General:
Project name: Projekt9
TIA Portal: V15
Station name: S71500/ET200MP-Station_2
Module name: PLC_4
Module type: CPU 1516-3 PN/DP

Status:
Operating Mode: RUN
Status: ✓ OK
Mode selector: RUN

CPU operator panel:

Edition 11/2023

FUNCTION MANUAL

SIMATIC

S7-1500, ET 200SP, ET 200pro

Web server

support.industry.siemens.com

SIEMENS

Introduction	1
Safety and security instructions	2
General information	3
Web pages	4
API (Application Programming Interface)	5

SIMATIC

S7-1500, SIMATIC Drive Controller, ET 200SP, ET 200pro Web server

Function Manual

11/2023
A5E03484625-AK

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.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E03484625-AK
Ⓟ 09/2023 Subject to change

Table of contents

Introduction 1

Purpose of the documentation

This documentation supports you in the operation of the Web server.

The Web server offers, among other things, web page access to diagnostic data and to process data of the CPU.

Basic knowledge required

The following knowledge is required in order to understand the documentation:

- General knowledge in the field of automation technology
- Knowledge of the SIMATIC industrial automation system
- Experience of working with Windows-based computers
- Knowledge about how to use STEP 7 (TIA Portal)

Conventions

STEP 7: In this documentation, "STEP 7" is used as a synonym for all versions of the configuring and programming software "STEP 7 (TIA Portal)".

Please also observe notes marked as follows:

NOTE

A note contains important information on the product described in the documentation, on the handling of the product and on the section of the documentation to which particular attention should be paid.

Scope of the documentation

This documentation is valid for CPUs as of firmware version V2.5 and contains illustrations of the Web server user interface. The illustrations used can be transferred to the following CPUs:

- The CPUs of the SIMATIC S7-1500 automation system
- The CPUs of the S7-1500 Software Controller for the Windows and Industrial OS operating systems
- The CPUs of the SIMATIC Drive controller
- The CPUs of the ET 200SP distributed I/O system
- The CPUs 1516pro-2 PN and 1513pro-2 PN of the ET 200pro distributed I/O system

The displayed illustrations can differ from the interface of the Web server in some details, e.g. depending on the web browser used.

What's new in the Web Server Function Manual, Version 11/2023 compared to Version 10/2022

What's new?		What are the customer benefits?	Where can I find information?
New contents	New Web API methods	Many new API methods extend your access options to the CPU via the Web API.	You can find an overview of all Web API methods in the section Supported clients (Page 153).
	Introduction of the local user management for CPUs with firmware version ≥ V3.1	You define and manage the users of CPUs in a TIA Portal project. You assign roles and rights to the users. The user accounts are valid project-wide.	Section Configuring the Web server (Page 26)
Changed contents	Scope of this function manual expanded to include the R/H-CPU of the redundant system S7-1500R/H	You can apply Web API methods on the R/H-CPUs.	Section API (Application Programming Interface) (Page 153)

What's new in the Web Server Function Manual, Version 10/2022 compared to Version 05/2021

What's new?		What are the customer benefits?	Where can I find information?
New contents	New Web API methods	Many new API methods extend your access options to the CPU via the Web API.	Section API (Application Programming Interface) (Page 153)
	Management of certificates in runtime	In addition to loading Web server certificates via the TIA Portal, this option makes it possible to provide such certificates from a certificate management server at runtime. This allows you, for example, to update expiring certificates in time without interrupting the process.	Section Managing certificates in runtime (Page 33)
	Motion Control Diagnostics (T CPUs)	Display of the diagnostics of the axis status for the "Encoder homed" signal of the encoders of all 4 possible axes.	Section Motion Control Diagnostics (Page 64) Function manuals for S7-1500/1500T Motion Control on the Internet (https://support.industry.siemens.com/cs/ww/de/view/109751049)
Changed contents	Extension of the scope of this function manual to the CPU 1514SP-2 PN and the CPU 1514SPT-2 PN of the ET 200SP distributed I/O system.	Web server functions can now also be used on these CPUs.	• Equipment Manual CPU 1514SP-2 PN (https://support.industry.siemens.com/cs/ww/en/view/109813105) • Equipment Manual CPU 1514SPT-2 PN (https://support.industry.siemens.com/cs/ww/en/view/109813106)

What's new in the Web Server Function Manual, Version 05/2021 compared to Version 11/2019

What's new?		What are the customer benefits?	Where can I find information?
New contents	New Web API methods: • Plc.ReadOperatingMode • Plc.RequestChangeOperatingMode	Reading and changing CPU operating mode via Web API	Section Reading and changing the operating mode (Page 206)
	Ticket mechanism and ticket methods for handling the tickets: • Api.BrowseTickets • Api.CloseTicket	With the Web server as of firmware version V2.9, you can use the ticket mechanism of the Web API. You can use the ticket mechanism to transfer large amounts of data outside of the JSON-RPC protocol. The ticket mechanism is the basis for all file-based methods, such as file upload/download of resources.	Section Ticket mechanism (Page 164)
	Web applications of the Web API that can be loaded by the user: Web applications that can be loaded by the user provide you with an additional set of methods to manage web applications via Web API. You can use all available Web API methods within the web application.	Web applications that can be loaded by the user offer you the following major advantages as of firmware version V2.9 compared to the older method that provided the user-defined pages via the system function SFC 99 in STEP 7: Only the TIA Portal project in the SIMATIC.S7S directory on the SIMATIC Memory Card changes. Your TIA Portal project is extended by the option of saving resources (e.g. HTML, CSS, JavaScript, etc.) in the project but outside of the data blocks of the user program. The resources are saved in the associated web application. Via the Web API you can download the resources to your PC, edit them and upload them back to the CPU. This procedure results in significantly reduced development times of user-defined pages. You can access resources independent of the CPU operating mode (e.g. RUN, STOP) and update these. Web applications are also available in the STOP mode of the CPU.	Section Web applications that can be loaded by the user (Page 250)
	Creating user-defined pages with TIA Portal WinCC Unified as of Version V17	Compared to creating user-defined pages with a random HTML editor, creating and loading user-defined pages with WinCC Unified for SIMATIC S7-1500 CPUs offers the following advantages: • No HTML code expertise required • You can make changes to the user-defined pages while the CPU is in the RUN mode.	Section User-defined pages (Page 124) WinCC Unified online help

What's new?		What are the customer benefits?	Where can I find information?
Changed contents	Diagnostics	The display in the "Memory" tab is extended by information on the data type memory so that the user program can be changed accordingly when the available memory is not sufficient.	Section Diagnostics (Page 54)
	Motion Control Diagnostics	Comprehensive diagnostic options for Motion Control applications: • Diagnostics information is available for all technology objects • Improved display and grouping of the tags	Section Motion Control Diagnostics (Page 64)
	Communication	Extension of the "Parameter" tab to assign the network configuration via a DHCP server and display the DNS settings	Section Communication (Page 78)

What's new in the Web Server Function Manual, Version 11/2019 compared to Version 12/2017

What's new?		What are the customer benefits?	Where can I find information?
New contents	The CPU has a web-based API (Application Programming Interface) as an interface for: • Reading and writing CPU data • Executing functions (e.g. backing up and restoring the CPU configuration, changing the operating mode) The Web API supports all common browsers and command line programs, such as cURL and Wget.	• Established standard mechanisms for creating web pages: Automation Web Programming commands (AWP commands) are no longer required for output of CPU data • No dependency between custom Web pages and CPU program: No synchronization between user program and Web server required by the SFC 99 instruction • Lower communication load: A smaller data packet (JSON instead of HTML) of the custom web page generated by the CPU is transferred between the server and the client. This improves the communication performance. The CPU needs less runtime to generate the information and make it available. • Secure data traffic: the Web API only supports the transmission protocol "HTTPS"	Section API (Application Programming Interface) (Page 153)
Changed contents	Extension of the scope of this function manual to the CPUs of the SIMATIC Drive Controller	Webserver functions which you are familiar with from the CPUs of the SIMATIC S7-1500 can now also be used on the CPUs of the SIMATIC Drive Controller.	• System Manual SIMATIC Drive Controller (https://support.industry.siemens.com/cs/ww/en/view/109766665) • Equipment Manual SIMATIC Drive Controller (https://support.industry.siemens.com/cs/ww/en/view/109766666)

What's new in the Web Server Function Manual, Version 12/2017 compared to Version 09/2016

What's new?		What are the customer benefits?	Where can I find information?
New contents	New web page "User files"	You can download ASCII files (files in binary format) from the SIMATIC Memory Card, directory UserFiles\ to the web page and delete them.	Section User files (Page 122)
	Time display as Coordinated Universal Time (UTC) or as PLC local time	The display of the UTC allows you to use a uniform time for the web pages. You can set the format of the time display to Coordinated Universal Time (UTC) or PLC local time (default setting).	Section Start page with general CPU information (Page 47)
	Automated downloading, reading out and archiving of DataLogs	You can, for example, read out and archive DataLogs daily from one or more CPUs at a specific time via the Web server. Automatic downloading of DataLogs is realized either by the execution of scripts in, for example, Bash or via JavaScript on your HTML user-defined page.	Section Automated reading out of DataLogs (Page 120)
	The "Permit access only with HTTPS" check box is activated in the default setting of a configured CPU.	The web pages are transmitted by default via a secure connection and are protected from attacks by third parties.	Section Configuring the Web server (Page 26)
Changed contents	Web page "Module information": New column Device number	You can read the assignment of the device number to the device name.	Section Module information (Page 69)
	Web page "Topology": Selection of the available PROFINET interfaces, for example X1, X2, CM 1542-1	You can select the topology display for the PROFINET interfaces X1, X2 and for connected PROFINET communication modules.	Section Topology (Page 84)
	Web page "DataLogs": New column for deleting DataLog files	You can delete DataLog files via the Web server.	Section DataLogs (Page 119)
	Web page "Record": Changes in the display of Trace recordings	You can evaluate the Trace recordings in more detail through the extension of the display.	Section Record (Page 101)
	Web page "Record": New arithmetic functions	In the case of completed measurements you can combine the measured signals mathematically with each other and this generate signals that were not recorded. You can, for example, form the difference of two signals in order to better display the deviation of the current pressure of a boiler from the set setpoint value.	

What's new in the Web Server Function Manual, Version 09/2016 compared to Version 12/2014

What's new?		What are the customer benefits?	Where can I find information?
New contents	Handling of certificate modified	You protect the Web server connection against tapping or distortion of the communication through access via the secure transmission protocol "HTTPS" including a special Web server certificate.	Section Configuring the Web server (Page 26)
	Four additional languages for the Web server interface	You can set the Web server interface to the following languages: • Korean • Russian • Turkish • Portuguese (Brazil)	Section Start page with general CPU information (Page 47)
	Assignment of different project languages extended	You can assign up to three different project languages for comments, alarm texts and diagnostics information to the user interface languages of the Web server.	Section Language settings (Page 37)
	"Start page" web page extended	The display of the TIA project name immediately indicates whether the desired project is selected.	Section Start page with general CPU information (Page 47)
	"Diagnostics" web page extended by one tab: • "Program protection" • "Runtime information" • "Fail-safe" (with an F-CPU)	Here you can find information about: • Know-how protection or copy protection of the PLC program • Program/communication load and cycle time • F-collective signatures, cycle times and runtimes of the F-runtime group(s)	Section Diagnostics (Page 54)
	"Alarms" web page extended	You can acknowledge alarms of the CPU via the Web server.	Section Alarms (Page 76)
	"Tag status" and "Watch tables" web pages extended	You can change the value of tags and write them to the CPU, also using the absolute address.	• Section Tag status (Page 93) • Section Watch tables (Page 95)
	New web page "Online backup"	You can back up and restore the CPU configuration to/from the SIMATIC Memory Card via the Web server.	Section Online backup (Page 97)
	New "Motion Control Diagnostics" web page	You can monitor statuses, errors, technology alarms and the current values of configured technology objects (TOs) with the Web server without STEP 7.	Section Motion Control diagnostics (Page 64)
	New "Record" web page	You can read, view and save trace recordings via the Web server and thus obtain plant and project information for diagnostics and maintenance without STEP 7.	Section Record (Page 101)
Changed contents	Extension of the scope of this function manual to the CPUs of the ET 200SP distributed I/O system and the CPU 1516pro-2 PN	Functions that you will be familiar with from the SIMATIC S7-1500 CPUs are implemented in CPUs in other designs (ET 200SP) and in the CPU 1516pro-2 PN (degree of protection IP65, IP66 and IP67).	• Equipment Manual CPU 1510SP-1 PN (https://support.industry.siemens.com/cs/ww/en/view/90157130)

What's new?		What are the customer benefits?	Where can I find information?
Changed contents			• Equipment Manual CPU 1512SP-1 PN (https://support.industry.siemens.com/cs/ww/en/view/90157013) • Operating instructions CPU 1516pro-2 PN (https://support.industry.siemens.com/cs/ww/en/view/109482416)
	Web page "Watch tables": Note added on the maximum configuration limits.		Section Watch tables (Page 95)
	Web page "User-defined pages": Note added on the maximum size of the HTML pages.		Section User pages (Page 124)

Recycling and disposal

For environmentally sustainable recycling and disposal of your old equipment, contact a certified electronic waste disposal service and dispose of the equipment according to the applicable regulations in your country.

Industry Mall

The Industry Mall is the catalog and order system of Siemens AG for automation and drive solutions on the basis of Totally Integrated Automation (TIA) and Totally Integrated Power (TIP).

You can find catalogs for all Automation and Drives products on the Internet (https://mall.industry.siemens.com).

1.1 Function manuals documentation guide

1.1.1 Information classes Function Manuals

The documentation for the SIMATIC S7-1500 automation system, for the 1513/1516pro-2 PN, SIMATIC Drive Controller CPUs based on SIMATIC S7-1500 and the SIMATIC ET 200MP, ET 200SP, ET 200AL and ET 200eco PN distributed I/O systems is arranged into three areas.
This arrangement enables you to access the specific content you require.
You can download the documentation free of charge from the Internet (https://support.industry.siemens.com/cs/ww/en/view/109742705).

Basic information

The system manuals and Getting Started describe in detail the configuration, installation, wiring and commissioning of the SIMATIC S7-1500, SIMATIC Drive Controller, ET 200MP, ET 200SP, ET 200AL and ET 200eco PN systems. Use the corresponding operating instructions for 1513/1516pro-2 PN CPUs.
The STEP 7 online help supports you in the configuration and programming.
Examples:

- Getting Started S7-1500
- System manuals
- Operating instructions ET 200pro and 1516pro-2 PN CPU
- Online help TIA Portal

Device information

Equipment manuals contain a compact description of the module-specific information, such as properties, wiring diagrams, characteristics and technical specifications.
Examples:

- Equipment manuals for CPUs
- Equipment manuals for interface modules
- Equipment manuals for digital modules
- Equipment manuals for analog modules
- Equipment manuals for communication modules
- Equipment manuals for technology modules
- Equipment manuals for power supply modules
- Equipment manuals for BaseUnits

General information

The function manuals contain detailed descriptions on general topics relating to the SIMATIC Drive Controller and the S7-1500 automation system.

Examples:

- Function Manual Diagnostics
- Function Manual Communication
- Function Manuals Motion Control
- Function Manual Web Server
- Function Manual Cycle and Response Times
- PROFINET Function Manual
- PROFIBUS Function Manual

Product Information

Changes and supplements to the manuals are documented in a Product Information. The Product Information takes precedence over the device and system manuals.

You will find the latest Product Information on the Internet:

- S7-1500/ET 200MP (https://support.industry.siemens.com/cs/de/en/view/68052815)
- SIMATIC Drive Controller (https://support.industry.siemens.com/cs/de/en/view/109772684/en)
- Motion Control (https://support.industry.siemens.com/cs/de/en/view/109794046/en)
- ET 200SP (https://support.industry.siemens.com/cs/de/en/view/73021864)
- ET 200eco PN (https://support.industry.siemens.com/cs/ww/en/view/109765611)

Manual Collections

The Manual Collections contain the complete documentation of the systems put together in one file.

You will find the Manual Collections on the Internet:

- S7-1500/ET 200MP/SIMATIC Drive Controller (https://support.industry.siemens.com/cs/ww/en/view/86140384)
- ET 200SP (https://support.industry.siemens.com/cs/ww/en/view/84133942)
- ET 200AL (https://support.industry.siemens.com/cs/ww/en/view/95242965)
- ET 200eco PN (https://support.industry.siemens.com/cs/ww/en/view/109781058)

1.1.2 Basic tools

Tools

The tools described below support you in all steps: from planning, over commissioning, all the way to analysis of your system.

TIA Selection Tool

The TIA Selection Tool tool supports you in the selection, configuration, and ordering of devices for Totally Integrated Automation (TIA).

As successor of the SIMATIC Selection Tools , the TIA Selection Tool assembles the already known configurators for automation technology into a single tool.

With the TIA Selection Tool , you can generate a complete order list from your product selection or product configuration.

You can find the TIA Selection Tool on the Internet.
(https://support.industry.siemens.com/cs/ww/en/view/109767888)

SIMATIC Automation Tool

You can use the SIMATIC Automation Tool to perform commissioning and maintenance activities on various SIMATIC S7 stations as bulk operations independent of TIA Portal.

The SIMATIC Automation Tool offers a wide range of functions:

- Scanning of a PROFINET/Ethernet system network and identification of all connected CPUs
- Assignment of addresses (IP, subnet, Gateway) and device name (PROFINET device) to a CPU
- Transfer of the date and the programming device/PC time converted to UTC time to the module
- Program download to CPU
- RUN/STOP mode switchover
- CPU localization through LED flashing
- Reading out of CPU error information
- Reading the CPU diagnostic buffer
- Reset to factory settings
- Firmware update of the CPU and connected modules

You can find the SIMATIC Automation Tool on the Internet.
(https://support.industry.siemens.com/cs/ww/en/view/98161300)

PRONETA

SIEMENS PRONETA (PROFINET network analysis) is a commissioning and diagnostic tool for PROFINET networks. PRONETA Basic has two core functions:

- In the network analysis, you get an overview of the PROFINET topology. Compare a real configuration with a reference installation or make simple parameter changes, e.g. to the names and IP addresses of the devices.
- The "IO test" is a simple and rapid test of the wiring and the module configuration of a plant, including documentation of the test results.

You can find SIEMENS PRONETA Basic on the Internet:
(https://support.industry.siemens.com/cs/ww/en/view/67460624)

SIEMENS PRONETA Professional is a licensed product that offers you additional functions. It offers you simple asset management in PROFINET networks and supports operators of automation systems in automatic data collection/acquisition of the components used through various functions:

- The user interface (API) offers an access point to the automation cell to automate the scan functions using MQTT or a command line.
- With PROFIenergy diagnostics, you can quickly detect the current pause mode or the readiness for operation of devices that support PROFIenergy and change these as needed.
- The data record wizard supports PROFINET developers in reading and writing acyclic PROFINET data records quickly and easily without PLC and engineering.

You can find SIEMENS PRONETA Professional on the Internet.
(https://www.siemens.com/proneta-professional)

SINETPLAN

SINETPLAN, the Siemens Network Planner, supports you in planning automation systems and networks based on PROFINET. The tool facilitates professional and predictive dimensioning of your PROFINET installation as early as in the planning stage. In addition, SINETPLAN supports you during network optimization and helps you to exploit network resources optimally and to plan reserves. This helps to prevent problems in commissioning or failures during productive operation even in advance of a planned operation. This increases the availability of the production plant and helps improve operational safety.

The advantages at a glance

- Network optimization thanks to port-specific calculation of the network load
- Increased production availability thanks to online scan and verification of existing systems
- Transparency before commissioning through importing and simulation of existing STEP 7 projects
- Efficiency through securing existing investments in the long term and the optimal use of resources

You can find SINETPLAN on the Internet
(https://new.siemens.com/global/en/products/automation/industrial-communication/profinet/sinetplan.html).

1.1.3 SIMATIC Technical Documentation

Additional SIMATIC documents will complete your information. You can find these documents and their use at the following links and QR codes.

The Industry Online Support gives you the option to get information on all topics. Application examples support you in solving your automation tasks.

Overview of the SIMATIC Technical Documentation

Here you will find an overview of the SIMATIC documentation available in Siemens Industry Online Support:

Industry Online Support International (https://support.industry.siemens.com/cs/ww/en/view/109742705)

Watch this short video to find out where you can find the overview directly in Siemens Industry Online Support and how to use Siemens Industry Online Support on your mobile device:

Quick introduction to the technical documentation of automation products per video (https://support.industry.siemens.com/cs/us/en/view/109780491)

YouTube video: Siemens Automation Products - Technical Documentation at a Glance (https://youtu.be/TwLSxxRQQsA)

Retention of the documentation

Retain the documentation for later use.

For documentation provided in digital form:

1. Download the associated documentation after receiving your product and before initial installation/commissioning. Use the following download options:
   - Industry Online Support International: (https://support.industry.siemens.com)

     The article number is used to assign the documentation to the product. The article number is specified on the product and on the packaging label. Products with new, non-compatible functions are provided with a new article number and documentation.
   - ID link:

     Your product may have an ID link. The ID link is a QR code with a frame and a black frame corner at the bottom right. The ID link takes you to the digital nameplate of your product. Scan the QR code on the product or on the packaging label with a smartphone camera, barcode scanner, or reader app. Call up the ID link.
2. Retain this version of the documentation.

Updating the documentation

The documentation of the product is updated in digital form. In particular in the case of function extensions, the new performance features are provided in an updated version.

1. Download the current version as described above via the Industry Online Support or the ID link.
2. Also retain this version of the documentation.

mySupport

With "mySupport" you can get the most out of your Industry Online Support.

Registration	You must register once to use the full functionality of "mySupport". After registration, you can create filters, favorites and tabs in your personal workspace.
Support requests	Your data is already filled out in support requests, and you can get an overview of your current requests at any time.
Documentation	In the Documentation area you can build your personal library.
Favorites	You can use the "Add to mySupport favorites" to flag especially interesting or frequently needed content. Under "Favorites", you will find a list of your flagged entries.
Recently viewed articles	The most recently viewed pages in mySupport are available under "Recently viewed articles".
CAx data	The CAx data area gives you access to the latest product data for your CAx or CAe system. You configure your own download package with a few clicks: • Product images, 2D dimension drawings, 3D models, internal circuit diagrams, EPLAN macro files • Manuals, characteristics, operating manuals, certificates • Product master data

You can find "mySupport" on the Internet. (https://support.industry.siemens.com/My/ww/en)

Application examples

The application examples support you with various tools and examples for solving your automation tasks. Solutions are shown in interplay with multiple components in the system - separated from the focus on individual products.
You can find the application examples on the Internet.
(https://support.industry.siemens.com/cs/ww/en/ps/ae)

Safety and security instructions 2

2.1 Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines, and networks.

In order to protect plants, systems, machines, and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens’ products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For more information on protective industrial cybersecurity measures for implementation, please visit (https://www.siemens.com/global/en/products/automation/topic-areas/industrial-cybersecurity.html).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' exposure to cyber threats.

To stay informed about product updates at all times, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
(https://new.siemens.com/global/en/products/services/cert.html).

General information 3

3.1 Properties of the Web server

Benefits of the web server

The web server enables monitoring and administering of the CPU by authorized users over a network. Evaluations, diagnostics, and modifications are thus possible over long distances. Monitoring and evaluation is possible without STEP 7, only a web browser is required. Note that you must take appropriate measures to protect the CPU from compromise (such as restricting network access, using firewalls).

Activating the web server

The web server is deactivated in the delivery state of the CPU. This means that you must load a project in which the web server is activated to enable access using the web browser.

Security functions

The web server provides the following safety functions:

- Access via the secure transmission protocol "HTTPS" using the CA-signed web server certificate
- User authorizations you can configure by means of user list
- Activation for specific interfaces

Web browser

You need a web browser to access the HTML pages of the CPU.

The web browsers listed below have been tested for communication with the CPU. Other web browsers may also work, especially newer versions. However, if problems occur with web browsers not mentioned here that cannot be rectified, use one of the following tested web browsers:

- Microsoft Edge (Version 116.0)
- Google Chrome (Version 116.0)
- Mozilla Firefox (Version 116.0.3)
- Opera (Version 102.0)
- Mobile Safari and Chrome (iOS 12.5.1, 16.5)
- Android browser (7.x, 8.x and 10.x, 11.x, 12.x)
- Chrome for Android (7.x, 8.x and 10.x, 11.x, 12.x)
- HMI Panels:
  - Basic Panel
  - Comfort Panel

NOTE

If you are using Internet Explorer, deactivate "Compatibility view" in the settings ("Options" menu).

NOTE

For access to display devices with low screen resolution, we recommend the use of basic web pages, see section Basic websites (Page 151).

NOTE

Older versions of the web browsers named above, which previously supported access to the HTML pages of the CPU, continue to allow this. However, these older versions do not support the new functions and HTML pages described in this edition.

NOTE

Two reserved communication connections are available to the web server for communication with the CPU.

Depending on the web browser used, different numbers of connections to the CPU are established. If more connections are available, more communication connections will be established.

If no more connections are available, display or functional problems may occur. This is because the web server will reject all other communication connections apart from the two that are reserved.
For this reason, the web pages may not load fully.

NOTE

If you access the web server of the CPU using a communications processor (CP), ensure that the cache (temporary Internet files) is activated in your web browser. Choose the "Automatically" option in the cache settings of your web browser.

If the cache is deactivated or if a setting other than "Automatically" is made in the cache settings of your web browser, this may result in slow access times and incomplete display.

NOTE

After a firmware update of the CPU, incorrect display of web pages can occur in various web browsers. This is caused by problems of the new CPU firmware with the cache of the web browser.

Solution: Press F5 or clear the web browser cache.

NOTE

Web browser behavior can be different if a certificate is not valid yet or is no longer valid. Siemens has no influence on this behavior. This means we cannot guarantee the reliable functionality of the web server with invalid certificates.

Reading out data

With the web server, you can read out the following data from the CPU and, in some cases, modify and write back the data to the CPU.

- Start page with general CPU information (Page 47)
- Information on Diagnostics (Page 54)
  - Identification
  - Program protection
  - Memory
  - Runtime information
  - Fail-safe (with an F CPU)
- Contents of the diagnostics buffer (Page 63)
- Module information (Page 69)
- Firmware update (Page 74)
- Alarms (Page 76)
- Information on Communication (Page 78)
  - Important interface parameters
  - Port statistics
  - Display of the communication resources
  - Display of the communication connections

- PROFINET-Topology (Page 84)
  - Graphical view (set and actual topology)
  - Table view (actual topology)
  - Status overview
- Tag status (Page 93)
- Watch tables (Page 95)
- User pages (Page 124)
- Filebrowser (Page 150)
- DataLogs (Page 119)
- User files (Page 122)
- Online backup and restoration of the configuration (Page 97)
- Diagnostics information for technology objects (Page 64)
- Evaluation of trace recordings (Page 101)
- Reading out service data (Page 151)
- Basic websites (Page 151)

The HTML pages are described in more detail on the following pages.

NOTE

Max. characters at data type WSTRING

Note that the data type WSTRING is limited to 254 characters for the display in the web server. If the 254 characters are exceeded, the web server does not display the superfluous characters.

Web access to the CPU via PG/PC, HMI devices and mobile end devices

Proceed as follows to access the web server:

1. Use STEP 7 to download a project in which the web server is activated to the CPU.
2. Connect the display device (PG/PC, HMI, mobile terminal device) with the CPU or a communications module using a PROFINET interface.
   If you are working with WLAN, activate the WLAN on the display device and establish a connection to the access point (e.g. SCALANCE W788-1RR or SCALANCE W784-1), which is in turn connected to the CPU.
3. Open the web browser on the display device.
4. Enter the IP address of the interface of the CPU which is connected to the client in the "Address" field of the web browser in the following format: https://a.b.c.d (entry example: https://192.168.3.141).
   The introduction page of the CPU opens. From the intro page you can navigate to more information.

More information on access using the secure transmission protocol "HTTPS" is available in the section Configuring the Web server (Page 26).

More information

Using a smartphone, you can access the web server of the CPU either via WLAN or access to the CPU via the SIMATIC S7 app (using web server functionality). You can find additional information in the FAQ entry ID 103473392 on the Service&Support (https://support.industry.siemens.com/cs/ww/en/view/103473392) Internet page.

Note: The web server must also be activated for access to the CPU via the SIMATIC S7 app.

The SIMATIC S7 app offers you additional functions. You can find a detailed application example with further documentation and example projects on the Service&Support (https://support.industry.siemens.com/cs/ww/en/view/84133612) Internet page.

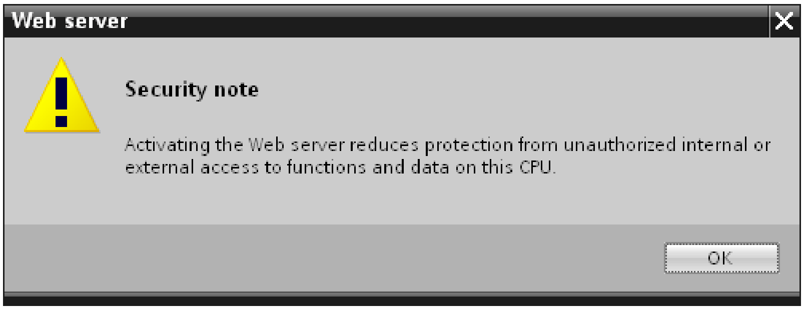
3.2 Configuring the Web server

To use the full functionality of the Web server, the following settings in STEP 7 are necessary.

Activate Web server on this module

The Web server is deactivated in the default setting of a configured CPU. Proceed as follows to activate the Web server:

1. Open the "Devices & Networks" view by double-clicking in the project tree in STEP 7.
2. Select the desired CPU in the device, network or topology view.
3. Navigate to the "Web server" area in the inspector window properties, "General" tab.
4. Select the "Activate Web server on this module" check box.
   The following note is output:



Web server

Security note

Activating the Web server reduces protection from unauthorized internal or external access to functions and data on this CPU.

Figure 3-1 Security note upon activation of the Web server in STEP 7

NOTE

When projects from deliveries are applied in which the Web server was already activated and configured on the module, this security note is not shown.

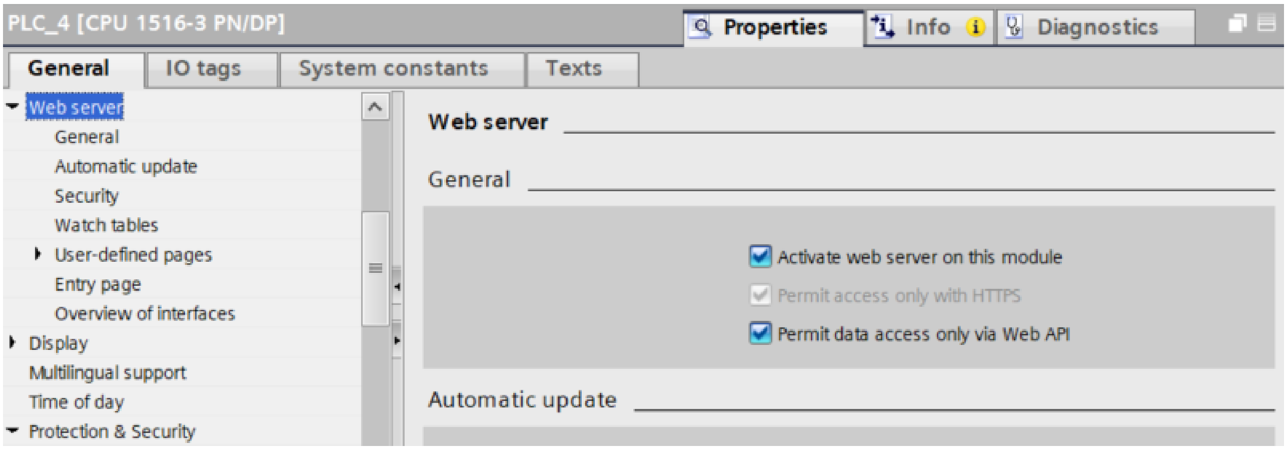
NOTE

Activating Web server for R/H-CPUs

For R/H-CPUs you can only activate the Web server for both R/H-CPUs and the identical settings. In addition to the Web server properties, this also applies to the "Overview of the interfaces", for the certificates and the user management.

You can use the IP addresses of both CPUs to use the Web server.

Depending on the CPU used, you can make your own settings or the settings are fixed.



☑ Activate web server on this module
☑ Permit access only with HTTPS
☑ Permit data access only via Web API

Figure 3-2 Web server settings in STEP 7

Permit access only with HTTPS

Note: A valid Web server certificate is required in the CPU to operate the Web server using the secure transfer protocol "HTTPS". You can find information on how to create and assign Web server certificates in the section Managing certificates via TIA Portal (Page 31).

To ensure secure access to the Web server the "Permit access only with HTTPS" check box is activated in the basic setting of a configured CPU.

The web pages are transmitted by default via a secure connection and are protected from attacks by third parties. Note that in this case the URL of the CPU starts with "https://".

The requirements for error-free HTTPS access to the CPU are as follows:

- The current date/time must be set in the CPU.

NOTE

When using secure communication (e.g. HTTPS), make sure that the corresponding modules have the current time of day and the current date. Otherwise the modules cannot check the validity period or evaluate the certificates used as invalid. Therefore, a secure connection cannot be established.

- The IP address of the CPU must be assigned.

- A valid Web server certificate offered by the CPU is installed in the web browser.

NOTICE

Safety-related functions only possible with CA-signed Web server certificate

The safety-related functions backup and restoring the CPU configuration, see section Online backup (Page 97), are only possible with a CA-signed Web server certificate.

A valid CA-signed Web server certificate in the CPU is also required:

- User management with password-protected users
- Saving and downloading diagnostics information in csv files

To use the full functionality of the Web server, we therefore recommend that you use the Certificate Manager to create a CA-signed server certificate in the global security settings and assign it to the CPU.

If no CA-signed Web server certificate is installed, a warning is output recommending that you do not use the page. To view the page, you may need to "Add an exception", depending on the web browser used.

A valid CA certificate is available for download from the "Intro" web page under "Download certificate".

You can find instructions for installing the certificate in the help system of your web browser and in the FAQ with the entry ID 103528224 at the Service&Support (https://support.industry.siemens.com/cs/ww/en/view/103528224) website.

NOTE

To protect against manipulation from the outside, download the certificate only in an environment that is guaranteed not to be compromised. Installation of the CA certificate has to be carried out once for each display device you wish to use.

Permit data access only via Web API

If you select the "Permit data access only via Web API" check box, only Web API-based functions are available, including:

- JSON-RPC interface
- Ticketing and web applications

Functions on the Web server via the unencrypted HTTP protocol and AWP commands are no longer accessible.

If you select the "Permit data access only via Web API" check box, the "Allow access only via HTTPS" check box is automatically selected and cannot be changed.

For R/H-CPUs, data access is only possible via Web API and is therefore preset and cannot be changed.

Access protection and user management

The encrypted connection created with the help of the certificate prevents eavesdropping or falsification of communication, but does not provide access protection. This means you have to protect your CPU from unauthorized access with the corresponding configuration in the user management.

The procedure for setting up the user management with password-protected users for the Web server is based on the configured firmware version of your project. You can find more information in the User management (Page 39) section.

You can find more information on access protection on the CPU in the STEP 7 online help, keyword: "Protection".

Activate automatic update

Automatic updating is activated in the default setting of a configured CPU.

The following web pages are updated automatically:

- Start page
- Diagnostics (memory, runtime information, fail-safe)
- Diagnostics buffer
- Motion Control Diagnostics
- Module information
- Alarms
- Communication
- Topology
- Tag status
- Watch tables
- Record
- DataLogs
- User files
- User-defined pages
- File browser

NOTE

The default activation interval is 10 seconds.
Larger data volumes or multiple HTTP/HTTPS-connections increase the update time.

Setting the language for the Web

In total, you can assign up to three different project languages to the user interface languages of the Web server.

In STEP 7, activate the project languages that you want to use and then assign one of the activated project languages to each of the Web server interface languages.

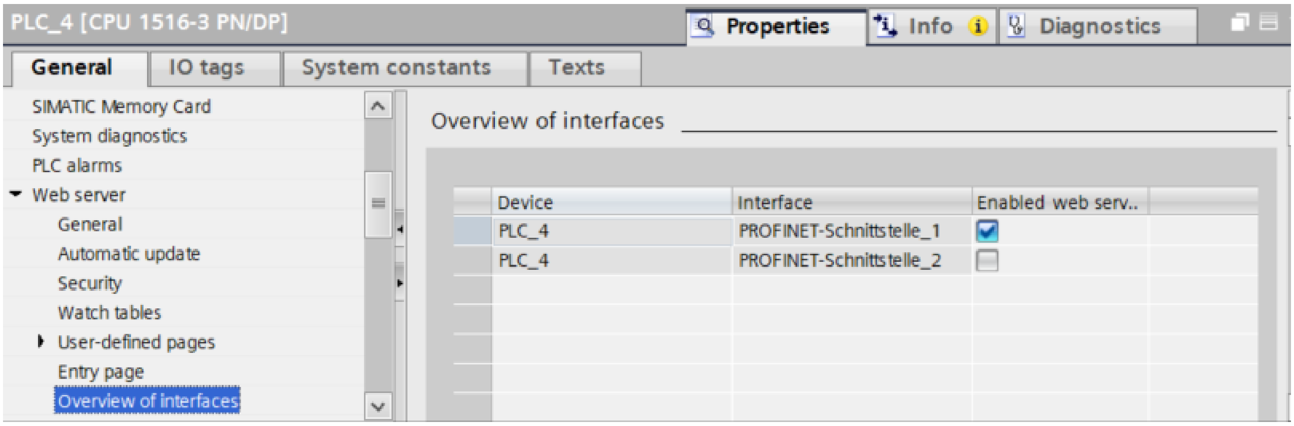
You can find more information about the language settings and a description of how to assign a project language to the interface languages in the section Language settings (Page 37).

User-defined pages

In the "User-defined pages" area, you can download your own web pages to the CPU and make your own web applications available via the web browser.
You can find more information in section User pages (Page 124).

Activation of the Web server for specific interfaces

In the "Overview of interfaces" area, you have the option to enable access to the Web server.



Overview of interfaces

Device	Interface	Enabled web serv..
PLC_4	PROFINET-Schnittstelle_1	☑
PLC_4	PROFINET-Schnittstelle_2	☐

Figure 3-3 Enabling access to the Web server via the interfaces

3.3 Certificate

3.3.1 Web server certificates

To secure the data exchange with a partner, different applications and communication functions of the CPU use device certificates that are managed specific to the application. In the case of the device certificate for the Web server, this involves a Web server certificate.

3.3.2 Managing certificates via TIA Portal

Creating and assigning a Web server certificate

Operation of the Web server using the secure transfer protocol "HTTPS" requires a valid Web server certificate.

For SIMATIC S7 1500 CPUs with firmware V2.0 and higher, you have to create the Web server certificate of the CPU yourself using STEP 7 and assign it to the Web server in the properties of the CPU. This certificate is also downloaded to the CPU automatically when the hardware configuration is downloaded.

NOTE

If you update the firmware of a SIMATIC S7 1500 CPU or ET 200SP CPU with a firmware version < V2.0 to a firmware version ≥ V2.0, a valid Web server certificate is automatically generated and used. The same applies to the replacement parts scenario in which a newer CPU replaces a CPU with firmware version < V2.0.

If you update or replace an already configured CPU, a valid Web server certificate is automatically generated and used for CPUs with a firmware version ≤ V1.8.

You can create different Web server certificates:

- If you use the global security settings for the certificate manager, the certification authority of the project (CA certificate) signs the device certificate of the Web server. During loading, the CA certificate of the project is automatically loaded as well.
- If you do not use the certificate manager in the global security settings, STEP 7 generates the device certificate as a self-signed certificate.

NOTICE

Utilizing the full functionality of the Web server

A valid CA-signed Web server certificate in the CPU is a requirement for the following functions:

- User management with password-protected users
- Saving and downloading diagnostic information in csv files
- Backup and restore of security-related functions such as the configuration of the CPU

To use the full functionality of the Web server, we therefore recommend you activate the global security settings of the certificate manager, create a CA-signed Web server certificate and assign it to the CPU.

Creating a self-signed Web server certificate

To create a self-signed Web server certificate with TIA Portal, follow these steps:

1. In the Inspector window Properties of the CPU, "General" tab, navigate to the "Web server > Security" area.
2. Click the "Add" button in the drop-down list to select a certificate.
   The "Create a new certificate" dialog opens.
3. Select the "Self-signed" check box in the follow-up dialog.

4. Enter the parameters for the new certificate or confirm the default settings.
   – Select "Web server" in the "Usage" box.
   – Enter the IP address(es) of the interface(s) or the domain name of the configured CPU in the "Subject Alternative Name" field.
5. Click "OK" to confirm.
6. Compile and load the configuration into the CPU.
   The device certificate of the Web server is a component of the configuration.

Creating and assigning a CA-signed Web server certificate

To create a CA-signed Web server certificate with TIA Portal, follow these steps:

1. Protect your project with the security settings "Protect this project".
   The "Security functions" appear in the project tree.
2. In the "General" tab of the Properties of the CPU Inspector window, navigate to the "Protection & Security > Certificate Manager" area and select the "Use global security settings for certificate manager" option.

NOTE

For managing certificates with the global security settings, you require the "Configure security" configuration permission.

3. Log in as a user in the project tree in the "Security settings" section. The "Administrator" role is the default for the first logon for a new project.
4. In the Inspector window Properties of the CPU, "General" tab, navigate to the "Web server > Security" area.
5. Click the "Add" button in the drop-down list to select a certificate.
   The "Create certificate" dialog opens.
6. In the follow-up dialog, select the "Signed by certificate authority" check box and select the certificate authority from the drop-down list.
7. Enter the parameters for the new certificate or confirm the default settings.
   – Select "Web server" in the "Usage" box.
   – Enter the IP address(es) of the interface(s) or the domain name of the configured CPU in the "Subject Alternative Name" field.
8. Click "OK" to confirm.
9. Compile and load the configuration in the CPU.
   The device certificate of the Web server and the CA certificate are components of the configuration.

NOTICE

Addressing the Web server of the CPU via domain names

If you enter the IP address(es) of the interface(s) of the configured CPU in the "Subject Alternative Name" field, the generated certificate may not be accepted by all Internet browsers. In addition, you must generate and load a new Web server certificate (end entity certificate) with each change of the IP address of an Ethernet interface of the CPU, since the identity of the CPU changes with the IP address.

You can avoid this problem by addressing the Web server of the CPU using domain names instead of IP address(es), e.g. "myconveyer-cpu.room13.myfactory.com". For this purpose, you have to manage the domain names of your CPU via a DNS server. Addressing via domain names is recommended especially for a configuration with reception of the IP address from a DHCP server, as in this case the assigned IP address is not known beforehand.

More information

For detailed information on local self-signed and global CA-signed certificates, on the "Public Key Infrastructure" (PKI) and on certificate management, refer to the Communications Function Manual (https://support.industry.siemens.com/cs/ww/en/view/59192925) and to the STEP 7 online help, keyword "Secure Communication".

The application example "The use of certificates with the TIA Portal (https://support.industry.siemens.com/cs/de/en/view/109769068)" includes detailed instructions on how to create a secure connection to the Web server of a SIMATIC S7-1500 CPU.

3.3.3 Managing certificates in runtime

If you manage certificates via the TIA Portal, load a certificate together with the hardware configuration into the CPU. To do this, the CPU must be in STOP mode. You cannot load a new certificate or renew an existing certificate without a RUN-STOP-RUN transition.

If you manage certificates at runtime of the CPU, loading or updating a certificate is also possible in RUN mode.

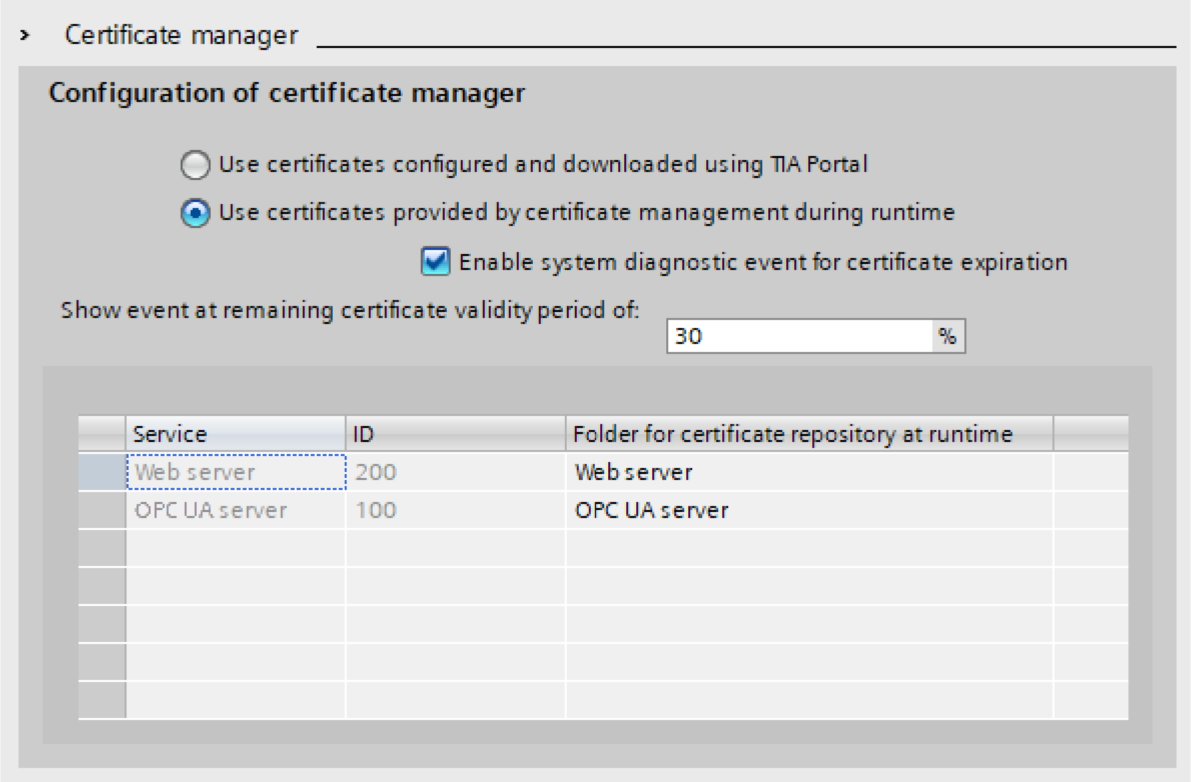
Managing the web server certificate during the CPU runtime

As of firmware version V3.0, it is also possible to transfer web server certificates to the CPU during runtime via the GDS server using OPC UA methods. The GDS server is part of the OPC UA server in the CPU. Through GDS push management functions, you can automatically update OPC UA certificates for the OPC UA server of the S7-1500 CPU.

You can find detailed information about the concept of automated certificate management with GDS (Global Discovery Services) in the Communication (https://support.industry.siemens.com/cs/ww/en/view/59192925) Function Manual.

Setting the type of certificate management

In the "Protection & Security" > "Certificate manager" category on the "General" tab of the "Properties" Inspector window, select how you want to handle certificates.



› Certificate manager

Configuration of certificate manager

○ Use certificates configured and downloaded using TIA Portal

● Use certificates provided by certificate management during runtime

☑ Enable system diagnostic event for certificate expiration

Show event at remaining certificate validity period of: 30 %

Service	ID	Folder for certificate repository at runtime
Web server	200	Web server
OPC UA server	100	OPC UA server

Figure 3-4 Configuration of the certificate manager

If you want to submit certificates via GDS at runtime, click the "Use certificates provided by the certificate management during runtime" option.

By selecting the "Enable system diagnostics event for certificate expiration" button, you specify that you want to be notified when a certificate expires. In the input field "Show event at remaining certificate validity period of:" enter a percent value. At the time this value is reached, the CPU triggers a system alarm with a maintenance request.

Example:

The certificate transferred via GDS on 01/06/2022 has a validity from 01/06/2022 to 30/06/2022 (30 days). You have input a percent value of 10 for the diagnostics event. On 27/06/2022, after 90% of the validity period has expired, the system diagnostics alarm reports that the transmitted certificate will expire on 30/06/2022.

Regardless of the configured percentage value, a message appears in any case when the validity period of a certificate expires.

NOTE

Time settings in the CPU

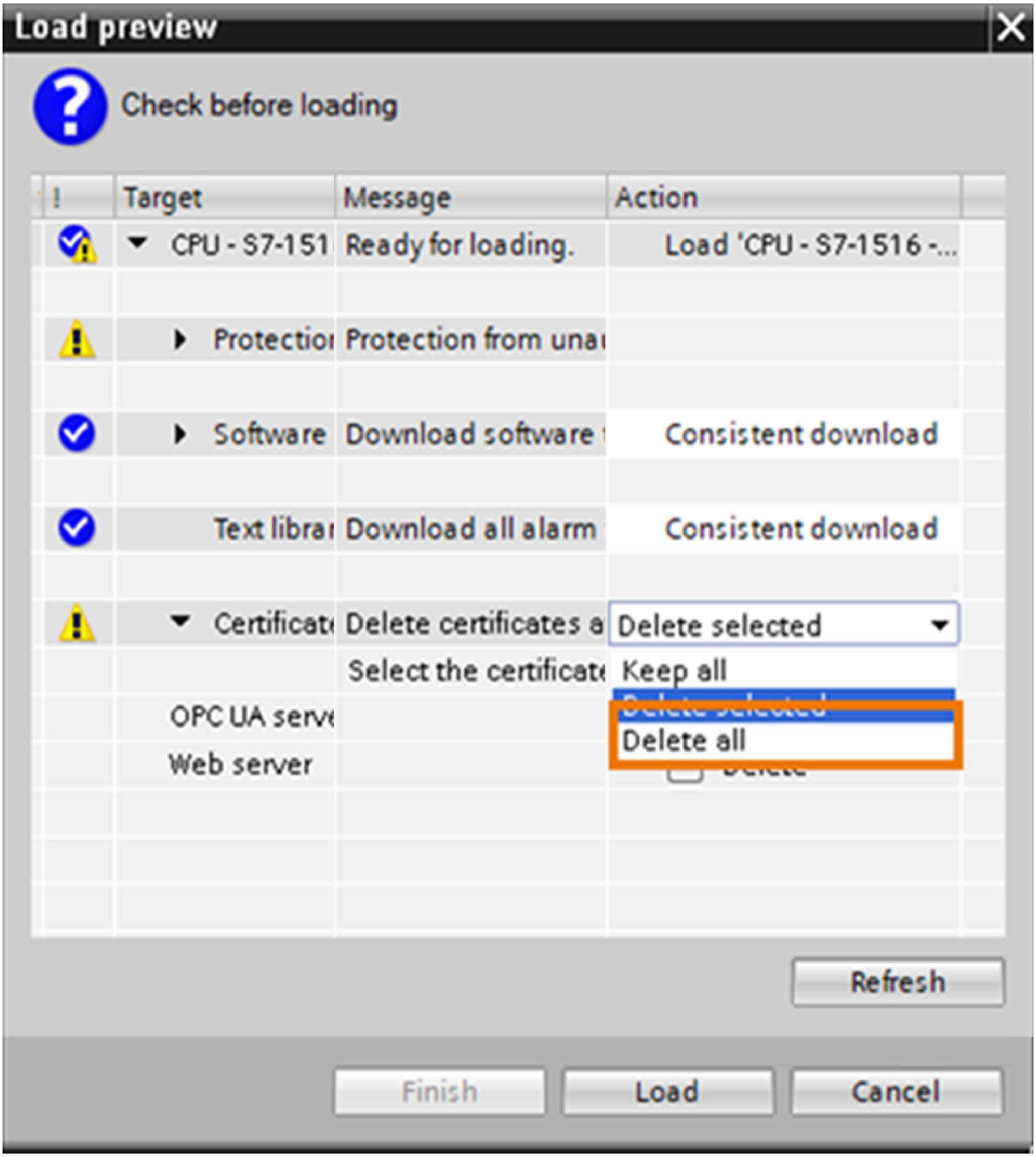
In order for the CPU to detect the expiration of a certificate, you must set the system time to Coordinated Universal Time (UTC). An incorrect system time can lead to incorrect messages regarding the expiration of certificates.

In the lower area of the "Certificate manager" category in the table, you can find a list of all CPU applications with certificates you may transfer to the CPU at runtime. In the list, the CPU applications are assigned an ID. Under the "Folder for certificate repository at runtime" column, you can find the changeable name of the certificate group.

Handling of existing certificates during loading

Before you load a project into the CPU, you may determine in the "Load preview" dialog window what should happen with the certificates of the CPU received at runtime.

As of firmware version V3.0, you can use the "Delete selected" option to delete certificates of selected CPU applications.



Load preview

Check before loading

!	Target	Message	Action
✓⚠	CPU - S7-151	Ready for loading.	Load 'CPU - S7-1516 -...
⚠	Protectio	Protection from una	
✓	Software	Download software	Consistent download
✓	Text librar	Download all alarm	Consistent download
⚠	Certificat	Delete certificates a	Delete selected
		Select the certificate	
	OPC UA serv		
	Web server		

Figure 3-5 Deleting certificates

3.4 Language settings

Introduction

The Web server provides the user interface in the following languages:

- German (Germany)
- English (USA)
- French (France)
- Italian (Italy)
- Spanish (traditional sort)
- Japanese
- Chinese (Simplified)
- Korean
- Russian
- Turkish
- Portuguese (Brazil)

Requirements for the availability of East Asian languages

The following requirements must be met for East Asian languages:

- The appropriate package for support of East Asian languages is installed on the display device (such as PC).
  For more information on installing files for East Asian languages, refer to your Windows documentation.
- STEP 7 for East Asian languages is installed on the programming device for configuring the CPU.

NOTE

SIMATIC HMI devices with the Windows CE operating system do not support East Asian languages.

Requirement for multilingual output of texts

In order for the Web server to correctly display messages, comments, and diagnostics information in the different project languages, you must assign a project language to each of the desired user interface languages of the Web server in STEP 7.

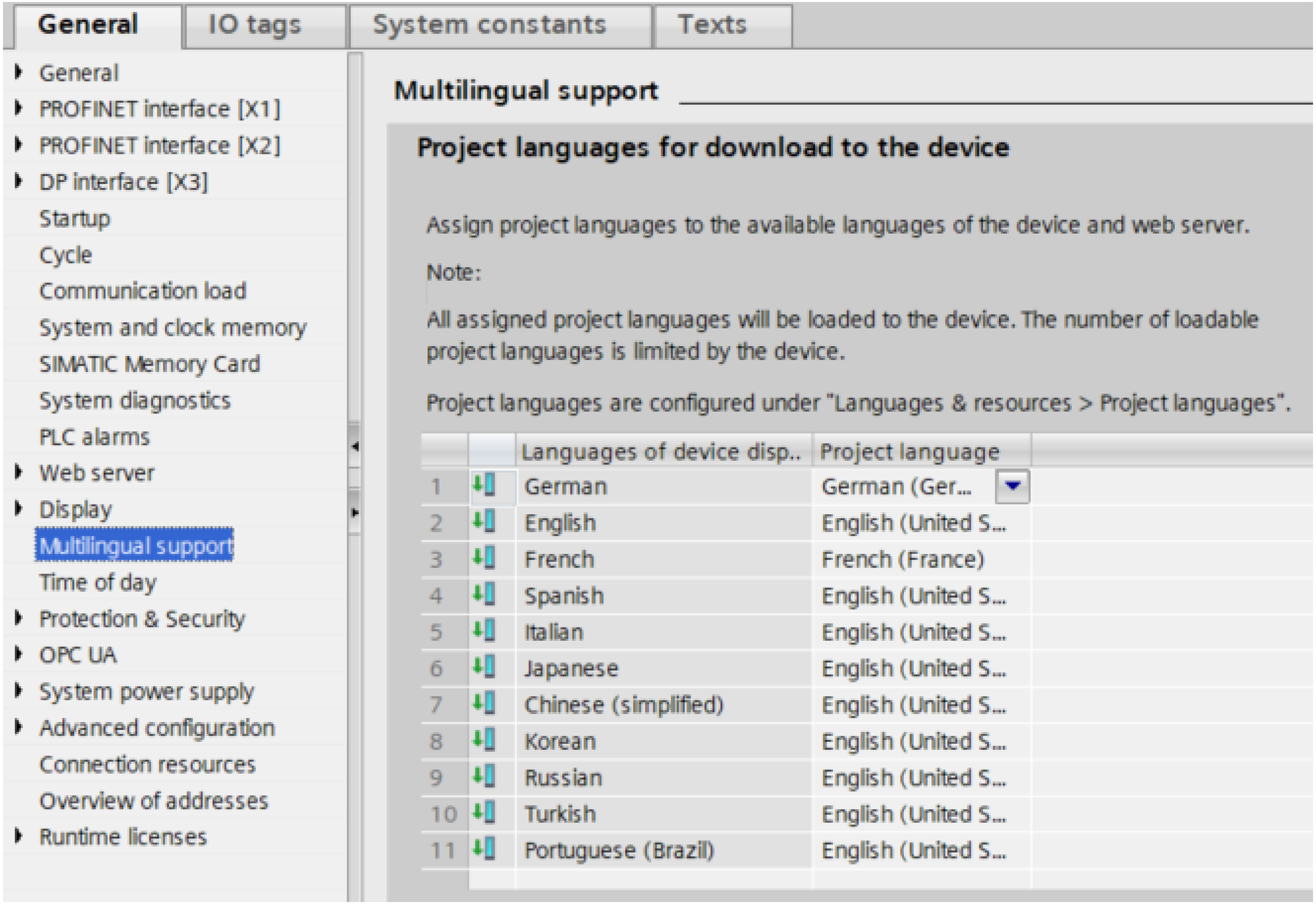
NOTE

The project languages of the STEP 7 project that you want to assign must be activated and the corresponding texts (translations) must be available in the project. You can find a list of available project languages in the project tree under "Languages & resources".

Configuring a project language for the Web server

Once you have activated the Web server on your module, assign a STEP 7 project language to each of the user interface languages of the Web server. This assignment affects the display of pages with project language-dependent texts, such as the diagnostics buffer.

1. Navigate to the "Multilingual support" area in the "General" tab of the CPU properties in the inspector window.
2. Assign a project language from the drop-down list to each user interface language of the Web server.



Multilingual support

Project languages for download to the device

Assign project languages to the available languages of the device and web server.

Note:

All assigned project languages will be loaded to the device. The number of loadable project languages is limited by the device.

Project languages are configured under "Languages & resources > Project languages".

	Languages of device disp..	Project language
1	German	German (Ger...
2	English	English (United S...
3	French	French (France)
4	Spanish	English (United S...
5	Italian	English (United S...
6	Japanese	English (United S...
7	Chinese (simplified)	English (United S...
8	Korean	English (United S...
9	Russian	English (United S...
10	Turkish	English (United S...
11	Portuguese (Brazil)	English (United S...

Figure 3-6 Language settings for the Web server in STEP 7

You can also assign user interface languages the same project language, for example:

- ① Project language German for user interface language German, English (USA) for English, French for French.
- ② Project language English (USA) for all other available user interface languages of the Web server.

In total, you can assign up to three different project languages of the STEP 7 project to the user interface languages of the Web server.

Reference

You can find more information on how to set the project language in STEP 7 in the STEP 7 online help, keyword: "Selecting project languages".

3.5 User management

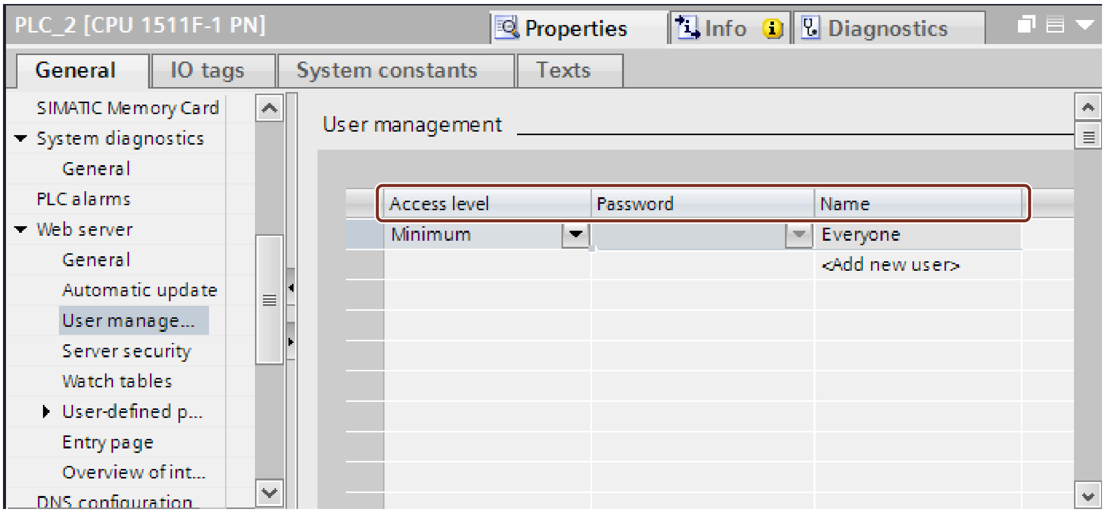
User authentication on the Web server always takes place via secure HTTPS communication.

Setting up user management

You set up the user management in the TIA Portal depending on the configured firmware version of your CPU in the project.

- The static (old) user management applies to CPUs with firmware version ≤ V3.0
- CPUs with configured firmware version ≥ V3.1 support both the static (old) user management and the local (new) user management, depending on whether an old or new project was loaded.

Static user management for CPUs with configured firmware version ≤ V3.0



User management

Access level	Password	Name
Minimum		Everyone
		<Add new user>

Figure 3-7 User management in TIA Portal for CPUs with configured firmware version ≤ V3.0

In the TIA Portal you can manage the user list in the "Web server > User management" area.

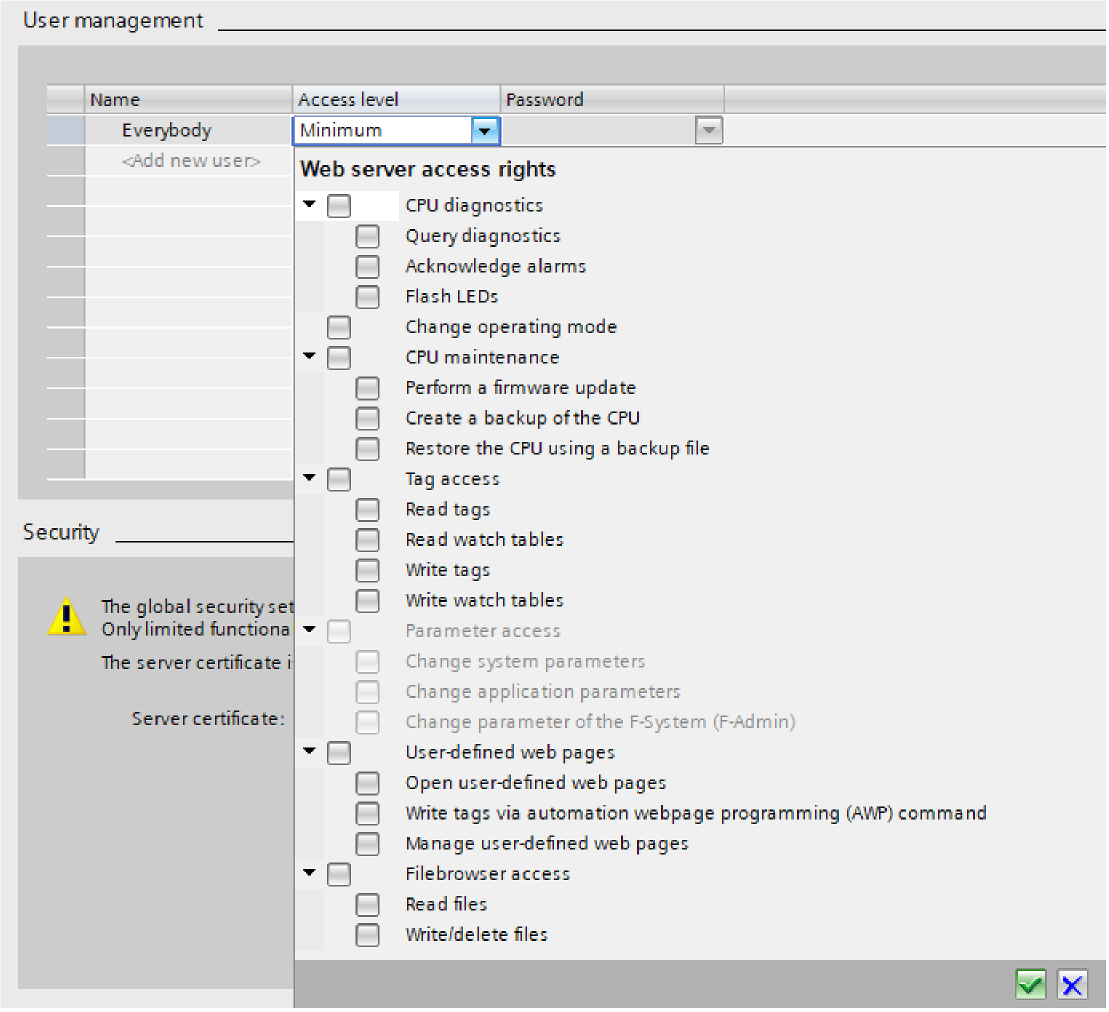
The user list provides the following options:

- Create users
- Specify access permissions
- Assign passwords

Users only have access to the options that are permanently linked to the access rights.

You can assign different user rights depending on the CPU and firmware used.

The available user rights can be available for selection in TIA Portal as follows:



User management

	Name	Access level	Password
	Everybody	Minimum	
	<Add new user>		

Web server access rights

- □ CPU diagnostics
  - □ Query diagnostics
  - □ Acknowledge alarms
  - □ Flash LEDs
- □ Change operating mode
- □ CPU maintenance
  - □ Perform a firmware update
  - □ Create a backup of the CPU
  - □ Restore the CPU using a backup file
- □ Tag access
  - □ Read tags
  - □ Read watch tables
  - □ Write tags
  - □ Write watch tables
- □ Parameter access
  - □ Change system parameters
  - □ Change application parameters
  - □ Change parameter of the F-System (F-Admin)
- □ User-defined web pages
  - □ Open user-defined web pages
  - □ Write tags via automation webpage programming (AWP) command
  - □ Manage user-defined web pages
- □ Filebrowser access
  - □ Read files
  - □ Write/delete files

Security

The global security set
Only limited functiona
The server certificate i

Server certificate:

Figure 3-8 Assignment of user rights in TIA Portal for CPUs with configured firmware version ≤ V3.0

Local user management for CPUs with configured firmware version ≥ V3.1

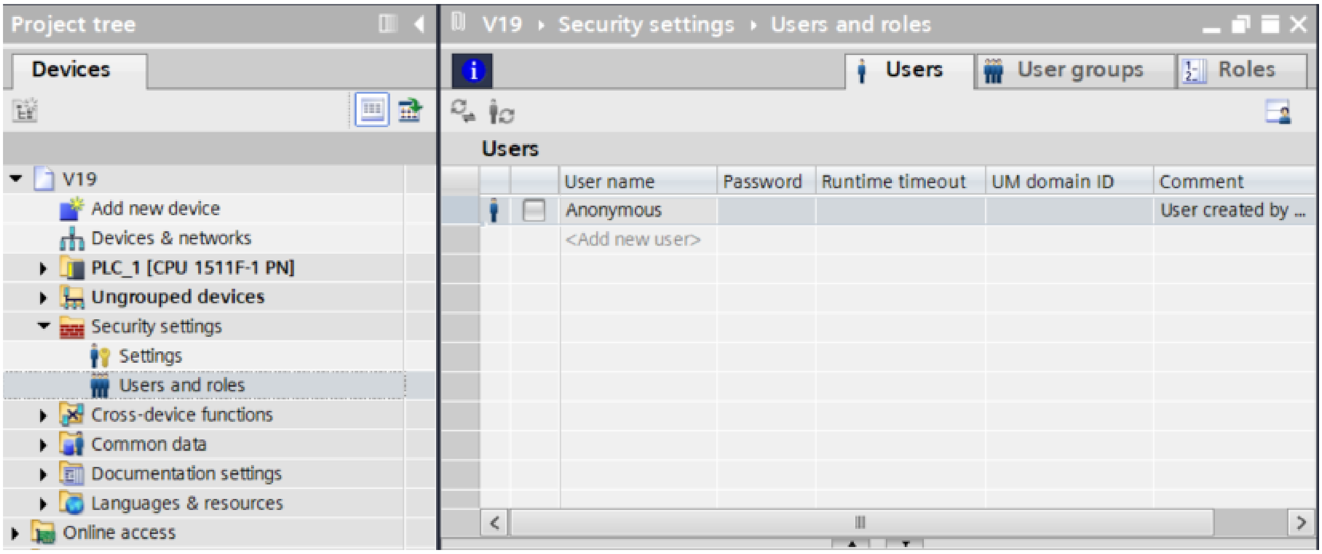
As of TIA Portal version V19 and CPU firmware version V3.1, S7-1500 CPUs feature improved management of users, roles, and CPU function rights (User Management & Access Control, UMAC). Software Controller as of version V30.1 also have this function. You manage all project users along with their rights (for example, access rights) for all CPUs in the project in the editor for users and roles of the project in the TIA Portal:

- Navigate to the "Security Settings > Users and roles" area in the project tree to manage users with their rights, for example, to control access rights.

The TIA Portal saves the assignment of the function rights of a CPU to user-defined roles and the assignment of these roles to users for each CPU. There are no system-defined roles with predefined function rights for CPUs. After loading the configuration, the user management becomes effective in the respective CPUs. After loading, every CPU "knows" who may access which service and execute specific functions.

The following settings can be made for a local project user:

- User name: Name of the local project user which must be used to log on to the project.
- Password: The password assigned by the administrator with which the project user can log on to the CPU. The project user can change the password later.
- Authentication method: For CPUs only possible via the password.
- Runtime timeout: Time span of inactivity after which a user is logged out from a device. You can use this timeout for your own web applications to have the application exit after a user-defined period of inactivity.
- Comment: Comment on the respective project user.
- Use the check box to the left of the user name to specify whether or not the project user should be downloaded to the CPU.



Users

		User name	Password	Runtime timeout	UM domain ID	Comment
	□	Anonymous				User created by ...
		<Add new user>				

Figure 3-9 User management: Users and roles

You assign the roles and rights to the user.

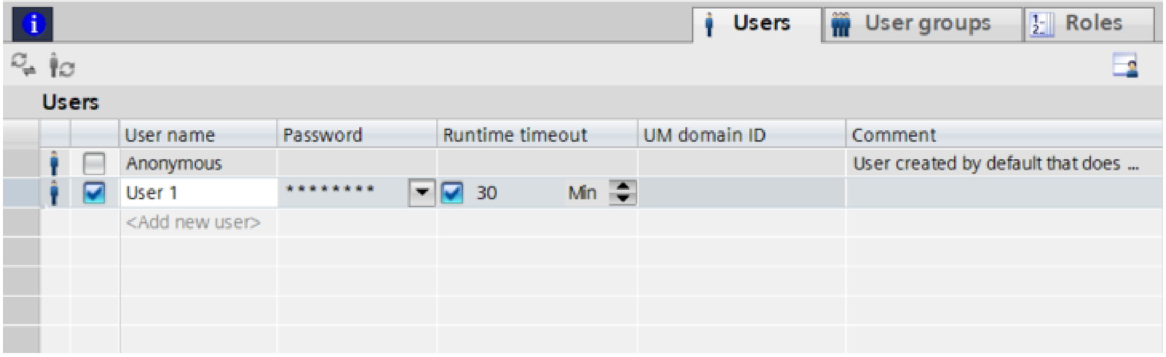
Requirement:

CPU parameter assignment: To be able to set up users, roles and function rights for a CPU, the "Enable access control" option in the "Protection & Security > Access control" tab must be selected (which is the default).

Procedure:

In the following example, you assign the "admin" role to the user "User_1" with all rights for the Web server:

1. Create a new local user in the "User" tab, in the example "User_1", and assign a password.

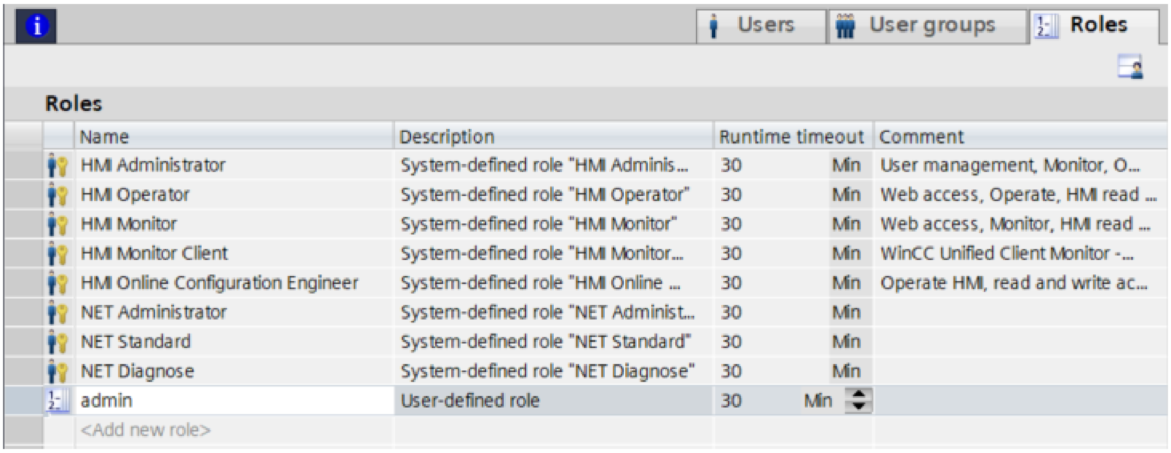


Users

		User name	Password	Runtime timeout	UM domain ID	Comment
	☐	Anonymous				User created by default that does ...
	☑	User 1	********	☑ 30 Min		
		<Add new user>				

Figure 3-10 User management: Creating a local user

2. Define one or more roles in the "Roles" tab, the role "admin" in the example.

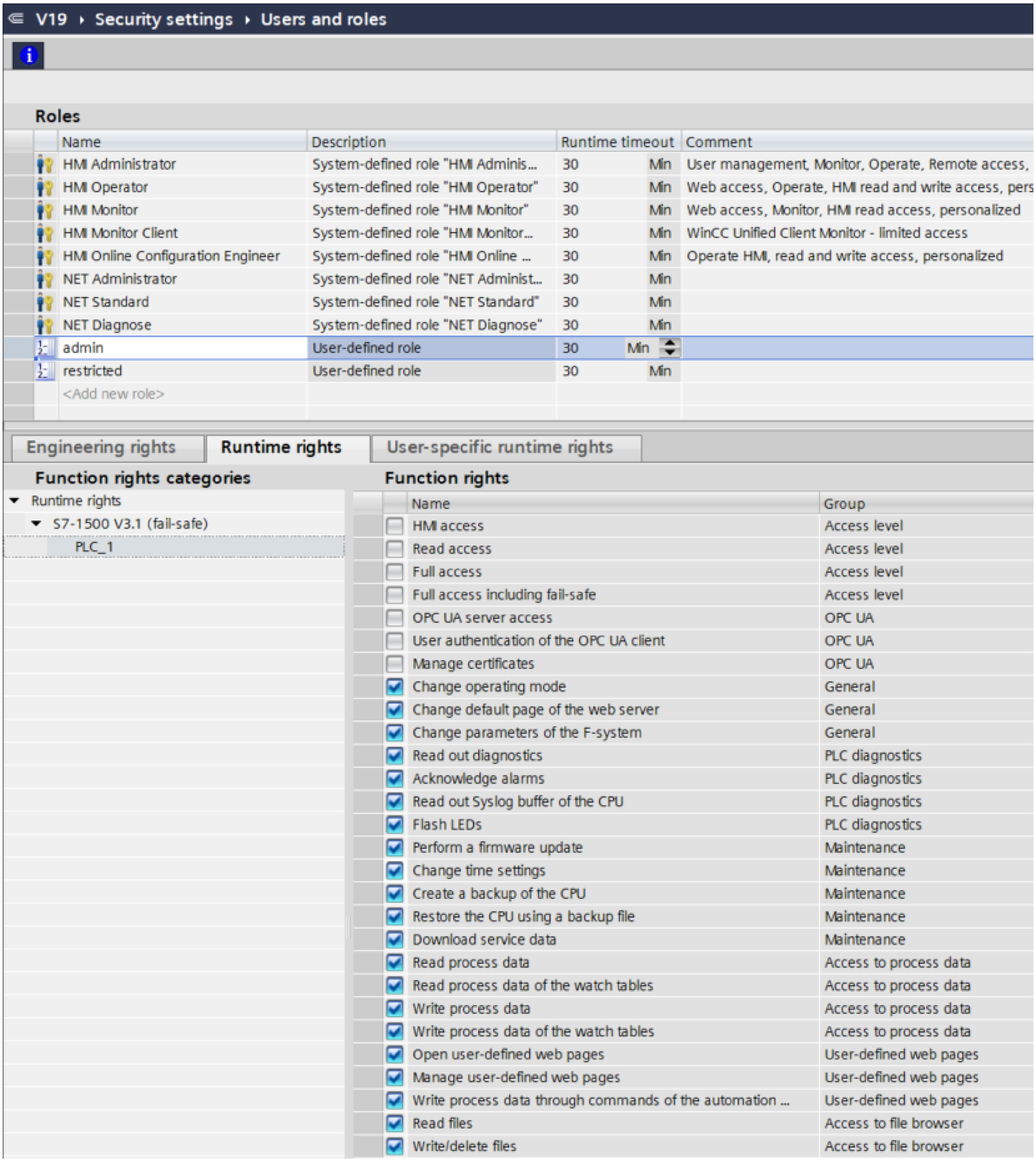


Roles

Name	Description	Runtime timeout	Comment
HMI Administrator	System-defined role "HMI Adminis...	30 Min	User management, Monitor, O...
HMI Operator	System-defined role "HMI Operator"	30 Min	Web access, Operate, HMI read ...
HMI Monitor	System-defined role "HMI Monitor"	30 Min	Web access, Monitor, HMI read ...
HMI Monitor Client	System-defined role "HMI Monitor...	30 Min	WinCC Unified Client Monitor -...
HMI Online Configuration Engineer	System-defined role "HMI Online ...	30 Min	Operate HMI, read and write ac...
NET Administrator	System-defined role "NET Administ...	30 Min	
NET Standard	System-defined role "NET Standard"	30 Min	
NET Diagnose	System-defined role "NET Diagnose"	30 Min	
admin	User-defined role	30 Min	
<Add new role>			

Figure 3-11 User management: Defining a role

3. Assign function rights for the Web server to the "admin" role. To do this, switch to the "Runtime rights" tab and select all function rights for the Web server for the example.



V19 ▸ Security settings ▸ Users and roles

Roles

Name	Description	Runtime timeout		Comment
HMI Administrator	System-defined role "HMI Adminis...	30	Min	User management, Monitor, Operate, Remote access,
HMI Operator	System-defined role "HMI Operator"	30	Min	Web access, Operate, HMI read and write access, pers
HMI Monitor	System-defined role "HMI Monitor"	30	Min	Web access, Monitor, HMI read access, personalized
HMI Monitor Client	System-defined role "HMI Monitor...	30	Min	WinCC Unified Client Monitor - limited access
HMI Online Configuration Engineer	System-defined role "HMI Online ...	30	Min	Operate HMI, read and write access, personalized
NET Administrator	System-defined role "NET Administ...	30	Min	
NET Standard	System-defined role "NET Standard"	30	Min	
NET Diagnose	System-defined role "NET Diagnose"	30	Min	
admin	User-defined role	30	Min	
restricted	User-defined role	30	Min	
<Add new role>				

Engineering rights | Runtime rights | User-specific runtime rights

Function rights categories

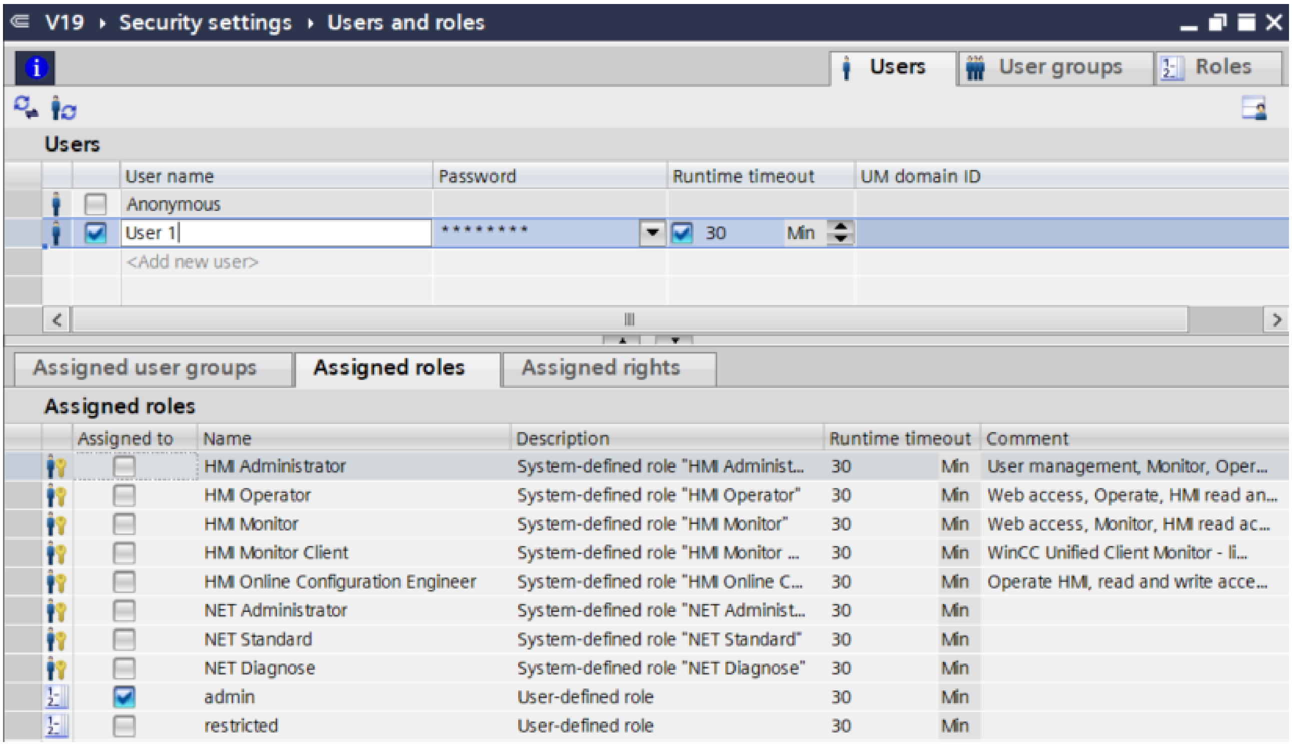
- Runtime rights
  - S7-1500 V3.1 (fail-safe)
    - PLC_1

Function rights

	Name	Group
□	HMI access	Access level
□	Read access	Access level
□	Full access	Access level
□	Full access including fail-safe	Access level
□	OPC UA server access	OPC UA
□	User authentication of the OPC UA client	OPC UA
□	Manage certificates	OPC UA
☑	Change operating mode	General
☑	Change default page of the web server	General
☑	Change parameters of the F-system	General
☑	Read out diagnostics	PLC diagnostics
☑	Acknowledge alarms	PLC diagnostics
☑	Read out Syslog buffer of the CPU	PLC diagnostics
☑	Flash LEDs	PLC diagnostics
☑	Perform a firmware update	Maintenance
☑	Change time settings	Maintenance
☑	Create a backup of the CPU	Maintenance
☑	Restore the CPU using a backup file	Maintenance
☑	Download service data	Maintenance
☑	Read process data	Access to process data
☑	Read process data of the watch tables	Access to process data
☑	Write process data	Access to process data
☑	Write process data of the watch tables	Access to process data
☑	Open user-defined web pages	User-defined web pages
☑	Manage user-defined web pages	User-defined web pages
☑	Write process data through commands of the automation ...	User-defined web pages
☑	Read files	Access to file browser
☑	Write/delete files	Access to file browser

Figure 3-12 User management: Assigning rights to a role

4. Assign one or more roles to the user "User_1" in the tab "Assigned roles". In the example, you assign the "admin" role to the user "User_1". The roles and rights assigned to a user can be found in the corresponding tabs.



Users

	User name	Password	Runtime timeout		UM domain ID
□	Anonymous				
☑	User 1	********	☑ 30	Min	
	<Add new user>				

Assigned roles

Assigned to	Name	Description	Runtime timeout		Comment
□	HMI Administrator	System-defined role "HMI Administ...	30	Min	User management, Monitor, Oper...
□	HMI Operator	System-defined role "HMI Operator"	30	Min	Web access, Operate, HMI read an...
□	HMI Monitor	System-defined role "HMI Monitor"	30	Min	Web access, Monitor, HMI read ac...
□	HMI Monitor Client	System-defined role "HMI Monitor ...	30	Min	WinCC Unified Client Monitor - li...
□	HMI Online Configuration Engineer	System-defined role "HMI Online C...	30	Min	Operate HMI, read and write acce...
□	NET Administrator	System-defined role "NET Administ...	30	Min	
□	NET Standard	System-defined role "NET Standard"	30	Min	
□	NET Diagnose	System-defined role "NET Diagnose"	30	Min	
☑	admin	User-defined role	30	Min	
□	restricted	User-defined role	30	Min	

Figure 3-13 User management: Assigned role

5. Download the configuration to the CPU.

You can find more information on roles and rights in the online help for TIA Portal, keyword "Basics on user management in TIA Portal".

Deactivating the user management

For CPUs with configured firmware version ≥ V3.1, you can disable access control in the hardware configuration in the CPU properties under "Protection & Security" > "Access control".

There is no user management in the project with deactivated access control. Users who are defined under "Users and roles" are not considered and authentication is not possible. The CPU itself creates an "Anonymous" user that has full access rights to the functions of the Web server and the CPU.

User name "Everybody"/"Anonymous"

As a non-logged-in user, you always access web pages as "Everybody" or "Anonymous" by default. It does not matter in this case whether you have configured additional users. The user name depends on the configured firmware version of the CPU:

- For CPUs with configured firmware version ≤ V3.0: Static user management with the user "Everybody"
- For CPUs with configured firmware version ≥ V3.1: Local user management with the user "Anonymous"

The user "Everybody" or "Anonymous" has minimum access authorizations. These are read-only access to the intro page and start page. The user "Everybody"/"Anonymous" is defined without assigning a password, but you can assign all access authorizations available in TIA Portal to it.

Number of users:

- Static user management: Minimum 1 "Everybody" user and maximum 20 users can be created
- Local user management: Minimum 1 "Anonymous" user and maximum 100 users can be created

If the "Anonymous" user has not been activated under "Users and roles", an anonymous user without authorizations is available automatically on the CPU.

Because the user "Everybody"/"Anonymous" is defined in TIA Portal without assignment of a password, be careful of the access authorizations that you assign to this user.

Certain authorizations, such as the possibility of changing the operating mode, could represent a security risk.

For the assignment of security-relevant authorizations, we recommend that you create a user with password protection in TIA Portal.

⚠ WARNING

For an F-CPU, do **not** assign the access authorization "Perform changes as F-Admin" to the user "Everybody"/"Anonymous".

Make sure that you observe the warnings relating to this in the section "Restoring a backup of the safety program to an S7-300/1500 F-CPU" in the manual SIMATIC Safety - Configuring and Programming (https://support.automation.siemens.com/WW/view/en/54110126).

Passwords should always be more than 8 characters in length and contain uppercase and lowercase characters as well as special characters and numbers (?!+%$1234...). Computer keyboard character strings and words from the dictionary are unsuitable. Change the password regularly.

You can read the password policies of the CPU as of firmware version V3.1 with the Api.GetPasswordPolicy method. In the TIA Portal you can read the policies and change these in the project tree via "Security settings" > "Settings" > "Password policies". You can find more information in the online help for the TIA Portal.

NOTE

When assigning rights, note that read and write access to tags of the "Watch tables (Page 95)" and "Tag status (Page 93)" web pages is retained, even if you have deactivated the attribute "Accessible from HMI/OPC UA/Web API" in the PLC tag table when configuring the data block in STEP 7.

3.6 Updating and saving information

Updating the screen content

Automatic updating is activated in the default setting. The preset update time is 10 s.
You update the web pages manually via the function key <F5>.

Disabling automatic updating for an individual web page

Click Off to temporarily deactivate automatic updating for a web page.
Note that the deactivation affects only the currently visited web page. Automatic updating is activated again when you change to a different web page.
You reactivate automatic updating by clicking On.

NOTE

If the load on the CPU is very high during operation, for example, due to a large number of PROFINET interrupts or extensive communication jobs, the updating of web pages may be significantly delayed for the duration of this high CPU load.

Printing web pages

The Web server offers you a print preview on most web pages. Click the symbol 🖨 to open it.
Created printouts always contain the current information in the CPU. This means that it is possible that the information in the print preview is more up-to-date than the information in the standard view.

Web pages

4

4.1 Start page with general CPU information

Connecting to the Web server

Establish a connection to the Web server by entering the IP address of the interface of the configured CPU which is connected to the client in the address bar of the web browser, for example, https://192.168.3.141. The connection is set up and the "Intro" page opens.

NOTE

Web pages for R/H-CPUs

Note that the web pages are not available for R/H-CPUs. If you enter the IP address in the address bar of the browser, the error message "404 NOT FOUND" appears.

The examples in the next section provide information about the different web pages.

Intro

The figure below shows the first page (Intro) called by the Web server.

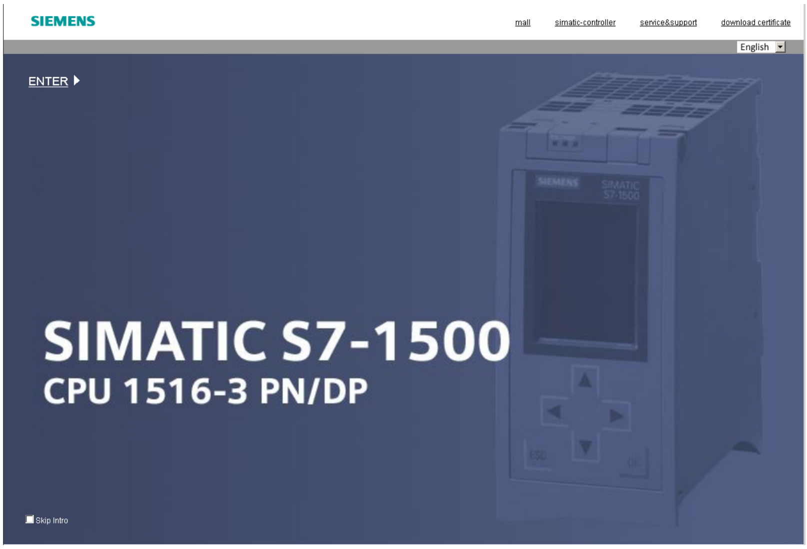


Figure 4-1 Introduction page of the Web server of the CPU 1516-3 PN/DP

Click the NEXT link to go to the Web server pages.

NOTE

Select the "Skip Intro" check box in order to skip the intro. The Web server will then take you directly to its start page in future. This setting is saved in the user profile of the current PC user.

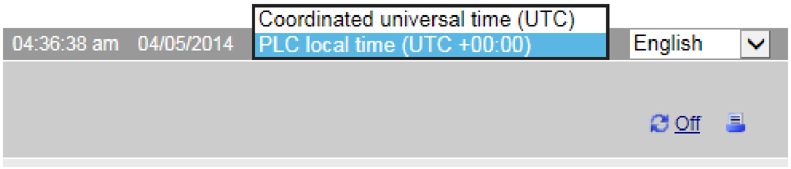
You can undo the setting "Skip Intro" by clicking the "Intro" link in the left-hand navigation bar of a web page.

Setting the user interface language

You can change the language for the Web server interface, e.g., from English to German, in the upper right corner. This option is available to you on all web pages of the Web server.

Switching the time display

You can set the format of the time display to Coordinated Universal Time (UTC) or PLC local time (default setting) on the left next to the language setting.



04:36:38 am 04/05/2014

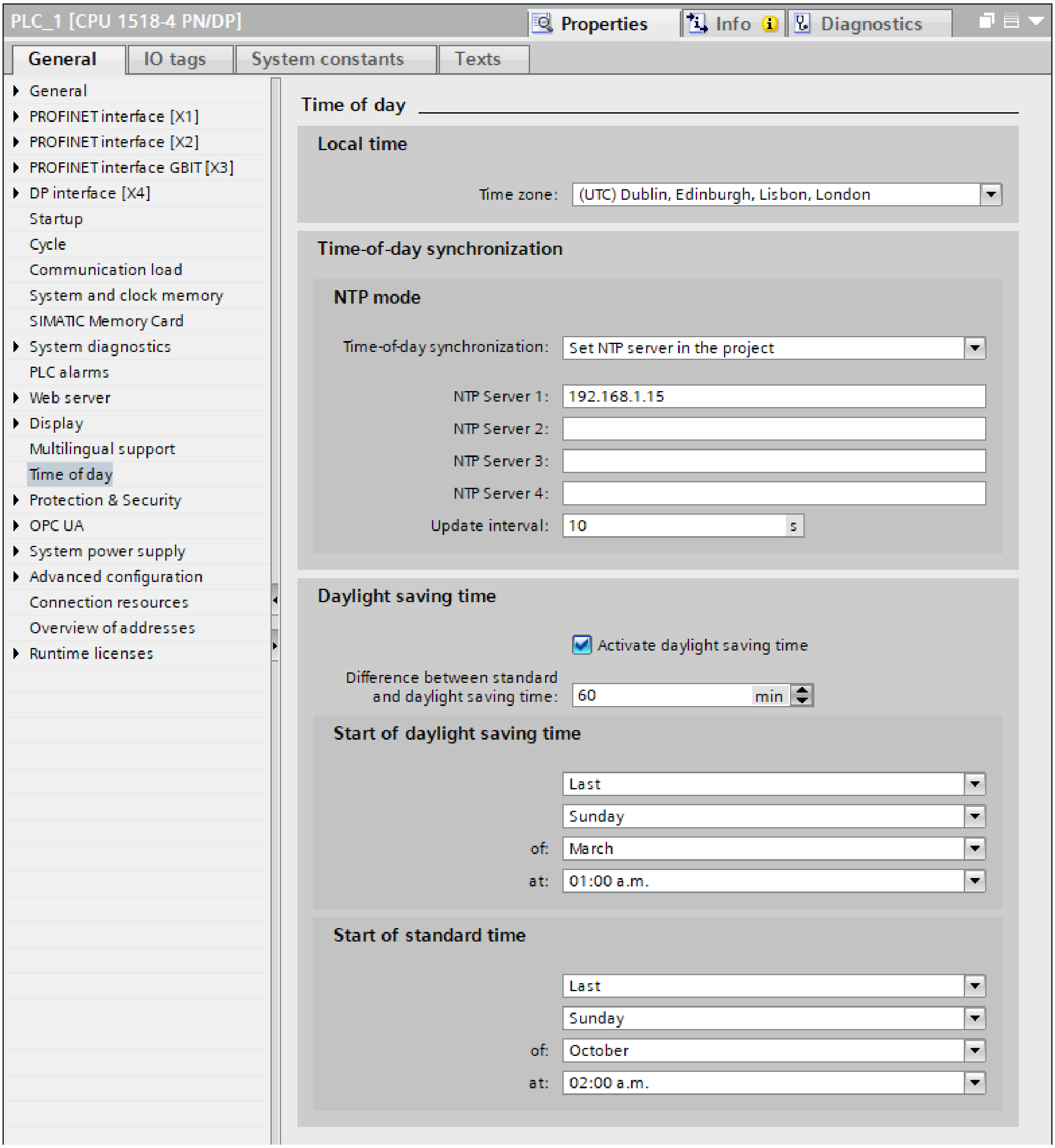
Coordinated universal time (UTC)
PLC local time (UTC +00:00)

English

Figure 4-2 Switching the time display

You can switch the time display on all the Web pages that provide this drop-down list.

The displayed PLC local time comes from the time zone set in the CPU properties without consideration of the daylight saving / standard time setting.



Time of day

Local time

Time zone: (UTC) Dublin, Edinburgh, Lisbon, London

Time-of-day synchronization

NTP mode

Time-of-day synchronization: Set NTP server in the project

NTP Server 1: 192.168.1.15

NTP Server 2:

NTP Server 3:

NTP Server 4:

Update interval: 10 s

Daylight saving time

☑ Activate daylight saving time

Difference between standard and daylight saving time: 60 min

Start of daylight saving time

Last

Sunday

of: March

at: 01:00 a.m.

Start of standard time

Last

Sunday

of: October

at: 02:00 a.m.

Figure 4-3 Setting the time in the CPU properties

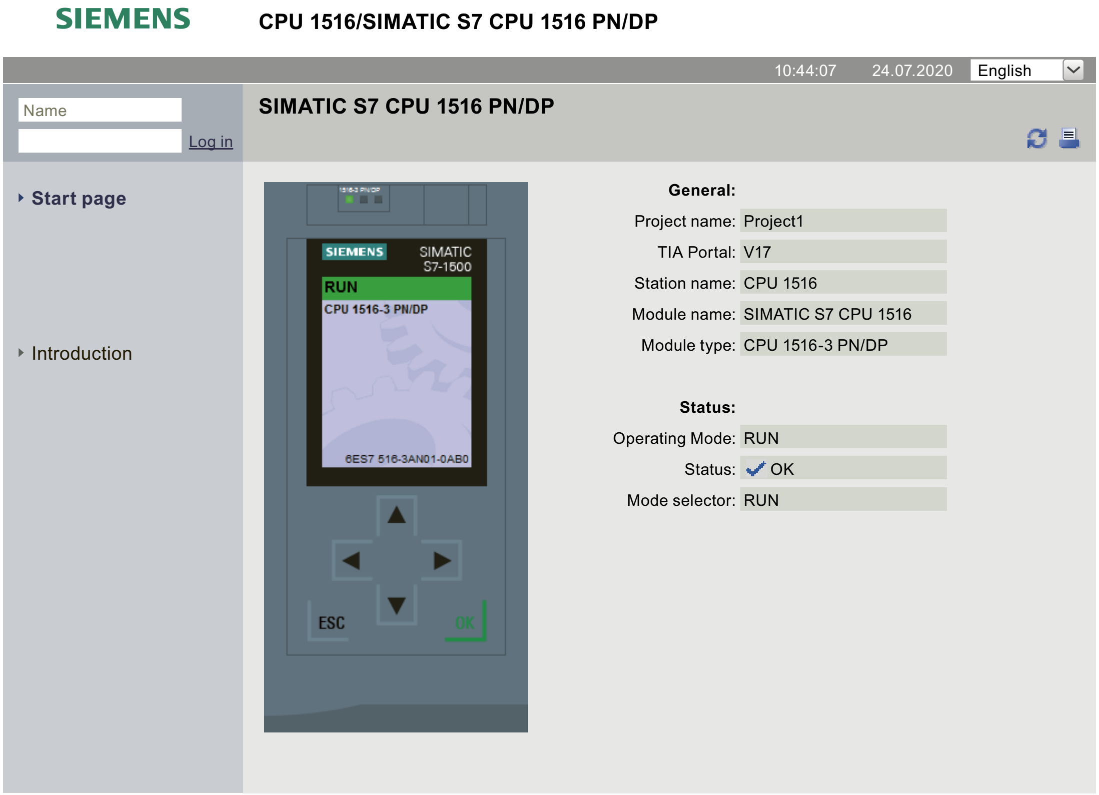
The switchover has an effect on the following Web pages:

Table 4-1 Switching the time display: Display on Web pages

Web pages	Display as Coordinated Universal Time (UTC) or as PLC local time
Start page	Last F-change
Diagnostics buffer	Date and time of the diagnostic buffer entry
Alarms	Date and time of the alarms
Online backup	Backup file with date and time of the backup
DataLogs	Date of change and time of change
File browser	Date of change and time of change
User files	Date of change and time of change
Save service data	File with time stamp of the storage

Start page

The start page before login offers information as shown in the figure below. The image of the CPU with LEDs shows its current status at the time of the data request.



SIEMENS CPU 1516/SIMATIC S7 CPU 1516 PN/DP

10:44:07 24.07.2020 English

Name

Log in

SIMATIC S7 CPU 1516 PN/DP

- Start page
- Introduction

General:

Project name: Project1

TIA Portal: V17

Station name: CPU 1516

Module name: SIMATIC S7 CPU 1516

Module type: CPU 1516-3 PN/DP

Status:

Operating Mode: RUN

Status: ✓ OK

Mode selector: RUN

Figure 4-4 General information before login

Log in

To use the full functionality of the web pages, you must be logged in. Log in with a user name and password specified in the Web configuration in STEP 7. You now have corresponding permissions to access the web pages released for this user. If you have not configured a user, read-only access is granted to intro and start pages by default.

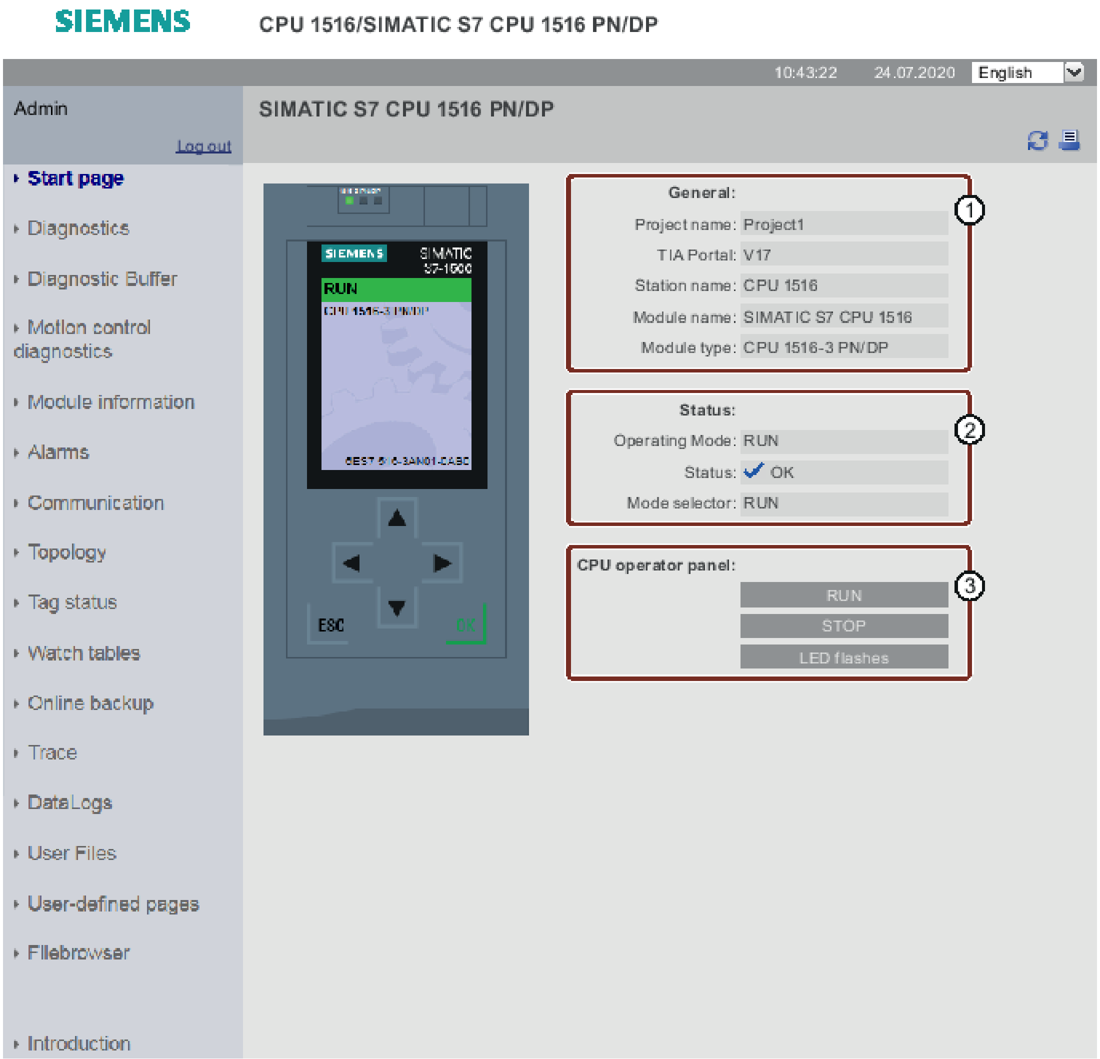
NOTE

After carrying out your required actions, log out explicitly from the Web server by clicking "Logout" in order to minimize the risk of unauthorized external access.

NOTE

Session timeout

The timeout for each started session is 30 minutes. After each update/automatic update, the session is automatically extended by another 30 minutes.



SIEMENS CPU 1516/SIMATIC S7 CPU 1516 PN/DP

10:43:22 24.07.2020 English

SIMATIC S7 CPU 1516 PN/DP

General:
Project name: Project1
TIA Portal: V17
Station name: CPU 1516
Module name: SIMATIC S7 CPU 1516
Module type: CPU 1516-3 PN/DP

Status:
Operating Mode: RUN
Status: ✓ OK
Mode selector: RUN

CPU operator panel:

Figure 4-5 Start page after login

① "General"

"General" contains information about the CPU whose Web server you are currently connected to, as well as the project name and the version of the TIA Portal with which the CPU was configured. The displayed TIA Portal version is at least required to load or edit the entire project.

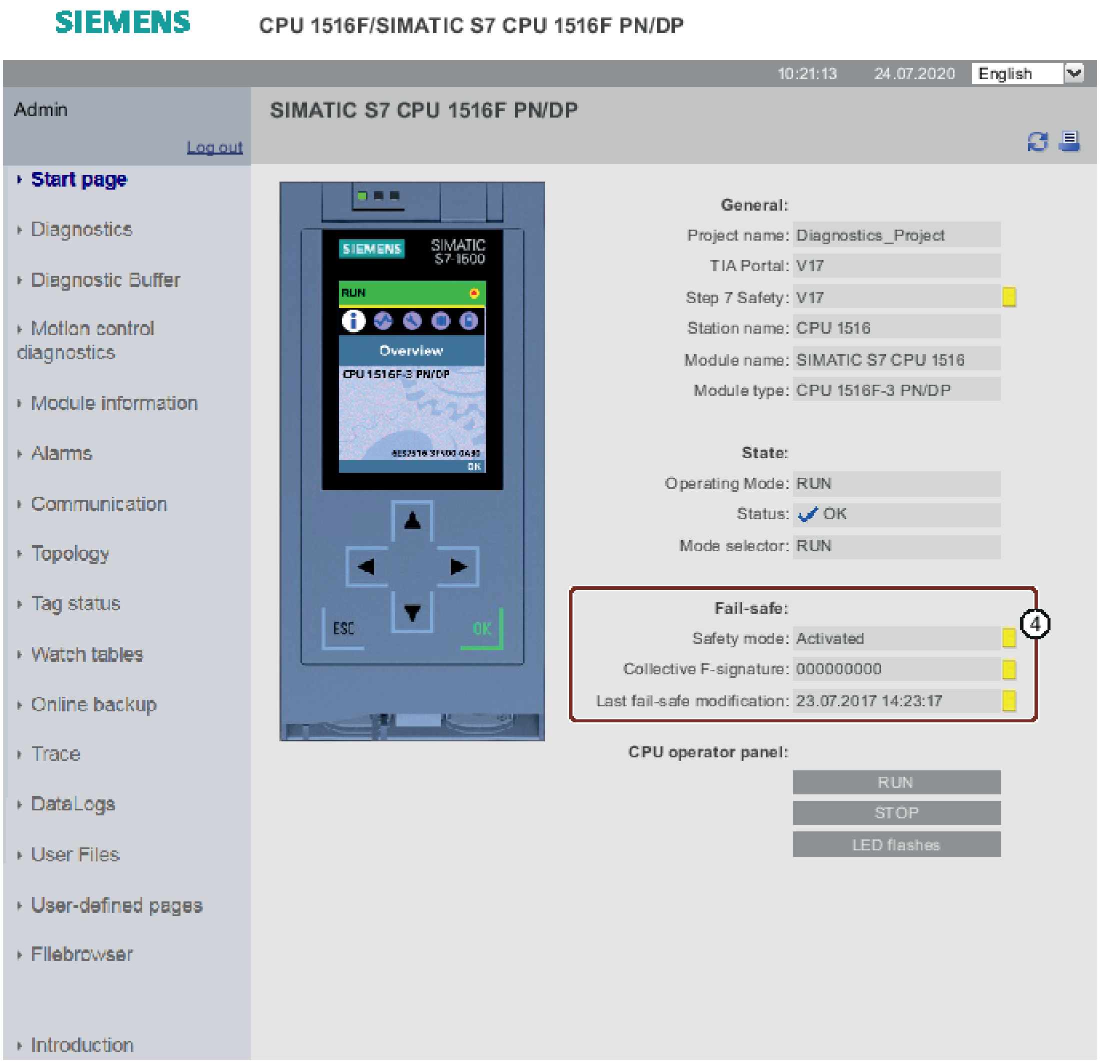
② "Status"

"Status" contains information about the CPU status at the time of the query.

③ "CPU operator panel"

In the area "CPU operator panel" you have the possibility to change the operating mode of the CPU ("RUN"/"STOP" buttons) or to have the LEDs blink ("LED blink" button) with corresponding access rights.

Additional information for F-CPUs



General:
Project name: Diagnostics_Project
TIA Portal: V17
Step 7 Safety: V17
Station name: CPU 1516
Module name: SIMATIC S7 CPU 1516
Module type: CPU 1516F-3 PN/DP

State:
Operating Mode: RUN
Status: ✓ OK
Mode selector: RUN

Fail-safe:
Safety mode: Activated
Collective F-signature: 000000000
Last fail-safe modification: 23.07.2017 14:23:17

Figure 4-6 Start page after login to an F-CPU

④ "Fail-safe"

"Fail-safe" contains additional information on the F-CPU. Further information about the specification is available in the Programming and Operating Manual SIMATIC Safety - Configuring and Programming (https://support.industry.siemens.com/cs/de/de/view/54110126/en?dl=en).

Reference

You can find additional information in the section Configuring the Web server (Page 26).

4.2 Diagnostics

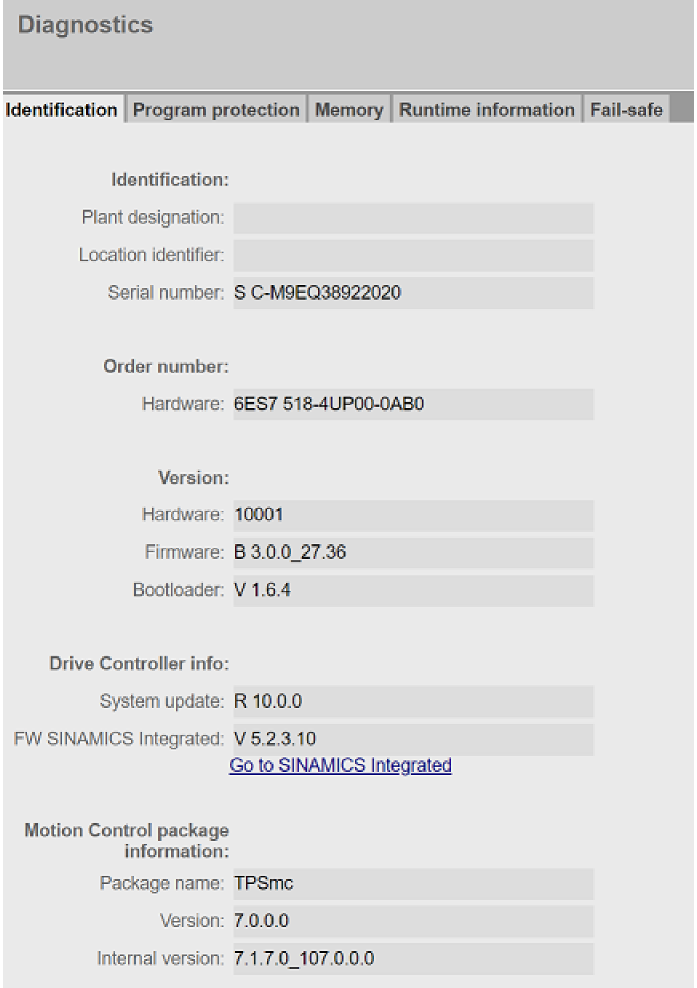
Overview

The "Diagnostics" web page provides more information about the tabs:

- Identification
- Program protection
- Memory
- Runtime information
- Fail-safe (with an F CPU)

"Identification" tab

The CPU characteristics are available in the "Identification" tab.



Identification:

Plant designation:

Location identifier:

Serial number: S C-M9EQ38922020

Order number:

Hardware: 6ES7 518-4UP00-0AB0

Version:

Hardware: 10001

Firmware: B 3.0.0_27.36

Bootloader: V 1.6.4

Drive Controller info:

System update: R 10.0.0

FW SINAMICS Integrated: V 5.2.3.10

Go to SINAMICS Integrated

Motion Control package information:

Package name: TPSmc

Version: 7.0.0.0

Internal version: 7.1.7.0_107.0.0.0

Figure 4-7 "Identification" tab

"Identification"

The "Identification" info box contains the plant and location designation and the serial number. Plant and location designations can be configured in STEP 7 in the properties dialog of the CPU in the "General" tab.

"Order number"

The "Order number" info box contains the order number for the hardware.

"Version"

You can find the information on hardware version, firmware version on the CPU and bootloader version in the "Version" info box.

"Drive Controller Info"

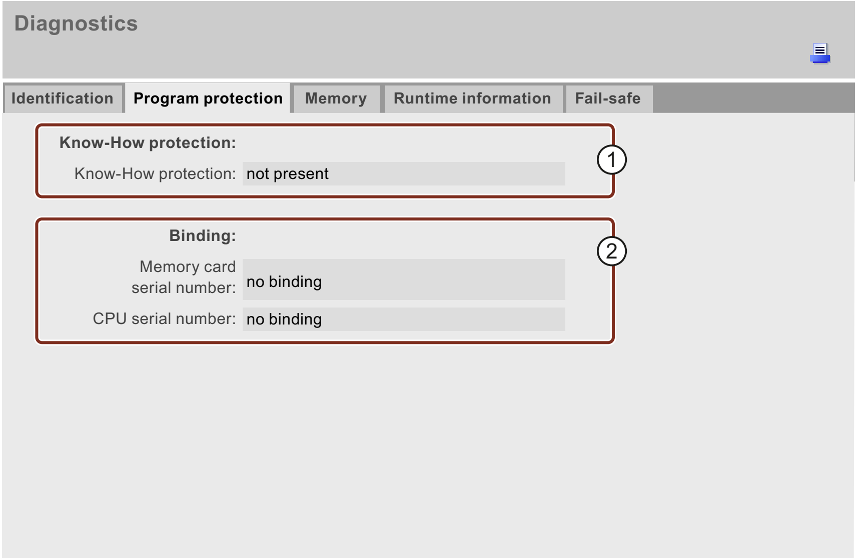
If you are using a SIMATIC drive controller, you will find information about the hardware update version of the drive controller, the firmware version of SINAMICS Integrated, and a link to the readable information of SINAMICS Integrated.
You can find more information in the SIMATIC Drive Controller (https://support.industry.siemens.com/cs/ww/en/view/109766665) System Manual.

"Motion control package information"

Display of the name and version of the Motion Control technology package used (Technology Package Standard Motion Control or Technology Package Motion Control KinPlus).
For more information, refer to the SIMATIC S7-1500 S7-1500T Kinematics Functions (https://support.industry.siemens.com/cs/ww/en/view/109781850) manual.

"Program protection" tab

The "Program protection" tab provides information on whether the PLC program contains know-how protection or copy protection.



Know-How protection:
Know-How protection: not present

Binding:
Memory card serial number: no binding
CPU serial number: no binding

Figure 4-8 "Program protection" tab

"①Know-how protection"

Information on whether the PLC program contains at least one block with know-how protection or not can be found in the info field "Know-how protection".

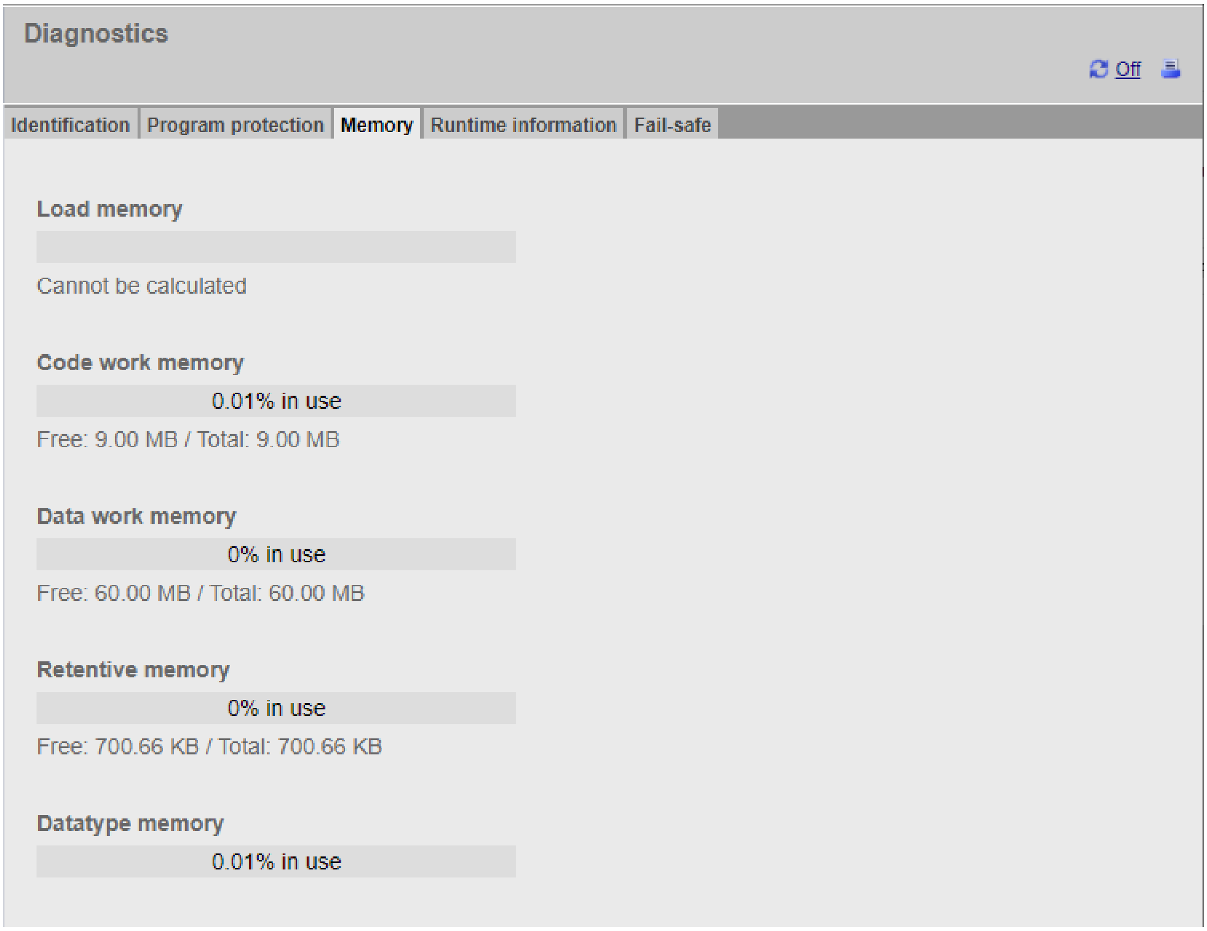
②"Binding"

In the info field "Binding" you can find information on whether copy protection has been activated by binding at least one program block of the PLC program to the serial number of the CPU or memory card.

- "Binding"
- "No binding"
- "Binding mismatch": At least one block is bound to a different serial number (load process is rejected)

"Memory" tab

The "Memory" tab contains current values on the memory currently in use.



Load memory

Cannot be calculated

Code work memory

0.01% in use

Free: 9.00 MB / Total: 9.00 MB

Data work memory

0% in use

Free: 60.00 MB / Total: 60.00 MB

Retentive memory

0% in use

Free: 700.66 KB / Total: 700.66 KB

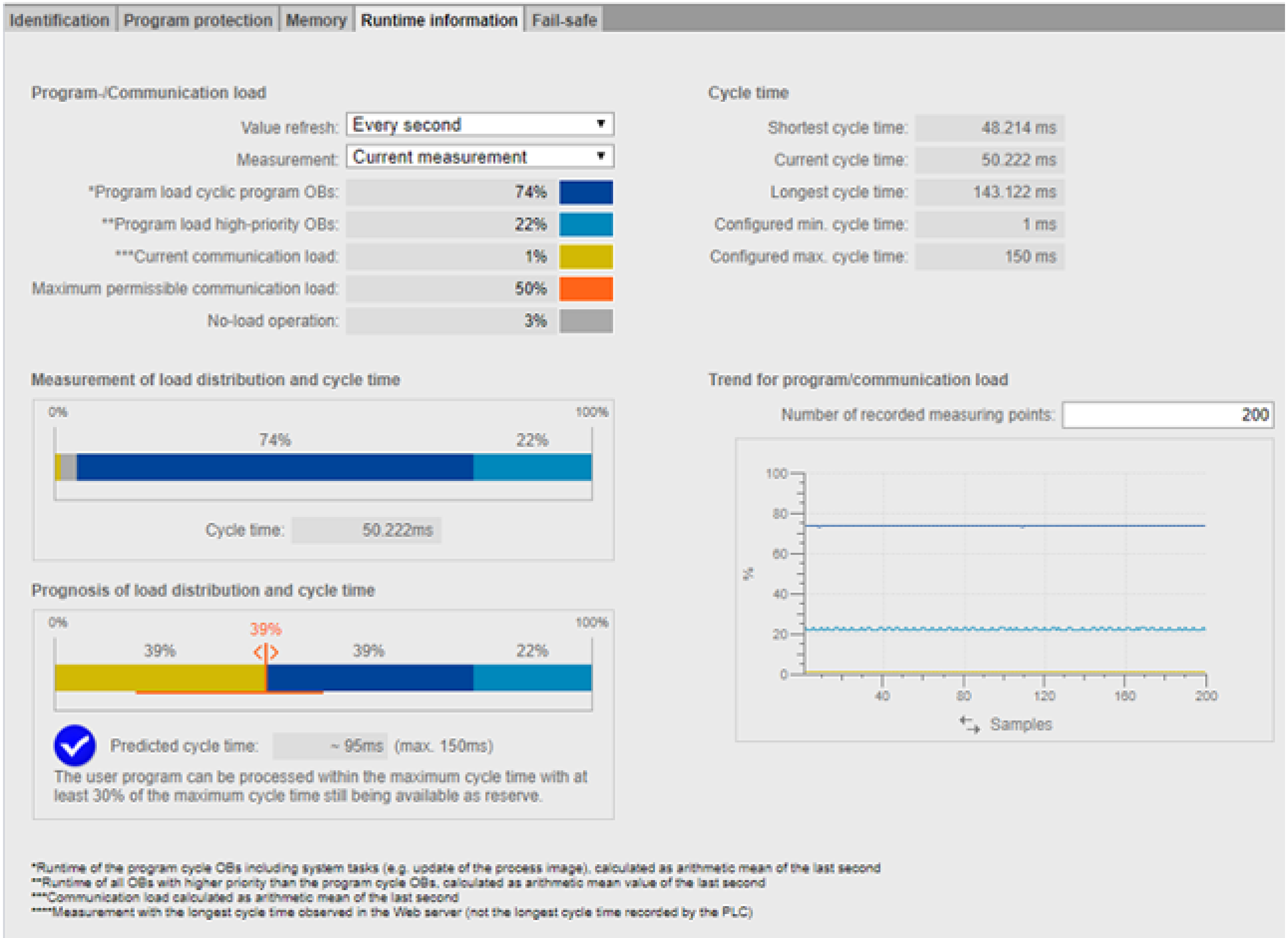
Datatype memory

0.01% in use

Figure 4-9 "Memory" tab

"Runtime information" tab

Current information on program/communication load and cycle time can be found in the "Runtime information" tab. This enables you to see whether there may be runtime problems during execution of your user program.



Program-/Communication load

Value refresh: Every second

Measurement: Current measurement

*Program load cyclic program OBs: 74%

**Program load high-priority OBs: 22%

***Current communication load: 1%

Maximum permissible communication load: 50%

No-load operation: 3%

Cycle time

Shortest cycle time: 48.214 ms

Current cycle time: 50.222 ms

Longest cycle time: 143.122 ms

Configured min. cycle time: 1 ms

Configured max. cycle time: 150 ms

Measurement of load distribution and cycle time

Cycle time: 50.222ms

Trend for program/communication load

Number of recorded measuring points: 200

Prognosis of load distribution and cycle time

Predicted cycle time: ~ 95ms (max. 150ms)

The user program can be processed within the maximum cycle time with at least 30% of the maximum cycle time still being available as reserve.

*Runtime of the program cycle OBs including system tasks (e.g. update of the process image), calculated as arithmetic mean of the last second
**Runtime of all OBs with higher priority than the program cycle OBs, calculated as arithmetic mean value of the last second
***Communication load calculated as arithmetic mean of the last second
****Measurement with the longest cycle time observed in the Web server (not the longest cycle time recorded by the PLC)

Figure 4-10 "Runtime information" tab

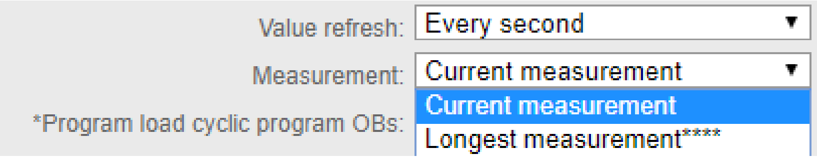
Program-/Communication load

With the "Value refresh" function, you update the data displayed in the bar charts:

- At intervals of 1 second
- Automatic (as configured in STEP 7)

With the "Measurement" function, you can decide which measurement the bar charts display. You can choose between:

- The current measurement
- The measurement of the longest cycle time



Value refresh: Every second

Measurement: Current measurement

*Program load cyclic program OBs:

Figure 4-11 Program-/Communication load

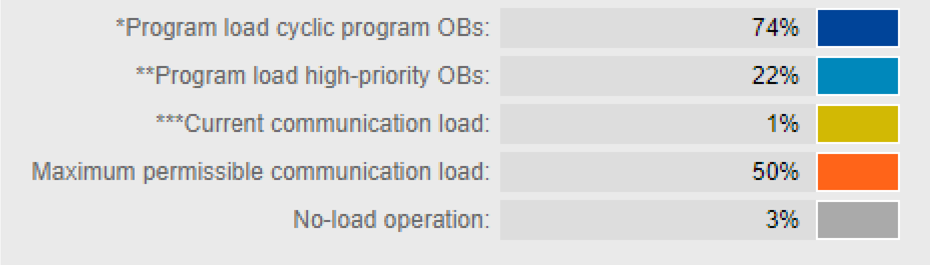
The legend of the program-/communication load shows information on the following values, highlighted in color:

- "Program load cyclic program OBs"
  required computing time in percent within a cycle for cyclic program OBs
- "Program load high-priority OBs"
  Required calculation time in percent within a cycle for higher-priority OBs (priority ≥ 15)
- "Current communications load"
  Required calculation time in percent for current communications tasks within a cycle
- "Maximum permissible communication load"
  The configured maximum communication load as a percentage
- "No-load operation"
  There is no program-/communication load

NOTE

When you have configured a minimum cycle time, it can happen that no-load operation displays a high percentage value, although the value of the cycle time is also high.

The reason for this is that the loads are recorded as mathematical average of the last second, but the cycle time relates to the last cycle.



*Program load cyclic program OBs:	74%	🟦
**Program load high-priority OBs:	22%	🟦
***Current communication load:	1%	🟨
Maximum permissible communication load:	50%	🟧
No-load operation:	3%	⬜

Figure 4-12 Color legend

If you click on a specific color, the selected color is highlighted in the chart. If you click on a highlighted color, you remove the highlighting.

Measurement of load distribution and cycle time

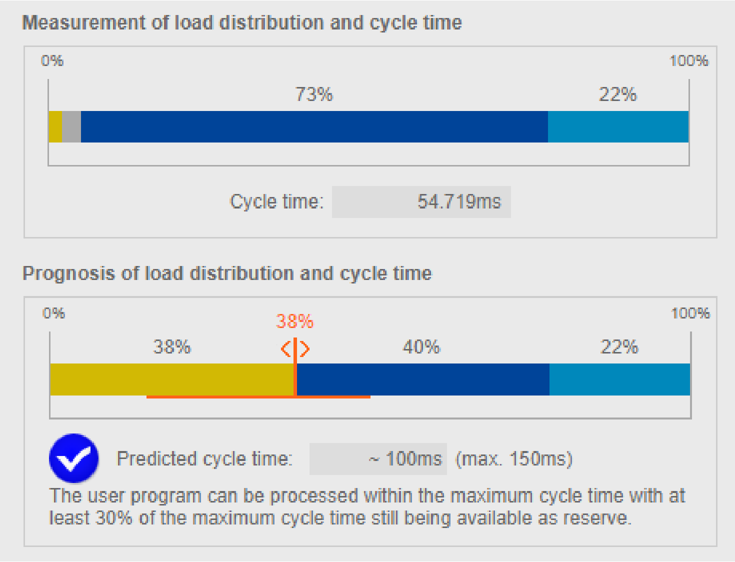
The "Measurement of load distribution and cycle time" bar chart shows the percentage of the calculation time within a cycle for the following values:

- "Program load cyclic program OBs"
- "Program load high-priority OBs"
- "Current communications load"
- "No-load operation"

Prognosis of load distribution and cycle time

The "Prognosis of load distribution and cycle time" bar chart predicts whether the CPU can process the user program with maximum communication load within the maximum cycle time.

Example 1:



Measurement of load distribution and cycle time

Cycle time: 54.719ms

Prognosis of load distribution and cycle time

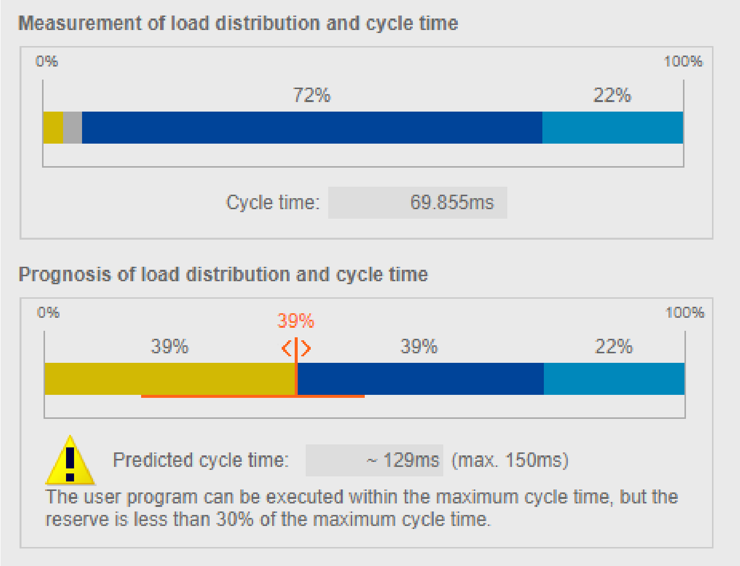
✓ Predicted cycle time: ~ 100ms (max. 150ms)

The user program can be processed within the maximum cycle time with at least 30% of the maximum cycle time still being available as reserve.

Figure 4-13 Cycle time < 70% of the maximum cycle time

Example 1 shows that the CPU can process the user program within the maximum cycle time of 150 ms when the maximum communication load of 38% is reached. The predicted cycle time is < 70% of the configured maximum cycle time.

Example 2:



Measurement of load distribution and cycle time

Cycle time: 69.855ms

Prognosis of load distribution and cycle time

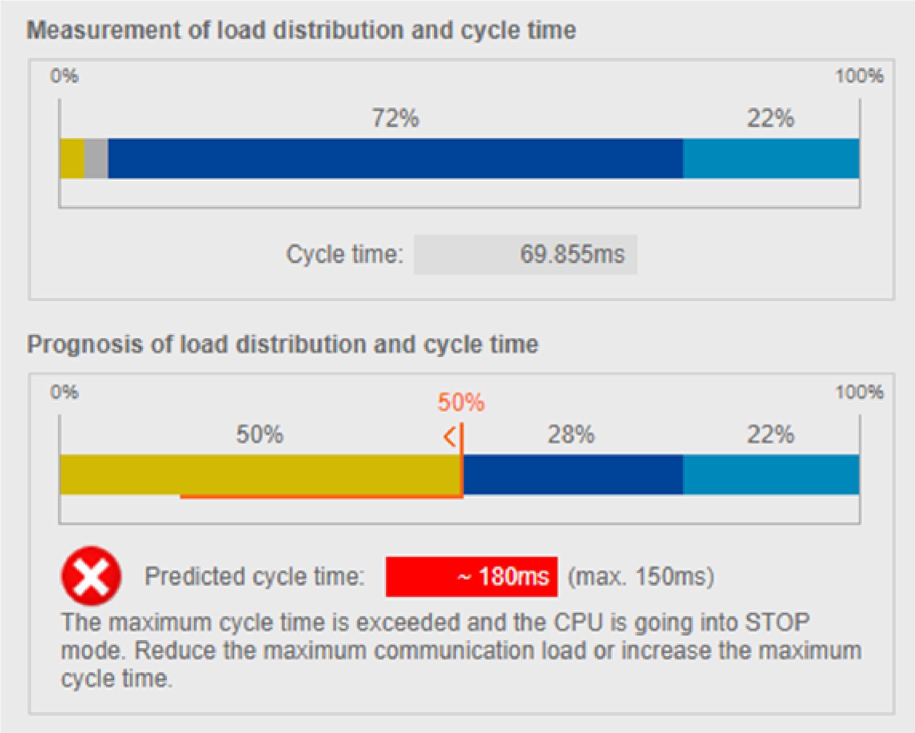
⚠ Predicted cycle time: ~ 129ms (max. 150ms)

The user program can be executed within the maximum cycle time, but the reserve is less than 30% of the maximum cycle time.

Figure 4-14 Cycle time ≥ 70% of the maximum cycle time

In example 2, the CPU can also process the user program with maximum communication load within the maximum cycle time. However, the predicted cycle time is already at 129 ms. As soon as the predicted cycle time is ≥ 70% of the maximum cycle time, the chart outputs a warning.

Example 3:



Measurement of load distribution and cycle time

Cycle time: 69.855ms

Prognosis of load distribution and cycle time

Predicted cycle time: ~ 180ms (max. 150ms)

The maximum cycle time is exceeded and the CPU is going into STOP mode. Reduce the maximum communication load or increase the maximum cycle time.

Figure 4-15 Cycle time longer than maximum cycle time

Example 3 shows that the CPU can no longer process the user program within the maximum cycle time when the maximum communication load is reached. If the predicted cycle time is longer than the maximum cycle time, the chart outputs an error message.

If it is predicted that the maximum cycle time will be exceeded, use the following controller in order to reduce the maximum communication load.

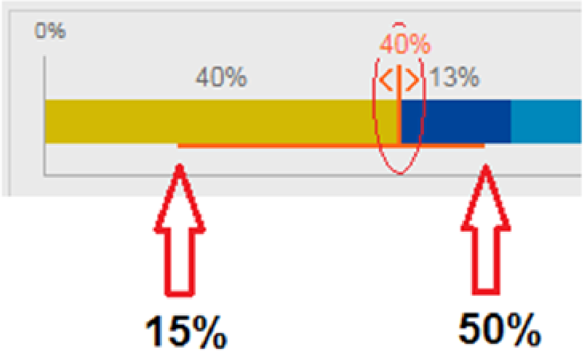


Figure 4-16 Controller for setting the maximum communication load

---

NOTE

Setting the communication load

The controller predicts the effects of the changed communication load on the cycle time. You configure the maximum communication load in STEP 7.

NOTE

For non-measurable fluctuations in the user program, e.g. for future changes in the user program, plan a sufficiently low value for the maximum communication load.

NOTE

Due to the different acquisition bases of cycle time and load, a settled system state is required to display reliable measured values.

You can find more information about the influence of the communication load on the cycle time in the Cycle and Response Times Function Manual (https://support.industry.siemens.com/cs/us/en/view/59193558).

Trend for program/communication load

If your web browser supports the display of SVG (Scalable Vector Graphics), the display in the "Runtime information" tab is expanded to show the trend for program/communication load.

With the line charts in the "Trend for program/communication load" area, you can track the progression of the following values:

- "Program load of the cyclic program OBs"
- "Program load high-priority OBs"
- "Current communications load"

With the "Number of recorded measuring points" option, you can choose between the last 20 to 1 000 measured values for the display of the measured values.

For the trend on the x-axis, you can choose between "Time" (CPU time) and "Samples" by clicking on the desired unit.

NOTE

If you have selected the "Time" unit on the x-axis, all measured values that are more than 24 hours old are deleted automatically.

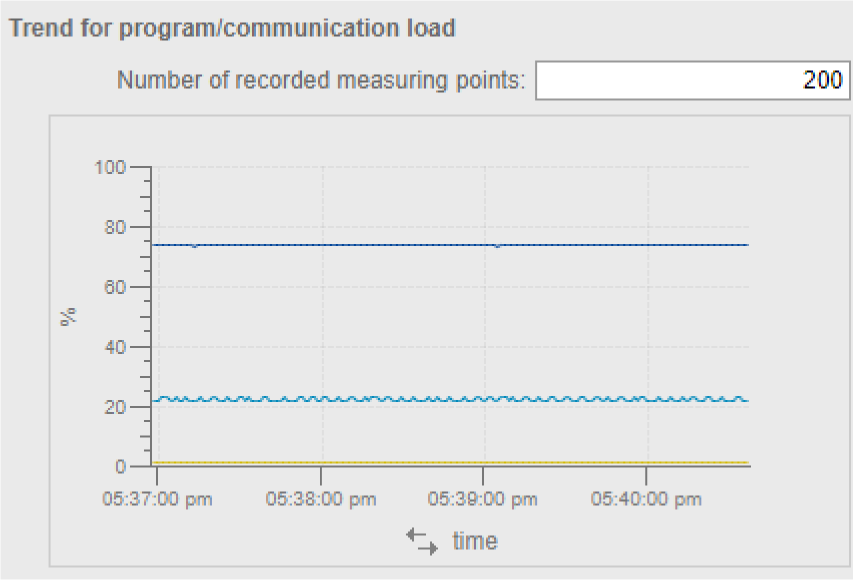
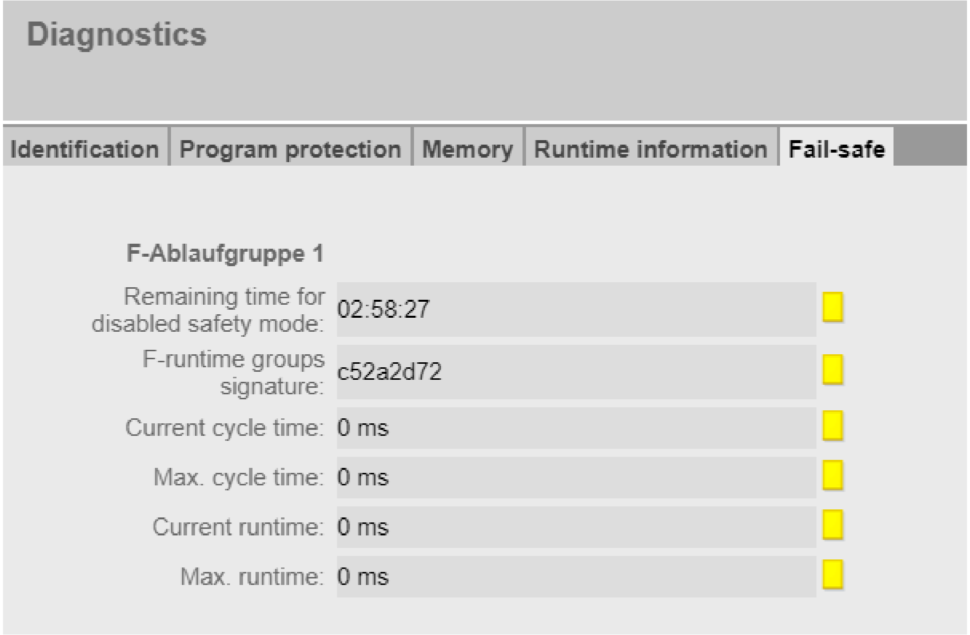


Figure 4-17 Line chart

"Fail-safe" tab (with an F-CPU)

The safety program of an F-CPU consists of one or two F-runtime groups. You can find their F-runtime group signature, cycle times (F-monitoring time) and runtimes in the "Fail-safe" tab.



F-Ablaufgruppe 1

Remaining time for disabled safety mode:	02:58:27
F-runtime groups signature:	c52a2d72
Current cycle time:	0 ms
Max. cycle time:	0 ms
Current runtime:	0 ms
Max. runtime:	0 ms

Figure 4-18 "Fail-safe" tab

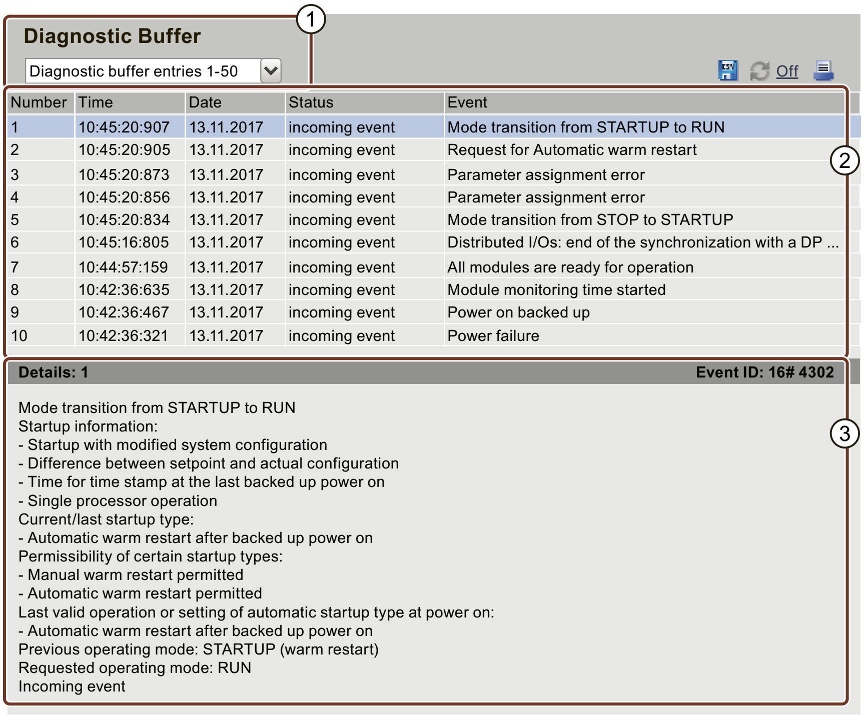
4.3 Diagnostics buffer

Requirements

The web server is activated, languages are set, the text libraries are loaded and the project has been compiled and downloaded with STEP 7.

Diagnostics buffer

The content of the diagnostic buffer is displayed by the browser on the web page "Diagnostics buffer".



Diagnostic buffer entries 1-50

Number	Time	Date	Status	Event
1	10:45:20:907	13.11.2017	incoming event	Mode transition from STARTUP to RUN
2	10:45:20:905	13.11.2017	incoming event	Request for Automatic warm restart
3	10:45:20:873	13.11.2017	incoming event	Parameter assignment error
4	10:45:20:856	13.11.2017	incoming event	Parameter assignment error
5	10:45:20:834	13.11.2017	incoming event	Mode transition from STOP to STARTUP
6	10:45:16:805	13.11.2017	incoming event	Distributed I/Os: end of the synchronization with a DP ...
7	10:44:57:159	13.11.2017	incoming event	All modules are ready for operation
8	10:42:36:635	13.11.2017	incoming event	Module monitoring time started
9	10:42:36:467	13.11.2017	incoming event	Power on backed up
10	10:42:36:321	13.11.2017	incoming event	Power failure

Details: 1 Event ID: 16# 4302

Mode transition from STARTUP to RUN
Startup information:
- Startup with modified system configuration
- Difference between setpoint and actual configuration
- Time for time stamp at the last backed up power on
- Single processor operation
Current/last startup type:
- Automatic warm restart after backed up power on
Permissibility of certain startup types:
- Manual warm restart permitted
- Automatic warm restart permitted
Last valid operation or setting of automatic startup type at power on:
- Automatic warm restart after backed up power on
Previous operating mode: STARTUP (warm restart)
Requested operating mode: RUN
Incoming event

Figure 4-19 Diagnostics buffer

① "Diagnostics buffer entries 1-50"

The diagnostics buffer can accommodate different numbers of alarms depending on the CPU used.

For information on the maximum number of diagnostics buffer entries, refer to the technical specifications of the CPU used.

Select an interval for the entries from the drop-down list. Each interval comprises 50 entries.

② "Event"

The "Event" info box contains the diagnostics interrupts with date and time.
Note that the diagnostic events are displayed in the project language of the STEP 7 project that is assigned to the current web server interface language. You can find out how to assign project languages to interface languages in section Language settings (Page 37).

③ "Details"

This field outputs detailed information about a selected event. Select the corresponding event from the ② "Event" info field.

Saving diagnostics buffer entries

You can save diagnostics buffer entries to a csv file for further processing in a spreadsheet program or database program.
Save the data by clicking the 💾 icon.
A dialog opens in which you can specify the file name and target directory.

4.4 Motion Control diagnostics

Overview

On the "Motion Control Diagnostics" web page, you will find the status, error, and warning bits as well as the latest values for the configured technology objects (TO).
The Web server supports the following technology objects:

- Speed-controlled axis (TO_SpeedAxis)
- Positioning axis (TO_PositioningAxis)
- Synchronous axis (TO_SynchronousAxis)
- External encoder (TO_ExternalEncoder)
- Measuring input (TO_MeasuringInput)
- Output cam (TO_OutputCam)
- Cam track (TO_CamTrack)
- Cam disk (TO_Cam, TO_Cam_10k) (S7-1500T)
- Leading axis proxy (TO_LeadingAxisProxy) (S7-1500T)
- Kinematics (TO_Kinematics) (S7-1500T)
- Interpreter (TO_Interpreter) (S7-1500T)

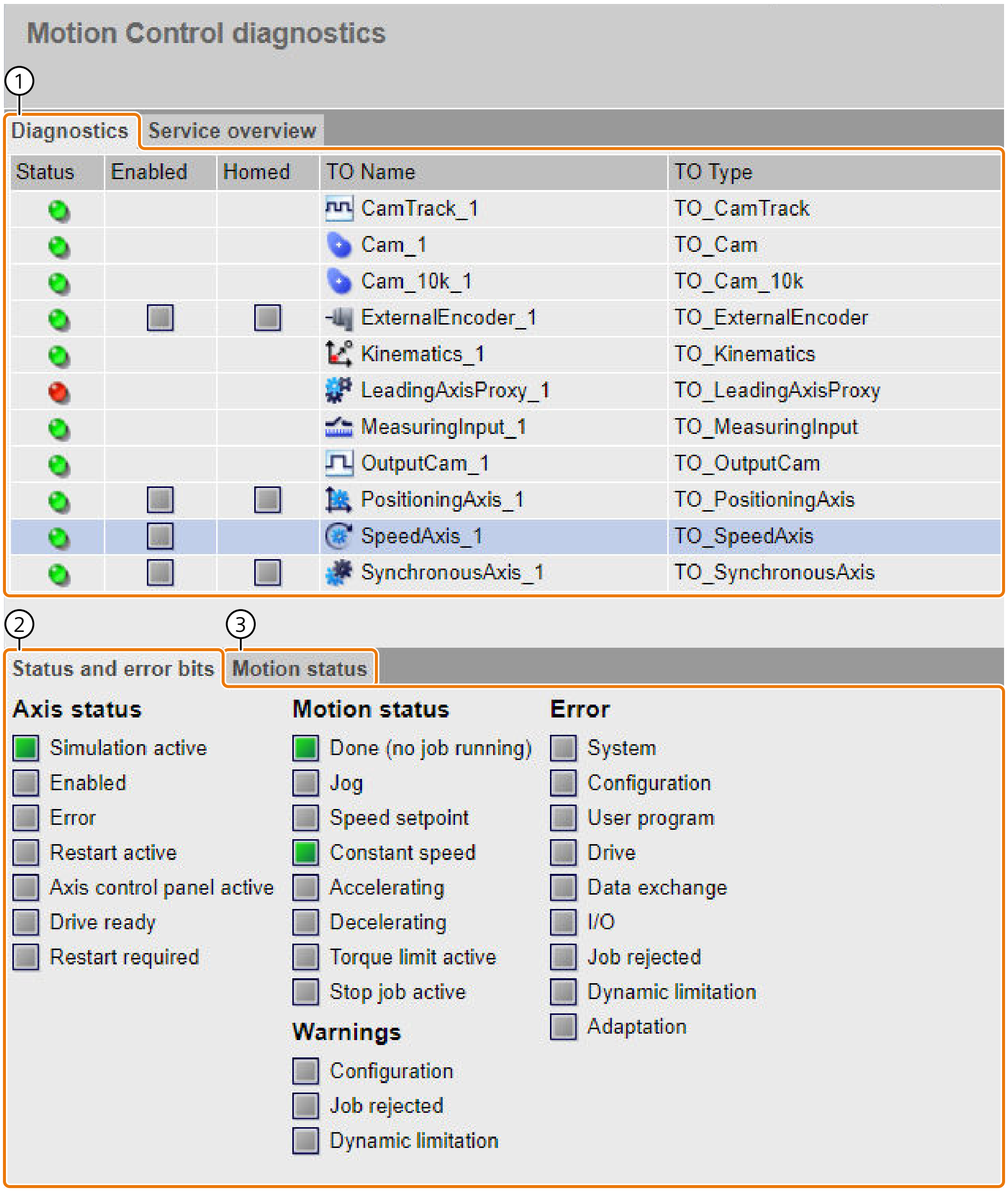
The web page is subdivided into the tabs:

- Diagnostics
- Service overview

"Diagnostics" tab

In the "Diagnostics" view you will see:

- ① A list of the configured technology objects
- ② The status and error bits of a selected technology object
- ③ Specific diagnostic information for the technology objects



Status	Enabled	Homed	TO Name	TO Type
🟢			CamTrack_1	TO_CamTrack
🟢			Cam_1	TO_Cam
🟢			Cam_10k_1	TO_Cam_10k
🟢	☐	☐	ExternalEncoder_1	TO_ExternalEncoder
🟢			Kinematics_1	TO_Kinematics
🔴			LeadingAxisProxy_1	TO_LeadingAxisProxy
🟢			MeasuringInput_1	TO_MeasuringInput
🟢			OutputCam_1	TO_OutputCam
🟢	☐	☐	PositioningAxis_1	TO_PositioningAxis
🟢	☐		SpeedAxis_1	TO_SpeedAxis
🟢	☐	☐	SynchronousAxis_1	TO_SynchronousAxis

Axis status

- 🟩 Simulation active
- ☐ Enabled
- ☐ Error
- ☐ Restart active
- ☐ Axis control panel active
- ☐ Drive ready
- ☐ Restart required

Motion status

- 🟩 Done (no job running)
- ☐ Jog
- ☐ Speed setpoint
- 🟩 Constant speed
- ☐ Accelerating
- ☐ Decelerating
- ☐ Torque limit active
- ☐ Stop job active

Warnings

- ☐ Configuration
- ☐ Job rejected
- ☐ Dynamic limitation

Error

- ☐ System
- ☐ Configuration
- ☐ User program
- ☐ Drive
- ☐ Data exchange
- ☐ I/O
- ☐ Job rejected
- ☐ Dynamic limitation
- ☐ Adaptation

Figure 4-20 Diagnostics - Speed axis status and error bits

① List of the configured technology objects

Check the following diagnostic information in the list of technology objects:

- Status (green = no warning, no alarm; yellow = warning is pending; red = error is pending)
- Enabled
- Homed

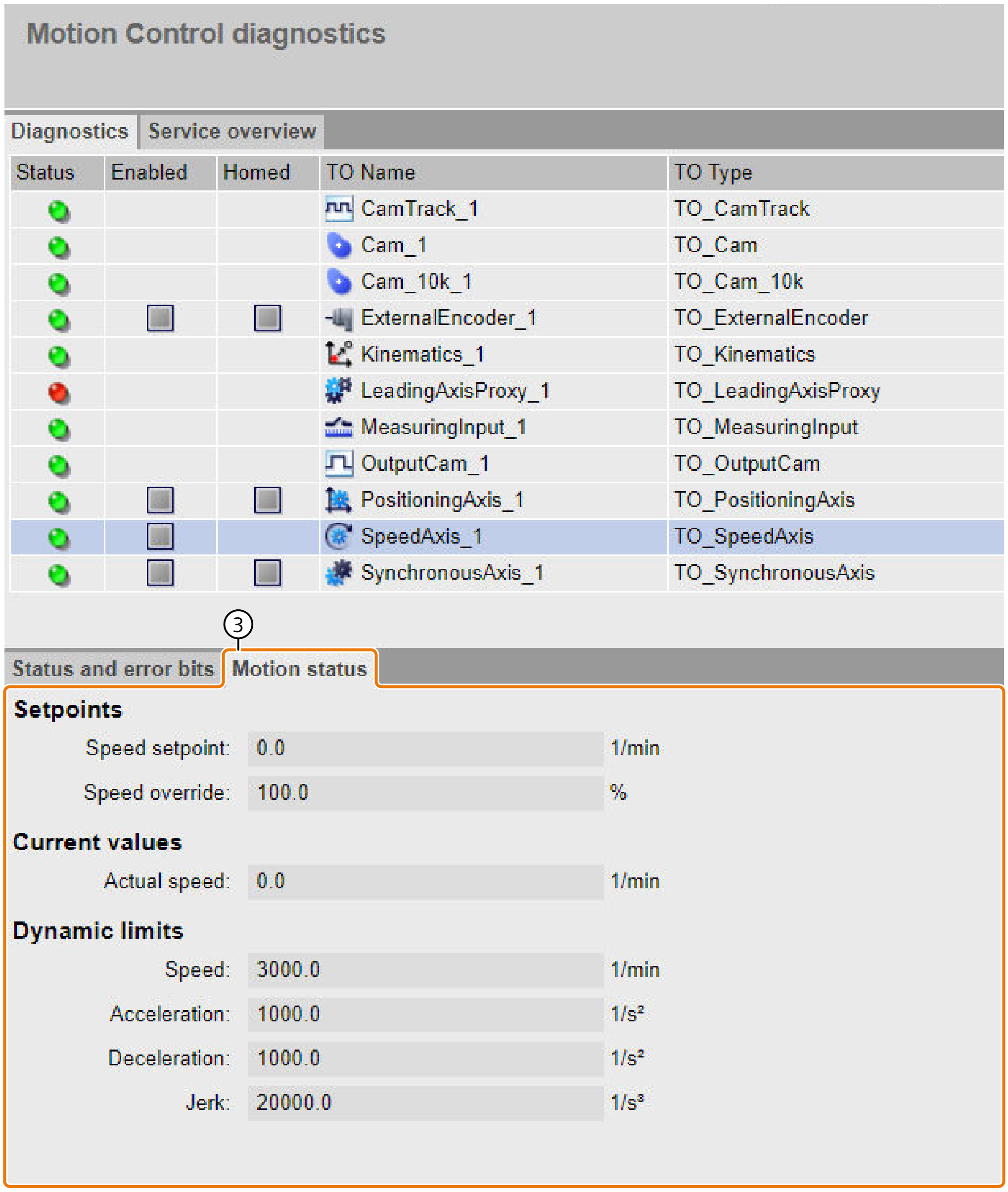
Click on a technology object in the list to display the current status and error bits ② as well as specific diagnostic information ③ for this technology object.

② Status and error bits

The displayed status and error bits correspond to the diagnostics of the technology object in STEP 7.

③ Specific diagnostic information for the technology objects

Depending on the type of technology object, you will receive additional diagnostic information, for example, the status of a speed axis. The displayed diagnostic information corresponds to the diagnostics of the technology object in STEP 7.



Motion Control diagnostics

Diagnostics | Service overview

Status	Enabled	Homed	TO Name	TO Type
🟢			CamTrack_1	TO_CamTrack
🟢			Cam_1	TO_Cam
🟢			Cam_10k_1	TO_Cam_10k
🟢	□	□	ExternalEncoder_1	TO_ExternalEncoder
🟢			Kinematics_1	TO_Kinematics
🔴			LeadingAxisProxy_1	TO_LeadingAxisProxy
🟢			MeasuringInput_1	TO_MeasuringInput
🟢			OutputCam_1	TO_OutputCam
🟢	□	□	PositioningAxis_1	TO_PositioningAxis
🟢	□		SpeedAxis_1	TO_SpeedAxis
🟢	□	□	SynchronousAxis_1	TO_SynchronousAxis

Status and error bits | Motion status

Setpoints

Speed setpoint: 0.0 1/min

Speed override: 100.0 %

Current values

Actual speed: 0.0 1/min

Dynamic limits

Speed: 3000.0 1/min

Acceleration: 1000.0 1/s^2

Deceleration: 1000.0 1/s^2

Jerk: 20000.0 1/s^3

Figure 4-21 Diagnostics - Speed axis motion status

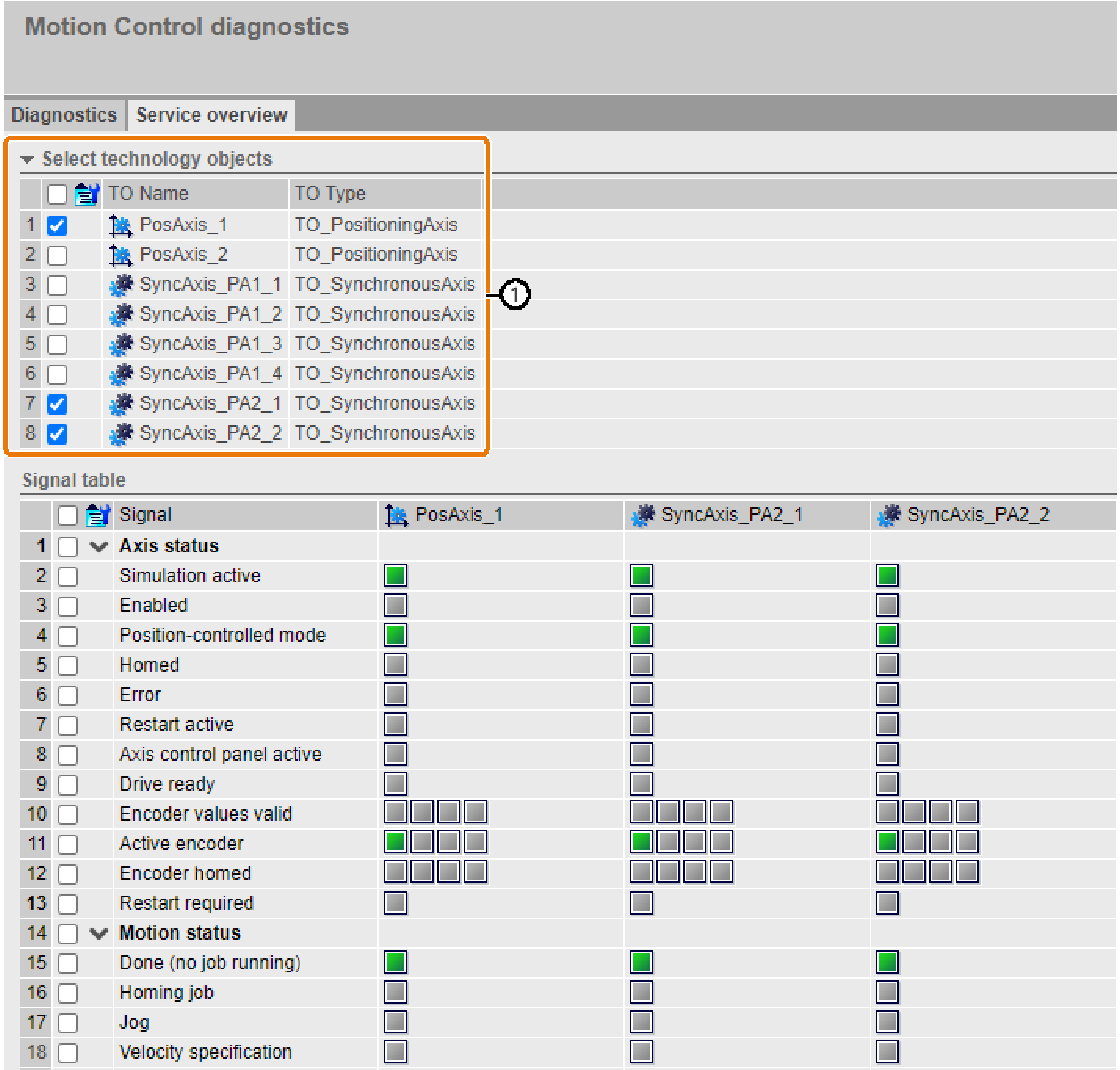
"Service overview" tab

In the "Service overview" tab, you observe selected signals of multiple technology objects at the same time, for example, the actual position of multiple axes.
The "Service overview" is subdivided into the areas:

- ① Select technology objects
- ② Signal table

① Select technology objects

In this area, you select which technology objects you want to observe in the signal table.



Motion Control diagnostics

Diagnostics | Service overview

Select technology objects

	□	TO Name	TO Type
1	☑	PosAxis_1	TO_PositioningAxis
2	□	PosAxis_2	TO_PositioningAxis
3	□	SyncAxis_PA1_1	TO_SynchronousAxis
4	□	SyncAxis_PA1_2	TO_SynchronousAxis
5	□	SyncAxis_PA1_3	TO_SynchronousAxis
6	□	SyncAxis_PA1_4	TO_SynchronousAxis
7	☑	SyncAxis_PA2_1	TO_SynchronousAxis
8	☑	SyncAxis_PA2_2	TO_SynchronousAxis

Signal table

	□	Signal	PosAxis_1	SyncAxis_PA2_1	SyncAxis_PA2_2
1	□	**Axis status**			
2	□	Simulation active	🟩	🟩	🟩
3	□	Enabled	⬜	⬜	⬜
4	□	Position-controlled mode	🟩	🟩	🟩
5	□	Homed	⬜	⬜	⬜
6	□	Error	⬜	⬜	⬜
7	□	Restart active	⬜	⬜	⬜
8	□	Axis control panel active	⬜	⬜	⬜
9	□	Drive ready	⬜	⬜	⬜
10	□	Encoder values valid	⬜⬜⬜⬜	⬜⬜⬜⬜	⬜⬜⬜⬜
11	□	Active encoder	🟩⬜⬜⬜	🟩⬜⬜⬜	🟩⬜⬜⬜
12	□	Encoder homed	⬜⬜⬜⬜	⬜⬜⬜⬜	⬜⬜⬜⬜
13	□	Restart required	⬜	⬜	⬜
14	□	**Motion status**			
15	□	Done (no job running)	🟩	🟩	🟩
16	□	Homing job	⬜	⬜	⬜
17	□	Jog	⬜	⬜	⬜
18	□	Velocity specification	⬜	⬜	⬜

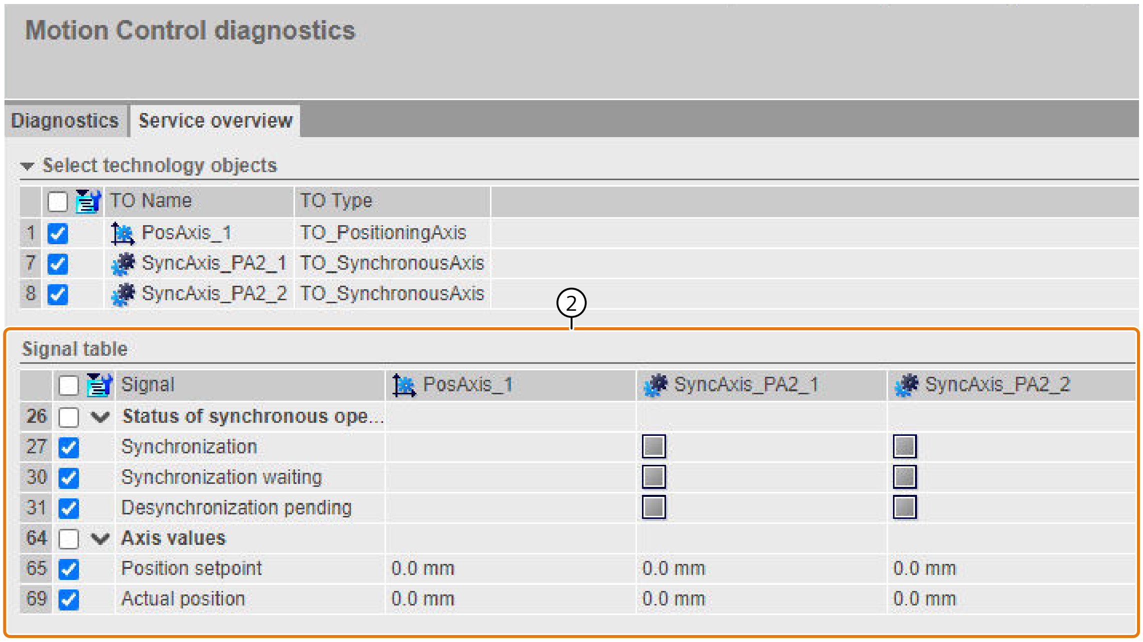
Figure 4-22 Service overview - Select technology objects

1. To sort the list of technology objects, click on the column header "TO Name" or "TO Type".
2. To observe a technology object in the signal table, select the check box in front of the technology object.
   A column is only displayed for the selected technology objects in the signal table. Signals that do not exist for the selected technology objects are hidden.

3. Click the icon.
   Only the selected technology objects are displayed in the "Select technology objects" area.
4. Click the icon again.
   All technology objects are displayed once again in the "Select technology objects" area.

② **Signal table**

In this area, you select which signals of the selected technology objects you want to observe. The signals are grouped.



Motion Control diagnostics

Diagnostics | Service overview

Select technology objects

	□	TO Name	TO Type
1	☑	PosAxis_1	TO_PositioningAxis
7	☑	SyncAxis_PA2_1	TO_SynchronousAxis
8	☑	SyncAxis_PA2_2	TO_SynchronousAxis

Signal table

	□	Signal	PosAxis_1	SyncAxis_PA2_1	SyncAxis_PA2_2
26	□	**Status of synchronous ope...**			
27	☑	Synchronization		▨	▨
30	☑	Synchronization waiting		▨	▨
31	☑	Desynchronization pending		▨	▨
64	□	**Axis values**			
65	☑	Position setpoint	0.0 mm	0.0 mm	0.0 mm
69	☑	Actual position	0.0 mm	0.0 mm	0.0 mm

Figure 4-23 Service overview - Signal table

1. To expand or collapse a scaling group, click the arrow in front of the scaling group.
2. To only observe selected scaling groups or signals, select the check box in front of the scaling group or signal.
3. Click the icon.
   Only the selected signal groups or signals are displayed.
4. Click the icon again.
   All signal groups and signals are displayed.

Displaying and acknowledging technology alarms

On the "Alarms (Page 76)" web page you will find an overview of the pending technology alarms with error numbers.

You can acknowledge the pending technology alarms on the "Alarms" web page.

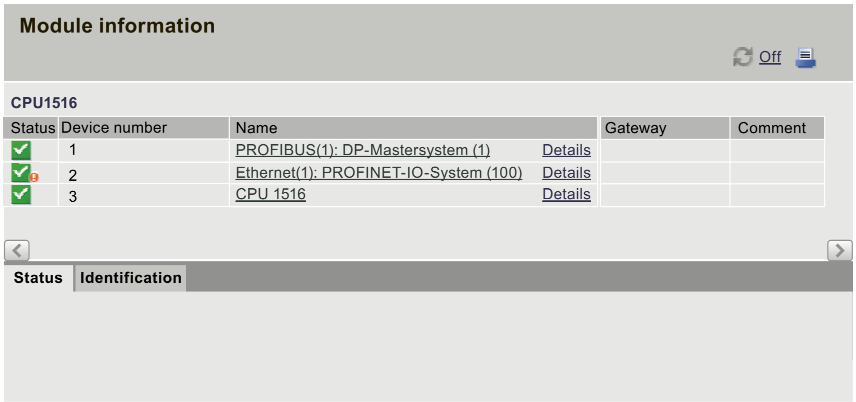
More information

You can find explanations for the diagnostics functions of the individual technology objects in the STEP 7 online help or the S7-1500/1500T Motion Control function manuals on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109751049).

4.5 Module information

Module information

The status of a device is indicated by means of symbols and comments on the "Module information" web page.



Module information

CPU1516

Status	Device number	Name		Gateway	Comment
✓	1	PROFIBUS(1): DP-Mastersystem (1)	Details		
✓!	2	Ethernet(1): PROFINET-IO-System (100)	Details		
✓	3	CPU 1516	Details		

Figure 4-24 Module information

Meaning of the symbols in the "Status" column

Table 4-2 Meaning of symbols

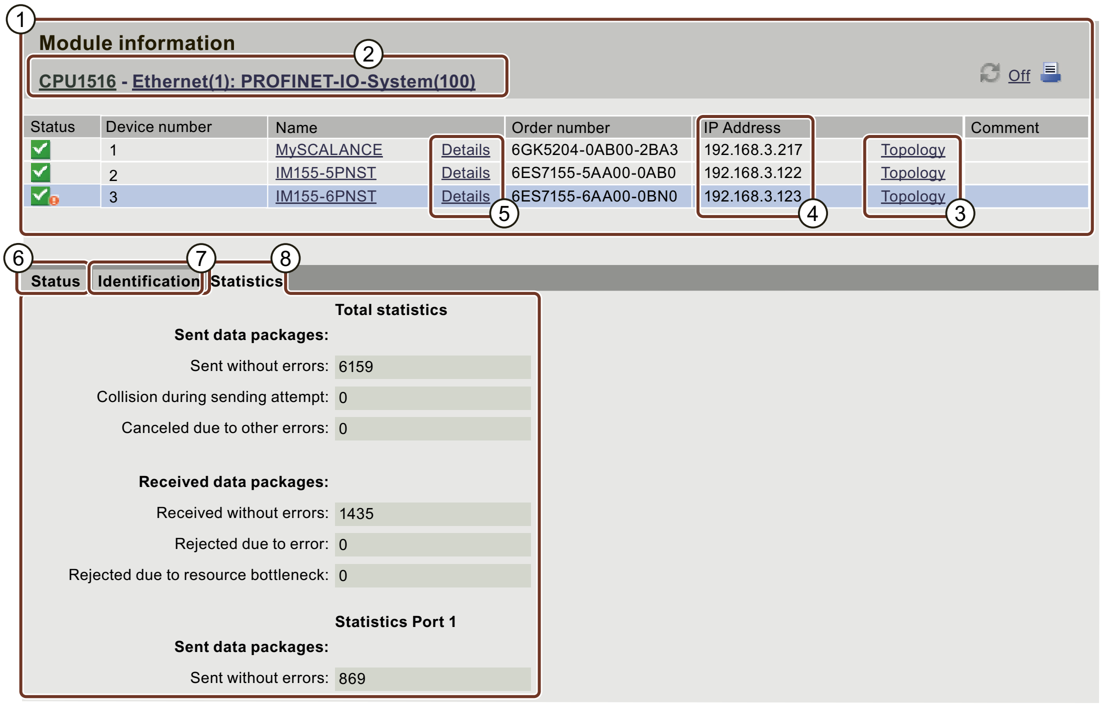
Symbol	Symbol color	Meaning
✓	green	Component is OK
✓	gray	Deactivated PROFIBUS or PROFINET devices.
?	gray	State cannot be determined • "State cannot be determined" is displayed during system diagnostics for all configured I/O modules and I/O systems after restart of the CPU. • However, this state can also be displayed temporarily during operation if a diagnostics interrupt burst occurs for all modules. • It is not possible to determine the status of modules on a subsystem that is connected to a CP.
☒	red	Component "not reachable" "Not reachable" is displayed when a module has been removed or a module has been configured but does not exist.
0101	black	No input or output data available. Input or output channels of the (sub)module are disabled.
🔧	green	Maintenance required (Maintenance Required)

Symbol	Symbol color	Meaning
🔧	yellow	Maintenance demanded (Maintenance Demanded)
🔧	red	Error - component faulty or not available due to an incorrect type
❗	red	A module in a lower module level does not have the status "Component OK"

Navigation to further module levels

The status of individual components/modules/submodules is displayed when you navigate to the further module levels:

- To the next higher module level using the links in the display of the module levels
- To the next lower module level using the links in the "Name" column



Module information

CPU1516 - Ethernet(1): PROFINET-IO-System(100)

Status	Device number	Name		Order number	IP Address		Comment
✓	1	MySCALANCE	Details	6GK5204-0AB00-2BA3	192.168.3.217	Topology	
✓	2	IM155-5PNST	Details	6ES7155-5AA00-0AB0	192.168.3.122	Topology	
✓❗	3	IM155-6PNST	Details	6ES7155-6AA00-0BN0	192.168.3.123	Topology	

Total statistics

Sent data packages:

Sent without errors: 6159

Collision during sending attempt: 0

Canceled due to other errors: 0

Received data packages:

Received without errors: 1435

Rejected due to error: 0

Rejected due to resource bottleneck: 0

Statistics Port 1

Sent data packages:

Sent without errors: 869

Figure 4-25 Navigation to further module levels

① "Module information"

Depending on the selected level, the table contains information on the rack, the DP master system, the PROFINET IO master system, the stations, the individual modules or the modules or submodules of the station.

② "Display of the module levels"

The links are used to access the "Module information" of the higher module levels.

③ "Topology"

The two web pages, "Topology" and "Module information", are linked. A click on "Topology" of the selected module automatically takes you to this module in the graphic view of the set topology on the "Topology" web page. The module is displayed in the visible area of the "Topology" web page. The device header of the selected module flashes for a few seconds.

④ "IP address"

If a link is available, you can use it to access the Web server of the configured device you selected.

⑤ "Details"

More information about the selected module is provided in the "Status" and "Identification" tabs via the "Details" link.

⑥ "Status" tab

The tab contains information about the status of the selected module when a fault or alarm exists.

⑦ "Identification" tab

The tab contains data on the identification of the selected module.

NOTE

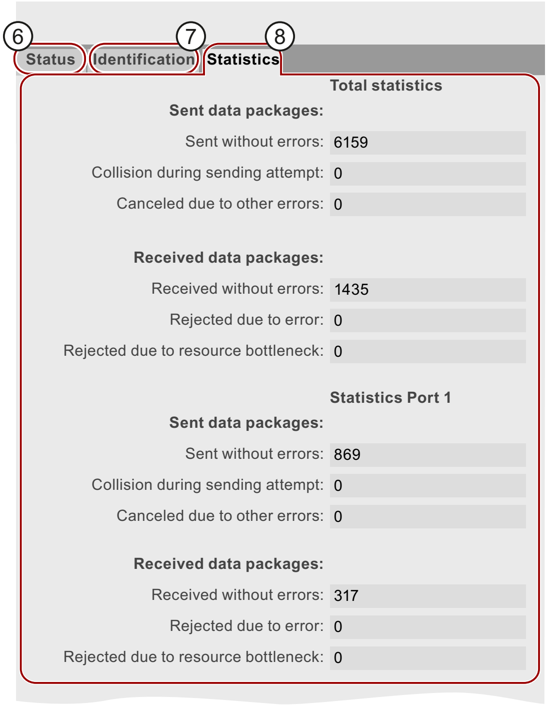
This tab displays only the data configured offline of the module.

⑧ "Statistics" tab

The tab is only displayed for PROFINET IO devices and contains the following information on the communication statistics of the selected IO device:

- "Total statistics - Sent data packages"
  You can assess the data transmission on the transmit line based on the key data in this info box.
- "Total statistics - Received data packages"
  You can assess the data transmission on the receive line based on the key data in this info box.
- "Statistics port x - Sent data packages"
  You can assess the data transmission on the transmit line for each port based on the key data in this info box.

- "Statistics port x - Received data packages"
  You can assess the data transmission on the receive line for each port based on the key data in this info box.



Total statistics

Sent data packages:

Sent without errors: 6159

Collision during sending attempt: 0

Canceled due to other errors: 0

Received data packages:

Received without errors: 1435

Rejected due to error: 0

Rejected due to resource bottleneck: 0

Statistics Port 1

Sent data packages:

Sent without errors: 869

Collision during sending attempt: 0

Canceled due to other errors: 0

Received data packages:

Received without errors: 317

Rejected due to error: 0

Rejected due to resource bottleneck: 0

Figure 4-26 "Statistics" tab

Reference

You can find more information in the "Statistics" tab in the section Communication (Page 78).

"Safety" tab for fail-safe I/O

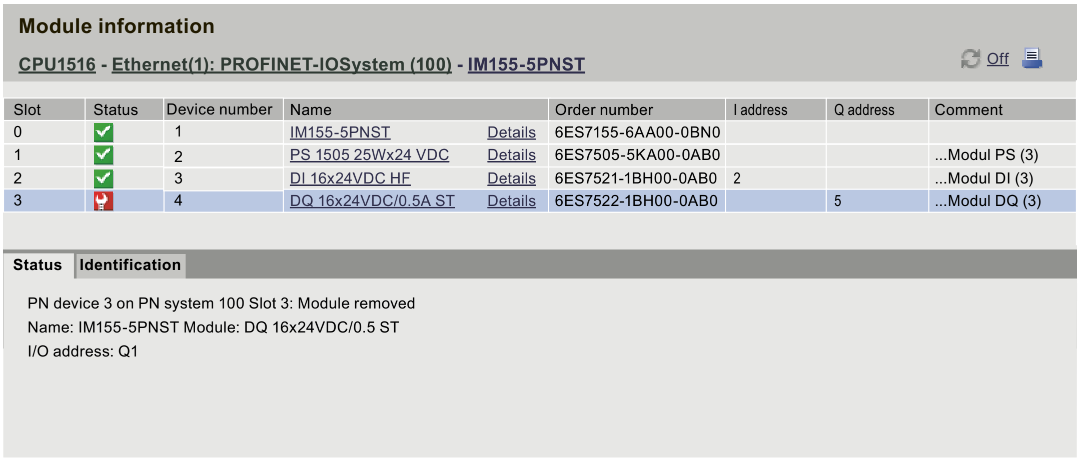
The tab is only displayed for fail-safe I/O and contains the following parameters:

- F-Par_CRC (with addresses)
- F-monitoring time
- F-source address
- F-destination address

Reference

You can find more information about the F-parameters in the Programming and Operating Manual SIMATIC Safety - Configuring and Programming (https://support.automation.siemens.com/WW/view/en/54110126).

Example: Module information - module



Module information

CPU1516 - Ethernet(1): PROFINET-IOSystem (100) - IM155-5PNST

Slot	Status	Device number	Name		Order number	I address	Q address	Comment
0	✓	1	IM155-5PNST	Details	6ES7155-6AA00-0BN0			
1	✓	2	PS 1505 25Wx24 VDC	Details	6ES7505-5KA00-0AB0			...Modul PS (3)
2	✓	3	DI 16x24VDC HF	Details	6ES7521-1BH00-0AB0	2		...Modul DI (3)
3	🟥	4	DQ 16x24VDC/0.5A ST	Details	6ES7522-1BH00-0AB0		5	...Modul DQ (3)

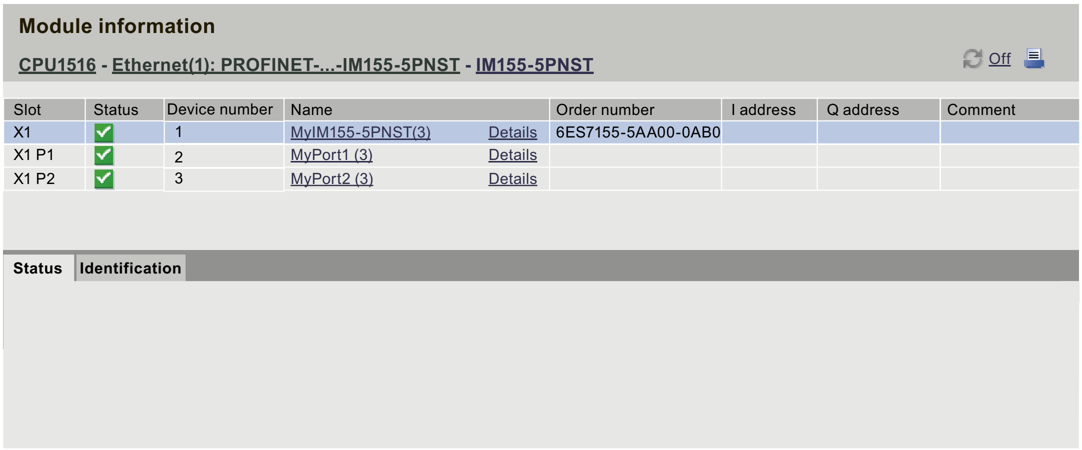
PN device 3 on PN system 100 Slot 3: Module removed
Name: IM155-5PNST Module: DQ 16x24VDC/0.5 ST
I/O address: Q1

Figure 4-27 Example: Module information - module

NOTE

If you are using the function Configuration control (option handling) in the central configuration of your plant, the information text in the headings area of the web page informs you that the status of the I/O modules may be displayed inconsistently. No corresponding text is displayed for the distributed I/O.

Example: Module information - submodule



Module information

CPU1516 - Ethernet(1): PROFINET-...-IM155-5PNST - IM155-5PNST

Slot	Status	Device number	Name		Order number	I address	Q address	Comment
X1	✓	1	MyIM155-5PNST(3)	Details	6ES7155-5AA00-0AB0			
X1 P1	✓	2	MyPort1 (3)	Details				
X1 P2	✓	3	MyPort2 (3)	Details				

Figure 4-28 Example: Module information - submodule

Reference

You can find more information on the "Module information" in the STEP 7 online help, keyword: "Module information".

4.6 Firmware update

Introduction

You can update the firmware as a user with the corresponding access rights on the "Module information" web page at the module level. You can find information on user management in section Configuring the Web server (Page 26) under "Amending user management".
With the help of an update file, you can update the firmware of the CPU, the display of the CPU, or individual centralized and distributed modules. Note that all modules you want to update must be compatible with the TIA Portal as of V12.0.

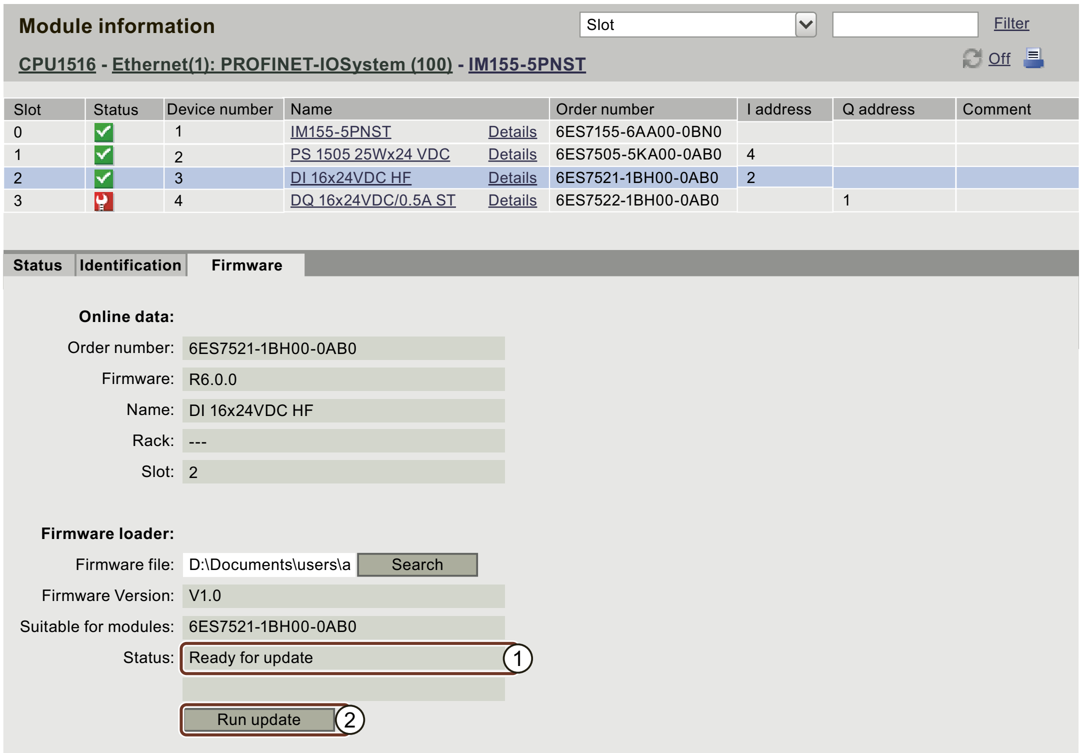
NOTE

A firmware update is not possible if access is via a mobile terminal device with the "iOS" operating system.

Procedure

The following steps are required to perform a firmware update:

- Click on "Browse" in the Firmware Loader area.
- Select the file you would like to use for the firmware update. You can find the available firmware updates on the Service&Support page on the Internet (https://support.automation.siemens.com).



Module information

Slot | Filter

CPU1516 - Ethernet(1): PROFINET-IOSystem (100) - IM155-5PNST

Off

Slot	Status	Device number	Name		Order number	I address	Q address	Comment
0	✓	1	IM155-5PNST	Details	6ES7155-6AA00-0BN0			
1	✓	2	PS 1505 25Wx24 VDC	Details	6ES7505-5KA00-0AB0	4		
2	✓	3	DI 16x24VDC HF	Details	6ES7521-1BH00-0AB0	2		
3	🟥	4	DQ 16x24VDC/0.5A ST	Details	6ES7522-1BH00-0AB0		1	

Status | Identification | Firmware

Online data:

Order number: 6ES7521-1BH00-0AB0
Firmware: R6.0.0
Name: DI 16x24VDC HF
Rack: ---
Slot: 2

Firmware loader:

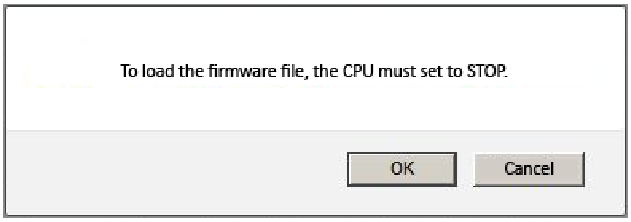
Firmware file: D:\Documents\users\a
Firmware Version: V1.0
Suitable for modules: 6ES7521-1BH00-0AB0
Status: Ready for update

① Status of the selected firmware file

② Button to execute the update

Figure 4-29 Module information, "Firmware" tab, "Ready for update" status

- If the status is "Ready for update", click "Run update". If the CPU is in RUN mode during the update, the following alarm is output:

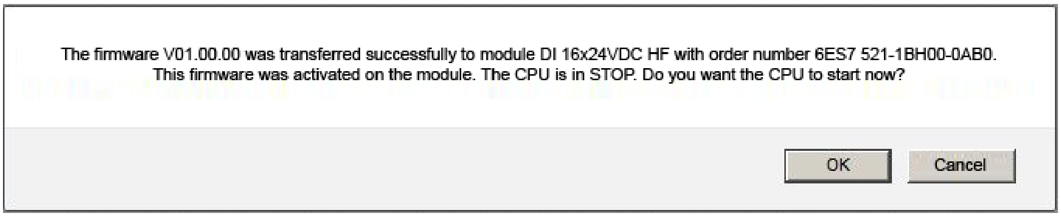


To load the firmware file, the CPU must set to STOP.

Figure 4-30 Alarm after clicking "Run update"

Acknowledge the alarm output by clicking "OK". The CPU is set to STOP mode and the firmware update is executed.
If you click "Cancel", the CPU remains in the current operating mode and the firmware update is canceled.

- A alarm informs you about the order number and version ID of the updated firmware once the update is complete.
  The CPU is automatically placed in RUN mode when the mode selector of the CPU is in RUN and when you acknowledge the alarm with "OK". This may take a few minutes; there is no progress indicator.
  If you click "Cancel", the CPU remains in STOP mode and you can run additional updates.



The firmware V01.00.00 was transferred successfully to module DI 16x24VDC HF with order number 6ES7 521-1BH00-0AB0.
This firmware was activated on the module. The CPU is in STOP. Do you want the CPU to start now?

Figure 4-31 Alarm: Firmware successfully transferred

Reference

For more information on the topic of firmware update, refer to the STEP 7 online help and the following FAQ on the Internet (https://support.industry.siemens.com/cs/ww/en/view/67190848).

4.7 Alarms

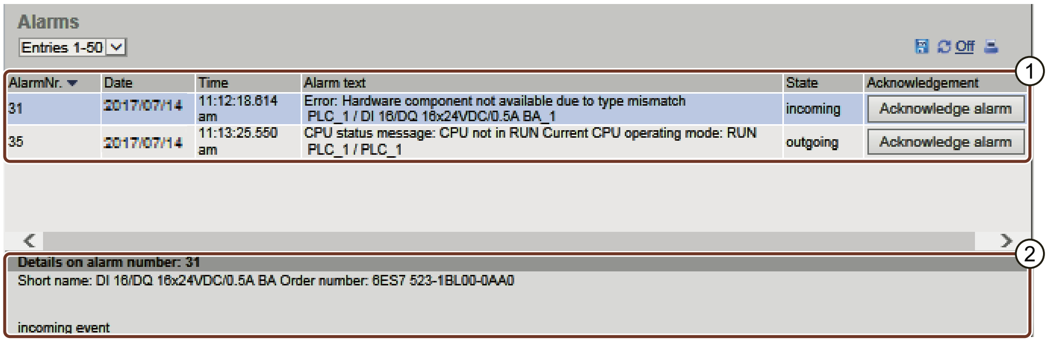
Requirements

The alarm texts were configured in the user-specific languages. You can find information about the configuration of alarm texts in STEP 7.

Alarms

To receive compact information on fault analysis, we recommend that you always first read out the content of the alarm buffer. This is the most effective method to get an overview of the pending faults.

The browser displays the content of the alarm buffer on the "Alarms" web page.



AlarmNr.	Date	Time	Alarm text	State	Acknowledgement
31	2017/07/14	11:12:18.614 am	Error: Hardware component not available due to type mismatch PLC_1 / DI 16/DQ 16x24VDC/0.5A BA_1	incoming	Acknowledge alarm
35	2017/07/14	11:13:25.550 am	CPU status message: CPU not in RUN Current CPU operating mode: RUN PLC_1 / PLC_1	outgoing	Acknowledge alarm

Details on alarm number: 31
Short name: DI 16/DQ 16x24VDC/0.5A BA Order number: 6ES7 523-1BL00-0AA0

incoming event

Figure 4-32 Alarms

① "Alarms"

Alarms of the CPU are displayed in descending chronological order with **date** and **time** in info box ①.

The **alarm text** parameter is an entry which contains the alarm texts configured for the corresponding fault definitions.

Note that the message texts are displayed in the project language of the STEP 7 project that is assigned to the current Web server interface language. You can find out how to assign project languages to interface languages in section Language settings (Page 37).

Sorting

You also have the option to display the individual parameters of the currently displayed web page (max. 50 entries) sorted in ascending or descending order. For this purpose, click on one of the parameters in the column header:

- Alarm number
- Date
- Time (of the CPU)
- Alarm text
- Status
- Acknowledgment

The alarms are returned in chronological order when you click the "Date" entry. Incoming and outgoing events are output at the **Status** parameter.

If you have the appropriate user rights (see section Configuring the Web server (Page 26)), for alarms which can be acknowledged, a button is available to you in the "Acknowledgment" column with which you can acknowledge the alarm.

② "Details on alarm number"

You can view detailed alarm information in this info box. Select the corresponding alarm from the info field ②.

Saving alarms

You can save alarms to a csv file for further processing in a spreadsheet program or database program.

Save the data by clicking the CSV icon.

A dialog opens in which you can specify the file name and target directory.

4.8 Communication

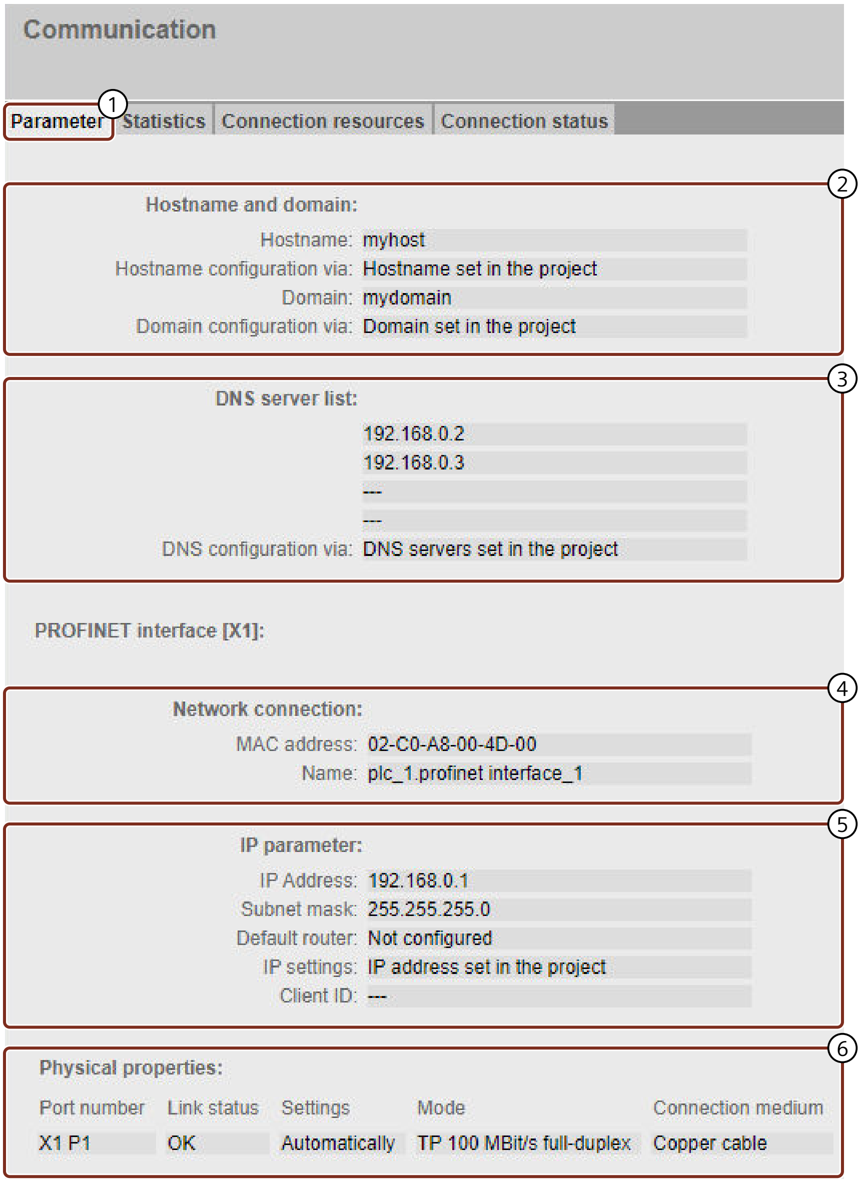
Overview

The "Communication" web page provides detailed information about the following tabs:

- Parameters
- Statistics
- Resources
- Connections

① "Parameter" tab

A summary of the information on the PROFINET and Ethernet interfaces of the selected CPU is available in the "Parameter" tab.



Hostname and domain:

Hostname: myhost
Hostname configuration via: Hostname set in the project
Domain: mydomain
Domain configuration via: Domain set in the project

DNS server list:

192.168.0.2
192.168.0.3

DNS configuration via: DNS servers set in the project

PROFINET interface [X1]:

Network connection:

MAC address: 02-C0-A8-00-4D-00
Name: plc_1.profinet interface_1

IP parameter:

IP Address: 192.168.0.1
Subnet mask: 255.255.255.0
Default router: Not configured
IP settings: IP address set in the project
Client ID: ---

Physical properties:

Port number	Link status	Settings	Mode	Connection medium
X1 P1	OK	Automatically	TP 100 MBit/s full-duplex	Copper cable

Figure 4-33 "Parameter" tab for communication via PROFINET

② "Host name and domain"

The CPU can use the DHCP communication protocol to assign the network configuration via a DHCP server. The CPU can make its host name and the domain available to the DHCP server.

Here, you will receive information on the host name and the domain of the CPU and where the configuration was specified, for example, remotely via DHCP, in the project or the user program.

③ "DNS server list"

Here, you will receive information on which DNS servers can communicate with the CPU and where the DNS configuration was specified, for example, remotely via DHCP, in the project or in the user program.

④ "Network connection"

Under "Network connection", you will find information for identification of the integrated PROFINET and Ethernet interfaces of the corresponding CPU. The MAC address is located on the CPU above the respective PROFINET or Ethernet interface.

⑤ "IP parameter"

This parameter includes information on the configured IP address and number of the subnet in which the corresponding CPU is located. If a client ID for DHCP was configured for the CPU, it is displayed here.

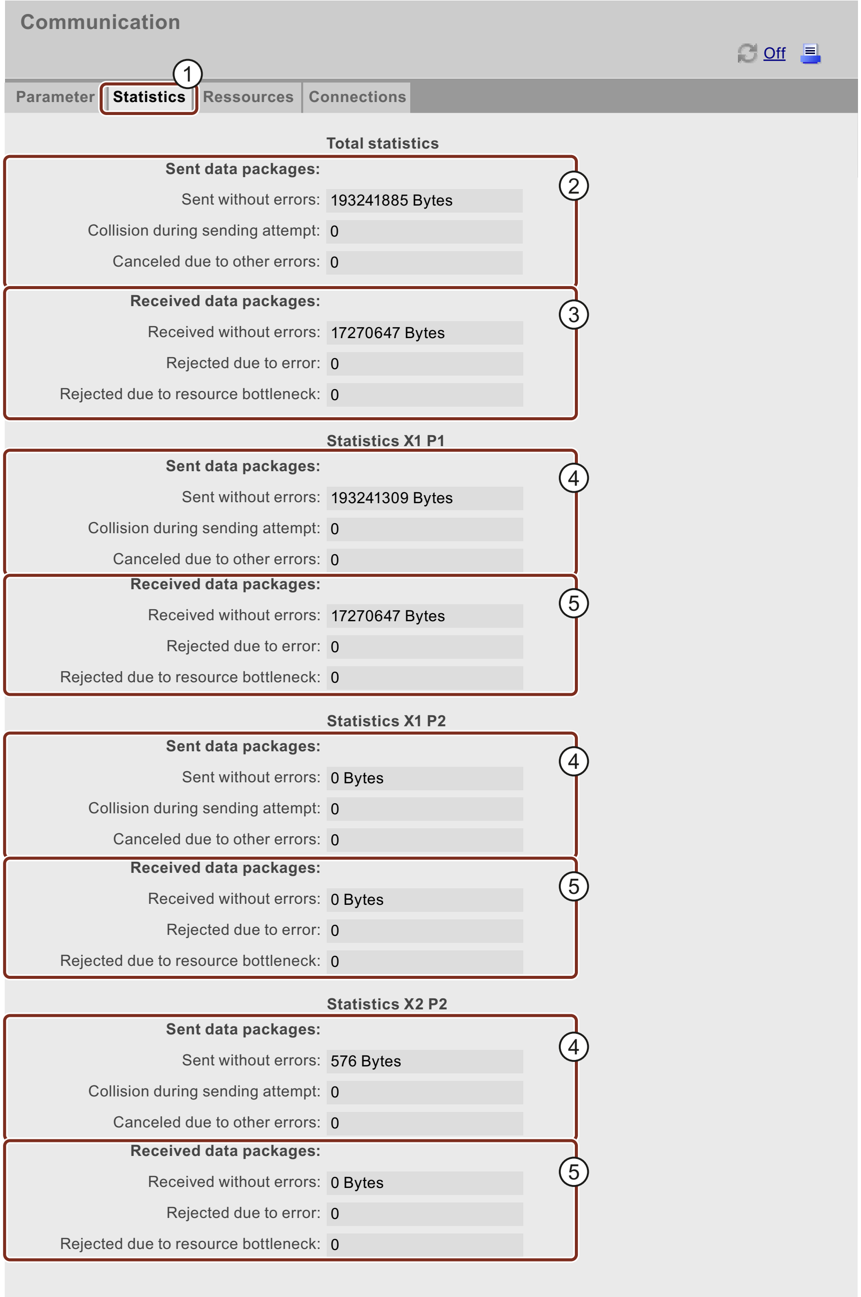
⑥ "Physical properties"

The following information on the interface hardware is available in the "Physical properties" field:

- Port number
- Link status
- Settings
- Mode
- Connection medium

① "Statistics" tab

Information on the data transmission can be found on the "Statistics" tab.



Total statistics

Sent data packages:
Sent without errors: 193241885 Bytes
Collision during sending attempt: 0
Canceled due to other errors: 0

Received data packages:
Received without errors: 17270647 Bytes
Rejected due to error: 0
Rejected due to resource bottleneck: 0

Statistics X1 P1

Sent data packages:
Sent without errors: 193241309 Bytes
Collision during sending attempt: 0
Canceled due to other errors: 0

Received data packages:
Received without errors: 17270647 Bytes
Rejected due to error: 0
Rejected due to resource bottleneck: 0

Statistics X1 P2

Sent data packages:
Sent without errors: 0 Bytes
Collision during sending attempt: 0
Canceled due to other errors: 0

Received data packages:
Received without errors: 0 Bytes
Rejected due to error: 0
Rejected due to resource bottleneck: 0

Statistics X2 P2

Sent data packages:
Sent without errors: 576 Bytes
Collision during sending attempt: 0
Canceled due to other errors: 0

Received data packages:
Received without errors: 0 Bytes
Rejected due to error: 0
Rejected due to resource bottleneck: 0

Figure 4-34 "Statistics" tab with key data on data transmission

② "Total statistics - Sent data packages"

You can assess the data transmission on the transmit line based on the key data in this info box.

③ "Total statistics - Received data packages"

You can assess the data transmission on the receive line based on the key data in this info box.

④ "Statistics Port x - Sent data packages"

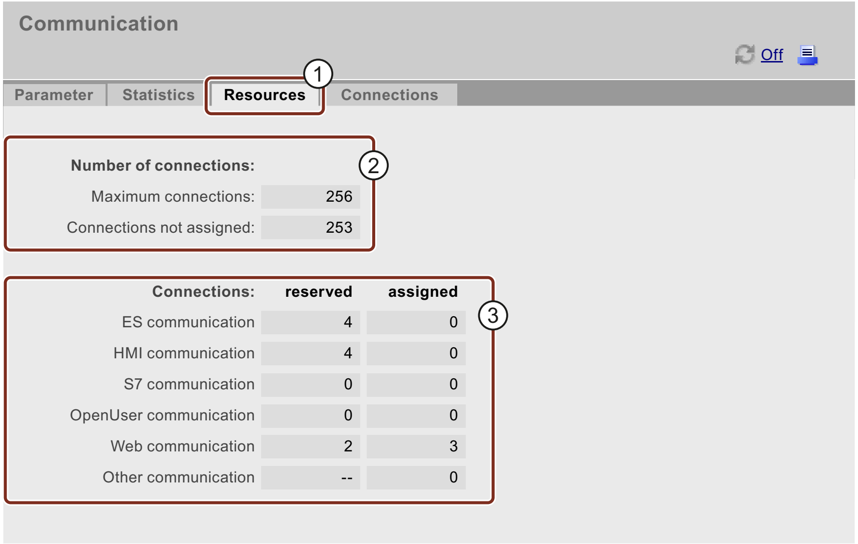
You can assess the data transmission on the transmit line for each port based on the key data in this info box.

⑤ "Statistics port x - Received data packages"

You can assess the data transmission on the receive line for each port based on the key data in this info box.

① The "Resources" tab

For information about the resource consumption of the connections, refer to the "Resources" tab.



Number of connections:	
Maximum connections:	256
Connections not assigned:	253

Connections:	reserved	assigned
ES communication	4	0
HMI communication	4	0
S7 communication	0	0
OpenUser communication	0	0
Web communication	2	3
Other communication	--	0

Figure 4-35 "Resources" tab

② Number of connections

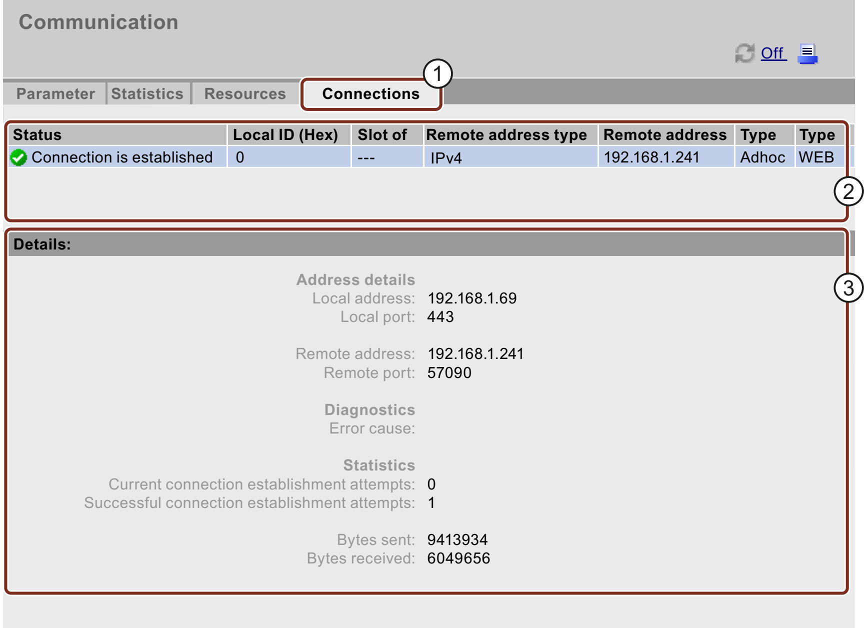
Under "Number of connections", you will find information on the maximum number of connections and the number of connections not assigned.

③ Connections

The item "Connections" provides information on the number of connections reserved or used for ES, HMI, S7, OpenUser, web communication and other communication functions.

① "Connections" tab

The "Connections" tab contains information on the status of the communication connections.



Status	Local ID (Hex)	Slot of	Remote address type	Remote address	Type	Type
✓ Connection is established	0	---	IPv4	192.168.1.241	Adhoc	WEB

Details:

Address details
Local address: 192.168.1.69
Local port: 443

Remote address: 192.168.1.241
Remote port: 57090

Diagnostics
Error cause:

Statistics
Current connection establishment attempts: 0
Successful connection establishment attempts: 1

Bytes sent: 9413934
Bytes received: 6049656

Figure 4-36 "Connections" tab

② Status

Under "Status", you will find an overview of the communication connections being established and the already established communication connections.

For each of these connections, the table contains information about the connection status, the local ID, the slot of the gateway, the remote address (IP address), the corresponding remote address type, and the connection types.

③ Details

Under "Details", you will find detailed information about the selected connection.

Reference

For an explanation of the error message displayed when a connection is interrupted or an attempt to establish a connection fails, refer to the STEP 7 online help.

4.9 Topology

4.9.1 Introduction

Topology of the PROFINET devices

The "Topology" web page provides information on the topological configuration and status of the PROFINET devices on your PROFINET IO system.

There are three tabs for the following views:

- Graphical view (set and actual topology)
- Table view (actual topology only)
- Status overview (excluding topological correlations)

You can print the table view and status overview. Before printing, use the print preview of your browser and, if necessary, correct the format.

Set topology

The set topology is displayed if you have topologically interconnected the connections in the configuration with STEP 7.

This view identifies the topological assignment of PROFINET devices that have failed, the differences between the set and actual topology, and interchanged ports.

NOTE

The configured set topology is always displayed by default in the following scenarios:

- When the "Topology" web page is called via the navigation bar
- When you change from the overview of PROFINET IO devices on the "Module information" web page to the "Topology" web page by means of the "Topology" link.

If a setpoint topology was not configured, the actual topology is displayed.

Actual topology

Displays the current topological structure of the "configured" PROFINET devices of a PROFINET IO system and the directly adjacent, non-configured PROFINET devices (display of the neighbor relationships, provided these can be determined; but the status of these adjacent PROFINET devices is not displayed).

4.9.2 Graphical view

Requirements

For error-free operation of the topology, the following conditions must be met:

- You have made the Language settings (Page 37).
- In the Topology Editor of STEP 7, you configured the topological interconnection of ports (requirement for display of the set topology and the corresponding topological target connections).
- The project has been compiled in STEP 7.
- The project is completely loaded.

Set and actual topology - graphical view

You can select the interface with the topology you want to display (X1, X2, X3 or PROFINET communication modules such as CM 1542-1) at the top left of the "Topology" Web page.

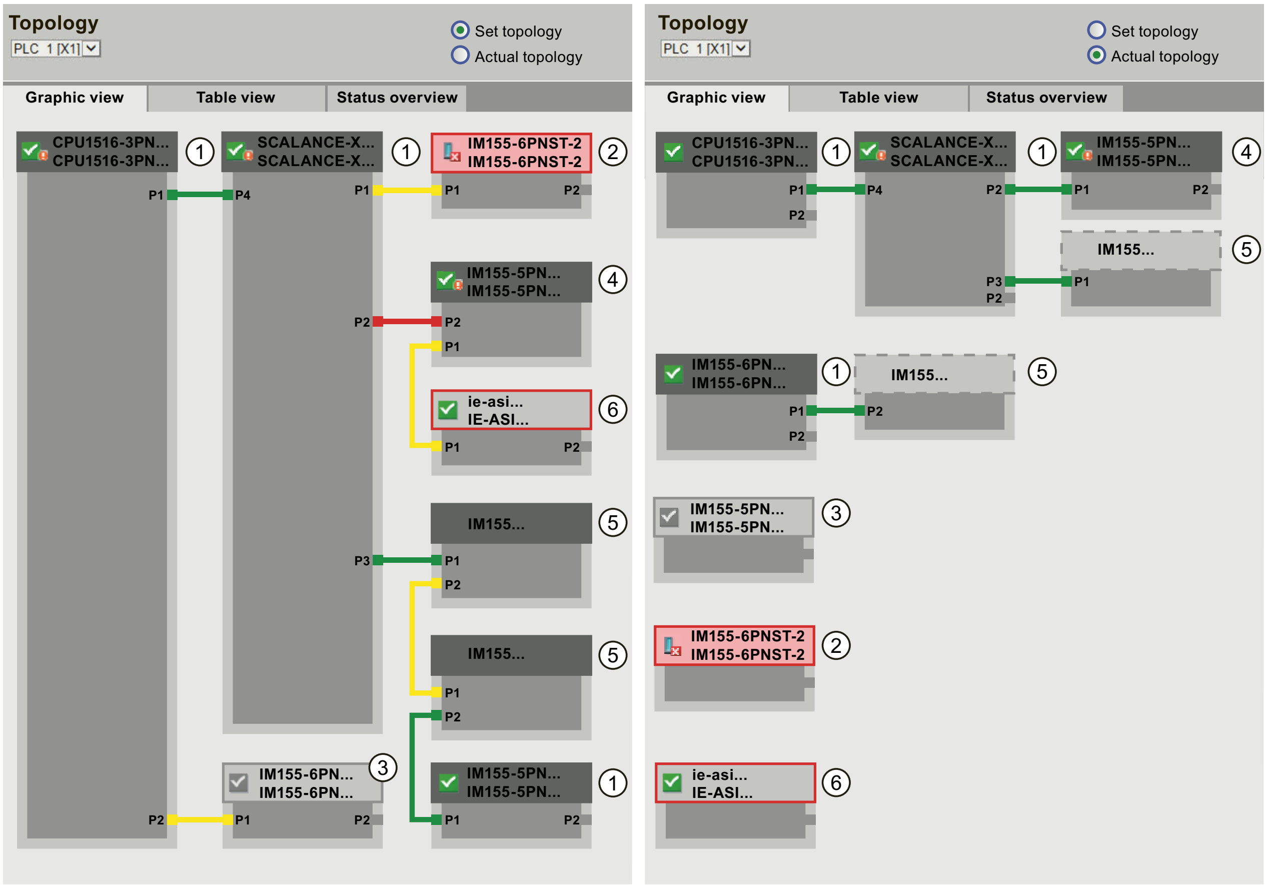


Figure 4-37 Graphical view - Set and actual topology

Meaning of the colored connections in the set/actual topology:

Table 4-3 Meaning of the colored connections in the set/actual topology

Connection	Meaning	
	Set topology	Actual topology
Green	The current actual connection matches the configured target connection.	Connections detected
Red	Mismatch between the current actual connection and the configured target connection (e.g., port interchanged).	-
Yellow	Connection diagnostics not possible. Causes: • Malfunction of communication with a device (e.g., cable was removed) • Connection to a passive component (e.g., switches or cables) • Connection to devices/PROFINET devices on a different IO controller or IO subsystem.	-

① Configured and accessible PROFINET devices

Configured and accessible PROFINET devices are displayed in dark gray. Connections indicate the ports used to connect the PROFINET devices of a station.

② Configured but inaccessible PROFINET devices

Configured but inaccessible PROFINET devices are indicated in pink with red frame (e.g., device failure, cable disconnected).

③ Deactivated devices

All deactivated, configured PROFINET devices are displayed in light gray.

④ Interchanged ports

Interchanged ports are highlighted in red in the set topology view. The actual topology view indicates the actually connected ports, while the set topology view displays the configured target connections.

⑤ PROFINET devices of a different PROFINET IO subsystem

- In the set topology:
  A PROFINET device of a different PROFINET IO subsystem is indicated by means of a green link (or red link for interchanged ports) if it is available on the bus and directly adjacent to an accessible configured PROFINET device ①. If the PROFINET device of a different PROFINET IO subsystem is inaccessible, it is identified by means of a yellow connecting line.
  The connection between two PROFINET devices which both belong to a different PROFINET IO subsystem cannot be identified and is always indicated in yellow color.
- In the actual topology:
  The PROFINET device of a different PROFINET IO subsystem is not displayed unless it is directly adjacent to a configured PROFINET device. The PROFINET device is shown in light gray with a dashed line around the device header.

The status of PROFINET devices of a different PROFINET IO subsystem is **not** displayed in the device header.

⑥ Displaying faulty neighbor relationships

Devices from which the relation data could not be read completely or with error are highlighted in light gray with a red frame.

NOTE

Displaying faulty neighbor relationships

If a device does not have the matching firmware, the relationships cannot be displayed correctly. This means a firmware update of the respective device is required in case a faulty neighbor relationship is displayed.

Views after changes to the configuration

- If a device fails, it remains at the same position in the "Set topology" view. This error state is indicated with a red border around the device header and the icon ☒.
- If a device fails, it is displayed in the "Actual topology" view. This error state is indicated separately in the bottom area with a red border around the device header and the icon ☒.

Link between the "Topology" and "Module information" web pages

The two web pages, "Topology" and "Module information", are linked. A click on the header of a selected module in the topology view automatically takes you to this module on the "Module information" web page.
You can find additional information on this in the section Module information (Page 69).

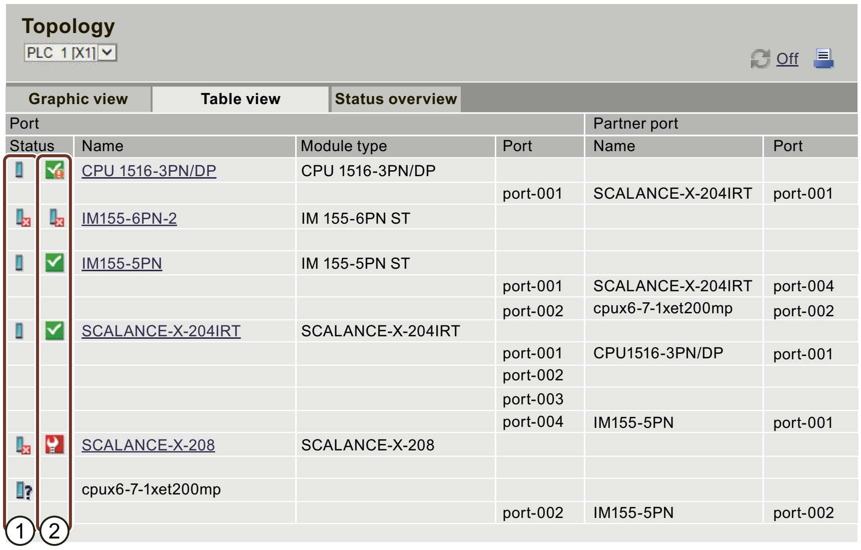
Reference

Additional examples for graphical topology view are available in the section Examples for graphical topology views (Page 90).

4.9.3 Tabular view

Topology - tabular view

The "Tabular view" always shows the "Actual topology".



Port					Partner port	
Status		Name	Module type	Port	Name	Port
▯	✅!	CPU 1516-3PN/DP	CPU 1516-3PN/DP			
				port-001	SCALANCE-X-204IRT	port-001
▯☒	▯☒	IM155-6PN-2	IM 155-6PN ST			
▯	✅	IM155-5PN	IM 155-5PN ST			
				port-001	SCALANCE-X-204IRT	port-004
				port-002	cpux6-7-1xet200mp	port-002
▯	✅	SCALANCE-X-204IRT	SCALANCE-X-204IRT			
				port-001	CPU1516-3PN/DP	port-001
				port-002		
				port-003		
				port-004	IM155-5PN	port-001
▯☒	🔧	SCALANCE-X-208	SCALANCE-X-208			
▯?		cpux6-7-1xet200mp				
				port-002	IM155-5PN	port-002

Figure 4-38 Topology - tabular view

① Meaning of the symbols relating to the status of the PROFINET devices

Table 4-4 Meaning of the symbols relating to the status of the PROFINET devices

Symbol	Meaning
▯	Configured and accessible PROFINET devices
▯?	Unconfigured and accessible PROFINET devices
▯☒	Configured but inaccessible PROFINET devices
▯!	Devices for which neighbor relations cannot be determined, or for which the neighbor relationship could not be read out completely, or only with errors

② Meaning of the symbols relating to the module status of the PROFINET devices

Table 4-5 Meaning of the symbols relating to the module status of the PROFINET devices

Symbol	Color	Meaning
✓	green	Component is OK.
✓	gray	Deactivated PROFIBUS or PROFINET devices.
?	black	State cannot be determined • For example, "Status cannot be determined" is always displayed while the CPU is in STOP mode, or during startup evaluation of "Report system error" for all configured I/O modules and I/O systems after a CPU restart. • However, this status can also be displayed temporarily during operation if a diagnostics interrupt burst occurs at all modules. • It is not possible to determine the status of modules on a subsystem that is connected to a CP.
☒	red	Component failed or is not reachable • "Not reachable" is displayed for e.g. a module that has been removed or a module that has been configured but does not exist.
🔧	green	Maintenance required (Maintenance Required)
🔧	yellow	Maintenance demanded (Maintenance Demanded)
🔧	red	Error - component faulty or not available due to an incorrect type.
!	-	A module in a lower module level does not have the status "Component OK".

Reference

For additional information on the "Report System Error" function, refer to the STEP 7 online help, keyword: "System diagnostics".

4.9.4 Status overview

Topology - status overview

The "Status overview" provides a clear presentation of all PROFINET IO devices/PROFINET devices (without connection relations) on one page. A quick error diagnostics is possible based on the symbols that show the module statuses.

The overview also provides a link of the modules to the Web page Module information (Page 69).

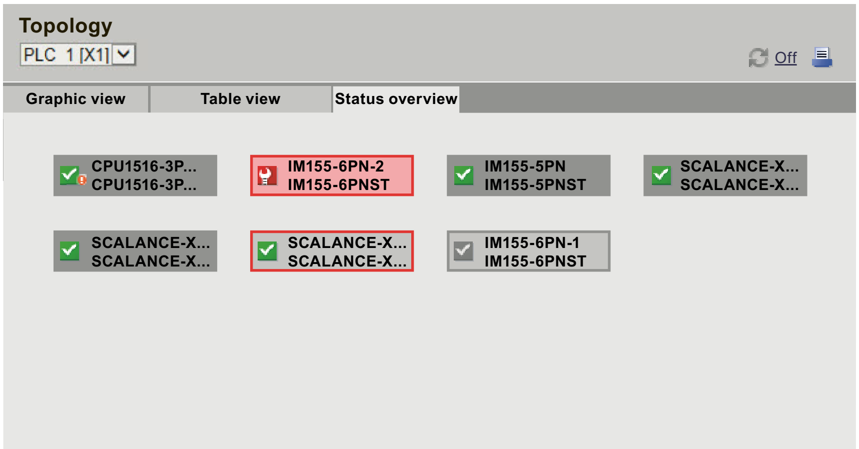


Figure 4-39 Topology - status overview

4.9.5 Examples for graphical topology views

The following section shows, as an example, some displays of the different topology views for a simple project.

"Set topology" is OK

Here you see the connections as they are configured in the topology editor by STEP 7. The configuration and wiring match.

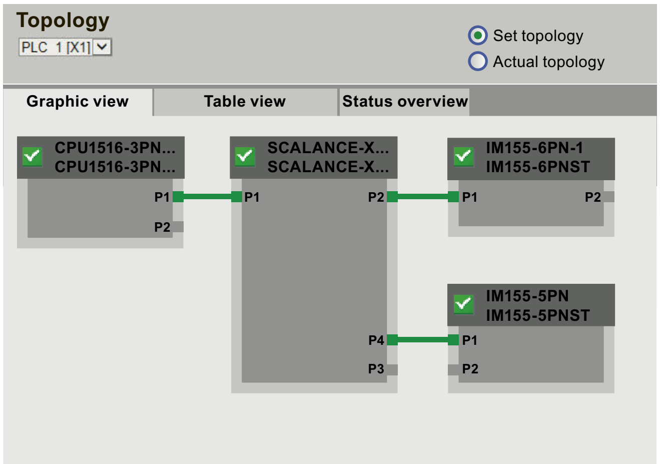


Figure 4-40 "Set topology" is OK

"Actual topology" is OK

Shows the actual layout of all configured devices that can be reached topologically.

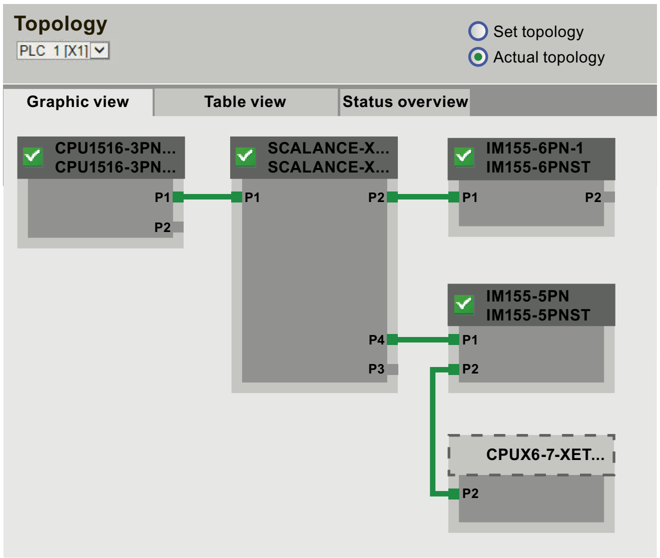


Figure 4-41 "Actual topology" is OK

"Set topology" with failed device

If a device has failed in the meantime, this device remains in the same place in the "Set topology" view. The failed device is displayed with a red border around the device header and the icon.

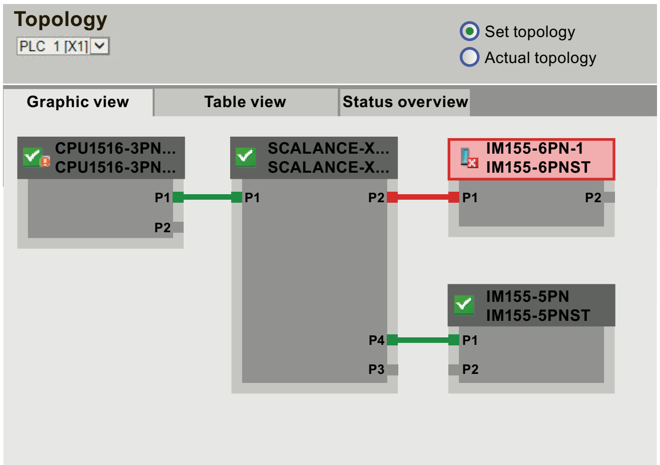


Figure 4-42 "Set topology" with failed device

"Actual topology" with failed device

In the "Actual topology" view, the device that has failed in the meantime is displayed separately in the bottom area of the view. The failed device is displayed with a red border around the device header and the icon.

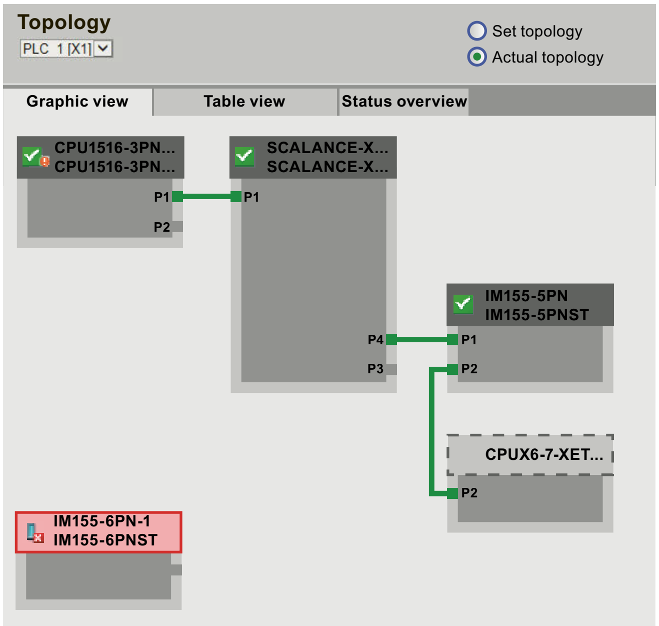


Figure 4-43 "Actual topology" with failed device

"Set topology" with interchanged ports

If a port was interchanged for a configured, directly adjacent PROFINET device, this device remains in the same place in the "Set topology" view. The interchanged connection is indicated by a red line.

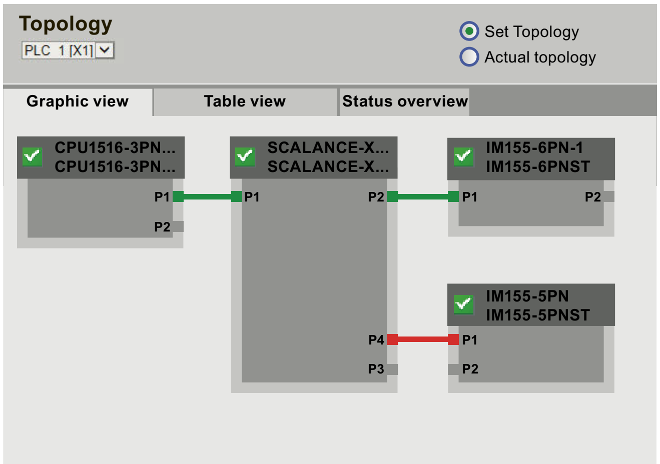


Figure 4-44 "Set topology" with interchanged ports

4.10 Tag status

Tag status

The web browser outputs the tag status on the web page of the same name.

NOTE

Saving the tag status as a bookmark

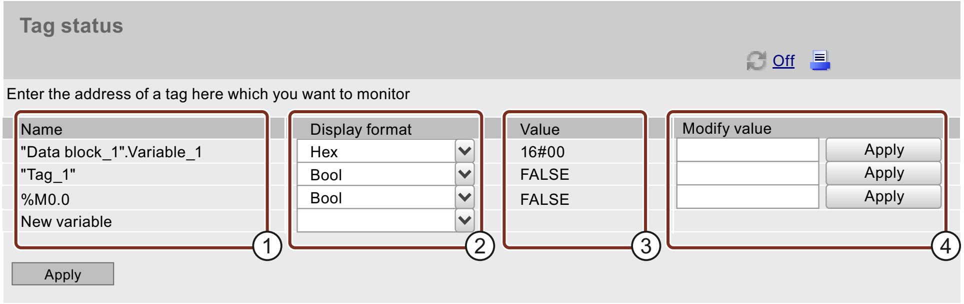
When the page is exited, the entries made on it are not saved. If you want to monitor the same entered tags again later on, then create a bookmark in your web browser for the "Tag status" page. Otherwise, you will have to enter the tags again when the page is reopened.

NOTE

Selected tag addresses are copied to the URL

The maximum number of characters for the URL of the tag status page is 2083. You can see the URL which corresponds to your current tag status page in the address bar of your web browser.

To monitor several tags, we recommend the use of the watch tables (Page 95).



Tag status

Off

Enter the address of a tag here which you want to monitor

Name	Display format	Value	Modify value	
"Data block_1".Variable_1	Hex	16#00		Apply
"Tag_1"	Bool	FALSE		Apply
%M0.0	Bool	FALSE		Apply
New variable				

Apply

Figure 4-45 Tag status

① "Name"

In the "Name" text box, enter the address of the tag whose behavior you want to monitor. This may be a symbolic or absolute address.

- PLC tags (inputs and outputs, bit memories, times and counters) and DB tags in blocks with standard access have an absolute and a symbolic address.
- DB tags in blocks with optimized access have a symbolic address and no absolute address.

Example for access to the absolute address of a data block with standard access:
The absolute address consists of the preceding address ID %, the number of the data block and the absolute address of the tags in the data block, separated by a period: `%DB1.DBX1.0` = absolute addressing of the tags "DBX1.0" in the global data block "DB1".

Invalid entries are displayed in red font.

② "Display format"

Using the drop-down list box, select the desired display format for the tag. If the tag cannot be displayed in the desired format, it will be displayed in hexadecimal format.

③ "Value"

Under "Value", the value of the corresponding operand is displayed in the selected format.

④ "Modify value"

You can change the value of tags and write them to the CPU in this column. To transfer several changed values in one operation, click the "Apply" button below the table.

To be able to read values and write values to the CPU, you need to have configured a user with the appropriate access rights in STEP 7.

If the value you entered is not valid (e.g. binary value in a BOOL field), the entry is not applied and the corresponding input field remains empty. A specific message relating to this is not output.

You can change the values of the following data types:

- Bool, Byte
- DWord, LWord, Word
- Int, DInt, LInt, SInt, UDInt, UInt, ULInt, USInt
- Real, LReal
- LDT
- Counter, Date
- Time, LTime, Time_Of_Day, LTime_Of_Day, Timer
- S5Time
- Char, WChar, String

NOTE

The following generally applies: To be able to write data, the "Referrer" transfer must be activated in your web browser (this is the default in all common browsers).

Special considerations when changing languages

You can change the language, e.g., from German to English, in the upper right corner. Note that the German mnemonics differ from those of the other languages.

For monitoring available data types

Basically, you can monitor all data types of PLC tags via the web Server, which you can also monitor in STEP 7.

Note that structured data types such as ARRAY, STRUCT and DTL are not available as data types for PLC tags due to their data volume.

Reference

You can find additional information on the available data types in the STEP 7 online help, keyword: "Overview of the valid data types".

4.11 Watch tables

Watch tables

The browser displays the content of the configured, web-enabled watch tables on the web page of the same name.

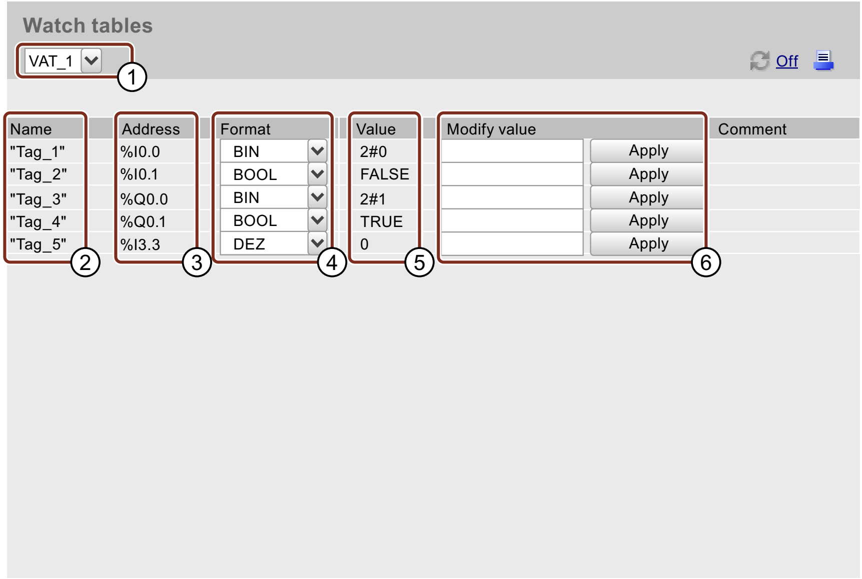
NOTE

Please note that you can observe a maximum of 50 of the watch tables configured in STEP 7 in the Web server.

Each of these tables is displayed in the Web server with a maximum of 200 entries.

If you are monitoring many large watch tables in the Web server, the update time may increase due to the large data volumes.

The number of watch tables that you can monitor download into the CPU also depends on the size of the SIMATIC memory card used.



Watch tables

VAT_1

Off

Name	Address	Format	Value	Modify value		Comment
"Tag_1"	%I0.0	BIN	2#0		Apply	
"Tag_2"	%I0.1	BOOL	FALSE		Apply	
"Tag_3"	%Q0.0	BIN	2#1		Apply	
"Tag_4"	%Q0.1	BOOL	TRUE		Apply	
"Tag_5"	%I3.3	DEZ	0		Apply	

Figure 4-46 Watch tables

① Selection

Select one of the configured watch tables from the drop-down list.

② "Name"

The symbolic name of the tag is shown in this info box.

③ "Address"

The absolute address of the tags is displayed within this info field (if present, e.g. for inputs or outputs; DB tags in blocks with optimized access have no absolute address).

③ "Format"

Select the display format of the respective tag from the drop-down list.

⑤ "Value"

This column shows the values in the corresponding display format.

⑥ "Modify value"

You can change the value of tags and write them to the CPU in this column.

To be able to read values and write values to the CPU, you need to have configured a user with the appropriate access rights in STEP 7.

If the value you entered is not valid (e.g. binary value in a BOOL field), the entry is not applied and the corresponding input field remains empty. A specific message relating to this is not output.

NOTE

The following generally applies: To be able to write data, the "Referrer" transfer must be activated in your Web browser (this is the default in all common browsers).

Note that the comments are displayed in the project language of the STEP 7 project that is assigned to the current user interface language of the Web server. You can find out how to assign project languages to interface languages in section Language settings (Page 37).

Reference

You can find additional information on the available data types in the STEP 7 online help, keyword: "Overview of the valid data types".

4.12 Online backup

Backing up and restoring the CPU configuration

You can back up a CPU configuration using the Web server with the corresponding access rights. If necessary, you can also restore this configuration at a later time using the Web server.

You can create as many backups as you want and store a variety of configurations for a CPU.

NOTICE

Prior to every restoration of the CPU configuration, always first perform an online backup of the current CPU configuration and save this backup file to a local directory of your PC.

This ensures that you can undo a restoration which failed (e.g. due to a damaged backup file) or which does not show the desired result.

NOTE

You can also perform online backup and restoration of the CPU configuration in STEP 7 (see STEP 7 online help, keyword: "Creating a backup of an S7 CPU").

When backing up using STEP 7, the backup file is saved within the STEP 7 project. With a backup using the Web server, the backup file is saved to a local directory of your PG/PC (e.g. "Downloads" directory). Web server backup files cannot be restored via STEP 7, nor can STEP 7 backup files be restored directly using the web server.

To restore a STEP 7 backup file using the web server, first save the STEP 7 backup file to a local directory of your programming device/PC (e.g. "Downloads" directory). From there, you can restore the backup with the Web server.

NOTE

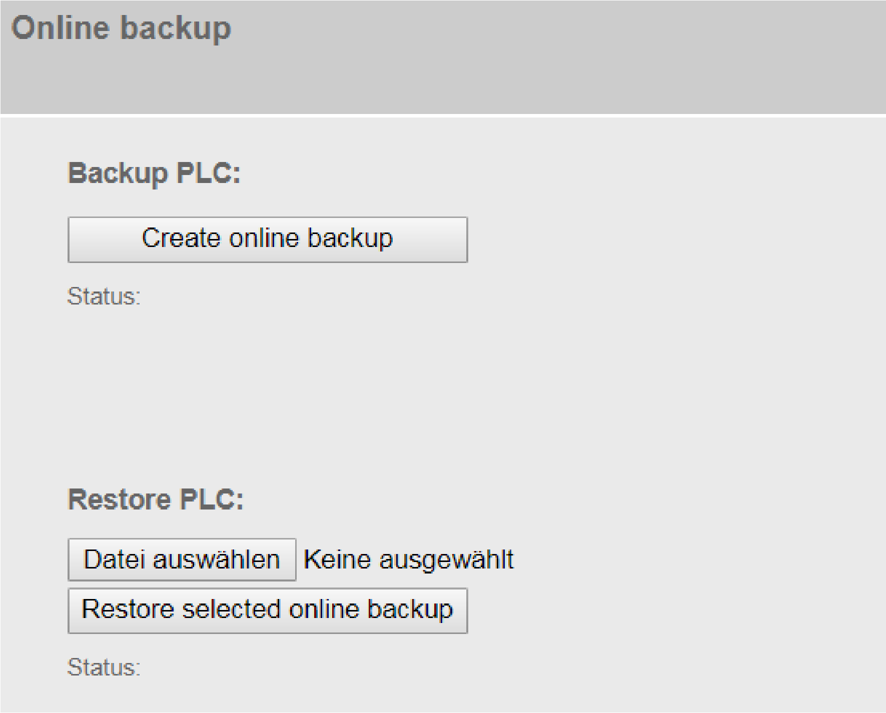
The "Online backup" function is not available if you access the web server via:

- a virtual IP address
- a communication module (CM)
- a communication processor (CP)

Requirements

- You access the CPU via the secure transmission protocol "HTTPS".
- A valid CA-signed certificate is installed in the Web browser; see section Configuring the Web server (Page 26).

Online backup



Online backup

Backup PLC:

Status:

Restore PLC:

Keine ausgewählt

Status:

Figure 4-47 Online backup

Perform online backup of the configuration

To perform an online backup of the CPU configuration, proceed as follows:

1. Click the "Create online backup" button in the "PLC backup" area.
2. If the CPU is in RUN mode, the following alarm is output:
   "Creation of an online backup requires PLC STOP. Do you want to set the PLC to STOP mode?"
   Acknowledge the alarm output by clicking "OK". The CPU is set to STOP mode and online backup is performed. (If you click "Cancel", the CPU remains in the current mode and the online backup is canceled.)
3. Save the backup file to a local directory of your PC.
4. Set the CPU back to RUN mode ("RUN" button in the "CPU operator panel" area of the start page).

NOTE

During the execution of the online backup, some data is not available in the web page view of the web server.

Scope of the backup

The backup includes all data needed to restore a particular state of a CPU, i.e. the specific combination of the configuration of the CPU with the current values of the user-related retentive data.

The following data of the configuration of the CPU is backed up:

- The contents of the SIMATIC memory card, e.g. configuration, program code, recipes and archives, DataLogs

The following user-relevant retentive data is backed up:

- Retentive memory areas of data blocks, bit memories, counters and timers
- Front-panel settings, dynamic IP configuration data, operating hours counters, retentive motion control sensor data

Note:

- Entries in the diagnostics buffer are not included in the backup.
- With a SIMATIC S7-1500 CPU, the current time is not saved.
- The complete content of the SIMATIC memory card is saved, i.e. also any data stored on the card (e.g. PDF files, GSD files).
- The backup file is assigned the name of the CPU and the project with the time and date of the backup, e.g. "2015-09-10_11-01_03_online backup_PLC69_machineControl.s7pbkp".
- The backup file of an F-CPU also contains the collective signature of the safety program in the file name. Check whether it is the expected F-collective signature.
- You can rename the backup, but you cannot make any changes to the contents of the backup.
- For the motion control technology objects positioning axis, synchronous axis and external encoder with absolute encoder, the position actual value does not match the actual mechanical axis position after the configuration is restored. Reference the technology object once again using absolute adjustment.

Restoring the configuration

To restore the CPU configuration, follow these steps:

1. Enter the password of the currently logged-in user in the "Restore PLC" area.
2. Click the "Select file" button and select the backup file of the configuration that you want to restore.
3. Click "Restore selected online backup".
4. If the CPU is in RUN mode, the following alarm is output:
   "Download online backup to device. The CPU is set to STOP and the contents of the CPU will be overwritten. Do you want to continue?"

   If the CPU is already in STOP mode, the following alarm is output:
   "Download online backup to device. The contents of the CPU will be overwritten. Do you want to continue?"

   Acknowledge the alarm output by clicking "OK". The CPU is set to "STOP" mode if required, and the online backup is downloaded. (If you click "Cancel", the CPU remains in the current mode and downloading is canceled.)
5. An alarm informs you that you must not leave the web page during the "restore procedure". Acknowledge the alarm output by clicking "OK".

   The restoration of the CPU configuration starts and you will be continuously informed of the current status:

   – "Download of online backup has been started."
   – "Checking backup file."
   – "Formatting memory card and resetting CPU."

6. If you have started the restoration procedure with a user name and password defined in the Web server configuration, you will be asked to enter these again after restoration of the CPU. Enter the required information and click "Login".

   If you have started the restoration procedure as the user "Everybody"/"Anonymous" without a password (but with appropriate access rights), this prompt is not displayed.

   NOTE

   To restore the configuration of an F-CPU whose security program and/or password has been changed for the F-CPU in the meantime, you also need the access authorization "Perform changes as F-Admin"; see "Amending user management" in the section Configuring the Web server (Page 26).

   ⚠ **WARNING**

   The authorization "Perform changes as F-Admin" on the Web server without password protection (User "Everybody" or "Anonymous") is only for test purposes, commissioning, etc. In other words, only when the system is not in productive operation. In this case, you have to ensure the security of the plant through other organizational measures, e.g. through spatial protection.

   Before the transition to productive operation, you must remove the right "Perform changes as F-Admin" from the user "Everybody" or "Anonymous".

   The password of the user of the web server with the right "Perform changes as F-Admin" must only be accessible to authorized persons.

   The restoration of the CPU configuration starts and you will be continuously informed of the current status:

   – "Loading configuration."
   – "Resetting CPU."

   This may take a few minutes.

7. When the procedure is complete, you will be logged out and the "Reload page..." button will be displayed.

   If you did not receive an error message during the restoration procedure, the restoration of the CPU configuration was successfully completed and you will receive a corresponding message.

   Click the "Reload page..." button and log on to the newly downloaded CPU configuration with your user name and password.

   You will receive the following error message if:

   – the newly downloaded CPU configuration does not contain the same IP address as the former one
   – the Web server is deactivated in the newly downloaded CPU configuration
   – the browser does not receive a response from the CPU after 3 minutes

   Error message: "The CPU is not reachable anymore. Please check the IP address and the Web server configuration. The result of the restore can be checked in the ASLog."

4.13 Record

Trace and logic analyzer function

You record device tags and evaluate the recordings with the trace and logic analyzer function. Tags are, for example, drive parameters or system and user tags of a CPU.

The recordings are saved on the device and can be read out by users with appropriate access rights via the Web server and saved. The trace and logic analyzer function is therefore suitable for monitoring highly dynamic processes in the Web server.

Requirements

- A trace configuration has been created, i.e. you have defined the recording and trigger conditions and selected the signals to be recorded.
  - Note: You can only display measurements stored on the SIMATIC Memory Card on the "Record" web page.
    In order for the CPU to save the measurements on the SIMATIC Memory Card, you must make the following settings in the trace configuration in STEP 7:
    1. Set the "Trigger mode" to "Trigger on tag".
    2. Select the "Save measurements on device (memory card)" check box.
- You have transferred the trace configuration to the device and activated it there.
- You have been assigned the access right "The user is authorized to..." > "...query diagnostics" in the user administration of the Web server; see section Configuring the Web server (Page 26).

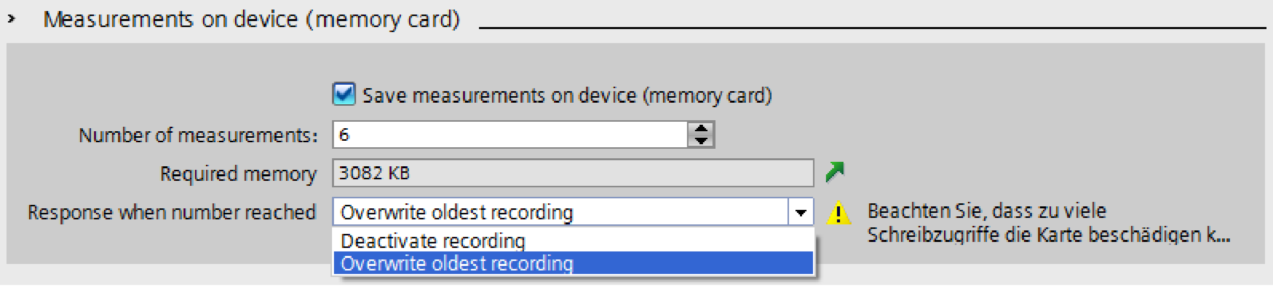
Space requirements for storing trace recordings

The "Save measurements on device (memory card)" function in STEP 7 saves Trace recordings on your SIMATIC Memory Card.

Response when number reached

The "Deactivate recording" parameter repeats the measurements until the configured "Number of measurements" is reached.

The "Overwrite oldest recording" parameter replaces the oldest measurement with the latest measurement when the configured "Number of measurements" is reached. Please note, however, that continuously writing data to the SIMATIC Memory Card shortens its service life.



Measurements on device (memory card)

☑ Save measurements on device (memory card)

Number of measurements: 6

Required memory 3082 KB

Response when number reached Overwrite oldest recording

Beachten Sie, dass zu viele Schreibzugriffe die Karte beschädigen k...

Figure 4-48 Dialog of the settings for saving measurements to the Memory Card in STEP 7

Number of measurements

The CPU supports a maximum of 999 measurements. While the CPU writes the trace recordings to the load memory of the memory card, it pauses monitoring of the trigger conditions for the trace job. After the CPU has terminated the storing of Trace recordings, the CPU continues checking of the trigger conditions.

NOTICE
Memory required on the SIMATIC Memory Card When the trace function "Measurements on device (memory card)" requires more memory than is available on the SIMATIC Memory Card, undesired effects may result. Ensure there is always sufficient free storage space to use the "Measurements on device (memory card)" function. In addition to the "Measurements on device (memory card)" Trace function, other functions, such as storing DataLogs, use storage space on the SIMATIC Memory Card. Make sure that enough memory space is available for all functions that occupy memory.

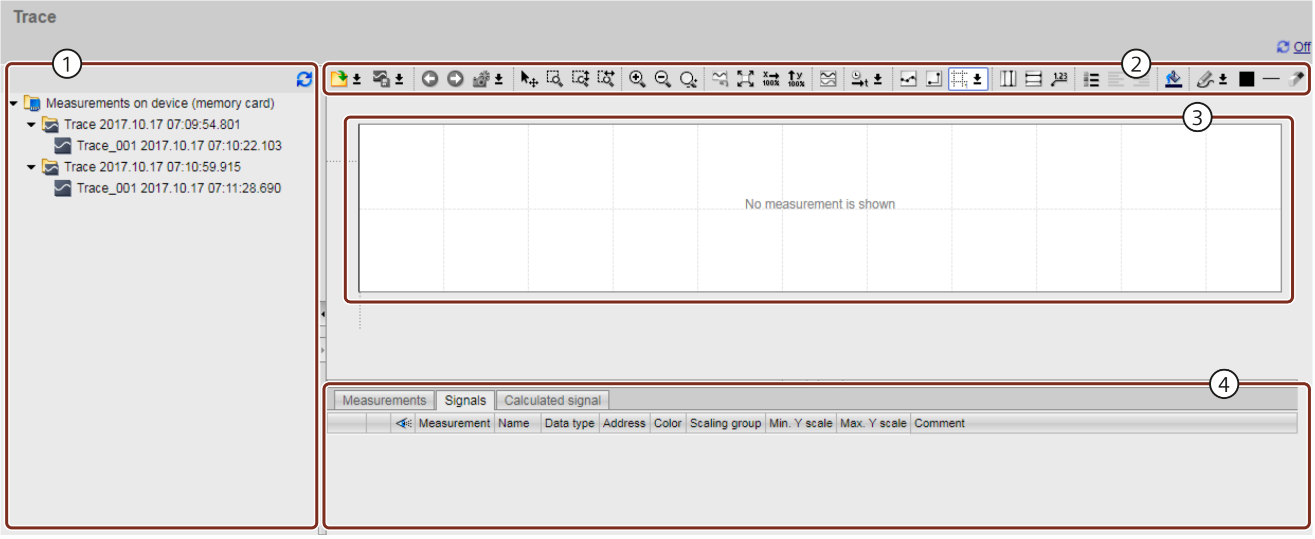
You can view the current values on the currently used storage space in the load memory in the "Memory" tab on the "Diagnostics" web page.

More information

The user interface of the "Trace" web page is largely the same as that of the trace function in STEP 7. See the Using the trace and logic analyzer Function Manual (https://support.industry.siemens.com/cs/ww/en/view/64897128) and the online help for STEP 7 for more details.

Displaying the trace recordings

The web page of the trace and logic analyzer function consists of several areas. The example in the figure below shows how the Web server user interface is divided when the "Trace" web page is first called.



Trace

- Measurements on device (memory card)
  - Trace 2017.10.17 07:09:54.801
    - Trace_001 2017.10.17 07:10:22.103
  - Trace 2017.10.17 07:10:59.915
    - Trace_001 2017.10.17 07:11:28.690

No measurement is shown

Measurement	Name	Data type	Address	Color	Scaling group	Min. Y scale	Max. Y scale	Comment

① Trace recordings

② Toolbar of the trend diagram

③ Trend diagram and bit trace

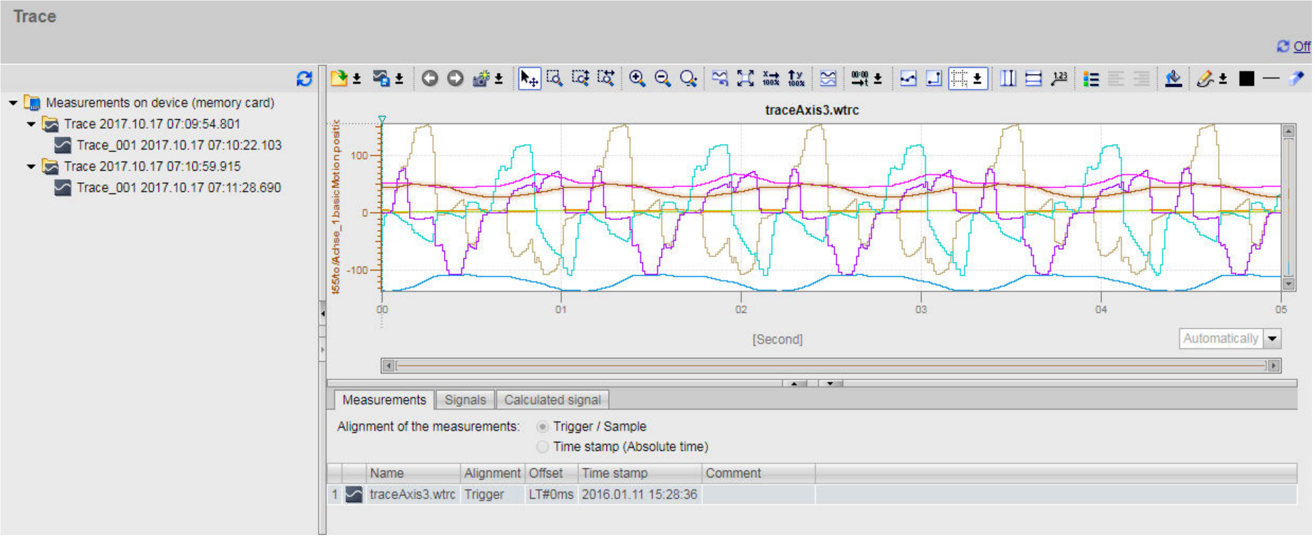
④ Signal tables

Figure 4-49 Trace start page without measurement

Opening measurements

To open a measurement, right-click on a measurement to select it from the "Trace recordings" area. Then select the command "Show in chart" in the shortcut menu.

The measurement is displayed in the "Trend diagram and bit trace" area.



Trace

Off

- Measurements on device (memory card)
  - Trace 2017.10.17 07:09:54.801
    - Trace_001 2017.10.17 07:10:22.103
  - Trace 2017.10.17 07:10:59.915
    - Trace_001 2017.10.17 07:11:28.690

traceAxis3.wtrc

Measurements | Signals | Calculated signal

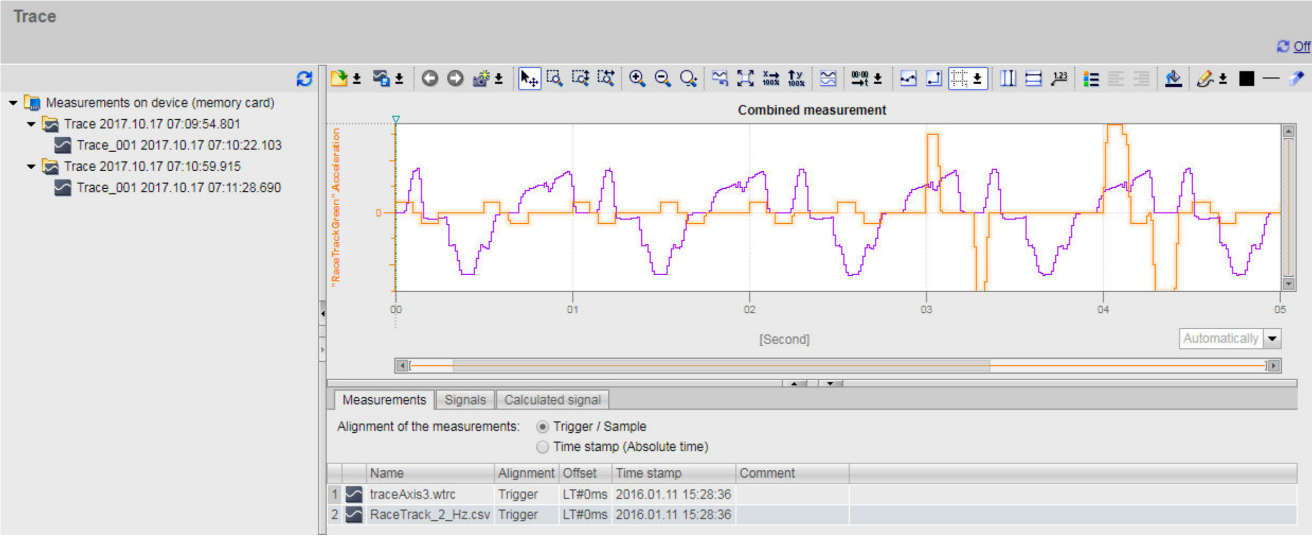
Alignment of the measurements: ● Trigger / Sample
○ Time stamp (Absolute time)

	Name	Alignment	Offset	Time stamp	Comment
1	traceAxis3.wtrc	Trigger	LT#0ms	2016.01.11 15:28:36	

Figure 4-50 Displaying an individual measurement

To display multiple measurements at once, right-click on a measurement to select it from the "Trace recordings" area. Then select the command "Add to table" in the shortcut menu.

The measurements are displayed in the "Trend diagram and bit trace" area.



Trace

Off

- Measurements on device (memory card)
  - Trace 2017.10.17 07:09:54.801
    - Trace_001 2017.10.17 07:10:22.103
  - Trace 2017.10.17 07:10:59.915
    - Trace_001 2017.10.17 07:11:28.690

Combined measurement

Measurements | Signals | Calculated signal

Alignment of the measurements: ● Trigger / Sample
○ Time stamp (Absolute time)

	Name	Alignment	Offset	Time stamp	Comment
1	traceAxis3.wtrc	Trigger	LT#0ms	2016.01.11 15:28:36	
2	RaceTrack_2_Hz.csv	Trigger	LT#0ms	2016.01.11 15:28:36	

Figure 4-51 Displaying several measurements

Trace recordings

The "Trace recordings" area shows a list of all existing measurements, sorted by date and time of the trace recordings. A measurement always consists of a trace configuration with an associated recording.

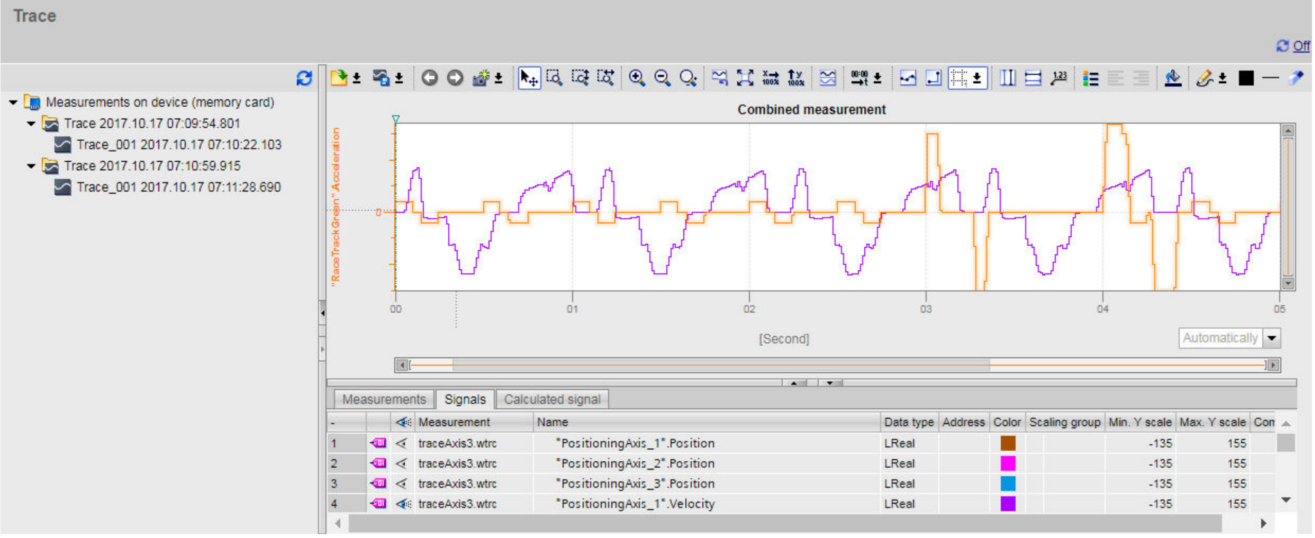
The following table shows the special Web server shortcut menu commands in the Trace recordings area:

Table 4-6 Web server shortcut menu commands in the trace recordings area

Shortcut menu command	Description
"Delete"	Deletes the selected measurement on the memory card of the CPU. A confirmation dialog opens. After deletion, the display in the trend diagram is not automatically overwritten.
"Save as"	Saves the selected measurement.
"Show in chart"	Loads the selected measurement to the display area of the Web server.
"Add in table"	Inserts the selected measurement into the table in the "Measurements" tab.

Some data types offer the display of individual bit traces. Activate the individual bit traces of the signal opened in the signal table using the ◁ icon.

You can adjust the display of the signals in the signal table and with the aid of the toolbar of the trend diagram.



- Measurements on device (memory card)
  - Trace 2017.10.17 07:09:54.801
    - Trace_001 2017.10.17 07:10:22.103
  - Trace 2017.10.17 07:10:59.915
    - Trace_001 2017.10.17 07:11:28.690

-			Measurement	Name	Data type	Address	Color	Scaling group	Min. Y scale	Max. Y scale	Con
1		◁	traceAxis3.wtrc	"PositioningAxis_1".Position	LReal		🟫		-135	155	
2		◁	traceAxis3.wtrc	"PositioningAxis_2".Position	LReal		🟪		-135	155	
3		◁	traceAxis3.wtrc	"PositioningAxis_3".Position	LReal		🟦		-135	155	
4		◁	traceAxis3.wtrc	"PositioningAxis_1".Velocity	LReal		🟪		-135	155	

Figure 4-52 Trace measurement - All areas visible

Toolbar of the trend diagram

As in STEP 7, the buttons in the toolbar of the trend diagram provide you with tools for adjusting the display.

The following table shows the Web server buttons in the trend diagram toolbar:

Table 4-7 Buttons of the trend diagram toolbar

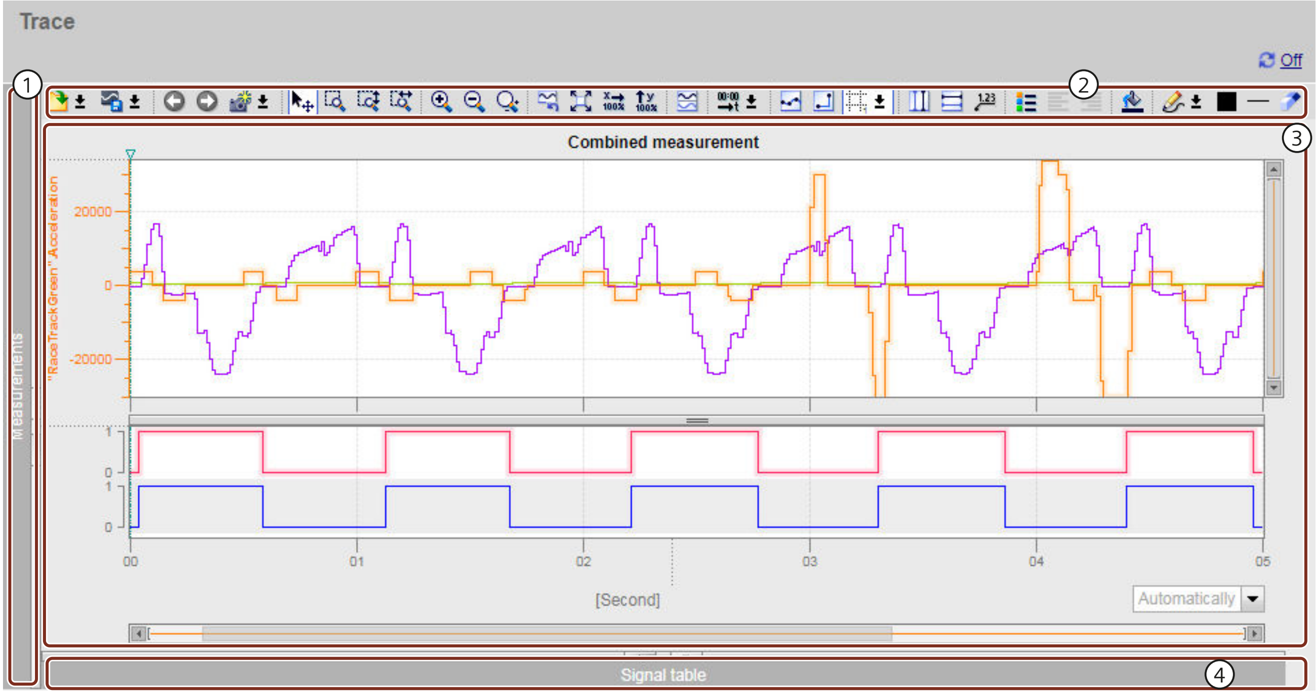
Symbol	Function	Description
📂 ⤓	Open / add measurement	Opens measurements or adds measurement to an existing measurement.
💾 ⤓	Save as	Saves measurement(s) as a file with the extension .csv, .wtrc (SIMOTION format for saving Trace data) or .ttrecx (TIA Portal format for saving Trace data). In addition to the measured data the command also saves the diagram, snapshots, marking and calculated signals. Condition for saving in the .wtrc format: • Only one measurement in the .wtrc format was loaded to the "Trace recordings" area. Condition for saving in the .ttrecx format: • Only one measurement in the .ttrecx format was loaded to the "Trace recordings" area. In all other cases, you can only save the measurements in the .csv format.
⬅	Undo move / zoom	Undoes the move / zoom function executed last. If you have carried out several move / zoom functions, you can undo these step-by-step.
➡	Repeat move / zoom	Redoes the last undone move / zoom function. If you have undone several move / zoom functions, you can redo these step-by-step.
📷 ⤓	Snapshot	Saves the current view as a snapshot (see the section "Settings and displays of the Snapshot symbol").
↖✥	Move view	Moves the display with a pressed mouse button - corresponds to the ✋ button in STEP 7.
⬚🔍	Zoom selection	Selection of an arbitrary range with the mouse button pressed. The button scales the display to the range selection.
⬚🔍↕	Vertical zoom selection	Selection of a vertical range with the mouse button pressed. The button scales the display to the range selection.
⬚🔍↔	Horizontal zoom selection	Selection of a horizontal range with the mouse button pressed. The button scales the display to the range selection.
🔍+	Zoom in	Enlargement of the display. The ranges of the X axis and Y axis are reduced every time the button is clicked. The curves are displayed larger.
🔍−	Zoom out	Reduction of the display. The ranges of the X axis and Y axis are reduced every time the button is clicked. The curves are displayed smaller.
🔍	Scaling	Scales all the signals or also just signal / signal group vertically and horizontally.
↶	Restore standard view	The button undoes scaling and move commands. The view is reset to the status at the time of loading of the measurement. Hidden signals are also reset but remain deactivated.
⛶	Display all	The button moves all the signals completely into the display area without changing the relative positions of the signals to each other.

Symbol	Function	Description
x→ 100%	Scale X automatically	Automatic scaling of all visible signals on the horizontal X area.
↑y 100%	Scale Y automatically	Automatic scaling of all visible signals on the vertical Y area.
	Arrange in tracks	Activate or deactivate the trace arrangement. When the trace arrangement is activated the signals are arranged among themselves with the relevant value axes. Signal groups are displayed in the same trace. This setting does not affect the display for the bit traces.
0.000 →t / →n	Unit switching of the time axis	Switching the unit of the time axis You can enter the following information: • Measuring points • Time (relative time related to the trigger time) • Time stamp of the measurement points
	Display measurement points	The button displays the measurement point as small circles on the curves.
/	Interpolation on/off	The buttons activate / deactivate the interpolation of the data of the trend diagram.
	Grid	The button activates / deactivates the grid of the trend diagram and regulates its brightness in the Levels 1 to 9.
	Vertical measurement cursor	Display of vertical measurement cursor. The vertical position of the two measurement cursors can be moved with the mouse. The values of the signals and the difference between two measuring points are displayed in the signal table for all displayed signals and also in the trend diagram for the selected signal. The measuring point or the relative/absolute time to the measurement cursors is displayed depending on the set unit of the time axis (X axis) in the movable pop-up window "Measuring points/Time values".
	Horizontal measurement cursor	Display of the horizontal measurement cursors. The horizontal position of the two measurement cursors can be moved with the mouse. The Y values of the measurement cursor for the selected signal are displayed in the movable pop-up window "Y values".
1.23	Difference of the measurement cursor	Display of the difference of the horizontal and vertical measurement cursors and the Y values at the intersections with the vertical measurement cursors.
	Show legend	Showing or hiding of the legend in the trend diagram and the bit trace labels.
	Align the chart legend to the left	Display of the legend and the bit trace labels on the left side of the trend diagram.
	Align the chart legend to the right	Display of the legend and the bit trace labels on the right side of the trend diagram.
	Change background color	Changeover between various background colors.
	Identification	The following table provides an overview of marked signal areas. Note that selections are possible only for analog and real signals (no calculated signals).

All icons in the toolbar are equipped with tooltips.

Trend diagram

The trend diagram displays the selected signals of a recording. Bits are shown in the lower diagram as a bit trace.



① Trace recordings (minimized)

② Toolbar of the trend diagram

③ Trend diagram and bit trace

④ Signal table (minimized)

Figure 4-53 Trace measurement - only trend diagram visible

The following table shows the special Web server shortcut menu commands of a selected signal in the trend diagram:

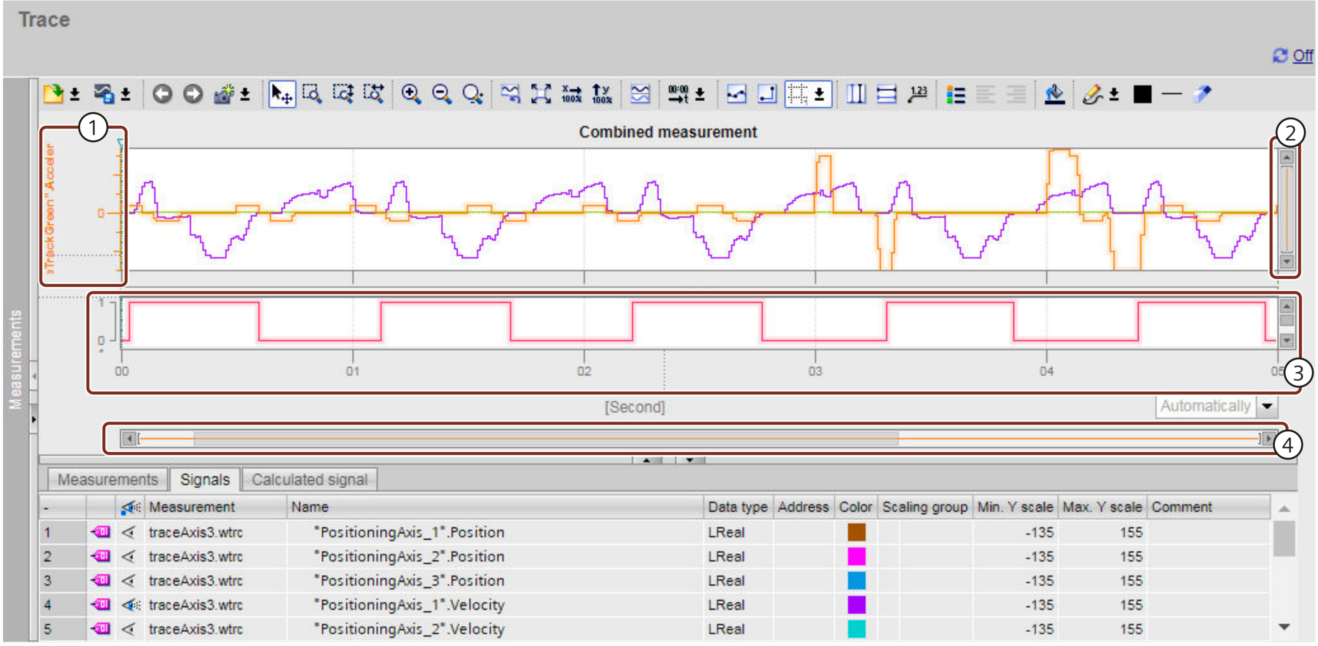
Table 4-8 Web server shortcut menu commands in the trend diagram area

Shortcut menu command	Description
"Scale Y automatically"	Automatic scaling of the selected signal in the Y direction.
"Hide signal"	Hides the selected signal in the trend diagram.

Use of the trend diagram

You can zoom the display area as you like. Measurement cursors (see "② Toolbar of the trend diagram") can be used to select individual values for display in the signal table.

The following image shows how you can change the display area of the trend diagram as required with rulers and scroll bars.



-			Measurement	Name	Data type	Address	Color	Scaling group	Min. Y scale	Max. Y scale	Comment
1			traceAxis3.wtrc	"PositioningAxis_1".Position	LReal		🟫		-135	155	
2			traceAxis3.wtrc	"PositioningAxis_2".Position	LReal		🟪		-135	155	
3			traceAxis3.wtrc	"PositioningAxis_3".Position	LReal		🟦		-135	155	
4			traceAxis3.wtrc	"PositioningAxis_1".Velocity	LReal		🟪		-135	155	
5			traceAxis3.wtrc	"PositioningAxis_2".Velocity	LReal		🟦		-135	155	

① Vertical ruler

② Vertical scroll bar

③ Horizontal ruler

④ Horizontal scroll bar

Figure 4-54 Trace measurement - rulers and scroll bars

Using the vertical ruler

- If you click the vertical ruler at the top or the bottom, you increase the size of the display at the top or bottom.
- If you click the vertical ruler at the top or the bottom while keeping the shift key pressed, you scale both ends.
- If you click the vertical ruler at the top or the bottom while keeping the Ctrl key pressed, you move the display up or down.

Using the horizontal ruler

- If you click the horizontal ruler on the left or the right, you increase the size of the display on the left or right.
- If you click the horizontal ruler on the left or right while keeping the shift key pressed, you scale both ends.
- If you click the horizontal ruler on the left or the right while keeping the Ctrl key pressed, you move the display to the left or right.

Using the mouse wheel

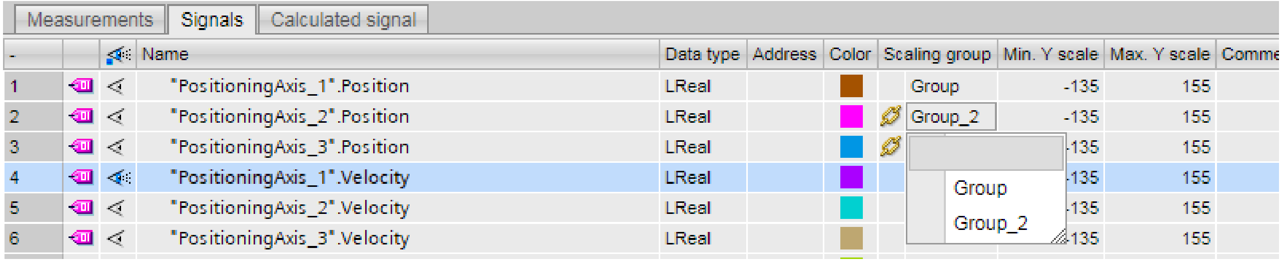
- If you activate the mouse wheel in the display, you move the display up or down.
- If you activate the mouse wheel in the display while keeping the shift key pressed, you move the display to the left or right.
- If you activate the mouse wheel in the display while keeping the Ctrl key pressed, you increase/reduce the size of the display at the position of the mouse pointer.

Signal tables

The signal tables list the signals of the selected measurement and provides setting options for some properties. The area of the signal tables is divided into the tabs "Measurements", "Signals" and "Calculated signal".

Settings and displays in the "Signals" tab

The following figure shows the signal table of the "Signals" tab.

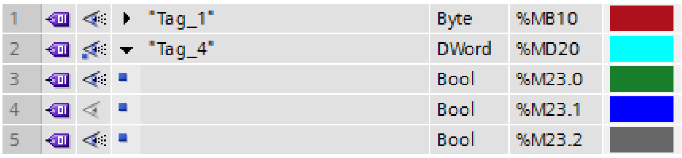


-		◁	Name	Data type	Address	Color	Scaling group	Min. Y scale	Max. Y scale	Comme
1	🏷	◁	"PositioningAxis_1".Position	LReal		🟫	Group	-135	155	
2	🏷	◁	"PositioningAxis_2".Position	LReal		🟪	Group_2	-135	155	
3	🏷	◁	"PositioningAxis_3".Position	LReal		🟦		-135	155	
4	🏷	◁	"PositioningAxis_1".Velocity	LReal		🟪		-135	155	
5	🏷	◁	"PositioningAxis_2".Velocity	LReal		🟦		-135	155	
6	🏷	◁	"PositioningAxis_3".Velocity	LReal		🟫		-135	155	

Figure 4-55 Display in the "Signals" tab

The following table shows the settings and displays of the recorded signals of the "Signals" tab:

Column	Description
Signal or error symbol	
🏷	Signal symbol
FO	Symbol for calculated signals (formulas)
◁	Selection for the display in the trend diagram
◁	The point indicates that at least one bit has been selected for display as bit trace for the signal in the bit selection.
Signal number	Automatically generated number of the signal The signal can be accessed via the signal number in the formulas.
Name	Display of the signal name A click on the name of a displayed signal updates the scale in the trend diagram.
▸	Open bit selection Individual bits can also be selected for the following data types for display as a bit trace in the lower trend diagram: • Byte, Word, DWord, LWord • SInt, USInt, Int, UInt, DInt, UDInt, LInt, ULInt

Column	Description
	Example of an opened bit selection for the DWORD data type:  1 ▶ "Tag_1" Byte %MB10 🟥 2 ▼ "Tag_4" DWord %MD20 🟦 3 Bool %M23.0 🟩 4 Bool %M23.1 🟦 5 Bool %M23.2 ⬛ Select or deselect the relevant bit for display by clicking the ◁ icon.
Data type	Display of the data type
Address	Display of the address of the signal The field remains empty with optimized / type correct tags.
Color	Display and setting option for the color of the signal
Signal group	Display or input of the signal group name for one signal group. The Y-scales are scaled identically for all signals of one signal group. Enter an identical signal group name for those signals that you want to scale identically. To remove signals from a signal group: • Delete the signal group name. • Click the empty entry in the shortcut menu of the signal group. Note that you cannot group binary signals.
Gray field for the chain icon of the signal group	To add or delete the signal from a signal group, move the mouse pointer over the grey field or the chain icon (🔗 or 🔗), Clicking the 🔗 chain icon adds the signal to a signal group or creates a new signal group. Clicking the 🔗 chain icon removes the signal from the signal group. For a selected signal with signal group, the 🔗 chain icon displays all signals of the same signal group.
Input field of the signal group	The input field displays the signal group name. As an alternative to the chain icon, you can assign or delete a group name via text input in this field.
Min. Y-scale	Display or input of the minimum value for the scaling of the signal.
Max. Y-scale	Display or input of the maximum value for the scaling of the signal.
Comment	Display and input option for a comment about the signal
Y(t1)	Display of the value at the position of the first measuring cursor
Y(t2)	Display of the value at the position of the second measuring cursor
ΔY	Display of the value difference between the first and the second measuring cursor

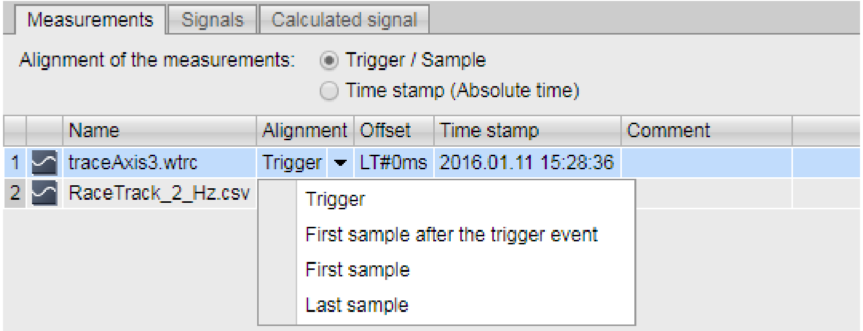
The following table shows the possible Web server shortcut menu commands of the "Signals" tab:

Table 4-9 Web server shortcut menu commands of the "Signals" tab

Shortcut menu command	Description
"Scale Y automatically"	Automatic scaling of the selected signal in the Y direction.
"Show signal"	Shows the signal in the trend diagram
"Hide signal"	Hides the signal in the trend diagram
"Show all bits"	Shows all the bits of a signal
"Hide all bits"	Hides all the bits of a signal
"Use for determined signal"	Calculates a new signal based on the selected real signal
"Process calculated signal"	Switches the selected calculated signal to the editing mode.
"Delete calculated signal"	Deletes the selected calculated signal

Settings and displays in the "Measurements" tab

The following figure shows the display of the "Measurements" tab and the shortcut menu of the "Alignment" column of a selected measurement.



Measurements | Signals | Calculated signal

Alignment of the measurements: ● Trigger / Sample
○ Time stamp (Absolute time)

		Name	Alignment	Offset	Time stamp	Comment
1	📈	traceAxis3.wtrc	Trigger ▾	LT#0ms	2016.01.11 15:28:36	
2	📈	RaceTrack_2_Hz.csv				

Figure 4-56 Tab "Measurements" with shortcut menu

The following table shows the settings and displays for the measurements:

Column	Description
Alignment of the measurements	
Trigger/Sample	Alignment of the measurements in accordance with the trigger or measurement point. The individual zero point for the measurement is predefined in the table under the "Alignment" column.
Time stamp (absolute time)	Alignment of the measurements in accordance with their time stamp. The signals are aligned in accordance with the time from the absolute time stamp.
Table columns	
📈	Static display of the measurement icon
Name	Display and change options for the name Note that the name must be unique.
Alignment	Alignment of the measurement (only adjustable with the "Trigger/Sample" check box selected). Determines the individual zero point for a measurement. All signals for the measurement are displayed in relation to this zero point. The following settings are possible: • Trigger • First sample after the trigger event • First sample • Last sampling
Offset	Offset related to the time axis Moves the measurement left or right by the offset stated on the time axis. If you enter solely a numerical value without a unit of measurement, the system automatically assigns the unit "ms" (for example 0=0ms, 100=100ms, 1000=1s, -1001=-1s 1ms , LT#2000ms=2s, LT#-3605000ms=-1h 5s , LT#-1h5s=-1h 5s)
Time stamp	Display of the trigger time
Comment	Display and input option for a comment about the signal
Shortcut menu commands	
"Save as WTRC"	Saves the measurement(s) as a file with the extension .csv, .wtrc (SIMOTION format for saving Trace data) or .ttrecx (TIA Portal format for saving Trace data).

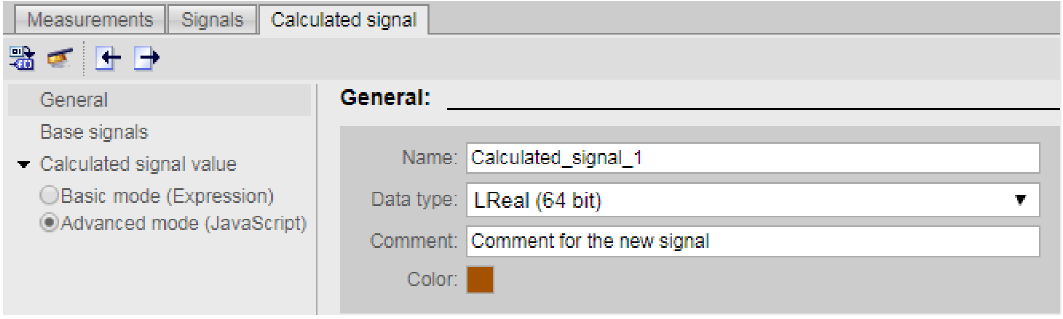
Column	Description
"Save as CSV"	Exports a measurement as a file with the file extension .´csv. Note that the command only saves the measured data. The command does not save the diagram, snapshots, markings and calculated signals.
"Edit name"	Switches the name in editing mode
"Edit offset"	Switches the offset in editing mode
"Edit comment"	Switches the comment in editing mode
"Delete measurement"	Deletes the measurement

Settings and displays in the "Calculated signal" tab

You can use this function to calculate new signals based on real signals. The system calculates the Y-values of the signal points in the process.

To parameterize the signals to be calculated, open the "Calculated signal" tab.

In the "General" section specify the name, the data type and color of a signal to be calculated. Note that the name of the signal to be calculated must differ from the name of a real signal.



General:

Name: Calculated_signal_1

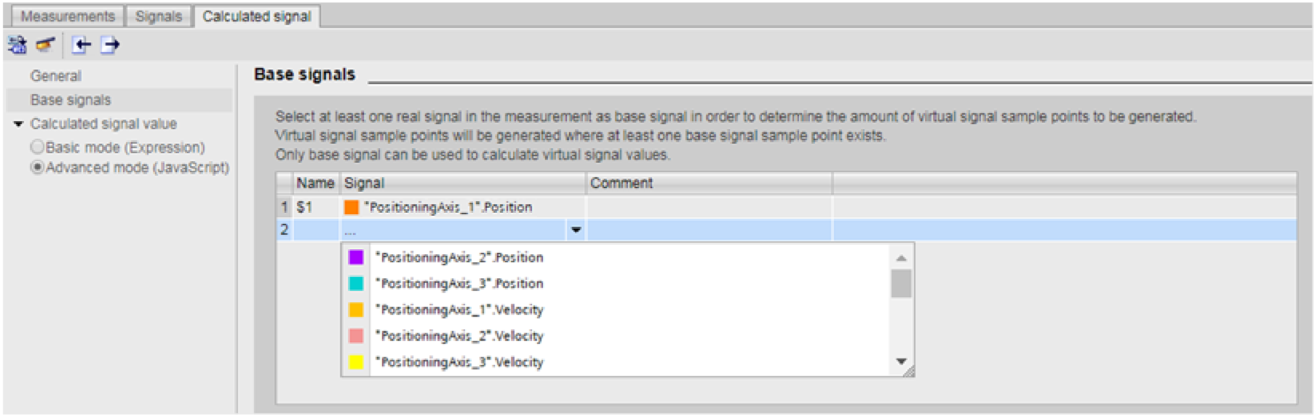
Data type: LReal (64 bit) ▼

Comment: Comment for the new signal

Color: 🟧

Figure 4-57 "General" section of the "Calculated signal" tab

Add the basic signals in the "Basic signals" section. The basic signals form the basis for calculating the new signal. You can change the default name of the variable in the "Name" column. You select real signals for the specification of the number of signal points to be calculated in the "Signal" column.



Base signals

Select at least one real signal in the measurement as base signal in order to determine the amount of virtual signal sample points to be generated.
Virtual signal sample points will be generated where at least one base signal sample point exists.
Only base signal can be used to calculate virtual signal values.

	Name	Signal	Comment
1	$1	🟧 "PositioningAxis_1".Position	
2		... ▼	

🟪 "PositioningAxis_2".Position
🟦 "PositioningAxis_3".Position
🟧 "PositioningAxis_1".Velocity
🟥 "PositioningAxis_2".Velocity
🟨 "PositioningAxis_3".Velocity

Figure 4-58 "Basic signals" section of the "Calculated signal" tab

Enter the code for the calculation of the Y values of the signal points in the "Calculated signal value" section. The section is divided into:

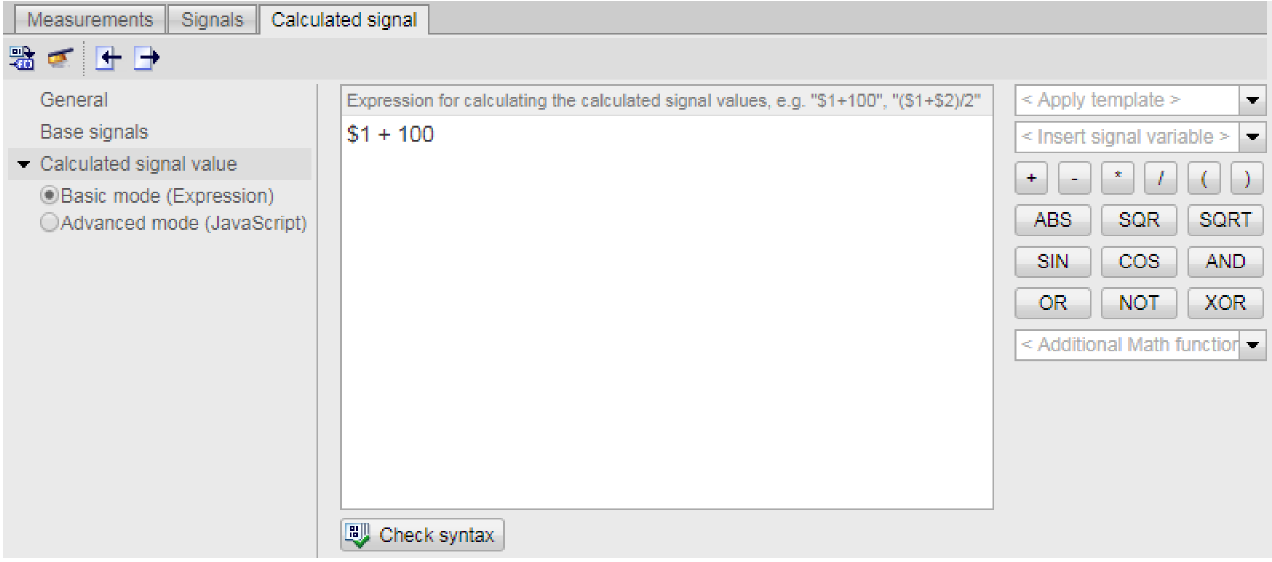
- Basic mode (expression)
- Advanced mode (JavaScript)

Use the JavaScript syntax to enter the code for both modes. Take into account, however, that not the full JavaScript functionality is supported.

Basic mode (expression)

In this mode you use the following to create your code:

- Standardized JavaScript expressions and operators (for example +, -, /, *, %, ~, &, |, ?, !)
- Standardized math libraries
- The tag names specified under "Basic signals"
  (the system writes the Y-value directly to the tag name)



● Basic mode (Expression)
○ Advanced mode (JavaScript)

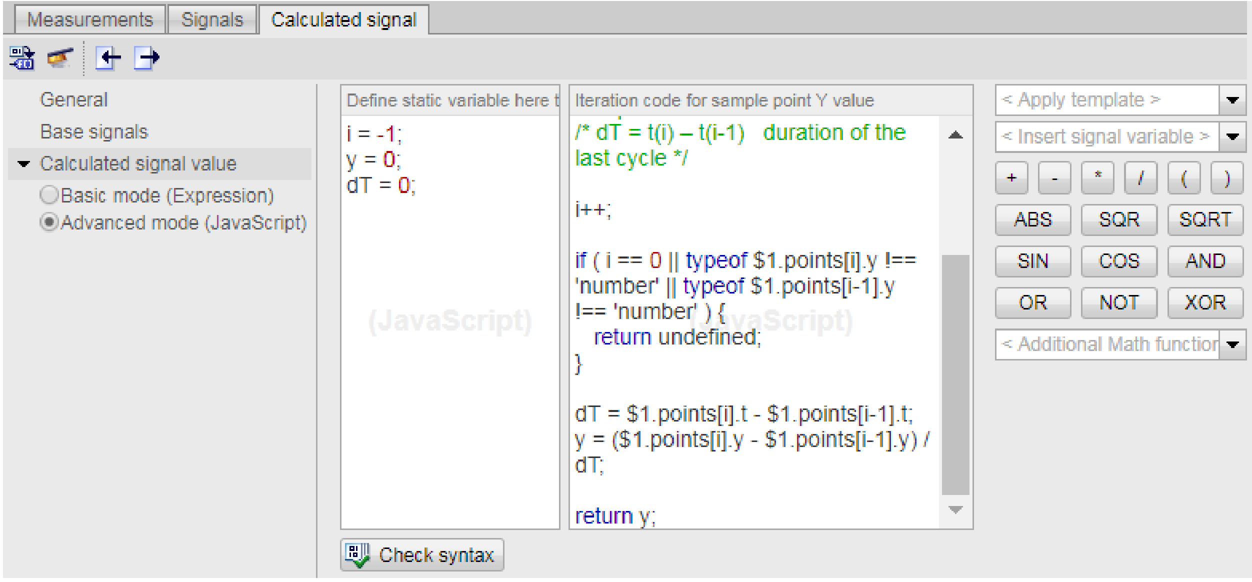
Expression for calculating the calculated signal values, e.g. "$1+100", "($1+$2)/2"

```
$1 + 100
```

Figure 4-59 Basic mode (expression)

Advanced mode (JavaScript)

This mode offers you an advanced functional scope for the calculation of the Y value by means of a complex JavaScript code. You can, for example, specify own static tags for the iteration of code sections:


Measurements
Signals
Calculated signal
General
Base signals
Calculated signal value
○ Basic mode (Expression)
● Advanced mode (JavaScript)
Define static variable here t
Iteration code for sample point Y value
(JavaScript)
(JavaScript)
< Apply template >
< Insert signal variable >
+
-
*
/
(
)
ABS
SQR
SQRT
SIN
COS
AND
OR
NOT
XOR
< Additional Math functior
Check syntax

```
i = -1;
y = 0;
dT = 0;
```

```
/* dT = t(i) – t(i-1)   duration of the
last cycle */

i++;

if ( i == 0 || typeof $1.points[i].y !==
'number' || typeof $1.points[i-1].y
!== 'number' ) {
   return undefined;
}

dT = $1.points[i].t - $1.points[i-1].t;
y = ($1.points[i].y - $1.points[i-1].y) /
dT;

return y;
```

Figure 4-60 Advanced mode (JavaScript)

The following table shows all the elements that you can use for your code:

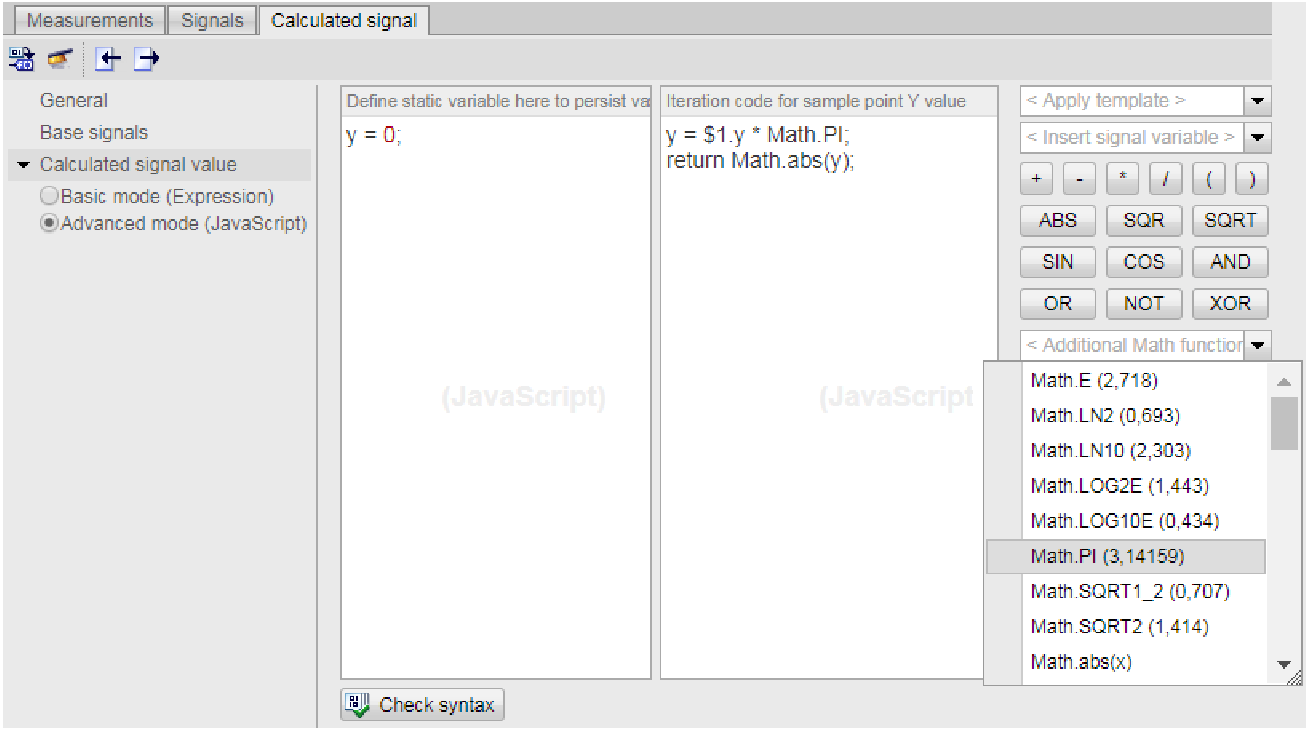
Reserved words (case-sensitive)						
if	var	Math	Array	catch	typeof	unescape
encodeURI	in	for	else	break	escape	delete
Infinity	decodeURI	do	Date	case	false	String
Object	continue	parseFloat	new	case	true	while
return	Number	parseInt	try	null	isNaN	throw
switch	default	undefined				
Operators						
%	-	*=	!=	<=	|=	&&
!==	+	--	/	>	=	||
^	===	++	-=	/=	>=	==
&	~	+=	*	!	<	|
&=	?					
Delimiters						
(	)	{	}	[	]	.
,	:	;				
Comments						
/*	*/					

Every signal point consists of the following attributes:

1. x (measuring point)
2. t (relative ^time in milliseconds)
3. y (Y value)
4. points (number of signal points that are available for the calculation of a new signal)

In "Advanced Mode (JavaScript)" you can access all four attributes (e.g. "$1.y", "$1.t", "$1.x", $1.points[i].y, ...).

The following function tables support you in writing expressions or complex code in JavaScript. The function table contains the functions used most often:



○ Basic mode (Expression)
● Advanced mode (JavaScript)

Define static variable here to persist va

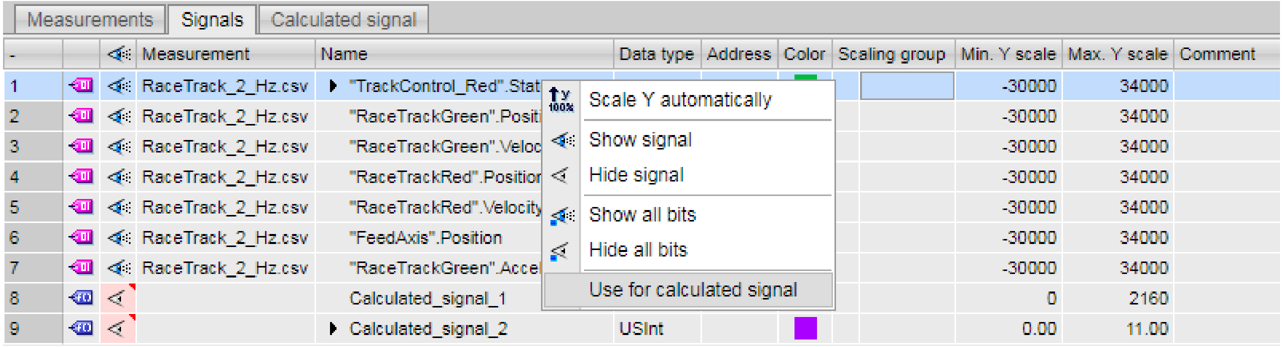
```
y = 0;
```

Iteration code for sample point Y value

```
y = $1.y * Math.PI;
return Math.abs(y);
```

Figure 4-61 "Calculated signal" tab with function table

You can also generate the JavaScript code of a real signal from the "Signals" tab. Alternatively, you can also select one of predefined templates of the function table, change the code and generate a calculated signal.

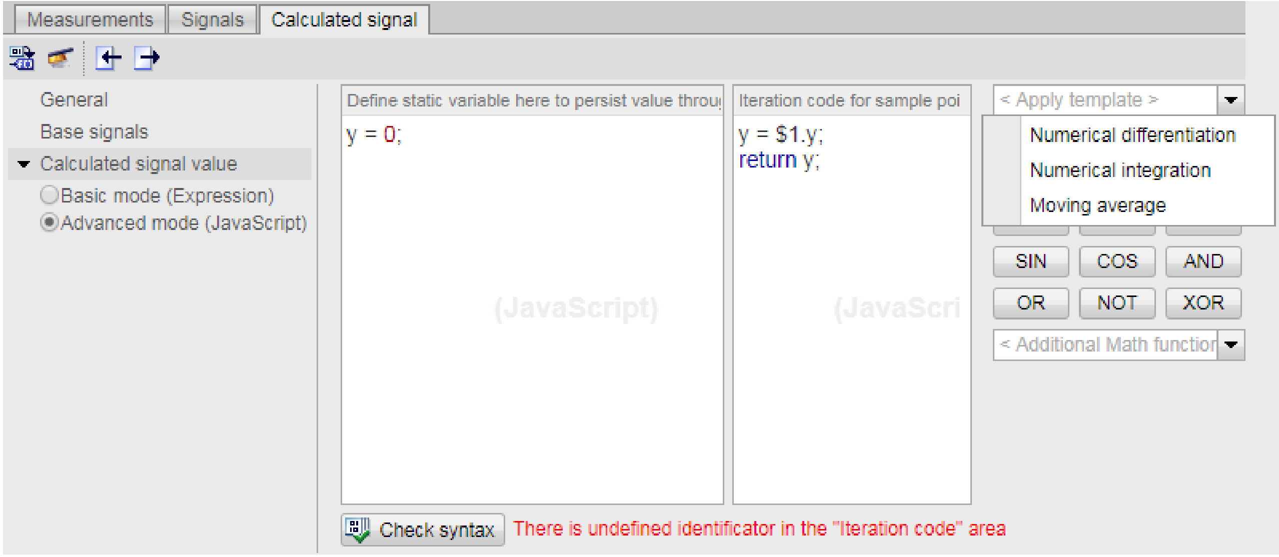


-			Measurement	Name	Data type	Address	Color	Scaling group	Min. Y scale	Max. Y scale	Comment
1			RaceTrack_2_Hz.csv	"TrackControl_Red".Stat			🟩		-30000	34000	
2			RaceTrack_2_Hz.csv	"RaceTrackGreen".Positi					-30000	34000	
3			RaceTrack_2_Hz.csv	"RaceTrackGreen".Veloc					-30000	34000	
4			RaceTrack_2_Hz.csv	"RaceTrackRed".Position					-30000	34000	
5			RaceTrack_2_Hz.csv	"RaceTrackRed".Velocity					-30000	34000	
6			RaceTrack_2_Hz.csv	"FeedAxis".Position					-30000	34000	
7			RaceTrack_2_Hz.csv	"RaceTrackGreen".Accel					-30000	34000	
8				Calculated_signal_1					0	2160	
9				Calculated_signal_2	USInt		🟪		0.00	11.00	

Figure 4-62 Real signals of the "Signals" tab

The following templates are available in the function table:

- Numerical differentiation
- Numerical integration
- Arithmetic mean


Measurements
Signals
Calculated signal
General
Base signals
Calculated signal value
○ Basic mode (Expression)
● Advanced mode (JavaScript)
Define static variable here to persist value throu
Iteration code for sample poi
(JavaScript)
(JavaScri
< Apply template >
Numerical differentiation
Numerical integration
Moving average
SIN
COS
AND
OR
NOT
XOR
< Additional Math functior
Check syntax

```
y = 0;
```

```
y = $1.y;
return y;
```

There is undefined identificator in the "Iteration code" area

Figure 4-63 Templates

To check the syntax, click the symbol "Check syntax" or generate the signal by clicking the icon in the toolbar. If the code has any errors, these are displayed to the right of the "Check syntax" icon in red lettering. If the code does not have any errors, the "Syntax check successful" message is displayed.

How does the system calculate a new signal?

The system checks:

- Whether you have selected a basic signal for the calculation of a new signal
- The name of the signal to be calculated
- The syntax of your JavaScript code

Subsequently, the system defines the counting of the measuring points and executes the code for each measuring point to be calculated. In each iteration the system stores the measuring points of the new signal on the basis of the following four attributes:

- Measuring point
- Relative time
- Calculated Y value
- Signal point with x-, t, y-values

The course is displayed after calculation has been completed.

Example for calculating on the basis of a basic signal

You use a single basic signal to calculate the new signal. The basic signal consists of 1000 measuring points.

In this case, the system carries out your written code a thousand times. The calculated signal then consists of 1000 calculated signal points with the same x- and t-values, but with own y-values.

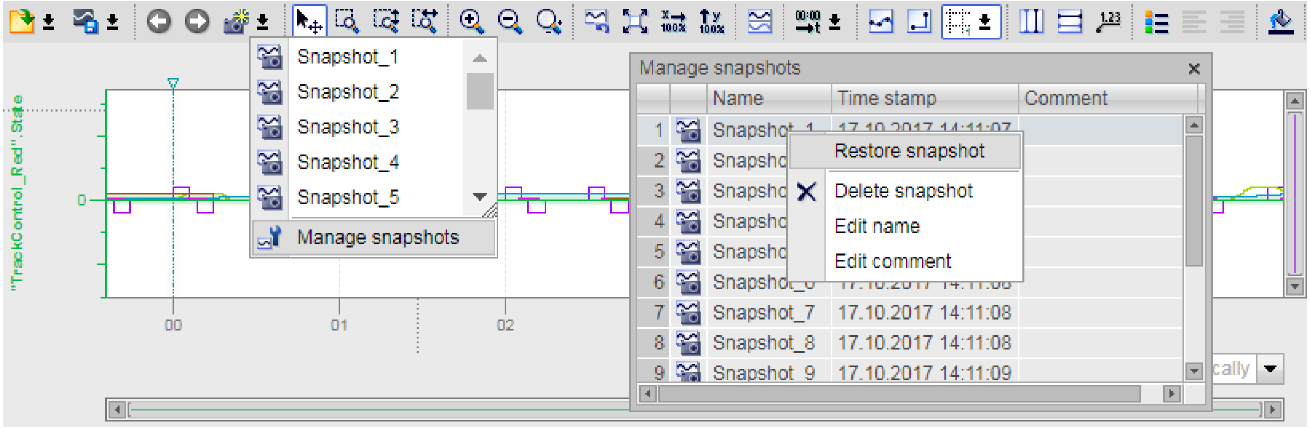
Import/export settings

You can import/export certain parameters (formulas, calculation method, signal type and signal name. To import parameters, click the icon in the toolbar of the "Calculated signal" tab. To export parameters, click the icon in the toolbar of the "Calculated signal" tab.

Settings and displays of the Snapshot icon

With the "Snapshot" icon in the toolbar of the trend diagram, you save the current signal course in the form of a snapshot.

To create a snapshot of the signal course click the icon. To manage the created snapshots, click the arrow on the left next to the "Snapshot" icon and select the entry "Manage snapshots".



		Name	Time stamp	Comment
1	📷	Snapshot_1	17.10.2017 14:11:07	
2	📷	Snapsho		
3	📷	Snapsho		
4	📷	Snapsho		
5	📷	Snapsho		
6	📷	Snapshot_6	17.10.2017 14:11:08	
7	📷	Snapshot_7	17.10.2017 14:11:08	
8	📷	Snapshot_8	17.10.2017 14:11:08	
9	📷	Snapshot_9	17.10.2017 14:11:09	

Figure 4-64 Managing snapshots

The following table shows the settings and displays of the "Managing snapshots" window.

Column	Description
📷	Static display of the snapshot icon
Name	Display and change options for the name
Time stamp	Display of the creation time of the snapshot
Comment	Display and input option for a comment

The following table shows the possible shortcut menu commands:

Shortcut menu command	Description
"Restore snapshot"	Shows the measurement with the saved view in the "Diagram" tab.
"Remove snapshot"	Deletes the snapshot.
"Edit name"	Switches the name in editing mode
"Edit comment"	Switches the comment in editing mode

NOTE

Delete measurement

If you delete a measurement of a calculated signal, all the associated snapshots are also deleted.

4.14 DataLogs

DataLogs

On the DataLogs web page, you can view all the DataLogs that you have created.

You can sort the DataLogs according to individual parameters in ascending or descending order.

For this purpose, click on one of the parameters in the column header:

- Name
- Size
- Changed

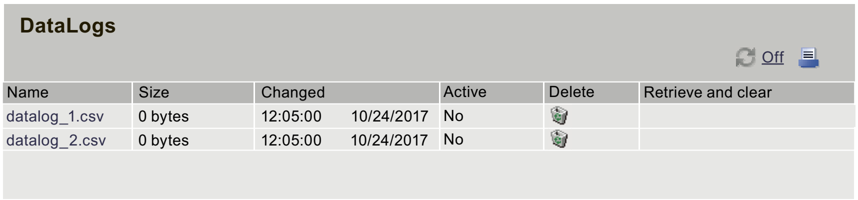
You can download the relevant DataLog file by clicking the file name.

The "Active" column shows whether the respective DataLog file is used (is active) or not.

When the DataLog file is active, you can call (download) and empty the relevant DataLog file by clicking the icon 💾. The file must be closed. The empty DataLog file is still maintained in the list of DataLogs.

You delete the DataLog file by clicking the 🗑 icon in the "Delete" column. The file must be closed.

You close an opened DataLog file in STEP 7 by using the "DataLogClose" instruction.



DataLogs

Name	Size	Changed	Active	Delete	Retrieve and clear
datalog_1.csv	0 bytes	12:05:00 10/24/2017	No	🗑	
datalog_2.csv	0 bytes	12:05:00 10/24/2017	No	🗑	

Figure 4-65 DataLogs

4.14.1 Automated reading out of DataLogs

In addition to the downloading of individual DataLogs via the user interface of the Web server, you can download, read out and archive DataLogs. Automatic downloading of DataLogs is realized either by the execution of scripts in, for example, Bash or on your HTML customer page via JavaScript.

A typical application for this functionality is the daily reading out and archiving of DataLogs from one or more CPUs at a specific time.

Calling up of the DataLogs from the SIMATIC memory card

The CPU makes a URL in the following format available so that you can download DataLogs automatically from the SIMATIC memory card of your CPU:

```
https://[ip]/DataLogs?Action=List
```

Enter the correct IP address of the interface of your CPU at this URL and use the appropriate transfer protocol (HTTP or HTTPS), for example `https://192.168.2.132/DataLogs?Action=LIST`. Subsequently call up the URL in your browser or command line interpreter.

The URL returns a list of the DataLogs on the SIMATIC memory card. Each entry returns the URL by which you download the corresponding DataLog.

For simple syntax analysis of the list by command line interpreters (such as Bash) or web-based programming languages (such as JavaScript) the individual URLs are separated by a line break `<CR><LF>`. The following example shows the syntax of two URLs that access the DataLog files `Test.txt` and `Test2.txt`:

```
/DataLogs?Path=/DataLogs/Test.txt&Action=DOWNLOAD&E=1<CR><LF>
/DataLogs?Path=/DataLogs/Test2.txt&Action=DOWNLOAD&E=1<CR><LF>
<CR><LF>
```

When the URLs are called successfully, the CPU returns the status code `200 OK`. The CPU also returns this status code if no DataLogs exist on the SIMATIC memory card.

NOTE

Access authorization to the CPU for the reading out of data

In order to download DataLogs from the CPU, the user has to dispose of reading rights on the CPU. If the user does not have the required rights, the CPU returns the status code `403 FORBIDDEN` in the HTTP Response.

Downloading of the DataLogs via Bash scripts

The following example shows how you can download DataLogs automatically from the CPU by using a Bash script. Replace the URL of the example by the correct IP address of the interface of your CPU at this URL and use the appropriate transfer protocol (HTTP or HTTPS).

```
wget --content-disposition -i
"https://192.168.2.132/DataLogs?Action=LIST"
```

Downloading of the DataLogs via JavaScript

The following example shows how you can download DataLogs automatically by using JavaScript. Replace the URL of the example by the correct IP address of the interface of your CPU at this URL and use the appropriate transfer protocol (HTTP or HTTPS).

```
<html>
<head>
<title>DataLog JavaScript Test</title>
<script type="text/javascript" src="jquery-1.12.4.min.js"></script>
</head>
<body>
<h1>DataLog JavaScript Test</h1>
<div><button id="load">Load DataLogs</button></div>
<div><ul id="list"></ul></div>
<script type="text/javascript">
$('#load').click(function(){
$.get('https://192.168.2.132/DataLogs', {'Action': 'LIST'},
function(data){
$('#list').empty();
$.each(data.split(/\r\n/), function(){
if (this.length == 0) continue;
$('#list').append('<li><a href="https://192.168.2.132' +
this + '">' + this + '</a></li>');
});
});
});
</script>
</body>
</html>
```

4.15 User files

Introduction

You read and write with the instructions "FileReadC" (Compact Read Data of a File) or "FileWriteC" (Compact Write Data to a File) in STEP 7 ASCII files (files in binary format).

Requirements

You need to save the UserFiles in the "UserFiles" directory on the SIMATIC memory card. You specify the storage location in the path of the "FileReadC" or "FileWriteC" instruction.
Path and file name for UserFiles have to fulfill the following rules:

- The file name must not be longer than 55 characters.
- The following characters are permitted for the directory and file name: 0 to 9, a to z in upper- and lower-case, "-" and "_"
- The path name must not start with "/", "\" or "."
- The path name must not contain ".."

Examples:

- UserFiles\Lift16_DataBase.txt
- UserFiles\2017-04-13_ErrorLog.bin

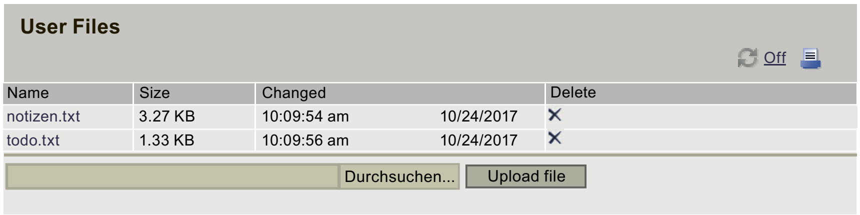
"User files" Web page

The browser displays the content of the SIMATIC memory card, directory UserFiles\ on the "User files" Web page.
You can sort the UserFiles according to the individual parameters in ascending or descending order.
To do this, click on one of the parameters in the column header:

- Name
- Size
- Changed

You can download, delete, and upload files.
You download the UserFile by clicking the file name.

By clicking on the ✕ icon, you can delete the UserFile. The file must be closed.



User Files

Name	Size	Changed		Delete
notizen.txt	3.27 KB	10:09:54 am	10/24/2017	✕
todo.txt	1.33 KB	10:09:56 am	10/24/2017	✕

Figure 4-66 User files view

> **NOTICE**
>
> **Size UserFiles**
>
> When you open a large UserFile through this Web page, the processing times of the instructions that process this file can increase notably.

4.15.1 Automatically read or upload user files

In addition to the Web server's user interface, you can automatically list, delete, download and upload UserFiles. Use JavaScript or Bash for this, for example.

Opening UserFiles from the SIMATIC memory card

The CPU makes a URL in the following format available so that you can list UserFiles automatically from the SIMATIC memory card of your CPU:

```
https://[ip]/UserFiles?Action=List
```

Enter the correct IP address of the interface of your CPU at this URL and use the appropriate transfer protocol (HTTP or HTTPS), e.g. `https://192.168.2.132/UserFiles?Action=LIST`. Subsequently call up the URL in your browser or command line interpreter.

The URL returns a list of the UserFiles on the SIMATIC memory card. Each entry returns the URL by which you download or delete the corresponding UserFile from the CPU. The actions to be performed are separated by a vertical line "|".

For simple syntax analysis of the list by web-based programming languages (such as JavaScript), the individual URLs are separated by a line break `<CR><LF>`. The following example shows the syntax of two URLs that access the UserFiles `File1.csv` and `File2.csv`:

```
File1.csv|/UserFiles?Name=File1.csv&Action=DOWNLOAD&E=1|/UserFiles?Name=File1.csv&Action=DELETE&E=1<CR><LF>
File2.csv|/UserFiles?Name=File2.csv&Action=DOWNLOAD&E=1|/UserFiles?Name=File2.csv&Action=DELETE&E=1<CR><LF>
<CR><LF>
```

When the URLs are called successfully, the CPU returns the status code `200 OK`. The CPU also returns this status code if there are no UserFiles on the SIMATIC memory card.

NOTE

Access authorization to the CPU for the reading out of data

In order to download UserFiles from the CPU, the user must have reading rights for the CPU. If the user does not have the required rights, the CPU returns the status code `403 FORBIDDEN` in the HTTP Response.

Downloading UserFiles via Bash scripts

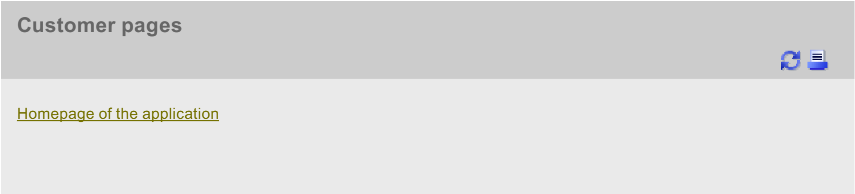
The following example shows how you can download UserFiles automatically from the CPU using a Bash script. Replace the URL of the example by the correct IP address of the interface of your CPU at this URL and use the appropriate transfer protocol (HTTP or HTTPS).

```
wget --content-disposition -i
"https://192.168.2.132/UserFiles?Action=LIST"
```

4.16 User pages

User-defined pages

In the "User-defined pages" area of the Web server you can upload HTML pages you have created yourself for reading out data of the target system.



Customer pages

Homepage of the application

Figure 4-67 User-defined pages

You create the pages with an HTML editor of your choice from which you generate data blocks (Web control DB and Fragment DBs) in STEP 7 and download them to the CPU. The "WWW" instruction synchronizes the user program with the Web server on the CPU and initializes the Web server. With the first call of the "WWW" instruction, the link to the user page is displayed on the web page of the CPU. A click on the link opens the user page in a new window.

NOTE

Write access to user-defined pages allows the process parameters and, thus, the operation of the CPU to be influenced.
To prevent external manipulation, always assign a password for users with write access to user pages in the user management. You can find information on user management in section Configuring the Web server (Page 26) under "Amending user management".

Creating user-defined pages

To create your own user page(s), you can use TIA Portal WinCC Unified as of version V17 or any HTML editor.

Creating user-defined pages with WinCC Unified in the TIA Portal (as of version V17)

With "View of Things" (VoT), you use WinCC Unified Engineering in the TIA Portal to create web pages for a SIMATIC S7-1500 CPU, ET 200SP CPU or CPU 1513/1516pro and to map the process.

You connect to the Web server of the CPU via a web browser. You can operate the objects that you have created with WinCC Unified on the "ViewOfThings" web page.

Compared to a random HTML editor, creating and loading with WinCC Unified offers you the following advantages:

- No HTML code expertise required; the web pages can be created as screen in the WinCC Unified editor and then downloaded to the CPU.
- You can make changes to the user-defined pages while the CPU is in the RUN mode.

More information on creating, loading and operating user-defined pages with VoT as well as special features in the hardware configuration, is available in the WinCC Unified online help under the keyword "View of Things".

Requirements for creating user-defined pages with an HTML editor

- You have assigned symbolic names to the tags you want to use on your web page in STEP 7 .
- In the Inspector window under "Properties > General > Web server", you have at least:
  - Activated the Web server
  - Assigned read-only or read and write permissions to the users for user-defined pages (see section Configuring the Web server (Page 26))
- You have completed all necessary communication settings (IP address parameter, subnet mask, etc.).
- You have downloaded the configuration.
- You have created your user page in an HTML editor of your choice.

Creating user-defined pages with an HTML editor

To create your user page(s) with any HTML editor, make sure that your HTML code complies with the standards of the W3C (World Wide Web Consortium), because STEP 7 does not check the HTML syntax in any way. In addition to the simple HTML code, you can also use JavaScript commands in your user-defined pages.

Proceed as follows:

1. Create the HTML file for your user page with an HTML editor.
   To allow data from the CPU to be output on your web page, integrate the AWP commands as HTML comments (see section AWP commands (Page 127)).
2. Store the HTML file and all associated source files (e.g., *.gif, *.jpg, *.js, etc.) in a directory on your PG/PC and note the storage path.

3. Call the "WWW" instruction in STEP 7 and program it (see section Programming the WWW instruction (Page 140)).
4. Configure the user page in STEP 7 (see section Configuring user pages (Page 139)). In this way, you compile the contents of your HTML files, among other things, into data blocks.
5. Download the configuration and the user program to the CPU.
6. Open your user page with your display device by means of a web browser in the Web server of the CPU.

NOTE

Extensive HTML pages, especially those with a lot of images, take up a lot of space in the load memory. Make sure you select a SIMATIC memory card with enough storage space to provide sufficient load memory.

If the **sum of the HTML pages** > 1 MB, performance losses may occur as only 1 MB data is saved in the cache.

We recommend that you create **each individual file of an HTML page** with a size not exceeding 512 KB; otherwise, problems can occur when sending the file from the Web server to the web browser. You can view the size of the respective file in the file explorer of the directory.

Updating user-defined pages

User-defined pages are not updated automatically in the web browser. You can program the HTML code so that the pages are updated automatically.
Pages which read out data from the controller are always up-to-date due to regular updates.

NOTE

If the HTML page contains form fields for data input, automatic update can impair the correct data input by the user.

To update the entire page automatically, you can add the following instruction to the <head> area of your HTML page, whereby the number "10" stands for the update interval in seconds:

```
<meta http-equiv="refresh" content="10">
```

References

The description of a user page is available in the section Example of a user page (Page 144).

Additional help for visualization with user-defined pages is available in the application examples on the Internet:

- Creating and using your own web pages for S7-1200 (https://support.automation.siemens.com/WW/view/en/58862931)
- Creating and using your own web pages for S7-1200 / S7-1500 (https://support.industry.siemens.com/cs/de/en/view/68011496)
- Visualizing with user-defined web pages on SIMATIC CPUs with PROFINET interface (https://support.automation.siemens.com/WW/view/en/44212999)
- How do you integrate the string contents in your user-defined web page of the S7-1500 CPU as of Firmware V1.6? (https://support.industry.siemens.com/cs/ww/en/view/98754370)

You will find more information on JavaScript commands in the ECMAScript specification on the Internet (https://www.ecma-international.org/ecma-262/5.1/).

For more information about how to automatically update web pages and how to incorporate user-defined pages with relative path names, refer to the FAQ with entry ID 62543256 on the Service&Support (https://support.industry.siemens.com/cs/ww/en/view/62543256) Internet page.

4.16.1 AWP commands

Overview

Automation Web Programming (AWP) commands are a special command syntax for exchanging data between the CPU and the customer page (HTML file).

AWP commands are entered as HTML comments and offer you the following options for your customer pages:

- Reading PLC tags
- Writing PLC tags
- Reading special tags
- Writing special tags
- Defining enum types
- Assigning enum types to tags
- Defining data block fragments
- Importing data block fragments
- Accessing the values of an array
- Accessing the values of a PLC tag of the data type STRUCT

General syntax

All AWP commands, except for the command for reading a PLC tag, have this structure:

```
<!-- AWP_< command name and parameter> -->
```

Files including AWP commands:

- must be UTF-8 encoded.
  To define UTF-8 as the character set of the page, include the following line in your HTML code:
  ```
  <meta http-equiv="content-type" content="text/html; charset
  utf-8">
  ```

NOTE

Saving the HTML page

Make sure that you save the file in the editor in UTF 8 character encoding as well.

- may not include the following sequence:]]>
- may not include the following sequence outside "read tag areas" (:="<Varname>":):
- depending on the use, must identify special characters in tag names or data block names with character escape sequences or quotation marks
- are case-sensitive

- should be additionally enclosed by JavaScript comments ("/*...*/") in JavaScript files
- must not exceed a file size of 64 KB
  To reduce the size of a file, you can subdivide the file into individual fragments (dynamic files). The maximum size of each individual fragment within the file (AWP command) is 64 KB.

Overview of AWP commands

Table 4-10 AWP commands

Function	Representation
Reading PLC tags	:=<Varname>:
Writing PLC tags	<!-- AWP_In_Variable Name='<Varname1>' -->
Reading special tags	<!-- AWP_Out_Variable Name='<Typ>:<Name>' -->
Writing special tags	<!-- AWP_In_Variable Name='<Typ>:<Name>' -->
Defining enum types	<!-- AWP_Enum_Def Name='<Name Enum-Typ>' Values='0: "<Text_1>",1:"<Text_2>",...,x:"<Text_y>"' -->
Assigning enum types to tags	<!-- AWP_Enum_Ref Name='<Varname>' Enum='<Name Enum-Typ>' -->
Defining data block fragments	<!-- AWP_Start_Fragment Name='<Name>'[Type=<Typ>] [ID=<Id>] -->
Importing data block fragments	<!-- AWP_Import_Fragment Name='<Name>' -->
Accessing the values of an array	<!-- AWP_Start_Array Name='"<DB name>".<array name>' --> ... <!-- AWP_End_Array -->
Accessing the values of a PLC tag of the data type STRUCT	<!-- AWP_Start_Struct Name='"<DB name>".<struct name>' --> ... <!-- AWP_End_Struct -->

4.16.1.1 PLC tags

Introduction to PLC tags

User pages can read PLC tags from the CPU and write data to the CPU.
To do so, PLC tags must:

- be enclosed by double quotation marks ("...").
- also be enclosed by single quotation marks ('" ... "') or with quotation marks masked with a backslash ("\" ... "\").
- be specified by a PLC tag name.
- if the PLC tag name includes the characters \ (backslash) or ', identify these characters with the escape sequence \\ or \' as normal characters of the PLC tag name.
- be enclosed with single quotation marks ('...'), if an absolute address (input, output, bit memory) is used in the AWP command.

Reading PLC tags

These out-tags (output direction as seen from the controller) are inserted at any place in the HTML text with the syntax described below.

Syntax

```
:=<Varname>:
```

<Varname> corresponds to the tag to be read from your STEP 7 project and can be a simple shared tag or a complete tag path to a structural element. Make sure that you use the name of the data block and not its number when you use data blocks.

Examples

```
:="Conveying speed":
:="My_datablock".bitmemory1:
:=%MW100:
```

Reading tags of the String and Character type

Below, these types of quotation marks are used in the explanation: single quotation marks ('), double quotation marks (").

As of firmware V1.6, with the "Read PLC tags" function, the CPU outputs tags of the String or Character type enclosed in single quotation marks to the browser.

For example:

- "Varname".MyString = ABC string tag
- You read the tag in HTML using the function :="Varname".MyString:
- The Web server outputs the character string 'ABC' to the browser

Using String or Character tags in expressions

On your HTML page, you use an expression in which the character string for reading a tag is enclosed in quotation marks, for example in forms.

Possible HTML code used:

```
<input type="text" name="appfield" value="myvalue">
```

If you read the displayed value for the "value" attribute from a PLC tag in this expression, the HTML code appears as follows:

```
<input type="text" name="appfield" value=":="Varname".MyString:">
```

By reading the PLC tag, the Web server outputs the value 'ABC'. In HTML, the code is then represented as follows:

```
<input type="text" name="appfield" value=" 'ABC' ">
```

If you have used single quotes instead of double quotes in your HTML code to enclose the attributes, the web server passes on the content of the tag enclosed in two single quotes to the browser. As a result of this, the browser does not output the content of the String or Character tag, since two consecutive single quotation marks each form a closed sequence. The values to be read are located between these sequences and are not output by the browser.

In this context, note in particular that the character string with double quotation marks is not identical to two single quotation marks even if they appear to be identical.

NOTE

The code is not adapted automatically during an update to firmware as of V1.6.
Adapt your HTML code if you have used single quotation marks to enclose attributes for the "Read PLC tags" function.

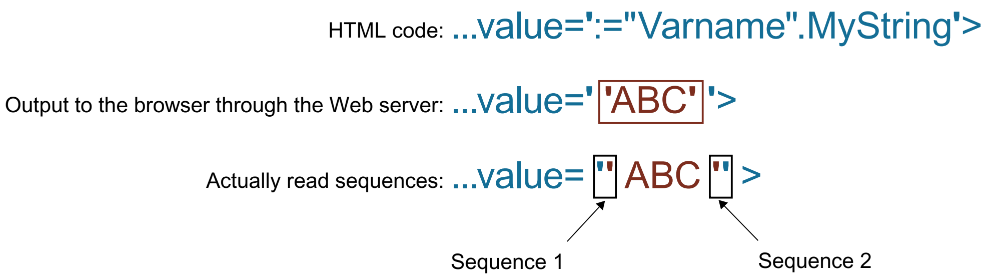

HTML code: ...value=':="Varname".MyString'>
Output to the browser through the Web server: ...value=''ABC''>
Actually read sequences: ...value='' ABC ''>
Sequence 1
Sequence 2

Figure 4-68 Example of HTML code with attribute in single quotation marks

Writing PLC tags

These in-tags (input direction as seen from the controller) are set on the browser page. This can take place in a form on your HTML page, for example, with text input or list selection boxes that correspond to the tags that can be written.
The tags are either set in the HTTP header (by cookie or POST method) or in the URL (GET method) by the browser in the HTTP request and are then written by the web server into the respective PLC tag.

NOTE

Write access during operation

In order for data to be written to the CPU from a user page, a user with corresponding write rights must be set up and the user must be logged in as this user. This applies to all write accesses of web pages to the CPU.

Syntax

```
<!-- AWP_In_Variable Name='"<Varname1>"' Name='"<Varname2>"'
Name='"<Varname3>"' -->
```

If the name of the tag that you are using for the web application is not identical with the name of the PLC tag, you can assign it to a PLC tag with the "Use" parameter.

```
<!-- AWP_In_Variable Name='<Varname_Webapp>' Use='<PLC_Varname>' -->
```

Examples with HTML input fields

```
<!-- AWP_In_Variable Name='"Target_Level"' -->
<form method="post">
<p>Input Target Level: <input name='"Target_Level"'
type="text"><input type="submit" value="Write to PLC"> </p>
</form>

<!-- AWP_In_Variable Name='"Data_block_1".Braking' -->
<form method="post">
<p>Braking: <input name='"Data_block_1".Braking' type="text"> <input
type="submit" value="Write to PLC"></p>
</form>
```

Example with HTML selection list

```
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable' -->
<form method="post">
<select name='"Data_block_1".ManualOverrideEnable'>
<option value=1>Yes</option>
<option value=0>No</option>
</select><input type="submit" value="submit setting"> </form>
```

4.16.1.2 Special tags

Special tags

The special tags are mainly the so-called HTTP tags that are defined in the definitions of the World Wide Web Consortium (W3C). Special tags are also used for cookies and server tags.

Reading special tags

The Web server can read PLC tags and pass them to special tags in the HTTP response header. You can, for example, read out a path name from a PLC tag to redirect the URL to another storage location with the special tag "HEADER:Storage location".

Syntax

```
<!-- AWP_Out_Varible Name='<Type>:<Name>' Use='<Varname>' -->
```

<Type> corresponds to the type of the special tag.
Options are:

- HEADER
- COOKIE_VALUE
- COOKIE_EXPIRES

<Name> corresponds to the name of the HEADER tag or the cookie:

- HEADER tags:
  - Status: HTTP status code (if no other value is set, status code 302 is returned).
  - Location: Path for redirection to another page. Status code 302 must be set.
  - Retry-After: Time for which the service is most likely not available. Status code 503 must be set.
- COOKIE_VALUE:name: Value of the named cookies.
- COOKIE_EXPIRES:name: Delay time of the named cookie in seconds.

Examples

The HTTP HEADER tag is written to the PLC tag of the same name:

```
<!-- AWP_Out_Variable Name='"HEADER:Status"' -->
```

If the name of the special tag is not identical with the name of the PLC tag, you can assign it to a PLC tag with the "Use" parameter:

```
<!-- AWP_Out_Variable Name='"HEADER:Status"' Use='"Status"' -->
```

Writing special tags

The Web server allows you to write values of special tags written in the HTTP header to the CPU. In STEP 7, for example, you can store information on the cookie of a user-defined page, or on the user accessing a page.

Syntax

```
<!-- AWP_In_Variable Name='<Type>:<Name>' Use='Varname' -->
```

<Type> corresponds to the type of the special tag.
Options are:

- HEADER
- SERVER
- COOKIE_VALUE

<Name> corresponds to the name of the HEADER tag or the cookie:

- HEADER tags:
  - Accept-Language: Accepted or preferred language
  - User-Agent: Browser information
  - Authorization: Credentials for a requested resource
- SERVER tags:
  - current_user_id: Indicates whether a user is logged in:
    current_user_id=0: No user is logged on / user "Everybody" or "Anonymous" has access.
    current_user_id=1: At least one user is logged on.
  - current_user_name: User name of the logged-on user
- COOKIE_VALUE:name: Value of the named cookies.

Examples

The HTTP SERVER tag is written to the PLC tag of the same name:

```
<!-- AWP_In_Variable Name='"SERVER:current_user_id"' -->
```

The HTTP SERVER tag is written to the PLC tag "My_User ID":

```
<!-- AWP_In_Variable Name='"SERVER:current_user_id"'
Use='"My_User-ID"' -->
```

4.16.1.3 Enum types

Enumeration types (enum types)

Enum types convert numerical values from the PLC program into texts or vice versa. The numerical values may also be assigned for use with several languages.

Define enum types

You can define enum types in your user pages and assign the values in an AWP command.

Syntax

```
<!-- AWP_Enum_Def Name='<Name Enum-Typ>' Values='0:"<Text_1>",
1:"<Text_2>",...,x:"<Text_y>"' -->
```

Examples

To store German values as HTML file in the "de" folder of the HTML directory:

```
<!-- AWP_Enum_Def Name="Enum1" Values='0:"an", 1:"aus", 2:"Störung"'
-->
```

To store English values as HTML file in the "en" folder of the HTML directory:

```
<!-- AWP_Enum_Def Name="Enum1" Values='0:"on", 1:"off", 2:"error"'
-->
```

Assigning enum types to tags

The assignment of tags from the user program to the individual enum types takes place by means of a separate AWP command. The used tag can be used at a different location of the user pages in a read operation or in a write operation.

During a read operation, the web server replaces the value read from the CPU with the correspondingly defined enum text value. During a write operation, the web server replaces the defined enum text value with the corresponding integer value of the enumeration before the value is written to the CPU.

Syntax

```
<!-- AWP_Enum_Ref Name='<Varname>' Enum="<Enum-Type>" -->
```

<Varname> is the symbolic tag name from the user program, <Enum-Type> of the previously defined name of the enum type.

Example for a declaration

```
<!-- AWP_Enum_Ref Name='"Alarm"' Enum="AlarmEnum" -->
```

Example of how to use when reading a tag

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No Alarm", 1:"Vessel
is full", 2:"Vessel is empty"' -->
<!-- AWP_Enum_Ref Name='"Alarm"' Enum="AlarmEnum" -->
...
<p> The current value of "Alarm" is :="Alarm": </p>
```

If the value of "Alarm" in the CPU is 2, the HTML page shows 'The current value of "Alarm" is container is empty' because the definition of the enum type assigns the numerical value 2 to the character sequence "Container is empty".

Example of how to use when writing a tag

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No Alarm", 1:"Vessel
is full", 2:"Vessel is empty"' -->
<!-- AWP_In_Variable Name='"Alarm"' -->
<!-- AWP_Enum_Ref Name='"Alarm"' Enum="AlarmEnum" -->
...
<form method="post">
<p><input type="hidden" name ='"Alarm"' value='Vessel is full'
/></p>
<p><input type="submit" value='Set vessel is full' /></p>
</form>
```

The value 1 is written to the PLC tag "Alarm" because the definition of the enum type assigns the numerical value 1 the text "Container is full".

Note that the name specified in "AWP_In_Variable" must be exactly the same as the name specified in "AWP_Enum_Ref".

4.16.1.4 Fragments

Fragments

Fragments are "logical sections" of a web page to be processed by the CPU individually.

Fragments are usually entire pages, but they can be individual elements, such as files (e.g. images) or documents.

NOTE

In each fragment in which enum texts are referenced by a PLC tag, this PLC tag must be assigned to the enum type name with the appropriate AWP command.

Defining fragments

A fragment extends to the beginning of the next fragment or to the end of the file.

Syntax

```
<!-- AWP_Start_Fragment Name='<Name>' [Type="<Typ>"] [ID="<Id>"]
[Mode="<Mode>"] -->
```

This command specifies the start of a fragment.

- <Name> Specifies the name of the fragment. The name must start with a letter [a-zA-Z] or an underscore (_). This first character can be followed by letters, underscores or numbers [0-9].
- <Type> Specifies the type of the fragment.
  - "automatic": The page is edited automatically (default)
  - "manual": Do not use this fragment type. Use the preset fragment type "automatic".
- <Id> You can specify a numerical ID for the fragment. If no ID is assigned, the fragment is automatically assigned an ID. For manual pages (<Type>=manual), the fragment can be addressed with this ID in the user program of the CPU.

  NOTE

  ID assignment

  Set the ID as low as possible because the highest ID influences the size of the Web control DB.

- <Mode>
  - "visible": The contents of the fragment are displayed on the user page (default).
  - "hidden": The contents of the fragment are **not** displayed on the user page.

Importing fragments

You can specify a fragment in an HTML page and import this fragment into other websites.

NOTE

Ensure that no AWP command for importing fragments is positioned between an enum assignment and enum usage, because this import can result in the enum assignment being located in a different fragment than the enum usage.

Example

A company logo is to be displayed on all websites of a web application.
The HTML code for the fragment that displays the company logo exists only once. You can import the fragment as many times and into as many HTML files as required.

Syntax

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

<Name> corresponds to the name of the fragment to be imported.

Example

HTML code within a web page that creates a fragment for displaying an image:

```
<!-- AWP_Start_Fragment Name='Mein_Firmenlogo' -->
<p><img src="CompanyLogo.jpg"></p>
```

HTML code that imports the created fragment into another web page:

```
<!-- AWP_Import_Fragment Name='My_Company_Logo' -->
```

4.16.1.5 Arrays

Arrays

The Web server provides the user program commands AWP_Start_Array and AWP_End_Array for accessing all values of an array.
Only one-dimensional arrays are supported.
Multidimensional arrays of the form array[x][y] are not supported.

Syntax

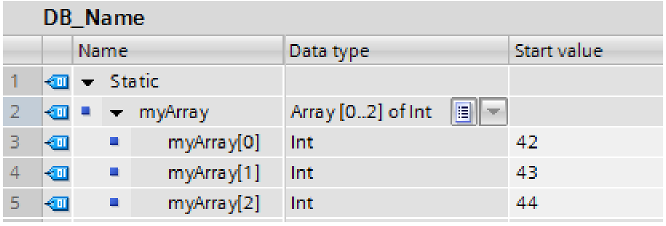
```
<!-- AWP_Start_Array Name='"<DB name>".<array name>' -->
... Content of the array, utilized keywords: ArrayIndex and value..
<!-- AWP_End_Array -->
```

Parameter

- <Name> defines the name of the array with the elements you want to access.
  - You require the DB name and the name of the array corresponding to the data block structure defined in STEP 7.
  - The name must be within single or double quotation marks.
  - The DB name is within double quotation marks.
- <ArrayIndex> Index of an array element
- <value> Value of an array element

Example

The example reads all elements of the "MyArray" structure in the "DB_Name" data block of the CPU and displays the index and the values of the tags on the user-defined web page.



DB_Name

	Name	Data type	Start value
1	Static		
2	myArray	Array [0..2] of Int	
3	myArray[0]	Int	42
4	myArray[1]	Int	43
5	myArray[2]	Int	44

```
<!-- AWP_Start_Array Name='"DB_Name".MyArray' -->
```

```
Index: :=ArrayIndex: Value: :=value:
<!-- AWP_End_Array -->
```

The code indicated above generates the following display:

Index: 1 Value: 42
Index: 2 Value: 43
Index: 3 Value: 44

Representation of arrays of the BOOL data type

The output of arrays of the BOOL type is always filled to the next full 8 bits. This particular feature only occurs with BOOL arrays.

Example:

```
"DB_1".bitArray is a BOOL array with 5 elements.
<!-- AWP_Start_Array Name='"DB_1".bitArray' -->
:=ArrayIndex: -> :=value:
<!-- AWP_End_Array -->
```

Edition:

```
0 -> Value from "DB_1".bitArray[0]
1 -> Value from "DB_1".bitArray[1]
2 -> Value from "DB_1".bitArray[2]
3 -> Value from "DB_1".bitArray[3]
4 -> Value from "DB_1".bitArray[4]
5 -> 0
6 -> 0
7 -> 0
```

4.16.1.6 Structures

Structures

The web server provides AWP commands for accessing structures in order to access the values of a PLC tag of the data type STRUCT.

Syntax

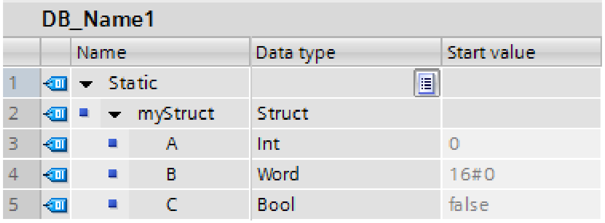
```
<!-- AWP_Start_Struct Name='"<DB name>".<struct name>' -->
... Content of structure ...
<!-- AWP_End_Struct -->
```

Parameter

- <Name> defines the name of the structure with the elements you want to access.
  - You require the DB name and the name of the structure corresponding to the data block structure defined in STEP 7.
  - The name must be within single or double quotation marks.
  - The DB name is within double quotation marks.

Example

The example reads elements of the "MyStruct" structure in the "DB_Name" data blocks of the CPU and displays the value of the tag on the user-defined web page.



DB_Name1

	Name	Data type	Start value
1	▼ Static		
2	▼ myStruct	Struct	
3	A	Int	0
4	B	Word	16#0
5	C	Bool	false

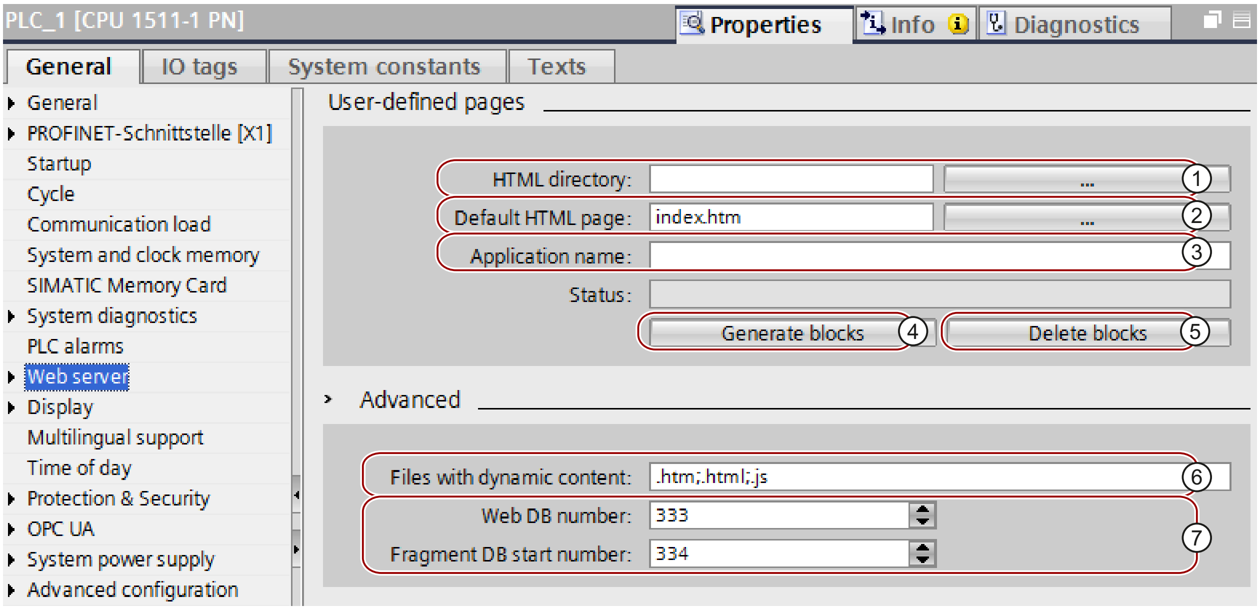
```
<!-- AWP_Start_Struct Name='"DB_Name".MyStruct' -->
:=A:
:=B:
:=C:
<!-- AWP_End_Struct -->
```

The code indicated above corresponds to the following commands:

```
:="DB_Name".MyStruct.A:
:="DB_Name".MyStruct.B:
:="DB_Name".MyStruct.C:
```

4.16.2 Configuring user pages

Configuring customer pages



User-defined pages

HTML directory:

Default HTML page: index.htm

Application name:

Status:

Advanced

Files with dynamic content: .htm;.html;.js

Web DB number: 333

Fragment DB start number: 334

Figure 4-69 Configuring customer pages in STEP 7

To configure the customer pages in STEP 7, follow these steps:

1. Select the CPU in the device configuration.
2. Open the settings in the Inspector window of the CPU under "Properties > General > Web server".
3. Select the folder on your display device in which you have saved your HTML page in the "Customer pages" area under ① "HTML directory".
4. Enter the name of the HTML page under ② "Default HTML page" that is to open when you start the application.
5. You can also specify a name for your application under ③ "Application name". This name is used to further divide or group the webpages. If an application name already exists, the URL is displayed in the following format: https://a.b.c.d/awp/<Application name>/<Page name>.html
6. In the "Extended" area, enter the file extensions that have to be checked for AWP commands in input box ⑥ "Files with dynamic contents". By default, STEP 7 analyses files with the extensions "htm" and "html". Here you can enter other file extensions that you have used when creating your customer page.
7. You can accept the number for the Web DB ⑦ and the fragment DB start number ⑦ or you can assign a new number of your choice that is not assigned.

8. Click the button ④ "Create blocks" to create data blocks from the source files. The generated data blocks are stored in the STEP 7 project tree in the folder "System blocks > Web server". These data blocks consist of a control data block (Web control DB) that controls the display of the webpages and one or several data block fragments (fragment DBs) with the compiled webpages.
9. In the network view, select the CPU to be loaded and then select the "Download to device" command in the "Online" menu to download the blocks. The compilation of the blocks is implicitly triggered before the download. If errors are signaled during this process, they must be remedied before you can download the configuration.

Deleting data blocks

Click the "Delete blocks" button ⑤ to delete previously generated data blocks. STEP 7 deletes the Web Control DB and all fragment DBs from the project in which your customer pages are located.

4.16.3 Programming the WWW instruction

The WWW instruction

The instruction WWW initializes the Web server of the CPU or synchronizes the user-defined pages with the user program in the CPU. The Web Control DB is the input parameter for the WWW instruction and specifies the content of the pages as they are displayed in the fragment DBs, as well as the status and control information. STEP 7 creates the Web-Control DB when you click on the "Generate blocks" button.

NOTE

DB number of the web control DB.

If you change the DB number of the DB 333, the user pages in the Web server can no longer be reached at renewed downloading into the CPU. Error code W#16#007F is output at the parameter RET_VAL. Therefore observe the default setting DB 333 for the Web Control DB.

If you want to change the DB number nevertheless, you have to switch the CPU POWER-OFF → POWER ON, so that the user pages in the Web server can be reached.

NOTE

Calling the WWW instruction

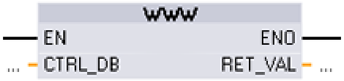
You have created your user page in an HTML editor of your choice.

- Use automatic HTML pages if you want to **deactivate** control of the page layout by means of the user program (requires at least one call of SFC 99) Operating mode changes from RUN to STOP do not affect the call of the user-defined pages.
- Use manual HTML pages if you want to **enable** control of the page layout by means of the user program (cyclic call of SFC 99 required).

Programming the WWW instruction

The user program must execute the WWW instruction in order that the user-defined pages can be called in the Web server.

Table 4-11 WWW instruction

LAD/FBD	SCL	Description
	ret_val :=WWW(ctrl_db:=uint_in_);	Access to user pages by means of the Web server

Parameters

The following table shows the parameters of the WWW instruction.

Table 4-12 Parameters

Parameters	Declaration	Data type	Description
CTRL_DB	Input	DB_WWW	Data block that describes the user pages (Web control DB)
RET_VAL	Output	INT	Error information

RET_VAL parameter

Table 4-13 RET_VAL

Error code (W#16#...)	Explanation
0000	No error has occurred. There are no pending website requests that must be released by the user program.

00xy	x: Indicates whether an error occurred during initialization of the Web Control DB (CTRL_DB):
	x=0: No errors occurred.
	x=1: Error(s) occurred. The error is encoded in the byte "CTRL_DB.last_error" of the Web Control DB.
	y: Number of the pending request. Several requests are possible (e.g., request "0" and "1" are pending: y="3").
	y="1": Request "0"
	y="2": Request "1"
	y="4": Request "2"
	y="8": Request "3"

803A	The specified Web Control DB does not exist on the CPU.
8081	Incorrect version or incorrect format of the Web Control DB.
80C1	There are no resources available to initialize the web application.

4.16.4 Defining the user page as start page

Defining the customer page as start page

In addition to the default introduction page, you can also define the start page of your customer pages as the start page of the Web server.

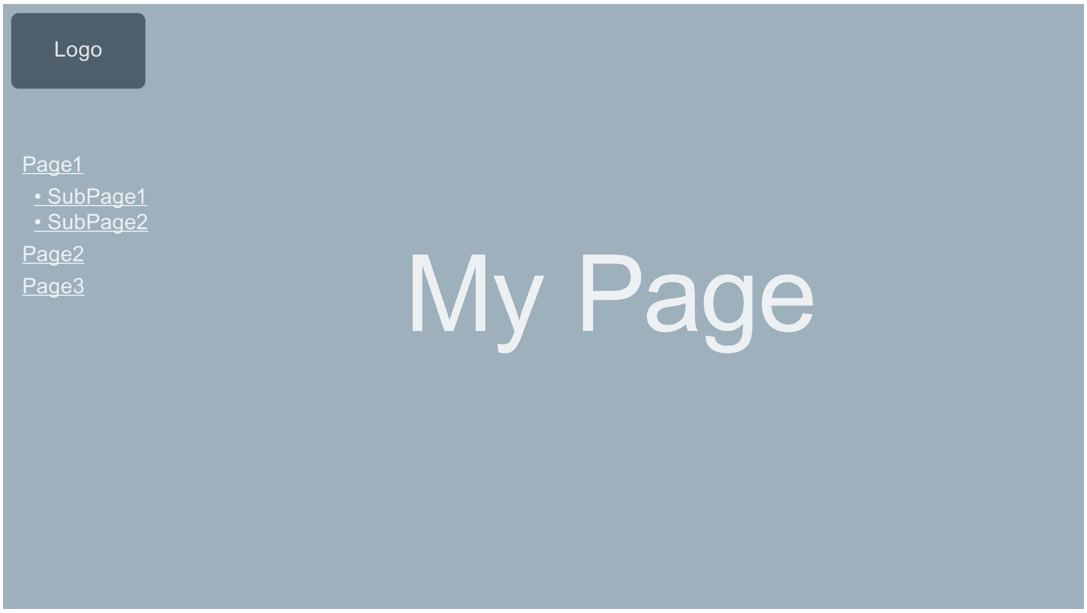


Figure 4-70 Example of customer page as start page of the Web server

Requirements

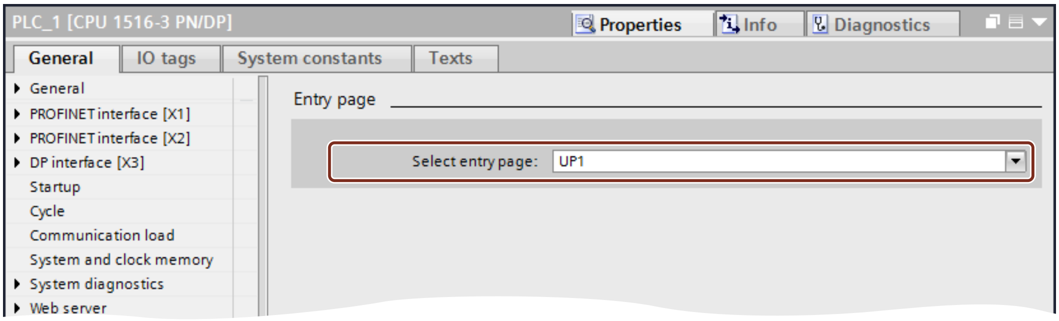
The following requirements must be met before the customer page is displayed as the start page of the Web server:

- You have configured the customer page as the start page.
- You have configured a user in STEP 7 whom you have assigned at least the authorization "... open customer web pages".
- The CPU is in RUN mode.

Procedure

To define the customer pages in STEP 7 as the start page of the Web server, proceed as follows:

1. Select the CPU in the device configuration.
2. Open the settings in the Inspector window of the CPU under "Properties > General > Web server".
3. Select the entry "AWP1" in the area "Entry page" under "Select entry page".



Entry page

Select entry page: UP1

Figure 4-71 Set customer page as start page in STEP 7

4. Download the configuration to the CPU.

If you now enter the IP address of the CPU in the browser, a connection is automatically established to your customer pages.

If you want to access the web pages of the CPU again, link the web pages from your customer pages, e.g. via the URL "https://a.b.c.d./Portal/Portal.mwsl?PriNav=Start" or "https://a.b.c.d/Portal/Portal.mwsl?PriNav=Start". The specification "a.b.c.d" is an example of the IP address of the configured CPU.

Example of link in HTML:

```
<a href="/Portal/Portal.mwsl?PriNav=Start">SIMATIC web pages</a>
```

Reading out service data

If you define your customer page as start page of the Web server, all direct access to the web pages for reading out the service data is also disabled.

If you want to continue to read out service data via the Web server when servicing is required, here is how you can link the service data page directly to your customer page. Just as for the web pages of the CPU, link the service data page, e.g. via the URL "https://a.b.c.d/save_service_data" or "https://a.b.c.d/save_service_data", the "a.b.c.d" here is an example of the IP address of the configured CPU.

Example of link in HTML:

```
<a href="/save_service_data">Service data</a>
```

Reference

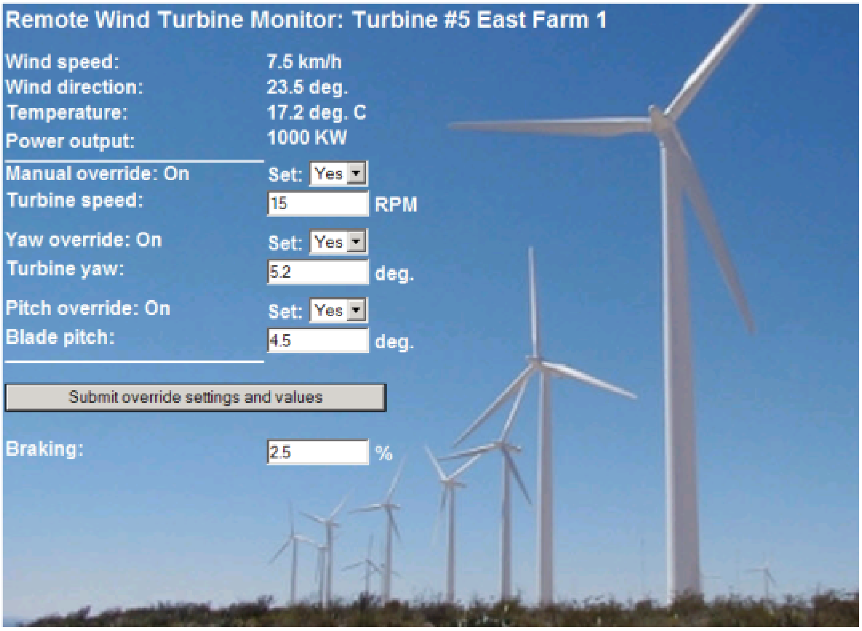
You can find additional information on the topic of customer page as start page in the FAQ with entry ID 67184104 on the Service&Support (https://support.industry.siemens.com/cs/ww/en/view/67184104) Internet page.

4.16.5 Example of a user page

4.16.5.1 Website for monitoring and controlling a wind turbine

Example of a user page

Here you see a user page for monitoring and controlling a wind turbine:



Remote Wind Turbine Monitor: Turbine #5 East Farm 1

Wind speed: 7.5 km/h
Wind direction: 23.5 deg.
Temperature: 17.2 deg. C
Power output: 1000 KW

Manual override: On Set: Yes
Turbine speed: 15 RPM

Yaw override: On Set: Yes
Turbine yaw: 5.2 deg.

Pitch override: On Set: Yes
Blade pitch: 4.5 deg.

Braking: 2.5 %

Figure 4-72 Overview of user page wind turbine

The user page was created in English in this example, but you can select any language you wish when you create your own user page.

In this application, each wind turbine of the wind farm in STEP 7 has a data block with specific data for the respective location and respective turbine.

The user page gives you the option to access the turbine remotely with a display device. Users can call the standard web pages of a CPU of a specific wind turbine and switch to the user page "Remote Wind Turbine Monitor", where they can view the turbine data. A user with the corresponding access permissions can also set the turbine into the manually controlled mode and thus control the tags for speed, orientation and angle of attack of the turbine by means of the website. The user can also specify a brake value regardless of manual or automatic control of the turbine.

STEP 7 checks the Boolean values for override of the automatic control and, if set, uses the values for speed, orientation and angle of attack of the turbine as defined by the user.

Files used

Three files are used in the application example:

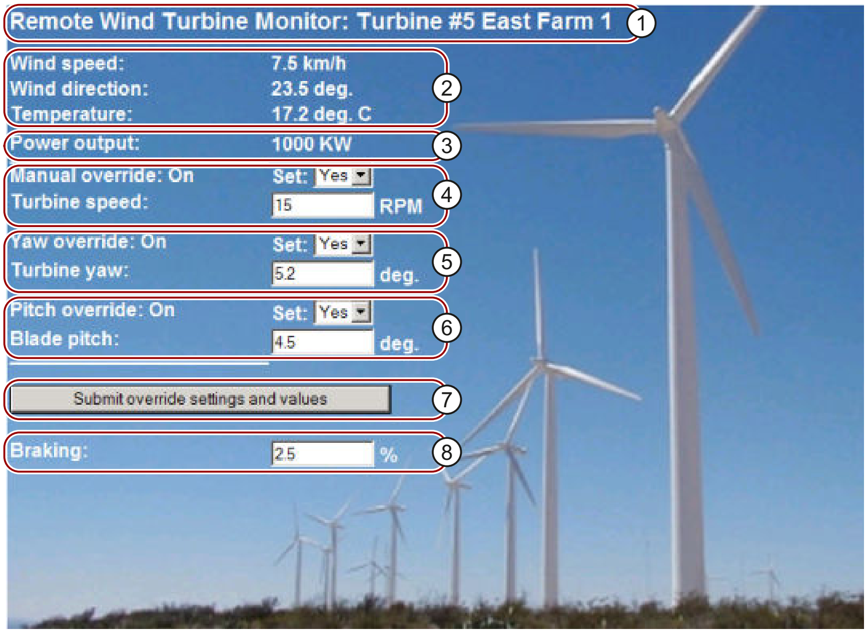
- Wind_turbine.html: The user page in the figure shown above. The control data is accessed by AWP commands.
- Wind_turbine.css: The Cascading Style Sheet which includes the formatting specifications of the user page. The use is optional but can simplify the design of the user page.
- Wind_turbine.jpg: The background image displayed on the user page. The use of images is optional, user pages with lots of images require a lot more memory in the load memory.

These files are not part of your installation but they are described as an example below.

Implementation

The user page uses AWP commands to read out values from the CPU as well as writing values to it. The user page also uses AWP commands for the definition of enum types, such as the assignment of tags to enum types for handling the ON/OFF settings.

The user page is structured as follows:



Remote Wind Turbine Monitor: Turbine #5 East Farm 1

Wind speed: 7.5 km/h
Wind direction: 23.5 deg.
Temperature: 17.2 deg. C

Power output: 1000 KW

Manual override: On Set: Yes
Turbine speed: 15 RPM

Yaw override: On Set: Yes
Turbine yaw: 5.2 deg.

Pitch override: On Set: Yes
Blade pitch: 4.5 deg.

Submit override settings and values

Braking: 2.5 %

① Header of the website with number and location of the wind turbine.

② Atmospheric conditions at the turbine, wind speed, wind direction and current temperature are displayed.

③ Read-out power output.

④ Manual override: Activates manual override of the turbine. To make manual settings for speed, orientation and angle of attack, the STEP 7 user program requires that manual override has been activated.

⑤ Override of the orientation: Activates manual override of the turbine orientation.

⑥ Override of the angle of attack: Activates manual override of the angle of attack of the rotor blades.

⑦ By clicking this button, you transfer the override settings to the CPU.

⑧ Manual setting of a percentage value for braking. The setting "Manual override" is not required to enter a brake value.

Figure 4-73 Overview of user page wind turbine

In addition, the user page uses an AWP command that writes the special tag to the tag table. The tag table contains the ID of the user who is currently accessing the page.

4.16.5.2 Reading and displaying data from the CPU

Example of HTML code for reading and displaying data from the CPU

This part of the HTML code is used to display the power output on the user page.

On the left-hand side the "Power Output": text is displayed, on the right-hand side, the value of the tags for the power output including the unit ("KW").

The AWP command `:="Data_block_1".PowerOutput` executes the read operation. The data block is referenced here by its symbolic name and not by its number ("Data_block_1" instead of "DB1").

The code used in the example is:

```
<tr style="height:2%;">
<td>
<p>Power output:</p>
</td>
<td>
<p style="margin-bottom:5px;"> :="Data_block_1".PowerOutput: KW</p>
</td>
</tr>
```

4.16.5.3 Using enum types

Definition of enum types

The described user page uses enum types in three locations. "On" or "Off" is displayed for a Boolean value at these locations.

The enum type for "On" results in a value of 1, the enum type for "Off" results in a value of 0. The following statements from the HTML code of the user page show the declaration of an enum type with the name "OverrideStatus" and the values "0" and "1" for "Off" or "On" as well as the definition of an enum type reference from "OverrideStatus" to the tag "ManualOverrideEnable" in the data block "Data_block_1".

NOTE

Assignment of enum types

If the user page writes into a tag by using an enum type, there has to be a declaration "AWP_In_Variable" for each "AWP_Enum_Ref" declaration.

The code used in the example is:

```
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable' -->
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
<!-- AWP_Enum_Ref Name='"Data_block_1".ManualOverrideEnable'
Enum="OverrideStatus" -->
```

The following code describes a display box for displaying the current status of "ManualOverrideEnable". A normal read command for tags is used but because of the declared and referenced enum type, the website displays the values "On" and "Off" instead of "1" and "0".

```
<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
```

```
<p>Manual override: :="Data_block_1".ManualOverrideEnable:</p>
</td>
```

The following code describes a drop-down list for changing "ManualOverrideEnable" by the user. The drop-down list consists of the "Yes" and "No" options that are assigned to the "On" or "Off" values by means of the enum type reference. If you make no selection, the status remains the same.

```
<select name='"Data_block_1".ManualOverrideEnable'>
<option value=':"Data_block_1".ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option selected value="Off">No</option>
</select>
```

The drop-down list is included in the form on the website. The form is uploaded, when the user clicks on the "Submit override settings and values" button. If the user has selected "Yes", the value "1" is written in the tag "ManualOverrideEnable" in the "Data_block_1" data block; if the user has selected "No", the value "0" is written.

4.16.5.4 Writing user inputs into the controller

Setting options

The user page "Remote Wind Turbine Monitor" includes different AWP commands for writing data into the controller. A user with the corresponding access permissions can control the wind turbine manually, activate the override for the turbine speed and the turbine orientation as well as the angle of attack of the rotor blades with the declaration of different "AWP_In_Variable" write commands. The user can also specify floating-point numbers for turbine speed, orientation angle of attack and percentage of braking. The user page uses an HTTP command in the format "POST" to write the tags into the controller.

The code used in the example for setting the brake value is:

```
<!-- AWP_In_Variable Name='"Data_block_1"' -->
...
<tr sytle="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Braking:</p></td>
<td>
<form method="POST">
<p><input name='"Data_block_1".Braking' size="10" type ="text">
%</p>
</form>
</td>
</tr>
```

This excerpt from the HTML code first defines a "AWP_In_Variable" for the "Data_block_1" data block which enables the user page to write any number of tags into the data block. The text "Braking:" is displayed on the left-hand side; on the right-hand side is a box in which the user can make entries for the "Braking" tag in the data block.

The user page reads out the actual braking value from the controller and displays it in the text box. A user with the corresponding access permissions can then write a brake value that controls the braking process into the data block of the CPU.

NOTE

Declaration of data blocks

If you declare an entire data block by means of a "AWP_In_Variable", each tag in the data block can be written by means of the user page. If only certain tags in the data block are to be writable, you declare this specifically using `<!-- AWP_In_Variable Name='"Data_block_1".Braking' -->`, for example.

4.16.5.5 Writing special tags

Using special tags

The user page "Remote Wind Turbine Monitor" writes the special tag "Server:current_user_id" into a tag of the CPU. The tag value contains the value "1" if a user is logged on and "0" if a user is not logged on. In this example, a user is logged on, so the tag value is set to "1". The special tag is written into the CPU by the user page and does not need a user interface.

The code used in the example is:

```
<!-- AWP_in_variable Name="SERVER:current_user_id" Use="User_ID" -->
```

4.16.5.6 HTML code of the user page "Remote Wind Turbine Monitor"

Example

You can find the complete HTML code of the sample customer page "Remote Wind Turbine Monitor" as well as the used Cascading Style Sheet (CSS) on the Internet (https://support.industry.siemens.com/cs/ww/en/view/107623221/71849346059).

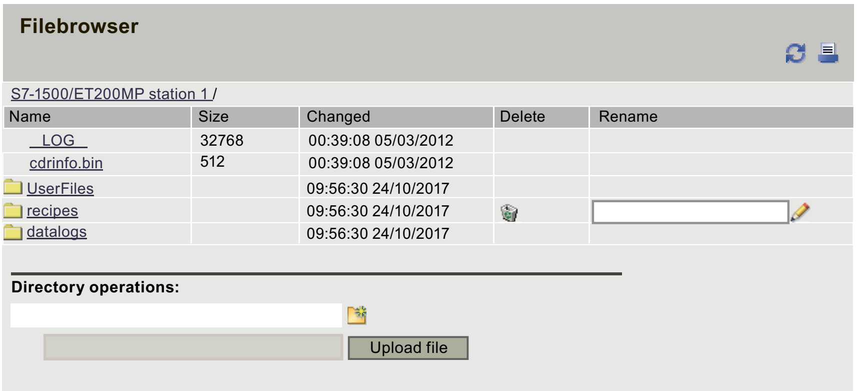
4.17 Filebrowser

Requirements

Access rights must be assigned for the user in the user management.

Filebrowser

The contents of the SIMATIC memory card are displayed by the browser on the "Filebrowser" web page. This means, for example, that you can read and edit the log files generated by the CPU without having to use STEP 7.



Filebrowser

S7-1500/ET200MP station 1 /

Name	Size	Changed	Delete	Rename
__LOG__	32768	00:39:08 05/03/2012		
cdrinfo.bin	512	00:39:08 05/03/2012		
UserFiles		09:56:30 24/10/2017		
recipes		09:56:30 24/10/2017	🗑	✎
datalogs		09:56:30 24/10/2017		

Directory operations:

Upload file

Figure 4-74 Filebrowser view

The file browser lists all existing files and directories located on the SIMATIC memory card. You can download, delete, rename and upload the files. You can create, delete and rename the directories.

NOTE

The Filebrowser only grants you read access to the “DataLogs”, "Backups" and “UserFiles” directories.

Exception system files

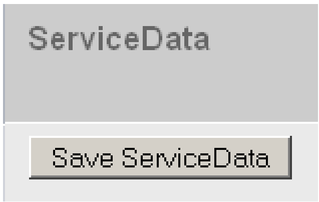
The system files include the job file and all special directories including their contents to which the job file refers. System files are not displayed, and cannot be changed or deleted.

4.18 Reading out service data

The Web server gives you the option to save service data. In addition to the content of the diagnostics buffer, they include more information on the internal status of the CPU. If you should encounter a problem with the CPU that cannot be resolved otherwise, you therefore have the option to submit the service data to the Service&Support team.

Procedure

1. Enter the following address in the address bar of your web browser:
   "https://<CPU IP address>/save_service_data", e.g.
   "https://192.168.3.141/save_service_data"
2. Your screen displays the service data page with a button for saving the service data.



3. Save the service data locally on your display device by clicking on "Save ServiceData".

Result

The data is stored in a .dmp file with the following naming convention: "<MLFB><Serial number><Time stamp>.dmp." The user can change the file name at a later time.

NOTE

If you have defined your customer page as the start page, observe the note on reading out service data in section Defining the user page as start page (Page 142).

4.19 Basic websites

Web pages with reduced contents

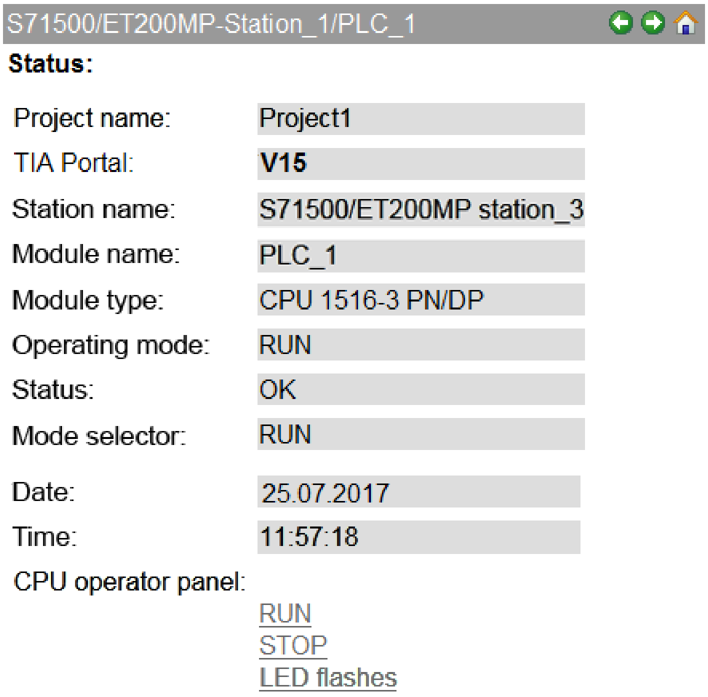
Basic websites are offered for display devices with smaller screens, for example HMI, on the Web server.
Basic websites are web pages with reduced content that are adapted to the requirements of small screens with low resolution.
These sites do not support JavaScript for the sake of fast access. This means that not all standard websites are available as basic websites. The basic website can also have fewer functions than the standard website.

The switch to basic websites takes place automatically for HMI devices.
You access basic websites from other end devices by entering the IP address of the configured CPU and the extension "/basic" in the address bar of the Web browser, for example, https://192.168.3.141/basic.

The following standard websites are also available as basic websites:

- Start page (in Basic: "Status")
- Diagnostics (without the "Program protection", "Runtime information" and "Fail-safe" tabs)
- Diagnostics buffer
- Memory usage
- Module information
- Alarms (without acknowledgment option)
- Communication
- Tag status
- Watch tables
- Customer pages
- Filebrowser (read access only)
- DataLogs
- Intro

The basic websites are displayed as follows:



S71500/ET200MP-Station_1/PLC_1

Status:

Project name:	Project1
TIA Portal:	**V15**
Station name:	S71500/ET200MP station_3
Module name:	PLC_1
Module type:	CPU 1516-3 PN/DP
Operating mode:	RUN
Status:	OK
Mode selector:	RUN
Date:	25.07.2017
Time:	11:57:18

CPU operator panel:

Figure 4-75 Example basic websites, "Status" web page

API (Application Programming Interface) 5

Web API

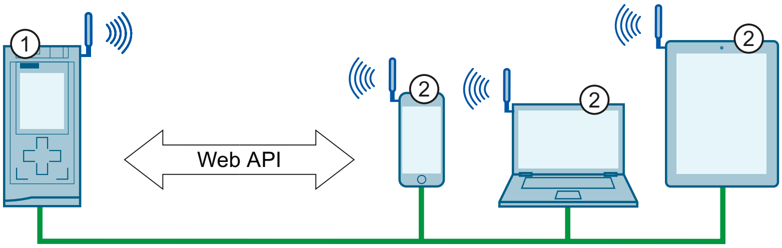
The CPU offers you a Web-based API (Web API) as an interface for reading and writing CPU data.

The Web API enables you to:

- Implement web applications at the latest state-of-the-art technology
- Communicate with the Web server of the CPU via script and programming languages
- Create web applications that connect to multiple CPUs at the same time, for example, to create dashboards that visualize the status of multiple CPUs

Connection established between CPU, Web API and terminal device.

The following graphic shows an example of the Web API between CPU and terminal device.



① CPU

② Terminal devices

Figure 5-1 Web API

Communication between the CPU and the terminal device takes place via PROFINET or WLAN integration.

NOTE

Security information

Please note that the following graphic only shows the role of the Web API between CPU and terminal device.

For the correct setup of a secure WLAN connection, observe the Security Information in the preface of this manual.

Requirements

The Web API can only be used for CPUs as of firmware version from V2.8 for the following systems:

- The standard F and T-CPUs of the automation system S7-1500
- The CPUs 1504D TF and 1507D TF of the SIMATIC Drive Controller
- The CPUs of the ET 200pro distributed I/O system
- The standard F and T-CPUs of the distributed I/O system ET 200SP
- The standard F and T-CPUs of the S7-1500 software controller
- As of firmware version V3.1, the R/H-CPUs of the automation system S7-1500

The following requirements must be met before you can use the Web API.

- You have assigned the correct firmware version (≥ V2.8) in the Hardware catalog of STEP 7.
- You have created and configured a project and downloaded it to the CPU.
- You have ensured that the following check box is selected in STEP 7:
  - Activate Web server on this module

Overview of the Web API methods depending on the firmware version of the CPU

The Web API is being continuously extended. The following table shows the mechanisms and methods you can use with a specific firmware version of the CPU:

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Basic functions			
Api.Login (Page 171)	V2.8	V2.8	V3.1
Api.GetPermissions (Page 175)	V2.8	V2.8	V3.1
Api.Browse (Page 176)	V2.8	V2.8	V3.1
Api.Version (Page 177)	V2.8	V2.8	V3.1
Api.Ping (Page 178)	V2.8	V2.8	-
Api.GetCertificateUrl (Page 178)	V2.8	V2.8	-
Api.Logout (Page 178)	V2.8	V2.8	V3.1
Api.GetQuantityStructures (Page 179)	V3.1	V3.1	V3.1
Api.ChangePassword (Page 179)	V3.1	V3.1	V3.1
Api.GetPasswordPolicy (Page 181)	V3.1	V3.1	V3.1
Api.GetAuthenticationMode (Page 182)	V3.1	V3.1	V3.1

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Setting the Web server default page			
WebServer.SetDefaultPage (Page 183)	V3.1	V3.1	-
WebServer.ReadDefaultPage (Page 184)	V3.1	V3.1	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Ticket mechanism (Page 164)			
Api.BrowseTickets (Page 168)	V2.9	V2.9	V3.1
Api.CloseTicket (Page 170)	V2.9	V2.9	V3.1

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Reading and writing process data			
PlcProgram.Read (Page 190)	V2.8	V2.8	-
PlcProgram.Write (Page 192)	V2.8	V2.8	-
PlcProgram.DownloadProfilingData (Page 194)	V3.1	V3.1	-
PlcProgram.Browse (Page 200)	V2.8	V2.8	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Reading and changing the operating mode			
Plc.ReadOperatingMode (Page 206)	V2.9	V2.9	-
Plc.RequestChangeOperatingMode (Page 206)	V2.9	V2.9	-
Plc.ReadModeSelectorState (Page 207)	V3.1	V3.1	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Performing time settings via Web API			
Plc.ReadSystemTime (Page 208)	V3.0	V3.0	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Plc.SetSystemTime (Page 209)	V3.1	V3.1	-
Plc.ReadTimeSettings (Page 210)	V3.0	V3.0	-
Plc.SetTimeSettings (Page 212)	V3.1	V3.1	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Reading diagnostics and service data			
Alarms.Browse (Page 216)	V3.1	V3.1	-
Alarms.Acknowledge (Page 221)	V3.1	V3.1	-
Syslog.Browse (Page 222)	V3.1	V3.1	V3.1
DiagnosticBuffer.Browse (Page 225)	V3.1	V3.1	-
Modules.DownloadServiceData (Page 228)	V3.1	V3.1	-
Project.ReadLanguages (Page 215)	V3.1	V3.1	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Backing up and restoring the configuration			
Plc.CreateBackup (Page 229)	V3.0	V3.0	-
Plc.RestoreBackup (Page 231)	V3.0	V3.0	-

Method	Standard/T-CPU as of firmware version	F-CPUs as of firmware version	R/H-CPUs as of firmware version
Accessing the contents of the SIMATIC Memory Card			
Files.Browse (Page 234)	V3.0	V3.0	V3.1
Files.Download (Page 237)	V3.0	V3.0	V3.1
Files.Create (Page 238)	V3.0	V3.0	V3.1
Files.Rename (Page 240)	V3.0	V3.0	V3.1
Files.Delete (Page 241)	V3.0	V3.0	V3.1
Files.CreateDirectory (Page 243)	V3.0	V3.0	V3.1
Files.DeleteDirectory (Page 244)	V3.0	V3.0	-
DataLogs.DownloadAndClear (Page 245)	V3.0	V3.0	V3.1

Method	**Standard/T-CPU as of firmware version**	**F-CPUs as of firmware version**	**R/H-CPUs as of firmware version**
Reading information from SIMATIC Safety			
Failsafe.ReadRuntimeGroups (Page 247)	-	V3.1	-
Failsafe.ReadParameters (Page 248)	-	V3.1	-

Table 5-1 Overview of web applications that can be loaded by the user

Web applications that can be loaded by the user (Page 250)	**Standard/T-CPU as of firmware version**	**F-CPUs as of firmware version**	**R/H-CPUs as of firmware version**
WebApp.Create (Page 254)	V2.9	V2.9	-
WebApp.Delete (Page 255)	V2.9	V2.9	-
WebApp.Rename (Page 256)	V2.9	V2.9	-
WebApp.Browse (Page 257)	V2.9	V2.9	-
WebApp.SetState (Page 259)	V2.9	V2.9	-
WebApp.SetDefaultPage (Page 260)	V2.9	V2.9	-
WebApp.SetNotFoundPage (Page 261)	V2.9	V2.9	-
WebApp.SetNotAuthorizedPage (Page 262)	V2.9	V2.9	-
WebApp.BrowseResources (Page 264)	V2.9	V2.9	-
WebApp.CreateResource (Page 266)	V2.9	V2.9	-
WebApp.DeleteResource (Page 268)	V2.9	V2.9	-
WebApp.RenameResource (Page 269)	V2.9	V2.9	-
WebApp.DownloadResource (Page 270)	V2.9	V2.9	-
WebApp.SetResourceVisibility (Page 271)	V2.9	V2.9	-
WebApp.SetResourceETag (Page 273)	V2.9	V2.9	-
WebApp.SetResourceMediaType (Page 274)	V2.9	V2.9	-
WebApp.SetResourceModificationTime (Page 275)	V2.9	V2.9	-

API endpoint

As an RPC protocol, JSON-RPC V2.0 is based on HTTP. The Web API can be reached via POST Requests to the following URL:

```
https://[ip]/api/jsonrpc
```

The Web API supports bulk operations as defined in JSON-RPC 2.0. While bulk operations are not explicitly limited by a fixed number of method calls, there is a limit for the HTTP request body. The limit differs depending on the firmware version of the CPU:

- Limit of 64 KB for CPUs with firmware version ≤ V3.0
- Limit of 128 KB for CPUs as of firmware version ≥ V3.1

As of firmware version V3.1, you can use the API method Api.GetQuantityStructures to read the limit.

An example of the required structure of an HTTP Request and HTTP Response for successfully making a Web API request can be found in the chapter Web API integration (Page 160).

Compatibility with regard to future extensions of the Web API

The order of attributes within a JSON object does not affect API clients.

Web server responses to requests via the API may be extended with new JSON attributes in future firmware versions, e.g. to enrich results with more details.

Error codes of possible API error messages based on JSON-RPC may change in future firmware versions, if applicable, and existing error messages may be made more precise.

NOTE

To check if your API request was successful, first check if the request was successful in general. You can evaluate the JSON-RPC error codes for detailed information.

Textual error information only provides information. If you want to implement error evaluation in a way specific to an application, use the corresponding numeric error codes of the error message.

The precision of floating point values may differ in the display from other clients, such as the TIA Portal.

Bulk requests are always processed in descending order, whereby the individual requests are contained in the HTTP request body in descending order.

If an API method includes time stamps, these time stamps are always returned based on UTC time. Examples of this are the time stamps of the Files.Browse and Api.BrowseTickets methods. With the API methods Plc.ReadSystemTime and Plc.ReadTimeSettings, you can read out the system time and determine the PLC local time.

Supported clients

The products and versions listed in the following table were tested for the Web API. The column "As of Version" lists the tested version as of which the clients are supported:

Product *	As of Version	Supported functions
Chrome-based Desktop web browser (e.g. Google Chrome) (https://chromium.woolyss.com/)	75.x (Windows and Android)	Web API access with JavaScript for asynchronous requests
Mozilla Firefox (https://www.mozilla.org/en-US/firefox/)	64.x (long-term support)	
Microsoft Internet Explorer	11.x (Windows 7, Windows 10)	
Microsoft Edge	44.x Windows 10	
Apple Safari	12.x iOS	
Opera	58.x	
SIMATIC HMI Panels	2	
Microsoft C# (https://docs.microsoft.com/en-us/dotnet/api/system.net.webrequest?view=netframework-4.7.2) with WebRequest class and Json.Net library (https://www.newtonsoft.com/json)	.Net Framework 4.7	Web API access for pure HTTP requests and Json.Net for generating and parsing content
GNU Wget (https://www.gnu.org/software/wget/)	1.20 Windows	Web API access for pure HTTP requests, e.g. for automatic archiving of DataLogs.
cURL (https://curl.haxx.se/)	7.63.x Windows	
Python (https://www.python.org/downloads/)	3.10	
Microsoft PowerShell	5.0	Web API access for pure HTTP Requests with Invoke WebRequest and ConvertTo-Json/ConvertFrom-Json for generating and parsing content
WebserverApi Client Library for .NET (https://github.com/siemens/simatic-s7-webserver-api)	1.0.1	Web API access for pure HTTP requests in C#. The library is also available as a NuGet package at the following address (https://www.nuget.org/packages/Siemens.Simatic.S7.Webserver.API).

* Not included in the scope of delivery of the product described here

Github

You can also find examples of using API methods in practice on Github at the following repository (https://github.com/siemens/simatic-s7-webserver-api).

5.1 Web API integration

In the following section you will find examples of how to integrate the Web API into your application.

Structure of an HTTP Request and HTTP Response

The following section shows the example of the required structure of an HTTP Request and HTTP Response for successfully making a Web API request.

```
POST /api/jsonrpc HTTP/1.1
Host: 192.168.3.14
Content type: application/json
Content length: 92

[{"jsonrpc":"2.0","method":"Api.Login","params":{"user":"User1","password":"SecurePassword"},"id":999}]

HTTP/1.1 200 OK
Content type: application/json
Cache-Control: no-cache
Pragma: no-cache
Expires: 0
Access-Control-Allow Origin: *
Access-Control-Allow Headers: Content-Type,X-Auth Token
Access-Control-Allow Methods: POST,OPTIONS
transfer-encoding: chunked
date: Tue, 23 Apr 2019 17:50:31 GMT

[{"jsonrpc":"2.0","id":999,"result":{"token":"Sy8pe3VNv86rTMldzFBsYzmw12Lg"}}]
```

Web API examples

The following section contains related examples of how you can use the described methods in the Web API.
The examples use HTML, JSON and JQuery library for asynchronous requests.

NOTE

Information used in the examples

Note that the names of the methods, parameters and the JavaScript code are specified without liability and can deviate from the current API methods.

Example 1

Example 1 shows a code section that wants to maintain a session with JavaScript. For this purpose, a one-time ping request is sent using the Api.Ping method. If the intervals at which the ping request is sent are less than the timeout of 2 minutes, the user remains permanently logged in.
A maintained session lends itself to scenarios such as operating and monitoring tasks.

```
$.post({
url:"https://192.168.2.132/api/jsonrpc",
headers:{
'X-Auth-Token':"Sy8pe3VNv86rTMldzFBsYzmw12Lg"
},
data:JSON.stringify({"jsonrpc":"2.0", method:"Api.Ping", "id":1}),
success:function(data){ console.log(data); }
dataType:"text",
contentType:"application/json"
});
```

NOTE

X-Auth-Token

To extend the session, you must send the token (X-Auth-Token) as an HTTP header to the CPU.

When you call the Api.Ping method without a token, the session is not extended because the CPU cannot assign a token to the user.

Example 3 shows an example of a token in the HTTP request.

In the example, the selected user has the necessary authorizations. The methods after the login request were successfully carried out, as the following result shows.

```
{"jsonrpc":"2.0","id":1,"result":"ZWlmbnJwZmplb3Nwd2l1Y3N3dWE="}
```

Example 2

Example 2 shows a client that wants to log in to the CPU with JavaScript and calls several methods within an HTTP request using a bulk request.

```
$.post({
url:"https://192.168.2.132/api/jsonrpc",
data:JSON.stringify([
{"jsonrpc":"2.0", "id":1, method:"Api.Login",
params:{user:"Admin",password:"12345"} },
{"jsonrpc":"2.0", "id":2, method:"Api.GetPermissions" },
{"jsonrpc":"2.0", "id":3, method:"Api.Browse" }]),
success: function(data){ console.log(data); },
dataType: "text",
contentType: "application/json"});
```

The following section shows an example of a bulk request response. The selected user has the necessary authorizations. The methods after the login request were successfully carried out with the authorizations of the authenticated user.

```
[
"jsonrpc": "2.0", "id": 2, "result":[
{ "name": "Api.Browse" },
{ "name": "Api.Login" },
{ "name": "Api.Logout" },
{ "name": "Api.GetPermissions" },
{ "name": "PlcProgram.Read" },
{ "name": "PlcProgram.Write" }
]
```

Example 3

Example 3 shows a bulk request for read and write access to a stack of tags in a single HTTP request. This procedure is recommended for bulk requests, as it is more efficient than a series of single accesses and therefore places less load on the CPU.

```
$.post({
url:"https://192.168.2.132/api/jsonrpc",
data:JSON.stringify([
{"jsonrpc":"2.0", "id":1, method:"PlcProgram.Read"},
{"jsonrpc":"2.0", "id":2, method:"PlcProgram.Read",
params:{"var":"\"MyDB\".InvalidField"}},
{"jsonrpc":"2.0", "id":3, method:"PlcProgram.Read",
params:{"var":"MyDB.MyDate"} },
{"jsonrpc":"2.0", "id":4, method:"PlcProgram.Write",
params:{"var":"\"BoilerControl\".TempSetPoint", value:9001} }]),
success: function(data){ console.log(data); },
dataType: "text",
contentType: "application/json",
headers: {
"X-Auth-Token":"d29kamV3cGxtdm5keHNhcXd1aXJ0empkZXN3cQ=="}});
```

The bulk request contains an invalid tag with an error message providing information about this. All other methods were successfully carried out, as the following result shows.

```
[
{"jsonrpc":"2.0","id":1,"result":{"value":42}},
{"jsonrpc":"2.0","id":2,"error":{"code":201,"message":"Invalid
address"}},
{"jsonrpc":"2.0","id":3,"result":{"value":"1990-01-01"}},
{"jsonrpc":"2.0","id":4,"result":true}
]
```

5.2 Web API sessions

Timeout for Web API sessions

NOTE

If a Web API call is not made within a session before 120 seconds have elapsed, the CPU terminates the session with a logout event. A timeout reset is initiated by every successful action of the user in which a toke is supplied.

Call the Api.GetPermission or Api.Ping method cyclically within the timeout grid to ensure that:

- Your session remains active
- Your authorizations for the call of other methods remain active

Limitations for Web API sessions

The CPU limits the number of active sessions. The following table shows the limitations based on the memory platform used.

CPUs	Limitation
1510 to 1513	50
1514, 1515, 1516 and 1504D TF	100
1517, 1518 and 1507D TF	200

Limitation of the active Web API sessions

If you request another authentication token and none are available, then the request is rejected.

Changes to CPU user administration

The following applies to configured CPUs with firmware version ≤ V3.0: If the configuration of the CPU user management changes (by downloading the HW configuration in the TIA Portal), e.g. password changed or user removed, the CPU terminates all sessions with a logout event.

The following applies to configured CPUs with firmware version ≥ V3.1: If a user has been authenticated and the project is subsequently loaded into the CPU, deleted or deactivated users are logged out. If only the password or role of the user changes, the user remains authenticated.

Security events

The following applies to configured CPUs with firmware version ≤ V3.0: The CPU generates a security event for successful and failed logins. The CPU enters this security event in the diagnostics buffer.

For configured CPUs with firmware version ≥ V3.1, security events are logged in the CPU's internal syslog buffer. You can read out the entries with the Syslog.Browse (Page 222) method.

You can find more information about syslog messages in the Communication function manual (https://support.industry.siemens.com/cs/ww/en/view/59192925).

5.3 Ticket mechanism

With the web server as of firmware version V2.9, you can use the ticket mechanism of the Web API . The ticket mechanism is the basis for all file-based methods, such as the download of resources from the CPU.

The ticket mechanism enables you to:

- Transfer large amounts of data outside of the JSON-RPC protocol.
- Call status information, for example, to implement progress indicators in your application and respond conveniently to mode changes

NOTE

4 tickets per session

The ticket mechanism enables you to use up 4 tickets simultaneously per session.

The number of tickets that can be created per type is additionally limited. For more information, see the API methods that create a ticket:

- WebApp.CreateResource (Page 266)
- WebApp.DownloadResource (Page 270)
- Plc.CreateBackup (Page 229)
- Files.Download (Page 237)
- Files.Create (Page 238)
- PlcProgram.DownloadProfilingData (Page 194)

Principle of the ticket mechanism

Data transfers outside the JSON-RPC protocol are initiated by Web API methods, such as WebApp.DownloadResource. This method returns a specific identifier, a so-called "ticket". The ticket can be redeemed by a subsequent request to another HTTP end point. In the request, the data are exchanged with standard HTTP mechanisms.

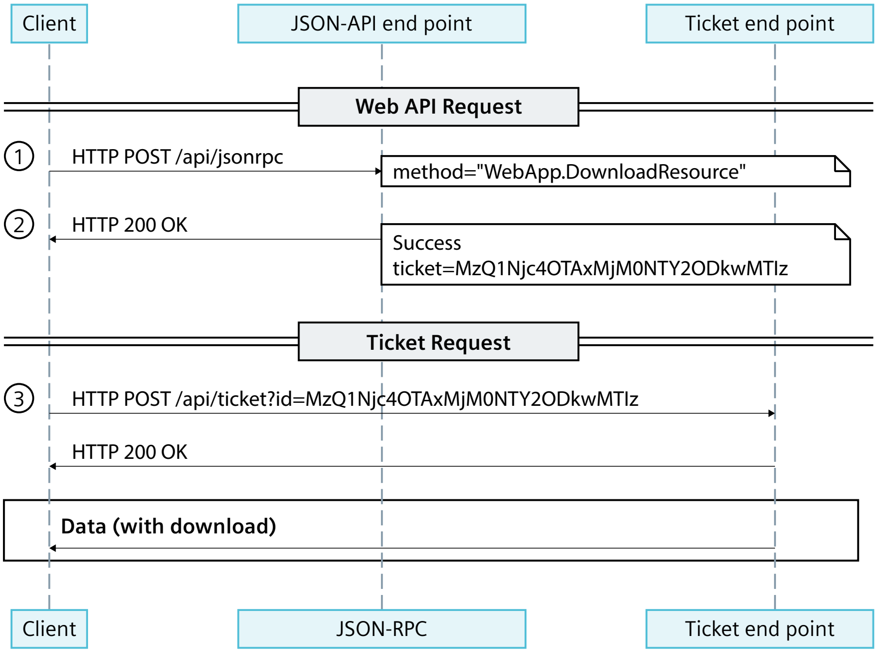


Figure 5-2 Ticket mechanism

① To request the upload or download of a resource from/to the CPU, a client sends a request to the JSON-RPC.

② After successful authorization and resource check, the client receives a valid ticket ID.

③ The client sends a request to the ticket end point with the valid ticket ID and an X-Auth-Token. The data is included here during upload to the CPU. The request is confirmed to the client with "HTTP 200 OK" or "204 No Content". The data is returned in a download from the CPU.
Ticket IDs are one-time tokens that may not be reused by the client or by the CPU after being redeemed.

NOTE

Changes in firmware version V3.0

As of firmware version V3.0, no X-Auth-Token is required for the ticket end point.

Do not share the token with third parties, because the owner of the token gets control over a ticket.

As of firmware version V3.0, GET requests for downloads are possible in addition to POST requests. For GET requests, a content disposition header is returned with the HTTP response. It contains a default file name for storage in the web browser. The browser uses the file name as the storage location. However, you can also evaluate the file name programmatically using a programming language (for example, via C#).

Ticket end point

The ticket end point is accessible via GET (for downloads only), POST and OPTIONS requests to the following URL:

```
https://[ip]/api/ticket?id=<ticket-id>
```

The ticket mechanism alone is not enough to execute a file action. To do this, you must call a specific method. In the web server of the firmware version V2.9, you use the ticket mechanism for the following methods:

- WebApp.CreateResource (Page 266)
- WebApp.DownloadResource (Page 270)

Procedure to execute a file action

To create a resource or download one from the CPU, follow these steps:

1. Call the corresponding method with the necessary method parameters, for example, WebApp.DownloadResource. The method returns a character string that includes a valid ticket ID.
2. Call the ticket end point via POST request with the returned ticket ID. The method will then execute the download, for example.

For more information, see the paragraph "Principle of the ticket mechanism".

NOTE

In some cases, programming languages perform preprocessing of text files before the upload. For example, a UTF-8 BOM-encoded (Byte Order Mark) file in Javascript is converted to a UTF8 file in advance.

Examples

Below you will find two examples for further processing of the tickets. The examples use HTML, Json and JQuery library.

WebApp.CreateResource (upload a resource):

```
<td><input id="fileForticketCustomerExampleUpload" type="file"
onchange="fReadFile()" /></td>
var fileReader = new FileReader();
function fReadFile() {
   fileReader = new FileReader();
   fileReader.readAsText(this.files[0]);
}
$("#ApiUploadticketCustomerExample").click(function () {
     var ticketId = $("#iApiUploadticketIdCustomerExample").val();
     var content = fileReader.result; // e.g. <!DOCTYPE
html><html><head>...
     $.post({
       url: "https://" + window.location.hostname +
"/api/ticket?id=" + ticketId,
       headers: { "X-Auth-Token": token, "Content-Type":
"application/octet-stream" },
       contentType: "application/json",
       data: content,
       // ticketing: status = 204: No content (=> no data) upload 
has finished successfully:
       success: function (data, textStatus, jqXHR) { if 
(jqXHR.status == 204) { 
$("#ApiUploadticketCustomerExampleRes").text("true"); } },
//print error to the document
       error: function (jqXHR, textStatus, errorThrown) { 
$("#ApiUploadticketCustomerExampleRes").html("<tr><td>code: 
</td><td>" + jqXHR + "</td></tr><tr><td>textStatus:</td><td>" + 
textStatus + "</td></tr><tr><td>error:</td><td>" + errorThrown + 
"</td></tr>"); }
     });
   });
```

WebApp.DownloadResource (download a resource):

```
// Function to save data in a file - will be stored in the default 
download folder
function saveDataInFile(data, filename, type) {
   var file = new Blob([data], { type: type });
   if (window.navigator.msSaveOrOpenBlob) // IE10+
     window.navigator.msSaveOrOpenBlob(file, filename);
   else { // Others
     var a = document.createElement("a"),
       url = URL.createObjectURL(file);
     a.href = url;
     a.download = filename;
     document.body.appendChild(a);
     a.click();
     setTimeout(function () {
       document.body.removeChild(a);
       window.URL.revokeObjectURL(url);
     }, 0);
   }
}
 $("#ApiDownloadTicketCustomerExample").click(function () {
     var ticketId = $("#iApiDownloadTicketIdCustomerExample").val();
     var filename = 
$("#iApiDownloadTicketCustomerExampleFileName").val();  //e.g. index
     var type = 
$("#iApiDownloadTicketCustomerExampleFileType").val();    //e.g. 
text/html
     $.post({
       url: "https://" + window.location.hostname + 
"/api/ticket?id=" + ticketId,
       headers: { "X-Auth-Token": token, "Content-Type": 
"application/octet-stream" },
       contentType: "application/json",
       // ticketing: status = 200: download has finished 
successfully:
       success: function (data, textStatus, jqXHR) { if 
(jqXHR.status == 200) { download(data, filename, type); } },
//print error to document
       error: function (jqXHR, textStatus, errorThrown) { 
$("#ApiDownloadTicketCustomerExampleRes").html("<tr><td>code: 
</td><td>" + jqXHR + "</td></tr><tr><td>textStatus:</td><td>" + 
textStatus + "</td></tr><tr><td>error:</td><td>" + errorThrown + 
"</td></tr>"); }
     });
});
```

Ticket methods

Two methods are available for handling your tickets as an authenticated user:

- You can use the Api.BrowseTickets (Page 168) method to find out which tickets are currently active for you and read out the status for the active tickets in each case.
- After the action, you can delete your ticket with the Api.CloseTicket (Page 170) method to clean the application with regard to active tickets.

5.3.1 Api.BrowseTickets

The Api.BrowseTickets method calls the status of all active tickets of a logged-in user. Alternatively, the status of a specific ticket can be called. This will provide information on the ticket status. You can also call this method during an active file action to retrieve the current status. Depending on the file action, additional information is provided by Api.BrowseTickets.

To call the Api.CloseTicket method, you do not need an authorization but you do need a valid session token. You can find information on the session token in the section Api.Login (Page 171).

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-2 Api_BrowseTickets_Request (object)

Name	Required	Data type	Description
id	No	string	The ticket ID that was returned by an API method for use by the ticket system. If the parameter is not specified, then the response structure returns all valid tickets of the user.

Example

In the following example, a ticket ID is transferred as parameter.

```
{
"id": "U2VyaW91c2x5LCBTdG9wIGl0ISE6"
}
```

Response structure

The following table shows you the structure of the server response to a successful request.

Table 5-3 Api_BrowseTickets_Response (object)

Name	Required	Data type	Description
max_tickets	yes	number	Maximum number of tickets for one session (4)
tickets	Yes	array of Api_ BrowseTickets_ Ticket_ Response	Ticket ID
id	yes	string	Ticket ID
date_created	yes	string	ISO8601 time stamp as string. Time of the ticket creation based on the CPU time
provider	yes	string	Name of the method that has created the ticket, for example, WebApp.DownloadResource
state	yes	string	Current ticket status. The following alternatives are possible: "created", "active", "", "completed" or "failed"
data	No	object	Additional ticket data: Some ticket-based methods offer users additional information. This additional information is provided via "data" and described in the sections of the respective API methods.

Example

A response is shown in the following example.

```
{
"max_tickets": 4,
"tickets":
[
{
"id": "U2VyaW91c2x5LCBTdG9wIGl0ISE6",
"date_created": "2021-01-15T08:00:00-05:00",
"provider": "WebApp.DownloadResource",
"state": "active"
}
]
}
```

NOTE

Additional ticket data (as of version V18)

As of version V18, each ticket is extended by the entry "bytes_processed" as part of the "data" object. This entry specifies how many bytes have been transferred when downloading or uploading a ticket until the Api.BrowseTicket method is called.

Possible error messages

The following table shows the possible error messages of the Api.BrowseTickets method.

Error code	Error message	Meaning
400	Not Found	The returned ticket ID does not exist in the ticket list of the user or does not match the assigned session token.

5.3.2 Api.CloseTicket

You use the Api.DeleteTicket method to delete a ticket provided by the system that is assigned to the current user session.

To call the Api.CloseTicket method, you do not need an authorization but a valid session token. You can find information on the session token in the section Api.Login (Page 171).

Structure of the request

The following table informs you about the necessary parameters for the request.

Table 5-4 Api_CloseTicket_Request (object)

Name	Required	Data type	Description
id	yes	string	The ticket ID that was returned by an API method for use by the ticket system.

Example

In the following example, a ticket ID is transferred as parameter.

```
{
"id": "U2VyaW91c2x5LCBTdG9wIGl0ISE6"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the Api.GetTicketStatus method.

Error code	Error message	Meaning
400	Not Found	The returned ticket ID does not exist in the ticket list of the user. The returned ticket ID or does not match the assigned session token.

5.4 Web API basic functions

The following section gives an overview of all Web API basic functions with extracts from the corresponding HTML code.

NOTE

Files which contain Web API methods must be encoded and stored in the UTF-8 character encoding.

For detailed examples of an integration of the Web API into your web application, refer to the section Web API integration (Page 160).

5.4.1 Api.Login

The Api.Login method checks the login data of the user and on successful verification opens a new Web API session. The method requests the name and the password of the user in plain text as proof of authorization. The user name and the password are encrypted before they are transferred to the server.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-5 Api_Login_Request (object)

Name	Required	Data type	Description
user	Yes	string	The user name
password	Yes	string	The current password in plain text
include_web_application_cookie	No	bool	As of CPU firmware version V2.9, Web Server supports the optional parameter "include_web_application_cookie". This parameter specifies: • Whether a "web_application_cookie" cookie was generated for access to protected web applications • Whether you want to return the cookie with the response to the successful login

Examples

The following examples show the parameters that are required or optional for calling the Api.Login method.

```
{
"user": "User1",
"password": "1234"
}
```

Parameter "include_web_application_cookie":

```
{
"user": "User1",
"password": "SecurePassword",
"include_web_application_cookie": true
}
```

After successful authentication the user receives a token. The token identifies the user as successfully authenticated.

In addition to the token, the Api.Login method optionally returns a cookie that is required to access web applications:

```
{
"token": "eG9mcHdhaGR0dWVsdm5teGFxcGw=",
"web_application_cookie": "Cb5xdhgiokr0dWVsdm5teGCncFb="
}
```

The "web_application_cookie" cookie is set with a value from the HTTP response of the login as "siemens_web_secure" cookie.

Token

The token comprises a 28-byte string. The token is transferred in encrypted form.

For every additional request which requires authentication, you have to specify the assigned token in the HTTP request header. If further communication no longer takes place in the meantime, the token becomes invalid after maximum 2.5 minutes. Each new request within a session extends the validity of the token by another 2 to 2.5 minutes, calculated from the completion of the request processing by the server.

The token is not required for methods that do not require authentication. However, you can still enter the token. If the token is passed when a method is called, the timeout of the corresponding session is reset.

When you call the Api.Ping method without a token, for example, the session is not extended because the CPU cannot assign a token to the user.

The following methods work with and without tokens:

- Api.Browse
- Api.Ping
- Api.GetPermissions

Receiving a token for a passwordless user account without password

If a user wants to use a user account without password to access the Web API, they must authenticate with the API method Api.Login using the user name "Everybody"/"Anonymous" and an empty password (""). The name of the specific user depends on the firmware version of the CPU:

- For projects with a firmware version ≤ V3.0: Static user management with the user "Everybody"
- For projects with a firmware version ≥ V3.1: Local user management with the user "Anonymous"

The "Anonymous" user can be deactivated in STEP 7 and is then not available on the CPU. Unverified users are not downloaded to the CPU. This means that a call of the API method Api.Login with the user name "Anonymous" fails if the user was deactivated in STEP 7.

NOTE

User "Everybody"/"Anonymous"

Note that, unlike TIA Portal, the API only accepts the English notation "Everybody" or "Anonymous".

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-6 Api_Login_Response (object)

Name	Required	Data type	Description
token	Yes	string	The token indicates that its holder has successfully authenticated themselves at the Web API. The returned token must be passed via the HTTP request header "X-Auth-Token" in subsequent Web API requests.
web_application_cookie	No	string	Only present if "include_web_application_cookie" is "true". The holder of the token has successfully authenticated themselves with the Web API and has authorization to access protected web applications.
password_expiration	No	object of type Api_Login_PasswordExpiration_Response	This object contains information on the expiration of the password, if: • The "local" authentication mode is used for the CPU and • The password policy is activated on the CPU
runtime_timeout	No	string	If the "local" authentication mode is used for the CPU: ISO 8601 time span as string Time span of inactivity in seconds after which a client application is to perform a logout using the API method Api.Logout.

Table 5-7 Api_Login_PasswordExpiration_Response (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO8601 time stamp as a string in Coordinated Universal Time (UTC) Indicates when the user password expires. The accuracy must be specified in seconds.
warning	Yes	bool	This parameter specifies whether the warning threshold was reached before the password expired.

Examples

Successful login for a user without expiration of the password (either static user management or password expiration deactivated):

```
{
  "token": "TXlMdWdnYWdlSGFzVGhlU2FtZSE="
}
```

Successful login for a user with password expiration:

```
{
  "token": "TXlMdWdnYWdlSGFzVGhlU2FtZSE=",
  "password_expiration":
  [
    "timestamp": "2023-11-05T18:25:43Z",
"warning": true
  }
}
```

Successful login for a user with expiration of the password and runtime_timeout of 30 minutes:

```
{
  "token": "TXlMdWdnYWdlSGFzVGhlU2FtZSE=",
  "runtime_timeout": "PT30M",
  "password_expiration":
  [
    "timestamp": "2023-11-05T18:25:43Z",
"warning": true
  }
}
```

Possible error messages

The following table shows the possible error messages of the Api.Login method.

Error code	Error message	Meaning
4	No resources	The system does not have the required resources to carry out this request. Perform the request again as soon as enough resources are available again.
100	Login failed	The user name and/or password are not permissible. Assign a permissible user name and a permissible password. Another reason why the login failed may be an active brute force attack.
101	Already authenticated	The current X-Auth-Token is already authenticated. Use Api.Logout before you authenticate yourself again.
102	Login Failed - Password expired	The password of the user account has expired. The user must change the password in order to be able to successfully authenticate again.

Example

You can find an example with further processing of the cookie "web_application_cookie" in section Example: Web page for monitoring and controlling a wind turbine.

5.4.2 Api.GetPermissions

After the successful login, the Api.GetPermissions returns a list of actions for whose execution the user is authorized.

Response structure

The following table shows the structure of server responses to successful requests:

Table 5-8 Api_GetPermissions_Response (array of objects)

Name	Required	Data type	Description
Name	Yes	string	Name of authorization

Example

The following example shows the actions for which the user is authorized.

```
[
{ "name": "read_value" },
{ "name": "change_operating_mode" }
]
```

Checkable authorizations

You can use the Web API to check the authorizations for the following functions.

Action	User authorization	As of the firmware version of the CPU
flash_leds	Identify device	V2.8
acknowledge_alarms	Acknowledge alarms	V2.8
read_diagnostics	Query diagnostics data from the CPU without being permitted to change data.	V2.8
read_value	Read process data from the CPU.	V2.8
write_value	Write process data to the CPU.	V2.8
open_user_pages	Call user-defined pages on the CPU.	V2.8
manage_user_pages	Change user-defined pages on the CPU.	V2.9
read_file*	Read the contents of files on the CPU.	V2.8
write_file*	Change the contents of files and folders on the CPU.	V2.8
change_operating_mode	Change the operating mode.	V2.8
backup_plc	Back up the CPU configuration.	V2.8

* Also available for the R/H-CPU of the redundant S7-1500R/H system (as of firmware version V3.1).

Action	User authorization	As of the firmware version of the CPU
restore_plc	Restore configuration of the CPU.	V2.8
failsafe_admin	Make fail-safe changes on the CPU.	V2.8
update_firmware	Perform firmware update	V3.0
read_watch_table_value	Read value of a tag in watch table	V3.1
write_watch_table_value	Write value of a tag in watch table	V3.1
read_syslog*	Read the SysLog buffers of the CPU	V3.1
change_time_settings	Change the system time settings of the CPU	V3.1
change_webserver_default_page	Change the default page of the Web server	V3.1
download_service_data	Load the service data of the CPU	V3.1

* Also available for the R/H-CPU of the redundant S7-1500R/H system (as of firmware version V3.1).

For projects with firmware version ≤ V3.0, the Web API checks the authorization based on the rights and passwords assigned in STEP 7 in the Inspector window in the "Web server > User management" area.

For projects with firmware version ≥ V3.1, the Web API checks the authorization based on the rights and passwords assigned in STEP 7 in the project tree in the "Security settings > Users and roles" area.

A description of the user management can be found in section "Configuring the Web server (Page 26)".

5.4.3 Api.Browse

The Api.Browse method gives you a list of all methods that you can call via the Web API with the current firmware. This provides you with an overview of all the methods supported by the CPU.

No authorization is required for calling the Api.Browse method.

Example

The following example shows the HTTP request and a possible response to the request for the Api.Browse method.

```
POST /api/jsonrpc HTTP/1.1
Host: 192.168.3.14
Content type: application/json
Content length: 48

[{"jsonrpc":"2.0","method":"Api.Browse","id":1}]
```

Response of the server:

```
[
{ "name": "Api.Browse" },
{ "name": "Api.GetCertificateUrl" },
{ "name": "Api.GetPermissions" },
{ "name": "Api.Login" },
{ "name": "Api.Logout" },
{ "name": "Api.Ping" },
{ "name": "Api.Version" },
{ "name": "PlcProgram.Browse" },
{ "name": "PlcProgram.Read" },
{ "name": "PlcProgram.Write" }
]
```

NOTE

Checking authorizations

The Api.Browse method does not filter the list of the available methods by the individual authorizations of users.

The list of available methods may therefore contain methods which the user may not execute without authorization.

Possible error messages

The following table shows the possible error messages of the Api.Browse method.

Error code	Error message	Meaning
4	No resources	The system does not have the necessary resources to execute the Web API request. Perform the request again as soon as enough resources are available again.

5.4.4 Api.Version

Use the Api.Version method to request the current version number of the Web API. You can draw conclusions from the version number:

- The functions supported by the CPU version
- The hardware functional status of the CPU

This information lets you implement applications that dynamically adapt to the scope of functions offered by the contacted CPU. An application can support multiple CPU versions.

No authorization is required for calling the Api.Version method.

Example

The following example shows a possible result of calling the Api.Version method.

```
3.28
```

The version number is displayed as a floating-point number and is incremented with every release and every change of the Web API.

5.4.5 Api.Ping

The Api.Ping method outputs a unique ID for the CPU used. You can use it to determine whether the CPU can be reached. The CPU ID comprises a 28-byte string. The system assigns a new, unique CPU ID after each restart (POWER ON - POWER OFF) or warm start of the CPU. By comparing this with previously output IDs, you can also determine whether the CPU was restarted in the meantime.
No authorization is required for calling the Api.Ping method.

Example

The following example shows the output of a CPU ID:

```
"ZWlmbnJwZmplb3Nwd2l1Y3N3dWE="
```

5.4.6 Api.GetCertificateUrl

The Api.GetCertificateUrl method returns a relative URL (https://<IP>) with which you can retrieve the SSL certificate of the Web server.

Example

The following example shows the result of the Api.GetCertificateUrl method call.

```
"/MiniWebCA_Cer.crt"
```

The method outputs a string with a relative URL to the root directory of the CPU Web server (https://[ip-address]).
If the Web server has not been configured with a CA certificate generated via the global security settings, the method outputs an empty string.

5.4.7 Api.Logout

The Api.Logout method removes the token from the list of active Web API sessions and ends the session.
The Api.Logout method returns the status of whether the logout was successful or not. For security reasons, however, the method always returns the Boolean value "true" even if the token is invalid.

5.4.8 Api.GetQuantityStructures

The Api.GetQuantityStructures method returns different structure information of the Web server.

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-9 Api_GetQuantityStructures_Response (object)

Name	Required	Data type	Description
webapi_max_http_request_body_size	Yes	number	Maximum size of the HTTP request body of a Web API request in bytes
webapi_max_parallel_requests	Yes	number	Maximum number of parallel requests at the API end point
webapi_max_parallel_user_sessions	Yes	number	Maximum number of parallel user sessions that use the API end point

Example

The following example shows the response with the read parameters of a CPU.

```
{
  "webapi_http_request_body_size": 131072,
  "webapi_parallel_requests": 4,
  "webapi_parallel_user_sessions": 200
}
```

5.4.9 Api.ChangePassword

You can change the password for a user account with the Api.ChangePassword method.

Recommendation: Before changing a password, read the password policy from the CPU using the Api.GetPasswordPolicy API method. If the new password does not conform to the password policy of the CPU, a corresponding error message is returned.

No prior authorizations are required to call the Api.ChangePassword method, but you must enter the current password for this call.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-10 Api_ChangePassword_Request (object)

Name	Required	Data type	Description
username	Yes	string	The user name for which the password was changed.
password	Yes	string	The current password of the specified user.
new_password	Yes	string	The new password.

Example

In the following example, the password is changed for the user account "HappyUser".

```
{
"username": "HappyUser",
"password": "mycurrentpassword",
"new_password": "mynewpassword"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows possible error messages with the Api.ChangePassword method.

Error code	Error message	Meaning
5	System is read-only	The memory card is write-protected. Therefore, the password cannot be changed.
6	Not accepted	The password change is not performed because a CPU was configured with firmware version < V3.1. The method can only be used with CPUs as of firmware version V3.1.
100	Login failed	The user name and password combination is invalid. Assign a permissible user name and a permissible password. Another reason why the login failed may be an active brute force attack.
103	New password does not follow password policy	The provided new password does not match with the required password policy. Assign a password conforming to the password policy. The Api.GetPasswordPolicy method provides you with the password policy of the CPU, if the CPU is in "local" authentication mode.
104	New password matches former password	The new password is identical with the previous password. Assign a different password. Note that the CPU does not store a password history. The comparison is therefore only performed between the new and previous password.

5.4.10 Api.GetPasswordPolicy

Passwords must fulfill specific criteria to be secure. The Api.GetPasswordPolicy method provides you with the password policy of the CPU. The password policy is a global setting in the STEP 7 project and applies for all users of the Web server. The method does not contain any information on the expiration of the password. Any user, including unauthenticated users ("Anonymous"), can call this API method.

No authorization is required for calling the Api.GetPasswordPolicy method.

Response structure

The following table shows you the structure of server responses to successful requests.

Table 5-11 Api_GetPasswordPolicy_Response (object)

Name	Required	Data type	Description
password_policy	Yes	object	The object represents the current password policy of the CPU.
min_password_length	Yes	number	The minimum length of the password in UTF-8 character encoding. Value range: 8 to 255
max_password_length	Yes	number	The maximum length of the password in UTF-8 character encoding.
min_digits	Yes	number	The minimum number of numerals (0 to 9) within the password. Value range: 0 to 255
min_special_characters	Yes	number	The minimum number of special characters within the password. Special characters are: !#$%&()*+,-./:;<=>?@[\]_{|}~^ Value range: 0 to 255
requires_uppercase_characters	Yes	bool	The password contains at least one uppercase character A to Z. Values: true or false
requires_lowercase_characters	Yes	bool	The password contains at least one lowercase character a to z. Values: true or false

Example

CPU project with a password policy

```
{
"password_policy":
{
"min_password_length": 8,
"max_password_length ": 120,
"min_digits": 2,
"min_special_characters": 1,
"requires_uppercase_characters": true,
"requires_lowercase_characters": true
}
}
```

Possible error messages

The following table shows possible error messages with the Api.GetPasswordPolicy method.

Error code	Error message	Meaning
6	Not accepted	The password policy cannot be read because a CPU is configured with a firmware version < V3.1. The method can only be used with CPUs as of firmware version V3.1.

5.4.11 Api.GetAuthenticationMode

You can read the current authentication mode of the CPU with the Api.GetAuthenticationMode method.
No authorization is required for calling the Api.GetAuthenticationMode method.

Possible authentication modes

The following authentication modes are available to you with the Web server.

Table 5-12 Authentication modes

Mode	String	Meaning	Supported from configured FW version of the CPU
Static user management	`static`	In this mode, changes are only possible by downloading the hardware configuration from the CPU. Passwords cannot be changed during runtime of the user program.	V2.8 to V3.0
Access control deactivated	`disabled`	No user management and authentication is possible in this mode. Only the "Anonymous" user is available. This specific user has full access/rights to the CPU and Web server functionality. As with every other user the Api.Login API method must be called. The returned X-Auth Token allows the full access right to all API functionalities.	V3.1
Local user management	`local`	In this mode, the authorizations and user roles continue to be configured as part of the STEP 7 project and a password change is possible via the Web API.	V3.1

Response structure

The following table shows you the structure of server responses to successful requests.

Table 5-13 Api_GetAuthenticationMode_Response (object)

Name	Required	Data type	Description
authentication_mode	Yes	array of strings	The parameter describes the current authentication mode supported by the CPU.

Example

In the following example, the authentication mode "local" (local user management) is read by the CPU.

```
{
"authentication_modes": ["local"]
}
```

5.5 Setting the Web server default page

The Web server of the CPU will use the default page if you do not specify a path along with the requested IP address, the domain and the host name in the web browser. You no longer have to configure the default page using the TIA Portal. This also eliminates the need to download the hardware configuration when changes are made.

5.5.1 WebServer.SetDefaultPage

Use the API method WebServer.SetDefaultPage to configure a web application as the default page of the Web server.

You can only configure web applications as the start page, but not web pages or user-defined pages of the CPU's Web server.

To call this method, you need "change_webserver_default_page" authorization.

Structure of the request

The following table contains information about the parameters of the request:

Table 5-14 Webserver_SetDefaultPage_Request (object)

Name	Required	Data type	Description
default_page	Yes	string	The name of the default page you want to use.

Examples

In the following example, the default page named "index.html" for the web application "webapp1" is used:

```
{
"default_page": "/~webapp1/index.html"
}
```

In the following example, the default page is set to "webapp1": In this case, you need to configure the web application to reference the specific default page of the web application.

```
{
"default_page": "/~webapp1/"
}
```

In the following example, the value for the default page name is empty. In this case, the system falls back to the default page as defined in the hardware configuration in the TIA Portal.

```
{
"default_page": ""
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
5	System is read only	The system is currently in a write-protected state. Changes are not currently permitted.
1300	Invalid default page	The default page provided is invalid. Make sure that you have used the default page name correctly.

5.5.2 WebServer.ReadDefaultPage

You use the API method WebServer.ReadDefaultPage to read the default page set with WebServer.SetDefaultPage.

Response structure

If successful, the response returns the following structure:

Table 5-15 Webserver_ReadDefaultPage_Response

Name	Required	Data type	Description
default_page	Yes	string	The default page currently configured during runtime. The returned value does not take into account the configuration of the hardware configuration start page in the TIA Portal.

5.6 Reading and writing process data

5.6.1 Supported data types

Binary representation

The Web API presents the values of primitive data types as pure binary data ("raw"). The binary data is formatted as a JSON array. Each element within an array represents a single data byte.

Simple display ("simple")

The Web API formats primitive data types into a readable form while preserving the ability to process the data using a program. The following section describes how primitive data types are represented as JSON data type.

Supported data types

The following table shows:

- The data types supported by the Web API for reading and writing process values
- The representation in the Web API
- The respective match of the data type in the TIA Portal

Name of the data type in the TIA Portal	Name of the data type in Web API	Representation in the Web API
Bool	bool	Boolean JSON value: **true** or **false**
Byte	Byte	JSON unsigned integer in a range from 0 to 255
USInt	usint	
Word	word	JSON unsigned integer in a range from 0 to 65 535
UInt	uint	
HW_ANY	hw_any	
HW_IOSYSTEM	hw_iosystem	
HW_DPMASTER	hw_dpmaster	
HW_DEVICE	hw_device	
HW_DPSLAVE	hw_dpslave	
HW_IO	hw_io	
HW_MODULE	hw_module	
HW_SUBMODULE	hw_submodule	
HW_HSC	hw_hsc	
HW_PWM	hw_pwm	
HW_PTO	hw_pto	
HW_INTERFACE	hw_interface	
HW_IEPORT	hw_ieport	

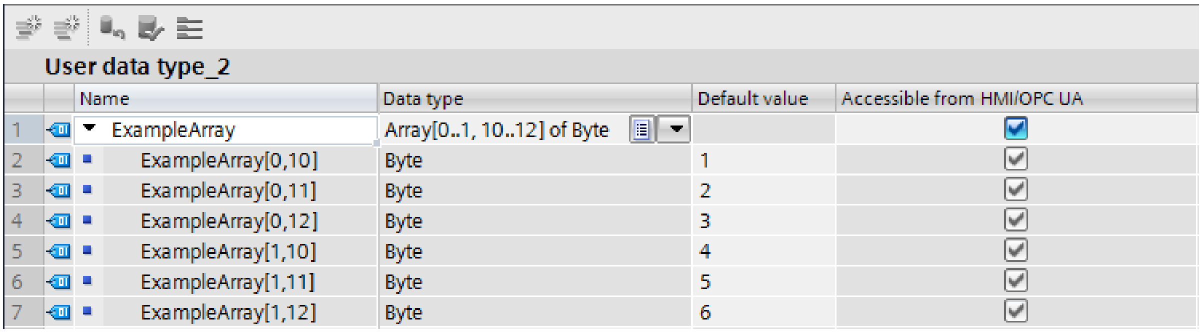
Name of the data type in the TIA Portal	Name of the data type in Web API	Representation in the Web API
CONN_ANY	conn_any	JSON unsigned integer in a range from 0 to 65 535
CONN_PRG	conn_prg	
CONN_OUC	conn_ouc	
PORT	port	
RTM	rtm_id	
PIP	pip	
DB_ANY	db_any	
DB_WWW	db_www	
DB_DYN	db_dyn	
DWord	dword	JSON unsigned integer in a range from 0 to 4 294 967 295
UDInt	udint	
AOM_IDENT	aom_ident	
EVENT_ANY	event_any	
EVENT_ATT	event_att	
EVENT_HWINT	event_hwint	
CONN_R_ID	conn_r_id	
LWord	lword	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from 0 to 18 446 744 073 709 551 615
ULInt	ulint	
SInt	sint	JSON signed integer in a range from -128 to 127
Int	int	JSON signed integer in a range from -32 768 to 32 767
OB_ANY	ob_any	
OB_DELAY	ob_delay	
OB_TOD	ob_tod	
OB_CYCLIC	ob_cyclic	
OB_ATT	ob_att	
OB_PCYCLE	ob_pcycle	
OB_HWINT	ob_hwint	
OB_DIAG	ob_diag	
OB_TIMEERROR	ob_timeerror	
OB_STARTUP	ob_startup	
DInt	dint	JSON signed integer in a range from -2 147 483 648 to 2 147 483 647.
LInt	lint	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from -9 223 372 036 854 775, 808 to 9 223 372 036 854 775 807
Real	real	JSON floating-point number If the floating-point number is infinite or NaN (not-a-number), the Web API returns the value **null** when reading a tag of this type.
LReal	lreal	
Character	char	JSON string with a single ASCII character in a valid range from 0 to 127 If a tag of this type is read with a value outside the valid range, the Web API outputs the value **null**.

Name of the data type in the TIA Portal	Name of the data type in Web API	Representation in the Web API
WChar	wchar	JSON string with a UTF-8 string that represents a single UCS-2 character in a valid range from 0 to 55 295. If a tag of this type is read with a value outside the valid range, the Web API returns the value **null**.
String	string	JSON string with a UTF-8 string If a tag of this type is read with a value outside the valid UTF-8 string (max. length 254 characters), the Web API returns the value **null**.
WString	wstring	JSON string with a UTF-8 string that represents a USC-2 string in a valid range from 0 to 55 295. If a tag of this type is read with a value outside the valid range (max. length 254 characters), the Web API outputs the value **null**.
Date	date	JSON unsigned integer in a range from 0 to 65 535 This value represents the number of days since 01.01.1990.
Time_Of_Day	time_of_day	JSON unsigned integer in a range from 0 to 4 294 967 295 This value represents the number of milliseconds since the beginning of the day.
LTime_Of_Day	ltime_of_day	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from 0 to 18 446 744 073 709 551 615 This value represents the number of nanoseconds since the beginning of the day.
Time	time	JSON signed integer in a range from -2 147 483 648 to 2 147 483 647) This value represents the number of milliseconds since a user-defined point in time.
LTime	ltime	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from -9 223 372 036 854 775 808 to 9 223 372 036 854 775 807 This value represents the number of nanoseconds since a user-defined point in time.
S5Time	s5time	JSON object with the keys **basis** and **value**: • The **basis** value is a JSON Unsigned Integer with a value of either 10, 100, 1000, or 10 000. The value **basis** represents the millisecond multiplier of the value **value**. • The value **value** is a JSON unsigned integer in the range from 0 to 999. The value **basis** multiplied by the value **value** gives the timer interval in milliseconds.
Date_And_Time	date_and_time	JSON object with the keys **year**, **month**, **date**, **hour**, **minute**, **second**, and **day_of_week**: • **year** is a JSON unsigned integer with a value in the range from 1 990 to 2 089 • **month** is a JSON unsigned integer with a value in the range 1 to 12 • **day** is a JSON unsigned integer with a value in the range 1 to 31 • **hour** is a JSON unsigned integer with a value in the range 0 to 23 • **minute** is a JSON unsigned integer with a value in the range 0 to 59 • **second** is a JSON floating point number with a value in the range 0 to 60 • **day_of_week** is a JSON string with a value of either **sun**, **mon**, **tue**, **wed**, **thu**, **fri**, or **sat**
LDT	ldt	JSON string with a numerical representation of an unsigned integer based on the number 10 in a range from 0 to 18 446 744 073 709 551 615 This value represents the number of nanoseconds since 01.01.1970 midnight (12:00:00.0 am).

Name of the data type in the TIA Portal	Name of the data type in Web API	Representation in the Web API
Struct	struct	Structured data type whose data structure can be determined using the PlcProgram.Browse method.
IEC_COUNTER	iec_counter	
IEC_TIMER	iec_timer	
DTL	dtl	
IEC_LTIMER	iec_ltimer	
IEC_SCOUNTER	iec_scounter	
IEC_DCOUNTER	iec_dcounter	
IEC_LCOUNTER	iec_lcounter	
IEC_UCOUNTER	iec_ucounter	
IEC_USCOUNTER	iec_uscounter	
IEC_UDCOUNTER	iec_udcounter	
IEC_ULCOUNTER	iec_ulcounter	
ERROR_STRUCT	error_struct	
NREF	nref	
CREF	cref	

Arrays

Arrays are displayed as JSON objects. The key is a string with a numeric representation of the index.
The following example shows the representation in the TIA Portal:



User data type_2

	Name	Data type	Default value	Accessible from HMI/OPC UA
1	▼ ExampleArray	Array[0..1, 10..12] of Byte		☑
2	ExampleArray[0,10]	Byte	1	☑
3	ExampleArray[0,11]	Byte	2	☑
4	ExampleArray[0,12]	Byte	3	☑
5	ExampleArray[1,10]	Byte	4	☑
6	ExampleArray[1,11]	Byte	5	☑
7	ExampleArray[1,12]	Byte	6	☑

Figure 5-3 Representation in the TIA Portal

Addressing the user data

The Web API supports the following addressing formats:

- Symbolic addressing of a tag in the tag table, e.g. Tag_1
- Symbolic addressing of a tag in a data block, e. g. "MyDB".Static_1

5.6.2 Parameter assignment of the block properties

Configuring access to the Web API in the TIA Portal

To restrict the read and write access to data blocks of your project, you can define the desired parameters in the attributes of the respective block.
To allow the Web API to access the data block, select the check box "DB accessible from Webserver".

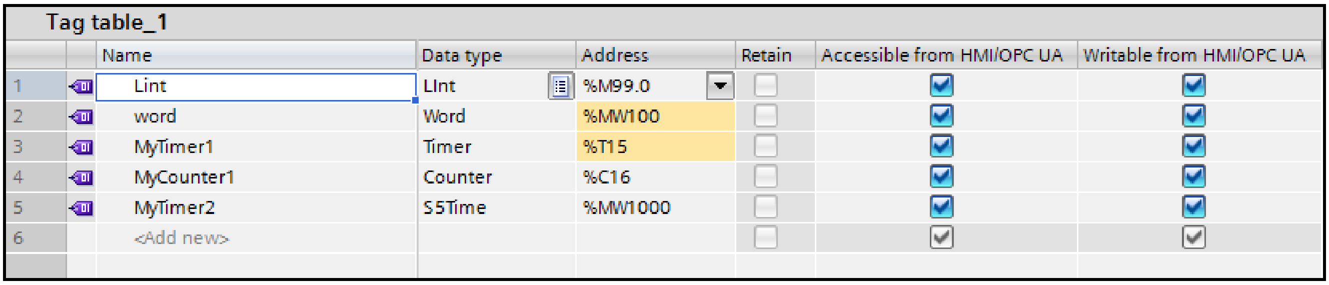
NOTE

F-blocks

Note that fail-safe blocks allow read access only. It is not possible to write tags into fail-safe blocks.

Parameter assignment for access to the Web API in tag tables

For read and write access of Web API to tags, the options "Accessible from HMI/OPC UA/Web API" and "Writable from HMI/OPC UA/Web API" must be enabled:



Tag table_1

	Name	Data type	Address	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA
1	LInt	LInt	%M99.0	□	☑	☑
2	word	Word	%MW100	□	☑	☑
3	MyTimer1	Timer	%T15	□	☑	☑
4	MyCounter1	Counter	%C16	□	☑	☑
5	MyTimer2	S5Time	%MW1000	□	☑	☑
6	<Add new>			□	☑	☑

Figure 5-4 Tag table in the TIA Portal

5.6.3 PlcProgram.Read

Use the PlcProgram.Read method to read a single variable from a CPU.
To call the PlcProgram.Read method, you need the "read_value" authorization.

Structure of the request

The following table informs you about the properties of the tag to be read.

Table 5-16 PlcProgram_Read_Request (object)

Name	Required	Data type	Description
var	yes	string	Name of the tag to be read.
mode	no, default is "simple"	string	Enumeration that determines the response format for this method: • "simple": Returns tag values according to the "simple" representation (see section Supported data types (Page 185)). • "raw": Returns tag values according to the "raw" representation (see section Supported data types (Page 185)).

Examples

In the following example, the user requests a global tag in the "simple" representation.

```
{
"var": "\"MotorSpeed\""
}
```

In the following example, the user requests a data block tag in the "raw" representation.

```
{
"var": "\"MyDB\".MyVariable",
"mode": "raw"
}
```

Response structure

If the request to the server was successful, the server returns JSON data values.

Examples

The following example shows the result of reading in a tag of type "int" in the "simple" representation.

```
-42
```

The following example shows the result of reading in a tag of type "dword" in the "raw" representation.

```
[ 1, 47, 233, 0 ]
```

Possible error messages

The following table shows the possible error messages of the PlcProgram.Read method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	The system does not have the necessary resources to read the requested address. Carry out the request again as soon as sufficient resources are available.
200	Address does not exist	The requested address does not exist or the Web server cannot access it.
201	Invalid address	The name structure of the symbolic address is not correct.
203	Invalid array index	The dimensions and limits of the array indexes do not correspond to the type information of the CPU.
204	Unsupported address	The data type of the address cannot be read.

NOTE

Tag access with the methods PlcProgram.Read, PlcProgram.Write, and PlcProgram.Browse

With these methods it is not yet possible to access all tags in firmware version V3.0.

There are selective restrictions when reading tags of technology objects. If access to specific tags is not possible, the API returns the message "unsupported".

5.6.4 PlcProgram.Write

Use the PlcProgram.Write method to write a single process tag to the CPU.
To call the PlcProgram.Write method, you need the "write_value" authorization.

Structure of the request

The following table informs you about the properties of the tag to be written.

Table 5-17 PlcProgram_Write_Request (object)

Name	Required	Data type	Description
var	yes	string	Name of the tag to be written.
value	yes	variant	of the value to be written; The value depends on the operating mode.
mode	no, default is "simple"	string	Enumeration that specifies the format of "value": • "simple": The user must specify the values according to the "simple" representation (see section Supported data types (Page 185)) • "raw": The user must specify the values according to the "raw" representation (see section Supported data types (Page 185))

Examples

In the following example, the user writes a global tag in the "simple" display.

```
{
"var": "\"MotorSpeed\"",
"value": 9001
}
```

In the following example, the user writes a tag to a data block in the "raw" representation.

```
{
"var": "\"MyDB\".MyVariable",
"value": [ 255, 77, 1, 99 ],
"mode": "raw"
}
```

In the following example, the user writes a string tag consisting of 10 characters to the "simple" representation:

```
{
"var": "\"MyDB\".MyString",
"value": "Short Str",
"mode": "simple"
}
```

In the following example, the user writes a string tag consisting of 10 characters with the text "Short Str" in the "raw" representation:

```
{
"var": "\"MyDB\".MyString",
"value": [ 10, 9, 83, 104, 111, 114, 116, 32, 83, 116, 114, 0 ],
"mode": "raw"
}
```

In the following example, the user writes a Wstring tag consisting of 6 characters in the "simple" representation:

```
{
"var": "\"MyDB\".MyWString",
"value": "Hello",
"mode": "simple"
}
```

In the following example, the user writes a string tag consisting of 6 characters with the text "Hello" in the representation "raw":

```
{
"var": "\"MyDB\".MyWString",
"value": [ 0, 6, 0, 5, 0, 72, 0, 101, 0, 108, 0, 108, 0, 111, 0, 0
],
"mode": "raw"
}
```

Response structure

If the write operation was successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the PlcProgram.Read method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	The system does not have the necessary resources to write the requested address. Carry out the request again as soon as sufficient resources are available.
200	Address does not exist	The requested address does not exist or the Web server cannot access the requested address.

Error code	Error message	Meaning
201	Invalid address	The name structure of the symbolic address is not correct.
203	Invalid array index	The dimensions and limits of the array indexes do not correspond to the type information of the CPU.
204	Unsupported address	The data type of the address cannot be written.

NOTE

Tag access with the methods PlcProgram.Read, PlcProgram.Write, and PlcProgram.Browse

With these methods it is not yet possible to access all tags in firmware version V3.0.

There are selective restrictions when reading tags of technology objects. If access to specific tags is not possible, the API returns the message "unsupported".

5.6.5 PlcProgram.DownloadProfilingData

The PlcProgram.DownloadProfilingData method supplies you with detailed runtime data for the user program in the CPU. The API method returns a ticket that you use to download the runtime data from the CPU. You can evaluate and graphically display the information later in order to analyze the program flow.

This information assists you in the following tasks:

- Runtime optimization of the user program
- Error diagnostics
- Evaluating the power reserve of the automation system
- Quality assurance of the application

To call the API method PlcProgram.DownloadProfilingData, you need "read_value" authorization.

Response structure

If successful, the method returns a string with a ticket ID. You can use this ticket ID to download the runtime data from the CPU.

You can find more information about the ticket mechanism in the section Ticket mechanism (Page 164).

Example

The following example shows a generated ticket ID for downloading the runtime data.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

File name

The file name for downloading the runtime data to a web browser is structured as follows:
`[project_name]_[module_name]_YYYY-MM-DD_HH-mm-ss_profiling.bin`
Example: `[1500_example01]_[plc_1]_2023-11-03_12-20-05_profiling.bin`
The file name is returned as an HTTP Content-Disposition header in the server response.

Structure and contents of the file

The file contains a list of entries in which the byte sequence of each entry is based on a specific entry type. The first byte of an entry in each case contains the entry type.

Table 5-18 Block events

Byte	0	1	2	3
0	Entry type	Block type	Block number	
4	Number of OB currently being executed in the program		Call hierarchy of the OB	Priority of the OB
8	Time stamp in 10^{-12} seconds			
12				

The entry types of the following table are supported.

Table 5-19 Entry type

Value	Entry type
0x00	Invalid
0x01	Block event: Start of a block
0x02	Block event: End of a block
0x10	Read process image partition
0x11	Write process image partition
0x20	Direct read access
0x21	Direct write access
0x30	Error ID
0x31	Communication load and cycle time

Entry type: Block events

The file is structured as follows:

The following table explains the meaning of the possible block types:

Table 5-20 Block type

Value	Block type
0x00	Invalid
0x01	Organization block (OB)

Value	Block type
0x02	Function (FC)
0x03	Function block (FB)
0x11	System function (SFC)
0x12	System function block (SFB)

The following table provides an example for block events:

Table 5-21 Example for block events

Byte	0	1	2	3
0	0x01	0x02	0x12	0x00
4	0x01	0x00	0x01	0x01
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x01: Start of a block
0x02: Block type = FC
0x0012: Block number = 18
0x0001: OB number = 1
0x01: Call hierarchy of the OB = 1
0x01: Priority of the OB = 1
0x0102030405060708: Time stamp

Entry type: Read/write process image partition

The file is structured as follows:

Table 5-22 Read/write process image partition

Byte	0	1	2	3
0	Entry type	-	Process image partition number	
4	-	-	-	Priority of the OB
8	Time stamp in 10^{-12} seconds			
12				

The following table provides an example for read process image partition:

Table 5-23 Example for read/write process image partition

Byte	0	1	2	3
0	0x10	-	0x01	0x00
4	-	-	-	0x01
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x10 Read process image partition
0x0001: Process image partition number = 1
0x01: Priority of the OB = 1
0x0102030405060708: Time stamp

Entry type: Direct read/write access

The file is structured as follows:

Table 5-24 Direct read/write access

Byte	0	1	2	3
0	Entry type	Event type	DB number	
4	-			Priority of the OB
8	Time stamp in 10^{-12} seconds			
12				

The following table explains the meaning of the possible entry types for direct I/O access:

Table 5-25 Event type

Value	Event type
0x00	Invalid
0x01	I/O
0x02	Motion Control
0x03	Diagnostics

The following table provides an example for direct reading:

Table 5-26 Example for direct reading

Byte	0	1	2	3
0	0x20	0x02	0x12	0x00
4	-	-	-	0x01
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x20: Direct reading
0x02: Motion Control = 2
0x0012: DB number = 18
0x01: Priority of the OB = 1
0x0102030405060708: Time stamp

Entry type: Error ID

The file is structured as follows:

Table 5-27 Entry in the diagnostics buffer

Byte	0	1	2	3
0	Entry type	-	Error type	
4	-	-	-	-
8	Time stamp in 10^{-12} seconds			
12				

The following table provides an example for a diagnostics buffer entry:

Table 5-28 Example for diagnostics buffer entry

Byte	0	1	2	3
0	0x30	-	0x42	0x29
4	-	-	-	-
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x30: Entry in the diagnostics buffer
0x2942: Error type = I/O read access error
0x0102030405060708: Time stamp

Entry type: Communication load and cycle time

The file is structured as follows:

Table 5-29 Entry in the diagnostics buffer

Byte	0	1	2	3
0	Entry type	-	Communication load in %	
4	Last cycle time in µs			
8	Time stamp in 10^{-12} seconds			
12				

The following table provides an example for the communication load:

Table 5-30 Example for communication load

Byte	0	1	2	3
0	0x31	-	0x12	0x00
4	0x03	0x02	0x01	0x00
8	0x08	0x07	0x06	0x05
12	0x04	0x03	0x02	0x01

0x31: Communication load and cycle time
0x0012: Communication load = 18%
0x00010203: Last cycle time = 66051µs
0x0102030405060708: Time stamp

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.

Further processing of runtime data

With the "SIMATIC S7-1500 Profiling" library (https://support.industry.siemens.com/cs/at/de/view/109750245/en), Siemens offers a comprehensive analysis tool for analyzing the runtime behavior of your user programs. All relevant information is displayed graphically using a web-based visualization. For further analysis purposes, you can output the recorded data as a CSV file and evaluate it in a spreadsheet.

You can also find examples of using API methods in practice on Github under the following repository (https://github.com/siemens/simatic-s7-webserver-api).

5.6.6 PlcProgram.Browse

The PlcProgram.Browse method allows you to search for tags and the corresponding metadata according to your individual requirements.
To call the PlcProgram.Browse method, you need the "read_value" authorization.

Structure of the request

The following table informs you about the properties of the tag to be searched.

Table 5-31 PlcProgram_Browse_Request (object)

Name	Required	Data type	Description
var	Yes/no, see "Description" column	string	Name of the tag to be searched. If this attribute is present, it cannot be an empty string. • If "mode" = "var", then this attribute is required. The Browse method searches for the provided tag to retrieve the metadata of the tag. • If "mode" = "children", this attribute is optional. The Browse method searches for the tag and returns a list of child tags and metadata.
mode	Yes	string	Enumeration that determines the behavior of this method: • "var": Displays information about the specified tag. • "children": Outputs information about the immediate descendants (children) of the specified tags.
type	No	array of string	Possible array entries are: • "code_blocks": Reads all code blocks • "data_blocks": Reads all data blocks • "tags": Removes all tags If no "type" parameter is selected for compatibility reasons, only DBs and tags are returned.

Example 1

In the following example the user searches the root node ("root") of the CPU.

```
{
"mode": "children"
}
```

The following example shows a possible response from the server.

```
[
{
"name": "TestDB",
"has_children": true,
"db_number": 2,
"datatype": "datablock"
},
{
"name": "GenUsrMsg_Ret",
"address": "%MW43",
"area": "M",
"datatype": "int"
}
]
```

Example 2

In the following example, the user searches the descendants (children) of a data block.

```
{
"var": "\"MyDB\"",
"mode": "children"
}
```

The following example shows a possible response from the server.

```
[
{
"name": "Static_1",
"db_number": 1,
"datatype": "int"
},
{
"name": "Static_2",
"db_number": 1,
"datatype": "int"
}
]
```

Example 3

In the following example, the user requests information about a specific tag.

```
{
"var": "\"MyDB\".MyStruct.MyField",
"mode": "var"
}
```

The following example shows a possible response from the server.

```
[
{
"name": "MyDateTimeValue",
"db_number": 2,
"datatype": "date_and_time",
"array_dimensions": [
{
"start_index": 0,
"count": 3
}
]
}
]
```

Example 4

In the following example, the user searches code blocks of a CPU.

```
{
"mode": "children",
"type": ["code_blocks"]
}
```

The following example shows a possible response from the server.

```
[
{

"name": "MainOB",

"block_number": 1,

"block_type": "ob"

},
{
"name": "MotorControlConveyor",
"block_number": 23,
"block_type": "fb"
}
{
"name": "CREATE_DB",
"block_number": 86,
"block_type": "sfc",

}
{
…
}
]
```

Example 5

In the following example, the user searches the data blocks of a CPU.

```
{
"mode": "children",
"type": ["data_blocks"]
}
```

The following example shows a possible response from the server.

```
[
{
"name": "TestDB",
"block_number": 2,
"datatype": "datablock"
},
{
"name": "MotorControlConveyorDB",
"block_number": 23,
"datatype": "datablock"
}
]
```

Example 6

The following example shows the result of searching the root node "root" of the CPU when the "type" parameter and all 3 possible array entries "data_blocks", "code_blocks" and "tags" are selected:

```
{
"mode": "children",
"type": ["data_blocks, "code_blocks", "tags"]
}
```

The following example shows a possible response from the server.

```
[
{
"name": "TestDB",
"has_children": true,
"db_number": 2,
"datatype": "datablock"
},
{
"name": "GenUsrMsg_Ret",
"address": "%MW43",
"area": "M",
"datatype": "int",
}
{
"name": "MainOB",
"block_number": 1,
"block_type": "ob"
}
]
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-32 PlcProgram_Browse_Response (array of objects)

Name	Required	Data type	Description
name	Yes	string	Tag name; can be used as a string identifier for the field.
address	No	string	Address of the tag in STEP 7; only applicable for the tags in the ranges M, I, Q, timer and counter and tags in non-optimized data blocks. The representation corresponds to the addresses in the watch tables in the TIA Portal.
read_only	No	bool	Query whether the tag is read-only. The only valid value is "true". If the tag is to be written, this attribute does not appear.
has_children	No	bool	Query whether the tag is a structured tag with child tags. The only valid value is "true". If the tags are an unstructured data type, this attribute is not displayed.
db_number	No	number	Numerical data block identifier; appears when "datatype" = "datablock" and for each child element of a data block (with corresponding data block number).
area	No	string	Letter which defines the range (M/I/Q/timer/counter) of the tag. If the tag is not in one of these ranges, the attribute does not appear.
datatype	Yes	string	Data type of the tag For data blocks, this is the "datablock" data type; for tags, see the table in the section Supported data types (Page 185). If the data type is not supported, the data type is "unsupported".
max_length	No	number	If the data type is a "string" or "wstring", this value is the maximum number of characters allowed in the tag.
array_dimensions	No	PlcProgram_Browse_Response_ArrayData	Object arrays arranged from the most significant to the least significant array dimension. The attribute is only displayed if the tag is an array.
block_number	No	number	Number of the logic block

Name	Required	Data type	Description
block_type	No	string	The type of the logic block: • "ob" • "fc" • "fb" • "sfc" • "sfb"

Table 5-33 PlcProgram_Browse_Response_ArrayData (array of objects)

Name	Required	Data type	Description
start_index	Yes	integer	Start index for this array dimension, as specified in the TIA Portal project.
count	Yes	integer	Number of elements in this array dimension

Possible error messages

The following table shows possible error messages of the PlcProgram.Browse method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
3	System is busy	The desired operation cannot be performed because the system is currently performing a different request. Restart the query as soon as the current operation is complete.
4	No resources	The system does not have the required resources to retrieve the type information. Perform the request again as soon as enough resources are available again.
200	Address does not exist	The requested address does not exist or the Web server cannot access the requested address.
201	Invalid address	The name structure of the symbolic address is not correct.
202	Variable is not a structure	It is not possible to search the specific address because the tag is not a structured data type.
203	Invalid array index	The dimensions and limits of the array indexes do not correspond to the type information of the CPU.

NOTE

Tag access with the methods PlcProgram.Read, PlcProgram.Write, and PlcProgram.Browse

It is not yet possible to access all tags with these methods in firmware version ≥ V3.0.

There are selective restrictions when reading tags of technology objects. If access to specific tags is not possible, the API returns the message "unsupported".

5.7 Reading and changing the operating mode

5.7.1 Plc.ReadOperatingMode

With the Plc.ReadOperatingMode method you can read the operating mode of the CPU.
To call the Plc.ReadOperatingMode method, you need the "read_diagnostics" authorization.

Response structure

If the request to the server was successful, the server returns the operating mode as a string. The following strings are possible for operating modes:

String	Operating mode
`stop`	STOP
`startup`	STARTUP
`run`	RUN
`hold`	HOLD
`unknown`	-

Possible error messages

The following table shows the possible error messages of the Plc.ReadOperatingMode method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.

5.7.2 Plc.RequestChangeOperatingMode

With the Plc.RequestChangeOperatingMode method, you request a new operating mode for the CPU.

Note that this is only a request for an operating mode. The conditions for an operating mode change must be given at the CPU, e.g. by the corresponding position of the mode selector. You can use the Plc.ReadOperatingMode (Page 206) method to check whether the operating mode change on the CPU was successful.

To call the Plc.RequestChangeOperatingMode method, you need the "change_operating_mode" authorization.

Structure of the request

The following table informs you about the necessary parameters for the request.

Table 5-34 Plc_RequestChangeOperatingMode_Request (object)

Name	Required	Data type	Description
mode	yes	string	Requested operating mode: "stop" STOP mode "run" RUN mode

Example

In the following example, the RUN mode is requested.

```
{
"mode": "run"
}
```

Response structure

If the request to the server was successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the Plc.RequestChangeOperatingMode method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.

5.7.3 Plc.ReadModeSelectorState

The Plc.ReadModeSelectorState method reads the current position of the mode switch to the CPU.
To call the Plc.ReadModeSelectorState method, you need "read_diagnostics" authorization.

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-35 Plc_ReadModeSelectorState_Response (object)

Name	Required	Data type	Description
redundancy_id	Yes for R/H-CPUs No for all other CPUs	number	The parameter redundancy ID must be available if the request is performed on an R/H-CPU. The redundancy ID has the value 1 or 2. For all other CPUs, the parameter does not have to be part of the request.
mode_selector	Yes	string	Possible values: • "run" • "stop" • "no_switch" • "unknown"

Example

In the following example, RUN mode is read.

```
{
"mode_selector": "run"
}
```

Possible error messages

The following table shows possible error messages with the Plc.ReadModeSelectorState method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.

5.8 Change time settings via Web API

5.8.1 Plc.ReadSystemTime

This API method provides the system time of the CPU. If you have synchronized the system time of the CPU, for example via the TIA Portal function "Online & diagnostics", the system time corresponds to Coordinated Universal Time (UCT).

Response structure

The following table shows you the structure of server responses to successful requests.

Table 5-36 Plc_ReadSystemTime_Response (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO8601 time stamp as string in nanoseconds

Example

The following example shows the structure of the time stamp.

```
{
"timestamp": "2012-04-23T18:25:43.123456789Z"
}
```

5.8.2 Plc.SetSystemTime

Use this API method to set the system time of the CPU (PLC local time).
To call the Plc.SetSystemTime method, you need the "change_time_settings" permission.

Structure of the request

The following table contains information about the parameters of the request:

Table 5-37 Plc_SetSystemTime_Request (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO 8601 timestamp as a string in nanoseconds; represents the time stamp of the system time to be set

Example

The following example shows the structure of the time stamp.

```
{
"timestamp": "2023-11-05T18:25:43.515154511Z"
}
```

Response structure

The server always returns the Boolean value "true" for successful requests.

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.
900	Invalid timestamp	The time stamp used does not match the required time-stamp format (ISO time-stamp defaults).
901	Time not within allowed time range	The time stamp is not within the allowed period for time stamps. The end of the possible timespan is 2200-12-31T23:59:59.999999999Z

5.8.3 Plc.ReadTimeSettings

This API method returns the currently active time, the deviation of the time zone from Coordinated Universal Time (UCT), and any daylight saving time rules.

Structure of the request

The following tables contain information about the individual parameters of the request.

Table 5-38 Plc_ReadTimeSettings_Response (object)

Name	Required	Data type	Description
current_offset	Yes	string the time range, shown in accordance with ISO 8601	The currently active deviation of the PLC local time from the coordinated universal time (UTC) configured on the CPU in minutes, e.g. "PT1H30M" for a deviation of 1 hour and 30 minutes.
utc_offset	Yes	string the time range, shown in accordance with ISO 8601	The deviation of the time zone from the coordinated universal time (UTC) in minutes without consideration of the daylight saving time rules.
rule	No	object of type Plc_ ReadTimeSettings_ Rule_Response	Displays the daylight saving time settings. If no settings are active, daylight saving time is not set and only the "utc_offset" value is used to calculate the PLC local time.

Table 5-39 Plc_ReadTimeSettings_Rule_ Response (object)

Name	Required	Data type	Description
std	Yes	object of type Plc_ ReadTimeSettings_ StdRule_Response	Stands for the zone time setting.
dst	Yes	object type Plc_ ReadTimeSettings_ DstRule_Response	Stands for the daylight saving time settings

Table 5-40 Plc_ReadTimeSettings_StdRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_ ReadTimeSettings_ Start_Response	Stands for the start time of the zone time.

Table 5-41 Plc_ReadTimeSettings_DstRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_ ReadTimeSettings_ Start_Response	Stands for the start time of daylight saving time.
offset	Yes	string the time range, shown in accordance with ISO 8601	Stands for the deviation of daylight saving time from the zone time in minutes (only used in the dst object).

Table 5-42 Plc_ReadTimeSettings_Start_Response (object)

Name	Required	Data type	Description
month	Yes	number	Used in the "dst.start/std.start" objects. Stands for the month in which the time starts.
week	Yes	number	Used in the "dst.start/std.start" objects. Stands for the week in which the time change is performed. Value 1: First occurrence of the day of the week in the month Value 5: Last occurrence of the day of the week in the month
day_of_week	Yes	string	Used in the "dst.start/std.start" objects. Describes the day of the week on which the time change is performed as a string with 3 characters. The following values are possible: • sun • mon • tue • wed • thu • fri • sat
hour	Yes	number	Used in the "dst.start/std.start" objects. Describes the hour in which the time change is performed.
minute	Yes	number	Used in the "dst.start/std.start" objects. Describes the minute in which the time change is performed.

Example 1

A daylight saving time rule is configured in the following example.

```
{
"current_offset": "PT1H",
"utc_offset": "PT2H",
"rule":
{
"dst":
{
"offset": "PT1H",
"start":
{
"month": 4,
"week": 4,
"day_of_week": "sun",
"hour": 3,
"minute": 0
}
},
"std":
{
```

```
"start":
{
"month": 10,
"week": 5,
"day_of_week": "sun",
"hour": 2,
"minute": 0
}
}
}
}
```

Example 2

In the following example, no daylight saving time rule has been configured and a time offset of 1 hour exists.

```
{
"current_offset": "PT1H",
"utc_offset": "PT2H",
}
```

5.8.4 Plc.SetTimeSettings

With this API method you can set the time settings of the CPU.
To call the Plc.SetTimeSettings method, you need the "change_time_settings" permission.

Structure of the request

The following tables contain information about the individual parameters of the request.

Table 5-43 Plc_SetTimeSettings_Response (object)

Name	Required	Data type	Description
utc_offset	Yes	string	ISO 8601 timespan as a string; the deviation of the time zone from the coordinated universal time (UTC) without consideration of the daylight saving time rules.
rule	No	object of type Plc_SetTimeSettings_Rule_Response	Displays the daylight saving time settings. If no settings are active, daylight saving time is not set and only the "utc_offset" value is used to calculate the local time.

Table 5-44 Plc_SetTimeSettings_Rule_Response (object)

Name	Required	Data type	Description
std	Yes	object of type Plc_ SetTimeSettings_ StdRule_Response	Stands for the zone time setting.
dst	Yes	object of type Plc_ SetTimeSettings_ DstRule_Response	Stands for the daylight saving time settings

Table 5-45 Plc_SetTimeSettings_StdRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_ SetTimeSettings_ Start_Response	Stands for the start time of the zone time.

Table 5-46 Plc_SetTimeSettings_DstRule_Response (object)

Name	Required	Data type	Description
start	Yes	object of type Plc_ SetTimeSettings_ Start_Response	Stands for the start time of daylight saving time.
offset	Yes	string	ISO 8601 timespan as a string; the deviation of daylight saving time from the zone time in minutes (only used in the dst object).

Table 5-47 Plc_SetTimeSettings_Start_Response (object)

Name	Required	Data type	Description
month	Yes	number	Used in the "dst.start/std.start" objects. Stands for the month in which the time starts.
week	Yes	number	Used in the "dst.start/std.start" objects. Stands for the week in which the time change is performed Value 1: First occurrence of the day of the week in the month Value 5: Last occurrence of the day of the week in the month
day_of_week	Yes	string	Used in the "dst.start/std.start" objects. Describes the day of the week on which the time change is performed as a string with 3 characters. The following values are possible: • sun • mon • tue • wed • thu • fri • sat
hour	Yes	number	Used in the "dst.start/std.start" objects. Describes the hour in which the time change is performed.
minute	Yes	number	Used in the "dst.start/std.start" objects. Describes the minute in which the time change is performed.

Example 1

The following example shows how to set a rule with daylight saving time parameters.

```
{
"utc_offset": "PT2H",
"rule":
{
"dst":
{
"offset": "PT1H",
"start":
{
"month": 4,
"week": 4,
"day_of_week": "wed",
"hour": 3,
"minute": 0
}
},
"std":
{
"start":
{
"month": 10,
"week": 5,
"day_of_week": "wed",
"hour": 2,
"minute": 0
}
}
}
}
```

Example 2

The following example shows the setting of a rule without daylight saving time parameters.

```
{
"utc_offset": "PT6H",
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read-only	The system cannot be written to (SIMATIC Memory Card is read-only). Changes are currently not permitted.
902	Invalid time rule	The rule provided is invalid. Check that the time settings are correct.
903	Invalid UTC offset	The Coordinated Universal Time (UTC) deviation provided is invalid. Check that the time settings are correct.

5.9 Reading diagnostics and service data

5.9.1 Project.ReadLanguages

This API method returns a list with the project languages available on the CPU.

You can then use the "Alarms.Browse" or "DiagnosticBuffer.Browse" API methods in one of the available languages to get alarm messages or diagnostic messages in the available languages.

To call the Project.ReadLanguages method, you need the "read_diagnostics" permission.

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-48 Project_ReadLanguages_Response (Array of objects)

Name	Required	Data type	Description
languages	Yes	array of Project_ReadLanguages_Entry_Response	Object array, where each object represents a project language.

Table 5-49 Project_ReadLanguages_Entry_Response (object)

Name	Required	Data type	Description
language	Yes	string	String with the project language.

Example

The following example shows the output of a project language as a string.

```
{
  "languages":
  [
   {
     "language": "en-US"
}
  ]
}
```

NOTE

No project language configured

If no project language was configured, the object array is empty.

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

5.9.2 Alarms.Browse

With this method you can determine which alarms are currently active on the CPU, and when the last change occurred within the diagnostics buffer.

To call the Alarms.Browse method, you need the "read_diagnostics" permission.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-50 Alarms_Browse_Request (object)

Name	Required	Data type	Description
language	Yes	string	The desired language in which the text is returned in RFC 4647 format, e.g. "en-US"
count	No	number	The maximum number of alarm entries that are returned. The default value is 50. If you want to determine the current status of the diagnostics buffer, enter 0 as "count".
alarm_id	No	string	The alarm ID of the CPU for which you are requesting data. If the Alarm ID is included, only the "count" parameter can be offered as a filter.
filters	No	object of type Alarms_Browse_Filters_Request	Optional object containing parameters for filtering the response

Table 5-51 Alarms_Browse_Filters_Request (object)

Name	Required	Data type	Description
mode	Yes	string	The mode that determines whether attributes are to be included or excluded in the response. The following modes are available: • include • exclude
filters.attributes	Yes	array of strings	Possible array entries are: • "alarm_text" • "info_text" • "status" • "timestamp" • "acknowledgement" • "alarm_number" • "producer"

Examples

The following example shows a request for reading a single alarm with all alarm areas in the English language:

```
{
"language": "en-US",
"alarm_id": "1231231231"
}
```

The following example shows the request for reading a single alarm without the alarm areas excluded under "exclude":

```
{
"language": "en-US",
"alarm_id": "1231231231",
"filters":
{
"mode": "exclude",
"attributes": ["alarm_text", "info_text"],
}
}
```

The following example shows the request for reading 50 alarms with the alarm ranges included in "include".

```
{
"language": "en-US",
"count": 50,
"filters":
```

```
{
"mode": "include",
"attributes": ["status", "acknowledgement"]
}
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-52 Alarms_Browse_Response (object)

Name	Required	Data type	Description
language	No, optional and only available when entries are returned; not relevant if, for example, count = 0 is requested	string	The language of the response in which the message text is output. If no valid language was requested, the server outputs the message "invalid".
last_modified	Yes	string	ISO 8601 time stamp in UTC as string; The time stamp with the last change of the alarm system since you made the last read request. The time stamp allows you to see when the last alarm change occurred in the system without having to check individual alarms in detail.
count_current	Yes	number	The number of active alarms.
count_max	Yes	number	The maximum number of active alarms.
entries	No	array of objects of Alarm_Browse_Entry_response	The list of pending alarms, where each alarm entry is represented as an object.

Table 5-53 Alarms_Browse_Entry_Response (object)

Name	Required	Data type	Description
id	Yes	string	Alarm ID.
alarm_number	Yes, if not excluded by user	number	Alarm number.
status	Yes, if not excluded by user	string	Alarm status; is either "incoming" or "outgoing".
timestamp	Yes, if not excluded by user	string	ISO 8601 time stamp as a string; Time stamp in Coordinated Universal Time (UTC) of the time at which the alarm assumed the status "incoming" or "outgoing".
producer	Yes, if not excluded by user	string	Possible alarm trigger:

Name	Required	Data type	Description
			• "program_alarm" • "system_diagnostics" • "motion" • "security" • "sinumerik" • "graph7" • "prodiag" • "other"
hwid	No	number	Contains the hardware ID if the alarm producer is "system_diagnostics".
acknowledgement	No	object	Appears when an alarm needs to be acknowledged.
acknowledgement.state	Yes	string	String in readable form, which provides information about the status of the acknowledgement: • "not_acknowledged" • "acknowledged"
acknowledgement.timestamp	Yes	string	ISO 8601 time stamp as a string; If the current status ("incoming" or "outgoing") has been acknowledged, the time stamp provides information about the time of acknowledgement.
alarm_text	Yes, if not excluded by user	string	Alarm text in the selected language.
info_text	Yes, if not excluded by user	string	Info text in the selected language.
text_inconsistent	No	boolean	If the alarm text or info text is inconsistent, this flag returns "true".

Examples

The following example shows the response to a request with a non-acknowledged alarm and all attributes (no filters set).

```
{
"entries":
[
{
"id": "121651651651",
"timestamp": "2023-11-05T18:25:43.511987654Z",
"status": "incoming",
"alarm_number": 37,
"producer": "system_diagnostics",
"hwid": 49,
"acknowledgement":
{
"state": "not_acknowledged"
},
"alarm_text": "CPU maintenance demanded: Emergency IP suite
parameter for IE interface activated PLC_1516 / Current CPU
operating mode: STOP",
"info_text": "Short name: CPU general Order number: 6ES7
516-3AP03-0AB0",
"text_inconsistent": false
}
],
"last_modified": "2023-11-05T18:25:43.511546151Z",
"count_current": 1,
"count_max": 5000,
"language": "en-US"
}
```

The following example shows the response to a request with an acknowledged alarm and all attributes (no filters set):

```
{
"entries":
[
{
"id": "121651651651",
"timestamp": "2023-11-05T18:25:43.511987654Z",
"status": "incoming",
"alarm_number": 35,
"producer": "system_diagnostics",
"hwid": 49,
"acknowledgement":
{
"state": "acknowledged",
"timestamp": "2023-11-05T18:25:50.123456789Z"
},
"alarm_text": "CPU status message: CPU not in RUN Current CPU
operating mode: STOP",
"info_text": "Short name: CPU general Order number: 6ES7
516-3AP03-0AB0",
"text_inconsistent": false
}
],
"last_modified": "2023-11-05T18:25:43.511546151Z",
"count_current": 1,
"count_max": 5000,
"language": "en-US"
}
```

The following example shows the filtered response to a query with only alarm IDs:

```
{
"entries":
[
{
"id": "121651651651",
"producer": "system_diagnostics"
}
{
"id": "121651651652"
"producer": "system_diagnostics",
}
{
"id": "121651651653"
"producer": "program_alarm",
}
],
"last_modified": "2023-11-05T18:25:43.511546151Z",
"count_current": 3,
"count_max": 5000,
"language": "en-US"
}
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
800	Invalid alarm ID	The alarm ID provided is invalid.
801	Invalid parameters	The request is invalid because provided parameters are invalid (e.g. the parameters "count" and "id" are present at the same time).

5.9.3 Alarms.Acknowledge

Use this method to acknowledge individual alarms.

To call the Alarms.Acknowledge method, you need the "acknowledge_alarms" permission.

Structure of the request

The following table contains information about the parameters of the request:

Table 5-54 Alarms_Acknowledge_Request (object)

Name	Required	Data type	Description
id	Yes	string	The acknowledgement ID of the alarm to be acknowledged. The acknowledgement ID can be found in the alarm object returned by the Alarm.Read method.

Examples

The following example shows the ID of an alarm to be acknowledged:

```
{
"id": "9979413824317259784"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log in with a user account that has sufficient authorizations to call this method.

5.9.4 Syslog.Browse

The Syslog.Browse method reads the entries of the CPU's internal syslog buffer.
To call the Syslog.Browse method, you need "read_syslog" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-55 Plc_ReadModeSelectorState_Request (object)

Name	Required	Data type	Description
redundancy_id	Yes for R/H-CPUs No for all other CPUs	number	The parameter redundancy ID must be available if the request is performed on an R/H-CPU. The redundancy ID has the value 1 or 2. For all other CPUs, the parameter does not have to be part of the request.
count	No	number	The maximum number of syslog entries is requested. The default and maximum value is 20. If the value is 0, all syslog entries are omitted from the response. Only the attributes last_modified, count_total and count_lost are returned.
first	No	number	"first" specifies the latest entry to be read. When count > 1 is set, the values are read backward. For example, if you request the value 42 with count 3, the entries 42, 41 and 40 are returned. When the value 0 is set, the latest <count> elements are read.

Example 1

In the following example, the user requests up to 20 syslog entries, beginning with the latest entries.

```
{
"count": 20
}
```

Example 2

In the following example, the user requests up to 20 syslog entries, beginning with entry 853.

```
{
"count": 20,
"first": 853
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-56 Syslog_Browse_Response (object)

Name	Required	Data type	Description
entries	Yes	array of Syslog_Browse_ Entry_Request	Contains an array of objects where each object is a single syslog entry of the CPU's internal syslog buffer.
count_total	Yes	number	This attribute contains the overall number of the entries in the syslog buffer since startup of the CPU.
count_lost	Yes	number	This attribute contains the number of entries of the syslog buffer which were lost. This is the number of the entries which were not overwritten through new entries and not stored on the syslog server.

Table 5-57 Syslog_Browse_Entry_Response (object)

Name	Required	Data type	Description
raw	Yes	string	An unformatted syslog entry according to RFC 3164.

Example 1

In the following example, an individual syslog entry is returned.

```
{
"entries":
[
{
"<35>1 2023-08-24T22:22:50.468Z 192.168.0.1 Webserver - ID6
[device@4329.6.100.1.1500 devVendor="Siemens" devProduct="CPU1500
(PLCSIM)"
FWVersion="T31.16.20"][session@4329.6.100.1.1500 protocolType="HTTPS
" userName="Anonymous" src="192.168.0.99"
sessionID="1"][function@4329.6.100.1.1500 fct="login"
result="success"] SE_DEFAULT_USER_AUTHENTICATION_USED"
}
],
"count_total": 546875456,
"count_lost": 0
}
```

Example 2

In the following example, an empty syslog response is returned, if previously no syslog message was available in the syslog buffer.

```
{

"entries": [],
"count_total": 0,
"count_lost": 0
}
```

Example 3

In the following example, an empty syslog response is returned, if count = 0 is requested. This query can be used to request the ID of the last syslog entry without data to detect if new entries are available.

```
{

"entries": [],
"count_total": 546875456,
"count_lost": 0
}
```

Possible error messages

The following table shows possible error messages with the Plc.ReadModeSelectorState method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	The system does not have the resources required to read the requested address. Perform the request again as soon as enough resources are available again.
7	Partner not accessible	The data of a partner CPU of an R/H system is not available. This can be the case when the CPU is not in RUN-Redundant system state.

5.9.5 DiagnosticBuffer.Browse

With this method you read out entries from the diagnostics buffer of the CPU.
To call the DiagnosticBuffer.Browse method, you need the "read_diagnostics" permission.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-58 DiagnosticBuffer_Browse_Request (object)

Name	Required	Data type	Description
language	Yes	string	The desired language in which the text is returned in RFC 4647 format, e.g. "en-US"
count	No	number	The maximum number of alarm entries that are returned. The default value is 50. If you want to determine the current status of the diagnostics buffer, enter 0 as "count".
filters	No	object of type DiagnosticBuffer_ Browse_Filters_ Request	The object that represents the different filtering options.

Table 5-59 DiagnosticBuffer_Browse_Filters_Request (object)

Name	Required	Data type	Description
attributes	Yes	array of strings	The following attributes are possible for the diagnostics buffer entries: • short_text • long_text • help_text
mode	Yes	string	The mode that determines whether attributes are to be included or excluded in the request. The following modes are available: • include • exclude

Example

The following example shows a request of the diagnostic entries as LCID value 1033 (dec value), which stands for "English – United States".

```
{
"language": "en-US",
"count": 50,
"entries":
{
"mode": "include",
"attributes": ["short_text", "long_text", "help_text"]
}
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-60 DiagnosticBuffer_Browse_Response (object)

Name	Required	Data type	Description
entries	No	array of objects of DiagnosticBuffer_Browse_Entry_Response	Array of diagnostics buffer entries, where each object represents one diagnostics buffer entry.
last_modified	Yes	string	ISO 8601 time stamp as a string; time stamp of the last change in the diagnostics buffer.
count_current	Yes	number	Number of available diagnostics buffer entries.
count_max	Yes	number	Maximum number of possible diagnostics buffer entries
language	No, optional and only available when entries are returned; not relevant if, for example, count = 0 is requested	string	The language in which the response is output.

Table 5-61 DiagnosticBuffer_Browse_Entry_Response (object)

Name	Required	Data type	Description
timestamp	Yes	string	ISO 8601 time stamp as a string; The attribute is provided in Coordinated Universal Time (UTC) and not in PLC local time.
status	Yes	string	The status parameter for events is either "incoming" (incoming events) or "outgoing" (outgoing events).
event	Yes	object	Contains the event ID of the diagnostics buffer which consists of the text list ID and the text ID of the event
event.textlist_id	Yes	number	Text list ID of the event

Name	Required	Data type	Description
event.text_id	Yes	number	Text ID of the event
long_text	Yes, if not excluded by the user	string	Diagnostics buffer entries in the long form
short_text	Yes, if not excluded by the user	string	Diagnostics buffer entries in the short form
help_text	Yes, if not excluded by the user	string	Help text for an incoming event.

Example

The following example shows the representation of a single entry in the diagnostics buffer (system diagnostics message).

```
{
"entries":
[
{
"timestamp": "2023-11-05T18:25:43.511854547Z",
"status": "outgoing",
"long_text": "CPU info: Boot up
memory card type: Program card (external load memory)
CPU changes from OFF to STOP (initialization) mode

PLC_2 / PLC_2",
"short_text": "Boot up - CPU changes from OFF to STOP
(initialization) mode",
"help_text": "",
"event"
{
"textlist_id": 2,
"text_id": 16385
}
}
],
"last_modified": "2023-11-05T18:25:43.514678521Z",
"count_current": 1234,
"count_max": 3200,
"language": "en-US"
}
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.

5.9.6 Modules.DownloadServiceData

The API method Modules.DownloadServiceData returns a ticket that you use to download service data from the CPU. You can then forward the service data to Customer Support, for example for an analysis of your production data in the event of an error.

To call the API method Modules.DownloadServiceData, you need "download_service_data" authorization.

You can find more information about the ticket mechanism in the section Ticket mechanism (Page 164).

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-62 Modules_DownloadServiceData_Request (object)

Name	Required	Data type	Description
hwid	Yes	number	Hardware ID of the module whose service data you want to read out.

Response structure

If successful, the method returns a string with a ticket ID. Use this ticket ID to download the service data.

NOTE

A maximum of only one Modules.DownloadServiceData ticket can be created for all users. A new ticket can be created for this method only after this ticket has been closed.

Example

The following example shows a generated ticket ID for service data download.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.
7	Partner not accessible	The data of the CPU of an R/H system is not available. This can be the case when the system is in SYNCUP operating state or RUN-Redundant system state or when the service data of the partner CPU is queried.
600	No service data resources	Only one ticket resource for service data is available for all users at the same time.
1100	Invalid hardware identifier	The specified hardware ID is not valid for the current request. Make sure that you have used the correct hardware ID.

5.10 Backing up and restoring the configuration

5.10.1 Plc.CreateBackup

With this API method, you request a ticket to create a backup file of the CPU configuration.
To call the Plc.CreateBackup method, you need the "backup_plc" permission.

Response structure

The method returns a string with a ticket ID for creating a backup file.

Example

The following example shows a generated ticket ID for creating a backup file.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 164) section.

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.
1000	Backup creation in progress	The creation of a backup file is in progress.
1001	Backup restoration in progress	The restoration of a saved configuration is currently being carried out. It is not possible to perform both operations at the same time.
1004	PLC not in STOP	A backup file can only be created when the CPU is in STOP mode. Set the CPU to STOP mode and execute the request again.

Overview of creating a backup file

Perform the steps below to create a backup file:

1. Authenticate yourself with the API method Api.Login.
2. Request a ticket for creating a backup file using the Plc.CreateBackup method.
   If you are authorized to call this method and the CPU is in STOP, the CPU creates a ticket. Once a ticket has been created for the creation of a backup file, it is no longer possible to switch to the RUN operating state. This ensures the consistency of the backup file.
3. Use the ticket end point to start downloading the backup file.
   The CPU informs you about the current status of the generation in the additional ticket information. Additional information is available using the Api.BrowseTickets(id) method.
4. After the download has been completed successfully, the CPU sets the ticket to the "completed" status.
5. Close the ticket using the Api.CloseTicket(id) method.
6. Now you can set the CPU back to RUN mode.

Format of the backup file name

The default file name of the backup file, which is returned by the HTTP content disposition header, contains the following information:

- Time stamp in Coordinated Universal Time (UTC)
- Module name
- Name of the TIA Portal project
- F-collective signature (for F-CPUs)

5.10.2 Plc.RestoreBackup

Use this API method to request a ticket that restores the configuration of a CPU via a backup file.
To call the Plc.RestoreBackup method, you need the "restore_plc" permission.

Structure of the request

The following table contains information about the parameters of the request:

Table 5-63 Plc_RestoreBackup_Request (object)

Name	Required	Data type	Description
password	Yes	string	The required password for the logged on user An empty password string is transferred for the "Everybody" or "Anonymous" user. The "passwort" attribute is therefore always required. The password must be the password of the user who previously authenticated via Api.Login and whose session token was used to call the Plc.RestoreBackup method.

Example

The following example shows how to enter a password.

```
{
"password": "SecurePassword"
}
```

Response structure

The method returns a string. The string contains a ticket ID that you can use to restore the configuration on a backup file.

Example

The following example shows a generated ticket ID for restoring the configuration of a CPU.

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 164) section.

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method.
4	No resources	You have exhausted all tickets in this user session. Close existing tickets to free up resources. Then call the method again.
5	System is read-only	The system cannot be written to (SIMATIC Memory Card is read-only). Changes are currently not permitted.
1000	Backup creation in progress	The creation of a backup file is in progress.
1001	Backup restoration in progress	The restoration of a saved configuration is currently being carried out. It is not possible to perform both operations at the same time.
1003	Restore not possible through this interface	Calling the method via CM/CP modules, via IP the address or via a virtual CP is not allowed. Perform the recovery via the IP address of one of the network interfaces of a CPU.
1004	PLC not in STOP	A backup file can only be created when the CPU is in STOP mode. Set the CPU to STOP mode and execute the request again.
1005	Legitimation failed	The user legitimation was not successful. Reasons for this can be the entry of password for the "Everybody" or "Anonymous" user, or the entry of an invalid password.

Overview of the recovery of the CPU configuration

The following section shows you all steps that are required to restore the CPU configuration.

NOTE

Tracking the recovery process via Api.BrowseTickets

The Api.BrowseTickets method provides information about the current status of the recovery.

Use this method to found out the recovery process phase, and whether the recovery process was successful.

1. Authenticate yourself with the API method Api.Login.
2. Request a ticket for restoring a CPU configuration using the Plc.RestoreBackup method.
3. Use the ticket end point to start the upload of the backup file.

   The CPU receives the file header and checks whether it is valid. After a successful check, the CPU restarts after 3 seconds. If the file header is invalid, the CPU aborts the restoration process and the ticket changes to the `failed` status.

   When checking the restoration process, the following states may occur. These states can be read out via the API method Api.BrowseTickets.

Status	Description
`waiting`	The waiting state is active until the upload of the backup file is started.
`ongoing`	As soon as the upload of the backup file has started, the state changes to `ongoing`.

Status	Description
`rebooting_format`	The CPU is restarting. The reason for the restart is the formatting of the SIMATIC Memory Card.
`rebooting`	The CPU is restarting. The reason for the restart is the activation of the restored project.
`failed`	An error occurred during the upload of the backup file. You can abort the upload when an error occurs.
`failed_failsafe`	An error occurred during the execution of the fail-safe function. Ensure that the password passed to the Plc.RestoreBackup method is correct and the user has the F-Admin function right.
`failed_wrong_interface`	You have started the upload of the backup file to the CPU via a CM or CP interface.

NOTE

Loss of the configuration during the restoration process

Note that the CPU loses the configuration during the restoration process.

4. Before restarting the CPU, it is possible to read the status with the Api.BrowseTickets(id) method. The additional information informs you about restarting the CPU and formatting the SIMATIC Memory Card as next steps.

   To be informed about the process and all messages, we recommend that you read the information of the Api.BrowseTickets method cyclically, e.g. every second.
5. The CPU then restarts and formats the SIMATIC Memory Card.

   During the restart you can use the Api.Ping method to determine when the CPU is available again.

   During the restart, the Api.BrowseTickets and Api.Ping methods do not respond.
6. The CPU puts the web server into a state with reduced functionality. Only a limited number of API methods is available to you during this time.

NOTE

If you want to restore the CPU to its normal state during the restoration process, perform a download via the TIA Portal. After the download, the CPU and the web server are again in normal operation and all functions are available for use.

7. Use the Api.Login method to log on with the logon data that were also valid at the beginning of the restoration process.
8. Request a ticket for the restore using the Plc.RestoreBackup method.
9. Upload the backup file via the ticket end point.

   After successful upload, the CPU restarts after 3 seconds.
10. Before restarting the CPU, it is possible to read the status with the Api.BrowseTickets(id) method. The additional information informs you about the restart and about the successful upload.

    The restart is required to activate the new project.

11. During the restart you can use the Api.Ping method to determine when the CPU is available again.
12. As soon as the CPU is available again, the restoration process is completed and the recovered project is loaded into the CPU.

You can now log in with the credentials of a user of the project loaded into the CPU, if desired.
Further work in the API requires a new login with the Api.Login method.

5.11 Accessing the contents of the SIMATIC Memory Card

The methods described in this section allow you to access the files in the file system on the SIMATIC Memory Card. You can access standard files as well as your own user files (UserFiles), DataLogs, and recipes.

NOTE

Access to the file system for R/H-CPUs

The file API only offers limited access to the SIMATIC Memory Card. Access is restricted to the "UserFiles", "DataLogs", and "Recipes" folders. Other content on the SIMATIC Smart Card is not accessible via the API.

5.11.1 Files.Browse

This method returns a list of the contents of subdirectories and attributes of specific directories or files which are located on the SIMATIC Memory Card of a CPU.
To call the Files.Browse method, you need the "read_file" permission.

NOTE

For R/H-CPUs, only a maximum of three folders – Recipes, UserFiles and DataLogs – are returned in the root directory of the SIMATIC memory card. No other folders are returned.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-64 Files_Browse_Request (object)

Name	Required	Data type	Description
resource	No	string	Path to the directory or file from the root node. For the root node, the use of a "/" is necessary. You can optionally use a "/" for the root node. If the attribute "resource" is missing or empty, the system interprets it as "/".

Example

The following example shows a request specifying the desired path to a txt file:

```
{
"resource": "/myfolder/file.txt"
}
```

The following example shows a request specifying the desired path to a csv file (datalog).

```
{
"resource": "/Datalogs/datalog1.csv"
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-65 Communication_ReadStatistics_Response (object)

Name	Required	Data type	Description
resources	yes	array of files_Browse_Entry_Response	Resource list.

Table 5-66 Files_Browse_Entry_Response (object)

Name	Required	Data type	Description
name	yes	string	Name of the entry.
type	yes	string	Type of entry, either "file" or "dir".
size	No	number	Size of the file in bytes (if type is "file").
last_modified	yes	string	ISO8601 time stamp as string; time stamp of the last change.
state	No	string	Attribute reserved for active or inactive DataLogs in the "DataLogs" folder

Example

The following example shows the response to a request with a file and a folder:

```
{
"resources":
[
{
"name": "my_dir",
"type": "dir",
"last_modified": "2012-04-23T18:25:43Z"
},
{
```

```
"name": "my_file.txt",
"type": "file",
"size": 87654567,
"last_modified": "2012-04-23T18:25:43Z"
}
]
}
```

The following example shows the response to a query with an active csv file:

```
{
"resources":
[
{
"name": "datalog1.csv",
"type": "file",
"size": 87654567,
"last_modified": "2012-04-23T18:25:43Z",
"state": "active"
}
]
}
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method.
4	No resources	The system does not have the required resources to carry out this request.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The file or path to be accessed by the "resource" parameter does not exist.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.

5.11.2 Files.Download

This method creates a ticket that you can use to download a file from the CPU.
To call the Files.Download method, you need the "read_file" permission.
For more information about the ticket mechanism, see the Ticket mechanism (Page 164) section.

Structure of the request

The following table contains information about the individual parameters of the request:

Table 5-67 Files_Download_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path to the file from the root node of the SIMATIC Memory Card

Examples

The following example shows a request specifying the path to a txt file:

```
{
"resource": "/myfolder/file.txt"
}
```

The following example shows a request specifying the path to a csv file (datalog):

```
{
"resource": "/Datalogs/datalog1.csv"
}
```

Response structure

If successful, the method returns a string with a ticket ID.

Example

The following example shows a generated ticket ID for downloading the file:

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

NOTE

Ticket-based file downloads

For all ticket-based file downloads, the ticket returns a file name in the HTTP content disposition header. You can use this file name as the default file name, or as the name used by the web browser as the default name.

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The user does not have read permissions.
4	No resources	The system does not have the required resources to carry out this request.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The file or path to be accessed by the "resource" parameter does not exist.
303	Entity in use	The file or path to be accessed by the "resource" parameter is locked because of another operation (e.g. because of a write access).
305	Entity not a directory	The directory or subdirectory specified under "resource" is attempting to access a file.
306	Entity not a file	The file name specified under "resource" is attempting to access a directory.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.

5.11.3 Files.Create

This method creates a ticket that you use to upload a file to the CPU.

To call the Files.Create method, you need the "write_file" permission.

For more information about the ticket mechanism, see the Ticket mechanism (Page 164) section.

NOTE

Uploading a file on R/H-CPUs

If a file is uploaded during SYNCUP, the R/H system aborts the upload of this file. During SYNCUP, the R/H does not wait until the file has uploaded.

NOTE

Storage location of files on R/H-CPUs

Files can only be created in the Recipes and UserFiles folders and not in the root directory or outside these two folders.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-68 Files_Create_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path to the file from the root node of the SIMATIC Memory Card

Example

The following example shows a request specifying the path to the desired file:

```
{
"resource": "/mydir/file.txt"
}
```

Response structure

If successful, the method returns a string with a ticket ID.

Example

The following example shows a generated ticket ID for uploading the file:

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The user does not have read permissions.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. File changes are not possible because the SIMATIC Memory Card is read-only.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The directory or subdirectory to be accessed by the "resource" parameter does not exist.
304	Entity already exists	The parameter specified under "resource" is attempting to create a file that already exists.
305	Entity not a directory	The directory or subdirectory specified under "resource" is attempting to access a file.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.

5.11.4 Files.Rename

This method changes the name of a file or directory. You can also use this method to move files from one directory to another.
To call the Files.Rename method, you need the "write_file" permission.

NOTE

Note that you cannot use this method for the "DataLogs" folder.

Structure of the request

The following table contains information about the individual parameters of the request:

Table 5-69 Files_Rename_Request (object)

Name	Required	Data type	Description
resource	yes	string	Current file path or directory path.
new_resource	yes	string	New file path or directory path.

Example

The following example shows a change of the file name.

```
{
"resource": "/folder/old_file_name.txt",
"new_resource": "/folder/new_file_name.txt"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The user does not have write permission for the file.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to (SIMATIC Memory Card is read-only). Changes are currently not permitted.
300	Path contains an illegal sequence	The parameter specified under "resource" and/or "new_resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" and/or under "new_resource" is subject to access restrictions.

Error code	Error message	Meaning
302	Entity does not exist	The file or path to be accessed by the parameter "resource" and/or under "new_resource" does not exist.
303	Entity in use	The parameter specified under "resource" is accessing a file or directory that is already locked by another operation (e.g. read or write).
304	Entity already exists	The parameter specified in "new_resource" is attempting to create a file or directory that already exists.
305	Entity not a directory	The parameter specified under "resource" and/or "new_resource" is attempting to access a file.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.
308	Directories cannot be moved	The parameter specified under "resource" and/or "new_resource" is attempting to move a directory structure. Moving a directory structure is not allowed.

5.11.5 Files.Delete

This method deletes files from the CPU.
To call the Files.Delete method, you need the "write_file" permission.

NOTE

Deleting DataLog files

You can also delete DataLog files with this method, but only if the file is not currently in use. If the DataLog file is currently in use, error message 303 appears: Entity in use.

NOTE

Deleting inactive DataLog files

If you have created a ticket for DataLogs.DownloadAndClear or Files.Download on an inactive DataLog file, you can still use the Files.Delete method to delete this file.
As a result, a download that has already been started or will be started in the future will fail with these tickets.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-70 Files_Delete_Request (object)

Name	Required	Data type	Description
resource	Yes	string	Path to the file from the root node of the SIMATIC Memory Card

Example

The following example shows a request specifying the path to a txt file:

```
{
"resource": "/myfolder/file.txt"
}
```

The following example shows a request specifying the path to a csv file (datalog).

```
{
"resource": "/Datalogs/datalog1.csv"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The user does not have write permission for the file.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. File changes are not possible because the SIMATIC Memory Card is read-only.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The directory or subdirectory to be accessed by the "resource" parameter does not exist.
303	Entity in use	The parameter specified under "resource" is accessing a file or directory that is already locked by another operation (e.g. read or write).
305	Entity not a directory	The directory or subdirectory specified under "resource" is attempting to access a file.
306	Entity not a file	The file name specified under "resource" is attempting to access a directory.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.

5.11.6 Files.CreateDirectory

This method creates a new directory on the CPU.
To call the Files.CreateDirectory method, you need the "write_file" permission.

NOTE

Available folders for R/H-CPUs

You can only create the DataLogs, Recipes and UserFiles folders. The folders are created on the system with the corresponding spelling UserFiles, Recipes, DataLogs, regardless of whether you specify resource="/datalogs", for example.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-71 Files_CreateDirectory_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path to the file from the root node of the SIMATIC Memory Card

Example

The following example shows a request specifying the path to the desired directory:

```
{
"resource": "/SPH_Storage/OPCUA"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The user does not have write permissions.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. Changes are not possible because the SIMATIC Memory Card is read-only.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The directory to be accessed by the "resource" parameter does not exist.

Error code	Error message	Meaning
304	Entity already exists	The parameter specified under "resource" is attempting to create a directory that already exists.
305	Entity not a directory	The parameter specified under "resource" is attempting to access a file.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.

5.11.7 Files.DeleteDirectory

This method deletes an existing directory from the CPU.
To call the Files.DeleteDirectory method, you need the "write_file" permission.

NOTE

Recursive deletion

Note that recursive deletion is not possible, and that the directory must be empty before you can delete it.

Structure of the request

The following table contains information about the individual parameters of the request:

Table 5-72 Files_DeleteDirectory_Request (object)

Name	Required	Data type	Description
resource	yes	string	Path of the directory to be deleted.

Example

The following example shows a request specifying the path to the desired directory:

```
{
"resource": "/SPH_Storage"
}
```

Response structure

If successful, the method returns the Boolean value "true".

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The user does not have write permissions.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. Changes are not possible because the SIMATIC Memory Card is read-only.
300	Path contains an illegal sequence	The parameter specified under "resource" violates the naming convention (e.g. contains invalid characters).
301	Entity access is restricted	The parameter specified under "resource" is subject to access restrictions.
302	Entity does not exist	The directory to be accessed by the "resource" parameter does not exist.
303	Entity in use	The parameter specified under "resource" is accessing a directory that is already locked by another operation (e.g. read or write).
305	Entity not a directory	The parameter specified under "resource" is attempting to access a file.
307	Maximum path depth exceeded	The parameter specified under "resource" exceeds the maximum path length.

5.11.8 DataLogs.DownloadAndClear

The DataLogs.DownloadAndClear method creates a ticket to download DataLogs from the CPU and delete them after processing.

NOTE

If you do not want to delete the contents of DataLogs after downloading, use the Files.Download method instead.

For more information about this method, see the Files.Download (Page 237) section.

NOTE

You can apply the DataLogs.DownloadAndClear method only if the DataLog is not currently in use. If the DataLog is currently in use, the error message 303: Entity in use appears.

To call the Files.DownloadAndClear method, you need the write_file permission.

Structure of the request

The following table contains information about the individual parameters of the request:

Table 5-73 DataLogs_DownloadAndClear_Request (object)

Name	Required	Data type	Description
resource	yes	string	The name of the DataLog you want to download. Alternatively, you can use a path starting with /datalogs/ (see examples below).

Examples

The following example shows a request specifying the name of the desired DataLog.

```
{
"resource": "datalog1"
}
```

The following example shows a request with specification of the path to the desired DataLog:

```
{
"resource": "/datalogs/datalog1"
}
```

Response structure

If successful, the method returns a string with a ticket ID.

Example

The following example shows a generated ticket ID for downloading and deleting the DataLog:

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The user does not have write permissions.
4	No resources	The system does not have the required resources to carry out this request.
5	System is read-only	The system cannot be written to at present. Changes to the log file are not possible because the SIMATIC Memory Card is write-protected.
303	Entity in use	The file or path to be accessed by the "resource" parameter is locked because of another operation (e.g. the user program is currently accessing the DataLog).
309	Entity is not a valid data log	The parameter specified under "resource" is attempting to access an unlinked DataLog. An unlinked DataLog cannot be deleted.

5.12 Reading information from SIMATIC Safety

5.12.1 Failsafe.ReadRuntimeGroups

This method outputs a list with all available F-runtime groups.
To call the Failsafe.ReadRuntimeGroups method, you need the "read_diagnostics" permission.

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-74 Failsafe_ReadRuntimeGroups_Response (object)

Name	Required	Data type	Description
groups	Yes	array of Failsafe_ReadRuntimeGroups_Entry_Response	Object array in which each object represents one F-runtime group.

Table 5-75 Failsafe_ReadRuntimeGroups_Entry_Response (object)

Name	Required	Data type	Description
name	Yes	string	Name of F-runtime group
signature	Yes	string	Signature of F-runtime group as array of decimal numbers Each number represents one byte of the signature.
cycle_time_current	Yes	string	ISO 8601 time span as string; Current cycle time in milliseconds.
cycle_time_max	Yes	string	ISO 8601 time span as string; Maximum cycle time in milliseconds.
runtime_current	Yes	string	ISO 8601 time span as string; Current runtime in milliseconds.
runtime_max	Yes	string	ISO 8601 time span as string; Maximum runtime in milliseconds.

Examples

The following example shows the parameters of the response to a query with an F-runtime group with remaining time "remaining_time".

```
{
  "groups":
  [
    {
      "name": "RTG_1",
      "signature": "FD62F235",
      "cycle_time_current": "PT0.110S",
      "cycle_time_max": "PT0.200S",
      "runtime_current": "PT0.050S",
      "runtime_max": "PT0.080S"
    }
  ]
}
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.

5.12.2 Failsafe.ReadParameters

With this method, you can read out fail-safe parameters of a fail-safe CPU or a fail-safe module via the hardware ID of the module.

To call the Failsafe.ReadParameters method, you need the "read_diagnostics" permission.

Structure of the request

The following table provides information about the individual parameters of the request.

Table 5-76 Failsafe_ReadModuleParameters_Request (object)

Name	Required	Data type	Description
hwid	Yes	number	Hardware ID of the module whose parameters you want to read out.

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-77 Failsafe_ReadParameters_Response (object)

Name	Required	Data type	Description
safety_mode	No	string	Status message indicating whether safety mode is active ("enabled") or not ("disabled"). Note that this status message only applies to the CPU and not to other modules.
type	Yes	string	Defines whether the required hardware ID is the fail-safe module or represents a different fail-safe module.
parameters	No	object	Indicates whether the required hardware ID is the fail-safe module with safety program or represents a different fail-safe module. The returned object is alternatively of type: • Failsafe_ReadParameters_Cpu_Response • Failsafe_ReadParameters_Module_Response

Table 5-78 Failsafe_ReadParameters_Cpu_Response (object)

Name	Required	Data type	Description
last_f_program_modification	Yes	string	ISO 8601 time stamp in UTC as string; time stamp of the last change in the safety program.
collective_signature	Yes	string	Collective F signature as byte array with 4 numbers for representing a 32-bit signature.
remaining_time	No	string	ISO 8601 time span as string; Remaining time in milliseconds (as of firmware version V2.9).

Table 5-79 Failsafe_ReadParameters_Module_Response (object)

Name	Required	Data type	Description
f_monitoring_time	Yes	number	ISO 8601 time stamp in UTC as string; F-monitoring time in milliseconds
f_source_address	Yes	number	F-source address
f_destination_address	Yes	number	F-destination address
f_par_crc	Yes	array of number	CRC signature of the F parameters as a byte array with 4 numbers for representing a 32-bit signature.

Examples

The following example shows the parameters of a fail-safe CPU in active safety mode.

```
{
"safety_mode": "enabled",
"type": "cpu",
"parameters":
{
"last_f_program_modification": "2023-11-05T18:25:43.510Z",
"collective_signature": "C572BC16"
}
}
```

The following example shows the parameters of a fail-safe module in active safety mode.

```
{
"type": "module",
"parameters":
{
"f_monitoring_time": "PT0.123S",
"f_source_address": 123,
"f_destination_address": 123,
"f_par_crc": "F062F235"
}
}
```

Possible error messages

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
1100	Invalid hardware identifier	The specified hardware ID is not valid for the current request. Make sure that you have used the correct hardware ID.

5.13 Web applications that can be loaded by the user

With the Web server as of firmware version V2.9, you can use web applications that can be loaded by the user (user-defined). In the following, we will refer to "web applications" for short.

Web applications offer you a set of methods to manage web applications via Web API. You can use all available Web API methods within a web application.

NOTE

HTTP-Range-Requests

For access to web application resources, the web Server provides you with a limited support for HTTP range requests. These requests allow you to read individual areas of a resource.

Advantages

Web applications offer you advantages compared to the older method that provided customer pages via the system function SFC 99 in STEP 7:

- The resources are saved in the associated web application. Via the Web API, you can download the resources to your PC, edit them and upload them back to the CPU. This procedure results in significantly reduced development times of customer pages.
- Unlike the previous customer pages, you can test the web application during implementation without having to load it onto the CPU.
- You can access resources independent of the CPU mode (e.g. RUN, STOP) and update these.
- Web applications are also available in the STOP mode of the CPU.
- No synchronization between the user program and Web server required by the SFC 99 instruction.
- You can access multiple CPUs within a web application using the Web API .
- The CPU supports saving multiple web applications that you can access simultaneously via the HTTP end point.
- Access to the resources of a web application via the HTTP end point can be activated or deactivated for each application. In so doing, an administrator can deactivate access to a web application, for example, to update the resources consistently.
- You can specify a standard entry page for each application. When you visit the basic URL of a web application, such as https://[ip]/~[application_name], you are automatically being forwarded to the configured home page.
- You can specify an individual media type (MIME type) for each resource.

- You can specify a visibility flag for each resource:
  - Public resource: Access without user authentication
  - Protected resource: Access limited to authenticated users with access right "open_user_pages"
- You can specify for each resource:
  - The time stamp of the resource change
  - The value for the HTTP header ETag

  In so doing, you reduce the communication load on the CPU. This enables the caching of resource files by the web browser.

NOTE

When you use the Web API for managing web applications, the TIA Portal project in the SIMATIC.S7S directory on the SIMATIC Memory Card changes. Your TIA Portal project is extended by the option of saving resources (e.g. HTML, CSS, JavaScript, etc.) in the project but outside of the data blocks of the user program.

As with the previous customer pages, the web applications must not contain any instructions in the STEP 7 user program and are thus purely static files without dynamic content.

Methods for managing web applications

The following methods are available to manage web applications using Web API:

Table 5-80 Methods for managing web applications

Method	Explanation
WebApp.Create (Page 254)	Enables users to create a new web application.
WebApp.Delete (Page 255)	Enables users to delete an existing web application.
WebApp.Rename (Page 256)	Enables users to change the name of an existing web application.
WebApp.Browse (Page 257)	Enables users to display a list of web applications with the associated properties.
WebApp.SetState (Page 259)	Enables users to activate /deactivate a web application for access from the HTTP end point.
WebApp.SetDefaultPage (Page 260)	Enables users to define a default page for the web application.
WebApp.SetNotFoundPage (Page 261)	Enables users to define a page that is loaded when a requested resource does not exist in the web application.
WebApp.SetNotAuthorizedPage (Page 262)	Enables users to define a page that is loaded when a requested resource is not public (protected) in the web application.

Methods for managing resources

The following methods are available to manage the resources of a web application using Web API:

Table 5-81 Web applications: Methods for managing resources

Method	Explanation
WebApp.BrowseResources (Page 264)	Enables users to display all resources with their properties that are assigned to a web application.
WebApp.CreateResource (Page 266)	Enables users to create a new resource in a web application.
WebApp.DeleteResource (Page 268)	Enables users to delete an existing resource in a web application.
WebApp.RenameResource (Page 269)	Enables users to change the name of an existing resource in a web application.
WebApp.DownloadResource (Page 270)	Allows the user to download a resource from a web application from the CPU.
WebApp.SetResourceVisibility (Page 271)	Enables users to change the visibility of a resource in a web application.
WebApp.SetResourceETag (Page 273)	Enables users to change or delete the ETag value of a resource in a web application.
WebApp.SetResourceMediaType (Page 274)	Enables users to change the media type of a resource in a web application.
WebApp.SetResourceModificationTime (Page 275)	Enables users to set the modification time of a resource in a web application.

End point for web applications

Web applications are only accessible via secure HTTPS communication. This increases security when accessing the resources of the CPU. When the Web server was configured for HTTP access, requests via HTTP are automatically rerouted to an HTTPS connection.

A web application is accessible via the following URL, provided that the default page (default_page) or the substitute page of an application (not_found_page) is configured:

```
https://[ip]/~[application_name]
```

A resource of a web application can be reached via the following URL:

```
https://[ip]/~[application_name]/[resource_name]
```

NOTE

The tilde symbol "~" is important in the path and must always exist for web applications.

Procedure to access resources of a protected web application

If resources of a web application are protected and you want to access/modify data of this web application, then you have to authenticate yourself first, for example on a public page.

For example, perform the following step on the default page:

- From JavaScript, call the Api.Login method using the "include_web_application_cookie" parameter.

  Result: If authentication is successful, the Api.Login method returns the session token and a cookie for accessing the protected files of a web application (see also section Api.Login (Page 171)).

  JavaScript is used to set the cookie "web_application_cookie" with a value from the HTTP response of the login as cookie "siemens_web_secure".

NOTE

To ensure that the API token is not lost after the web page is called up again, you can save the API token in a cookie or in the LoadStorage of the web browser, for example, using JavaScript.

The behavior after timeout for protected web applications is the same as the behavior after timeout for web API sessions: After a timeout, the cookie becomes invalid for access to the protected web applications. Reloading a resource file of a web application does not extend the session. Use appropriate methods of the Web API to stay logged in (see section Web API Sessions (Page 163)).

Rules for a valid application name

The application name may be max. 100 letters/characters long. The following letters and characters are permitted for the application name:

`A-Z, a-z, 0-9, -_.+"`

Rules for a valid resource name

The resource name can be max. 200 letters/characters in length. The following letters and characters are permitted for the resource name:

`A-Z, a-z, 0-9, -_.+()/|,*!'"`

Rules for a valid name for the media type

The web browser needs a media type to display a file correctly or to open it. The format of the media type is standardized. Valid media types are, for example, "text/html" or "image/jpeg". You can use all valid media types.

More information can be found on the Internet by entering "MIME type" or "media type".

5.13.1 WebApp.Create

With the WebApp.Create method, you can create a new web application in the CPU.
To call the WebApp.Create method, you need the "manage_user_pages" authorization.

Structure of the request

The following table informs you about the necessary parameters for the request.

Table 5-82 WebApp_Create_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the user-defined web application.
state	no, default is "enabled"	string	The status of the application is: • "disabled": Pages cannot be reached via HTTP end point • "enabled": Pages can be reached via HTTP end point

Example

In the following example, the user creates an application with the name "application_1".

```
{
"name": "application_1",
"state": "enabled"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.Create method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
500	Application name already exists	An application with the name already exists. Assign a name that does not exist yet.
502	Application limit reached	The maximum number of web applications has been reached. Delete applications that you do not need to free up resources for new applications.
503	Invalid application name	The name of the application is invalid. Assign an application name that meets the rules for a valid application name (see section Web applications that can be loaded by the user (Page 250)).

5.13.2 WebApp.Delete

With the WebApp.Delete method, you delete an existing web application that was loaded by the user with all its web page resources.
To call the WebApp.Delete method, you need the "manage_user_pages" authorization.

Structure of the request

The following table informs you about the necessary parameters for the request.

Table 5-83 WebApp_Delete_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application that is deleted

Example

In the following example, the user deletes an application with the name "application1".

```
{
"name": "application1"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.Delete method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
3	System is busy	The desired operation cannot be performed because the system is currently performing a different request. Restart the query as soon as the current operation is complete.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
501	Application does not exist	An application with this name does not exist.

5.13.3 WebApp.Rename

With the WebApp.Rename method, you change the name of the web application to a new name.
To call the WebApp.Rename method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-84 WebApp_Rename_Request (object)

Name	Required	Data type	Description
name	yes	string	The current name of the web application.
new_name	yes	string	The new name of the web application.

Example

In the following example, the user changes the name of the application from "application1" to "application_new".

```
{
"name": "application1"
"new_name": "application_new"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.Rename method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
500	Application name already exists	An application with the name already exists. Assign a name that does not exist yet.
501	Application does not exist	An application with this name does not exist.
503	Invalid application name	The name of the application is invalid. Assign an application name that meets the rules for a valid application name (see section Web applications that can be loaded by the user (Page 250)).

5.13.4 WebApp.Browse

The WebApp.Browse method returns alternatively:

- A list of all web applications with the associated properties.
- The properties for a specific web application

No authorization is required for calling the WebApp.Browse method.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-85 WebApp_Browse_Request (object)

Name	Required	Data type	Description
name	No, default is an empty string	string	If this parameter does not exist, all applications will be returned by the method. If the parameter is available, the list will contain only the application whose name matches this parameter. If you have not specified a name, the list may also be empty, depending on whether applications are present or not.

Example

In the following example, the user specifies the name of the "application1" for the list.

```
{
"name": "application1"
}
```

Response structure

The following tables show the structure of server responses to successful requests.

Table 5-86 WebApp_Browse_Request (object)

Name	Required	Data type	Description
max_applications	yes	number	Maximum number of applications supported by the CPU
applications	yes	array of WebApp_Browse_ Application_Respon-se	List of the applications in the CPU

Table 5-87 WebApp_Browse_Application_Response (object)

Name	Required	Data type	Description
name	yes	string	The name of the application
state	yes	string	The status of the application: • "disabled": Pages cannot be reached via HTTP end point • "enabled": Pages can be reached via HTTP end point
type	yes	string	The type of application: • "user": created by the user

Name	Required	Data type	Description
default_page	No	string	Default page of the application when no resource name was specified when accessing the web application
not_found_page	No	string	Substitute page in an application when the requested resource does not exist
not_authorized_ page	No	string	If the user has tried to access a protected resource to which the user does not have access. This page can, for example, be implemented as a login page.

Example

In the following example, the user requests responses from the server.

```
{
"max_applications": 4,
"applications": [
{
"name": "application1",
"state": "enabled",
"type": "user",
"default_page": "index.html"
},
{
"name": "vottheapp",
"state": "enabled",
"type": "vot"
}
]
}
```

Possible error messages

The following table shows the possible error messages of the WebApp.Browse method.

Error code	Error message	Meaning
501	Application does not exist	An application with this name does not exist.

5.13.5 WebApp.SetState

With the WebApp.SetState method, you can activate or deactivate a web application that was loaded by the user. A web application that was loaded by the user and is deactivated cannot be called by the HTTP end point.

To call the WebApp.SetState method, you need the "manage_user_pages" authorization.

Structure of the request

The following table informs you about the necessary parameters for the request.

Table 5-88 WebApp_SetState_Request (object)

Name	Required	Data type	Description
name	yes	string	Name of the web application whose status is changed
state	yes	string	The status of the application is: • disabled; pages cannot be reached via HTTP end point • enabled; pages can be reached via HTTP end point

Example

In the following example, the user deactivates the application with the name "webapp":

```
{
"name": "webapp",
"state": "disabled"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetState method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
501	Application does not exist	An application with this name does not exist.

5.13.6 WebApp.SetDefaultPage

With the WebApp.SetDefaultPage method, you set a default page for a user-defined web application. This page is loaded if you have not assigned a resource name when accessing the web application.

To call the WebApp.SetDefaultPage method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-89 WebApp_SetDefaultPage_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application for which the default page is to be configured
resource_name	yes	string	The name of the resource in the web application that was loaded by the user An empty character string indicates that the default page is to be deleted.

Example

In the following example, the page "index.html" is used as a resource in the web application "application_1".

```
{
"name": "application_1",
"resource_name": "index.html"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetDefaultPage method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.

Error code	Error message	Meaning
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	The requested resource does not exist in the application. When calling this method, select a resource in the application.
505	Resource visibility is not public	The requested resource is not marked as "public". You should change the resource to "public" or select another resource that is already "public".

5.13.7 WebApp.SetNotFoundPage

With the WebApp.SetNotFoundPage method, you set a page for a user-defined web application. This page is loaded if you have used a resource name that does not exist when accessing the web application.

To call the WebApp.SetNotFoundPage method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-90 WebApp_SetNotFoundPage_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application whose page must be changed
resource_name	yes	string	The name of the resource in the web application that was loaded by the user An empty character string indicates that the PageNotFound page should no longer be set.

Example

In the following example, the page "404.html" is used as a resource in the "application_1" web application.

```
{
"name": "application_1",
"resource_name": "404.html"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetNotFoundPage method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	The requested resource does not exist in the application. When calling this method, select a resource in the application.
505	Resource visibility is not public	The requested resource is not marked as "public". You should change the resource to "public" or select another resource that is already "public".

5.13.8 WebApp.SetNotAuthorizedPage

With the WebApp.SetNotAuthorizedPage method, you set a publicly visible page for a user-defined web application. This page is loaded if you have used a resource name that cannot be accessed publicly (protected) when accessing the web application, thus accessing the web application without a valid cookie.

You can get a valid cookie using the Api.Login (Page 171) method with the "include_web_application_cookie" parameter.

To call the WebApp.SetNotAuthorizedPage method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-91 WebApp_SetDefaultPage_Request (object)

Name	Required	Data type	Description
name	yes	string	The name of the web application whose public page must be changed
resource_name	yes	string	The name of the resource in the web application that was loaded by the user An empty character string indicates that the non-public page is to be deleted.

Example

In the following example, the page "login.html" is used as a resource in the "application_1" web application.

```
{
"name": "application_1",
"resource_name": "login.html"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetNotAuthorizedPage method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	The requested resource does not exist in the application. When calling this method, select a resource in the application.
505	Resource visibility is not public	The requested resource is not marked as "public". You should change the resource to "public" or select another resource that is already "public". An example with a set cookie "web_application_cookie" for access to protected web applications can be found in section Example: Web page for monitoring and controlling a wind turbine.

5.13.9 WebApp.BrowseResources

The WebApp.BrowseResources method provides a list of all resources with their properties that are assigned to a web application.

To call the WebApp.BrowseResources method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-92 WebApp_BrowseResources_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application whose list is provided
name	No, default is an empty string	string	If this parameter does not exist, all resources must be returned. Otherwise, the list of returned resources only contains one resource whose name matches this parameter. If no such resource is found, then: • The return list must be empty • It is not an error

Example

In the following example, the user specifies the name of the "application1" application for the list.

```
{
"app_name": "application_1"
}
```

Response structure

The following tables show the structure of the server response to a successful request.

Table 5-93 WebApp_BrowseResources_Response (object)

Name	Required	Data type	Description
max_resources	yes	number	Maximum number of resources supported by the CPU.
resources	yes	array of WebApp_BrowseResources_ Resource_Response	List of the resources in the specific application.

Table 5-94 WebApp_BrowseResources_Resource_Response (object)

Name	Required	Data type	Description
name	yes	string	The name of the resource.
size	yes	integer	The size of the resource in bytes.
media_type	yes	string	The media type of the resource.

Name	Required	Data type	Description
etag	No	string	The ETag value of the resource.
visibility	yes	string	The visibility of the resource.
last_modified	yes	string	ISO8601 time stamp as string. The time stamp of the last modification.

Example

In the following example, the user requests responses from the server.

```
{
"max_resources": 200,
"resources": [
{
"name": "index.html",
"size": 24387,
"media_type": "text/html",
"etag": "896a9s8df0897g098a",
"visibility": "public",
"last_modified": "2020-08-24T07:08:06Z"
},
{
"file_name": "secret.js",
"size": 97826348,
"media_type": "application/javascript",
"visibility": "protected",
"last_modified": "2020-08-24T07:08:06Z"
}
]
}
```

Possible error messages

The following table shows the possible error messages of the WebApp.BrowseResources method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. When calling this method, select a resource in the application.

5.13.10 WebApp.CreateResource

With the WebApp.CreateResource method, you create a new resource in a web application that was loaded by the user on the CPU.

To call the WebApp.CreateResource method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-95 WebApp_CreateResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application (Page 250) for which a resource must be created
name	yes	string	The name of the resource (Page 250) that is uploaded
media_type	yes	string	The media type of the resource (Page 250).
visibility	no; the default value is "public"	string	The visibility of the resource (Page 271).
etag	no; the default value is an empty character string	string	The ETag value of the resource (Page 273).
last_modified	yes	string	ISO8601 time stamp as string. The time stamp of the last modification.

Example

In the following example, the user creates a new resource "index.html" of the application "application_1" with the media type "text/html".

```
{
"app_name": "application_1",
"name": "index.html",
"media_type": "text/html",
"last_modified": "2020-08-24T07:08:06Z"
}
```

Response structure

This method returns a character string that includes a valid ticket ID. The user uses this ticket to upload the content of the resource to the CPU using the ticket end point.

NOTE

A maximum of only one WebApp.CreateResource ticket can be created. A new ticket can be created for this method only after this ticket has been closed.

Example of ticket ID:

`"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"`

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 164) section.

Possible error messages

The following table shows the possible error messages of the WebApp.CreateResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	You have exhausted all tickets in this user session or already have a ticket for WebApp.CreateResource that has not yet been closed. Close existing tickets to free up resources. Then call the method again.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
507	Resource alredy exists	A resource with the specified name already exists for this application. Select a new resource name or delete/rename the resource before you call this method.
508	Invalid resource name	The resource name is invalid. Correct the resource name (Page 250) before you call this method.
509	Resource limit reached	The maximum number of resources has been exhausted for this application. Delete some resources before you call this method.
511	Invalid modification time	The planned modification time is not within the permissible time window of the modification time. Reduce the modification time accordingly before you call this method.
512	Invalid media type	The media type is invalid. Correct the media type (Page 250) before you call this method.
513	Invalid ETag	The ETag value is invalid. Correct the ETag value (Page 273) before you call this method.

5.13.11 WebApp.DeleteResource

With the WebApp.Delete method, you can delete a resource from a specific web application. To call the WebApp.DeleteResource method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-96 WebApp_DeleteResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the user-defined web application from which the resource is to be deleted
name	yes	string	The name of the resource that is to be deleted

Example

In the following example, the user deletes the resource "img/wrong.png" of the application "application_1".

```
{
"app_name": "application1",
"name": "img/wrong.png"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.DeleteResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with the specified name does not exist. When calling this method, select a resource in the application.

5.13.12 WebApp.RenameResource

With the WebApp.RenameResource method, you change the name of a resource in a specific web application.

To call the WebApp.RenameResource method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-97 WebApp_RenameResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the user-defined web application in which the name of the resource is changed
name	yes	string	The name of the resource that is changed
new_name	yes	string	The new name of the resource.

Example

In the following example, the user changes the name of the resource to "corrspelled.png" in the "application1" application.

```
{
"app_name": "application1",
"name": "msspelled.png",
"new_name": "corrspelled.png"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.RenameResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.

Error code	Error message	Meaning
506	Resource does not exist	A resource with this name does not exist. When calling this method, select a resource in the application.
507	Resource already exists	A resource with the new name already exists for this application. Select a new resource name or delete/rename the resource before you call this method.
508	Invalid resource name	The new resource name is invalid. Correct the resource name (Page 250) before you call this method.

5.13.13 WebApp.DownloadResource

With the WebApp.DownloadResource method, you can load a resource to a web application that was loaded by the user from the CPU.
To call the WebApp.DownloadResource method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-98 WebApp_DownloadResource_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is downloaded

Example

In the following example, the user downloads the resource "secrets.pdf" of the application "application_1".

```
{
"app_name": "application_1",
"name": "secrets.pdf"
}
```

Response structure

This method returns a character string that includes a valid ticket ID. The user uses this ticket to download the content of the resource from the CPU using the ticket end point.

NOTE

A maximum of 1 WebApp.DownloadResource ticket can be created at the same time.

Example of ticket ID:

```
"NDU2Nzg5MDEyMzQ1Njc4OTAxMjM0"
```

An example of further processing of the ticket ID can be found in the Ticket mechanism (Page 164) section.

Possible error messages

The following table shows the possible error messages of the WebApp.DownloadResource method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
4	No resources	You have exhausted all tickets in this user session or already have a ticket for WebApp.DownloadResource that has not yet been closed. Close existing tickets to free up resources. Then call the method again.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
514	Resource content has been corrupted	The file content has been damaged as a result of file manipulations on the SIMATIC Memory Card. You can rectify the damage by deleting and recreating the resource file via the API.
504	Resource content is not ready	The content of the resource is not yet ready because it is currently being uploaded. Wait until the upload has finished.

5.13.14 WebApp.SetResourceVisibility

With the WebApp.SetResourceVisibility method, you change the visibility of a resource, public or protected, in a web application that was loaded by the user. A protected resource cannot be called by the HTTP end point without authentication.

To call the WebApp.SetResourceVisibility method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-99 WebApp_SetResourceVisibility_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is changed
visibility	yes	string	The visibility of the resource. public: public protected: only for authorized users An example with a set cookie "web_application_cookie" for access to protected web applications can be found in section Example: Web page for monitoring and controlling a wind turbine.

Example

In the following example, the user sets the "secrets.html" resource of the "myapp" application to "protected":

```
{
"app_name": "myapp",
"name": "secrets.html",
"visibility": "protected"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetResourceVisibility method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
505	Resource visibility must be public	The respective resource is referenced: • as default page, • as unauthorized page or • as page not found You cannot mark the resource as protected. You must set the visibility in the application to "public" before calling this method.

5.13.15 WebApp.SetResourceETag

With the WebApp.SetResourceETag method, you change or delete the ETag attribute that is returned when accessing the resource via the HTTP header. ETag (Entity Tag) is an HTTP header field. It only serves to determine changes at the requested resource and is used to avoid redundant data transfers.

To call the WebApp.SetResourceETag method, you need the "manage_user_pages" authorization.

Rules for a valid ETag value

You can use any character string as ETag value as seen in the example below. The length is limited to a maximum of 128 characters.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-100 WebApp_SetResourceETag_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is to be changed.
etag	yes	string	The ETag value of the resource. An empty character string indicates that the value is to be deleted.

Example

In the following example, the user sets the ETag value to "09as7df09h8j23r" for the "secrets.html" resource in the "myapp" application.

```
{
"app_name": "myapp",
"name": "secrets.html",
"etag": "09as7df09h8j23r"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetResourceETag method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
513	Invalid ETag	The ETag value is invalid. Correct the value before you call this method.

5.13.16 WebApp.SetResourceMediaType

With the WebApp.SetResourceMediaType method, you change the media type of a resource in a web application that was loaded by the user.
To call the WebApp.SetResourceMediaType method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-101 WebApp_SetResourceMediaType_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is changed
media_type	yes	string	The media type of the resource (Page 250).

Example

In the following example, the user sets the media type to "image/jpeg" for the "secrets.jpg" resource in the "myapp" application.

```
{
"app_name": "myapp",
"name": "secrets.jpg",
"media_type": "image/jpeg"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetResourceMediaType method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
512	Invalid media type	The media type is invalid. Correct the media type (Page 250) before you call this method.

5.13.17 WebApp.SetResourceModificationTime

With the WebApp.SetResourceModificationTime method, you set the modification time of a resource in a web application that was loaded by the user.

To call the WebApp.SetResourceModificationTime method, you need the "manage_user_pages" authorization.

Structure of the request

The following table provides information about the required parameters for the request.

Table 5-102 WebApp_SetResourceModificationTime_Request (object)

Name	Required	Data type	Description
app_name	yes	string	The name of the web application that contains the resource
name	yes	string	The name of the resource that is changed
last_modified	yes	string	ISO8601 time stamp as a string; the time stamp of the last change

Example

In the following example, the user sets the modification time to "24.08.2020 07:08:06" for the "secrets.html" resource in the "myapp" application:

```
{
"app_name": "myapp",
"name": "secrets.html",
"last_modified": "2020-08-24T07:08:06Z"
}
```

Response structure

If successful, the server returns the Boolean value "true".

Possible error messages

The following table shows the possible error messages of the WebApp.SetResourceModificationTime method.

Error code	Error message	Meaning
2	Permission denied	The current authentication token is not authorized to call this method. Log on with a user account that has sufficient privileges to call this method.
5	System is read only	The system is currently in a write-protected state. Changes to web applications are currently not permitted.
6	Not accepted	The method cannot be executed because it is not supported for this application.
501	Application does not exist	An application with this name does not exist.
506	Resource does not exist	A resource with this name does not exist. Select a resource name that exists in the application before you call the method.
511	Invalid modification time	The planned modification time is not within the permissible time window of the modification time. Reduce the modification time accordingly before you call this method.

Glossary

API

Application Programming Interface

Automation system

An automation system is a programmable logic controller that consists of at least one CPU, various input and output modules, as well as operating and monitoring devices.

AWP

Automation Web Programming

AWP commands

Special command syntax for exchanging data between CPU and HTML file.

Configuration

Systematic arrangement of individual modules (design).

CSS

A CSS (Cascading Style Sheet) specifies how an area or content marked up in HTML is displayed.

Device

Device that can send, receive or amplify data via the bus, e.g., IO controller.

Diagnostics

The detection, localization, classification, visualization and further evaluation of errors, malfunction and alarms.

Diagnostics provides monitoring functions that run automatically during plant operation. This increases the availability of plants by reducing commissioning times and downtimes.

EBNF

Extended Backus-Naur form: Formal metalanguage to represent context-free grammar. Formal languages can be expressed in EBNF; especially used in computer science for the definition of programming languages. The EBNF is standardized by ISO as ISO/IEC 14977:1996(E).

Firewall

A firewall is used to restrict the network access based on sender or target address of the used services. The firewall decides based on specified rules which of the network packets it handles are forwarded and which are not. This way the firewall tries to prevent unauthorized network access.

It is not the function of a firewall to detect attacks. It only implements rules for network communication.

HTTP

Hypertext Transfer Protocol (HTTP). Protocol for data transmission across a network.

HTTPS

Hypertext Transfer Protocol Secure (HTTPS). Protocol for tap-proof transmission of sensitive data across a network.

Identification data

Identification data is stored on a module, and contains information which supports the user in

- Checking the system configuration
- Locating hardware changes in a system
- Correcting errors in a system

Modules can be clearly identified online using the identification data.

JSON

JSON (JavaScript Object Notation) is a compact data format in an easy-to-read text form and is used to exchange data between applications. JSON is independent of the programming language. Parsers and generators exist in all common programming languages.

Master

The master in possession of the token is an active device. This master has the option to receive data from other devices and to send data to other devices.

PROFIBUS

PROcess **FIeld BUS**, German process and fieldbus standard that is defined in IEC 61784-1:2002 Ed1 CP 3/1. This standard specifies functional, electrical, and mechanical properties for a bit-serial fieldbus system.

PROFIBUS is available with the protocols DP (= Distributed I/O), FMS (= Fieldbus Message Specification), PA (= Process Automation), or TF (= Technological Functions).

PROFINET

Within the context of Totally Integrated Automation (TIA), PROFINET is the systematic continuation of:

- PROFIBUS DP, the established fieldbus
- Industrial Ethernet, the communications bus for the cell level

Experiences from both systems have been are integrated in PROFINET.

PROFINET as an Ethernet-based automation standard of PROFIBUS International (formerly PROFIBUS User Organization e.V.) thus defines a manufacturer-independent communication, automation and engineering model.

PROFINET components

A PROFINET component includes all data of the hardware configuration, the parameters of the modules, and the associated user program. The PROFINET component comprises the following elements:

- Technological Function

  The (optional) technological (software) function includes the interface to other PROFINET components as interconnectable inputs and outputs.
- Device

  The device is the representation of the physical programmable controller or field device including the I/O, sensors and actuators, mechanical parts, and the device firmware.

PROFINET IO

Within the context of PROFINET, PROFINET IO is a communication concept for implementing modular, distributed applications.

PROFINET IO allows you to create automation solutions familiar from PROFIBUS.

Implementation of PROFINET IO is carried out on the one hand via the PROFINET standard for programmable controllers and on the other by using the engineering tool STEP 7.

This means that you have the same application layer in STEP 7 – regardless of whether you are configuring PROFINET or PROFIBUS devices. The programming of your user program is identical for PROFINET IO and PROFIBUS DP when using the blocks and system status lists expanded for PROFINET IO.

PROFINET IO controller

Device used to address connected I/O devices. This means that the IO controller exchanges input and output signals with assigned field devices. The IO controller is often the controller on which the automation program runs.

PROFINET IO device

A distributed field device that is assigned to one of the IO controllers (e.g., remote IO, valve terminals, frequency converters, switches).

URL

Uniform Resource Locator (URL). Identifies and localizes a source, such as a web page, uniquely via the method of access used and the location of the source in computer networks.

UTF-8

UTF-8 is the abbreviation for 8-bit UCS (Universal Character Set) Transformation Format. UTF-8 is the most widely used encoding of Unicode characters.

Each Unicode character is assigned a specially encoded byte string of variable length. UTF-8 supports up to four bytes onto which all Unicode characters can be mapped.

Web browser

Web browsers are visualization programs for web pages and can communicate with Web servers.

Typical web browsers are, for example:

- Microsoft Internet Explorer
- Mozilla Firefox

Index