-14
3-10	Defining Table Objects	3-17
3-11	Editing an Array	3-19

3-12	Editing Recipe Elements	3-20
3-13	Entering Values for Recipe Elements	3-21
3-14	Editing Subroutine Math	3-22
3-15	Adding a Template	3-24
3-16	Examining List of Template Elements	3-25
3-17	Editing a Template Element	3-25
3-18	Displaying the Completion Aids	3-27
3-19	Example Cross Reference: Items That the Specified Object References	3-28
3-20	Example Cross Reference: Items That the Specified Object Is Referenced By	3-29
4-1	Parallel and Selection Branches	4-2
4-2	SFC Drawing Icons	4-4
4-3	Using the Drawing Icons	4-6
4-4	SFC Controls and Options	4-7
4-5	Exiting an SFC	4-11
4-6	Using the SFC Sub-Editor	4-12
4-7	Marking a Block of Text	4-13
5-1	Selecting the Block Type	5-3
5-2	Connected Control Blocks	5-5
5-3	CFC Controls and Options	5-6
5-4	Using the CFC Form Sub-Editor	5-8
5-5	Using the CFC Math Sub-Editor	5-10
6-1	Compiler Control File — 545	6-8
6-2	Compiler Control File — 545L	6-9
6-3	Compiler Control File — 555	6-10
6-4	Compiler Control File — 575	6-11
6-5	Compiler Control File — 560	6-12
6-6	Compiler Control File — 560T	6-12
6-7	Compiler Control File — 565	6-13
6-8	Compiler Control File — 565T	6-13
6-9	Compiler Control File — 565P	6-14
6-10	Printing or Showing the Compile Report	6-19
6-11	Compiler Control File — CPU928B	6-32
6-12	Compiler Control File — CPU948	6-33
6-13	Compiling an APT Program	6-38
6-14	Printing or Showing the Compile Report	6-39
6-15	Using Object Reports (Series 505)	6-42
6-16	Using Object Reports (S5)	6-43
6-17	Download Requirements and I/O Assignments for Series 505	6-47
6-18	Example Program Memory Requirements for Series 505	6-48
6-19	Download Dialog Windows	6-52

6-20	Verifying a Program	6-58
7-1	Debug Hierarchy	7-3
7-2	Controls and Options (Keys F2 and F3)	7-6
7-3	Series 505 AUX Functions (Key F5)	7-8
7-4	S5 AUX Functions (Key F5)	7-10
7-5	Accessing the Build Chart Option	7-19
7-6	Example of Chart Comments	7-22
7-7	Monitoring Variables	7-24
7-8	Chart Fields during an Edit	7-25
7-9	Trend Window Range Configuration	7-27
7-10	Trend Window	7-27
7-11	Animating MSS Extensions	7-28
7-12	Reading the Direct Address of a Variable	7-29
7-13	Modifying the Value of a Variable	7-31
7-14	Forcing a Variable	7-32
7-15	Using Breakpoint	7-35
7-16	Using Single Step	7-36
7-17	Using Step Logger	7-37
7-18	Activating an SFC Step	7-39
7-19	Activating a Loop	7-40
7-20	Activating a CFB	7-40
7-21	Activating a Device	7-41
8-1	RBE Data for S5 Compiler Control File	8-3
8-2	Example of Marking OSx Tags	8-6
8-3	Example of Tag Translate Editor	8-7
8-4	Example of Tag Translate Sub-Editor	8-8
8-5	Example of Tag Attributes Sub-Editor	8-9
8-6	Menu for Process Groups	8-11
8-7	Translating Tags to OSx	8-17
8-8	Offline Tag Translation to OSx	8-18
8-9	Translating Tags to TISTAR	8-19
8-10	Offline Tag Translation to TISTAR	8-20
8-11	Offline Tag Translation to TISTAR without APT	8-22
8-12	Offline Tag Translation to a TISTAR Model 20 System	8-25
8-13	Creating TISOFT Comments and Synonyms	8-29
8-14	Creating STEP 5 Comments and Synonyms	8-31
9-1	Sample PROFIBUS Network	9-4
9-2	New Master System Dialog Box	9-5
9-3	DP Master System PROFIBUS Address 1 Window	9-6
9-4	PROFIBUS Address Dialog Box	9-7

9-5	Slave Parameters Dialog Box	9-7
9-6	Configure Dialog Box	9-8
9-7	505 Module Selection Dialog Box	9-9
9-8	Configuring a Module	9-10
9-9	Saving the PROFIBUS Configuration	9-11
9-10	Selecting a Directory Path for the Export File	9-12
9-11	Export Binary File Dialog Box	9-13
9-12	Defining I/O Points	9-19
9-13	Creating the PROFIBUS.CFG Report	9-20
9-14	Naming the PROFIBUS.CFG File	9-21
9-15	Example PROFIBUS.CFG File	9-21
9-16	Format and Examples for PROFIBUS.CFG File	9-22
9-17	Selecting PROFIBUS-DP Mode	9-23
9-18	Selecting Slave	9-24
9-19	Specifying the Slave Address	9-24
9-20	Setting the Slave Mode	9-25
9-21	Selecting Operational Mode	9-26
9-22	Setting the Operational Mode	9-27
9-23	Selecting PROFIBUS-DP Synchronization Mode	9-28
9-24	Synchronizing PROFIBUS-DP I/O with the RLL Scan	9-29
9-25	Selecting Read PROFIBUS-DP I/O	9-30
9-26	Specifying the Slave Address	9-30
9-27	PROFIBUS-DP Slave I/O Configuration Report	9-31
A-1	RS-232/C 25-pin to 25-pin Connectors	A-26
A-2	RS-232/C 9-pin to 25-pin Connectors	A-26
A-3	RS-232/C 9-pin to 9-pin Connectors	A-27
A-4	APT/Series 505 Controller Communication Link	A-29
A-5	APT/S5 Controller Communication Link	A-31
B-1	Industrial Ethernet Network with APT, OSx Station, and S5 and Series 505 Controllers	B-2
B-2	Network Node Names for Series 505	B-3
B-3	Example Network Node Addresses for Series 505	B-4
B-4	Logical Layout for OSx with Series 505 Controllers	B-5
B-5	Network Node Names for S5	B-9
B-6	Example Network Node Addresses for S5	B-11
B-7	Logical Layout for PCS with S5 Controllers	B-13
B-8	Three Memory Manager Setup Examples	B-16
B-9	Adding the CP1613 Card to the PG/PC Interface List	B-19
B-10	Defining the Point of Access for the CP1613 Card	B-20
B-11	Testing the CP1613 Card	B-21
B-12	Association for a CP1434TF Module	B-23
B-13	Matching Association for a CP1413/CP1613 Card	B-23

B-14	Sample S5-H1 Configuration File	B-29
B-15	Sample AUTOEXEC and CONFIG Files	B-30
B-16	Configuring the CP1613 Card Properties	B-31
C-1	DOS Memory Configuration	C-3
C-2	AUTOEXEC.BAT and CONFIG.SYS for Basic Computer Configurations	C-8
C-3	Content of CGI.CFG	C-9
C-4	Compaq LTE Elite CONFIG.SYS and AUTOEXEC.BAT for APT	C-10
C-5	Compaq LTE Elite CONFIG.SYS and AUTOEXEC.BAT for APT under QEMM	C-12
C-6	386/ATM Coprocessor CONFIG.SYS and AUTOEXEC.BAT for APT	C-14
C-7	TISTAR 2.0 Model 70 Operator Station CONFIG.SYS and AUTOEXEC.BAT for APT	C-16
C-8	TISTAR 1.3 Model 70 Operator Station CONFIG.SYS and AUTOEXEC.BAT for APT	C-18
C-9	Compaq Deskpro 486 CONFIG.SYS and AUTOEXEC.BAT for APT	C-20
C-10	Pentium CONFIG.SYS and AUTOEXEC.BAT for APT	C-24
C-11	Pentium CONFIG.SYS and AUTOEXEC.BAT for APT under QEMM	C-26
C-12	MS-DOS 5.0 Additional Boot Files	C-28
D-1	Taskbar Properties Dialog Box	D-10
D-2	Start Menu Program Explorer Window	D-11
D-3	MS-DOS Prompt Properties Dialog Box	D-12
D-4	Advanced Program Settings Dialog Box	D-13
D-5	Example of Memory Usage Report	D-17
D-6	APT Installation Program, Final Screen	D-19
D-7	Memory Usage	D-21
D-8	Change Icon Dialog Box	D-24
D-9	MS-DOS Mode Properties Dialog Box	D-27
D-10	Advanced Program Settings Dialog Box	D-28
D-11	Memory Usage Report for Standard Configuration	D-31
D-12	Memory Usage Report for Real Mode DOS Environment	D-33
D-13	MS-DOS Prompt Properties Dialog Box	D-37
D-14	Advanced Program Settings Dialog Box	D-38
D-15	Memory Usage Report for Preliminary New Configuration	D-41
D-16	Memory Usage Report for New Configuration	D-43
E-1	Program Folder of the PIF	E-3
E-2	Windows NT PIF Settings	E-5
E-3	APT_CMD Properties: Program	E-9
E-4	APT_CMD Properties: Screen	E-9
F-1	APT Configured to Run on One Drive	F-3
F-2	APT Configured to Run on Two Drives	F-4

F-3	APT Configured to Run on Three Drives	F-5
F-4	APT Configured to Run on a Virtual Drive	F-6
F-5	Programs That Load/Save Files to/from a Virtual Drive	F-8
H-1	The SHOWLINE Utility	H-8

List of Tables

1-1	APT Function Keys	1-10
1-2	APT Screen Paging Keys	1-12
2-1	Options at Hierarchy Level	2-10
2-2	Report Options	2-16
3-1	Controls (F2) and Options (F3) for Defining Modules	3-7
3-2	Controls (F2) and Options (F3) for Defining I/O Points	3-9
3-3	Controls (F2) and Options (F3) for I/O Table Object List	3-11
3-4	Field Entries for S5 I/O	3-13
3-5	Controls (F2) and Options (F3) for Table Object List	3-15
3-6	Controls (F2) and Options (F3) for Defining Table Objects	3-18
3-7	Controls (F2) and Options (F3) for Subroutine Sub-Editor	3-23
3-8	Parameter Options	3-26
4-1	APT Find Command	4-8
4-2	Options for SFC Graphic Editor	4-10
4-3	Options for SFC Sub-Editor	4-14
5-1	Options for CFC Editor	5-7
5-2	Options for CFC Form Sub-Editor	5-9
5-3	Options for CFC Math Sub-Editor	5-11
6-1	Controller Memory Configuration	6-5
6-2	CPU928B Program Interrupts	6-26
6-3	CPU948 Program Interrupts	6-29
6-4	CP/CPU Configuration Options for the 948	6-30
7-1	Debug Options Defined	7-7
7-2	Series 505 AUX Functions Defined	7-9
7-3	S5 AUX Functions Defined	7-11
7-4	Chart Editing Functions	7-23
8-1	Controllers Compatible with OSx 4.x	8-4
A-1	APT Configuration Parameters	A-9
A-2	DOS Environment Variables	A-17
B-1	Example Node Names and Descriptions for Series 505	B-3
B-2	Example Node Names and Addresses for Series 505	B-4
B-3	Full Listing of Nodes and Associations	B-5
B-4	Reduced Listing of Nodes and Associations	B-5
B-5	Node TSAPs	B-6
B-6	APT Node Configuration Data	B-7
B-7	OSx Station Configuration Data	B-7
B-8	Controller Configuration Data	B-8

B-9	Example Node Names and Descriptions for S5	B-9
B-10	Example Node Names, Addresses, and Paths for S5	B-10
B-11	Full Listing of Nodes and Associations	B-12
B-12	Reduced Listing of Nodes and Associations	B-12
B-13	CP1413 Configuration Prompts	B-14
B-14	Single Node Configuration	B-22
B-15	Memory Requirements for APT-H1 Drivers	B-34
B-16	APT H1 Server Registry Entries	B-37
B-17	Sub-key Registry Entries	B-38
B-18	APT H1 Server Error Messages	B-40
C-1	DOS Memory Types	C-2
C-2	APT Memory Requirements	C-5
C-3	Hardware Cards That Require Reserved Memory	C-7
C-4	Memory Configuration for the Compaq LTE Elite	C-11
C-5	Memory Configuration for the Compaq LTE Elite under QEMM	C-13
C-6	Memory Configuration for the 386/ATM Coprocessor	C-15
C-7	Memory Configuration for the TISTAR 2.0 Model 70	C-17
C-8	Memory Configuration for the TISTAR 1.3 Model 70	C-19
C-9	Memory Configuration for the Compaq Deskpro 486	C-21
C-10	Region Sizes in Kbytes Determined by Starting Address	C-23
C-11	Memory Configuration for the Pentium	C-25
C-12	Memory Configuration for the Pentium under QEMM	C-27
F-1	APT Program Organization	F-2

New Features of APT

SIMATIC Application Productivity Tool — APT is a software package that you can use to design and implement a solution to your process control problem. The capabilities of APT have been enhanced in Software Release 1.9A. The documented differences between APT Release 1.9 and Release 1.9A are indicated by change bars in the manual page margins.

Controller Families

APT continues to support two controller families, the Series 505 and the SIMATIC S5. Most programming tasks, like writing a program, downloading, or debugging, are handled the same way in APT regardless of your controller type. The way APT treats direct memory addressing and I/O is determined by whether you have an S5 or a Series 505 controller.

Using APT Documentation

The APT manual set is organized to make it easy both to use the manuals and to follow the program design process that is appropriate for APT. The APT manual organization is described below.

- *SIMATIC APT User Manual* (Volume 1) is a guide for using the operator interface to enter your program.
- *SIMATIC APT Programming Reference (Tables) Manual* (Volume 2) and *SIMATIC APT Programming Reference (Graphics/Math) Manual* (Volume 3) provide the information that you need to design your process control solution. These manuals describe the APT programming languages, the characteristics of APT objects, and the tables that you use to configure APT objects. Information is presented in the order that provides for the most efficient and logical design of an APT program.
- *SIMATIC APT Applications Manual* (Volume 4) is intended to help you design and write an application program using APT. It includes programming hints, specific examples, and a recommended approach to designing the controls for a factory process.
- *SIMATIC APT MAITT User Manual* (Volume 5) provides the information that you need to design and execute a test program for an application program.
- *SIMATIC APT Release Notes* have important information not included in the manual set.
- The APT manual set is available both in paper form (APT-8200-T) and in electronic form on CD-ROM (APT-8200-CD).

NOTE: Unless otherwise specified, the term “OSx” is used throughout this manual to designate SIMATIC TISTAR Releases 1.x and 2.x in addition to SIMATIC PCS Release 3.x and OSx 4.x.

Getting Started with APT

1.1	Starting Up APT	1-2
	Starting APT on a DOS Computer (Series 505 and S5)	1-2
	Starting APT on an OSX Engineering Node (Series 505)	1-3
	Starting APT on an OSX Engineering Node (S5)	1-4
	Starting APT on a TISTAR Operator Station (Series 505)	1-5
	Starting APT on a DOS Computer Connected to a TISTAR System (Series 505)	1-6
	Configuring Colors	1-7
1.2	Managing APT Screens	1-8
	Screen Characteristics	1-8
	APT Icons	1-9
	Using the Keyboard	1-10
	Using the Mouse	1-11
	Paging	1-12
	Using a Split Screen	1-13
1.3	Understanding the APT Structure	1-14
	APT Hierarchy	1-14
	APT Utilities	1-14
1.4	APT Menus	1-15
	Menu Windows	1-15
	Using Menus	1-16
	Using Completion Aids	1-17
1.5	Using On-line Help	1-18
	Overview	1-18
	Accessing Help	1-18
1.6	Exiting APT	1-19
	Returning to the Operating System	1-19
	Changing DOS Configuration	1-19

1.1 Starting Up APT

APT supports the Series 505 controller family and the S5 controller family. Before using APT, refer to [Appendix A](#) and install the software as described. Then, depending upon your system, choose one of the following five procedures to start the APT program. The procedures are labeled by controller family.

Starting APT on a DOS Computer (Series 505 and S5)

If you are using APT on an MS-DOS computer, you can use the following procedure whether you have a Series 505 controller or an S5 controller.

From the DOS prompt, type APT in upper or lower case letters and press **Enter**. The title screen appears, and APT appears as shown in [Figure 1-1](#).

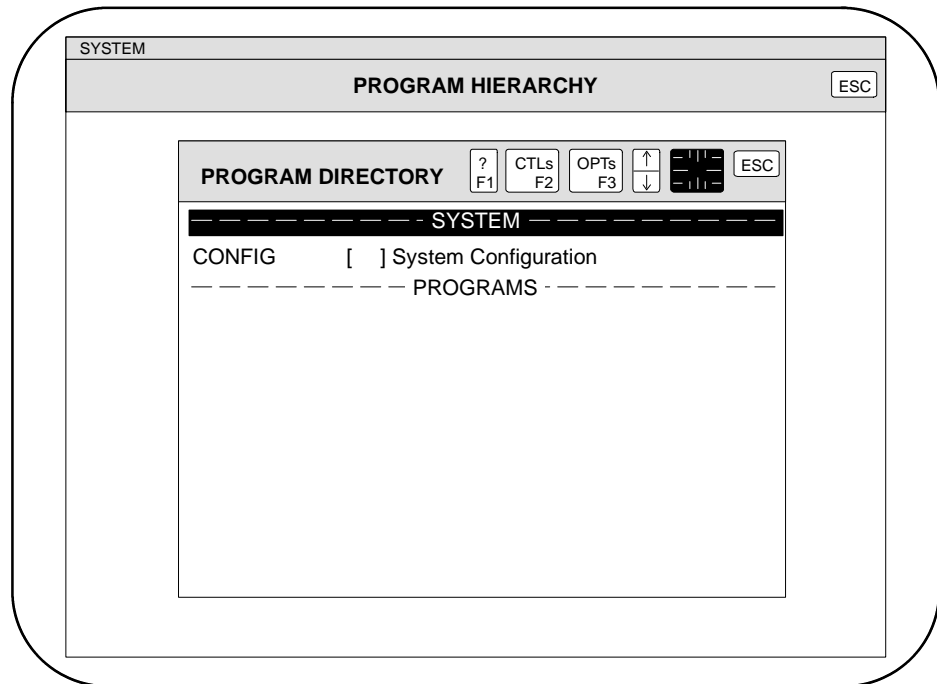


Figure 1-1 Starting APT

Starting APT on an OSX Engineering Node (Series 505)

If you are using a Series 505 controller and you are running APT on an OSX engineering node on the Industrial Ethernet (H1) network, then you can do the following types of data transfer ([Figure 1-2](#)).

- Develop your application (controller) program with APT and then transfer tag information to the OSx operation node.
- Download and debug an application to the controller through the Industrial Ethernet network.

To set up APT with the Industrial Ethernet network, refer to [Appendix B](#). After the hardware and software have been configured, you can start up APT as follows:

From the DOS prompt, type APT and press **[Enter]**. The title screen appears, and APT appears as shown in [Figure 1-1](#).

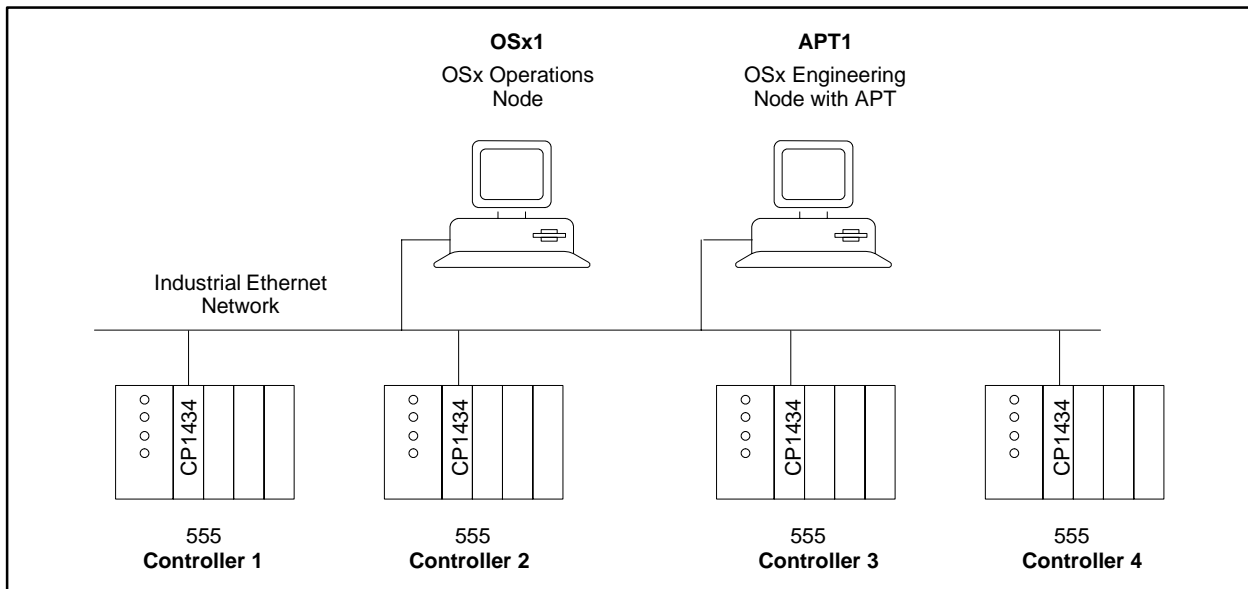


Figure 1-2 Data Transfer on OSx with Series 505 Controllers

Starting Up APT (continued)

Starting APT on an OSX Engineering Node (S5)

If you are using an S5 controller and you are running APT on an OSX engineering node on the Industrial Ethernet network, then you can do the following types of data transfer ([Figure 1-3](#)).

- Develop your application (controller) program with APT and then transfer tag information to the OSX operation node.
- Download and debug an application to the controller through the Industrial Ethernet network.

To set up APT with the Industrial Ethernet network, refer to [Appendix B](#). After the hardware and software have been configured, you can start up APT as follows:

From the DOS prompt, type APT and press **[Enter]**. The title screen appears, and APT appears as shown in [Figure 1-1](#).

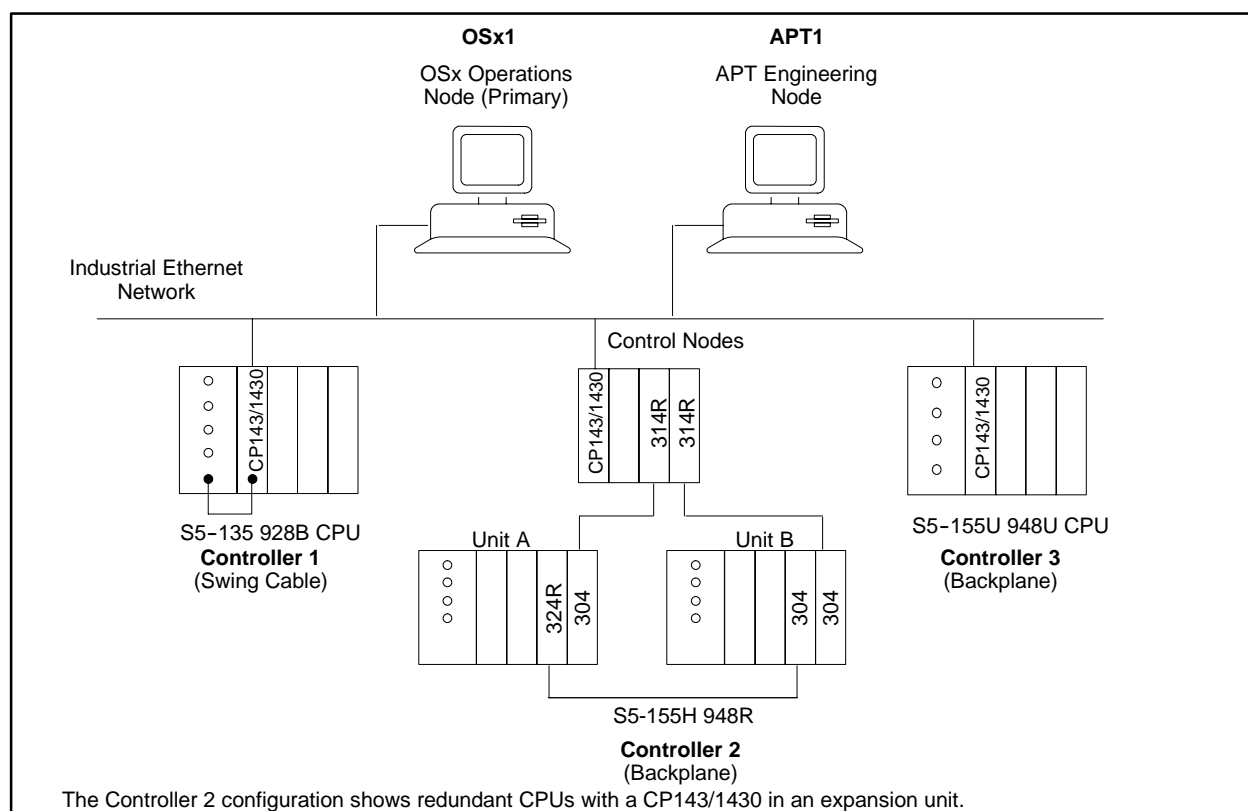


Figure 1-3 Data Transfer on OSX with S5 Controllers

Starting APT on a TISTAR Operator Station (Series 505)

If you are using a Series 505 controller and a TISTAR Operator Station, you can use the following procedure. TISTAR 1.x and 2.x with TIWAY communications do not support S5 controllers.

1. Select **Programming Utilities** from the TISTAR Main Menu.
2. Enter your user identification and password.
3. Select **APT** from the Program Menu.

The title screen appears, and APT appears as shown in [Figure 1-1](#).

Starting Up APT (continued)

Starting APT on a DOS Computer Connected to a TISTAR System (Series 505)

If you are using a Series 505 controller and you have the APT computer connected to a TISTAR system, then you can do the types of data transfer that are listed below and shown in [Figure 1-4](#).

- Develop your application (controller) program with APT and then transfer tag information to the TISTAR system.
- Download and debug an application program to the controller through the TISTAR TIWAY link.

To start APT from a DOS computer configured as a TISTAR operator station, follow these steps.

1. From the DOS prompt, type `APT -T` and press `[Enter]`.
2. Enter your TISTAR user ID.

The title screen appears, and APT appears as shown in [Figure 1-1](#).

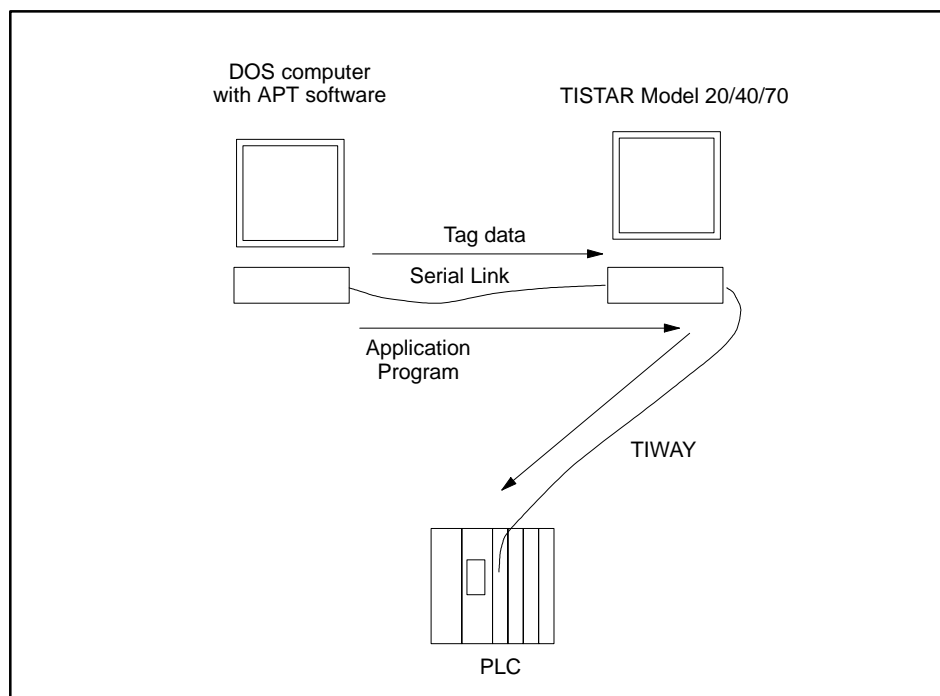


Figure 1-4 Data Transfer with APT as a TISTAR Operator Station

Configuring Colors

APT provides basic color schemes that you can use with your system. The colors that are available depend on your hardware configuration.

To configure the colors for your APT screens, follow these steps.

1. Under the **SYSTEM** label, use the arrow keys to highlight **CONFIG [] System configuration**
2. Press **F10** or **E** or **Enter**.

The screen appears as shown in [Figure 1-5](#). The letter that designates the current color configuration is listed as the default selection.

3. Enter the letter corresponding to your choice.

The cursor highlights **OKAY**.

4. Press **Enter**.

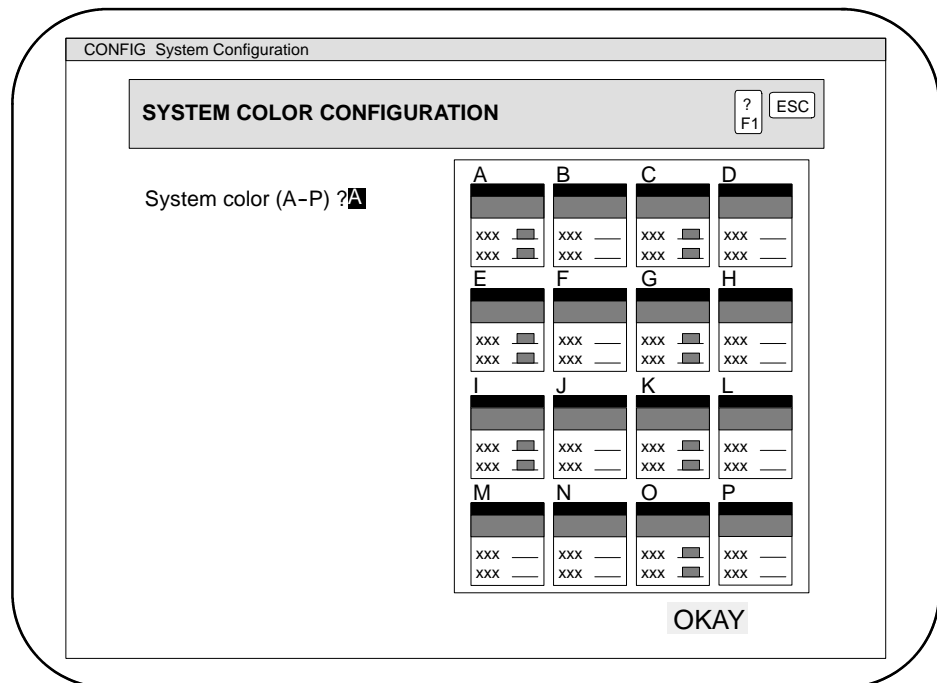


Figure 1-5 Color Configuration Dialog Window

1.2 Managing APT Screens

Screen Characteristics

All APT screens have the same general appearance. [Figure 1-6](#) identifies the specific features that characterize all APT screens.

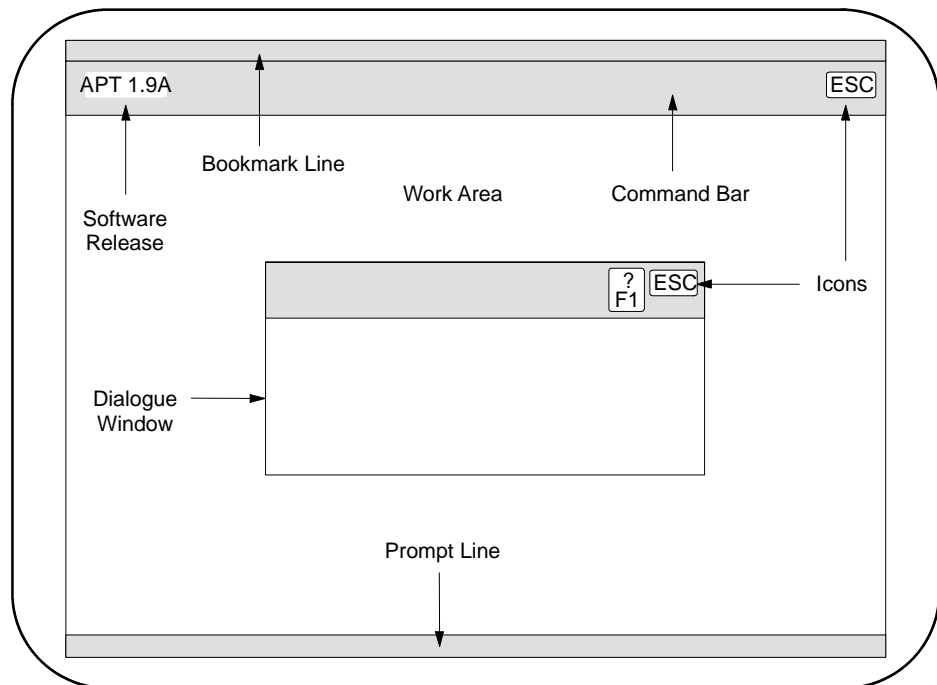


Figure 1-6 APT Screen Layout

Bookmark line indicates your location in the hierarchy and uses the following format: *Program_Name Unit_Name Active_Utility [description]*. The current date and time appear in the right portion of the bookmark line.

Command bar shows all of the controlling icons available for the active utility.

Work area contains the information you enter and edit in the active utility or editor.

Dialog window appears for certain commands and must be completed before APT accepts that command.

Prompt line shows instructions or system messages in response to actions performed in the work area.

Icons in the command bar provide special functions that help you move through the hierarchy and utilities.

Software release indicates the current software version that you are using.

APT Icons

In general, the command bar on the APT screen contains the icons illustrated in [Figure 1-7](#).

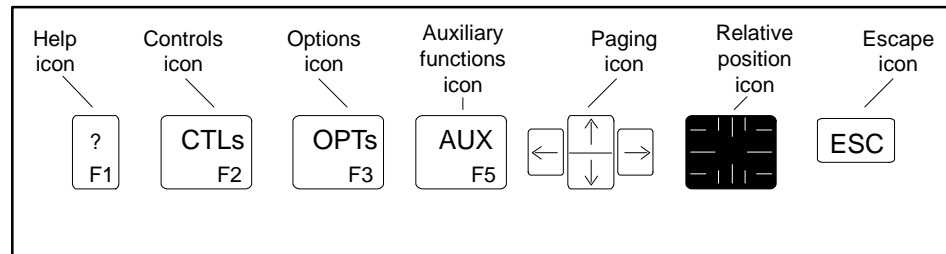


Figure 1-7 APT Icons

Help icon provides on-line help at all levels. When you either select the help icon or press `F1`, you see an index of the available utilities along with a sub-directory that corresponds to your activity level on the screen.

Controls icon allows you to access additional features in the APT system. When you either select the controls icon or press `F2`, a list of available actions appears, along with the keys you use to initiate each action. Only the items in black are available at the current text-cursor position.

Options icon allows you to access additional actions at your current level in the hierarchy or utility. When you either select the options icon or press `F3`, a list of available actions appears, along with the keys you use to initiate each action. Only the items in black are available at the current text-cursor position.

Auxiliary functions icon allows you to execute controller Auxiliary functions. This option is available only from Debug menus. For a more detailed explanation of the operation of the Auxiliary functions, refer to your controller manual and [Chapter 7](#).

Paging icon allows you to change the position of items in the work area. Clicking on the up or down arrows in the paging icon has the same effect as pressing the PgUp or PgDn keys; clicking on the left arrow has the same effect as pressing the Home key, and clicking on the right arrow has the same effect as pressing the End key. If you are not using a mouse, use the PgUp, PgDn, Home, and End keys, not the arrow keys.

Relative position icon gives an indication of your position inside a utility when the screen cannot display all of the available information. The icon fills in proportion to your edit activity on the screen. See [page 1-12](#).

Escape icon is always available within APT. Selecting this icon closes the current activity and moves you one step back in your sequence of actions.

Managing APT Screens (continued)

Using the Keyboard

APT defines each action with a designated key or set of keystrokes. Some function keys, listed in [Table 1-1](#), are common throughout APT. Other command keys are specific to a utility or editor but can be identified by selecting the **CTLs** or **OPTs** icon from the command bar.

Table 1-1 APT Function Keys

Function	Keys	Function	Keys
Help	F1	Exit	Esc
Controls	F2	Delete	Del
Options	F3	Insert	Ins
Show Completion Aids	F4	Page Up	PgUp
AUX ¹	F5	Page Down	PgDn
Cross Reference	F6	Page Left	Home
Split Screen	F7	Page Right	End
Unsplit Screen/Exit	Shift F7	Text Cursor Up	↑
Edit Next	F8	Text Cursor Down	↓
Edit Previous	Shift F8	Character Left	←
Mark Block	F9	Character Right	→
Edit/Expand ²	F10	Enter/Move to Next Field	Tab
Mark All Objects	Shift F9	Enter/Move to Previous Field	Shift Tab
Unmark All Objects	Alt F9		

1 This option appears in the Debug utility only.
 2 Function varies, depending on location of the cursor.

The block-shaped text cursor shown in [Figure 1-8](#) highlights an area of the screen and is under keyboard control. Use the arrow keys to move the text cursor to the position where you want to request an action or enter data.

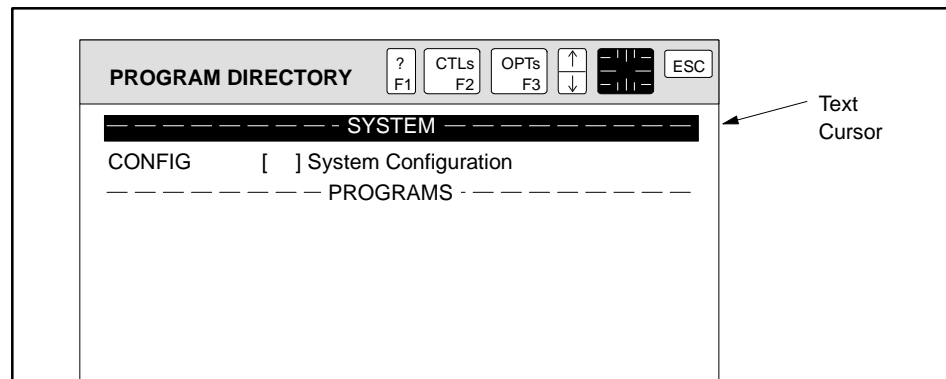


Figure 1-8 Using the Text Cursor

Using the Mouse

You can use a mouse or a trackball to move a cross-hair freely over any part of the screen. [Figure 1-9](#) shows the cross-hair as it appears when you are using the mouse or trackball.

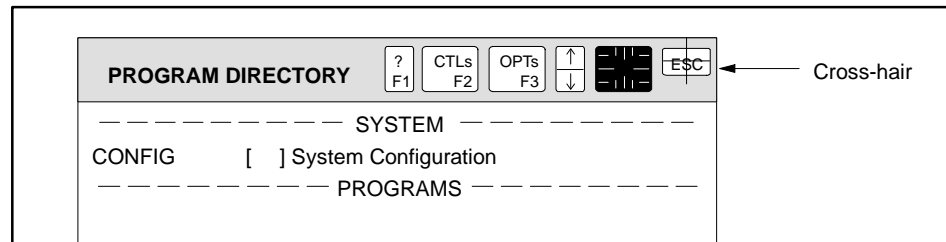


Figure 1-9 Using the Cross-Hair

[Figure 1-10](#) identifies the functions of a three-button mouse, which are explained below.

- The left button selects the item under the cross-hair by moving the text cursor to that item.
- The middle button marks an item under the cross-hair. When you draw lines in the graphical editors, the middle button also lets you change the direction of the line. Press both buttons of a two-button mouse for the middle button functions.
- The right button, or **F10**, selects and expands the item under the cross-hair.

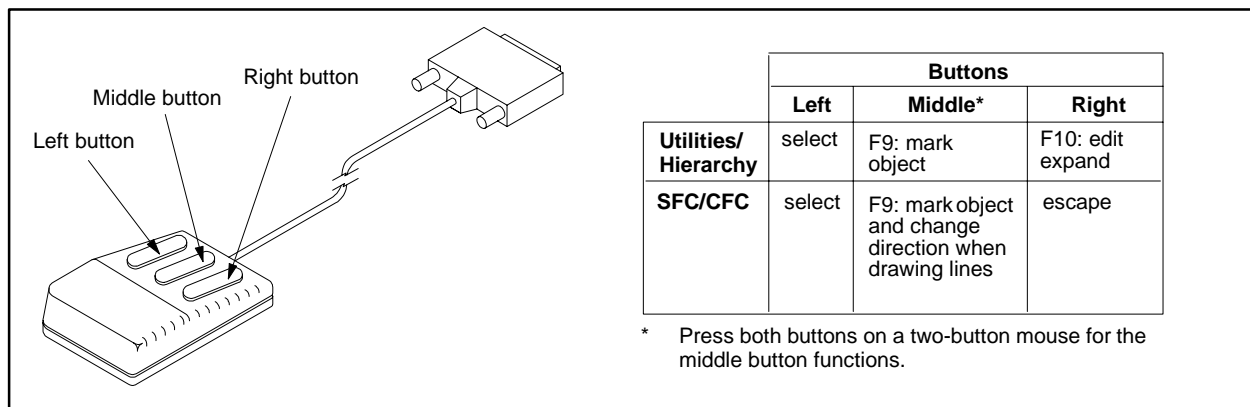


Figure 1-10 Using the Mouse

Managing APT Screens (continued)

Paging

The relative position icon indicates the portion of information that currently appears on the screen, as shown in [Figure 1-11](#).

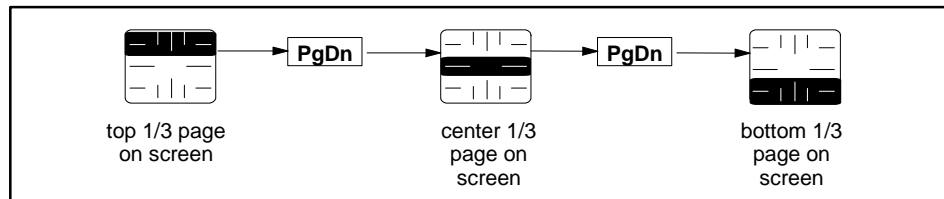


Figure 1-11 Relative Position Icon

APT provides several methods of paging through the screens to change the position of the items in your work area. [Table 1-2](#) shows the keys and icons that you use to determine the area of information on the screen.

Table 1-2 APT Screen Paging Keys

Key	Icon	Result
PgUp		Everything in the work area moves down to display the preceding page of information.
PgDn		Everything in the work area moves up to display the next page of information.
Home		Everything in the work area moves to the right, so that you display items to the left of the current work area.
End		Everything in the work area moves to the left, so that you display items to the right of the current work area.

Using a Split Screen

The split-screen function in APT allows you to view one section of your program while you work in another, and to move easily between the two areas.

When you are in a utility or in an editor, APT allows you to display the hierarchy on the bottom half of the screen. When the text cursor is in the hierarchy area, you can select another activity to run in the lower portion of the screen, as shown in [Figure 1-12](#). You cannot, however, show two directories at one time, or display the same object in both halves of the screen.

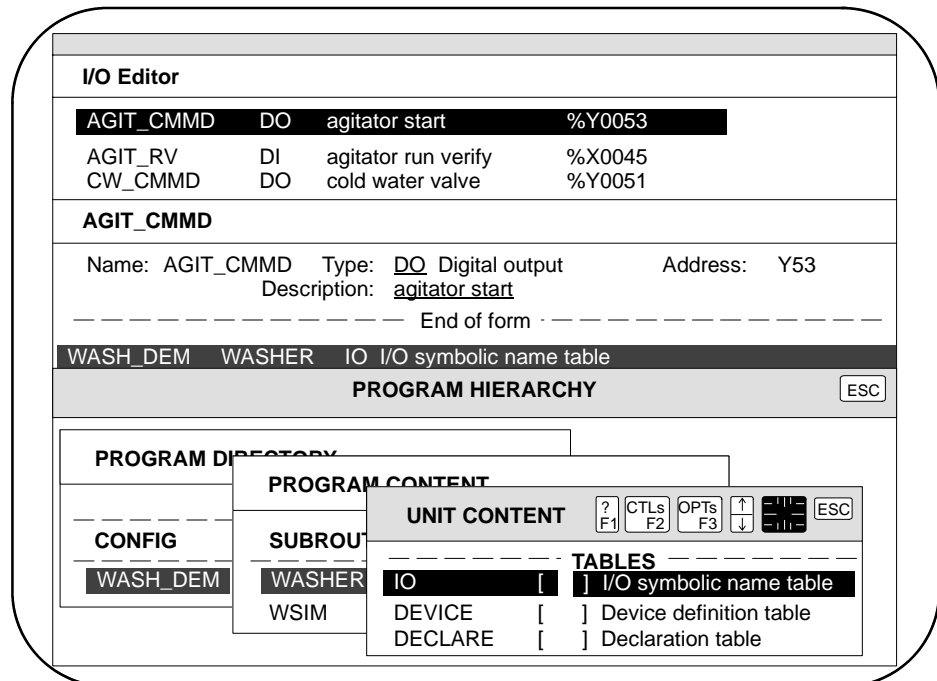


Figure 1-12 Using a Split Screen

When you operate with a split screen, only one half the screen is active, and you can work only in the active portion. The command bar in the inactive portion of the screen is white and shows no icons.

To split the screen, press **F7**. To activate the other half of the screen, press **F7** again. To return to a single screen showing only one half, press **Esc** from either half of the screen until you exit from the hierarchy. You can exit immediately from the split screen by pressing **Shift F7**.

1.3 Understanding the APT Structure

APT Hierarchy

APT has three levels of directories, shown in [Figure 1-13](#). The Program Directory, the top level in the hierarchy, contains system-wide information. The Program Content, the middle level, contains all the information for a single program. A Program Content directory is available for each program listed at the Program Directory level. The Unit Content, the lowest level, contains the logical sub-components of a program. A Unit Content directory is available for each unit listed in the Program Content directories.

[Chapter 2](#) explains the APT hierarchy in more detail.

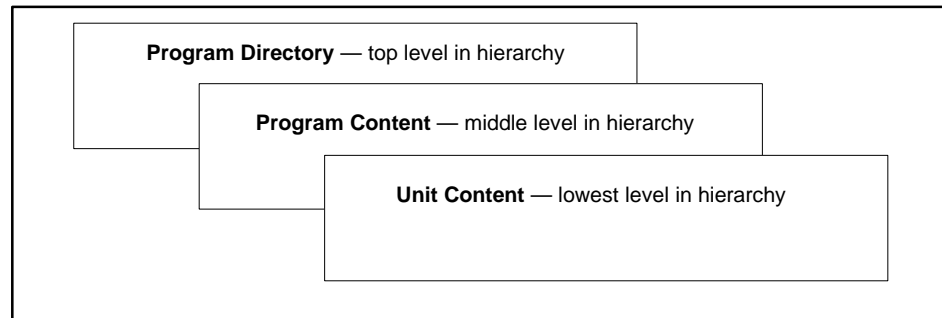


Figure 1-13 APT Program Hierarchy

APT Utilities

APT provides the following three types of utilities for you to use within the hierarchical structure.

Table Definition Utilities allow you to define information used in your APT program. This information includes I/O points, modules, devices, constants and variables (declarations), recipes, and subroutines. [Chapter 3](#) provides instructions for using the table definition utilities.

Programming Utilities allow you to design the program logic to solve your process control problem, and include two graphical languages and two textual languages. [Chapters 4](#) and [5](#) explain how to use the graphical and textual editors.

Operational Utilities allow you to compile, download, verify, debug, archive, restore, and translate your program. [Chapters 6, 7, and 8](#) describe how to perform these operations. When it is necessary to work directly with the program code, e.g., RLL and SFPGMs for Series 505, or STL for S5, you can call SIMATIC TISOFT or STEP 5 from APT. See [Chapter 8](#) of this manual.

The MAITT programming language enables you to test your APT programs in both batch and interactive modes.

1.4 APT Menus

Menu Windows

APT provides menu windows with a list of actions available at your current position. Three types of menus are available.

- The **CTLs** menu window lists actions dealing with screen management. These vary according to your position in the hierarchy but include splitting the screen, marking blocks, marking files, and expanding the sub-editors.
- The **OPTs** menu window lists actions to do on an APT table, unit, report, etc. These vary according to your position in the hierarchy but include actions such as editing, copying, deleting, etc.
- Completion aids are listed in menu windows available in the Table Definition Utilities. APT displays the available choices or subset of choices (depending on the table that you are currently viewing).

Figure 1-14 shows the **CTLs** menu that is available at the Program Content Level.

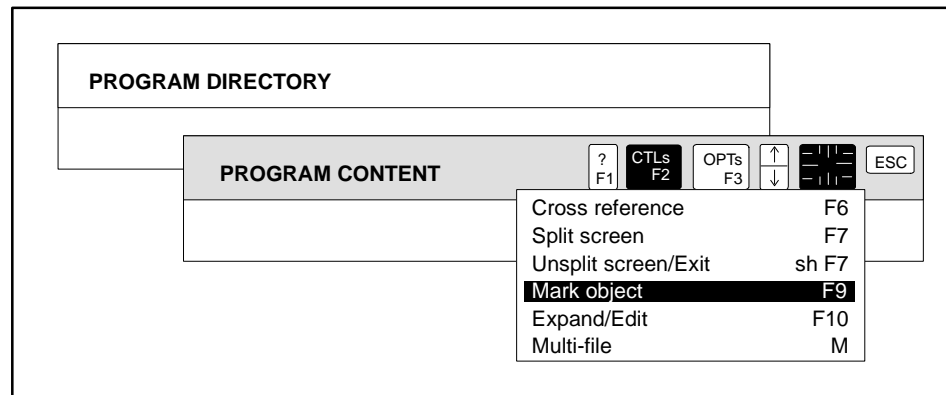


Figure 1-14 Using Menu Windows

The appearance of an action in the menu indicates whether or not it is a valid selection at this point in the hierarchy.

- The highlighted item in the menu is the default selection.
- Items printed in black are valid selections.
- Items that are gray cannot be initiated from your current position in the hierarchy.

APT Menus (continued)

Using Menus

APT provides alternate methods of accessing the **CTLs** and **OPTs** menus.

- Place the cross-hair on the **CTLs** or **OPTs** icon, and press the left button on the mouse.
- Press the function key corresponding to the menu you want to display: **F2** for the **CTLs** menu; **F3** for the **OPTs** menu.

APT also provides alternate methods of selecting actions from the menus:

- Press **Enter** to select the highlighted item in the menu.
- Place the cross-hair on another item in the menu, and press the left button.
- Use the arrow keys to highlight another item in the menu, and press **Enter**.
- Press the key or key-combination that is listed with the item in the menu.

NOTE: When you know the key or key-combination for an action, you do not need to open the menu window to select that action.

APT provides alternate methods to exit from the menu window:

- Press **Esc** to exit; your changes are saved automatically.
- Use the cursor to move down the menu, and select **OKAY**. (Your changes are saved.)

Using Completion Aids

Many fields in the APT forms have a limited number of valid entries. You can get help for entering data into these fields by selecting the **Completion Aids** option. APT displays the available choices, or an appropriate subset of the choices, depending on the table that you are currently viewing.

If completion aids are available, a black dot appears on the right of the field. To display the completion aids, place the mouse cursor on the dot and click the mouse **Select** (left) button. You can also move the cursor to the field and press **F4** or select the **Show Completion Aids** function on the CTLs **F2** menu. See [Figure 1-15](#).

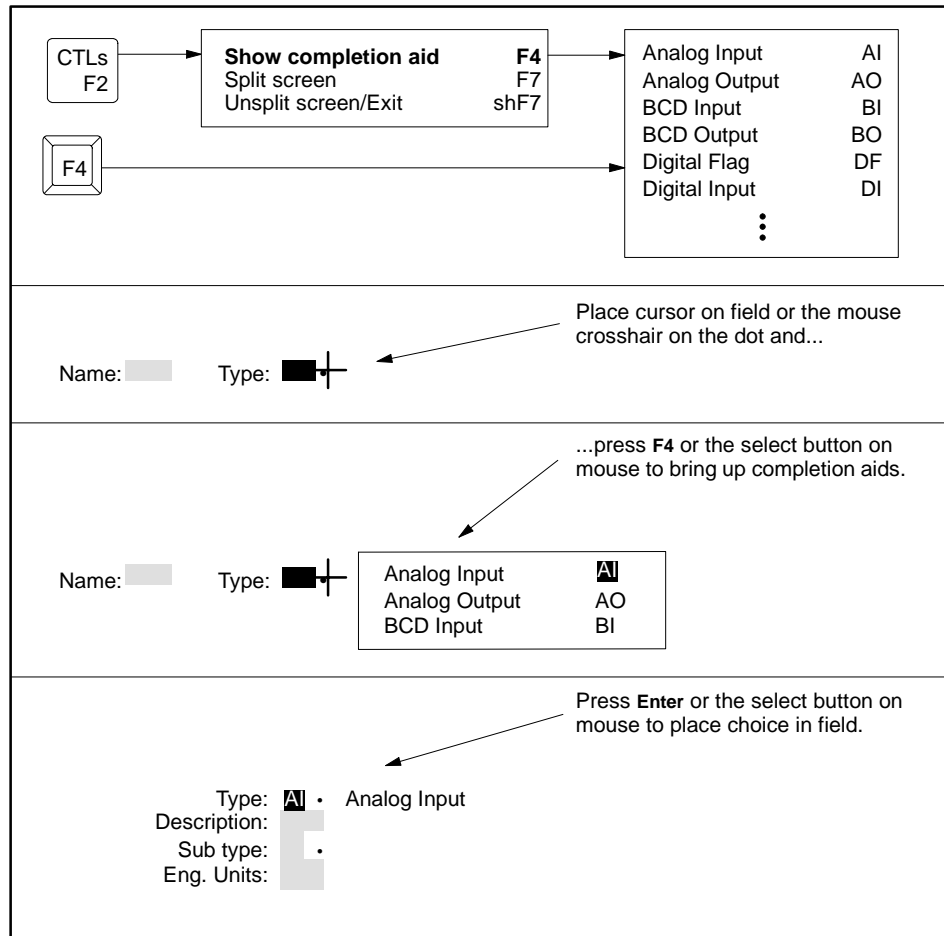


Figure 1-15 Displaying the Completion Aids

See [Chapter 3](#) for more information about accessing the completion aids.

1.5 Using On-line Help

Overview

On-line help is available at all levels of APT. On-line help provides a Main Index and a Sub-directory. When you access help from inside a utility, the index appears with your current position highlighted, as shown in [Figure 1-16](#).

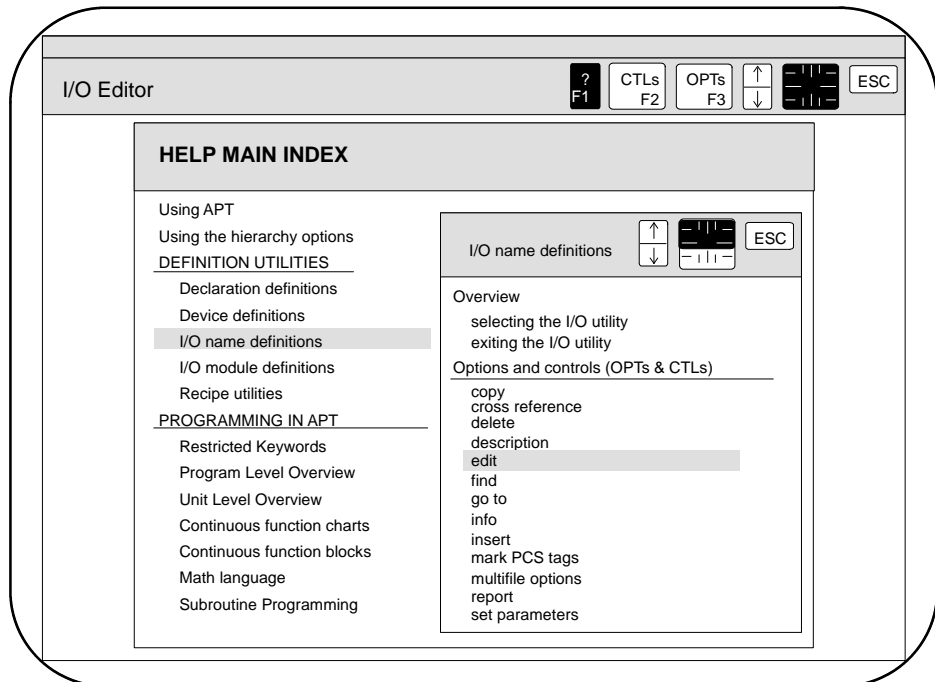


Figure 1-16 On-line Help

Accessing Help

To use the on-line help, follow these steps.

1. Press **F1**.
2. To access help on the highlighted topic, press **Enter**. To access help on a different topic, move the text cursor to the desired topic and press **Enter**.

To page through information on the selected topic, either use the paging icons or the PgUp/PgDn keys. (Clicking on the up or down arrows in the paging icon has the same effect as pressing the PgUp or PgDn keys.)

To move sequentially through the topics without returning to the index, use the **NEXT** and **PREV** icons; press **Esc** to return to the index and select another topic.

3. To exit the help screen, press **Esc** from each level of the index.

1.6 Exiting APT

Returning to the Operating System

To exit APT, follow these steps.

1. From the hierarchy, press **[Esc]** until the dialog window shown in [Figure 1-17](#) appears on the screen.
2. Press **[Y]** or use the arrow keys to select **Yes**.
3. Press **[Enter]**.

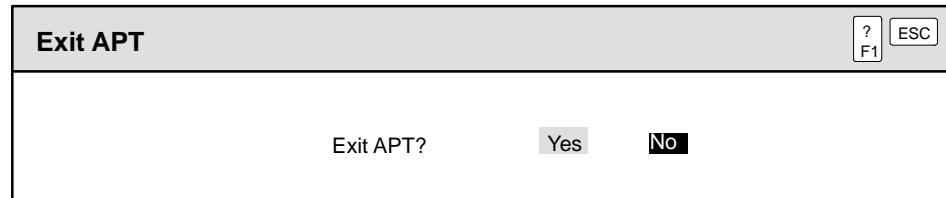


Figure 1-17 Exit Dialog Window

Changing DOS Configuration

APT requires a customized system configuration that may not be compatible with the other software on your system.

During the installation process, APT saves your original **config.sys** and **autoexec.bat** files so that you can use that configuration with other programs.

- To return to your original system configuration after exiting APT, type **DOSMODE** at the system prompt and press **[Enter]**.
- To change to the APT system configuration, type **APTMODE** at the system prompt and press **[Enter]**.

APT checks for the current mode and then reboots the system. Refer to [Appendix A](#) for more information about the **DOSMODE** and **APTMODE** commands.

NOTE: You do not have to change from **APTMODE** to **DOSMODE** to exercise DOS commands. If you can execute APT and all of your other DOS programs from one **config.sys** and **autoexec.bat**, then you do not need to execute **APTMODE** and **DOSMODE**.

Understanding the APT Hierarchy

2.1	Understanding the Directory Levels	2-2
	Program Directory	2-2
	Program Content	2-4
	Unit Content	2-6
2.2	Accessing Levels in the Hierarchy	2-8
	Moving Down the APT Hierarchy	2-8
	Accessing Another Program or Unit	2-9
	Moving Up the Hierarchy	2-9
2.3	Hierarchy Level Activities	2-10
	Using the Options Icon	2-10
	Adding a Program	2-12
	Adding a Unit	2-12
	Using Multi-File Command	2-13
	Archiving and Restoring	2-14
	Copying and Deleting	2-15
	Validating	2-15
2.4	Reports	2-16
	Using the Report Option	2-16
	Table Reports	2-17
	SFC and CFC Reports	2-18
	Object Code Reports	2-20
	Individual Cross Reference Reports	2-21
	Viewing Saved Reports	2-21

2.1 Understanding the Directory Levels

Program Directory The Program Directory Level shown in [Figure 2-1](#) is the highest level of the APT hierarchy and always contains two major headings: System and Programs.

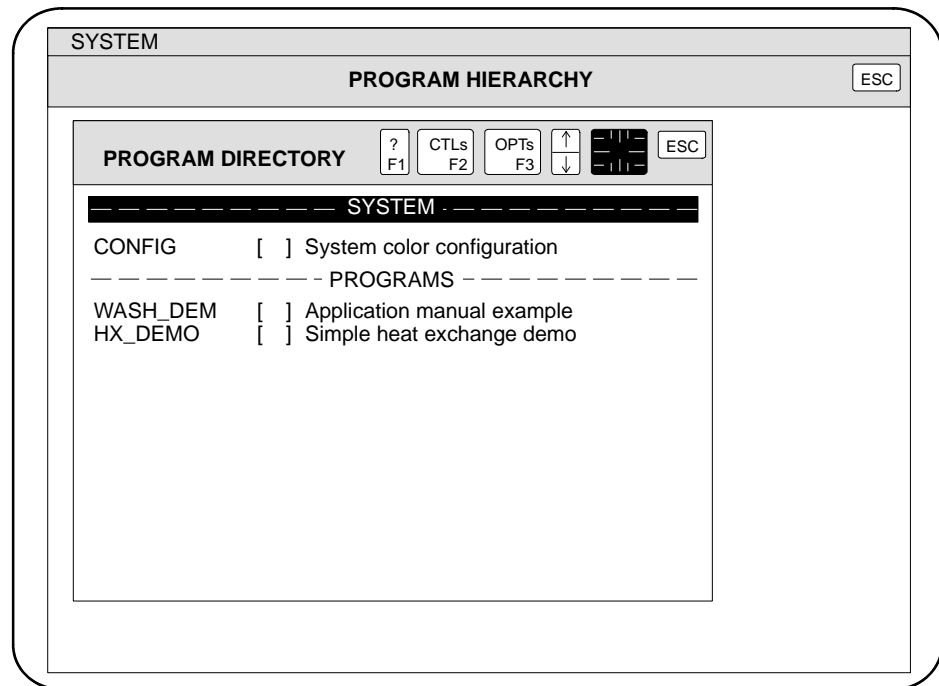


Figure 2-1 APT Program Directory Level

At the Program Directory Level, you can do the following:

- Configure the screen colors for your APT system.
- Work with entire programs (add, edit, copy, delete, etc.).
- Move into the Program Content Level for any of the programs listed.
- Access help information on any APT topic.
- Exit APT.

Figure 2-2 shows the menus that appear when you select the CTLs and OPTs icons in each section of this level. See Table 2-1 on page 2-10 for definitions of these functions.

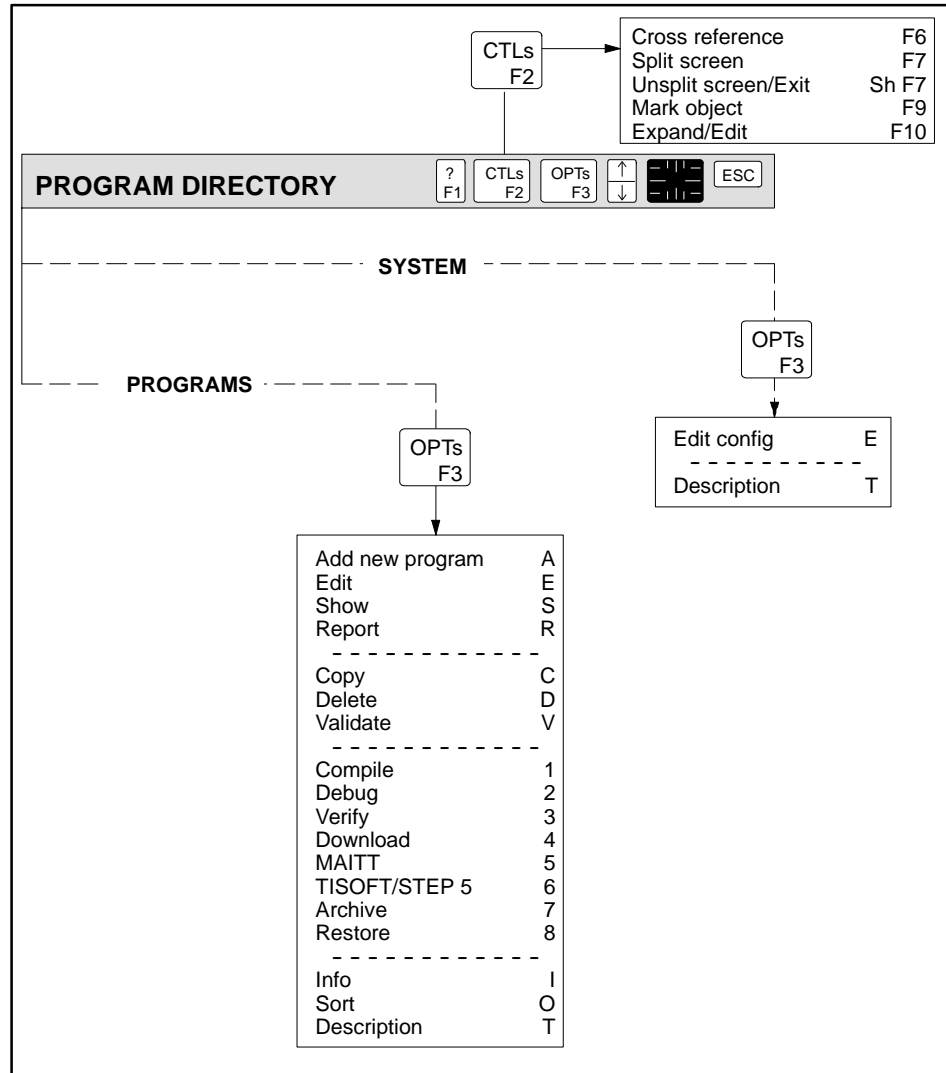


Figure 2-2 Program Directory Controls and Options

Understanding the Directory Levels (continued)

Program Content

The Program Content Level shown in [Figure 2-3](#) is the middle level of the hierarchy and always contains seven major headings: Compile, Tables, Units, Recipe Templates, SFCs, CFCs, and Saved Reports.

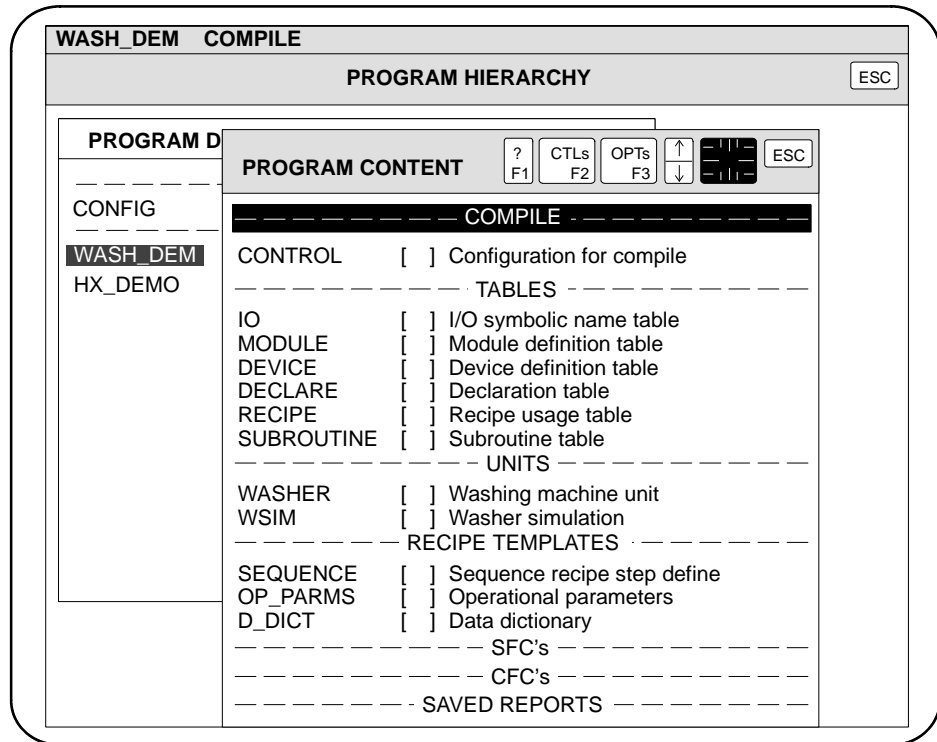


Figure 2-3 Program Content Level

At the Program Content Level, you can do the following:

- Configure the parameters used to compile the program.
- Work with the entire program (compile, debug, etc.).
- Work with tables, units, recipe templates, SFCs, and CFCs for the program (add, edit, copy, delete, etc.).
- Display, print, delete, and rename saved reports.
- Access help information on any APT topic.
- Move either into a Unit Content Level, or back to the Program Directory Level.

Figure 2-4 shows the menus that appear when you select the CTLs and OPTs icons in each section of this level. See Table 2-1 for definitions of these functions.

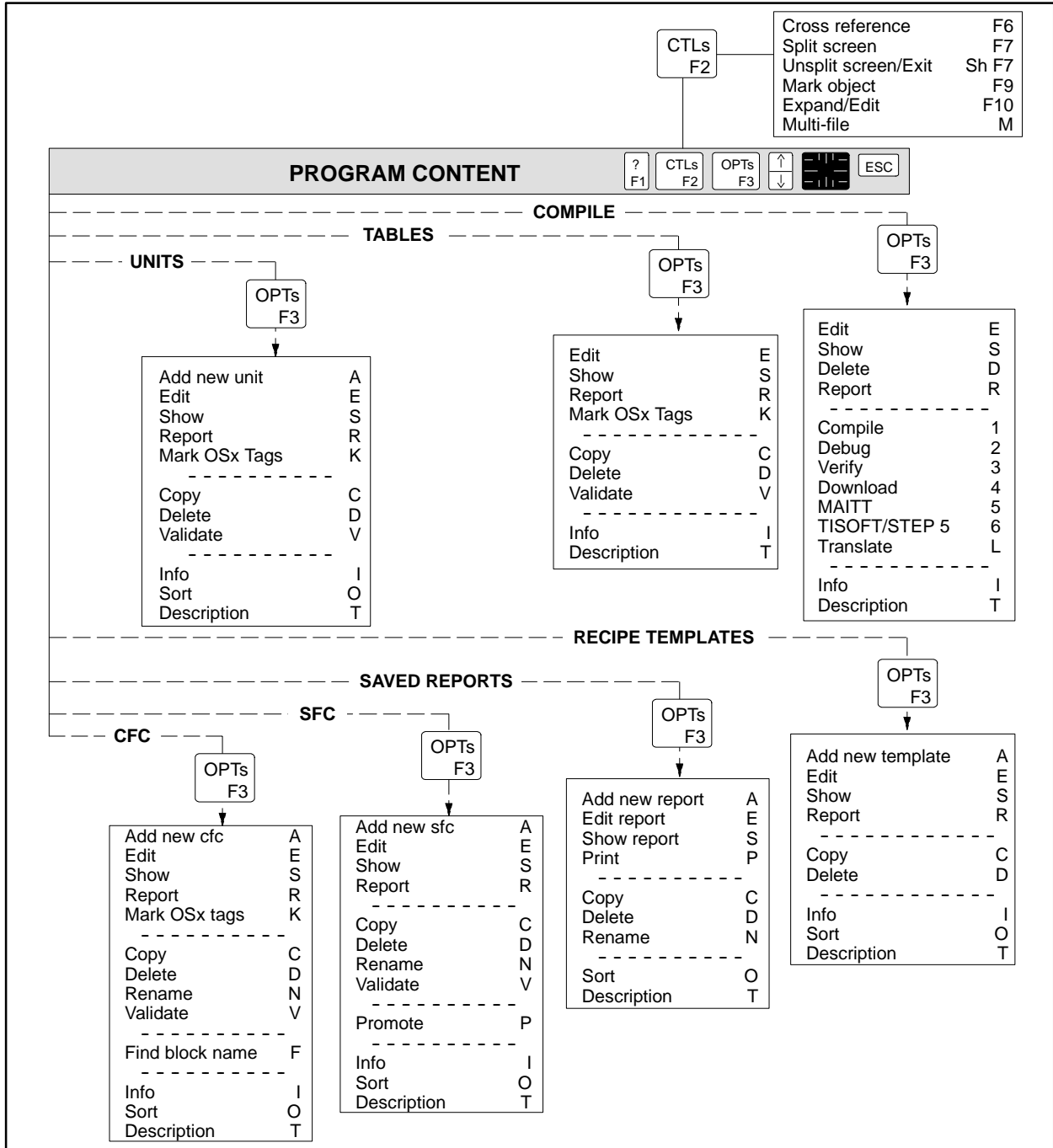


Figure 2-4 Program Content Controls and Options

Understanding the Directory Levels (continued)

Unit Content

The Unit Content Level shown in [Figure 2-5](#) is the lowest level in the hierarchy and always contains four major headings: Tables, SFCs, CFCs, and Saved Reports.

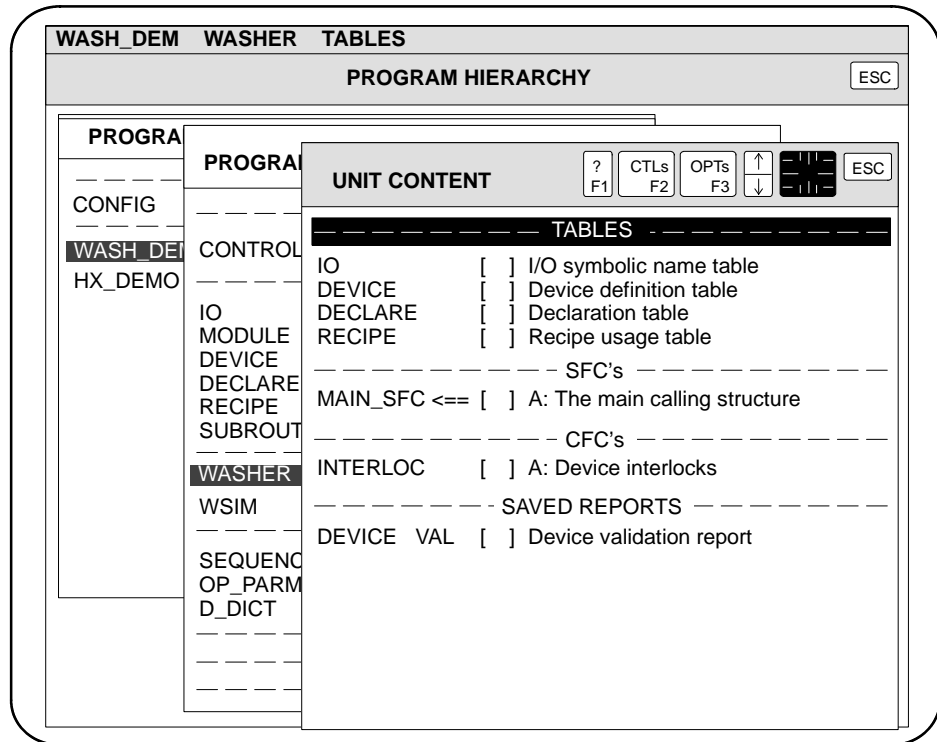


Figure 2-5 Unit Content Level

At the Unit Content Level, you can do the following:

- Work with tables, SFCs, and CFCs for the unit (add, edit, copy, delete, etc.).
- Display, print, delete, and rename saved reports.
- Access help information on any APT topic.
- Move back to the Program Content Level, or Program Directory Level.

Figure 2-6 shows the menus that appear when you select the CTLs and OPTs icons in each section of this level. See Table 2-1 for definitions of these functions.

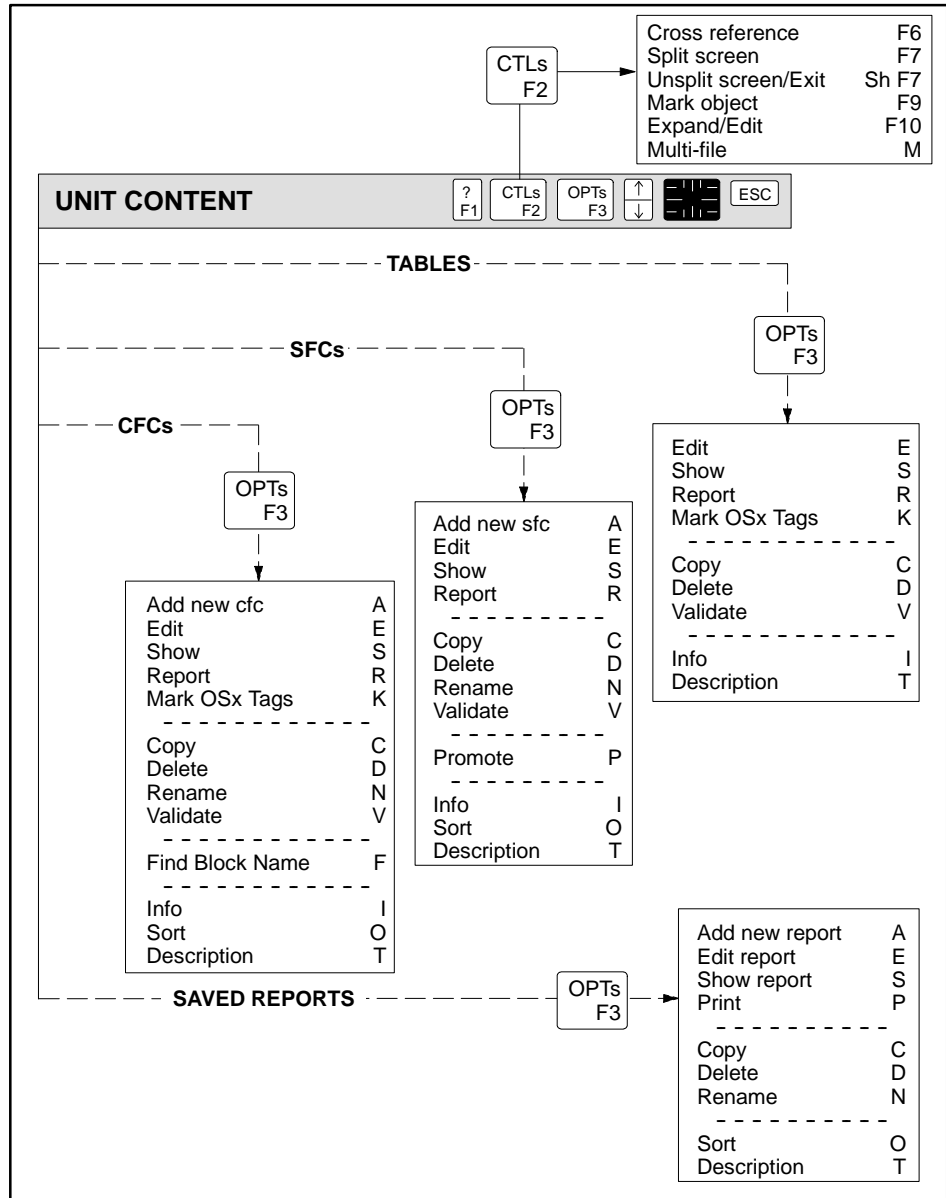


Figure 2-6 Unit Content Controls and Options

2.2 Accessing Levels in the Hierarchy

Moving Down the APT Hierarchy

You can move down the hierarchy from the Program Directory to the Program Content Level by following these steps.

1. Place the text cursor on the name of a program in the Program Directory as shown in [Figure 2-7](#).
2. To move to the Program Content Level, do any one of the following:
 - * With the cross-hair on the program name, press the right mouse button.
 - * Press **F10**, or use the **CTLs** icon to select **Expand/Edit**.
 - * Press **E**, or use the **OPTs** icon to select **Edit**.
 - * Press **Enter**. The **Enter** key can substitute for **F10** or **E**, except where it is specifically stated that you must use **F10**.

To move from the Program Content Level to the Unit Content Level, place the text cursor on the name of a unit and use any of the methods in Step 2.

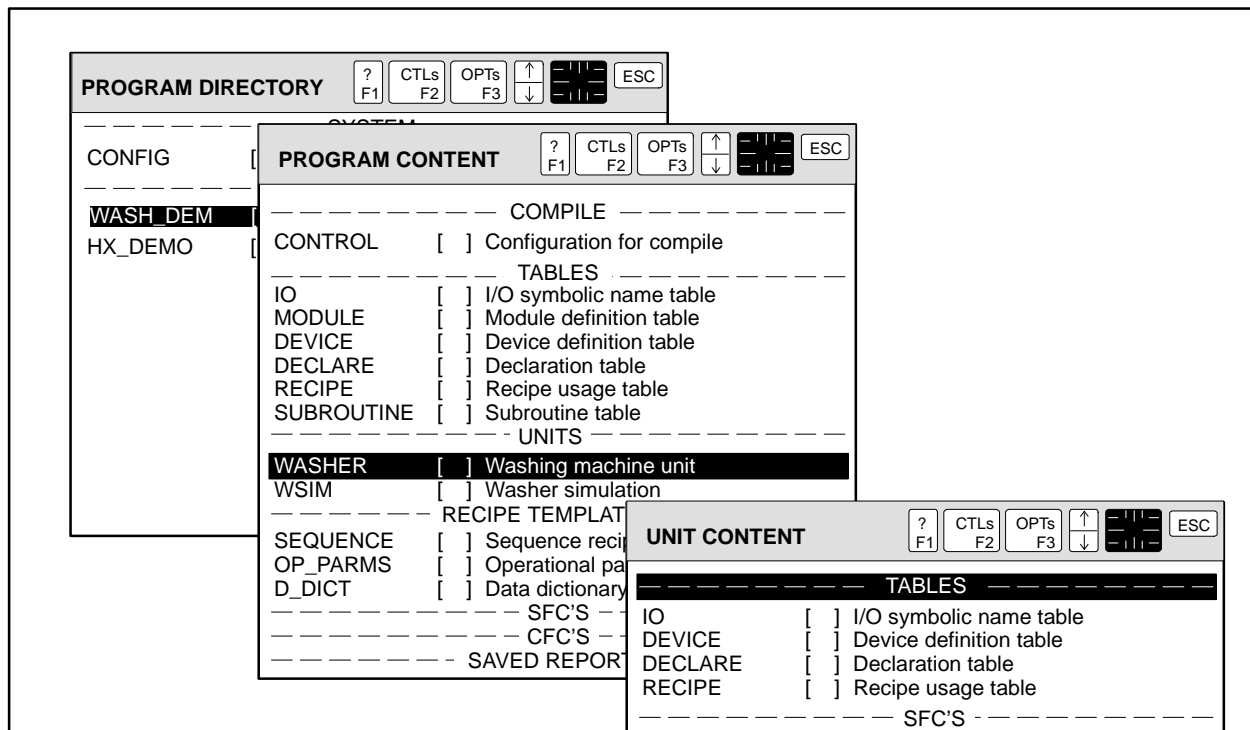


Figure 2-7 Moving Down the APT Hierarchy

Accessing Another Program or Unit

From any level of the hierarchy, you can directly access another level by using the mouse. From the Unit Content Level, for example, you can access another unit in the same program, or another program.

To access another unit or program:

1. Use the pointing device to position the cross-hair on the name of a unit or on the name of a program. See [Figure 2-8](#).
2. To move to that level of the hierarchy, press the left button; to expand (edit) the object under the cross-hair, press the right button.

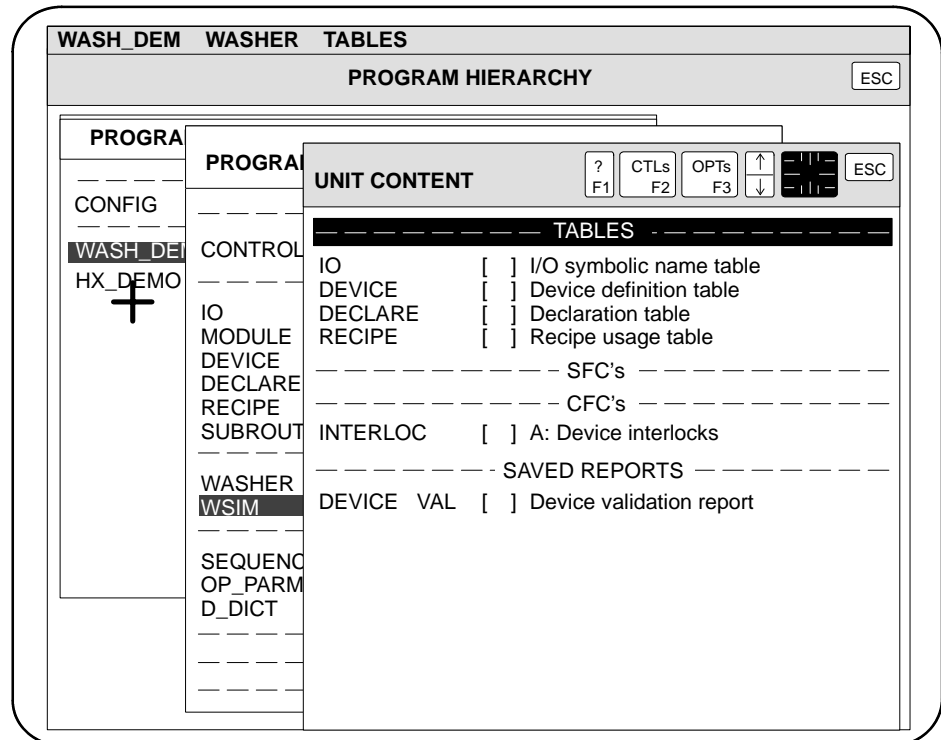


Figure 2-8 Selecting Another Program

Moving Up the Hierarchy

You can move from the Unit Content Level back to the Program Content Level, or from the Program Content Level back to the Program Directory Level, by pressing **Esc** or by selecting the **ESC** icon in the command bar of the current level. (Pressing **Esc** or selecting the **ESC** icon automatically saves your changes.)

2.3 Hierarchy Level Activities

Using the Options Icon

Table 2-1 lists the options available from the hierarchy and the predefined key that you can use to initiate each command. To use the **OPTs** icon when you are at the hierarchy level, follow these steps.

1. Press , or select the **OPTs** icon.

The command menu appears at the top of the screen. Only the commands in black are valid for the selected object.

2. Select a command from the menu, or press the associated key.
3. Follow the directions in the dialog window.

Table 2-1 Options at Hierarchy Level

Command	Key(s)	Object(s)/Remarks
Add New	<input type="text" value="A"/>	Program, Unit, SFC, CFC, Template, Report: Creates a new object in the hierarchy with name specified in the dialog window.
Archive	<input type="text" value="7"/>	Program: Creates a backup of the program on hard disk or floppy disk. See page 2-14 .
Compile	<input type="text" value="1"/>	Program, Compile: Compiles program and prepares it for execution in the controller. See Chapter 6 .
Copy	<input type="text" value="C"/>	Program, Table, Unit, SFC, CFC, Template, Report: Duplicates object under name specified in dialog window. See page 2-15 .
Debug	<input type="text" value="2"/>	Program, Compile, Object: Tests operation of a compiled program downloaded to the controller. See Chapter 7 .
Delete	<input type="text" value="D"/>	Program, Compile, Table, Unit, SFC, CFC, Recipe Template, Report, Object: Removes object (or contents of table) from the hierarchy and from the APT disk. Only archived objects can be recovered. See page 2-15 .
Description	<input type="text" value="T"/>	System, Program, Compile, Table, Unit, SFC, CFC, Template, Report, Object: Allows you to edit the description that appears in the hierarchy.
Download	<input type="text" value="4"/>	Program, Compile, Object: Sends a compiled program to the controller. See Section 6.5 .
Edit/Expand	<input type="text" value="E"/> or <input type="text" value="F10"/> or <input type="text" value="ENTER"/>	System, Program, Compile, Table, Unit, SFC, CFC, Template, Report: Allows you to edit the selected object.
Find Block	<input type="text" value="F"/>	CFC: Locates CFB inside CFC.

Table 2-1 Options at Hierarchy Level (continued)

Command	Key(s)	Object(s)/Remarks
Info	<input type="text" value="I"/>	Program, Compile, Table, Unit, SFC, CFC, Template, Object: Displays size information and date and time of last modification and compile.
MAITT	<input type="text" value="5"/>	Program, Compile, Object: Calls MAITT utility for interactive and batch mode testing of APT program. Refer to the SIMATIC APT MAITT Manual .
Mark OSx Tags	<input type="text" value="K"/>	Table, CFC, Unit: Allows you to mark APT names to translate to OSx.
Print	<input type="text" value="P"/>	Report: Sends saved report to printer.
Promote	<input type="text" value="P"/>	SFC: Sets up SFC as the main, or starting, SFC in a unit.
Rename	<input type="text" value="N"/>	Report, SFC, CFC: Allows you to change name of selected object.
Report	<input type="text" value="R"/>	Program, Compile, Table, Unit, SFC, CFC, Template, Object: Generates a report for selected object(s) and sends report to a file, a printer, or both. See Section 2.4 .
Restore	<input type="text" value="8"/>	Program: Reconstructs a previously archived program and all associated files on the APT system. See page 2-14 .
Show	<input type="text" value="S"/>	Program, Compile, Table, Unit, SFC, CFC, Template, Saved Report: Displays selected object.
Sort	<input type="text" value="O"/>	Program, Unit, SFC, CFC, Template, Saved Report, MAITT programs and macros: Reorders items alphabetically, by name or description.
STEP 5	<input type="text" value="6"/>	Program, Compile, Object: Calls S5's STEP 5 program for execution. Refer to the STEP 5 user manual (not included in the APT documentation).
TISOFT	<input type="text" value="6"/>	Program, Compile, Object: Calls Series 505's TISOFT program for execution. Refer to the TISOFT user manual (not included in the APT documentation).
Translate	<input type="text" value="L"/>	Compile, Object: Translates selected objects from APT format into OSx format. See Chapter 8 .
Verify	<input type="text" value="3"/>	Program, Compile, Object: Confirms that a downloaded program on a controller corresponds to the local program objects.
Validate	<input type="text" value="V"/>	Program, Compile, Table, Unit, SFC, CFC, Template: Checks the accuracy and completeness of the objects within the APT utilities.

Hierarchy Level Activities (continued)

Adding a Program

To add a program to the Program Directory, follow these steps.

1. In the Program Directory, highlight:
----- **PROGRAMS** -----
2. Press , or use the **OPTs** icon to select **Add new program**.
3. Enter a program name and description in the dialog window, and press .
4. Press again to complete the action.

Adding a Unit

To add a unit at the Unit Content Level, follow these steps.

1. At the Program Content Level, highlight:
----- **UNITS** -----
2. Press , or use the **OPTs** icon to select **Add new unit**.
3. Enter a unit name and description in the dialog window, and press .
4. Press again to complete the action.

Using Multi-File Command

The Multi-File command allows you to select multiple objects for certain operations. With the Multi-File command you can validate, report, and delete programs, tables, units, SFCs, CFCs, and recipe templates. You can also print and delete saved reports.

To use the Multi-File command, follow these steps from the Program Content or Unit Content Level.

1. Highlight an object in the hierarchy.
2. To mark the object, press **[F9]**, or use the **CTLs** icon to select **Mark object**.
3. When you have selected all the objects you want, press **[M]**, or use the **CTLs** icon to select **Multi-File**.

The dialog window appears as shown in [Figure 2-9](#).

4. Highlight the appropriate choice, and press **[Enter]**.
5. Press **[Enter]** again to confirm your selection.

To unmark the object, remove a highlighted **X** by pressing the spacebar. You can also toggle **[F9]** to unmark an object.

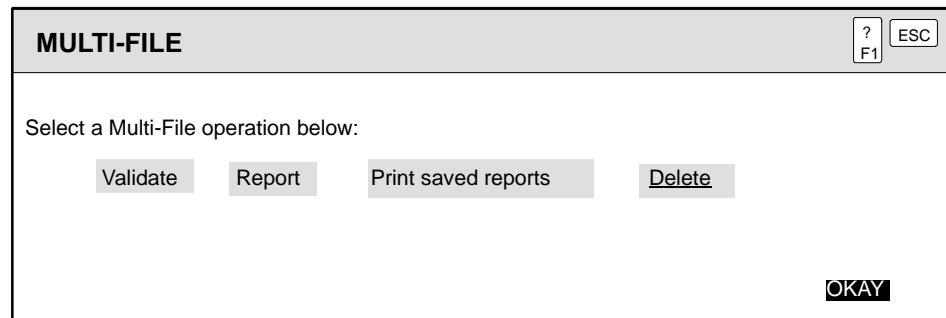


Figure 2-9 Using the Multi-File Command

Hierarchy Level Activities (continued)

Archiving and Restoring

The **Archive** command allows you to make a backup of your entire program from the Program Directory. You cannot, however, archive individual parts of an APT program. In order to save your program, you must use the Archive function. Archive organizes all the information in your program so that it can be restored to APT. The Archive command sends the selected program (and all its associated files) either to the hard disk or to one or more diskettes on your APT machine for off-line storage. This command does not delete or modify the program in the APT system. You can reduce the size of the archived file significantly if you archive the source code only. You can also reduce the size of the archive if you delete the object code that is created when you compile your program before you archive.

When you select the Archive option, APT prompts you for the following information.

Path Using DOS conventions, enter a path name such as A:\ to archive the program to a diskette in drive A:. To archive the program to the hard disk, enter a path (such as C:\ for the hard disk. APT accepts both the slash (/) and the backslash (\) in the path.

Program name Follow the standard DOS conventions for file names.

Source code Archive only the source code and the reports. This option creates a smaller archive file and takes less time to execute.

The **Restore** command allows you to reload an archived program file from disk storage. This command reconstructs the APT program to its form at the time of archival. Restoring a single program has no effect on any other program in the APT system.

CAUTION

When an APT program is archived, the archive name is identical to the program name. The RESTORE command deletes the existing program and replaces it with the restored program.

You can lose data if you have a program in APT with the same name as the archive you are restoring.

If you do not wish to overwrite your program, use the APT copy function and make a copy of your APT program under a different name, before you restore the archived program.

When you select the **Restore** option, APT prompts you for a path. To restore the program from a diskette, enter A:\ for the path. To restore the program from the hard disk, enter a path (such as C:\) for the hard disk. APT then prompts you for the program name. APT accepts both the slash (/) and the backslash (\) in the path.

If you are uncertain of the program name, press **F4** for a list of all archived programs located on the specified path.

You can archive/restore programs from DOS without running APT. Refer to [Section H.1](#) of this manual. To cancel a current archive operation, press **Esc**. APT may require several seconds to close files and process other overhead before you can continue your work.

Copying and Deleting

Use the **Copy** command to duplicate a program, unit, table, recipe template, SFC, CFC, or saved report. When you copy a program, APT duplicates all of the program components except the object file. When you copy a unit, APT duplicates all the unit components except the I/O addresses.

Use the **Delete** command to remove a program, unit, recipe template, SFC, CFC, saved report, object code, or the contents of a table from the hierarchy and from the APT disk.

CAUTION

APT keeps many files in DOS containing system and program information. If you delete an object from the hierarchy, you will not be able to recover it again unless you restore your program from an archive. Do not attempt to copy or delete files from DOS.

Validating

Use the **Validate** command to check the accuracy and completeness of the objects within the APT utilities. For example, you can use the validation operation to identify undeclared variables.

You can validate portions of your program as you build them. This is a more efficient procedure than validating the entire program, and is an easier way to identify errors when you compile the program.

The messages that are displayed during validation are listed in the appendix on error messages in the [SIMATIC APT Programming Reference Graphics/Math Manual](#).

2.4 Reports

Using the Report Option

The Report option allows you to create a report on an object. You press **R**, or use the **OPTS** option to select **Report**. You can also mark items with an **X** and use the **Multi-File** option to create multiple reports. A dialog window prompts you to specify the destination of the report to a file, a printer, or both, and displays other report options that are available. After you have specified the options that you need, the specifications become the defaults for any subsequent reports of the same type that you generate.

[Table 2-2](#) lists the reports that you can generate from within APT. The types of reports that you can create from the object file after you compile a program are listed on [page 2-20](#).

Table 2-2 Report Options

Report Source	Report Options
Program Directory	<ul style="list-style-type: none"> • Complete documentation for an entire program, based on the options that you select in the report menu.¹ • Marked tags for an entire program.
Program Content Level	<ul style="list-style-type: none"> • All program-level tables: I/O, Module, Device, Declare, Recipe, Subroutine.² • Complete documentation for an entire unit based on the options that you select in the report menu.¹ Report formats include screen image for tables, graphical overview, graphical subsection, and text for SFCs and CFCs. • All marked tags for the unit. • Marked tags for all units. • Documentation for a compiled program.
Unit Content Level	<ul style="list-style-type: none"> • All unit-level tables: I/O, Device, Declare, Recipe. • Selected details for SFCs and CFCs. • All marked tags for the unit.
<p>1 These reports can be extremely lengthy, depending on your choices in the report menus.</p> <p>2 For Series 505 controllers, select a module report that lists all the I/O points and their addresses. For S5 controllers, select an I/O report at the program level that lists all the I/O points and their addresses.</p>	

Table Reports

When you select the Report option for a table, a screen similar to the one in [Figure 2-10](#) is displayed. The Report option is a useful way to gather certain information; for instance, you can get an I/O report, or, if you have a Series 505 controller, a module report, that lists all your I/O points and addresses. You can create a report for any of the tables with the following format options.

Screen image For each object in the table, APT creates a report format based on the fields of the object editor as they appear when you are editing the object.

Tabular APT creates a format of up to twelve columns of information for each object in the table. Each column represents a field in the object editor and is user-specified. The default width of the report is 80 characters. If a greater width is required, APT automatically prints the report in a compressed mode, 132 characters wide.

Marked tags screens For each tag that has been marked in the table, APT creates a report format based on the fields of the Mark OSx Tags editor as they appear when you are marking tags.

APT automatically appends the following extension to the various reports.

- Screen image report - <table_name>.RPT
- Tabular report - <table_name>.TAB
- Marked tag report - <table_name>.TAG

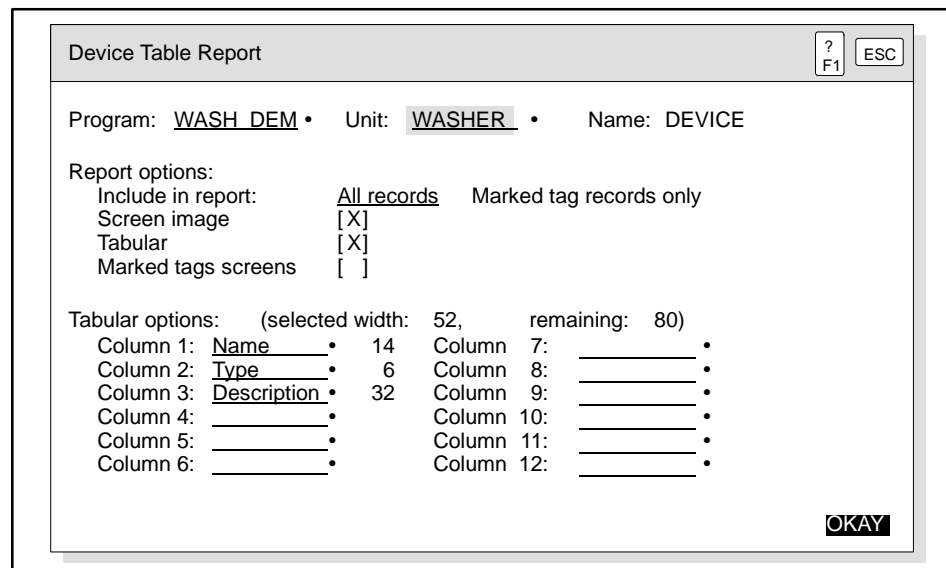


Figure 2-10 Device Table Report Options

Reports (continued)

SFC and CFC Reports

When you select the Report option for an SFC or CFC, a screen similar to the one in [Figure 2-11](#) (SFCs) or [Figure 2-12](#) (CFCs) is displayed. For SFCs you have the following format options.

Graphical overview APT creates a report format based on graphical display of the steps and transitions that comprise the SFC. This type of report can be sent to a printer, but cannot be saved and viewed from APT with the **Show** option.

Graphical subsection APT creates a report format based on a zoomed view of the graphical display of the steps and transitions. The size of the report and the number of pages required to print the report depends on the location of the step/transition graphical connections within the SFC editor. For example, when you place the step/transition connections in the upper left-hand corner of the editor, only one page is needed to print the report. When you place the step/transition connections in the lower right-hand corner of the editor, four pages are needed to print the report. This type of report can be sent to a printer, but cannot be saved and viewed from APT with the **Show** option.

Textual information APT creates a text-based report that describes each step and transition. You can sort textual information by cell number, graphic type, or label number. This type of report can be sent to a printer, or it can be saved and viewed from APT with the **Show** option.

SFC Report

Program: WASH_DEM•
Unit: WASHER•
SFC Name: MAN OP•

Report options:
Graphical overview []
Graphical subsection []
Textual information [X]

Subsection cell width 06 09 15 25 58

Textual information sorted by
Cell Number Graphic type Label number

OKAY

Figure 2-11 SFC Report Options

For CFCs you have the following format options.

Graphical overview APT creates a report format based on graphical display of the CFBs that comprise the CFC. This type of report can be sent to a printer, but cannot be saved and viewed from APT with the **Show** option.

Graphical zoomed views APT creates a report format based on a zoomed view of the graphical display of the CFBs. A zoomed view of each CFB is printed on a report page. This type of report can be sent to a printer, but cannot be saved and viewed from APT with the **Show** option.

Textual information APT creates a text-based report that describes each CFB. This type of report can be sent to a printer, or it can be saved and viewed from APT with the **Show** option.

Marked tags For each tag that is marked in the CFC, APT creates a report format based on the fields of the Mark OSx Tags editor as they appear when you are marking tags. This type of report either can be sent to a printer, or it can be saved and viewed from APT with the **Show** option.

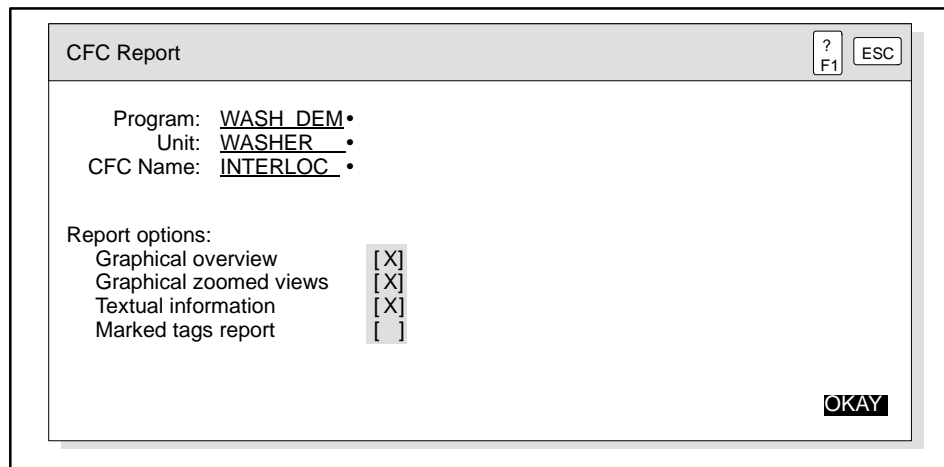


Figure 2-12 CFC Report Options

Reports (continued)

Object Code Reports

When you successfully compile a program, you can create reports from the Object File. APT provides a variety of different report options, listed below.

- APT symbol name to controller address: **symbol.rpt** lists the APT symbolic names in alphabetical order with the corresponding controller address for each.
- Controller address to APT symbol name: **address.rpt** lists all types of controller memory: C, K, V, S, etc., with the corresponding I/O symbolic name that is assigned to each address.
- SFC/CFC Math to controller address: **cfb.rpt** lists SFC Math blocks and CFBs with the corresponding addresses and any related SFPGMs.
- SFC cross-reference: **sfc.rpt** lists the main SFC with the related safe-state and subordinate SFCs.
- SFC steps and transitions: **steps.rpt** lists each step and transition with the corresponding control relay.
- Translated tags: **tag.rpt** lists the OSx tag types and information about the attributes (address and whether or not they are scanned).
- Cross reference: **xref.rpt** is a list of cross references for all objects defined in the program.
- TISOFT comment files: This report contains the files used by TISOFT to put comments in the ladder logic. This report can only be saved to a file for use by TISOFT and cannot be printed or viewed from APT.
- TISOFT synonym files: This report contains the files used by TISOFT to put synonyms in the ladder logic. This report can only be saved to a file for use by TISOFT and cannot be printed or viewed from APT.
- STEP 5 symbol files: This report contains the files used by STEP 5 to put synonyms in STL. This report can only be saved to a file for use by STEP 5 and cannot be printed or viewed from APT.

The STEP 5 symbol file is stored in the **apt\program\program_name\pr** subdirectory. The file name is the first six characters of the APT program name, followed by **z0.seq**. For instance, if your APT program is TEST1, then the name of the symbol file is **test1@z0.seq**. To use this symbol file in STEP 5, copy it from APT to the STEP 5 working directory, and then invoke it from the assignment list. You need to set your symbol length to 24 and your comment length to 30 for compatibility with APT.

Individual Cross Reference Reports

You can select an individual object or direct memory address from anywhere within APT, do a cross reference for the object or direct memory address, and save the results as a report. The report format is based on which cross reference option you select, **Referenced by** or **References**. The report name is *object_name.xrf*, where *object_name* is the name of the object you select.

NOTE: You can do a cross reference only on objects that have been validated. Both the **Validate** option and the **Compile** option (after Phase 3 of the compile process is completed) will validate objects.

Viewing Saved Reports

All reports made are listed under the saved reports section of APT. These reports are also stored in the `apt\program\<program_name>\pr` subdirectory in DOS. In the Saved Reports section, you can:

- Add a new report.
- Edit a report.
- Show (display) a report.
- Print a report.
- Copy, delete, or rename a report.

When displaying a report using **Show**, you do not get any editing capabilities. When editing a report you can add and delete lines, split and join lines, and find and replace text. You can also insert a block from a buffer which enables you to put Math blocks from SFCs and CFCs in a report.

Using Table Definition Utilities

3.1	Getting Started with Table Definition Utilities	3-2
	Overview	3-2
3.2	Defining I/O for a Series 505 Controller	3-4
	Defining Modules	3-4
	Defining the I/O Points	3-8
3.3	Defining I/O for an S5 Controller	3-10
	Accessing the I/O Table	3-10
	Defining the I/O Points	3-12
3.4	Editing I/O, Device, Declaration, Recipe, Subroutine Tables	3-14
	Accessing a Table	3-14
	Entering Data into a Table	3-16
	Entering Data into an Array	3-19
	Entering Data into a Recipe	3-20
	Entering Subroutine Math	3-22
3.5	Building a Recipe Template	3-24
	Adding a Template	3-24
	Editing a Template	3-24
3.6	More about the Controls (F2) and Options (F3)	3-26
	Setting Parameters (Alt P)	3-26
	Selecting Completion Aids	3-27
	Using the Cross Reference (F6)	3-28
	Copying Selected Objects (Alt C)	3-30
	Moving Selected Objects (Alt M)	3-31

3.1 Getting Started with Table Definition Utilities

Overview

The APT table definition utilities allow you to define I/O symbolic names, modules, devices, declarations (constants, variables, and arrays), recipes, and subroutines.

You define modules (Series 505 controllers only), the template for recipes, and subroutines only at the Program Content Level. You define I/O names, devices, declarations, and specific recipes at both the Program Content and Unit Levels. [Figure 3-1](#) illustrates the location of the tables and templates in the hierarchy.

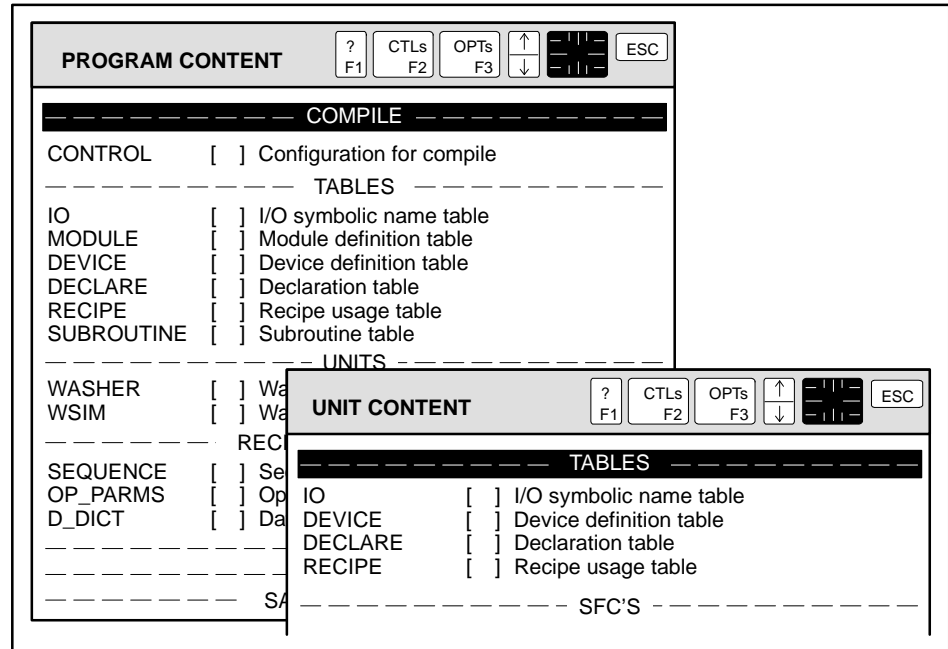


Figure 3-1 Availability of Table Definition Utilities

Except for the module editor, which you use only for Series 505 controllers, the table editors are composed of two sections (Figure 3-2). The upper section shows the objects that have been defined in the table. The lower section contains the fields for information specific for each object. In the device table editor, for example, the upper section is a list of devices. The lower section contains fields for device name, type, description, commands, etc., for each device. Details about using the other table editors are given in Section 3.4.

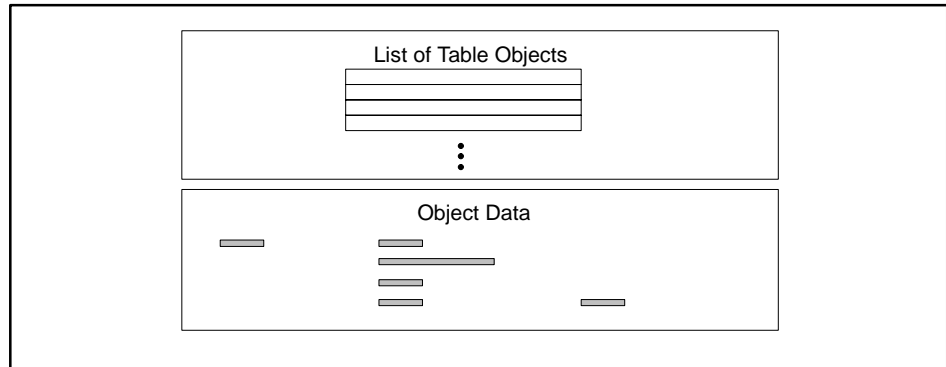


Figure 3-2 Table Sections for All Other Table Editors

For Series 505 controllers, you must define the I/O modules in the module editor. In the three sections of the module table editor (Figure 3-3), you enter the module type, I/O address, unit and I/O names, and I/O descriptive data. Details about using the module editor are given in Section 3.2. For configuring S5 I/O, refer to Section 3.3.

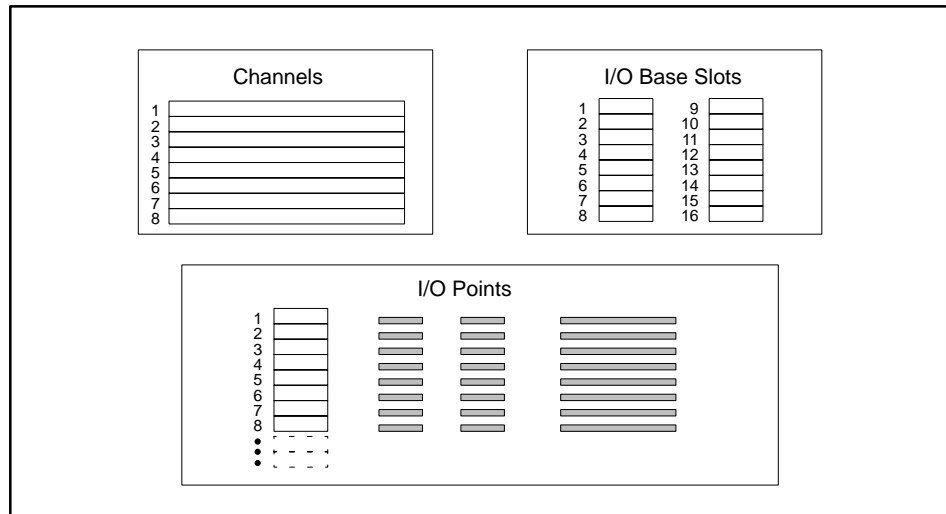


Figure 3-3 Table Sections for the Module Editor

3.2 Defining I/O for a Series 505 Controller

Defining Modules

To define an I/O module for a Series 505 controller, follow these steps.

1. At the Program Content Level, highlight: **Modules []**
2. Press **F10** or **E** or **Enter**.

The channel/base and slot fields appear (Figure 3-4). The active window (outlined with a thick border) is on the left, showing channel and base numbers.

3. Use the mouse or the arrow keys to place the cursor (hollow square) on the channel/base combination for configuration.
4. Press **F10** or **Enter** to edit the selected base. The window on the right, showing individual slots for the base, becomes the active window.
5. Use the mouse or arrow keys to select the appropriate slot.

The screenshot displays the 'MODULE Module definition table' interface. The 'Module Table' on the left shows a grid with 'Channel' (0-8) and 'Base' (0-15). A thick border highlights the area around channel 1 and base 1. The 'I/O Series' window on the right shows a list of slots (1-16) with their types and addresses. Slot 9 is highlighted with a thick border, showing type 'AO' and address '65'. Two pop-up menus are visible: one for 'Show completion aid' (F4) and another for 'Delete' (alt D) and other actions like 'Move', 'Undo', 'Find', etc.

Channel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		<input type="checkbox"/>														
2																
3																
4																
5																
6																
7																
8																

Slot	Typ	Addr	Slot	Type	Addr
1	8DI	1	9	AO	65
2	8DO	9	10		
3	8AI	17	11		
4	8AO	25	12		
5	8DO	33	13		
6	8DI	41	14		
7	8DO	49	15		
8	8AI	57	16		

Show completion aid F4
 Cross Reference F6
 Split screen F7
 Unsplit screen/Exit sh F7
 Edit channel/base F10

Delete alt D
 Move alt M
 Undo alt U
 Find alt F
 Go to alt G
 Change I/O series alt C
 Specify slots alt S

 Set parameters alt P

Figure 3-4 Defining Modules

-
- If you know the module name you want, you can type it in and press **Enter**. Alternatively, you can select a module name from the list by bringing up the Completion Aid window:

Press **F4** to display the Completion Aid window. Use the arrow keys to select the desired module name, and press **Enter**.

or

Use the select button of the mouse to bring up the Completion Aid window. Select the desired module from the names listed.

- After you enter the module type, APT moves the cursor into the Address field and displays the address of the next available I/O eight-point boundary. You can either accept the default address or enter a different address, and then press **Enter**.
- Verify that the configured I/O matches the physical I/O. Use **Alt S** to change the slot width if necessary.

WARNING

APT cannot verify that the I/O configuration you specified matches the physical I/O connected to the controller.

An I/O mismatch can cause unpredictable operation that could result in death or serious injury to personnel, and/or damage to equipment.

Ensure that your I/O configuration matches the physical I/O prior to placing the controller in RUN mode.

- You can either use **Esc** or **Shift F7** to exit the module editor, or you can press **F10** to define the I/O points (described on [page 3-8](#)).

Defining I/O for a Series 505 Controller (continued)

APT creates an icon for each base you enter into the table. For base #1 of channel #1 in [Figure 3-4](#), for example, the icon shows modules in slots 1-9.

[Figure 3-5](#) shows the relationship between the icon and the base.

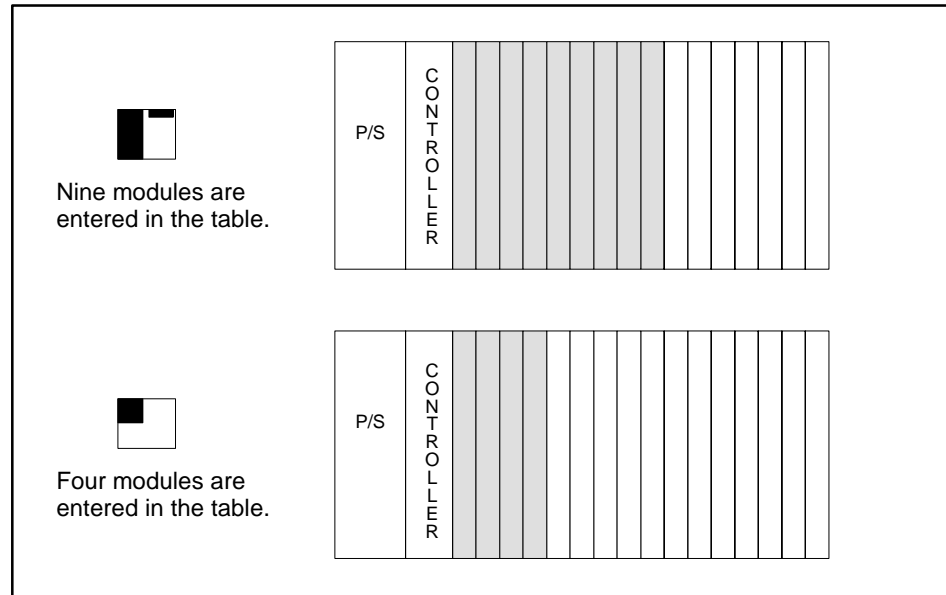
























Figure 3-5 Base Slot Icon

You can change the I/O Series from Series 500 to Series 505, and vice versa. Use **Alt C** to change the I/O Series 500 to Series 505, and select the series that you want. APT reports any incompatible modules when you select this option and prompts you about how to delete them. If you select **Yes**, the modules are removed from the table. If you select **No**, the I/O Series change is aborted and no changes are made in the channel/base status.

Table 3-1 lists the options and controls available for defining modules.

Table 3-1 Controls (F2) and Options (F3) for Defining Modules

Icon	Command	Keys	Function
F2 (CTLs)	Show Completion Aids		Displays a list of valid choices for a specific entry field.
	Cross Reference		Lists objects that are referenced by the selected object, and lists objects that reference the selected object. See Section 3.6 .
	Split Screen		Divides the screen in two, allowing you to display separate areas of APT functions.
	Unsplit Screen/Exit	 	Undoes the split screen option. If the screen is already united, prompts you to exit the table.
	Edit Channel/Base Edit Module		Depending on current location of the cursor, shifts cursor to the slot window or the I/O point window.
F3 (OPTs)	Delete	 	Deletes item currently highlighted.
	Move	 	Allows you to move a module with all underlying I/O names from one location to another.
	Undo	 	Cancels action before you press the ENTER key when making an entry in a table.
	Find	 	Locates the item you specify in the dialog window.
	Go to	 	Pages the screen to display the portion of the table (TOP or BOTTOM) that you select in the dialog window.
	Change I/O Series*	 	Changes I/O from Series 500 to Series 505 or from Series 505 to Series 500.
	Specify Slots*	 	Allows you to allocate the number of I/O slots to be reserved for a module.
	Set Parameters	 	Sets system to provide completion aids for each applicable entry and to change paging parameters. See Section 3.6 .
* This option is valid for all Series 505 controllers except the 545L (545 Lite).			

Defining I/O for a Series 505 Controller (continued)

Defining the I/O Points

To define the I/O points and complete the definition of a module, refer to [Figure 3-6](#) and follow these steps.

1. While the module table editor is open, place the cursor on the slot that you want to edit and press **F10**.
2. Enter the I/O symbolic name and other necessary information for each address that you want to access.
3. Use **Esc** or **F10** to exit the I/O point editor.
4. When you have finished with the module definition table, use **Esc** or **Shift F7** to exit the module editor.

WASH_DEM MODULE Module definition table

Module Table

Channel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1																
2																
3																
4																
5																
6																
7																

Channel: 1 Base: 1 Slot: 6 8DI

Slot	Type	Addr	Slot	Type	Addr
1	8DI	1	9	8AO	65
2	8DO	9	10		
3	8AI	17	11		
4	8AO	25	12		
5	8DO	33	13		
6	8DI	41	14		
7	8DO	49	15		
8			16		

Pos	Address	Description
1	WX 0001	
2	WX 0002	
3	WX 0003	
4	WX 0004	H2O_IN_TEMP
5	WX 0005	H2O_OUT_TEMP
6	WX 0006	
7	WX 0007	
8	WX 0008	





















Action	Key
Show completion aid	F4
Cross Reference	F6
Split screen	F7
Unsplit screen/Exit	sh F7
Exit to slot window	F10

Action	Key
Delete	alt D
Validate	alt V
Undo	alt U
Find Next	alt N
Mimic line above	alt M
Go to	alt G
Set parameters	alt P

Figure 3-6 Defining I/O Points

Table 3-2 lists the options and controls available for defining I/O points.

Table 3-2 Controls (F2) and Options (F3) for Defining I/O Points

Icon	Command	Keys	Function
F2 (CTLs)	Show Completion Aids		Displays a list of valid choices for a specific entry field.
	Cross Reference		Lists objects that are referenced by the selected object, and lists objects that reference the selected object. See Section 3.6 .
	Split Screen		Divides the screen in two, allowing you to display separate areas of APT functions.
	Unsplit Screen/Exit	 	Undoes the split screen option. If the screen is already united, prompts you to exit the table.
	Exit to Slot Window		Shifts cursor to the slot window from the I/O point window.
F3 (OPTs)	Delete	 	Deletes item currently highlighted.
	Validate	 	Checks the accuracy and completeness of the objects within the APT utilities.
	Undo	 	Cancels action before you press the ENTER key when making an entry in a table.
	Find Next	 	Locates next occurrence of the item last specified in the Find option initiated in the module-define editor.
	Mimic Line Above	 	Places a copy of the information in the preceding line into the current line.
	Go to	 	Pages the screen to display the portion of the table (TOP or BOTTOM) that you select in the dialog window.
	Set Parameters	 	Sets system to provide completion aids for each applicable entry and to change paging parameters. See Section 3.6 .

3.3 Defining I/O for an S5 Controller

Accessing the I/O Table

To define the I/O for an S5 controller, you must enter the I/O in the I/O Symbolic Name Table. To access the I/O symbolic name table editor, follow these steps.

1. Highlight the table in the Program Content or Unit Content Directory:

I/O Symbolic Name Table []

2. Press **F10** or **E** or **Enter**.

After you access the I/O symbolic name table editor, APT displays a list of defined I/O. If you have not defined any I/O, APT displays a single line, as shown in [Figure 3-7](#).

3. You can use either **Esc** or **Shift F7** to exit the editor, or you can press **F10** or **Enter** to enter I/O data (described on [page 3-12](#)).

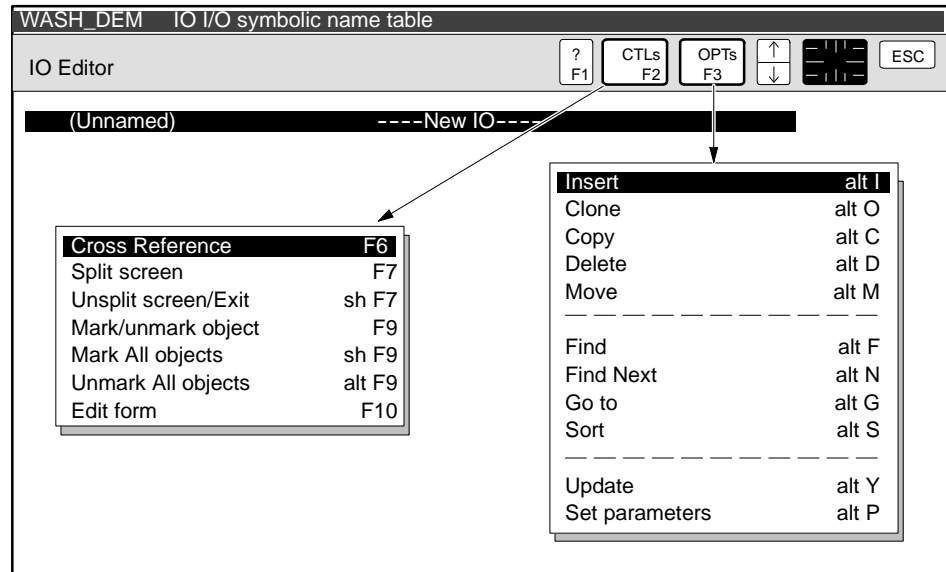


































Figure 3-7 I/O Symbolic Name Table Object List

[Table 3-3](#) lists the APT options and controls capabilities for the I/O Symbolic Name Table editor.

Table 3-3 Controls (F2) and Options (F3) for I/O Table Object List

Icon	Command	Keys	Function
F2 (CTLs)	Cross Reference		Lists objects that are referenced by the selected object, and lists objects that reference the selected object. See Section 3.6 .
	Split Screen		Divides the screen in two, allowing you to display separate areas of APT functions.
	Unsplit Screen/Exit	 	Undoes the split screen option. If the screen is already united, prompts you to exit the table.
	Mark/Unmark Object		Mark objects for other options.
	Mark All Objects	 	Mark all objects for other options.
	Unmark All Objects	 	Unmark all objects.
	Edit Form		Shifts cursor to the object data window from the object list window.
F3 (OPTs)	Insert	 	Adds an item above current cursor position in the table.
	Clone	 	Copies information from one I/O entry to the next and increments the I/O address. You have the option of replacing the contents of any of the fields.
	Copy	 	Duplicates highlighted/selected objects. See Section 3.6 .
	Delete	 	Deletes highlighted/selected object.
	Move	 	Moves highlighted/selected objects from one location to another. See Section 3.6 .
	Find	 	Locates the item you specify in the dialog window. You can search for a text pattern; that is, you do not have to enter the entire name.
	Find Next	 	Locates next occurrence of the item last specified in the Find option.
	Go to	 	Pages the screen to display the portion of the table (TOP or BOTTOM) that you select in the dialog window.
	Sort	 	Arranges items according to name in alphabetical order or according to type or address.
	Update	 	Forces an immediate update of the database.
	Set Parameters	 	Sets system to provide completion aids for each applicable entry and to change paging parameters. See Section 3.6 .

Defining I/O for an S5 Controller (continued)

Defining the I/O Points

To define the individual I/O points, refer to [Figure 3-8](#) and follow the steps listed below.

1. While the I/O Symbolic Name Table editor is open, use the arrow keys to position the cursor over the I/O point to be edited, and press **Enter**. If you are entering a new I/O point, press **Enter** and enter the name of the I/O point.
2. Use **Enter** or **Tab** to position the cursor in the appropriate entry field.
3. Refer to [Table 3-4](#), and enter the necessary data required in each field. See the *SIMATIC APT Programming Reference (Tables) Manual* for detailed information about each type of I/O and its table entries.

If you make an error, you can cancel the entry by pressing **Alt** **U** (Undo) before you exit from the form.

4. Press **Enter** or **Tab** to complete the entry and move to the next field.
5. Repeat Steps 2 and 3 until you complete all necessary entries.

APT automatically saves your entries whenever you exit the data entry form.

6. Use **Esc** to exit the editor and to return to the list of I/O points.
7. After you enter an I/O point, use the Clone option to enter data for related points on a module. For example, consider a module with eight digital outputs. After filling out the form for output, valve1.CMMD, at address Q1.1, press **Alt** **O**. APT displays the form for Q1.2 with fields filled in based on data that you entered for Q1.1.
8. Verify that the configured I/O matches the physical I/O.

WARNING

APT cannot verify that the I/O configuration you specified matches the physical I/O connected to the controller.

An I/O mismatch can cause unpredictable operation that could result in death or serious injury to personnel and/or damage to equipment.

Ensure that your I/O configuration matches the physical I/O prior to placing the controller in RUN mode.

9. You can use either **Esc** or **Shift** **F7** to exit the editor.

Figure 3-8 shows the I/O Table with seven objects entered. The first one, H2OTMP, is being edited.

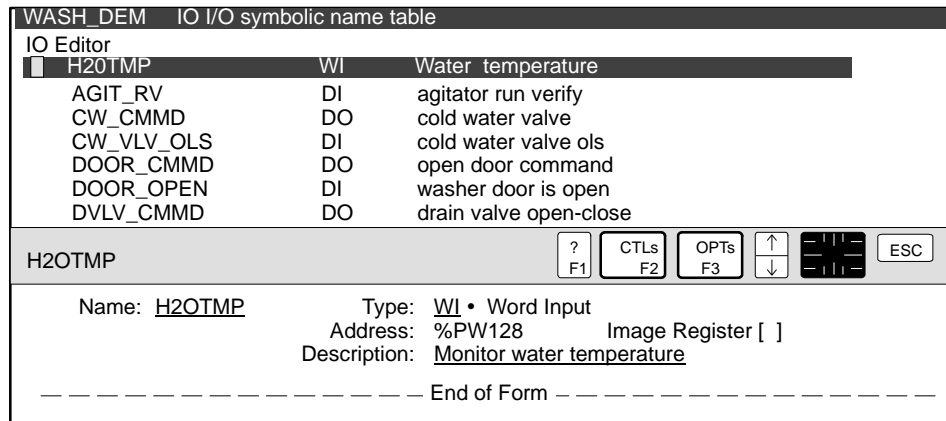


Figure 3-8 S5 I/O Symbolic Name Table Objects

Table 3-4 Field Entries for S5 I/O

Field	Entry	Definition	
Name	Symbolic name of I/O.	Enter a 12-character name that identifies the I/O point.	
Type	I/O module category.	Valid module types are: AI, AO, DI, DO, DF, WI, WO, BI, BO	
Address	Address of I/O point.	I/O module	Valid Addresses ¹
		AI	PW, OW, IW
		AO	PW, OW, QW
		DI	I
		DO	Q
		DF	Q
		WI	PW, OW, IW
		WO	PW, OW, QW,
		BI	PW, OW, IW
BO	PW, OW, QW		
Image Register	Enter X to select option.	Enter X if no I/O modules are physically connected. ²	
Description	Optional description of I/O.	Enter a 30-character description that identifies the I/O.	

1 You must enter an address using a percent sign as a prefix, followed by the address type, followed by the address number, e.g., %PW128.

2 This option allows you to test a program based on data in an image register in APT. It is not necessary to remove the X after connecting the I/O, because the presence of physical I/O takes precedence over this option.

3.4 Editing I/O, Device, Declaration, Recipe, Subroutine Tables

Accessing a Table To access any of the tables, follow these steps.

1. Highlight the table name in the Program Content or Unit Content Directory.
2. Press **F10** or **E** or **Enter**.

After you have accessed one of the table editors, APT displays a list of defined objects. For example, you can select a list of devices from the Device Table. If you have not defined any objects, APT displays a single line, as shown in [Figure 3-9](#) for the Device Table.

3. You can either use **Esc** or **Shift F7** to exit the table editor, or you can press **F10** to enter object data (described on [page 3-16](#)).

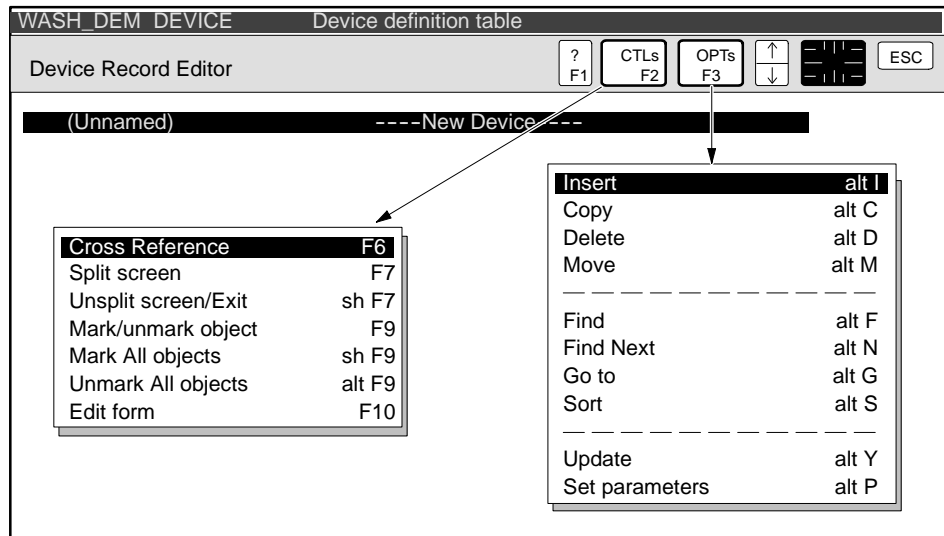


Figure 3-9 Device Table Object List

The APT options and controls capabilities ([Table 3-5](#)) are the same for all tables described in this section. However, the Clone option for S5 controllers is only used in the I/O Symbolic Table.

Table 3-5 Controls (F2) and Options (F3) for Table Object List

Icon	Command	Keys	Function
F2 (CTLs)	Cross Reference		Lists objects that are referenced by the selected object, and lists objects that reference the selected object. See Section 3.6 .
	Split Screen		Divides the screen in two, allowing you to display separate areas of APT functions.
	Unsplit Screen/Exit		Undoes the split screen option. If the screen is already united, prompts you to exit the table.
	Mark/Unmark Object		Mark object for other options.
	Mark All Objects		Mark all objects for other options.
	Unmark All Objects		Unmark all objects.
	Edit Form		Shifts cursor to the object data window from the object list window.
F3 (OPTs)	Insert		Adds an item above current cursor position in the table.
	Clone		(S5 controllers only) Copies information from one I/O entry to the next and increments the I/O address. You have the option of replacing the contents of any of the fields.
	Copy		Duplicates highlighted/selected objects. See Section 3.6 .
	Delete		Deletes highlighted/selected object.
	Move		Moves highlighted/selected objects from one location to another. See Section 3.6 .
	Find		Locates the item you specify in the dialog window. You can search for a text pattern; that is, you do not have to enter the entire name.
	Find Next		Locates next occurrence of the item last specified in the Find option.
	Go to		Pages the screen to display the portion of the table (TOP or BOTTOM) that you select in the dialog window.
	Sort		Arranges items according to name in alphabetical order or according to type or address.
	Update		Forces an immediate update of the database.
	Set Parameters		Sets system to provide completion aids for each applicable entry and to change paging parameters. See Section 3.6 .

Editing I/O, Device, Declaration, Recipe, Subroutine Tables (continued)

Entering Data into a Table

All the tables described in this section require you to enter an item name and type. Other entries depend on the table and item type that you select. See the *SIMATIC APT Programming Reference (Tables) Manual* for specific information about the entry fields.

To enter information in a table for a specific object, follow these steps.

1. While the table editor is open, use the arrow keys to position the cursor over the object to be edited, and then press **F10** or **Enter**.
2. Use the arrow keys to position the cursor in the appropriate entry field.
3. Enter the necessary information in the blank.

If you make an error, you can cancel the entry by pressing **Alt** **U** (Undo) before you exit from the form.

4. Press **Enter** or **Tab** to complete the entry and move to the next field.
5. Repeat Steps 2 and 3 until you complete all necessary entries.

APT automatically saves your entries whenever you exit the data entry form.

6. Use **Esc** to exit the object data editor.

Figure 3-10 shows the I/O Table with seven objects entered. The first one, AGIT_CMD, is being edited.

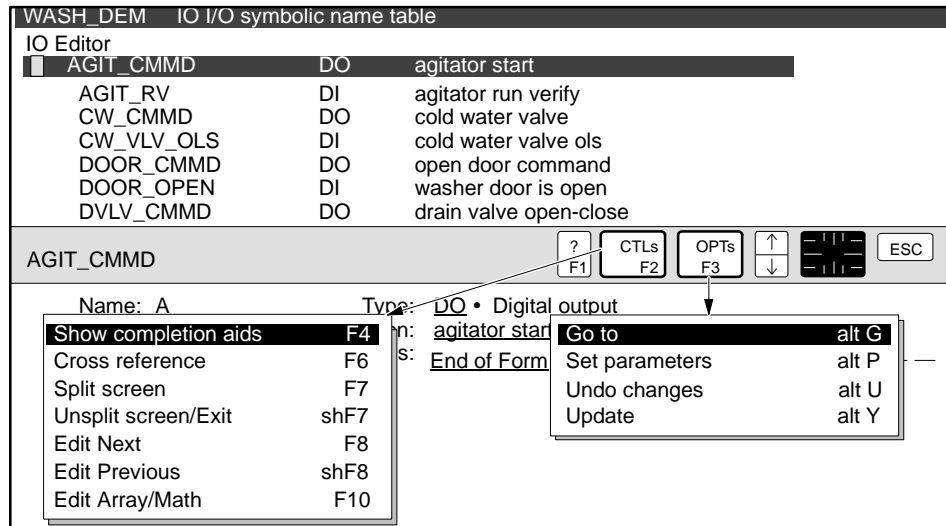



















Figure 3-10 Defining Table Objects

See Table 3-6 for a listing of the options and controls available for defining table objects.

Editing I/O, Device, Declaration, Recipe, Subroutine Tables (continued)

Table 3-6 shows the options and controls available for defining table objects.

Table 3-6 Controls (F2) and Options (F3) for Defining Table Objects

Icon	Command	Keys	Function
F2 (CTLs)	Show Completion Aids		Displays a list of valid choices for a specific entry field.
	Cross Reference		Lists objects that are referenced by the selected object, and lists objects that reference the selected object. See Section 3.6 .
	Split Screen		Divides the screen in two, allowing you to display separate areas of APT functions.
	Unsplit Screen/Exit	 	Undoes the split screen option. If the screen is already united, prompts you to exit the table.
	Edit Next		Shifts cursor to next object in list.
	Edit Previous	 	Shifts cursor to previous object in list.
	Edit Array/Math		Allows you to enter values into an array in the declaration table, enter recipe elements in the recipe table, or enter math in the subroutine table.
F3 (OPTs)	Go to	 	Pages the screen to display the portion of the table (TOP or BOTTOM) that you select in the dialog window.
	Set Parameters	 	Sets system to provide completion aids for each applicable entry and to change paging parameters.
	Undo	 	Cancels action before you press the ENTER key when making an entry in a table.
	Update	 	Forces an immediate update of the database.

Entering Data into an Array

The Declaration table editor has a sub-editor to enter values into an array. The controls and options available in the Declaration Array sub-editor are shown in [Figure 3-11](#) and defined in [Table 3-6](#).

To specify the values of array elements, follow these steps.

1. To access the Declaration Array sub-editor, place the cursor on the name of an array, and press **F10** or **Enter**.
2. Press **F10** again to enter the sub-editor.
3. Enter the appropriate value for each element.
4. To exit the sub-editor, press **Esc** until you return to the hierarchy.

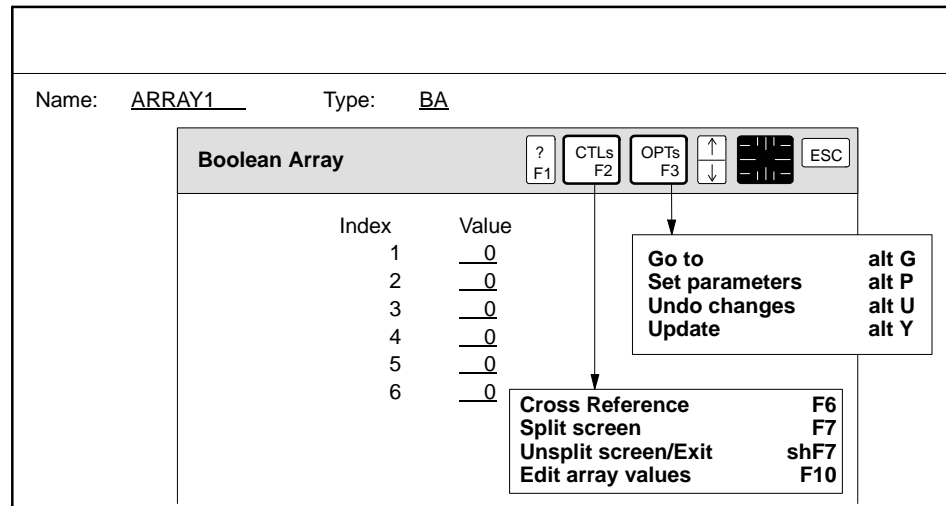


Figure 3-11 Editing an Array

Editing I/O, Device, Declaration, Recipe, Subroutine Tables (continued)

Entering Data into a Recipe

The Recipe table editor has a sub-editor to enter an initial value or a declared constant for elements that you defined in the recipe template. The controls and options available in the Recipe table sub-editor are shown in [Figure 3-12](#) and [Figure 3-13](#), and defined in [Table 3-6](#).

NOTE: Before building a recipe table, you must have a recipe template. To create a recipe template, see [Section 3.5](#).

After you have accessed the Recipe table editor, APT displays a list of defined recipes. For example, in [Figure 3-12](#) the two recipes WR and SR are listed. To specify the values for recipe elements, follow these steps.

1. To access the Recipe sub-editor, place the cursor on the name of a recipe and press **F10** or **Enter**.

The screen shown in [Figure 3-12](#) is displayed.

2. Enter the appropriate information, e.g., template, description, etc.

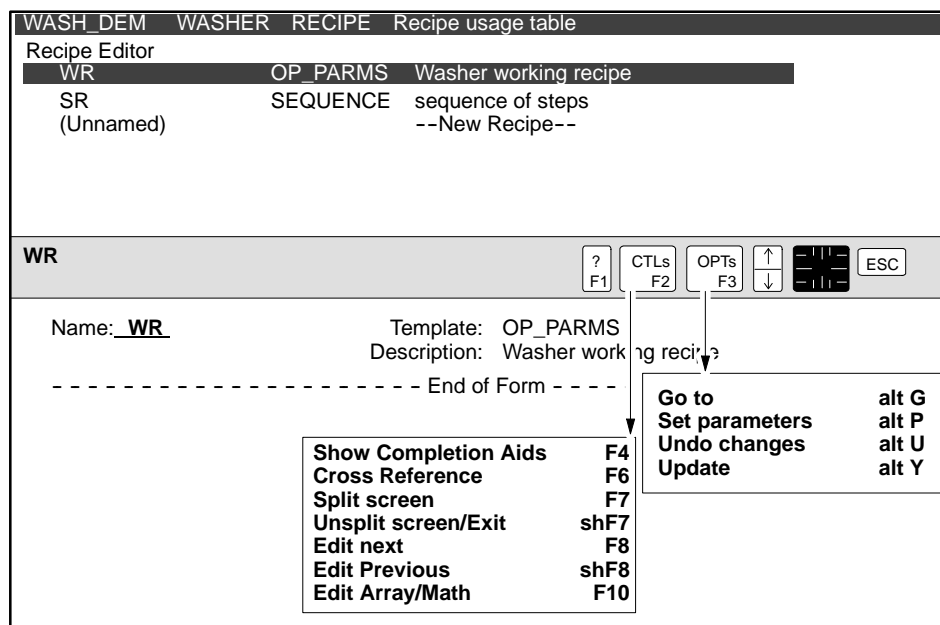


Figure 3-12 Editing Recipe Elements

3. To enter specific values for each recipe element, press **F10**.
The screen shown in **Figure 3-13** is displayed.
4. Enter the appropriate value for each recipe element.
5. To exit the sub-editor, press **Esc** until you return to the hierarchy.

The screenshot shows the 'Recipe Editor' window with the following structure:

- Header:** WASH_DEM WASHER RECIPE Recipe usage table
- Sub-headers:** Recipe Editor, WR OP_PARMS Washer working recipe, SR SEQUENCE sequence of steps, (Unnamed) --New Recipe--
- Toolbar:** ? F1, CTLs F2, OPTs F3, navigation arrows, ESC
- Table:**

VALUE	ENG. UNITS	ELEMENT NAME	DESCRIPTION	TYPE
1		NUM_RUNS	Number of loads to wash	I
MED_LEVEL	high/low	FILL_SP	fill cycle setpoint	I
WARM	hot/cold	TEMP_SETPT	wash/rinse cycle te	I
1.5	min	WASH_TIME1	agitator wash time	R
- Action List (Right):**
 - Go to alt G
 - Set parameters alt P
 - Undo changes alt U
 - Update alt Y
- Action List (Bottom):**
 - Cross Reference F6
 - Split screen F7
 - Unsplit screen/Exit shF7
 - Edit array values F10

Figure 3-13 Entering Values for Recipe Elements

Editing I/O, Device, Declaration, Recipe, Subroutine Tables (continued)

Entering Subroutine Math

The Subroutine table editor has a sub-editor to enter the subroutine math. The controls and options available in the Subroutine sub-editor are shown in [Figure 3-14](#), and defined in [Table 3-7](#).

To enter the math for a subroutine, follow these steps.

1. To access the Subroutine sub-editor, place the cursor on the name of a subroutine, and press **F10** or **Enter**.
2. Press **F10** again to enter the sub-editor.
3. Enter the appropriate math.
4. To exit the sub-editor, press **F10** or **Esc** until you return to the hierarchy.

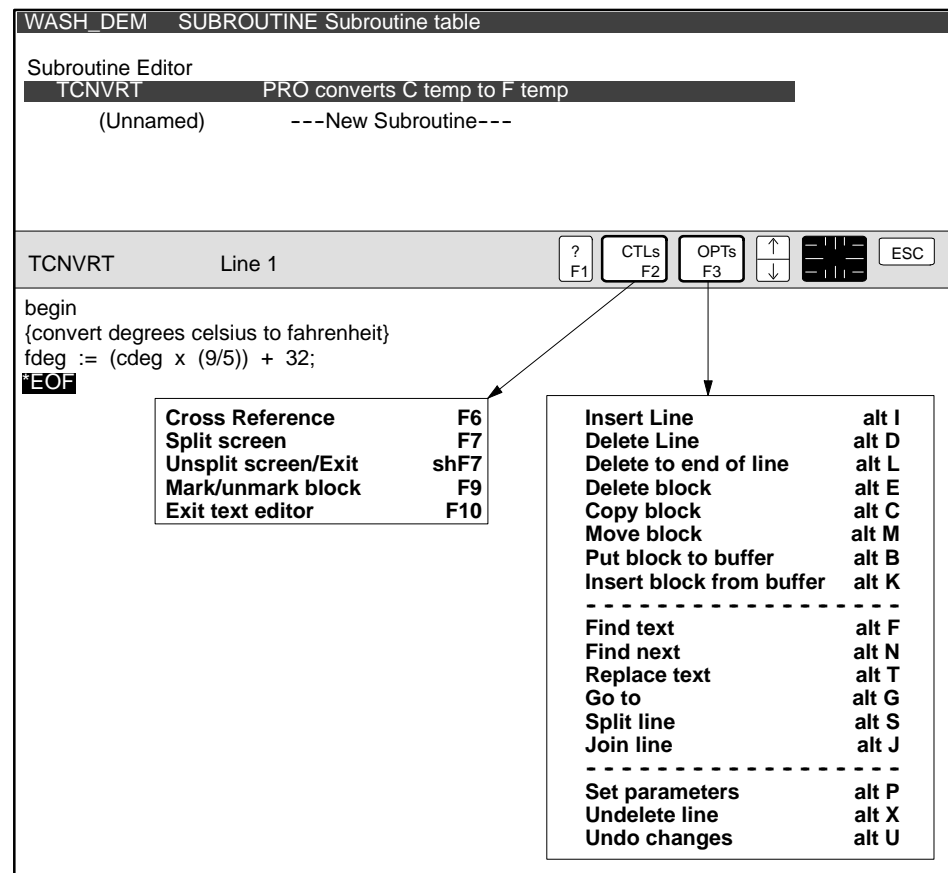


Figure 3-14 Editing Subroutine Math

Table 3-7 Controls (F2) and Options (F3) for Subroutine Sub-Editor

Icon	Command	Keys	Function
F2 (CTLs)	Cross Reference	<input type="button" value="F6"/>	Lists objects that are referenced by the selected object, and lists objects that reference the selected object. See Section 3.6 .
	Split Screen	<input type="button" value="F7"/>	Divides the screen in two, allowing you to display separate areas of APT functions.
	Unsplit Screen/Exit	<input type="button" value="Shift"/> <input type="button" value="F7"/>	Undoes the split screen option. If the screen is already united, prompts you to exit the table.
	Mark/Unmark Block	<input type="button" value="F9"/>	Marks block for other options.
	Exit Text Editor	<input type="button" value="F10"/>	Exits the text editor and return to APT hierarchy.
F3 (OPTs)	Insert Line	<input type="button" value="Alt"/> <input type="button" value="I"/>	Allows you to add one blank line at the current cursor position.
	Delete Line	<input type="button" value="Alt"/> <input type="button" value="D"/>	Removes one line of text.
	Delete to End of Line	<input type="button" value="Alt"/> <input type="button" value="L"/>	Removes text from cursor position to end of line.
	Delete Block	<input type="button" value="Alt"/> <input type="button" value="E"/>	Removes all text in the marked block.
	Copy Block	<input type="button" value="Alt"/> <input type="button" value="C"/>	Copies all text in a marked block to the current cursor position.
	Move Block	<input type="button" value="Alt"/> <input type="button" value="M"/>	Relocates all text in a marked block to the current cursor position.
	Put Block to Buffer	<input type="button" value="Alt"/> <input type="button" value="B"/>	Places marked block in a buffer.
	Insert Block from Buffer	<input type="button" value="Alt"/> <input type="button" value="K"/>	Moves marked block from buffer into text.
	Find Text	<input type="button" value="Alt"/> <input type="button" value="F"/>	Locates a specific character string in the text.
	Find Next	<input type="button" value="Alt"/> <input type="button" value="N"/>	Continues search initiated by Find text and locates next occurrence in the text.
	Replace Text	<input type="button" value="Alt"/> <input type="button" value="T"/>	Locates and replaces a character string in the text.
	Go to	<input type="button" value="Alt"/> <input type="button" value="G"/>	Pages the screen to display the portion of the text area (TOP or BOTTOM) that you select in the dialog window.
	Split Line	<input type="button" value="Alt"/> <input type="button" value="S"/>	Breaks line at current cursor position and places remainder of that line in a new line.
	Join Line	<input type="button" value="Alt"/> <input type="button" value="J"/>	Combines two successive short text lines into a longer line.
	Set Parameters	<input type="button" value="Alt"/> <input type="button" value="P"/>	Allows you to specify the size of the text area, the amount of paging that occurs when you press PgUp or PgDn, and whether or not the editor inserts a line when you press ENTER.
Undelete Line	<input type="button" value="Alt"/> <input type="button" value="X"/>	Replaces deleted line of text.	
Undo Changes	<input type="button" value="Alt"/> <input type="button" value="U"/>	Restores text to condition at start of the edit session and erases all modifications.	

3.5 Building a Recipe Template

Adding a Template

Before you can build a recipe table, you must create a recipe template that identifies all the possible elements to use in a set of similar recipes.

To create a recipe template, follow these steps.

1. At the Program Content Level, highlight:

----- Recipe Templates -----

2. Press **[A]**, or use the **OPTs** icon to select **Add new template**.

The dialog window appears as shown in [Figure 3-15](#).

3. Enter a name and description for the recipe template, and press **[Enter]**.

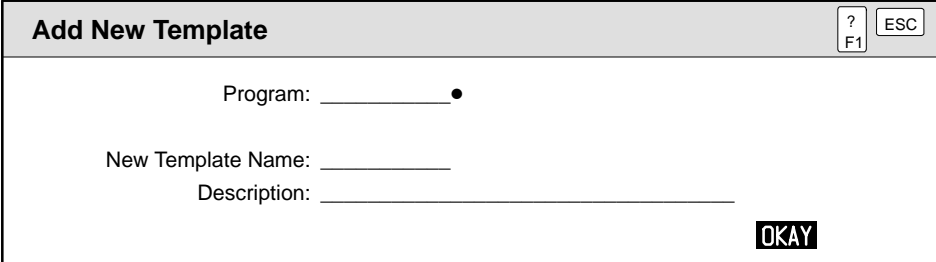


Figure 3-15 Adding a Template

Editing a Template

To add elements to a recipe template, follow these steps.

1. Place the cursor on the template name in the Program Content and press **[F10]** or **[E]** or **[Enter]**.

The screen shown in [Figure 3-16](#) is displayed. If no elements have been added to the template, the following message appears:

(Unnamed) ---New Template Element---

2. Place the cursor on **New Template Element**, and press **[F10]** or **[E]** or **[Enter]**.

The screen shown in [Figure 3-17](#) is displayed.

3. Enter the element name and type, and press **[Enter]**.
4. Enter the description, engineering units, and any other appropriate data, and press **[Enter]**.
5. To exit the sub-editor, press **[Esc]** until you return to the hierarchy.

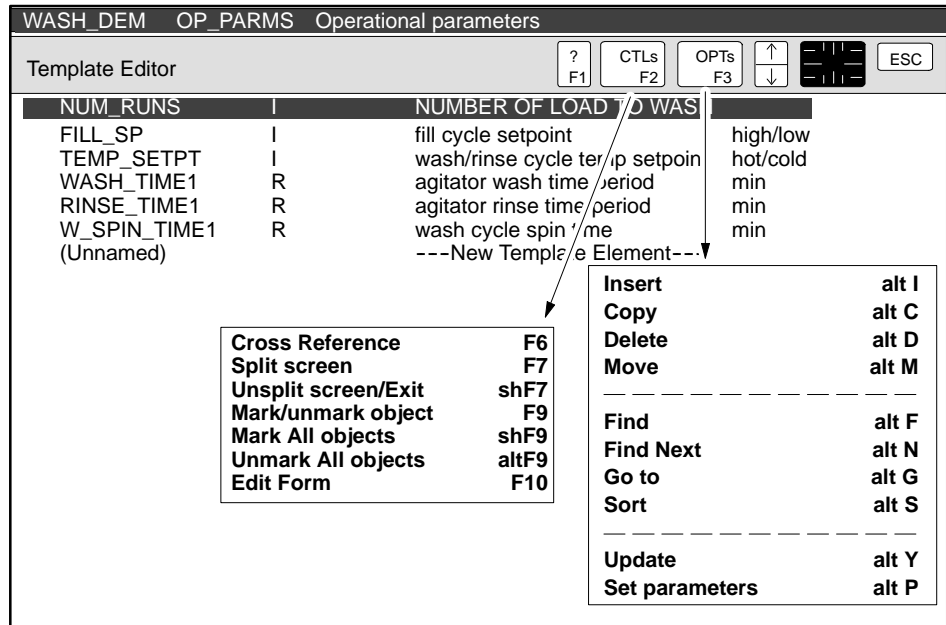


Figure 3-16 Examining List of Template Elements

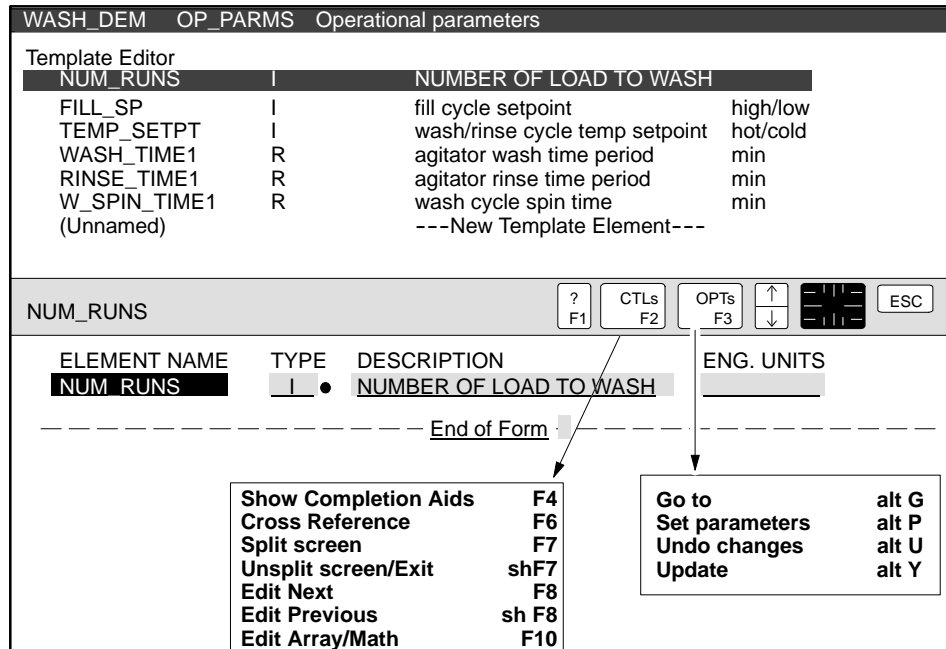


Figure 3-17 Editing a Template Element

3.6 More about the Controls (F2) and Options (F3)

Setting Parameters (Alt P)

In the table editors, you can set default options to suit your editing style. These options include the display of completion aids, the display of object data, the number of lines skipped when you page, etc.

You can change the parameters at any point while you are editing a table by pressing , or by using the **OPTs** icon to select the **Set parameters** function. [Table 3-8](#) lists the parameters that you can set.

Table 3-8 Parameter Options

Table	Parameter	Function
Module (Series 505)	Show Completion Aids	Displays a list of valid choices for a specific entry field. The Completion Aids option is described more fully on page 3-27 .
	Amount to Page: All 3/4 1/2 1/4 1 Name	Sets number of lines to skip when you page up or down.
	Default I/O Series	Sets the default I/O Series to Series 500 or Series 505.
I/O, Device, Declaration Recipe, Subroutine: Object List Editor	Amount to Page Up/Down	Sets number of lines to skip when you page up or down.
	Display Object Data on Entry	Displays the object data for the object currently selected when you enter the table object list editor.
I/O, Device, Declaration Recipe, Subroutine: Object Data Editor	Maximum Amount of Display for Form	Sets number of lines of the object data form that are displayed.
	Amount of Display to Page Up/Down	Sets number of lines to skip when you page up or down.
	Display Object Data	Displays the object data for the object currently selected when the cursor is in the object list editor.
	Show Completion Aids	Displays a list of valid choices for a specific entry field. The Completion Aids option is described more fully on page 3-27 .

Selecting Completion Aids

Many fields in the APT forms have a limited number of valid entries. You can get help for entering data into these fields by selecting the **Completion Aids** option. APT displays the available choices, or an appropriate subset of the choices, depending on the table that you are currently viewing. In the device table, the completion aids list global I/O symbolic names when you are displaying global devices, and unit I/O names when you are displaying unit-level devices.

Each field with available completion aids has a black dot to its right. To display the completion aids, place the mouse cursor on the dot and click the left mouse button. You can also move the cursor to the field and press **F4**, or select the **Show Completion Aids** function on the **CTLs (F2)** menu. See [Figure 3-18](#).

You can enable the automatic display of completion aids for a field by selecting the **Set parameters** function and placing an **X** between the brackets for the **Show Completion Aids** option. To remove the completion aids, press the spacebar.

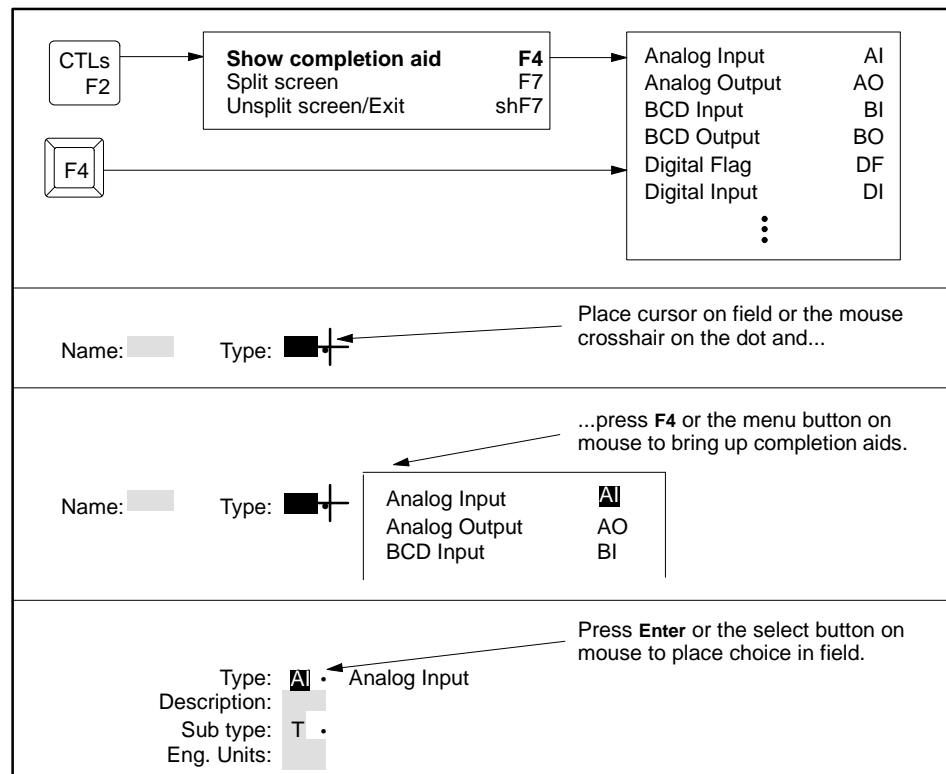


Figure 3-18 Displaying the Completion Aids

More about the Controls (F2) and Options (F3) (continued)

Using the Cross Reference (F6)

If you need to know which objects are referenced by other objects in the program, use the **Cross Reference** option. This option lists:

- All the items that the specified object references, or
- All the items that the specified object is referenced by.

In [Figure 3-19](#) a cross reference option (References) has been selected for the device named AGITATOR. AGITATOR references the two I/O points, Agit_Rv and Agit_Cmmd. The cross reference option selected in [Figure 3-20](#) (Referenced By) shows the three CFBs, AGIT_ILK, LOCK_DEV, and STRT_AGI, and the transitions and steps in the SFC named AGIT, that AGITATOR is referenced by.

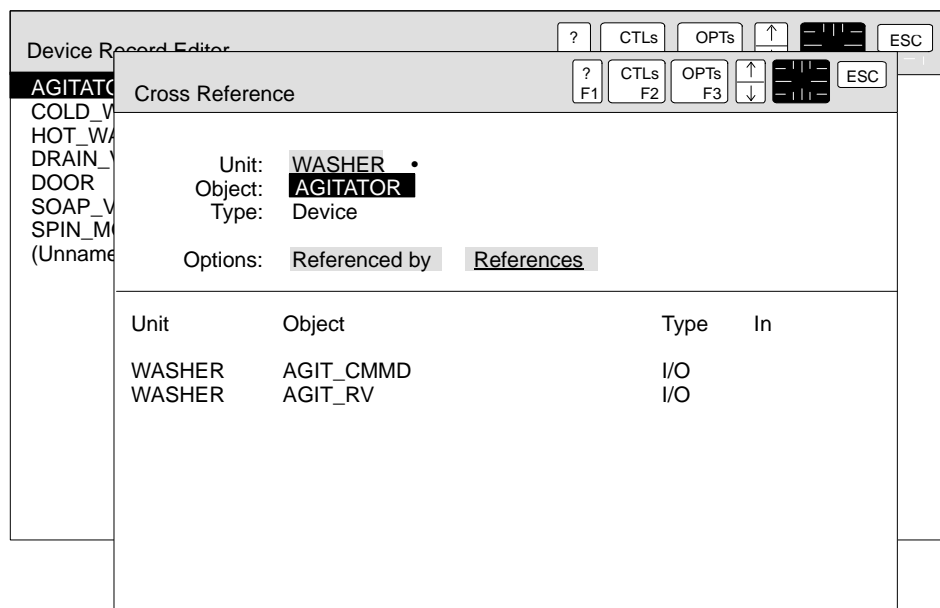


Figure 3-19 Example Cross Reference: Items That the Specified Object References

You can do a cross reference on objects only after the objects and their references are validated. If you attempt to do a cross reference on an object before all its cross references are validated, APT lists only those references that have been validated. Since validation is part of the compile procedure, you can do a cross-reference check at any point of the hierarchy after a program compile.

To do a cross-reference check on an object, place the cursor on the object and press **F6**. A dialog box, similar to the ones in [Figure 3-19](#) and [Figure 3-20](#), is displayed. You can also press **F6** from any point in the hierarchy and enter the name of the object that you want to check. Press **Alt S** to save the information in the form of a report.

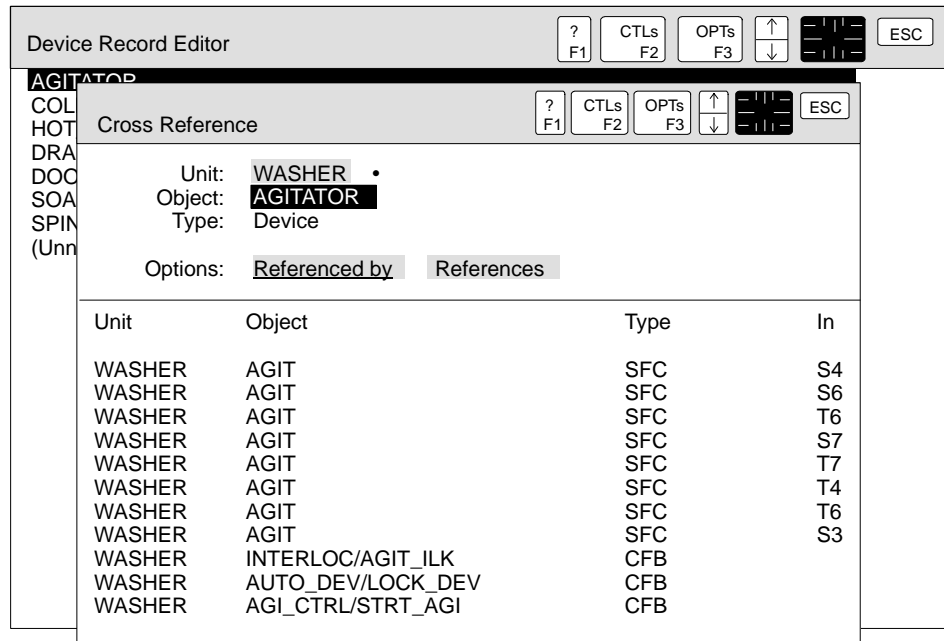


Figure 3-20 Example Cross Reference: Items That the Specified Object Is Referenced By

Cross Reference is one of the report options for the object code of a compiled program. After compiling a program, you can generate a list of cross references for all the objects defined in the program.

Follow these cross-reference guidelines.

- The default contents of the header fields of the dialog box depend on the object that you cross reference.
- The program is referenced by all objects that reference the program extensions. The program does not reference any other objects.
- A unit is referenced by all objects that reference the unit extensions. A unit does not reference any other objects.
- No objects can reference objects in the Series 505 module table.
- No objects in the I/O table can reference any other objects in the program.
- You cannot do a cross reference on objects that are in reports or in MAITT programs.

More about the Controls (F2) and Options (F3) (continued)

Copying Selected Objects (Alt C)

When you need to enter similar objects into a table, use the **Copy** option to save time. For example, if you have already entered Valve_H2O into a device table and you need to enter Valve_OIL, which is similar, make a copy of Valve_H2O and modify it appropriately for Valve_OIL. You can also place a copy elsewhere in the program, e.g., in another unit, but the table must be of the same type.

The copy option makes a copy of the object that is currently highlighted by the cursor. You can copy multiple objects by selecting them with the **Mark** option first.

Follow these object-copying guidelines.

- You must place the copy into the same type of table, e.g., copy a device to another device table.
- If objects already exist in the destination table, APT displays a **Replace duplicates?** message with these options.

Yes. The existing object is overwritten.

No. The existing object is not overwritten. If you are copying multiple (marked) objects, only unique objects are added (inserted) to the table.

Prompt. You can choose which objects in the destination table to overwrite.

- I/O addresses are not copied. Therefore, you must assign addresses in either the module table for Series 505 or the I/O table for S5.
- Objects that are not copied into the destination table remain marked.
- Objects that are marked for translation do not copy the marked information if they are being copied to the same program.

Moving Selected Objects (Alt M)

Use the **Move** option to transfer objects from one table to another. This option makes a copy of the object currently highlighted by the cursor, deletes the original object, and places the copy in the destination table. You can move multiple objects by selecting them with the **Mark** option first.

Follow the following guidelines about moving objects.

- You must move the object into the same type of table, e.g., move a device to another device table.
- If objects already exist in the destination table, APT displays a **Replace duplicates?** message with these options.

Yes. The existing object is overwritten.

No. The existing object is not overwritten. If you are moving multiple (marked) objects, only unique objects are added (inserted) to the table.

Prompt. You can choose which objects in the destination table to overwrite.

- I/O addresses are moved within a program. Therefore, you do not have to reassign addresses in the module table for Series 505 or the I/O table for S5.
- Objects that are not moved into the destination table remain marked.
- Objects that are marked for translation move the marked information also. The object is assigned to a new controller address. APT displays a **Move translated items?** message with these options:

Yes. The marked object is moved.

No. The marked object is not moved.

Prompt. You can choose which marked objects are moved.

NOTE: When you do a block move of multiple objects to a table that contains any objects, a message is displayed indicating that objects already exist in the destination table. This message appears even when no duplicate objects are present in the destination. If you continue with the move procedure, APT prompts you before it attempts to overwrite any existing objects.

Chapter 4

Using the SFC Editor

4.1	Creating an SFC	4-2
	Overview	4-2
	Step/Transition Labels	4-3
	Adding an SFC	4-3
	Accessing the SFC Editor	4-3
	Understanding Drawing Icons	4-4
	Using Drawing Icons	4-5
4.2	Changing an SFC	4-7
	Overview	4-7
	Finding Objects	4-8
	Inserting Graphics	4-8
	Marking a Block	4-9
	Copying a Block to a Buffer	4-9
	Resequencing	4-9
	Using the Options Icon	4-10
	Exiting an SFC	4-11
4.3	Using the SFC Sub-Editor	4-12
	Overview	4-12
	Entering Text	4-12
	Marking a Block of Text	4-13
	Changing Text	4-14

4.1 Creating an SFC

Overview

In developing an APT program, you create structures called Sequential Function Charts (SFCs) that specify the sequence of events in your control process. SFCs are composed of steps and transitions. Each step can contain one or more commands; each transition contains one conditional expression.

Parallel branches are sets of steps that execute at the same time. Parallel branches allow you to follow a single transition with more than one step, as shown in [Figure 4-1](#).

Selection branches are a series of steps that represent a choice in the program. Selection branches allow you to follow a single step with more than one transition, as shown in [Figure 4-1](#). The first transition to become true determines which step executes. APT tests the transitions from left to right.

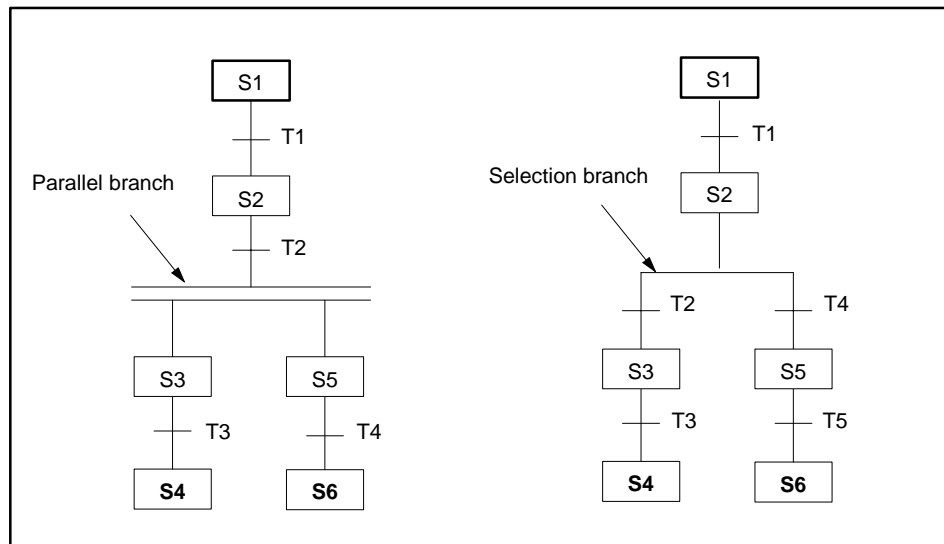


Figure 4-1 Parallel and Selection Branches

Step/Transition Labels

APT automatically numbers each step with a unique label from S1 to S500 and each transition with a unique label from T1 to T500.

This numbering system reflects the order in which you create and place objects on the drawing but can be changed by using the **Sequence Labels** command. For information about the **Sequence Labels** command, see [page 4-9](#).

Adding an SFC

To add an SFC, follow these steps.

1. At the Program or Unit Content Level, highlight:

- - - - - **SFCs** - - - - -

2. Press , or use the **OPTs** icon to select **Add new SFC**.
3. Enter a name and description in the dialog window and press .

An SFC name can contain up to eight alphanumeric characters. The name must include at least one letter and can include an underscore. An SFC name cannot begin with a number.

4. Press again to complete the action.

Accessing the SFC Editor

To access an SFC, follow these steps.

1. At the Program or Unit Content Level, highlight the name of the SFC.
2. Press or or .

Because the SFC editor operates in system memory, you can receive a message that the SFC has reached its memory limits. This can occur if you have numerous steps and transitions, or steps that contain many statements.

If you receive this message, save the SFC, and then do one of the following.

- Exit and re-enter the SFC editor. This eliminates fragmented memory that occurred in the previous editing session.
- Reduce the number of steps and transitions and/or the number of statements in each step.
- Use subordinate SFCs to replace steps that have many statements.

Creating an SFC (continued)

Understanding Drawing Icons

APT organizes the SFC graphic by using a coordinate system that is 25 cells wide and 100 cells deep. The 25 columns across are labeled A to Y, and the 100 rows are numbered 1 to 100. The SFC editor allows you to place steps and transitions on the graphic by using the icons shown in [Figure 4-2](#).

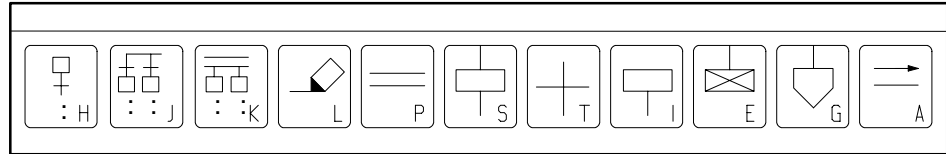













Figure 4-2 SFC Drawing Icons

	H	Step Transition Macro places a single column of alternating step and transition symbols in the SFC.
	J	Selection Macro places multiple columns of alternating transition and step symbols. They are joined at the top or bottom by a selection branch line.
	K	Parallel Macro places multiple columns of alternating step and transition symbols that are joined at the top or bottom by a parallel branch line.
	L	Line Draw icon places vertical and horizontal lines between SFC graphic symbols.
	P	Parallel places a parallel branch line in the SFC.
	S	Step places one step symbol in the SFC.
	T	Transition places one transition symbol in the SFC.
	I	Initialization Step places one initial-step symbol in the SFC.
	E	End Step places one end-step symbol in the SFC.
	G	Graphic Connections places one graphic-connection symbol in the SFC to transfer control to another step.
	A	Arrows places one arrow in the SFC to indicate the direction of program logic flow.

Using Drawing Icons

You select a drawing icon the same way you select other icons in APT: you either press the associated letter key, or place the cross-hair over the icon and press the left button.

The drawing icons act in the same way as toggle switches. As long as an icon is highlighted, it remains active. Whenever you press **Enter**, the highlighted symbol appears on the screen.

To turn off an icon (i.e., make it inactive), either press the associated letter key again, or place the cross-hair on the icon and press the left button. You can also turn off an icon by selecting another icon. To cancel an action, press **Esc**, or the right button.

Single drawing icons include the initial step (I), the end step (E), the graphic connection (G), the transition (T), and the step (S).

To place a single step, transition, or graphic connection in an SFC, follow these steps.

1. Press the appropriate letter key.
2. Place the text cursor over the cell where you want to place the symbol.
3. Press **Enter**.

Creating an SFC (continued)

Macro-drawing icons include the step transition macro (H), the selection macro (J), the parallel macro (K), the single line draw (L), and the parallel line draw (P). Macro-drawing icons use an anchor point from which you can move to place symbols in the SFC.

To place lines or macros in an SFC, follow these steps.

1. Press the appropriate letter key.
2. Place the text cursor over the cell that you want to use as the anchor point for drawing the symbols.
3. Press **Enter**.

Select the ending cell or press ESC to cancel... appears in the prompt line at the bottom of the screen; see [Figure 4-3](#).

4. Move the cursor along an appropriate path to locate the ending cell.
5. Press **Enter**.

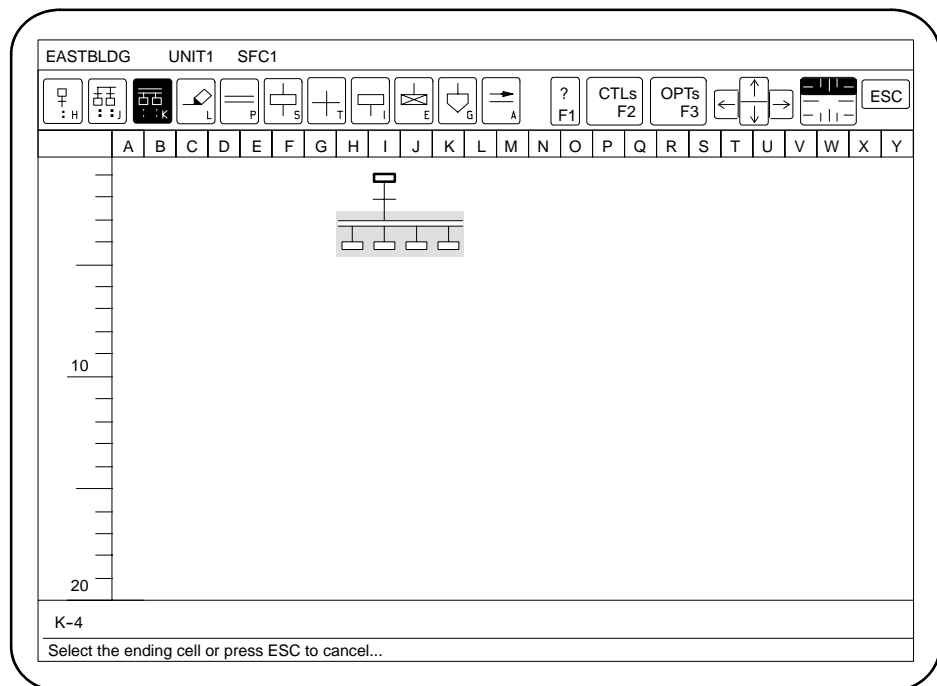


Figure 4-3 Using the Drawing Icons

4.2 Changing an SFC

Overview

When you build an SFC, you have the controls and options available that are shown in [Figure 4-4](#).

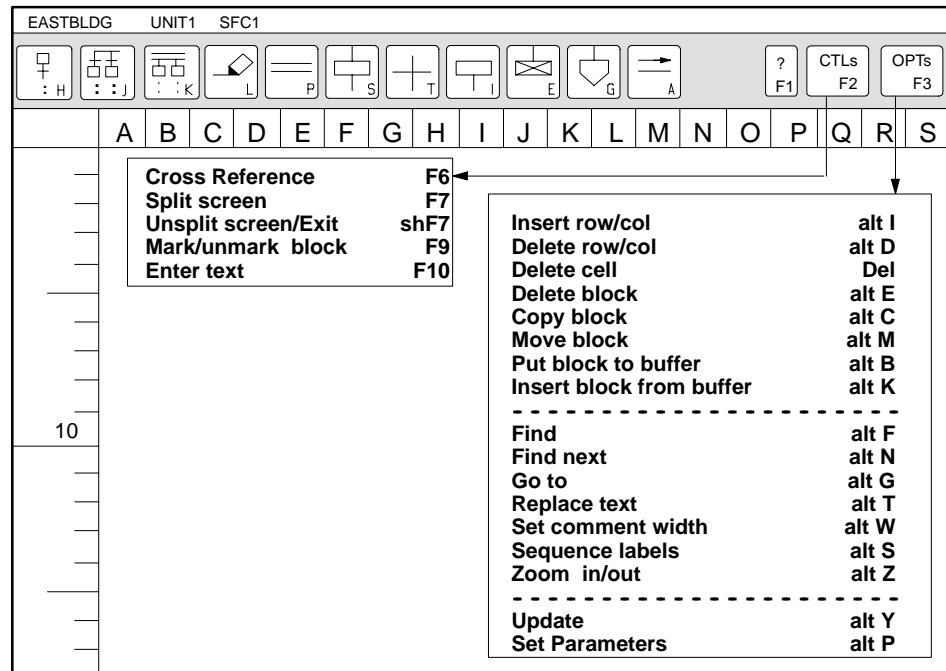


Figure 4-4 SFC Controls and Options

Changing an SFC (continued)

Finding Objects

In the SFC editor, you can locate a specific cell, step, transition, or text string by using the Find command. You must use a specific entry, as listed in [Table 4-1](#). The Find command searches from the current cursor position to the end of the drawing. When APT locates the requested object, the cursor moves to the location on the grid and opens the appropriate sub-editor.

Table 4-1 APT Find Command

Object	Entry	Example
Cell	Grid coordinates	D33
Step	Step label	S105
Transition	Transition label	T11
Character string	Text	PT101.HHA

Inserting Graphics

You can insert a row or column in an existing SFC. When you insert a column, the existing graphics move to the right. When you insert a row, the graphics move down from the current cursor position.

If you attempt to insert graphics that could push the SFC beyond Column Y or Row 100, the editor beeps and does not allow you to complete the insertion.

Marking a Block

After you build an SFC, you can use the **CTLs** icon to mark the block of cells you want to copy, delete, or move. To mark a block, follow these steps.

1. Position the text cursor over the first cell you want to mark.
2. Press **[F9]**, or use the **CTLs** icon to select **Mark/unmark block**.

A colored box highlights the cell, which then serves as an anchor point for you to begin marking the block.

3. Move the cursor until the colored box includes all the cells you want to mark.
4. To complete the marking process, press **[Enter]** or **[F9]**.
To abort the marking process, press **[Esc]**.

To copy, move, or delete the block of cells, use the commands available through the **OPTS** icon, which are outlined in [Table 4-2](#).

Copying a Block to a Buffer

You can copy a block to a buffer in order to use it later in your process. Press **[Alt] [B]** to copy the marked block to a buffer. The marked block will contain the SFC graphic symbols and their underlying steps and transition texts. You can copy this marked block into another SFC. Press **[Alt] [K]** to insert the block from the buffer into the displayed SFC. If you copy the contents of a step of an SFC to a buffer, you can insert the buffer into another SFC, a CFB, or a report.

Resequencing

After you make a series of editing changes, the step and transition labels may no longer be sequential. You can reorder these by using the Sequence Labels command. This command re-numbers steps and transitions from top to bottom, left to right.

If you have a Graphic Connection icon in the SFC, the command automatically adjusts the label reference according to the new number scheme. However, if you have a Graphic Connection icon but have deleted the original step or transition to which it refers, this command deletes the reference and re-numbers the labels for all steps and transitions, except the label for the connection.

Changing an SFC (continued)

Using the Options Icon [Table 4-2](#) lists the commands available using the the OPTs icon.

Table 4-2 Options for SFC Graphic Editor






















Command	Keys	Remarks
Copy Block	 	Duplicates the marked block at the current cursor position, along with all associated step and/or transition commands.
Delete Block	 	Removes the marked block, along with all associated step and/or transition commands. This command does not make any adjustments to remaining cells.
Delete Cell		Removes contents of single cell, along with all associated commands.
Delete Row/Column	 	Removes the column or row that you specify in the dialog window. This command adjusts the chart right to left, or bottom to top, to fill in the deleted column or row.
Find	 	Locates specific grid coordinates, a step label, a transition label, or a text string in a step or transition.
Find Next	 	Continues search started by the Find command.
Go to	 	Pages the screen to display the portion of the SFC (TOP, BOTTOM, LEFT, or RIGHT) that you select in the dialog window.
Insert Block from Buffer	 	Allows you to copy SFC graphic symbols and their underlying steps and transition texts from a buffer into the displayed SFC.
Insert Row/Column	 	Allows you to add a blank column or row in the SFC. All existing graphics move down or to the right. If you attempt to insert graphics that would push the SFC beyond Column Y or Row 100, the editor does not allow you to complete the insertion.
Move Block	 	Relocates the marked block, with all associated step and transition commands, to the current cursor position.
Put Block to Buffer	 	Copies the marked block, with all associated step and transition commands, to a buffer.

Table 4-2 Options for SFC Graphic Editor (continued)

Command	Keys	Remarks
Replace Text	<input type="button" value="Alt"/> <input type="button" value="T"/>	Replaces the specified text with another character string. This command begins at the current cursor position and continues to the end of the chart. You have the option of replacing all occurrences without repeating the command.
Sequence Labels	<input type="button" value="Alt"/> <input type="button" value="S"/>	Reassigns label numbers to steps and transitions in the drawing. This command renumbers steps and transitions from top to bottom, left to right.
Set Comment Width	<input type="button" value="Alt"/> <input type="button" value="W"/>	Allows you to adjust the width of the cell and comment box that appear next to the graphical step symbols in Zoom mode.
Set Parameters	<input type="button" value="Alt"/> <input type="button" value="P"/>	Allows you to specify a percentage of the screen to move up, down, right, or left when you page through the displayed SFC. This affects paging with the PgUp, PgDn, Home, and End keys.
Update	<input type="button" value="Alt"/> <input type="button" value="Y"/>	Saves the displayed SFC to disk without leaving the edit session.
Zoom In/Out	<input type="button" value="Alt"/> <input type="button" value="Z"/>	Changes the display mode of the SFC. This serves as a toggle key that enlarges or reduces the display size of specified cells based on the cursor location.

Exiting an SFC

To exit an SFC, follow these steps.

1. Press .

If you have made changes to the SFC, the dialog window appears as shown in [Figure 4-5](#).

2. Highlight the appropriate response, and press .

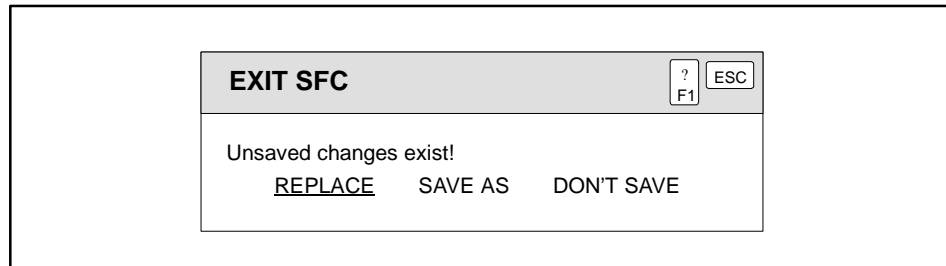


Figure 4-5 Exiting an SFC

4.3 Using the SFC Sub-Editor

Overview

The SFC provides a sub-editor for entering textual commands associated with the steps and transitions of the process control problems. Steps contain commands that manipulate APT objects, and a Math section containing calculations. Each transition consists of a boolean expression to control program flow through the SFC. The text in the SFC step or transition cannot include a tilde (~) or a caret (^).

Figure 4-6 shows the controls and options available in the SFC sub-editor.

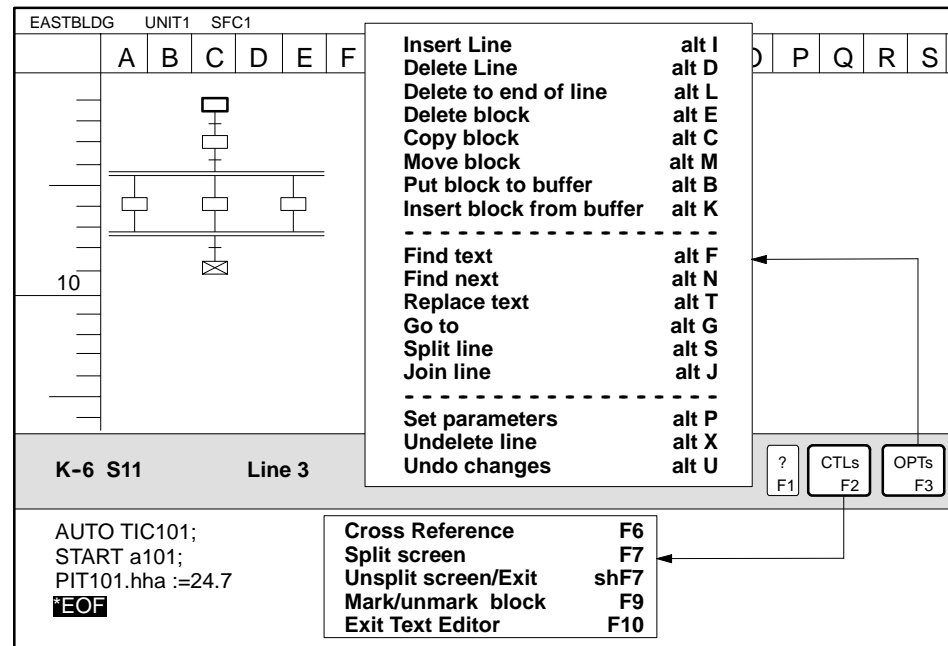


Figure 4-6 Using the SFC Sub-Editor

Entering Text

To enter text in an SFC step, transition, or graphic connection, follow these steps.

1. Position the text cursor over the step, transition or graphic-connection symbol you want to edit.
2. Press **F10**, or use the **CTLs** icon to select **Enter text**.
3. Enter the text for the step or transition.
4. To exit the sub-editor, press **F10** or **Esc**.

When you exit, the sub-editor automatically saves your commands and returns to the SFC.

Marking a Block of Text

After you enter text in a step or transition, you can use the **CTLs** icon to mark a block of text to copy, delete, or move. To mark a block, follow these steps.

1. Position the text cursor at the beginning of the text that you want to copy, delete, or move.
2. Press **F9**, or use the **CTLs** icon to select **Mark/unmark block**.

A colored box marks the beginning of the text.

3. Move the cursor until the colored box includes all the text that you want to mark. See [Figure 4-7](#).
4. Press **Enter** or **F9**, or use the **CTLs** icon to select **Mark/unmark block**.

To copy, move, or delete the block of text, use the commands available through **OPTs** icon, which are outlined in [Table 4-3](#).

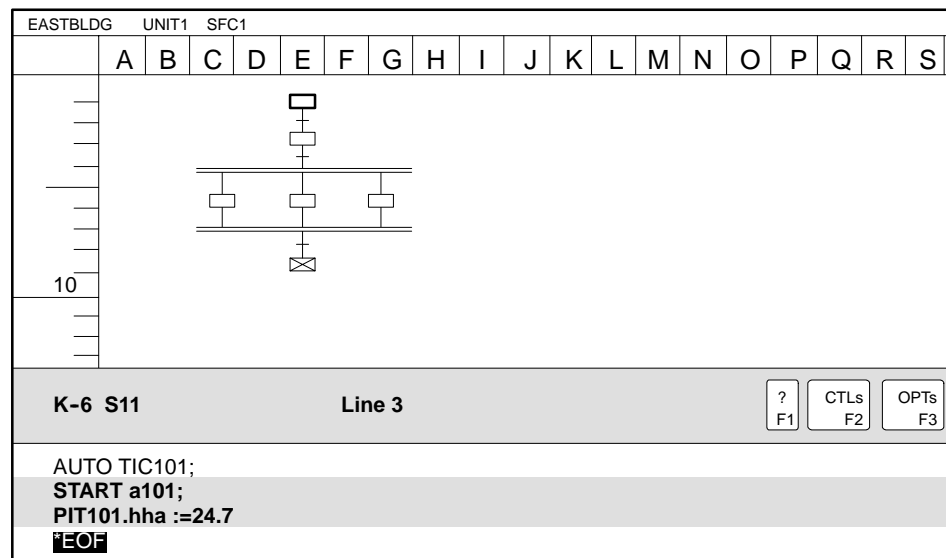


Figure 4-7 Marking a Block of Text

Using the SFC Sub-Editor (continued)

Changing Text

While you are in the SFC sub-editor, you have available the options listed in [Table 4-3](#).

Table 4-3 Options for SFC Sub-Editor

Command	Keys	Remarks
Copy Block	<input type="button" value="Alt"/> <input type="button" value="C"/>	Copies all text in a marked block to the current cursor position.
Delete Block	<input type="button" value="Alt"/> <input type="button" value="E"/>	Removes all text in the marked block.
Delete Line	<input type="button" value="Alt"/> <input type="button" value="D"/>	Removes one line of text.
Delete to End of Line	<input type="button" value="Alt"/> <input type="button" value="L"/>	Removes text from cursor position to end of line.
Find Text	<input type="button" value="Alt"/> <input type="button" value="F"/>	Locates a specific character string in the text.
Find Next	<input type="button" value="Alt"/> <input type="button" value="N"/>	Continues search initiated by Find text and locates next occurrence in the text.
Go to	<input type="button" value="Alt"/> <input type="button" value="G"/>	Pages the screen to display the portion of the text area (TOP or BOTTOM) that you select in the dialog window.
Insert Block from Buffer	<input type="button" value="Alt"/> <input type="button" value="K"/>	Moves marked block from buffer into text.
Insert Line	<input type="button" value="Alt"/> <input type="button" value="I"/>	Allows you to add one blank line at the current cursor position.
Join Line	<input type="button" value="Alt"/> <input type="button" value="J"/>	Combines two successive short text lines into a longer line.
Move Block	<input type="button" value="Alt"/> <input type="button" value="M"/>	Relocates all text in a marked block to the current cursor position.
Put Block to Buffer	<input type="button" value="Alt"/> <input type="button" value="B"/>	Places marked block in a buffer.
Replace Text	<input type="button" value="Alt"/> <input type="button" value="T"/>	Locates and replaces a character string in the text.
Set Parameters	<input type="button" value="Alt"/> <input type="button" value="P"/>	Allows you to specify the size of the text area, the amount of paging that occurs when you press PgUp or PgDn, and whether or not the editor inserts a line when you press Enter.
Split Line	<input type="button" value="Alt"/> <input type="button" value="S"/>	Breaks line at current cursor position and places remainder of that line in a new line.
Undelete Line	<input type="button" value="Alt"/> <input type="button" value="X"/>	Replaces deleted line of text.
Undo Changes	<input type="button" value="Alt"/> <input type="button" value="U"/>	Restores text to condition at start of the edit session and erases all modifications.

Chapter 5

Using the CFC Editor

5.1	Creating a CFC	5-2
	Overview	5-2
	Adding a CFC	5-2
	Accessing the CFC Editor	5-2
	Understanding Block Icons	5-2
	Using Block Icons	5-4
	Linking Control Blocks	5-4
	Connecting Blocks	5-5
	Disconnecting Blocks	5-5
5.2	Changing a CFC	5-6
	Overview	5-6
	Marking a Block	5-6
	Using the Options Icons	5-7
	Exiting a CFC	5-7
5.3	Using the Form Sub-Editor	5-8
	Editing a Form	5-8
	Using the Options Icon	5-9
5.4	Using the Math Sub-Editor	5-10
	Accessing the Math Editor	5-10
	Changing Text	5-11

5.1 Creating a CFC

Overview

In your control process, some functions need to operate continuously; for example, PID loops, interlock functions, analog alarms, and other math-related operations. To track these continuous functions, use a Continuous Function Chart (CFC). A CFC is a graphical worksheet in which you create graphic representations of continuous functions.

Each continuous function is represented graphically as a Continuous Function Block (CFB). Most CFBs have inputs and/or outputs, which are represented by arrows that indicate the direction of control to or from the CFB.

Adding a CFC

To add a CFC at the Unit Content Level, follow these steps.

1. At the Unit Content Level, highlight:

----- CFCs -----

2. Press , or use the **OPTs** icon to select **Add new CFC**.
3. Enter a CFC name and description in the dialog window, and press .

A CFC name can contain up to 8 alphanumeric characters. The name must include at least one letter and can include an underscore.

4. Press again to complete the action.

Accessing the CFC Editor

To access a CFC, follow these steps.

1. At the Unit Content Level, highlight the name of the CFC.
2. Press or or .

Understanding Block Icons

APT uses a grid to organize the CFC graphics. Each CFB requires an area four cells wide and two cells deep. With the graphical portion of the CFB, you specify the type of object, its name, and graphical connection to other objects.

APT provides eight types of control blocks. Each type is represented by a block icon. Two other block icons allow you to connect and disconnect blocks as explained on [page 5-4](#).

[Figure 5-1](#) shows the block icons and the key that you use to select each of the 45 predefined CFBs.

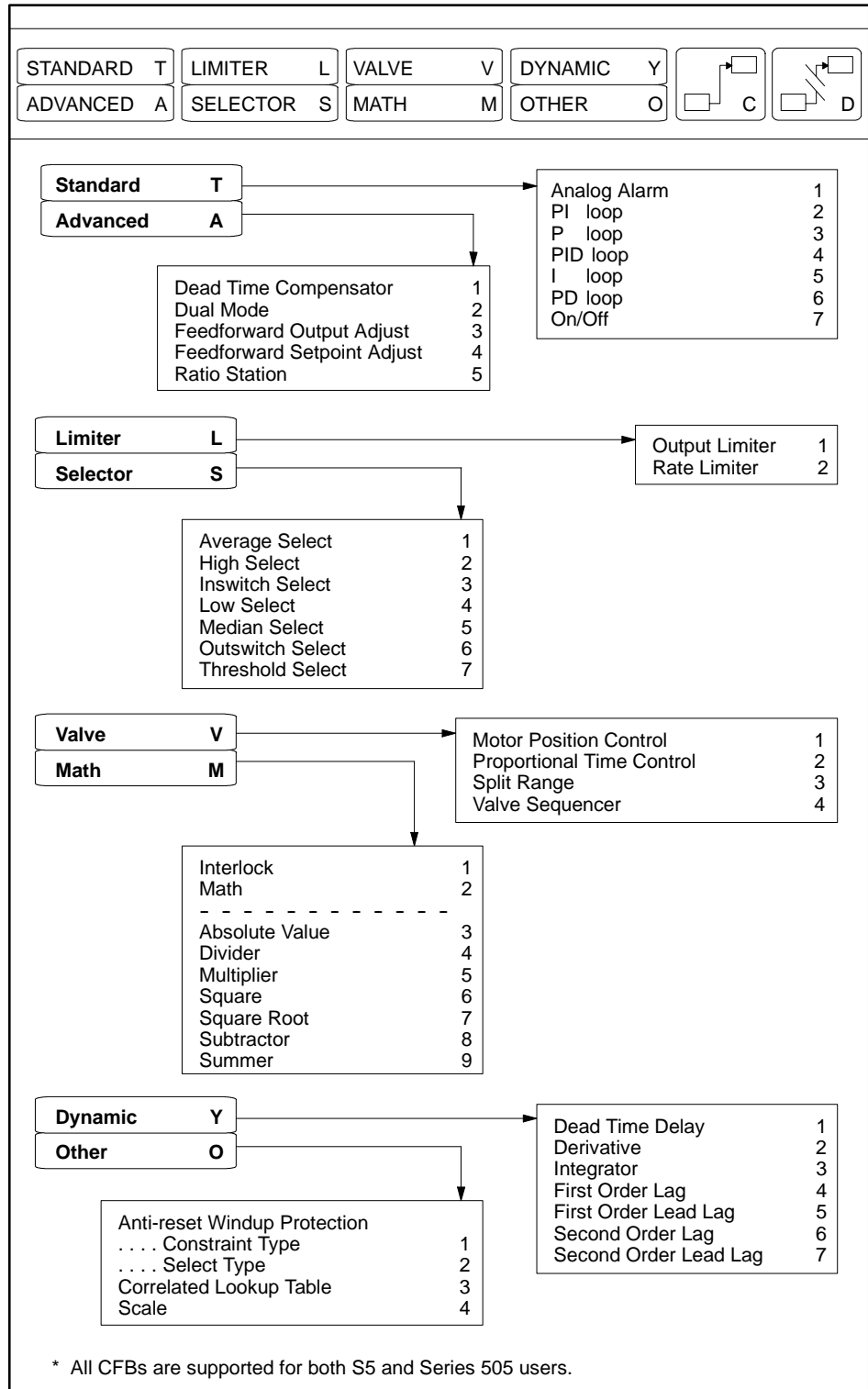


Figure 5-1 Selecting the Block Type

Creating a CFC (continued)

Using Block Icons

You select a block icon the same way you select other icons in APT: you press the associated letter keys, or you place the cross-hair over an icon and press the left button.

To place a CFB on a Continuous Function Chart, follow these steps.

1. Select the block type by pressing the associated key.

As long as a block icon is active, the cursor appears as a large box on the screen.

2. Press the number following the name that corresponds to the specific block you want.
3. Move the text cursor to the appropriate location on the chart and press **Enter**.
4. Enter the name of the CFB in the dialog window, and press **Enter**.

Linking Control Blocks

A CFC can consist of connected CFBs, unconnected CFBs, or a combination of connected and unconnected CFBs. Connection lines link function blocks together to indicate that the output of one CFB serves as the input to another.

When you connect two CFBs, the input of one block inherits the name of the output from the other block. For example, if you connect block1 to block2, the output variable for block1 is block1.OUTPUT, and the input variable for block2 is also block1.OUTPUT, as shown in [Figure 5-2](#).

You can change the name of a graphically connected input only by disconnecting the blocks or changing the name of the output in the form. If you modify either the form or the graphic in a set of connected blocks, the changes appear automatically throughout the control block configuration.

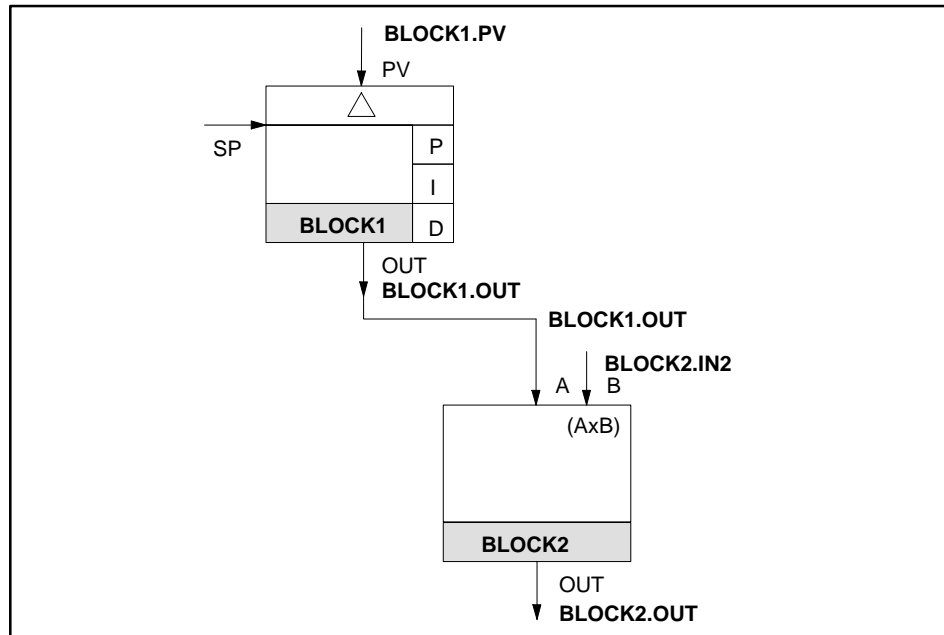


Figure 5-2 Connected Control Blocks

Connecting Blocks

To draw a connecting line between two CFBs, follow these steps.

1. Press **[C]**.
2. Move the text cursor to the appropriate arrow on one of the CFBs you want to connect, and press **[Enter]**.
3. Move the cursor along a path to connect with the arrow on the other CFB, and press **[Enter]**.

Disconnecting Blocks

To remove a connecting line between two blocks, follow these steps.

1. Press **[D]**.
2. Move the cursor to the appropriate arrow on either of the blocks you want to disconnect, and press **[Enter]**. The line is deleted.

As long as the C-block or D-block icon is active, the cursor is a small box on the screen. When you want to return the cursor to its usual size, either press **[C]** or **[D]** so that they become inactive, or else select another block type.

5.2 Changing a CFC

Overview

When you build a CFC, you have the controls and options available as shown in [Figure 5-3](#).

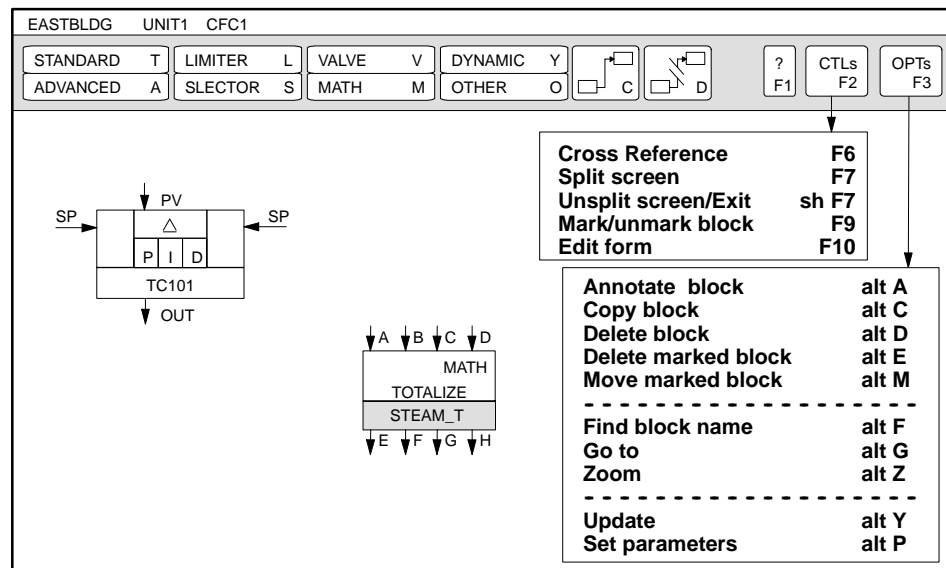


Figure 5-3 CFC Controls and Options

Marking a Block

After you create a CFC, you can use the **CTLs** icon to mark a section of the CFC that you want to delete, move, or copy.

To mark a portion of the CFC, follow these steps.

1. Position the cursor on the upper left cell of the block that you want to delete or move.
2. Press **F9**, or use the **CTLs** icon to select **Mark/unmark block**. A colored box marks the block.
3. Move the text cursor until the colored box covers the block you want to mark.
4. Press **Enter** or **F9**, or use the **CTLs** icon to select **Mark/unmark block**.

To copy, move, or delete the block of cells, use the **OPTs** icon, as explained in [Table 5-1](#).

Using the Options Icons

Table 5-1 lists the commands available using the options icon in the CFC Editor.

Table 5-1 Options for CFC Editor

Command	Keys	Remarks
Annotate Block	<input type="button" value="Alt"/> <input type="button" value="A"/>	Adds a comment to the Math or Interlock CFB graphical symbol.
Copy Block	<input type="button" value="Alt"/> <input type="button" value="C"/>	Duplicates the single marked block at the current cursor position along with the associated math.
Delete Block	<input type="button" value="Alt"/> <input type="button" value="D"/>	Removes the specified block along with the form and any associated math from the CFC. This command does not make any adjustments to remaining cells.
Delete Marked Block	<input type="button" value="Alt"/> <input type="button" value="E"/>	Removes the marked block from CFC.
Find Block Name	<input type="button" value="Alt"/> <input type="button" value="F"/>	Moves the cursor to the block that you specify in the dialog window.
Go To	<input type="button" value="Alt"/> <input type="button" value="G"/>	Scrolls the screen to display the portion of the CFC (TOP or BOTTOM) that you select in the dialog window.
Move Marked Block	<input type="button" value="Alt"/> <input type="button" value="M"/>	Relocates marked portion of CFC.
Set Parameters	<input type="button" value="Alt"/> <input type="button" value="P"/>	Allows you to specify a percentage of the screen to move up, down, right, or left when you page through the displayed CFC. This affects paging with the PgUp, PgDn, Home, and End keys.
Update	<input type="button" value="Alt"/> <input type="button" value="Y"/>	Saves the displayed CFC to disk without leaving the edit session.
Zoom	<input type="button" value="Alt"/> <input type="button" value="Z"/>	Graphically displays the names of inputs and outputs; displays block diagrams of advanced blocks.

Exiting a CFC

APT automatically saves your entries when you exit the CFC. To exit a CFC, follow these steps.

1. Press .
2. To return to the hierarchy, press .

5.3 Using the Form Sub-Editor

Editing a Form

APT provides a CFC sub-editor to complete the forms associated with each block. You use this Form Editor to enter additional information to define a CFB and to change the system-supplied input and output names. Some entry fields in a form cannot be altered because they are graphically connected in the CFC; restricted fields are marked with the following symbol: <--->

To edit a CFB form, follow these steps.

1. Position the text cursor over the CFC block you want to edit.
2. Press **F10**, or use the **CTLs** icon to select **Edit Form**.
3. Complete the information in the required entry fields.

To select an item followed by brackets [], place an **X** within the brackets. To cancel a selection, place the cursor on the **X** and press the spacebar.

4. To exit the sub-editor, press **Esc**.
5. When you exit, the sub-editor automatically saves the form and returns to the CFC.

Figure 5-4 shows the controls and options that are available in the form sub-editor.

The screenshot displays the CFC Form Sub-Editor for a PID loop block (TC101). The interface is divided into several sections:

- Block Diagram:** Shows a PID loop with inputs SP (Setpoint) and PV (Process Variable), and output OUT. The controller is labeled TC101.
- Menu:**
 - Find text (alt F)
 - Find next (alt N)
 - Replace text (alt T)
 - Delete math text (alt M)
 - Go to (alt G)
 -
 - Set parameters (alt P)
 - Undo changes (alt U)
- Keyboard Shortcuts:**
 - Cross Reference (F6)
 - Split screen (F7)
 - Unsplit screen/Exit (shF7)
 - Edit math text (F10)
- Block Information:**
 - PID loop TC101**
 - Name: TC101
 - Description:
 - Sample time: _____ sec
 - Algorithm type: POSITION VELOCITY
 - User Assigned Address?: []
- Process Variable Information:**
 - Process variable: TC101.PV ●

Figure 5-4 Using the CFC Form Sub-Editor

Using the Options Icon

Table 5-2 lists the commands available using the **OPTs** icon.

Table 5-2 Options for CFC Form Sub-Editor

Command	Keys	Remarks
Delete Math Text	<input type="button" value="Alt"/> <input type="button" value="M"/>	Removes associated math text from CFB.
Find Text	<input type="button" value="Alt"/> <input type="button" value="F"/>	Locates a specific character string in the form.
Find Next	<input type="button" value="Alt"/> <input type="button" value="N"/>	Continues search initiated by Find text and locates next occurrence in the text file.
Go to	<input type="button" value="Alt"/> <input type="button" value="G"/>	Pages the screen to display the portion of the file (TOP or BOTTOM) that you select in the dialog window.
Replace Text	<input type="button" value="Alt"/> <input type="button" value="T"/>	Locates and replaces a character string in the form.
Set Parameters	<input type="button" value="Alt"/> <input type="button" value="P"/>	Allows you to specify the size of the form area and the number of lines that page when you press PgUp or PgDn.
Undo Changes	<input type="button" value="Alt"/> <input type="button" value="U"/>	Restores form to condition at start of the edit session and erases all modifications.

5.4 Using the Math Sub-Editor

Accessing the Math Editor

In some forms, you have the option of programming associated math for a CFB. For example, the PID block has options for PV Cycle, Auto, Cascade, and Output. To specify associated math and enter the math statements, follow these steps. The procedure may vary slightly, depending on the type of CFB.

1. Place an **X** between the brackets following the appropriate math type.
2. With the cursor on the appropriate math type, press **F10**.

Another form appears at the bottom of the screen with controls and options available as shown in [Figure 5-5](#).

3. Enter the appropriate math statements. The text in the CFC math statements cannot include a tilde (~) or a caret (^). In a CFB, a MATH statement is not needed.
4. To exit the sub-editor, press **F10** or **Esc**.

When you exit, the sub-editor automatically saves your entries and returns to the form.

To enter math statements in a Math or Interlock CFB, press **F10** and continue with steps 3 and 4 above.

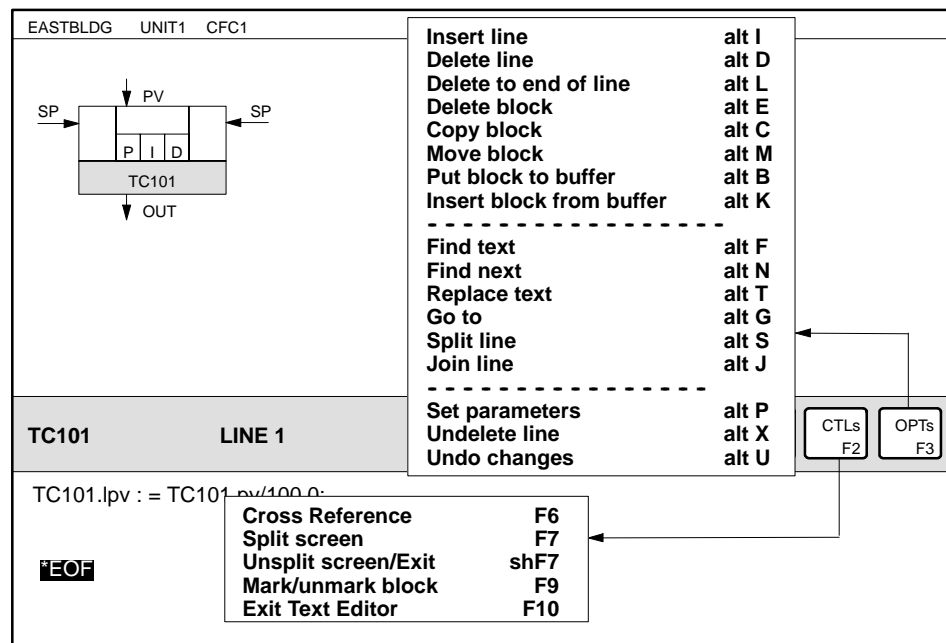


Figure 5-5 Using the CFC Math Sub-Editor

Changing Text

While you are in the CFC math editor, the options listed in [Table 5-3](#) are available. Use **F9** to mark the beginning and ending of a block of text that you want to copy, delete, move, or put in a buffer. You can copy a block to a buffer and use it later, for instance, in another CFC or in a report.

Table 5-3 Options for CFC Math Sub-Editor

Command	Keys	Remarks
Copy Block	Alt C	Copies all text in a marked block to the current cursor position.
Delete to End of Line	Alt L	Removes text from cursor position to end of line.
Delete Block	Alt E	Removes all text in the marked block.
Delete Line	Alt D	Removes one line of text.
Find Text	Alt F	Locates specific character string in the text.
Find Next	Alt N	Continues search initiated by Find text and locates next occurrence in the text.
Go to	Alt G	Pages the screen to display the portion of the text area (TOP or BOTTOM) that you select in the dialog window.
Insert Block from Buffer	Alt K	Moves marked block from buffer into text
Insert Line	Alt I	Allows you to add one blank line at the current cursor position.
Join Line	Alt J	Combines two successive short text lines into one line.
Move Block	Alt M	Relocates all text in a marked block to the current cursor position.
Put Block to Buffer	Alt B	Places marked block in buffer.
Replace Text	Alt T	Locates and replaces a character string in the text.
Set Parameters	Alt P	Allows you to specify the size of the text area, whether or not the editor inserts a line when you press ENTER, and the amount of paging that occurs when you press PgUp or PgDn.
Split Line	Alt S	Breaks line at current cursor position and places remainder of that line in a new line.
Undelete Line	Alt X	Replaces deleted line of text.
Undo Changes	Alt U	Restores text to condition at start of the edit session and erases all modifications.

Compiling an APT Program

6.1	Understanding the Compile Operation	6-2
	Compile Phases	6-2
	Compile Options	6-2
6.2	Compiling for Series 505 Controllers	6-4
	Understanding the Compiler Control File	6-4
	Editing the Compiler Control File	6-15
	Starting the Compile Operation	6-17
	Compile Screen	6-18
	Using the Compile Report	6-19
6.3	Compiling for S5 Controllers	6-20
	Understanding the Compiler Control File	6-20
	Operating System Defaults (928B CPU)	6-24
	Error Processing Checks (928B CPU)	6-25
	RBE Options (928B CPU)	6-26
	Operating System Defaults (948 CPU)	6-28
	RBE Options (948 CPU)	6-30
	Editing the Compiler Control File (928B and 948 CPUs)	6-34
	Completing the Compiler Control File (928B CPU)	6-36
	Completing the Compiler Control File (948 CPU)	6-37
	Starting the Compile Operation	6-38
	Using the Compile Screen	6-39
	Using the Compile Report	6-39
6.4	Object File Reports	6-41
	Overview	6-41
	Using Object Reports	6-42
	Deleting Object Code	6-43
6.5	Downloading Program to Controller	6-44
	When to Use the Incremental Download	6-45
	Quick Verification Option	6-46
	Obstacles to Incremental Download	6-46
	Overflow CRs and Incremental Download	6-47
	Incremental Download for S5 Redundant System	6-47
	Incremental Download and RUN Mode	6-48
	Downloading a Program from APT	6-50
	Downloading a Program from DOS (Series 505)	6-54
	Downloading a Program from DOS (S5)	6-56
6.6	Verifying a Downloaded Program	6-57
	Understanding Verify	6-57
	Using Verify	6-58

6.1 Understanding the Compile Operation

Compile Phases

The process of compiling a program includes these seven phases.

- Phase 1: Coordinate Compile
- Phase 2: Validation and Package Expand
- Phase 3: SFC Compiler
- Phase 4: NNL Compiler
- Phase 5: Assembler Directive Pre-processor
- Phase 6: Assembler
- Phase 7: Translate File Build (Optional)

The validation process in Phase 2 confirms that definitions, graphic connections, math syntax, and other program items are accurate. If errors exist in the program, they usually are found during this phase or in Phase 3 when the SFCs are validated.

NOTE: When errors are identified, the compile process stops; therefore, not all errors are found the first time that you compile a program.

Compile Options

When you compile a program, APT provides options that determine how the program is compiled by the system. [Figure 6-1](#) through [Figure 6-9](#) show the Compiler Control Files for the Series 505 controllers. [Figure 6-11](#) and [Figure 6-12](#) show the Compiler Control Files for the S5 controllers. For a detailed description of the Compiler Control File, see [Section 6.2](#) for Series 505 controllers and [Section 6.3](#) for S5 controllers.

Force compile is an option that allows you to specify how much of the program needs to be recompiled.

If you choose a force compile, APT compiles all the units that you mark, whether they have changed since the last compile or not.

If you do not choose a force compile, APT selects the parts of the program code to compile. If a portion of the program has not been changed since the last compile, APT does not recompile that portion. When you are ready to download the program to the controller, you can request that APT do an incremental download. Only the changed code, including added or deleted objects, is downloaded.

Build Translate Table reallocates all OSx tags for OSx scan. You have three options:

- When you select **No**, existing marked tags are not modified and Phase 7 of the compile, Translate File Build, is not executed.
- When you select **Yes**, all marked tags are placed in the **install.tag** file, which is sent to OSx when translating.
- When you select **Append**, all marked tags are placed in the **install.tag** file, and tags marked since the last translate build are also placed in a separate file, **append.tag**.

Refer to [Section 8.4](#) for detailed information about these options.

NOTE: If you use the Transoff program to translate tags as described in [Chapter 8](#) of this manual, phase 7 of the compile operation warns you of tag mismatches. For non-OSx operator stations, you should generate a new **symbol.rpt** or **tag.rpt** report and check for new addresses.

Create debug version is an option that provides additional features (breakpoint, single step, step logger) to use in debugging your program for Series 505 controllers. These additional features are not supported for S5. If you do not select this option, the breakpoint, single scan/step, and step logger functions for SFCs are not available within the Debug Utility. All other debug functions can be used for both Series 505 and S5 controllers, even when you do not select this option. Your program file is larger and requires more controller memory when you select this option. See [Chapter 7](#) for more information about debugging an APT program.

Units to compile is an option that allows you to include all units in the compile, or to mark individual units through a dialog window. This allows you to identify the units that you want to include in the object file that is downloaded to the controller.

Reserved locations is an option that allows you to specify what you want to reserve for each listed memory type. This option allows you to use existing programs already resident in the controller. APT does not use reserved memory, and when you download a program, reserved memory locations are not changed or cleared.

6.2 Compiling for Series 505 Controllers

This section explains how to enter data in the Compiler Control File for a Series 505 controller and how to run the compile process.

Understanding the Compiler Control File

Before compiling a program, you must fill out the Compiler Control File. File content depends upon the controller model, as shown in [Figure 6-1](#) through [Figure 6-9](#) on [pages 6-8](#) through [6-14](#). Field definitions are listed below. Refer to the Controller Support section of the APT Overview chapter in the *SIMATIC APT Programming Reference (Tables) Manual*.

Controller type Enter the controller type, e.g., 545, 555, 575, etc., preceded by TI. For a 565-2120 card with a 560-2820 controller, enter TI565T. For a 565-2820 card with a firmware release 6.x 560-2820 controller, enter TI565P. For a 545-1103 (545 Lite), enter TI545L.

Controller name The controller name is optional unless you are using OSx, in which case this is the name of the secondary or association name specified in the OSx Network Configuration. OSx allows these characters: letters A-Z (upper case only), numbers 0-9, the period (.), the underscore (_), and the hyphen (-). Do not use the space character in the name; APT treats it as a delimiter.

NOTE: When you use the Industrial Ethernet network, the controller name field entry should be the (or one of the) association names that you entered during the network configuration process.

Controller release (field not displayed for the 560, 565, or 565P) Enter the firmware release of the controller. Refer to the Controller Support section of the APT Overview chapter in the *SIMATIC APT Programming Reference (Tables) Manual* for a list of the controller firmware releases supported by APT.

Application ID (field displayed for the 575 only) Enter the letter for this controller's application ID in the backplane.

OSx/PCS release Enter the appropriate OSx/PCS release number for which tags will be installed. In this instance, PCS release 1.x and 2.x refer to TISTAR 1.x and 2.x, PCS 3.x refers to SIMATIC PCS 3.x, and PCS 4.x refers to SIMATIC OSx 4.x. PCS 3.x and OSx 4.x only support Release 3.x or greater of the following controllers: 575-2103/2104/2105/2106, 555-1101 to -1106, and 545-1102 to -1106. OSx/PCS does not support the 560 or 565 series.

Number of RCC cards (field displayed for the 560/565/T/P only) Enter the number of Remote Channel Controller cards that you have installed in your hardware system; the default is 1.

Memory size Enter the total memory available in your system. The total memory configurations for the various controllers are listed in [Table 6-1](#). You can also use the **P/C AUX** function in Debug or **Report Controller** in MAITT to determine your controller memory configuration.

Table 6-1 Controller Memory Configuration

Controller Type	K Bytes	Controller Configuration	Controller Type	K Bytes	Controller Configuration
545 2.1	192	545-1101 ¹	560T/565T/565P	351	Standard configuration
545 2.1	448	545-1101 + Memory Expansion Module	560T/565T/565P	479	+ 64 Kw card
545 3.x	192	545-1102	560T/565T/565P	607	+ 128 Kw card
545 4.x	192	545-1104	560T/565T/565P	735	+ 64 Kw card , + 128Kw card
545 5.x	192	545-1106	560T/565T/565P	767	+ 208 Kw card
545L 4.x	96	545-1103	560T/565T/565P	863	+ 256 Kw card
545L 5.x	96	545-1105	560T/565T/565P	895	+ 64 Kw card, + 208 Kw card
555 3.x	384	555-1101	560T/565T/565P	991	+ 64 Kw card, + 256 Kw card
555 3.x	1920	555-1102	560T/565T/565P	1023	+ 128 Kw card, + 208 Kw card
555 4.x	384	555-1103	560T/565T/565P	1023	+ 128 Kw card + 208 Kw card
555 4.x	1920	555-1104	575 3.x	320	575-2102 ²
555 5.x	384 1856	555-1105, 555-1106	575 3.x	1024	575-2103
560/565	96	Standard configuration	575 4.x	832	575-2104
560/565	224	+ 64 Kw card	575 5.x	832 1856	575-2105, 575-2106
560/565	352	+ 128 Kw card	CPU928B	64	928-3UB12 + 64 Kb RAM Module
560/565	480	+ 64 Kw card, + 128 Kw card	CPU948/948R	640	948-3UA11/948-3UR11
560/565	512	+ 208 Kw card	CPU948/948R	1664	948-3UA21/948-3UR21

- 1 If you have a PPX:545-1101 with no Memory Expansion Module installed (192 Kbytes available memory), you must exit the Compiler Control File and leave APT. Use the DOS **copy** command to copy the information in the file `\apt\lib505\ref\545.192` to the configuration file `\apt\lib505\ref\545.2x` before re-entering APT and the Compiler Control File.
- 2 If you have a PPX:575-2102 installed (320 Kbytes available memory), you must exit the Compiler Control File and leave APT. Use the DOS **copy** command to copy the information in the file `\apt\lib505\ref\575.320` to the configuration file `\apt\lib505\ref\575.cfg` before re-entering APT and the Compiler Control File.

Force compile Mark **X** or leave blank.

An **X** indicates that you want the system to compile the entire program.

Blank indicates that you want the system to compile only those portions of the code that have changed since the last compile. In the Download Dialog Window, **Download Incremental Changes Only** becomes one of the default options.

Create debug version Mark **X** or leave blank.

An **X** indicates that you want to use breakpoint, single scan/step, and step logger options for SFCs when you debug your program. The compiled program file is larger when you select this option.

The blank field indicates these options will not be available for debugging.

You do not need to **X** this field in order to use Debug.

Build translate table Select **Yes**, **No**, or **Append** to allocate OSx tags for optional OSx scan. Refer to [Section 8.4](#) for detailed information about these options.

If you select **No**, the existing marked tags keep the same memory locations reserved at the last Build Translate Table compile. Tags that were added or removed are not updated in the compile. Phase 7, Translate Build, does not execute.

If you select **Yes**, all marked tags are placed into the OSx format. Memory locations are blocked optimally for the most efficient access. You may have to reassign some OSx names if you have added some tags.

If you select **Append**, the tags that you have added are appended to existing files containing previously translated tags. Memory locations for appended information are blocked optimally. However, because they are not integrated into existing blocks, memory allocation for all tags (old and appended tags) is not blocked optimally. If you have deleted tags, the memory locations that were assigned to them are not available and cannot be used until you select the **Yes** option for the build translate table.

Uninterruptible Power Supply (UPS) Select this option when you have a UPS and want to remove power fail recovery logic from the code of program and device objects.

Do not convert TOD to integer fields Select this option when you want to remove logic that converts the time status (Time of Day) words of the controller to program extensions in V-Memory locations.

Units to compile To include all units, place an **X** between the brackets; or, to mark individual units, press **[F10]** to open the window and place an **X** in front of the units that you want to include.

Reserved locations This option allows you to reserve memory locations by specifying an upper limit of memory for your use. For example, if you reserve 100 words in Ladder memory for your own use, APT begins assigning memory locations at address 101. When you download a program, reserve memory locations are not changed or cleared. Most APT applications do not use the Reserve Memory option, but rather let APT determine the memory allocations. Refer to the appendix on direct memory addressing in the *SIMATIC APT Programming Reference (Graphics/Math) Manual* for more information.

U-memory filename (field displayed for the 545, the 545L, the 555, and the 575 only). Enter the name of the user program stored in the `..\program\program_name\pr` directory.

Report by exception (field only displayed for Release 3.x or greater of the 545, 545L, 555, and 575 CPUs). Enter the number of Kbytes needed for RBE data for OSx. The default allocation of 20 Kbytes is the minimum allocation. Any entry of less than 20 K results in a 20 Kbyte allocation.

TI5x5 Time Line Configuration (field displayed for the 545, 555 and 575 only). You can assign a value for each time slice in the analog portion of the scan for the controller. Refer to the *SIMATIC 505 Programming Manual* for the details of each field.

Application Dependencies (field displayed for the 575 only) Allows you to configure the interactions between the applications. Enter the appropriate application letters next to each other, without spaces.

Required Applications must be installed on the backplane before the current application can transition to the run mode.

Optional Applications interact with the current application. They do not have to be installed on the backplane before the current application can transition to the run mode.

Mode-locked Applications causes all of the listed applications to transition to the same mode when one of the listed applications changes mode.

NOTE: To reference G-memory from another application, you must identify that application in either the required application or in the optional application list. If you do not, the controller does not go into run mode.

Compiling for Series 505 Controllers (continued)

Figure 6-1 through Figure 6-9 on pages 6-8 through 6-14 show the contents of the Compiler Control Files for the various controller types.

```

Controller type:  TI545 •
Controller name:  TI545
Controller release:  5.x •
PCS release:      4.x •

Memory size:      192 Kbytes
Force compile?    [X]
Create debug version? [ ]
Build translate table? No Append Yes
Uninterruptible power (UPS)? [ ]
Do not convert TOD to integer fields? [ ]

Units to compile -----
Include all?      [ ] 0 of 0 included
                  (Press F10 to mark individual units)

Reserved locations -----
Ladder (L): 0 Words Nonretentive cr's: 0
Variable (V): 0 Words Retentive cr's: 0
Constant (K): 0 Words Timer/Counters: 0
Special Memory (S): 0 Words Drum/Edrums: 0
U-Memory (U): 0 Words Shift registers: 0
Loops: 0 Table Moves: 0
Analog Alarms: 0 One Shots: 0
SF programs: 0 Dset: 0
SF subroutines: 0 Tset: 0
Report by Exception: 0 Kbytes (if Rel. 3.x or 4.x)

U-Memory filename: _____

TI545 time line configuration -----
Scan type: Fixed Variable Upper Limit
Total scan time: 0 MSEC
Loop time slice: 34 MSEC
Analog alarm time slice: 6 MSEC
Cyclic SF program time slice: 4 MSEC
Priority SF program time slice: 4 MSEC
Normal SF program time slice: 2 MSEC
Ladder SF sub time slice: 1 MSEC
Normal communication time slice: 2 MSEC
Priority communication time slice: 3 MSEC
Ladder SF sub zero: 1 MSEC
Report by Exception time slice: 5 MSEC (if Rel. 3.x or 4.x)
Cyclic RLL scan time: 32767 MSEC
----- End of Form -----

```

Figure 6-1 Compiler Control File — 545


```

Controller type:  TI545L •
Controller name:  TI545L
Controller release:  4.x •
PCS release:      4.x •

Memory size:      _96 Kbytes
Force compile?    [X]
Create debug version? [ ]
Build translate table? No Append Yes
Uninterruptible power (UPS)? [ ]
Do not convert TOD to integer fields? [ ]

Units to compile -----
Include all?      [ ] 0 of 0 included
                  (Press F10 to mark individual units)

Reserved locations -----
Ladder (L): _0 Words Nonretentive cr's: _0
Variable (V): _0 Words Retentive cr's: _0
Constant (K): _0 Words Timer/Counters: _0
Special Memory (S): _0 Words Drum/Edrums: _0
U-memory (U): _0 Words Shift registers: _0
Loops: _0 Table Moves: _0
Analog Alarms: _0 One Shots: _0
SF programs: _0 Dset: _0
SF subroutines: _0 Tset: _0
Report by Exception: _0 Kbytes

U-Memory filename: _____

TI545L time line configuration -----
Scan type: Fixed Variable Upper Limit
Total scan time: _0 MSEC
Loop time slice: _34 MSEC
Analog alarm time slice: _6 MSEC
Cyclic SF program time slice: _4 MSEC
Priority SF program time slice: _4 MSEC
Normal SF program time slice: _2 MSEC
Ladder SF sub time slice: _1 MSEC
Normal communication time slice: _2 MSEC
Priority communication time slice: _3 MSEC
Ladder SF sub zero: _1 MSEC
Report by Exception time slice: _5 MSEC
Cyclic RLL scan time: _32767 MSEC

----- End of Form -----

```

Figure 6-2 Compiler Control File — 545L

Compiling for Series 505 Controllers (continued)

```

Controller type:  TI555 •
Controller name:  TI555
Controller release:  5.x
PCS release:     4.x •

Memory size:     384 Kbytes
Force compile?   [X]
Create debug version? [ ]
Build translate table? No Append Yes
Uninterruptible power (UPS)? [ ]
Do not convert TOD to integer fields? [ ]

Units to compile -----
Include all?    [ ] 0 of 0 included
                (Press F10 to mark individual units)

Reserved locations -----
Ladder (L):    0 Words           Nonretentive cr's: 0
Variable (V):  0 Words           Retentive cr's:   0
Constant (K):  0 Words           Timer/Counters:  0
Special Memory (S): 0 Words       Drum/Edrums:     0
U-Memory (U):  0 Words           Shift registers:  0
Loops:         0                  Table Moves:     0
Analog Alarms: 0                 One Shots:       0
SF programs:   0                  Dset:            0
SF subroutines: 0                 Tset:            0
Report by Exception: 0 Kbytes

U-Memory filename: _____

TI555 time line configuration -----
Scan type:      Fixed   Variable   Upper Limit
Total scan time:  0 MSEC
Loop time slice: 34 MSEC
Analog alarm time slice: 6 MSEC
Cyclic SF program time slice: 4 MSEC
Priority SF program time slice: 4 MSEC
Normal SF program time slice: 2 MSEC
Ladder SF sub time slice: 1 MSEC
Normal communication time slice: 2 MSEC
Priority communication time slice: 3 MSEC
Ladder SF sub zero: 1 MSEC
Report by Exception time slice: 5 MSEC
Cyclic RLL scan time: 32767 MSEC
----- End of Form -----

```

Figure 6-3 Compiler Control File — 555

```

Controller type:  TI575 •
Controller name:  TI575
Controller release:  5.x •
Application Id:  __ (A-Z)
PCS release:  4.x •

Memory size:  __320 Kbytes
Force compile?  [X]
Create debug version?  [ ]
Build translate table?  No Append Yes
Uninterruptible power (UPS)?  [ ]
Do not convert TOD to integer fields?  [ ]

Units to compile -----
Include all?  [ ] 0 of 0 included
(Press F10 to mark individual units)

Reserved locations -----
Ladder (L):__0 Words Nonretentive cr's:__0
Variable (V):__0 Words Retentive cr's:__0
Constant (K):__0 Words Timer/Counters:__0
Special Memory (S):__0 Words Drum/Edrums:__0
U-Memory (U):__0 Words Shift registers:__0
Loops:__0 Table Moves:__0
Analog Alarms:__0 One Shots:__0
SF programs:__0 Dset:__0
SF subroutines:__0 Tset:__0
Report by Exception:__0 Kbytes

U-Memory filename:_____

TI575 time line configuration -----
Scan type: Fixed Variable Upper Limit
Total scan time: 0 MSEC
Loop time slice: 34 MSEC
Analog alarm time slice: 6 MSEC
Cyclic SF program time slice: 4 MSEC
Priority SF program time slice: 4 MSEC
Normal SF program time slice: 2 MSEC
Normal communication time slice: 5 MSEC
Ladder SF sub time slice: 1 MSEC
Normal communication time slice: 2 MSEC
Priority communication time slice: 3 MSEC
Ladder SF sub zero: 1 MSEC
Network communication time slice: 5 MSEC
Report by Exception time slice: 5 MSEC
Cyclic RLL scan time: 32767 MSEC

Application Dependencies -----

Required Applications: _____
Optional Applications: _____
Mode-locked Applications: _____

----- End of Form -----

```

Figure 6-4 Compiler Control File — 575

Compiling for Series 505 Controllers (continued)

Controller type:	<u>T1560</u> •
Controller name:	<u>T1560</u>
PCS release:	<u>2.x</u> •
Number of RCC cards:	<u>1</u>
Memory size:	<u>96</u> Kbytes
Force compile?	<input checked="" type="checkbox"/> [X]
Create debug version?	<input type="checkbox"/> []
Build translate table?	<u>No</u> Append Yes
Uninterruptible power (UPS)?	<input type="checkbox"/> []
Do not convert TOD to integer fields?	<input type="checkbox"/> []

Units to compile	-----
	Include all? <input type="checkbox"/> [] 0 of 0 included (Press F10 to mark individual units)

Reserved locations	-----
Ladder (L): <u>0</u> Words	Nonretentive cr's: <u>0</u>
Variable (V): <u>0</u> Words	Retentive cr's: <u>0</u>
Constant (K): <u>0</u> Words	Timer/Counters: <u>0</u>
Role swap: <u>0</u>	Drum/Edrums: <u>0</u>
	Shift registers: <u>0</u>
	Table Moves: <u>0</u>
	One Shots: <u>0</u>
	Dset: <u>0</u>
	Tset: <u>0</u>

----- End of Form -----	

Figure 6-5 Compiler Control File — 560

Controller type:	<u>T1560T</u> •
Controller name:	<u>T1560T</u>
Controller release:	<u>6.x</u> •
PCS release:	<u>2.x</u> •
Number of RCC cards:	<u>1</u>
Memory size:	<u>351</u> Kbytes
Force compile?	<input checked="" type="checkbox"/> [X]
Create debug version?	<input type="checkbox"/> []
Build translate table?	<u>No</u> Append Yes
Uninterruptible power (UPS)?	<input type="checkbox"/> []
Do not convert TOD to integer fields?	<input type="checkbox"/> []

Units to compile	-----
	Include all? <input type="checkbox"/> [] 0 of 0 included (Press F10 to mark individual units)

Reserved locations	-----
Ladder (L): <u>0</u> Words	Nonretentive cr's: <u>0</u>
Variable (V): <u>0</u> Words	Retentive cr's: <u>0</u>
Constant (K): <u>0</u> Words	Timer/Counters: <u>0</u>
Role swap: <u>0</u>	Drum/Edrums: <u>0</u>
	Shift registers: <u>0</u>
	Table Moves: <u>0</u>
	One Shots: <u>0</u>
	Dset: <u>0</u>
	Tset: <u>0</u>

----- End of Form -----	

Figure 6-6 Compiler Control File — 560T

```

Controller type: TI565 •
Controller name: TI565
PCS release: 2.x •

Number of RCC cards: 1
Memory size: 96 Kbytes
Force compile?  [X]
Create debug version?  [ ]
Build translate table? No Append Yes
Uninterruptible power (UPS)?  [ ]
Do not convert TOD to integer fields?  [ ]

Units to compile -----
Include all?  [ ] 0 of 0 included
(Press F10 to mark individual units)

Reserved locations -----
Ladder (L): 0 Words Nonretentive cr's: 0
Variable (V): 0 Words Retentive cr's: 0
Constant (K): 0 Words Timer/Counters: 0
Special Memory (S): 0 Words Drum/Edrums: 0
Loops: 0 Shift registers: 0
Analog Alarms: 0 Table Moves: 0
SF programs: 0 One Shots: 0
SF subroutines: 0 Dset: 0
Role swap: 0 Tset: 0
----- End of Form -----

```

Figure 6-7 Compiler Control File — 565

```

Controller type: TI565T •
Controller name: TI565T
Controller release: 6.x •
PCS release: 2.x •

Number of RCC cards: 1
Memory size: 351 Kbytes
Force compile?  [X]
Create debug version?  [ ]
Build translate table? No Append Yes
Uninterruptible power (UPS)?  [ ]
Do not convert TOD to integer fields?  [ ]

Units to compile -----
Include all?  [ ] 0 of 0 included
(Press F10 to mark individual units)

Reserved locations -----
Ladder (L): 0 Words Nonretentive cr's: 0
Variable (V): 0 Words Retentive cr's: 0
Constant (K): 0 Words Timer/Counters: 0
Special Memory (S): 0 Words Drum/Edrums: 0
Loops: 0 Shift registers: 0
Analog Alarms: 0 Table Moves: 0
SF programs: 0 One Shots: 0
SF subroutines: 0 Dset: 0
Role swap: 0 Tset: 0
----- End of Form -----

```

Figure 6-8 Compiler Control File — 565T

Compiling for Series 505 Controllers (continued)

Controller type:	<u>T1565P</u> •
Controller name:	<u>T1565P</u>
PCS release:	<u>2.x</u> •
Number of RCC cards:	<u> 1</u>
Memory size:	<u> 351</u> Kbytes
Force compile?	<input checked="" type="checkbox"/> [X]
Create debug version?	<input type="checkbox"/> []
Build translate table?	<u>No</u> Append Yes
Uninterruptible power (UPS)?	<input type="checkbox"/> []
Do not convert TOD to integer fields?	<input type="checkbox"/> []

Units to compile	-----
	Include all? <input type="checkbox"/> [] 0 of 0 included
	(Press F10 to mark individual units)

Reserved locations	-----
Ladder (L):	<u> 0</u> Words Nonretentive cr's: <u> 0</u>
Variable (V):	<u> 0</u> Words Retentive cr's: <u> 0</u>
Constant (K):	<u> 0</u> Words Timer/Counters: <u> 0</u>
Special Memory (S):	<u> 0</u> Words Drum/Edrums: <u> 0</u>
Loops:	<u> 0</u> Shift registers: <u> 0</u>
Analog Alarms:	<u> 0</u> Table Moves: <u> 0</u>
SF programs:	<u> 0</u> One Shots: <u> 0</u>
SF subroutines:	<u> 0</u> Dset: <u> 0</u>
Role swap:	<u> 0</u> Tset: <u> 0</u>
----- End of Form -----	

Figure 6-9 Compiler Control File — 565P

Editing the Compiler Control File

To configure the control file, follow the steps below. As you edit the control file, you can press to cancel all changes before you exit the form.

1. Position the text cursor on the following line in the Compile section of the Program Content Directory.

CONTROL [] Configuration for compile

2. Press or or .
3. Enter the controller type. The rest of the control file is displayed after you press .

NOTE: If you have already entered modules into the Module Definition Table, you cannot enter a controller model that does not support the channel containing these modules. For example, if you added modules to channel 4, you cannot enter 545 into the Compiler Control File. You must first reenter the Module Table and delete the modules from the unsupported channel. Then you can configure the Compiler Control File for the appropriate controller.

4. If you intend to translate the program or communicate with OSx, specify the controller name used in OSx. If you intend to communicate to the controller using TIWAY, specify the NIM address. If you intend to communicate to the controller using the Industrial Ethernet network, specify the association name. Association names must be in upper case.
5. For 545/545L/555/575/560T/565T, specify the controller release. Refer to the section on controller support in the APT overview chapter of the *APT Programming Reference (Tables) Manual* for a list of the controller firmware releases supported by APT.
6. For 575, specify the application ID of the controller in the backplane.
7. If you intend to translate the program to OSx, specify the OSx release.
8. For 560/565/565T/565P, specify the number of RCC cards that are installed in your system.

Compiling for Series 505 Controllers (continued)

9. Specify in Kbytes the total memory size available in your system. Memory size depends on the number of global memory cards that you have installed, but also includes memory in the main CPU. The defaults are as follows:

- 545L — 96 Kbytes
- 545 — 192 Kbytes
- 555 — 384 Kbytes
- 575 — 320 Kbytes
- 560T/565T/565P — 351 Kbytes
- 560/565 — 96 Kbytes

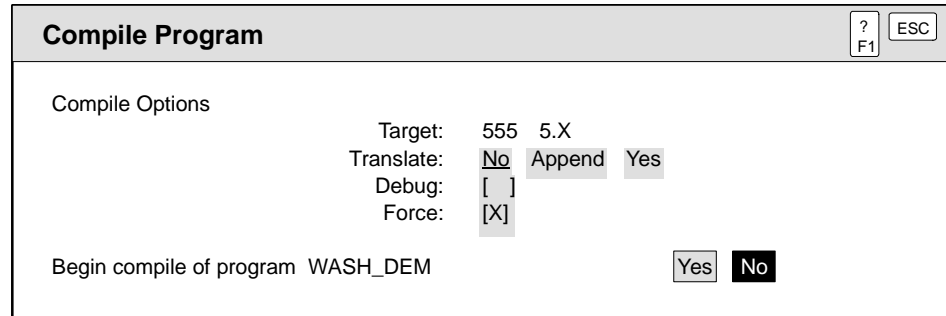
The memory size of each global memory card is specified in Kwords on the front bezel. (To convert this to Kbytes, multiply the number of Kwords by 2). For 560/565, you can add memory by installing global memory cards. With the Memory Expansion Module (PPX:545-1111) installed, the PPX:545-1101 supports a total of 448 Kbytes of user memory. For more information, refer to the programming reference manual for your controller.

10. Place an **X** between the brackets if you want to select **Force compile** and/or **Create debug version**.
11. For the **Build Translate Table** option, move the cursor to the option that you need (**No**, **Append**, or **Yes**) and press .
12. Place an **X** between the brackets if you want to select **Uninterruptible Power**.
13. Place an **X** between the brackets if you want to select **Do not convert TOD to integer fields**.
14. To include all units, place an **X** between the brackets; or, to mark individual units, press to open the window and place an **X** in front of the units that you want to include.
15. Specify any memory locations you want to reserve.
16. For 545/545L/555/575, optionally specify a U-memory file name, a user program file name, located in the `..program\program_name\pr` directory.
17. For 545/545L/555/575, you can enter the RBE memory if you need more than 20 Kbytes.
18. For 545/545L/555/575, select the scan type and enter the time slice values that you need to specify.
19. For the 575, specify the **Required Applications**, **Optional Applications**, and the **Mode-locked Applications**.
20. To return to the hierarchy, press .

Starting the Compile Operation

You can compile a program either from the Program Directory or from the Program Content. To compile an APT program, follow these steps.

1. In the Program Directory, place the cursor on the program you want to compile; or, at the Program Content Level, place the cursor on:
Control [] Configuration for compile
2. Press **[1]**, or use the **OPTs** icon to select **Compile**. The dialog window appears as shown below.



3. If the target controller is incorrect, press **[Esc]**. Then repeat, starting with Step 1.
4. If the translate, debug, or force options are incorrect, any change here is reflected in the compile and in the control file unless you press **[Esc]** before compiling.
5. When ready to compile, place the text cursor on **Yes**, and press **[Enter]**.

Compiling for Series 505 Controllers (continued)

Compile Screen

While compiling the program, APT updates the status of the compile. To stop the compile operation at any point, press **[Esc]**. APT requires a few seconds to close files. At a break point, APT acknowledges **[Esc]** by displaying **Escape**. When the object being compiled finished, APT returns from the compile.

The compile screen window appears as shown below.

```
APT PROGRAM COMPILE

Phase 1 of 6:  Coordinate Compile

Objects:    0
Errors:     0
Warnings:   0

NOTE:  Press ESCAPE to cancel
```

If you selected translating OSx tags by selecting **Append** or **Yes** for Translate Build, there will be seven phases of compile, and the seventh phase builds the translate table.

Using the Compile Report

The dialog window in [Figure 6-10](#) appears at the end of the compile operation and allows you to show or print the compilation report, or to exit to the hierarchy.

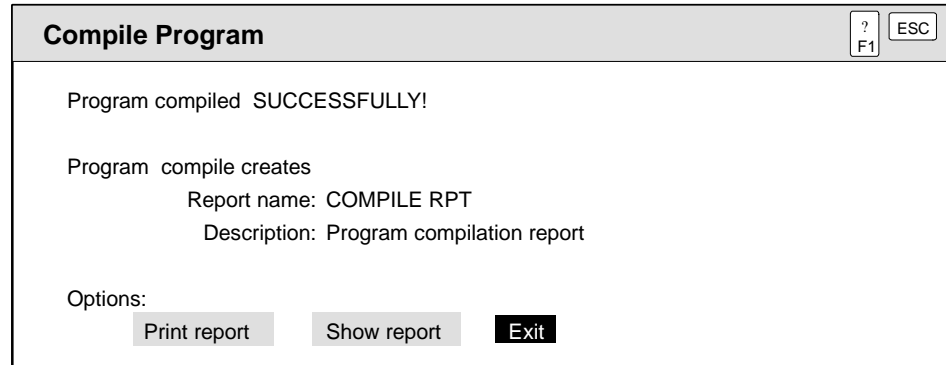


Figure 6-10 Printing or Showing the Compile Report

To show or print the report, follow these steps.

1. Use the arrow keys to move the text cursor to the appropriate option (see [Figure 6-10](#)).
2. Press **Enter**.

Refer to the appendix on error messages in the *SIMATIC APT Programming Reference (Graphics/Math) Manual* for information about errors and warnings that can appear in the compile report.

Refer to [Section 6.4](#) for a description of other reports that are available.

To return to the hierarchy, press **Esc**; or move the text-cursor to **Exit** and press **Enter**.

6.3 Compiling for S5 Controllers

This section explains how to enter data into the Compiler Control File for an S5 controller and how to run the compile process.

Understanding the Compiler Control File

Before compiling a program, you must fill out the Compiler Control File. File content depends upon the controller model, as shown in [Figure 6-11](#) and [Figure 6-12](#) on [pages 6-32](#) and [6-33](#). Field definitions are listed below. Refer to the Controller Support section of the APT Overview chapter in the *SIMATIC APT Programming Reference (Tables) Manual*.

Controller Type Enter the controller CPU type: CPU928B or CPU948. For an S5-135U, enter CPU928B; for an S5-155U or 155H, enter CPU948. If you are configuring a 155H redundant system, see the chapter on redundant configuration for S5, in the *SIMATIC APT Applications Manual*, for information about how to ensure compatibility with APT when you use COM 155H to set up your operating system parameters.

Controller Name The controller name is optional unless you are using OSx, in which case this is the name of the secondary or association name specified in the OSx Network Configuration. OSx allows these characters: letters A-Z (upper case only), numbers 0-9, the period (.), the underscore (_), and the hyphen (-). Do not use the space character in the name; APT treats it as a delimiter.

OSx/PCS Release If you are using OSx in your application, the release number must be 3.0, or greater.

Programmer Support If you have a PG programmer or STEP 5 for DOS, you can view the STL code that APT generates. If selected, APT generates code that can be displayed on the PG or in STEP 5.

Memory Size Enter the total memory available in your system. The total memory configurations for the various controllers are listed in [Table 6-1](#) on [page 6-5](#). You can also use the **P/C AUX** function in Debug or **Report Controller** in MAITT to determine your controller memory configuration.

Force Compile Mark **X** or leave blank.

An **X** indicates that you want the system to compile the entire program.

Blank indicates that you want the system to compile only those portions of the code that have changed since the last compile. In the Download Dialog Window, **Download Incremental Changes Only** becomes one of the default options.

Build translate table Select **Yes**, **No**, or **Append** to allocate OSx tags for OSx scan. Refer to [Section 8.4](#) for detailed information about these options.

If you select **No**, the existing marked tags keep the same memory locations reserved at the last Build Translate Table compile. Tags that were added or removed are not updated in the compile. Phase 7, Translate Build, does not execute.

If you select **Yes**, all marked tags are translated into the OSx format. Memory locations are blocked optimally for the most efficient access. You may have to reassign some OSx names if you have added some tags.

If you select **Append**, the tags that you have added are appended to the existing files that contain previously translated tags. Memory locations for appended information are blocked optimally. However, because they are not integrated into existing blocks, memory allocation for all tags (old and appended tags) is not blocked optimally. If you have deleted tags, the memory locations that were assigned to them are not available and cannot be used until you select **Yes** for the build translate table.

RBE Code Included Select this option if you are using OSx as your operator interface. If you select this option, APT does a forced compile.

Units to compile To include all units, place an **X** between the brackets; or, to mark individual units, press **F10** to open the window and place an **X** in front of the units that you want to include.

Reserved locations This option allows you to specify locations that you want to reserve for each memory type listed. Most APT applications do not use the Reserve Memory option, but allow APT to determine the memory allocations. Refer to the appendix on direct memory addressing in the [SIMATIC APT Programming Reference \(Graphics/Math\) Manual](#) for more information.

You can specify memory locations individually or as ranges, using a comma as a delimiter. Use a colon to specify a range. For example, to reserve data blocks 3, 4, 7, 8, 9, 10, 11, and 255, enter the following in the Data Block field:

```
3,4,7:11,255
```

It is also correct to enter the following for the same data blocks:

```
3:4,7,8,9:11,255
```

Use the same convention for all the memory types. APT does not use these memory locations.

Compiling for S5 Controllers (continued)

Operating System Defaults (928B CPU)

If you have a 928B CPU, fill out the following fields in the Compiler Control File to establish the defaults for your operating system. See [page 6-28](#) for the operating system defaults for the 948 CPU.

Restarting After Powerup Choose the restart mode after a power failure that is appropriate for your application.

WARNING

Whether the various items in controller memory are retentive or are reset after power failure, and what happens to the execution of your process, is determined by the restart mode you choose and the way you write your code. If you do not take the consequences of power failure into account when you select operating system defaults and write your code, an unexpected power failure could cause unpredictable controller operation.

Unpredictable controller operation can cause death or serious injury to personnel, and/or damage to equipment.

Pick your restart mode carefully, and know what will happen to your process in the event of a power failure. Monitor your process very carefully after a restart.

Coldstart A coldstart calls OB20, resets flags, timers, and counters to zero, and causes the controller to start at the top of the scan. When you restart the process, you must re-enable your devices, timers, and counters. The other types of memory (besides flags, timers, and counters) retain the value they had before power failure. All SFCs are reset to the first step. Since the SFC transitions are retentive, follow good programming techniques to ensure your transitions are re-evaluated prior to execution.

Warmstart A warmstart calls OB22, retains current values for flags, timers, and counters, and causes the controller to resume the scan at the point where the power failure occurred. However, your process does not pick up at the point it was before the power failure, because all SFCs are reset to the first step, and you must also re-enable your devices. Since the SFC transitions are retentive, follow good programming techniques to ensure your transitions are re-evaluated prior to execution.

Addressing Error Monitoring An X indicates that you want the CPU to monitor for I/O errors. Otherwise, the CPU ignores I/O errors when they occur.

Cycle Time Monitoring Enter the maximum cycle time in milliseconds. The default is 150 ms. If the cyclic scan takes longer than the maximum cycle time, then an error occurs.

Process Interrupt Servicing Choose Level or Edge for your process interrupt servicing.

Error Processing Checks (928B CPU)

The error processing checks described below allow you to choose how to handle program execution for a 928B CPU when an error occurs and you have not programmed an error OB. Refer to the chapter on user subroutines in the *SIMATIC APT Programming Reference (Tables) Manual* for information about how to program OBs for APT.

Runtime Error An **X** indicates that you want the controller to stop if your program attempts to jump to, or call, a non-existent block. Leave the field blank if you want the controller to continue running when this error occurs. If you have programmed OB19, OB31, and OB32, they take priority over your choice in this field. For the 928B, APT programs OB31 to catch runtime errors. They are stored in .ECODE1 and .ECODE2.

Acknowledge Error An **X** indicates that you want the controller to stop if an I/O acknowledgement delay (QVZ error) occurs. Leave the field blank if you want the controller to continue running when this error occurs. If you have programmed OB23 and OB24, they take priority over your choice in this field.

Addressing Error An **X** indicates that you want the controller to stop if an I/O error (ADF error) occurs. Leave the field blank if you want the controller to continue running when this error occurs. If you have programmed OB25, it takes priority over your choice in this field.

Cycle Error An **X** indicates that you want the controller to stop if the program overruns the specified cycle time (ZYK error). Leave the field blank if you want the controller to continue running when this error occurs. If you have programmed OB26, it takes priority over your choice in this field.

Command Code Error An **X** indicates that you want the controller to stop if a machine code error occurs. Leave the field blank if you want the controller to continue running when this error occurs. If you have programmed OB27, OB29, and OB30, they take priority over your choice in this field.

Timer Error An **X** indicates that you want the controller to stop if a timer OB is triggered to start before another timer OB has completed. Leave the field blank if you want the controller to continue running when this error occurs. If you have programmed OB33, it takes priority over your choice in this field.

Controller Error An **X** indicates that you want the controller to stop when an error occurs during the processing of standard function block (R64). Leave the field blank if you want the controller to continue running when this error occurs. If you have programmed OB34, it takes priority over your choice in this field.

Interruptibility of User Program by Interrupts The mode that you choose determines which of the interrupts in [Table 6-2](#) is allowed and when it can occur.

Table 6-2 CPU928B Program Interrupts

Mode	Interrupt Mode	Occurrence
1	All interrupts at block boundaries.	Interrupt cannot occur until a block has completed execution.
2	All interrupts at operation boundaries.	Interrupt cannot occur until an operation has completed execution.

RBE Options (928B CPU)

If you are using OSx as your operator interface application, you must place an **X** in the RBE Code Included field. The following RBE options appear. Refer to the *SIMATIC PCS 7 OSx Interface to S5 Controller User Manual* for more information about choosing options.

Number of RBE Points OSx supports 250 or 500 points on a CP143/CP1430. Enter the same value that you use to configure the CP143/CP1430.

Number of CP143 Cards in Base Unit You can install one or two CP143/CP1430 communications cards in the S5 base. Select 1 or 2.

Number of Connections on 1st CP143 Enter the number of OSx operations nodes (1-16) that are configured to communicate with the first CP143/CP1430.

Number of Connections on 2nd CP143 Enter the number of OSx operations nodes (1-16) that are configured to communicate with the second CP143/CP1430.

Read Time from CP143 If the CP143/CP1430 is synchronized with a network time (provided by either the SIMATIC NET Real Time Transmitter or a CP143/CP1430 module), then you can choose whether the time is acquired from the CP143/CP1430 to provide an RBE exception timestamp.

Option 0 — Do not read time from a CP143/CP1430.

Option 1 — Read time from first CP143/CP1430.

Option 2 — Read time from second CP143/CP1430.

SSNR for 1st CP143 The default value is 0. One CP module (not both) can communicate directly to the CPU across the backplane. If this CP module is supposed to communicate across the backplane, change its SSNR value to 232.

SSNR for 2nd CP143 The default value is 4. One CP module (not both) can communicate directly to the CPU across the backplane. If this CP module is supposed to communicate across the backplane, change its SSNR value to 232.

NOTE: You cannot assign the same value to the SSNR field of both CP143/CP1430 modules. For example, do not try to assign a value of 232 to both modules; you can only choose one module to communicate across the backplane to the CPU.

Operating System Defaults (948 CPU)

If you have a 948 CPU, fill out the following fields in the Compiler Control File to establish the defaults for your operating system. See [page 6-24](#) for the Operating System Defaults for the 928B CPU.

Restarting After Powerup Choose the restart mode after a power failure that is appropriate for your application.

WARNING

Whether the various items in controller memory are retentive or are reset after power failure, and what happens to the execution of your process, is determined by the restart mode you choose and the way you write your code. If you do not take the consequences of power failure into account when you select operating system defaults and write your code, an unexpected power failure could cause unpredictable controller operation.

Unpredictable controller operation can cause death or serious injury to personnel, and/or damage to equipment.

Pick your restart mode carefully, and know what will happen to your process in the event of a power failure. Monitor your process very carefully after a restart.

Coldstart A coldstart calls OB20, resets flags, timers, and counters to zero, and causes the controller to start at the top of the scan. When you restart the process, you must re-enable your devices, timers, and counters. The other types of memory (besides flags, timers, and counters) retain the value they had before power failure. All SFCs are reset to the first step. Since the SFC transitions are retentive, follow good programming techniques to ensure your transitions are re-evaluated prior to execution.

Warmstart A warmstart calls OB22, retains current values for flags, timers, and counters, and causes the controller to resume the scan at the point where the power failure occurred. However, your process does not pick up at the point it was before the power failure, because all SFCs are reset to the first step, and you must also re-enable your devices. Since the SFC transitions are retentive, follow good programming techniques to ensure your transitions are re-evaluated prior to execution.

Manual Restart Manual restart means that the operator must restart the controller and choose which method is appropriate. The controller remains in the stop mode until restarted manually. A manual cold restart calls OB20; a manual warm restart calls OB21.

Cycle Time Monitoring Enter the maximum cycle time. APT multiplies the value that you enter by 10 to get a time in milliseconds. The default is 150, which means a maximum scan time of 1500 ms. If the cyclic scan takes longer than the cycle time, then an error occurs.

Basic Clock Enter a multiplier for the CPU clock speed. For example, enter 10 for OB13 to interrupt every 1.0 seconds; enter 1 for OB13 to interrupt every 100 milliseconds. A value of 1 is required for RBE, loops, analog alarms, and devices to work correctly.

Time Interrupt Servicing Enter a priority for timed interrupts. For high priority, enter 1; for low priority, enter 2.

Process System Interrupt A/B Enter a priority for Interrupt A/B system interrupts. For high priority, enter 1; for low priority, enter 2.

Process System Interrupt E Enter a priority for Interrupt E system interrupts. For high priority, enter 1; for low priority, enter 2.

Process System Interrupt F Enter a priority for Interrupt F system interrupts. For high priority, enter 1; for low priority, enter 2.

Process System Interrupt G Enter a priority for Interrupt G system interrupts. For high priority, enter 1; for low priority, enter 2.

Interruptibility of User Program by Interrupts The mode that you choose determines which of the interrupts in [Table 6-3](#) are allowed and when they can occur.

Table 6-3 CPU948 Program Interrupts

Mode	Interrupt Mode	Occurrence
1	All interrupts at block boundaries.	Interrupt cannot occur until a block has completed execution.
2	All interrupts at operation boundaries.	Interrupt cannot occur until an operation has completed execution.

Compiling for S5 Controllers (continued)

RBE Options (948 CPU)

If you are using OSx as your operator interface application, you must place an **X** in the RBE Code Included field. The following RBE options appear. Refer to the *SIMATIC PCS 7 OSx Interface to S5 Controller User Manual* for more information about choosing options.

Number of RBE Points OSx supports 250 or 500 points on a CP143/CP1430. Enter the same value that you use to configure the CP143/CP1430.

Number of CP143 Cards in Base Unit You can install one or two CP143/CP1430 communications cards in the S5 base. Select 1 or 2.

CP/CPU Configuration Select the option listed in [Table 6-4](#) that is appropriate for your system.

Table 6-4 CP/CPU Configuration Options for the 948

Option	Selection	Configuration
0	S5-135U, S5-155U.	No redundancy. One CPU controls the process.
1	S5-155H Switched Network CP Configuration.	Redundant CPUs, with switched I/O, on a single network.
2	S5-155H One-Sided Network with 1st CP having CP1.	Redundant CPUs with redundant networks, with CP1 located in same base as the master CPU.
4	S5-155H One-Sided Network with 1st CP having CP2.	Redundant CPUs with redundant networks, with CP2 located in same base as the master CPU.

Number of Connections on 1st CP143 Enter the number of OSx operations nodes (1-16) that are configured to communicate with the first CP143/CP1430.

Number of Connections on 2nd CP143 Enter the number of OSx operations nodes (1-16) that are configured to communicate with the second CP143/CP1430.

Read Time from CP143 If the CP143/CP1430 is synchronized with a network time (provided by either the SIMATIC NET Real Time Transmitter or a CP143/CP1430 module), then you can choose whether the time is acquired from the CP143/CP1430 to provide an RBE exception timestamp.

Option 0 — Do not read time from a CP143/CP1430.

Option 1 — Read time from first CP143/CP1430.

Option 2 — Read time from second CP143/CP1430.

SSNR for 1st CP143 The default value is 0. One CP module (not both) can communicate directly to the CPU across the backplane. If this CP module is supposed to communicate across the backplane, change its SSNR value to 232.

SSNR for 2nd CP143 The default value is 4. One CP module (not both) can communicate directly to the CPU across the backplane. If this CP module is supposed to communicate across the backplane, change its SSNR value to 232.

NOTE: You cannot assign the same value to the SSNR field of both CP143/CP1430 modules. For example, do not try to assign a value of 232 to both modules; you can only choose one module to communicate across the backplane to the CPU.

Compiling for S5 Controllers (continued)

Figure 6-11 and Figure 6-12 show the contents of the Compiler Control File for the 928B and 948 CPUs.

Controller type:	<u>CPU928B</u> •
Controller name:	<u>CPU928B</u>
OSx release:	4.x
Programmer Support:	<u>None</u> PG
Memory size:	<u>64</u> Kbytes
Force compile?	<input checked="" type="checkbox"/> [X]
Build translate table?	<u>No</u> Append Yes
RBE code included?	<input type="checkbox"/> []

Units to compile -----

Include all? [] 0 of 0 included
(Press F10 to mark individual units)

Reserved locations -----

Data Blocks (DB): _____

Ext. Data Blocks (DX): _____

Function Blocks (FB): _____

Ext. Func. Blocks (FX): _____

Program Blocks (PB): _____

Sequence Blocks (SB): _____

Flag Bytes: _____

S-Flag Bytes: _____

Timers: _____

Operating System Defaults -----

Restart After Power Up: Warm Cold

Addressing Error Monitoring: [X]

Cycle Time Monitoring: 150 (msec)

Process Interrupt Servicing: Level Edge

System stop if event occurs and error OB does not exist:

Runtime Error (OB19,31,32): [X]

Acknowl. Error (OB23,24): []

Addressing Error (OB25): [X]

Cycle Error (OB26): [X]

Command Code Error (OB27,29,30): [X]

Timer Error (OB33): [X]

Controller Error (OB34): [X]

Interruptibility of user program by interrupts: Mode: 1 •

Report by Exception -----
{if select RBE code included}

Number of RBE Points:	<u>250</u> 500
Number of CP143 cards in base unit:	<u>1</u> 2
Number of connections on 1st CP143:	<u>1</u>
Number of connections on 2nd CP143:	<u>0</u>
Read time from CP143:	<u>0</u> •
SSNR for 1st CP143:	<u>0</u>
SSNR for 2nd CP143:	<u>4</u>

----- End of Form -----

Figure 6-11 Compiler Control File — CPU928B


```

Controller type: CPU948 •
Controller name: CPU948
OSx release: 4.x
Programmer Support: None PG
Memory size: 640 Kbytes
Force compile? [X]
Build translate table? No Append Yes
RBE code included? [ ]

Units to compile -----
Include all? [ ] 0 of 0 included
(Press F10 to mark individual units)

Reserved locations -----
Data Blocks (DB): _____
Ext. Data Blocks (DX): _____
Function Blocks (FB): _____
Ext. Func. Blocks (FX): _____
Program Blocks (PB): _____
Sequence Blocks (SB): _____
Flag Bytes: _____
S-Flag Bytes: _____
Timers: _____

Operating System Defaults -----
Restart After Power Up: Warm Cold Manual

Cycle Time Monitoring: 150 (X 10 msec)
Basic Clock: 1 (X 10 msec)

Time Interrupt Servicing: 1 priority
Process System Interrupt A/B: 0 priority
Process System Interrupt E: 0 priority
Process System Interrupt F: 0 priority
Process System Interrupt G: 0 priority
Interruptibility of user program by interrupts: Mode: 1 •

Report by Exception -----
{if select RBE code included}
Number of RBE Points: 250 500
Number of CP143 cards in base unit: 1 2
CP/CPU Configuration: 0 •
Number of connections on 1st CP143: 1
Number of connections on 2nd CP143: 0
Read time from CP143: 0 •
SSNR for 1st CP143: 0
SSNR for 2nd CP143: 4
----- End of Form -----

```

Figure 6-12 Compiler Control File — CPU948

Compiling for S5 Controllers (continued)

Editing the Compiler Control File (928B and 948 CPUs)

To configure the control file, follow the steps below. As you edit the control file, you can press to cancel all changes before you exit the form.

1. Position the text cursor on the following line in the Compile section of the Program Content Directory.
CONTROL [] Configuration for compile
2. Press or or .
3. Enter the controller type. The rest of the control file is displayed after you press .
4. Enter a meaningful name for the controller, e.g., ALUM_LINE. If you intend to translate the program or communicate with OSx, specify the controller name used in OSx.
5. If you have a PG programmer or STEP 5 for DOS, you can view the STL code that APT generates.
6. Specify in Kbytes the total memory size available in your system. The defaults are listed below:
 - 64 Kbytes for the 928B.
 - 640 Kbytes for the 948.

-
7. Place an **X** between the brackets if you want to select **Force compile**.
 8. For the **Build Translate Table** option, move the cursor to the option that you need (**No**, **Append**, or **Yes**) and press **[Enter]**. APT underlines the option.
 9. If you are using OSx in your application, place an **X** between the brackets to select **RBE Code Included**.
 10. To include all units, place an **X** between the brackets; or, to mark individual units, press **[F10]** to open the window and place an **X** in front of the units that you want to include.
 11. Specify any memory locations you want to reserve.
 12. Select the desired **Restart After Power Up** option.

Compiling for S5 Controllers (continued)

Completing the Compiler Control File (928B CPU)

For a 928B CPU, follow the steps below to complete the Compiler Control File. For a 948 CPU, refer to [page 6-37](#).

1. Place an **X** between the brackets if you want the CPU to monitor for I/O errors.
2. Enter the CPU cycle time.
3. Select **Level** or **Edge** for Process Interrupt Servicing.
4. Place an **X** between the brackets of each option, **Runtime Error**, **Acknowledge Error**, **Addressing Error**, etc., if you want to halt program execution when that error occurs.
5. Select the interrupt mode.

If you mark the **RBE Code Included** field, make selections for the options in steps 6 through 12.

6. Select the number of RBE points.
7. Select the number of CP143/CP1430 cards.
8. Enter the number of OSx nodes configured to communicate with the first CP143/CP1430.
9. Enter the number of OSx nodes configured to communicate with the second CP143/CP1430.
10. Select the time update options appropriate for your application.
11. Specify the SSNR (or accept the default value) for the first CP143/CP1430.
12. Specify the SSNR (or accept the default value) for the second CP143/CP1430.
13. To return to the hierarchy, press .

Completing the Compiler Control File (948 CPU)

For a 948 CPU, follow the steps below to complete the Compiler Control File. For a 928B CPU, refer to [page 6-36](#).

1. Enter a time for cycle time monitoring.
2. Enter a multiplier for the CPU clock speed.
3. Enter a priority (1 or 2) for timed interrupt servicing.
4. Enter a priority (1 or 2) for system interrupts A/B, E, F, and G.
5. Select the interrupt mode.

If you mark the **RBE Code Included** field, make selections for the options in steps 6 through 13.

6. Select the number of RBE points.
7. Select the number of CP143/CP1430 cards.
8. Select the option for **CP/CPU Configuration** that is appropriate for your system, according to [Table 6-4](#).
9. Enter the number of OSx nodes configured to communicate with the first CP143/CP1430.
10. Enter the number of OSx nodes configured to communicate with the second CP143/CP1430.
11. Select the time update options appropriate for your application.
12. Specify the SSNR (or accept the default value) for the first CP143/CP1430.
13. Specify the SSNR (or accept the default value) for the second CP143/CP1430.
14. To return to the hierarchy, press .

Compiling for S5 Controllers (continued)

Starting the Compile Operation

You can compile a program either from the Program Directory or from the Program Content. To compile an APT program, follow these steps.

1. In the Program Directory, place the cursor on the program you want to compile; or, at the Program Content Level, place the cursor on:
Control [] Configuration for compile
2. Press 1, or use the **OPTs** icon to select **Compile**. The dialog window appears as shown in [Figure 6-13](#).

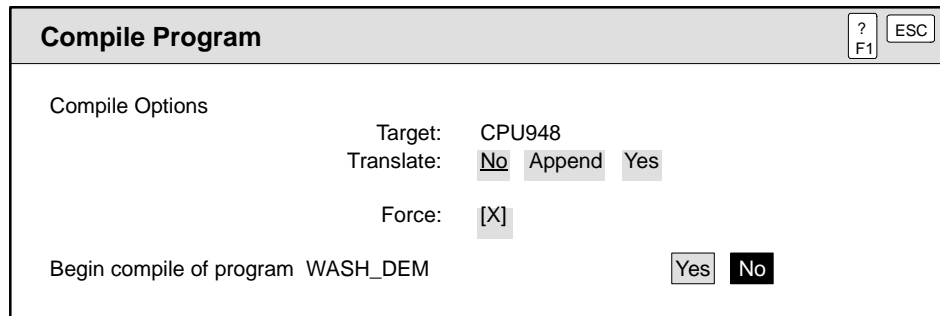


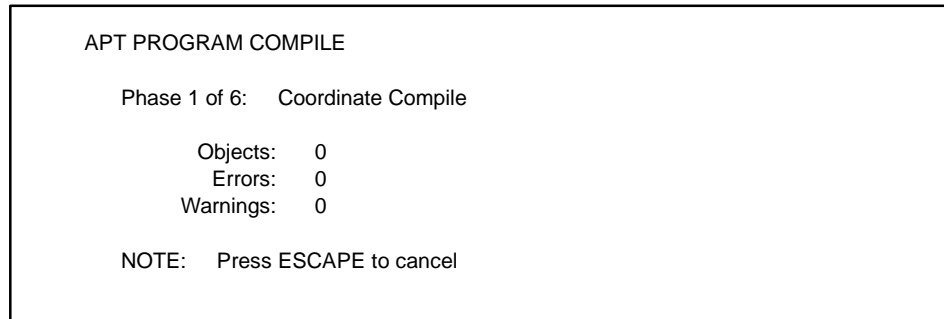
Figure 6-13 Compiling an APT Program

3. If the target controller is incorrect, press Esc.
4. If the translate or force options are incorrect, any change here is reflected in the compile and in the control file unless you press Esc before compiling.
5. When ready to compile, place the text cursor on **Yes**, and press Enter.

Using the Compile Screen

While compiling the program, APT updates the status of the compile. To stop the compile operation at any point, press **[Esc]**. APT requires a few seconds to close files. At a break point, APT acknowledges **[Esc]** by displaying **Escape**. When the object being compiled finished, APT returns from the compile.

The compile screen window appears as shown below.



If you selected translating OSx tags by selecting **Append** or **Yes** for Translate Build, there will be seven phases of compile, and the seventh phase builds the translate table.

Using the Compile Report

The dialog window in [Figure 6-14](#) appears at the end of the compile operation and allows you to show or print the compilation report, or to exit to the hierarchy.

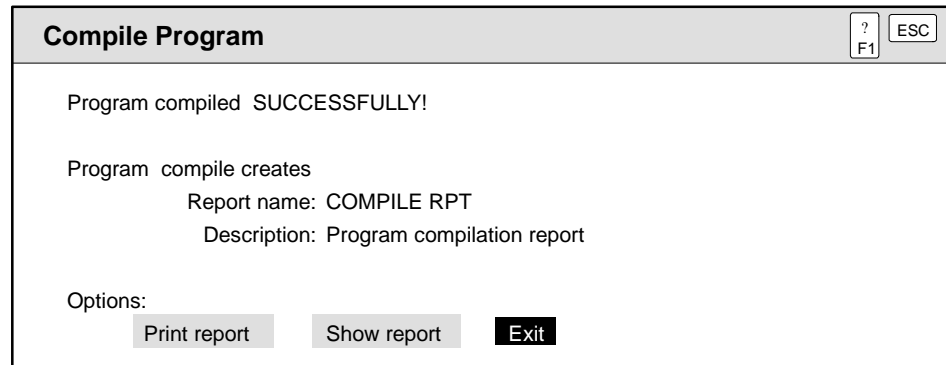


Figure 6-14 Printing or Showing the Compile Report

Compiling for S5 Controllers (continued)

To show or print the report, follow these steps.

1. Use the arrow keys to move the text cursor to the appropriate option (see [Figure 6-14](#)).
2. Press `Enter`.

Refer to the appendix on error messages in the *SIMATIC APT Programming Reference (Graphics/Math) Manual* for information about errors and warnings that can appear in the compile report.

Refer to [Section 6.4](#) for a description of other reports that are available.

To return to the hierarchy, press `Esc`; or move the text-cursor to **Exit** and press `Enter`.

6.4 Object File Reports

Overview

When you have a successfully compiled program, you can create reports from the Object File. APT provides the following reports.

- APT symbol name to controller address: **symbol.rpt** lists the APT symbolic names in alphabetical order with the corresponding controller address for each.
- Controller address to APT symbol name: **address.rpt** lists all types of controller memory, e.g., C, K, V, S, etc., for Series 505, and FW, DB:D, PW, etc., for S5, with the corresponding symbolic name that is assigned to each address.
- SFC/CFC math to controller address: **cfb.rpt** lists SFC math blocks and CFBs with the corresponding addresses and any related SFPGMs.
- SFC cross-reference: **sfc.rpt** lists the main SFC with the related safe-state and subordinate SFCs.
- SFC steps and transitions: **steps.rpt** lists each step and transition with the corresponding control relay.
- Translated tags: **tag.rpt** lists the OSx tag types and information about the attributes (address and whether or not they are scanned).
- Cross reference: **xref.rpt** is a list of cross references for all objects defined in the program.
- TISOFT comment files: (For Series 505 only) This report contains the files used by TISOFT to put comments in the ladder logic. This report can only be saved to a file for use by TISOFT and cannot be printed or viewed from APT.
- TISOFT synonym files: (For Series 505 only) This report contains the files used by TISOFT to put synonyms in the ladder logic. This report can only be saved to a file for use by TISOFT and cannot be printed or viewed from APT.
- STEP 5 symbol files: (For S5 only) This report contains the files used by STEP 5 to put synonyms in STL. This report can only be saved to a file for use by STEP 5 and cannot be printed or viewed from APT.

The STEP 5 symbol file is stored in the **apt\program\program_name\pr** subdirectory. The file name is the first six characters of the APT program name, followed by **z0.seq**. For instance, if your APT program is TEST1, then the name of the symbol file is **test1@z0.seq**. To use this symbol file in STEP 5, copy it from APT to the STEP 5 working directory, and then invoke it from the assignment list. You need to set your symbol length to 24 and your comment length to 30 for compatibility with APT.

Object File Reports (continued)

Using Object Reports

To access any of the object reports, follow these steps. In the Program Content Directory, place the cursor on:

OBJECT [] Successfully compiled program

1. Press , or use the **OPTs** icon to select **Report**. The dialog window appears as shown in [Figure 6-15](#) for a Series 505 controller, or [Figure 6-16](#) for an S5 controller.
2. Place an **X** between the brackets following the appropriate selection(s), to either save or to print the reports.
3. After selecting reports, move the cursor to **OKAY** and press .
4. Return to hierarchy and read the reports in Saved Reports.

REPORTS			<input type="button" value="F1"/>	<input type="button" value="ESC"/>
	Save	Print		
APT symbol name to P/C address	[]	[]		
P/C address to APT symbol name	[]	[]		
SFC/CFC math to P/C address	[]	[]		
SFC cross-reference	[]	[]		
SFC steps and transitions	[]	[]		
Translated tags	[]	[]		
Cross Reference	[]	[]		
TISOFT comment files	[]			
TISOFT synonym files	[]			
				<input type="button" value="OKAY"/>

Figure 6-15 Using Object Reports (Series 505)

Figure 6-16 shows the dialog window that appears if you have an S5 controller when you choose to view an object report.

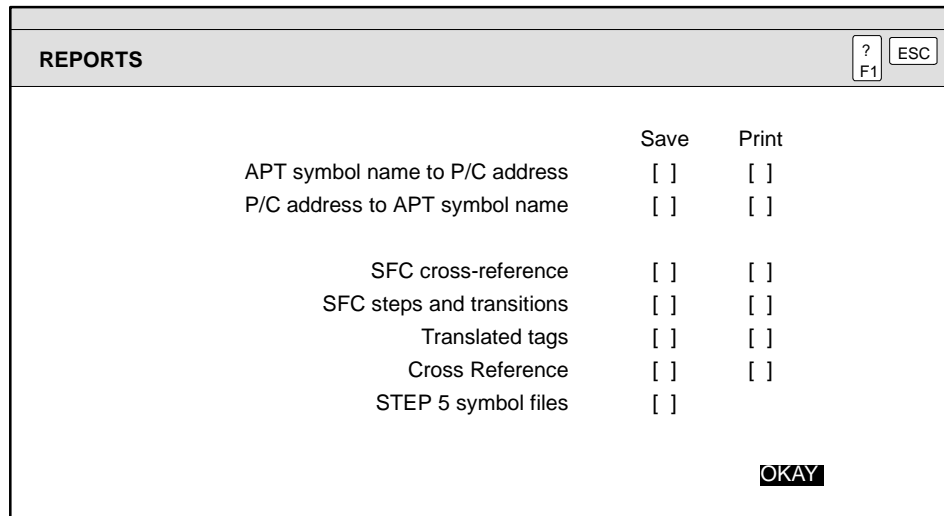


Figure 6-16 Using Object Reports (S5)

Deleting Object Code

APT allows you to delete the object code created by the Compile option. Although the program can no longer be executed by the controller when the object code has been deleted, you reduce the size of your program files.

To delete the object code follow these steps.

1. In the Program Content Directory, place the cursor on:
OBJECT [] Successfully compiled program
2. Press **[D]**, or use the **OPTs** icon to select **Delete**. A dialog window appears.
3. Press **[Enter]** to continue, or **[Esc]** to cancel the operation.
4. Select **Yes** to confirm the deletion of the object code, or **No** to cancel.

6.5 Downloading Program to Controller

When you have a successfully compiled program, you can download the program to your Series 505 or S5 controller. You can download from either the Program Directory or the Program Content Level. Be sure that you download your program to the correct controller. A program created for one controller cannot be downloaded to another.

WARNING

Downloading an APT program involves the mode of the controller and the execution of equipment.

Downloading a program could cause unexpected application operation that could result in death or serious injury to personnel, and/or damage to equipment.

Ensure that you are fully familiar with all the issues involving the downloading of an APT program before you download to your controller.

The download dialog window offers different options that depend upon the APT_MODE that you selected during APT installation.

If you have a Series 505 controller, when you are using TISTAR, you have the option of downloading the program through the TISTAR operator station, or directly to the controller. When you are using both the CP1413 or CP1613 card and the SIMATIC NET Technological Functions (TF) with your APT computer, you can download your programs across the Industrial Ethernet network. When your APT computer is connected to the controller using the CVU TIWAY Adapter or the Unilink Host Adapter, you have the option of downloading the program across TIWAY or directly to the controller.

S5 controllers can communicate with APT by means of the Industrial Ethernet (H1) network, or through a direct connection.

All APT_MODEs allow the following options. (See [Section B.4](#) for a description of APT_MODE.)

- Clear the controller memory. For Series 505, this option clears all controller memory, including reserved memory locations, unforces all forced variables, and clears all I/O points. For S5 controllers, this option performs an overall RESET in the controller; it clears all controller memory, including reserved memory locations, and clears all I/O points. All reserved FB/FXs, PBs, SBs, and DB/DXs must be downloaded to the controller after clearing memory.
- Download the I/O configuration (Series 505 only). When you use TISOFT to customize the I/O configuration in the controller, select **No** for this option to avoid overwriting the I/O and PROFIBUS configurations.

-
- Do an incremental download. Use this option to reduce the “bump” to your process control. That is, objects that are not downloaded retain their current states (loops remain in their current mode, the same steps remain active in SFCs, values do not change, etc.) It reduces the download time since only those objects that you have changed, added, or deleted from the program are downloaded. Objects that you can download incrementally include the CFB, SFC, device, declaration, recipe, subroutine, I/O point, and, for Series 505 only, I/O base configuration. Depending on what the program changes are, you may be able to download the program while the controller is still in the RUN mode.

For Series 505, if you choose to do an incremental download in the PROGRAM mode, the controller goes into a Program Freeze condition until you change the mode back to RUN. For S5, if you choose to do an incremental download in the PROGRAM mode, the controller goes into STOP mode until you change the mode back to RUN with either a COLDSTART or a WARMSTART.

When to Use the Incremental Download

Before you compile the program, you have the option of forcing a complete compile by placing an **X** in the **Force Compile** field of the Compile Control File. Leaving this field blank indicates that you want the system to compile only those portions of the code that have changed, including added/deleted objects, since the last compile.

If APT has compiled only a portion of the code, you may then be able to do an incremental download. In the Download Dialog Window, the option **Download Incremental Changes Only** becomes one of the default options if you do not select a forced compile on the Compiler Control File.

APT cannot report whether an incremental download can be done, or whether the controller must be in PROGRAM mode, until you actually start a program download. After you select the **Download** option, APT checks the controller to see which of these download options are possible:

- Incremental download with controller in RUN mode
- Incremental download with controller in PROGRAM mode
- Full program download with controller in PROGRAM mode

Downloading Program to Controller (continued)

Quick Verification Option

If you need to know which objects will be downloaded before you do an incremental download, select the **Quick Verification** option (not currently available for S5 controllers), described in [Section 6.6](#). The **Quick Verification** option checks for changes in the date-time stamps, and if the program in the controller is older, generates a list of those objects that must be downloaded.

Obstacles to Incremental Download

Your program code must satisfy several conditions before you can do an incremental download. These are described on [pages 6-46 to 6-49](#).

You cannot do an incremental program download if any one of the following conditions is true.

- You select the **Force Compile** option in the Compiler Control File.
- APT determines that a force compile is required because you made changes in the Compiler Control File, e.g., you have changed reserved memory, controller, etc.
- You have restored a program that was created under a previous release of APT and have not compiled it again under the current release.
- You delete the object code for the program.
- You delete a unit or unmark a unit that had been marked for compile.
- You mark new tags and select **YES** for the **Build Translate Table** option in the Compiler Control File.
- You have selected the **YES** option for the **Build Translate Table** option in the Compiler Control File since doing the last program compile.
- You have not done a successful compile of the program.
- You use TISOFT, STEP 5, or another programming package to make changes to the RLL or STL code that was generated by an APT compile.
- The program to be downloaded is not a modification of the program that is currently loaded in the controller. That is, code version information, such as the program time stamp, has changed.
- For an S5 controller, you add or delete the RBE code for OSx tag translation. RBE code is added when you select **RBE code included** on the Compiler Control file.

Overflow CRs and Incremental Download

For Series 505, if you make assignments to I/O points that APT used for control relays (CRs) in the last compile, you cannot do an incremental download. The compile report lists these points as Ys used as CRs (overflow CRs).

If the compile report lists no overflow CRs, you can make any new I/O assignments and still do an incremental download.

If the compile report lists overflow CRs, you may still be able to do an incremental download. APT begins assigning I/O points to CRs after the highest address that you use for I/O. If possible, assign any new points to addresses below your highest assigned address. See the example in [Figure 6-17](#).

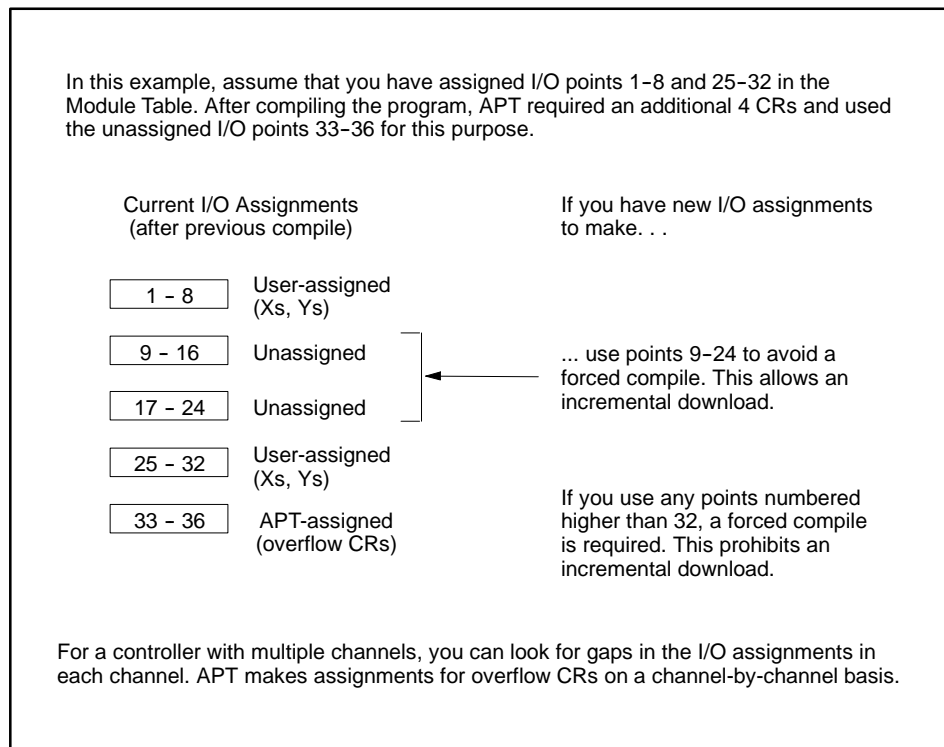


Figure 6-17 Download Requirements and I/O Assignments for Series 505

Incremental Download for S5 Redundant System

For an S5 redundant system, you can do an incremental program download, but you must also communicate changes, such as new I/O modules and new data blocks, in COM 155H. Implementation with COM 155H is discussed in the chapter on redundant configuration for S5, in the [SIMATIC APT Applications Manual](#).

Downloading Program to Controller (continued)

Incremental Download and RUN Mode

You cannot do an incremental program download while the controller is in the RUN mode if any one of the following conditions is true:

- Incremental download is not permitted, as described on [page 6-46](#).
- You resequenced steps (selected the **Sequence Labels** option) while you were editing an SFC.
- The 575 Application Dependencies or time slot options are modified.
- APT automatic allocation of controller memory changes. This generally happens when you make substantive changes to your program.

When you compile a program for Series 505, APT automatically allocates controller memory, with the ratio of memory types based on program requirements: xx Kbytes for L-Memory, xx Kbytes for V-Memory, etc. This is comparable to the memory allocation that is done manually on the Memory Configuration screen in TISOFT.

The size of each memory allocation is generally larger than that actually required by the program. Therefore, you can make changes to the program that fit within the previously allocated blocks. If changes do not fit, the memory is reallocated and it will be necessary to do a download with the controller in Program or Stop mode.

You can examine the compile report (**compile.rpt**) to see how close your program is to the ratio limits for each memory type. Refer to the example shown in [Figure 6-18](#). For S5 controllers, it is not necessary to allocate memory. Therefore, S5 controllers do not have this limitation.

A summary of the compiled program memory requirements is listed at the end of the compile report. In this example, controller memory was allocated in the following manner:

	Actual Usage	Memory Used/ Memory Configured
Ladder (L)	11695 words	68/162 Kbytes
Variable (V)	792 words	1/5 Kbytes
Constant (K)	52 words	0/1 Kbytes
Special (S)	1065 words	2/12 Kbytes
TMR/CTR/DCAT/MCAT	53 words (21 boxes)	0/5 Kbytes
Drum/Edrum	0 words (0 boxes)	0/3 Kbytes
Shift Register	0 words (0 boxes)	0/1 Kbytes
Table Move	2 words (2 boxes)	0/2 Kbytes
One Shot	33 words (66 boxes)	0/1 Kbytes
Nonretentive CRs	1189	
Retentive CRs	28	
Ys used as CRs	0	

To determine how much of the configured L-Memory is required by your program, solve the following equation.

$$\text{Program L Memory} = \frac{\left(11695 \text{ words L Memory} \times \frac{3 \text{ words overhead}}{\text{word L Memory}} \times \frac{2 \text{ Kbytes}}{\text{word}} \right)}{\frac{1024 \text{ bytes}}{\text{Kbytes}}} = 68.53 \text{ Kbytes}$$

Since APT allocated 162 Kbytes for L-Memory, you can add RLL code without risking a reallocation of controller memory. Note that L-Memory is the only memory type with 3:1 overhead. For other memory types, the second term in the equation is 1.

Figure 6-18 Example Program Memory Requirements for Series 505

WARNING

Downloading an APT program to the controller while the controller is in the RUN mode is a complicated series of events.

Downloading a program to a controller while the controller is in the RUN mode could cause unexpected application operation that could result in death or serious injury to personnel, and/or damage to equipment.

To avoid unexpected controller operation before you start a download, be aware of the following:

When you download a program while the controller is in RUN mode, the new program takes immediate control of the process. Check the program for logic errors before executing the download to a running process.

Do not attempt an APT program download if communications between your computer and the controller are unstable. If this communications link is broken during download, you cannot continue, and the controller may contain an invalid program. You must initiate a new, full program download while the controller is in PROGRAM mode.

A program download can cause an active SFC to become inactive if you have deleted its active steps. Therefore, the deletion of steps or an entire SFC can leave the unit or global sequential charts without any active steps.

If you have changed any initial values, the new values are downloaded to the controller.

For a Series 505 controller, APT sets the controller to EDIT mode for a program download and then to RUN mode after the download is complete. The controller must recompile the program code before it reenters the RUN mode. The program is not scanned during the transition from EDIT to RUN.

For an S5 controller, APT attempts to keep the controller in RUN mode if possible.

For 545 and 575 controllers, for which you do not do a forced compile and you add, move, or delete an I/O module in the Module Table Editor:

All input points and words on the I/O base that are downloaded are set to zero for one RLL scan after an incremental download.

Downloading Program to Controller (continued)

Downloading a Program from APT

To download a program, follow these steps.

WARNING

For Series 505, APT and other external programs allow you to force elements within the controller. APT does not attempt to clear these forced elements during download unless you first select the **CLEAR P/C BEFORE DOWNLOAD** option.

Downloading a new APT program with forced elements in the controller could result in unexpected operation, including death or serious injury to personnel, and/or equipment damage.

Do not download with forced elements in the controller unless you fully understand all issues involved.

WARNING

APT allows a download to be aborted at any time by the user. If you abort a download, the controller may contain a partial or invalid program.

Execution of an invalid program could result in unexpected operation which could result in death or serious injury to personnel, and/or damage to equipment.

Be sure to observe the guidelines under the **System Commissioning** section of the **Safety Considerations** document included with your documentation.

1. In the Program Directory place the cursor on the program you want to download; or, in the Program Content, place the cursor on either of the following lines:

CONTROL [] **Configuration for compile**

or

OBJECT [] **Successfully compiled program**

2. Press 4 or use the **OPTs** icon to select **Download**. The dialog window appears as shown in [Figure 6-19](#).

-
3. Select the options appropriate for your system by placing an **X** on the option. A blank field means that the option is not selected. If you have selected an object that you do not want, then press the space bar to leave the field blank.

NOTE: When you download an APT program, the **Download I/O Configuration** option in the Download dialog box applies to PROFIBUS-DP as well.

The following communication paths are available:

- Direct connection to the controller: Download the program directly to the controller (Series 505 and S5 controllers).
- CVU TIWAY Adapter/Unilink Host Adapter: Download the program across TIWAY or directly to the controller (Series 505 controllers only).
- TISTAR: Download the program through the TISTAR operator station or directly to the controller. TISTAR must be in the OPER state before you start the download (Series 505 controllers only).
- OSx: Download the program across the Industrial Ethernet network and tell OSx that you are downloading (Series 505 and S5 controllers).
- H1: Download the program across the Industrial Ethernet network (Series 505 and S5 controllers).

NOTE: When you download an APT program using either an OSx/PCS or TISTAR operator interface, OSx or TISTAR automatically takes the associated controller offscan. After the download is complete, the scan is turned on again. The system does not turn off the scan if the program is downloaded by another method. Turn the scan off and on manually from the Network Utilities menu for TISTAR 1.x and 2.x, or from the Network Setup Menu for OSx/PCS.

4. Select **OKAY** to begin the download.
5. If the controller is in RUN mode and you need to set the mode to PROGRAM, select **Yes** to change the mode.

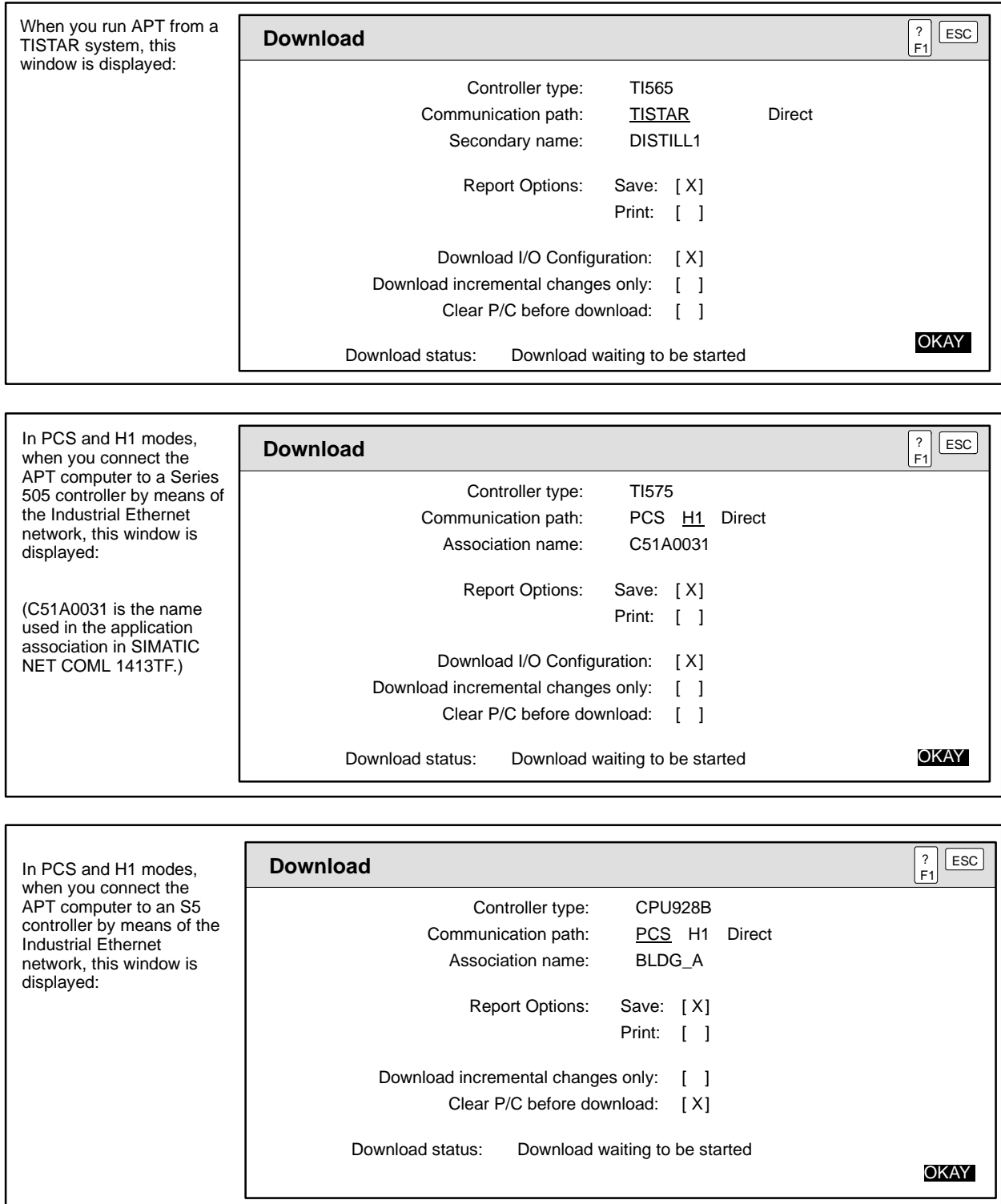


Figure 6-19 Download Dialog Windows (continued)

Downloading Program to Controller (continued)

Downloading a Program from DOS (Series 505)

The **down.exe** utility enables you to download a program to a Series 505 controller from a DOS computer without running APT. This utility is located in the `\apt` subdirectory, unless you specified another subdirectory during APT installation.

The syntax for **down.exe** is the following:

```
DOWN [-cinprt] <program_name> [<secondary_name>]
```

Items in brackets [] are optional. For `program_name`, substitute the name of the APT program. For `secondary_name`, substitute the name of the OSx/TISTAR secondary (controller). The parameters provide the following services.

- c Clears the controller memory before downloading the program.
- i Downloads only those portions of the code that have changed since the last compile was done.
- n Causes the I/O configuration not to be downloaded with the program.
- p Sets the controller to PROGRAM mode before downloading the program.
- r Sets the controller to RUN mode after downloading the program.
- t Indicates that you want to notify OSx that a download is about to occur and to take the controller off scan.

For example, when you type at the DOS prompt: `DOWN -cpr Wash_dem` these actions occur:

- The controller memory is cleared.
- The controller is set to PROGRAM mode.
- The program called WASH_DEM is downloaded to the controller.
- The controller is set to RUN mode.

If an error occurs, refer to the error messages appendix in the [SIMATIC APT Programming Reference \(Graphics/Math\) Manual](#) for more information.

NOTE: If you download an APT program when you are using either a TISTAR (Series 505 only) or an OSx/PCS (Series 505 and S5) operator interface, the system automatically turns off the scan to the associated controller. After the download is complete, the scan is turned on again. The system does not turn off the scan if the program is downloaded by another method. Turn the scan off and on manually from the Network Utilities menu for TISTAR 1.x and 2.x, or from the Network Setup Menu for OSx/PCS.

Downloading Program to Controller (continued)

Downloading a Program from DOS (S5)

The **DOWN_MC5.EXE** utility enables you to download a program to an S5 controller from a DOS computer without running APT. This utility is located in the **VPT** subdirectory, unless you specified another subdirectory during APT installation. The syntax for **down_mc5.exe** is the following:

```
DOWN_MC5 [-cioprtw] <program_name> [<secondary_name>]
```

Items in brackets [] are optional. For `program_name`, substitute the name of the APT program. For `secondary_name`, substitute the name of the OSx controller. The controller name is required if you plan to download over the Industrial Ethernet. The parameters provide the following services.

- c Clears the controller memory before downloading the program. (Performs an overall reset.)
- i Downloads only those portions of the code that have changed since the last compile was done.
- o Sets the controller to RUN mode using a coldstart after downloading the program.
- p Sets the controller to PROGRAM mode before downloading the program.
- r Sets the controller to RUN mode using a coldstart after downloading the program.
- t Indicates that you want to notify OSx that a download is about to occur and to take the controller off scan.
- w Sets the controller to RUN mode using a warmstart after downloading the program.

6.6 Verifying a Downloaded Program

Understanding Verify

If you want to compare the program in the controller with the compiled object file, you use the verify option from the Program Directory or from the Program Content Level. The verify option is not available for S5 controllers.

APT provides three options for the verification process.

- **Full Verification** provides an exact comparison of the compiled object file with the entire program in the controller and reports any differences.

Typically, this option is used to verify that a complete download has occurred. After the controller has been placed in run mode, initial values can change and would, therefore, be reported as discrepancies.

- **Partial Verification** checks the portions of the program code that should not change during normal operation. Portions that should not change include L-Memory, K-Memory, and S-Memory (except for some loop and analog alarm variables). Portions of code that do typically change during normal operation of the controller include values for V-Memory, CRs, I/O locations, and some loop and analog alarm variables.
- **Quick Verification** checks for changes in the date-time stamps, and if program in the controller is older, generates a list of those objects that must be downloaded if you want to do an incremental compile.

If you need to know which objects will be downloaded before you do an incremental download, select the **Quick Verification** option.

During the verification process, APT indicates the percent of the verification that has been completed and the number of discrepancies that have been found.

The actual discrepancies are listed in the **verify.rpt** file, which includes controller addresses and symbolic names. The **P/C address to APT symbol name** report, which is created from the object file, also lists the addresses and names.

Verifying a Downloaded Program (continued)

Using Verify

To verify that the program in the controller is the same as the compiled APT program, follow these steps.

1. In the Program Directory, place the text cursor on the program you want to verify; or, in the Program Content, place the text cursor on either:

CONTROL [] **Configuration for compile**
or

OBJECT [] **Successfully compiled program**

2. Press , or use the **OPTs** icon to select **Verify**. The following dialog window appears (Figure 6-20).

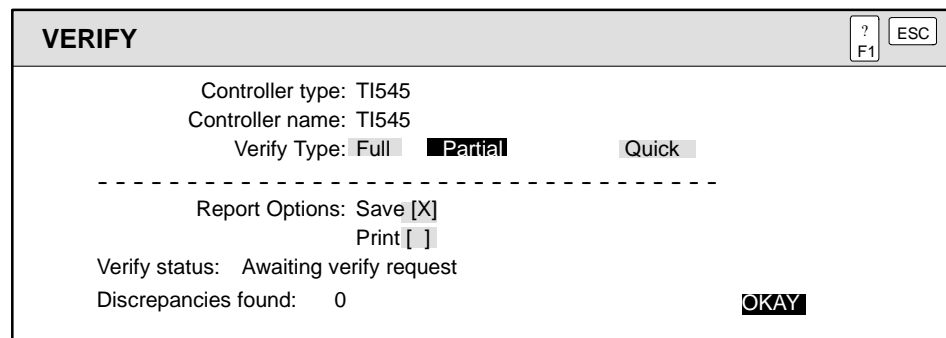


Figure 6-20 Verifying a Program

3. Select a destination (Printer and/or File) for the report of any errors in the verification.
4. Press again to confirm the action.
5. To see your Verify report, select **Show** in the Saved Reports section.

Debugging an APT Program

7.1	Understanding the Debug Operation	7-2
	Overview	7-2
	Starting Debug	7-4
	Using Debug	7-6
	Using the Series 505 AUX Functions	7-8
	Using the S5 AUX Functions	7-10
	Debugging Any Controller Program	7-12
	Using Prog/Run Option	7-14
	Using Sort Entries Option	7-17
	Using Split Screen Option	7-17
	Exiting Debug (and Forced I/O Display for Series 505)	7-17
7.2	Monitoring Variables	7-18
	Overview	7-18
	Working with Charts	7-18
	Manually Building a Chart	7-18
	Automatically Building a Chart	7-19
	Completing the Build Monitor Chart Dialog Box	7-20
	Editing a Chart	7-22
	Listing Charts	7-23
	Changing Charts	7-23
	Using the Monitor and Sample Options	7-24
	Using Trend Option	7-26
	Using Animate Option	7-28
	Using Direct Address Option	7-29
7.3	Changing I/O and Variables	7-30
	Overview	7-30
	Using Modify Option	7-31
	Using Force Option (Series 505 Only)	7-32
	Hints: Forcing Variables	7-33
7.4	Tracing SFC Program Flow	7-34
	Overview	7-34
	Using Animate Option	7-34
	Using Breakpoint Option (Series 505 Only)	7-35
	Using Single Step Option (Series 505 Only)	7-36
	Using Step Logger Option (Series 505 Only)	7-37
7.5	Using the Activate Option	7-38
	Overview	7-38
	Activating SFC Steps	7-38
	Activating CFBs	7-40
	Activating Devices	7-41

7.1 Understanding the Debug Operation

Overview

When you have successfully compiled and downloaded a program to the controller, APT allows you to test the program with the debug operation. You can debug an entire program, individual units, or individual SFCs and CFBs.

In debug, you can place the controller in either Run or Program (Stop) mode, and the options that you then have include the ability to:

- Monitor and trend variables.
- Set and test variables.
- Force variables (Series 505 only).
- Tune loops.
- Single-step through the program (Series 505 only).
- Check the logic in SFCs and CFBs.
- Set up the controller with the AUX function.
- Perform limited debugging of a program that is not in the hierarchy. The program does not have to be created with APT.

Debug displays the controller mode at the left side of the command bar instead of showing the software release version.

The debug operation provides a three-tiered hierarchy similar to the hierarchy you used in designing your program. This hierarchy contains Program Content as the first level, Unit Content as the second level, and a third content level that can be I/O, Device, Declare, Recipe, SFC content, or CFC content. The program- and unit-level Tables list the individual I/O symbolic names, Devices, Declarations, Recipes, SFCs, and CFCs defined for that program or unit.

Figure 7-1 shows an example hierarchy with the following three levels.

- The Program Content lists the program-level Tables, compiled Units, and a Chart level that contains a list of the charts to build.
- The Unit Content lists unit-level Tables, Sequential Function Charts, and Continuous Function Charts.
- The Continuous Function Chart Directory lists the individual Continuous Function Blocks in the selected CFC.

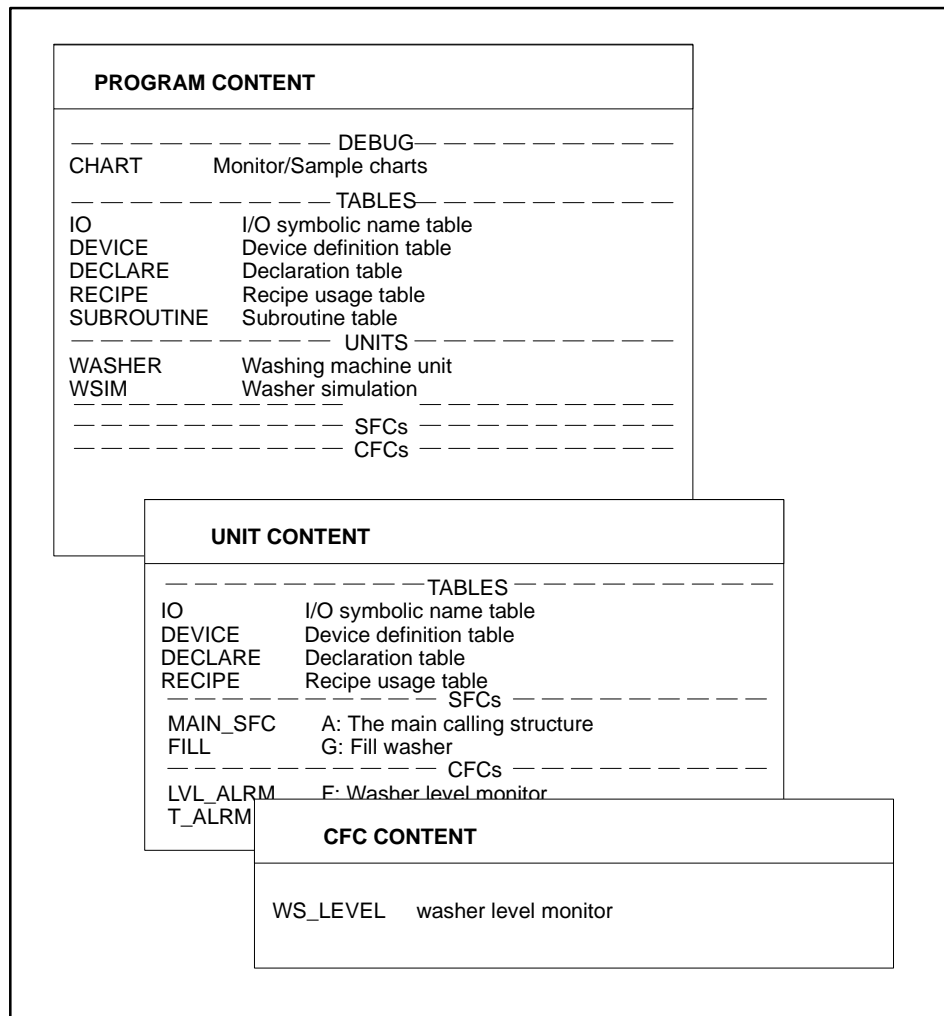


Figure 7-1 Debug Hierarchy

Understanding the Debug Operation (continued)

Starting Debug

If you only need to build charts or to copy existing charts, you can run Debug without connecting a controller to the APT computer. To run the Debug Utility, follow these steps.

NOTE: If you want to debug a program that you created under a previous release of APT and restored to a system executing the current APT release, you must recompile the program first.

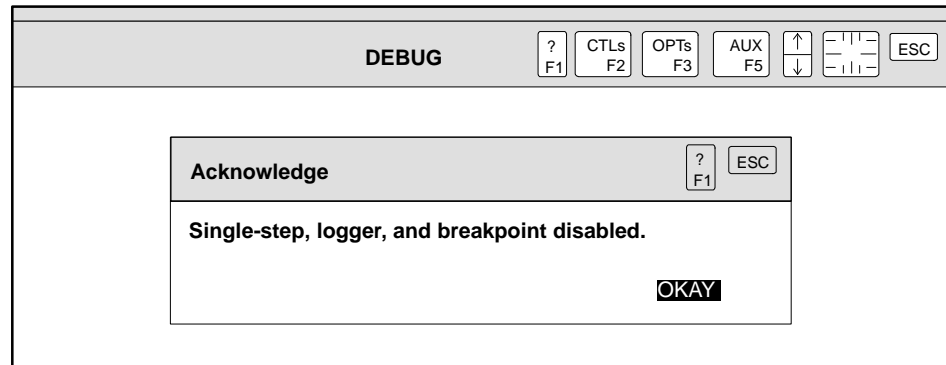
1. In the Program Directory, place the cursor on the program you want to debug; or, in the Program Content of a program, place the cursor on:

CONTROL [] Configuration for compile

or

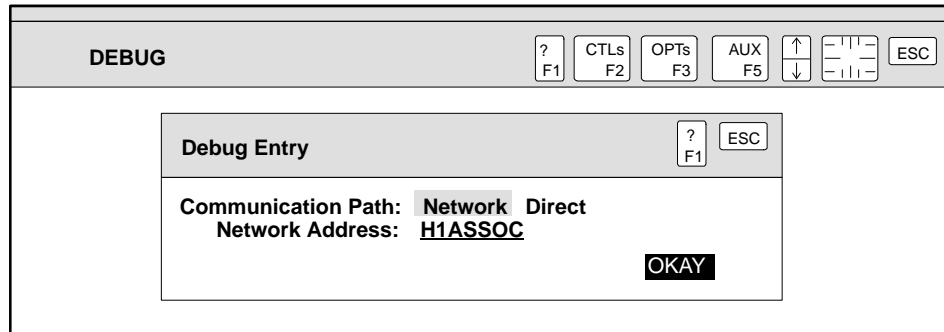
OBJECT [] Successfully compiled program

2. Press or use the **OPTs** icon to select **Debug**.
3. If you did not select **Create debug version** when you compiled the program, the following prompt appears:



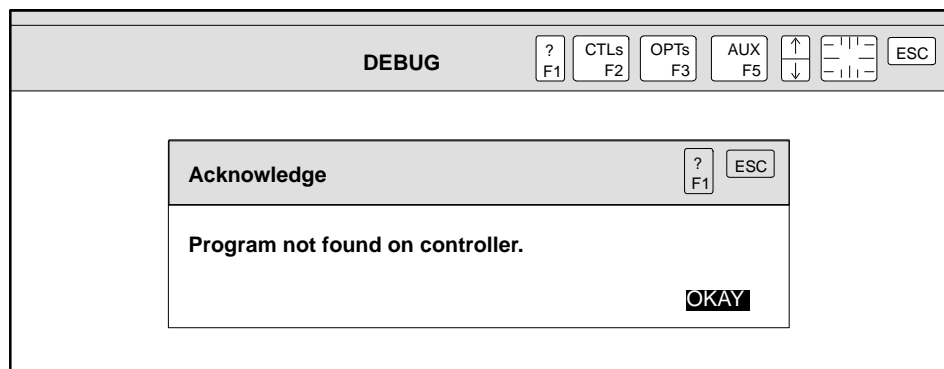
NOTE: You can use the Debug option even if you did not select **Create debug version** when you compiled the program. The Breakpoint, Single-Step, and Step Logger options, which are only supported on Series 505 controllers, are not available when the **Create debug version** option is not selected.

- If you are not using a direct connection (i.e., if APT_MODE is set to H1, PCS, TISTAR, CVU, or TIWAY for Series 505 controllers, or PCS or H1 for S5 controllers), the following prompt appears, asking you to define the communication path. The network address is the secondary name. The dialog window appears as shown below.



- If no controller is connected, APT indicates that communications could not be established. If you only want to build or copy a chart, select the **Ignore** option.
- When the program in the controller is different from the program that you are currently debugging, the dialog window appears as shown below.

This message appears on the screen if you have made any changes, and the program in the controller and the APT program have different date/time stamps. Selecting **OKAY** allows you to use some of the debug functions, but only with direct addressing. [Table 7-2](#) shows the available Debug Auxilliary (AUX) functions.



- To select an item from any directory in the debug hierarchy for debugging, place the text cursor on the name of the object that you want to debug, and press **F10** or **Enter**.

Understanding the Debug Operation (continued)

You can begin to debug at the program level, by selecting a table or an SFC or a CFC at the program level, or you can select a single unit to debug. If you select a unit, the next level of the debug hierarchy (the Unit Content directory), appears on the screen.

At the Unit Content Level, you can debug at the unit level, or you can select a table, or an SFC or a CFC within a unit, to debug. If you select a CFC, a third level (the CFC Content directory), appears so that you can select a single CFB.

Debug operates on the object code created from the last compile operation that you have done. If you have made changes in your program source code, you cannot debug these changes until you execute another compile.

If you have changed a unit since the last compile operation, APT reminds you of the change before you continue. However, until you compile and download, your changes to the unit are not visible in Debug. For example, if you have deleted 10 of 30 declarations, Debug still lists all 30 declarations until you recompile.

NOTE: If you have changed an SFC since the last compile, you cannot use Debug to debug this SFC. Debug removes the SFC from the list of SFCs.

Using Debug

During the debug operation, the **CTLs** and **OPTs** icons provide the actions that are shown in [Figure 7-2](#) and are explained in [Table 7-1](#).

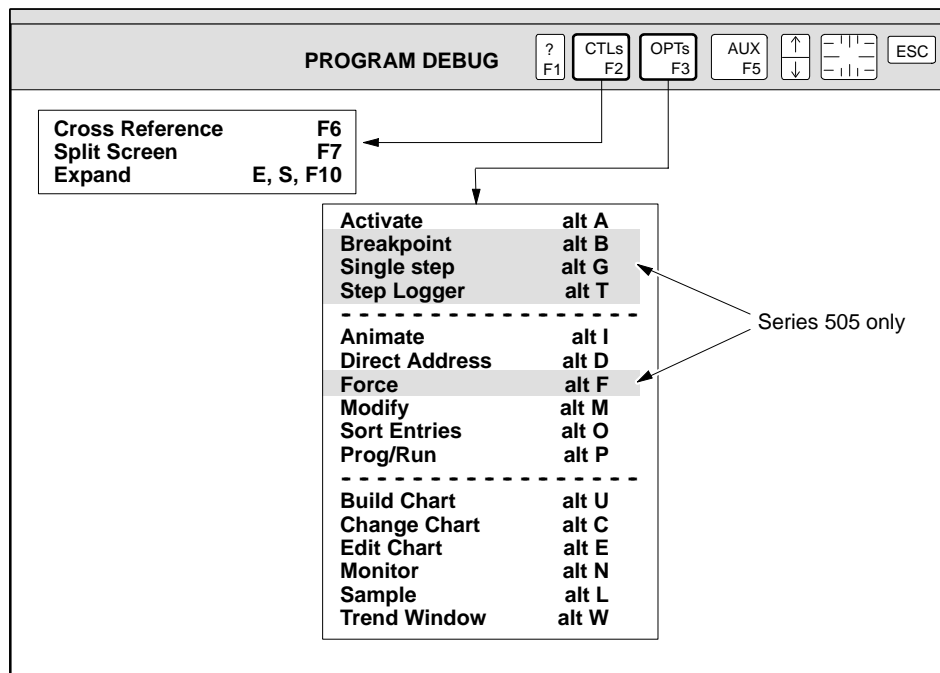


































Figure 7-2 Controls and Options (Keys F2 and F3)

Table 7-1 Debug Options Defined

Command	Keys	Remarks
Activate	 	Activates an SFC step or CFB. See Section 7.5 .
Animate	 	Shows the status of the set of variables or extensions associated with an object. See page 7-28 . Shows the active steps in an SFC. See page 7-34 .
Breakpoint ^{1,2}	 	Sets a breakpoint, or stopping place, in an SFC. See page 7-35 .
Build Chart	 	Automatically generates a debug monitor chart for an I/O or declaration table or a CFB, SFC, or subroutine. See page 7-19 .
Change Chart	 	Displays another chart of variables and sets the default chart. See page 7-23 .
Direct Address	 	Displays the controller address for a symbol. See page 7-29 .
Edit Chart	 	Displays a chart in an edit mode, allowing you make changes to the chart. See page 7-22 .
Force ²	 	Forces the values of variables. See page 7-32 .
Modify	 	Changes the value of a variable. See page 7-31 .
Monitor	 	Displays a chart of values, which are updated with each scan of the controller. See page 7-24 .
Prog/Run	 	(Series 505) Changes the controller from run to program or from program to run and places the sequencer and loop card in different modes. (S5) Changes the controller from run to stop mode or from stop to run mode with a warm start or a cold start. This option can also be used to determine the current controller mode. See page 7-14 .
Sample	 	Displays a chart of values, which are sampled on demand. See page 7-24 .
Single Step ^{1,2}	 	Allows you to stop the SFC after each step. See page 7-36 .
Sort Entries	 	Sorts the list of objects in a Declaration, Device, I/O, or Recipe table by name or type. See page 7-17 .
Step Logger ^{1,2}	 	Produces a trace record of the program steps as each becomes active. Icons allow you to review the record or clear the record and start the Step Logger again. See page 7-37 .
Trend Window	 	Displays the first 12 chart entries as a trend. See page 7-27 .
<p>1 Available only if you select Create debug version when you compile the program.</p> <p>2 Not supported for S5 controllers.</p>		

Understanding the Debug Operation (continued)

Using the Series 505 AUX Functions

During Debug operation, the **AUX** icon allows you to execute the Series 505 controller auxiliary (AUX) functions that are shown in [Figure 7-3](#) and defined in [Table 7-2](#). For a more detailed explanation of the operation of the AUX functions, refer to your controller user manual.

To execute an AUX function, press **F5**, or use the **AUX** icon to select one of the AUX functions.

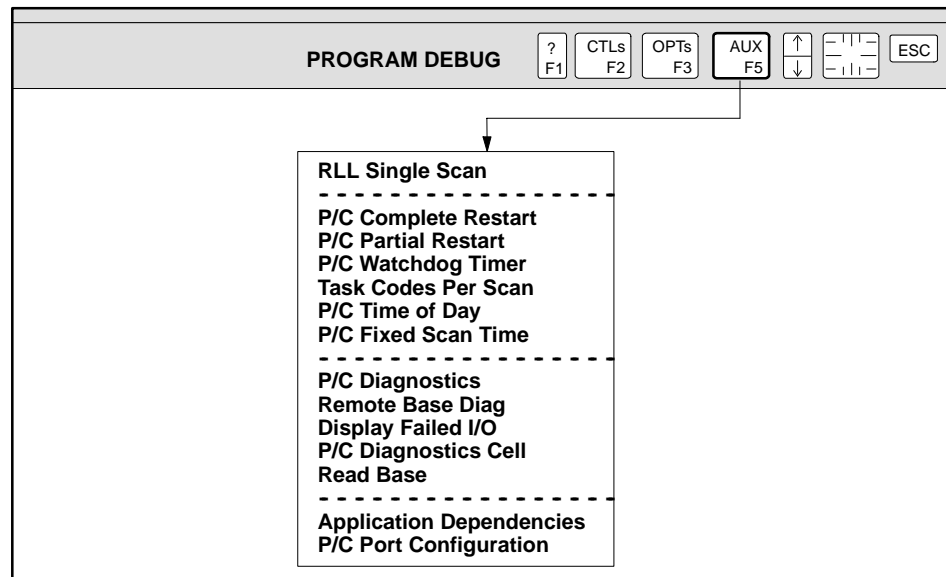


Figure 7-3 Series 505 AUX Functions (Key F5)

⚠ WARNING

The **P/C Complete Restart** option, available from the **AUX** icon (**F5**) menu, clears all booleans in the controller. To return the booleans to their proper state, you must perform a full download. If you choose **P/C Complete Restart** and do not perform a full download, you do not have control of your process.

Losing control of your process can cause death or serious injury to personnel, and/or damage to equipment.

Be sure you understand all the consequences before choosing the **P/C Complete Restart** option, and be sure that you perform a full download after choosing this option.

Table 7-2 Series 505 AUX Functions Defined

AUX Function	Remarks
RLL Single Scan	Causes the Series 505 controller to execute one scan of the ladder logic. Controller must be in PROGRAM mode.
P/C Complete Restart	Clears discrete I/O image registers, word I/O image registers, non-retentive CRs, retentive CRs, and fatal error. TMR/CTR/Drum presets are downloaded. Forced elements and memory locations that are not mentioned are not cleared. Places the controller in Program mode if a fatal error is present, or if the previous mode was Program.
P/C Partial Restart	Clears discrete I/O image registers, non-retentive CRs, and fatal error. TMR/CTR/Drum presets are not downloaded. Word I/O image registers, retentive CRs, forced elements, and memory locations that are not mentioned are not cleared. Places the controller in Program mode if a fatal error is present, or if the previous mode was PROGRAM.
P/C Watchdog Timer	Allows you to read or set the controller watchdog timer.
Task Codes Per Scan	Allows you to set the number of task codes that the controller executes per scan, per channel.
P/C Time of Day	Allows you to read or set the controller real-time clock.
P/C Fixed Scan Time	Allows you to set the controller scan time either to a fixed value or to vary from scan to scan; also allows you to set the time slices in the time line configuration.
P/C Diagnostics	Allows you to initiate controller diagnostics. The controller must be in Program mode before you can execute this AUX function.
Read Base	Allows you to read the I/O configuration for a Series 500/505 base or a Series 505 FIM module and its slaves. Displays information on screen and saves a report with name CHxBAYy.RPT where x is the channel number and yy is the base number.
Remote Base Diagnostics	Allows you to run diagnostics on each remote base controller configured in the system.
Display Failed I/O	Displays the locations of any failed I/O modules which are capable of diagnosing and indicating failure; also displays I/O mismatches, indicating that one or more I/O modules do not agree with the I/O configuration.
P/C Diagnostics Cell	Checks the operational status of the controller and displays the results.
Application Dependencies	Configure the Required, Optional, and Mode-Locked application dependencies of a 575.
P/C Port Configuration	Configures one of the communication ports of any 575 application.

Understanding the Debug Operation (continued)

Using the S5 AUX Functions

During Debug operation, the **AUX** icon allows you to execute S5 controller auxilliary functions that are shown in [Figure 7-4](#) and defined in [Table 7-3](#).

To execute an AUX function, press **F5**, or use the **AUX** icon to select one of the AUX functions.

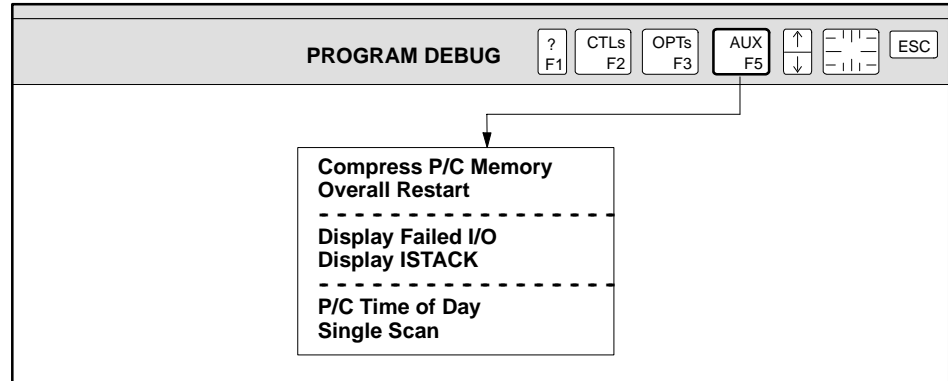


Figure 7-4 S5 AUX Functions (Key F5)

Table 7-3 S5 AUX Functions Defined

AUX Function	Remarks
Compress P/C Memory	Compresses S5 controller memory by deleting all holes in memory that were created during the deletion of objects.
Overall Restart	Clears all memory: flags, extended flags, program area (FBs, PBs, SBs, OBs), data blocks, extended data blocks, digital I/O, timers and counters.
Display Failed I/O	(948 only) Displays failed I/O locations in the PII and PIQ area.
Display ISTACK	Displays two levels of ISTACK. Information about controller error causes and locations are returned.
P/C Time of Day	Allows you to read or set the controller real-time clock.
Single Scan	Performs a single cyclic scan. First the single scan mode must be enabled by selecting ENABLE, then single scans are performed by selecting SCAN, and finally you must disable the single scan mode by selecting DISABLE.

 WARNING

The Overall Restart option, available from the AUX icon (F5) menu, clears all memory in the controller. To restore memory, you must perform a full download. If you choose Overall Restart and do not subsequently perform a full download, you do not have control of your process.

Losing control of your process can cause death or serious injury to personnel, and/or damage to equipment.

Be sure you understand all the consequences before choosing the P/C Overall Restart option, and be sure that you perform a full download after choosing this option.

Understanding the Debug Operation (continued)

Debugging Any Controller Program

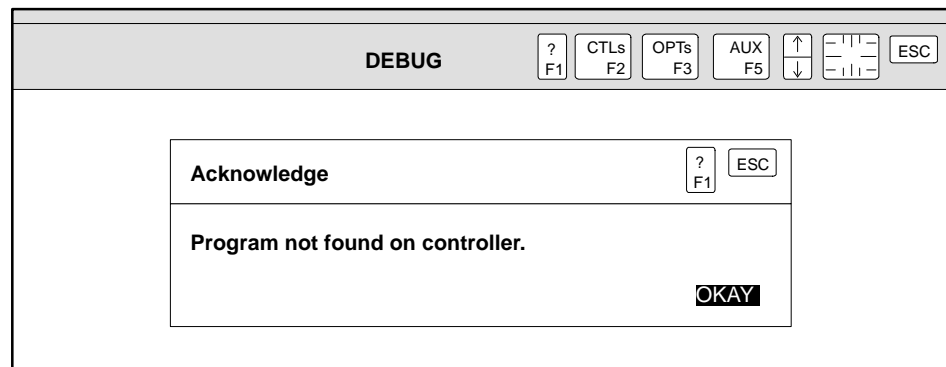
You can do a limited debug of a controller program for which there is no APT program or name listed in the Program Content. The program does not have to be created in APT. For example, a RLL program that you created with TISOFT, or an STL program created with STEP 5, can be debugged through APT when it is present in the controller.

The following options are available to you during this limited debug procedure:

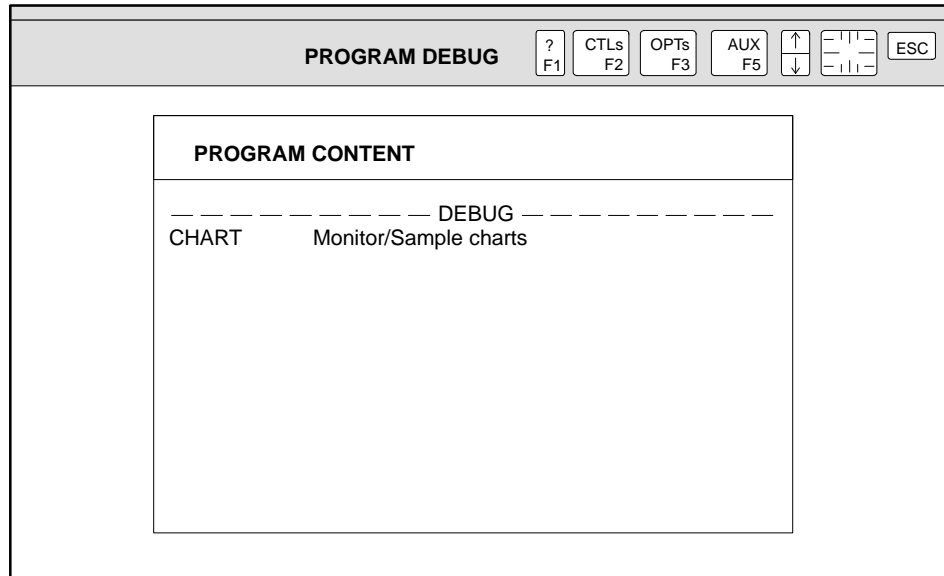
Change Chart	Program/Run modes
Edit Chart	AUX Functions
Force (Series 505 only)	Sample
Modify	Trend Window
Monitor	

To access a non-APT program, or an APT program not listed in the Program Content, follow these steps.

1. In the Program Directory, place the cursor on any listed program.
2. Press , or use the **OPTs** icon to select **Debug**. The following dialog window appears.



-
3. To debug the program, press **Enter**. The Program Content screen appears:



You must use direct addressing to access variables. This means that when you build a chart, for example, you specify the actual memory locations within the controller memory, such as %V100, %WX34, or %LERR1 for Series 505, and %FW100, %PW128, or %I12.1 for S5.

Understanding the Debug Operation (continued)

Using Prog/Run Option

While you are in the Debug Utility, the Prog/Run option allows you to check the current mode of the controller as well as to change modes.

While Debug is being executed the following information is displayed in the upper left corner of the screen.

P/C PGM: the controller is in Program (Stop) mode.

P/C RUN: the controller is in Run mode.

COMM: communications to the controller have been interrupted.

(Series 505)

When you change from run to program mode, you have these options for setting the controller.

Program Freeze leaves all I/O points and boolean variables in their current state when the controller goes to program mode.

Program Off turns off outputs and leaves all other memory locations in their current state.

WARNING

APT allows you to place the controller in the PROGRAM OFF state. In this state, all discrete outputs (Ys) are turned off. Any Ys that are being used by APT as overflow CRs will also be turned off. For TISTAR and OSx: "Secondary Commands from Network Utilities" on a TISTAR system changes to PROGRAM OFF state. PCS 3.x and OSx 4.x do not change mode of the controller.

Incorrect de-energization of output Ys and overflow CRs could cause unexpected operations and result in death or serious injury to personnel, and/or damage to equipment.

Fully understand what will happen when you transition to the PROGRAM OFF state. Only qualified personnel should be allowed to make the decision to change the controller state to PROGRAM OFF.

If you want to change from program to run mode, display the Program/Run dialog window and choose the Run option. After you select the loop mode, the controller is in run mode.

When you change the loop mode, you have these options.

Follow sets the loop mode to follow the mode that you specify for the controller.

Program allows you to set the loop mode to program mode when you set the controller to program mode. You cannot set the loop mode to program mode at the same time that you set the controller to run mode.

Run allows you to set the loop mode to run when the controller is in program mode.

(S5)

When you change from Stop to Run mode, you have these options for setting the controller.

Stop puts an S5 controller in Stop mode. All I/O points and flag variables are left in their current state when the controller goes to Stop mode. All digital outputs go low. The PII, PIQ, flags, and other memory retain the value that they contained when the CPU went into Stop mode.

Coldstart puts an S5 controller in Run mode, using a coldstart where all flags, counters, and timers are reset to zero. Program executes at OB20 and then starts at the beginning of OB1.

Warmstart puts an S5 controller in Run mode, using a warmstart where all flags, counters, and timers are retained. Program executes OB21 or OB22, then continues from where program was stopped. Outputs are not updated until a complete program scan occurs.

If you want to change from Stop to Run mode, select Run in the Prog/Run option.

WARNING

APT allows you to place the controller in the Stop mode. In this mode, all discrete outputs (Qs) are turned off. PCS 3.x and OSx do not change mode of the controller.

Incorrect de-energization of output Qs could cause unexpected operations which could result in death or serious injury to personnel, and/or damage to equipment.

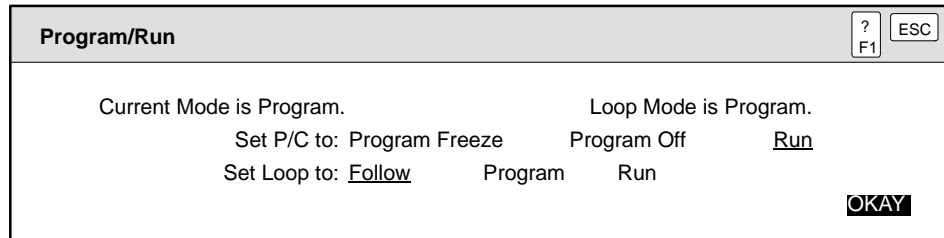
Fully understand what will happen when you transition to the Stop mode. Only qualified personnel should be allowed to make the decision to change the controller mode to Stop.

Understanding the Debug Operation (continued)

(Series 505)

To change the mode of your controller, follow these steps.

1. Press **Alt P**; the dialog window appears:

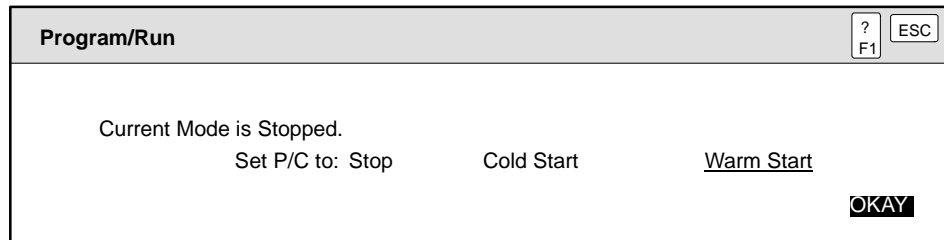


2. Move the text cursor to highlight the appropriate choices for controller mode and for loop mode, and press **Enter**.
3. Press **Enter** again to complete the action.

(S5)

To change the mode of your controller, follow these steps.

1. Press **Alt P**; the dialog window appears:



2. Move the text cursor to highlight the appropriate choices for controller mode and press **Enter**.
3. Press **Enter** again to complete the action.

Using Sort Entries Option

Use the Sort Entries option, **[Alt] [O]**, to sort the objects in a Declaration, Device, I/O, or Recipe table by either name or type.

NOTE: Sorting the entries only affects the way the listing is displayed in the Debug hierarchy. It does not affect the order of the objects when you view the table in the Program hierarchy.

Using Split Screen Option

You cannot see the code for the SFCs, CFBs, or subroutines when you are in the Debug Utility; however, you can use the Split Screen option to view code for an SFC, a CFC, or any of the tables (I/O, Device, Declare, Recipe, Subroutine). The Split Screen option displays the same view as the Show option in the APT program edit environment (before you run Debug). If you use the Split Screen, Debug appears in the top half of the screen and the APT source code appears in the bottom half of the screen.

You can split the screen while you are displaying the Debug hierarchy, or while you are using the Animate or Step Logger option.

- From the Debug hierarchy, place the text cursor on the object (name of an SFC, CFC, CFB, I/O table, Device table, etc.) and press **[Alt] [S]** or **[F7]**. The selected object appears on the lower portion of the screen.
- From within an object that you have already expanded or animated, press **[Alt] [S]**, or **[F7]**. The object appears on the lower portion of the screen.

When the lower portion of the screen is active, you can expand the object (SFC step, CFB, table object) and you can use the Zoom mode. When you are in the Debug Utility, however, you cannot make any changes to the object. To return to the upper portion of the screen, press **[F7]**. To exit the Split Screen option, make sure that you are in the bottom portion of the screen and press **[Shift] [F7]** or **[Esc]** until you exit from the Show screen.

Exiting Debug (and Forced I/O Display for Series 505)

To exit the Debug Utility, press **[Esc]**, or select the **ESC** icon from the program content level in the hierarchy. A dialog window is displayed prompting you to confirm return to the hierarchy. For Series 505 controllers, your confirmation to exit causes a second dialog window to appear if there are any forced I/O locations. You can then either request the I/O to be displayed or exit to the hierarchy.

If the forced I/O is displayed, each point forced is listed by its controller address. If a forced compile has not been done since the point was forced, its symbol name is also listed.

7.2 Monitoring Variables

Overview

APT provides four methods to observe the values of variables.

- You can use the Monitor option to view a chart of user-selected variables that are updated continuously from values in the controller. APT supports up to 100 charts, with up to 42 variables displayed in each.

You can use the Sample option to view a chart of user-selected variables that are updated from values in the controller once after each request. As with the Monitor option, you can use the Sample option to view/edit up to 100 charts, with up to 42 variables displayed in each.

- You can use the Trend option to view a trend of the first 12 entries in a chart.
- You can use the Expand and Animate options to display a pre-built chart of single Table entries in the IO, Device, CFB, or Declare tables. The values of the variables or extensions associated with the entry that you choose are updated continuously.
- You can use the Modify option to display the value of a variable.

Working with Charts

From the Chart Content level of the Debug hierarchy, you can:

- Copy a chart: press
- Delete a chart: press
- Rename a chart: press
- Change the description of a chart: press
- Sort charts by name or description: press

You can also use the **OPTs** icon in the Chart Content level to select these options.

There are two ways to create a chart: you can build the chart manually, or use the Build Chart option to generate a chart automatically.

Manually Building a Chart

Before you can manually build a chart, you must display the Chart Content level of the Debug hierarchy. After starting Debug, place the cursor on **CHART Monitor/Sample charts** in the Program Content directory, and press , or , or use the **CTLs** icon to select **Expand**. This takes you to the Chart Content level of the Debug hierarchy.

The default chart \$DEBUG is always present in Chart Content. Until you enter some variables to be displayed in it, however, it is empty. You can enter data in the \$DEBUG chart, or add a new chart. To add a new, empty chart, press , or use the **OPTs** icon to select **Add**. APT prompts you for a chart name and description.

To manually build your chart, whether it is the \$DEBUG chart or one that you have added, you must edit the chart using the procedure described on [page 7-22](#).

Automatically Building a Chart

You can use the Build Chart option, **[Alt] [U]**, to generate a debug monitor chart automatically from an I/O or Declarations table, or from a specified CFB, SFC, or subroutine. The maximum size possible for a chart generated with the Build Chart option is 42 elements.

To use the Build Chart option, you must first open the desired table, CFB, SFC, or subroutine from the Debug hierarchy. You can either press **[Alt] [U]**, or use the **OPTs** icon to select Build Chart. [Figure 7-5](#) gives an example of how to select Build Chart from within an object table.

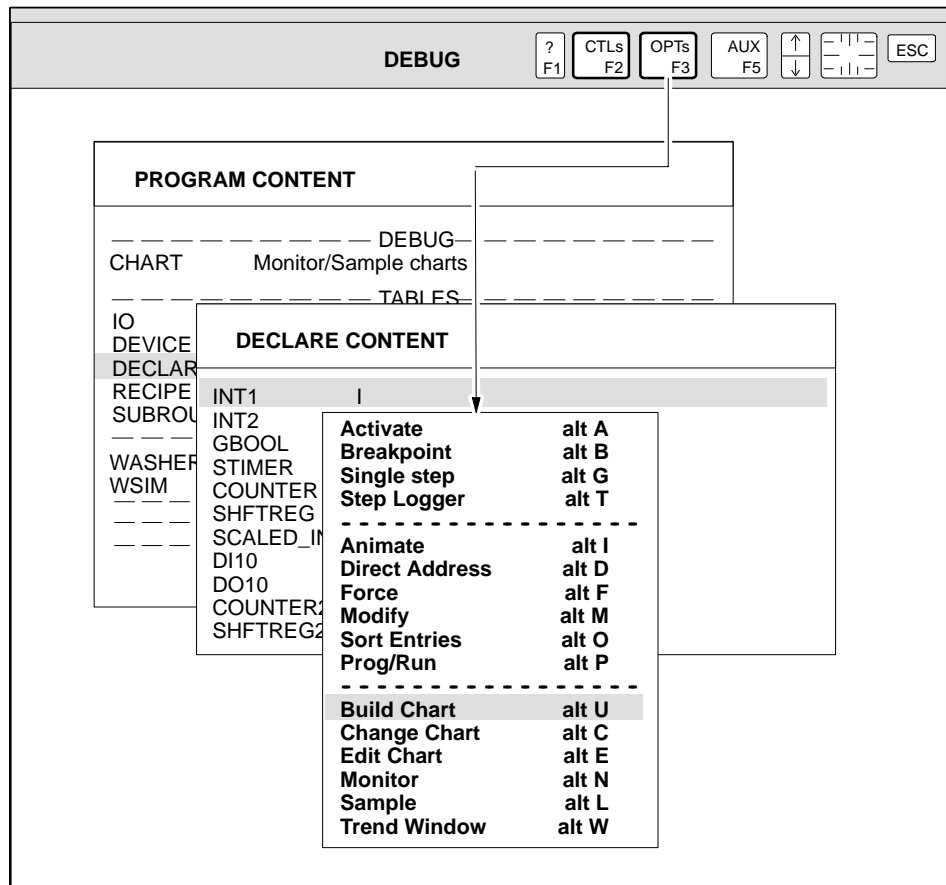
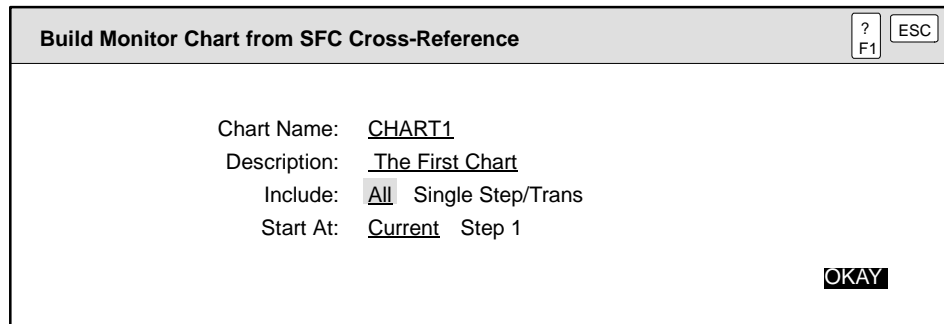


Figure 7-5 Accessing the Build Chart Option

Monitoring Variables (continued)

Completing the Build Monitor Chart Dialog Box

The fields in the Build Monitor Chart dialog box vary slightly depending on the nature of the object for which you are building a chart.



Build Monitor Chart from SFC Cross-Reference

Chart Name: CHART1

Description: The First Chart

Include: All Single Step/Trans

Start At: Current Step 1

OKAY

I/O Fill in the Chart Name field with a unique, meaningful name up to 8 characters long. The Description field is optional and has a maximum length of 30 characters. For the Start At field, choose either Current, to begin on the element that your cursor is highlighting, or First, to start the chart at the first element in the table.

Declare Fill in the Chart Name field with a unique, meaningful name up to 8 characters long. The Description field is optional and has a maximum length of 30 characters. For the Start At field, choose either Current, to begin on the element that your cursor is highlighting, or First, to start the chart at the first element in the table.

CFB Fill in the Chart Name field with a unique, meaningful name up to 8 characters long. The Description field is optional and has a maximum length of 30 characters.

SFC Fill in the Chart Name field with a unique, meaningful name up to 8 characters long. The Description field is optional and has a maximum length of 30 characters. The Include field allows you to choose between charting elements selectively (Single Step/Trans) or exhaustively (All). To use the Single Step/Trans option, your cursor must be highlighting the element of the SFC that you want to chart; otherwise, you must select All. The Start At field allows you to decide where the chart should begin; if your SFC contains more than 42 elements, you can create multiple charts and sequence them by using this field. Select Step 1 if you want the chart to start at the beginning of the SFC. If you want to select Current, so that this chart picks up where a previous chart left off, your cursor must be highlighting the desired step or transition.

NOTE: If you edit an SFC after compiling, your step and transition numbers may get out of sequence. Build Chart cannot detect this. To avoid referencing out-of-order steps in your chart, use the Sequence Labels option to correct step and transition numbers while you are in the SFC Editor, and recompile the SFC. Then go to Debug and run Build Chart again.

Subroutine Fill in the Chart Name field with a unique, meaningful name up to 8 characters long. The Description field is optional and has a maximum length of 30 characters.

When you generate a monitor chart with the Build Chart option, this chart becomes the default chart, and is displayed when you choose the Monitor option, Alt N.

Charts generated by the Build Chart command can be edited using the same procedure that applies to a manually constructed chart. See [page 7-22](#).

Monitoring Variables (continued)

Editing a Chart

All charts, whether they have been created manually or automatically (using the Build Chart option), can be edited by the same method. To edit a new or existing chart, go to the Chart Content level of the Debug hierarchy, place the cursor on the chart name, and either press **F10** or **Enter**, or use the **CTLs** icon to select **Expand**. The **OPTs** icon displays options you can use to make changes in your chart entries, or to display a different chart. These chart editing options are defined in [Table 7-4](#). To enter variables into the chart, follow these steps.

1. Enter the name of the variable in the appropriate field, and press **Enter**. For local math block variables, this must be the complete variable name. The complete variable name includes the SFC and step names, or the CFB block names.

For locally-declared variables in a Math CFB use

CFB_name.NNL.Local_Variable (NNL is always used in the designation.)

For locally-declared variables in an SFC step use

\$SFC_name.SMATHx.Local_Variable (x represents the step number).

CFB_name and **SFC_name** are the names of your CFB or SFC, respectively; **Local_Variable** is the name of the locally defined variable.

2. If the variable is defined at the Unit Content level, you must enter the name of the unit containing the variable, and press **Enter**. This field is blank for a variable defined at the Program Content level.
3. Integers can be displayed in integer, ASCII, binary, or hexadecimal format. To display an integer value in BCD or hexadecimal format, put **H** in the format field; to display an integer value in ASCII, use **A**. To display an integer value in binary format, put **L** in the format field to display the lower byte of the integer, or **U** to display the upper byte. The default is to display in integer format (**I**).
4. For S5 controllers, you can display bytes and double words. For a byte, put **Y** in the format field. For a double word, put **D** in the format field.

You can enter comments into a chart for documentation purposes. Place the comment in the **Name** field, and precede the comment with a brace character (**{**) as shown in [Figure 7-6](#). A trailing brace character (**}**) is optional.

WASH_DEM CHART		DRAIN	
P/C RUN	RANGE ALTS	DEBUG	ESC
DRAIN	Name	Unit	Format
	STRT_DRAIN {THIS IS COMMENT TEXT}	WASHER .	.
	DRAIN_DONE {THIS IS MORE COMMENT TEXT}	WASHER .	.
	{THIS IS A COMMENT LINE	.	.
	R_DRAIN	.	.
	STOP_DRAIN	WASHER .	.
	{THIS IS A COMMENT LINE	.	.
	WASH_OP {THIS IS COMMENT TEXT	WASHER .	.
	WASH	.	.

Figure 7-6 Example of Chart Comments

Table 7-4 Chart Editing Functions

Option	Keys	Remarks
Insert Line	<input type="button" value="Alt"/> <input type="button" value="I"/>	Allows you to insert a chart entry.
Delete Line	<input type="button" value="Alt"/> <input type="button" value="D"/>	Allows you to remove a chart entry.
Erase Chart	<input type="button" value="Alt"/> <input type="button" value="E"/>	Allows you to remove all entries from the chart.
Undelete Line	<input type="button" value="Alt"/> <input type="button" value="X"/>	Allows you to replace the last line that you deleted.
Undo Changes	<input type="button" value="Alt"/> <input type="button" value="U"/>	Replaces changes you have made with the original information.
Find Text	<input type="button" value="Alt"/> <input type="button" value="F"/>	Allows you to locate a specified string of text.
Replace Text	<input type="button" value="Alt"/> <input type="button" value="T"/>	Replaces a specified string of text with another. You have the option of replacing a single line, or all occurrences, of a string of text.
Mimic Line Above	<input type="button" value="Alt"/> <input type="button" value="M"/>	Copies information from one line to the next. If the line contains an array, the index is automatically incremented by one.
Change Chart	<input type="button" value="Alt"/> <input type="button" value="C"/>	Allows you to display another chart.

Listing Charts

To display a list of charts, place the cursor on **CHART Monitor/Sample charts** in the Program Content directory and press , or , or use the **CTLs** icon to select **Expand**. This takes you to the Chart Content level of the Debug hierarchy.

The chart \$DEBUG is always listed and is the default chart at the beginning of each Debug session, until you perform an operation on another chart. The indicator <=== denotes the default chart. The default chart is the one APT displays when you select **Monitor** or **Sample** from the Debug hierarchy.

Changing Charts

When a chart is displayed, you can press to change to another chart. This action also changes the default chart to the new chart. If you are uncertain of the chart name, press to see a list of all charts.

Monitoring Variables (continued)

Using the Monitor and Sample Options

You can use the Monitor and Sample options to view a chart of user-selected variables.

- The Monitor option updates the values of the variables continuously from values in the controller.
- The Sample option updates the values each time you select the option.

You can select **Sample** to switch from continuous polling to a single poll mode; you can select **Monitor** to switch from the single poll mode to continuous polling.

You can view the chart using the Monitor or Sample options from any point in the Debug utility, except from the bottom half of a split screen.

To monitor values, press **Alt N**, or use the **OPTs** icon to select **Monitor**. To sample values, press **Alt L**, or use the **OPTs** icon to select **Sample**.

APT polls the controller continuously (Monitor option) or a single time (Sample option). The current values of the variables listed in the default chart, if you have entered any variables, are displayed. The screen is similar to the one shown in [Figure 7-7](#).

WASH_DEM CHART							
P/C RUN	DEBUG	? F1	CTLs F2	OPTs F3	AUX F5	↑ ↓	ESC
DRAIN	Name	Unit	Value				
	STRT_DRAIN	WASHER	FALSE				
	DRAIN_DONE	WASHER	FALSE				
	R_DRAIN		1056				
	STOP_DRAIN	WASHER	FALSE				
	WASH_OP	WASHER	32				
	WASH		32				
	RINSE		40				

Figure 7-7 Monitoring Variables

You can make changes to the variables. Move the text cursor to a variable and press **Alt M** to **Modify** the variable. For Series 505 controllers, you can also press **Alt F** to **Force** a variable. APT displays a dialog window so that you can designate changes to the variable. Refer to [Section 7.3](#) for information about changing I/O and variables.

You can edit the contents of the chart that you are viewing. Press **Alt E** or use the **OPTs** icon to select **Edit Chart**. The fields are displayed as shown in [Figure 7-8](#).

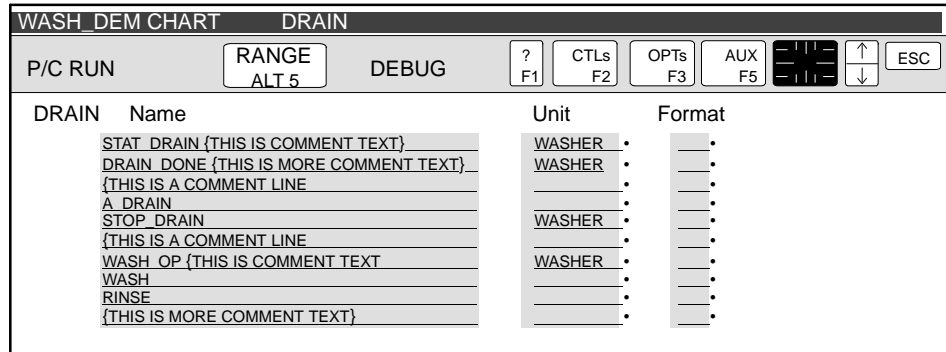


Figure 7-8 Chart Fields during an Edit

If you need to see another chart, press **Alt C**, or use the **OPTs** icon to select **Change Chart**. APT prompts you for the name of the new chart. Use the Completion Aids if you do not remember the chart name.

Monitoring Variables (continued)

Using Trend Option You can use the Trend option to view a trend of the first 12 entries of a chart.

You can view the trend using the Trend option from any point in the Debug utility, except from the bottom half of a split screen.

To trend values, press **Alt** **W**, or use the **OPTs** icon to select **Trend**.

- You can edit the contents of the chart you are viewing by pressing **Alt** **E**, or you can use the **OPTs** icon to select **Edit Chart**.
- You can switch to the Monitor option by pressing **Alt** **N**, or you can use the **OPTs** icon to select **Monitor**.
- You can switch to the Sample option by pressing **Alt** **L**, or you can use the **OPTs** icon to select **Sample**.
- You can use the **STOP** icon or **Alt** **3** to freeze the current trend. The trend stops when it reaches the right hand edge of the screen. To restart the current variable trending, press **Alt** **3** again. The trend information for the elapsed time is not reported.
- You can use the **RANGE** icon or **Alt** **5** to specify the high and low values, the color, and bold line format for each variable. Additionally, the chart time range can be specified. See [Figure 7-9](#).

The trend screen is similar to the screen shown in [Figure 7-10](#). If you exit the trend screen, it will start a new trend when you return to it.

If you need to see another chart, press **Alt C**, or use the **OPTs** icon to select **Change Chart**. APT prompts you for the name of the new chart. Use the Completion Aids if you do not remember the chart name.

Trend Ranges						? F1	ESC
Name	Unit	Low	High	Color	Bold		
FILL_DONE	WASHER			Blue	• []		
R_FILL		0	100	Blue	• []		
STOP_FILL	WASHER			Red	• []		
WS_STOP_CMMD	WASHER	0	100	Red	• []		
FILL_SETPT	WASHER	0.000	100.000	Grey	• []		
WS_LEVEL.IM	WASHER	0.000	100.000	Green	• []		
COLD_WATER.OPND	WASHER			Grey	• []		
HOT_WATER.OPND	WASHER			Green	• []		
TEMP_ALM.IN	WASHER	0.000	100.000	Cyan	• []		
ADD_SOAP_TMR.TCC	WASHER	0	500	Black	• []		
FILL_LEVEL	WASHER	0	100	Blue	• [X]		
LOW_LEVEL	WASHER	0	100	Red	• [X]		

Time Range: **1** Seconds Minutes Hours
OKAY

Figure 7-9 Trend Window Range Configuration

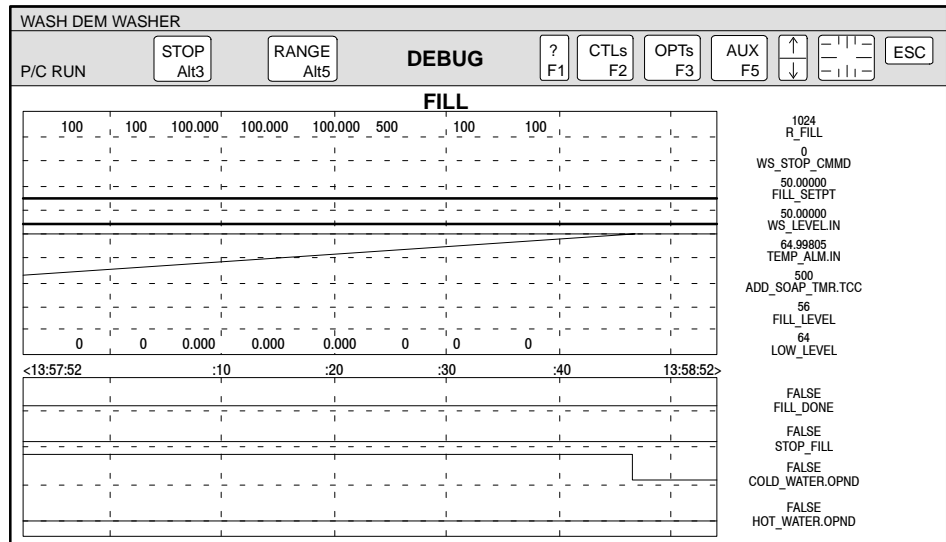


Figure 7-10 Trend Window

Monitoring Variables (continued)

Using Animate Option

To observe the current status of the set of variables or extensions associated with a single Table entry or CFB, place the cursor on the appropriate table (IO, Device, Declare, SFC, or CFB) and press **E**, **F10** or **Enter**. Then place the cursor on the appropriate entry within the table and press **Alt**, **I**, **Enter**, or use the **OPTs** icon to select **Animate**. Refer to [Section 7.4](#) for more details about the Animate option. For example, APT displays this dialog window for a Single-drive/Single-feedback Motor (MSS) as shown in [Figure 7-11](#).

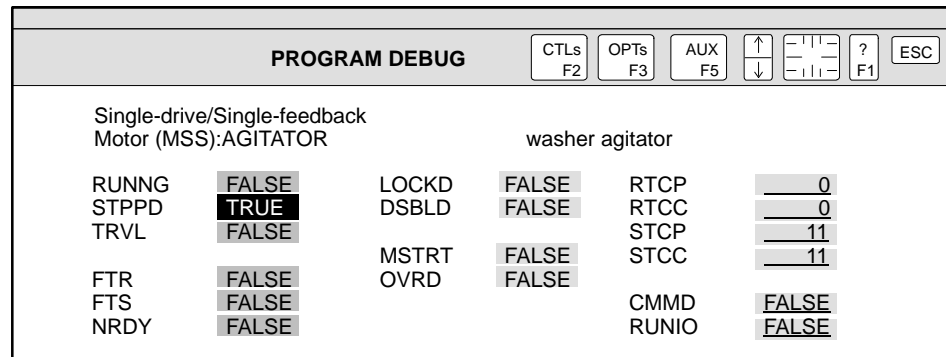


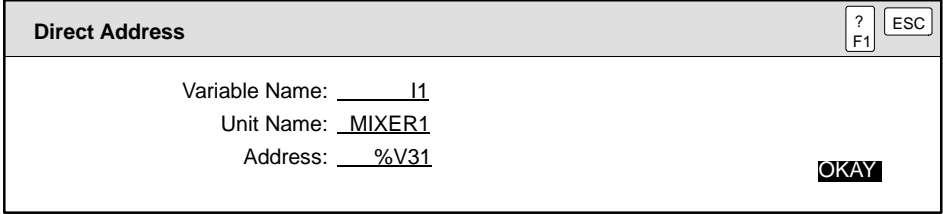
Figure 7-11 Animating MSS Extensions

You can make changes to the variables. Move the text cursor to a variable and press **Alt**, **M** to **Modify** the variable. For Series 505 controllers, you can also press **Alt**, **F** to **Force** a variable. APT displays a dialog window for you to designate the changes to the variable. Refer to [Section 7.3](#).

Using Direct Address Option

To determine the controller address for a variable, press **Alt D**, or use the **OPTs** icon to select **Direct Address**. The dialog window appears as shown in [Figure 7-12](#).

When you enter the variable and unit names, Debug displays the controller address that is associated with the variable, such as `%V100` for Series 505, or `%DB6:DW24` for S5.



Direct Address		? F1	ESC
Variable Name:	_____I1		
Unit Name:	MIXER1		
Address:	_____%V31		
			OKAY

Figure 7-12 Reading the Direct Address of a Variable

NOTE: Addresses may change when you do another compile of the program.

7.3 Changing I/O and Variables

Overview

During the debug operation, APT provides two methods for changing controller values.

Modify allows you to make changes to I/O points and variables.

Force (Series 505 only) allows you to set I/O points and boolean variables to specified values and force them to remain in that condition until you unforce them. The Force option provides three operations:

- **Force** allows you to change I/O points to specified values. When an I/O point is forced, attempts to change that value from the program have no effect.
- **Unforce** allows you to unforce a single I/O point or boolean variable.
- **Unforce All** is a single step that allows you to unforce all forced I/O points, remove all breakpoints, and remove all single-step options that are turned on in SFCs.

WARNING

Forced elements on Series 505 controllers affect hardwired inputs and outputs as well as internal control relays (CR memory). Inputs and outputs are set to On or Off and can affect the operation of machinery. Forced elements remain forced when you download a new program to the controller.

Forced elements could result in unexpected operations which could cause death or serious injury to personnel, and/or equipment damage.

Fully understand what will happen when you force an element or download a program to a controller with forced elements. Only qualified personnel should be allowed to use the FORCE functions.

Using Modify Option

To use the Modify option, follow these steps.

1. Press **Alt M**, or use the **OPTs** icon to select **Modify**. The dialog window appears as shown in [Figure 7-13](#).

The screenshot shows a dialog box titled "Modify". At the top right of the dialog are buttons for "? F1" and "ESC". The main content area contains the following labels and input fields:

- Variable Name: _____
- Unit Name: _____ .
- Hex: [] Binary: []
- Value: _____

An "OKAY" button is positioned in the bottom right corner of the dialog's main area.

Figure 7-13 Modifying the Value of a Variable

2. Enter the name of the variable in the appropriate field, and press **Enter**. For local math block variables, this must be the complete variable name. The complete variable name includes the SFC and step names, or the CFB block names.
3. If the variable is defined at the program level, you must remove the unit name from the field.
4. To enter a value in hexadecimal or BCD, place an **X** in the Hex field. To enter a value in binary, place an **X** in the Binary field. APT displays the current value for the variable. If you select binary, all 16 binary digits (bits) of the word are displayed. The 8 bits on the left comprise the upper byte and the 8 bits of the right comprise the lower byte.
5. Enter the new value in the Value field, and press **Enter**. APT determines whether or not the value is Boolean or integer and automatically adjusts the value entry type.

If you select binary, you can enter up to 16 bits (zeroes and ones). Leading zeroes are not required, and you can insert spaces to improve readability. Bits are right justified; the rightmost bit corresponds to the least significant bit in the integer. If you enter fewer than 16 bits, the remaining bits of the integer are set to zero.

6. Press **Enter** again to confirm the action.

Changing I/O and Variables (continued)

Using Force Option (Series 505 Only)

To use the Force option, follow these steps.

1. Press , or use the **OPTs** icon to select **Force**. The dialog window appears as shown in [Figure 7-14](#).

The screenshot shows a dialog box titled "Force". In the top right corner, there are buttons for "? F1" and "ESC". The main content area contains the following text:
Operation: Force Unforce Unforce All
Variable Name: _____
Unit Name: _____ .
Hex: [] Binary: []
Value: _____
An "OKAY" button is located in the bottom right corner of the dialog.

Figure 7-14 Forcing a Variable

2. Select the appropriate operation.
3. Enter the variable you want to force or unforce, and press . For local math block variables, this must be the complete variable name. The complete variable name includes the SFC and step names, or the CFB block names.
4. If the variable is defined at the program level, you must remove the unit name from the field.
5. To enter a value in hexadecimal or BCD, place an **X** in the Hex field. To enter a value in binary, place an **X** in the Binary field. APT displays the current value for the variable. If you select binary, all 16 binary digits (bits) of the word are displayed. The 8 bits on the left comprise the upper byte and the 8 bits of the right comprise the lower byte.
6. Enter the appropriate value in the Value field, and press . APT determines if the value is Boolean or integer and automatically adjusts the value entry type.

If you select binary, you can enter up to 16 bits (zeroes and ones). Leading zeroes are not required, and you can insert spaces to improve readability. Bits are right justified; the rightmost bit corresponds to the least significant bit in the integer. If you enter fewer than 16 bits, the remaining bits of the integer are set to zero.

-
7. Press again to complete the action.

NOTE: The Force option acts on the discrete (X, Y), control relay (C), and word (WX, WY) image registers. You cannot force values in V-memory, such as declared integers or real numbers.

Hints: Forcing Variables

Whenever you force a variable in the controller, remember to document the variable that you forced. This simplifies the task of locating the forced variable at a later time. Also, if you do a force compile and full download of your APT program, you must re-force the variables. Consider one of the following actions as a means of documenting forced locations:

- Keep a written log near your computer and write down all locations that you have forced.
- Add a report to the global hierarchy in APT and save all force information there. That way, the information is stored with the program and is readily available when you need it.
- Create a monitor chart in Debug containing the forced variables and comments. You can then animate this chart and force or unforce elements quickly. Be sure to include the forced values with the comments for each variable. Refer to [Section 7.2](#) for information about creating monitor charts.

7.4 Tracing SFC Program Flow

Overview

APT provides several means of allowing you to follow SFC program flow during the Debug operation.

- Animate displays the SFC and can be used in combination with activate (see [page 7-38](#)), breakpoint, and single step to observe program flow.
- Breakpoint (Series 505 only) allows you to stop the program inside specified steps.
- Single Step (Series 505 only) stops the program between steps.
- Step Logger (Series 505 only) records the program flow and allows you to review the trace record.

The Breakpoint, Single Step, and Step Logger options are available only if you select **Create debug version** when you compile the program.

WARNING

The Breakpoint and Single Step options create internal forced variables. If you do not remove the Breakpoint or turn off Single Step, those internal variables remain forced even if you download a new program to the controller.

Forced variables can cause unexpected operations which could result in death or serious injury to personnel, and/or equipment damage.

Fully understand what will happen when you use Breakpoint and Single Step. Be sure to disable these options before re-downloading a new program.

Using Animate Option

Used with an SFC, the Animate option displays the SFC and highlights each step as it becomes active. Two colors can appear during animation: blue indicates that a step is active; yellow means that a step is waiting to become active.

To initiate the Animate option, place the cursor on the SFC, and press ; or press **OPTs** and select **Animate**.

NOTE: The screen display of active steps in an SFC may not always be able to keep up with the actual program flow. To observe a particularly fast program, use the Step Logger option.

Using Breakpoint Option (Series 505 Only)

The Breakpoint option allows you to set a breakpoint (or stopping place) in the specified steps of an SFC. Breakpoint provides the following options in the dialog window shown in [Figure 7-15](#).

- Set a breakpoint by specifying a step. When you set a breakpoint, the program stops at the beginning of the designated step. This step, as well as the preceding step, remains inactive until you continue the program flow or remove the breakpoint.
- Remove a breakpoint in the step.
- Allow the program to continue through a specified step that contains a breakpoint.
- Allow the program to continue through all steps that contain a breakpoint and that are waiting to become active.
- Show the next step that contains a breakpoint.

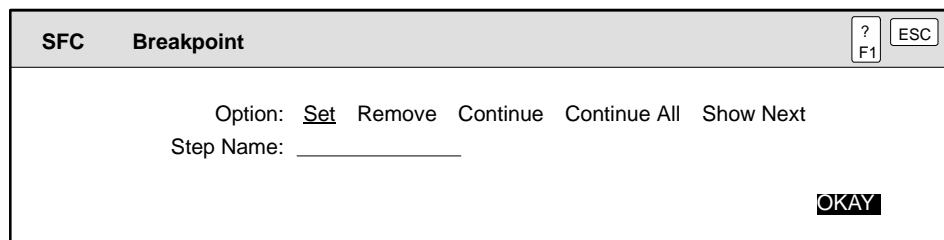


Figure 7-15 Using Breakpoint

To initiate the Breakpoint option, follow these steps.

1. Press **Alt** **B**, or use the **OPTs** icon to select **Breakpoint**.
2. Place the cursor on the appropriate response, and press **Enter**.
3. Enter the Step Name, and press **Enter**.

If you do not know the step name and are using the Animate option, move the cursor in the SFC to the cell containing the step. The prompt line at the bottom of the screen indicates the cell and step name.

4. Press **Enter** again to complete the action.

Before you can turn off the Breakpoint option, you must place the cursor either inside the expanded SFC, or over the name of the SFC in the Debug hierarchy. You can also use **Force** (press **Alt** **F** and select **Unforce All**) to turn off the Breakpoint (and Single Step) options for all SFCs.

Tracing SFC Program Flow (continued)

Using Single Step Option (Series 505 Only)

The Single Step option stops program flow at the beginning of each SFC step, allowing you to check other program conditions. To use the Single Step option, follow these steps.

1. Place the cursor on the SFC that you want to check, and press **Alt** **G**, or use the **OPTs** icon to select **Single Step**. The dialog window appears as shown in [Figure 7-16](#).

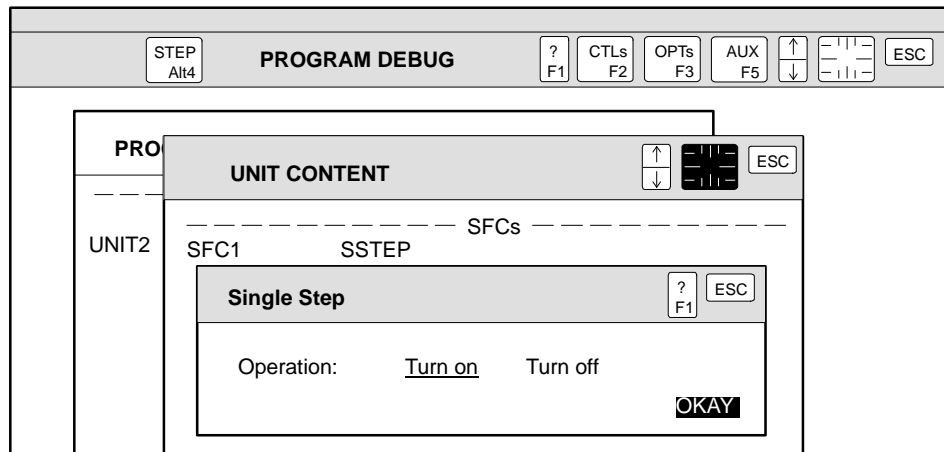


Figure 7-16 Using Single Step

2. Place the cursor on the appropriate response, and press **Enter**.
3. Press **Enter** again to complete the action.

The **STEP** icon appears at the top of the screen and the message **SSTEP** appears adjacent to the SFC that you have chosen.

After you turn on the Single Step operation, use the **STEP** icon, **Alt** **4**, to advance from one step to the next. Program flow moves to the next step only if the preceding step has completed execution.

NOTE: The **STEP** icon affects all SFCs in the current unit that have the Single Step option turned on.

Before you can turn off the Single Step option, you must place the cursor either inside the expanded SFC, or over the name of the SFC in the Debug hierarchy. You can also use **Force** (press **Alt** **F** and select **Unforce All**) to turn off the Single Step (and Breakpoint) options for all SFCs.

Using Step Logger Option (Series 505 Only)

The Step Logger option records up to 256 steps of program execution. Each unit of the program contains its own step log.

To use the Step Logger option, press **Alt** **T**, or use the **OPTs** icon to select **Step Logger**. The screen appears with the icons as shown in [Figure 7-17](#).

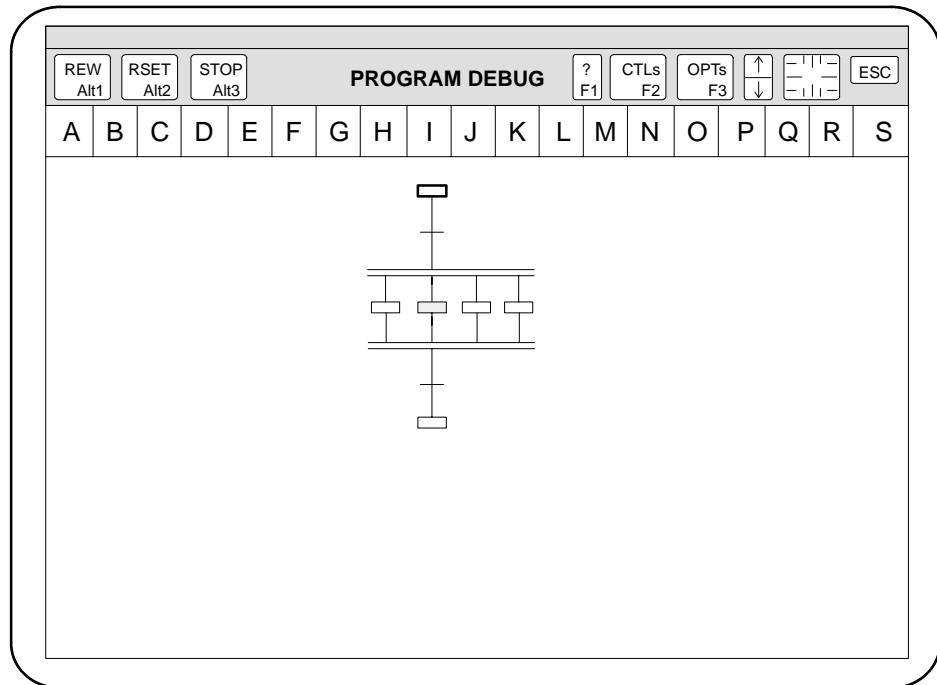


Figure 7-17 Using Step Logger

- The **REW** icon rewinds the recording and allows you to review the program execution.
- The **RSET** icon allows you to erase the current recording and to trace the next 256 steps as they are executed.
- The **STOP** icon allows you to temporarily suspend the playback of the recording. When you select **STOP**, the icon changes to **PROC**.
- The **PROC** icon allows you to continue reviewing the program execution. To speed up this playback of the recording, press the spacebar. When you select **PROC**, the icon changes to **STOP**.

7.5 Using the Activate Option

Overview

The Activate option allows you to activate and deactivate an SFC step or device, or a CFB. Typically, this option is used to debug small areas of your program.

WARNING

The Activate option allows you to alter the normal sequence of operations in your application.

Altering the flow of a program could cause unexpected operations that could result in death or serious injury to personnel, and/or equipment damage.

Fully understand the implications when you activate an SFC step. Only qualified personnel who understand the application should use this feature.

If the actual program is running in another portion of an SFC when you activate a step, it is possible to create two paths of execution in the program: one is the normal path, and the other is the path that begins where you activated a step.

Activating SFC Steps

You can select the Activate option for an SFC when the SFC name is highlighted by the cursor (at the Program or Unit Content level), or when you are using the Animate option. For Series 505 controllers, you can also activate an SFC while using the Step Logger option. Press Alt A or use the **OPTs** icon to select **Activate**. When you activate an SFC step, you have these options.

- **Active:** If you select **Active**, the step executes and the program continues from that point without affecting the actual program that is running in the controller. In this case, you can start a second path in the program.
- **Inactive:** If you select **Inactive**, the dialog window shows this line:

Force Inactive: **Yes** **No**

If you select **No**, the step remains inactive until the program flow reaches that step or until you activate it.

If you select **Yes**, the step remains inactive and program execution stops at this step until you release it. To release the step, either select **Active**, or **Force Inactive No**.

WARNING

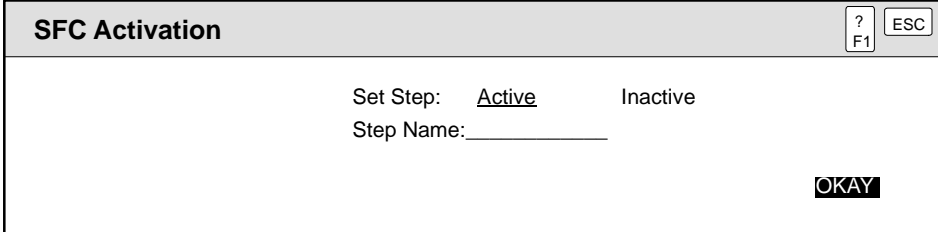
For Series 505 controllers, the system creates internal forced variables when you force a step inactive. If you do not remove the Force Inactive option, those internal variables remain forced even if you download a new program to the controller.

Forced variables can cause unexpected operations which could result in death or serious injury to personnel, and/or equipment damage.

Fully understand the implications when you Force Inactive an SFC step. Only qualified personnel who understand the application should use this feature.

To activate an SFC step, follow these steps.

1. Press Alt A . The dialog window appears as shown below.



SFC Activation

Set Step: Active Inactive

Step Name: _____

OKAY

Figure 7-18 Activating an SFC Step

2. Select the appropriate response, and press Enter .
3. Type in the name of the step, and press Enter .

If you do not know the step name and are using the Animate option, move the cursor in the SFC to the cell containing the step. The prompt line at the bottom of the screen shows the cell and step name.

WARNING

If a transition follows the step that you want to activate and that transition is true, program flow proceeds to the next step.

Altering the flow of your program could cause unexpected operations that could result in death or serious injury to personnel, and/or equipment damage.

If you know the logic contained in the following transition, you can force the transition condition to false, and stop program flow from proceeding. Only qualified personnel who understand the application should use this feature.

Using the Activate Option (continued)

Activating CFBs

To activate a CFB, you must be at the third level of the hierarchy, the CFC Content. To activate a CFB, follow these steps.

1. Place the text cursor on the name of a CFB.
2. Press **Alt** **A**. Loop CFBs appear with this dialog window.

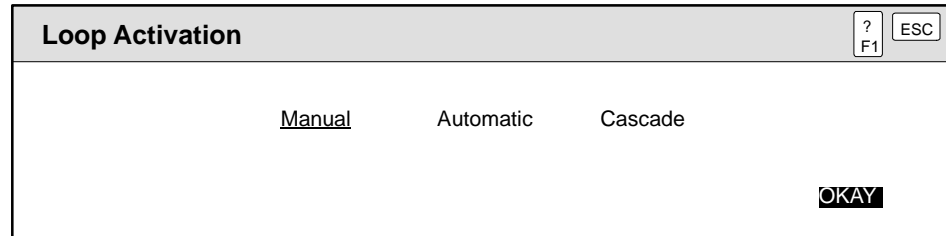


Figure 7-19 Activating a Loop

Other CFBs appears with this dialog window.

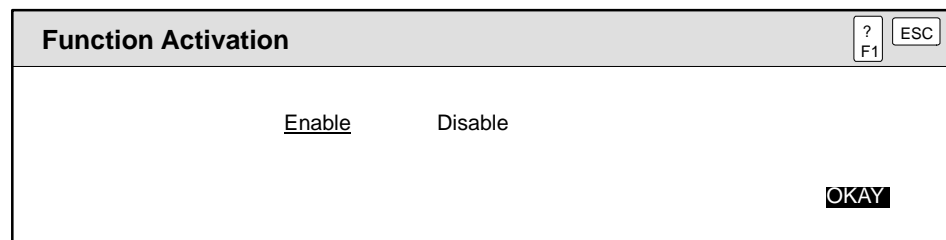


Figure 7-20 Activating a CFB

3. Select the appropriate response, and press **Enter**.

Activating Devices

When you activate a device, APT displays a chart of verbs associated with the type of device. You can then select the verbs that you need to execute. The current state of the device is the default.

To activate a device, follow these steps.

1. Place the text cursor on the name of a device.
2. Press .

For example, APT displays the following dialog window for a Single-drive/Null-feedback Motor (MSN).

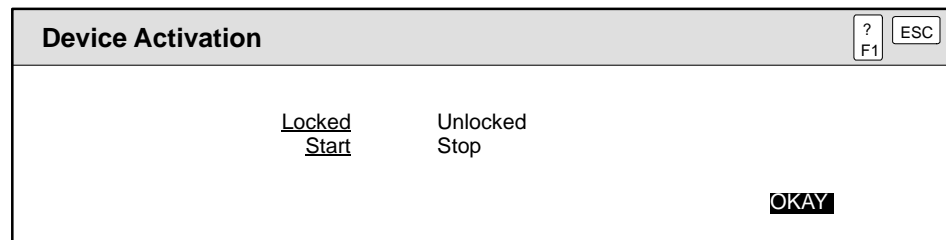


Figure 7-21 Activating a Device

3. Select the appropriate action(s), and press .

In this example, the MSN must be unlocked before you can manually execute a start or stop.

APT, OSx/TISTAR, and TISOFT/STEP 5

8.1	Translating APT Objects to OSx Tags	8-2
	For S5 Users	8-3
8.2	Tag Translation — Choosing an OSx/PCS Release	8-4
	Choosing an OSx/PCS Release	8-4
	Appending or Building the Translate Table	8-5
8.3	Tag Translation — Marking Objects	8-6
	Using the Tag Translate Editor	8-6
	Using the Tag Translate Sub-Editor	8-8
	Using the Attributes Sub-Editor	8-9
	Using the Process Groups Sub-Editor	8-10
	Unmarking an Object	8-11
8.4	Tag Translation — Compiling the Program	8-12
	Compile Options	8-12
	Compiling a Program	8-12
8.5	Tag Translation — Selecting the Translate Utilities	8-16
8.6	Translation on an OSx Engineering Node	8-17
8.7	Translation from an APT Workstation to an OSx System	8-18
8.8	Translation on a TISTAR Operator Station with APT	8-19
8.9	Translation from APT Workstation to TISTAR Operator Station with APT	8-20
8.10	Translation from APT Workstation to TISTAR Operator Station w/o APT	8-22
	Placing File TRANS.EXE on the Operator Station	8-22
	Placing File INSTALL.TAG or APPEND.TAG on the Operator Station	8-24
	Executing Translation	8-24
8.11	Translation from APT Workstation to TISTAR Model 20 System	8-25
	Transferring INSTALL.TAG to a Diskette	8-25
	Translating the Tags	8-26
8.12	Using TISOFT with APT	8-28
	Creating TISOFT Synonyms and Comments	8-28
	Examining APT Code	8-29
8.13	Using STEP 5 With APT	8-30
	Creating STEP 5 Synonyms and Comments	8-30
	Examining APT Code	8-31

8.1 Translating APT Objects to OSx Tags

NOTE: Unless otherwise specified, OSx is used throughout this chapter to refer to SIMATIC TISTAR Releases 1.x and 2.x in addition to SIMATIC PCS 3.x and OSx 4.x. S5 controllers are supported by PCS 3.x and OSx 4.x.

APT allows you to translate APT objects to OSx tags. The procedure consists of the following four steps.

- For Series 505 controllers, choose the OSx release (TISTAR 1.x and 2.x, PCS 3.x, or OSx 4.x). For S5 controllers, the default is PCS 4.x and programs must include RBE code. For both controller families, indicate whether to build a new translate table or to append to an existing translate table.
- Mark objects for OSx by selecting objects for translation to OSx.
- Compile the program. This step builds the executable code used by the controller and creates a file that contains tag data.
- Translate the tags. Translation places the tag data into a formatted OSx database.

Each of the four steps for translating tags is discussed in detail further in this chapter.

After objects are translated to OSx, they are stable; that is, they keep the same controller location whether the program is changed or recompiled. In order for the addresses to stay constant, you must set the **Build Translate Table** option in the Compiler Control File to **No** or **Append**. If you need to add tags and keep the old tags at the same address, use the **Append** option. This is necessary for OSx configuration, which requires the controller addresses of your marked objects to stay constant. Refer to [Section 8.4](#) for additional details on these Build Translate Table options.

NOTE: Objects that you mark for OSx cannot be deleted and fields that affect the address locations of the marked object cannot be changed, unless you first unmark the affected objects.

For specific information about tag translation, refer to [Sections 6.1](#) and [6.2](#) of this manual, and the chapter about APT and the OSx operator interface in the *SIMATIC APT Applications Manual*. For a detailed list of the fields that are translated for each object, see the appendix about OSx (and TISTAR) tag translation in the *SIMATIC APT Applications Manual*.

For S5 Users

In order to translate tag data to OSx, S5 users must select **RBE code included** in the Compiler Control File, and you must fill in the appropriate RBE information that appears after you make this selection ([Figure 8-1](#)). Refer to [Section 6.3](#) for more detailed information.

Number of RBE Points:	<u>250</u> 500
Number of CP143 cards in base unit:	<u>1</u> 2
CP/CPU Configuration:	<u>0</u> •
Number of connections on 1st CP143:	<u>1</u>
Number of connections on 2nd CP143:	<u>0</u>
Read time from CP143:	<u>0</u> •
SSNR for 1st CP143:	<u>0</u>
SSNR for 2nd CP143:	<u>4</u>

Figure 8-1 RBE Data for S5 Compiler Control File

If you are translating analog outputs, word inputs, or word outputs (%PW128-254, %OW0-254) to OSx, you must select **Image Register** for each point in the I/O Symbolic Name Table. Otherwise, these points are not translated.

8.2 Tag Translation — Choosing an OSx/PCS Release

Choosing an OSx/PCS Release

For Series 505 controllers, you can choose any one of four releases: TISTAR Release 1.x, TISTAR Release 2.x, PCS 3.x, and OSx 4.x. For S5 controllers, you must use PCS 4.x, the default. [Table 8-1](#) lists the controllers that are compatible with the current release of OSx.

Table 8-1 Controllers Compatible with OSx 4.x

Series 505 Controllers	S5 Controllers
545-1102 (Rel. 3.1)	S5-135U (CPU 928B)
545-1103 (Rel. 4.0)	S5-155U (CPU 948)
545-1104 (Rel. 4.0)	S5-155H (CPU 948R)
545-1105 (Rel. 5.0)	
545-1106 (Rel. 5.0)	
555-1101 (Rel. 3.1)	
555-1102 (Rel. 3.1)	
555-1103 (Rel. 4.0)	
555-1104 (Rel. 4.0)	
555-1105 (Rel. 5.0)	
555-1106 (Rel. 5.0)	
575-2103 (Rel. 3.1)	
575-2104 (Rel. 4.0)	
575-2105 (Rel. 5.0)	
575-2106 (Rel. 5.0)	

(Series 505)

In the Compiler Control File, enter the release (TISTAR 1.x or 2.x, PCS 3.x, or OSx 4.x) being used on the OSx release line.

TISTAR 1.x and 2.x create similar files for tag data; however, TISTAR Release 2.x can translate more types of tags. PCS 3.x and OSx 4.x have different formats for their files of tag data, namely, comma delimited. You can import and export the file to and from spreadsheet programs where you can review and edit it. For additional information about the format of the tag file, see the appendix on defining a variable's address in the [SIMATIC APT Applications Manual](#).

(S5)

In the Compiler Control File, use the default selection, PCS 4.x, on the PCS release line. PCS 4.x uses a comma-delimited file of tag data that you can import from/export to a spreadsheet program where you can review and edit it.

Appending or Building the Translate Table

In order to create a file of tag data, you build a translate table on the Compiler Control File by choosing **Yes** or **Append** for the **Build Translate Table** option in the Compiler Control File.

When you choose **Yes**, tag data is placed in the file named **install.tag**.

When you choose **Append**, APT creates two files: **append.tag**, which contains only the tags marked since the last program compile; and **install.tag**, which contains all marked tag data, including the data in **append.tag**.

[Section 8.4](#) provides more details about the differences between appending tag data and creating a new file of tag data.

8.3 Tag Translation — Marking Objects

APT provides four editors for marking objects for translation: the tag translate editor, the tag translate sub-editor, the tag attributes sub-editor, and the process groups sub-editor.

Use the tag translate editor to mark simple APT objects for translation to OSx. After you mark an object, APT provides default names for both the OSx name and the OSx description.

You also use the tag translate editor to mark complex objects such as CFBs, recipes, or declarations that have more than one associated extension. After marking a complex object, you can expand/edit the object in order to mark individual extensions of the object in the tag translate sub-editor.

Use the tag attributes sub-editor to specify scan status, upload, download, autolog, manual set, or 20% offset. APT provides default values, depending on the object type.

Use the process groups sub-editor to specify the OSx process groups with which a tag is associated. The default value assigns all tags to all 32 process groups. Unmark all unused process groups to avoid unintended memberships occurring in OSx between users, tags, alarm groups and action requests.

Using the Tag Translate Editor

Determine the level in the hierarchy at which you want to mark objects for translation: unit, CFC, or table (recipe, declare, device, or I/O). Then use the tag translate editors to select the individual objects to be translated. Follow the steps below:

1. Place the cursor on the hierarchy location (unit, CFC, table) and press K, or use the **OPTs** icon to select **Mark PCS Tags**. A screen similar to [Figure 8-2](#) appears.

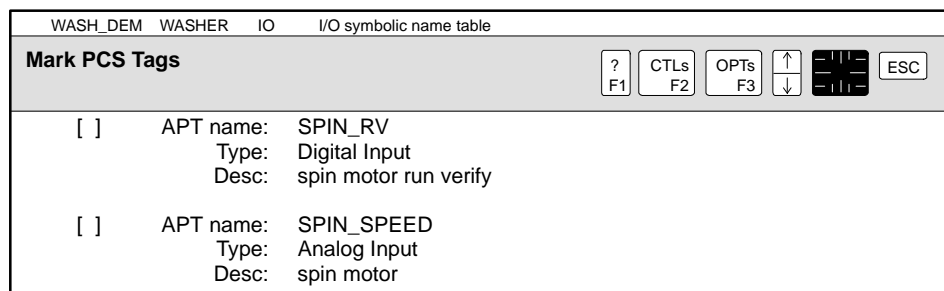


Figure 8-2 Example of Marking OSx Tags

- Place an **X** within the brackets to mark an object.

For a simple object, e.g., SPIN_RV, the additional fields PCS Name, Type, and Desc appear on the screen, as shown in [Figure 8-3](#).

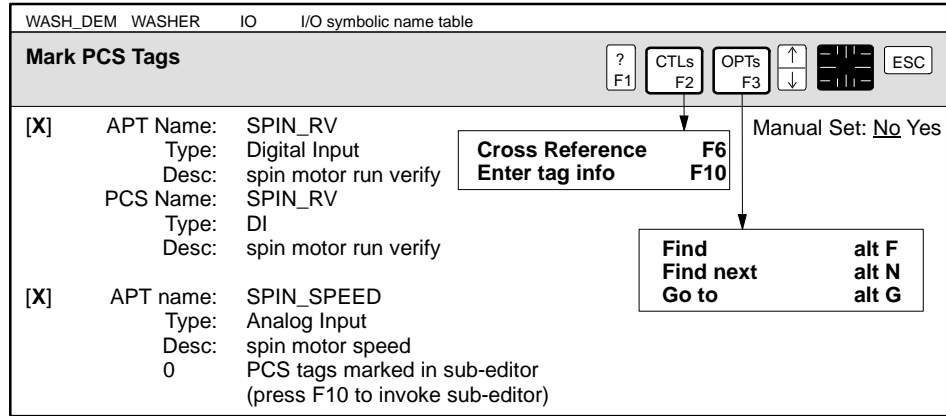


Figure 8-3 Example of Tag Translate Editor

- If you want the OSx name for the tag to be different from the APT name that appears as the default, enter the OSx name in the PCS Name field and press **Enter**. Do not use the characters , ; ! " \ in this field, as they are reserved characters in PCS and OSx.

NOTE: APT names can be longer than 12 characters but OSx names cannot. APT names longer than 12 characters are automatically truncated in OSx. To avoid the possibility of names in OSx that are not unique, rename any tags that have APT names long enough to be subject to truncation in OSx. See the OSx hints chapter in the [SIMATIC APT Applications Manual](#).

- If you want the OSx description to be different from the APT description that appears as the default, you can enter the OSx description in the Desc field and press **Enter**. Do not use the characters , ; ! " \ in this field, as they are reserved characters in OSx.
- If an object has a low/high range or a scan deadband, enter the appropriate values and press **Enter**.
- To enable an OSx operator to enter trend data manually for a tag, select **Yes** in the Manual Set field.
- To exit the editor, press **Esc**.

Tag Translation — Marking Objects (continued)

Using the Tag Translate Sub-Editor

For a more complex object, e.g., SPIN_SPEED in [Figure 8-3](#), APT indicates that no tags are currently marked in the tag translate sub-editor.

1. To mark extensions for a complex object, press **[F10]** from the tag translate editor.
2. Place an **X** within the brackets to mark the extensions for translation. [Figure 8-4](#) shows the extensions for SPIN_SPEED. Notice that the PCS name for SPIN_SPEED.RAW has been truncated to SPIN_SPEED.R.
3. Follow steps 3–6 ([page 8-7](#)) for marking a simple object.
4. To exit the sub-editor, press **[Esc]**.

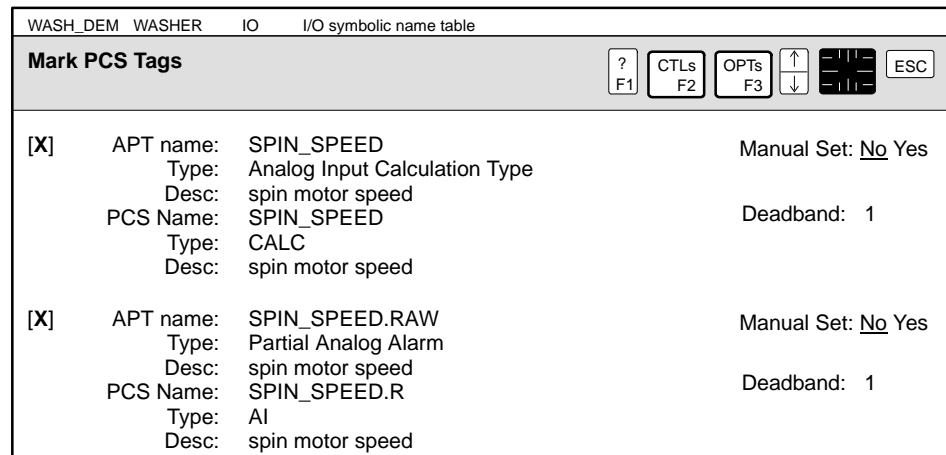


Figure 8-4 Example of Tag Translate Sub-Editor

Using the Attributes Sub-Editor

Use the attributes sub-editor to mark options for OSx attributes; both simple and complex objects have options. Follow these steps to use the Attributes Sub-Editor.

1. To access the Attributes Sub-Editor, place the cursor on the object (one of the extensions, if a complex object) and press **[F10]**. The attribute fields are displayed, as shown for SPIN_SPEED in [Figure 8-5](#).
2. Place an **X** within the brackets for the appropriate fields and press **[Enter]**.
 For Series 505 and S5 controllers with a PCS 3.x or OSx 4.x operator interface, the attributes are scan, upload, autolog, and 20%.
 For Series 505 controllers with a TISTAR 1.x or 2.x operator interface, the attributes are scan, upload, download, and 20%.
3. To exit the sub-editor, press **[Esc]**.

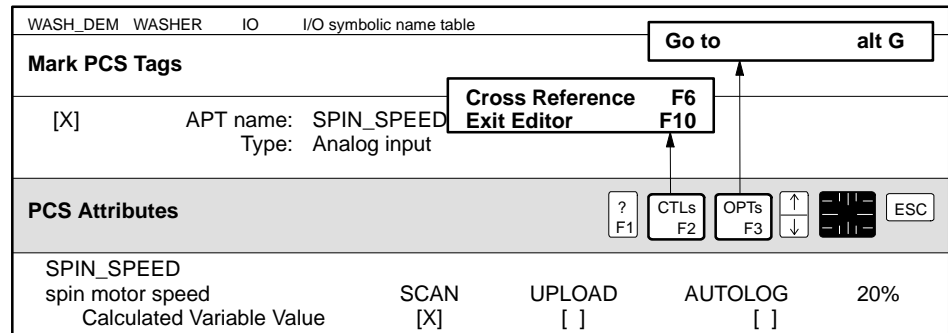


Figure 8-5 Example of Tag Attributes Sub-Editor

Tag Translation — Marking Objects (continued)

Using the Process Groups Sub-Editor

Use the process groups sub-editor to specify with which OSx process groups a tag is associated. Both simple and complex objects can belong to process groups. The default value assigns all tags to all 32 process groups. Follow the steps below.

1. To access the process groups sub-editor, place the cursor on the object (or on the extension, if a complex object) and press ALT S . [Figure 8-6](#) shows the process groups for SPIN_SPEED.
2. Enter the names of the OSx process groups with which the tag is to be associated. Place an **X** within the brackets for the appropriate process group and press Enter .

All process groups are initially selected [X]. If you do not want a process group, make a blank within the brackets by pressing the space bar []. Always unmark [] unused process groups so that unintended memberships in OSx do not occur between users, tags, alarm groups and action requests.

3. Select **OKAY** in the lower right hand corner to save your process group selections and to exit the sub-editor. Do not press Esc to exit the process group, unless you do not want to save your changes.

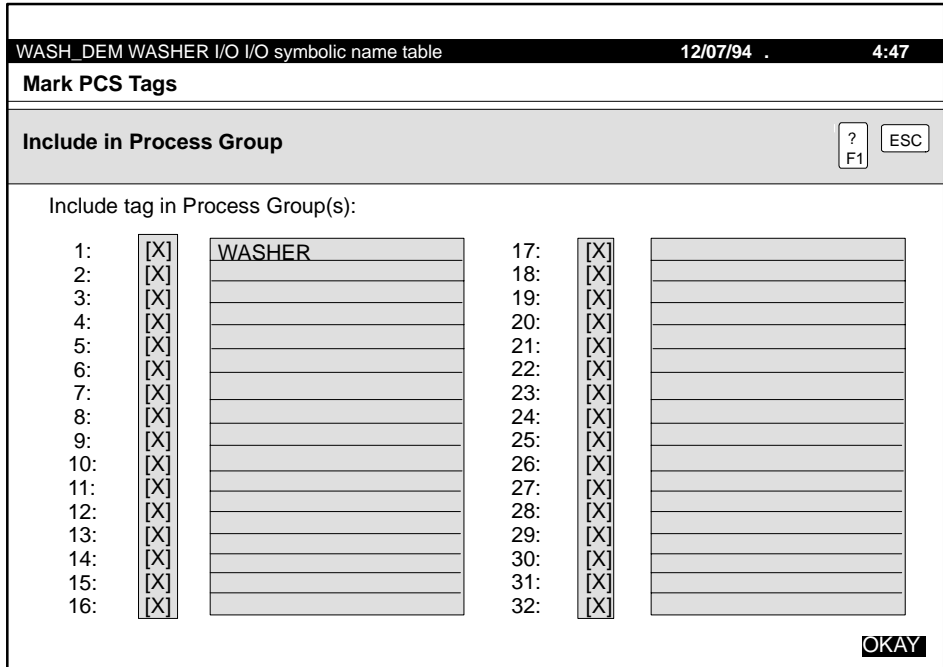


Figure 8-6 Menu for Process Groups

Unmarking an Object

Objects that you mark for OSx cannot be deleted, and fields that change the address location of the object cannot be changed, unless you first unmark the affected objects. In order to unmark an object, place a blank within the brackets by pressing the space bar.

8.4 Tag Translation — Compiling the Program

Compile Options

When you compile the program, APT builds the executable code and fixes controller addresses for objects. It is not essential to mark all objects before you compile a program because you can append tag information to the tag data file at any time. However, decide which objects are required by the operator interface and mark them all before compiling. You can use the APT **Compile** option for any type of operator interface.

The first time you compile a program after marking objects, select these options in the Compiler Control File:

```
Force compile? [X]
Build translate table? No Append Yes
Units to compile -----
Include all? [X] 1 of 1 included
```

Compiling a Program

Keep these points in mind when you fill out the Compiler Control File.

Force Compile Use a force compile if you are compiling a program for the first time. After a successful compile, subsequent compiles, without a forced compile, execute more quickly after minor code changes because some sections may not be revalidated. Refer to [Chapter 6](#) for a complete discussion about the program compile operation.

Build Translate Table The translate table reallocates all OSx tags for translation to OSx. You have three options:

- When you select **No**, existing marked tags are not modified and Phase 7 of the compile is not executed.
- When you select **Yes**, all marked tags are placed in the **install.tag** file that is sent to OSx.
- When you select **Append**, all 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**.

Select **Yes** for the Build Translate Table option when you build the translate table for the first time. Selecting **Yes** for the Build Translate Table option fixes the controller addresses for objects. If you have already compiled the program using the **Yes** option, addresses for the existing objects could change. Memory locations are blocked optimally for the most efficient accessing. Refer to the chapter on APT and the OSx operator interface and the appendix on defining a variable's address in the [SIMATIC APT Applications Manual](#) for more information.

Select **No** for Build Translate Table option when you do not want addresses for marked objects to change after the initial program compile. Tags that were added or removed are not updated in the compile. Phase 7, Translate Build, does not execute. Typically, you do this during subsequent program compilations. Exceptions are listed in the note at the bottom of this page. If you need to include additional tag data, see Adding Tags, below.

Adding Tags Select **Append** for the Build Translate Table when you recompile a program, if you want to add tags without affecting the addresses of previously marked objects. The **Append** selection adds new object addresses to the file **append.tag**, which contains tag data for appended tags, but does not change the addresses of previously marked objects.

If you select **Append**, the tags that you have added are appended to existing files containing previously translated tags. Memory locations for appended information are blocked optimally. However, because they are not integrated into existing blocks, memory allocation for all tags, (old and appended tags) is not blocked optimally. If you have deleted tags, the memory locations that were assigned to them are not available and cannot be used until you select the **Yes** option for the build translate table.

If you unmark a tag to change a parameter and then re-mark it, APT treats the re-marked tag like a new tag.

When you select **Append** (and deselect the **Compile** option for one or more units that had been previously selected for compiling), no addresses are moved. However, memory is allocated inefficiently and has unusable gaps.

If you select **Append** and later do a forced compile, large gaps in memory may occur in your controller memory allocation. The RLL or STL memory requirements may also increase. To repack the files efficiently, select the Build Translate Table with the **Yes** option.

If you want to recompile the whole program after you have appended tag data, select **No** in the Build Translate Table, and no object addresses are changed.

NOTE: The **No** and **Append** choices for the Build Translate Table in the Compiler Control File do not change addresses marked for translate unless you do any of the following.

Change the controller in the Compiler Control File.

Add RCC cards to a 560/565 system.

Change the amount of reserved memory.

Re-mark a previously translated unit for compile.

Mark Units The **Units to Compile, Include All** option must remain selected if you do not want the address of previously marked objects to change.

Handling Complex Tags When you mark a complex object, such as a valve or a motor, only a subset of the object extension addresses is fixed. A detailed list of the extensions fixed by the compile is given in the appendix of the *SIMATIC APT Applications Manual*.

You may require an extension to have a fixed address, but APT does not fix the address for this particular extension. Examples of this are the control relays created in the PID and Analog Alarm blocks. If you want to use these objects in your operator interface, you can follow the procedure listed below.

Recipes In certain instances when you do a force compile, or when an assembly force compile occurs, APT cannot recalculate the addresses of Boolean elements in recipes that contain Boolean arrays. When this occurs, the assembler generates an error (12058) during program compilation. You must then recompile your program with Translate set to **Yes** and re-install your tags to the operator station.

The extension is a simple boolean status report read-only variable.

1. Assign the value of the PID extension to a declared variable in a CFB math block.
2. Mark the declared variable in order to fix its address.
3. Use the fixed address for your operator interface.

For example, if you want to display the most significant bits of the C-Flag for PID Loop BL_TC on your operator interface, follow the steps below.

1. Use the Declarations editor to declare a real variable to hold the status, e.g., BL_TC_CFH.
2. In a CFB math block, assign the value of **BL_TC.CFH** to the declared variable as follows: `BL_TC_CFH := BL_TC.CFH`
3. Use the **Mark Tags** option to mark **BL_TC_CFH** to fix the address.
4. Use the fixed address for your operator interface.

8.5 Tag Translation — Selecting the Translate Utilities

Translation is the last function to do after compiling and downloading an APT program. This option takes all the marked objects and places them with the appropriate format in the OSx/PCS or TISTAR database. You can use the following procedures to translate your objects into the OSx or TISTAR format.

- For Series 505 or S5: If you are compiling your APT program on an OSx engineering node, you can select the **Translate** option to convert tags into the OSx/PCS format, as described in [Section 8.6](#).
- For Series 505 or S5: If you are translating tags for PCS 3.x or OSx 4.x and there is no communications link with OSx, follow the procedure described in [Section 8.7](#).
- For Series 505 only: If you are compiling your APT program on a TISTAR operator station, you can select the **Translate** option while executing APT on the operator station, as described in [Section 8.8](#).
- For Series 505 only: If you are compiling your APT program on a workstation other than a TISTAR operator station, and you want to use the **Translate** option while executing APT on your TISTAR operator station, follow the procedure described in [Section 8.9](#).
- For Series 505 only: If you are compiling your APT program on a workstation other than a TISTAR operator station, and you want to translate tags without executing APT from your TISTAR operator station, follow the procedure described in [Section 8.10](#).
- For Series 505 only: If you are translating tags for use with a TISTAR Model 20, and there is no communication link with the Model 20 system, follow the procedure described in [Section 8.11](#).

(Series 505 and S5)

For PCS 3.x and OSx 4.x you can transfer translation data across the Industrial Ethernet network using the CP1413 or CP1613 card. For OSx 4.x, PCS 3.x, and TISTAR 1.x and 2.x, you can also transfer translation data to a diskette and then install the data from the diskette.

(Series 505 only)

For all TISTAR models, you can transfer translation data using the TISTAR Communications Card (PFS-3619). This card is RS-232/422 compatible and communicates only with a TISTAR system.

8.6 Translation on an OSx Engineering Node

(Series 505 and S5)

If you are executing APT on an OSx engineering node, you can select the **Translate** option to translate tags into the OSx format. Your system must be in communication with OSx through the Industrial Ethernet network. To translate APT tags to OSx, follow the steps below.

1. In the Program Content, place the cursor on:
OBJECT () Successfully compiled program
2. Press **L** or use the **OPTs** icon to select **Translate**. [Figure 8-7](#) shows an example dialog box.

TRANSLATE [?] [ESC]

PCS node: PCS_CNFG
Controller name: REACTOR1
Controller type: TI555

Append tags only: []
Delete unused tags on secondary: []

Translate creates
Report name: TRANSLT.RPT
Description: PCS tag translation report.

Report options
Print: []
Save: [X]

Begin translate? **OKAY**

Figure 8-7 Translating Tags to OSx

3. If you want to translate only the appended tags, place an **X** between the brackets of the **Append tags only** field. If you want to translate all tags, leave the field blank.
4. Place an **X** between the brackets to select **Delete unused tags on secondary** to delete any unused tags in the OSx database. (This option is ignored if you are appending tags.) If you unmark a tag in APT, it is not included in the next INSTALL.TAG file, but it remains in the OSx database unless you choose this option.
5. The translation information is transferred to OSx, where it is validated, and the translate report is sent back to APT.

The report generated from OSx is called **translat.rpt** and is saved under Saved Reports in the program content. From DOS, you can access the report under the `\apt\program\program_name\PRR` subdirectory, where **program_name** is the name of your APT program.

The **PCS_NODE** name cannot be changed from this dialog. If you wish to translate to other than the default OSx station, you must set the environment variable **APT_PCS_NODE**. See [Table A-2](#) for more information.

8.7 Translation from an APT Workstation to an OSx System

(Series 505 and S5)

If you have compiled your application on a system that is not in communication with OSx, and need to translate tags to OSx, follow the steps below:

1. At the Program Content level, place the cursor on:
OBJECT () Successfully compiled program
2. Press L or use the **OPTs** icon to select **Translate**. [Figure 8-8](#) shows an example dialog box.

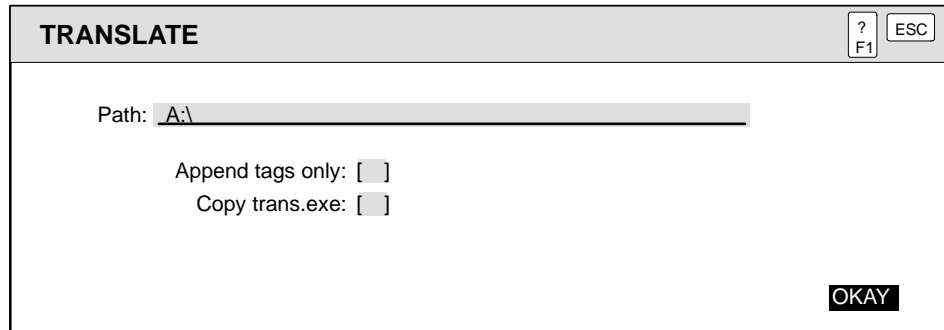


Figure 8-8 Offline Tag Translation to OSx

3. At the **Path:** prompt, type the path to your diskette drive. For example, if your disk drive is A:, type **A:**.
4. If you want to translate only the appended tags, place an **X** between the brackets of the **Append tags only** field. If you want to translate all tags, leave the field blank.
5. If you need the **trans.exe** file on the system you are translating to, place an **X** between the brackets of the **Copy trans. exe** field.
6. Place the storage diskette into the disk drive that you specified above, and press Enter.

In addition to copying files, this option informs your non-APT/OSx system that a tag installation has occurred. This removes compiler warnings until tag information changes again.

7. Insert the diskette in the OSx unit. Select **Install Tags** from the **Data** option of the menu bar in OSx. More information about tag translation can be found in the chapter on configuring tags of the *SIMATIC PCS 7 OSx Process Configuration Manual*.

8.8 Translation on a TISTAR Operator Station with APT

(Series 505 only)

If you are executing APT on the TISTAR operator station, you can select the Translate option to convert tags into the TISTAR format. You must be executing APT from the TISTAR Programming Utilities Menu.

To translate APT tags to TISTAR, follow these steps.

1. Set the TISTAR system to the OFLN state.
2. In the Program Content, place the cursor on:
OBJECT [] Successfully compiled program
3. Press , or use the **OPTs** icon to select **Translate**. [Figure 8-9](#) shows an example dialog box.

TRANSLATE ? ESC

Controller name: BLDG5

Controller type: TI565

Append tags only:

Translate creates

Report name: TRANSLT.RPT

Description: PCS tag translation report.

Report options

Print:

Save: [X]

Begin translate? **OKAY**

Figure 8-9 Translating Tags to TISTAR

4. Place an **X** between the brackets to select **Append tags only** to translate only the appended tags. Leave the field blank to translate all tags.
5. Press again to complete the action.
6. The TISTAR tag information is transferred to TISTAR, where it is validated, and the translate report is sent back to APT.

The report generated from TISTAR is called **translat.rpt** and is saved under Saved Reports in Program Content. From DOS, you can access the report under the **\apt\program\program_name\pr** subdirectory, where **program_name** is the name of your APT program.

In the TISTAR UNIX environment, the report is stored in the **/usr/tistar/data/tmp** subdirectory. The report is called **apt_rpt.i-4** if the installation was made from station 4. If the installation was made from station 3, the report is called **apt_rpt.i-3**.

8.9 Translation from APT Workstation to TISTAR Operator Station with APT

(Series 505 only)

If you have compiled your application program on an APT workstation, and need to translate tags by selecting the **Translate** option while executing APT on your TISTAR operator station, follow the steps below.

1. Restore or compile any APT program on the TISTAR operator station. You can save time by using a small program. The purpose is to ensure that the following line appears in the Program Content.

OBJECT [] Successfully compiled program

2. Press **L** or use the **OPTs** icon to select **Translate**. [Figure 8-10](#) shows an example dialog box.

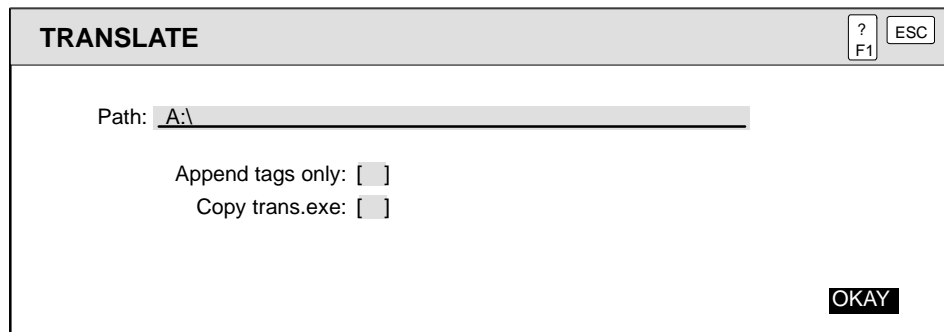


Figure 8-10 Offline Tag Translation to TISTAR

3. At the Path: prompt, type the path to your diskette drive. For example, if your disk drive is A:, type the following: A:\
4. If you want to translate only the appended tags, place an **X** between the brackets of the **Append tags only** field. If you want to translate all tags, leave the field blank.
5. If you need the **trans.exe** file on the system you are translating to, place an **X** between the brackets of the **Copy trans. exe** field.
6. Place the storage diskette into the disk drive that you specified above, and press **OKAY**.

-
7. Place the diskette in the TISTAR operator station and use the DOS command
Copy A:\install.tag C:\apt\program*small_program*>
to copy the file **install.tag** to the operator station.

If you have appended tags (selected the **Append** option on the Compiler Control File) substitute `append.tag` for `install.tag`.

small_program is the name of the program that you restored or compiled in Step 1.

8. Enter **IGT** and press to return to TISTAR. Place the TISTAR system in the OFFLINE state.
9. Execute APT from the Programming Utilities menu of TISTAR.
10. Place the cursor on the OBJECT box of the APT program *small_program* that you restored or compiled in step 1. Press , or use the **OPTs** icon to select **Translate**.
11. Enter the appropriate information into the dialog window using the TISTAR secondary name in the Controller name prompt. An example of a dialog window is shown on [page 8-19](#).
12. Press again to complete the action.
13. The TISTAR tag information is transferred to TISTAR, where it is validated, and the translate report is sent back to APT.

The report generated from TISTAR is called **translat.rpt** and is saved under Saved Reports in Program Content. From DOS, you can access the report under the `\apt\program\small_program\pr` subdirectory, where *<small_program>* is the name of the program that you restored or compiled in step 1.

In the TISTAR UNIX environment, the report is stored in the `/usr/tistar/data/tmp` subdirectory. The report is called **apt_rpt.i-4** if the installation was made from station 4. If the installation was made from station 3, the report is called **apt_rpt.i-3**.

8.10 Translation from APT Workstation to TISTAR Station without APT

(Series 505 only)

The following procedure allows you to translate tags without executing APT. You can compile your application program on a workstation other than a TISTAR operator station and then translate tags without executing APT from a TISTAR operator station. This procedure is faster than those described on [Sections 8.8](#) and [8.9](#). The APT software does not have to be on the operator station, and you do not need an APT key for the operator station.

You will be prompted for a valid user ID and password defined in the security configuration of the TISTAR DEU Utilities Menu.

Placing File *TRANS.EXE* on the Operator Station

This procedure uses a program called **trans.exe**, which must exist on the operator station. To place **trans.exe** on the operator station, follow the steps below.

1. In the Program Content, place the cursor on:
OBJECT () Successfully compiled program
2. Press **L** or use the **OPTs** icon to select **Translate**. [Figure 8-11](#) shows an example dialog box.

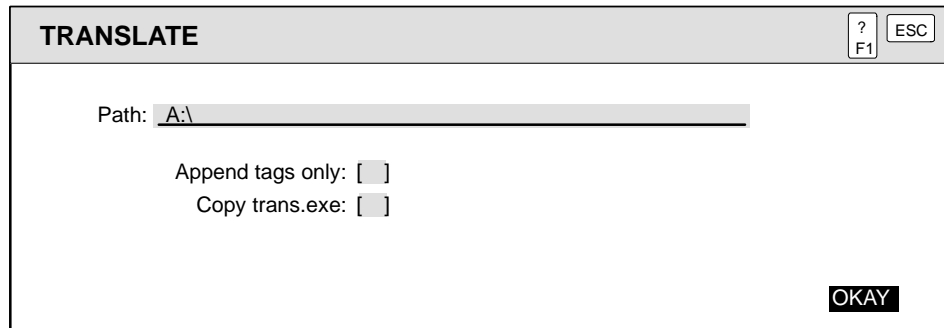


Figure 8-11 Offline Tag Translation to TISTAR without APT

3. At the **Path:** prompt, type the path to your diskette drive. For example, if your disk drive is A:, type the following: A:\
4. If you want to translate only the appended tags, place an **X** between the brackets of the **Append tags only** field. If you want to translate all tags, leave the field blank.

-
5. If you need the **trans.exe** file on the system you are translating to, place an **X** between the brackets of the **Copy trans. exe** field.
 6. Place the storage diskette into the disk drive that you specified above, and press .

In addition to copying files, this option informs your non-TISTAR workstation that a tag installation has occurred. This removes compiler warnings until tag information changes again.

7. Place the diskette into the operator station and use the DOS command **Copy A:\trans.exe C:** to copy the file **trans.exe** to the operator station.

Translation from APT Workstation to TISTAR Station w/o APT (continued)

Placing File *INSTALL.TAG* or *APPEND.TAG* on the Operator Station

The APT file **install.tag** or **append.tag** must also exist on the operator station. To place either file on the operator station, follow the steps below.

1. From the operator station, place the TISTAR system in the OFFLINE state and then enter the DOS environment.

2. From the operator station, use the DOS command

```
MD \apt\program\loadtags
```

to create a subdirectory on the operator station. You can substitute any name for `loadtags`.

3. From the operator station, use the DOS command

```
MD \apt\program\loadtags\pr
```

to create a second subdirectory on the operator station.

4. Place the diskette into the diskette drive of the operator station and use the DOS command

```
copy A:\install.tag C:\apt\program\loadtags
```

to copy **install.tag** to the operator station.

If you have appended tags (selected the **Append** option on the Compiler Control File) substitute `append.tag` for `install.tag`.

Executing Translation

Begin the translation process by executing the **trans.exe** program. At the DOS prompt, type `trans loadtags <secname>` and press **Enter**.

For **secname** substitute the name of the TISTAR secondary.

If you have appended tags, use `trans loadtags -a <secname>` to translate the appended tags.

You will be prompted for a valid user ID and password.

The report generated from TISTAR is called **translat.rpt** and is saved under Saved Reports in Program Content. From the DOS environment you can access the report under the `\apt\program\loadtags\pr` subdirectory.

In the TISTAR UNIX environment, the report is stored in the `/usr/tistar/data/tmp` subdirectory. The report is called **apt_rpt.i-4** if the installation was made from station 4. If the installation was made from station 3, the report is called **apt_rpt.i-3**.

8.11 Translation from APT Workstation to TISTAR Model 20 System

(Series 505 only)

The following procedure allows you to translate tags developed on your APT DOS workstation for a TISTAR Model 20 system. In this procedure, you transfer the APT tag file **install.tag** to the Model 20 from a diskette and then execute a translation program that does the actual tag translation.

NOTE: Your Model 20 must be configured to read either drive A or drive B as a DOS drive. A diskette drive that is configured as a high-density drive cannot access a low-density diskette. You can run the UNIX utility **config_fd** to verify that a drive is configured correctly. Refer to your TISTAR documentation for more information.

Transferring *INSTALL.TAG* to a Diskette

Follow these steps to transfer the APT tag file **install.tag** to a diskette.

1. In the Program Content, place the cursor on:
OBJECT () Successfully compiled program
2. Press **[L]** or use the **OPTs** icon to select **Translate**. [Figure 8-12](#) shows an example dialog box.

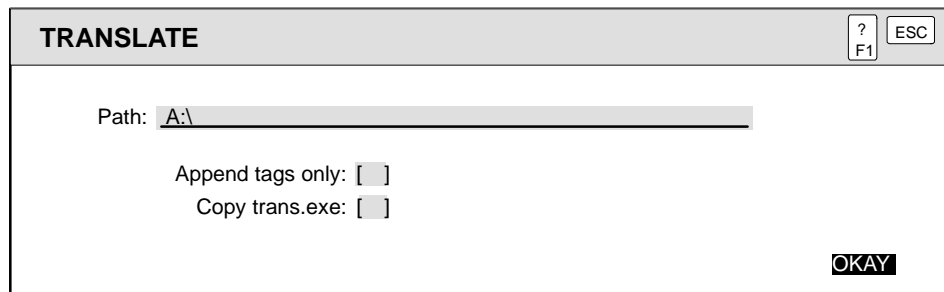


Figure 8-12 Offline Tag Translation to a TISTAR Model 20 System

3. At the Path: prompt, type the path to your diskette drive. For example, if your disk drive is A:, type the following: A:\
Append tags only:
4. If you want to translate only the appended tags, place an **X** between the brackets of the **Append tags only** field. If you want to translate all tags, leave the field blank.
5. If you need the **trans.exe** file on the system you are translating to, place an **X** between the brackets of the **Copy trans.exe** field.
6. Place the storage diskette into the disk drive that you specified above, and press **[OKAY]**.

If you have appended tags, rename the **append.tag** file on the floppy disk to **install.tag**. For example, if the floppy disk is drive A:, the DOS command is `RENAME A:\append.tag A:\install.tag`.

Translation from APT Workstation to TISTAR Model 20 System (continued)

Translating the Tags

Follow these steps to translate the APT tags for TISTAR.

1. Set TISTAR to the OFLN state.
2. Select the **DEU Terminal** mode.
3. Log in using the TISTAR User ID.
4. Type `trans` and press `[Enter]`. Be sure to use lowercase letters.

TISTAR prompts you with the following options.

- 1) Floppy diskette from drive A (DOS format)**
- 2) Floppy diskette from drive B (DOS format)**
- 3) Streaming tape (cpio format)**

You can select option **3** and use a tape to transfer the `install.tag` file from one TISTAR system to another. To display UNIX help for the `cpio` command, type `man cpio` at the UNIX command line.

5. Type **1**, **2**, or **3** to select the appropriate option, or **q** to quit. Then press `[Enter]`.

TISTAR prompts you for a **secondary name**.

6. Enter the name of the secondary and press `[Enter]`. This entry is not optional.

TISTAR prompts you for a **program name**.

7. Enter the name of the program and press `[Enter]`. Since the program name is used only in the APT report that is generated by the translation, this entry is optional; you can press `[Enter]`.

TISTAR prompts you for a **program descriptor**.

-
8. Enter a description for the program. The program description is used only in the APT report that is generated by the translation. Therefore, this entry is optional; you can press `Enter`.

TISTAR prompts you to place the diskette containing the APT tag file **install.tag** into the diskette drive.

9. Put the diskette into drive A: or drive B: of the Model 20 system. Remember that the **install.tag** file must be located in the root directory of the diskette. The UNIX **trans** utility does not search subdirectories.

When the translation is complete, TISTAR displays a message indicating that the process was successful. However, you need to check the APT report to verify that there was not a failure during the translation. The report is called **apt_rpt.i-1** and is located under the **/usr/tistar/data/tmp** subdirectory.

10. After checking the APT translation report, type `exit` and press `Enter` to return to TISTAR.

8.12 Using TISOFT with APT

Creating TISOFT Synonyms and Comments

From APT, Series 505 users can generate the synonyms and comments that TISOFT uses to annotate the RLL. APT creates the synonyms from the I/O symbolic names, and the comments label the general areas of the Relay Ladder Logic.

To generate synonyms and comments to be used with TISOFT Release 4.x or greater on your APT computer, follow the steps below. For using TISOFT on an OSx station, follow the procedure in [Section 8.6](#).

1. From APT, compile the program as described in [Section 6.2](#).
2. Select the Report option by placing the cursor on **OBJECT [] Successfully compiled program**
3. Press , or use the **OPTs** icon to select **Report**.
4. Place an **X** between the brackets (as shown in [Figure 8-13](#)).
5. Press .
6. Exit the Reports option.

For the APT and TISOFT programs to be compatible, they need to access the same communications port. The default port for TISOFT is COM1. If a different port is configured for APT, you need to edit the **T1505.bat** batch file used to execute TISOFT. The **T1505.bat** file is located in your TISOFT subdirectory. Using any ASCII editor, modify the line that says `pc5rt p1 T01 1`, following the guidelines below.

If port 2 is configured, edit the line to say: `pc5rt p2 T01 1`

If you are using a CVU, edit the line to say: `pc5rt CVU T01 1`

If you are using the Industrial Ethernet network (available only with TISOFT 5.x and greater), edit the line to say: `pc5rt H1 T01 1`

Figure 8-13 gives an example of the Reports screen where you select the TISOFT Comments and Synonyms screen.

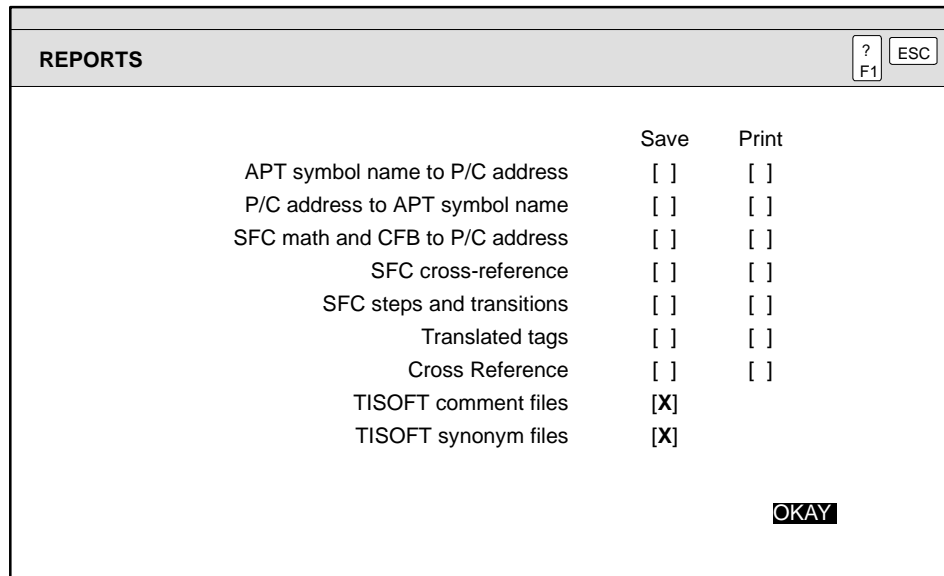


Figure 8-13 Creating TISOFT Comments and Synonyms

Examining APT Code

You can use TISOFT to examine the object code generated by APT. To view the code generated by APT that is resident in the controller, follow the steps below.

1. In the Program Directory, place the cursor on the program name; or, in the Program Content of a program, place the cursor on either:

CONTROL [] Configuration for compile
or
OBJECT [] Successfully compiled program

2. Press , or use the **OPTs** icon to select **TISOFT**.

If TISOFT is installed and the DOS path includes the TISOFT files, then the system exits from APT and enters the TISOFT environment. You can examine the RLL comments and synonyms from either the Online or the Offline mode.

8.13 Using STEP 5 With APT

Creating STEP 5 Synonyms and Comments

From APT, S5 users can generate the synonyms and comments that are used by STEP 5 to annotate the STL. APT creates the synonyms from the I/O symbolic names, and the comments label the general areas of the STL.

To generate synonyms and comments to be used with STEP 5, follow the steps below.

1. From APT, compile the program as described in [Section 6.2](#).
2. Select the Report option by placing the cursor on **OBJECT [] Successfully compiled program**
3. Press , or use the **OPTs** icon to select **Report**.
4. Place an **X** between the brackets (as shown in [Figure 8-14](#)).
5. Press .
6. Exit the Reports option.

The STEP 5 symbol file is stored in the `apt\program\program_name\pr` subdirectory. The file name is the first six characters of the APT program name, followed by `z0.seq`. For instance, if your APT program is `test1`, then the name of the symbol file is `test1@z0.seq`. To use this symbol file in STEP 5, copy it from APT to the STEP 5 working directory, and then invoke it from the assignment list. From the Project Settings screen, set the **Symbols** option to **Yes**, set your symbol length to 24, and set your comment length to 30 for compatibility with APT.

For the APT and STEP 5 programs to be compatible, they need to access the same communications port.

Figure 8-14 gives an example of the Reports screen where you select the STEP 5 Comments and Synonyms screen.

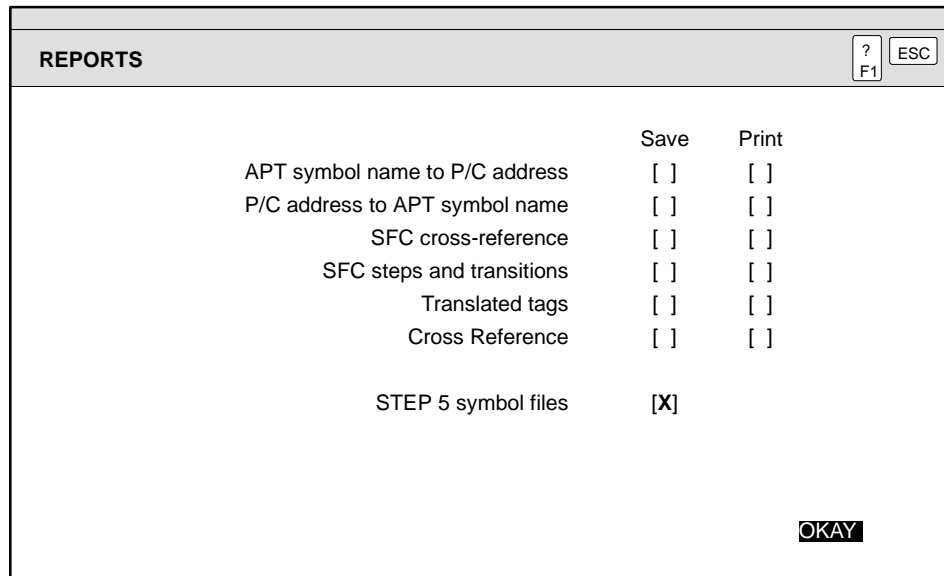


Figure 8-14 Creating STEP 5 Comments and Synonyms

Examining APT Code

You can use STEP 5 to examine the object code generated by APT. To view the code generated by APT that is resident in the controller, follow the steps below.

1. In the Program Directory, place the cursor on the program name; or, in the Program Content of a program, place the cursor on either:

CONTROL [] Configuration for compile

or

OBJECT [] Successfully compiled program

2. Press , or use the **OPTs** icon to select **STEP 5**.

If STEP 5 is installed and the DOS path includes the STEP 5 files, then the system exits from APT and enters the STEP 5 environment. You can examine the STL comments and synonyms from either the Online or the Offline mode.

Using PROFIBUS with APT

9.1	Overview	9-2
9.2	Configuring the PROFIBUS Network (Version 3.3)	9-4
9.3	Exporting the Binary File to APT (COM PROFIBUS V 3.3)	9-12
9.4	Configuring the PROFIBUS Network (Version 5.0)	9-14
9.5	Exporting the Binary File to APT (COM PROFIBUS V 5.0)	9-16
9.6	Adding PROFIBUS Modules in APT	9-17
9.7	Creating a PROFIBUS Configuration File in APT	9-20
	Guidelines for PROFIBUS.CFG Report	9-22
9.8	Debugging PROFIBUS in APT	9-23
	Setting the Slave Mode	9-23
	Setting the Operational Mode	9-26
	Setting the PROFIBUS-DP Synchronization Mode	9-28
	Reading PROFIBUS-DP I/O	9-30

PROFIBUS-DP enables you to integrate a wider variety of automation products and provides greater speed of communications than ever before. If you use PROFIBUS devices on your network, you can configure the 505 controllers that support PROFIBUS in APT. You can configure up to 112 slaves, including controllers such as the ET 200.

PROFIBUS is supported by the following controllers:

- 545-1104 and later (with PROFIBUS-DP Annex Card, Order No. PPX: 505-CP5434-DP)
- 555-1104 and later
- 575-2104 and later (with PROFIBUS-DP Annex Card, Order No. PPX: 505-CP5434-DP)
- 545-1103 ((with PROFIBUS-DP Annex Card, Order No. PPX: 505-CP5434-DP); limited to 16 slaves (Channel 1/Base 0, 16 slots)

For all PROFIBUS devices to be used within APT, you must also perform the following tasks:

- Configure the PROFIBUS-DP network using COM PROFIBUS V 3.3 (Section 9.2) or using COM PROFIBUS V 5.0 (Section 9.4).
- Export a binary file from COM PROFIBUS V 3.3 (Section 9.3) or from COM PROFIBUS V 5.0 (Section 9.5) to the APT program .prg directory.
- Place the modules in the module table with APT (Section 9.6).
- Create a PROFIBUS.CFG report within APT (Section 9.7).
- Compile and download the APT program (Chapter 6).

NOTE: When you download an APT program, the **Download I/O Configuration** option in the Download dialog box applies to PROFIBUS-DP as well.

9.2 Configuring the PROFIBUS Network (Version 3.3)

In order to use PROFIBUS with APT, you must first configure the PROFIBUS network. See the *SIMATIC ET 200 Distributed I/O System Manual* for instructions on how to install and use COM PROFIBUS, and then follow the procedures in this section.

Figure 9-1 shows an example of a PROFIBUS network.

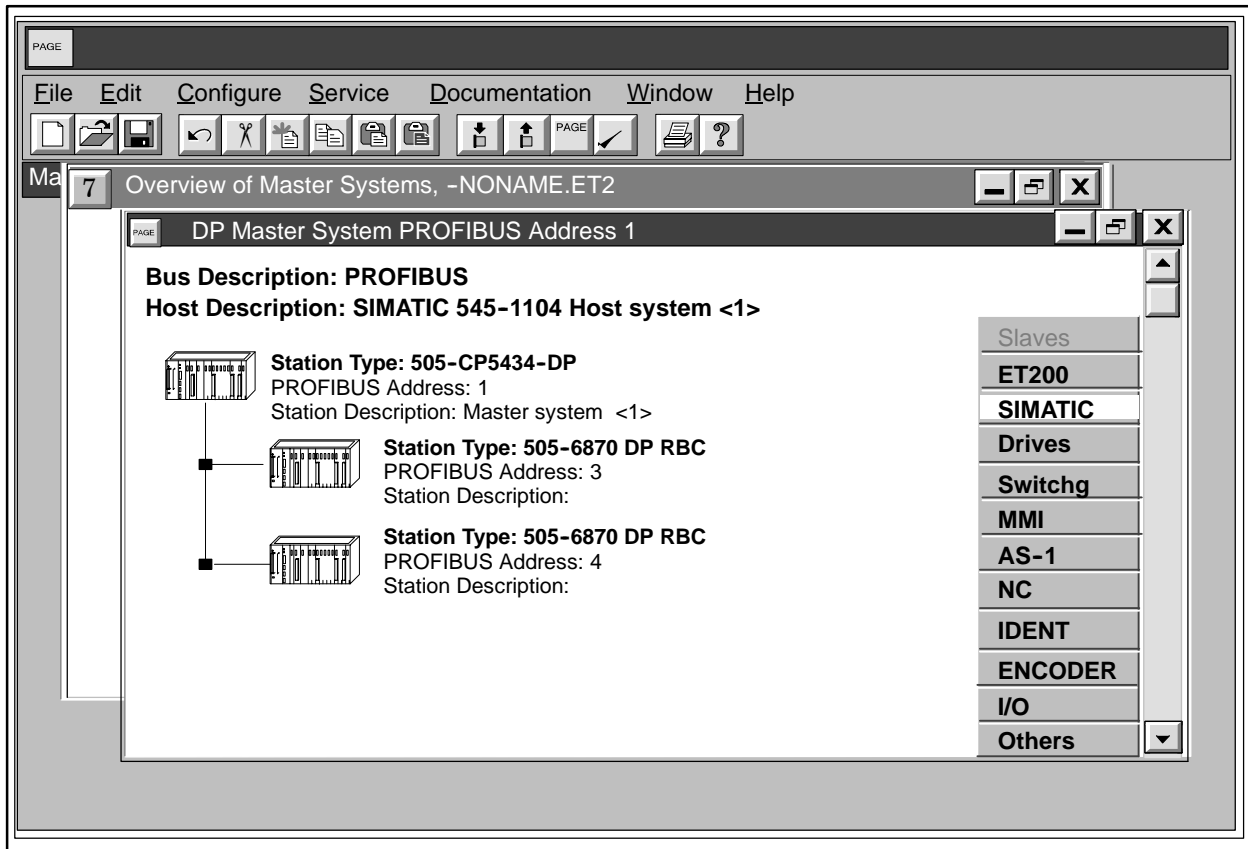


Figure 9-1 Sample PROFIBUS Network

To configure the PROFIBUS network, follow these steps:

1. From the Windows environment, double-click on the COM PROFIBUS icon. COM PROFIBUS opens.
2. Select **File->New**. The new Master System dialog box appears (Figure 9-2).
3. Select Master Station Type **505-CP5434-DP** and the Host Station Type that corresponds to the controller that you want to use as the master on the PROFIBUS network.

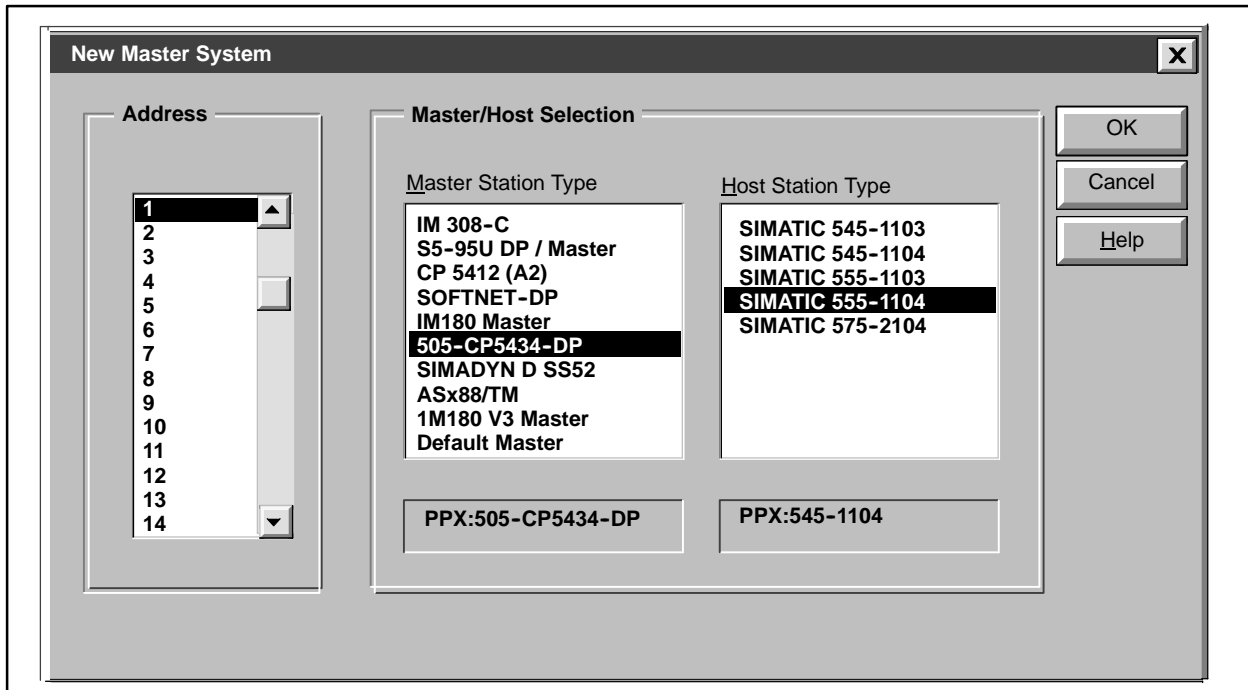


Figure 9-2 New Master System Dialog Box

Configuring the PROFIBUS Network (Version 3.3) (continued)

4. Select **OK** to confirm. The DP Master System PROFIBUS Address 1 window is displayed, showing the master system that you have selected and a list of slave types ([Figure 9-3](#)).

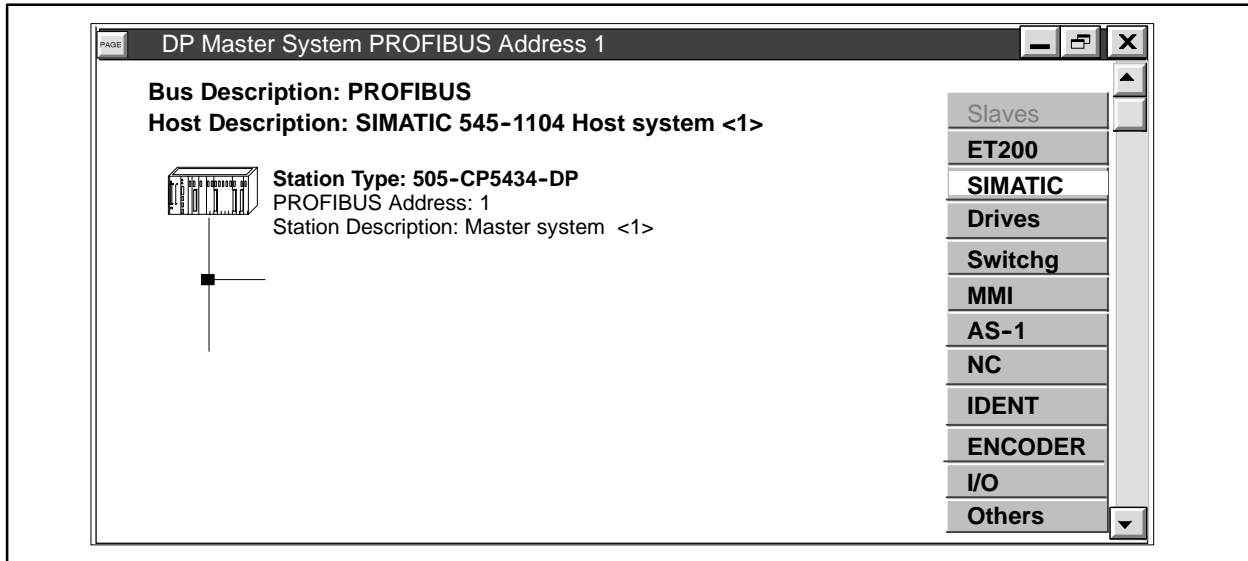


Figure 9-3 DP Master System PROFIBUS Address 1 Window

- From the list of slave types, select the PROFIBUS device that you want to add to your network. Click and drag it to a position underneath the master system. The PROFIBUS Address dialog box opens (Figure 9-4).

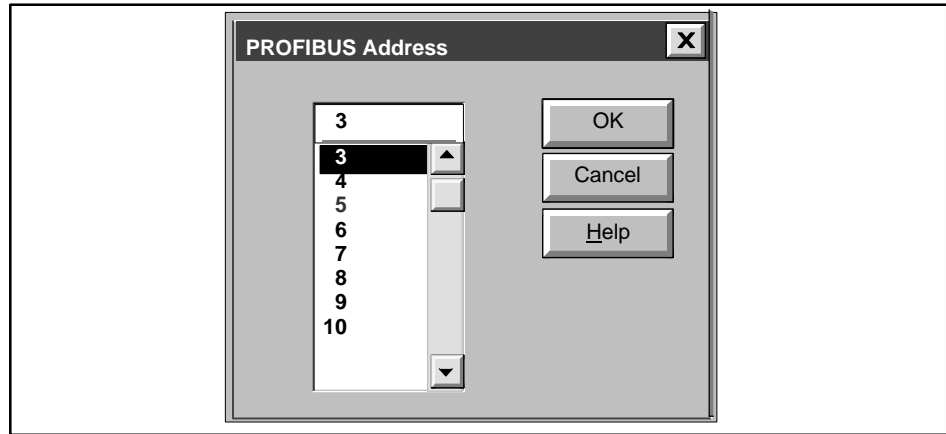


Figure 9-4 PROFIBUS Address Dialog Box

- Select the appropriate address and click **OK** to confirm. The Slave Parameters dialog box opens (Figure 9-5).

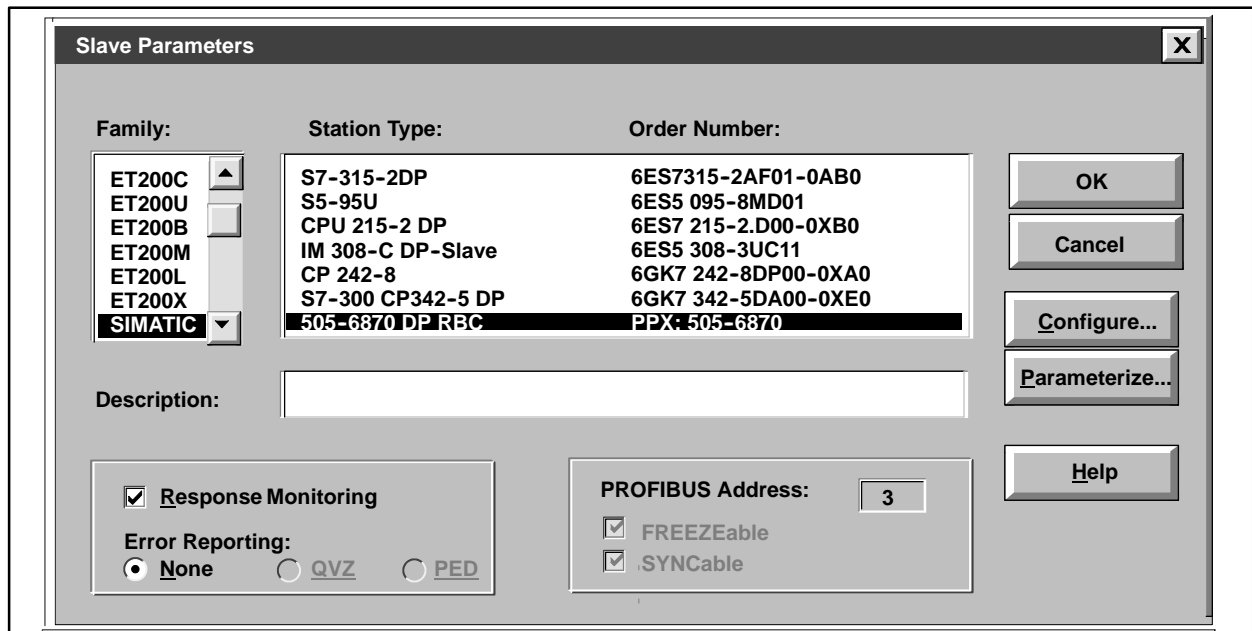


Figure 9-5 Slave Parameters Dialog Box

Configuring the PROFIBUS Network (Version 3.3) (continued)

- From the Slave Parameters dialog box, select the family and station type for the module that you want to insert as a slave on the PROFIBUS network.

If you select **505-6870 DP RBC**, you need to click **Parameterize** and change the 505 Mismatch Mode to **Enable** if you want the slave to be polled even when there is a module mismatch. If this parameter is left disabled the controller does not communicate with the slave when there is a module mismatch. Keep in mind that a module mismatch also occurs when an I/O module fails.

- Click the **Configure** button. The Configure dialog box appears. For different selections, this dialog box may require different information. [Figure 9-6](#) shows the Configure dialog box for the SIMATIC 505-6870 DP RBC. Steps 9-15 describe how to insert slave modules for the 505 line. For other kinds of modules, see your PROFIBUS documentation for details.

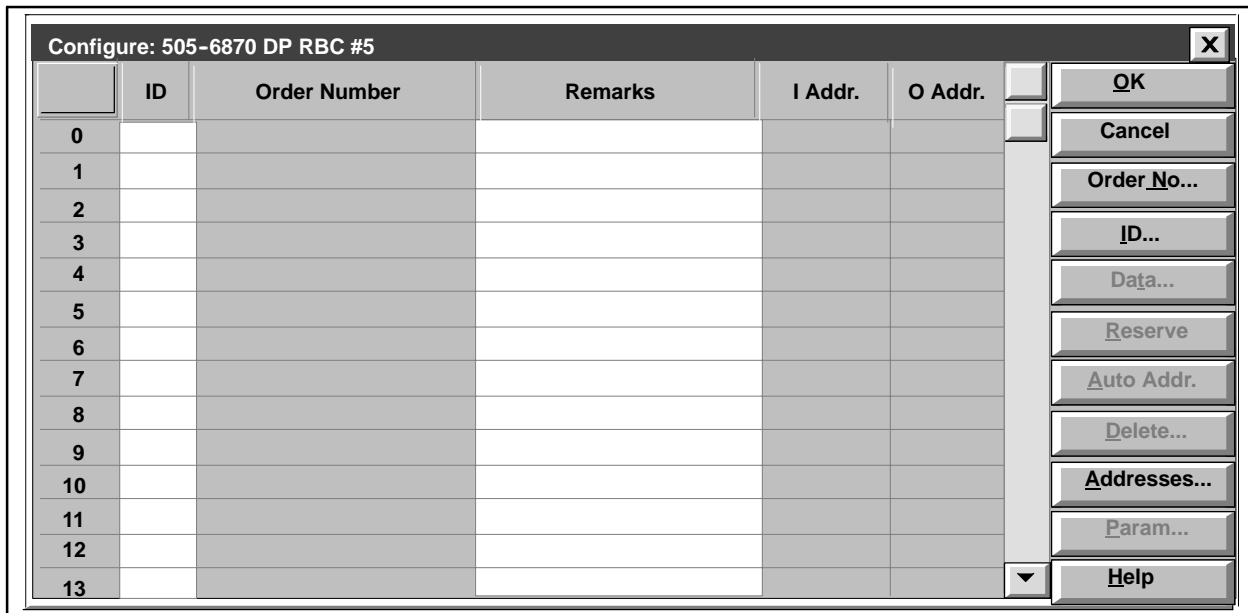


Figure 9-6 Configure Dialog Box

9. If you have selected from the SIMATIC family, double-click in the first box in the Order Number column. A selection list of 505 modules appears (Figure 9-7).
10. Select the module that corresponds to the hardware that you want to add to the first slot of the RBC.
11. Click **Accept** to confirm. Continue selecting modules until you have filled all the slots for which you have hardware.
12. Click **Close** to close the select list.

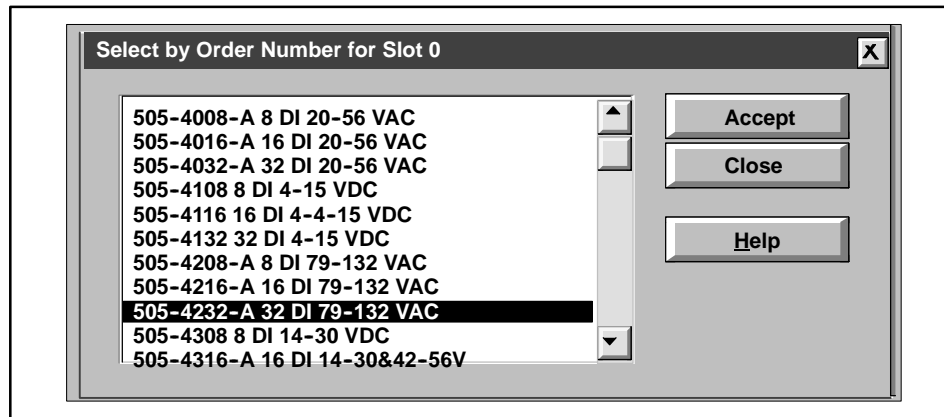


Figure 9-7 505 Module Selection Dialog Box

Configuring the PROFIBUS Network (Version 3.3) (continued)

- The module appears in the Configure dialog box (Figure 9-8). Click **OK** on the Configure dialog box to confirm your configuration.
- Click **OK** on the Slave Properties dialog box. The RBC is added to the PROFIBUS network in the DP Master System window. You do not need to configure addresses in COM PROFIBUS.
- Repeat steps 5-14 for any additional 505 RBCs that you want to add to the PROFIBUS network.

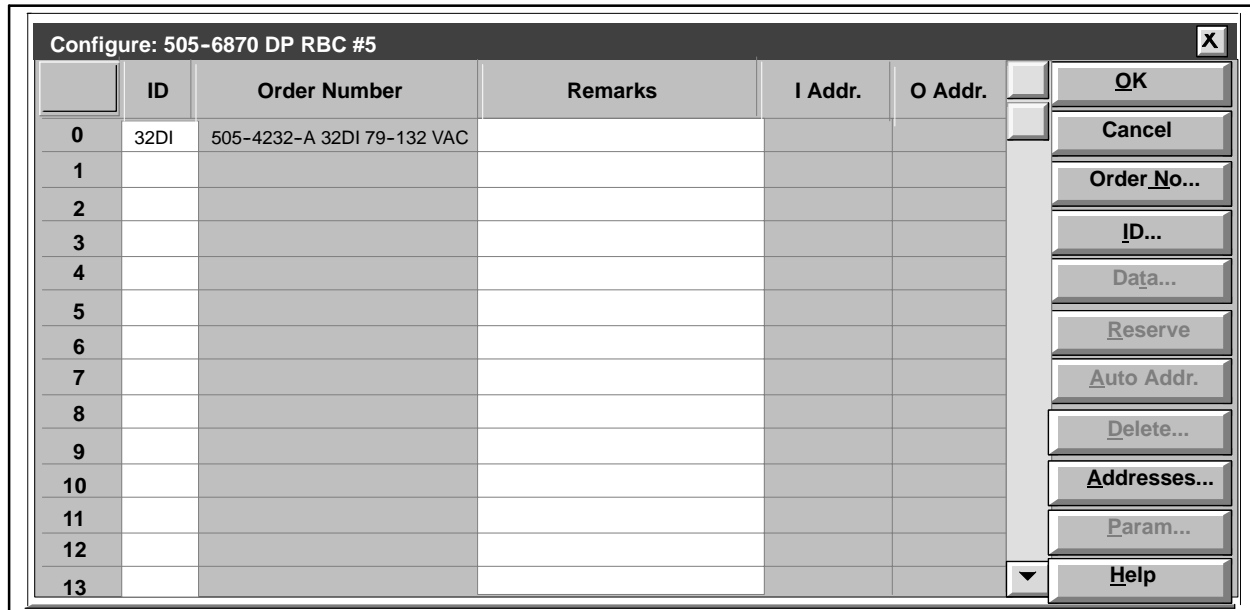


Figure 9-8 Configuring a Module

-
16. Select **File->Save As** to save the entire configuration. The **Save As** dialog box appears (Figure 9-9).
 17. Enter a file name and confirm it by clicking **OK**. The recommended directory is shown in Figure 9-9. Placing the file in this directory assures that it will be archived with your APT program. In this example, the APT program name is **example1**.

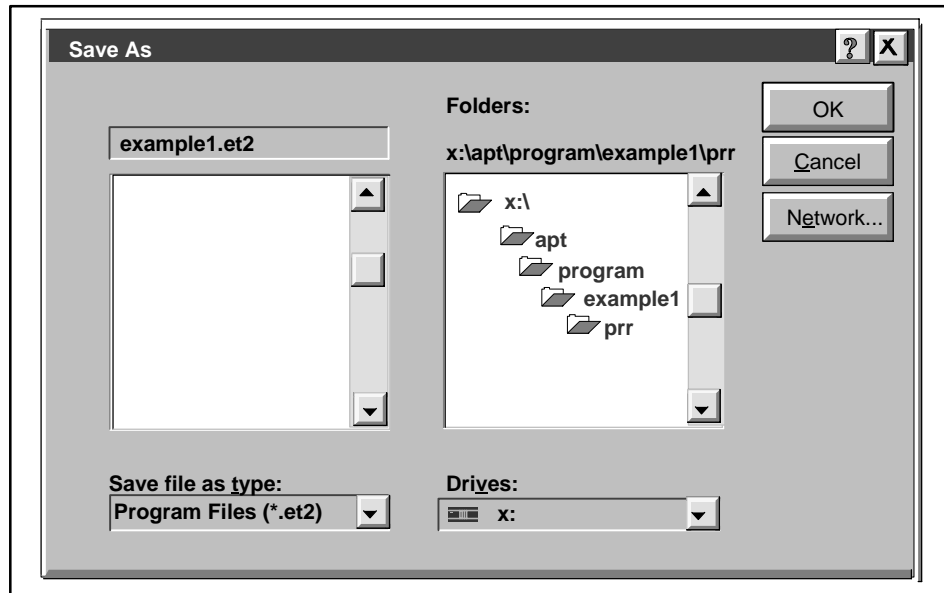


Figure 9-9 Saving the PROFIBUS Configuration

9.3 Exporting the Binary File to APT (COM PROFIBUS V 3.3)

Now that you have configured the PROFIBUS network with COM PROFIBUS, you need to export a binary file of that network to APT. Follow the steps below.

1. Select **File->Export->Binary File**. The **Export Binary File As** dialog box appears.
2. Change the file name to **export.2bf** (Figure 9-10).
3. Save the binary file in the **x:/APT/PROGRAM/<program_name>/PRR** directory in APT. This puts the file under the program hierarchy and ensures that the file is saved when you archive the program. In this example, the APT program name is **example1**.
4. Click **OK** on the Export Binary File As dialog box to confirm the directory path that you have selected.

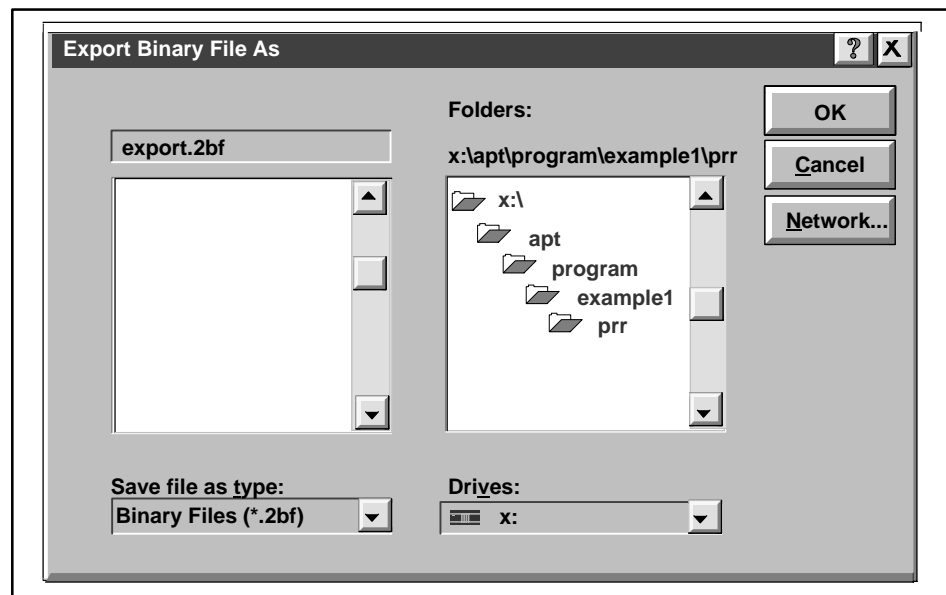


Figure 9-10 Selecting a Directory Path for the Export File

-
5. Click **OK** on the confirmation dialog box (Figure 9-11). The file is exported to APT.

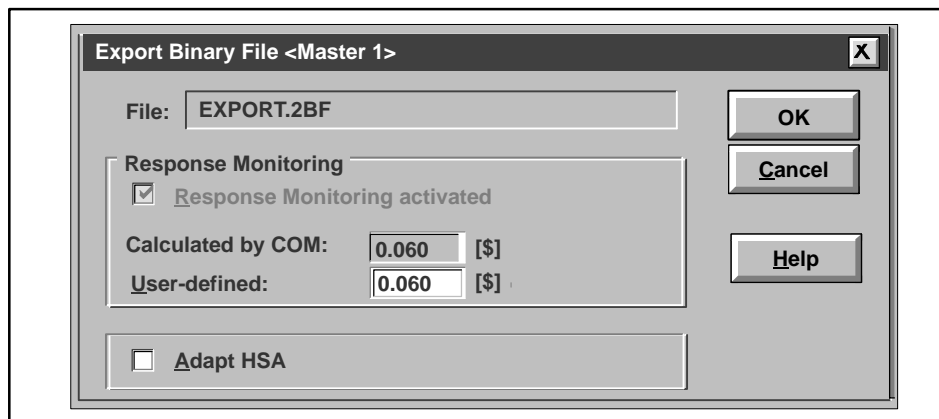


Figure 9-11 Export Binary File Dialog Box

In order to use PROFIBUS with APT, you must first configure the PROFIBUS network. See the *SIMATIC ET 200 Distributed I/O System Manual* for instructions on how to install and use COM PROFIBUS, and then follow the procedures in this section.

To configure the PROFIBUS network, follow these steps:

1. From the Windows environment, start COM PROFIBUS. (See the COM PROFIBUS documentation for instructions.) COM PROFIBUS opens with a new file.
2. From the tree view in the left-hand pane, expand **DP Master** and then expand the Master Station Type. For this example, the type will be **505-CP5434-DP**.
3. Select the controller that you want to use as the master on the PROFIBUS network. An image of the controller that you select appears in the right-hand pane of the window.

Right-click the image to display a context-sensitive menu for this master. You can change the parameters by selecting one of the active menu items.

4. From the tree view in the left-hand pane, expand **DP Slave**. Then expand **SIMATIC** and select **505-6870 DP RBC** (PPX: 505-6870) for this example. The RBC is added to the network graphic in the right-hand pane.
5. Double-click on the slave. The Slave Properties dialog box opens. Change the PROFIBUS address if necessary.
6. If you select **505-6870 DP RBC**, you need to click **Parameterize** and change the 505 Mismatch Mode to **Enable** if you want the slave to be polled even when there is a module mismatch. If this parameter is left disabled the controller does not communicate with the slave when there is a module mismatch. Keep in mind that a module mismatch also occurs when an I/O module fails. Press **OK**.
7. Click the **Configure** button. The Configure dialog box appears. For different selections, this dialog box may require different information. Steps 8-15 describe how to insert slave modules for the SIMATIC 505-6870 DP RBC. For other kinds of modules, see your PROFIBUS documentation for details.

-
8. Click in the first box in the Identifier column.
 9. Click on the **Module...** button. A selection list of 505 modules appears.
 10. Select the module that corresponds to the hardware that you want to add to the first slot of the RBC.
 11. Click **Apply** to confirm. Continue selecting modules until you have filled all the slots for which you have hardware.
 12. Click **Close** to close the select list. The module appears in the Configure dialog box.
 13. Click **OK** on the Configure dialog box to confirm your configuration.
 14. Click **OK** to exit the Slave Properties dialog box.
 15. Repeat steps 4-14 for any additional 505 RBCs that you want to add to the PROFIBUS network.
 16. Select **File->Save As** to save the entire configuration.
 17. Enter a file name and confirm it by clicking **OK**. The recommended directory is **apt\program\example1\pr**. Placing the file in the **pr** directory assures that it will be archived with your APT program. In this example, the APT program name is **example1**.

9.5 Exporting the Binary File to APT (COM PROFIBUS V 5.0)

Now that you have configured the PROFIBUS network with COM PROFIBUS, you need to export a binary file of that network to APT. Follow the steps below.

1. Select the DP Master in the network on the right-hand pane of the window.
2. Select **File->Export->Binary File**. The **Save As** dialog box appears, with the file type **COM PROFIBUS binary file (*.2bf)**.
3. Change the file name to **export**.
4. Save the binary file in the **x:\APT\PROGRAM\\PRR** directory in APT. This puts the file under the program hierarchy and ensures that the file is saved when you archive the program. In this example, the APT program name is **example1**.
5. Click **Save** on the Save As dialog box to confirm the directory path that you have selected.
6. Click **OK** on the confirmation dialog box. The file is exported to APT.

9.6 Adding PROFIBUS Modules in APT

The I/O modules that are on the PROFIBUS network must be added to the APT Module Table in unused slots. Since the I/O on the PROFIBUS network must be configured on Channel 1 along with the standard 505 I/O, the total number of I/O for the system is limited to 16 bases with 16 I/O modules (standard and/or PROFIBUS) per base.

The recommended placement of I/O modules for the PROFIBUS network is on a base that is not being used for standard 505 I/O. For system expansion, begin with base 15 for I/O on the PROFIBUS network, and then work your way back through other bases not used. For example, if you have a controller with five standard I/O bases numbered consecutively 1 through 5, then configure the PROFIBUS I/O on base 15, reserving bases 6 through 14 for further expansion of I/O. (This does not apply to the 545-1103 controller which can use only base 0.)

You can configure PROFIBUS I/O on unused slots in bases that contain 505 I/O, but this can cause confusion for maintenance when the rack in the field does not match the modules in the Module Table. If you choose to configure PROFIBUS I/O in 505 bases with unused slots, you should add syntax to the I/O name or description to indicate the type of I/O that you have configured.

For devices other than 505 I/O modules, such as an ET 200, you add the USER module in the APT module table. Since the APT USER module is limited to 64 I/O points, you can not use more than 64 I/O points on any single PROFIBUS device. The USER module is limited to those I/O mixes that are valid on 505 controllers.

Once the module is placed in the Module Table, APT automatically assigns the addresses to the I/O. You can change these addresses manually. See [page 3-8](#) for naming and giving descriptions to the I/O.

Adding PROFIBUS Modules in APT (continued)

In addition to configuring the modules in COM PROFIBUS, you must also add a corresponding module to the Module Table in APT. To add the I/O modules for the PROFIBUS network to the APT Module Table, follow the steps below.

1. In the APT program, press **F10** in Program Content to open the Module Table (Figure 9-12).
2. Select the base where you want to insert the modules for the PROFIBUS network. It is recommended that you start with base 15.
3. Press **F10** to configure the modules in the slots.
4. Use the arrow keys to select the slot where you want to place the I/O module.
5. Press **F4** to display a list of modules.
6. Use the arrow keys to select the module, and press **Enter**. Be sure that the module matches the one that you entered in COM PROFIBUS. For example, if you configure a 32DI module, you must add a 32DI module here. The module is added to the appropriate slot in the base.
7. APT automatically assigns an address to the I/O. You can use this address or modify it to any valid address.

8. Repeat steps 4-7 for each I/O module on the PROFIBUS network.
9. Press **F7** to split the screen. Use the split screen mode to create the PROFIBUS.CFG report (Section 9.7). See page 1-13 for more information on using split screens.

NOTE: Keep in mind that PROFIBUS devices from different vendors often use different data formatting rules and algorithms. This is especially true for scaling in analog devices. Do not assume that the APT I/O types such as AI, RT, and TC I/O will correctly scale data from your PROFIBUS devices.

Module Table

Channel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1																
2																
3																
4																
5																
6																
7																

Channel: 1 Base: 15 Slot: 1 32DI

I/O Series: 505

Slot	Type	Addr	Slot	Type	Addr
1	32DI	89	9		
2	32DO	121	10		
3	32DI	153	11		
4	32DO	105	12		
5			13		
6			14		
7			15		
8			16		

POS	ADDRESS	UNIT	I/O NAME	DESCRIPTION
1	WX 0089	_____	ALPHA_SWITCH	
2	WX 0090	_____	DRAIN_SWITCH	
3	WX 0091	_____	_____	
4	WX 0092	_____	_____	
5	WX 0093	_____	_____	
6	WX 0094	_____	_____	
7	WX 0095	_____	_____	
8	WX 0096	_____	_____	

Figure 9-12 Defining I/O Points

9.7 Creating a PROFIBUS Configuration File in APT

In order for APT to link the I/O modules in the Module Table with the actual I/O modules on the PROFIBUS network, you need to create a PROFIBUS configuration file in APT. Follow these steps:

1. In Program Content, use the arrow keys to highlight **SAVED REPORTS**.
2. Select **F3**, and then select **Add New Report** (Figure 9-13).
3. Name the report PROFIBUS.CFG (Figure 9-14).

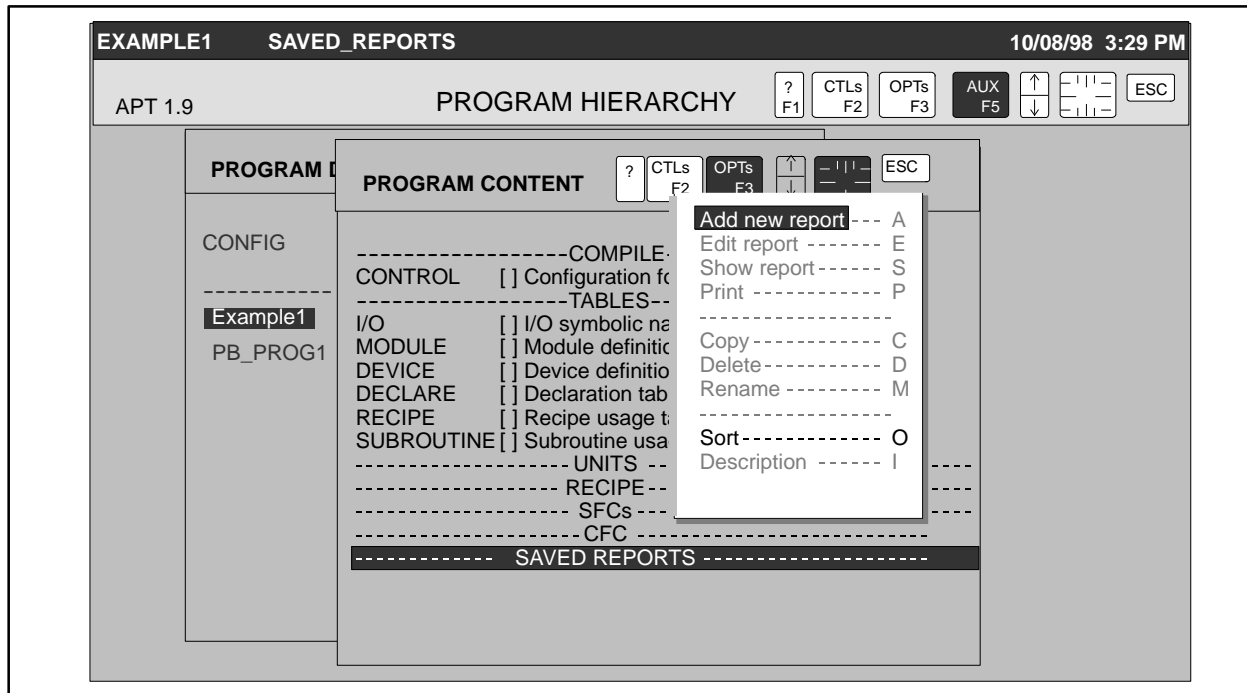


Figure 9-13 Creating the PROFIBUS.CFG Report

NOTE: If you use the split screen feature of APT, you can keep the Module Table open while you create your PROFIBUS configuration report. See [page 1-13](#) for more information on using split screens.

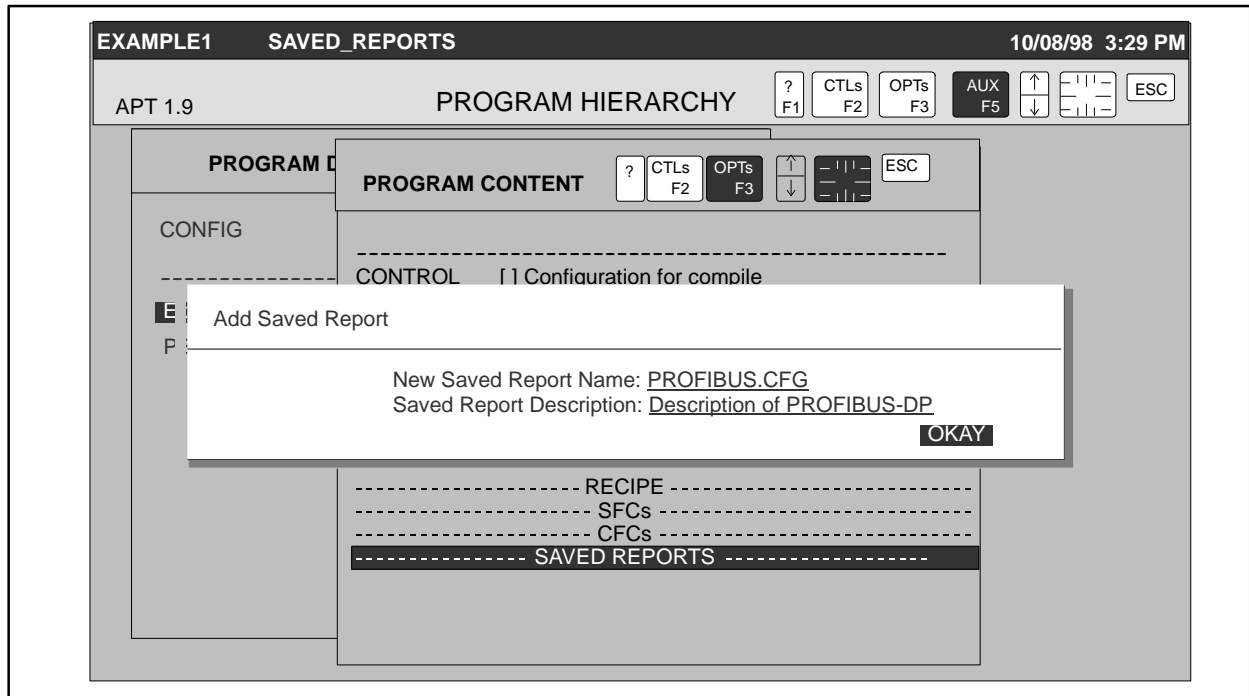


Figure 9-14 Naming the PROFIBUS.CFG File

4. Edit the PROFIBUS configuration file, following the guidelines on page 9-22. Figure 9-15 shows an example.

In Figure 9-15, the I/O Module configured in COM PROFIBUS for Slave 7, Modules 2-5 will be linked to Slots 7, 8, 9, and 10 on Channel 1, Base 15 in the Module Table of APT.

5. Save the PROFIBUS.CFG report. When you compile the program, APT uses this file to locate the modules on the PROFIBUS network.

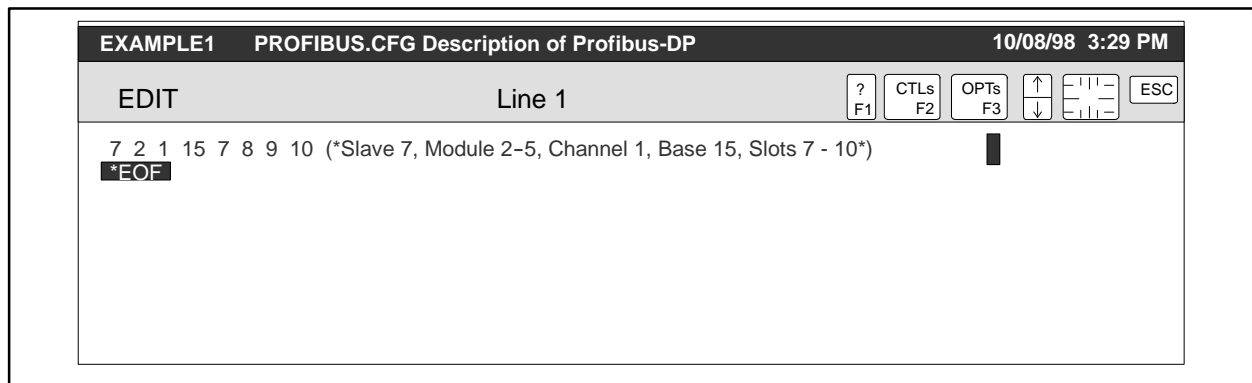


Figure 9-15 Example PROFIBUS.CFG File

Creating a PROFIBUS Configuration File in APT (continued)

Guidelines for PROFIBUS.CFG Report

The columns of the **PROFIBUS.CFG** file (with the valid range of values in parentheses) are listed below:

- Slave (1 . . 112)
- Module (0 . . 15)
- Channel (1 . . 8). APT Release 1.9 accepts only Channel 1.
- Base (0 . . 15)
- Slot (1 . . 16). You can add as many Slot columns to the end of a line as needed. If no slot is specified, then all slots (1 . . 16) are used, starting with Slot 1.

Each line of the report must consist of numbers only, except for comments. The columns can be separated by spaces or commas. Use one line only for each definition.

Comments can be marked off by a combination of parentheses and asterisks (* *) or by curly brackets { }. All comments in [Figure 9-16](#) are inserted for clarity and are not necessary for compilation.

Invalid numbers (for example, 113 for Slave, or 0 for Slot) will cause an error.

See further examples in [Figure 9-16](#).

```
{Format: SLAVE    MODULE    CHANNEL    BASE    SLOT    [SLOT ...]
Ranges  [1 . . 112]  [0 . . 15]  [1 . . 8]  [0 . . 15]  [1 . . 16] }
```

(*The following example for Slave 5 shows a format where each line maps one module to a slot:*)

```
5 0 1 15 1 (*Slave 5, Module 0, Channel 1, Base 15, Slot 1*)
5 1 1 15 2 (*Slave 5, Module 1, Channel 1, Base 15, Slot 2*)
5 2 1 15 3 (*Slave 5, Module 2, Channel 1, Base 15, Slot 3*)
```

{A more efficient format maps multiple modules to slots in a single line:}

```
6 0 1 15 4 5 6 (*Slave 6, Module 0, Channel 1, Base 15, Slots 4, 5, 6*)
```

{If you do not enter a value for the Slot, all slots are used:}

```
4 0 1 15 (*Slave 4, Module 0, Channel 1, Base 15, Slots 1-16*)
```

Figure 9-16 Format and Examples for PROFIBUS.CFG File

9.8 Debugging PROFIBUS in APT

Under the AUX menu in the APT Debug utility, there are three menu items specifically related to PROFIBUS:

- **PROFIBUS-DP Mode** — sets the mode for the slave or for the operational mode.
- **PROFIBUS-DP Sync Mode** — selects whether or not to synchronize to the RLL scan.
- **Read PROFIBUS-DP I/O** — displays the I/O configuration (x, y, wx, wy) and addresses for a module on the PROFIBUS network. This option also creates and saves a report, which you can view or print.

Setting the Slave Mode

You can enable and disable slaves from Debug in APT. To set the mode for a slave module on the PROFIBUS network in APT, follow these steps:

1. In Debug, press **F5**. The AUX functions are displayed.
2. Use the arrow keys to highlight **PROFIBUS-DP Mode** (Figure 9-17).

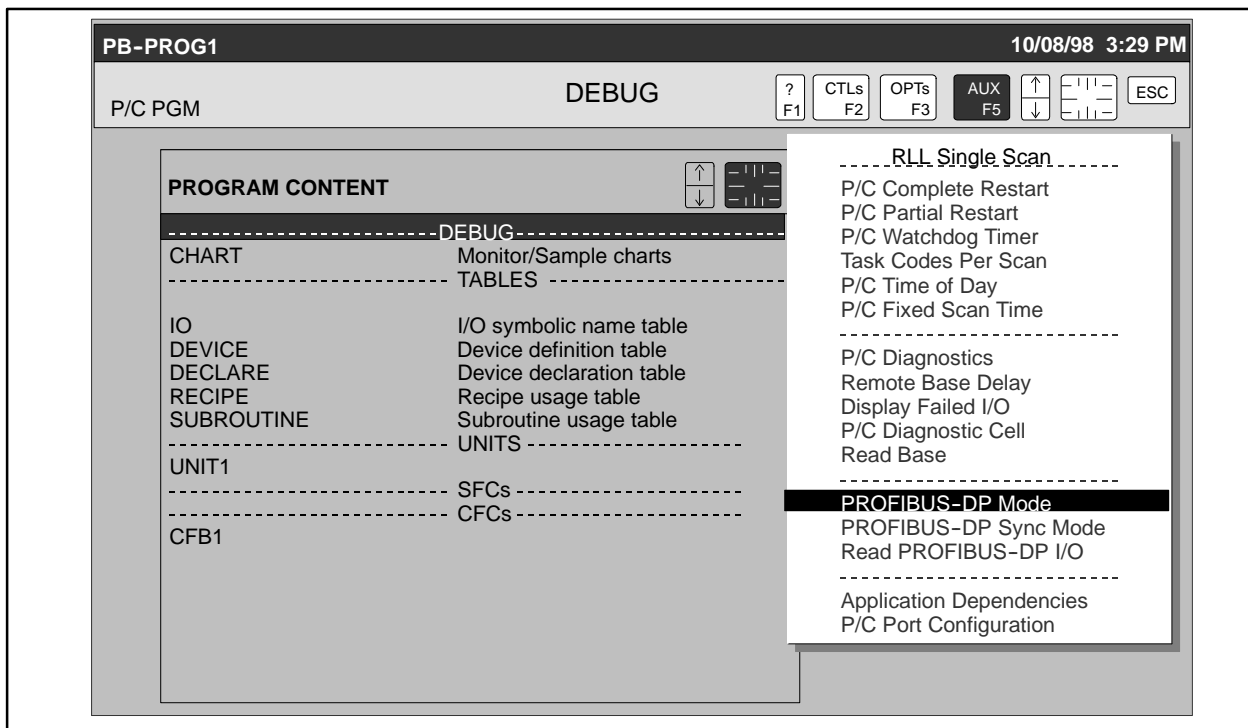


Figure 9-17 Selecting PROFIBUS-DP Mode

Debugging PROFIBUS in APT (continued)

3. Press **Enter**. The PROFIBUS-DP Set Mode dialog box appears (Figure 9-18). **Slave** is highlighted.

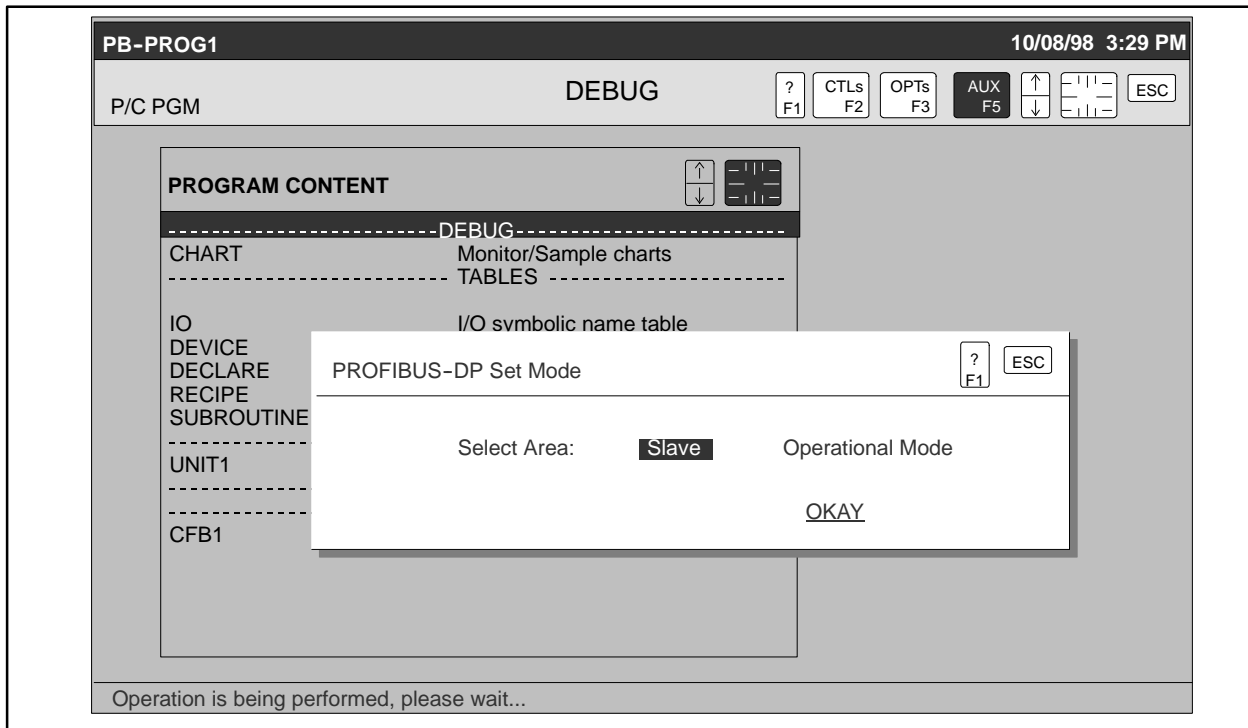


Figure 9-18 Selecting Slave

4. Press **Enter** twice. The PROFIBUS-DP Slave dialog box appears (Figure 9-19).

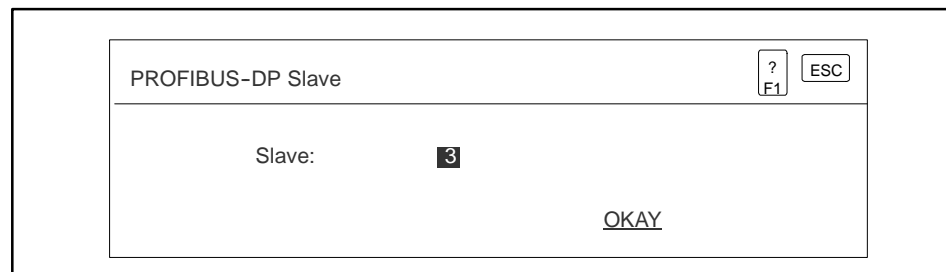



Figure 9-19 Specifying the Slave Address

5. Enter the slave address on the PROFIBUS network in the Slave field, and press **Enter** twice. The PROFIBUS-DP Set Mode — Slave dialog box appears (Figure 9-20).
6. Use the arrow keys to highlight **Activate** to enable the slave or **Deactivate** to disable the slave, and press **Enter** twice. The slave is set to the mode that you selected.

 WARNING
<p>If you deactivate (disable) a slave while the process is running, your slave I/O will not operate as intended.</p> <p>Unpredictable controller operations can result in serious injury or death to personnel, and/or damage to equipment.</p> <p>Do not deactivate (disable) slaves that are involved in an active process.</p>

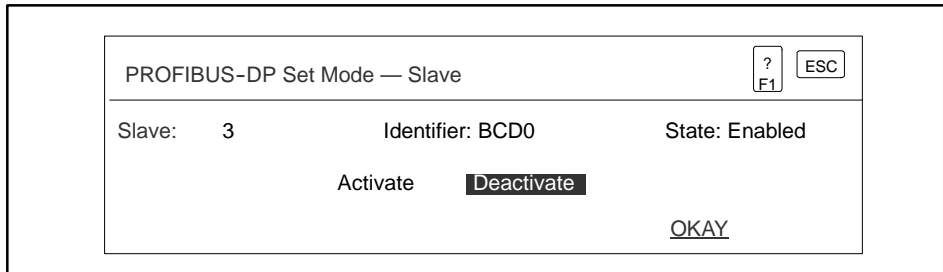


Figure 9-20 Setting the Slave Mode

Debugging PROFIBUS in APT (continued)

Setting the Operational Mode

To set the operational mode for the PROFIBUS network, follow the steps below.

1. In Debug, press **F5**.
2. Use the arrow keys to highlight **PROFIBUS-DP Mode** (Figure 9-17).
3. Press **Enter**. The PROFIBUS-DP Set Mode screen appears (Figure 9-21).

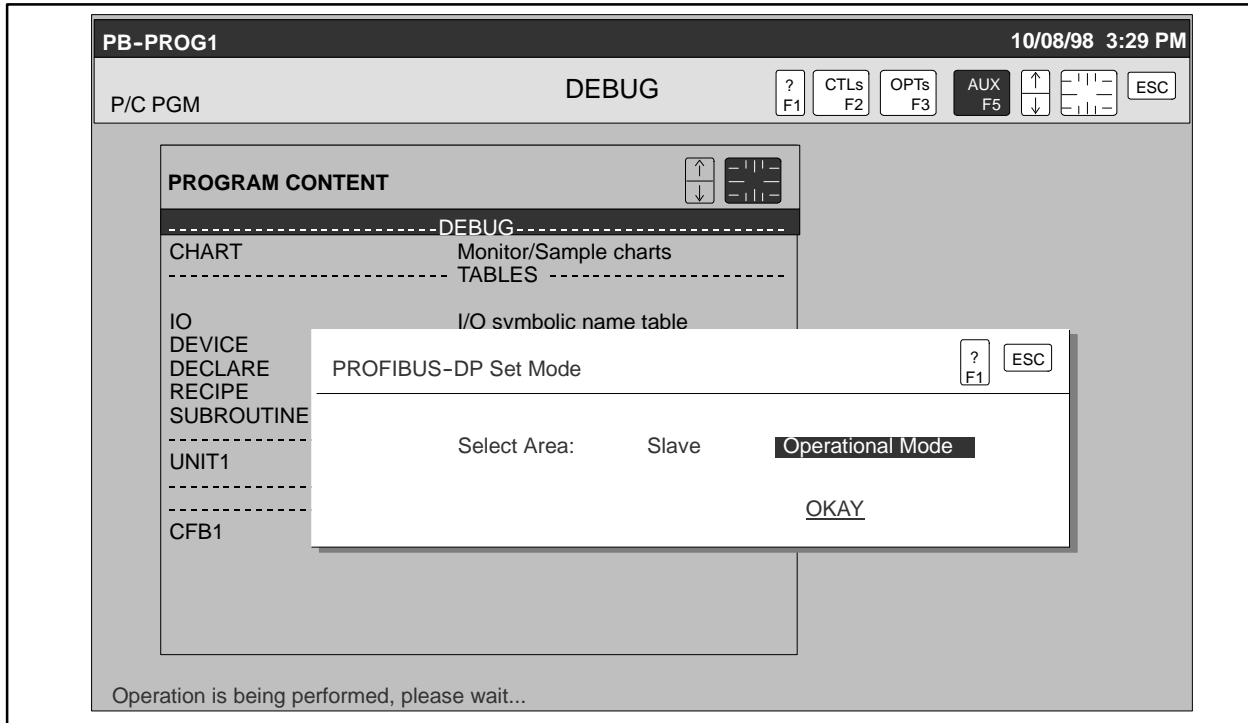


Figure 9-21 Selecting Operational Mode

4. Select **Operational Mode**, and press **Enter** twice. The PROFIBUS-DP Set Mode — Operational Mode dialog box appears (Figure 9-22).
5. Highlight the operational mode that you want to select, and press **Enter** twice.

In **Operate**, the inputs from the slaves are passed to the master and the outputs from the master are passed to the slaves.

In **Clear**, the inputs from the slaves are read and passed to the master. Outputs from the master are ignored.

In **Stop**, the slaves are not polled.

⚠ WARNING
<p>If you set the operational mode of the PROFIBUS-DP to CLEAR or STOP while the process is running, your slave I/O will not operate as intended.</p> <p>Unpredictable controller operations can result in serious injury or death to personnel, and/or damage to equipment.</p> <p>Do not set the operational mode of the PROFIBUS-DP to CLEAR or STOP while the process is running.</p>

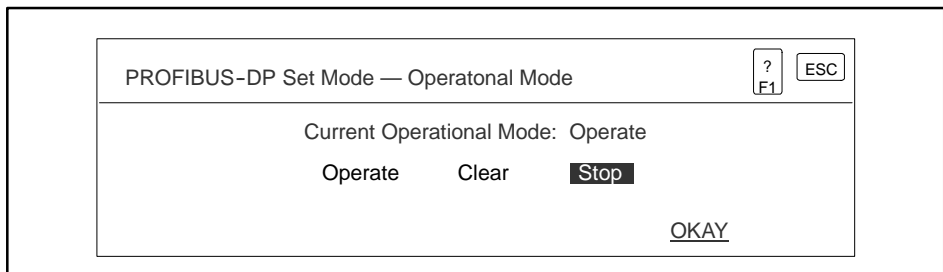


Figure 9-22 Setting the Operational Mode

Debugging PROFIBUS in APT (continued)

Setting the PROFIBUS-DP Synchronization Mode

In Debug, you can specify whether to synchronize the PROFIBUS network to the RLL scan or not. Normally, the PROFIBUS-DP I/O and the RLL scans are not coordinated; that is, they operate asynchronously. If the worst case PROFIBUS-DP I/O scan time is not at least two times as fast as the best case RLL scan time, it is possible to lose output transitions. If this occurs, place the system in synchronous mode.

You can set the synchronization mode for the PROFIBUS-DP I/O at any time. However, it does not take effect until the next transition of the PROFIBUS-DP I/O from STOP to CLEAR mode. Follow these steps:

1. In Debug, press **F5**. The AUX functions are displayed.
2. Use the arrow keys to highlight **PROFIBUS-DP Sync Mode** (Figure 9-23).
3. Press **Enter**. The PROFIBUS-DP Synchronization Mode screen appears (Figure 9-24).

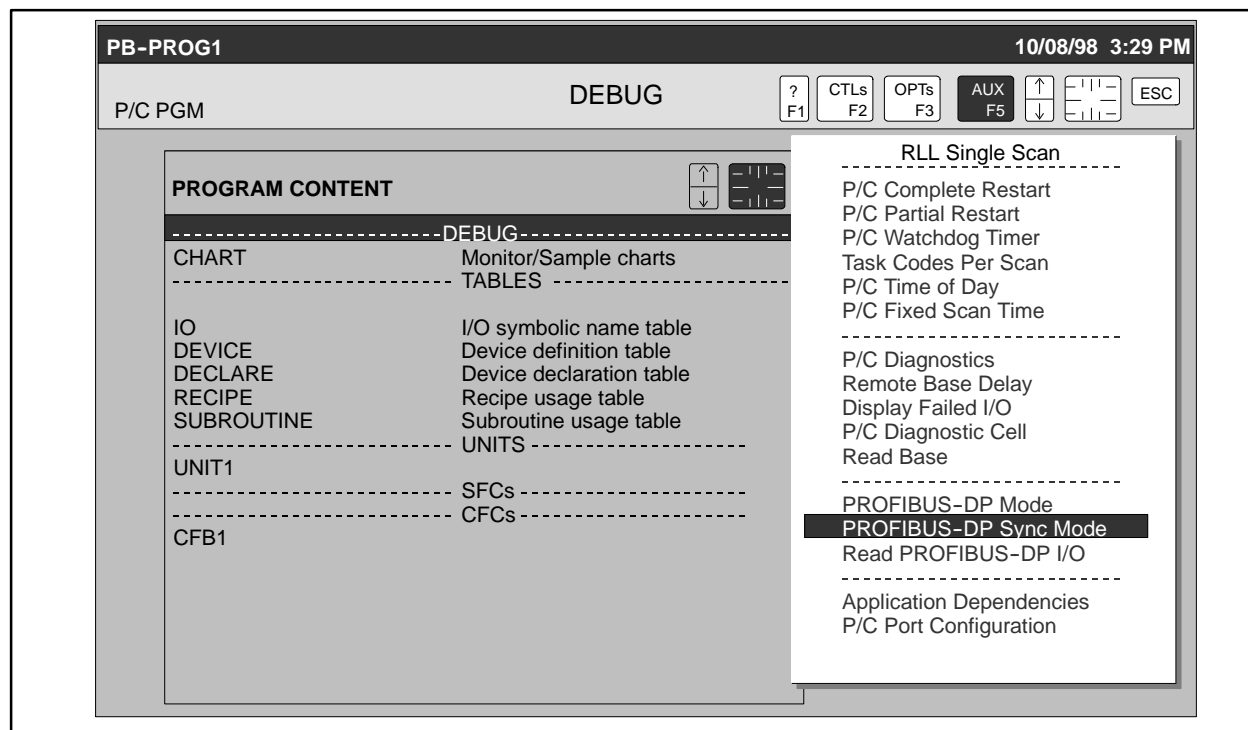


Figure 9-23 Selecting PROFIBUS-DP Synchronization Mode

4. Use the arrow keys to highlight **Yes** if you want to synchronize the PROFIBUS-DP I/O with the RLL scan, **No** if you do not.
5. Press **Enter** twice. The synchronization mode is set as you selected. However, it does not take effect until the next transition of the PROFIBUS-DP I/O from STOP to CLEAR mode. See [page 9-26](#), “Setting the Operational Mode,” to cause this transition.

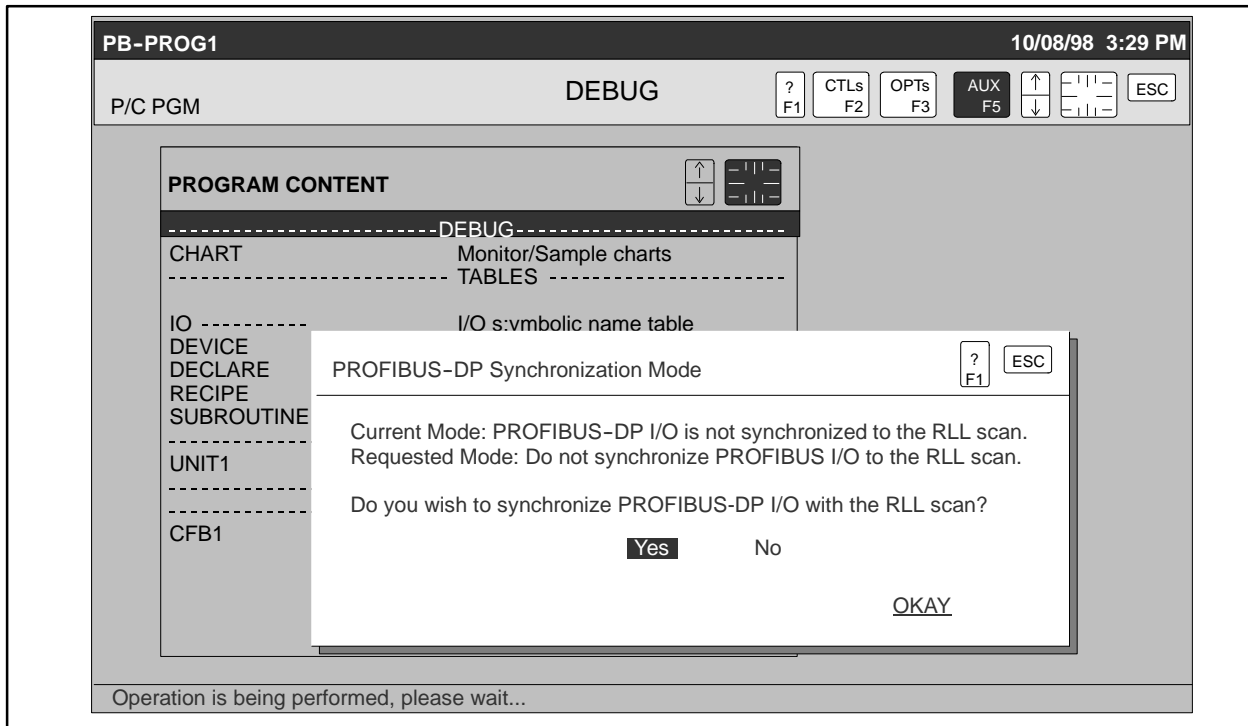


Figure 9-24 Synchronizing PROFIBUS-DP I/O with the RLL Scan

Debugging PROFIBUS in APT (continued)

Reading PROFIBUS-DP I/O

In Debug, you can read the PROFIBUS device to view the I/O modules and addresses that are configured in the controller for a particular slave. Follow these steps:

1. In Debug, press **F5**. The AUX functions are displayed.
2. Use the arrow keys to highlight **Read PROFIBUS-DP I/O** (Figure 9-25).
3. Press **Enter**. The PROFIBUS-DP Slave dialog box appears (Figure 9-26).

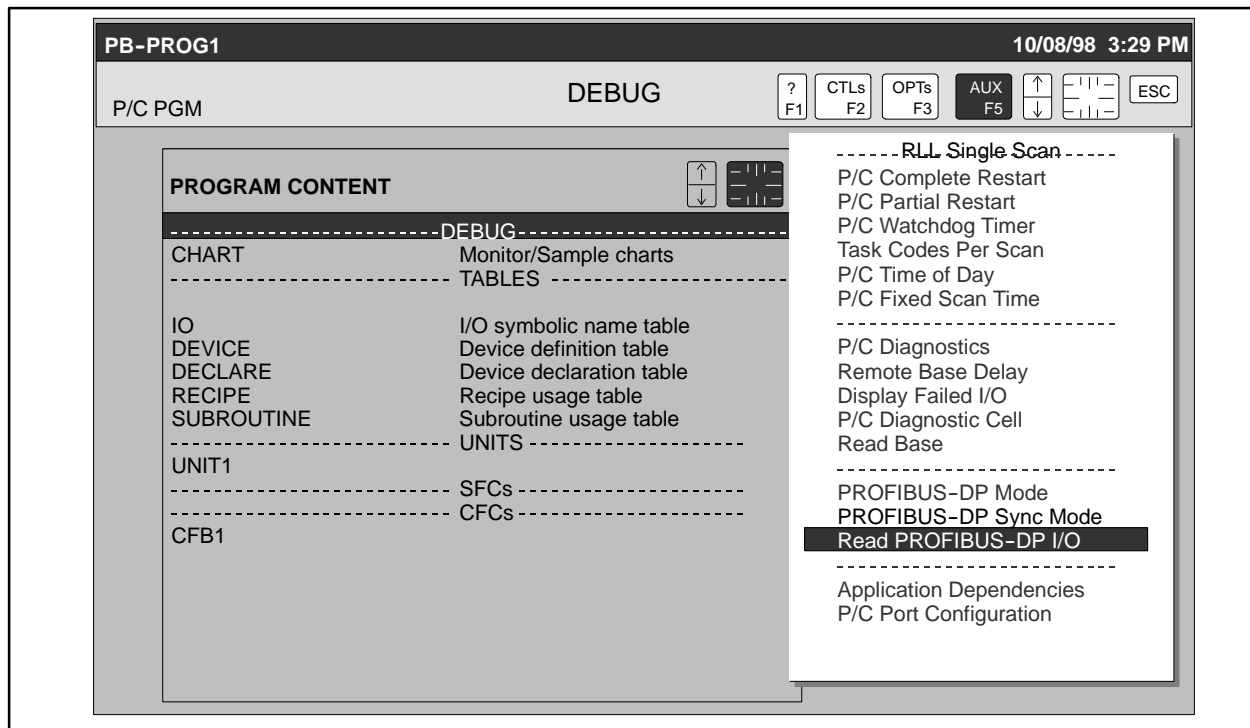


Figure 9-25 Selecting Read PROFIBUS-DP I/O

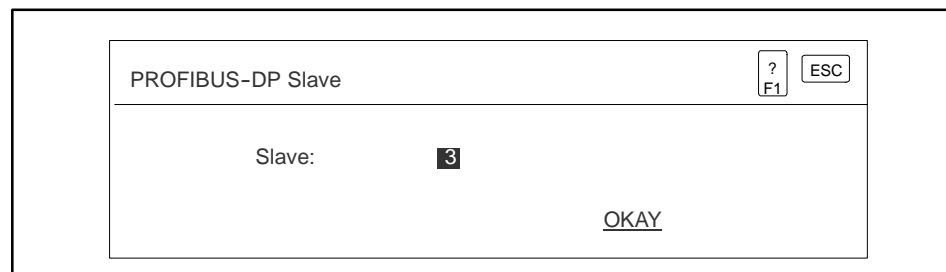


Figure 9-26 Specifying the Slave Address

4. Enter the address of the slave that you want to view, and press **Enter** twice. The PROFIBUS-DP Slave I/O Configuration report appears (Figure 9-27).

Viewing this report also saves it under the Saved Reports in Program Content with the name **SLAVE<nnn>.RPT**, where **nnn** stands for the slave address; for example, **SLAVE003.RPT**.

5. Press **Enter** to dismiss the screen.

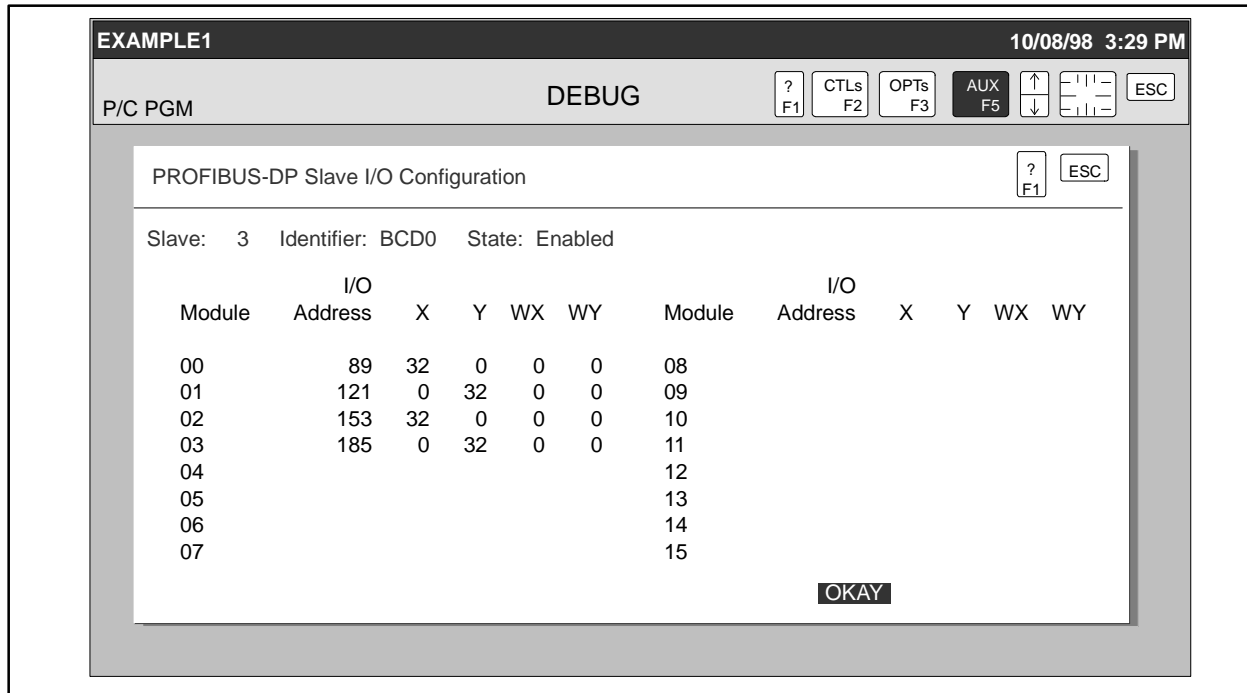


Figure 9-27 PROFIBUS-DP Slave I/O Configuration Report

Appendix A

APT Installation

A.1	Before Installing APT	A-2
	Compatibility	A-2
	Minimum Hardware and Software	A-3
	Recommended Hardware and Software	A-3
	Setting Up the CONFIG.SYS and AUTOEXEC.BAT Files	A-4
	DOS Memory	A-4
	Microsoft Mouse or Compatible	A-4
A.2	Installing APT	A-5
	Overview	A-5
	Creating Diskettes	A-5
	Starting the Installation	A-6
	APT Installation	A-7
	Configuring the System	A-8
	Copying the Files	A-11
	Setting Computer to APT Mode (DOS)	A-11
	Running APT	A-12
	Loading an Example Program	A-12
	Changing between APT Mode and DOS Mode (DOS)	A-14
	Configuring the CVU/TIWAY Adapter (Series 505 Only) (DOS/Win95 Only)	A-15
	Configuring the 386/ATM (Series 505 Only) (DOS Only)	A-15
A.3	Setting DOS Environment Variables	A-16
A.4	Changing the APT Graphic Configuration	A-24
	CGI.CFG File	A-24
	Printer Support	A-25
A.5	Hardware Connections	A-26
	Port Pinouts	A-26
	Series 505 Communication Links	A-28
	S5 Communication Links	A-30

A.1 Before Installing APT

Compatibility

You can install APT in any of the following configurations.

- TISTAR operator station (Series 505 only)
- APT engineering node in the OSx system
- IBM Personal Computer or compatible, with 80386, 80486, or Pentium processor, EGA, VGA, TISTAR/CVU MATROX, or TISTAR TIGA graphics card, and DOS 5.0 or higher.

APT requires a customized system configuration that may not be compatible with other software on your system. See [Page A-11](#) for information about saving your original configuration.

If you want to use a mouse with APT, choose a model that is compatible with the three-button Mouse Systems mouse or the two-button Microsoft mouse.

Before you install APT, make certain that you have at least 30 Mbytes, but preferably 60 to 100 Mbytes, of free space available on your hard disk.

Minimum Hardware and Software

For acceptable speed and performance by the APT software, your system should be equipped with the hardware and software described in this section.

Hardware For your system unit, use an 80486-compatible computer with a clock speed of 50 MHz. The hard disk should have an access speed of 15 ms or less, and 60 to 100 Mbytes of free space available for APT and your programs.

A very large program may require over 25 Mbytes of disk space when restored and over 10 Mbytes of disk space when archived. Do not allow your APT hard disk partition to become more than 95% full.

Software These software packages are not included with APT, but they can be very useful for customizing your APT system. Other packages are commercially available.

For memory management, e.g., loading drivers into high memory, installing a RAM disk, disk-caching, etc., consider MS-DOS Rel. 6.x, Quarterdeck Expanded Memory Manager (QEMM), or equivalent.

NOTE: The DOS memory requirements of APT necessitate that you use DOS 5.0 (or later) with APT. Additional TSR (Terminate and Stay Resident) requirements, such as network drivers, may require better memory management than DOS 5.0. For memory management techniques, refer to [Appendix C](#) of this manual.

For disk-caching, file defragmenting, and file repairs, consider packages such as Norton Utilities, PC-Tools, or equivalent.

NOTE: You can enhance APT performance by using a disk cache of 500 Kbytes or more.

For file compression utilities, consider PKZIP and PKUNZIP or equivalent programs.

Recommended Hardware and Software

For optimal performance of APT, use a computer with a Pentium III or better processor and a state-of-the-art operating system such as Windows NT or Windows 2000.

Before Installing APT (continued)

Setting Up the CONFIG.SYS and AUTOEXEC.BAT Files

The number of files listed in the **config.sys** file must be at least 40. If your system does not specify 40 files or more, add `files=40` to the **config.sys** file, and reboot the computer before installing APT.

For optimum performance of APT, specify 20 buffers in the **config.sys** file. If you are using a disk cache, you can experiment with smaller values after installation: `buffers=10`

NOTE: For new computers, the default values for **files** and **buffers** are often less than 20. Be sure to check these values before installing APT on a new computer.

DOS Memory

APT requires at least 520 Kbytes of DOS conventional memory to run; however, depending upon the size, contents and configuration of the APT program that you create, the APT compilation process may require considerably more. Refer to [Appendix C](#) for information on DOS memory types.

In order to run, APT must load special graphic drivers that can require up to 90 Kbytes of memory. If your **config.sys** and **autoexec.bat** do not yield at least 610 Kbytes of DOS memory before the APT graphics drivers are loaded, you may not be able to execute APT.

[Appendix C](#) and [D](#) discuss memory management and suggest possible **config.sys** and **autoexec.bat** configurations. If you do not have enough conventional memory for APT and the graphic drivers, you may be able to load the drivers in upper memory after the APT installation.

If your current configuration does not allow you to add APT and its drivers, APT sets up an APTMODE and DOSMODE configuration. Refer to [page A-14](#) for more information on APTMODE and DOSMODE.

Microsoft Mouse or Compatible

In order to use a Microsoft (or compatible) mouse with APT, you must install the driver that came with the mouse. Typically the file is called **mouse.com**. Refer to your mouse user manual for installation instructions.

A.2 Installing APT

Overview

The APT software media consists of a single CD-ROM, which you use to install APT on your system. If diskette media are required for your system, the contents of the CD is organized to permit generation of diskettes in either 1.2 MB or 1.44 MB format. Instructions for creating these diskettes are included below in the following subsection. The installation procedure takes about ten minutes and consists of these tasks:

- Configuring the system
- Copying files to a hard drive
- Optionally setting your computer to APT mode (DOS and Windows 95 only)
- Optionally loading an example

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_n folder name immediately after it is created.

Starting the Installation

Select the appropriate procedure below for starting the installation of APT from your media (CD-ROM or diskette) to your system (DOS or Windows).

CD-ROM on DOS Systems: To install APT from CD-ROM on a DOS system, follow these steps:

1. Place the APT software CD in the CD-ROM drive.
2. At the DOS command prompt, enter the following command:

```
<CD-ROM drive>:\DISK1\INSTALL
```

The APT Installation program starts. Proceed to “APT Installation.”

CD-ROM on Windows Systems: To install APT from CD-ROM on a Windows 95, Windows NT, or Windows 2000 system, follow these steps:

1. Place the APT software CD in the CD-ROM drive.
2. From the Windows Taskbar, click **Start->Run**.
3. In the Run dialog box, click **Browse**, and then browse to your CD-ROM drive.
4. Double-click on the folder named **DISK1**.
5. Double-click on **Install.exe**.
6. Click **OK**.

The APT Installation program starts. Proceed to “APT Installation.”

Diskettes on DOS Systems: To install APT from diskettes on a DOS system, follow these steps:

1. Place the APT software diskette labeled “DISK1” in the Floppy drive.
2. At the DOS command prompt, enter the following command:

```
<Floppy drive>:\INSTALL
```

The APT Installation program starts. Proceed to “APT Installation.”

Diskettes on Windows Systems: To install APT from diskettes on a Windows 95, Windows NT, or Windows 2000 system, follow these steps:

1. Place the APT software diskette labeled “DISK1” in the Floppy drive.
2. From the Windows Taskbar, click **Start->Run**.
3. In the Run dialog box, click **Browse**, and then browse to your Floppy drive.
4. Double-click on **Install.exe**.
5. Click **OK**.

The APT Installation program starts. Proceed to “APT Installation.”

APT Installation

After starting, the APT Installation screen appears with installation instructions. You need to know your system configuration. Check [Table A-1](#) and make sure that you know which option to choose for each configuration parameter. To proceed with the installation, follow these steps:

1. Press **F10** to continue.
2. Fill in the registration information. The serial number is located on the back of the APT software CD case and on the shipping box. Press **Esc** when you are finished.
3. After you have completed the installation, remove your installation media and store the CD (and the diskettes, if you created them) in a safe place.

Installing APT (continued)

Configuring the System

To configure APT, you need to specify the items listed in [Table A-1](#) on [page A-9](#) as they apply to your system. Follow these steps.

1. Use the arrow keys to highlight an item and press **Enter** to see the available options.
2. Highlight the appropriate option and press **Enter**. Help is available by pressing **F1**.
3. When you have completed the changes, press **F10** to continue and answer **Yes** to save the configuration.

Follow these guidelines for configuring the system:

- For a Logitech three-button Trackman that emulates a Microsoft mouse, select the Microsoft mouse.
- If APT is going to communicate to the controller through PCS 3.x or OSx 4.x and/or an Industrial Ethernet network, then you must perform the hardware and software network configuration before you install APT. APT uses files and subdirectories that are created by the configuration utility to set up its communication. See [Appendix B](#) for information about connecting to the Industrial Ethernet network.

NOTE: Do not install APT on substituted drives. You will not be able to run APT if the same substitution is not set up again.

Table A-1 APT Configuration Parameters

Item	Options	Comments
Graphics Card	EGA compatible, MATROX (CVU), MATROX (TISTAR), TIGA (TISTAR), VGA compatible	
Hard Disk Drive	C, D, E, F, other	Install APT on the hard disk drive. Requires 30 Mbytes. The "other" option cannot be A or B.
Boot Disk Drive	C, D, other	Requires 50 Kbytes. The "other" option cannot be A or B.
Mouse Port	COM1, COM2, COM3, COM4, none, bus	
Mouse Type	None, Mouse Systems-3 button, Microsoft-2 button, TISTAR Trackball	
Download Port	COM1, COM2, none	Select the port through which programs are to be downloaded to the controller.
Printer Port	LPT1, LPT2, COM1, COM2, COM3, COM4, none	You must have a serial printer if you use COM1, COM2, COM3, or COM4.
Printer Type	TI850/TI855 and compatibles	Does not print SFC/CFC graphics with a TIGA graphics card. For other graphics cards, see page A-24.
	Epson EX/FX/MX and compatibles	For Epson printers you can the select A4 paper size option (NO, YES).
	HP LaserJet, DeskJet, PaintJet	For Hewlett-Packard printers you can the select A4 paper size option (NO, YES). You can also select the printer resolution option: 75, 150, or 300 DPI for the DeskJet and LaserJet, 90 or 180 DPI for the PaintJet. If you do not see your printer listed, consult the user documentation for your printer to see which drivers that it supports.
	None	
Communication Path	CVU TIWAY Adapter, Direct Connect, TIWAY Host Adapter, 386/ATM Coprocessor Module, TISTAR 1.x or 2.x, H1 (Industrial Ethernet), PCS 3.x and OSx 4.x	Select the means by which your computer communicates with the controller. Refer to Figure A-4 for Series 505 controllers. Refer to Figure A-5 for S5 controllers.
Log Installation to Printer	NO, YES	Sends a report of the installation to a printer
Configure STEP 5 Directory	NO, YES	If you select YES, then enter the complete path to indicate where the STEP 5 program is located. The subdirectory must exist. The default is C:\STEP 5. This selection creates APTS5.BAT and places it in the subdirectory where the APT executable files reside.

Installing APT (continued)

Table A-1 APT Configuration Parameters (continued)

Item	Options		Comments
Configure Directories	NO		Use the default options.
	YES	Directory	
		Executable files: (all APT <i>.EXE</i> files)	Default path is <code>\apt</code> . These files can be shared, e.g., by network users. APT changes the <code>autoexec.bat</code> file to add this line to your DOS path. This is the only change the installation makes in the <code>autoexec.bat</code> file when the default options are selected. ¹
		Common files: (all APT tables, forms, and libraries)	Default path is <code>\apt</code> . These files can be shared, e.g., by network users. APT adds the following subdirectories under the directory that you specify: <code>\lib505</code> , <code>\libmc5</code> , <code>\text</code> , <code>\forms</code> . ¹
		Temporary files: (APT files that are deleted after a session)	Default path is <code>\apt\temp</code> . These files must be unique for each user. For fastest execution by APT, use a RAM disk when possible for the temporary files. ¹
		Database files: (APT diagram tables, hierarchy structure, CFC graphics)	Default path is <code>\apt\database</code> . These files must be unique for each user. ¹
Local ² files: (file structures)	Default path is <code>\apt</code> . These files must be unique for each user. APT adds the following subdirectories under the directory that you specify: <code>\program</code> , <code>\temp</code> , ³ <code>\sys</code> , <code>\pr</code> . ¹		
<p>1 Because of the DOS limit for pathname length, it is recommended that you select short directory names. See Table A-2 for maximum pathname lengths.</p> <p>2 APT creates the selected file structure when APT is installed. If you select directories that are different from the defaults, APT adds environment variables to the <code>\apt\sys\autoexec.bat</code> file.</p> <p>3 The temporary environment variables take precedence over this subdirectory. See Table A-2.</p>			

Copying the Files

When you press **F10** after finishing system configuration, APT begins copying files to the specified hard disk and, if you are using diskettes, prompts you to insert the next diskette at the appropriate time. If APT exists on the disk that you specified during system configuration, you are prompted to confirm the deletion of these APT files with these options.

YES This option deletes the existing APT files, including any APT programs on the drive.

NO If you have already done a partial installation, this option enables you to skip diskettes that have already been installed.

NOTE: Installation may be cancelled at any time by pressing either **Esc** or **F9**.

Setting Computer to APT Mode (DOS)

The installation program makes copies of the **autoexec.bat** and **config.sys** files and names the copies **autoexec.dos** and **config.dos**, respectively. Customized versions of these files, needed to run APT, are also made and named **autoexec.apr** and **config.apr**. Before you can run APT, you need to reboot the computer using the customized files. You can select from these options: **Copy and Reboot** and **Don't Copy**.

NOTE: In order to run, APT must load special graphic drivers that can require up to 90 Kbytes of memory. If your current **config.sys** and **autoexec.bat** do not yield 610 Kbytes of DOS memory, you may not be able to execute APT. Refer to [Appendix C](#) for information on how to configure your **autoexec.apr** and **config.apr** in the **\apt\sys** directory.

Copy and Reboot The installation program replaces **autoexec.bat** with **autoexec.apr** and **config.sys** with **config.apr** and then reboots your computer under the APT environment. To select this option, press **F10**.

Don't Copy The installation program exits to DOS and your computer remains in its original environment. Before you can run APT, you need to execute the **APTMODE** command described below. To select this option, press **Esc**.

Installing APT (continued)

The installation program places these two files into the root directory of your boot drive: **aptmode.bat** and **dosmode.bat**. Whenever you need to set your computer to the APT environment, type **APTMODE** and press **[Enter]**. To set the computer to the environment that you configured before installing APT, type **DOSMODE** and press **[Enter]**.

The **DOSMODE** and **APTMODE** commands work as follow:

- **DOSMODE** replaces **autoexec.bat** with **autoexec.dos** and **config.sys** with **config.dos** and then reboots your computer under the environment that you had configured before installing APT.
- **APTMODE** replaces **autoexec.bat** with **autoexec.apr** and **config.sys** with **config.apr** and then reboots your computer under the APT environment.

Running APT

After you have set the computer to APT mode, type **APT** and press **[Enter]** to run APT.

To input an application, follow the steps outlined in Chapters 1 through 8 of this manual, or load a demo program as described in the next section. You must exit APT to load the demo program. Press **[Esc]** to exit APT.

If errors are encountered when invoking APT, consult the rest of this appendix and [Appendix C](#) to debug your system. The most common errors are that there is not enough conventional memory or that the video drivers are not correctly installed.

Loading an Example Program

After APT installation has successfully completed, you may optionally load an example APT program (*WASH_DEM*) that is described in the *SIMATIC APT Applications Manual*. You can install this example program and the other examples by following these steps:

1. Insert the APT software CD in your CD-ROM drive; or, if you created diskettes from the CD, insert DISK8 in your diskette drive.
2. Ensure that the current DOS “path” environment includes the path to the APT executables directory. For example, if APT is installed at D:\APT, enter the following command:

```
PATH
```

If D:\APT is not included in the list output by **PATH**, add it by entering the following command:

```
PATH=%PATH%;D:\APT
```

-
3. If you are loading *WASH_DEM* from the CD, enter the following command, substituting your diskette drive for E, and your disk drive containing the APT database for D:

```
E:\DISK8\LOADDEMO E:\DISK8\WASH_DEM D:
```

If you are loading *WASH_DEM* from diskette, enter the following command, substituting your diskette drive for A, and your disk drive containing the APT database for D:

```
A:\LOADDEMO A:\WASH_DEM D:
```

NOTE: You must have at least 539K free conventional memory to load this demo program.

The APT database is typically on the drive where you installed the APT software, unless you selected the **Configure Directories** option during installation or reconfigured your system at a later time.

If you have configured APT so that the APT software is installed on multiple drives and/or in non-standard directories, then you must ensure that the environment variables `APT_DB` and `APT_COMMON` exist and are assigned appropriate values. See [Table A-2](#) for more information.

If these environment variables are properly set and you are loading from CD drive G, enter the following command:

```
G:\DISK8\LOADDEMO G:\DISK8\WASH_DEM
```

If you are loading from diskette drive B, enter the following command:

```
B:\LOADDEMO B:\WASH_DEM
```

NOTE: The example programs access direct memory addresses to simulate an actual process. For this reason, warnings are generated the first time that the programs are compiled. On subsequent compiles, no warnings occur unless you choose a forced compile.

Installing APT (continued)

Changing between APT Mode and DOS Mode (DOS)

APT uses graphic drivers that require as much as 90 Kbytes of DOS conventional memory. With these APT drivers loaded, you may be unable to run your other DOS programs. With the drivers loaded for the other DOS programs and for APT, you may have insufficient memory to run APT.

Changing Modes Use the **DOSMODE** and **APTMODE** commands to set up your computer to run either APT or the other DOS programs.

Using One Mode It is not necessary to change modes if either of these conditions is true for your system.

- APT and your other DOS programs can run with all the appropriate drivers loaded, or
- You run only APT on your system.

Refer to [Appendix C](#) for information on sample **config.sys** and **autoexec.bat** with one mode.

Refer to [Appendix D](#) for information on using APT with Windows 95.

Refer to [Appendix E](#) for information on using APT with Windows NT/2000.

Changing to DOS Mode To return to your original system configuration after exiting APT, type **DOSMODE** at the DOS prompt and press **[Enter]**. APT reboots the system.

NOTE: If you change your **config.sys** or **autoexec.bat** files while you are in the DOS mode, copy the corresponding new files to **config.dos** and **autoexec.dos** before you return to APT mode. These files are located in the **\apt\sys** subdirectory.

Changing to APT Mode To change to the APT system environment, type **APTMODE** at the DOS prompt and press **[Enter]**. APT reboots the system.

NOTE: If you change your **config.sys** or **autoexec.bat** files while you are in the APT mode, copy the corresponding new files to **config.apr** and **autoexec.apr** before you return to DOS mode. These files are located in the **\apt\sys** subdirectory.

**Configuring the
CVU/TIWAY
Adapter
(Series 505 Only)
(DOS/Win95 Only)**

For Series 505 controllers, if you have difficulty using the CVU/TIWAY Adapter, you may need to make the following changes to the **config.sys** and **autoexec.bat** files or the **config.apr** and **autoexec.apr** files in the **\apr\sys** directory.

For the **config.xxx** file, add the following line:

```
device=z:\apr\drivers\cvuta.dev
```

where z is the drive on which you installed APT.

For the **autoexec.xxx** file, add the following line:

```
set apr_mode=cvu
```

NOTE: If you use an upper memory manager, you must exclude the region C800-C900.

**Configuring the
386/ATM
(Series 505 Only)
(DOS Only)**

The SIMATIC 386/ATM module ships with a driver that is used to communicate with the controller. This driver, **386atm.exe**, requires 80 Kbytes of conventional memory. If you are having trouble running APT because of memory limitations, you can use the drivers that were previously shipped with the 386/ATM module. These drivers, **bs_386.exe** and **aptds.sys**, are shipped with APT and are located in the **\apr\drivers** subdirectory.

For the **config.xxx** file, add the following lines:

```
device=z:\apr\drivers\bs_386.exe  
device=z:\apr\drivers\aptds.sys
```

where z is the drive on which you installed APT.

For the **autoexec.xxx** file, add the following line:

```
set apr_mode=cvu
```

NOTE: If you use an upper memory manager, you must exclude the region C800-C900.

A.3 Setting DOS Environment Variables

The execution of APT is influenced by several variables defined in the DOS environment. The value for some of these variables is set automatically by APT, or by PCS if you have a PCS system. All the variables can be changed with the DOS **set** command using this syntax:

```
set variable=value
```

Do not put spaces before or after the equals (=) sign in the set command. Do not have trailing blanks after the word `value`.

To clear the variable, type the entry without the value after the equals mark. Refer to your DOS user manual for a detailed description of the DOS environment and how the **set** command operates.

The environment variables used by APT are listed in [Table A-2](#). An example with a description of the fields follows.

Variable: Variable name.

Description: Description of the functions provided by the variable.

Value: Values that can be assigned to a variable. A value of **Yes** means that the variable must simply be present in the DOS environment and can have any value. For example, when you define `NO_TBP` with the command `set no_tbp=yes`, then the transparent binary protocol is not used to communicate with a controller that is connected by a direct link.

Default: Value used by APT when the variable is not present.

Set at APT Install? Indication that the variable is automatically set by the APT installation program based on your choices during installation. **Yes** means that the variable is defined during installation. **No** means that you have to define the variable with a DOS **set** command.

If you want to define any environment variables for your system, you can include the **set** statements in the `config.sys` and `autoexec.bat` files or the `config.apr` and `autoexec.apr` files in the `\apt\sys` directory.

Table A-2 DOS Environment Variables

Communication Variables				
Variable	Description	Values	Default	Set at APT Install?
APT_BAUD	Sets the baud rate at which the APT computer and the controller communicate. If you have trouble establishing communications with the controller, use the APT_BAUD variable to set the baud rate to the same value that the controller is using. If you continue to have trouble, set both the controller and APT_BAUD to successively lower values.	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200, 600, 300	Automatic, based on the baud rate of the controller.	No
APT_H1_INTNUM	Used to change the software interrupt by which APT communicates with the CP1413.	0x00-0xFF	0x82	No

This table is continued on the next page.

Setting DOS Environment Variables (continued)

Table A-2 DOS Environment Variables (continued)

Communication Variables					
Variable	Description	Values		Default	Set at APT Install?
APT_MODE	Sets method of communication that you are using.	DIRECT	APT computer is connected to the controller by means of an RS-232 or RS-422 (Series 505) or current loop (S5) interface.	The default is DIRECT unless APT is called from TISTAR, in which case the default is TISTAR. If communication with PCS is selected, default is PCS.	Yes
		TISTAR (Series 505 only)	APT is installed on a TISTAR operator station, or communication between the APT computer and the TISTAR Model 20 system is by a TISTAR Communications Card (PFS-3619).		
		CVU (Series 505 only)	APT computer is connected to the controller via the CVU TIWAY Adapter (2587716-8011), or APT is installed on the 386/ATM Coprocessor Module.		
		HOST (Series 505 only)	APT computer is connected to the controller via the UNILINK Host Adapter.		
		H1	APT computer is connected to the controller via the Industrial Ethernet network using the CP1413 or CP1613 card.		
		PCS	APT computer is connected to the controller and the PCS/OSx configuration node via the Industrial Ethernet network using the CP1413 or CP1613 card.		

This table is continued on the next page.

Table A-2 DOS Environment Variables (continued)

Communication Variables				
Variable	Description	Values	Default	Set at APT Install?
APT_OS	For Windows NT/2000 only. When this variable is set, APT uses one of the APT communications servers (APTH1SRV or APTPCSRV)	NT, 2000	NT	Yes
APT_ROOT	For Windows NT/2000 only. Contains the path to the APT executables	Path specifier	C:\apt	Yes
APT_PCS_NODE	Used to select the name of the PCS/OSx configuration node. This is the node to which APT sends all tag data during a tag translation.	ASCII string of up to 8 characters (must be upper case)	PCS_CNFG	No
APT_PORT	Sets the communication port on the APT computer when the APT_MODE is DIRECT or HOST.	1, 2	1	Yes
APT_RETRIES	Sets the number of times that APT attempts to communicate with the controller after a communication error is detected. Note that not all communications are re-sent, since some information can corrupt the controller when it is received twice.	0 to 32767	3	No
APT_SPEED (Series 505 only)	Sets the number of task codes to be packed into one communications packet. Set this variable to lower values for early hardware releases if they are overloaded by the large amount of data transmitted at the default value. This reduces throughput somewhat, but reduces communications errors. This variable is applicable only when APT_MODE is TISTAR, PCS, CVU, or H1.	1, 2, 3, 4, 5, 6, 7	7	No
APT_TIMEOUT	Sets the number of seconds that APT waits for a response from the controller before timing out or retrying a communication.	1 to 32767	15	No
APT_TSCAN	Limits the number of taskcodes that will be processed per controller scan during program download.	1 to 8	controller default value	No

This table is continued on the next page.

Setting DOS Environment Variables (continued)

Table A-2 DOS Environment Variables (continued)

Communication Variables				
Variable	Description	Values	Default	Set at APT Install?
BD (Series 505 only)	Sets the baud rate at which the APT computer and the TISTAR DEU communicate. Note: This variable is set automatically when APT is called from a TISTAR menu.	19200, 9600, 4800, 2400, 1200, 600, 300	19200	No
NO_TBP (Series 505 only)	When this variable is set in the DOS environment, then the transparent binary protocol (TBP) is not used to communicate with a controller that is connected via a Direct link. Some automatic (smart) switch boxes do not support the TBP. Because communication is slower when this variable is set, use the default "not present." This variable is applicable only when APT_MODE is Direct. To reset NO_TBP, from the DOS prompt type: SET NO_TBP= This makes NO_TBP Not Present.	Yes	Not Present	No
PA (Series 505 only)	Sets the parity for communications between the APT computer and the TISTAR DEU. Note: This variable is set automatically when APT is called from a TISTAR menu.	N(one), E(ven), O(dd)	N(one)	No
PO (Series 505 only)	Sets the port used by the APT computer to communicate with the TISTAR DEU. Note: This variable is set automatically when APT is called from a TISTAR menu.	1-9	7	No
File Locations				
Variable	Description	Values	Default	Set at APT Install?
APT_COMMON	Sets the path for these APT subdirectories: FORMS , LIB505 , LIBMC5 , and TEXT . These directories contain read-only data used by APT. You can use this environment variable to split APT files over several drives for networking, RAM disks, etc. In Windows NT/2000, these are set in the System Variables area (Appendix E).	User-selectable. Length of variable cannot exceed 34 characters.	\apt	Yes ¹
APT_LOCAL	Sets the path for these APT subdirectories: PROGRAM , TEMP , PRR , and SYS . You can use this environment variable to split APT files over several drives for networking, RAM disks, etc.	User-selectable. Length of variable cannot exceed 19 characters.	\apt	Yes ¹
1 If you choose a path different from the default, this variable is set during installation.				

Table A-2 DOS Environment Variables (continued)

File Locations (continued)				
Variable	Description	Values	Default	Set at APT Install?
APT_TEMP TMP TEMP	<p>Each of these three environment variables can be used to set the path where APT can create temporary files. For example, if you have a RAM disk configured for your APT computer, then for faster program execution consider using the following command to write all temporary files to the RAM disk: SET APT_TEMP = H:\TEMP where H:\ is the RAM drive.</p> <p>APT looks for APT_TEMP first. If APT_TEMP is not present, then APT looks for TMP, then TEMP, and finally APT_LOCAL\TEMP. You can use these environment variables to split APT files over several drives for networking, RAM disks, etc.</p> <p>Note: APT_TEMP should not be set to the root directory of any drive.</p>	User-selectable for each subdirectory. Length of variable cannot exceed 63 characters.	\apt\temp	Yes ¹
APT_DB	Sets the path where the APT database is located. You can use this environment variable to split APT files over several drives for networking, RAM disks, etc.	User-selectable for each subdirectory. Length of variable cannot exceed 63 characters.	\apt\database	Yes ¹
CGIPATH	Indicates to the GSS environment where the files <i>CGI.CFG</i> and <i>S5H1_CFG.TXT</i> are located.	User-selectable. Length of variable cannot exceed 63 characters.	X:\apt where X indicates where APT is loaded.	Yes
PATH	Sets the path where the APT executable files are located. You can use this environment variable to split APT files over several drives for networking, RAM disks, etc.	User-selectable	c:\apt; . . . ²	Yes
TM_LOG	Defines the name of the log file in which APT writes unusual or unexpected errors.	User-selectable	\apt\temp\tmstdout.log	No

¹ If you choose a path different from the default, this variable is set during installation.
² c:\apt is added to the end of the existing path.

Setting DOS Environment Variables (continued)

Table A-2 DOS Environment Variables (continued)

Graphic Environment				
Variable	Description	Values	Default	Set at APT Install?
BLACK	Sets the RGB components of the color black. You can use this variable to improve the contrast on your monitor.	0-999 0-999 0-999	0 0 0	No
GREY	Sets the RGB components of the color grey. You can use this variable to improve the contrast on your monitor.	0-999 0-999 0-999	500 500 500	No
WHITE	Sets the RGB components of the color white. You can use this variable to improve the contrast on your monitor.	0-999 0-999 0-999	999 999 999	No
MOUSE	Identifies the type of mouse being used: Mouse Systems or Microsoft.	MOUSESYS MSMOUSE	MOUSESYS	Yes
MOUSESYS	Identifies the serial communications port used by the Mouse Systems mouse.	COM1, COM2, COM3, COM4, NONE	COM1	Yes
NOMOUSE	Indicates that no mouse is to be used and no mouse device driver is installed. You can remove the mouse driver to increase available DOS memory. The mouse driver must be installed for you to use a mouse. To reset NOMOUSE, from the DOS prompt type: SET NOMOUSE= This makes NOMOUSE Not Present.	Yes	Not present — Mouse is present.	Yes

This table is continued on the next page.

Table A-2 DOS Environment Variables (continued)

Printing				
Variable	Description	Values	Default	Set at APT Install?
PAPER	Sets the paper size for report generation and printouts. Set this environment variable to ISOA4 for European page sizes. The default, not present, indicates that US letter size is used (8.5 " × 11 ").	ISOA4	Not present — US letter size is used.	Yes
PRINT_SCREEN	<p>Enables the Print Screen key. Normally this key is disabled to prevent the accidental print of a graphic screen display. To reset Print_Screen, from the DOS prompt type</p> <pre>SET PRINT_SCREEN=</pre> <p>This makes PRINT_SCREEN Not Present.</p> <p>Note: The graphic print screen driver must be loaded prior to executing APT for DOS. For Windows 95, the driver is not necessary, because Print Screen copies the screen to the clipboard, from which it can be pasted into Paint and then printed.</p>	Yes	Not present — Print Screen key is disabled.	No
RESOLUTION	Sets the printer resolution in dots per inch (DPI) for Hewlett-Packard printers. A resolution of 75 DPI provides the fastest printing.	75 150 300	75	Yes
LINES_PER_PAGE	Sets the lines to print per page for all text reports, after an initial blank line.	20 to 32767	59 (64 if paper size is ISOA4)	No

A.4 Changing the APT Graphic Configuration

CGI.CFG File

When you install APT, APT creates a **cgi.cfg** file in the `\apt` directory. This file instructs which APT graphic drivers should be loaded. Below is an example of the **cgi.cfg** file configured for a VGA graphics card, Hewlett-Packard Laserjet printer, and Microsoft mouse. Reboot your computer after changing **cgi.cfg**.

```
CGI.CFG

; video card driver
driver=c:\apt\drivers\ibmvga12.sys /g:display
;
If not using printer or mouse, these can be
commented out to reduce
the size of your drivers. {
; printer driver
driver=c:\apt\drivers\laserjet.sys /g:display
;
; mouse driver
driver=c:\apt\drivers\msmouse.sys /g:input
;
; gss*cgi graphics driver
driver=c:\apt\drivers\gsscgi.sys /r
```

To change to an EGA graphic card, replace `ibmvga12` with `ibmega`. To change to a Mouse Systems mouse, replace `msmouse` with `mousesys` and include the following environment variables in **autoexec.bat** or **autoexec.apr** in the `\aptsys` directory.

```
SET mouse=mousesys
SET mousesys=comx
```

where `x` is the number of the communication port that the mouse is installed on.

NOTE: For the MS mouse, the configuration for the serial port is the same as the configuration for the mouse port.

To change to an Epson printer, replace `laserjet` with `epsonx`. To change to a Hewlett-Packard DeskJet, replace `laserjet` with `hpdjet`. To change to a Hewlett-Packard PaintJet at 90 DPI resolution, replace `laserjet` with `hppj90`. To change to a Hewlett-Packard PaintJet at 180 DPI resolution, replace `laserjet` with `hppj180`. Refer to [Table A-2](#) for printer environment variables.

Configuring a TI855 printer to print SFC and CFC graphics requires a different approach. Refer to the next section for information on configuring a TI855 printer.

Printer Support

The following printers are supported by APT:

- Epson EX/FX/MX and compatibles
- HP DeskJet and compatibles
- HP LaserJet and compatibles
- HP PaintJet and compatibles
- TI850/TI855 and compatibles

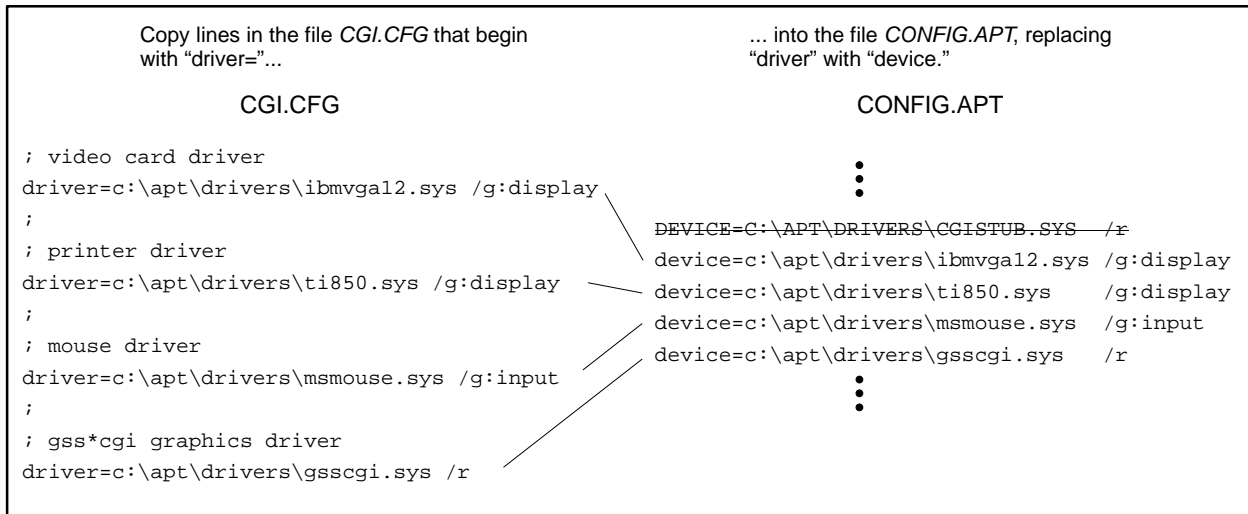
APT does not support a TI855 printer in the **cgi.cfg** file for SFC and CFC graphic printouts. If you need to use a TI855, you must make some changes in the **config.sys** or **config.apr** file.

Examine the file **cgi.cfg**. Make a copy of every line that is preceded by **driver=** and place it in the **config.sys** or **config.apr** file, replacing the term **driver** with **device**. Also delete the following line from the file:

```
device=c:\apt\drivers\cgistub.sys /r
```

NOTE: The mouse driver is required even if you do not use a mouse.

For an example of how to revise the **config.sys** or **config.apr** file, refer to the example below.



A.5 Hardware Connections

Port Pinouts

The port pinouts for RS-232/C communications between the programming device and a Series 505 controller are shown in [Figure A-1](#), [Figure A-2](#), and [Figure A-3](#).

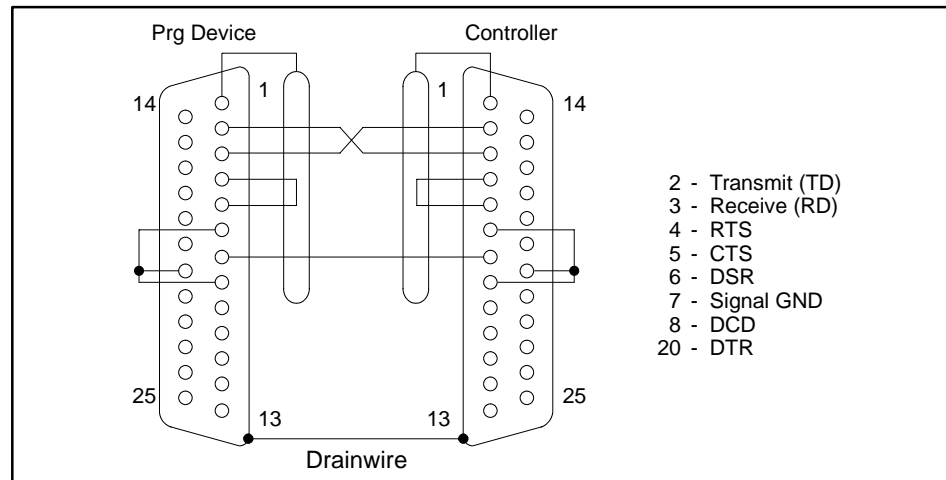


Figure A-1 RS-232/C 25-pin to 25-pin Connectors

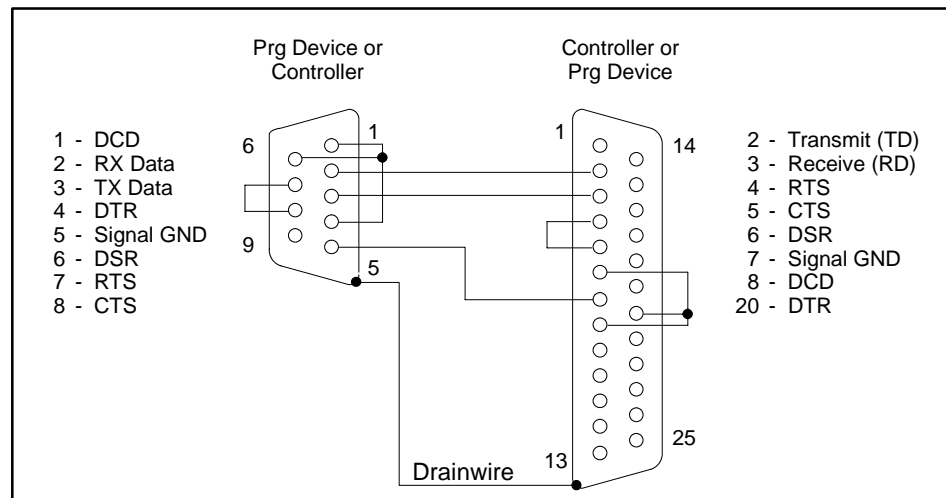


Figure A-2 RS-232/C 9-pin to 25-pin Connectors

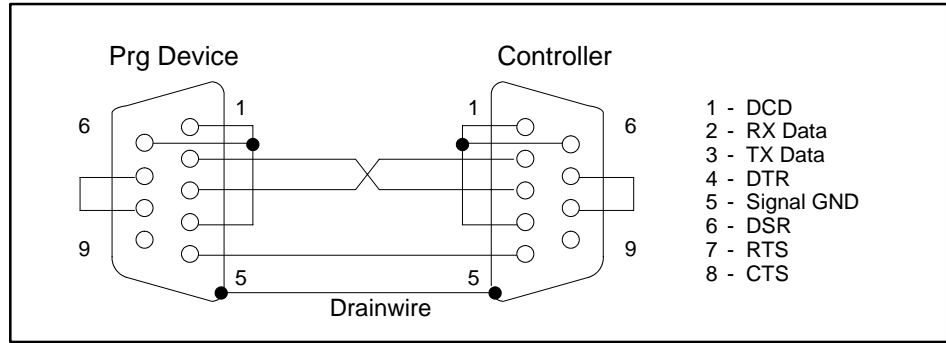


Figure A-3 RS-232/C 9-pin to 9-pin Connectors

NOTE: Use a shielded cable for the connection to your programming device to prevent electrical noise interference in communications.

Hardware Connections (continued)

S5 Communication Links

[Figure A-5](#) illustrates the means for linking your APT computer to an S5 controller. The cables needed to make the connections are identified below.

- Cable 1 15-pin ethernet cable (6ES5 727-1BD20)
- Cable 2 3.2m/10ft programming cable (6ES5 734-1BD20)
- Cable 3 3.2m/10ft 15-pin ethernet cable (6ES5 727-1BD20)
- Cable 4 0.9m/3ft drop cable (6ES5 725-0AK00)
- Other cable lengths available are:

6ES5 727-1BD20 = 3.2m/10ft
6ES5 727-1CB00 = 10m/33ft
6ES5 727-1CB50 = 15m/49ft
6ES5 727-1CC00 = 20m/65ft
6ES5 727-1CD20 = 32m/105ft
6ES5 727-1CF00 = 50m/164ft

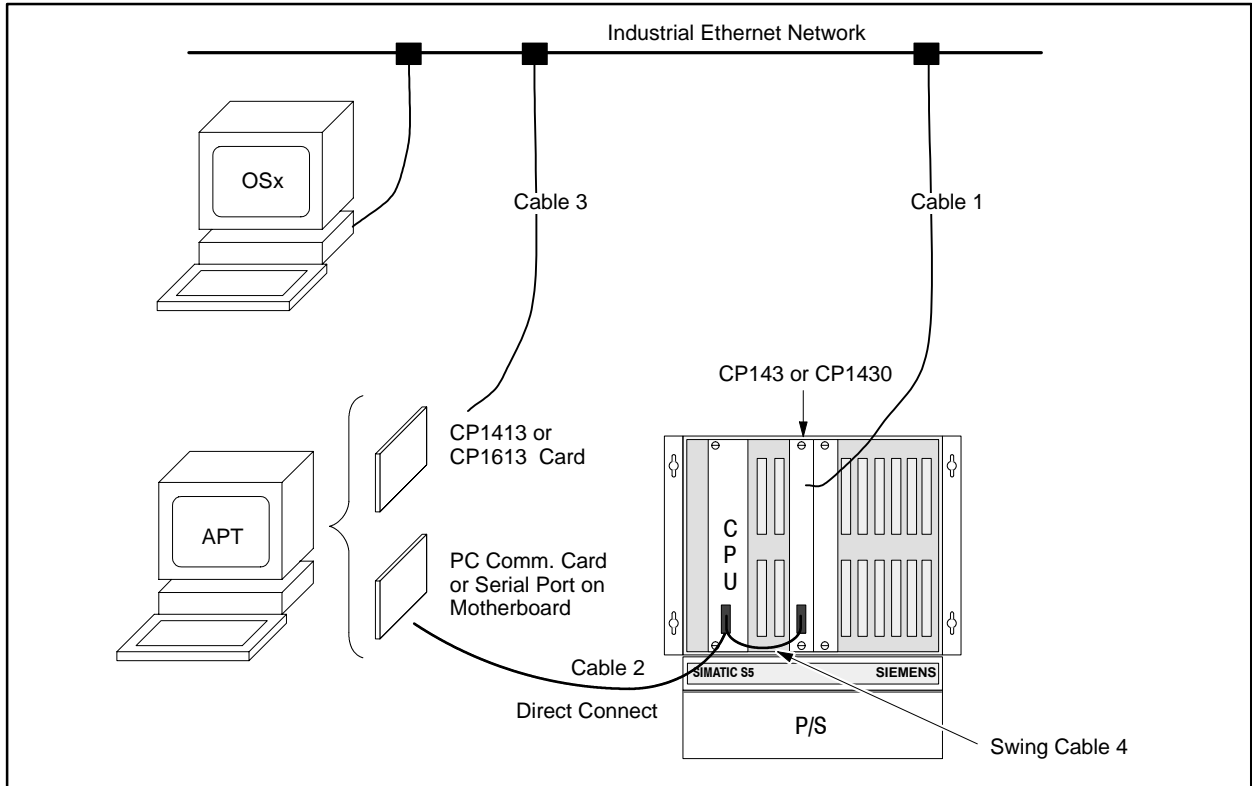


Figure A-5 APT/S5 Controller Communication Link

APT on an Industrial Ethernet Network

B.1	Overview	B-2
B.2	Planning the Network Configuration for Series 505	B-3
	Assigning Node Names	B-3
	Setting Ethernet Addresses	B-4
	Setting Logical Network Connections	B-5
	Assigning TSAPs	B-6
	Gathering Configuration Data for Each Node	B-7
B.3	Planning the Network Configuration for S5	B-9
	Assigning Node Names	B-9
	Setting Ethernet Addresses and Paths	B-10
	Setting Logical Network Connections	B-12
B.4	Connecting APT to the Industrial Ethernet Network (DOS/Windows 95)	B-14
	Beginning Hardware Configuration	B-14
	Configuring the CP1413 Card (Series 505 and S5)	B-14
	If Configuration Fails (Series 505 and S5)	B-15
	Installing a Memory Manager (Series 505 and S5)	B-16
	Verifying CP1413 Configuration (Series 505 and S5)	B-17
B.5	Connecting APT to the Industrial Ethernet Network (Windows NT/2000)	B-18
	Before You Begin	B-18
	Beginning Hardware Configuration	B-18
	Loading the SIMATIC NET Software	B-18
	Defining the Point of Access for the CP1613 Card	B-20
	Testing the CP1613 Card Installation	B-21
	Installing Service Packs	B-21
B.6	Configuring the Network	B-22
	Network Configuration (Series 505 Only)	B-22
	Creating a TF Service with the H1 Configurator (Series 505 Only)	B-24
	Creating a Matching Association with COML (Series 505 Only)	B-26
	Network Configuration (S5 Only)	B-28
	APT Configuration (DOS only)	B-30
	APT Configuration (NT/2000 Only)	B-31
	Testing the APT Configuration	B-32
	Memory Usage and the APT Drivers (DOS only)	B-33
	Loading and Starting the APT H1 Server (NT/2000 Only)	B-35
	Specifying the H1 Driver for APT	B-36
	APT H1 Server Registry Values	B-37
	APT H1 Server Error Messages	B-40
	Using APT with the Industrial Ethernet Network	B-41

B.1 Overview

The configuration required to connect an APT computer to an Industrial Ethernet (formerly called SINEC H1) network is not difficult or technically challenging. However, you must take extreme care with the details. You must configure all parameters correctly, or problems may occur. Before starting, review the following documents:

- *SIMATIC TI505 SINEC H1 Communications Processor User Manual*
- *SIMATIC PCS 7 OSx Interface to S5 Controllers Manual*
- *TF-NET 1413/MS-DOS Windows Manual*
- *SIMATIC PCS 7 OSx System Administration Manual*
- *CP1430 TF with NCM COM 1430 TF Manual*
- *TF-NET 1613/ Windows NT/2000 Manual*

Before you can begin configuration, you must become familiar with the physical topology of your network. [Figure B-1](#) shows a simple Industrial Ethernet network containing Series 505 and S5 controllers, an APT engineering node, and an OSx station.

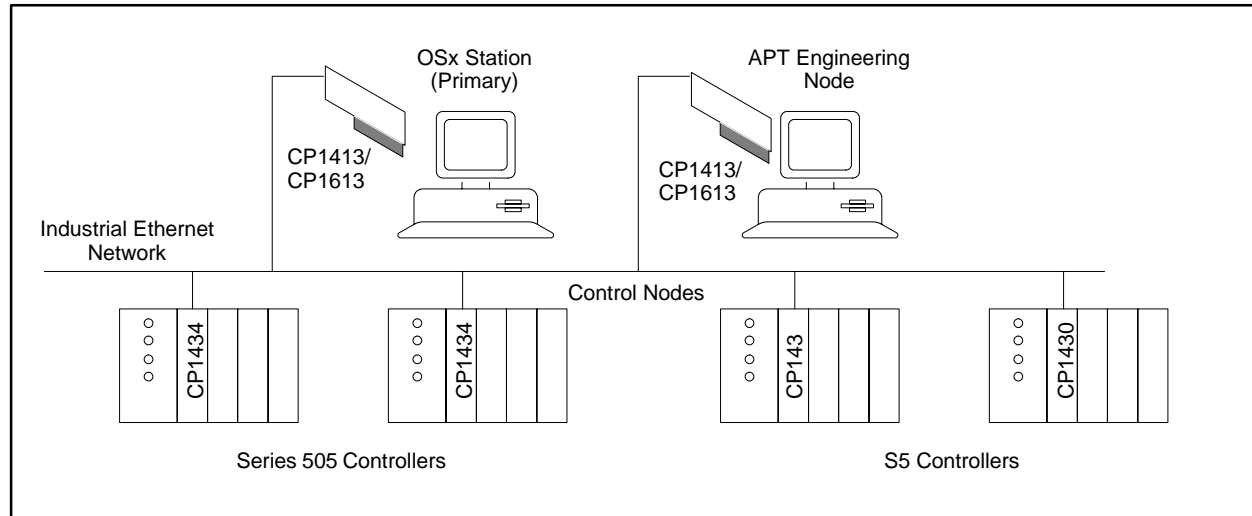


Figure B-1 Industrial Ethernet Network with APT, OSx Station, and S5 and Series 505 Controllers

B.2 Planning the Network Configuration for Series 505

Assigning Node Names

This section describes how to plan the network configuration for Series 505 controllers. For S5 controllers, see [Section B.3](#). The first step in configuration is to give a meaningful name to each node on the network. A node is defined as a piece of hardware that is physically connected to the network. A node can be a controller, such as the 555. A node can also be a personal computer. [Table B-1](#) lists the node names and descriptions and [Figure B-2](#) illustrates them.

Table B-1 Example Node Names and Descriptions for Series 505

Node Name	Description
APT1	OSx Engineering Node with APT
OSx1	OSx Station (must be the OSx primary)
PLC1	555 controller number 1
PLC2	555 controller number 2
PLC3	555 controller number 3
PLC4	555 controller number 4

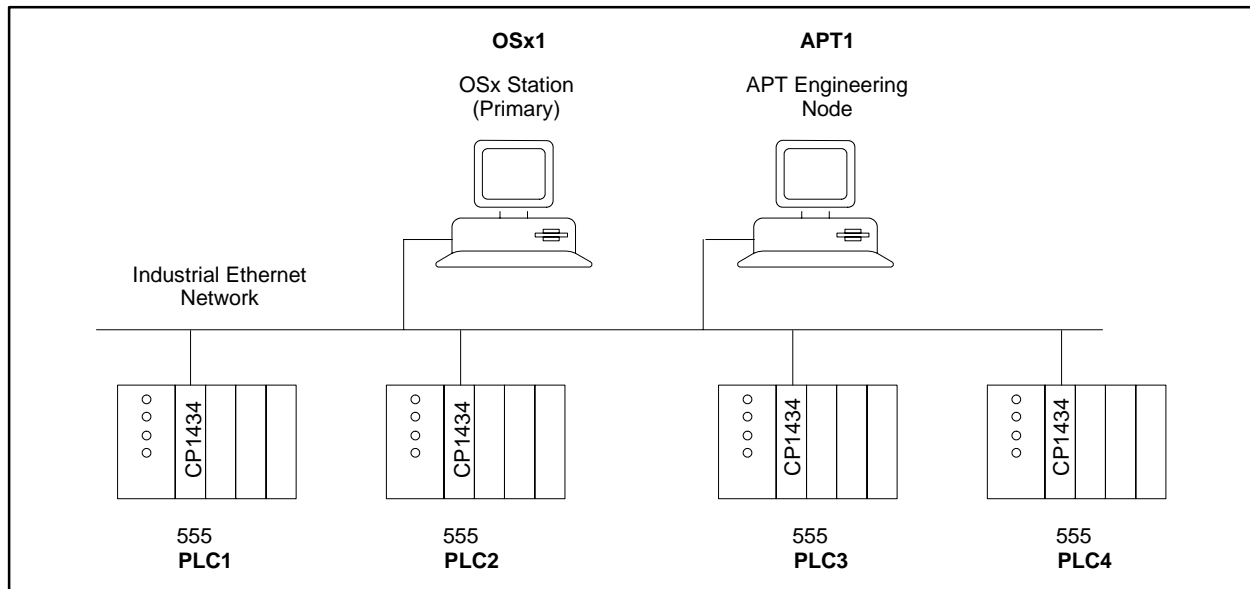


Figure B-2 Network Node Names for Series 505

Planning the Network Configuration for Series 505 (continued)

Setting Ethernet Addresses

After you have chosen a name for each node, assign a unique ethernet address for each node (Table B-2 and Figure B-3).

⚠ WARNING
Using the same ethernet address more than once on the same network can cause unpredictable controller behavior.
Unpredictable controller behavior could cause death or serious injury to personnel, and/or damage to equipment.
Ensure that each ethernet address on the network is unique.

Table B-2 Example Node Names and Addresses for Series 505

Node Name	Description	Address
APT1	HP Vectra XU 5/90 running APT	08.00.06.01.00.01
OSx1	HP Vectra XU 5/90 running OSx	08.00.06.01.00.02*
PLC1	555 controller number 1	08.00.06.01.00.03
PLC2	555 controller number 2	08.00.06.01.00.04
PLC3	555 controller number 3	08.00.06.01.00.05
PLC4	555 controller number 4	08.00.06.01.00.06

* Must be address of the OSx primary.

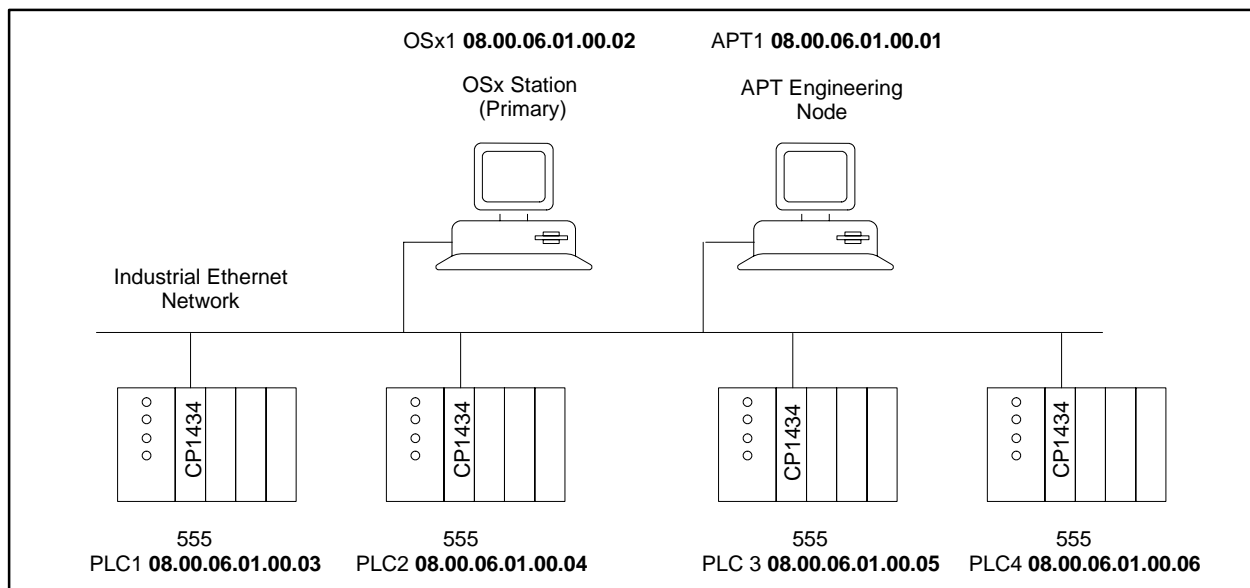


Figure B-3 Example Network Node Addresses for Series 505

Setting Logical Network Connections

Next, determine the logical layout of the network. The previous figures and tables describe the physical layout of the network. The logical layout describes the communications between nodes (Table B-3 and Figure B-4). Table B-4 simplifies the associations shown in Table B-3. OSx1 and APT1 can communicate with one another and with all the controllers. However, the controllers can only communicate with OSx1 and APT1, not with one another.

Table B-3 Full Listing of Nodes and Associations

Node Name	Association				
APT1	OSx1	PLC1	PLC2	PLC3	PLC4
OSx1	APT1	PLC1	PLC2	PLC3	PLC4
PLC1	APT1	OSx1			
PLC2	APT1	OSx1			
PLC3	APT1	OSx1			
PLC4	APT1	OSx1			

Table B-4 Reduced Listing of Nodes and Associations

Node Name	Association				
APT1	OSx1	PLC1	PLC2	PLC3	PLC4
OSx1		PLC1	PLC2	PLC3	PLC4

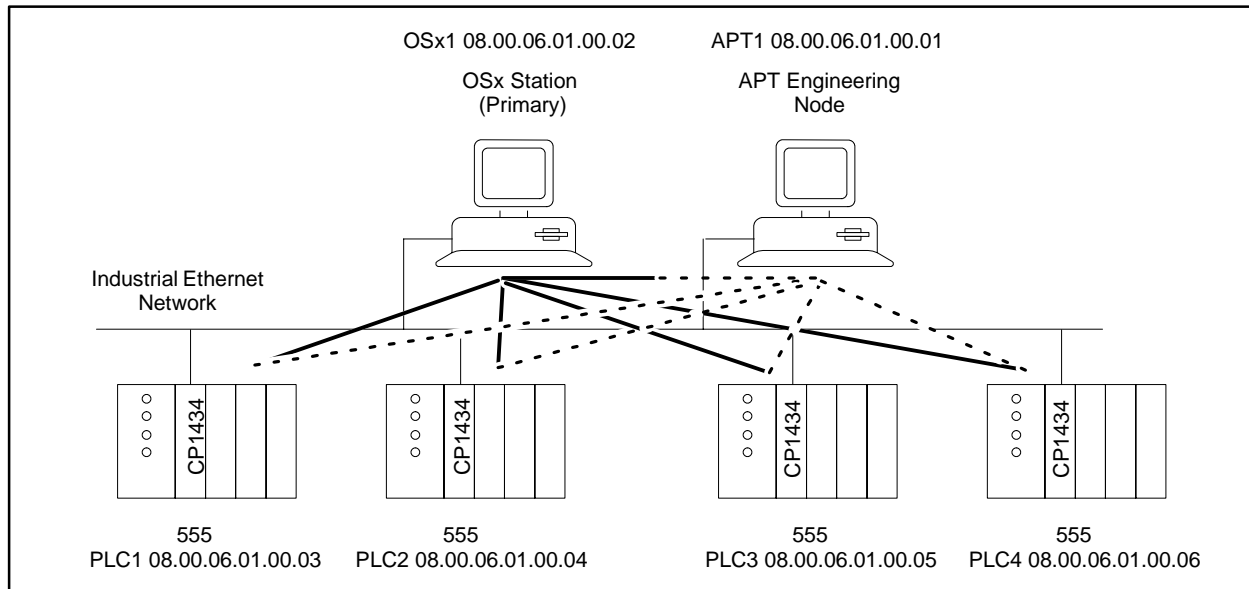


Figure B-4 Logical Layout for OSx with Series 505 Controllers

Planning the Network Configuration for Series 505 (continued)

Assigning TSAPs

To create the logical connections, you must load information about each association into the CP1413/CP1613 cards and the CP1434TF modules connected to the network. Organize the information about each association into a common table (Table B-5) and include the node identifier, its Transport Service Access Point (TSAP). See Section B.4 for more information about assigning TSAPs to nodes.

Table B-5 Node TSAPs

Association	Node Name	Address	Association Name	Local TSAP	Remote TSAP
APT1 and OSx1 ¹	APT1	08.00.06.01.00.01	OSx_CNFG ²	APT1****	OSx1****
	OSx1	08.00.06.01.00.02	APT1	OSx1****	APT1****
APT1 and PLC1	APT1	08.00.06.01.00.01	PLC1	APT1****	PLC1****
	PLC1	08.00.06.01.00.03	APT1	PLC1****	APT1****
APT1 and PLC2	APT1	08.00.06.01.00.01	PLC2	APT1****	PLC2****
	PLC2	08.00.06.01.00.04	APT1	PLC2****	APT1****
APT1 and PLC3	APT1	08.00.06.01.00.01	PLC3	APT1****	PLC3****
	PLC3	08.00.06.01.00.05	APT1	PLC3****	APT1****
APT1 and PLC4	APT1	08.00.06.01.00.01	PLC4	APT1****	PLC4****
	PLC4	08.00.06.01.00.06	APT1	PLC4****	APT1****
OSx1 and PLC1	OSx1	08.00.06.01.00.02	PLC1	OSx1****	PLC1****
	PLC1	08.00.06.01.00.03	OSx1	PLC1****	OSx1****
OSx1 and PLC2	OSx1	08.00.06.01.00.02	PLC2	OSx1****	PLC2****
	PLC2	08.00.06.01.00.04	OSx1	PLC2****	OSx1****
OSx1 and PLC3	OSx1	08.00.06.01.00.02	PLC3	OSx1****	PLC3****
	PLC3	08.00.06.01.00.05	OSx1	PLC3****	OSx1****
OSx1 and PLC4	OSx1	08.00.06.01.00.02	PLC4	OSx1****	PLC4****
	PLC4	08.00.06.01.00.06	OSx1	PLC4****	OSx1****
<p>1 Must use the OSx primary for this association.</p> <p>2 If this is changed you must change the DOS environment variable, APT_PCS_NODE.</p>					

Gathering Configuration Data for Each Node

After you have noted all the node names, addresses, TSAPs, etc., for the network, group this information by node. Then, you need to enter the information for each node and its associations as described below.

APT Node Table B-6 shows the information for the APT node. Use the Configuration Management Local (COML) software described in Section B.4 to enter this information.

Table B-6 APT Node Configuration Data

Node Name and Address	Association	Local TSAP	Remote TSAP	Remote Address
APT1 08.00.06.01.00.01	PCS_CNFG	APT1****	PCS1****	08.00.06.01.00.02
	PLC1	APT1****	PLC1****	08.00.06.01.00.03
	PLC2	APT1****	PLC2****	08.00.06.01.00.04
	PLC3	APT1****	PLC3****	08.00.06.01.00.05
	PLC4	APT1****	PLC4****	08.00.06.01.00.06

OSx Node Table B-7 shows the information for the OSx station. This must be the OSx primary. Refer to the *SIMATIC PCS 7 OSx System Administration Manual* for detailed information about how to configure the OSx station.

Table B-7 OSx Station Configuration Data

Node Name and Address	Association	Local TSAP	Remote TSAP	Remote Address
OSx1 08.00.06.01.00.02	APT1	PCS1****	APT1****	08.00.06.01.00.01
	PLC1	PCS1****	PLC1****	08.00.06.01.00.03
	PLC2	PCS1****	PLC2****	08.00.06.01.00.04
	PLC3	PCS1****	PLC3****	08.00.06.01.00.05
	PLC4	PCS1****	PLC4****	08.00.06.01.00.06

Controller Nodes Table B-8 shows the information for the controller nodes. Use the 505-CP1434TF H1 Configurator software described in Section B.4 to enter this information.

Table B-8 Controller Configuration Data

Node Name and Address	Association	Local TSAP	Remote TSAP	Remote Address
PLC1 08.00.06.01.00.03	APT1	PLC1****	APT1****	08.00.06.01.00.01
	PCS1	PLC1****	PCS1****	08.00.06.01.00.02
PLC2 08.00.06.01.00.04	APT1	PLC2****	APT1****	08.00.06.01.00.01
	PCS1	PLC2****	PCS1****	08.00.06.01.00.02
PLC3 08.00.06.01.00.05	APT1	PLC3****	APT1****	08.00.06.01.00.01
	PCS1	PLC3****	PCS1****	08.00.06.01.00.02
PLC4 08.00.06.01.00.06	APT1	PLC4****	APT1****	08.00.06.01.00.01
	PCS1	PLC4****	PCS1****	08.00.06.01.00.02

B.3 Planning the Network Configuration for S5

Assigning Node Names

This section describes how to plan the network configuration for a group of S5 controllers. S5 controllers differ from Series 505 controllers in that APT communicates to them using a virtual connection. No association needs to be programmed into the CP in order to facilitate APT communications. The first step in configuration is to give a meaningful name to each node on the network. A node is defined as a piece of hardware that is physically connected to the network. A node can be a controller, such as the 928B. A node can also be a personal computer. [Table B-9](#) lists the node names and descriptions and [Figure B-5](#) illustrates them.

Table B-9 Example Node Names and Descriptions for S5

Node Name	Description
APT1	PCS Engineering Node with APT
PCS1	PCS Operations Node (must be the PCS primary)
PLC1	928B controller number 1
PLC2	948R controller number 2
PLC3	948U controller number 3

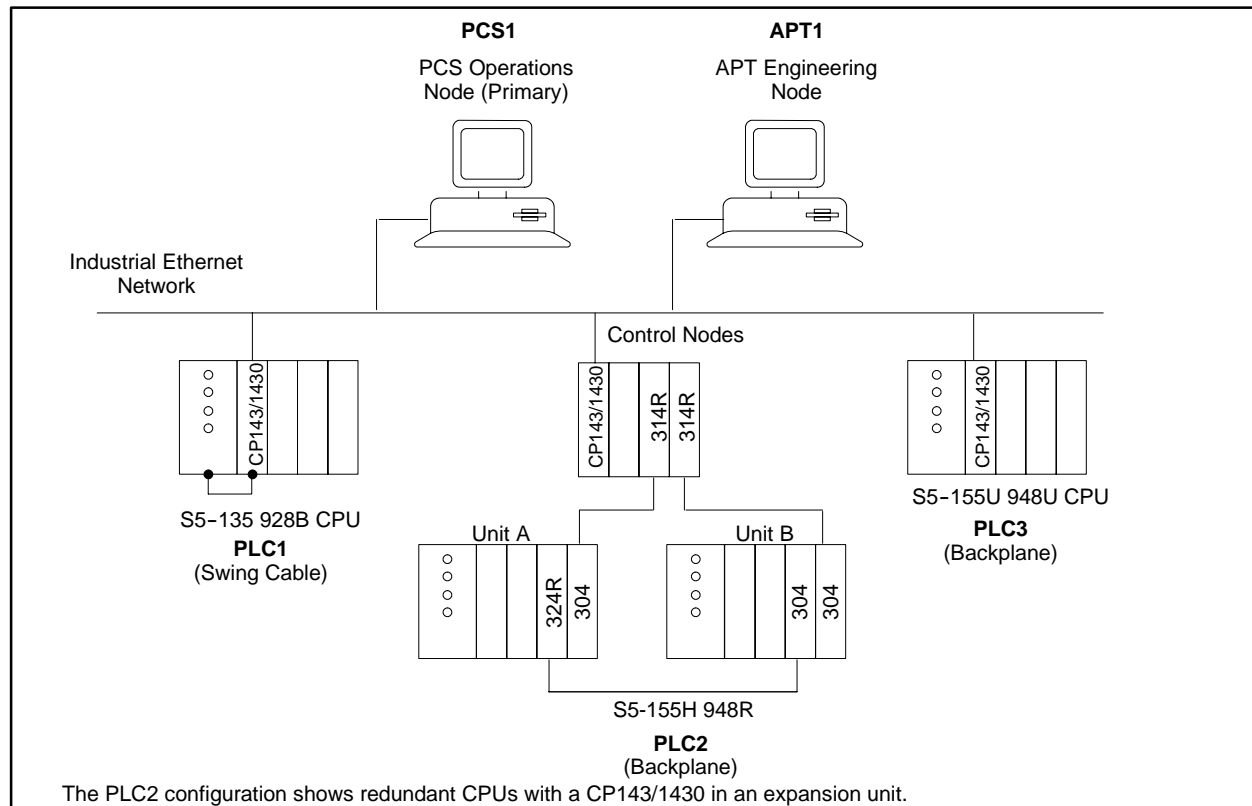


Figure B-5 Network Node Names for S5

Planning the Network Configuration for S5 (continued)

Setting Ethernet Addresses and Paths

After you have chosen a name for each node, assign a unique ethernet address and select a path for each node (Table B-10 and Figure B-6).

WARNING

Using the same ethernet address more than once on the same network can cause unpredictable controller behavior.

Unpredictable controller behavior could cause death or serious injury to personnel, and/or damage to equipment.

Ensure that each ethernet address on the network is unique.

There are only two valid paths for APT to S5 communications: backplane and swing. If you choose backplane, communications between the CP and the controller occur over the controller backplane. If you choose swing, communications take place through the swing cable. The swing cable connects the programming port on the CP to the programming port on the controller. Consult your CP143/CP1430 manual for more details.

Table B-10 Example Node Names, Addresses, and Paths for S5

Node Name	Description	Address	Path
APT1	PCS Engineering Node with APT	08.00.06.01.00.01	N/A
PCS1	PCS Operations Node	08.00.06.01.00.02*	N/A
PLC1	928B controller number 1	08.00.06.01.00.03	Swing
PLC2	948R controller number 2	08.00.06.01.00.04	Backplane
PLC3	948U controller number 3	08.00.06.01.00.05	Backplane

* Must be address of the PCS primary.

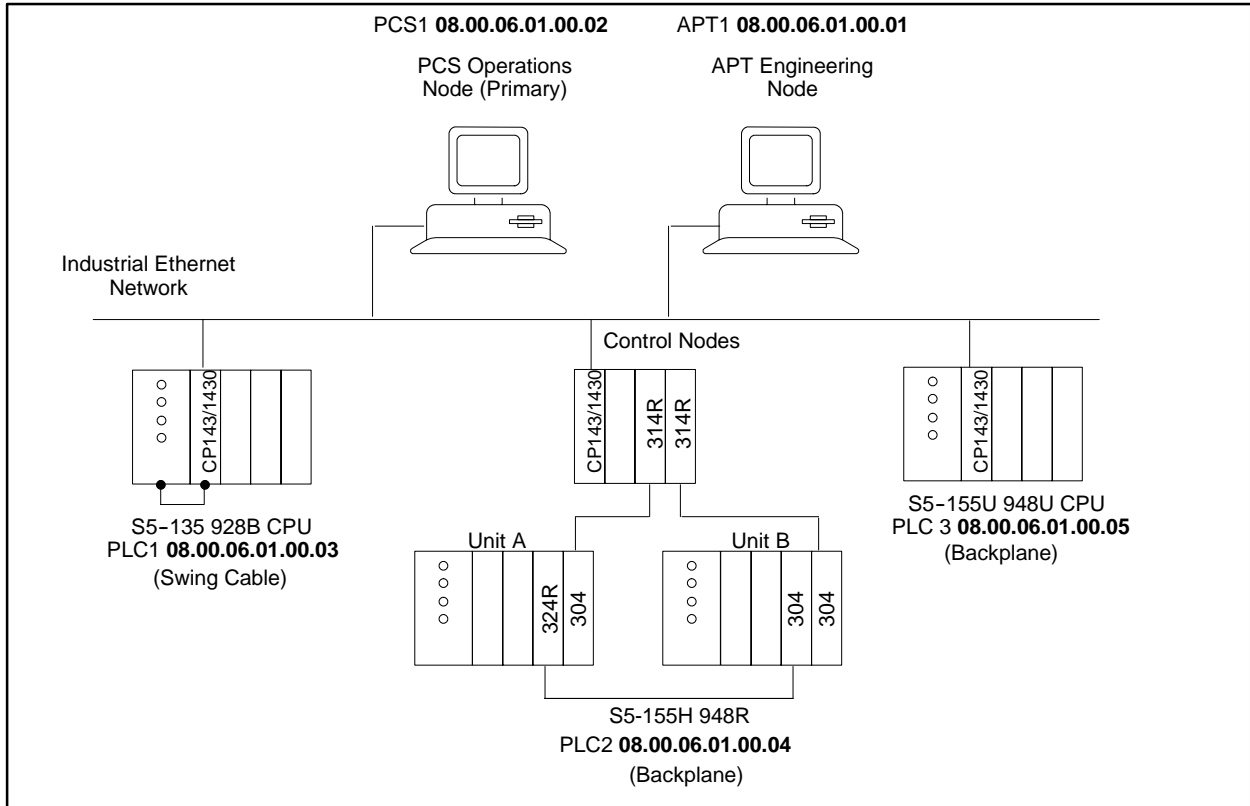


Figure B-6 Example Network Node Addresses for S5

Planning the Network Configuration for S5 (continued)

Setting Logical Network Connections

Next, determine the logical layout of the network. The previous figures and tables describe the physical layout of the network. The logical layout describes the communications between nodes ([Table B-11](#) and [Figure B-7](#)). [Table B-12](#) simplifies the associations shown in [Table B-11](#). PCS1 and APT1 can communicate with one another and with all the controllers. However, the controllers can only communicate with PCS1 and APT1, not with one another.

Table B-11 Full Listing of Nodes and Associations

Node Name	Association					Path
APT1	PCS1	PLC1	PLC2	PLC3	PLC4	N/A
PCS1	APT1	PLC1	PLC2	PLC3	PLC4	N/A
PLC1		PCS1				Swing
PLC2		PCS1				Backplane
PLC3		PCS1				Backplane

Table B-12 Reduced Listing of Nodes and Associations

Node Name	Association/Path			
APT1	PCS1	PLC1/Swing	PLC2/Backplane	PLC3/Backplane
PCS1		PLC1	PLC2	PLC3

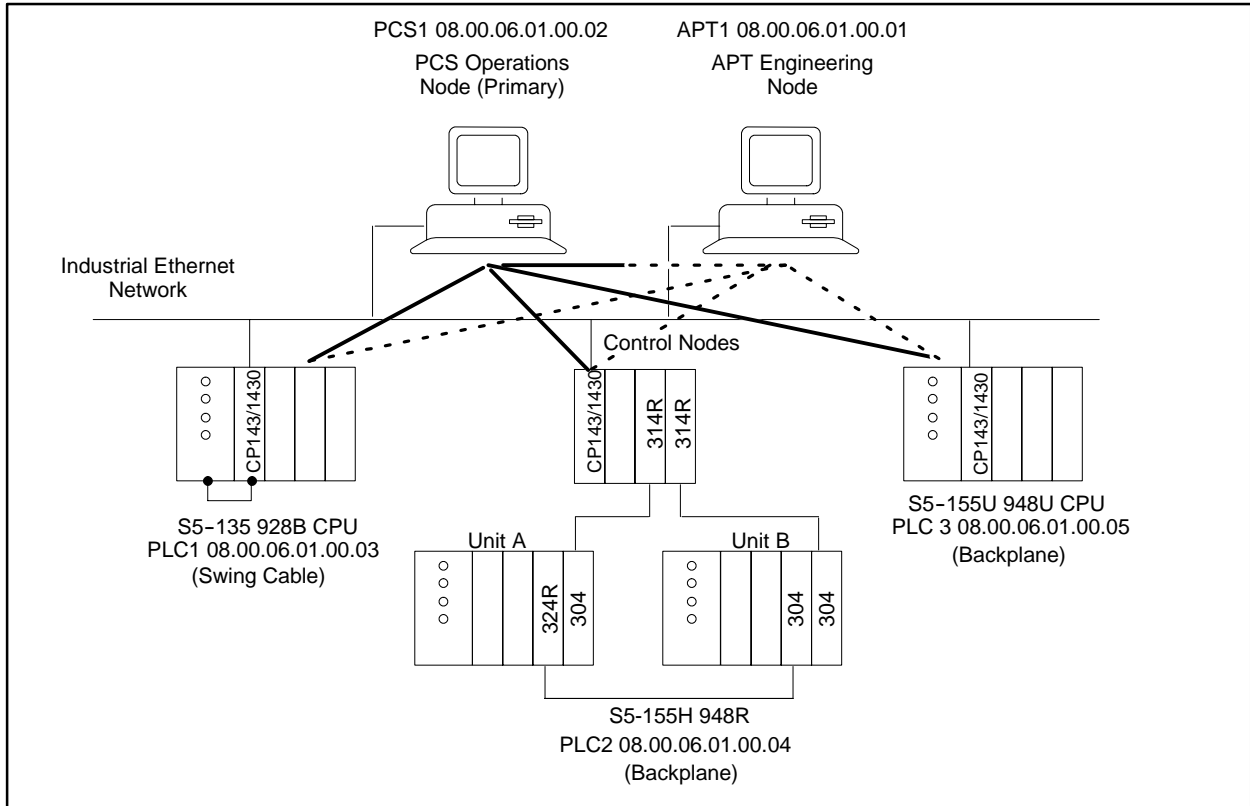


Figure B-7 Logical Layout for PCS with S5 Controllers

B.4 Connecting APT to the Industrial Ethernet Network (DOS/Windows 95)

Beginning Hardware Configuration

APT access to the Industrial Ethernet (H1) network is only possible under DOS or Windows 95 if you use a CP1413 card and the Siemens Technological Functions library. Purchase the card and the library at the same time. To order a CP1413 card and the DOS/Windows English-language version of the STF library, specify part number 6GK11413AE00. If you intend to use these libraries under Windows 95, Release 1.7 of the TF-NET1413 software is required.

To begin, install the CP1413 card in an AT slot in the host computer for APT. Follow the procedures outlined in the Hardware Installation chapter (Chapter 2) of the *SINEC TF-NET1413/MS-DOS, Windows Manual*.

Configuring the CP1413 Card (Series 505 and S5)

Once you have installed the card, run the INSTALL utility located on the first TF-NET1413 disk provided with your card. This batch program loads all software onto your hard drive and then runs `netinst.exe`. The NETINST utility generates a series of prompts that you follow in order to configure the CP1413 card correct. Table B-13 lists selections that work for most installations.

Table B-13 CP1413 Configuration Prompts

Screen	# / Description	Select
1	Installation mode	Detailed mode
2	Number of installed boards	New installation
3	Which board to install	CP1413
4	Layer 2 communications	No
5	Layer 4 communications	Yes
6	Layer 7 communications	Yes
7	Dual-port RAM address	0D0000H-64K
8	Ethernet address	08.00.06.01.xx.xx ¹
9	AT bus interrupt	INT 10
10	Configuration register	03E0H
11	Local database	\sinec\data\startup.LDB
1	The letters xx.xx represent the last four digits of the unique ethernet address that you assigned to the APT engineering node.	

The NETINST utility creates the directory `\sinec` on the selected hard drive. This directory holds the associated configuration files and the drivers needed to communicate over the Industrial Ethernet network.

For further information about running these utilities, consult the Software Installation chapter of the *SINEC TF-NET1413/MS-DOS, Windows Manual*.

**If Configuration
Fails
(Series 505 and S5)**

If the default configuration operates incorrectly, verify that the dual-port RAM address and the AT bus interrupt are correct for the host computer.

Dual-Port RAM Address This parameter specifies an area of high memory that will be used by the CP1413 card. It is possible that this area of memory is already in use by another piece of hardware in the system. Network cards and SCSI controllers often require an area of high memory; check them for conflict with the CP1413. If it is not possible to change the address of a conflicting card, then you will have to change the dual-port RAM address of the CP1413 to 0E0000H.

AT Bus Interrupt The CP1413 card needs exclusive use of one hardware interrupt in order to function. This parameter specifies that address. Occasionally, another device, such as a mouse or a multi-media card, uses this interrupt. In most instances, changing the interrupt from 10 to 12 works. Two other options are INT 5 and INT 15.

Configuration Register The I/O address must be one that is not currently in use on your computer. If 03E0H is being used, try 0390H, 03E8H, or 0100H.

For changes to your configuration to take effect, you must run `c:\sinec\bin\netinst.exe` and reconfigure your system, and then power cycle your computer. It is not enough to reboot; you must power cycle.

Connecting APT to the Ethernet Network (DOS/Windows 95) (continued)

Installing a Memory Manager (Series 505 and S5)

In order to run APT with the Industrial Ethernet network, you need to install a memory manager. APT has been successfully configured with H1 using CEMM, QEMM, and DOS 6.x MEMMAKER. When you edit your **config.sys** to install the memory manager of your choice, you must include a memory exclusion command which identifies the region you have set aside for use by the CP1413 card; the memory manager cannot have access to the region of memory reserved for the CP1413.

If your memory manager does not provide an installation program, then use an ASCII text editor to install the memory manager of your choice in your **config.sys**. [Figure B-8](#) shows the appropriate instruction for installing CEMM, QEMM, or DOS 6.x MEMMAKER, in that order. The memory exclusion command for each line is represented by the bolded text.

```
rem Compaq DOS 5.0
device = c:\dos\cemm.exe 384 noems romcompress x = d000-dfff

rem Quarterdeck expanded memory manager
device = c:\qemm\qemm386.sys ram frame = none exclude = d000-dfff

rem EMM386 from DOS 6.0/Windows 95
device = c:\dos\emm386.exe noems x = d000-dfff
```

Figure B-8 Three Memory Manager Setup Examples

NOTE: After you modify the **config.sys** file to install the memory manager, power cycle the system to verify that it will boot up correctly.

**Verifying CP1413
Configuration
(Series 505 and S5)**

To confirm that you have successfully accomplished the configuration of your CP1413 card and the installation of your memory manager, you can generate a textual database and use it to test the system.

1. Display the `\sinec\com` directory. This directory contains the default database description H1.TXT, which was installed along with the TF-NET 1413 software. You can use the CVH1TF utility to create a sample local database (LDB) from the `h1.txt` database description.

From the DOS command prompt, enter:

```
cvh1tf h1.txt \sinec\data\startup.ldb
```

2. Next, change to the `\sinec\bin` directory.

To start up the CP card, enter:

```
startcp.bat
```

If the system is correctly configured, the messages **CP ok** and **LDB downloaded** appear on the screen. If a problem is reported, or the system locks up, it is probably due to a conflict in either the dual-port address or the AT bus interrupt, which you can correct. See [page B-15](#) for troubleshooting tips.

3. If the system did not lock up, and no errors were displayed on the screen, you can continue. From the DOS command line at the `c:>sinec/bin` prompt, enter:

```
scp_mon -s
```

This returns a box describing the status of the CP1413 card. Verify that the database is consistent and that all tasks are running.

NOTE: For further information about starting the module and the driver, and using the SCP monitor, consult the Software Installation chapter (Sections 3.2 and 3.3) of the *SINEC TF-NET1413/MS-DOS, Windows Manual*.

B.5 Connecting APT to the Industrial Ethernet Network (Windows NT/2000)

Before You Begin

Before you install the CP1613 card or load the SIMATIC NET software on your system, make sure that you have Administrator privileges under your current logon. Consult your computer user manuals or contact your MIS department for help in verifying or setting Administrator access on your computer.

Beginning Hardware Configuration

APT access to the Industrial Ethernet (H1) network under Windows NT and Windows 2000 is only possible if you use the CP1613 card and the SIMATIC NET TF-1613 software drivers. Purchase the card and the software at the same time. To order, you must specify part numbers for both the CP1613 card (6GK11613AA00) and the TF-1613 for Windows NT/2000 software (6GK17161TB213AA0). The minimum CP1613 software version required for use with APT is Version 2.1 + SP1.

To install the CP1613 card in a PCI slot in your computer, follow the installation instructions that are included with the card. Installation of the NDIS drivers is not required and is not recommended when using the CP1613 card with APT.

Loading the SIMATIC NET Software

Once you have installed the card in your computer, you must load the TF-1613 and PG-1613 software from your SIMATIC NET CD. Follow these steps:

1. Insert the CD in your CD-ROM drive. If installation does not begin automatically, execute **SETUP.EXE** from the root directory of the CD. When setup begins, the SIMATIC NET Software splash screen is displayed.
2. Select the **Install Software SIMATIC NET** option. A list of SIMATIC products appears.
3. Select **IE PG-1613** and **IE TF-1613**, and follow the rest of the installation instructions. Note that the release number of these software products is Version 2.1. After you complete this installation, you must install the service pack (SP1). The service pack is usually provided on one or more diskettes.

When the installation is complete, the setup utility starts the PC/PG Interface Configurator. This software utility is used in Windows NT and 2000 to configure and debug the communications processors and their software drivers.

Before you can configure the CP1613, you must add the CP1613 card to the list of recognized interfaces. Follow these steps:

1. Click **Select** on the PC/PG Interface Configurator to bring up the Installing/Uninstalling Interfaces dialog box (Figure B-9).
2. Click **CP1613** in the Selection list in the left window.
3. Click the **Install-->** button. The CP1613 appears in the right window.
4. Click **Close**.

The CP1613 card is now recognized as PC/PG interface.

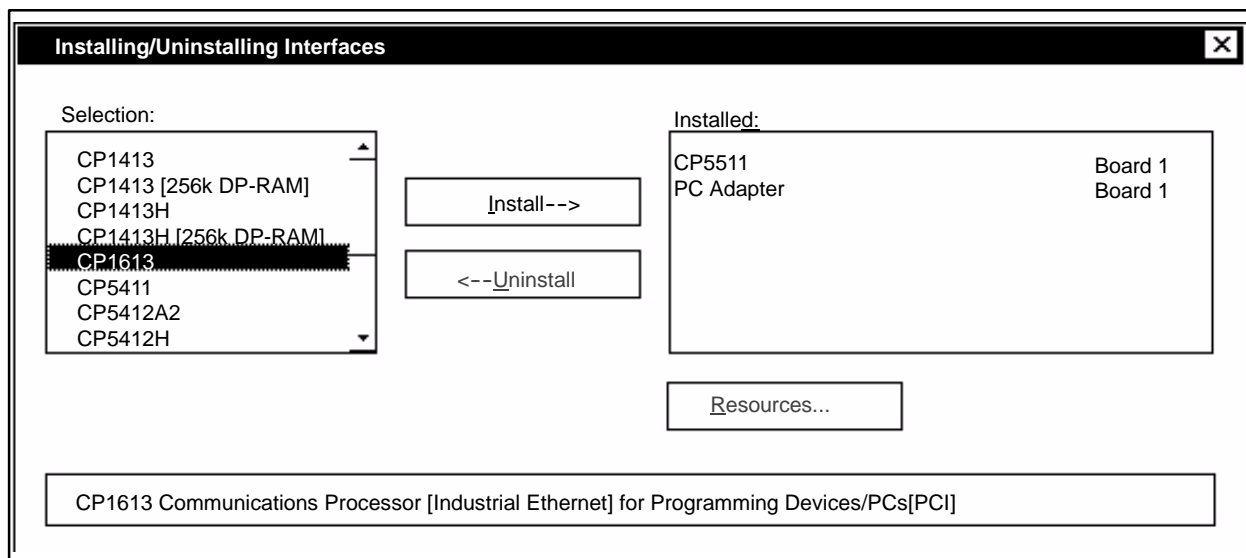


Figure B-9 Adding the CP1613 Card to the PG/PC Interface List

Connecting APT to the Ethernet Network (Windows NT/2000) (continued)

Defining the Point of Access for the CP1613 Card

Once the CP1613 is known to the system, you must define its point of access. Follow these steps:

1. Click the completion aid button to the right of the Access Point of Application field on the PC/PG Interface Configurator and select **CP_H1_1**.
2. In the Interface Parameter Assignment Used field, select **CP1613(ISO)**. The entry **CP_H1_1: -> CP1613(ISO)** should now appear in the Access Point of the Application field (Figure B-10).

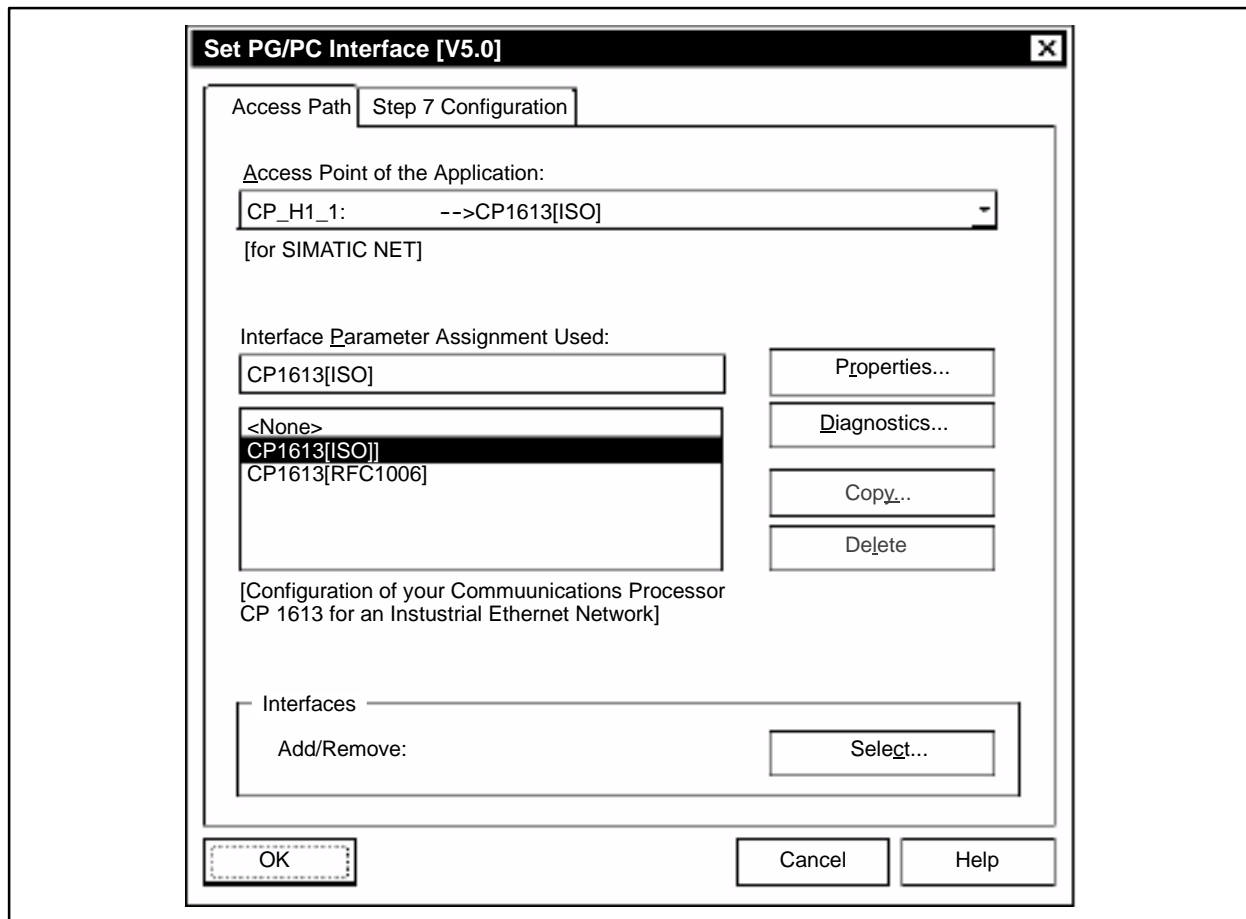


Figure B-10 Defining the Point of Access for the CP1613 Card

Testing the CP1613 Card Installation

To ensure that you have installed and configured the CP1613 card correctly, follow these steps:

1. On the Set PG/PC Interface dialog box, click **Diagnostics**. The SIMATIC NET Diagnosis CP1613 dialog box appears.
2. Select the **IE Net Diagnosis** tab (Figure B-11).
3. Click the **Test** button. You should see an OK indication in the window immediately to the right of the **Test** button. If you do not, refer to the installation instructions provided with the CP1613 card to diagnose the problem.
4. Click **OK** to exit the Diagnostics dialog box.
5. Click **OK** again to exit the PG/PC Interface Configurator.

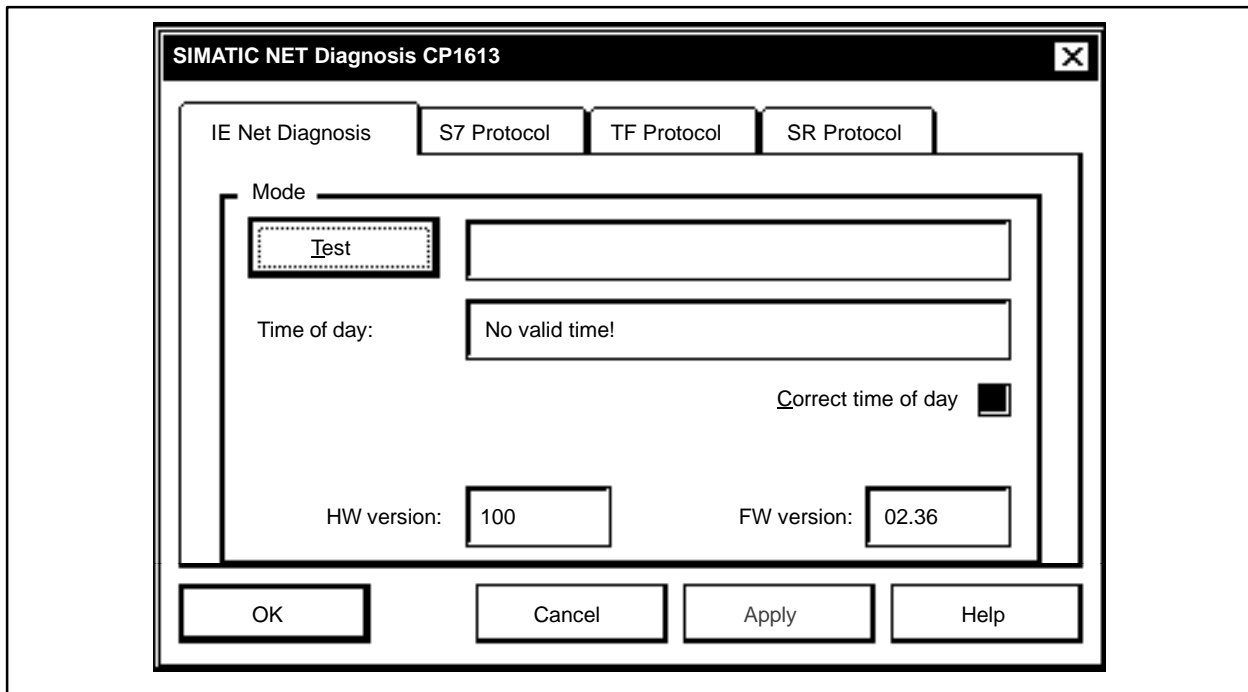


Figure B-11 Testing the CP1613 Card

Installing Service Packs

If the software you ordered has an additional service pack that must be applied, it will be provided on one or more diskettes. Place the first diskette into your drive and execute Setup. Follow the instructions in Setup.

B.6 Configuring the Network

Network Configuration (Series 505 Only)

When the CP card and all required DP card software have been installed, you can proceed to configure Series 505 controllers and PCS nodes on the network. You need two software tools: the 505-CP1434TF H1 Configurator, which is shipped with the CP1434TF modules, and Configuration Management Local (COML), which is automatically installed when you run INSTALL to set up the CP card. Both of these items run under Windows.

Before APT can communicate with the Communications Processor (CP1413/CP1613), the associations on the CP card and the CP1434TF module must be correctly configured. The 505-CP1434TF H1 Configurator allows you to create a Technology Function (TF) Service for the CP1434TF module; you can then use COML to create a matching association for the CP1413 and CP1613 cards.

NOTE: If you have not yet installed the 505-CP1434TF H1 Configurator, you must do so before continuing. Consult the chapter on Installing the Software in the *SIMATIC TI505 SINEC H1 Communication Processor User Manual* for the proper procedure to follow.

In order for one node to communicate with another, each node must be configured to recognize the other one. When you create associations, you give each node an identifier (its Transport Service Access Point, or local TSAP) and you tell it the identifier (remote TSAP) of the other node.

[Figure B-12](#) and [Figure B-13](#) are examples of associations created between a CP1434TF module and a CP1413/CP1613 card.

For the purpose of the example figures, assume that you have made the decisions shown in [Table B-14](#) about the layout of your network:

Table B-14 Single Node Configuration

CP1434TF Module		CP1413/CP1613 Card	
Node Name	PLC1	Node Name	APT1
Ethernet Address	08.00.06.01.00.03	Ethernet Address	08.00.06.01.00.01
TSAP	PLC1****	TSAP	APT1****

TF Service

Association Name:

Association Profile: ▾

Local

TSAP Length: HEX: ASCII:

Multiplexing Address:

Remote

Ethernet Address:

TSAP Length: HEX: ASCII:

Figure B-12 Association for a CP1434TF Module

SINEC COM1 1413TF -(unnamed)

File Help

Node name: Station address: CP type: ▾

Application association list

PLC1

Delete

Edit application association selected on left

Name:

Remote Addr:

Local TSAP: hex:

Remote TSAP: hex:

Server ID:

Connection establ. active passive

Connection type. static dynamic

PDU size: ▾

IPDU size: ▾

Figure B-13 Matching Association for a CP1413/CP1613 Card

■ Creating a TF Service with the H1 Configurator (Series 505 Only)

The first association that you create is for the CP1434TF module, allowing it to recognize and communicate with the CP1413/CP1613 card. Follow these steps to create a TF Service for your CP1434TF module:

1. Select the H1 icon from Windows. The 505-CP1434TF H1 Configurator window opens.
2. From the drop-down File menu, select **New** to create a new configuration file which will hold the association that you are about to create.
3. The Local Ethernet Address box appears. Assign an address to your CP1434TF module.
4. To generate a TF Service, select the TF box and then select the **Insert** button. A TF Service window opens.
5. Fill out the following fields and boxes in the TF Service window (Figure B-12):

■ **Association Name** Enter the name of the node that the module is communicating with (that is, the node name of the CP1413/CP1613 card).

Association Profile Select the drop-down selection box and select **APROF_0510_MDT**.

Local This box allows you to assign a Transport Service Access Point (TSAP) to the node being configured, the CP1434TF module.

TSAP Length Accept the default (8).

HEX This will change to reflect whatever you put in the ASCII box.

ASCII The TSAP must be exactly 8 characters long. For purposes of identification and tracking, use the node name you have assigned to the CP1434TF module as a portion of your TSAP.

Multiplexing Address Accept the default (00).

Remote This box allows you to assign a Transport Service Access Point (TSAP) to the node you wish to communicate with, i.e., the CP1413/CP1613 card. ■

Ethernet Address Enter the ethernet address that you assigned to the CP card when you configured it. ■

TSAP Length Accept the default (8).

HEX This changes to reflect whatever you put in the ASCII box.

ASCII The TSAP must be exactly 8 characters long. For purposes of identification and tracking, use the node name you have assigned to the CP card as a portion of your TSAP. ■

6. Close the TF Service window (press **OK**). This returns you to the 505-CP1434TF H1 Configurator window.
7. Assign a meaningful file name. To distinguish this association from others you may create for your CP1434TF module, you can include the node name of the CP card as a portion of the filename. **Save** the file. ■
8. To download the association you have just configured to the CP1434TF module, pull down the Transfer menu from the 505-CP1434TF H1 Configurator window and select **Download Configuration to CP**.
9. To leave the H1 Configurator and return to the main screen in Windows, pull down the File menu from the 505-CP1434TF H1 Configurator window, and select **Exit**.

NOTE: You can obtain additional information about using the 505-CP1434TF H1 Configurator and creating associations by consulting the *SIMATIC TI505 SINEC H1 Communication Processor User Manual*. See the chapters on Communication Overview and Creating and Editing Configuration Files (Chapters 4 and 5) and the appendix on reference data (Appendix A).

Configuring the Network (continued)

■ Creating a Matching Association with COML (Series 505 Only)

The next association that you create is for the CP card, allowing it to recognize and communicate with the CP1434TF module. Follow these steps to create an association for your CP card that matches the TF Service that you just filled out on the CP1434TF:

1. Select the COML icon from Windows. This opens the COML 1413TF window.

NOTE: COML is named COMTF under Windows NT and Windows 2000.

2. From the drop-down File menu, select **New** to create a new configuration file to hold the association that you want to make.
3. Fill out the following fields in the COML 1413TF window ([Figure B-13](#) on [page B-23](#)).

Node name Use the node name that you have assigned to the CP card in your network.

Station address Enter the ethernet address that you assigned to your CP card when you configured it.

CP type If you are using a CP1413 card, select CP1413TF. If you are using a CP1613 card, select CP1613TF.

Name Enter the name of the node that the card is communicating with (that is, the node name of the CP1434 module).

Remote Address Enter the ethernet address of the node that the card is communicating with (that is, the node name of the CP1434 module).

Local TSAP This must be the same as the Remote TSAP you specified in your TF Service.

Remote TSAP This must be the same as the Local TSAP you specified in your TF Service.

Server ID Enter serv01.

Connection established Accept the default (active).

Connection type Accept the default (static).

PDU size From the drop-down list, select the value 1000.

TPDU size Accept the default (512).

Application association list To distinguish this association from others you may create for your CP card, enter the name of the node that the card is communicating with (that is, the node name of the CP1434 module).

4. You need to store the association you have just created in a database. From the drop-down File menu of your COML window, select **Generate binary DB as....** This opens the Check and Store Binary Database window.
5. At the File Name field, type in `\sinec\data\startup.ldb` and press **OK** to confirm your choice and return to the COML window.
6. To exit COML, select **End** from the drop-down File menu.

Configuring the Network (continued)

■ Network Configuration (S5 Only)

When the CP card and all of the required software have been installed, you can proceed to configure the S5 controllers on the network. Unlike the Series 505 controllers, which require a dedicated connection, communication to the S5 controllers is accomplished using virtual connections. Therefore, you do not have to program a dedicated connection into either the CP card or the CP143/CP1430.

However, each connection must exist in the S5 H1 configuration file, called **s5h1_cfg.txt**. This file contains the name, ethernet address, and path for each association and is located in the APT drivers directory, **\apt\drivers**.

In order for APT to recognize the H1 configuration for S5 controllers, you must also place a copy of the **s5h1_cfg.txt** file in same directory as the **cgi.cfg** file as well. The **cgi.cfg** file is stored by default in the **\apt** directory. Make sure that the two files are in the same directory as indicated by the CGIPATH environment variable used by APT.

NOTE: Since there is not a dedicated connection between APT and the S5 controller, only one APT node at a time can talk to a given S5 controller.

A sample configuration file is created by the APT installation process. This file is in ASCII format and can be edited with any text editor.

Figure B-14 shows a sample S5 H1 configuration file. The fields in this file are described below.

Association This is the name given to the controller. This name is used by APT to identify the controller.

Network Address This is the address of the CP module. This address must be loaded into the module using COM143 or COM1430 TF.

Path This is the path through which communications from the CP143/CP1430 to the controller occurs. Backplane refers to communications across the system backplane to the controller. Swing refers to the swing cable (Siemens part number 6ES5725-0AK00) connected between the programming port of the CP and the programming port of the controller.

```
;This file contains the H1 communications links between APT and
;the S5 controllers located in Building A, Site 14.
;
;
;Association      Network Address      Path      Comments
PLC1              08.00.06.01.00.03    SWING     928B controller 1
PLC2              08.00.06.01.00.04    BACKPLANE 948R controller 2
PLC3              08.00.06.01.00.05    BACKPLANE 948U controller 3
;
;Notes:
;PLC1 controls palletization
;PLC2 controls site heating and cooling
;PLC3 controls polymer production
```

Figure B-14 Sample S5-H1 Configuration File

Configuring the Network (continued)

APT Configuration (DOS only)

After you do hardware and network configuration, you can load APT on the host computer. First, at the DOS command line, run the APT Installation utility. A **Communications type** prompt allows you to select either **PCS 3.x** or **SINEC H1** as the communications path, depending on your needs.

After completing the APT installation, edit the **autoexec.bat** and **config.sys** files. There will not be enough memory to execute APT and the H1 drivers in DOS memory; you must use a memory manager. The memory managers provided with MS-DOS 5.x and later work well. [Figure B-15](#) shows example **autoexec.bat** and **config.sys** files, designed for a Compaq DeskPro 486/66M running Compaq DOS version 5.0.

```
rem ----- CONFIG.SYS for H1 mode
rem ----- COMPAQ DOS 5.0
STACKS = 0,0
BREAK = ON
BUFFERS = 7
FILES = 40
device = c:\dos\himem.exe
device = c:\dos\cemm.exe 384 noems romcompress i=b000-b7ff x=d000-dfff
device = c:\dos\runhi.exe /block=2 c:\dos\vdisk.sys 12000 /E
device = C:\dos\ansi.sys
device = c:\apt\drivers\cgistub.sys /r
shell=\command.com /p /e:1024
dos = high

:: ----- AUTOEXEC.BAT for H1 mode
:: ----- COMPAQ DOS 5.0
@echo off
::
::          SET UP PATH, DISK CACHE
PATH c:\apt;c:\;c:\dos;c:\win;d:\ti
prompt $p$e$g
c:\dos\runhi /block=3 c:\win\smartdrv 16384

::          SET APT ENVIRONMENT, LOAD APT DRIVES
set apt_temp=i:\          :: ramdrive
set apt_mode=H1
set CGIPATH=c:\apt\drivers
c:\apt\drivers\DRIVERS.EXE

::          BRING UP H1 CARD
c:
cd \sinec\bin
c:\dos\runhi /block=2 tfnetdrv.exe
scp_mon -c
scp_mon -l
c:\dos\runhi /block=1 c:\apt\drivers\apthltc.exe 0x82 pl p2

cd \apt
```

Figure B-15 Sample AUTOEXEC and CONFIG Files

APT Configuration (NT/2000 Only)

After you have successfully installed the CP1613 card in your system and your SIMATIC NET software has been loaded and tested, you must specify a database for TF-1613. The TF services are used to communicate with the 505 controllers and OSx. Use the PG/PC Interface Configurator to load and check your TF database. (See the section on Network Configuration for information on creating a database.) Follow these steps:

1. Click **Start->SIMATIC->SIMATIC NET->Setting the PG/PC Interface**.
2. Make sure **CP_H1_1:->CP1613(ISO)** is displayed in the Access Point of the Application field.
3. Click the **Properties** button. The Properties dialog box appears (Figure B-16).

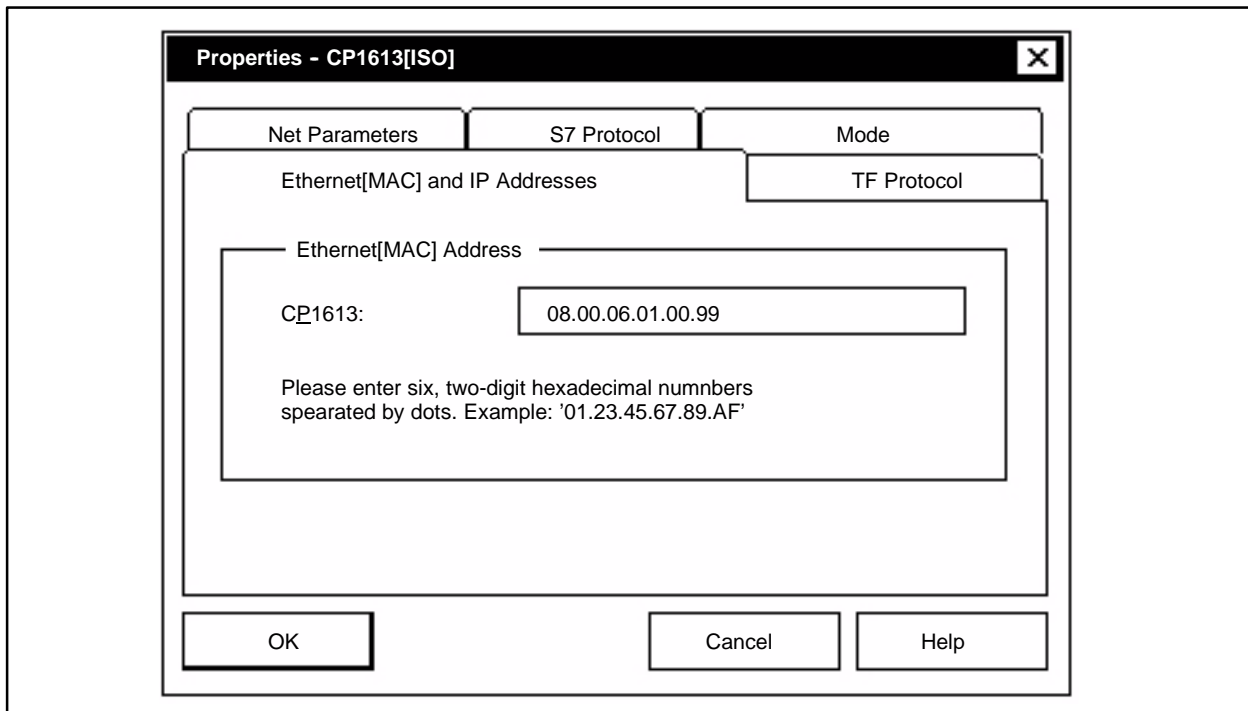


Figure B-16 Configuring the CP1613 Card Properties

4. Select the **Ethernet(MAC) and IP Addresses** tab, and enter the Ethernet address for this card. This address must match the one that you assigned to the CP1613 in COML.
5. Select the **TF Protocol** tab.
6. Select **Activate TF**.
7. Enter the location and name of the TF database that you created using COML. The Browser can help you locate this file.
8. Select the **Mode** tab and click **Restart**. If everything is okay, the message **OK, Restart CP was completed successfully** appears in the status field.
9. Click **OK** to confirm your CP1613 properties configuration.

Testing the APT Configuration

Now that you have specified the database for the TF services, you need to verify that the TF configuration is correct and that the **Send/Receive** services are running. Follow these steps:

1. On the Set PG/PC Interface dialog box, click the **Diagnostics** button.
2. Select the **TF Protocol** tab and click the **Test** button.
3. Verify that all of the associations that you programmed using COML are active. The status of each association is displayed in the window located immediately under the **Test** button.
4. Select the **SR Protocol** tab and click the **Test** button.
5. Verify that the SRMD services are running. The status of the SRMD services is displayed in the window to the right of the **Test** button. If the services are running, you should see the following status report:

```
-->CP_H1_1:  
SRMD_Set[..] OK  
SRMD_Reset[..] OK
```

If you have problems, refer to the TF and Send/Receive user manuals.

6. Click **OK** to exit.

Memory Usage and the APT Drivers (DOS only)

During the APT installation process, the APT H1 driver, named **apth1tc.exe**, is added to your **autoexec** file. If, after attempting to optimize memory, you find that you still have insufficient memory to run APT, the memory requirements of the APT H1 driver can be reduced.

In order to communicate with PCS, Series 505 controllers, and S5 controllers, the driver **apth1tc.exe** must be loaded. If you require communications only with Series 505 controllers, the driver **h1tc.exe** can be substituted for **apth1tc.exe** in your **autoexec** file. This driver requires less memory than **apth1tc.exe**. The **h1tc.exe** driver is shipped with APT and is located in the **\apt\drivers** directory.

In order to use this driver, you must edit your **autoexec.bat** file. For example, to edit the **autoexec** file shown in [Figure B-15](#), follow the steps below.

1. Find the line on which the **apth1tc.exe** driver is loaded. In this sample, that line reads as follows:

```
c:\dos\runhi /block=1 c:\apt\drivers\apth1tc.exe 0x82 p1 p2
```

2. Edit the line to read as follows:

```
c:\dos\runhi /block=1 c:\apt\drivers\h1tc.exe 0x82 p1 p2
```

Note that the parameters required for each driver are identical.

If you cannot substitute **h1tc** because you must communicate with S5 controllers, you can reduce the number of allowed associations by adding a command line parameter. For the example shown in [Figure B-15](#), follow the steps below.

1. Find the line on which the **apth1tc.exe** driver is loaded:

```
c:\dos\runhi /block=1 c:\apt\drivers\apth1tc.exe 0x82 p1 p2
```

2. Edit the line to read as follows:

```
c:\dos\runhi /block=1 c:\apt\drivers\apth1tc.exe -32 0x82 p1 p2
```

This parameter specifies the maximum number of associations that can be open at any given moment. The maximum associations for APT Release 1.7a was 32; this number was increased to 64 and made user configurable.

Table B-15 depicts approximate memory requirements for the APT-H1 drivers loaded with different association options. The listing for APT Release 1.7a is for purposes of comparison only.

Table B-15 Memory Requirements for APT-H1 Drivers

Number of Associations	Memory Required for APTH1TC	Memory Required for H1TC	Memory required for APTH1TC in Rel. 1.7a
64	17,264	12,480	N/A
32	16,208	11,424	16,032
16	15,696	10,896	N/A
8	15,296	10,624	N/A
4	15,296	10,496	N/A

Manually Loading and Starting the APT H1 Server (NT/2000 Only)

When APT Release 1.9A is loaded on your computer, an executable is placed in the `x:\APT` directory. This executable is named `APTH1SRV.EXE` and is essential for communications with the CP1613 card on your computer.

When you click the `APT_WIN` icon on your desktop for the first time, the server is started. When the server is started, it executes as a background task, visible only in the Task Manager.

If you want, you can start the server manually each time you want APT to communicate over the Industrial Ethernet Network. To launch the APT H1 Server manually at any time, follow these steps:

1. Click **Start->Run** from the Windows taskbar.
2. Type `x:\apt\APTH1SRV` in the dialog box that appears, and click **OK**.

You can also launch the APT H1 Server manually by entering `START x:\APT\APTH1SRV` at the DOS command line.

Do not launch more than one APT server at a time. If you try to start a second server when one is already running, it detects that the first one is running and aborts the startup. If you want to launch a new server, you must first use the Task Manager to kill the running server .

Configuring the Network (continued)

Specifying the H1 Driver for APT

In order to differentiate between the H1 drivers under DOS and those under Windows NT/2000, a new environment variable has been added, **APT_OS**. If this environment variable is not present, APT assumes that the DOS/Windows 95 H1 drivers have been installed to facilitate communications with the CP1413 card. This variable must be set to either NT or 2000 in order for APT to recognize that it is communicating with the CP1613 card.

If you are running APT under the Windows NT operating system, place the following line in your **autoexec.bat** file:

```
Set APT_OS=NT
```

If you are running APT under the Windows 2000 operating system, place the following line in your **autoexec.bat** file:

```
Set APT_OS=2000
```

For more information, see [Appendix E](#), Configuring APT under Windows NT/2000.

APT H1 Server Registry Values

Configuration and tuning parameters for the APT H1 Server are stored in the system registry under the following key:

HKEY_LOCAL_MACHINE->SOFTWARE->SIEMENS->AUTSW->APTH1SRV

[Table B-16](#) lists all values stored under this key, with a description and the initial value of each. Do not change these registry values unless you understand exactly what the consequences will be.


 WARNING	
<p>Changing the registry values that are used by the APT H1 Server changes its operating behavior.</p>	
<p>A change in the operating behavior could cause unexpected problems in communicating with the controller, including downloading and debugging controller programs. Any unpredictable communications behavior could cause unexpected operation which could result in death or serious injury to personnel, and/or damage to equipment.</p>	
<p>Be sure that you fully understand the impact that changing registry values will have on the APT H1 Server before you make these changes and initiate communications with the controller.</p>	

Table B-16 APT H1 Server Registry Entries

Registry Entry	Description	Initial Value
CPUUsed	Name of the current CP that is being used by the server. This is the name of the registry sub-key that contains configurations for this specific card.	“CP16131”
icTmo	Amount of time in milliseconds between byte reads from the serial port.	100 ms
Priority	Server priority in NT/2000.	4 - High priority
serPriority	Priority to assign to the serial port services	0 - Low priority
Sharing	Enables sharing of the TF and SR interfaces with other applications.	0 - Not shared
TraceLog	Name of the debug trace log. Additional debug information is written to this log when tracing is enabled.	“C:\ \APTH1ServerLog.txt”
TraceMode	Enables or disables trace logging.	0 - Not enabled

Configuring the Network (continued)

Table B-17 lists the registry entries associated with the sub-key specified by the CPused entry. In this case, the sub-key is for the CP16131 card.

Table B-17 Sub-key Registry Entries

Registry Entry	Description	Initial Value
AppRelSleepTime	Amount of time for the server to wait after recovering the application relation ID from the CP1613. Value is in milliseconds.	10
CloseSleepTime	Amount of time for the server to wait after closing down its session with the CP1613. Value is in milliseconds.	0
InitiateRecvSleepTime	Amount of time for the server to wait after initiating a TF connection with the CP1613. Value is in milliseconds.	10
PipeDogTimeout	Amount of time before watchdog fires. The watchdog monitors the processing of a server command. If the command does not complete in a timely manner, the watchdog shuts down the server. Value is in seconds.	120
PipeName	The name of the pipe through which the server is communicating with APT.	"\\.\pipelaph1srv"
RecvExchSleepTime	Amount of time to wait between sending receive commands to the CP1613. Value is in milliseconds.	10
RecvMsgExchRetries	Number of times to retry a message exchange in the event of error.	0 - No retries
SRAccessPoint	Name of the Send/Receive access point configured for this CP1613.	"CP_H1_1"
SRCloseReqTmo	Amount of time to wait after closing down a send/receive session. Value is in milliseconds.	20000
SRConnReqTmo	Amount of time to wait for the completion of a Connect request. Value is in milliseconds.	20000
SROpenReqTmo	Amount of time to wait for the completion of an Open request. Value is in milliseconds.	20000
SRRceiveDatagramTmo	Amount of time to wait for the completion of a Receive Datagram request. Value is in milliseconds.	20000
SRRceiveDataTmo	Amount of time to wait for the completion of a Receive Data request. Value is in milliseconds.	20000
SRSendDataTmo	Amount of time to wait for the completion of a Send Data request. Value is in milliseconds.	20000
SRSendEOMDataTmo	Amount of time to wait for the completion of a Send EOM Data request. Value is in milliseconds.	20000
TFAccessPoint	Name of the TF access point on the CP1613.	"/CP_H1_1:/AP/"
TFcnfTimeout	Amount of time to wait for a message exchange retry. Value is in seconds.	0

Table B-17 Sub-key Registry Entries (continued)

Registry Entry	Description	Initial Value
TFOrdTimeout	Amount of time to wait for the completion of a message exchange with the controller. Value is in seconds.	20
TFReceiveTimeout	Amount of time to wait for a response from messages sent to the controller. Value is in seconds.	20
Version	Server version information.	"v1.0"

Configuring the Network (continued)

APT H1 Server Error Messages

If an error occurs that the APT H1 Server cannot correct or recover from, an error message is displayed. You must press the **Enter** key to continue. When you do, the server is terminated. Before you attempt to reconnect APT to the Industrial Ethernet network, you should identify and correct the problem that caused the error. Then start the server again, as described on [page B-35](#).

[Table B-18](#) lists the error messages generated by the server and suggests corrective actions for each error.

Table B-18 APT H1 Server Error Messages

Error Message	Description	Corrective Action
Error reading registry	The registry entries used by the Server have been corrupted.	Rerun the registry script APTH1SRV.REG located in the \APT directory.
Could not establish communications with CP, error = %!x	The CP1613 is not configured correctly.	Verify that the CP1613 is installed correctly. Verify the the CP1613 software was installed correctly by a user with administrator privileges.
Error opening Send/Receive session	The Send/Receive interface is not running.	Verify that PG-1613 is correctly loaded on your system. Run system diagnostics from PG-PC interface utility.
Error creating pipe to DOS	The server could not create a pipe to APT.	Make sure that APT is not attempting to communicate while you are starting the server. Make sure that if you modified the pipe name entry in the registry that it is correct. Use the task manager to see if an APT H1 server is already running.
Error closing Send/Receive session	An error occurred attempting to shut down the send/receive services on the CP1613 card.	Verify that PG-1613 is correctly loaded on your system. Run system diagnostics from PG-PC interface utility.
Server request has timed out	A communication request took too long to process.	Verify that the H1 network, including cables to your computer and the controller are connected. Use the PG-PC interface diagnostics to assure that the TF and SR services are running. Verify the entries you made in COML-TF, COM1430, and the CP1434 configurator are correct.
APT H1 Server shut down	The server is shutting down because of one of the previously listed errors.	Try corrective actions above.

Using APT with the Industrial Ethernet Network

Now that you have configured your system, you can use it on the Industrial Ethernet network. The Industrial Ethernet network support only affects five areas of APT:

- Debug
- Download
- Verify
- MAITT
- Tag Translate

When you use the Industrial Ethernet, you should place the association name you programmed into the CP1413/CP1613 Application Association list (for Series 505) or into the S5-H1 configuration file (for S5) in the controller name field of the compile control editor. In the example provided in [Figure B-12](#), PLC1 would be the name to use. Whenever you perform an operation requiring network communications, you use this controller name to make the connection. A dialog window is provided, which allows user override of the controller name if desired. This is useful when the same program must be downloaded to a number of different controllers on the network.

Appendix C

Using APT with DOS

C.1	DOS Memory Management	C-2
	Understanding the DOS Memory Types	C-2
	Using a Memory Manager	C-4
C.2	APT and DOS Memory	C-5
	Using Memory Effectively	C-5
	Configuring Memory Areas	C-5
	Effects of Hardware on Memory	C-7
C.3	Memory Configurations: Introduction	C-8
C.4	Memory Configurations: Compaq LTE Elite	C-10
	Using MS-DOS 5.0	C-10
	Using MS-DOS 6.22 with QEMM 7.04	C-12
C.5	Memory Configurations: 386/ATM Coprocessor Module	C-14
	Using MS-DOS 5.0 or Higher	C-14
C.6	Memory Configurations: TISTAR Model 70	C-16
	Using TISTAR 2.0 Operator Station	C-16
	Using TISTAR 1.3 Operator Station	C-18
C.7	Memory Configurations: Compaq Deskpro 486	C-20
	Using Compaq DOS 5.0 or Higher and a Network	C-20
C.8	Memory Configurations: HP XU 5/90C	C-24
	Using MS-DOS 6.22 and a Network	C-24
	Using MS-DOS 6.22 with QEMM 7.04 and a Network	C-26
C.9	Memory Configuration: Additional Boot Files	C-28
	Using MS-DOS 5.0 or Higher	C-28

C.1 DOS Memory Management

APT requires at least 520 Kbytes of DOS conventional memory to run. However, even if your computer has 16 Mbytes of memory, you may be unable to execute APT. This appendix describes how to manage your computer memory to run APT, using your computer memory as efficiently as possible.

Understanding the DOS Memory Types

The DOS operating system recognizes five types of memory in IBM-compatible personal computers. These memory types are listed in [Table C-1](#), and their relative address locations are illustrated in [Figure C-1](#).

Table C-1 DOS Memory Types

Type	Size (Kbytes)	Function
Conventional Memory ¹	640	The memory addresses between 0 Kbytes and 640 Kbytes are called conventional memory. Application programs such as APT are executed in conventional memory. The DOS operating system also loads utilities and device drivers from conventional memory. Some or all portions of the DOS operating system must reside in conventional memory.
Upper Memory ²	384	The memory addresses between 640 Kbytes and 1 Mbyte are called upper memory. System ROM, system hardware, e.g., video ROM, terminate-stay-resident (TSR) utilities, and device drivers, can be loaded into segments of memory called Upper Memory Blocks (UMB). The UMBs require a memory manager for control. The MS-DOS 5.0 or higher utility EMM386.EXE provides this service for 80386, 80486 and Pentium computers.
High Memory (HMA)	64	The memory addresses above 1 Mbyte are called extended memory. The first 64 Kbytes of extended memory can be reserved by the memory manager. This 64 Kbyte block is called the high memory area or (HMA). The MS-DOS 5.0 or higher utility HIMEM.SYS provides this service for 80386, 80486, Pentium, and some 80286 computers that have memory above 1 Mbyte.
Extended Memory	0 to maximum RAM installed	Extended memory addresses, located above HMA, provide a continuation of conventional memory. Some application programs control extended memory directly; others can use extended memory, but allow the operating system to manage it. The MS-DOS 5.0 or higher utility HIMEM.SYS provides this service.
Expanded Memory	0 to maximum RAM installed	Expanded memory is managed by an expanded memory manager that uses a 64 Kbyte block of space in the upper memory area to reference the expanded memory. The physical location of the expanded memory is typically a memory expansion board. Like extended memory, expanded memory provides a continuation of conventional memory that can be used by an application program. The MS-DOS 5.0 or higher utility EMM386.EXE enables extended memory to emulate expanded memory on 80386, 80486 and Pentium computers.
<p>1 Conventional memory is often referred to as low memory.</p> <p>2 Upper memory is often referred to as high memory. When DOS commands LOADHIGH and DEVICEHIGH are used, the action is to load into upper memory.</p>		

Figure C-1 shows the relative address locations of the five memory types that DOS recognizes if you have an IBM-compatible personal computer.

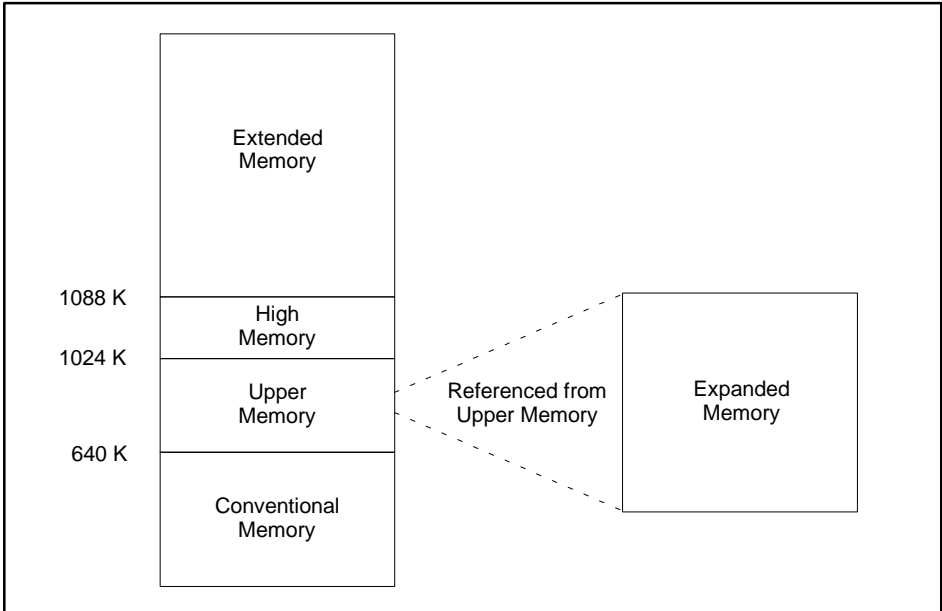


Figure C-1 DOS Memory Configuration

DOS Memory Management (continued)

Using a Memory Manager

Without a DOS memory manager, all application programs, device drivers, Terminate-Stay-Resident (TSR) programs, and the operating system must reside in conventional memory. You may not be able to load all the utilities that you need on your computer and still run APT or other application programs that require over 500 Kbytes of memory.

A memory manager can help resolve the memory constraints of the DOS operating system. You can load some device drivers and TSRs into Upper Memory, allowing more conventional memory to be available for application programs. Most memory managers can operate on 80386, 80486, and Pentium computers that have at least 2 Mbytes of RAM. The MS-DOS 5.x and 6.x operating systems both have memory managers. The Quarterdeck Expanded Memory Manager is an example of a commercially available memory manager.

Install a memory manager on your APT computer if you encounter any of these problems:

- You cannot execute APT.
- You cannot execute APT with essential utilities such as disk caching installed.
- You cannot execute APT with essential utilities and other utilities installed.

NOTE: This appendix mentions several software products and manufacturers of personal computers. These examples are given for illustration only. Siemens Energy & Automation does not endorse any specific product or manufacturer.

C.2 APT and DOS Memory

Using Memory Effectively

Table C-2 lists the amount of memory required to load and execute APT. Cache software is not included with APT, but a disk cache is recommended to improve performance.

Table C-2 APT Memory Requirements

Program	Memory Required to Load (Kbytes)	Memory Required to Execute (Kbytes)
APT	—	520
GSS: APT Graphics Driver (Largest Configuration)	84.1	80.7
Memory Cache	—	15-30

Configuring Memory Areas

Follow the suggestions below to use your conventional and upper memory most effectively.

- Plan to leave as much conventional memory free as is possible. APT must reside in conventional memory. Try to load all utilities and device drivers into your computer's upper memory. Try installing different combinations of drivers/utilities in conventional and upper memory to see which combination gives you the greatest amount of conventional memory.

Use the `mem /f` command to view the amount of memory available in each region.

Use the `mem /c` command to view the size of drivers currently installed.

Consider these alternatives:

Load large utilities and drivers, such as the GSS graphics driver, and cache and network drivers into upper memory. Install only the smaller utilities and drivers into conventional memory.

If your upper memory is small, load GSS into upper memory and all other large utilities into conventional memory. Collectively, the other utilities may not use as much conventional memory as the GSS drivers.

If you install the GSS graphics driver in conventional memory and all other utilities into upper memory, you can unload GSS when you are not using APT. Conventional memory increases by 80 Kbytes when GSS is removed.

To make the monochrome region available (if it does not need to be reserved by the system), add the following parameter to the memory manager line in your **config.sys** file: `i=B000-B7FF`

- Load items in an optimal order. Some utilities require more memory to load than to run. For example, 40 Kbytes of free upper memory may not be sufficient for a 35 Kbyte utility if it requires 45 Kbytes to load. Load this utility before another utility that does not require more memory when loading.
- Do not use expanded memory if you do not need it. If none of your programs require expanded memory, instruct your memory manager not to use it. An expanded page frame requires 64 Kbytes of upper memory and reduces memory available for drivers and TSRs.
- Configure the hardware for extended memory. Some computers can configure memory as extended or expanded. Configure all memory above the conventional memory as extended when possible. If you have an application that requires expanded, allow a memory manager to convert some extended memory to expanded memory.
- Disable shadow RAM and any video BIOS if your computer allows this. This may provide more upper memory. If your hard disk requires a special configuration, you may be unable to disable the system shadow BIOS.
- If MS-DOS 5.0 does not provide enough upper memory for your needs, obtain another memory manager that provides more upper memory.
- If you use MS-DOS 6.22, use the following guidelines.

Part of DOS can be loaded high using the MemMaker utility. Most of DOS can be loaded high using QEMM 7.0 DOS-UP drivers.

The DriveSpace utility, which replaces DOS DoubleSpace, is a compression program that frees up space on your hard disk. However, the driver is 38K, and the utility also requires 45-50K of DOS overhead.

The MemMaker utility looks at available memory and your desired drivers and tries to make a best fit. You can use MemMaker for your first pass, and then if MemMaker has not produced enough conventional memory follow up with an iterative approach, loading large drivers first and adding smaller ones next.

Effects of Hardware on Memory

Before you purchase hardware, consider how it may affect available upper memory. Some devices may reduce the available upper memory. Follow the suggestions below.

- Some computers have many blocks of system ROM or device hardware scattered in upper memory, resulting in several small, less-useful regions of free memory. Ideally, your computer should provide a maximum of upper memory with a minimum of scattered blocks of system ROM.

You can obtain commercially available programs, such as QEMM, that help you determine the memory map for a computer.

- Some hard disks affect how much upper memory is available. Most SCSI hard disks require device drivers that require memory. Some hard disks do not allow you to disable the system shadow BIOS.
- Some hardware cards, e.g., some ethernet cards, require a block of memory in upper memory. You can either use an ethernet card that does not require upper memory (performance may be affected) or select a location for the memory that minimizes the number of upper memory regions. Some cards give a choice of regions that you can configure with jumpers. Experiment with configurations that provide the largest block of contiguous memory.

Table C-3 shows cards that require specific reserved memory regions.

Table C-3 Hardware Cards That Require Reserved Memory

Card	Region	Size in Kbytes
TISTAR MATROX Video Card	B000-C7FF	96
TISTAR TIGA Video Card	D000-D1FF	8
CVU TIWAY Adapter Card	C800-C8FF	4
386/ATM Module	C800-C8FF	4
CP1413 Card	D000-DFFF	64

C.3 Memory Configurations: Introduction

The purpose of [Sections C.4 to C.8](#) is to help you determine how best to configure your computer for running APT. These sections describe memory configurations for several common computer models and give suggestions for how to install drivers. It is recommended that you look at all the example configurations. Although you may not have the hardware or software described, you can use the examples as guidelines for configuring your own system.

If you require assistance in configuring your computer, call the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at (800) 333-7421. Outside the U.S.A., call 49-911-895-7000.

The basic computer configurations shown in [Figure C-2](#) are based on a computer that has not been set up for any particular application, but which has a memory manager loaded. These basic computer configurations are the starting points for the APT configuration presented in [Sections C.4 through C.9](#), which are specifically set up for running APT.

The **autoexec.bat** and **config.sys** files are the same for the basic configuration for each computer. The contents of these files for the basic configuration are shown in [Figure C-2](#). Note that three **config.sys** files are shown, the differences depending on the memory manager used.

<p>CONFIG.SYS All computer models All basic configurations Memory manager is <u>DOS 5.0</u></p>	<pre>device=c:\dos\himem.sys device=c:\dos\emm386.exe noems dos=high,umb shell=c:\command.com c:\dos /e:500 /p buffers=20 files=50</pre>
<p>CONFIG.SYS All computer models All basic configurations Memory manager is <u>QEMM</u></p>	<pre>device=c:\qemm\qemm386.sys ram noems dos=high shell=c:\command.com c:\dos /e:500 /p buffers=20 files=50</pre>
<p>CONFIG.SYS All computer models All basic configurations Memory manager is <u>CEMM</u></p>	<pre>device=c:\dos\himem.exe device=c:\dos\cemm dos=high shell=c:\command.com c:\dos /e:500 /p buffers=20 files=50</pre>
<p>AUTOEXEC.BAT All computer models All basic configurations</p>	<pre>path c:\dos set prompt=\$!\$p\$g</pre>

Figure C-2 AUTOEXEC.BAT and CONFIG.SYS for Basic Computer Configurations

The APT file **cgi.cfg**, which specifies the drivers used by APT, has the same format for all valid configurations. The content of this file is shown in [Figure C-3](#). At the DOS prompt, enter the `drivers` command to load **cgi.cfg**.

```
;These device drivers are needed to run APT.
;
; video card driver
driver=c:\apt\drivers\ibmvga12.sys/g:display
;
; printer driver
driver=c:\apt\drivers\laserjet.sys/g:display
;
; mouse driver
driver=c:\apt\drivers\msmouse.sys/g:input
;
; gss*cgi graphics driver
driver=c:\apt\drivers\gsscgi.sys/r
```

Figure C-3 Content of CGI.CFG

C.4 Memory Configurations: Compaq LTE Elite

Using MS-DOS 5.0 In this example, the Compaq LTE Elite has 6 Mbytes of RAM and uses the MS-DOS 5.0 operating system without Windows, and the **himem.sys** and **emm386.exe** memory managers provided by DOS.

The **config.sys** and **autoexec.bat** files shown in [Figure C-4](#) are set up for running APT.

<p><i>CONFIG.SYS</i> Compaq LTE Elite set up to run APT</p>	<pre>device=c:\dos\himem.sys device=c:\dos\emm386.exe noems frame=none devicehigh=c:\apt\drivers\cgistub.sys /r shell=c:\command.com /p /e:1024 dos=high,umb buffers=10,0 files=50</pre>
<p><i>AUTOEXEC.BAT</i> Compaq LTE Elite set up to run APT</p>	<pre>path c:\apt;c:\dos set prompt=\$p\$g set cgipath=c:\apt set mouse=msmouse set lmouse=c:\mouse loadhigh c:\apt\drivers.exe loadhigh c:\dos\doskey.com loadhigh c:\dos\smartdrv.exe loadhigh c:\mouse\mouse</pre>

Figure C-4 Compaq LTE Elite CONFIG.SYS and AUTOEXEC.BAT for APT

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The APT configuration for a Compaq LTE Elite distributes memory in the allocations shown in [Table C-4](#).

Largest executable program: 625,600 (610.9K)
Upper memory available: 46,720 (45.6K)

Table C-4 Memory Configuration for the Compaq LTE Elite

Conventional Memory		Upper Memory	
File	Size (Kbytes)	File	Size (Kbytes)
MSDOS	15.0	SYSTEM	160.0
HIMEM	1.2	CGISTUB	0.0
EMM386	9.2	MOUSE	16.7
COMMAND	3.3	DRIVERS	65.1
FREE	0.1	DOSKEY	4.0
FREE	611.1	SMARTDRV	28.4
		FREE	45.6

NOTE: The APT DRIVERS (65.1 Kbytes) program is loaded before all other programs. Otherwise, DRIVERS does not have enough memory to load.

Memory Configurations: Compaq LTE Elite (continued)

Using MS-DOS 6.22 with QEMM 7.04

In this example, the Compaq LTE Elite has 6 Mbytes of RAM and uses the MS-DOS 6.22 operating system, and the Quarterdeck Expanded Memory Manager (QEMM).

The **config.sys** and **autoexec.bat** files shown in [Figure C-5](#) are set up for running APT.

```
CONFIG.SYS
Compaq LTE Elite
set up to run APT
device=c:\qemm\dosdata.sys
device=c:\qemm\qemm386.sys ram fr=none r:1
device=c:\qemm\dos-up.sys @c:\qemm\dos-up.dat
device=c:\qemm\loadhi.sys /r:2 /size=560 c:\apt\drivers\cgistub.sys/r
shell=c:\qemm\loadhi.com /r:2 /res=2992 /sqf c:\command.com /p /e:1024
dos=high,umb
files=50
buffers=10,0

AUTOEXEC.BAT
Compaq LTE Elite
set up to run APT
path c:\apt;c:\dos
set prompt=$p$g
set mouse=msmouse
set lmouse=c:\mouse
set cgipath=c:\apt
c:\qemm\loadhi /r:2 c:\apt\drivers.exe
c:\qemm\loadhi /r:2 c:\dos\smartdrv.exe
c:\qemm\loadhi /r:1 c:\dos\doskey.com
c:\qemm\loadhi /r:2 c:\mouse\mouse
```

Figure C-5 Compaq LTE Elite CONFIG.SYS and AUTOEXEC.BAT for APT under QEMM

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The APT configuration for a Compaq LTE Elite distributes memory in the allocations shown in [Table C-5](#).

Largest executable program: 633 Kbytes
 Upper memory available: 61 Kbytes

Table C-5 Memory Configuration for the Compaq LTE Elite under QEMM

Conventional Memory		Upper Memory	
File	Size (Kbytes)	File	Size (Kbytes)
QEMM	1	COMMAND	3
LOADHI	0	FILES	3
COMMAND	1	LASTDRIV	0
FREE	633	INSTALL	0
		DOSKEY	4
		DOS-UP	0
		DOSDATA	5
		CGISTUB	0
		STACKS	2
		MOUSE	17
		DRIVERS	65
		SMARTDRV	28
		FREE	61

In this configuration, the QEMM `loadhi` parameter, `/r: x`, directs the small files to Region 1 (`x=1`) and `DRIVERS` to Region 2 (`x=2`).

C.5 Memory Configurations: 386/ATM Coprocessor Module

Using MS-DOS 5.0 or Higher

In this example, the 386/ATM Coprocessor has 4 Mbytes of RAM and uses the MS-DOS 5.0 operating system, and the **himem.sys** and **emm386.exe** memory managers provided by DOS.

The **config.sys** and **autoexec.bat** files shown in [Figure C-6](#) are set up for running APT.

<pre>CONFIG.SYS 386/ATM Coprocessor set up to run APT</pre>	<pre>device=c:\dos\himem.sys device=c:\dos\emm386.exe noems x=C800-C900 dos=high,umb device=c:\dos\setver.exe device=c:\dos\smartdrv.sys 1024 1024 device=c:\dos\ramdrive.sys 1000 /e device=c:\drivers\bs_386.exe device=c:\drivers\aptdd.sys device=c:\apt\drivers\cgistub.sys/r shell=c:\command.com e:500/p break=on buffers=20 files=50</pre>
<pre>AUTOEXEC.BAT 386/ATM Coprocessor set up to run APT</pre>	<pre>path c:\dos;c:\apt set prompt=\$l\$p\$g set cgipath=c:\apt loadhigh drivers loadhigh c:\dos\doskey set mouse=msmouse mouse set apt_mode=cvu set apt_temp=e:\</pre>

Figure C-6 386/ATM Coprocessor CONFIG.SYS and AUTOEXEC.BAT for APT

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The APT configuration for a 386/ATM Coprocessor distributes memory in the allocations shown in [Table C-6](#).

Largest executable program: 567.1 Kbytes

Table C-6 Memory Configuration for the 386/ATM Coprocessor

Conventional Memory		Upper Memory	
File	Size (Kbytes)	File	Size (Kbytes)
MSDOS	15.1	SYSTEM	168.0
HIMEM	2.8	DRIVERS	80.7
EMM386	8.2	DOSKEY	4.0
SETVER	0.4	FREE	0.1
SMARTDRV	14.2	FREE	3.1
RAMDRIVE	1.2		
BS_386	6.0		
APTDD	6.5		
CGISTUB	0.0		
COMMAND	2.8		
MOUSE	15.0		
FREE	0.1		
FREE	567.3		

NOTE: RAMDRIVE and DOSKEY cannot both be loaded into upper memory under this configuration. DOSKEY alone is loaded into upper memory since it is the larger utility.

C.6 Memory Configurations: TISTAR Model 70

Using TISTAR 2.0 Operator Station

In this example, the AST Premium has 4 Mbytes of RAM and uses a TIGA graphics adapter, the MS-DOS 5.0 operating system, the **himem.sys** and **emm386.exe** memory managers, and disk caching provided by DOS.

The **config.sys** and **autoexec.bat** files shown in [Figure C-7](#) are set up for running APT.

```
CONFIG.SYS
TISTAR 2.0 Model 70 Operator
Station set up to run APT

device=c:\dos\himem.sys
device=c:\dos\emm386.exe noems x=d000-d3ff
devicehigh=c:\dos\smartdrv.sys 2048
devicehigh=c:\dos\ansi.sys
device=d:\apt\drivers\cgistub.sys/r
dos=high,umb
buffers=20
files=40
shell=c:\command.com c:\dos/e:660/p

AUTOEXEC.BAT
TISTAR 2.0 Model 70 Operator
Station set up to run APT

path=c:\200;c:\tiga;c:\dos;c:\;d:\apt;
set prompt=$l$p$g
set apt_mode=direct
set video_card=tiga
set tiga=-mc:\tiga -lc:\tiga -i0x61
set tigafont=c:\tigafont;
del c:\tigafont\tiga.cfg>nul
c:\tiga\tigacd
c:\tiga\tigaload
set bd=19200
set vtbd=9600
pushkey check^My^Migt^M
set igt_path=c:\200
set mouse=msmouse
c:\msmouse

TIAPT.BAT
TISTAR 2.0 Model 70 Operator
Station set up to run APT

if "%video_card%" == "tiga" tigacd -u
apptiga

APTTIGA.BAT
TISTAR 2.0 Model 70 Operator
Station set up to run APT

::Setup to run APT
::
d:
cd \apt
set apt_mode=tistar
set cgipath=d:\apt\sys\exe
d:\apt\drivers\drivers.exe
apt

::Reset to run TISTAR
set apt_mode=direct
d:\apt\drivers\drivers.exe /a
set cgipath=
cd \
if "%video_card%" == "tiga" tigacd
if "%video_card%" == "tiga" tigaload
igt
```

Figure C-7 TISTAR 2.0 Model 70 Operator Station CONFIG.SYS and AUTOEXEC.BAT for APT

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The memory configuration for an AST Premium, after loading the **config.sys** and **autoexec.bat** files, provide the following amounts of free memory.

Largest executable program:	566.4 Kbytes
Upper memory available (Region 1):	27.9 Kbytes
Upper memory available (Region 2):	21.5 Kbytes

The TISTAR configuration for an AST Premium distributes memory in the allocations shown in [Table C-7](#).

Table C-7 Memory Configuration for the TISTAR 2.0 Model 70

Conventional Memory		Upper Memory	
File	Size (Kbytes)	File	Size (Kbytes)
MSDOS	14.9	SYSTEM	180.0
HIMEM	1.2	SMARTDRV	22.3
EMM386	8.2	ANSI	4.1
CGISTUB	0.0	FREE	27.9
COMMAND	3.0	FREE	21.5
MSMOUSE	6.9		
TIGACD	38.8		
FREE	0.1		
FREE	566.7		

The following special considerations apply to this configuration:

- When APT loads, the TIGACD driver is unloaded before DRIVERS is loaded. As with the APT DRIVERS, TIGACD needs to be the last one loaded.
- The printer configured for APT is the PaintJet (180 DPI resolution) instead of the LaserJet.

Memory Configurations: TISTAR Model 70 (continued)

Using TISTAR 1.3 Operator Station

In this example, the AST Premium has 4 Mbytes of RAM and uses a MATROX graphics adapter, the AST DOS 3.3 operating system, the Quarterdeck Expanded Memory Manager 386, and the AST version of disk caching.

The **config.sys** and **autoexec.bat** files shown in [Figure C-8](#) are set up for running APT.

```
device=c:\qemm\qemm386.sys ram m=120 noems x=b000-c800
device=c:\mousesys.sys /g:input
device=c:\tivdi.sys /g:display
device=d:\apt\drivers\hppj180.sys /g:display
device=c:\gsscgi.sys /r
device=c:\qemm\loadhi.sys c:\ansi.sys
buffers=10
files=40
shell=c:\command.com c:\ /e:660/p

CONFIG.SYS
TISTAR 1.3 Model 70 Operator
Station set up to run APT

path=c:\;c:\sys;d:\apt;c:\qemm;c:\astutil
set prompt=$l$p$g
set apt_mode=direct
loadhi astcache /e/r:o/b-/t-
d:\apt\drivers\pgdownl d:\apt\drivers\ticvu>matrox.ver
set mouse=mousesys
set mousesys=com1
mode t=on>null
del null
mode co80
cfgbaud

AUTOEXEC.BAT
TISTAR 1.3 Model 70 Operator
Station set up to run APT

TIAPT.BAT
TISTAR 1.3 Model 70 Operator
Station set up to run APT

apptiga

::Setup to run APT
::
d:
cd \apt
set apt_mode=tistar
set cgipath=c:\apt
apt

APTTIGA.BAT
TISTAR 1.3 Model 70 Operator
Station set up to run APT

::Reset to run TISTAR
set apt_mode=direct
c:
cd \
igt
```

Figure C-8 TISTAR 1.3 Model 70 Operator Station **CONFIG.SYS** and **AUTOEXEC.BAT** for APT

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The memory configuration for an AST Premium, after loading the **config.sys** and **autoexec.bat** files, provides the following amounts of free memory.

Largest executable program:	567.7 Kbytes
Upper memory available (Region 1):	38.0 Kbytes
Upper memory available (Region 2):	8.0 Kbytes

The TISTAR/APT configuration for an AST Premium distributes memory in the allocations shown in [Table C-8](#).

Table C-8 Memory Configuration for the TISTAR 1.3 Model 70

Conventional Memory		Upper Memory (<i>LOADHI</i>)	
File	Size (Kbytes)	File	Size (Kbytes)
The DOS command CHKDSK indicates that there are 567.7 Kbytes free.		Region1: ANSI	1.5
		Region1: ASTCACHE	51.0
		Region1: AVAILABLE	38.0
		Region2: AVAILABLE	8.0

The following special considerations apply to this configuration:

- The MATROX graphics adapter is similar to CGA graphics, and QEMM can provide 704 Kbytes of free memory.
- QEMM cannot load the TISTAR/APT device drivers high under MS-DOS 3.3. Upgrade to MS-DOS 5.0 if you need to load these drivers high.
- If TISTAR does not operate correctly with the cache installed, you must load the cache only when you are running APT. Follow these steps.

1. Delete the following line from the **autoexec.bat** file:

```
loadhi astcache /e/r:o/b-/t-
```

2. Add the following lines to the **apptiga.bat** file:

```
astcache /e/r:o/b-/t-
apt
astcache /u
```

These changes will load a cache just before APT is executed, and will unload the cache when APT is finished.

C.7 Memory Configurations: Compaq Deskpro 486

Using Compaq DOS 5.0 or Higher and a Network

In this example, the Compaq Deskpro has 36 Mbytes of RAM, uses the Compaq DOS 5.0 operating system, and the **himem.sys**, **cemm.exe** and **runhi.exe** memory managers provided by Compaq DOS. The system has a CVU TIWAY Adapter installed and is connected to a Novell network.

The **config.sys** and **autoexec.bat** files shown in [Figure C-9](#) are set up for running APT.

<pre>CONFIG.SYS Compaq 486 set up to run APT</pre>	<pre>device=c:\dos\himem.sys device=c:\dos\cemm.exe noems rom x=c800-c8FF device=c:\dos\runhi.exe c:\dos\cache.exe 3072 /ext device=c:\dos\runhi.exe c:\dos\vdisk.sys 32768 device=c:\dos\runhi.exe c:\apt\drivers\cvuta.dev -i11 device=c:\apt\drivers\cgistub.sys /r shell=c:\command.com /e:500 /p lastdrive=x dos=high buffers=20 files=50</pre>
<pre>AUTOEXEC.BAT Compaq 486 set up to run APT</pre>	<pre>path c:\dos;c:\novell;c:\apt; set prompt=\$l\$p\$g set cgipath=c:\apt runhi drivers runhi c:\novell\ipx runhi c:\novell\netx /ps=xxxx runhi c:\dos\doskey runhi c:\dos\append set apt_mode=cvu mouse</pre>

Figure C-9 Compaq Deskpro 486 CONFIG.SYS and AUTOEXEC.BAT for APT

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The APT configuration for a Compaq Deskpro 486 distributes memory in the allocations shown in [Table C-9](#).

Largest executable program: 598.8 Kbytes

Table C-9 Memory Configuration for the Compaq Deskpro 486

Conventional Memory		Upper Memory (RUNHI)	
File	Size (Kbytes)	File	Size (Kbytes)
IBMDOS	17.2	CEMM	3.6
HIMEM	2.8	CACHE	17.3
CEMM	4.1	VDISK	0.7
RUNHI	0.3	CVUTA	2.0
RUNHI	0.2	FREE	0.1
RUNHI	0.1	DRIVERS	76.7
CGISTUB	0.0	IPX	16.3
COMMAND	2.8	NETX	42.7
MOUSE	13.2	DOSKEY	4.0
FREE	0.1	APPEND	8.9
FREE	598.9	FREE	11.4

See the next page for guidelines to follow when you configure your system.

Memory Configurations: Compaq Deskpro 486 (continued)

The following special considerations apply to this configuration:

- The mouse (Logitech Trackman) cannot be loaded into upper memory after APT DRIVERS under this configuration. DOSKEY and APPEND were loaded instead.
- The addresses C800-C8FF are reserved for the CVU TIWAY Adapter.
- The DOS append command is used for the network.
- APT can be executed in the large 32 Mbyte RAM disk.
- The Novell drivers are for the NE2000 Ethernet card. This card does not use the upper memory region.

If you have a card that requires 16 Kbytes of upper memory, your region will be smaller because the memory area used by the card must be excluded (x= parameter for **cemm.exe**). [Table C-10](#) shows examples of region sizes, based on the exclusion of various ranges of addresses. The TIWAY adapter is not considered.

NOTE: Exclude memory locations C800-CBFF to create the largest single region. However, you cannot exclude this region (C800-CBFF) when a CVU TIWAY Adapter is installed.

Table C-10 Region Sizes in Kbytes Determined by Starting Address

Starting Address	Range	Region 1	Region 2
C800	C800-CBFF	176	-
D000	D000-D3FF	32	144
D800	D800-DBFF	64	112

- If you do not have a video card that uses addresses B0000-B7FF, e.g., monochrome, super VGA, Matrox, etc., this memory area can be included (i=B000-B7FF parameter for **cemm.exe**). This creates a new Region 1 of 32 Kbytes. The old Region 1 becomes Region 2.

QEMM typically includes these addresses automatically. See [page C-12](#) for an example of how to use QEMM.

C.8 Memory Configurations: HP XU 5/90C

Using MS-DOS 6.22 and a Network

In this example, the HP Vectra XU 5/90C, a Pentium processor, has 80 Mbytes of RAM and uses the MS-DOS 6.22 operating system connected to a Novell network.

The **config.sys** and **autoexec.bat** files shown in [Figure C-10](#) are set up for running APT.

```
CONFIG.SYS
Pentium processor
set up to run APT

device=c:\dos\himem.sys
device=c:\dos\emm386.exe noems highscan I=B000-B7FF
devicehigh /L:1,5888=c:\dos\ramdrive.sys 32767 /e
devicehigh /L:1,272=c:\gss\cgistub.sys /r
devicehigh /L:1,9072=c:\dos\ansi.sys
shell=c:\command.com/e:3096 /p
dos=umb,high
files=50
buffers=30,0
lastdrive=x
fcbs=4,0
stacks=0,0

AUTOEXEC.BAT
Pentium processor
set up to run APT

@echo off
path c:\dos;c:\novell;c:\apt;
lh /L:2,69616 c:\gss\drivers.exe
lh /L:1,9296 c:\novell\lsl.com
lh /L:2,34352 c:\novell\3c5x9.com
lh /L:2 c:\novell\ipxodi.com
c:\novell\netx.exe
lh /L:1,6384 c:\dos\doskey.com
lh /L:2 c:\dos\smartdrv.exe /q 2048 2048
```

Figure C-10 Pentium CONFIG.SYS and AUTOEXEC.BAT for APT

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The APT configuration for a Pentium processor distributes memory in the allocations shown in [Table C-11](#).

Largest executable program: 578.8 Kbytes

The monochrome region is included as part of the memory allocated for the largest executable program.

Table C-11 Memory Configuration for the Pentium

Conventional Memory		Upper Memory	
File	Size (Kbytes)	File	Size (Kbytes)
RAMDRIVE	1.2	MSDOS	15.6
CGISTUB	0.96	HIMEM	1.2
ANSI	4.2	EMM386	4.1
LSL	5.0	COMMAND	5.8
DOSKEY	4.1	NETX	48.7
DRIVERS	69.1	FREE	37.9
3C5X9	11.5		
IPXODI	16.3		
SMARTDRV	29.0		
FREE	579.4		

Memory Configurations: HP XU 5/90C (continued)

Using MS-DOS 6.22 with QEMM 7.04 and a Network

In this example, the HP Vectra XU 5/90C, a Pentium processor, has 80 Mbytes of RAM, uses the MS-DOS 6.22 operating system with the QEMM 7.04 memory manager, and communicates using the Novell network.

The **config.sys** and **autoexec.bat** files shown in [Figure C-11](#) are set up for running APT.

CONFIG.SYS Pentium processor set up to run APT	<pre>device=c:\qemm\dosdata.sys device=c:\qemm\qemm386.sys frame=none ramx=C800-CBFF x=F400-F4FF r:1 device=c:\qemm\dos-up.sys@c:\qemm\dos-up.dat device=c:\qemm\loadhi.sys/r:1/size=16128 c:\dos\vdisk.sys 32767 device=c:\qemm\loadhi.sys/r:1/size=560 e:\gss\cgistub.sys/r dos=high,umb files=50 buffers=50,0 stacks=9,256 lastdrive=x shell=c:\qemm\loadhi.com/r:2 c:\command.com c:\ /p /e:2048</pre>
AUTOEXEC.BAT Pentium processor set up to run APT	<pre>@echo off path c:\dos;c:\qemm;c:\APT;c:\gss set mouse=c:\windows\mouse set cgipath=e:\gss c:\qemm\loadhi /r:1 c:\nv\ls1 c:\nv\3c5x9.com c:\qemm\loadhi /r:2 c:\nv\ipxodi c:\qemm\loadhi /r:2 c:\nv\netx c:\qemm\loadhi /r:2 c:\gss\drivers.exe c:\qemm\loadhi /r:1 c:\dos\doskey.com</pre>

Figure C-11 Pentium CONFIG.SYS and AUTOEXEC.BAT for APT under QEMM

Use the DOS `mem /c` command to examine how DOS has allocated your computer memory.

The APT configuration for a Pentium processor distributes memory in the allocations shown in [Table C-12](#).

Largest executable program: 622 Kbytes

Table C-12 Memory Configuration for the Pentium under QEMM

Conventional Memory		Upper Memory	
File	Size (Kbytes)	File	Size (Kbytes)
SYSTEM	4.3	SYSTEM	724.9
COMMAND	4.7	COMMAND	2.1
QEMM386	0.9	DOS-UP	0.2
LOADHI	0.1	DOSDATA	5.7
3C5X9	11.5	VDISK	0.8
FREE	636.4	CGISTUB	0.1
		FILES	2.7
		FCBS	0.3
		WKBUFFER	0.5
		LASTDR	2.1
		STACKS	3.0
		INSTALL	0.2
		LSL	5.0
		DOSKEY	4.1
		IPXODI	16.3
		NETX	48.7
		DRIVERS	71.2
		FREE	6.3

C.9 Memory Configuration: Additional Boot Files

Using MS-DOS 5.0 or Higher

APT is shipped with two additional system files, **autoexec.def** and **config.def**, installed in the **apt\sys** directory. These files are designed for use with MS-DOS 5.0 or later, and take advantage of the DOS memory management features. You can tailor these files to suit your individual system as required. However, if you use these files, the **APT_DRIVE** environment variable in the **autoexec.def** file must reflect the drive where APT is installed.

If you use these files as your system files, you will have enough available memory both to run APT and to load the APT drivers into upper memory. Although these files do not contain the same settings as your non-APT working environment, you can make incremental additions to them and verify each change. In doing this, you create a custom boot file. You must manually copy these files to the root of your boot drive and then re-boot your system before these changes are added to the boot file. You can then copy **autoexec.def** and **config.def** to **autoexec.apr** at the **apt\sys** directory and **config.apr** and use **aprmode.bat** and **dosmode.bat** to switch between environments. [Figure C-12](#) shows the **autoexec.def** and **config.def** files.

```
AUTOEXEC.DEF      @echo off
                   set APT_DRIVE=C:
                   set PATH=%APT_DRIVE%\apt;c:\;c:\dos
                   set PROMPT=$p$g
                   set CGIPATH=%APT_DRIVE%\apt
                   lh %APT_DRIVE%\apt\drivers.exe
                   lh c:\dos\smartdrv.exe 2048
                   set TMPDIR=c:\tmp
                   set TMP=c:\tmp
                   set TEMP=c:\tmp
                   set APT_TEMP=c:\tmp
                   %APT_DRIVE%
                   cd \apt
                   set APT_DRIVE=

CONFIG.DEF         break      =on
                   files      =40
                   buffers    =20
                   stacks     =9,256
                   dos        =high, umb
                   device     =C:\DOS\himem.sys
                   device     =C:\DOS\emm386.exe 2048 NOEMS
                   device     =D:\APT\DRIVERS\cgistub.sys /R
                   device     =C:\DOS\command.com C:\DOS\ /E:1024 /p
```

Figure C-12 MS-DOS 5.0 Additional Boot Files

Using APT with Windows 95

D.1	Running APT under Windows 95	D-2
	Overview	D-2
	Selecting the Type of DOS Emulation	D-2
	Virtual DOS Environment under Windows 95	D-4
	Real Mode DOS Environment with Standard Configuration	D-6
	Real Mode DOS Environment with New Configuration	D-8
D.2	Installing APT in a Virtual DOS Environment	D-10
	Overview	D-10
	Creating a DOS Icon for APT	D-10
	Setting the Properties for the DOS Window	D-12
	Configuring Memory Management	D-15
	Installing APT	D-19
	Writing a Configuration Batch File	D-23
	Running APT in the Window	D-24
	Resource Sharing with Other DOS and Windows Programs	D-25
D.3	Installing APT in Real Mode DOS Environment (Standard Configuration)	D-26
	Overview	D-26
	Creating a DOS Icon for APT	D-26
	Setting the Properties for the DOS Window	D-27
	Configuring Memory Management	D-29
	Installing APT	D-32
	Running APT in DOS Mode	D-35
D.4	Installing APT in Real Mode DOS Environment with New Configuration	D-36
	Overview	D-36
	Creating a DOS Icon for APT	D-36
	Setting the Properties for the DOS Window	D-37
	Configuring Memory Management	D-39
	Installing APT	D-42
	Running APT in DOS Mode	D-45

D.1 Running APT under Windows 95

Overview

This appendix explains how to install and operate APT under Windows 95. The sections describe the types of DOS environments available from Windows 95, installation of APT and modification of the **autoexec.bat** and **config.sys** files, operation of APT under Windows 95, and trouble-shooting suggestions.

You can successfully install and use APT under Windows 95, but to ensure that everything operates correctly, you need to do some additional configuration. First, you have to set up memory management for conventional memory in the DOS environment. This is similar to setting up memory management in a DOS-only system. Then you need to configure a properties sheet for the DOS session that you use to run APT. You must manually configure parts of the **autoexec.bat** and **config.sys** files. When you complete installation, sample properties sheets (PIFs) and icons will reside in the **\apt** directory on your hard drive.

Selecting the Type of DOS Emulation

Windows 95 supports two types of DOS environments:

Virtual DOS session A virtual DOS session runs as a window under the control of Windows 95. You can press **Alt** **Enter** to toggle the window to full screen, but it is still a virtual session under Windows 95 control.

DOS mode When you start DOS mode, Windows 95 unloads and places the computer under the control of a real mode version of DOS. In this mode, Windows 95 simply installs a small program that causes the computer to reboot and load Windows 95 when you exit from the DOS session.

Windows 95 supports the two following ways of starting DOS mode:

- Using the current DOS configuration. The drivers and environment established by the Windows 95 **autoexec.bat** and **config.sys** files are used when the real mode DOS session starts.
- Using a new DOS configuration. This involves writing new **autoexec.bat** and **config.sys** files especially for this DOS session.

These environment options, therefore, offer three possible ways to run APT using Windows 95:

- A virtual DOS session under Windows 95 ([Section D.2](#)).
- A real mode DOS session using the existing Windows 95 **config.sys** and **autoexec.bat** files ([Section D.3](#)).
- A real mode DOS session using new **config.sys** and **autoexec.bat** files ([Section D.4](#)).

The relative advantages and disadvantages of these three methods of running APT under Windows 95 are discussed in the following pages.

Running APT under Windows 95 (continued)

Virtual DOS Environment under Windows 95

The virtual DOS environment allows you to run APT in a window on the Windows 95 desktop. This allows APT to be multi-tasked with other programs. APT can run on any type of video display in full screen mode. Some video cards can support DOS programs in graphic mode. If this is the case, APT can run in a window on the desktop. Otherwise, when APT is switched from full screen to windowed mode, Windows 95 suspends execution until APT is returned to full screen mode.

There are a number of advantages to running APT in a virtual DOS environment:

- APT loads immediately, taking about the same amount of time to load as it does on a DOS-only machine. There is no time delay to switch between Windows 95 and the DOS session.
- Windows 95 provides support for a number of devices and services that previously required drivers to be loaded from **autoexec.bat** or **config.sys**, thus taking up memory. Windows 95 provides support for LAN connections, CD-ROMs and other storage devices, disk-caching, and mouse support, to name a few. You can use any device that Windows 95 supports without taking memory from the DOS session and APT.
- Other programs can run at the same time APT is running. You do not have to exit from APT to use another program.
- Communications devices that use drivers, such as the Industrial Ethernet, work in this environment.

There are, however, some disadvantages to using APT from a virtual DOS environment:

- Only one instance of APT can be run at a time. APT's database is not designed to support access by multiple users on different programs at the same time. Only one instance of APT or any of APT's utilities can be running at any given time.
- There can be a performance penalty when running APT in this environment. The video is slower because it is being managed by Windows 95. Compile times can be slower because CPU time is consumed by other tasks. The actual performance degradation depends on the speed of the computer and the amount of CPU time required by other tasks.

Because of the multi-tasking in the Windows 95 environment, real-time operations such as communications may be adversely affected. Perhaps the most serious effect is on program download. To minimize the impact to your process during program download, make sure that the window in which APT is running is the active window. This allows download to proceed efficiently. Using the active window is also recommended for other APT utilities requiring communications, such as MAITT and Debug.

Although it is unlikely, APT running in a window that is not the active window may experience communications problems. The error prompt allows you to retry the operation when this occurs. During a program download, a communications failure could cause the operation to be aborted, thus leaving the program in an undefined state.

 **WARNING**

Downloading a program to the controller from APT in any window that is not the active window may result in delays or loss of communication, thus terminating program download.

A partially downloaded APT program can cause your process to react unpredictably, which can result in death and/or serious injury to personnel, and/or damage to equipment.

Always download your APT program from the active window.

APT's database, program structure, and the access key do not support access by multiple programs. Only one instance of APT can be run at a time. If you attempt to run more than one copy of APT at a time, or of the APT command line utilities, such as APTARCH, APTREST, TRANSOFF, and DC, problems may result. This means that you cannot leave APT running in one window and start another DOS window to run an APT utility.

CAUTION

APT does not support multiple accesses to its database or program information.

Attempting to run multiple instances of APT can cause unexpected loss or corruption of program data.

Do not run multiple copies of APT or any of the APT utilities at the same time.

Running APT under Windows 95 (continued)

Real Mode DOS Environment with Standard Configuration

The real mode DOS environment allows you to run APT with a DOS operating system, rather than as part of the multi-tasking Windows 95 operating system. This allows APT direct access to the hardware on the computer. The main reason for choosing the real mode DOS environment over the virtual DOS environment in Windows 95 is APT execution speed. In the real mode DOS environment, APT has exclusive use of system resources.

This version of the real mode DOS environment uses the existing real mode configuration established by Windows 95. Device drivers loaded in the **config.sys** and **autoexec.bat** files when Windows 95 originally booted up are left in place for the real mode DOS session. You must load any drivers that are required by the programs executed in this DOS environment into the Windows 95 **config.sys** file. The **autoexec.bat** drivers required by the DOS environment can be handled separately. You do not have to load them with Windows 95.

There are a number of advantages to running APT in this real mode DOS environment:

- Program communications are faster. Without the Windows 95 multi-tasking, the APT communications drivers get exclusive and uninterrupted access to the communications hardware on the computer. This results in faster program downloads, quicker MAITT program execution, and faster screen updates in Debug.
- APT performs better. Without the multi-tasking of Windows 95, APT does not have to share CPU time or other resources with any other programs. This results in faster screen updates and shorter program compile times.
- You can enter DOS mode without rebooting. When you use the standard configuration, Windows 95 does not have to reboot the computer in order to reload **config.sys**. Without the reboot, unloading Windows 95 and reloading real mode DOS only requires a few seconds.

There are some disadvantages to running in a real mode DOS environment as opposed to running in a virtual DOS environment under Windows 95.

- A reboot is required to reload Windows 95. When you exit from the DOS environment, the computer reboots, and Windows 95 performs a complete startup. This can take several minutes on some systems. It is inconvenient to switch from DOS mode back to Windows 95.
- The default **config.sys** file and DOS memory management for Windows 95 must support APT. The **cgistub.sys** device driver required by APT and any device drivers required by other DOS programs must be in the **config.sys** file used by Windows 95.
- DOS mode is a single-tasking environment. You must exit APT in order to run any other programs.
- The real mode DOS environment does not support Windows programs. If you need access to any Windows programs, you must exit this DOS environment and go back to Windows 95.

Running APT under Windows 95 (continued)

Real Mode DOS Environment with New Configuration

Like the real mode DOS environment with standard configuration, this real mode DOS environment allows you to run APT with a DOS operating system, rather than as part of the multi-tasking Windows 95 operating system. This allows APT direct access to the hardware on the computer. The main reason for choosing the real mode DOS environment over the virtual DOS environment in Windows 95 is APT execution speed. In the real mode DOS environment, APT has exclusive use of system resources.

In addition, this version of the real mode DOS environment uses its own **config.sys** and **autoexec.bat** files. This means that you do not need to load the drivers required by the programs executed in this DOS environment into the Windows 95 **config.sys** file. This can be useful if you currently use DOSMODE and APTMODE, because of memory usage problems between APT and other programs.

There are a number of advantages to running APT in a real mode DOS environment with new configuration:

- Program communications are faster. Without the Windows 95 multi-tasking, the APT communications drivers get exclusive and uninterrupted access to the communications hardware on the computer. This results in faster program downloads, quicker MAITT program execution, and faster screen updates in Debug.
- APT performs better. Without the multi-tasking of Windows 95, APT does not have to share CPU time or other resources with any other programs. This results in faster screen updates and shorter program compile times.
- Windows 95 does not have to support APT. This real mode DOS environment has its own **autoexec.bat** and **config.sys** files. It does not depend on Windows 95 for any of its device drivers. This allows the APT environment to be set up differently if it cannot function in your normal Windows 95 environment or with other DOS programs because of memory limitations.

There are some disadvantages to running APT in this real mode DOS environment, as opposed to running it in a virtual DOS environment or in a real mode DOS environment that uses the Windows 95 **config.sys** file.

- You must reboot in order to load both DOS and Windows 95. When you start the DOS session, the computer reboots so the new **config.sys** and **autoexec.bat** can be loaded. When you exit from the DOS session, the computer reboots, and Windows 95 performs a complete startup. It can require several minutes on some systems either to load DOS or to load Windows. It is inconvenient to switch back and forth between Windows 95 and DOS.
- DOS mode is a single-tasking environment. You must exit APT in order to run any other programs.
- The real mode DOS environment does not support Windows programs. If you need access to any Windows programs, you must exit from DOS and go back to Windows 95.

D.2 Installing APT in a Virtual DOS Environment

Overview

To install APT in a virtual DOS environment, you must perform the following tasks:

- Create an icon for the DOS window and configure its properties sheet.
- Configure the memory management for the DOS environment so there is enough conventional memory to run APT.
- Install APT using the DOS window that you have configured.
- Set up an APT configuration batch file and **autoexec.bat** file.
- Configure an APT icon and run APT in the DOS window.

Creating a DOS Icon for APT

To make things easier for the rest of the installation, you can create an icon for APT's DOS window that provides a shortcut to DOS. You need to configure the icon to run a DOS session full screen. Since there is usually an MS-DOS shortcut in the Start Menu, you can make a copy and drop it on the desktop. To create an icon for the DOS window for APT, follow the steps below:

1. Click the right mouse button over a blank spot on the taskbar, and select **Properties** from the menu. The system displays the Taskbar Properties dialog box (Figure D-1).

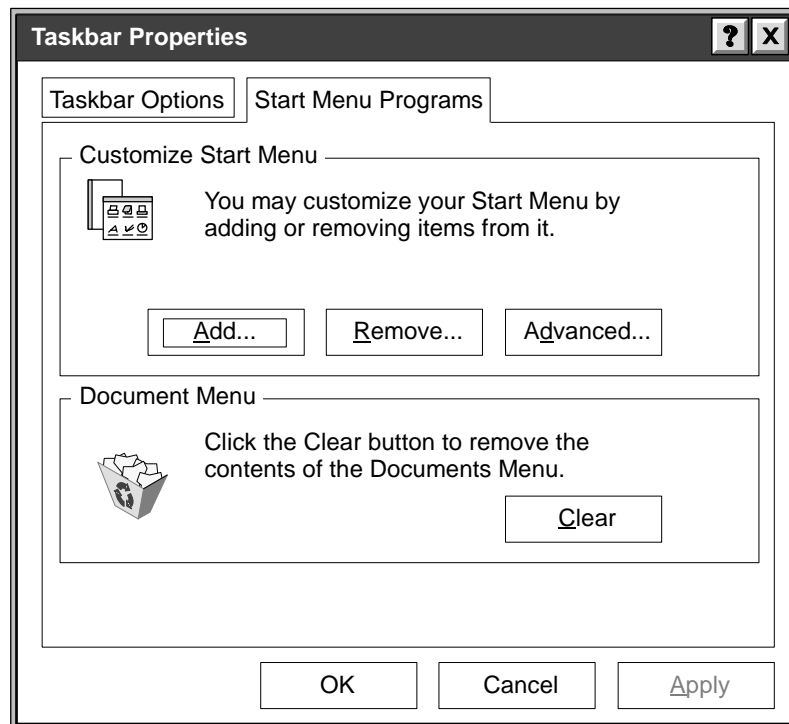


Figure D-1 Taskbar Properties Dialog Box

2. Select the **Start Menu Programs** tab and click on the **Advanced** button in the Customize Start Menu. This opens an explorer window for the Start Menu (Figure D-2).
3. Locate an MS-DOS shortcut, identified in the Type column in the Programs directory. Click the right mouse button on the MS-DOS shortcut and select **Create Shortcut** from the menu to create a copy of the MS-DOS shortcut in the explorer window.
4. Click the left mouse button on the copy of the MS-DOS shortcut, drag it out of the explorer window, and drop it on the desktop.

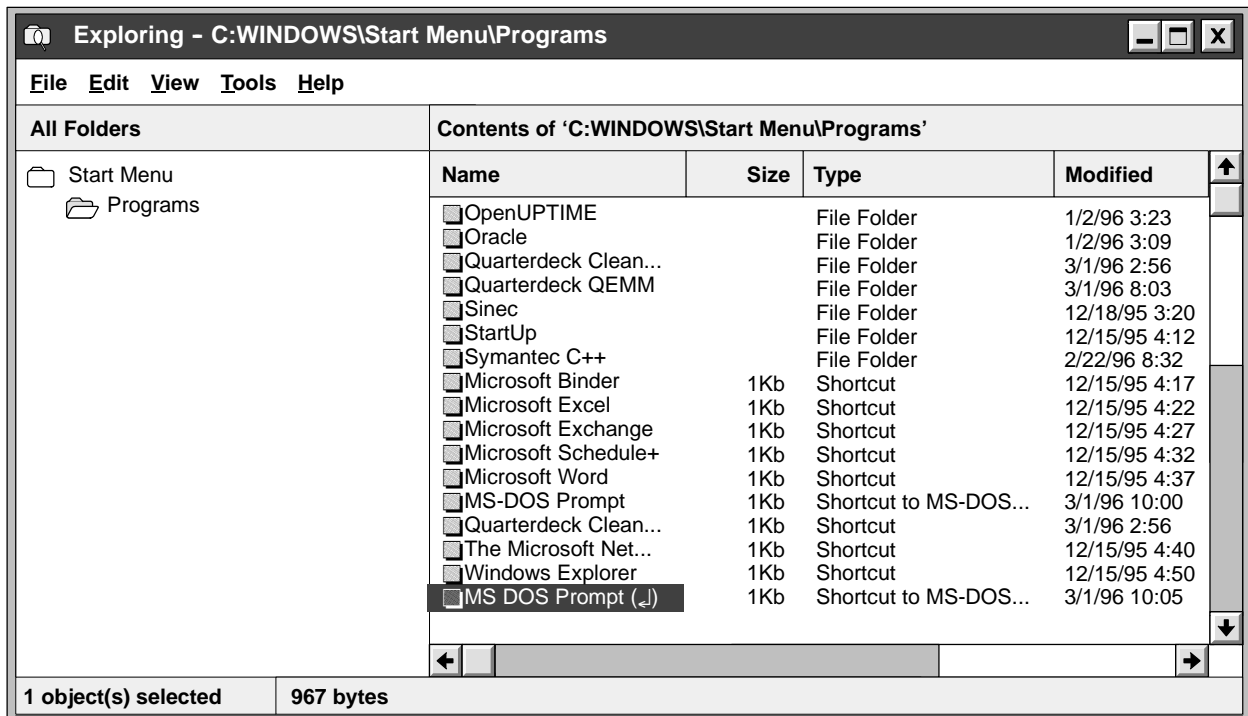


Figure D-2 Start Menu Program Explorer Window

Installing APT in a Virtual DOS Environment (continued)

Setting the Properties for the DOS Window

You have now created an MS-DOS shortcut icon for the DOS window for APT. The next task is to configure the properties sheet for this icon. There are five tabs on the properties sheet, and you must enter a configuration for each of them except the **Font** tab. Follow the steps below.

1. Click the right mouse button on the new icon, and select **Properties** from the menu. The system displays the MS-DOS Prompt Properties dialog box (Figure D-3).

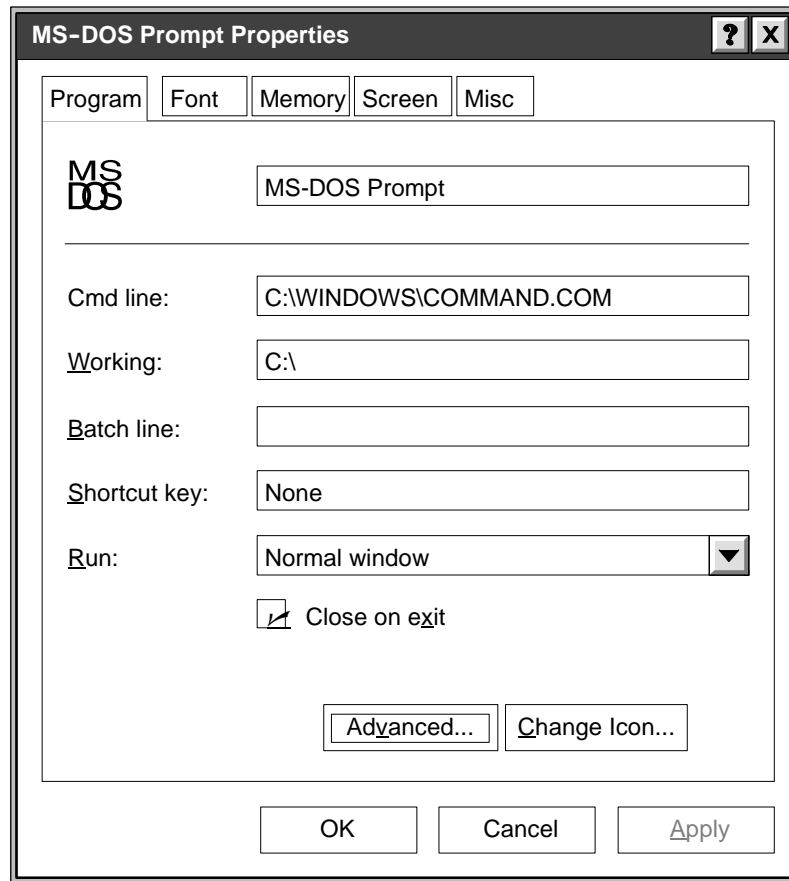


Figure D-3 MS-DOS Prompt Properties Dialog Box

2. Select the **Program** tab, and then select the **Advanced** button to access the Advanced Program Settings dialog box (Figure D-4).

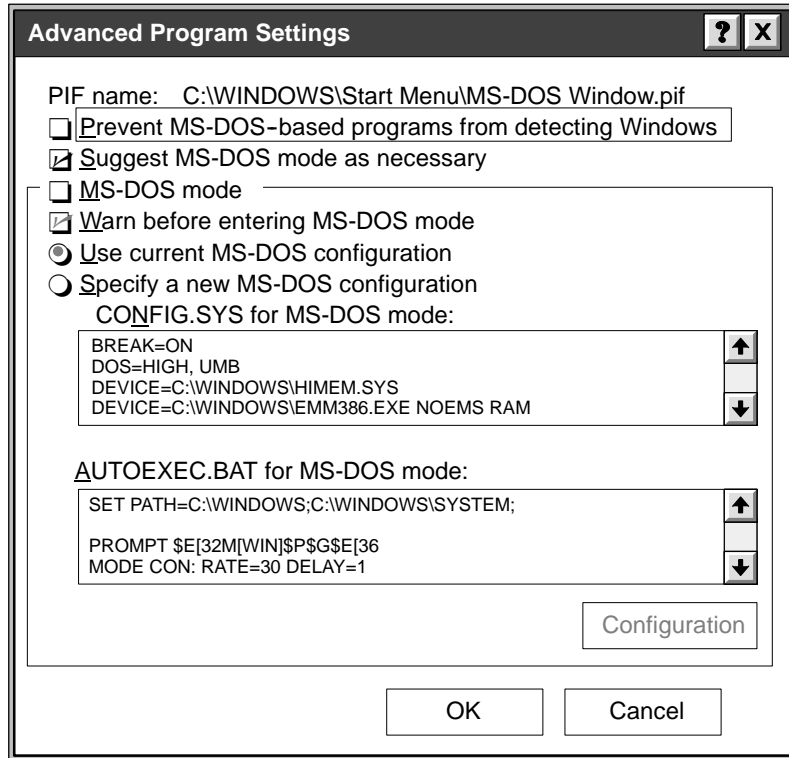


Figure D-4 Advanced Program Settings Dialog Box

3. Make sure the MS-DOS Mode box is not checked, and then select **OK**.

MS-DOS Mode [] (not checked)

Result: The Advanced Program Settings dialog box closes, returning you to the Program folder.

4. In the Program folder, set the following properties:

Cmd line	C:\WINDOWS\COMMAND.COM
Working	C:\
Batch File	
Shortcut Key	
Run	Normal Window
Close on Exit	[✓] (checked)

In the Program folder, the default is for APT and Windows 95 to be installed on the C: drive. If this is not the case, change the drive letter in the Command Line and Working fields to the appropriate drive.

Installing APT in a Virtual DOS Environment (continued)

- APT requires 520K of conventional memory in order to run. APT does not use expanded, extended, or protected-mode memory, nor does it use HMA blocks. Select the **Memory** tab and set the following properties:

Conventional Memory	Auto
Protected	<input type="checkbox"/> (not checked)
Initial Environment	1024
Expanded (EMS) Memory	None
Extended (XMS) Memory	None
Uses HMA	<input type="checkbox"/> (not checked)
Protected-mode (DPMI) Memory	Auto

- You must set the DOS session initially to run APT full screen because some display adapters do not support VGA graphics emulation in an overlapped window. You must set the number of lines on the text display at 25 so the compiler screen displays correctly. Select the **Screen** tab and set the following properties, leaving everything else unmarked:

Full Screen	<input checked="" type="checkbox"/> (checked)
Initial Size	25 lines
Restore settings on startup	<input checked="" type="checkbox"/> (checked)
Fast ROM Emulation	<input checked="" type="checkbox"/> (checked)

- To cause Windows 95 to give APT more CPU time, idle sensitivity should be set to **Low**. Select the **Misc** tab, and set the following properties, leaving everything else unmarked:

Idle sensitivity	Low (all the way)
Warn if still active	<input checked="" type="checkbox"/> (checked)
Windows shortcut keys	<input checked="" type="checkbox"/> (check all)

- Select **OK** at the bottom of the MS-DOS Prompt Properties dialog box to save the changes.

You have now configured an MS-DOS icon for starting a virtual DOS session that runs full screen with an 80-character by 25-line display.

Configuring Memory Management

Although the Windows 95 environment has no 640K memory limits, the virtual DOS session still does. A virtual DOS session requires memory management of the 640K conventional memory and upper memory just like the older versions of MS-DOS.

Windows 95 provides support for the mouse, networking, disk caching, and most other types of system hardware. You do not have to load drivers for these devices in the memory of the virtual DOS session. This frees up memory for use with APT and allows you to use devices that otherwise would not have room for their drivers. Windows 95 does require space for a windows driver and a virtual memory driver.

The **config.sys** file has to support DOS memory management, DOS Shell utilities, and APT. A sample **config.sys** file suitable for installing APT is listed below.

```
BREAK=ON
DOS=HIGH,UMB
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE NOEMS RAM
DEVICEHIGH=C:\WINDOWS\COMMAND\ANSI.SYS
```

If you are not running DOS programs that require expanded memory (EMS), then include the NOEMS parameter with EMM386. Supporting EMS requires a 64K page frame in upper memory. This is 64K that would otherwise be available for using the `devicehigh` or `loadhigh` commands. If you are currently using EMM386 or an equivalent, make sure to include the command line parameters currently in use.

If you have other entries in your **config.sys** that are needed for your computer system, be sure to include those also. **ansi.sys** is not required to run APT; however, it is a common addition to **config.sys** and is included in the example. Under Windows 95, the only drivers that you need to include in the **config.sys** file are drivers specific to the DOS shell, such as **ansi.sys**.

Installing APT in a Virtual DOS Environment (continued)

The **autoexec.bat** file that you need at this point is shown below. The **prompt** command sets the prompt string and font colors for all DOS sessions. The **mode** command speeds up the key repeat rate; this is a useful setting for APT. The **doskey** entries set up commands for switching between 25 and 50 lines of text. Notice the **lh** (**loadhigh**) command in front of the first **doskey** command. This command loads the specified program into high memory, if available. These are handy additions to the APT environment, but none of them are required, and you can leave them out if you prefer. A sample **autoexec.bat** file suitable for installing APT is listed below.

```
set path= c:\windows;c:\windows\system;.

prompt $e[32m[WIN]$p$g$e[36m
mode con: rate=30 delay=1

set tmp=c:\temp
set temp=c:\temp

lh doskey /bufsize=1024 /insert
doskey 25=mode con: lines=25_cls
doskey 50=mode con: lines=50_cls
```

After you reboot the computer with the new **config.sys** and **autoexec.bat** files, start a DOS session with the icon that you have created. Check the memory usage by using the `mem /c` command. The memory should look like the chart in [Figure D-5](#).

```
[WIN] C:\>mem /c
Modules using memory below 1 MB:

```

Name	Total	Conventional	Upper Memory
SYSTEM	18,720 (18K)	10,544 (10K)	8,176 (8K)
HIMEM	1,168 (1K)	1,168 (1K)	0 (0K)
EMM386	4,320 (4K)	4,320 (4K)	0 (0K)
WIN	3,616 (4K)	3,616 (4K)	0 (0K)
vmm32	55,840 (55K)	864 (1K)	54,976 (54K)
COMMAND	8,112 (8K)	8,112 (8K)	0 (0K)
ANSI	4,320 (4K)	0 (0K)	4,320 (4K)
IFSHLP	2,864 (3K)	0 (0K)	2,864 (3K)
SETVER	832 (1K)	0 (0K)	832 (1K)
DOSKEY	5,200 (5K)	0 (0K)	5,200 (5K)
Free	626,448 (612K)	626,448 (612K)	0 (0K)

```

Memory Summary:

```

Type of Memory	Total	Used	Free
Conventional	655,360	28,912	626,448
Upper	76,368	76,368	0
Reserved	393,216	393,216	0
Extended (XMS)	13,555,120	13,555,120	0
Total memory	14,680,064	14,053,616	626,448
Total under 1 MB	731,728	105,280	626,448
Largest executable program size		626,432 (612K)	
Largest free upper memory block		0 (0K)	
MS-DOS is resident in the high memory area.			

```
[WIN] C:\>
```

Figure D-5 Example of Memory Usage Report

Installing APT in a Virtual DOS Environment (continued)

In the “Modules using memory below 1 MB” chart in [Figure D-5](#), the row for free memory should have at least 600K of conventional memory free; the example has 612K. Notice that **vmm32** takes a sizable chunk of upper memory. The **vmm32** module is required to take a small amount of conventional memory; after that it allocates all remaining upper memory, bringing the total free upper memory to 0K. This prevents the `loadhigh` command from being used in a virtual DOS window. You must put anything that you want to load into high memory in the **autoexec.bat** file.

If you add drivers to upper memory, **vmm32** can reduce its memory allocation, so there is still room in upper memory for drivers if they are loaded from **autoexec.bat**. If there is no upper memory, this usually indicates a problem with **himem.sys** or **emm386.exe**. At this point, you do not need to add anything to the **autoexec.bat** file.

Installing APT

The DOS session is now prepared for APT installation. Start the APT installation program by following the “CD-ROM on Windows Systems” procedure on [page A-6](#) or the “Diskettes on Windows Systems” procedure on [page A-7](#). Follow the directions until the last screen appears. Press **F10** and take the selection that lets the installation program update the **autoexec.bat** and **config.sys**. Then reboot the computer. Windows 95 does not allow the DOS-based installation program to reboot the computer. The installation program can only close the DOS session.

Installing APT in a Virtual DOS Environment (continued)

When the installation program is finished, you need to remove some lines from the modified **autoexec.bat** and **config.sys** files. Remove the mouse from the **autoexec.bat** file. The mouse interface is provided by Windows 95.

```
set path=c:\apt;c:\windows;c:\windows\system;.

prompt $e[32m[WIN]$p$g$e[36m
mode con: rate=30 delay=1

set tmp=c:\temp
set temp=c:\temp

lh doskey /bufsize=1024 /insert
doskey 25=mode con: lines=25_cls
doskey 50=mode con: lines=50_cls

set cgipath=c:\apt
c:\apt\drivers.exe
set mouse=msmouse
mouse
set apt_mode=direct
set apt_port=1
set resolution=75
```

← remove this line

Remove the buffers and shell commands from the **config.sys** file. Windows 95 provides suitable defaults for these parameters.

```
BREAK=ON
DOS=HIGH,UMB
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE NOEMS RAM
DEVICEHIGH=C:\WINDOWS\COMMAND\ANSI.SYS

shell=c:\command.com /p /e:500
device=c:\apt\drivers\cgistub.sys /r
buffers=10
files=40
```

← remove this line

← remove this line

The **files=40** command allocates forty file handles for DOS programs to use. This sets the maximum number of files that can be opened at one time. Forty file handles represent the minimum number required for DOS and APT to operate together.

After the files have been saved, select **Shutdown** from the Start Menu and select **Restart the Computer** from the dialog box. When the computer reboots, you can make any final adjustments to the APT environment. At this point, when you start the DOS session and enter the `mem /c` command, the memory should look like the chart in [Figure D-6](#).

```
[WIN] C:\>mem /c
Modules using memory below 1 MB:

```

Name	Total	Conventional	Upper Memory
SYSTEM	19,328 (19K)	10,560 (10K)	8,768 (9K)
HIMEM	1,168 (1K)	1,168 (1K)	0 (0K)
EMM386	4,320 (4K)	4,320 (4K)	0 (0K)
CGISTUB	64 (0K)	64 (0K)	0 (0K)
WIN	3,696 (4K)	3,696 (4K)	0 (0K)
vmm32	58,864 (57K)	4,480 (4K)	54,384 (53K)
DRIVERS	78,160 (76K)	78,160 (76K)	0 (0K)
COMMAND	8,176 (8K)	8,176 (8K)	0 (0K)
ANSI	4,320 (4K)	0 (0K)	4,320 (4K)
IFSHLP	2,864 (3K)	0 (0K)	2,864 (3K)
SETVER	832 (1K)	0 (0K)	832 (1K)
DOSKEY	5,200 (5K)	0 (0K)	5,200 (5K)
Free	544,400 (532K)	544,400 (532K)	0 (0K)

```

Memory Summary:

```

Type of Memory	Total	Used	Free
Conventional	655,360	110,960	544,400
Upper	76,368	76,368	0
Reserved	393,216	393,216	0
Extended (XMS)	13,555,120	13,555,120	0
Total memory	14,680,064	14,135,664	544,400
Total under 1 MB	731,728	187,328	544,400
Largest executable program size		544,384 (532K)	
Largest free upper memory block		0 (0K)	

```

MS-DOS is resident in the high memory area.
[WIN] C:\>

```

Figure D-6 Memory Usage

Installing APT in a Virtual DOS Environment (continued)

Locate the entry for CGISTUB that is loaded from **config.sys** and the entry for DRIVERS that is loaded from **autoexec.bat**. If anything did not load, you can go through a step-by-step bootup. To do this, shut down Windows 95 and reboot the computer. When the reboot procedure starts, press the F8 key until the boot menu appears. Select **Step by Step** bootup; it is usually number 5. As the computer boots up, it prompts you at each line in the **config.sys** and **autoexec.bat** files.

When everything loads successfully, there should still be 532K of free conventional memory. All of APT's environment variables should be present. Type `set` and press `[Enter]` to display all of the environment variables. At the minimum, the following APT environment variables must be configured:

```
CGIPATH=c:\apt
MOUSE=msmouse
APT_MODE=direct
APT_PORT=1
RESOLUTION=75
```

If any of these variables are not present, check the **autoexec.bat** file. An icon, **{apt.ico}**, and the following sample PIF files are placed on your hard drive under **\apt** by the installation process:

- **apt_vdos.pif** - virtual DOS environment
- **apt_sdos.pif** - real mode DOS environment with standard configuration
- **apt_ndos.pif** - real mode DOS environment with new configuration

You can use these files as needed.

Writing a Configuration Batch File

The APT environment variables and the **drivers.exe** program should be in the **autoexec.bat** file. You can either leave them in the **autoexec.bat** file, or put them in an APT configuration batch file and have the DOS session run the batch file as it loads. The advantage of using a configuration file is that the 70K+ drivers program is not loaded in every DOS session. The drivers program is removed from **autoexec.bat** and is only loaded into a DOS session if the APT configuration batch is executed.

To create an APT configuration batch, you need to remove the APT commands from the **autoexec.bat** and put them in the configuration batch. First, open a copy of the note pad and load the **autoexec.bat** file. Select the section of the file with the APT settings and perform a cut command. Next, open a second copy of the note pad and paste the contents from the **autoexec.bat** file. Finally, select a name for the second file such as **aptnfg.bat** and save the contents of the file. Do not put the configuration file in the APT directory, because it will be deleted the next time you do an APT installation. You can save the file as **c:\aptnfg.bat**. These are the lines that should be removed from the **autoexec.bat** file and placed in the **aptnfg.bat** file:

```
set cgipath=c:\apt
c:\apt\drivers.exe
set mouse=msmouse
set apt_mode=direct
set apt_port=1
set resolution=75
```

You need to add the **aptnfg.bat** file to the properties sheet of the DOS icon. Display the properties sheet of the DOS icon and select the Program tab. The **aptnfg.bat** file goes in the Batch file field:

Batch file: c:\aptnfg.bat

Now when you double click on the DOS icon, the **aptnfg.bat** file executes while the window is starting.

Installing APT in a Virtual DOS Environment (continued)

Running APT in the Window

After APT has been successfully installed, you can start the DOS session from the icon. Make sure the window is in full screen mode. Check the memory configuration by using the `mem /c` command to make sure that the required APT device drivers are present, and that 532K+ of conventional memory is free. Check the environment variables with the `set` command to make sure that all required APT environment variables are present. If everything checks out, you should now be able to invoke and run APT with no problems. Type `apt` at the DOS prompt.

If you want, you can set up the DOS icon to run APT directly instead of the DOS shell. To do this, you need to change the `Cmd` line from `command.com` to `apt.exe` and change the icon to the APT symbol. Display the properties sheet of the DOS icon and select the **Program** tab. Change the `Cmd` line property to the **apt.exe** program:

Cmd line: `c:\apt\apt.exe`

Next, change the top line beside the icon from MS-DOS Prompt to APT.

Title: APT

Now click on the **Change Icon** button. To select the APT icon, enter the following command in the File Name field of the Change Icon dialog box (Figure D-7):

`c:\apt\apt.ico`

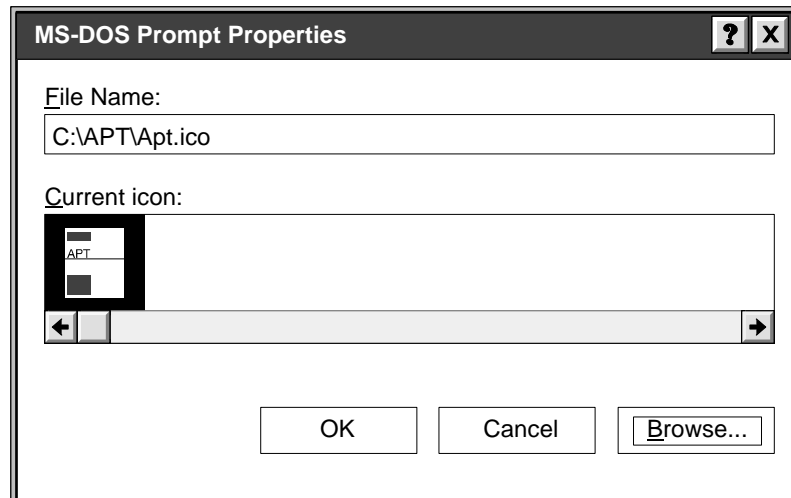


Figure D-7 Change Icon Dialog Box

Select **OK** for both the Change Icon and the MS-DOS Prompt Properties dialog boxes. After you finish with the properties sheet, you can change the icon label on the desktop. Click once, and then a second time, on the icon label, but not fast enough for a double click. This selects the text of the icon label and allows you to edit the text to say APT.

When you click on the APT icon, a full screen DOS session should start and load APT automatically. APT is now successfully installed on your Windows 95 system and ready to use.

**Resource Sharing
with Other DOS
and Windows
Programs**

When you run APT in a window, APT must share hardware resources with other programs running on your system, such as the serial ports and the printer port. When APT is using a serial port, Windows 95 does not allow other programs to access it. Although all APT print requests are placed in the Windows 95 print spooler, APT requires direct hardware control to access the copyright protection key. When any program directly accesses the parallel port, Windows 95 brings up an arbitration window, and you must select which program gets exclusive access to the port. Assign control to the window running APT.

D.3 Installing APT in Real Mode DOS Environment (Standard Configuration)

Overview

To install APT in a real mode DOS environment using the standard configuration, you must perform the following tasks:

- Create an icon for the DOS window and configure its properties sheet.
- Configure the memory management for the DOS environment so there is enough conventional memory to run APT.
- Install APT using the DOS window that you have configured.
- Set up an **autoexec.bat** file for APT.
- Configure an APT icon and run APT in the DOS window.

Creating a DOS Icon for APT

To make things easier for the rest of the installation, you can create an icon for APT's DOS window that provides a shortcut to DOS. You need to configure the icon to run a real mode DOS session. Since there is usually an MS-DOS shortcut in the Start Menu, you can make a copy and drop it on the desktop. Follow the procedure on [pages D-10 through D-11](#), and then set the properties for the DOS window according to the procedure which begins on [page D-27](#).

Setting the Properties for the DOS Window

You have now created an MS-DOS shortcut icon for the DOS window for APT. The next task is to configure the properties sheet for this icon. Follow the steps below.

1. Click the right mouse button on the new icon, and select **Properties** from the menu. The system displays the MS-DOS Mode Properties dialog box (Figure D-8).

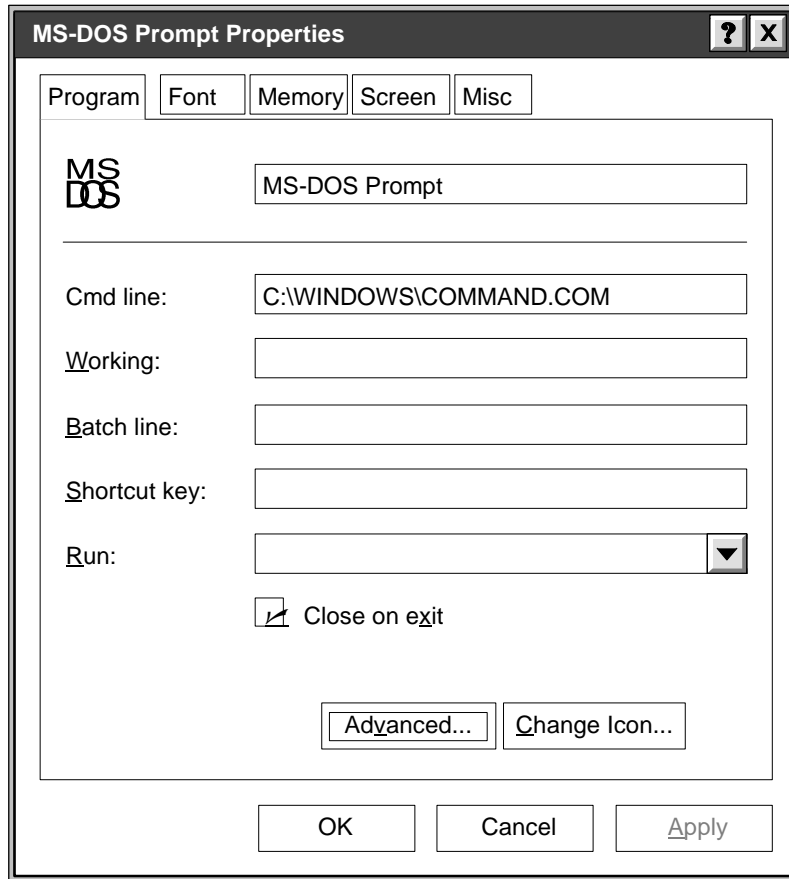


Figure D-8 MS-DOS Mode Properties Dialog Box

2. Select the **Program** tab, and then select the **Advanced** button to access the Advanced Program Settings dialog box (Figure D-9).

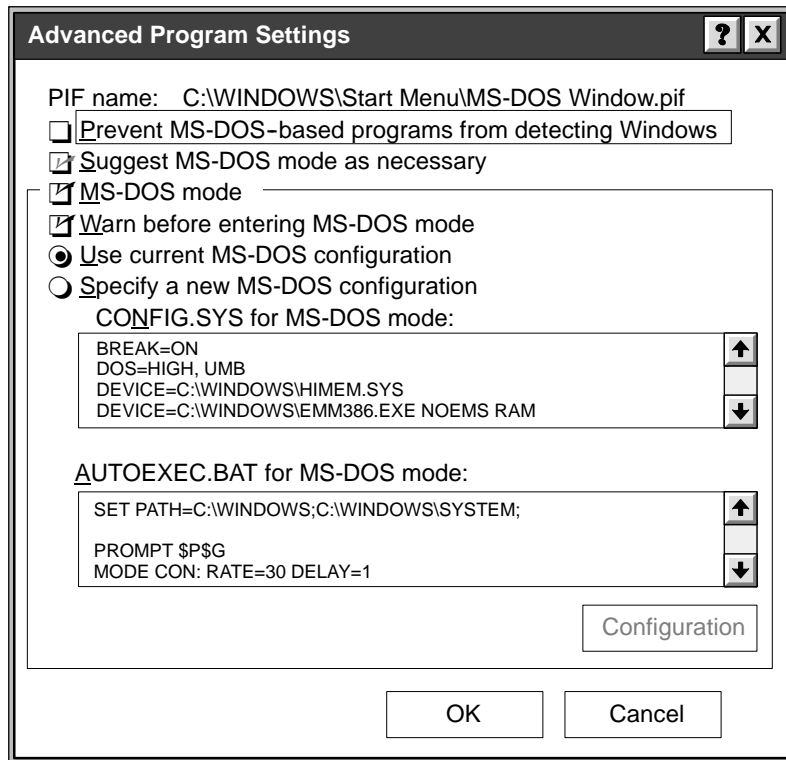


Figure D-9 Advanced Program Settings Dialog Box

3. Make sure the MS-DOS Mode box is checked, and select the **Use Current MS-DOS Configuration** button. Then select **OK**.

Since you have selected real mode DOS, you do not need to configure the Working, Batch File, Shortcut Key, and Run fields.

4. In the Program folder, set the following property:

Cmd line C:\WINDOWS\COMMAND.COM

In the Program folder, the default is for APT and Windows 95 to be installed on the C: drive. If this is not the case, change the drive letter in the Cmd line field to the appropriate drive.

You do not need to configure the Font, Memory, Screen, and Misc folders, since they are not used in DOS Mode.

5. Select **OK** at the bottom of the MS-DOS Prompt Properties dialog box to save the changes.

You have now configured an MS-DOS icon for starting a real mode DOS session.

Configuring Memory Management

A real mode DOS session is essentially the same environment provided by older versions of MS-DOS. DOS Mode requires memory management of the 640K conventional memory and upper memory just like the older versions of MS-DOS.

Windows 95 does not provide any type of support for a real mode DOS session. No support is included for the mouse, networking, disk caching, and most other types of system hardware. You must load drivers for these items into **config.sys**, which uses up memory that would otherwise be available to APT. Devices that are available under Windows 95 may have to be omitted in DOS Mode due to memory limitations.

The **config.sys** file must support all programs and devices to be used in this DOS environment. A sample **config.sys** file suitable for installing APT is listed below. See [Figure D-9](#).

```
BREAK=ON
DOS=HIGH,UMB
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE NOEMS RAM
```

If you are not running DOS programs that require expanded memory (EMS), then the NOEMS parameter should be included with **emm386**. Supporting EMS requires a 64K page frame in upper memory. This is 64K that would otherwise be available for using the `devicehigh` or `loadhigh` commands. If you are currently using EMM386 or an equivalent, be sure to include the command line parameters currently in use.

If there are other entries in your **config.sys** that you need for your computer system, be sure to include those also. The virtual DOS environment example in [Section D.2](#) included **ansi.sys**, which is optional and is not required to run APT. Due to the addition of **smartdrv.exe** and **mouse.exe**, some computer systems may not have room to include **ansi.sys**, so it is omitted from this example.

Installing APT in Real Mode DOS Environment — Standard (continued)

The **autoexec.bat** file that you need at this point is shown below. The `prompt` command sets the prompt string for all DOS sessions. The `mode` command speeds up the key repeat rate, which is a useful setting for APT. The virtual DOS environment example included **doskey**, which is optional and is not required to run APT. Due to the addition of **smartdrv.exe** and **mouse.exe**, some computer systems may not have room to include **doskey**, so it is omitted from the example. A sample **autoexec.bat** file suitable for installing APT is listed below. See [Figure D-9](#).

```
set path=c:\windows;c:\windows\system;

prompt $p$g
mode con: rate=30 delay=1

set tmp=c:\temp
set temp=c:\temp
```

Windows 95 has a new batch file called **dosstart.bat**, which is located in the **windows** directory. If DOS Mode is configured to use the current **autoexec.bat** and **config.sys** files loaded by Windows 95, **dosstart.bat** is executed when transitioning from Windows 95 to DOS Mode. This file is used to run programs that are needed in a real mode DOS environment but should not be run for a Windows 95 environment. This includes programs such as **mouse.exe** and **smartdrv.exe**. **Smartdrv.exe** or an equivalent disk cache should be loaded in order to keep APT's compile times efficient. **Mouse.exe** is required if you want to use a Microsoft mouse. The **mouse.exe** file is not included in Windows 95, so it is not in the **windows** directory. In the example, the file is located in the **\dos** directory. The location on your computer may be different. The sample **dosstart.bat** file is listed below:

```
lh c:\dos\mouse.exe
lh c:\windows\smartdrv.exe
```

After you reboot the computer with the new **config.sys** and **autoexec.bat** files, start a DOS session with the icon previously created. This executes the new **dosstart.bat** file. Check the memory usage by using the `mem /c` command. The memory should look like the chart in [Figure D-10](#).

```
C:\>mem /c
Modules using memory below 1 MB:

```

Name	Total	Conventional	Upper Memory
SYSTEM	18,720 (18K)	10,560 (10K)	8,160 (8K)
HIMEM	1,168 (1K)	1,168 (1K)	0 (0K)
EMM386	4,320 (4K)	4,320 (4K)	0 (0K)
CGISTUB	64 (0K)	64 (0K)	0 (0K)
WIN	3,632 (4K)	3,632 (4K)	0 (0K)
MOUSE	23,856 (23K)	272 (0K)	23,584 (23K)
SMARTDRV	29,040 (28K)	12,640 (12K)	16,400 (16K)
COMMAND	7,168 (7K)	0 (0K)	7,168 (7K)
IFSHLP	2,864 (3K)	0 (0K)	2,864 (3K)
SETVER	832 (1K)	0 (0K)	832 (1K)
DOSKEY	5,200 (5K)	0 (0K)	5,200 (5K)
COMMAND	7,168 (7K)	0 (0K)	7,168 (7K)
Free	627,440 (613K)	622,464 (608K)	4,976 (5K)

```

Memory Summary:

```

Type of Memory	Total	Used	Free
Conventional	655,360	32,896	622,464
Upper	76,352	71,376	4,976
Reserved	131,072	131,072	0
Extended (XMS)	24,303,040	2,360,768	21,942,272
Total memory	25,165,824	2,596,112	22,569,712
Total under 1 MB	731,712	104,272	627,440
Largest executable program size		622,336 (608K)	
Largest free upper memory block		3,632 (4K)	
MS-DOS is resident in the high memory area.			

```
C:\>
```

Figure D-10 Memory Usage Report for Standard Configuration

In the “Modules using memory below 1 MB” chart in [Figure D-10](#), the row for free memory should have at least 600K free of conventional memory; the example has 608K. If there is no upper memory, this usually indicates a problem with **himem.sys** or **emm386.exe**.

Installing APT

The DOS session is now prepared for APT installation. Start the APT installation program by following the “CD-ROM on DOS Systems” procedure on [page A-6](#) or the “Diskettes on DOS Systems” procedure on [page A-7](#). Follow the directions until the last screen appears. Press **F10** and select the option that lets the installation program update the **autoexec.bat** and **config.sys** files. Then reboot the computer. Since you are running real mode DOS, the computer modifies the files and reboots back into Windows 95.

When the installation program is finished, you need to remove some lines from the modified **autoexec.bat** and **config.sys** files. Earlier in this procedure, you placed the mouse command in the **dosstart.bat** file. Now remove the mouse command from the **autoexec.bat** file.

```
set path=c:\apt;c:\windows;c:\windows\system;

prompt$p$g
mode con: rate=30 delay=1

set tmp=c:\temp
set temp=c:\temp

set cgipath=c:\apt
c:\apt\drivers.exe
set mouse=msmouse
mouse                                     ← remove this line
set apt_mode=direct
set apt_port=1
set resolution=75
```

Remove the buffers and shell commands from the **config.sys** file. Windows 95 provides suitable defaults for these parameters.

```
BREAK=ON
DOS=HIGH,UMB
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE NOEMS RAM

shell=c:\command.com /p /e:500           ← remove this line
device=c:\apt\drivers\cgistub.sys /r
buffers=10                               ← remove this line
files=40
```

The **files=40** command allocates forty file handles for DOS programs to use. This sets the maximum number of files that can be opened at one time. Forty file handles represent the minimum number required for DOS and APT to operate together.

After the files have been saved, select **Shutdown** from the Start Menu and select **Restart the Computer** from the dialog box. When the computer reboots, you can make any final adjustments to the APT environment. At this point, when you start the DOS session and enter the `mem /c` command, the memory should look like the chart in [Figure D-11](#).

```
[WIN]C:\>mem /c
Modules using memory below 1 MB:

```

Name	Total	Conventional	Upper Memory
SYSTEM	18,720 (18K)	10,560 (10K)	8,160 (8K)
HIMEM	1,168 (1K)	1,168 (1K)	0 (0K)
EMM386	4,320 (4K)	4,320 (4K)	0 (0K)
CGISTUB	64 (0K)	64 (0K)	0 (0K)
WIN	3,696 (4K)	3,696 (4K)	0 (0K)
DRIVERS	78,160 (76K)	78,160 (76K)	0 (0K)
MOUSE	23,856 (23K)	272 (0K)	23,584 (23K)
SMARTDRV	29,040 (28K)	12,640 (12K)	16,400 (16K)
COMMAND	7,168 (7K)	0 (0K)	7,168 (7K)
COMMAND	7,168 (7K)	0 (0K)	7,168 (7K)
IFSHLP	2,864 (3K)	0 (0K)	2,864 (3K)
SETVER	832 (1K)	0 (0K)	832 (1K)
Free	554,368 (541K)	544,192 (531K)	10,176 (10K)

```

Memory Summary:

```

Type of Memory	Total	Used	Free
Conventional	655,360	111,168	544,192
Upper	76,352	66,176	10,176
Reserved	131,072	131,072	0
Extended (XMS)	24,303,040	2,360,768	21,942,272
Total memory	25,165,824	2,669,184	22,496,640
Total under 1 MB	731,712	177,344	554,368
Largest executable program size		543,792 (531K)	
Largest free upper memory block		6,976 (7K)	
MS-DOS is resident in the high memory area.			

```
[WIN]C:\>
```

Figure D-11 Memory Usage Report for Real Mode DOS Environment

Installing APT in Real Mode DOS Environment — Standard (continued)

Locate the entry for CGISTUB that is loaded from **config.sys** and the entry for DRIVERS that is loaded from **autoexec.bat**. If anything did not load, you can go through a step-by-step bootup. To do this, shut down Windows 95 and reboot the computer. When the reboot procedure starts, press the **[F8]** key until the boot menu appears. Select **Step by Step** bootup; it is usually number 5. As the computer boots up, it prompts you at each line in the **config.sys** and **autoexec.bat** files.

When everything loads successfully, there should still be 531K of free conventional memory. All of APT's environment variables should be present. Type `set` and press **[Enter]** to display all of the environment variables. At the minimum, the following APT environment variables must be configured:

```
CGIPATH=c:\apt
MOUSE=msmouse
APT_MODE=direct
APT_PORT=1
RESOLUTION=75
```

If any of these variables are not present, check the **autoexec.bat** file.

D.4 Installing APT in Real Mode DOS Environment with New Configuration

Overview

To install APT in a real mode DOS environment with a customized configuration, you must perform the following tasks:

- Create an icon for the DOS window and configure its properties sheet.
- Configure the memory management for the DOS environment so there is enough conventional memory to run APT.
- Install APT using the DOS window that you have configured.
- Set up an APT **config.sys** file and an **autoexec.bat** file.
- Configure an APT icon and run APT in the DOS window.

Creating a DOS Icon for APT

To make things easier for the rest of the installation, you can create an icon for APT's DOS window that provides a shortcut to DOS. You need to configure the icon to run a real mode DOS session on the desktop. Since there is usually an MS-DOS shortcut in the Start Menu, you can make a copy and drop it on the desktop. Follow the procedure on [pages D-10 through D-11](#), and then set the properties for the DOS window according to the procedure which begins on [page D-37](#).

Setting the Properties for the DOS Window

You have now created an MS-DOS shortcut icon for the DOS window for APT. The next task is to configure the properties sheet for this icon. Follow the steps below.

1. Click the right mouse button on the new icon, and select **Properties** from the menu. The system displays the MS-DOS Prompt Properties dialog box (Figure D-12).

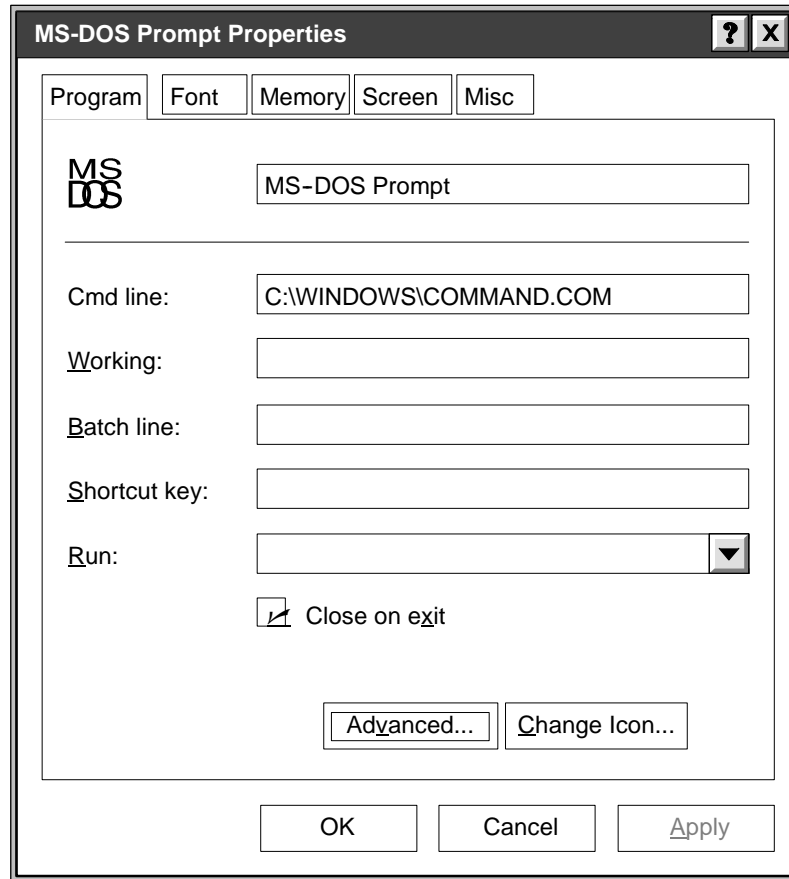


Figure D-12 MS-DOS Prompt Properties Dialog Box

2. Select the **Program** tab, and then select the **Advanced** button to access the Advanced Program Settings dialog box (Figure D-13).

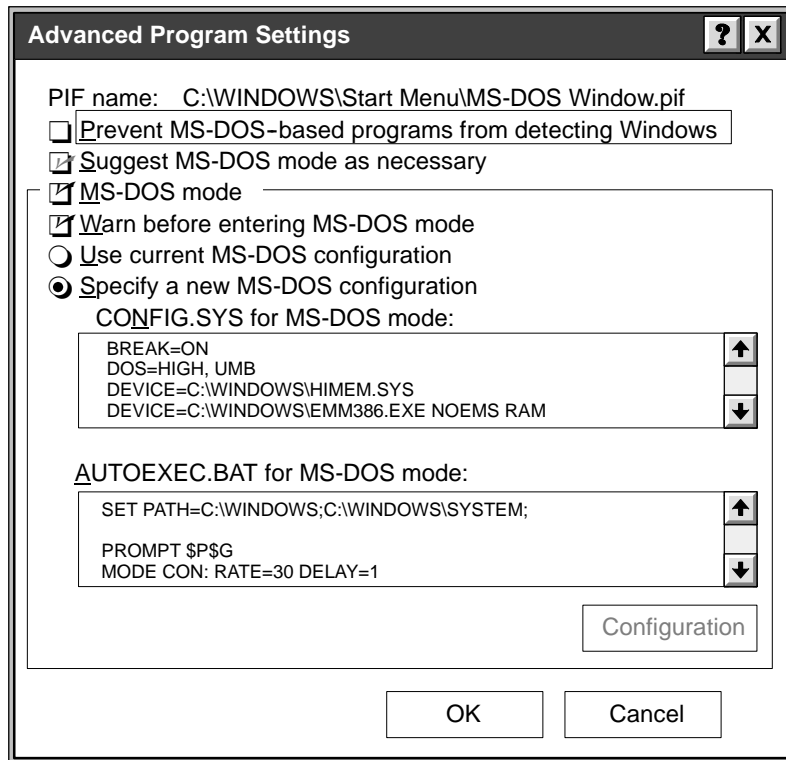


Figure D-13 Advanced Program Settings Dialog Box

3. Make sure the MS-DOS Mode box is checked, and select the **Specify a New MS-DOS Configuration** button. Then select **OK**.

Since you have selected real mode DOS, you do not need to configure the Working, Batch File, Shortcut Key, and Run fields.

4. In the Program folder, set the following property:

Cmd line `C:\WINDOWS\COMMAND.COM`

In the Program folder, the default is for APT to be installed on the C: drive. If this is not the case, change the drive letter in the Cmd line field to the appropriate drive.

You do not need to configure the Font, Memory, Screen, and Misc folders, since they are not used in DOS Mode.

5. Select **OK** at the bottom of the MS-DOS Prompt Properties dialog box to save the changes.

You have now configured an MS-DOS icon for starting a real mode DOS session.

Configuring Memory Management

A real mode DOS session is essentially the same environment provided by older versions of MS-DOS. DOS Mode requires memory management of the 640K conventional memory and upper memory just like the older versions of MS-DOS.

Windows 95 does not provide any type of support for a real mode DOS session. No support is included for the mouse, networking, disk caching, and most other types of system hardware. You must load drivers for these items into **config.sys**, which uses up memory that would otherwise be available to APT. Devices that are available under Windows 95 may have to be omitted in DOS Mode due to memory limitations.

The **config.sys** file must support all programs and devices to be used in this DOS environment. A sample **config.sys** file suitable for installing APT is listed below. See [Figure D-13](#).

```
BREAK=ON
DOS=HIGH,UMB
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE NOEMS RAM
```

You must put the text for the **config.sys** file in the properties sheet for the MS-DOS shortcut. To do this, follow the steps below.

1. Click the right mouse button on the new MS-DOS shortcut icon and select **Properties** from the menu. The MS-DOS Prompt Properties dialog box appears.
2. Select the **Program** tab, and then select the **Advanced** button to access the Advanced Program Settings dialog box.
3. Locate the text box labeled **config.sys** for MS-DOS mode, and type in the sample **config.sys**.

If you are not running DOS programs that require expanded memory (EMS), then the `noems` parameter should be included with **emm386**. Supporting EMS requires a 64K page frame in upper memory. This is 64K that would otherwise be available for using the `devicehigh` or `loadhigh` commands. If you are currently using **emm386** or an equivalent, make sure to include the command line parameters currently in use.

If there are other entries in your **config.sys** that you need for your computer system, be sure to include those also. The virtual DOS environment example in the previous section included **ansi.sys**, which is optional and is not required to run APT. Due to the addition of **smartdrv.exe** and **mouse.exe**, some computer systems may not have room to include **ansi.sys**, so it is omitted from this example.

The **autoexec.bat** file that you need at this point is shown below. The `prompt` command sets the prompt string for all DOS sessions. The `mode` command speeds up the key repeat rate, which is a useful setting for APT. The virtual DOS environment example included **doskey**, which is optional and is not required to run APT. Due to the addition of **smartdrv.exe** and **mouse.exe**, some computer systems may not have room to include **doskey**, so it is omitted from the example. A sample **autoexec.bat** file suitable for installing APT is listed below. See [Figure D-13](#).

```
set path=c:\windows;c:\windows\system;

prompt $p$g
mode con: rate=30 delay=1

set tmp=c:\temp
set temp=c:\temp

lh c:\dos\mouse.exe
lh c:\windows\smartdrv.exe
```

You must also put this text for the **autoexec.bat** file in the properties sheet for the MS-DOS shortcut. Since this **autoexec.bat** is used strictly for MS-DOS mode, it must contain all programs and drivers needed for the DOS session. This includes the **mouse.exe** and **smartdrv.exe** programs. To enter the text for this file, follow the steps below.

1. Click the right mouse button on the new MS-DOS shortcut icon and select **Properties** from the menu. The MS-DOS Prompt Properties dialog box appears.
2. Select the **Program** tab, and then select the **Advanced** button to access the Advanced Program Settings dialog box.
3. Locate the text box labeled **autoexec.bat** for MS-DOS mode, and type in the sample **autoexec.bat** file.

The **dosstart.bat** file is not used when new **autoexec.bat** and **config.sys** files are specified. After you have entered the **autoexec.bat** and **config.sys** information in the properties sheet, select **OK** on each dialog box until the entire properties sheet has been saved.

To enable the new configuration, double click on the MS-DOS icon. Windows 95 shuts down, and the computer reboots using the **autoexec.bat** and **config.sys** files in the properties sheet. When the MS-DOS session is running, check the memory usage by entering the `mem /c` command. The memory should look like the chart in [Figure D-14](#).

```

C:\>mem /c
Modules using memory below 1 MB:

```

Name	Total	Conventional	Upper Memory
SYSTEM	18,704 (18K)	10,544 (10K)	8,160 (8K)
HIMEM	1,168 (1K)	1,168 (1K)	0 (0K)
EMM386	4,320 (4K)	4,320 (4K)	0 (0K)
MOUSE	23,856 (23K)	272 (0K)	23,584 (23K)
SMARTDRV	29,040 (28K)	12,640 (12K)	16,400 (16K)
COMMAND	10,144 (10K)	0 (0K)	10,144 (10K)
IFSHLP	2,864 (3K)	0 (0K)	2,864 (3K)
SETVER	832 (1K)	0 (0K)	832 (1K)
COMMAND	6,976 (7K)	0 (0K)	6,976 (7K)
Free	633,536 (619K)	626,144 (611K)	7,392 (7K)

```

Memory Summary:

```

Type of Memory	Total	Used	Free
Conventional	655,360	29,216	626,144
Upper	76,352	68,960	7,392
Reserved	131,072	131,072	0
Extended (XMS)	24,303,040	2,360,768	21,942,272
Total memory	25,165,824	2,590,016	22,575,808
Total under 1 MB	731,712	98,176	633,536
Largest executable program size		626,128 (611K)	
Largest free upper memory block		4,368 (4K)	
MS-DOS is resident in the high memory area.			

```

C:\>

```

Figure D-14 Memory Usage Report for Preliminary New Configuration

In the “Modules using memory below 1 MB” chart in [Figure D-14](#), the row for free memory should show at least 600K free of conventional memory; the example has 611K. If there is no upper memory, this usually indicates a problem with `himem.sys` or `emm386.exe`.

Installing APT

The DOS session is now prepared for APT installation. Start the APT installation program by following the “CD-ROM on DOS Systems” procedure on [page A-6](#) or the “Diskettes on DOS Systems” procedure on [page A-7](#). Follow the directions until the last screen appears. Press **F10** and select the option that lets the installation program update the **autoexec.bat** and **config.sys** files. Then reboot the computer. Since you are running real mode DOS with a new configuration, the computer modifies the files and reboots back into DOS.

When the installation program is finished, you need to remove the `buffers` and `shell` commands from the **config.sys** file. Windows 95 provides suitable defaults for these parameters.

```
BREAK=ON
DOS=HIGH,UMB
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE NOEMS RAM

shell=c:\command.com /p /e:500      ← remove this line
device=c:\apt\drivers\cgistub.sys /r
buffers=10                          ← remove this line
files=40
```

The `files=40` command allocates forty file handles for DOS programs to use. This sets the maximum number of files that you can open at one time. Forty file handles represent the minimum number required for DOS and APT to operate together.

When you type `exit` to return to Windows 95, any modifications made to the **autoexec.bat** and **config.sys** files are saved in the properties sheet for this DOS session. Therefore, if you change the **autoexec.bat** or **config.sys** files while in DOS mode, you do not have to change the **autoexec.bat** and **config.sys** in the properties sheet.

After you have edited and saved **config.sys**, type **exit** at the DOS command prompt and allow the computer to reboot. When the computer reboots, you can make any final adjustments to the APT environment. At this point, when you start the DOS session and enter the **mem /c** command, the memory should look like the chart in [Figure D-15](#).

```
[WIN]C:\>mem /c
Modules using memory below 1 MB:

```

Name	Total	Conventional	Upper Memory
SYSTEM	18,720 (18K)	10,560 (10K)	8,160 (8K)
HIMEM	1,168 (1K)	1,168 (1K)	0 (0K)
EMM386	4,320 (4K)	4,320 (4K)	0 (0K)
CGISTUB	64 (0K)	64 (0K)	0 (0K)
MOUSE	23,856 (23K)	272 (0K)	23,584 (23K)
SMARTDRV	29,040 (28K)	12,640 (12K)	16,400 (16K)
DRIVERS	78,160 (76K)	78,160 (76K)	0 (0K)
COMMAND	10,144 (10K)	0 (0K)	10,144 (10K)
IFSHLP	2,864 (3K)	0 (0K)	2,864 (3K)
SETVER	832 (1K)	0 (0K)	832 (1K)
COMMAND	6,976 (7K)	0 (0K)	6,976 (7K)
Free	555,312 (542K)	547,920 (535K)	7,392 (7K)

```

Memory Summary:

```

Type of Memory	Total	Used	Free
Conventional	655,360	107,440	547,920
Upper	76,352	68,960	7,392
Reserved	131,072	131,072	0
Extended (XMS)	24,303,040	2,360,768	21,942,272
Total memory	25,165,824	2,668,240	22,497,584
Total under 1 MB	731,712	176,400	555,312
Largest executable program size		547,680 (535K)	
Largest free upper memory block		4,368 (4K)	
MS-DOS is resident in the high memory area.			

```
[WIN]C:\>
```

Figure D-15 Memory Usage Report for New Configuration

Installing APT in Real Mode DOS Environment — New (continued)

Locate the entry for CGISTUB that is loaded from **config.sys** and the entries for DRIVERS, MOUSE, and SMARTDRV that are loaded from **autoexec.bat**. If anything did not load, you can go through a step-by-step bootup. To do this from Windows 95, double click on the MS-DOS icon. When the reboot procedure starts, press the **F8** key until the boot menu appears. Select **Step by Step** bootup; it is usually number 5. As the computer boots up, it prompts you at each line in the **config.sys** and **autoexec.bat** files.

When everything loads successfully, there should still be 535K of free conventional memory. All of APT's environment variables should be present. Type `set` and press **Enter** to display all of the environment variables. At the minimum, the following APT environment variables must be configured:

```
CGIPATH=c:\apt
MOUSE=msmouse
APT_MODE=direct
APT_PORT=1
RESOLUTION=75
```

If any of these variables are not present, check the **autoexec.bat** file.

Using APT with Windows NT/2000

E.1	Installing APT on Windows NT/2000	E-2
	Overview	E-2
	DOS Execution under Windows	E-2
	Program Information File	E-3
	System Variables	E-6
	Communications Servers	E-7
	Communication Conflicts	E-7
	Communication Reliability	E-8
	PIF for Command Line Access	E-8

E.1 Installing APT on Windows NT/2000

Overview

Windows NT and Windows 2000 are true multi-tasking operating systems. Previous versions of Windows operating systems, such as Windows 95 and 98, were essentially built around DOS.

To preserve compatibility with older programs, a new DOS command interface was created within Windows NT and Windows 2000. Although this command line interface looks identical to the running DOS program, there are some differences. This appendix addresses some of these differences and how they affect APT.

The word “Windows” in the rest of this appendix refers to Windows NT and 2000.

DOS Execution under Windows

When you create a DOS window, it is not truly DOS, but an emulation of DOS running under Windows. Previous versions of DOS-based operating systems had the entire resources of the computer at their command. Therefore, a program operating under DOS could be sure that it had exclusive ownership of the computer hardware, such as communications ports, printer ports, and hard drives.

In Windows, the operating system owns all of the computer resources. So, in order to emulate the type of environment DOS programs require, Windows creates a virtual computer for the DOS window. This virtual computer is called the hardware abstraction layer.

Therefore, whenever APT is downloading a program through the serial port, it is not actually controlling the serial port hardware, but instead is controlling a software image of the serial port hardware. Whenever APT redraws a screen, it is sending commands to the video emulators instead of the video card.

Program Information File

The behavior of the DOS window can be customized to meet the needs of individual users. Customization information is stored in the Program Information File (PIF). As part of APT installation, a PIF was placed on your desktop. This PIF is named APT_WIN and is configured for efficient execution of APT in its own DOS window. Although it is beyond the scope of this document to detail all aspects of the PIF, the most important parameters are discussed below.

Click on the APT_WIN icon on your desktop and select **Properties**. Details of the APT_WIN PIF are displayed. Select the **Program** tab (Figure E-1). The fields of the Program folder are described on the following page.

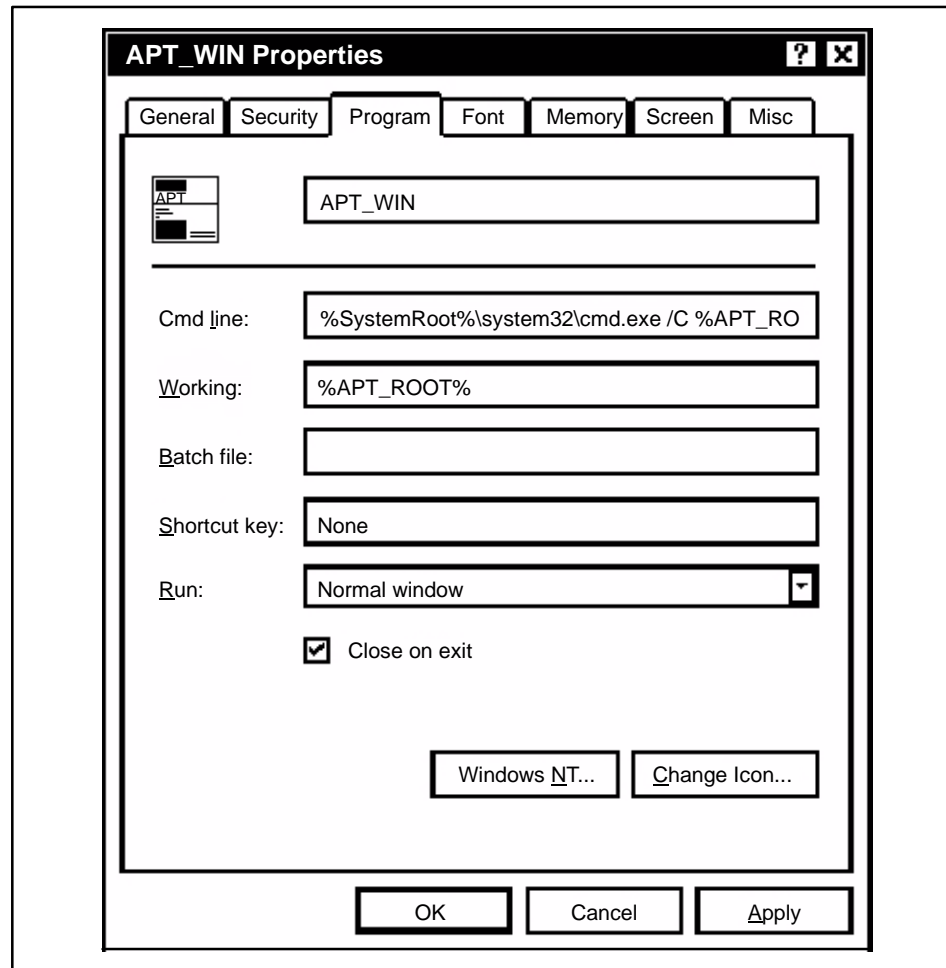


Figure E-1 Program Folder of the PIF

The “Cmd line” field contains the parameters that are required to invoke APT. By default, this field contains the following command:

%SystemRoot%\system32\cmd.exe /C %APT_ROOT%\startapt.bat

%SystemRoot%\system32\cmd.exe loads the DOS command shell.

/C %APT_ROOT%\startapt.bat specifies the batch file used to start APT. The contents of this file is listed below:

- **@echo off** - DOS command that turns off echoing of commands.
- **C:** - selects the drive from which APT is executed.
- **cd C:\apt** - selects the directory from which APT is executed.
- **start aptpcsrv** - starts up the NT/2000 communications server. If you selected H1 as your communications path, this line contains the entry **start APTH1SRV**. If you chose any other communications path, this line contains the entry **start aptpcsrv**.
- **apt** - starts APT.

The “Working” field contains the working directory for APT. This field is set to the **%APT_ROOT%** environment variable. The value of this variable is set by the APT program installation and is typically set to **C:\apt**.

“Close on Exit” is selected as the default. This option closes the DOS window when you exit APT.

At the bottom of the Program folder, click on the **Windows NT...** button. The Windows NT PIF Settings dialog box appears (Figure E-2). Like DOS, each DOS window that is created under Windows must have its own and **Config.sys** files. In this dialog box, you can specify the names of these files and where they are located.

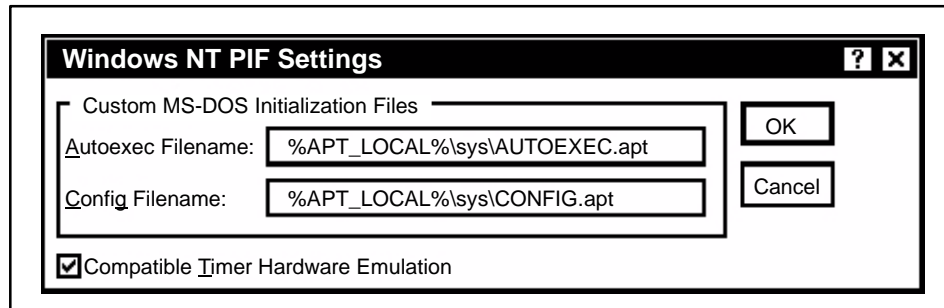


Figure E-2 Windows NT PIF Settings

The autoexec file contains statements for starting up drivers and environment variable definitions for selecting APT functionality such as printer resolution or communications settings. An example of the **Autoexec.apr** that was loaded during APT installation is shown below:

```
PATH=C:\WINNTsystem32;C:\WINNT;C:\;C:\APT
setcgitpath=c:\apt
c:\apt\drivers.exe
set mouse=msmouse
mouse
set apt_mode=direct
set apt_port=1
set resolution=75
set apt_os=NT
prompt$1$p$g
echo on
```

NOTE: If you are having problems with low DOS memory on your system, consider replacing the line `c:\apt\drivers.exe` with the following line:
`Lh c:\apt\drivers.exe`

The config file contains a list of system-level drivers. Such entities as memory managers are specified in this file. An example of the **Config.apt** file that was loaded during APT installation is shown below:

```
device=%SystemRoot%\system32\himem.sys
device=%SystemRoot%\system32\emm.exe NOEMS
dos=high,umb
device=c:\apt\drivers\cgistub.sys /r
buffers=10
files=40
```

In order for APT to function correctly, the “Compatible Timer Hardware Emulation” field at the bottom of the Windows NT PIF Settings dialog box must be selected.

System Variables

Two environment variables control where different components of APT are stored on your hard drive:

- APT_LOCAL
- APT_ROOT

In Windows NT and 2000, these variables are declared in the System Variables area. For details on the function of these variables, see [Appendix A](#).

To examine or modify the values of these variables, follow these steps:

1. Click **Start->Settings->Control Panel**.
2. Click **System**. The System Properties dialog box appears.
3. Click the **Environment** tab.
4. In the System Variables field, you can view or modify the values of the APT_LOCAL and APT_ROOT variables.

Communications Servers

No support is provided for using the CP1613 Industrial Ethernet communication processor from DOS. Therefore, a server has been created to handle communication with the CP1613 from within Windows. This server is named APTH1SRV and is loaded the first time APT is started from the APT_WIN PIF. APT attaches as a client to this server whenever the APT_OS environment variable is set. This server handles all H1, serial, and host communications when running under Windows NT and 2000.

If you do not have a CP1613 in your computer, another server named APTPCSRV is loaded the first time APT is started. This server handles all serial and host adapter communications when running under Windows NT and 2000.

Only one server, APTH1SRV or APTPCSRV, can be running at any time. If you attempt to start another server, it detects the presence of the existing server and terminates. You must either restart your computer or use the Task Manager to shut down a running server.

For more information on configuring and running these servers, see [Appendix B](#).

Communication Conflicts

Since Windows NT and Windows 2000 are multi-tasking operating systems, other programs may conflict with APT for usage of the serial ports. Programs that are likely to interfere with APT communications include Remote Access Software (RAS) and the synchronization software for personal data assistants.

**Communication
Reliability**

APT communications under Windows NT and Windows 2000 are very reliable. Potential conflicts can arise, however, when you download a program to the controller.

As a general rule, whenever you download an APT program, you must have APT in the foreground and should have no other programs executing in the background. When some programs are started, they can freeze execution of other programs. Microsoft Internet Explorer is a good example.

Be aware that if APT is not running full screen, Windows freezes execution of APT, disrupting the download and resulting in a partial program on your controller.

**WARNING**

Windows freezes execution of APT whenever it is not running in Full Screen mode. This will cause a program download to fail, which could result in a partial or incorrect program on your controller.

A partial or incorrect program on your controller could lead to unpredictable controller behavior, which could cause death or serious injury to personnel, and/or damage to equipment.

Ensure that APT runs in Full Screen mode during a program download.

**PIF for Command
Line Access**

In order to execute APT command line functions, you must create a PIF that provides you with a command line prompt, while at the same time setting up all of the parameters required for correct APT operation.

To create a PIF to give you command line access, follow these steps:

1. Copy the APT_WIN PIF on your desktop.
2. Rename the copied PIF to APT_CMD.
3. Right-click on the APT_CMD icon and select **Properties**.

- Click the **Program** tab (Figure E-3), and change the entry in the “Cmd line” field to the following: %SystemRoot%\system32\cmd.exe

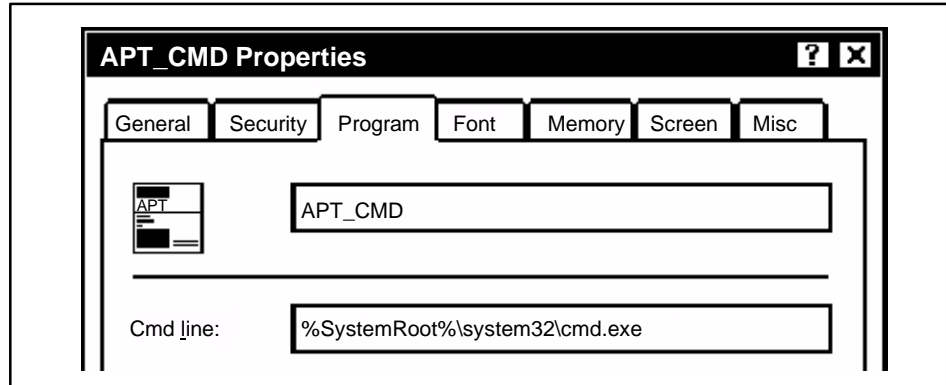


Figure E-3 APT_CMD Properties: Program

- Click the **Screen** tab (Figure E-4), and set the “Initial size” field to 25 lines.
- Click **OK** to save your changes.

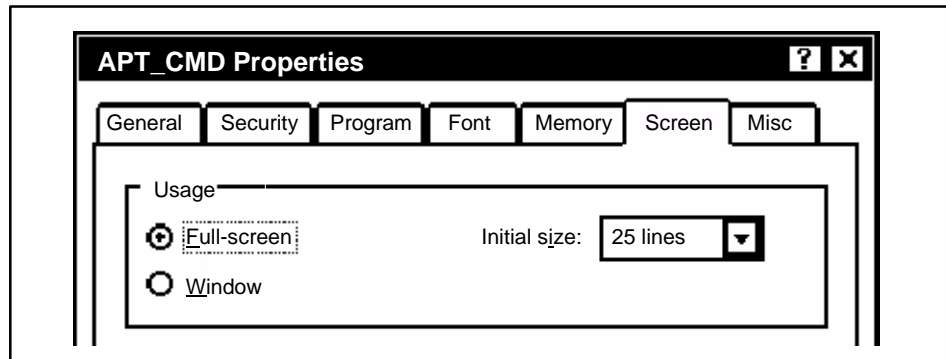


Figure E-4 APT_CMD Properties: Screen

Appendix F

Disk Management

F.1	Disk Management	F-2
	Multiple Drives or Drive Partitions	F-2
	APT Program Organization	F-2
	Using a Single Drive	F-3
	Using Two Drives	F-4
	Using Three Drives	F-5
	Using a Network Drive	F-5
	Using a Virtual (RAM) Drive	F-6

F.1 Disk Management

Multiple Drives or Drive Partitions

If you choose to enhance APT performance by using a virtual (RAM) disk, or if your disk drives or disk drive partitions are small, you can distribute the APT files over two or more drives or drive partitions.

APT Program Organization

Before you can separate the various APT files, it is useful to become familiar with their organization. APT consists of hundreds of files, distributed across dozens of subdirectories, and can be grouped in the broad categories listed in [Table F-1](#).

Table F-1 APT Program Organization

Category	Files	Environment Variable
Executable Files	All <i>.EXE</i> files	Path=X:\apt;
Common Files	Tables, Forms, Libraries	Set APT_COMMON=X:\apt
Database Files	Diagram Tables, Hierarchy Structure, CFC Graphics, Portions of the User Application Program	Set APT_DB=X:\apt\database
Program Files	Math Sections, SFCs, Reports, Compile Information, Portions of the User Application Program	Set APT_LOCAL=X:\apt
Temporary Files	APT files automatically deleted after each session	Set APT_TEMP=X:\apt\temp

You can place these various categories of files on separate computer drives when you install APT and then use the DOS environment variables to point to them. Installation of APT and the usage of environment variables are discussed in [Appendix A](#).

You can keep track of programs and their copies and the locations of the APT files either by installing additional disks in your computer or by partitioning a single disk into one or more extended DOS partitions.

Using a Single Drive

In this example ([Figure F-1](#)), the computer has a single hard disk with a single partition. At least 60 Mbytes of free space should be available for the APT software and for several programs and their expansion as they are edited. Since other programs and utilities may be executed on this computer, the disk needs to be at least 100 Mbytes in size.

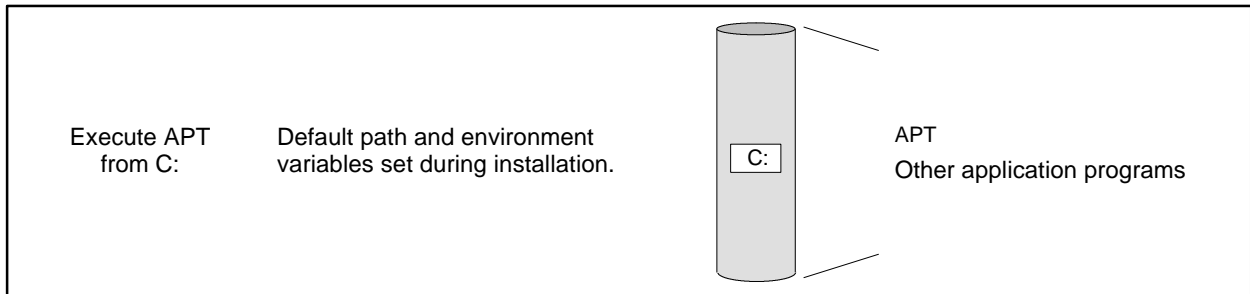


Figure F-1 APT Configured to Run on One Drive

Disk Management (continued)

Using Two Drives

In [Figure F-2](#), the computer has a hard disk with two partitions (or two hard disks). The APT executable and common files are separated from the database and program files, making it easier to reorganize your disk space.

For example, if the disk space on Drive D: becomes fragmented as you develop an application program, you may need to defragment D: periodically. You can defragment D: without disturbing the files on C:, which is faster than if all the files are on one disk. You can also locate and clear lost clusters without the risk of damaging your APT installation.

Other application programs, spreadsheets, database programs, word processing programs, etc., are located on drive C: too. Therefore, C: needs to be relatively larger than the other drive.

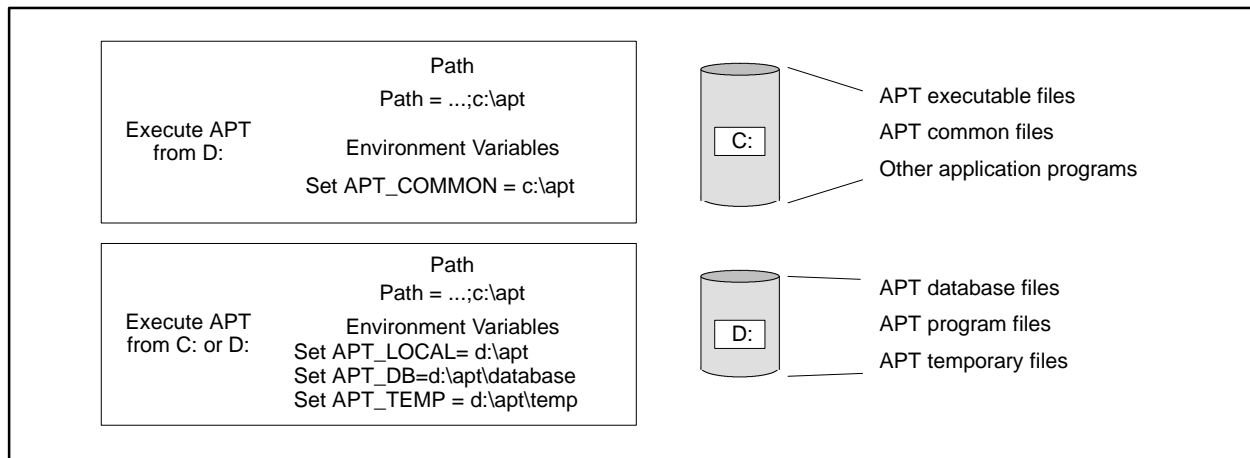


Figure F-2 APT Configured to Run on Two Drives

Using Three Drives

In this example (Figure F-3), the computer has a hard disk with three partitions (or three hard disks, or two hard disks with multiple partitions, etc.). When you install APT, you place the executable files and common files on drive C: and the database files and program files on drives D: and E:. You can edit the program located on drive E: and run the Debug utility on the program on drive D:. Note that the program on drive D: is the one that has been downloaded to the controller.

If you have other application programs, you might want to place these on drive C: too. Therefore, C: needs to be relatively larger than the other two drives.

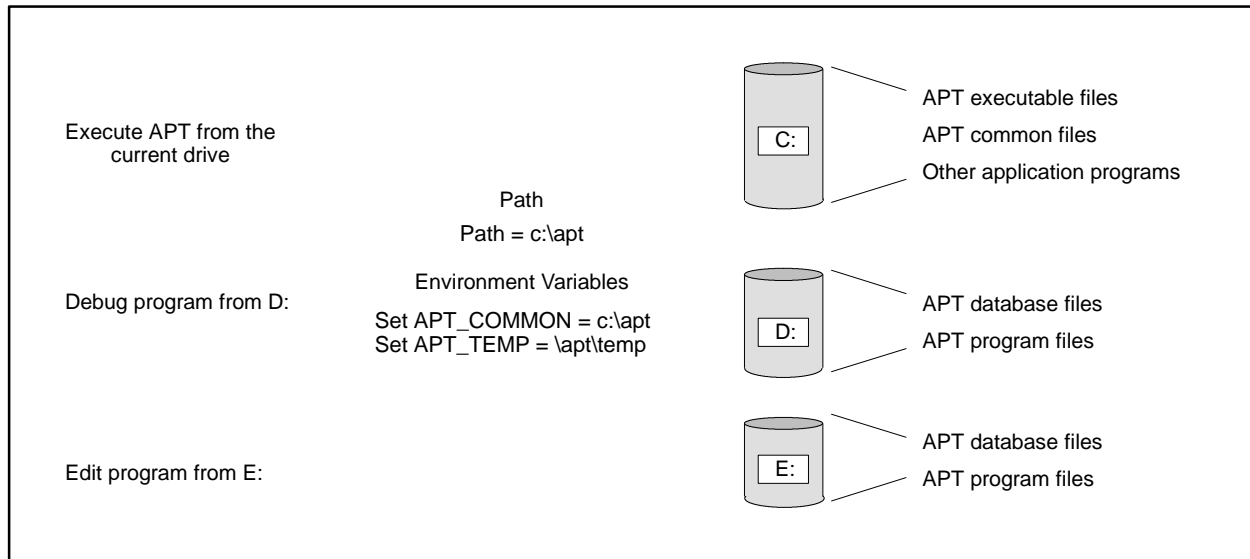


Figure F-3 APT Configured to Run on Three Drives

Using a Network Drive

During the installation of APT, you can enter the path for a network drive. When you have several APT computers at your plant site, this involves only one, instead of multiple, installations of APT. Moreover, if you load the executable and common files on the network, you can save space on the individual APT computers. Note that the temporary and program database files must be unique for each user of the network. Also, you need to use the environment variables to point to the files that you place on the network.

Disk Management (continued)

Using a Virtual (RAM) Drive

In this example (Figure F-4), the computer has a hard disk with two partitions (or two hard disks) and a virtual (RAM) drive. The APT executable and common files are separated from the database and program files for the reasons described on page F-4.

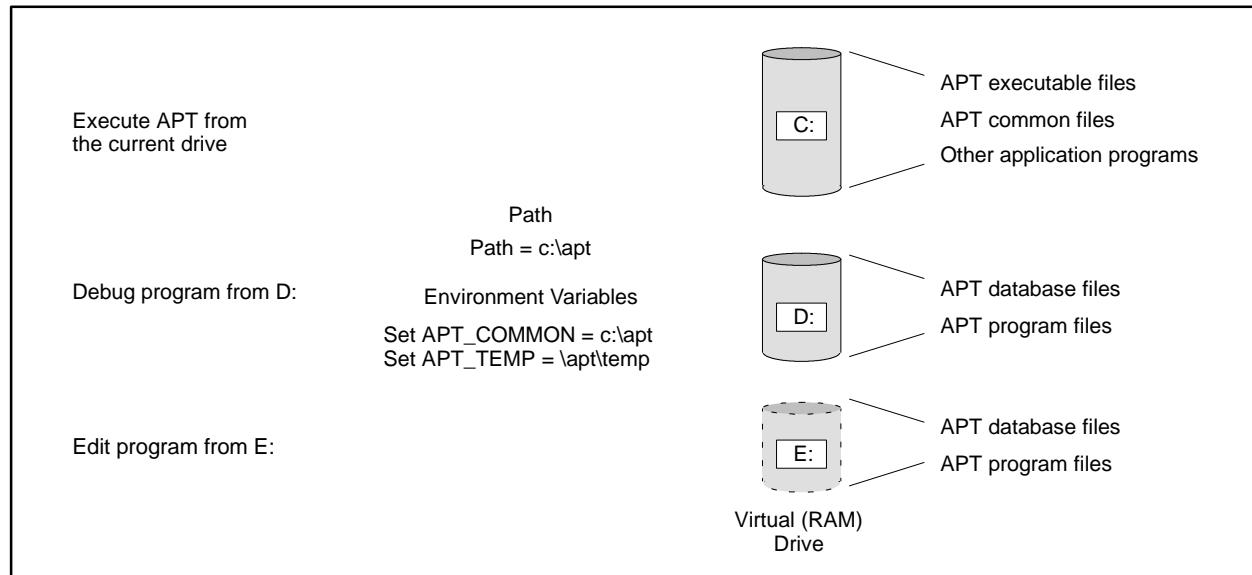


Figure F-4 APT Configured to Run on a Virtual Drive

To get the fastest performance from your computer, you can copy various APT files and subdirectories to the virtual drive before executing APT. The limiting factor for the number of files to copy is the size of the virtual drive. At the least, the virtual drive needs to be large enough (1 Mbyte) to hold the APT temporary files. A virtual drive as large as 25 Mbytes can hold the files in the `\apt\program` and `\apt\database` subdirectories, and the `frontend.exe` file, located in the `\apt` subdirectory.

The MS-DOS 5.0 `dir` command can help you to determine the amount of disk space used by all the files in a subdirectory. You can use this information to calculate how many files to copy to the virtual drive. Enter the following command to check files on Drive E:

```
dir e:\apt\program /s
```

Commercially available programs, such as Norton Utilities, provide similar information. If you choose to use the Norton Utilities option `FILESIZE`, enter the following command to check files on Drive C:

```
FS c:\apt\program /s/t
```

Other factors to consider about a virtual drive include these points.

- You must copy all files to the virtual drive every time you boot the computer; you must save your work back to the hard disk before powering down the computer.
- Unless your system has an Uninterruptible Power Supply, a power outage, a system lockup, or even your application program can cause you to lose information.

CAUTION

If you are using a virtual disk, you must back up your program frequently. Virtual disks are only temporary, and loss of program data can result if power is lost.

Use the APT archive utility to archive your data to the hard disk. Do not use the DOS XCOPY utility or the PKZIP utility to backup your program.

Despite these disadvantages, the use of a virtual drive is worth considering because of the marked improvement in your computer's performance.

Disk Management (continued)

You can use the batch program **aptram.bat**, listed in [Figure F-5](#), to load programs that have been zipped with PKZIP to a virtual drive. The three batch programs **pgmzip.bat**, **dbzip.bat**, and **savesys.bat**, all listed in [Figure F-5](#), save information from the virtual drive to the hard disk. Substitute any name for `pgm_name` and for `db_name` when you run these programs.

Batch Programs That Load Files to the Virtual Disk	
APTRAM.BAT This program loads APT files to the virtual drive. Calls are made to PGMUNZIP.BAT and DBUNZIP.BAT	<pre>REM This program, APTRAM.BAT, loads an APT program to a virtual disk REM (G:). APT is located on hard disk drive C:. REM If database cannot fit on RAM disk, point to it on hard disk REM by removing the "REM" from the next line. REM set apt_db=c:\apt\database set apt_common=c:\apt g: md \apt md \apt\program md \apt\temp md \apt\sys md \apt\database copy c:\apt\sys\ \apt\sys call pgmunzip call dbunzip</pre>
PGMUNZIP.BAT This program is called by APTRAM.BAT .	<pre>g: cd \apt dirdel -r program md program cd program REM pgm_name is the name of the zipped file, unzipped REM by PKUNZIP -- see PGMZIP.BAT pkunzip -d c:\pgm_name</pre>
DBUNZIP.BAT This program is called by APTRAM.BAT .	<pre>cd \apt\database REM db_name is the name of the zipped file -- see DBZIP.BAT pkunzip -o c:\db_name</pre>
Batch Programs That Save Files to the Hard Disk	
PGMZIP.BAT This program saves APT files to the hard disk.	<pre>REM pgm_name.ZIP is the name of the file after it is zipped. del c:\pgm_name.zip pkzip -a -es -p -r c:\pgm_name \apt\program*.*</pre>
DBZIP.BAT This program saves APT database files to the hard disk.	<pre>REM db_name.ZIP is the name of the file after it is zipped. pkzip -a c:\db_name \apt\database\app.*</pre>
SAVESYS.BAT This program saves certain APT parameter files to the hard disk.	<pre>copy \apt\sys c:\apt\sys</pre>

Figure F-5 Programs That Load/Save Files to/from a Virtual Drive

APT Computer Performance

G.1	Planning an Efficient System	G-2
	Defragmenting RAM	G-2
	Defragmenting the Hard Disk	G-2
	Disk-Caching	G-2
	Compressing the Database	G-3
G.2	Maintaining Data Integrity	G-4
	Using the APT Archive Option	G-4
	Using Other Backup Programs	G-4
	Using the DOS CHKDSK or SCANDISK Command	G-5
	Recovering from Power Failure or System Lockup	G-6
G.3	Transporting the Program	G-7
	Archiving the Program	G-7
	Compressing the Program	G-7

G.1 Planning an Efficient System

The information in this section is intended to help you improve the performance of your system. If you are unable to obtain the suggested hardware or software, or have questions about a procedure, call the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at (800) 333-7421. Outside the U.S.A., call 49-911-895-7000.

Defragmenting RAM

You should exit from APT occasionally (once every two or three hours) to allow your workstation RAM memory to reform into contiguous blocks. If you allow the RAM to become fragmented, you may experience problems in executing certain functions, and/or your system could “lock up.”

Defragmenting the Hard Disk

Any program that writes to and also deletes files can cause the files to become fragmented on the hard disk. Consequently, you need to monitor disk fragmentation and defragment the hard disk periodically. You can obtain commercially available disk-optimizing utilities to measure the degree of disk fragmentation, and to rewrite file data in contiguous blocks. If you have DOS 6.0 or higher, you can use **DEFRAG** to defragment a highly fragmented drive and improve hard drive performance.

Disk-Caching

You can improve system performance by using disk-caching. Disk-caching is included in some versions of DOS, and is also commercially available as a separate product. Disk-caching programs determine which portions of your application program are used most frequently and place these portions in RAM for faster access. This also reduces the frequency of the relatively slower hard disk accesses.

NOTE: You can enhance APT performance by using a disk cache of 500 Kbytes or more.

Compressing the Database

Like most commercially available database programs, APT expands its database as information is added to a user program. When you delete information, the file space is marked as being available for reuse. However, APT does not automatically release the file space to the system. This is a task that you must do on a periodic basis.

Whenever you delete a large program, or several small programs, follow the procedure described below to compress the APT database and free up disk space.

1. Archive all APT programs. You can do this from the APT hierarchy, or by using the APTARCH utility (see [Section H.1](#)).
2. Create an empty APT database by typing newdb, then press at the DOS prompt.

CAUTION

The batch file NEWDB.BAT deletes all database information and allows you to start with a clean system.

If you do not have an archive of existing programs at the time you execute NEWDB.BAT, you will lose all program information.

Archive all programs before executing NEWDB.BAT. If there is a possibility that these programs are corrupted, archive them to diskettes or paths that are separate from previous uncorrupted archives.

3. Restore your APT programs. You can do this from the APT hierarchy, or by using the APTREST utility (refer to [Section H.1](#), on DOS utilities).

G.2 Maintaining Data Integrity

The information in this section is intended as a guide for backing up your program information. If you have questions about a procedure, call the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at (800) 333-7421. Outside the U.S.A., call 49-911-895-7000.

NOTE: Do not cycle power or reboot the system while you are using APT. Your program information could become corrupted.

Using the APT Archive Option

It is strongly recommended that you execute the APT Archive option as a routine method of backing up your program information. The greater the volume of work, the more often you will need to archive. See [Appendix H, Section H.1](#), for more information on the APT Archive and Restore utilities.

NOTE: If you archive a program, do not use the DOS `rename (ren)` command to rename the program. The APT Restore command cannot restore a program that has been renamed.

Using Other Backup Programs

The APT Restore option cannot restore a program containing corrupted data. As an alternative to the Archive option, consider one of the options below.

Commercially Available Utility Program Bundled toolkit programs, such as Norton Utilities and PC Tools, provide backup utilities that you can use to back up your program files. To save time, you only need to back up the `\apt\program` and `\apt\database` subdirectories.

Commercially Available File Compression Program Some file compression utilities, such as PKZIP, can back up subdirectories as a part of their file compression. To save time, you only need to back up the `\apt\program` and `\apt\database` subdirectories. See [Section F.1](#) for more discussion about PKZIP as a backup utility.

DOS Backup Command You can use the DOS `backup` command to back up your program. This backup is fastest when the `\apt` subdirectory only contains the database and program. The syntax is `backup \apt x:\ /s`. The `x:` is the destination drive receiving the backed up files.

CAUTION

If you are using a virtual disk, you must back up your program frequently. Virtual disks are only temporary, and loss of program data can result if power is lost.

Use the APT archive utility to archive your data to the hard disk. Do not use the DOS XCOPY utility or the PKZIP utility to back up your program.

Using the DOS CHKDSK or SCANDISK Command

Execute the DOS `chkdsk /f` command on a periodic basis. This command locates lost clusters of information, and the `/F` parameter enables you to have the clusters written to files. Examine these recovered files, and delete them if they are unusable, in order to preserve disk memory.

Execute the DOS `scandisk` command on a periodic basis. This command locates lost clusters of information, and enables you to have the clusters written to files. Examine these recovered files, and delete them if they are unusable, in order to preserve disk memory.

If the `chkdsk /f` or `scandisk` commands reveal lost clusters on your disk, use the APT Restore option to restore all programs.

Maintaining Data Integrity (continued)

Recovering from Power Failure or System Lockup

In the event that a power failure or system lockup occurs while you are executing APT, your program information could be corrupted. To restore your program, follow the steps below.

1. Execute the DOS command `chkdsk /f` or `scandisk` as described on [page G-5](#) and check for lost clusters.
2. If lost clusters exist, or if you want to check the integrity of your program, execute the APT Copy option. Make a temporary copy to be deleted later.

Copy examines your program information and stops execution when an error is discovered. An error message is displayed at the bottom of the screen showing the location of the first error found. You can correct the error and run **Copy** again to locate any other errors in your program.

NOTE: The Copy option provides only limited error checking.

3. Should it appear that you cannot salvage your program, you need to restore the database structure. Create an empty APT database by typing `newdb` and then pressing .

CAUTION

The batch file `NEWDB.BAT` deletes all database information and allows you to start with a clean system.

If you do not have an archive of existing programs at the time you execute `NEWDB.BAT`, you will lose all program information.

Archive all programs before executing `NEWDB.BAT`. If there is a possibility that these programs are corrupted, archive them to diskettes or paths that are separate from previous uncorrupted archives.

4. Restore your APT program. You can do this from the APT hierarchy, or by using the `APTREST` utility (see [Section H.1](#)).

If you have questions about salvaging a corrupted program, call the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at (800) 333-7421. Outside the U.S.A., call 49-911-895-7000.

G.3 Transporting the Program

You may need to transfer your application program to another system. The information in this section is intended to help you prepare a copy of your program in a manageable size for transfer to another system.

Archiving the Program

Execute the APT archive option in order to consolidate your program information into one file. You can reduce the size of your program considerably by archiving only the source file. Select the hard disk as the target so that the program information is all on a single disk.

NOTE: If you archive only the source file, you must re-compile the program after restoring it on another system. Incremental download to the controller is not possible after a full program compile.

Compressing the Program

You can reduce the size of your APT archived program still further by using a commercially available file-compressing utility, such as PKZIP. These utilities are capable of reducing the size of a file by 50% or more. The time to transfer a program by modem, for example, can be reduced significantly. You must de-compress the program before using the APT Restore option to restore your program.

Appendix H

DOS Utilities

H.1	Using the Archive and Restore Utilities	H-2
	Archiving Programs	H-2
	Restoring Programs	H-3
H.2	Using the EXTRACT Utility	H-4
H.3	Using the DDS Utility	H-6
H.4	Using the SHOWLINE Utility	H-8
H.5	Using the DC Utility	H-9
H.6	Using the CHKVIDEO Utility	H-10
H.7	Additional Utilities	H-11
	Downloading a Program from DOS	H-11
	Translating Tags Outside the APT Environment	H-11
	TRANSOFF	H-11
	Recovering RAM	H-11

H.1 Using the Archive and Restore Utilities

Archiving Programs

The APTARCH utility archives a program from the currently defined database to a specified drive and path.

The APTARCH utility is located in the `\apt` subdirectory unless you specified a different subdirectory for executable (`.exe`) files during APT installation.

The syntax for using APTARCH is shown below:

```
APTARCH [ -o -t# -h -s ] <path\program_name>
      or
APTARCH [ -o -t# -h -s ] <program_name> [path]
```

Substitute for *path* the drive and/or directory path to which the program is to be archived. If the drive is not specified, the default is the current drive and subdirectory. Substitute the name of the program to be archived for *program_name*.

- o disables the prompt that queries whether or not the archival should overwrite an existing program.
- t# (0-7) designates the amount of information that is displayed during the archival. A value of t0 displays the least information, and t7 displays the most. The default is t1.
- h displays the command syntax and the parameters.
- s archive all information entered by user. APT-generated information, e.g., object code, is not archived.

For example, to archive the program called PACKAGER to the directory PACK on drive D: with no screen messages, type the following at the DOS prompt:

```
APTARCH -t0 D:\pack\packager
      or
APTARCH -t0 packager D:\pack
```

Error messages that may appear when you use the APTARCH utility are described in the appendix on error messages in the [SIMATIC APT Programming Reference \(Graphics/Math\) Manual](#).

To cancel a current archive operation, press `Ctrl C` or `Ctrl Break`. APT may require several seconds to close files and process other overhead before you can continue your work.

Restoring Programs

The APTREST utility restores an archived program located in a specified drive and path to the currently defined database.

The APTREST utility is located in the **\apt** subdirectory unless you specified a different subdirectory for executable (.exe) files during APT installation.

The syntax for using APTREST is shown below:

```
APTREST [ -i -o -t# -h ] <path\program_name>
      or
APTREST [ -i -o -t# -h ] <program_name> [path]
```

Substitute for *path* the drive and/or directory path from which the program is to be restored. If the drive is not specified, the default is the current drive and subdirectory. Substitute the name of the program to be restored for *program_name*.

- i displays information only about the archive including time stamps, APT archive release, controller information, etc. The program will not be restored if this switch is included.
- o disables the prompt that queries whether or not the restore should overwrite an existing program.
- t# (0-7) designates the amount of information that is displayed during the restore. A value of t0 displays the least information, and t7 displays the most. The default is t1.
- h displays the command syntax and the parameters.

For example, to restore the program called PACKAGER located in the directory PACK on drive D: with no screen messages, type the following at the DOS prompt:

```
APTREST -t0 D:\pack\packager.
```

To restore the program called CONVEYER located in the directory TRANSPRT on drive E: with all screen messages, type the following at the DOS prompt:

```
APTREST -t6 E:\transprt\conveyer
```

Error messages that may appear when you use the APTREST utility are described in the appendix on error messages in the [SIMATIC APT Programming Reference \(Graphics/Math\) Manual](#).

To cancel a current restore operation, press **Ctrl** **C** or **Ctrl** **Break**. APT may require several seconds to close files and process other overhead before you can continue your work.

H.2 Using the EXTRACT Utility

The EXTRACT utility converts a binary APT intermediate library file into a readable ASCII format.

The EXTRACT utility is located in the `\apt` subdirectory unless you specified a different subdirectory for executable (**.exe**) files during APT installation.

When the APT compiler identifies a line that contains a Phase 4, 5, or 6 compile error, APT lists a line number, object name, and file name for the error in the compile report. Because the file is in binary code, you need to use the EXTRACT utility to convert the file to ASCII format before you can examine it. After converting the file, you can use any text editor to examine it for more information about the error. Use the following syntax:

```
EXTRACT [-fhlt -c# -o out_file] <in_file> [obj_names]
```

Substitute the file name listed in the compile report for *in_file*.

Substitute the object name to be extracted for *obj_names*. You can enter more than one name, separating the names with spaces. Note that this name is case sensitive. If you do not designate an object, all objects are extracted.

- f causes the output file to be formatted. If you want the line number listed in the error message to correspond to the line in the converted file, you can use the `-f` option with the `-l` option, described below.
- h displays the command syntax and the parameters.
- l generates line numbers in the output file.
- t indicates that the text input file is not an APT library file. This option is used for factory testing.
- c# selects conversion format, where # stands for
 - 0 = no conversion, or
 - 1 = NNL binary to ASCII conversion, or
 - 2 = Assembly binary to ASCII conversion

This option is used for factory testing.

- o designates the output file to which the conversion is written. Substitute any file name for *out_file* when you use the `-o` parameter. If you do not designate an output file, the conversion output is sent to the screen.

For example, assume that Compile Error #13000 is generated as shown below.

```
13000  Assembly Block
        (YACC) Syntax error near token ' ^eof '
        near line 2 of object WASHER
        in \APT\PROGRAM\ASSEMBLY.565.
        Assembler aborted due to fatal errors.
```

To extract the object **washer** from the binary file **assembly.565** and convert it to the ASCII file **washer.asc**, type the following at the DOS prompt:

```
EXTRACT -fl -o washer.asc \apt\program\p\assembly.565 washer
```

H.3 Using the DDS Utility

The DDS utility copies the APT database to an ASCII file that you can view/edit with an ASCII editor. If you have problems that are related to the database, e.g., difficulty compiling programs, erratic programming behavior, etc., you may be able to resolve these problems by editing the database.

The general procedure for working on the database involves copying the database to an ASCII file, making the appropriate changes, and then rebuilding the database. Before you attempt to modify the APT database, call the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at (800) 333-7421. Outside the U.S.A., call 49-911-895-7000.

The DDS utility is located in the `\apt` subdirectory unless you specified a different subdirectory for executable (`.exe`) files during APT installation.

CAUTION

The DDS utility is primarily used as an external means to modify the APT database.

Whenever you modify the APT database, there is a potential risk of corrupting the database and making it unusable.

If you have not backed up the APT programs, stop now! Run APT and archive all APT programs. Place the archive on a different disk from the one currently holding the database.

-
1. After archiving the programs to another disk, as stated in the Caution, delete any programs that are not causing problems. This reduces the size of the database that you have to work with.
 2. Exit to DOS.
 3. Save the database to an ASCII file called *file_name* by typing
DDS -o <file_name> and pressing .

The default input file for the DDS utility is **app.dds**, located in the APT database subdirectory. If you have changed this file name, you need to use the **-s** parameter:

```
DDS -o <file_name> -s <new_app.dds>
```

Replace *new_app.dds* with the new name that you used for **app.dds**.

4. Use an ASCII editor to edit the file *file_name*.
5. Exit from the ASCII editor.
6. Set the default subdirectory to the APT database subdirectory.
7. Create an empty APT database by typing `pkunzip -o database.zip` and pressing .

NOTE: Do not use the **NEWDB** command to create a new database. This command deletes additional files in the **\apt** directory that you need to preserve.

8. Incorporate your changes into your APT program by typing
DDU -c <file_name> and pressing .

H.4 Using the SHOWLINE Utility

The SHOWLINE utility can display a line in any ASCII file. For example, when the APT compiler identifies a line that contains a compile error, you can use SHOWLINE to examine the line. The syntax is shown below:

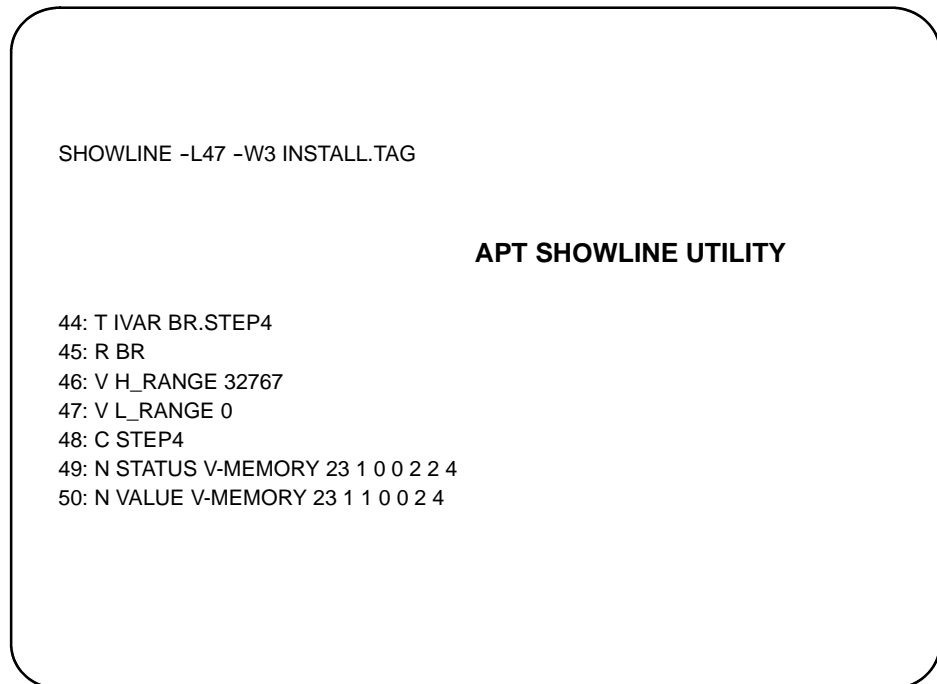
```
SHOWLINE [-h -l <number> -w <number>] <file_name>
```

- h displays SHOWLINE parameters.
- l designates the number of the line that you want to examine.
- w designates the number of lines, before and after the line specified by l, to be displayed.

The SHOWLINE utility is located in the `\apt` subdirectory unless you specified a different subdirectory for executable (`.exe`) files during APT installation.

If the source file is in binary code, use the EXTRACT utility ([Section H.2](#)) to convert the binary file into ASCII code before using SHOWLINE.

[Figure H-1](#) shows the results of executing SHOWLINE on the APT file `install.tag` to display three lines before and after line 47.



```
SHOWLINE -L47 -W3 INSTALL.TAG

                                APT SHOWLINE UTILITY

44: T IVAR BR.STEP4
45: R BR
46: V H_RANGE 32767
47: V L_RANGE 0
48: C STEP4
49: N STATUS V-MEMORY 23 1 0 0 2 2 4
50: N VALUE V-MEMORY 23 1 1 0 0 2 4
```

Figure H-1 The SHOWLINE Utility

H.5 Using the DC Utility

The DC utility offers all the functions that the APT Report option provides for an object, as well as resolving individual symbol addresses. The DC Resolve Symbol option is very useful for determining addresses if only a few locations are needed. Use the DC utility when you want to resolve symbol addresses in the DOS environment, as opposed to using Debug **Direct Address**.

The DC utility is located in the `\apt` subdirectory unless you specified a different subdirectory for executable (**.exe**) files during APT installation.

To run the DC utility (**Resolve Symbol**), follow the steps below.

1. At the DOS prompt for the drive containing the APT program, type DC and press .
2. At the menu prompt, type 5 (Resolve Symbol) and press .
3. Type the program name and press .
4. Type the unit name and press . (For a global variable, just press .)
5. Type the APT symbol name and press . Be sure you include the extension.
The system displays the address assigned to the variable.
6. Enter 0 to quit.

For example, consider the symbol **Tk1_a.opnd**, which has the controller address C88 and is located in unit **Tank_1** of a program called **Blender**. Use the DC utility, following the steps below.

1. At the DOS prompt, type DC and press .
 2. At the menu prompt, type 5 and press .
 3. Type Blender and press .
 4. Type Tank_1 and press .
 5. Type Tk1_a.opnd and press .
- The system displays the following message: **Resolved: C88.**

H.6 Using the CHKVIDEO Utility

The CHKVIDEO utility lists the video card that APT assumes is installed in your system. If you have difficulty executing APT because of a monitor problem, use the CHKVIDEO utility to check your video card.

The CHKVIDEO utility is located in the `\apt` subdirectory unless you specified a different subdirectory for executable (`.exe`) files during APT installation.

To run CHKVIDEO, type CHKVIDEO and press `Enter`. For a VGA system, APT displays **Video type is VGA**.

Be sure that the `cgi.cfg` file contains a line for the appropriate device driver. For example, for an IBM-compatible VGA system, `cgi.cfg` contains

```
DEVICE = C:\IBMVGA12.SYS /G:DISPLAY
```

A VGA system provides a large viewing area, especially in the split screen mode.

H.7 Additional Utilities

The utilities described below are located in the `\apt` subdirectory unless you specified a different subdirectory for executable (`.exe`) files during APT installation.

Downloading a Program from DOS

The `DOWN` (Series 505) and `DOWN_MC5` (S5) utilities allow you to download a program to the controller from a DOS computer without using an access key and running APT. The `DOWN` utility is described on [page 6-54](#) of this manual.

Translating Tags Outside the APT Environment

The `TRANS` utility allows you to translate tags to OSx/TISTAR without using an access key and running APT. The `TRANS` utility is described in [Section 8.10](#) of this manual.

TRANSOFF

The `TRANSOFF` utility allows you to copy the OSx/TISTAR translate files to a diskette or other directory location. The utility will also inform the APT program database of a successful translate. Subsequent compiles only list translate warnings if the addresses have changed for objects marked for translate since the last `TRANSOFF`. The `TRANSOFF` utility is described in [Section 8.9](#) of this manual.

Recovering RAM

The `DRIVERS` utility allows you to remove the APT drivers from memory and to load them when you need to run APT. Use this utility when you need to recover RAM for other applications.

To remove the APT drivers, type `DRIVERS /A` and press `[Enter]`. If you load any terminate-and-stay-resident programs (TSRs) after you first load the APT drivers, DOS may be unable to recover the RAM. In this case, run `DOSMODE`, described on [page A-14](#).

To load the APT drivers, type `DRIVERS` and press `[Enter]`.

Numbers

386/ATM module, A-15

A

Activate, debug option, 7-38, 7-40, 7-41

Address (controller)

accessing directly, 7-13

and APT symbols, H-9

report, creating from object file, 6-57

Animate, debug option, 7-28, 7-34

Append.tag file, 8-5, 8-13

APT

and ethernet, B-16

disk management, F-2–F-8

network drive, F-5

single drive, F-3–F-9

three drives, F-5–F-11

two drives, F-4–F-10

virtual drive, F-6

ethernet connection, B-14

file organization, F-2

installation, A-2

additional software, A-3

changing modes, A-12

configuration parameters, A-8

example programs, A-12

hardware, A-3

memory configuration, C-8, C-10–C-31

memory requirements, C-5

on ethernet, B-30

starting, A-6

startup, A-12

network connections, B-2

assigning ethernet address, B-4, B-10

assigning node names, B-3, B-9

assigning TSAPs, B-6, B-22

determining node associations, B-5, B-12,

B-22

startup, 1-2

APT mode, 1-19

command, A-11, A-14

apt_baud, DOS environment variable, A-17

apt_common, DOS environment variable, A-21

apt_db, DOS environment variable, A-21

apt_H1_intnum, DOS environment variable,
A-17

apt_local, DOS environment variable, A-21

apt_mode, DOS environment variable, A-18

apt_pc_nobios, DOS environment variable, A-17

apt_pcs_node, DOS environment variable, A-19

apt_port, DOS environment variable, A-19

apt_retries, DOS environment variable, A-19

apt_speed, DOS environment variable, A-19

apt_temp, DOS environment variable, A-21

apt_timeout, DOS environment variable, A-19

APTARCH utility, H-2

aptd.sys, A-15

apth1tc, B-33

APTREST utility, H-3

Archive, program, 2-14, G-4, H-2

Arrays, declaration table, 3-19

ASCII format

converting database, H-6

converting files, H-4

viewing files, H-8

Autoexec.bat file, A-4, A-11, A-14, B-30,
C-10–C-31

AUX functions, PROFIBUS-DP, 9-23

AUX functions icon, 1-9

debug, 7-8, 7-10

B

Backup, program, data integrity, G-4

bd (baud), DOS environment variable, A-20

Black, DOS environment variable, A-22

Breakpoint, debug option, 7-35

bs_386, A-15

Build chart, 7-19

C

Cable

- communication
 - S5, A-30
 - Series 505, A-28
- connection
 - S5, A-31
 - Series 505, A-29

CFB

- activating, 7-40
- defined, 5-2

CFC

- adding, 5-2
- block icons
 - connecting, 5-5
 - disconnecting, 5-5
 - linking, 5-4
 - marking, 5-6
 - using, 5-4
- CTLs icon, 5-6
- defined, 5-2
- editing options, 5-7
- exiting, 5-7
- form sub-editor, 5-8
 - OPTs icon, 5-9
- math sub-editor, 5-10
 - OPTs icon, 5-11
- OPTs icon, 5-6
- report, 2-18

cgi.cfg file, A-24, C-9

cgipath, DOS environment variable, A-21

cgistub.sys, A-25

Change chart, debug option, 7-23

Charts (debug)

- adding, 7-18
- changing, 7-23
- changing description, 7-18
- copying, 7-18
- deleting, 7-18
- editing, 7-22
- listing, 7-23
- renaming, 7-18
- sorting, 7-18

CHKVIDEO utility, H-10

Coldstart

- CPU928B, 6-24
- CPU948, 6-28

COML (configuration management local), B-7, B-22, B-26

Communication

- cable
 - S5, A-30
 - Series 505, A-28
- connection
 - S5, A-31
 - Series 505, A-29

Communication ports, A-26

Communications processor (CP1413)

- configuring, B-14
- TF service, B-24
- verifying configuration, B-17

Communications processor (CP143/CP1430), B-28

Communications processor (CP1434TF), B-22

Compiler control file, 6-4, 6-20, 8-12

- editing, 6-15, 6-34

Compiling programs

- appending tags, 8-5, 8-13
- build translate table, 6-3, 6-6, 6-22, 8-12
- compiler control file, 6-4, 6-20, 8-12
- create debug version, 6-3
- editing compiler control file, 6-15, 6-34
- force compile, 6-2
- object file reports, 6-41
- options, 6-2
- phases, 6-2
- report by exception, 6-7
- reserved locations, 6-3
- starting, 6-17, 6-38

Completion aids, 1-17, 3-27

Config.sys file, A-4, A-11, A-14, B-16, B-30, C-10–C-31

Configuration

- APT screen, 1-7
- CP1413, B-14
- DOS memory, C-8
- memory, C-10–C-31

Configuration (continued)
network connection
 assigning ethernet address, B-4, B-10
 assigning node names, B-3, B-9
 assigning TSAPs, B-6, B-22
 determining node associations, B-5, B-12,
 B-22

Continuous function chart. *See* CFC

Controllers
 memory configuration, 6-4, 6-5, 6-21
 mode, changing, 7-14
 supported by APT, 6-4, 6-5, 6-21

controllers, compatible with PCS, 8-4

Conventional DOS memory, C-2, C-5

CP143/CP1430, communications processors
 (S5), B-28

CP1434TF, H1 communications processor, B-22

Cross reference, 3-28

CRs, overflow, 6-47

CTLs icon, 1-9
 CFC, 5-6
 debug, 7-6
 defining I/O points, 3-9
 defining modules, 3-7
 defining table objects, 3-18
 program content, 2-5
 program directory, 2-3
 SFC, 4-7
 subroutine subeditor, 3-23
 symbolic name table, 3-11
 table objects, 3-15
 unit content, 2-7

Cursor, text, 1-10

CVU/TIWAY adapter, A-15

cvuta.dev, A-15

D

Database, conversion to ASCII, H-6

DC utility, H-9

DDS utility, H-6

Debug
 accessing direct controller address, 7-13
 activating CFBs, 7-40

Debug (continued)
 activating devices, 7-41
 activating SFC steps, 7-38
 adding charts, 7-18
 animating SFCs, 7-34
 animating variables, 7-28
 AUX functions icon, 7-8, 7-10
 build chart option, 7-19
 changing chart description, 7-18
 changing charts, 7-23
 copying charts, 7-18
 CTLs icon, 7-6
 deleting charts, 7-18
 direct controller addresses, 7-29
 editing charts, 7-22
 exiting, 7-17
 forcing/unforcing variables, 7-30
 listing charts, 7-23
 modifying variables, 7-30
 monitoring trends, 7-26
 monitoring variables, 7-24, 7-28
 non-APT program, 7-12
 OPTs icon, 7-6
 PROFIBUS-DP, 9-23
 PROG/RUN option, 7-14, 7-15
 program, 6-3, 7-2
 renaming charts, 7-18
 sort entries option, 7-17
 sorting charts, 7-18
 split-screen option, 7-17
 starting, 7-4
 tracing program flow
 animating SFCs, 7-34
 breakpoint option, 7-35
 single step option, 7-36
 step logger option, 7-37

Declaration table
 editor, 3-14
 subeditor, 3-19

Device
 activating, 7-41
 table editor, 3-14
 table report, 2-17

Direct address, debug option, 7-29

Direct controller addresses, 7-29
 and APT symbols, H-9
 reserving, 6-3

Direct memory addressing, from debug, 7-13

-
- Disk management, F-2–F-8
 - compressing database, G-3
 - defragmenting, G-2
 - lost clusters, G-5
 - network drive, F-5
 - single drive, F-3–F-9
 - three drives, F-5–F-11
 - two drives, F-4–F-10
 - virtual drive, F-6
 - Disk-caching, G-2
 - DOS
 - exiting to, 1-19
 - memory, A-4
 - DOS environment variables
 - apt_baud, A-17
 - apt_common, A-21
 - apt_db, A-21
 - apt_H1_intnum, A-17
 - apt_local, A-21
 - apt_mode, A-18
 - apt_pc_nobios, A-17
 - apt_pcs_node, A-19
 - apt_port, A-19
 - apt_retries, A-19
 - apt_speed, A-19
 - apt_temp, A-21
 - apt_timeout, A-19
 - bd (baud), A-20
 - black, A-22
 - cgipath, A-21
 - grey, A-22
 - lines_per_page, A-23
 - mouse, A-22
 - mousesys, A-22
 - no_tbp (transparent binary protocol), A-20
 - nomouse, A-22
 - pa (parity), A-20
 - paper, A-23
 - path, A-21
 - po (port), A-20
 - print_screen, A-23
 - resolution, A-23
 - tm_log, A-21
 - white, A-22
 - DOS memory
 - APT installation, C-5, C-8, C-10–C-31
 - Windows 95, D-15, D-29, D-39
 - DOS memory (continued)
 - configuration, C-8, C-10–C-31
 - Windows 95, D-15, D-29, D-39
 - conventional, C-5
 - defined, C-2
 - defragmenting hard disk, G-2
 - defragmenting RAM, G-2
 - expanded, C-6
 - extended, C-6
 - hardware effects, C-7
 - high, C-5
 - lost clusters, G-5
 - upper, C-5
 - DOS mode, 1-19
 - command, A-12, A-14
 - DOS utilities
 - archiving programs, H-2
 - check video card, H-10
 - controller addresses and APT symbols, H-9
 - converting database to ASCII, H-6
 - converting files to ASCII format, H-4
 - displaying files in ASCII, H-8
 - removing drivers from RAM, H-11
 - restoring programs, H-3
 - DOWN utility, 6-54
 - Downloading program, 6-44, 6-50
 - from APT, 6-50
 - incremental, 6-45
 - Drivers, C-9, H-11
 - See also* name of individual driver
 - memory usage, B-33
 - removing from RAM, H-11
- ## E
- Edit chart, debug option, 7-22
 - Environment variables, DOS. *See* DOS environment variables
 - epsonx.sys, A-24
 - Ethernet
 - address, B-4, B-10
 - and APT, B-16
 - APT connection, B-14
 - APT installation, B-30

Exit, from APT

CFC, 5-7

debug, 7-17

SFC, 4-11

Exit, from APT, to DOS, 1-19

Expanded DOS memory, C-2, C-6

Extended DOS memory, C-2, C-6

EXTRACT utility, H-4

F

Files

APT organization, F-2

conversion to ASCII, H-4

displaying in ASCII, H-8

Force, debug option, 7-30

Force compile, 6-2

Forcing variables, 7-30, 7-32

Function keys, 1-10

G

Graphics card

check, H-10

driver, A-24

Grey, DOS environment variable, A-22

gsscgi.sys, A-24

H

H1 communications processor, B-22

H1 configurator, B-8, B-22, B-24

H1 network. *See* Ethernet

Hardware, configuration

S5, A-31

Series 505, A-29

Help, online, 1-18

Help icon, 1-9

Hierarchy

accessing levels, 2-8

adding a program, 2-12

adding a unit, 2-12

OPTs icon, 2-10

program content, 1-14, 2-4

program directory, 1-14, 2-2

unit content, 1-14, 2-6

High DOS memory, C-2, C-5

hltc.exe, B-33

hpdjet.sys, A-24

hppj180.sys, A-24, C-18

hppj90.sys, A-24

I

I/O

point definition

CTLs icon, 3-9

OPTS icon, 3-9

S5, 3-10, 3-12

Series 505, 3-8

symbolic name table

CTLs icon, 3-11

defining points, 3-10

editor, 3-14

OPTs icon, 3-11

ibmega.sys, A-24

ibmvga12.sys, A-24

Icons

APT screen, 1-9

block, CFC, 5-4

drawing in SFC, 4-4

help, 1-18

hierarchy options, 2-10

modules, 3-6

paging, 1-12

program content, 2-5

program directory, 2-3

unit content, 2-7

Incremental download, 6-45

warnings, 6-49

Industrial Ethernet. *See* Ethernet

Install.tag file, 8-5

Installation, APT

- additional software, A-3
- and ethernet, B-30
- changing modes, A-12
- configuration, A-8
- example programs, A-12
- guidelines, A-2
- hardware, A-3
- memory configuration, C-8, C-10–C-31
- memory requirements, C-5
- starting, A-6
- startup, A-12

Installation, APT, Windows 95

- real mode DOS environment, D-26, D-36
- virtual DOS environment, D-10

K

Keys

- function, 1-10
- paging, 1-12

L

laserjet.sys, A-24

Lines_per_page, DOS environment variable, A-23

M

Math, CFC sub[]editor, 5-10

Memory

- APT drivers, B-33
- controller configuration, 6-4, 6-5, 6-21
- DOS, A-4
 - See also* DOS memory manager, B-16, C-4

Menus, 1-15

Mode, controller, changing, 7-14, 7-15

Module table (Series 505 only), editor, 3-4

- I/O points, 3-8

Monitor, debug option, 7-18, 7-24

Mouse

- DOS environment variable, A-22
- driver, A-4
- using, 1-11

Mousesys

- DOS environment variable, A-22
- mouse driver configuration, A-22

mousesys.sys, A-24, C-18

MSmouse, mouse driver configuration, A-22

msmouse.sys, A-24

Multi-file command, 2-13

N

Network, configuration

- assigning ethernet address, B-4, B-10
- assigning node names, B-3, B-9
- assigning TSAPs, B-6, B-22
- determining node associations, B-5, B-12, B-22

Network drive, F-5

no_tbp (transparent binary protocol), DOS environment variable, A-20

Node

- assigning ethernet address, B-4, B-10
- assigning names, B-3, B-9
- determining associations, B-5, B-12, B-22

Nomouse, DOS environment variable, A-22

O

Object code, report, 2-20

Object file, compile report, 6-41

Objects

- adding, 2-12
- copying, 2-15
- deleting, 2-15

Online help, accessing, 1-18

OPTs icon, 1-9

- CFC, 5-6
- CFC form sub-editor, 5-9
- CFC math sub[]editor, 5-11
- debug, 7-6

OPTs icon (continued)
defining I/O points, 3-9
defining modules, 3-7
defining table objects, 3-18
program content, 2-5
program directory, 2-3
SFC, 4-7, 4-10
subroutine subeditor, 3-23
symbolic name table, 3-11
table objects, 3-15
unit content, 2-7

Overflow CRs, 6-47

P

pa (parity), DOS environment variable, A-20

Paging icon, 1-9

Paging keys, 1-12

Paper, DOS environment variable, A-23

Parameters, table editor, 3-26

Path, DOS environment variable, A-21

PCS

- compatible controllers, 8-4
- marking APT objects, 8-6
- marking process groups, 8-10
- translating tags, 8-2, 8-16
 - choosing release, 8-4

pgdownl, C-18

po (port), DOS environment variable, A-20

Port pinouts, A-26

Position icon, 1-9

Power failure

- programs, G-6
- restart modes
 - S5 CPU928B, 6-24
 - S5 CPU948, 6-28

Print_screen, DOS environment variable, A-23

Process group, marking for PCS/TISTAR, 8-10

PROFIBUS-DP

- AUX functions, 9-23
- configuration report, 9-20
- configuring, 9-4, 9-14
- controller support, 9-2

PROFIBUS-DP (continued)

- debugging, 9-23
- example, 9-4
- exporting binary file to APT, 9-12, 9-16
- I/O modules, 9-17
- master, 9-5, 9-14
- setting operational mode, 9-26
- setting slave mode, 9-23
- setting synchronization mode, 9-28
- slave, 9-7, 9-14
- viewing I/O, 9-30

Prog/Run, debug option, 7-14

Program

- adding, 2-12
- archiving, 2-14, G-4, H-2
- backing up, G-4
- compiler control file
 - 545, 6-8
 - 545L, 6-9
 - 555, 6-10
 - 560, 6-12
 - 560T, 6-12
 - 565, 6-13
 - 565P, 6-14
 - 565T, 6-13
 - 575, 6-11
 - CPU928B, 6-32
 - CPU948, 6-33
- compiling
 - build translate table, 6-3, 6-6, 6-22
 - compiler control file, 6-4, 6-20
 - create debug version, 6-3
 - editing compiler control file, 6-15, 6-34
 - force compile, 6-2
 - object file reports, 6-41
 - options, 6-2
 - phases, 6-2
 - report, 6-19, 6-39
 - report by exception, 6-7
 - reserved locations, 6-3
 - starting, 6-17, 6-38
- content, 1-14, 2-4
- debug, 7-2
 - activating CFBs, 7-40
 - activating devices, 7-41
 - activating SFC steps, 7-38
 - starting, 7-4
- directory, 1-14, 2-2
- downloading to controller, 6-44, 6-50
 - incremental, 6-45

Program (continued)
recovering from power loss, G-6
restoring, 2-14, G-6, H-3
tracing flow
 animating SFCs, 7-34
 breakpoint option, 7-35
 single step option, 7-36
 step logger option, 7-37
transfer, G-7
validating, 2-15
verifying download, 6-57

R

RAM
 defragmenting, G-2
 removing drivers, H-11

RAM drive, F-6

RBE. *See* Report by exception

Real mode DOS session, Windows 95, D-2
 new configuration, D-8
 standard configuration, D-6

Recipe, template editor, 3-24

Recipe table
 editor, 3-14
 subeditor, 3-20

Report by exception (RBE), 6-7, 8-3

Reports
 APT
 address, from object file, 6-57
 APT symbol name to P/C address, 2-20
 creating, 2-16
 cross reference, 2-20
 object code, 2-20
 P/C address to APT symbol name, 2-20
 SFC cross reference, 2-20
 SFC math and CFB to P/C address, 2-20
 SFC steps and transitions, 2-20
 SFC/CFC, 2-18
 STEP 5 symbol files, 2-20
 tables, 2-17
 TISOFT comment files, 2-20
 TISOFT synonym files, 2-20
 translated tags, 2-20

Reports (continued)
 compile
 APT symbol name to P/C address, 6-41
 cross reference, 6-41
 P/C address to APT symbol name, 6-41
 SFC cross reference, 6-41
 SFC math and CFB to P/C address, 6-41
 SFC steps and transitions, 6-41
 STEP 5, 6-41
 TISOFT comment files, 6-41
 TISOFT synonym files, 6-41
 translated tags, 6-41
 controller address to APT symbol name, H-9

Reserve characters, PCS, 8-7

Resolution, DOS environment variable, A-23

Restart mode (S5)
 CPU928B
 coldstart, 6-24
 warmstart, 6-24
 CPU948
 coldstart, 6-28
 warmstart, 6-28

Restore, program, 2-14, G-6, H-3

RS-232 connections, A-26

S

S5
 APT direct connection, B-2
 APT network connection, B-2
 assigning ethernet address, B-10
 assigning node names, B-9

Sample, debug option, 7-24

Screen
 areas, 1-8
 colors, 1-7
 icons, 1-9
 paging, 1-12
 splitting, 1-13

Sequential function chart. *See* SFC

Series 505, APT network connection, B-2
 assigning ethernet address, B-4
 assigning node names, B-3
 assigning TSAPs, B-6, B-22
 determining node associations, B-5, B-12, B-22

SFC

- accessing, 4-3
- adding, 4-3
- animating, 7-34
- copying block to buffer, 4-9
- CTLs icon, 4-7
- defined, 4-2
- drawing icons, 4-4
- exiting, 4-11
- finding objects, 4-8
- inserting graphics, 4-8
- marking blocks, 4-9
- OPTs icon, 4-10
- report, 2-18
- resequencing labels, 4-9
- selection branches, 4-2
- sequencing labels, 4-9
- step/transition labels, 4-3
- sub□editor, 4-12
 - entering text, 4-12
 - marking text, 4-13
 - OPTs icon, 4-14

SHOWLINE utility, H-8

Single step, debug option, 7-36

Sort entries, 7-17

Split screen, 1-13, 7-17

Startup, APT, 1-2, A-12

STEP 5, and APT, 8-30

Step logger, debug option, 7-37

Subroutine

- subeditor
 - CTLs icon, 3-23
 - OPTs icon, 3-23
- table editor, 3-22

T

Table editors

- completion aids, 3-27
- copying objects, 3-30
- cross reference, 3-28
- declaration table, 3-14
 - subeditor, 3-19
- device table, 3-14
- I/O module, 3-3, 3-4

Table editors (continued)

- I/O symbolic name table, 3-14
- moving objects, 3-31
- parameters, 3-26
- recipe table
 - editor, 3-14
 - subeditor, 3-20
- recipe template, 3-24
- subroutine table, 3-22

Table object definition

- CTLs icon, 3-18
- OPTs icon, 3-18

Table object list

- CTLs icon, 3-15
- OPTs icon, 3-15

Tag translate

- attributes subeditor, 8-9
- editor, 8-6
- subeditor, 8-8

Tags

- appending, 8-5, 8-13
- marking objects for PCS/TISTAR, 8-6
- translating
 - append.tag file, 8-5
 - choosing PCS/TISTAR release, 8-4
 - install.tag file, 8-5
 - marking objects for PCS/TISTAR, 8-6
 - procedures, 8-16
 - S5, B-2
 - Series 505, B-2
 - to PCS/TISTAR, 8-2

TF service, B-24

ti850.sys, A-25

TISOFT, and APT, 8-28

TISTAR

- marking APT objects, 8-6
- marking process groups, 8-10
- translating tags, 8-2, 8-16
 - choosing release, 8-4

tivdi.sys, C-18

tm_log, DOS environment variable, A-21

TRANS utility, 8-22

Translate table, 6-3, 6-6, 6-22, 8-12

Translate tag, procedures, 8-16

Translating tags
 append.tag file, 8-5
 choosing PCS/TISTAR release, 8-4
 install.tag file, 8-5
 marking APT objects, 8-6
 S5, B-2
 Series 505, B-2
 to PCS/TISTAR, 8-2

Trend window, debug option, 7-26

Trends, monitoring, 7-26

TSAP, assigning, B-6, B-22

U

Unit
 adding, 2-12
 content, 1-14, 2-6

Upper DOS memory, C-2, C-5

V

Validation, program, 2-15

Variables
 animating, 7-28
 forcing, 7-32
 forcing/unforcing, 7-30
 modifying, 7-30
 monitoring, 7-24, 7-28

Verifying
 CP1413 configuration, B-17
 program download, 6-57

Video card, check, H-10

Virtual DOS session, Windows 95, D-2, D-4

Virtual drive, F-6

W

Warmstart
 CPU928B, 6-24
 CPU948, 6-28

White, DOS environment variable, A-22

Windows 95
 installing APT
 real mode DOS environment, D-26, D-36
 virtual DOS environment, D-10
 real mode DOS session, D-2
 new configuration, D-8
 standard configuration, D-6
 running APT, D-2
 virtual DOS session, D-2, D-4

Y

Ys, used as CRs, 6-47

Customer Response

We would like to know what you think about our user manuals so that we can serve you better. How would you rate the quality of our manuals?

	Excellent	Good	Fair	Poor
Accuracy	_____	_____	_____	_____
Organization	_____	_____	_____	_____
Clarity	_____	_____	_____	_____
Completeness	_____	_____	_____	_____
Graphics	_____	_____	_____	_____
Examples	_____	_____	_____	_____
Overall design	_____	_____	_____	_____
Size	_____	_____	_____	_____
Index	_____	_____	_____	_____

Would you be interested in giving us more detailed comments about our manuals?

Yes! Please send me a questionnaire.

No. Thanks anyway.

Your Name: _____

Title: _____

Telephone Number: (____) _____

Company Name: _____

Company Address: _____

Manual Name: SIMATIC APT User Manual

Text Number: 2601246-0010

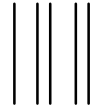
Order Number: PPX:APT-8101-10

Edition: Tenth

Date: 4/01

FOLD

SIEMENS ENERGY & AUTOMATION, INC.
3000 BILL GARLAND ROAD
P O BOX 1255
JOHNSON CITY TN 37605-1255



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

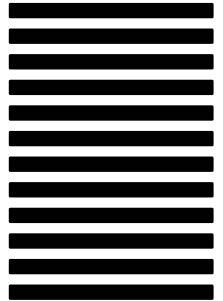
FIRST CLASS

PERMIT NO.3

JOHNSON CITY, TN

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN TECHNICAL COMMUNICATIONS M/S 1255
SIEMENS ENERGY & AUTOMATION INC
P O BOX 1255
JOHNSON CITY TN 37605-1255



FOLD