

SIMATIC

S7 SIMATIC Automation Tool user guide

Manual

Preface

SIMATIC Automation Tool overview	1
Prerequisites and communication setup	2
Tool operations	3
Saving your device table information	4
Menu, toolbar, and shortcut key reference	5
SIMATIC Automation Tool API for .NET framework	6
SIMATIC Automation Tool device support	7

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Preface

Service and support

In addition to our documentation, Siemens offers technical expertise on the Internet and on the customer support web site (<https://www.siemens.com/automation/>).

Contact your Siemens distributor or sales office for assistance in answering any technical questions, for training, or for ordering S7 products. Because your sales representatives are technically trained and have the most specific knowledge about your operations, process and industry, as well as about the individual Siemens products that you are using, they can provide the fastest and most efficient answers to any problems you might encounter.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. You can find more information about industrial security on the Internet (<https://support.industry.siemens.com/cs/us/en/>).

To stay informed about product updates as they occur, sign up for a product-specific newsletter. You can find more information on the Internet (<http://www.industry.siemens.com/topics/global/en/industrial-security/Pages/default.aspx>).

Table of contents

	Preface	3
1	SIMATIC Automation Tool overview	9
1.1	Managing networks.....	9
1.2	Network access.....	12
1.3	Network options	13
1.4	.NET API (application interface) .dll file	15
2	Prerequisites and communication setup	17
2.1	PG/PC Operating system requirement	17
2.2	Installing the SIMATIC Automation Tool.....	17
2.3	Starting the SIMATIC Automation Tool.....	17
2.4	CPU configuration requirements.....	18
2.5	Communication setup	21
3	Tool operations	25
3.1	Scan a network and use the device table	25
3.2	Download new IP, subnet, and gateway addresses.....	31
3.3	Download new PROFINET device names.....	32
3.4	Set CPUs to RUN or STOP mode	33
3.5	Flash the LEDs on devices	35
3.6	Download new programs to CPUs.....	36
3.7	Install new firmware in CPUs and modules	39
3.8	Backup and Restore CPU data.....	44
3.9	Reset CPUs and modules to factory default values	46
3.10	Reset CPU memory	47
3.11	Upload service data from CPUs	48
3.12	Set time in CPUs.....	50
3.13	Read diagnostic buffer in a CPU	51
3.14	Execution order of operations	52
4	Saving your device table information	53
4.1	Save/Save as - Device table stored in encrypted .sat format.....	53
4.2	Import/Export - Device table loaded from/stored in open .csv format	54

5	Menu, toolbar, and shortcut key reference.....	57
5.1	Main menu	57
5.1.1	File menu	57
5.1.2	Edit menu	58
5.1.3	Network menu	59
5.1.4	Tools menu	60
5.1.5	Help menu.....	62
5.2	Toolbar icons.....	63
5.3	Shortcut keys	64
6	SIMATIC Automation Tool API for .NET framework.....	65
6.1	API classes and methods overview	65
6.2	NetworkAPI methods	67
6.2.1	NetworkAPI constructor	67
6.2.2	ScanNetworkInterfaceCards	67
6.2.3	SetNetworkInterface	68
6.2.4	ScanNetworkDevices	68
6.2.5	FlashLED	69
6.2.6	SetIP	69
6.2.7	SetProfinetName.....	70
6.2.8	GetDeviceType	71
6.2.9	ResetToFactory (NetworkAPI)	72
6.3	DeviceAPI methods	73
6.3.1	DeviceAPI constructor	73
6.3.2	StartCommunications and StopCommunications	74
6.3.3	Connect.....	75
6.3.4	Disconnect	75
6.3.5	SetMode.....	76
6.3.6	GetMode	77
6.3.7	ResetToFactory (DeviceAPI)	78
6.3.8	MemoryReset.....	79
6.3.9	DownloadFirmware	80
6.3.10	DownloadProgram	85
6.3.11	UploadServiceData	86
6.3.12	GetStationName	86
6.3.13	GetPLCName.....	87
6.3.14	SetTime	88
6.3.15	GetTime	88
6.3.16	GetDeviceInformation	89
6.3.17	GetRackInformation	90
6.3.18	Backup	91
6.3.19	Restore.....	92
6.3.20	GetDiagnosticsBuffer	93

7	SIMATIC Automation Tool device support.....	95
7.1	S7-1200 CPU support.....	95
7.2	S7-1500 CPU support.....	96
7.3	ET 200SP CPU support (Distributed I/O controller based on ET 200SP and S7-1500)	97
7.4	ET 200SP support (PROFINET I/O)	97
7.5	ET 200MP support (PROFINET I/O)	98
7.6	ET 200AL support (PROFINET I/O)	98
7.7	ET 200S support (PROFINET I/O)	98
7.8	ET 200M support (PROFINET I/O).....	99
7.9	ET 200eco support (PROFINET I/O)	99
7.10	ET 200pro support (PROFINET I/O).....	100
	Index.....	101

SIMATIC Automation Tool overview

1.1 Managing networks

Managing networks of SIMATIC devices

After a control program is created and verified with the Siemens TIA Portal software, the SIMATIC Automation Tool can be used in the field for configuring, operating, maintaining, and documenting automation networks.

If you are managing a network with many devices, the SIMATIC Automation Tool can simplify operations and save time by automatically processing a group of devices with multiple processing threads. When a PG/PC processing thread is connected to a device and waiting for a reply message, other threads can use this time to communicate with the other devices in the group.

Automation Tool operations

- Scan the network and create a table that maps the accessible devices on the network. Unconfigured and configured CPUs and modules are included in the table that is stored in a secure *.sat project file.
- Flash LEDs on a device to help physically locate the device
- Download addresses (IP, subnet, gateway) to a device
- Download PROFINET name (station name) to a device
- Set the time in a CPU to the current time in your PG/PC (Programmer/Personal Computer)
- Download a new program to a CPU
- Download a firmware update to a CPU or module
- Put a CPU in RUN or STOP mode
- Perform a CPU Memory reset
- Read the diagnostic buffer of a CPU
- Upload service data from a CPU
- Backup/Restore CPU data to/from a backup file
- Reset devices to factory default values

SIMATIC device support

The SIMATIC Automation Tool supports the following devices. Standard and Fail-Safe CPUs and I/O modules are supported.

For some devices, some of the SIMATIC Automation Tool operations are not supported.

Click on a device for support details.

S7-1200 CPU (Page 95)

S7-1500 CPU (Page 96)

ET 200SP CPU (Distributed controller based on ET 200SP and S7-1500) (Page 97)

ET 200SP (Distributed I/O controller) (Page 97)

ET 200MP (Distributed I/O controller) (Page 98)

ET 200AL (Distributed I/O controller) (Page 98)

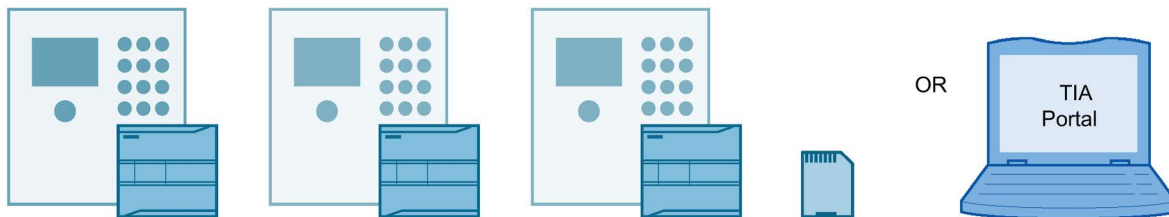
ET 200S (Distributed I/O controller) (Page 98)

ET 200M (Distributed I/O controller) (Page 99)

ET 200eco (Distributed I/O controller) (Page 99)

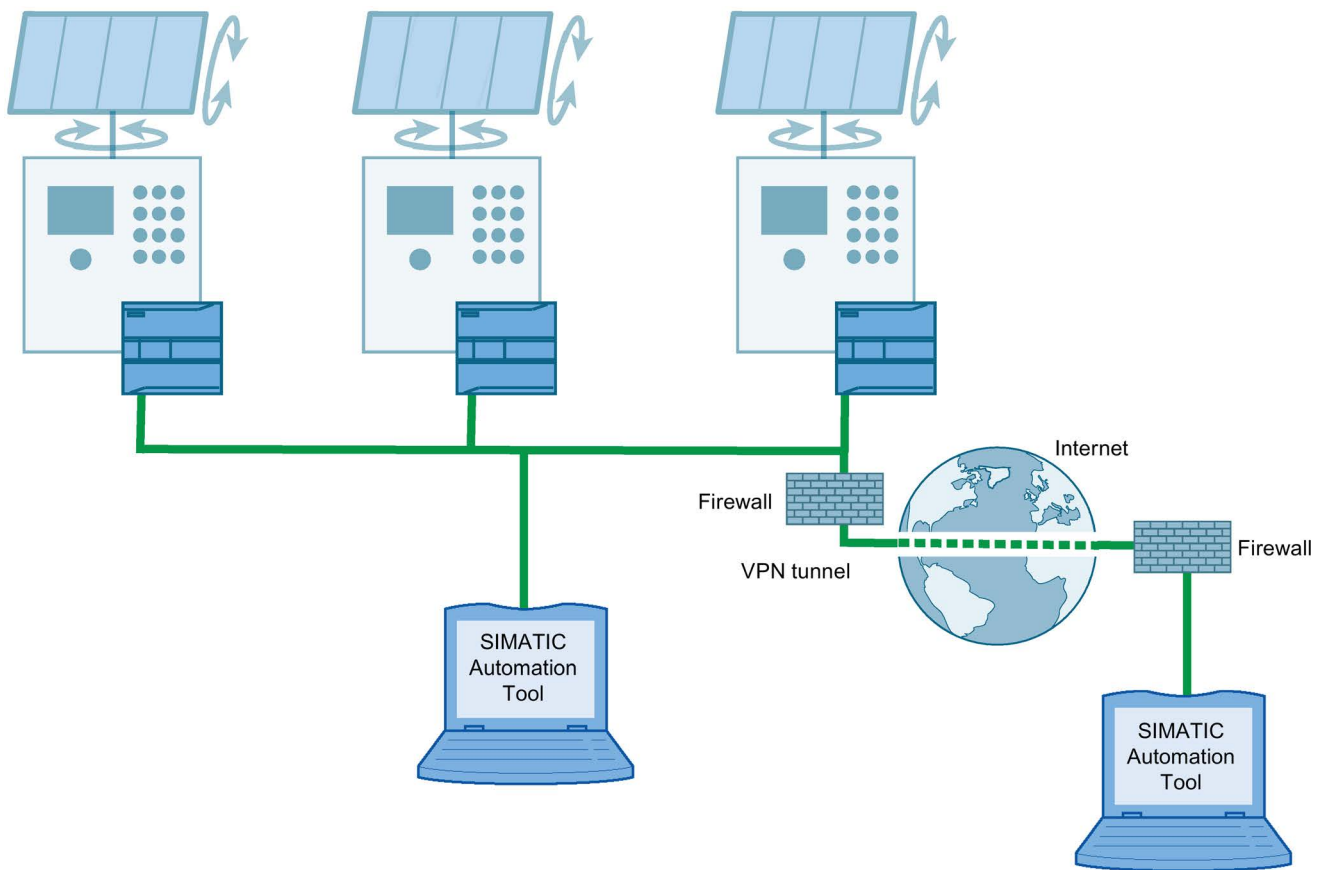
ET 200pro (Distributed I/O controller) (Page 100)

Example S7-1200 network installation



Panel shop fabrication and initial program load

- Each CPU uses the same hardware configuration and control program
- The CPU PROFINET configuration uses "Set IP Address on the device" and "Set PROFINET device name on the device" options
- Each Panel is identical
- The panel shop uses an SD Card or the TIA Portal to load the CPU program.



Field installation and operations

- Scan a network to find CPUs and modules
- Flash LEDs to identify a device
- Assign IP addresses and PROFINET names for each device, when you set up your network
- Set the time in CPUs to the current time in your PG/PC
- Update a CPU program/firmware and module firmware
- Put CPUs in RUN or STOP mode
- Read a CPU diagnostic buffer and upload service data
- Reset a CPU memory
- Perform CPU data backup and restore operations
- Document and save your network information in a standard text .csv file or an encrypted and password protected .sat file.
- Save time working with devices as a group and using parallel processing threads

1.2 Network access

Communicating with Siemens devices

PROFINET devices connected directly to a network are discovered by using the MAC (Media Access Control) address. A MAC address is unique to each device, cannot be changed, and is printed on the device. Connected PROFINET devices are discovered whether they are configured with an IP, subnet, and gateway address, or not configured (addresses are 0.0.0.0).

However, a directly connected CPU must have a valid IP address, before communication modules, signal modules, and decentralized I/O that are located behind the CPU become visible in the SIMATIC Automation Tool device table.

For example, an S7-1500 CPU has a PROFINET network connection to the SIMATIC Automation Tool and uses a local CP card to connect with another PROFINET network where decentralized I/O are connected. You must assign a valid IP address to the S7-1500 CPU before the decentralized I/O network is visible in the SIMATIC Automation Tool device table.

The type of network access you have depends on the command that you execute, as shown in the following table.

SIMATIC Automation Tool command	Device address used	Must provide CPU password for a protected CPU	PG/PC and device connectivity
Scan (discover CPUs and I/O devices)	MAC	No	<ul style="list-style-type: none"> Local network: You can access devices through Ethernet switches, but cannot access devices on another network through an IP address router. VPN (Virtual Private Network) connection to a local network
Flash LEDs on devices	MAC	No	
Set IP address, subnet mask, and gateway address on devices	MAC	No	
Set PROFINET name on devices	MAC	No	
Reset devices to factory default values (for PROFINET I/O devices only)	MAC	-	
Set CPU time to PG/PC time	IP	Yes	<ul style="list-style-type: none"> Local network: You can access devices through Ethernet switches. Remote network: You can access devices on another network through an IP address router. VPN connection to local network
Download program to CPUs	IP	Yes	
Download new firmware to CPUs and I/O devices	IP	Yes	
Put CPUs in RUN or STOP	IP	Yes	
Set CPU time to PG/PC time	IP	Yes	
Reset CPU memory	IP	Yes	
Read CPU Diagnostic buffer	IP	Yes	
Upload service data from CPUs	IP	Yes	
Backup/restore CPU data	IP	Yes	
Reset devices to factory default values (for CPU devices only)	IP	Yes	

Note**IP subnet mask assignment on a local network**

The PG/PC running the SIMATIC Automation Tool and devices connected to the local network must use the same subnet mask assignment.

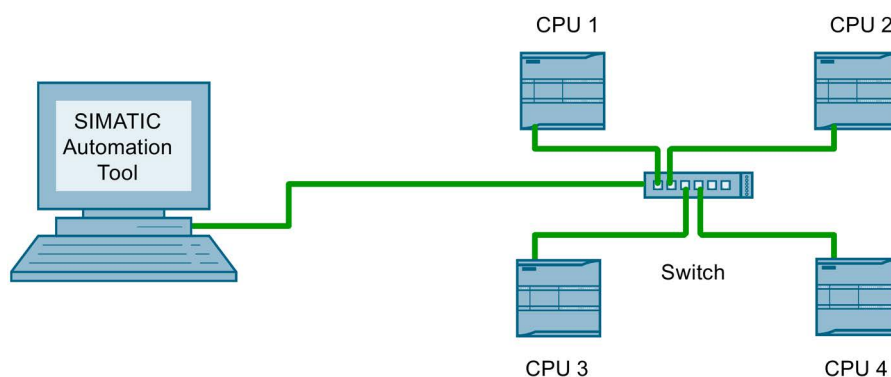
1.3 Network options

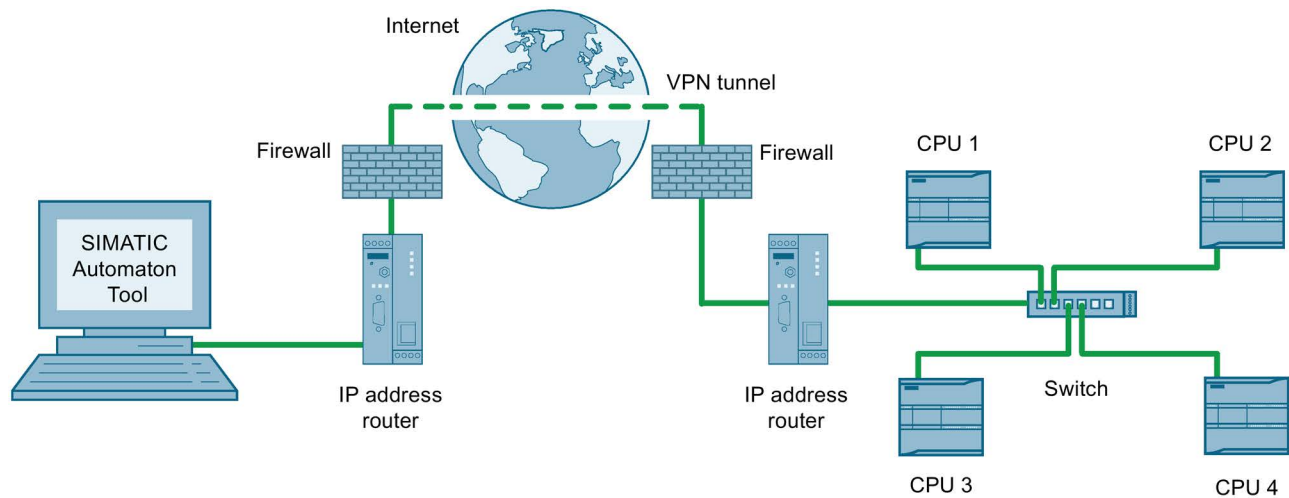
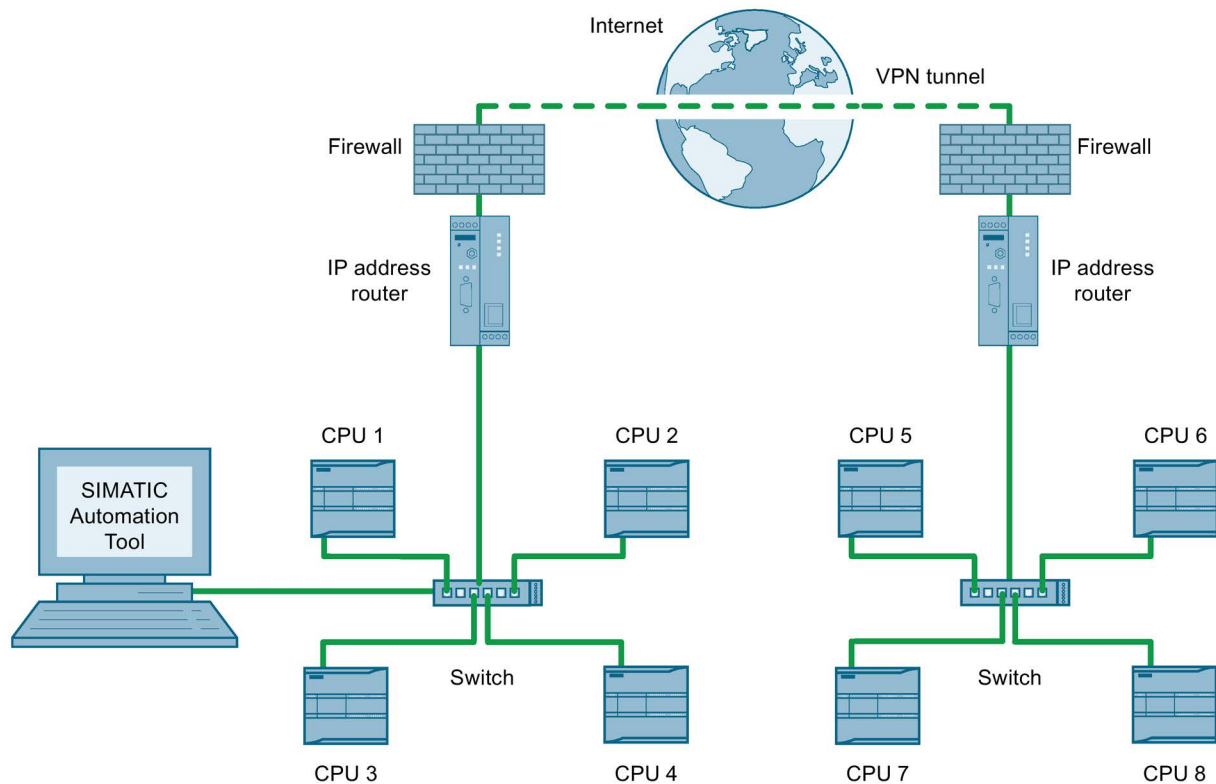
Local and remote networks

The following examples show local and remote networks that the SIMATIC Automation Tool can use. The diagrams are simplified to show basic connectivity and do not show the local I/O, CPs, and distributed I/O devices (PROFINET and PROFIBUS) that are also accessible. Different network topologies are also possible.

IP addressing and therefore valid network IP addresses are necessary to get access behind Ethernet IP routers. You can fill the SIMATIC Automation Tool device table by automatically scanning a network (Page 25) or by importing a list (Page 54) that identifies stations behind a router.

Example 1: S7-1200 local network



Example 2: S7-1200 remote network**Example 3: S7-1200 combined local and remote networks**

1.4 .NET API (application interface) .dll file

The Microsoft .NET API used by the SIMATIC Automation Tool is documented in this user guide. You can create your own application software that uses the API to perform the same device operations as the SIMATIC Automation Tool.

The SIMATIC Automation Tool must be installed on any PG/PC that uses this API. The SIMATIC Automation Tool and your application software use the API .dll file and additional S7 communication files. The SIMATIC Automation Tool installation provides all the files that you need.

API .dll file

The AutomationToolAPI.dll file is located in the folder where the SIMATIC Automation Tool is installed.

See also

SIMATIC Automation Tool API for .NET framework (Page 65)

Prerequisites and communication setup

2.1 PG/PC Operating system requirement

PG/PC operating systems

The SIMATIC Automation Tool can run on the 32-bit and 64-bit versions of Microsoft Windows 7, 8, and 8.1.

2.2 Installing the SIMATIC Automation Tool

Tool installation

1. On your PG/PC, save all your work in progress and close all applications, before installing the SIMATIC Automation Tool.
2. Execute the SIMATICAutomationToolSetup.exe file to begin the installation.

2.3 Starting the SIMATIC Automation Tool

Options for starting the SIMATIC Automation Tool:

- Double-click the SIMATIC Automation Tool shortcut icon on your desktop.
- Use the Windows Start button.
 - Click the Windows start button and "All Programs".
 - Click the "Siemens Automation" folder, then the "SIMATIC Automation Tool" folder, and finally "Automation Tool".
- Start the Windows command prompt (cmd.exe) and enter the executable file name and optional project parameter.
AutomationTool.exe [projectname.sat].

2.4 CPU configuration requirements

Ethernet address configuration

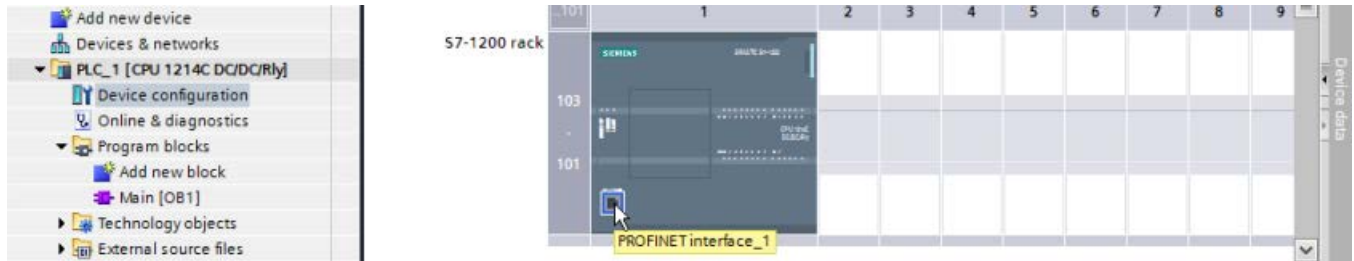
If you want the SIMATIC Automation Tool to set the IP address or PROFINET name of a CPU, then your TIA portal project must enable these actions in the CPU device configuration. Use the TIA portal to view and modify a program's IP protocol setting, as shown in the following S7-1200 example.

IP address and PROFINET name change

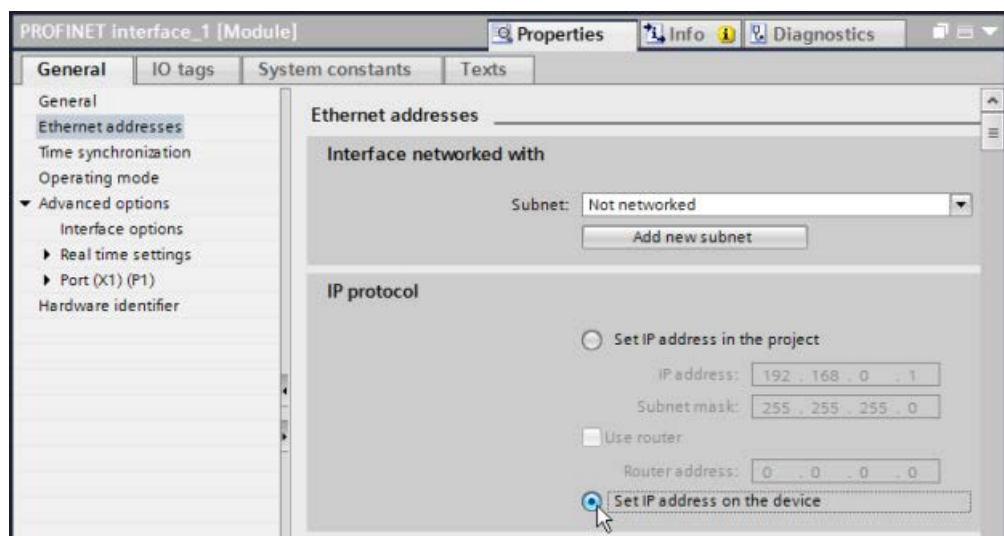
- It is **possible** for CPUs directly connected to the network that is connected to the SIMATIC Automation Tool (including connection through an Ethernet switch).
- It is **not possible** for CPU's with an indirect connection through a CP module, or a CPU's secondary Ethernet port when the direct connection is to the primary Ethernet port.
- It is **not possible** for CPU's on another network with a connection to the SIMATIC Automation Tool that passes through an IP address router.

S7-1200 example configuration

1. Click the PROFINET port on the device configuration CPU image, to view the port parameters.



2. On the **Properties** tab, click the **General** tab to view the **Ethernet addresses** options. Click the **SET IP address on the device** option. This option may be called **Set IP address using a different method**, depending on the TIA portal version that you have. For multi-port devices like the S7-1500 CPU, you can similarly configure all ports to enable IP address changes (when connected to the SIMATIC Automation Tool) or you can configure only the port you want to change.



3. Also on the **Ethernet addresses** options, click the **Set PROFINET device name on the device** option. This selection allows the SIMATIC Automation Tool to assign names. For multi-port devices like the S7-1500 CPU, you can similarly configure all ports to enable PROFINET name changes (when connected to the SIMATIC Automation Tool) or you can configure only the port you want to change.



4. Save your project and download the new configuration changes to the CPU.

Note

Default settings of PROFINET IP parameters

When you create a new TIA portal project, the default PROFINET parameter options are set to **"Set IP address in the project"** and **"Generate PROFINET device name automatically"**. With the default options, you cannot set IP addresses or PROFINET device names with the SIMATIC Automation Tool. However, you can use other CPU operations like RUN/STOP control, program/firmware updates, time setting, and service data/diagnostic analysis.

Note

Setting IP addresses for S7-1200 CPUs with V1 firmware

S7-1200 V1 CPU's do not support the "Set IP address on the device" option. If the IP addresses were previously set by a TIA portal program, then you must perform a factory reset operation to delete the old IP addresses before the SIMATIC Automation Tool can assign new IP addresses.

2.5 Communication setup

Identifying the network interface card connected to your device network

After you connect your PG/PC to a network, then you can use the Windows control panel to see the name of the network interface card.

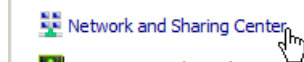
In the following example, S7-1200 CPUs are connected to a USB port on a PC running Windows 7. The network interface card is a USB to Ethernet converter device. The options that you actually see on your PG/PC depend on your network hardware.

Use the Windows Control Panel to identify the name of the device.

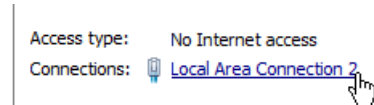
1. Open the Windows Control Panel



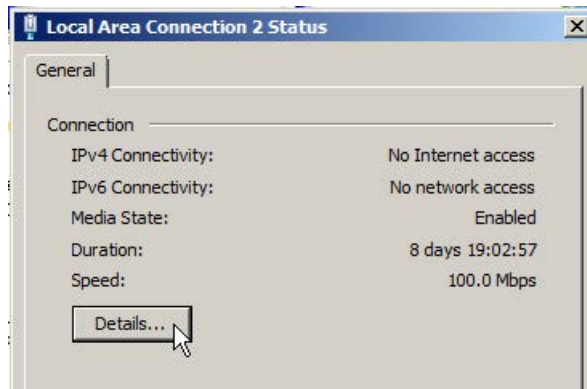
2. Click on the Network and Sharing center.



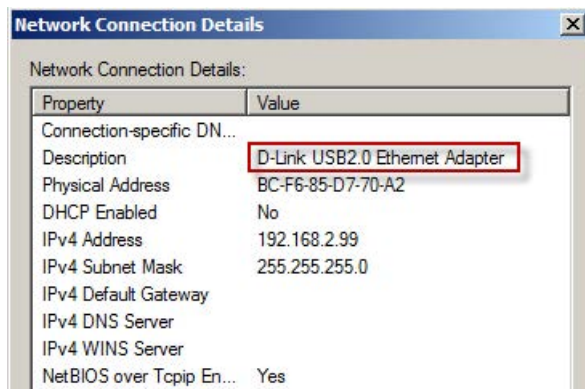
3. View your active networks and click on the network that is connected to the S7-1200 CPUs.



4. Click on the Details button in the connection status display.



5. View the description of the network interface.



Assigning the network interface card in the SIMATIC Automation Tool

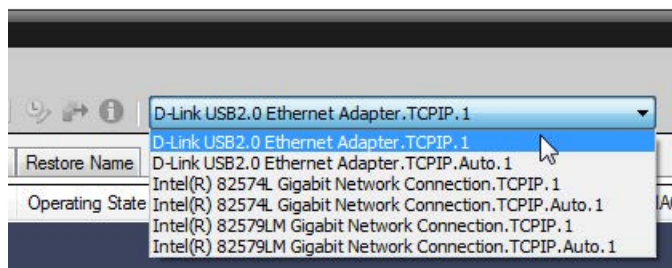
You must assign the network interface card to a new project, before communication can begin. Start the SIMATIC Automation Tool, click on the Network Interface Card drop-down list, and select the network card that is connected to your Siemens device network.

You may see different network cards from those shown in the following image, because the list shows the network interfaces that are available in your PG/PC.

If the network interface card is selected, but the devices do not have valid IP addresses, then you cannot use the IP address operations. However, you can use the MAC address based operations and set up valid IP addresses for your network.

MAC address operations

- Scan for CPUs and modules
- Flash LEDs to locate devices
- Set Ethernet IP addresses
- Set PROFINET names
- Reset to factory default values (for PROFINET I/O devices only)



As seen in the preceding image, there can be two entries for one network card and the difference is the addition of the characters ".Auto".

"Auto" refers to the auto-negotiation between two Ethernet stations for communication mode and speed.

Both options in the drop-list (auto and fixed) should work. However, some network hardware may require that you use only one of the settings to communicate efficiently. Try the simpler fixed selection without "auto" first.

For example, the S7-1200 CPU can communicate at either 10 or 100 Mbit/s (half or full duplex). You want PROFINET compatible full duplex 100 Mbit/s speed. If a problem occurs in the connection negotiation, the speed may be limited to half-duplex 10 Mbit/s communication.

If the network interface card is selected and the device IP addresses are valid, then you can use the SIMATIC Automation Tool operations that use an IP address.

IP address operations

- Set CPU time to PG/PC time
- Program/firmware update
- Put CPUs in RUN or STOP
- Reset CPU memory
- Read Diagnostic buffer
- Upload service data
- Backup and restore
- Reset to factory default values (for CPU devices only)

Note

Communication problems with the SIMATIC Automation Tool

For example, you send an operation command to multiple CPUs, but a CPU does not complete the operation and a communication error displayed for that CPU. However, other CPUs are communicating and executing the operation as expected.

If you experience communication problems, then reduce the maximum number of simultaneous operations (threads/connections) that is assigned in the **Tools > Options** dialog box. Close and restart the SIMATIC Automation Tool application, then try the group operation again.

Tool operations

3.1 Scan a network and use the device table

Basic operation

The device table for a new SIMATIC Automation Tool project is empty. You can assign a network interface card, scan a network, and fill the device table automatically.

The device table is filled with information collected by a network scan. You can see a parameter's existing value and enter new values into the appropriate column, for each device.

Device table cells have either a light gray or dark gray background:

- You can enter new values in the light gray cells
- You cannot enter new values in the dark gray cells
- A disabled dark gray cell prevents entering new values and indicates that this action is not supported, for this device.

After new IP addresses are entered in the device table, you can use the "Download" command to transfer the address assignments into selected directly connected devices.

Once the IP address assignments for all your devices are valid, you can use the SIMATIC Automation Tool operations that require IP addresses.

In addition, a network scan shows devices located behind CPUs and IP address routers.

Scan the network



1. Select Scan from the Network menu or click the Scan button on the toolbar.
2. The SIMATIC Automation Tool will fill a new device table or update an existing table with information from accessible devices.

The initial scan shows devices that have a direct connection to the SIMATIC Automation Tool and are shown at the top level (left-most) in the device hierarchy.

A directly connected device can use all MAC addressed operations (with IP address unconfigured or configured) and all IP addressed operations (with IP address configured), if the device firmware supports the operations.

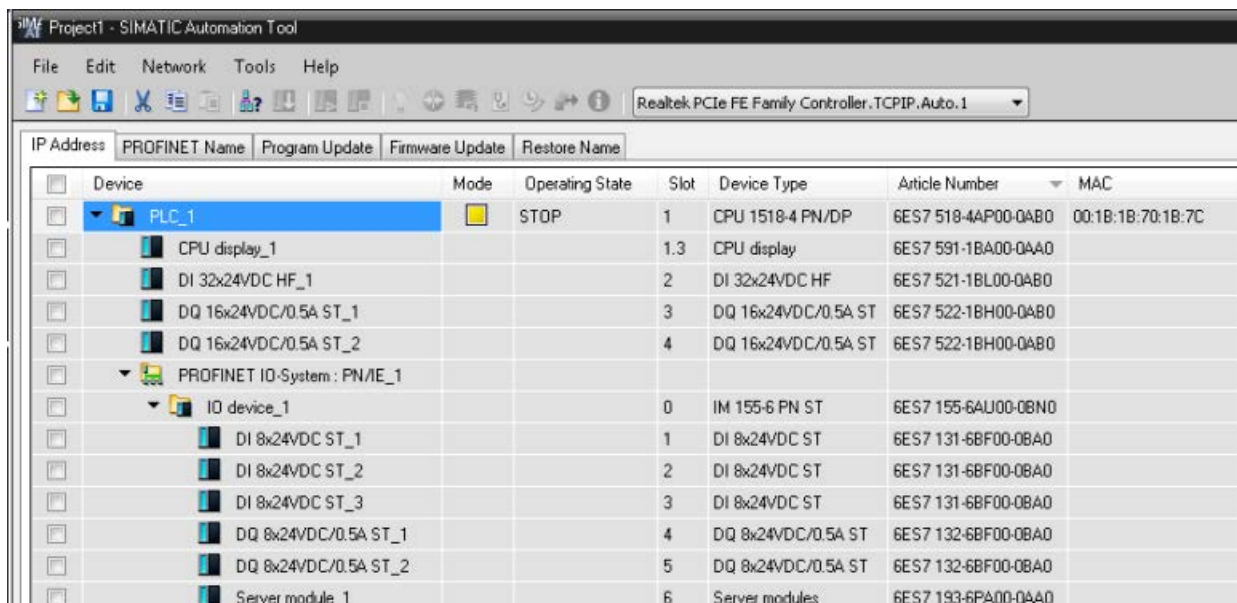
PROFINET I/O devices can be listed twice in the device table. Once on the top level, where a direct connection allows all SIMATIC Automation Tool operations, and a second entry at a lower level behind a CPU (with IP address configured), where an indirect connection restricts the operations to firmware update only.

The screenshot shows the SIMATIC Automation Tool window with the 'Network' menu open. The main area displays a table of discovered devices. The table has columns for Device, Mode, Operating State, Slot, Device Type, Article Number, and MAC. The devices listed include several PLC_1 units, a CPU_SPSC_2ch_2A unit, and a PLC1200 with ET200SP. The operating states are mostly STOP, with one CPU in RUN mode.

Device	Mode	Operating State	Slot	Device Type	Article Number	MAC
PLC_1	STOP	STOP	1	CPU 1518-4 PN/DP	6ES7 518-4AP00-0AB0	00:1B:1B:70:1B:7C
PLC_1	STOP	STOP	1	CPU 1516F-3 PN/DP	6ES7 516-3FN00-0AB0	28:63:36:81:D0:81
PLC_1	STOP	STOP	1	CPU 1516-3 PN/DP	6ES7 516-3AN00-0AB0	00:1B:1B:13:A0:F8
PLC_1	STOP	STOP	1	CPU 1515-2 PN	6ES7 515-2AM00-0AB0	00:1B:1B:13:77:66
CPU_SPSC_2ch_2A	RUN	RUN	1	CPU 1512SP-1 PN	6ES7 512-1DK00-0AB0	28:63:36:16:29:A6
CPU	STOP	STOP	1	CPU 1511-1 PN	6ES7 511-1AK00-0AB0	00:1B:1B:13:8D:D3
CPU	STOP	STOP	1	CPU 1510SP-1 PN	6ES7 510-1DJ00-0AB0	28:63:36:03:2D:D8
PLC_1	STOP	STOP	1	CPU 1217C DC/DC/DC	6ES7 217-1AG40-0XB0	28:63:36:80:60:42
CPU	STOP	STOP	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG40-0XB0	28:63:36:88:A7:54
PLC1200 with ET200SP	STOP	STOP	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG40-0XB0	28:63:36:88:A7:69

Expand the device rows and show local modules and decentralized I/O devices.

Click the expand icon to expand a device row. Use the right-click menu or Edit menu to expand/collapse all levels.



Device	Mode	Operating State	Slot	Device Type	Article Number	MAC
PLC_1	STOP		1	CPU 1518-4 PN/DP	6ES7 518-4AP00-0AB0	00:1B:1B:70:1B:7C
CPU display_1			1.3	CPU display	6ES7 591-1BA00-0AA0	
DI 32x24VDC HF_1			2	DI 32x24VDC HF	6ES7 521-1BL00-0AB0	
DQ 16x24VDC/0.5A ST_1			3	DQ 16x24VDC/0.5A ST	6ES7 522-1BH00-0AB0	
DQ 16x24VDC/0.5A ST_2			4	DQ 16x24VDC/0.5A ST	6ES7 522-1BH00-0AB0	
PROFINET IO-System : PN/IE_1						
IO device_1			0	IM 155-6 PN ST	6ES7 155-6AU00-0BN0	
DI 8x24VDC ST_1			1	DI 8x24VDC ST	6ES7 131-6BF00-0BA0	
DI 8x24VDC ST_2			2	DI 8x24VDC ST	6ES7 131-6BF00-0BA0	
DI 8x24VDC ST_3			3	DI 8x24VDC ST	6ES7 131-6BF00-0BA0	
DQ 8x24VDC/0.5A ST_1			4	DQ 8x24VDC/0.5A ST	6ES7 132-6BF00-0BA0	
DQ 8x24VDC/0.5A ST_2			5	DQ 8x24VDC/0.5A ST	6ES7 132-6BF00-0BA0	
Server module_1			6	Server modules	6ES7 193-6PA00-0AA0	

Only the firmware update operation is possible for indirectly connected devices.

Devices on the lower levels represent local I/O and decentralized I/O that are indirectly connected to the SIMATIC Automation Tool through a directly connected CPU. A valid IP configuration is necessary in the CPU, before a CPU's local devices are visible in the device table.

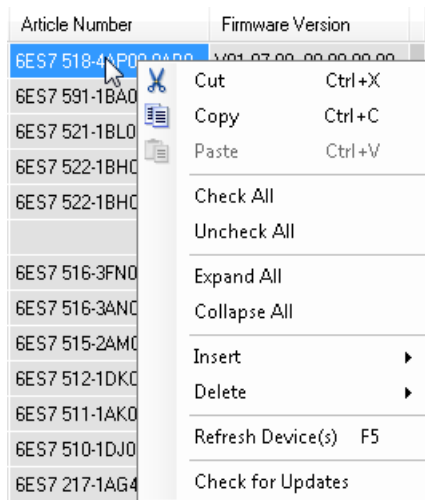
Devices on the third and fourth levels represent decentralized I/O devices (PROFINET and PROFIBUS devices). An IP configuration is necessary in a level two decentralized I/O controller, before the decentralized I/O (for example, head module and I/O modules) are visible in the device tree.

Working with the device table

- Use the Microsoft Excel compatible cell selection and copy/paste clipboard inside the device table or outside to/from another application.
- Click on a column header to sort or reverse sort the rows, by that column's data.
- Right-click a column header to show/hide any column.
- Click the checkbox column header or use the edit menu to Check All or Uncheck all rows.
- You can create row filters for the Device, Device type, and Article number columns.

3.1 Scan a network and use the device table

Right-click menu for all table cells



Cut current selection.

Copy current selection.

Paste current selection.

Check all device rows (first column).

Uncheck all device rows (first column).

Expand all rows in the Device column.

Collapse all rows in the Device column.


Insert a new device row

Delete selection or checked device(s)

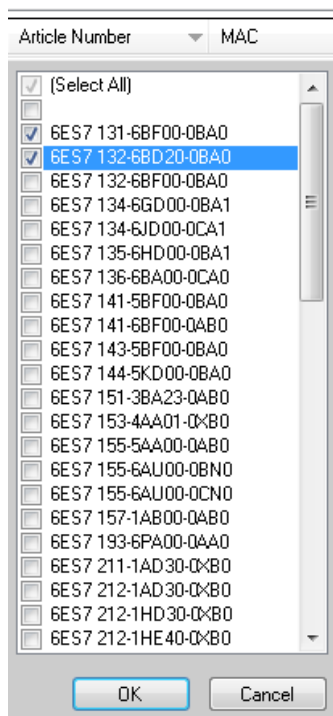
Refresh checked devices

Open SIEMENS industry support web page for current row's article number. Check for latest device information and firmware updates.

Filtering the displayed rows

Filtering is supported on the Device, Device Type, and Article number columns. When the mouse cursor hovers over a column that supports filtering, the filter button  appears in the column header. Click this button to open the filter window.

The following example article number filter has 6ES7 131-6BF00-0BA0 and 6ES7 132-6BD20-0BA0 selected. When you click the OK button, the device table only displays rows that have these article numbers.



Scan rules for existing table entries

- If a MAC address already exists in the table, then the row for that MAC address has the IP address, Subnet, and Gateway fields updated. The data in all other fields remains.
- If a MAC address is not listed, then a new row is created. The MAC address, IP Address, Subnet, and Gateway are added. For a new row all other fields are empty.

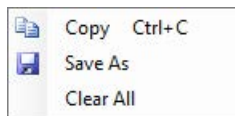
3.1 Scan a network and use the device table

Event log

The event log area is below the device table. When you select devices and start an operation, information about successful and unsuccessful results is displayed in the event log. The event log column headers have a right-click menu to show/hide columns.

Log						
	Date and time	Device	MAC	IP	Event	Result
✓	6/11/2015 8:07:33 AM	PLC_1	00:1B:1B:70:1B:7C	192.168.1.2	Transition to RUN	Success
✗	6/11/2015 7:57:20 AM	IM 155-6 PN ST	28:63:36:0F:D0:4D	192.168.1.9	Transition to STOP	This operation is not supported

Use a right-click in the Event log to open the menu shown below.




Copy log items to the clipboard
 Save log items to a file
 Clear all log items

3.2 Download new IP, subnet, and gateway addresses

Change IP addresses

Use the following steps to update the IP addresses.

1. Click the "IP Address" tab.
2. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
3. Enter address changes in the "New IP", "New subnet", and "New Gateway" columns.
-  4. Select Download from the Network menu or click the Download button on the toolbar
5. The Download operation sets the IP, subnet, and gateway addresses in the selected devices.
6. The Event log below the device table shows the results of this operation.

MAC	IP	Subnet	Gateway	New IP	New Subnet	New Gateway	Password
00:1B:1B:70:1B:7C	192.168.1.2	255.255.255.0	192.168.1.2	192.168.2.23	255.255.255.0	192.168.2.23	
28:63:36:0F:D0:4D	192.168.1.9	255.255.255.0	192.168.1.9	192.168.2.24	255.255.255.0	192.168.2.24	
28:63:36:0F:D0:70	192.168.1.10	255.255.255.0	192.168.1.10	192.168.2.26	255.255.255.0	192.168.2.26	
00:1B:1B:13:A0:F8	192.168.1.15	255.255.255.0	192.168.1.15				
28:63:36:83:82:5F	192.168.2.13	255.255.255.0	0.0.0.0				
28:63:36:03:2D:DB	192.168.2.20	255.255.255.0	192.168.2.20				
00:1B:1B:13:2B:40	192.168.2.22	255.255.255.0	192.168.2.22				*****
00:1B:1B:13:77:55	192.168.2.23	255.255.255.0	192.168.2.23				
00:1B:1B:13:77:66	192.168.2.24	255.255.255.0	192.168.2.24				

3.3 Download new PROFINET device names

Changing PROFINET device names

PROFINET name rules


Valid names follow the standard DNS (Domain Name System) naming conventions.

A maximum of 63 characters is allowed. Valid characters are the lower case letters "a" through "z", the digits 0 through 9, the hyphen character (minus sign), and the period character.

Invalid names

- The name must not have the format n.n.n.n where n is a value of 0 through 999.
- You cannot begin the name with the string port-nnn or the string port-nnnnnnnn, where n is a digit 0 through 9. For example, "port-123" and "port-123-45678" are illegal names.
- A name cannot start or end with a hyphen "-" or period "." character.

Use the following steps to update the PROFINET device names.

1. Click the "PROFINET name" tab.
2. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
3. Enter a new PROFINET name in the "New PROFINET Name" column.
4.  Select Download from the Network menu or click the Download button on the toolbar.
5. The Download operation sets new PROFINET names in the selected devices.
6. The Event log below the device table shows the results of this operation.

MAC	IP	PROFINET Name	New PROFINET Name	Password
00:1B:1B:70:1B:7C	192.168.1.2	plcxb1.profinetxainterf...	plcname1	
28:63:36:81:D0:B1	192.168.2.26	plcxb1.profinetxainterf...	plcname2	
00:1B:1B:13:A0:F8	192.168.1.15	plcxb1.profinetxainterf...	plcname3	
00:1B:1B:13:77:66	192.168.2.24	plcxb1.profinetxainterf...	plcname4	
28:63:36:16:29:A6	192.168.2.202	cpuxbspscxb2chxb2a...	plcname5	
00:1B:1B:13:8D:D3	192.168.2.52			
28:63:36:03:2D:DB	192.168.2.20	thermalkbtestffda		
28:63:36:80:60:42	192.168.2.10	plcxb1d0ed		
28:63:36:88:A7:54	192.168.2.80			

3.4 Set CPUs to RUN or STOP mode

Change CPUs to RUN mode

1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
2. Select RUN from the Network menu or click the RUN toolbar button. A valid program must exist in the CPU before it can enter RUN mode.
3. Selected CPUs are set to RUN mode.
4. The Mode and Operating state columns in the device table indicate the current CPU state. Yellow means STOP mode. Green means RUN mode, and RED means CPU fault.
5. The Event log below the device table shows the results of this operation.



Project1 - SIMATIC Automation Tool


File Edit Network Tools Help








Realtek PCIe FE Family Controller.TCPIP.Auto.1

IP Address PROFINET Name Program Update Firmware Update Restore Name

Device	Mode	Operating State	Slot	Device Type	Article Number	MAC
<input checked="" type="checkbox"/> CPU_SP5C_2ch_2A		RUN	1	CPU 1512SP-1 PN	6ES7 512-1DK00-0AB0	28:63:36:16:29:A6
<input checked="" type="checkbox"/> CPU		RUN	1	CPU 1511-1 PN	6ES7 511-1AK00-0AB0	00:18:1B:13:8D:D3
<input checked="" type="checkbox"/> CPU		RUN	1	CPU 1510SP-1 PN	6ES7 510-1DJ00-0AB0	28:63:36:03:2D:DB
<input checked="" type="checkbox"/> PLC_1		RUN	1	CPU 1217C DC/DC/DC	6ES7 217-1AG40-0XB0	28:63:36:80:60:42
<input checked="" type="checkbox"/> CPU		RUN	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG40-0XB0	28:63:36:88:A7:54
<input type="checkbox"/> PLC1200 with ET200SP		RUN	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG40-0XB0	28:63:36:88:A7:69
<input type="checkbox"/> PLC_2		RUN	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG31-0XB0	00:1C:06:09:49:DB

Change CPUs to STOP mode


1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
2.  Select STOP from the Network menu or click the STOP button on the toolbar.
3. Selected CPUs are set to STOP mode.
4. The Mode and Operating state columns in the device table indicate the current CPU state. Yellow means STOP mode. Green means RUN mode, and RED means CPU fault.
5. The Event log below the device table shows the results of this operation.

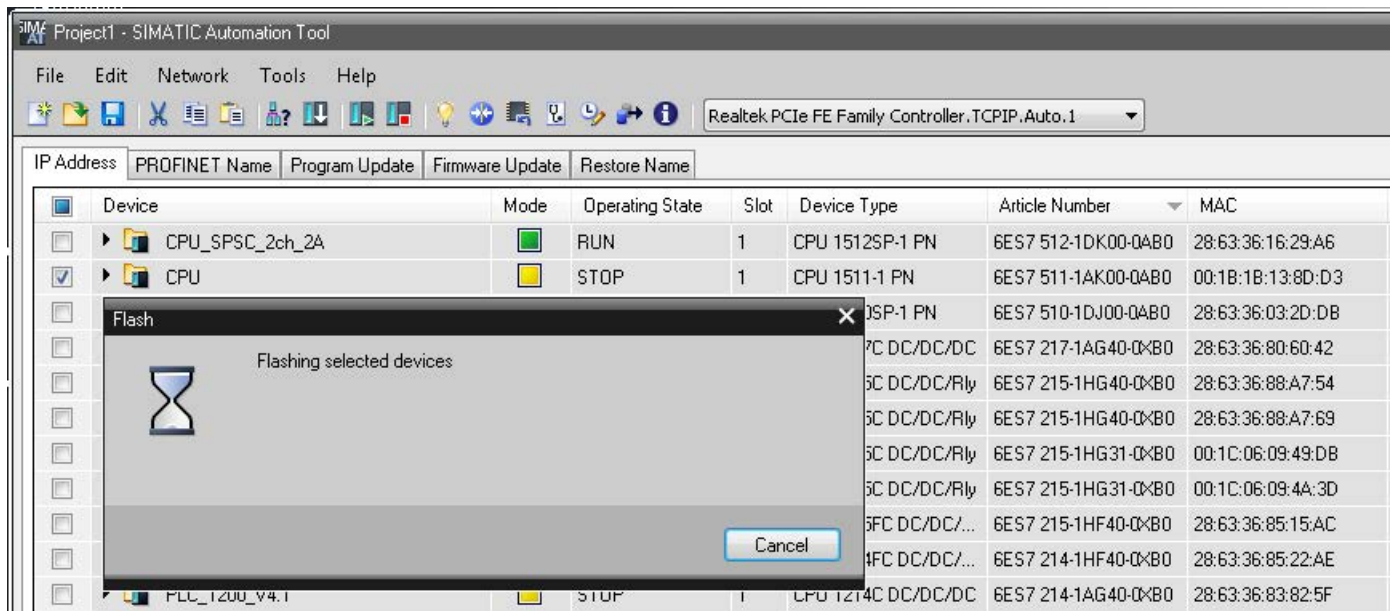
IP Address PROFINET Name Program Update Firmware Update Restore Name								
<input type="checkbox"/>	Device	Mode	Operating State	Slot	Device Type	Article Number	MAC	
<input checked="" type="checkbox"/>	CPU_SPSC_2ch_2A		STOP	1	CPU 1512SP-1 PN	6ES7 512-1DK00-0AB0	28:63:36:16:29:A6	
<input checked="" type="checkbox"/>	CPU		STOP	1	CPU 1511-1 PN	6ES7 511-1AK00-0AB0	00:1B:1B:13:8D:D3	
<input checked="" type="checkbox"/>	CPU		STOP	1	CPU 1510SP-1 PN	6ES7 510-1DJ00-0AB0	28:63:36:03:2D:D8	
<input checked="" type="checkbox"/>	PLC_1		STOP	1	CPU 1217C DC/DC/DC	6ES7 217-1AG40-0XB0	28:63:36:80:60:42	
<input checked="" type="checkbox"/>	CPU		STOP	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG40-0XB0	28:63:36:88:A7:54	
<input type="checkbox"/>	PLC1200 with ET200SP		RUN	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG40-0XB0	28:63:36:88:A7:69	
<input type="checkbox"/>	PLC_2		RUN	1	CPU 1215C DC/DC/Rly	6ES7 215-1HG31-0XB0	00:1C:06:09:49:D8	

3.5 Flash the LEDs on devices

Locate a device by flashing LED lights

The Flash button will flash the LEDs of selected devices. The Flash operation helps you physically locate which device has a specific MAC address. You can use the flash LEDs operation in RUN mode or STOP mode.

1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
2.  Select flash from the Network menu or click the Flash toolbar button.
3. Selected devices flash their LEDs.
4. Flashing continues until you click the cancel button.



3.6 Download new programs to CPUs

Preparing a program for use with the SIMATIC Automation Tool

A TIA portal program must be transferred by the TIA portal software to a SIMATIC memory card, USB flash drive, or another partition of your PG/PC hard drive before the program is usable with the SIMATIC Automation Tool. Refer to the TIA portal documentation about how to transfer a program to a storage device.

After the TIA portal transfers the program to a storage device, you can use the Windows Explorer to transfer the program to the folder that is used by the SIMATIC Automation Tool.

Copy the "SIMATIC.S7S" folder for each program

Follow these steps to make a program accessible to the SIMATIC Automation Tool

1. Run the SIMATIC Automation Tool and view the **Tools>Options** setting and note the path assignment for program update files. The default path is C:\ProgramData\Siemens\SIMATIC Automation Tool\Programs. You can modify the path, if you prefer to use another folder.
2. Create folders under the programs path that is assigned in the Options dialog. Create one folder for each program and create a folder name that identifies the program. The folder names that you create will appear in the SIMATIC Automation Tool program drop-down list.
3. Use the Windows Explorer to copy the "SIMATIC.S7S" folder (including all subfolders and files) to the folder assigned in the SIMATIC Automation Tool. The SIMATIC Automation Tool can only download programs to a CPU. Optional data log and recipe folders that may exist cannot be downloaded. A TIA portal program (a "SIMATIC.S7S" folder) can be put in a zip file archive and sent to a remote location.

Note

TIA portal program data is encrypted

Details like the project name or target CPU, of a TIA portal program, cannot be discovered from the encrypted data that is stored in a SIMATIC.S7S folder. You cannot identify one program's SIMATIC.S7S folder from another program's SIMATIC.S7S folder.

You must create and name subfolders under the SIMATIC Automation Tool program update folder that identify a program's function or target CPU. Copy a program's SIMATIC.S7S folder into the subfolder that you named. The folder names that you create appear in the SIMATIC Automation Tool "Program" column drop-down list and provide the path to the correct SIMATIC.S7S folder.

Download new programs to a CPU




WARNING

Verify that the CPU is not actively running a process before downloading a new program

Installing a new program causes the CPU to go to STOP mode, which could affect the operation of an online process or machine. Unexpected operation of a process or machine could result in death or injury to personnel and/or property damage.

After program data are copied to the program update folder, you can use the SIMATIC Automation Tool to load new programs in one or more CPUs.

1. Click the "Program Update" tab.
2. Click the left-side check box on CPUs to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
3. For each selected CPU, use the "Program" column drop-down list to select a program folder name. The drop-down list will show the folders that you created in the program update path.
-  4. Select the Download Command from the Network menu, or click the Download toolbar button to start the operation.
5. The Event log below the device table shows the results of this operation.

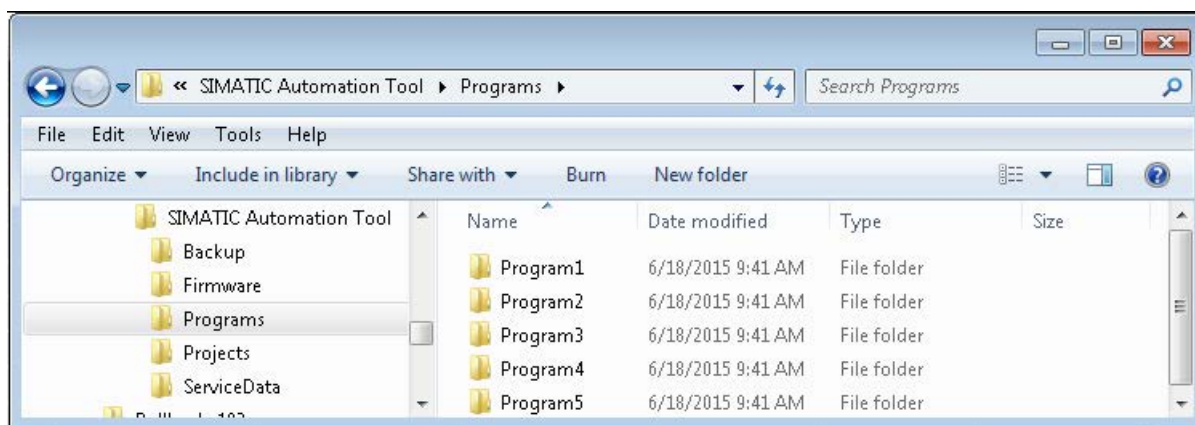
Example program update

The default path in the options dialog for program update is
C:\ProgramData\Siemens\SIMATIC Automation Tool\Programs.

If you want five different programs available for download, then you must create and name five folders under the path that is assigned in the Options dialog. Copy the entire "SIMATIC.S7S" folders to the five corresponding folders.

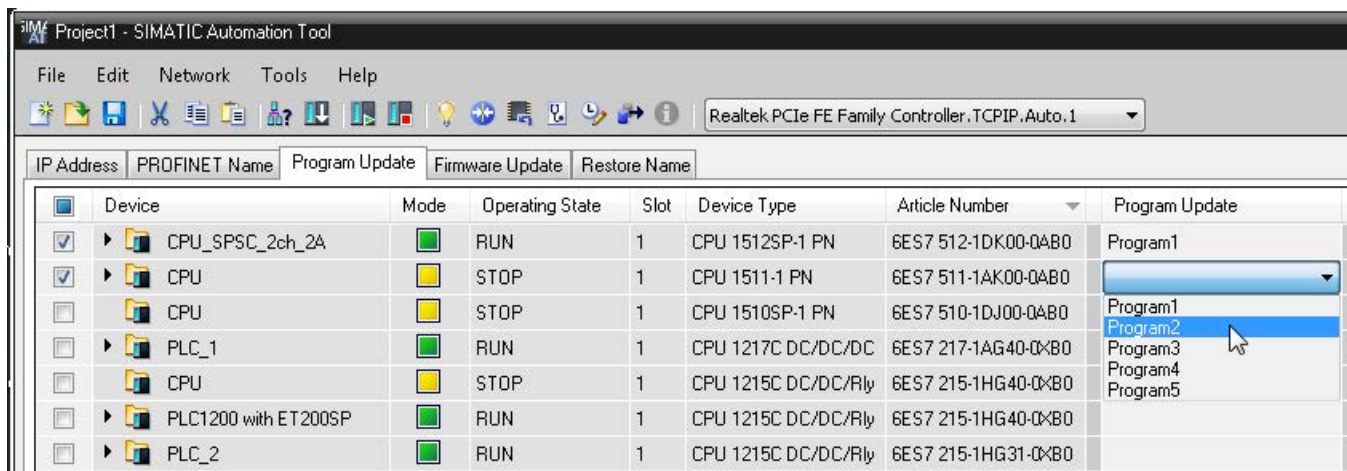
In this example, the folder names "Program1", "Program2", "Program3", "Program4", and "Program5" identify the available programs. You can use any folder name you want. The folder name could refer to a program function, or CPU location.

The following image shows the Windows Explorer view of the four subfolders under the Programs folder. The corresponding SIMATIC.S7S program folders are copied to these folders.



The following image shows the SIMATIC Automation Tool Program Update tab with the example folder names in the "Program" column drop-down list. Two S7-1500 CPUs are checked in the select column. You must use the drop-down list in the Program Update column to assign which program to use. If more than one CPU row is selected, then you must repeat the process and assign the correct program for each selected CPU.

Select the Download command on the Network menu, or click the toolbar Download button, to start the program update operation.



3.7 Install new firmware in CPUs and modules

Firmware updates

TIA Portal

A SIMATIC memory card can install firmware updates in CPUs and modules. Alternative firmware update methods include using the module information page of a CPU's built-in Web server, or using the TIA portal online and diagnostic functions.

SIMATIC Automation Tool

The SIMATIC Automation Tool can perform firmware updates on a group of CPUs and modules. You can use the new format single .upd file and the older (classic) format which uses three or more separate .upd files.

By default, the firmware update operation does not replace firmware with the same firmware version and allows only a single processing thread that must complete execution, before the next device operation is executed. To change this behavior, see the Tool menu (Page 60) topic. Be aware of the risk of doing multi-thread firmware updates that is described in that topic.

Note

S7-1200 CM communication modules must be configured before a firmware update

You can use the SIMATIC Automation Tool to update the firmware in unconfigured and configured SM and CM modules, except for left-side S7-1200 CM devices. For the S7-1200 CM, you must complete a TIA portal CM configuration and download the configuration, before you can update the CM firmware.

Note

CPU firmware downgrade

You can use the SIMATIC Automation Tool to downgrade CPU firmware (load a previous firmware version), but the IP address and program may be erased. In this case, the IP address is reset to 0.0.0.0 and a new network scan is required to communicate with this device. You must set the IP address to restore your previous network address.

Preparing firmware update files for use with the SIMATIC Automation Tool

- You can obtain firmware update software from the customer support (<https://www.siemens.com/automation/>) web site.
- Another option is to go directly to a device's customer support web page by right-clicking with the mouse cursor on a device row and then selecting the menu item "Check for updates". The Siemens support web page selection is controlled by the article number displayed in a device table row. For example, a "Check for updates" command on article number 6ES7 215-1HG31-0XB0 links to the corresponding S7-1215C web support page (<https://support.industry.siemens.com/cs/products/6es7215-1hg31-0xb0/cpu-1215c-dcdcrly-14di10do2ai2ao?pid=79072&ntp=Download&mlfb=6ES7215-1HG31-0XB0&lc=en-WW>)

For a CPU example, the firmware update file named **6ES7211_1AE40_0XB0_V04.00.02.exe** is only for the S7-1200 **CPU 1211C DC/DC/DC** model. If you use the .upd file within this package for any other S7-1200 CPU model, the update process will fail.

When you execute the update file and extract the files, you will see the following set of files and folders.

- file: S7-JOB.SYS
- folder: FWUPDATE.SYS contains the .upd file.
 - file: **6ES7 211-1AE40-0XB0 V04.00.02.upd** (.upd file used by the SIMATIC Automation Tool)

For an I/O module example, the firmware update file named **232-4HD32-0XB0_V203.exe** is only for the S7-1200 **SM1232 ANALOG OUTPUT 4AO** module. The self-extracting .exe file contains the file **6ES7 232-4HD32-0XB0 V02.00.03_00.00.00.00.upd** that is used by the SIMATIC Automation Tool.

Note

New format firmware update files

- The self-extracting .exe update package name must refer to the article number of the device that you want to update.
 - The extracted .upd file name must match the article number of the device and the firmware version that you want to load.
-

Note

Old format firmware update files

- The self-extracting .exe update package name must refer to the article number of the device that you want to update.
 - Contains three or more files depending on the firmware size.
 - Create a folder with any name underneath the C:\ProgramData\Siemens\SIMATIC Automation Tool\Firmware folder. You can name the folder with the article number and version number so it will be easier to identify, but you can use any name. The SIMATIC Automation Tool parses all firmware files at startup to confirm exact firmware version numbers.
-

Copy .upd files to the firmware update folder

The new format firmware update single .upd files have the target module model and version numbers in their file names. You can copy multiple .upd files to a single firmware folder and then identify the target module by the .upd file name.

1. Run the SIMATIC Automation Tool and view the **Tools>Options** setting and note the folder assignment for firmware update files. The default path is C:\ProgramData\Siemens\SIMATIC Automation Tool\Firmware. You can modify the default setting.
2. Copy all the .upd files you need to the firmware folder assigned by the **Tools>Option** command.




WARNING

Verify that the CPU is not actively running a process before installing firmware updates

Installing a firmware update for a CPU or module causes the CPU to go to STOP mode, which could affect the operation of an online process or machine. Unexpected operation of a process or machine could result in death or injury to personnel and/or property damage.

Download firmware updates to CPUs and modules

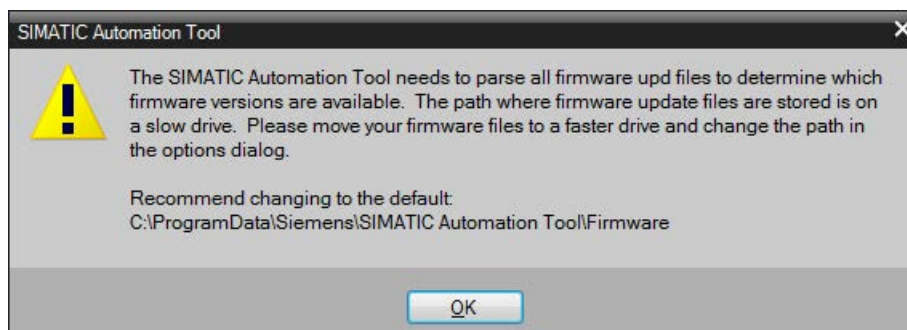
After .upd files are copied to the firmware update folder, you can use the SIMATIC Automation Tool to update the firmware in CPUs and modules.

1. Click the "Firmware Update" tab.
2. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
3. For each device row selected, use the "Firmware Version" column drop-down list and select a firmware version for either a CPU or module. The drop-down list will show the names of the .upd files that you copied to the firmware update path. If new firmware versions (.upd files) are available in the firmware update folder, then the latest version is automatically entered in the "Firmware version" column cell for a device.
4.  Select the Download Command from the Network menu, or click the Download toolbar button to start the operation.
5. Allow time for the firmware update to complete. Wait until the CPU lights stop flashing before attempting another operation with this CPU
6. The Event log below the device table shows the results of this operation.

Timeout error message due to slow communication with .upd file storage device

If you see the following error message box, then more than 5 seconds has elapsed and the SIMATIC Automation Tool has not completed processing all the .upd files in the firmware storage folder. The time required to open and scan all the .upd files is related to data access time and the number of .upd files in the folder.

- This timeout error can occur when communication with a remote storage device is too slow.
- To prevent this problem, assign a faster firmware data storage path with the **Tools>Options** menu command. Copy the .upd files you need to a faster local storage device and try the operation again.

**Example firmware update**

The default path for firmware update is C:\ProgramData\Siemens\SIMATIC Automation Tool\Firmware.

- If you want to have four different firmware versions available for downloading, then you must copy four different .upd files to the firmware subfolder.
- On the Firmware update tab, the SIMATIC Automation Tool lists the available versions (.upd files) in the "New Firmware Version" column drop-down list. One CPU was checked in the select column. So, you must use the FW Version drop-down list and assign a file for this CPU. If more than one CPU is selected, then you must repeat the process and assign the correct update file for each selected CPU.
- If multiple I/O modules of the same model exist, then one module firmware update will update all similar modules. I/O module firmware can be updated separately without updating firmware in the rack's CPU.
- Select the Download command on the Network menu, or click the toolbar Download button, to start the operation.



Note**You cannot update the firmware of some S7-1200 modules with the SIMATIC Automation Tool**

If you see the error message "The device requires both the CPU and module to support firmware update. This device can only be updated via SD card", then you cannot update the module firmware with the SIMATIC Automation Tool.

Modules that have article numbers containing xxx30 or xxx31 cannot be updated with the Automation Tool and you must use a SIMATIC memory card. This is only for S7-1200 modules installed on the left or right side of the CPU. For example, the middle part of the article number 6ES7232-4HD30-0XB0 contains 4DH30 and you cannot update the firmware of this module with the SIMATIC Automation Tool.

3.8 Backup and Restore CPU data


Backup device

New CPU data backup files are created and copied to the assigned folder. The default file path is C:\ProgramData\Siemens\SIMATIC Automation Tool\Backup.

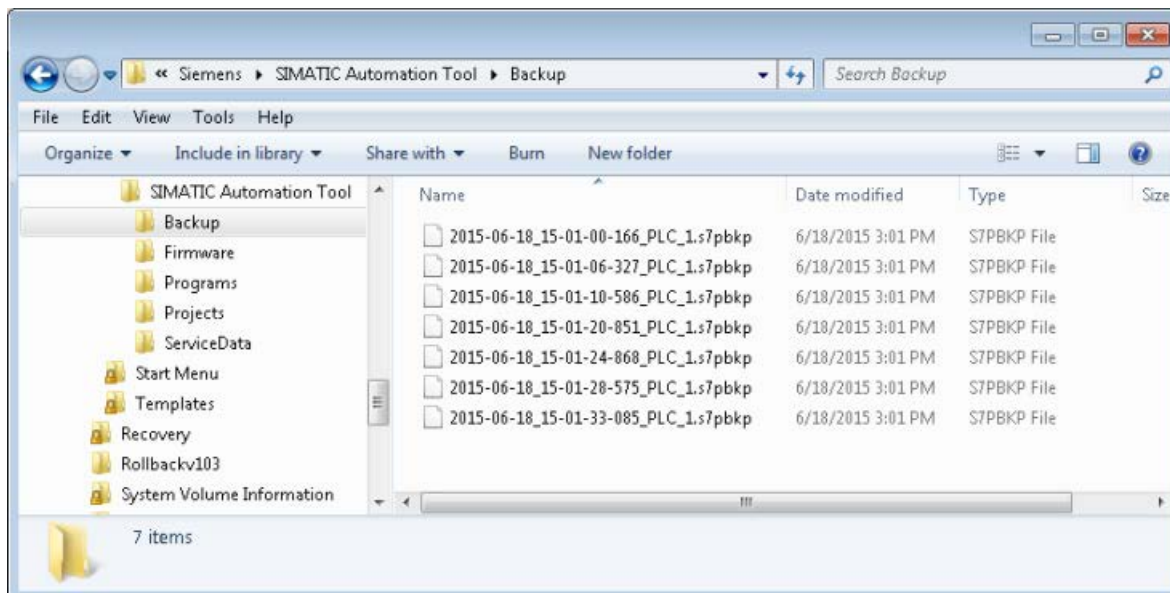
The backup file name contains the date, time, and PLC name similar to TIA Portal backup file names. You can use these files in the SIMATIC Automation Tool Restore Device operation.

You can start the backup operation from any tab selection (IP address, PROFINET Name, Program update, Firmware Update, or Restore Name).

Use the following steps to create new backup files.

1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
2.  Select "Backup Device" from the Network menu or click the "Backup device to file" toolbar button.
3. New CPU backup files are created and copied to the backup file path assigned in the Tools>Options menu.
4. The Event log below the device table shows the results of this operation.

The following image shows the backup file name format: Year-month-day-hour-second-millisecond_PLC name.s7pbkp.




Restore name

Restore CPU data from a backup file. You can create backup files using the TIA Portal or the SIMATIC Automation Tool. Backup files have the extension name ".s7pbkp" that must exist or be copied to the backup file path assigned in the Tools>Options menu. The default file path is C:\ProgramData\Siemens\SIMATIC Automation Tool\Backup.

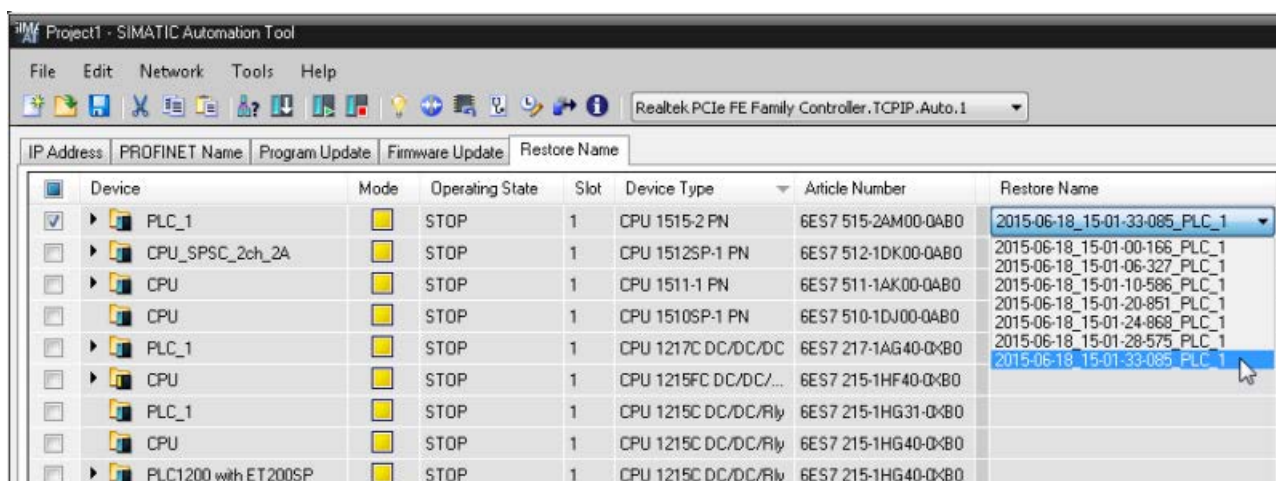
The cells in the columns "Restore" are used to enter the restore file name. You can create Backup files using the TIA Portal or the SIMATIC Automation Tool. These files have the extension of ".s7pbkp". The cells in the column "Restore" are used to enter the restore file name.

By default, the restore operation allows only a single processing thread that must complete execution, before the next device operation is executed. To change this behavior, see the Tool menu (Page 60) topic. Be aware of the risk of doing multi-thread firmware updates that is described in that topic.

Use the following steps to restore selected devices from a backup file.

1. Click the "Restore Name" tab.
2. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
3. For each device row selected, use the "Restore Name" column drop-down list and select a backup file name. The drop-down list will show the names of the .s7pbkp files that exist in the backup file path assigned in the Tools>Options menu.
4.  Select the Download Command from the Network menu, or click the Download toolbar button to start the operation.
5. The Event log below the device table shows the results of this operation.

In the following image, one CPU is selected, so only one backup file selection is required in the "Restore Name" column. If more than one device is selected, then you must repeat the backup file selection for each selected device.




3.9 Reset CPUs and modules to factory default values

Reset selected devices to factory default values

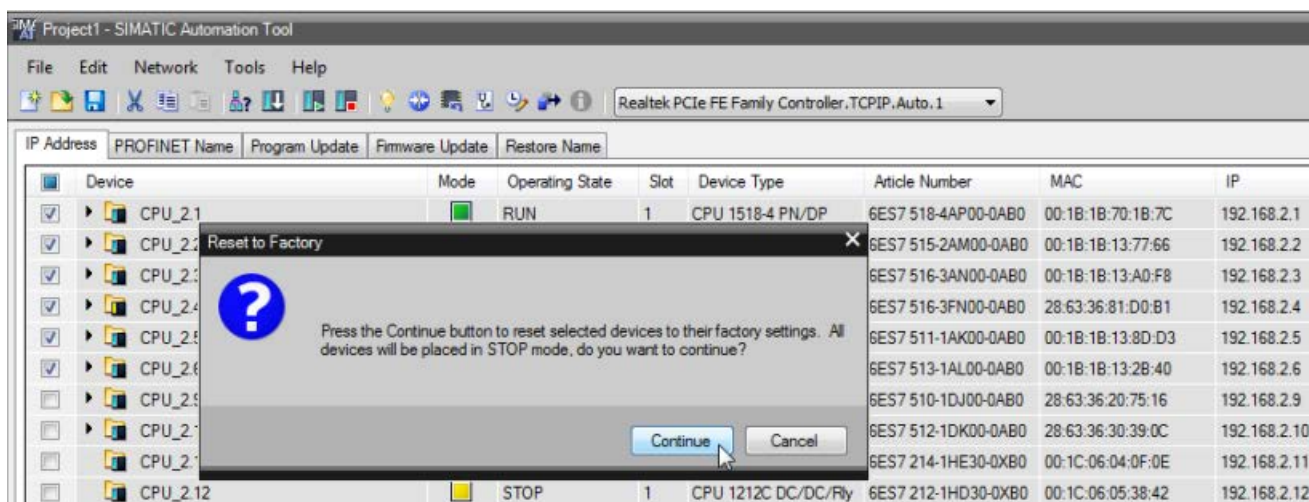
You can reset selected devices to factory default values, except for the IP address. The IP address that exists before the operation is retained, so your network IP assignments are preserved.

By default, the reset factory defaults operation allows only a single processing thread that must complete execution, before the next device operation is executed. To change this behavior, see the Tool menu (Page 60) topic. Be aware of the risk of doing multi-thread firmware updates that is described in that topic.

Use the following steps to reset selected devices to factory default values.

1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
2.  Select "Reset Factory Defaults" from the Network menu or click the "Reset Factory Defaults" toolbar button.
3. Click the "Continue" button on the "Reset to Factory" dialog box.
4. Selected devices are reset to factory default values.
5. The Event log below the device table shows the results of this operation.
6. Allow time for the reset to complete. Wait until the device lights stop flashing before attempting another operation.


The selected devices in the following image are reset to factory default values, when the Reset Factory Defaults command is executed.



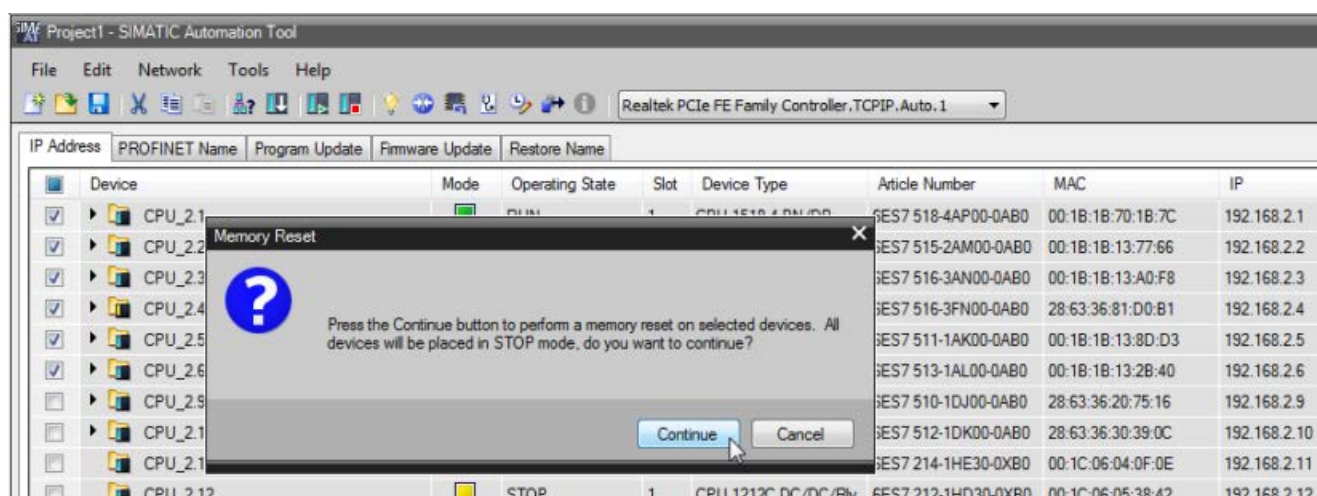
3.10 Reset CPU memory

Reset memory on selected CPUs

Use the following steps to reset CPU memory on selected devices.

1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
2.  Select "Memory Reset" from the Network menu or click the "Memory Reset" toolbar button.
3. Click the "Continue" button on the "Memory Reset" dialog box.
4. Selected CPUs perform a memory reset operation.
5. The Event log below the device table shows the results of this operation.

The selected devices in the following image perform a memory reset when the Memory Reset command is executed.




3.11 Upload service data from CPUs

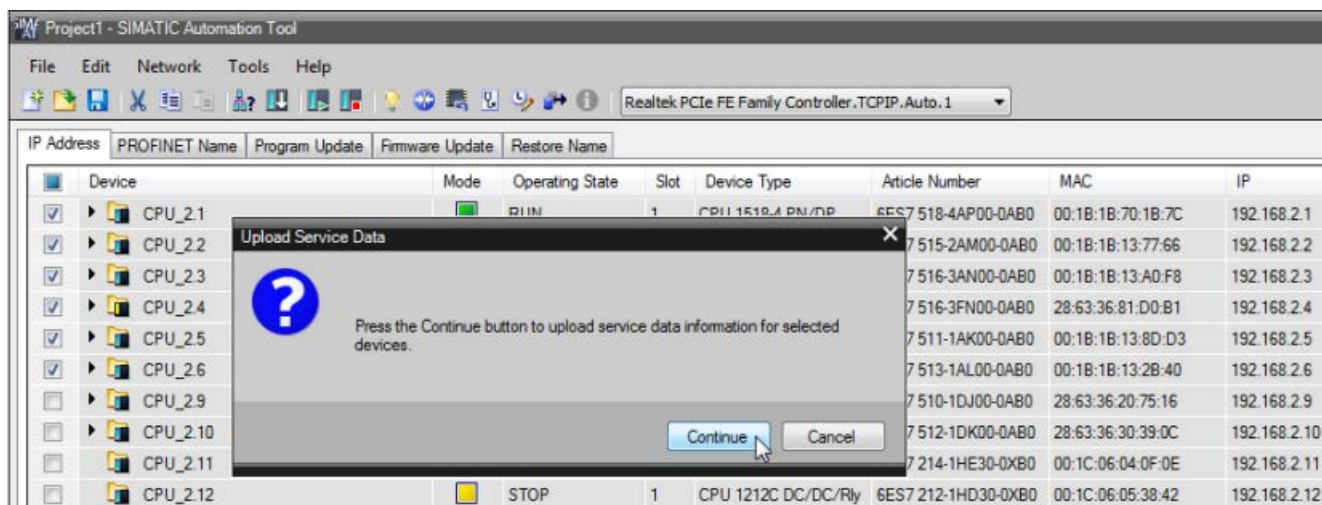
Get service data from selected CPUs

When a CPU enters a defective state, information is logged in the CPU that you can upload. You can send the service data files to Siemens customer support and help determine the cause of a fault. A folder is created under the assigned Fault folder (Default path is C:\ProgramData\Siemens\SIMATIC Automation Tool\Fault).

Use the following steps to upload service data from selected CPUs.

1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Unselect All" commands.
2.  Select "Upload Service data" from the Network menu or click the "Upload service data" toolbar button.
3. Click the "Continue" button on the "Upload Service Data" dialog box.
4. Selected CPUs upload service data to the file path assigned with the **Tools>options** menu.
5. The Event log below the device table shows the results of this operation.

The selected devices in the following image upload service data when the Upload Service Data command is executed.



Service data files

The name of the folder is based on the date, time, and PLC name similar to the TIA Portal.

Note

Uploading service data files from password protected CPUs

If a CPU is password protected, then you must provide the CPU's password to upload all the service data files. Enter CPU passwords in the SIMATIC Automation Tool's "Password" column, before the upload service data operation is executed.

Example service data file set:

2015-04-16_15-33-58_PLC_1\Fault.bin
2015-04-16_15-33-58_PLC_1\Fault2.bin (1200 V4.1 or later only).
2015-04-16_15-33-58_PLC_1\Alarms.txt
2015-04-16_15-33-58_PLC_1\ASLog.txt
2015-04-16_15-33-58_PLC_1\ASLog.txt
2015-04-16_15-33-58_PLC_1\DNN.txt
2015-04-16_15-33-58_PLC_1\PLCInformation.txt
2015-04-16_15-33-58_PLC_1\RAM.img (1200 only)
2015-04-16_15-33-58_PLC_1\NAND.img (1200 only)
2015-04-16_15-33-58_PLC_1\ResourceStats.txt (1200 only)
2015-04-16_15-33-58_PLC_1\CommBuffers.txt (1200 only)
2015-04-16_15-33-58_PLC_1\CommBuffers.txt (1200 only)

Note

Service data is stored in clear text

A malicious user could use the service data files to obtain status and configuration details about the control system. The service data files are stored in clear text on the CPU (binary encoding). A CPU password can control access to this information.

Use the TIA portal device configuration to set up CPU protection with a strong password. Strong passwords are at least eight characters in length, mixed letters, numbers, and special characters, are not words that can be found in a dictionary, and are not names or identifiers that can be derived from personal information. Keep the password secret and change it frequently.

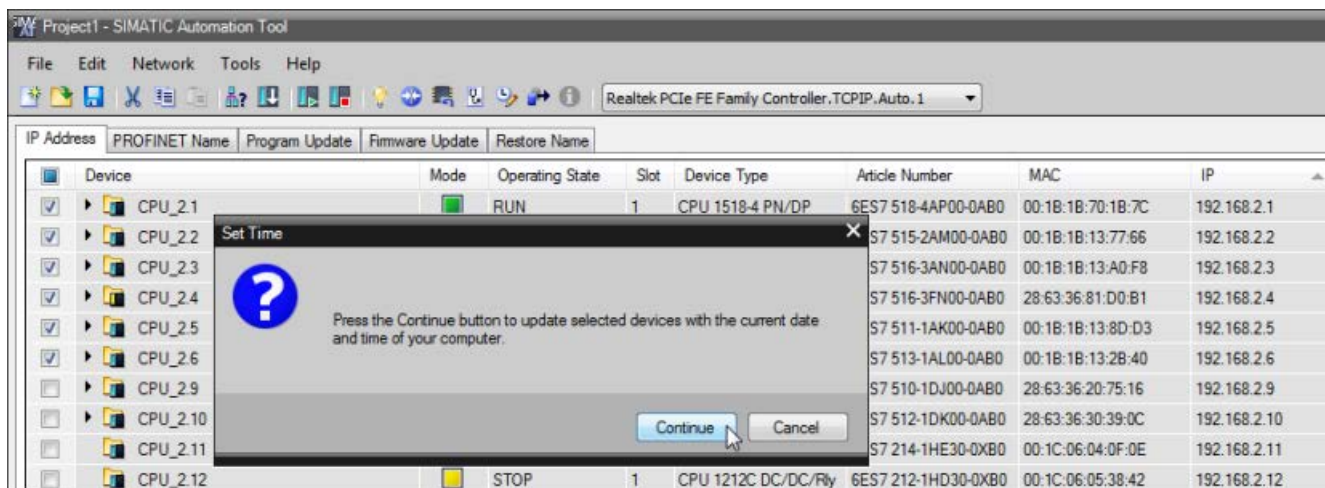
3.12 Set time in CPUs

Set time in CPUs to current PG/PC time

The Time button will set the time for selected CPUs to your current PG/PC time. Time transformation information for time zone and daylight saving time is not changed and must be modified in the TIA Portal Project.

1. Click the left-side check box on devices to include in the operation. You can use the top check box, right-click shortcut menu, or the Edit menu for "Select All" and "Un-select All" commands.
2. Select "Set Time" from the Network menu or click the "Set time in devices to PC time" toolbar button.
3. Click the "Continue" button on the "Set Time" dialog box.
4. The system time on selected CPUs is set to your current PG/PC time.
5. The Event log below the device table shows the results of this operation.


The selected devices in the following image set their time to your current PG/PC time when the Set Time command is executed.



3.13 Read diagnostic buffer in a CPU

CPU diagnostic buffer

The CPU diagnostics buffer contains an entry for each diagnostic event. Each entry includes the date and time the event occurred, an event category, and an event description. The entries are displayed in chronological order with the most recent event at the top. Up to 50 most recent events are available in this log. When the log is full, a new event replaces the oldest event in the log. When power is lost, the events are saved.

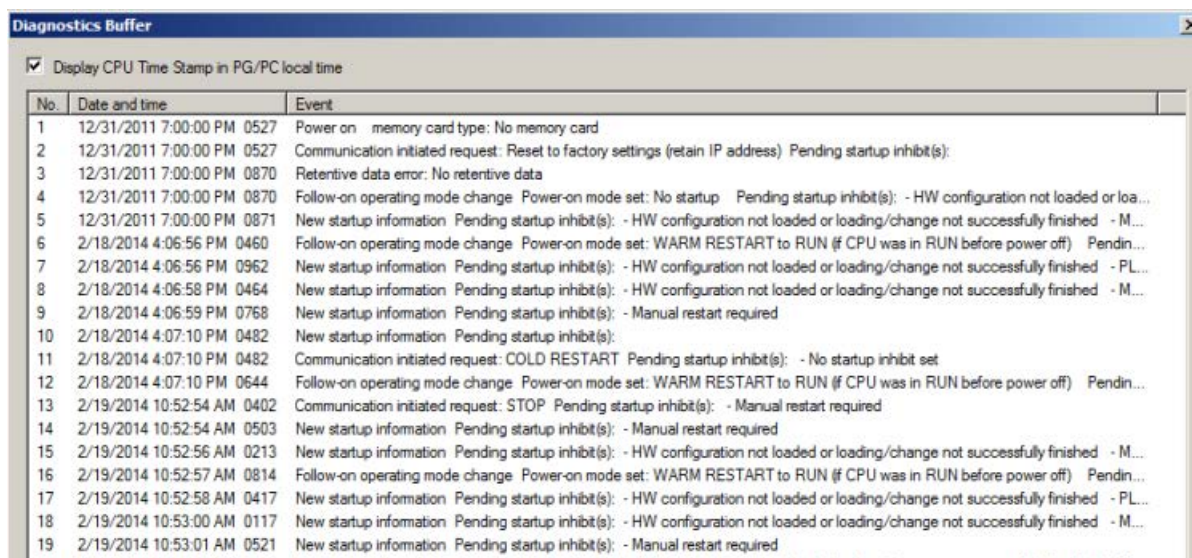
1. Click a check box in the "Select" column, for **one** CPU. You can use the right-click shortcut menu or the Edit menu, for the "Unselect All" command.
2.  Select "Diagnostics buffer" from the Network menu or click the "Diagnostics" button on the toolbar.
3. The SIMATIC Automation Tool will display the contents of a CPU's diagnostics log.

Example diagnostic log

The following types of events are recorded in the diagnostics buffer.

- System diagnostic event (each CPU error and module error)
- CPU state changes (each power up, each transition to STOP, each transition to RUN)

You can use the "Display CPU Time Stamp in PG/PC local time" check box to view time stamps in local time or UTC time (Coordinated Universal Time).



No.	Date and time	Event
1	12/31/2011 7:00:00 PM 0527	Power on memory card type: No memory card
2	12/31/2011 7:00:00 PM 0527	Communication initiated request: Reset to factory settings (retain IP address) Pending startup inhibit(s):
3	12/31/2011 7:00:00 PM 0870	Retentive data error: No retentive data
4	12/31/2011 7:00:00 PM 0870	Follow-on operating mode change Power-on mode set: No startup Pending startup inhibit(s): - HW configuration not loaded or loa...
5	12/31/2011 7:00:00 PM 0871	New startup information Pending startup inhibit(s): - HW configuration not loaded or loading/change not successfully finished - M...
6	2/18/2014 4:06:56 PM 0460	Follow-on operating mode change Power-on mode set: WARM RESTART to RUN (if CPU was in RUN before power off) Pendin...
7	2/18/2014 4:06:56 PM 0962	New startup information Pending startup inhibit(s): - HW configuration not loaded or loading/change not successfully finished - PL...
8	2/18/2014 4:06:58 PM 0464	New startup information Pending startup inhibit(s): - HW configuration not loaded or loading/change not successfully finished - M...
9	2/18/2014 4:06:59 PM 0768	New startup information Pending startup inhibit(s): - Manual restart required
10	2/18/2014 4:07:10 PM 0482	New startup information Pending startup inhibit(s):
11	2/18/2014 4:07:10 PM 0482	Communication initiated request: COLD RESTART Pending startup inhibit(s): - No startup inhibit set
12	2/18/2014 4:07:10 PM 0644	Follow-on operating mode change Power-on mode set: WARM RESTART to RUN (if CPU was in RUN before power off) Pendin...
13	2/19/2014 10:52:54 AM 0402	Communication initiated request: STOP Pending startup inhibit(s): - Manual restart required
14	2/19/2014 10:52:54 AM 0503	New startup information Pending startup inhibit(s): - Manual restart required
15	2/19/2014 10:52:56 AM 0213	New startup information Pending startup inhibit(s): - HW configuration not loaded or loading/change not successfully finished - M...
16	2/19/2014 10:52:57 AM 0814	Follow-on operating mode change Power-on mode set: WARM RESTART to RUN (if CPU was in RUN before power off) Pendin...
17	2/19/2014 10:52:58 AM 0417	New startup information Pending startup inhibit(s): - HW configuration not loaded or loading/change not successfully finished - PL...
18	2/19/2014 10:53:00 AM 0117	New startup information Pending startup inhibit(s): - HW configuration not loaded or loading/change not successfully finished - M...
19	2/19/2014 10:53:01 AM 0521	New startup information Pending startup inhibit(s): - Manual restart required

3.14 Execution order of operations

Operations are initiated with a toolbar button or menu item. For each toolbar button press, a single operation is added to the operations queue, for each selected device row. For example, if 20 different CPUs are selected and the RUN button is pressed, then 20 RUN operations are added to the queue.

For better performance, separate threads can run independently to initiate and execute the operations contained in the queue. The number of concurrent threads allowed is assigned in the Options dialog. Separate threads are not allowed to simultaneously start jobs on one CPU, to avoid race conditions where one job is putting the CPU in STOP and another job is placing the same CPU in RUN.

Execution examples

Example 1:

If the operations queue contains 10 go to RUN jobs for different CPUs, then multiple threads work in parallel to put all the CPUs in RUN mode. Since the threads execute in parallel, there is no guarantee of the order that CPUs complete the transition to RUN mode.

Communication speeds can be different and how fast the job completes can be different, for each CPU.

Example 2:

You can queue as many jobs of the same type as you want. For example, you can place 100 CPUs in STOP mode by selecting all 100 CPUs and clicking the STOP button. However, a dialog box with a progress bar is displayed until all 100 jobs are complete. This dialog box will block the start of another operation, until all the STOP operations are complete.

Saving your device table information

4.1 Save/Save as - Device table stored in encrypted .sat format

Use the **Save/Save as** commands or click the Save button to store your device table information in an encrypted .sat file. Once the SIMATIC Automation Tool project is saved, you can use the **File>Open** command to restore this project's device table.

- The .sat file save path is assigned by the **Tools>Options** command. The default path is C:\ProgramData\Siemens\SIMATIC Automation Tool\Projects. You can modify this path
- You must provide a valid password to save a SIMATIC Automation Tool .sat project file.
- You must enter the correct password to reopen an existing SIMATIC Automation Tool .sat project file.

SIMATIC Automation Tool .sat file security

Protect your SIMATIC Automation Tool project with a strong password. Strong passwords are at least ten characters in length, mixed letters, numbers, and special characters, are not words that can be found in a dictionary, and are not names or identifiers that can be derived from personal information. Keep the password secret and change it frequently.

SIMATIC Automation Tool password rules

- At least ten characters in length
- Mix of letters, numbers, and special characters is required

Integrity check for .sat file

Before opening a project, an internal checksum test verifies that the file data has not changed, since the last SIMATIC Automation Tool save operation.

4.2 Import/Export - Device table loaded from/stored in open .csv format

- The **File>Export** menu command saves the device table in .csv (comma separated values) text format.
- The **File>Import** menu command reads a .csv text file and puts that data in the SIMATIC Automation Tool device table.

The first text line is a description header followed by one or more data lines. Data text must match the expected format, with eleven "," comma characters on each line of text. Eleven comma characters separate the twelve data columns that you see in the export example.

The device table in the SIMATIC Automation Tool configures communication with a device group. If you put incorrect information in the cells of a device table or in an imported .csv file, then the affected device operation can fail. Correct the device data and try the operation again.

Note

CPU passwords are not exported

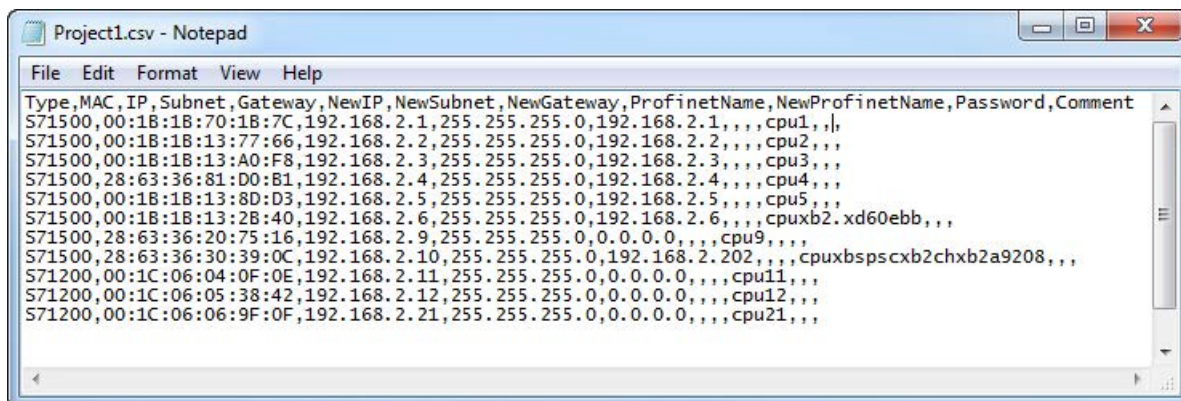
When you **export** a device table the tenth .csv file column (Password) is empty for security reasons.

You can edit a .csv file, add in the passwords, and then **import** the .csv file. The passwords then appear in the SIMATIC Automation Tool device table.

The .csv file path for import and export operations is assigned by the **Tools>Options** command. The default path is C:\ProgramData\Siemens\SIMATIC Automation Tool\Projects. You can modify the default path.

Export example

The following image shows the text format of a .csv file exported from the SIMATIC Automation Tool.



4.2 Import/Export - Device table loaded from/stored in open .csv format






The following image shows the same text file opened in Microsoft Excel.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Type	MAC	IP	Subnet	Gateway	NewIP	NewSubnet	NewGateway	ProfinetName	NewProfinetName	Password	Comment
2	S71500	00:1B:1B:70:1B:7C	192.168.2.1	255.255.255.0	192.168.2.1				cpu1			
3	S71500	00:1B:1B:13:77:66	192.168.2.2	255.255.255.0	192.168.2.2				cpu2			
4	S71500	00:1B:1B:13:A0:F8	192.168.2.3	255.255.255.0	192.168.2.3				cpu3			
5	S71500	28:63:36:81:D0:B1	192.168.2.4	255.255.255.0	192.168.2.4				cpu4			
6	S71500	00:1B:1B:13:8D:D3	192.168.2.5	255.255.255.0	192.168.2.5				cpu5			
7	S71500	00:1B:1B:13:2B:40	192.168.2.6	255.255.255.0	192.168.2.6				cpuxb2.xd60ebb			
8	S71500	28:63:36:20:75:16	192.168.2.9	255.255.255.0	0.0.0.0				cpu9			
9	S71500	28:63:36:30:39:0C	192.168.2.10	255.255.255.0	192.168.2.202				cpuxbspscxb2chxb2a9208			
10	S71200	00:1C:06:04:0F:0E	192.168.2.11	255.255.255.0	0.0.0.0				cpu11			
11	S71200	00:1C:06:05:38:42	192.168.2.12	255.255.255.0	0.0.0.0				cpu12			
12	S71200	00:1C:06:06:9F:0F	192.168.2.21	255.255.255.0	0.0.0.0				cpu21			

Menu, toolbar, and shortcut key reference

5.1 Main menu

5.1.1 File menu






Tool icon	Menu command	Description
	New	Creates a new SIMATIC Automation Tool project.
	Open	An "Open" dialog is displayed that can browse to a folder, select an .sat project file, and provide a password to open an encrypted project file. The default path is assigned in the Tools>Options dialog
	Save	The device table data is saved in a .sat file. If no filename and password are assigned, then this operation uses the "Save As" command.
	Save As...	The device table data is saved in a .sat file. You can browse to a folder, assign a .sat project filename, and assign a password to protect the project file. The default path is assigned in the Tools>Options dialog.
	Import...	Fill the device table with data from a file in .csv format.
	Export...	Save the device table data to a file in .csv format.
	Exit	Close the application. If the project was modified since the last save operation, then the "Save" operation is performed

See also












Save/Save as - Device table stored in encrypted .sat format (Page 53)

Import/Export - Device table loaded from/stored in open .csv format (Page 54)


5.1.2 Edit menu

Tool icon	Menu command	Description
	Cut	Cut the selected data and copy this data to the clipboard. Clipboard entries are compatible with Excel, so data can be shared between the two applications. Read-only cells are not deleted.
	Copy	Copy the selected data to the clipboard in Excel compatible format.
	Paste	Paste the data contained in the clipboard to selected field(s) in the SIMATIC Automation Tool. Read-only cells are not modified.
	Check All	Check (select) all rows of data on the visible tab.
	Uncheck All	Uncheck all rows on the visible tab.
	Expand All	Expand all rows for devices and modules.
	Collapse All	Collapse the rows for devices and modules.
	Insert Device	Insert a new device row at the selected row and push the following device rows downward. When a device cannot be discovered by a network scan, you can use this command to add the device to the device table. If you use this command to insert a device, the device name is colored blue. The blue color means that the MAC address based operations (flash LEDs, set IP address, and set PROFINET name) are not possible and the corresponding Device table cells are disabled.
	Delete	
	• Device	Delete one or more checked device rows.
	• Selection	Delete current selection in the device table.
	Refresh Device(s) F5	Refresh the checked devices.
	Check for Updates	Open the Siemens support Internet web page for the selected device.

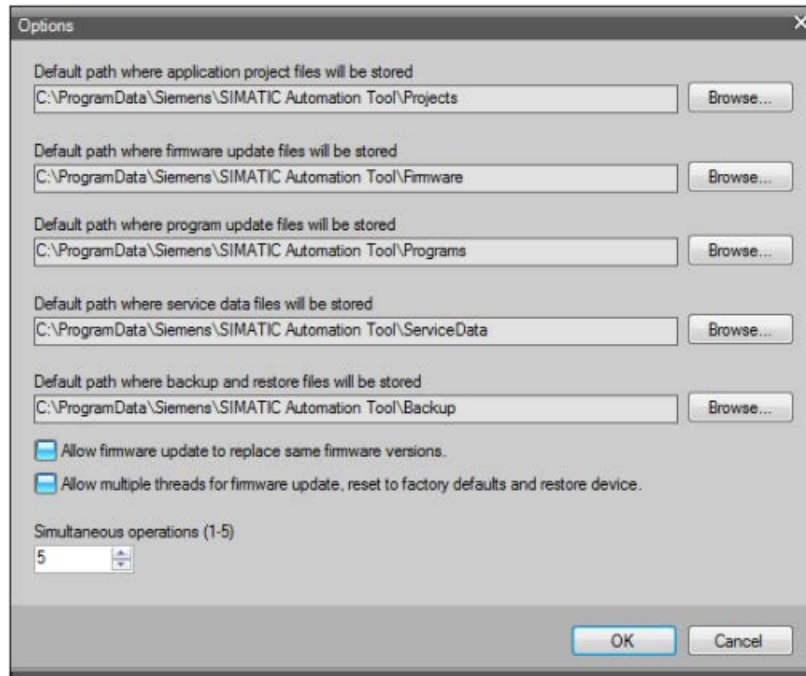
5.1.3 Network menu

Tool icon	Menu command	Description
	Scan	Scan the selected network interface for accessible CPUs and modules.
	Download	Download data entered in the SIMATIC Automation Tool to PLCs on the network. The type of download depends on the current tab selection, Download types: <ul style="list-style-type: none"> • IP address parameters • PROFINET name • Program update • Firmware update • Restore device data from named .s7pbkp backup file
	RUN	Put selected CPUs in RUN mode.
	STOP	Put selected CPUs in STOP mode.
	Flash LEDs	Flash the LEDs on selected devices. Use this feature to identify the physical location of a device.
	Reset to factory defaults	Perform reset to factory default values on selected devices.
	Memory reset	Perform a memory reset on selected CPUs.
	Upload service data	Upload service information from CPUs.
	Set time	Set time in selected CPUs to your PG/PC time.
	Backup	Perform a backup operation on selected CPUs of all CPU data.
	Diagnostics buffer	Read the diagnostics log from one CPU.

5.1.4 Tools menu

Tool icon	Menu command	Description
	Options	<p>Set file paths for the SIMATIC Automation Tool</p> <ul style="list-style-type: none"> • Path for SIMATIC Automation Tool project files • Path for CPU firmware update files • Path for program update files • Path to service data files • Path for Backup/Restore files <p>Allow firmware update to replace same firmware version The default is disabled to save time by preventing an unnecessary operation. You can force an update by enabling this option.</p> <p>Allow multiple threads for firmware update, reset to factory defaults, and restore device If your network has a star topology where each CPU has a separate connection to the PG/PC through an Ethernet switch, then you can enable the multiple threads option. If your network has a chain topology, you should disable this option to prevent one CPU from disrupting the communication to another CPU. For example, you have a chain connection (PG/PC to CPUa to CPUb to CPUc to ...). An ongoing operation for CPUb is disrupted when a separate thread causes a restart of CPUa.</p> <p>Set simultaneous operations limit Limit the number of communication connections and processing threads. Your selected device group can simultaneously process communication and operation requests coming from the SIMATIC Automation Tool. A maximum of 5 simultaneous operations is allowed.</p>

Default options




Note

Communication problems with the SIMATIC Automation Tool


















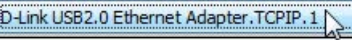
For example, you send an operation command to multiple devices, but a device does not complete the operation and a communication error displayed for that CPU. However, other devices are communicating and executing the operation as expected.

If you experience communication problems, then reduce the maximum number of simultaneous operations (threads/connections) that is assigned in the **Tools > Options** dialog box. Close and restart the SIMATIC Automation Tool application, then try the group operation again.

5.1.5 Help menu

Tool icon	Menu command	Description
	View user guide	<p>Open the SIMATIC Automation Tool user guide. The user guide is in .pdf format and you must have .pdf viewer software installed to view the user guide.</p> <p>English and German versions of the user guide are installed. The version you see depends on the regional settings for language and location, in your Windows operating system.</p> <p>The German user guide is displayed for these culture settings:</p> <ul style="list-style-type: none"> • de German • de-AT German (Austria) • de-DE German (Germany) • de-CH German (Switzerland) • de-LI German (Liechtenstein) • de-LU German (Luxembourg) <p>The English user guide is displayed in all other cases.</p>
	About	<p>Displays the About dialog that contains:</p> <ul style="list-style-type: none"> • Product name • Version

5.2 Toolbar icons

Tool icon	Description
	New: Create a new SIMATIC Automation Tool project file with the ".sat" file name extension.
	Open: Display an "Open" dialog that can browse to a folder, select a project file, and provide a password to open the encrypted project file.
	Save the opened project data to a file. If no filename and password are assigned, then the "Save As" dialog is displayed.
	Cut the selected data and copy the data to the clipboard. Clipboard data are compatible with Excel so data can be shared between the two applications.
	Copy the selected data to the clipboard.
	Paste the data contained in the clipboard to the selected field(s).
	Scan the selected network interface for accessible CPUs and modules.
	Download data entered in the SIMATIC Automation Tool to devices on the network. Depending on the current tab selection, either IP addresses, PROFINET names, program updates, firmware updates, or restore data from a backup file are downloaded.
	RUN: Put selected CPUs in RUN mode.
	STOP: Put selected CPUs in STOP mode.
	Flash the LEDs on selected devices. Use this feature to identify the physical location of a device.
	Reset factory default values in selected CPUs.
	Memory reset: Reset the memory on selected devices.
	Upload service data: Upload service information from a CPU.
	Set time: Set the system time in selected CPUs to current PG/PC time.
	Backup Device: Create .Backup data Perform a data backup operation.
	Diagnostics: Read a CPU diagnostic buffer
	Network interface drop-down list: Select the Ethernet network interface that is connected to the industrial control network.

5.3 Shortcut keys

CTRL+PgUp	Switches between tabs, from left to right
CTRL+PgDn	Switches between tabs, from right to left
CTRL+A	Selects the entire table
CTRL+C	Copies the selected cells
CTRL+O	Displays the project open dialog to open a new project file
CTRL+S	Displays the Save As dialog
CTRL+V	Pastes the contents of the clipboard at the insertion point and replaces any selection
CTRL+X	Cuts the selected cells
CTRL+Z	Undo the last edit or delete action
ARROW KEYS	Move one cell up, down, left or right
SHIFT+ARROW KEYS	Extends the selection of cells
DELETE	Removes the contents of the active cell
ENTER	Completes cell editing and validates data
ESC	Cancels cell editing restoring the cell to original value
HOME	Moves to the beginning of a row
CTRL+HOME	Moves to the beginning of the table
END	Moves to the end of a row
CTRL+END	Moves to the end of the table
PAGE DOWN	Moves one screen down in the table
PAGE UP	Moves one screen up in the table
SPACEBAR	Selects or clears the rows checkbox, or multiple rows, if selected
TAB	Moves one cell to the right

SIMATIC Automation Tool API for .NET framework

6.1 API classes and methods overview

Interface .dll file (dynamic link library)

The API interface consists of one .dll file.

`AutomationToolAPI.dll`

The .dll file was created with Microsoft Visual Studio 2010 and .NET Framework version 4.0.

.NET classes

The API consists of two main classes.

- **Siemens.Automation.AutomationTool.API.NetworkAPI** contains these methods that access a device using a MAC address:
 - NetworkAPI constructor (Page 67)
 - ScanNetworkInterfaceCards (Page 67)
 - SetNetworkInterface (Page 68)
 - ScanNetworkDevices (Page 68)
 - FlashLED (Page 69)
 - SetIP (Page 69)
 - SetProfinetName (Page 70)
 - GetDeviceType (Page 71)
 - ResetToFactory (Page 72)

- Siemens.Automation.AutomationTool.API.**DeviceAPI** contains these methods that access a device using an IP address:
 - DeviceAPI constructor (Page 73)
 - StartCommunications and StopCommunications (Page 74)
 - Connect (Page 75)
 - Disconnect (Page 75)
 - SetMode (Page 76)
 - GetMode (Page 77)
 - ResetToFactory (Page 78)
 - MemoryReset (Page 79)
 - DownloadFirmware (Page 80)
 - DownloadProgram (Page 85)
 - UploadServiceData (Page 86)
 - GetStationName (Page 86)
 - GetPLCName (Page 87)
 - SetTime (Page 88)
 - GetTime (Page 88)
 - GetDeviceInformation (Page 89)
 - GetRackInformation (Page 90)
 - Backup (Page 91)
 - Restore (Page 92)
 - GetDiagnosticsBuffer (Page 93)

6.2 NetworkAPI methods

6.2.1 NetworkAPI constructor

NetworkAPI constructor

The NetworkAPI constructor requires no parameters.

```
using Siemens.Automation.AutomationTool.API;  
  
NetworkAPI network = new NetworkAPI();
```

6.2.2 ScanNetworkInterfaceCards

Scan for network interface cards in a PG or PC

The "ScanNetworkInterfaceCards" method gets the NIC strings of accessible network interface cards. A NIC string is required for calling other methods in this class.

```
using Siemens.Automation.AutomationTool.API;  
  
NetworkAPI network = new NetworkAPI();  
List<String> aInterfaces = new List<String>();  
  
Error error = network.ScanNetworkInterfaceCards(aInterfaces);  
if (error.Succeeded)  
{  
    for (Int32 i = 0; i < aInterfaces.Count; i++)  
    {  
        String strNIC = aInterfaces[i];  
    }  
}
```

6.2.3 SetNetworkInterface

Set network interface card

The "SetNetworkInterface" method is used to change the access point used for IP communications. All DeviceAPI communications (IP address based operations) use the network card that is assigned by this method, Use this method to set the access path of the application instead of opening and changing the PG/PC Interface dialog. This method only needs to be called one time or when you change network cards.

```
using Siemens.Automation.AutomationTool.API;

NetworkAPI network = new NetworkAPI();

Error error = network.SetNetworkInterface("NIC Name");
if (error.Succeeded)
{
}
```

6.2.4 ScanNetworkDevices

Scan for CPUs and modules

The "ScanNetworkDevices" method identifies all accessible devices, on the current network.

```
using Siemens.Automation.AutomationTool.API;

NetworkAPI network = new NetworkAPI();

Error error = network.SetNetworkInterface("NIC Name");
if (error.Succeeded)
{
    List<ScanDevice> aDevices = new List<ScanDevice>();

    error = network.ScanNetworkDevices(aDevices);
    if (error.Succeeded)
    {
        for (int i = 0; i < aDevices.Count; i++)
        {
            DeviceType type = aDevices[i].Type;
            UInt64 nMAC = aDevices[i].MAC;
            UInt32 nIP = aDevices[i].IP;
            UInt32 nSubnet = aDevices[i].Subnet;
            UInt32 Gateway = aDevices[i].Gateway;
        }
    }
}
```

6.2.5 FlashLED

Flash the LEDs on a device

The "FlashLED" method flashes the LED lights of a device, on the current network.

```
using Siemens.Automation.AutomationTool.API;

NetworkAPI network = new NetworkAPI();

Error error = network.SetNetworkInterface("NIC Name");
if (error.Succeeded)
{
    error = network.FlashLED("11:22:33:44:55:66");
    // or
    error = network.FlashLED(0x112233445566);
}
```

6.2.6 SetIP

Set IP addresses on a device

The "SetIP" method is used to set or change an IP address of a device on the current network.

The TIA portal program running in a CPU must use the PROFINET port option "Set IP address on the device" or "Set IP by other means". If a CPU's TIA portal program has the default IP protocol configuration of "Set IP address in the project", then the SetIP method will fail.

```
using Siemens.Automation.AutomationTool.API;

NetworkAPI network = new NetworkAPI();

Error error = network.SetIP("11:22:33:44:55:66", "192.168.2.11",
    "255.255.255.0", "0.0.0.0");
    // or
error = network.SetIP(0x112233445566, 0xC0A8020A, 0xFFFFFFFF00,
    0x00000000);
```

6.2.7 SetProfinetName

Set PROFINET name on a device

The "SetProfinetName" method is used to set or change a PROFINET device name on the current network. The current running TIA portal program must have the Ethernet address option "Set PROFINET device name on the device" enabled.

```
using Siemens.Automation.AutomationTool.API;

NetworkAPI network = new NetworkAPI();

Error error = network.SetProfinetName("11:22:33:44:55:66",
    "mystationname");
//or
error = network.SetProfinetName(0x112233445566, "mystationname");
```

6.2.8 GetDeviceType

Get the device type of a PROFINET device

The "GetDeviceType" method gets the device type using only a MAC address.

These device types are supported: S7_1200, S7_1500, ET200SP, ET200MP, ET200AL, ET200pro, ET200eco, ET200s, and ET200m

```
using Siemens.Automation.AutomationTool.API;

NetworkAPI network = new NetworkAPI();

Error error = network.SetNetworkInterface("NIC Name");
if (error.Succeeded)
{
    DeviceType deviceType = new DeviceType();

    error = network.GetDeviceType("11:22:33:44:55:66", ref
deviceType);
    // or
    error = network.GetDeviceType(0x112233445566, ref deviceType);
    if (error.Succeeded)
    {
        if (deviceType == DeviceType.S7_1200)
            Call1200();
        if (deviceType == DeviceType.S7_1500)
            Call1500();
        if (deviceType == DeviceType.ET200SP)
            CallET200SP();
    }
}

void Call1200()
{
}

void Call1500()
{
}

void CallET200SP()
{
}
```

6.2.9 ResetToFactory (NetworkAPI)

Reset device to factory default values

The "ResetToFactory" method resets a PROFINET I/O device to the factory installed settings and default values.

Note

The NetworkAPI ResetToFactory method is only works with directly connected PROFINET I/O devices. If you attempt to execute this method with a directly connected CPU, the operation will fail.

For indirectly connected PROFINET I/O devices (behind a CPU), only the firmware update operation is possible.

You must use the DeviceAPI ResetToFactory method with directly connected CPUs.

```
using Siemens.Automation.AutomationTool.API;

NetworkAPI network = new NetworkAPI();

Error error = network.SetNetworkInterface("NIC Name");
if (error.Succeeded)
{
    error = network.ResetToFactory("11:22:33:44:55:66");
    // or
    error = network.ResetToFactory(0x112233445566);
}
```


6.3 DeviceAPI methods

6.3.1 DeviceAPI constructor

DeviceAPI constructor

The "DeviceAPI" constructor requires the valid enum of a supported device type.

These device types are supported: S7_1200, S7_1500, ET200SP, ET200MP, ET200AL, ET200pro, ET200eco, ET200s, and ET200m

```
using Siemens.Automation.AutomationTool.API;
```

```
DeviceAPI device1 = new DeviceAPI(DeviceType.S7_1200);  
DeviceAPI device2 = new DeviceAPI(DeviceType.S7_1500);  
DeviceAPI device3 = new DeviceAPI(DeviceType.ET200SP);
```

6.3.2 StartCommunications and StopCommunications

Start and stop communications

The "**StartCommunications**" method must be called one time only (to initialize communications) before any IP address based operations are performed.

The "**StopCommunications**" method should be called only once to release communication resources allocated by the "StartCommunications" method.

If StartCommunications is not called first, then the Connect method will not execute correctly.

These two methods are used together, as shown in the following examples. StartCommunications and StopCommunications are included with each code example, so the examples can compile and execute separately. Normally, you should call StartCommunications and StopCommunications only once, for one instance of your application.

```
using Siemens.Automation.AutomationTool.API;

// Initialize communications parameters
//(should only be called once per application instance)
DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    //Put one or more of your tasks here.

    device.Disconnect();
}

// Release communications resources allocated by
// "StartCommunications" method
DeviceAPI.StopCommunications();
```

6.3.3 Connect

Connect to a CPU

The "Connect" method establishes a connection with a CPU. An IP address and password are required. If the CPU has no password, then send an empty string.

```
using Siemens.Automation.AutomationTool.API;
DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);
Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
//or
    error = device.Connect(0xC0A8020A, new
EncryptedString("MyPassword"));

if (error.Succeeded)
    error = device.Disconnect();
DeviceAPI.StopCommunications();
```

6.3.4 Disconnect

Disconnect from a CPU

The "Disconnect" method breaks a previously successful connection. A limited number of connections can exist at one time.

The Connect and Disconnect methods are used together. If Disconnect is not executed after a successful Connect execution, then you will get an exception error when your application is closed.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));

if (error.Succeeded)
    error = device.Disconnect();

DeviceAPI.StopCommunications();
```

6.3.5 SetMode

Set the operating mode of a CPU

Set a CPU to RUN mode

The "SetMode" method can put a CPU in RUN mode.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    // Change CPU to RUN mode
    error = device.SetMode(AS_OperatingStateREQ.Run_REQ);

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

Set a CPU to STOP mode

The "SetMode" method can put a CPU in STOP mode.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    // Change CPU to STOP mode
    error = device.SetMode(AS_OperatingStateREQ.Stop_REQ);

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.6 GetMode

Get the operating mode of a CPU

The "GetMode" method gets the current mode of a CPU (RUN mode or STOP mode).

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    AS_OperatingState nState = AS_OperatingState.NotSupported;

    error = device.GetMode(ref nState);
    if (error.Succeeded)
    {
        bool bRunMode = (nState == AS_OperatingState.Run);
        bool bStopMode = (nState == AS_OperatingState.Stop);
    }
    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.7 ResetToFactory (DeviceAPI)

Reset a device to factory default values

The "ResetToFactory" method can reset device values to the factory default values.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);
device.ResetProcessed += new ResetDeviceHandler(ResetProcessed);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    // Reset device to Factory defaults
    error = device.ResetToFactory();

    device.Disconnect();
}

DeviceAPI.StopCommunications();
device.ResetProcessed -= new ResetDeviceHandler(ResetProcessed);

void ResetProcessed(object sender, ResetEventArgs e)
{
}
```

6.3.8 MemoryReset

Reset CPU memory

The "MemoryReset" method resets CPU memory.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
device.ResetProcessed += new ResetDeviceHandler(ResetProcessed);

if (error.Succeeded)
{
    // Reset CPU memory
    error = device.MemoryReset();

    device.Disconnect();
}

DeviceAPI.StopCommunications();

device.ResetProcessed -= new ResetDeviceHandler(ResetProcessed);

void ResetProcessed(object sender, ResetEventArgs e)
{
}
```

6.3.9 DownloadFirmware

Download new firmware to a device

- Verify in the SIMATIC Automation Tool device support (Page 95) chapter that you are using a model and version which supports the firmware update operation.
- The "DownloadFirmware" method can upgrade the firmware in a CPU or module.
- You must provide the path and name to a valid update file (an .upd file).

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();
DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

// Progress notification events
device.BlockDownloaded += new
BlockDownloadedHandler(DownloadFirmwareProgress);
device.FirmwareUpdated += new
FirmwareUpdatedHandler(FirmwareUpdated);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    // 1st parameter is the file path of the update file
    // 2nd parameter callback for progress.
    // 3rd parameter allows the same version of firmware
    //    to be replaced.
    // 4th parameter is the module ID returned from
    //    GetRackInformation method.

    // Download new version of the firmware
    error = device.DownloadFirmware(@"C:\6ES7 XXXXXX V04.00.00.upd",
this, true);
    // or
    error = device.DownloadFirmware(@"C:\6ES7 XXXXXX V04.00.00.upd",
this, true, 123);

    device.Disconnect();
}
device.BlockDownloaded -= new
BlockDownloadedHandler(DownloadFirmwareProgress);
device.FirmwareUpdated -= new
FirmwareUpdatedHandler(FirmwareUpdated);

DeviceAPI.StopCommunications();

void DownloadFirmwareProgress(object sender,
DownloadFirmwareEventArgs e)
{
    Int32 index = e.Index;
    Int32 count = e.Count;
```



```
        double percentComplete = 1.0f * index / count;
    }
    void FirmwareUpdated(object sender, FirmwareUpdatedEventArgs e)
    {
    }
}
```

Example of how to obtain a module's ID parameter

You must use the unique ID of a module, if you only want to update the firmware of that specific module. If the module ID is not provided, then the firmware of all accessible modules with a matching article number are updated. This ID must be obtained each time, before a call is made to the DownloadFirmware method.

If the module ID value passed to the API is 0 or does not exist, an error is returned stating "The module ID is not valid".

6.3 DeviceAPI methods

The following image shows devices connected to CPU_2.1 with the branch array index values and the example module (slot 10) colored red.

IP Address	PROFINET Name	Program Update	Firmware Update	Restore Name	
<input type="checkbox"/>	Device		Mode	Operating State	Slot
<input type="checkbox"/>	▼ CPU_2.1			STOP	1
<input type="checkbox"/>	CPU display_1				1.3
<input type="checkbox"/>	DI 32x24VDC HF_1				2
<input type="checkbox"/>	DQ 16x24VDC/0.5A ST_1				3
<input type="checkbox"/>	DQ 16x24VDC/0.5A ST_2				4
<input type="checkbox"/>	▼ DP-Mastersystem : PROFIBUS_2		RemoteInterface[0]		
<input type="checkbox"/>	▶ et200-10				0
<input type="checkbox"/>	▼ et200-11				0
<input type="checkbox"/>	DQ 8x24VDC/0.5A ST_1				1
<input type="checkbox"/>	DQ 8x24VDC/0.5A ST_2				2
<input type="checkbox"/>	DQ 8x24VDC/0.5A ST_3				3
<input type="checkbox"/>	DQ 8x24VDC/0.5A ST_4				4
<input type="checkbox"/>	Server module_1				5
<input type="checkbox"/>	▼ PROFINET IO-System : PN/IE_1		RemoteInterface[1]		
<input type="checkbox"/>	▶ IO device_121				0
<input type="checkbox"/>	▶ IO device_122				0
<input checked="" type="checkbox"/>	▼ IO device_171		RemoteDevices[3]		1
<input type="checkbox"/>	DI 16x24VDC HF_1				2
<input type="checkbox"/>	DI 16x24VDC HF_2				3
<input type="checkbox"/>	DI 16x24VDC HF_3				4
<input type="checkbox"/>	DI 16x24VDC HF_4				5
<input type="checkbox"/>	DQ 32x24VDC/0.5A ST_1				6
<input type="checkbox"/>	DQ 32x24VDC/0.5A ST_2				7
<input type="checkbox"/>	DQ 32x24VDC/0.5A ST_3				8
<input type="checkbox"/>	DQ 8x24VDC/2A HF_1				9
<input checked="" type="checkbox"/>	DQ 8x24VDC/2A HF_2		RemoteDevices.Modules[9]		10
<input type="checkbox"/>	▶ CPU_2.2			RUN	1
<input type="checkbox"/>	▶ CPU_2.3			RUN	1

You must call the GetRackInformation method to get the module data. The following image shows all the rack information that is available for the example module.

We want to perform a firmware update on the module with ID value 0x88e1012f located in slot 10 (1-based value) or 9 (0-based value).

ArticleNumber	Q - "6ES7 522-1BF00-0AB0"
BackupRestoreAllowed	false
Comment	Q - ""
Configured	true
Description	Q - "Digital output module DQ8 x 24VDC / 2A; grouping 4; 8 A per group;
Failsafe	false
FirmwareUpdateAllowed	true
FirmwareVersion	Q - "V02.00.00_00.00.00.00"
FirmwareVersionArray	{byte[0x00000004]}
HardwareVersion	0x0001
ID	0x88e1012f
ModuleType	Q - "DQ 8x24VDC/2A HF"
Name	Q - "DQ 8x24VDC/2A HF_2"
NewFirmwareVersion	Q - ""
PasswordAllowed	true
ProgramUpdateAllowed	false
Select	false
SerialNumber	Q - "S C-D4S655352013"
Slot	0x0000000a
SlotName	Q - "10"
Station	0x00000003
Status	0x00000000
SubSlot	0x00000001
Supported	true
Non-Public members	

The following C# code example shows the GetRackInformation call preceding the DownloadFirmware call, for the example module.

```
public void API_IP_2_1_FirmwareUpdate_IO_device_171_Slot10_V2_0_0()
{
    StartCommunications();

    Error error;
    RackInfo info = null;
    DeviceAPI device = new DeviceAPI(DeviceType.S7_1500);
    string strResults = string.Empty;

    error = device.Connect("192.168.2.1", new EncryptedString(""));
    if (error.Succeeded)
    {
        // Get Rack information
        error = device.GetRackInformation(ref info);
        if (error.Succeeded)
        {
            RemoteInterface interFace = info.RemoteInterfaces[1];
            RemoteDevice remoteDevice = interFace.RemoteDevices[3];
            Siemens.Automation.AutomationTool.API.Module module =
remoteDevice.Modules[9];

            // Use module.ID Slot 10 - DQ 8x24VDC/2A HF 6ES7 522-
1BF00-0AB0
            error =
device.DownloadFirmware(@"C:\ProgramData\Siemens\SIMATIC Automation
Tool\Firmware\6ES7 522-1BF00-0AB0 V02.00.00.upd",
"IO_device_171_Slot10", true, module.ID);
            if (error.Succeeded)
            {
            }
            else
                strResults += error.ErrorDescription.ToString() + "
DownloadFirmware " + "\r\n";
        }
        else
            strResults += error.ErrorDescription.ToString() + "
GetRackInformation " + "\r\n";

        device.Disconnect();
    }
    else
        strResults += error.ErrorDescription.ToString() + " Connect
" + "\r\n";

    StopCommunications();
}
```

6.3.10 DownloadProgram

Download a new program to a CPU

- The "DownloadProgram" method loads a new program in a CPU.
- A TIA portal program must be transferred by the TIA portal software to a SIMATIC memory card, USB flash drive, or another partition of your PG/PC hard drive before the program is usable with the SIMATIC Automation Tool. Refer to the TIA portal documentation about how to transfer a program to a storage device.
- After the TIA portal transfers the program to a storage device, you can use the Windows Explorer to transfer the program to the folder that is used by the SIMATIC Automation Tool. The SIMATIC.S7S folder and the files within are the program data. You must provide the path to a valid program.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    // Download program from this folder
    error = device.DownloadProgram(@"C:\ProgramCard");

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.11 UploadServiceData

Upload service data from a CPU

The "UploadServiceData" method can upload service data information from a defective CPU. A parameter assigns the path where the service data files are stored.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    // Upload service data from a CPU
    error = device.UploadServiceData(@"C:\FaultFolder");

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.12 GetStationName

Get the station name from a CPU

The "GetStationName" method gets the station name of the program in a CPU. You must assign a string that returns the name of the station.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    String strName = "";

    // Get station name
    error = device.GetStationName(ref strName);

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.13 GetPLCName

Get the PLC name from a CPU

The "GetPLCName" method gets the PLC name for the program loaded in a CPU. You must assign a string that returns the name of the PLC.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    String strName = "";

    // Get PLC name
    error = device.GetPLCName(ref strName);

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.14 SetTime

Set the time in a CPU

The "SetTime" method can set the current time in a CPU. The time transformation rules are not modified. Time zone and Daylight Saving Time properties are set with the TIA portal CPU device configuration.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    // Set to PC time
    error = device.SetTime(DateTime.Now);

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.15 GetTime

Get the time in a CPU

The "GetTime" method gets the current time in a CPU.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    DateTime dt = new DateTime();

    // Get CPU time
    error = device.GetTime(ref dt);

    device.Disconnect();
}

DeviceAPI.StopCommunications();
```


6.3.16 GetDeviceInformation

Get device information (CPU only)

The "GetDeviceInformation" method gets type, article number, and firmware version from a CPU. You must assign an object that returns the data.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    DeviceInfo info = null;

    // Get device information
    error = device.GetDeviceInformation(ref info);
    if (error.Succeeded)
    {
        String strType = info.CPUType;
        String strArticleNumber = info.ArticleNumber;
        String strFW = info.FirmwareVersion;
    }
    device.Disconnect();
}

DeviceAPI.StopCommunications();
```

6.3.17 GetRackInformation

Get rack Information (CPU and modules)

The "GetRackInformation" method gets rack information from a CPU. You must assign an object to return the data.

```

using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    RackInfo info = new RackInfo();

    // Get Rack information
    error = device.GetRackInformation(ref info);
    if (error.Succeeded)
    {
        foreach (Module module in info.LocalModules)
        {
            UInt32 id = module.ID
            bool bConfigured = module.Configured;
            String strName = module.Name;
            String strType = module.ModuleType;
            UInt32 slot = module.Slot;
            UInt32 subSlot = module.SubSlot;
            String strArticleNumber = module.ArticleNumber;
            String strSerialNumber = module.SerialNumber;
            String strFirmware = module.FirmwareVersion;
            UInt32 hwVersion = module.HardwareVersion;
        }
        // Process each remote interface
        foreach (RemoteInterface interFace in info.RemoteInterfaces)
        {
            String strInterfaceName = interFace.Name;

            // Process each remote device
            foreach (RemoteDevice remote in interFace.RemoteDevices)
            {
                // Process each module on this device
                foreach (Module module in remote.Modules)
                {
                    bool bConfigured = module.Configured;
                    String strName = module.Name;
                    String strType = module.ModuleType;
                    UInt32 slot = module.Slot;
                    UInt32 subSlot = module.SubSlot;
                    String strArticleNumber =
module.ArticleNumber;

```

```

        String strSerialNumber = module.SerialNumber;
        String strFirmware = module.FirmwareVersion;
        UInt32 hwVersion = module.HardwareVersion;
    }
}
}
device.Disconnect();
}

DeviceAPI.StopCommunications();

```

6.3.18 Backup

Backup data in a CPU to a .s7pbkp file

The "Backup" method performs a backup of the data in a supported CPU.

```

using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1500);

// Event for showing progress
device.BackupBlockUploaded += new
BackupBlockHandler(BackupBlockUploaded);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    error = device.Backup(@"C:\MyCPUBackupFile.s7pbkp", this);
}
device.BackupBlockUploaded -= new
BackupBlockHandler(BackupBlockUploaded);
device.Disconnect();

DeviceAPI.StopCommunications();

void BackupBlockUploaded(object sender, BackupBlockEventArgs e)
{
    UInt64 nBytesRead = e.BytesRead;
}

```

6.3.19 Restore

Restore CPU data from a .s7pbkp file

The "Restore" method restores CPU data from the .s7pbkp file created by the previous backup of a supported CPU.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1500);

device.RestoreBlockDownloaded += new
RestoreBlockHandler(RestoreBlockDownloaded);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    error = device.Restore(@"C:\MyCPUBackupFile.s7pbkp", this);
}
device.RestoreBlockDownloaded -= new
RestoreBlockHandler(RestoreBlockDownloaded);
device.Disconnect();

void RestoreBlockDownloaded(object sender, RestoreBlockEventArgs e)
{
    Int32 index = e.Index;
    Int32 count = e.Count;
}
DeviceAPI.StopCommunications();
```

6.3.20 GetDiagnosticsBuffer

Get diagnostic information from a CPU

The "GetDiagnosticsBuffer" method retrieves diagnostic information from a CPU.

```
using Siemens.Automation.AutomationTool.API;

DeviceAPI.StartCommunications();

DeviceAPI device = new DeviceAPI(DeviceType.S7_1200);

Error error = device.Connect("192.168.2.10", new
EncryptedString("MyPassword"));
if (error.Succeeded)
{
    DiagnosticsItem[] aDiagnosticsItems = null;

    error = device.GetDiagnosticsBuffer(ref aDiagnosticsItems);
    if (error.Succeeded)
    {
        foreach (DiagnosticsItem item in aDiagnosticsItems)
        {
            String strDescription = item.Description1 +
item.Description2;
            DateTime dt = item.TimeStamp;
        }
        device.Disconnect();
    }
}

DeviceAPI.StopCommunications();
```


SIMATIC Automation Tool device support

7.1 S7-1200 CPU support

S7-1200 operation support and firmware version

A check mark (✓) means that the operation is supported. Standard CPUs have only the firmware version number in the column header. Fail-Safe CPUs have "Fail-Safe" in the column header.

S7-1200	V1.x	V2.x	V3.x	V4.0	V4.1	Fail-Safe V4.1
Scan for devices	✓	✓	✓	✓	✓	✓
Flash LED	✓	✓	✓	✓	✓	✓
Set IP address	✓	✓	✓	✓	✓	✓
Set PROFINET name	✓	✓	✓	✓	✓	✓
Set time to PG/PC time	✓	✓	✓	✓	✓	✓
Program update	✓	✓	✓	✓	✓	
Firmware update				✓	✓	✓
Put CPU in RUN/STOP	✓	✓	✓	✓	✓	✓
Reset CPU memory	✓	✓	✓	✓	✓	
Read Diagnostic buffer	✓	✓	✓	✓	✓	✓
Upload service data	✓	✓	✓	✓	✓	✓
Backup and Restore						
Reset to factory defaults	✓	✓	✓	✓	✓	

7.2 S7-1500 CPU support

S7-1500 operation support and firmware version

A check mark (✓) means that the operation is supported. Standard CPU models have only the firmware version number in the column header. Fail-Safe CPU models have "Fail-Safe" in the column header.

S7-1500	V1.0	V1.1	V1.5	V1.6	V1.7	V1.8	Fail-Safe			
							V1.5	V1.6	V1.7	V1.8
Scan for devices	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Flash LED	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Set IP address	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Set PROFINET name	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Set time to PG/PC time	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Program update	✓	✓	✓	✓	✓	✓				
Firmware update	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Put CPU in RUN/STOP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reset CPU memory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Diagnostic buffer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Upload service data	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Backup and Restore			✓	✓	✓	✓				
Reset to factory defaults	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

7.3 ET 200SP CPU support (Distributed I/O controller based on ET 200SP and S7-1500)

ET 200SP CPU operation support and firmware version

A check mark (✓) means that the operation is supported. Standard CPUs have only the firmware version number in the column header. Fail-Safe CPUs have "Fail-Safe" in the column header.

ET 200SP CPU	V1.6	V1.7	V1.8	Fail-Safe	
				V1.7	V1.8
Scan for devices	✓	✓	✓	✓	✓
Flash LED	✓	✓	✓	✓	✓
Set IP address	✓	✓	✓	✓	✓
Set PROFINET name	✓	✓	✓	✓	✓
Set time to PG/PC time	✓	✓	✓	✓	✓
Program update	✓	✓	✓		
Firmware update	✓	✓	✓	✓	✓
Put CPU in RUN/STOP	✓	✓	✓	✓	✓
Reset CPU memory	✓	✓	✓	✓	✓
Read Diagnostic buffer	✓	✓	✓	✓	✓
Upload service data	✓	✓	✓	✓	✓
Backup and Restore	✓	✓	✓		
Reset to factory defaults	✓	✓	✓	✓	✓

7.4 ET 200SP support (PROFINET I/O)

ET 200SP operation support and firmware version

A check mark (✓) means that the operation is supported.

ET 200SP	V1.0	V1.1	V2.1	V2.2	V3.0	V3.1	V3.2
Scan for devices	✓	✓	✓	✓	✓	✓	✓
Flash LED	✓	✓	✓	✓	✓	✓	✓
Set IP address	✓	✓	✓	✓	✓	✓	✓
Set PROFINET name	✓	✓	✓	✓	✓	✓	✓
Firmware update	✓	✓	✓	✓	✓	✓	✓
Reset to factory defaults	✓	✓	✓	✓	✓	✓	✓

7.5 ET 200MP support (PROFINET I/O)

ET 200MP operation support and firmware version

A check mark (✓) means that the operation is supported.

ET 200MP	V1.0	V2.0	V3.0
Scan for devices	✓	✓	✓
Flash LED	✓	✓	✓
Set IP address	✓	✓	✓
Set PROFINET name	✓	✓	✓
Firmware update	✓	✓	✓
Reset to factory defaults	✓	✓	✓

7.6 ET 200AL support (PROFINET I/O)

ET 200AL operation support and firmware version

A check mark (✓) means that the operation is supported.

ET 200AL	V1.0
Scan for devices	✓
Flash LED	✓
Set IP address	✓
Set PROFINET name	✓
Firmware update	✓
Reset to factory defaults	✓

7.7 ET 200S support (PROFINET I/O)

ET 200S operation support and firmware version

A check mark (✓) means that the operation is supported.

ET 200S	V3.x	V5.x	V6.x	V7.x
Scan for devices	✓	✓	✓	✓
Flash LED	✓	✓	✓	✓
Set IP address	✓	✓	✓	✓
Set PROFINET name	✓	✓	✓	✓
Firmware update	✓	✓	✓	✓
Reset to factory defaults	✓	✓	✓	✓

Note

ET 200S CPU not supported

The ET 200S CPU is not supported by the SIMATIC Automation Tool

7.8 ET 200M support (PROFINET I/O)

ET 200M operation support and firmware version

A check mark (✓) means that the operation is supported.

ET 200M	V2.x	V3.x	V4.x
Scan for devices	✓	✓	✓
Flash LED	✓	✓	✓
Set IP address	✓	✓	✓
Set PROFINET name	✓	✓	✓
Firmware update	✓	✓	✓
Reset to factory defaults	✓	✓	✓

7.9 ET 200eco support (PROFINET I/O)

ET 200eco operation support and firmware version

A check mark (✓) means that the operation is supported.

ET 200eco	V6.x	V7.x
Scan for devices	✓	✓
Flash LED	✓	✓
Set IP address	✓	✓
Set PROFINET name	✓	✓
Firmware update	✓	✓
Reset to factory defaults	✓	✓

7.10 ET 200pro support (PROFINET I/O)

ET 200pro operation support and firmware version

A check mark (✓) means that the operation is supported.

ET 200pro	V2.x	V4.x	V5.x	V6.x	V7.x
Scan for devices	✓	✓	✓	✓	✓
Flash LED	✓	✓	✓	✓	✓
Set IP address	✓	✓	✓	✓	✓
Set PROFINET name	✓	✓	✓	✓	✓
Firmware update	✓	✓	✓	✓	✓
Reset to factory defaults	✓	✓	✓	✓	✓

Note

ET 200pro CPU not supported

The ET 200pro CPU is not supported by the SIMATIC Automation Tool

Index

A

API (application interface)
AutomationToolAPI.dll, 15
classes and methods overview, 65
Automation tool overview, 9

B

Backup Device, 44

C

Commands
Backup Device, 44
change IP address, 31
change PROFINET name, 32
download new program, 36
execution order, 52
flash LEDs, 35
import/export, 54
Install new firmware, 39
memory reset, 47
read diagnostics buffer, 51, 51
reset to factory default values, 46
Restore Name, 44
RUN/STOP, 33
save/save as, 53
scan, 25
set time in CPU, 50
upload service data, 48
Communication setup, 21
Contact information, 3
CPU
IP configuration requirement, 19
PROFINET name configuration requirement, 19
csv file, 54
Customer support, 3

D

Device support
ET 200AL, 98
ET 200eco, 99
ET 200M, 99

ET 200MP, 98
ET 200pro, 100
ET 200S, 98
ET 200SP, 97
ET 200SP CPU, 97
S7-1200, 95
S7-1500, 96
Device table, 25
DeviceAPI
Backup, 91
Connect, 75
DeviceAPI constructor, 73
Disconnect, 75
DownloadFirmware, 80
DownloadProgram, 85
GetDeviceInformation, 89
GetDiagnosticsBuffer, 93
GetMode, 77
GetPLCName, 87
GetRackInformation, 90
GetStationName, 86
GetTime, 88
MemoryReset, 79
ResetToFactory, 78
Restore, 92
SetMode, 76
SetTime, 88
StartCommunications and StopCommunications, 74
UploadServiceData, 86
Diagnostics buffer, 51, 51

F

Filtering device rows, 25
Firmware update, 39

H

Hotline, 3

I

IP address
configuration requirement, 19
device setting, 31

M

Memory reset, 47

Menu

- edit, 58

- file, 57

- help, 62

- network, 59

- tools, 60

N

Network

- access, 12

- options, 13

NetworkAPI

- FlashLED, 69

- GetDeviceType, 71

- NetworkAPI constructor, 67

- ResetToFactory, 72

- ScanNetworkDevices, 68

- ScanNetworkInterfaceCards, 67

- SetIP, 69

- SetNetworkInterface, 68

- SetProfinetName, 70

O

Operating system support, 17

P

PROFINET name configuration requirement, 19

R

Reset factory defaults, 46

Restore Name, 44

RUN mode, 33

S

sat file extension name, 53

Scan a network, 25

Service and support, 3

Set Time, 50

Shortcut keys, 64

Siemens technical support, 3

STOP mode, 33

Support, 3

T

Technical support, 3

Toolbar icons, 63

U

Upload service data, 48