

## SIMOTION IT Ethernet-Based HMI and Diagnostic Functions

Product Information

Preface

Introduction

1

SIMOTION IT OPC XML DA  
server

2

Trace Interface via SOAP

3

SIMOTION IT DIAG  
(standard diagnostics pages)

4

Variables providers

5

Secure Socket Layer

6

File access via FTP

7

Appendix

A

## Safety Guidelines

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

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

---

---

### Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

---

---

### Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

---

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

## Prescribed Usage

Note the following:



---

### Warning

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

---

## Trademarks

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

## Disclaimer of Liability

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

# Preface

## Hotline and Internet addresses

If you have any technical questions, please contact our hotline (worldwide):

A & D Technical Supports:

Phone: +49 (180) 50 50 222

Fax: +49 (180) 50 50 223

E-mail: [adsupport@siemens.com](mailto:adsupport@siemens.com)

Internet: <http://www.siemens.de/automation/support-request>

If you have any questions, suggestions, or corrections regarding the documentation, please send them to the following fax number or e-mail address:

Fax: +49 (9131) 98 63315

E-mail: [motioncontrol.docu@siemens.com](mailto:motioncontrol.docu@siemens.com)

## Siemens Internet address

The latest information about SIMOTION products, product support and FAQs can be found on the Internet at:

	<a href="http://www.siemens.de/simotion">http://www.siemens.de/simotion</a> (German)
	<a href="http://www.siemens.com/simotion">http://www.siemens.com/simotion</a> (international)
Product support	<a href="http://support.automation.siemens.com/WW/view/de/10805436">http://support.automation.siemens.com/WW/view/de/10805436</a>

## Further assistance

We also offer introductory courses to help you familiarize yourself with SIMOTION.

Please contact your regional training center or our main training center at D-90027 Nuremberg/Germany, phone +49 (911) 895 3202.



# Table of contents

	<b>Preface .....</b>	<b>iii</b>
<b>1</b>	<b>Introduction .....</b>	<b>1-1</b>
1.1	Overview .....	1-1
1.2	Schematic diagram of the function packages in the SIMOTION device.....	1-2
1.3	Form of delivery .....	1-3
<b>2</b>	<b>SIMOTION IT OPC XML DA server .....</b>	<b>2-1</b>
2.1	Overview .....	2-1
2.2	Comparison of OPC XML DA / SIMATIC NET OPC DA .....	2-3
2.3	Schematic diagram at the design stage.....	2-4
2.4	Schematic diagram at runtime .....	2-5
2.5	Installation .....	2-6
2.5.1	Hardware and software requirements at the design stage .....	2-6
2.5.2	Hardware and software requirements at runtime.....	2-7
2.5.3	Configuring SIMOTION device interface at runtime .....	2-7
2.6	Making unit variables available .....	2-8
2.7	Example of a client application .....	2-10
2.8	SIMOTION IT OPC XML DA server interface.....	2-13
2.8.1	Overview .....	2-13
2.8.2	Methods which can be called synchronously.....	2-13
2.8.3	Methods which can be called asynchronously.....	2-15
2.8.4	Access to variables .....	2-15
<b>3</b>	<b>Trace Interface via SOAP .....</b>	<b>3-1</b>
3.1	Trace Interface via SOAP .....	3-1
3.2	Procedure/terms .....	3-2
3.3	Error handling.....	3-4
3.4	Basics of subscriptions .....	3-4
3.5	Interface .....	3-6
3.5.1	Global definitions .....	3-6
3.5.2	Methods .....	3-7
3.5.3	Trace Interface via SOAP .....	3-9
3.5.4	Subscriptions .....	3-11
<b>4</b>	<b>SIMOTION IT DIAG (standard diagnostics pages) .....</b>	<b>4-1</b>
4.1	Overview .....	4-1
4.2	Installation .....	4-2
4.2.1	Hardware and software requirements.....	4-2
4.2.2	Configuring the SIMOTION device interface .....	4-2

4.2.3	WEBCFG.....	4-3
4.2.4	Log-in administration.....	4-3
4.3	Standard diagnostics pages.....	4-5
4.3.1	Requirements for displaying the standard diagnostics pages .....	4-5
4.3.2	Home page.....	4-6
4.3.3	Device Info .....	4-7
4.3.4	Diagnostics.....	4-8
4.3.5	Task runtime .....	4-9
4.3.6	Diag Buffer .....	4-10
4.3.7	Mem Pool .....	4-11
4.3.8	Active Traces .....	4-12
4.3.9	IP Config.....	4-13
4.3.10	Settings .....	4-14
4.3.11	Files.....	4-17
4.4	User-defined HTML pages.....	4-21
4.4.1	Integration in standard diagnostics pages .....	4-21
4.4.2	Server Side Includes (SSI).....	4-21
4.4.3	User-defined Home page.....	4-22
<b>5</b>	<b>Variables providers .....</b>	<b>5-1</b>
5.1	Overview .....	5-1
5.2	SIMOTION.....	5-2
5.3	SIMOTION diagnostics .....	5-4
5.3.1	Introduction .....	5-4
5.3.2	DeviceInfo group .....	5-5
5.3.3	ComplInfo group .....	5-6
5.3.4	CPUload group .....	5-8
5.3.5	MemoryLoad group .....	5-8
5.3.6	TaskRT group .....	5-9
5.3.7	DiagBuffer group .....	5-11
5.3.8	ActiveTraces group .....	5-13
5.3.9	Comparison with the device diagnostics of SIMOTION SCOUT .....	5-14
5.4	MemPool .....	5-16
<b>6</b>	<b>Secure Socket Layer .....</b>	<b>6-1</b>
6.1	Key files.....	6-2
6.2	Creating key files.....	6-3
6.2.1	Install the development environment .....	6-4
6.2.2	Generating the OpenSSL tool.....	6-5
6.2.3	Modifying the OpenSSL configuration .....	6-6
6.2.4	Create a Certification Authority (CA).....	6-7
6.2.5	Creating a server certificate request.....	6-8
6.2.6	Sign the server certificate request .....	6-8
6.2.7	Generate the SIMOTION files .....	6-9
6.2.8	Importing the certificate to the browser.....	6-10
6.3	Transferring the key files to SIMOTION .....	6-11
<b>7</b>	<b>File access via FTP .....</b>	<b>7-1</b>
<b>A</b>	<b>Appendix.....</b>	<b>A-1</b>
A.1	List of abbreviations .....	A-1
	<b>Index.....</b>	<b>Index-1</b>

**Tables**

Table 2-1	Basic procedure for accessing data.....	2-3
Table 2-2	Hardware requirements at the design stage.....	2-6
Table 2-3	Hardware requirements at runtime .....	2-7
Table 2-4	Configuring the interface.....	2-7
Table 4-1	Hardware Requirements .....	4-2
Table 4-2	Configuring the interface.....	4-2
Table 5-1	Variables of the DeviceInfo group.....	5-5
Table 5-2	Variables of the DeviceInfo group.....	5-5
Table 5-3	Variables of the ComplInfo group .....	5-6
Table 5-4	Variables of the ComplInfo group .....	5-6
Table 5-5	Variables of the ComplInfo group .....	5-6
Table 5-6	Variables of the ComplInfo group .....	5-7
Table 5-7	Variables of the ComplInfo group .....	5-7
Table 5-8	Variables of the CPUload group .....	5-8
Table 5-9	Variables of the MemoryLoad group.....	5-8
Table 5-10	Variables of the TaskRT group .....	5-9
Table 5-11	Task names.....	5-9
Table 5-12	Variables of the TaskRT group .....	5-10
Table 5-13	Variables of the TaskRT group .....	5-10
Table 5-14	Variables of the DiagBuffer group.....	5-12
Table 5-15	Variables of the DiagBuffer group - direct access .....	5-12
Table 5-16	Variables of the ActiveTraces group.....	5-13
Table 5-17	General variables.....	5-16
Table 5-18	Memory pool-specific variables.....	5-16





# Introduction

## 1.1 Overview

### Function packages

"SIMOTION IT Ethernet-based HMI and diagnostic functions" contains the following function packages (IT = Information Technology):

- SIMOTION IT OPC XML DA server

These function packages enable the connection of applications to a controller via Internet technology and access to data and operating modes on SIMOTION devices. Commands are transmitted via the SOAP (Simple Object Access Protocol) communication protocol.

- Trace Interface via SOAP

This function package enables tracing of variables from the SIMOTION Variable Provider management.

- SIMOTION IT DIAG

This function package enables a direct diagnosis of the SIMOTION devices. Access is with a standard browser (e.g. Internet Explorer) via the IP address of the SIMOTION device. You can use the standard diagnostic pages or your own HTML pages for access.

## 1.2 Schematic diagram of the function packages in the SIMOTION device

### Representation of the function packages

The following figure is a schematic diagram of the function packages in the SIMOTION device.

The data of the SIMOTION device can be reached via the "Variables providers".

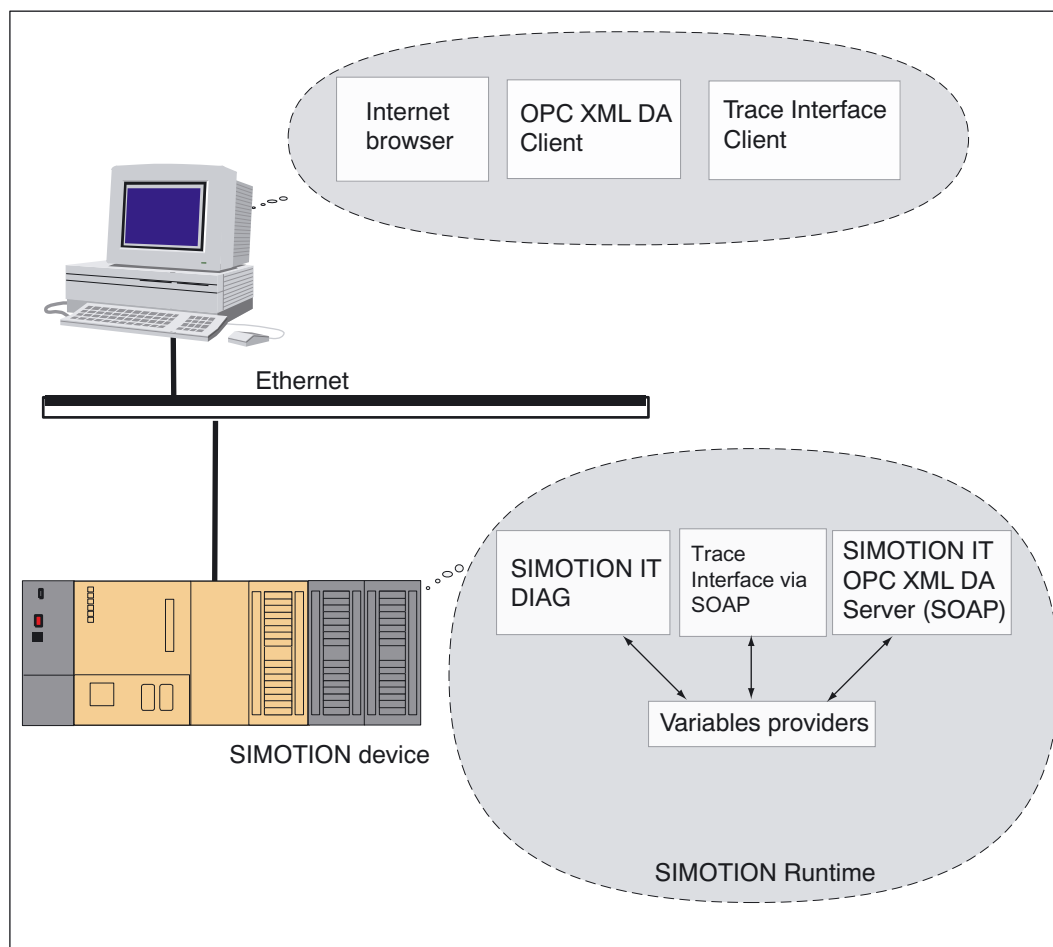


Figure 1-1 Function packages

## 1.3 Form of delivery

### Form of delivery

---

#### **Note**

As standard, SIMOTION devices (D, P, C) are delivered with firmware without "SIMOTION IT Ethernet-based HMI and diagnostic functions".

---

SIMOTION firmware with Ethernet-based HMI and diagnostic functions (for C, P, D, no runtime license) is provided on a separate CD.

Order no.: 6AU1 800-0KA40-0AA0

### CD contents

The CD contains the following components:

- Documentation, "SIMOTION IT Ethernet-based HMI and diagnostic functions" product information
- Documentation, "OPC XML DA R1.0 specification"
- Server configuration file (WEBCFG.XML)
- Minimum sample client for OPC XML DA V1.0 (see "Example for a client application")
- OPC XML-DA license contains Trace Interface via SOAP function
- Minimum sample client for Trace Interface via SOAP
- Firmware with the "Ethernet-based HMI and diagnostic functions" described in this document, for all current SIMOTION hardware versions (C, P, D)

### Runtime licenses

You need the following licenses depending on which functionality you use:

- SIMOTION IT DIAG:  
No.: 6AU1 820-8BA20-0AB0
- SIMOTION IT OPC XML DA:  
No.: 6AU1 820-8BA20-0AB0

---

#### **Note**

The functionality must also be activated in the SIMOTION project in SCOUT in the CPU hardware configuration. You can activate the "OPC XML/diagnostics pages" function in the "Extended Ethernet" tab in the object properties of the CPU hardware configuration.

---



# SIMOTION IT OPC XML DA server

## 2.1 Overview

SIMOTION IT OPC XML DA server enables access via Ethernet to data and operating modes of the SIMOTION device.

### What is OPC XML DA?

OPC is the abbreviation for OLE for Process Control and describes a standard interface for communication in automation systems.

With OPC XML DA, it is possible to communicate with a controller using Ethernet-based standard message frames. Commands are transmitted via the SOAP (Simple Object Access Protocol) communication protocol.

The interface is defined in a configuration file using a description language based on XML vocabulary. It defines the format of the request and response message streams with which function calls are transmitted (see OPC XML DA R1.0 specification).

This interface can only be used by client applications.

The following figure illustrates a client made available by the OPC Foundation.

The client enables browsing via the system, interface, I/O, and global device variables.

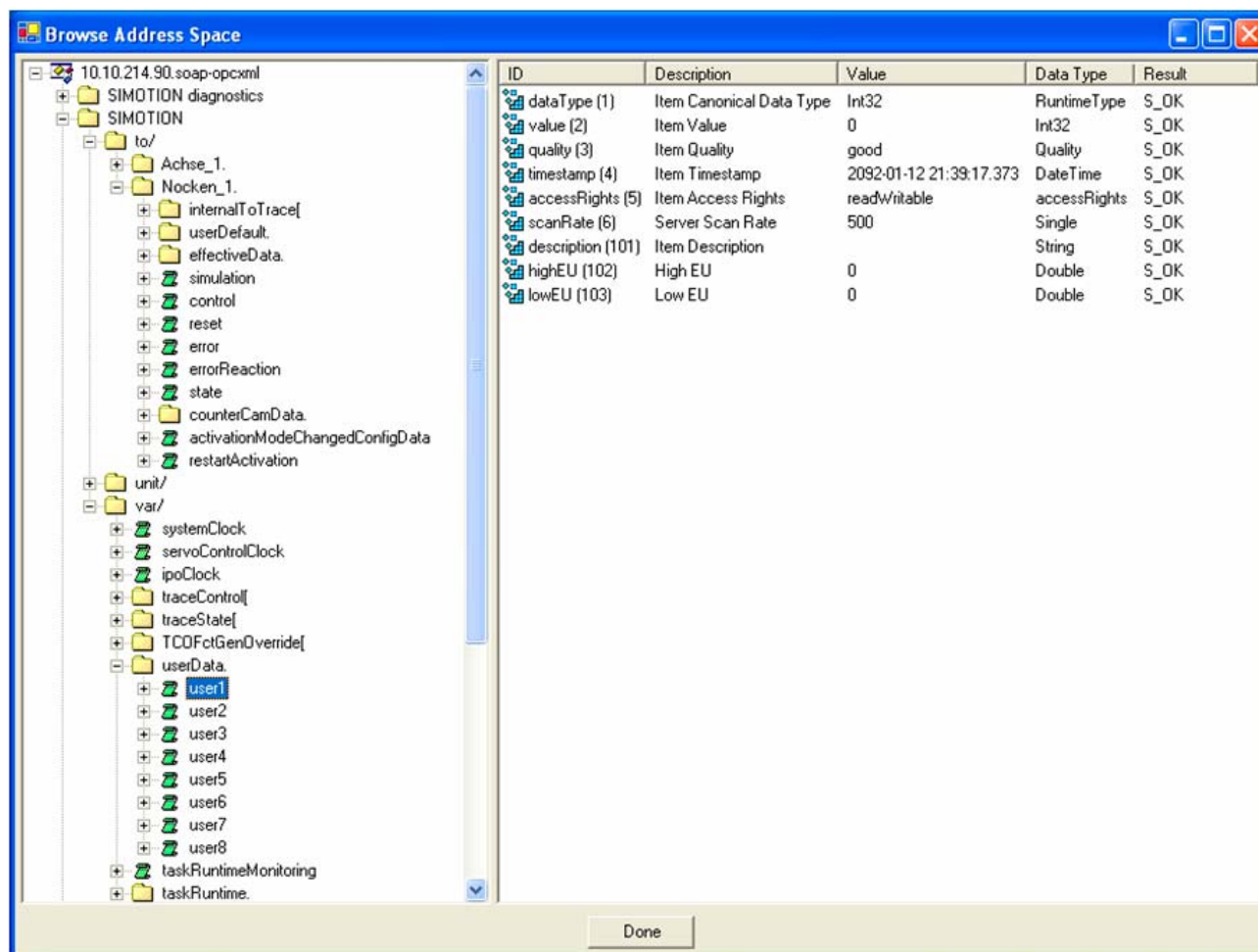


Figure 2-1 Client

## Purpose and benefits

The purpose and benefits of the SIMOTION IT OPC XML DA server are as follows:

- Addressable via Ethernet interface in accordance with the OPC XML DA V1.0 standard.
- The server can be addressed by any client application which conforms to the OPC XML DA V1.0 standard, regardless of its operating system (e.g. Linux).

## What previous knowledge is required?

In order to understand the SIMOTION IT OPC XML DA server described in this chapter, it is assumed that the user is familiar with OPC XML DA terminology (see OPC XML DA R1.0 specification).

## 2.2 Comparison of OPC XML DA / SIMATIC NET OPC DA

### Comparison

The "SIMATIC NET OPC Server for SIMOTION" product exists in addition to the SIMOTION IT OPC XML DA server. This package also allows access to data and operating modes of the SIMOTION device via SIMATIC NET OPC DA.

The following table compares the two packages and describes the basic procedure:

Table 2-1 Basic procedure for accessing data

SIMOTION IT OPC XML DA	SIMATIC NET OPC DA
No configuration (OPC export) with SCOUT required. Program variable can be activated via a switch.	OPC export with SIMOTION SCOUT required, which has to be repeated for every project change.
Symbols are interpreted in the SIMOTION device; communication in text format (XML).	Symbols are interpreted during OPC export and stored in the OPC server on the Windows system in binary format; binary -> higher data throughput.
At present only SIMOTION with OPC XML DA. Access to S7 devices not possible at present.	Simultaneous access to SIMOTION and S7 devices.
Client can run on any operating system.	Based on Windows COM/DCOM technology; client and server can only run on Windows operating systems.
Communication with standard protocols (TCP/IP, XML, SOAP), no manufacturer-specific (SIEMENS) tools, drivers required on the client system.	S7 protocol used for communication, appropriate manufacturer-specific drivers required on the client.
Communication only possible via Ethernet.	Communication via PROFIBUS/MPI and Ethernet is possible.
Direct addressing via firewalls is possible.	Generally, DCOM communication not released for firewalls.

## 2.3 Schematic diagram at the design stage

### Example arrangement

The figure below shows an arrangement example of the relevant software for the creation of a client application on a PC. The PC and the SIMOTION device are networked via Ethernet.

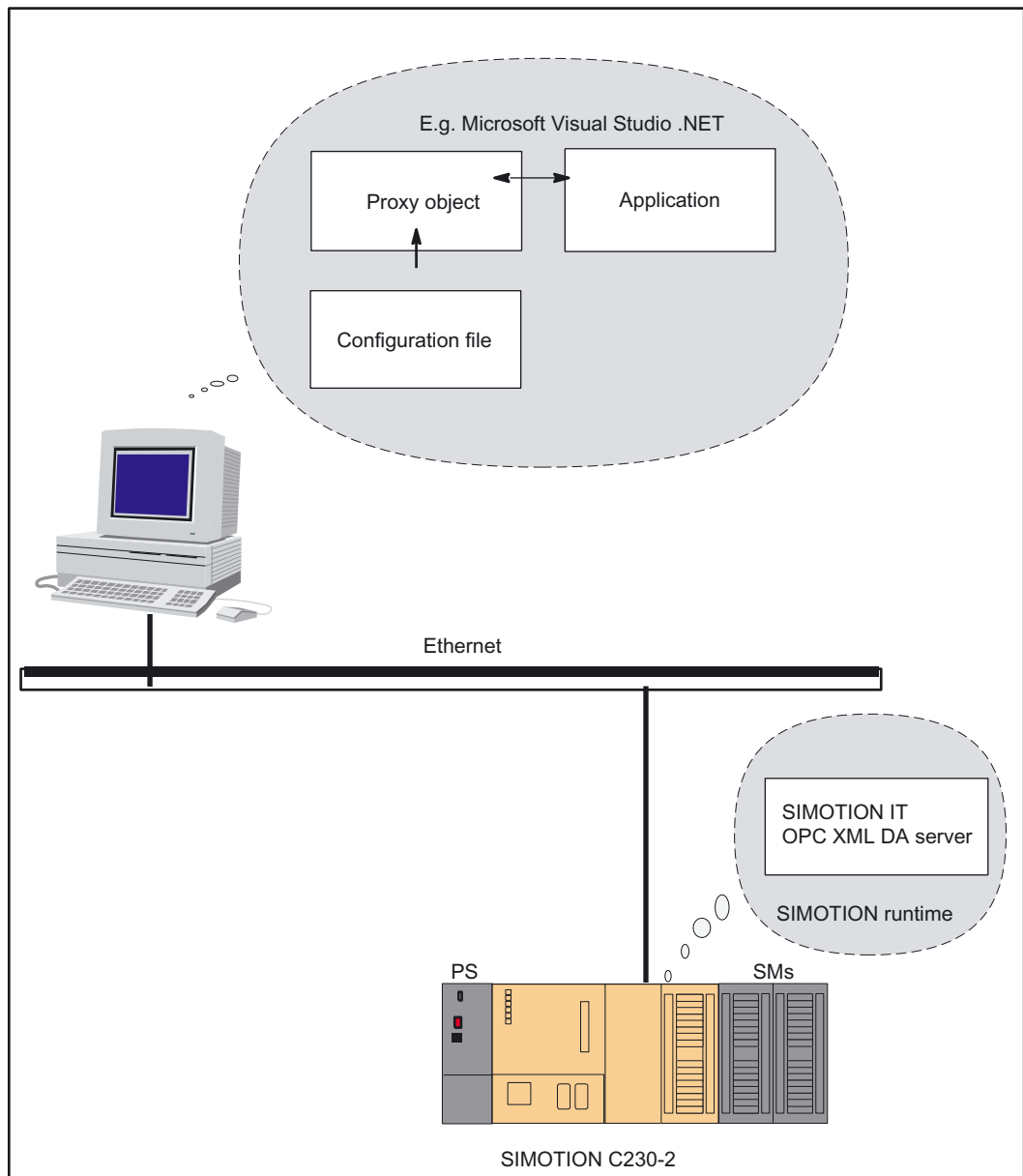


Figure 2-2 Design stage overview (example)



## 2.4 Schematic diagram at runtime

### Example arrangement

The figure below shows an arrangement example of the client application on a PC and the OPC XML DA server on the SIMOTION device at runtime. Both are networked via the Ethernet.

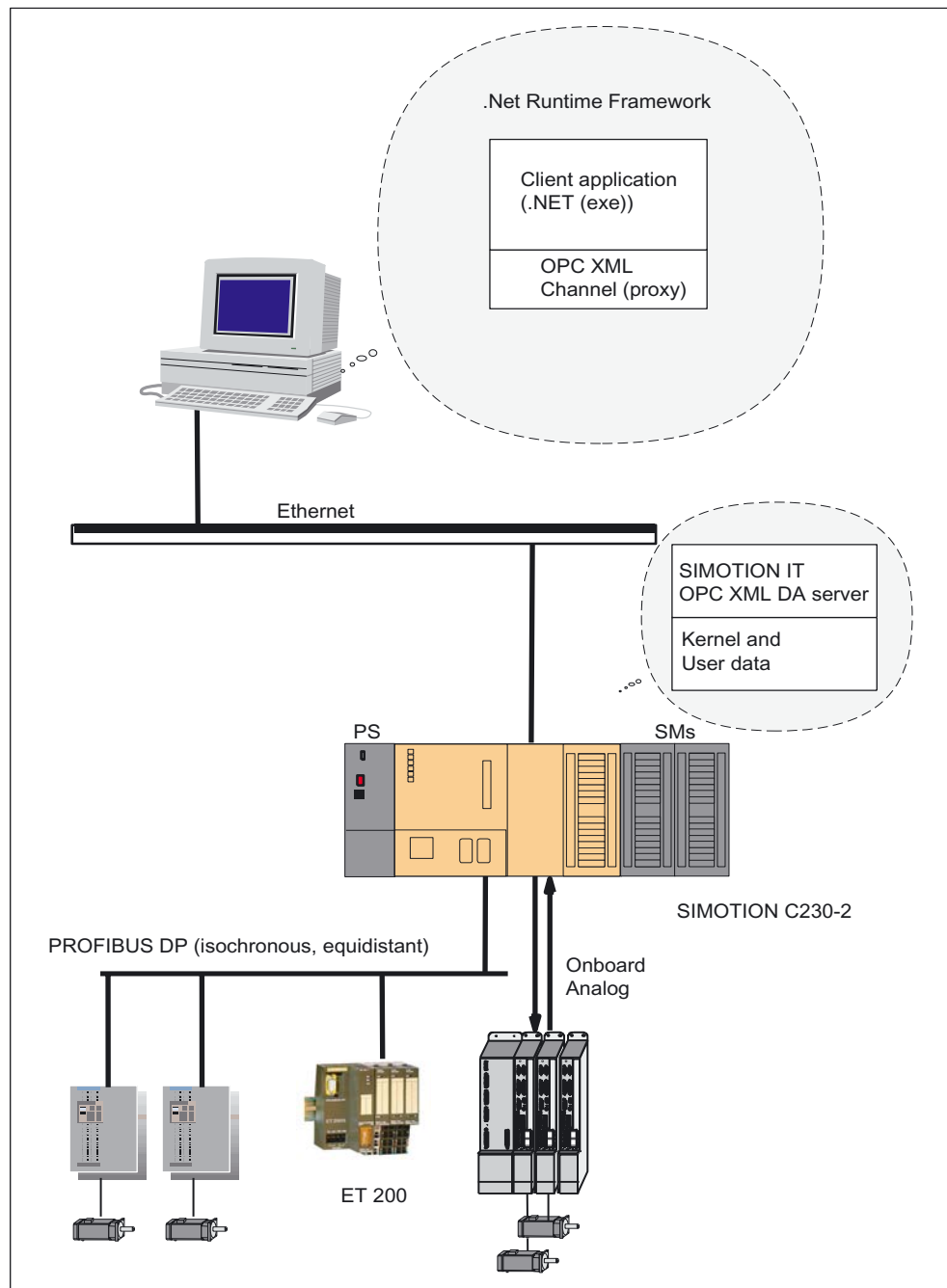


Figure 2-3 Overview at runtime (example)

## 2.5 Installation

### 2.5.1 Hardware and software requirements at the design stage

#### Hardware requirements at the design stage

---

**Note**

You can freely select the programming environment. The following requirements are an example for Microsoft Visual Studio .NET, but they are not binding.

---

Table 2-2 Hardware requirements at the design stage

Removal	Minimum requirement
Processor	Intel Pentium III or compatible, 800 MHz
Main memory	128 MB RAM

#### Software requirements at the design stage

---

**Note**

You can freely select the programming environment. The following requirements are an example for Microsoft Visual Studio .NET, but they are not binding.

---

- Microsoft Visual Studio .NET:  
<http://msdn.microsoft.com/vstudio/>  
<http://msdn.microsoft.com/net/>
- Configuration file (WSDL), in accordance with OPC XML R1.0 specification.

## 2.5.2 Hardware and software requirements at runtime

### Hardware requirements at runtime

Depending on the application you have set up, the hardware may need to meet the following requirements:

Table 2-3 Hardware requirements at runtime

Removal	Minimum requirements
Processor	Intel Pentium II, 266 MHz
Main memory	128 MB RAM
Network cards	Ethernet
SIMOTION device	E.g. SIMOTION C230-2 with micro memory card

### Software requirements at runtime

- .NET Runtime Framework: <http://msdn.microsoft.com/netframework/>
- SIMOTION device with the firmware (not standard firmware) installed from the CD
- Licensed SIMOTION IT OPC XML DA server (order no.: 6AU1820-8BB20-0AB0)

## 2.5.3 Configuring SIMOTION device interface at runtime

### Configuring the interface

In order to establish a connection between a PC and a SIMOTION device when the system is running, you must carry out the following steps for the configuration of the Ethernet interface:

Table 2-4 Configuring the interface

Step	Procedure
1	The functionality must be activated in the SIMOTION project during the hardware configuration of the CPU via the "Extended Ethernet" properties in the "OPC XML/diagnostic pages" function.
2	You must acquire a license for the SIMOTION IT OPC XML DA server.
3	The server address must be known. The preset IP address for the CPU (IP address 169.254.11.22.) can be found in the C2302 manual (order no. 6AU1 900-0AB21-0AA0). You can use HW Config to adapt this factory setting to local conditions and then download it to the SIMOTION device.

## 2.6 Making unit variables available

To make variables available on the SIMOTION IT OPC XML DA server, you have to declare them as VAR\_GLOBAL and activate them for OPC XML.

### Declaring unit variables in the interface

In the declaration table, you define the data type for each variable. Only variables declared as VAR\_GLOBAL are available in OPC XML.

The following figure shows an example of unit variable declarations in an MCC program.

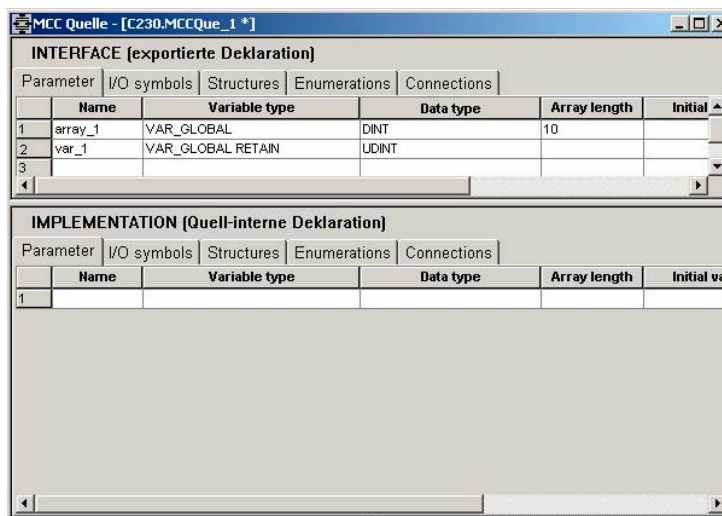


Figure 2-4 Declaring global variables

## Enable OPC-XML

To activate the variables for OPC XML:

1. Open the "Program properties" dialog box
2. Open the "Compiler" tab
3. Activate "Enable OPC-XML"

The following figure shows the activation of a unit variable from an MCC program.

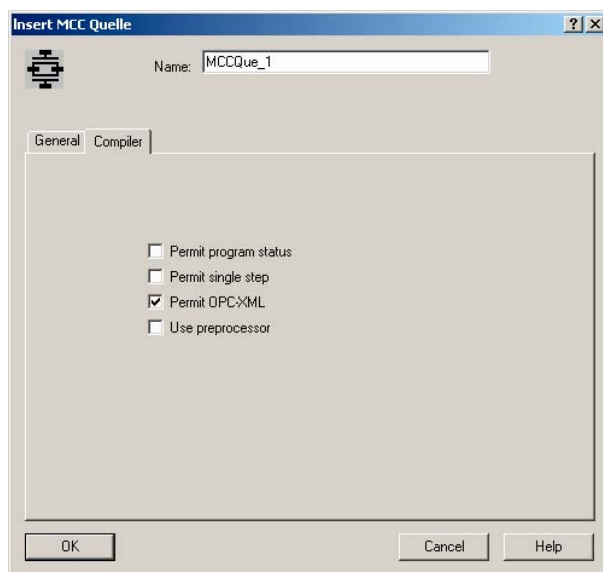


Figure 2-5 Making variables available for OPC XML

---

### Note

The OPC XML activation also applies to variables in LAD/FBD and ST programs. To make variables available for OPC XML in an ST program, they have to be defined in a global variable block. This block can appear in the interface or implementation section.

---

## 2.7 Example of a client application

### Example

The following example describes the most important steps for the "Read" method with the Microsoft Visual Studio .NET2003 tool.

The OPC client should execute the following application:

The application example displays a "Read" button in a dialog box. When the button is activated, the client connects to the SIMOTION IT OPC XML DA server and reads a variable. The result is displayed in the output field of the dialog box.

The dialog box of the application example is shown in the following figure:

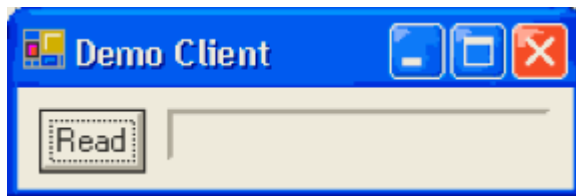


Figure 2-6 Demo client

### Programming steps

The following programming steps are needed:

1. Create a new project with Microsoft Visual Studio .NET and import the WSDL file as the interface description ("Add Web Reference" menu).
2. Create a dialog box with a text field and a "Read" button.
3. Enter the names specified for the reference, e.g. "OPCXMLServer", in the program (using `DemoClient.OPCXMLServer`).
4. Declare the server URL in the program as follows:  
`http://<IP address>/soap/opcxml`  
Enter the IP address of your SIMOTION device in place of `<IP address>`.
5. Instantiate the server proxy object in accordance with the code example and provide the call-up with the required parameters.
6. The required data is returned.

### Program section

```
using DemoClient.OPCXMLServer;  
private void ReadButton_Click(object sender,  
System.EventArgs e)  
{  
    Service C230_2_Server = new Service();
```

```
RequestOptions ReadOptions = new RequestOptions();
ReadOptions.ClientRequestHandle = "";
ReadOptions.LocaleID = "DE-AT";
ReadOptions.RequestDeadlineSpecified = false;
ReadOptions.ReturnDiagnosticInfo = false;
ReadOptions.ReturnErrorText = false;
ReadOptions.ReturnItemName = false;
ReadOptions.ReturnItemPath = false;
ReadOptions.ReturnItemTime = false;

RequestOptions ReadOptions = new RequestOptions();
ReadOptions.ClientRequestHandle = "";
ReadOptions.LocaleID = "DE-AT";
ReadOptions.RequestDeadlineSpecified = false;
ReadOptions.ReturnDiagnosticInfo = false;
ReadOptions.ReturnErrorText = false;
ReadOptions.ReturnItemName = false;
ReadOptions.ReturnItemPath = false;
ReadOptions.ReturnItemTime = false;

ReadRequestItemList ReadItemList = new
ReadRequestItemList();
ReadRequestItem[] ReadItemArray = new
ReadRequestItem[1];
ReadRequestItem ReadItem = new ReadRequestItem();

ReadItem.ItemPath = "SIMOTION";
ReadItem.ItemName = "var/userData.user5";
ReadItemArray[0] = ReadItem;
ReadItemList.Items = ReadItemArray;

ReplyItemList ReadReplyList;
OPCError[] ReadErrorList;

C230_2_Server.Url = "http://simotion/soap/opcxml";
System.Net.ICredentials myCredentials = new
System.Net.NetworkCredential("simotion","simotion");
C230_2_Server.Credentials = myCredentials;
C230_2_Server.PreAuthenticate = true;
System.Net.ServicePointManager.Expect100Continue =
false;

C230_2_Server.Read(ReadOptions,ReadItemList,out
ReadReplyList,out ReadErrorList);

if ((ReadReplyList.items[0] != null) &&
```

```
(ReadReplyList.Items[0].Value != null) &&
(ReadReplyList.Items[0].Value.GetType().Name !=
"XmlNode[]"))
Output.Text =
ReadReplyList.Items[0].Value.ToString();
else Output.Text = "<Error>";
}
```

---

**Note**

If the client application has been created with Microsoft Visual Studio .NET, then an "application configuration file" must be added to the solution with the "Add New Item" menu. This text file is required to deactivate the debug information in the SOAP message frame.

---

The "application configuration file" must have the following contents:

**Application configuration file**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.diagnostics>
    <switches>
      <add name="Remote.Disable" value="1"/>
    </switches>
  </system.diagnostics>
</configuration>
```



## **2.8 SIMOTION IT OPC XML DA server interface**

### **2.8.1 Overview**

#### **Introduction**

This section describes the methods you can run across the interface to the OPC XML DA V1.0 server. The server itself is integrated in the SIMOTION device and must be enabled by a license.

This is just a brief overview. A detailed description of these methods can be found in the document entitled "OPC XML DA Specification R1.0" from the OPC Foundation.

You can find the current detailed interface description on the home page of the OPC Foundation: <http://www.opcfoundation.org/webservices/XMLDA/V1.0/>

### **2.8.2 Methods which can be called synchronously**

The SIMOTION IT OPC XML DA server provides the following methods, which can be called synchronously, under the "OpcXmlDaService" type:

#### **Description of methods**

##### **Browse**

The "Browse" method allows you to navigate through the available variables.

##### **GetProperties**

The "GetProperties" method can query the settings for a specific variable (e.g. access rights, time stamp, data type).

##### **GetStatus**

The "GetStatus" method supplies information about the server status, the program version and the supported interface version.

##### **Read**

The "Read" method reads out variable lists.

## Subscribe

The "Subscribe" method passes a list of variable names and receives a handle for the subscription. This handle can be used in the SubscriptionPolledRefresh method to poll the values of the previously defined variables again.

## SubscriptionPolledRefresh

The "SubscriptionPolledRefresh" method returns the values of the variables written beforehand using the Subscribe method. The handle specifying the subscription is used as a parameter.

The "Holdtime" parameter defines the earliest possible response time. This limits the frequency of data transmission.

The "ReturnAllterms" parameter determines how the "WaitTime" parameter is used.

- True

"WaitTime" is ignored, all requested values are returned immediately.

- False

For the period set in the "WaitTime" parameter, the server checks whether one of the requested values has changed since the last call.

If the specified time expires without a value having been changed, an empty response is returned.

If values change during the specified time, the changed values are returned immediately and the polling ended.

## SubscriptionCancel

The "SubscriptionCancel" method cancels the subscription and returns the subscription handle.

Which subscription is to be canceled, must be specified at the call.

If an asynchronous call form is used, the client is informed later of which subscription has been canceled, via a client handle.

---

### Note

Once the subscription has been canceled, the subscription handle ceases to be valid for the client.

---

## Write

The "Write" method writes variable lists.

## 2.8.3 Methods which can be called asynchronously

### Methods which can be called asynchronously

Methods which can be called asynchronously are more complex than methods which can be called synchronously, but offer the same functionality. These methods are described in detail in the configuration file.

---

#### **Note**

In addition to methods which can be called synchronously, the SIMOTION IT OPC XML DA also provides methods which can be called asynchronously.

---

## 2.8.4 Access to variables

### Variable access using methods

Variables can be accessed via the methods which can be called synchronously and asynchronously.

---

#### **Note**

The user must map the following variable types by means of user-defined program variables:

- Process image
  - I/O variables
  - Global device variables
- 

---

#### **Note**

Before you can access unit variables, you need to declare them in the interface in SCOUT and activate them for OPC XML (see Section "Making unit variables available").

---

---

#### **Note**

Information for accessing unit variables in an MCC program can be found in the SIMOTION MCC programming manual.

Information for access in an ST program can be found in the SIMOTION ST programming manual.

---



# Trace Interface via SOAP

## Introduction

The SOAP-based service provides a trace service option.

## Trace-Service

The "Trace Interface via SOAP" function package enables variable values to be written to a buffer. The values are packed in files and can be retrieved asynchronously via a HTTP request.

This interface can only be used by client applications. The client enables the time characteristic of variables to be traced.

A WSDL file is available for creating the application.

---

### Note

The "Trace Interface via SOAP" function package contains a user example entitled "WebTrace Demo Client". This client supports:

- Setting up a trace
  - Starting the trace
  - Retrieving data
  - Deleting data
- 

## 3.1 Trace Interface via SOAP

### Introduction

When working with a trace, the trace can assume various states. The following graphic shows the possible states and transitions. The methods named are described in chapter "Trace interface".

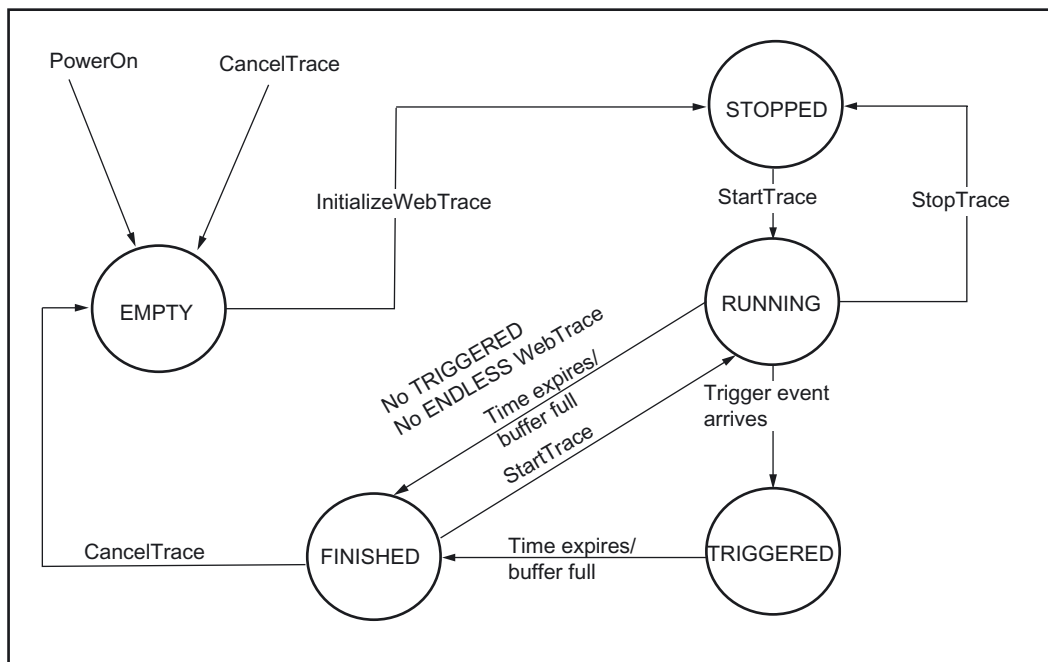


Figure 3-1 WebTrace

## States

After a trace has been created with "InitializeWebTrace", it is in the STOPPED state. With StartTrace, it starts up and writes the desired data to the buffer. Accordingly, it can be stopped again with "StopTrace". After the start, the trace switches to RUNNING state. If the time specified in the call has expired, the trace assumes the FINISHED state. A trace can be deleted at any time with "CancelTrace" in order to create a new trace, for example.

## 3.2 Procedure/terms

### HTTP methods - data exchange

The trace data are stored as compiled data on the RAM disk using the ReadData method. These data must be retrieved via ordinary HTTP requests.

#### Note

The data are not deleted by retrieval alone! To prevent the RAM disk from overflowing, an HTTP DELETE call to this URL must follow an HTTP GET call. (Reason: The UseCase is considered in which a client may have to request the same trace data more than once, e.g., to compare traces that have already been executed.) These temporary data are completely deleted only after a CancelTrace operation, regardless of whether they have already been retrieved or not.

## TRIGGERED

The trace offers a triggering option. Depending on the trigger method, different constants or variable symbols must be specified for this. The trace starts with:

- A rising edge (RE),  
if the variable exceeds the value of a constant.
- A falling edge (FE),  
if the variable falls below the value of a constant.
- Within a tolerance band (WIB),  
if the variable lies between two constants.
- Outside of a tolerance band (OOB),  
if the variable lies outside of a tolerance band.
- Bit mask has value (BHV),  
if the variable has a specified value after masking with a constant.

If the trace is set up in TRIGGERED mode, a trigger condition as described below must be issued. This trigger acts as a SingleShot. However, the MatchCountTriggerPoint parameter can be used to set the trigger for repeated occurrence (e.g., five: Trigger only the fifth time the trigger condition appears)

In this case, the trace takes place only after the trigger. The data are recorded for the duration specified during setup.

## IMMEDIATE / ENDLESS

The counterpart to the TRIGGERED Trace is the IMMEDIATE Trace, which begins the trace immediately after the StartTrace call has occurred. In this case as well, the data are recorded for the duration specified during setup.

The ENDLESS Trace uses a ring buffer trace. Trigger conditions are not evaluated. ENDLESS Trace starts as soon as the StartTrace event arrives. However, it is terminated only when StopTrace is called explicitly. The size of the ring buffer must also be specified using the duration for the initialization call. Thus, an appropriate value must be found that uses fewer resources, but is sufficient to retrieve data in a timely manner via HTTP.

The size of the ring buffer (B) is determined from the number of variables (N), their size (S), the transferred time duration (t) and the cycle clock (T) in which they are traced.

$$B = t/T * \sum_{i=0}^{n-1} S^i$$

Within the transferred time duration, the buffer must be discharged at least once by calling the "readData" function in order to prevent the oldest trace data from being overwritten each time.

The maximum size of the buffer is 256 Kbytes. If the buffer is larger, the duration of the trace is adapted accordingly.

### 3.3 Error handling

All implemented methods of the TVS (Trace Via SOAP) supply either the requested data or status information, or a SOAP\_FAULT. This behavior enables the use of the SoapFaultError in the .NET framework. The Try-Catch mechanism enables convenient error handling.

### 3.4 Basics of subscriptions

#### Introduction

"GetStatus" must be called in order to query the status of a trace. The fastest possible detection of a status change requires extremely frequent polling, which places an unnecessary load on the CPU in the controller and causes heavy traffic on the network.

To optimize this operation, OPCXML provides so-called subscriptions. With subscriptions, a query does not receive a response until a specific variable changes or a timeout occurs (variable does not change). Thus, the connection is kept open without causing traffic. As soon as relevant data are available for the client, these data are sent to the client.

This mechanism is also used by the TVS-WS. However, in this case, only the status of the trace object is checked, as this is the only valuable information in this environment.

As soon as the status changes (e.g. RUNNING -> FINISHED), the clients that issued the query receive a corresponding response. In essence, any number of clients is possible (as long as there are sufficient resources).

#### Operational sequence

The operational sequence of a subscription is as follows:



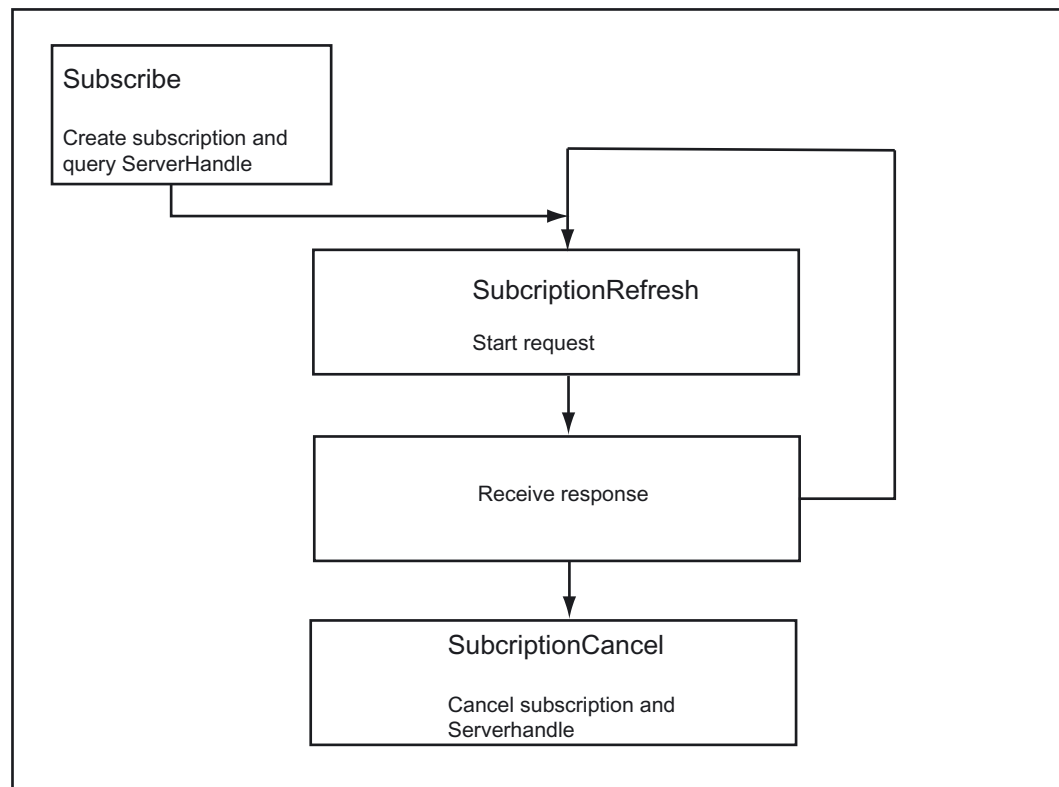


Figure 3-2 Subscription

First, a subscription must be created. It is answered with a unique ServerHandle, which is required for further communication.

SubscriptionRefresh can be called as often as necessary to start a new query. This request receives two time specifications in milliseconds as parameters:

- HoldTime:  
This time indicates the minimum hold time for the response, irrespective of whether the status has changed.
- WaitTime:  
After the HoldTime, an additional specified period of time can be allowed to elapse while waiting for the status to change. If the status changes, the response is sent during this time. If the status does not change, then a message containing the current status is sent once the WaitTime expires.

The exact method calls are explained in the next section.

In general, this procedure is only useful for the WebTrace (to see when data must/can be retrieved); however, it also works with an IOTrace.

## 3.5 Interface

### 3.5.1 Global definitions

#### TraceStateEnum

Enumerator that indicates the status of the trace object.

##### Declaration:

```
public enum TraceStateEnum { EMPTY, STOPPED, RUNNING, FINISHED, TRIGGERED,
ERROR }
```

#### TraceDataCycleEnum

Enumerator that specifies the cycle clock in which the data are to be traced. It must be noted here that large traces may cause a layer overflow.

##### Declaration:

```
public enum TraceDataCycleEnum {
IPO1,
IPO2,
SERVO,
}
```

#### Structure VDSC

Structure that contains information about the traced variables. These are:

- The variable name in VarName
- The variable type in VarType in S7 notation (e.g. DINT or BYTE)
- The offset of the variable within the data stream (relative to the start of the IOContainer)

##### Declaration:

```
public class VDSC {
public string VarName;
public string VarType;
public System.UInt32 VarOffset;
}
```

### 3.5.2 Methods

#### StartTrace

The `StartTrace` method starts an initialized trace. The "No Trace available" SoapFault is returned when a trace has not yet been initialized. This is ignored (with a positive result) if the trace is already in progress.

##### Call:

```
public TVS_Client.TVS.StartTrace_Response StartTrace ( )
public class StartTrace_Response {
    public TraceStateEnum TraceState;
}
```

#### StopTrace

The `StopTrace` method stops a trace in progress. The "No Trace available" SoapFault is returned when a trace has not yet been initialized. This is ignored (with a positive result) if the trace has already stopped.

##### Call:

```
public TVS_Client1.TVS.StopTrace_Response StopTrace ( )
public class StopTrace_Response {
    public TraceStateEnum TraceState;
}
```

#### CancelTrace

The `CancelTrace` method deletes an active trace. The traces switches to EMPTY status, and all trace data are deleted. (Note: Data blocks of the Webtrace that have been requested but have not yet been retrieved are also deleted (see `WebTrace::ReadData()`))

The "No Trace available" SoapFault is returned when a trace has not yet been initialized.

##### Call:

```
public TVS_Client.TVSIIO.CancelTrace_Response CancelTrace ( )
public class CancelTrace_Response{
    public TraceStateEnum TraceState;
}
```

#### GetStatus

The `GetStatus` method returns the current status of the trace. When a trace object is deleted or has become invalid, then `TracelsValid` will contain "false". In this case, the trace must be deleted via `CancelTrace`.

### 3.5 Interface

**Call:**

```
public TVS_Client.TVSIO.GetStatus_Response GetStatus ( )  
public class GetStatus_Response {  
    public bool TracelsValid;  
    public TraceStateEnum TraceState;  
}
```

**ReadData**

The `readData` method saves trace data on the RAM disk and supplies the URLs of the files in the return value. These data can be retrieved from the client with an HTTP-GET request.

The "No Tracedata available" SoapFault is returned if no trace data are available.

**Call:**

```
public ReadData_Response ReadData()  
public class ReadData_Response {  
    public TraceStateEnum TraceState;  
    public string[] URL;  
}
```

### 3.5.3 Trace Interface via SOAP

#### Introduction

For easier interface readability, the methods provided here are specified in Microsoft Visual Studio C#.

#### Data types

- Structure

**Call:**

```
public class TriggerCondition {  
    public string Variable;  
    public string Constant1;  
    public string Constant2;  
    public TriggerOperationType Operation;  
    public System.UInt32 MatchCountTriggerPoint;  
}
```

Structure indicating the trigger of a trace. It contains the variable to be compared in symbolic names according to VarProvider notation. MatchCountTriggerPoint indicates how many times the trigger condition must apply before the trigger is activated. For this purpose, the comparison type:

**Call:**

```
public enum TriggerOperationType {  
    RE,  
    FE,  
    WIB,  
    OOB,  
    BHV  
}
```

Describes the trigger algorithm to be used:

- RE – Rising Edge – Rising edge (One constant required)
- FE – Falling Edge – (One constant required)
- WIB – Within Bound – (Two constants required)
- OOB – Out of Bounds – (Two constants required)
- BHV – Bitmask has values – (Two constants required)

- Enumerator, which determines the type of trace

**Call:**

```
public enum TraceStartTypeEnum {
    IMMEDIATE,
    ENDLESS,
    TRIGGERED,
}
```

**Methods**

- InitializeWebTrace

A trace is created with `InitializeWebTrace`. `VariablesToTrace` is the list of symbolic names according to `VarProvider` notation. `TraceDataCycle` determines the cycle clock in which the data are to be traced. `TraceStartType` determines the type of trace. `Duration` indicates the duration of the trace in milliseconds. With an endless trace, this parameter specifies the size of the ring buffer in milliseconds. `MatchCountTracePoint` determines the intervals at which data are to be recorded (e.g. 3: only every third value is saved).

`MatchCountTriggerPoint` determines how often the trigger specified in the trigger information must be encountered as positive before the trigger takes place and thus begins the trace. `Pretrigger` specifies the number of values to be recorded before the trigger ("history").

**Call:**

```
public TVS_Client1.TVS.InitializeWebTrace_Response InitializeWebTrace (
    string[] VariablesToTrace ,
    TVS_Client.TVS.TraceDataCycleEnum TraceDataCycle ,
    TVS_Client.TVS.TraceStartTypeEnum TraceStartType ,
    System.UInt32 MatchCountTriggerPoint ,
    System.UInt32 MatchCountTracePoint ,
    System.UInt32 Pretrigger ,
    System.UInt32 Duration ,
    TVS_Client.TVS.TriggerCondition TriggerInformation )
```

```
public class InitializeWebTrace_Response {
    public VDSC[] CurrentlyTracedVariables;
    public TraceStateEnum TraceState;
}
```

- ReadData

With ReadData, the TVS service is requested to read out the buffer of the trace and pack the data in temporary files. These can then be accessed via HTTP under the relative paths specified in URL. If the buffer is empty, the request is responded to with the "No Tracedata available" SoapFault. Currently, up to 8 compiled files with up to 8,192 trace points are supplied per request.

### 3.5.4 Subscriptions

#### Introduction

The methods for subscriptions are presented below.

#### Subscribe

A subscription is created with the Subscribe method. The response is a ServerHandle that can be used to uniquely identify a subscription operation. In addition, the current TraceStatus is also supplied.

#### Call:

```
public TVS_Client.TVS.Subscribe_Response Subscribe ( )
```

```
public class Subscribe_Response {
    public System.UInt32 ServerHandle;
    public TraceStateEnum TraceState;
}
```

#### SubscriptionRefresh

With SubscriptionRefresh, a new subscribe operation is started. The server response is received either after HoldTime (milliseconds) + WaitTime (milliseconds) if the status has not changed during this time. Or, the response is received (at the earliest) after expiration of the HoldTime and before expiration of the WaitTime if the status of the trace changes during the WaitTime. Thus, the response arrives after the HoldTime in any case.

In the response, StateChanged indicates whether the status has changed between the request and the response (true) or whether the TraceState status matches the status during the request (false – WaitTime expired).

#### Call:

```
public TVS_Client.TVS.SubscriptionRefresh_Response SubscriptionRefresh (
    System.UInt32 ServerHandle ,
    System.UInt32 WaitTime ,
```

System.UInt32 HoldTime )

```
public class SubscriptionRefresh_Response {  
    public bool StateChanged;  
    public TraceStateEnum TraceState;  
}
```

### SubscriptionCancel

With `SubscriptionCancel`, a subscription is canceled, and the resource is enabled. The response indicates whether the Cancel operation was successful. Any current `SubscriptionRefreshes` are cancelled and answered immediately.

#### Call:

```
public TVS_Client.TVS.SubscriptionCancel_Response SubscriptionCancel (  
    System.UInt32 ServerHandle )
```

```
public class SubscriptionCancel_Response {  
    public bool SubscriptionCanceled;  
}
```



## SIMOTION IT DIAG (standard diagnostics pages)

### 4.1 Overview

The SIMOTION device administers prefabricated standard diagnostics pages. These pages can be displayed using a generally available browser via Ethernet. You can also create your own HTML pages and integrate servicing and diagnostics information.

#### Purpose and benefits

The purpose and benefits of HTML diagnostics pages are as follows:

- Prefabricated diagnostics pages are available to the user for the direct diagnosis of the SIMOTION device.
- Service and diagnostics information of the device can be accessed without manufacturer-specific programs to assist in production monitoring or diagnostics.
- User-defined HTML pages can be integrated.

## 4.2 Installation

### 4.2.1 Hardware and software requirements

#### Hardware Requirements

Table 4-1 Hardware Requirements

Removal	Minimum requirements
Processor	Intel Pentium II, 266 MHz
Main memory	128 MB RAM
Network cards	Ethernet
SIMOTION device	E.g. SIMOTION C230-2 with micro memory card

#### Software requirements

- SIMOTION device with the firmware (not standard firmware) installed from the CD
- Browser, e.g. Microsoft Internet Explorer as of Version 5.0
- License for SIMOTION DIAG  
(Order No.: 6AU1 820-8BA20-0AB0)

### 4.2.2 Configuring the SIMOTION device interface

#### Configuration of the Ethernet interface

In order to establish a connection between the standard diagnostics pages and a SIMOTION device via a browser, the following steps must be carried out to configure the Ethernet interface:

Table 4-2 Configuring the interface

Step	Procedure
1	The functionality must be activated in the SIMOTION project in SCOUT in the CPU hardware configuration. You can activate the "OPC XML/diagnostics pages" function in the "Extended Ethernet" tab in the object properties of the CPU hardware configuration.
2	License must be acquired for SIMOTION IT DIAG.
3	If required, change USER NAME and PASSWORD in WEBCFG.xml.
4	If the browser is to address the standard diagnostics pages, it has to know the address of the SIMOTION device, e.g. <a href="http://169.254.11.22">http://169.254.11.22</a> . The preset IP address for the CPU (IP address 169.254.11.22.) can be found in the C230-2 manual (order no. 6AU1 900-0AB21-0AA0). You can use HW Config to adapt

Step	Procedure
	this factory setting to local conditions and then download it to the SIMOTION device.

### 4.2.3 WEBCFG

The WEBCFG.XML file is used to configure the web server. The file is subdivided into several different sections, e.g. server options and user database. WEBCFG.XML can be re-loaded at runtime. This action restarts the web server. The modified settings are available after the restart.

#### Settings

This document describes two sections in which you can make changes:

- User database section: controls access to the diagnostics pages (see "Log-in administration" section)
- Server options section: replaces the home page of the standard diagnostics with your own home page (see "User-defined home page" section).

The WEBCFG.XML file can be found either on the supplied CD in the 3\_Configuration directory (in the default state) or on the SIMOTION memory card in directory USER\SIMOTION\HMI CFG\.

---

#### Notice

Please change only the settings described in this document. Changes to WEBCFG.XML settings not described in this manual can lead to errors.

---

### 4.2.4 Log-in administration

#### Structure of the log-in administration

The log-in system is structured as follows:

- There are users.
- Each user has a password
- There are secure areas (secure groups, or realms)
- Each secure area has a group of users who are authorized for access.
- A user can have access to different secure areas.

The user database is in WEBCFG.xml and can only be changed by reloading WEBCFG.xml.

---

**Note**

Please note that if faulty changes are made in the file, this can result in a crash.

---

```
<USERDATABASE>
  <USER NAME="simotion" PASSWORD="simotion">
    <DESCRIPTION>Default User</DESCRIPTION>
    <GROUP NAME="Administrator"/>
  </USER>
</USERDATABASE>
```

With USER "simotion", the password is specified in plain text ("simotion"). USER "simotion" can only access the "Administrator" secure area.

The "Administrator" group has been permanently programmed for the update of WEBCFG.xml. Which users belong to this group, is specified in the USERDATABASE.

Additional secure areas can be adapted to the respective requirements in WEBCFG.xml.

- Access to the settings:  
<settings.htm SECUREGROUP="Administrator">
- Access to the "Files" directory:  
<DEFAPP SECUREGROUP="Administrator">
- Writing of variables in the HTML diagnostics pages:  
<VarApp SECUREGROUP="Administrator">
- Update of the project and firmware:  
<FWUpdtApp SECUREGROUP="Administrator">
- Access protection to the OPC XML DA service:  
<URL BASE="OpcXml" SECUREGROUP="Administrator"/>

---

**Note**

The default setting for USER NAME and PASSWORD is "simotion". You can change these default settings.

Only a basic authentication is possible for these plain text passwords.

---

## 4.3 Standard diagnostics pages

### 4.3.1 Requirements for displaying the standard diagnostics pages

#### Requirements

The standard diagnostics pages described in the following sections are displayed after:

- You have installed the SIMOTION firmware **with** SIMOTION IT Ethernet-based HMI and diagnostics functions on the SIMOTION device.
- You have activated the functionality in the SIMOTION project in SCOUT during the hardware configuration of the CPU via the "Extended Ethernet" properties in the "OPC XML / diagnostic pages" function.
- You have acquired a license for SIMOTION DIAG.
- Name and password from WEBCFG.xml are known (only true for the "Settings" pages).
- You have established a connection to the SIMOTION device with a browser via the SIMOTION IP address.

#### Example:

`http://mysimotion`

`http://169.254.11.22`

---

#### Note

HTTPS connections are supported in SIMOTION V3.2 and higher.

You will find further information in Section "Secure Socket Layer".

---

The current SIMOTION device type (C230-2/D435/P350) is displayed on every page. Navigation is carried out via the links in the top section or on the left-hand side.

---

#### Note

Before you can display unit variables on the diagnostics pages, you need to declare them in the interface in SCOUT and activate them for OPC XML (see Section "Making unit variables available").

---

### 4.3.2 Home page

#### Data of the SIMOTION device

The following current data of the SIMOTION device is displayed on the Home page:

Order Number	Order no. (MLFB) of the device
Revision Number	Hardware version
Licence Serial Number	The license key is tied to this serial number. Depending on the hardware, either the MMC, CF or ISO board serial number is displayed here.
User Version Firmware	SIMOTION kernel user version
Operating State	Operating mode of the SIMOTION device (RUN, STOP, STOPU)
Systemtime	Current time-of-day of the SIMOTION device
Active Traces	Number of active traces

You can find further information on the current device data on the "Device Info" page.

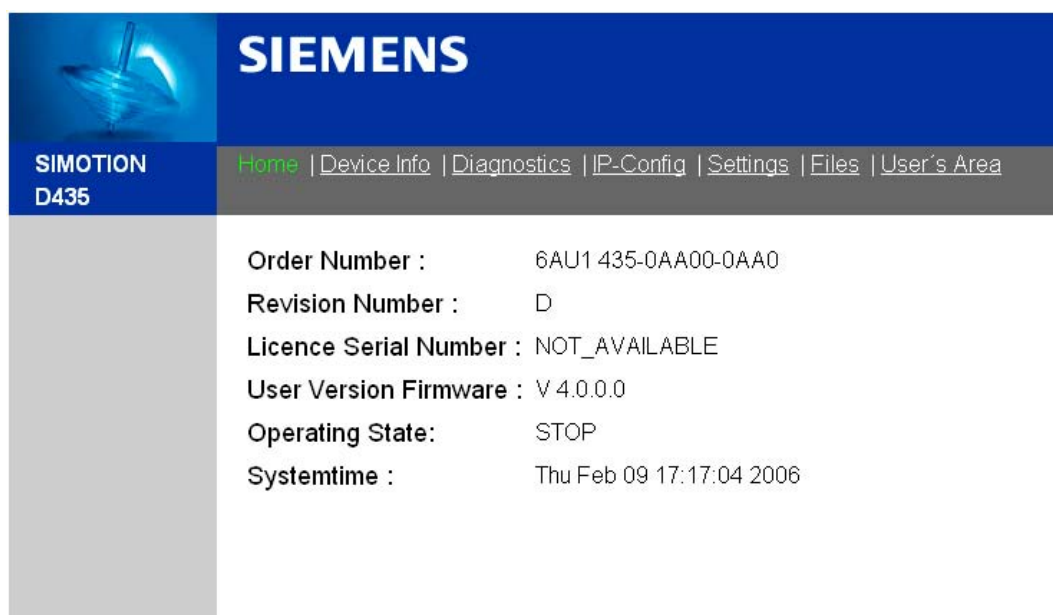


Figure 4-1 Home page

### 4.3.3 Device Info

#### Hardware and firmware information

The following current hardware and firmware information of the SIMOTION device is displayed on the "Device Info" page:

Manufacturer Name	Siemens AG
Order Number	Order no. (MLFB) of the device
Revision Number	Hardware version
Serial Number	Serial number of the SIMOTION device
User Version Firmware	SIMOTION kernel user version
Build Number	Internal version number
Additional Hardware	Installed components of the SIMOTION device including: Order no. (MLFB), serial no., revision no. firmware name, user version no., internal version no.
Technological Packages	Loaded technological packages including: Package name, user version no., internal version no.

**Device Info**

Manufacturer Name : [SIEMENS AG](#)  
Order Number : 6AU1 230-2AA00-0AA0  
Revision Number : A  
Serial Number : STN0000001  
User Version Firmware : V 3.1.0.0  
Build Number : V 30.0.0.0.Gmk\_30hkGHxG 6ntMD SSQV IFswE9x

**Additional Hardware**

MLFB	Serial-Nr.	Revision-Nr.	FW-Name	User-Ver.	Build-Nr.
6AU1700-0AA00-0AA0	3400190	01		V 0.0.0.0	V 0.0.0.0

**Technological Packages**

TP-Name	User-Ver.	Build-Nr.
tpcamming	V 3.1.0.0	V 30.0.0.27 GLZhs ACKZqpnGHxEatq 1Jjp9vHfQc

Figure 4-2 Device Info

### 4.3.4 Diagnostics

#### States of the SIMOTION device

The following states of the SIMOTION device are displayed on the "Diagnostics" page:

Systemtime	Current time-of-day of SIMOTION
Timezone	Current difference between the system time and GMT in minutes
CPU load by cyclic tasks	Processor time of servo and interpolator levels as a percentage of the total processor time
Memory Load	Size and allocation of the memory, the RAM disk, the memory card and the non-volatile memory in bytes and as a percentage
State	Current SIMOTION operating mode

Further details can be displayed using the links in the light-gray area.

**SIEMENS**

**SIMOTION D435**

[Home](#) | [Device Info](#) | [Diagnostics](#) | [IP-Config](#) | [Settings](#) | [Files](#) | [User's Area](#)

[Diagnostics](#)  
[Taskruntime](#)  
[Diag Buffer](#)  
[Mem Pool](#)  
[Active Traces](#)

## Diagnostics

**Systemtime:** Thu Feb 09 17:17:34 2006  
**Timezone:** GMT +60 min  
**CPU load by cyclic tasks:** 17%

**Memory Load:**

	Used Bytes	Size Bytes
RAM - Disk :	95232	13577216
RAM :	10476204	22544384
Memory Card :	442408960	523018240
Retentive Data :	1736	373240

**State:**

- ☒ DC5V
- ☐ RUN
- ☐ STOPPU
- ☐ STOP

Figure 4-3 Diagnostics

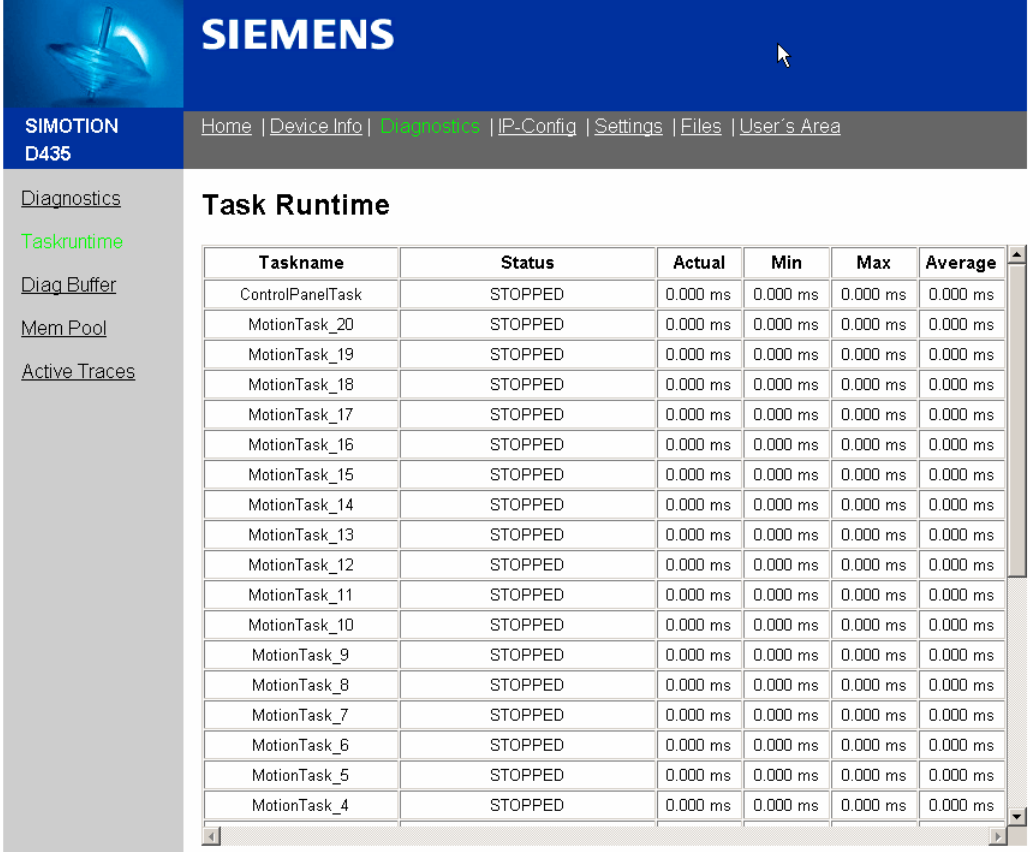


### 4.3.5 Task runtime

#### Information on task runtimes and states

On the "Task Runtime" page (opened via Diagnostics -> Task Runtime) you can view the following information:

Task name	Name of the task
Status	Current status of the task
Actual	Current runtime of the task in ms
Min	Minimum runtime of the task in ms
Max	Maximum runtime of the task in ms
Average	Average runtime of the task in ms



Taskname	Status	Actual	Min	Max	Average
ControlPanelTask	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_20	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_19	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_18	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_17	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_16	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_15	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_14	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_13	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_12	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_11	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_10	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_9	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_8	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_7	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_6	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_5	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms
MotionTask_4	STOPPED	0.000 ms	0.000 ms	0.000 ms	0.000 ms

Figure 4-4 Task runtime

### 4.3.6 Diag Buffer

#### DiagBuffer information

On the "Diag Buffer" page (opened via Diagnostics -> Diag Buffer) you can view information about events in the diagnostic buffer.

Time	Time of the event
Date	Date of the event
Event	Displays the event as text. If the language file DGBUFTXT.EDB is missing, the information is displayed in hexadecimal notation (see Section "DiagBuffer group")

#### Note

To display the event text in the desired language, you will need to download the DGBUFTXT.EDB file onto the SIMOTION memory card (see Section "DiagBuffer group").

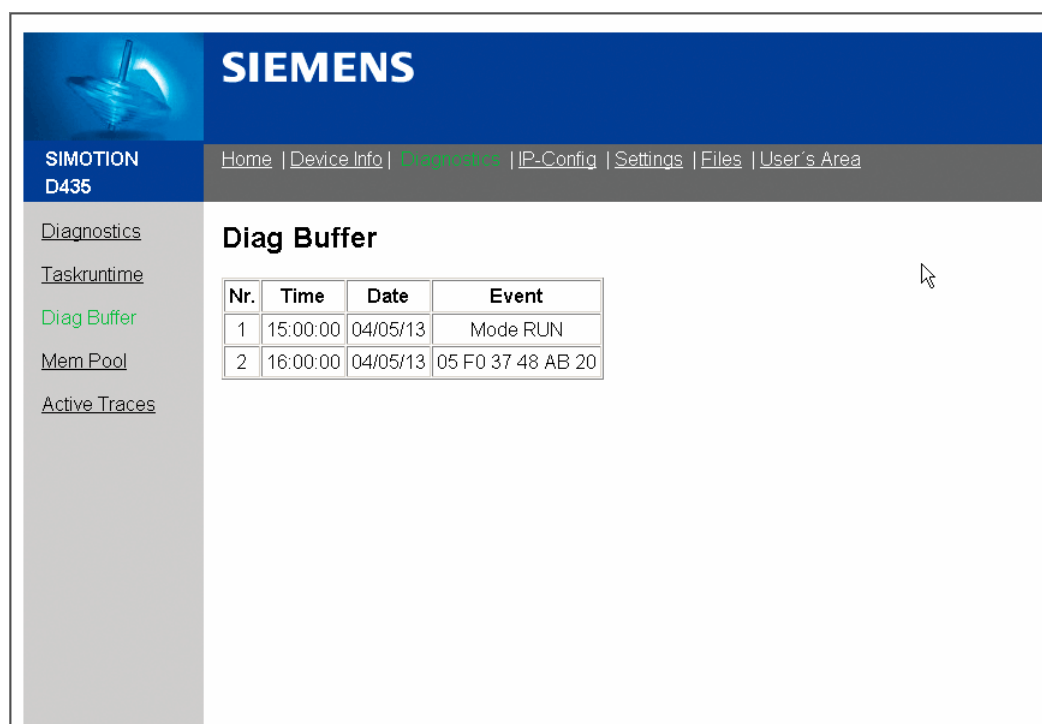



Figure 4-5 Diag Buffer

## 4.3.7 Mem Pool

### Mem Pool information

On the "Mem Pool" page (opened via Diagnostics -> Mem Pool) you can view the following internal memory management information (see also Section 5.4 MemPool):

Count	Number of memory pools
TotalSize	Total size of all pools in bytes
Name	Name of the relevant pool
TotalCount	Number of elements in the relevant pool
TotalSize	Total size of the relevant pool
MaxUsed	Maximum number of elements used in the relevant pool
CurrentUsed	Elements currently used in the relevant pool



SIMOTION

D435

[Home](#) | [Device Info](#) | [Diagnostics](#) | [IP-Config](#) | [Settings](#) | [Files](#) | [User's Area](#)

[Diagnostics](#)

[Taskruntime](#)

[Diag Buffer](#)

[Mem Pool](#)

[Active Traces](#)

Mem Pool

Mem Pool Count: 29

Name	TotalSize	ElementSize	TotalCount	MaxUsed	CurrentUsed
LISTMGMT_Entries	8000	16	500	70	69
LISTMGMT_Lists	3200	32	100	9	9
MEM_MallocPool_32	96000	32	3000	93	82
MEM_MallocPool_64	128000	64	2000	11	9
MEM_MallocPool_128	128000	128	1000	13	1
MEM_MallocPool_256	256000	256	1000	2	0
MEM_MallocPool_1024	40960	1024	40	4	2
MEM_MallocPool_8192	245760	8192	30	6	3
MEM_MallocPool_32768	327680	32768	10	4	2
CORE_Modules	1000	20	50	38	38
DEFAP_Simul_Downloads	327720	32772	10	1	1
DEFAP_Simul_Uploads	32780	32780	1	0	0
DEFAP_Simul_pFileBrowseData	126660	42220	3	0	0
EVENTSERVICE_Events	600	12	50	1	1
FILESYS_FileHandles	16400	820	20	1	1
FILESYS_FindHandles	816	272	3	0	0
TIMER_Timer	560	28	20	2	1
VARIF_ProviderEntries	560	56	10	4	4
VARIF_TransportStructs	1020	68	15	1	1

Figure 4-6 Mem Pool

### 4.3.8 Active Traces

#### ActiveTrace information

On the "Active Traces" page (opened via Diagnostics -> Active Traces) you can view the following information:

Count	Total number of active traces
Name	Name of the relevant trace
State	State of the relevant trace

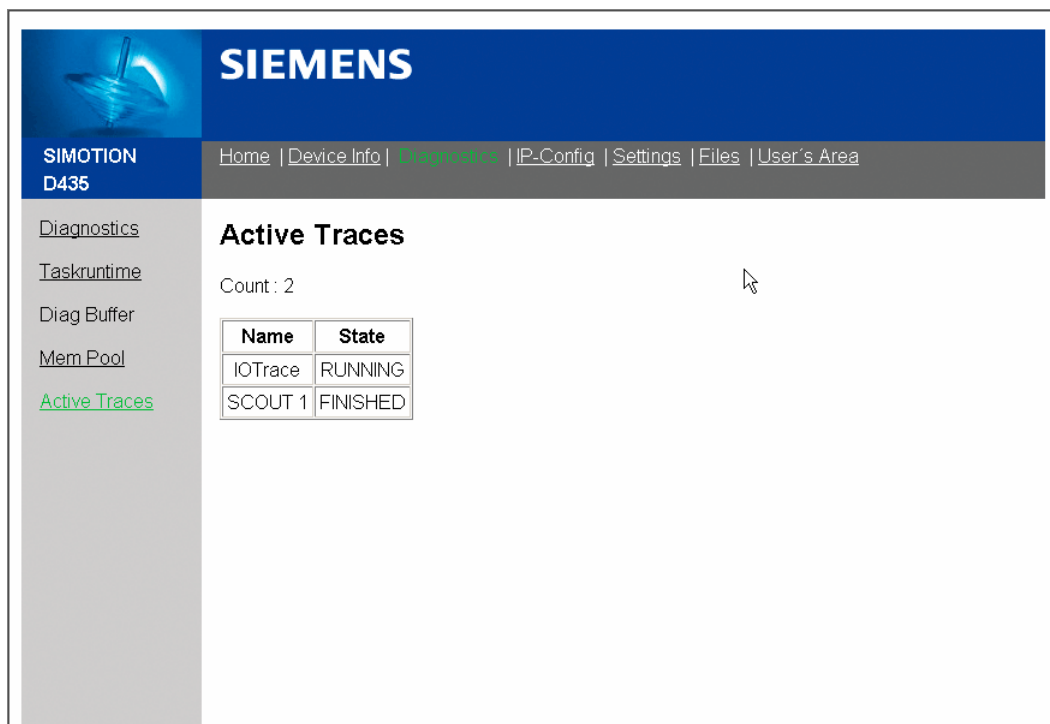


Figure 4-7 Active Traces

### 4.3.9 IP Config

#### Data of the SIMOTION device interface

The following current interface data of the SIMOTION device is displayed on the "IP-Config" page:

IP Address	Address of the TCP/IP interface
Subnet mask	Subnet mask of the interface
Gateway	Default gateway of the interface
MAC Address	Address of the network card



Figure 4-8 IP Config

#### Note

The IP address of the SIMOTION D435 device is the address of the X120 connector!

For D4xx, two IP addresses are displayed. The first from the left is the address of connector IE/X120 and the second is the address of connector IE/X130.

### 4.3.10 Settings

Via the "Settings" page, you can:

- Send new firmware, a new project and new configuration file to the SIMOTION device, as well as archive a project, in the upper part of the page.
- Change the operating state and the time zone of the SIMOTION device, in the lower part of the page.

---

#### Note

The "Settings" page is password-protected.

As delivered, the log-in is:

User = simotion

Password = simotion

You should change this password in order to avoid security problems. The password is specified in the WEBCFG.xml file.

---

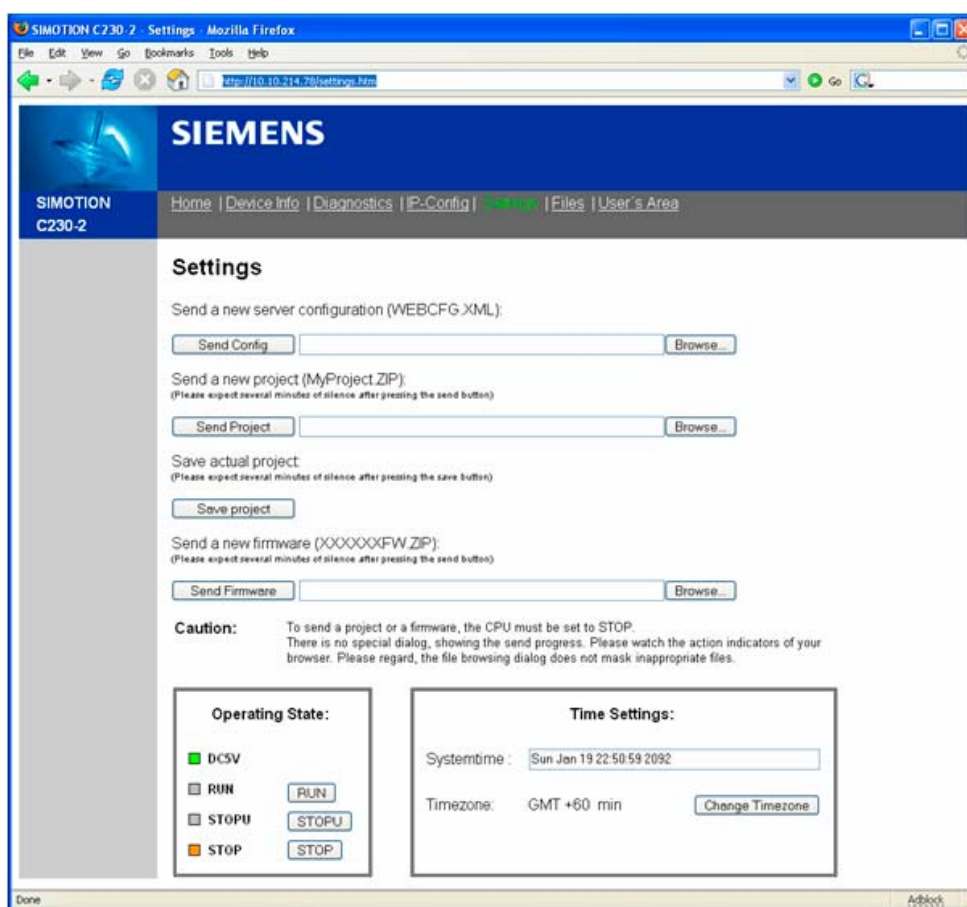


Figure 4-9 Settings

## Sending software to the SIMOTION device

The following device software can be sent from the "Settings" page to the SIMOTION device:

1. The configuration of the OPC XML server and standard diagnostics pages with the WEBCFG.xml file.
2. A new SCOUT project (e.g. MyProject.ZIP).

**Note:** The project zip file is created in compressed format in SIMOTION SCOUT via the menu command "Edit -> Save project (device) to memory card".

3. Current firmware of SIMOTION (XXXXXXFW.ZIP).

**Note:** The ZIP is generated exclusively by SIEMENS and is on the "SIMOTION Firmware with Diagnostics Functionality" CD. A firmware file in BIN format is NOT suitable for this update.

---

### Note

SIMOTION P does not support firmware download.

---

With the "Browse..." button, you can select the appropriate file on the local PC.

With the appropriate button, e.g. "Send Config", you can transfer the selected file to the SIMOTION device.



---

### Danger

To send a project or firmware, the CPU must be switched to STOP mode.

Type and contents of the file are not checked during transmission.

Sending inappropriate files, results in a crash of the SIMOTION software.

If the configuration is invalid, the WEBCFG.XML file must be deleted on the memory card in directory \USER\SIMOTION\HMICFG\.

The version of the SIMOTION SCOUT engineering tool, with which the project was created, must be identical to the version of the firmware.

---

Depending on the relevant file, SIMOTION automatically executes the following actions when the "Send" button is clicked:

- WEBCFG.xml

Restart of the diagnostics server.

**Note:** All OPC XML DA subscriptions are lost.

- MyProject.ZIP

Saving of the new project together with the Ethernet configuration on the (virtual) memory card and activation of the new project with a SIMOTION restart.

- XXXXXFW.ZIP

Saving of the firmware on the memory card and activation of the new firmware with a SIMOTION restart.

### Archiving a project

With the "Save Project" button, all files associated with the current project are written to a ZIP archive. This archive is then transferred via an HTTP download to the browser, which saves it.

This format can be uploaded back to the controller with "Send Project".

### Changing the state of the SIMOTION device

The state of the SIMOTION device, which can be changed here, comprises

- Operating mode
- System time and current time zone.

### Operating mode

In the field for the operating state of the SIMOTION device, the request to change the operating state can be triggered by clicking the respective RUN, STOPU or STOP button.

The switch on the CPU has a higher priority than this input, i.e., if the CPU switch is set to STOP, then RUN is not possible.

**Note:** For a transmission of the project or firmware, the current operating state must be set to STOP.



---

#### Danger

If the operating mode is not switched under controlled conditions, this may endanger the safety of personnel and the machine.

Please note the safety regulations before controlling a SIMOTION device via the mode selector in SIMOTION SCOUT.

---

### System time and time zone

The system time and the time zone are set in minutes, including sign, for the SIMOTION device, in the field for the time settings.

Systemtime  
Timezone

Local time-of-day of the SIMOTION device  
Difference between the local system time and GMT

The system time and the time zone are relevant for the OPC XML DA access.

The OPC XML client expects all times sent by the SIMOTION device as GMT. However a SIMOTION device is set to local time (GMT + X) and therefore a time zone must be set for the SIMOTION device.



With the "Change Timezone" button, a time zone can be selected from the displayed list of time zones.

For browsers which do not support the list display, the difference must be entered in minutes, with sign, in the range -720 to +780.

The time zone can also be set on the CPU in SIMOTION SCOUT under HW Config and then downloaded.

#### 4.3.11 Files

You can create, select and delete subdirectories on the (virtual) memory card in the SIMOTION device via the "Files" page. Files can still be saved, displayed and deleted.

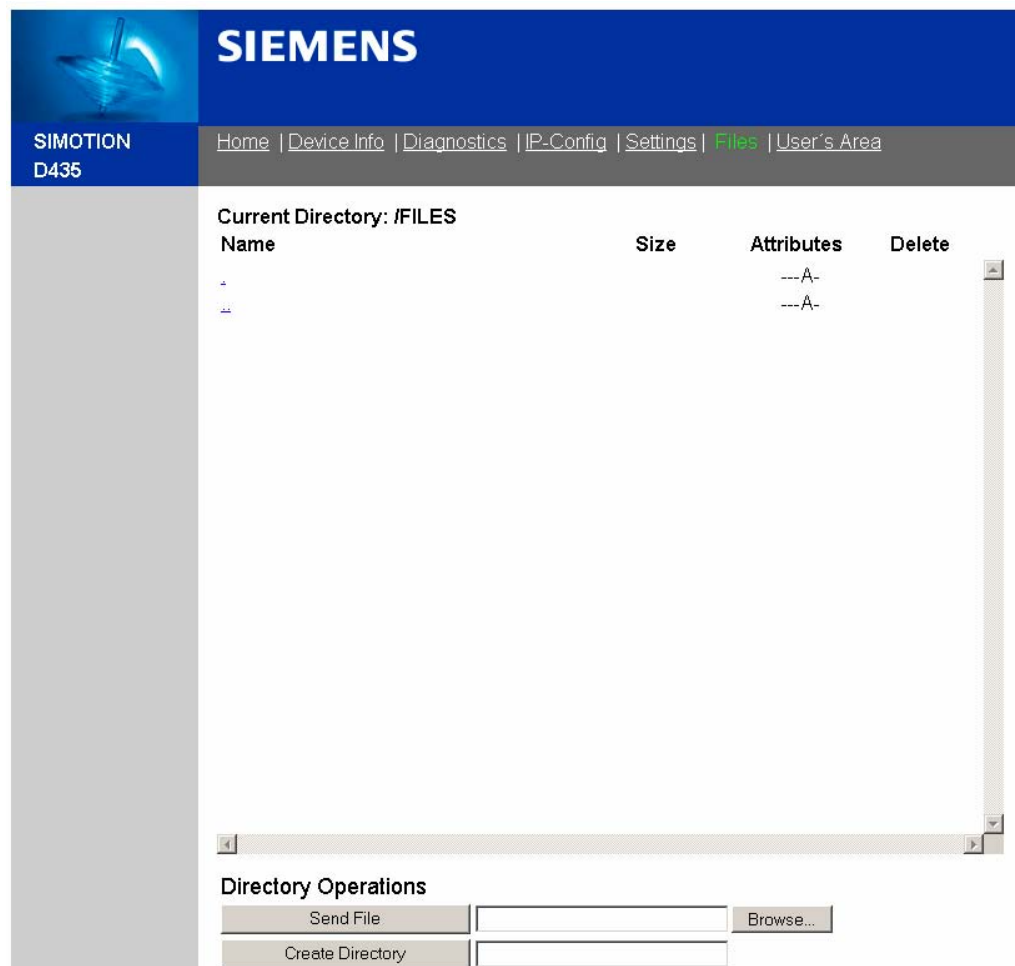


Figure 4-10 Files

#### Flash file system

The file system of the (virtual) memory card only supports standard upper-case ASCII file names in 8.3 format (8 characters and a dot to separate the extension with 3 characters). No time stamp (created/modified/read) is saved with the file.

## **File and directory management**

The user-specific directories and files are stored in the \USER\SIMOTION\HMI\FILES directory or, on the SIMOTION P350, in F:\SIMOTION\USER\CARD\USER\SIMOTION\HMI\FILES (in the case of the default installation).

To create subdirectories, enter the desired name in the input field and then confirm by clicking the "Create Directory" button.

You can delete files and directories using the Bin icon. You must make sure that a directory is empty before deleting it. If the directory contains files, these will have to be deleted first.

---

### **Note**

You can check the amount of memory available on the card on the Diagnostics Page in the "Memory Card" line.

---

## **Upload**

An upload is the transfer of a file from the local file system onto the SIMOTION CPU. You can use the "Browse" button to select a file from your local file system and click the "Send File" button to save it in the SIMOTION CPU.

---

### **Note**

If you upload a file with the same name as a file already saved in the SIMOTION CPU, the existing file will be overwritten.

---

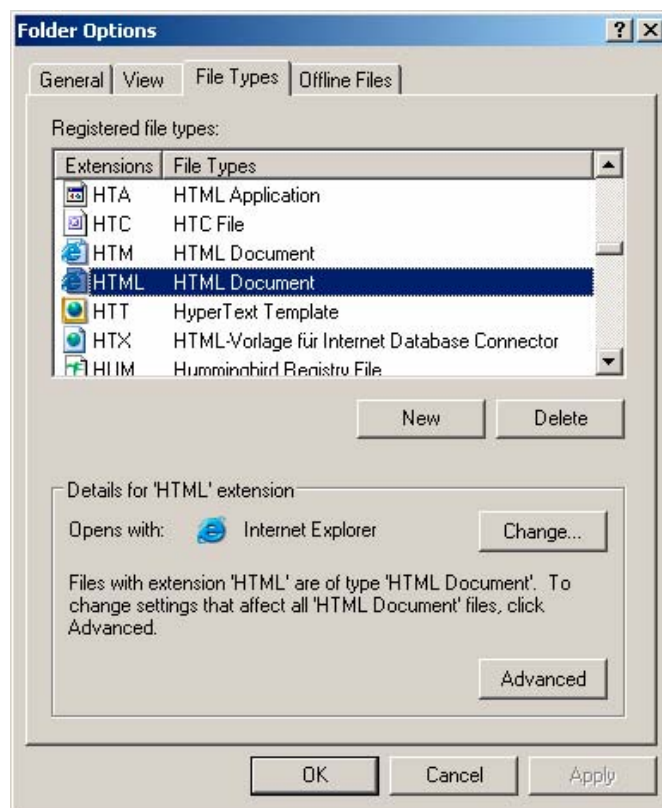
## Download

A download of user-specific HTML pages is the transfer of an HTML file, which is open in a browser (e.g. Internet Explorer), from the SIMOTION CPU to the local file system.

To enable an unfiltered transfer from the SIMOTION CPU to the local file system, all files are tagged as an "octet stream" during the download.

In the folder option settings in the Windows Explorer, you can specify that an HTML file is to be opened, e.g. in an editor, immediately after the download. An example is described below.

1. In the Windows Explorer menu, select Tools -> Folder Options to open the "Folder Options" dialog.
2. Click the "File Types" tab and select the HTM file type from the list.



3. Click the "Advanced" button.

4. In the dialog box that appears, select the "Edit" action. If you do not want to launch the editor automatically after the download, activate the "Confirm open after download" setting. Click the "Edit" button.

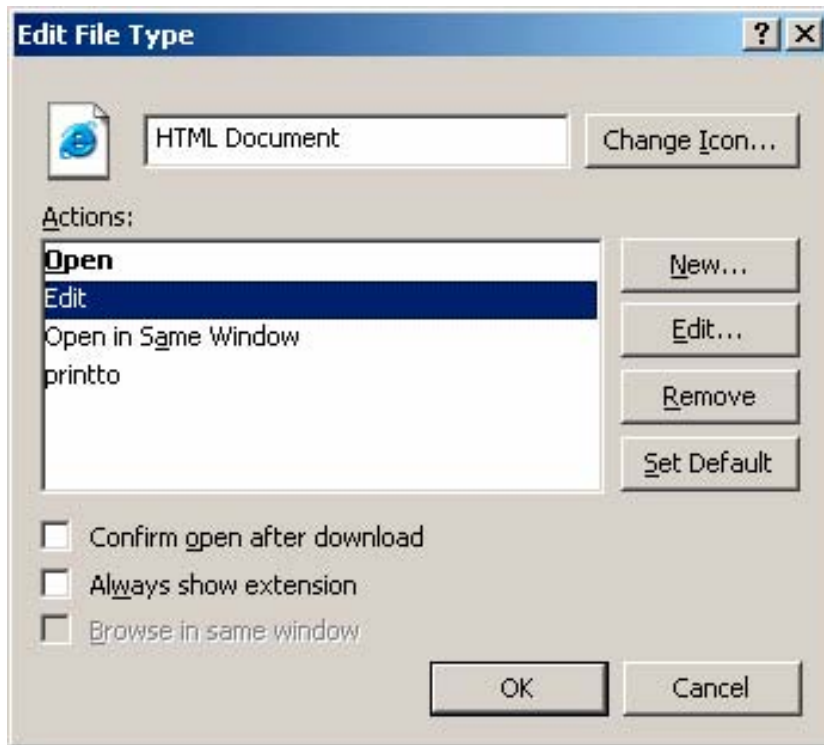


Figure 4-11 Editing the file type

5. In the "Editing action for type: HTM document" dialog, select the editor in which you want to open the document, e.g. Notepad.
6. Apply the changes by selecting the "OK" button.

This completes the settings in the Windows Explorer.

If you now click a file on the "Files" page (click the file name), it is automatically opened in the editor after the download.

## 4.4 User-defined HTML pages

### 4.4.1 Integration in standard diagnostics pages

---

**Note**

The integration of user-defined HTML pages in the standard diagnostics pages is supported in SIMOTION V3.2 and higher.

---

#### User's Area

The "User's Area" of the standard diagnostics pages is reserved for user-defined HTML pages. In this area, you can store user-defined HTML pages on the SIMOTION CPU using the flash file system.

The user-specific directories and files are stored in the \USER\SIMOTION\HMI\FILES directory. In the "Files" section of this manual, you can read how to manage files and directories in the flash file system.

In order to display your Home page in the User's Area, you must assign the file name USER.HTM as the Home page of the area. The HTML page can only be called up via the "User's Area" link under the name USER.HTM.

This document does not include a description of HTML. Information about the HTML language can be found on the Internet at <http://www.selfhtml.org>.

### 4.4.2 Server Side Includes (SSI)

#### Including process variables

You can include process variables in the user-defined HTML pages using Server Side Includes.

The variables are included in the HTML page using the `<%=IDENTIFIER %>` string. IDENTIFIER is a placeholder which you must replace with the variables of the variable provider. For example `<%=DeviceInfo.Board%>`: this variable returns the name of the CPU. On a D435, for example, the value is "D435".

Details about the variables and syntax can be found in the "Variable provider" section.

The source text below shows an example for including the variable `userData.user1`. First, the value of the variable is output (system variable `userData.user1`: `<%=var/userData.user1 %>`). The value of the variable is used as a default in the input field and can be overwritten by a user input.

```
<html>
```

```

<head>
  <title>Demo Page</title>
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000">
  Demopage<br>
  Systemvariable userData.user1 : <%= var/userData.user1 %> <br>
  <form method="post" action="/VarApp">
    C230: userData.user1:
    <input type="TEXT" name="var/userData.user1" value="<%=
var/userData.user1 %>" />
    <input type="submit" value="Write value" />
  </form>
</body>
</html>

```

#### 4.4.3 User-defined Home page

You can create your own Home page and display it instead of the Home page for the standard diagnostic pages of the CPU. To do this, you need to change the default page of the web server in the WEBCFG.XML file.

##### Procedure

1. Create your own Home page and save it, e.g. as default.htm.
2. Download the Home page onto the Simotion memory card from the "Files" page.
3. Open the WEBCFG.XML file in an available editor. The file can be found either on the supplied CD in the 3\_Configuration directory (in the default state) or on the SIMOTION memory card (possibly in a modified state) in directory USER\SIMOTION\HMICFG\.
4. Replace the file name index.htm in <SERVEROPTIONS> in element <DEFAULTDOCUMENT VALUE="index.htm" /> with the name of your Home page, including the path name "files" (all user-defined HTML pages are stored in the FILES directory).

##### Example:

```

<SERVEROPTIONS>
  <DEFAULTDOCUMENT VALUE="files/default.htm" />
</SERVEROPTIONS>

```

5. Save the modified WEBCFG.XML file to the memory card from the "Settings" page.

## Variables providers

### 5.1 Overview

#### Variables providers

The data of the SIMOTION device can be reached via the "Variables providers". Each provider enables access to certain variables.

At present there are three variable providers as described in the following.

- SIMOTION
- SIMOTION diagnostics
- MemPool

You can access the data presented by the variable providers from SIMOTION IT OPC XML-DA, from SIMOTION IT DIAG (standard diagnostics pages) or, if necessary, from user-defined HTML pages using Server Side Includes.

## 5.2 SIMOTION

Access to the SIMOTION process variables is possible via the "SIMOTION" provider.

---

### Note

An exact list with description can be found in the Online Help of SIMOTION SCOUT, in the Chapter, "System Functions, System Variables and Configuration Data".

---

### Variable syntax of the "SIMOTION" provider

With OPC XML DA V1.0, access to the variables of the SIMOTION device is via the terms "ItemPath" and "ItemName". With SIMOTION IT DIAG and Server Side Includes (SSI) in user-defined HTML pages, access is via the "ItemName".

#### ItemPath

The name for "ItemPath" is always "SIMOTION" for SIMOTION process variables.

ItemPath="SIMOTION"

---

### Note

The "ItemPath" is only needed for access via OPC XML DA, not for SIMOTION IT DIAG or for Server Side Includes in user-defined HTML pages.

---

#### ItemName

The name for "ItemName" varies.

For **system variables**, the syntax is:

ItemName="var/name"

Example: ItemName="var/userData.user3"

For **TO system variables**, the syntax is:

ItemName="to/name.variable"

Example: ItemName="to/Achse\_1.positioningState.actualPosition"



---

**Note**

The names of the system variables and TO system variables to be used can be found in the Online Help of SIMOTION SCOUT, in the Chapter, "System Functions, System Variables and Configuration Data".

**Upper and lower case notation must be strictly adhered to!**

---

For unit variables in the interface, the syntax is:

Example:            ItemName=" unit/name.variable"  
                  ItemName=" unit/prog\_1.var\_1"

---

**Note**

The names to be used for the unit variables in the interface correspond to the program and variable names **in lower case characters**.

---

## **5.3 SIMOTION diagnostics**

### **5.3.1 Introduction**

#### **Access to diagnostics variables**

The diagnostics variables of a SIMOTION device can be accessed via the "SIMOTION diagnostics" provider.

Most of the variables have read-only access and a few (e.g. operating state) also have write access. All variables are of the string type. Therefore, numerical values are converted into strings by the provider.

The number of variables varies and depends on the current configuration of the SIMOTION device. The provider supports browsing via OPC XML DA V1.0, and therefore the current number of variables can be viewed.

---

#### **Note**

Upper and lower case characters must be adhered to!

---

#### **Variable groups of the "SIMOTION diagnostics" provider**

The diagnostics variables of the "SIMOTION diagnostics" provider are combined in groups.

A variable name is made up of the group name and variable name:

E.g.: Group.Variable

## 5.3.2 DeviceInfo group

### General information about SIMOTION

The DeviceInfo group contains general information on the SIMOTION device. The 10 variables of this group are always available.

Table 5-1 Variables of the DeviceInfo group

Variable	Description
DeviceInfo.Board	Specifies the system being used, e.g. "C230-2", read-only
DeviceInfo.License-Serial-No	License serial number for this device, read-only
DeviceInfo.BZU	Access to the operating state, read and write, valid values for writing: STOP, STOPU, RUN
DeviceInfo.Systemtime	Access to the system time, read and write, the time must always be specified as in the following example: "Tue Aug 05 17:00:00 2003"; another format is not accepted.
DeviceInfo.Timezone	Time offset in minutes, read and write, valid values are -720 to +720
DeviceInfo.Active-MAC	Active MAC address, read-only
DeviceInfo.Remanent-MAC	Retentive MAC address, read-only
DeviceInfo.IP-Address	IP configuration data (address, subnet mask and gateway), read-only
DeviceInfo.Subnet-Mask	
DeviceInfo.Gateway	

### Additional variables of the DeviceInfo group

The following variables supply HTML color values ("#XXXXXX") which correspond to the colors of the DC5V, RUN, STOPU and STOP LEDs of the SIMOTION device. It is therefore possible, for example, to display the operating state as "traffic light information" via an HTML table (by means of the "background" attribute in the cells), similar to the display in SCOUT, as with "Operating state ..."

Access to these values is read-only.

Table 5-2 Variables of the DeviceInfo group

Variable	Description
DeviceInfo.LEDColor.DC5V	Color for the DC5V LED; as the server can only be addressed when a voltage is applied, the corresponding HTML color is always green ("#00FF00")
DeviceInfo.LEDColor.RUN	Color for the RUN LED; green in the RUN operating state ("#00FF00") otherwise gray ("#C0C0C0")
DeviceInfo.LEDColor.STOPU	Color for the STOPU LED; amber in the STOPU operating state ("#FF9900") otherwise gray ("#C0C0C0")
DeviceInfo.LEDColor.STOP	Color for the STOP LED; amber in the STOP operating state ("#FF9900") otherwise gray ("#C0C0C0")

### 5.3.3 ComplInfo group

This group supplies information on the components of the device. The number of variables varies in this group depending on the number of technology packages or additional hardware components.

Access to all variables is read-only.

#### Information on the CPU

The following variables supply information on the CPU:

Table 5-3 Variables of the ComplInfo group

Variable	Description
ComplInfo.Cpu.MLFB	CPU MLFB / order number
ComplInfo.Cpu.Serial-No	CPU serial number
ComplInfo.Cpu.Revision-No	Version number
ComplInfo.Cpu.Kernelname	Kernel name
ComplInfo.Cpu.Build-No	Build number
ComplInfo.Cpu.User-Version	User version (firmware)

#### Information on the technology packages (TPs) and hardware

The number of available TPs or hardware components can be determined with the following variables.

Table 5-4 Variables of the ComplInfo group

Variable	Description
ComplInfo.TP-Count	Number of available technology packages
ComplInfo.HW-Count	Number of available hardware components

If TPs are available, information on the individual TPs can be obtained with ComplInfo.TPx.Variable-Name (whereby x stands for the TP number).

The first TP is allocated the number 1 (not 0),  
for example: ComplInfo.TP1.Name

The following information is available:

Table 5-5 Variables of the ComplInfo group

Variable	Description
ComplInfo.TPx.Name	Name of the TP
ComplInfo.TPx.User-Version	User version of the TP
ComplInfo.TPx.Build-No	Build number of the TP

If additional hardware components are available, information on the individual hardware components can be obtained with ComplInfo.HWx.Variable-Name (whereby x stands for the HW number).

The first hardware component is allocated the number 1 (not 0),  
for example: ComplInfo.HW1.MLFB

The following information is available:

Table 5-6 Variables of the ComplInfo group

Variable	Description
ComplInfo.HWx.MLFB	MLFB / order number
ComplInfo.HWx.Serial-No	Serial number
ComplInfo.HWx.Revision-No	Version number
ComplInfo.HWx.FirmwareName	Firmware name
ComplInfo.HWx.Build-No	Build number
ComplInfo.HWx.User-Version	User version

As the information is dynamic and the scope is not known beforehand, the following variables also exist to simplify the display of hardware components and TPs in HTML:

Table 5-7 Variables of the ComplInfo group

Variable	Description
ComplInfo.TableHead.TP	Supplies the header of an HTML table with all the information about the TPs, e.g. "<tr><th>TP Name</th><th>User Ver.</th><th>Build No.</th></tr>"
ComplInfo.Table.TP	Supplies an HTML table with all the information about all the available TPs
ComplInfo.TableHead.HW	Supplies the header of an HTML table with all the information about the hardware components, e.g. "<tr><th>MLFB</th><th>Serial No.</th><th>Revision No.</th><th>Firmware Name</th><th>User Ver.</th><th>Build No.</th></tr> "
ComplInfo.Table.HW	Supplies an HTML table with all the information about all the available hardware components

#### Note

Separate access to the table and the table header enables separate formatting.

### 5.3.4 CPULoad group

#### Information on CPU load

The CPULoad group supplies information on the load of the CPU. Access to all variables is read-only.

Table 5-8 Variables of the CPULoad group

Variable	Description
CPULoad.Percent	CPU load in percent
CPULoad.Mintime	Minimum runtime of the BackgroundTask (free cycle) in ms with 5 decimal places
CPULoad.Acttime	Current runtime of the BackgroundTask (free cycle) in ms with 5 decimal places
CPULoad.Maxtime	Maximum runtime of the BackgroundTask (free cycle) in ms with 5 decimal places

### 5.3.5 MemoryLoad group

#### Information on memory load

The MemoryLoad group provides information about the load on memory devices in bytes or as a percentage. Variables can only be accessed in read-only mode.

Table 5-9 Variables of the MemoryLoad group

Variable	Description
MemoryLoad.Flash-Size	Size of the Flash memory
MemoryLoad.Flash-Used	Current amount of Flash memory used
MemoryLoad.RAM-Size	Size of the RAM
MemoryLoad.RAM-Used	Current amount of RAM used
MemoryLoad.RAMDisk-Size	Size of the RAM disk
MemoryLoad.RAMDisk-Used	Current amount of RAM disk used
MemoryLoad.Remanent-Size	Size of the retentive memory
MemoryLoad.Remanent-Used	Current amount of retentive memory used
MemoryLoad.Flash-Percent	Percentage of external Flash memory used
MemoryLoad.RAM-Percent	Percentage of RAM memory used
MemoryLoad.RAMDisk-Percent	Percentage of RAM disk used
MemoryLoad.Remanent-Percent	Percentage of internal Flash memory used

### 5.3.6 TaskRT group

#### Variables of the TaskRT group

The TaskRT group supplies information on the task runtimes and the task states of the SIMOTION device. The same values are supplied as in the SIMOTION SCOUT under device diagnostics, task runtimes. Access to all values is read-only. The number of variables varies and depends on the current configuration of the execution system in SIMOTION SCOUT.

Table 5-10 Variables of the TaskRT group

Variable	Description
TaskRT.TaskCnt	Supplies the number of currently available tasks

#### Task names

The following information can be obtained for the individual tasks via TaskRT.Task-name.Variable-Name. The following task names are valid (prerequisite is that the task already exists in the execution system):

Table 5-11 Task names

Task names for SIMOTION IT DIAG	Task names in the SIMOTION SCOUT execution system
BackgroundTask	BackgroundTask
StartupTask	StartupTask
ShutdownTask	ShutdownTask
IPOSynchTask, IPOSynchTask_2	IPOSynchronousTask, IPOSynchronousTask_2
TimeFaultTask	TimeFaultTask
TechnoFaultTask	TechnologicalFault
PeriphFaultTask	PeripheralFaultTask
ExecFaultTask	ExecutionFaultTask
TimeFaultBGTask	TimeFaultBackgroundTask
PWMSynchTask	PWMSynchronousTask
InputSynchTask_1, InputSynchTask_2	InputSynchronousTask_1, InputSynchronousTask_2
PostCtrlTask_1, PostCtrlTask_2	PostControlTask_1, PostControlTask_2
MotionTask_1 to MotionTask_20	MotionTask_1 to MotionTask_20
CtrlPanelTask	ControlPanelTask

#### Information on the task

The same information can be obtained for every task; here is an example of the first MotionTask.

##### Example:

TaskRT.MotionTask\_1.Status

Current task status, can be an appropriate combination of the following values:  
 STOP\_PENDING, STOPPED, RUNNING, STOP\_UNCOND, WAITING, SUSPENDED,  
 WAITING\_FOR\_NEXT\_CYCLE, WAITING\_FOR\_NEXT\_INTERRUPT, LOCKED,  
 SUSPENDED\_BY\_DEBUG\_MODE

### Additional variables of the TaskRT group

Table 5-12 Variables of the TaskRT group

Variable	Description
TaskRT.MotionTask_1.Actual	Current runtime of the task in s, with 5 decimal places
TaskRT.MotionTask_1.Min	Minimum runtime of the task in s, with 5 decimal places
TaskRT.MotionTask_1.Max	Maximum runtime of the task in s, with 5 decimal places
TaskRT.MotionTask_1.Average	Average runtime of the task in s, with 5 decimal places

As the information is dynamic and the scope is not known beforehand, the following variables also exist to simplify the display of task information in HTML:

Table 5-13 Variables of the TaskRT group

Variable	Description
TaskRT.TableHead	Supplies the header of an HTML table with all the information about the tasks, e.g. " <tr><th>Taskname</th><th>Status</th> <th>Actual</th><th>Min</th><th>Max</th> <th>Average</th></tr> "
TaskRT.Table	Supplies an HTML table with all the information about the available tasks; all runtime values are entered with the unit as, opposed to the individual value query, they can vary between s and ms. Three decimal places are displayed.



### **5.3.7 DiagBuffer group**

The DiagBuffer group supplies information on the events in the DiagBuffer. Access to all variables is read-only.

Events can be output in text format. The German, English and Italian languages are supported. The following condition must be met for text output to take place:

#### **Requirements**

To display the event text in the desired language, you will need to download the DGBUFTXT.EDB file onto the SIMOTION memory card.

Proceed as follows:

#### **Procedure**

1. Open the 3\_CONFIGURATION\<language> directory on the SIMOTION IT DIAG CD. You can choose between DE (German), EN (English) and IT (Italian). You will find the DGBUFTXT.EDB file in the corresponding directory.
2. Insert the SIMOTION memory card in a reader/writer.
3. Copy the DGBUFTXT.EDB file into the \USER\SIMOTION\HMICFG directory. Please create the directory if it does not already exist.
4. Insert the memory card in the SIMOTION device again.

#### **Procedure for P350**

1. Shut down the SIMOTION P.
2. Open the 3\_CONFIGURATION\<language> directory on the SIMOTION IT DIAG CD. You can choose between DE (German), EN (English) and IT (Italian). You will find the DGBUFTXT.EDB file in the corresponding directory.
3. Copy the DGBUFTXT.EDB file into the F:\SIMOTION\USER\CARD\USER\SIMOTION\HMICFG directory (for the default installation).
4. Start the SIMOTION P.

---

#### **Note**

Only one language can be saved in SIMOTION at a time.

If no language file is available on the SIMOTION device, the events are read out as hexadecimal text. If necessary, the output can be interpreted by Siemens Support.

---

## Variables of the DiagBuffer group

The following variables are available for enhancing the display:

Table 5-14 Variables of the DiagBuffer group

Variable	Description
DiagBuffer.TableHead	Supplies the header of an HTML table with all events. The contents are: <tr><th>Nr</th><th>Time</th><th>Date</th><th>Event</th></tr></td>
DiagBuffer.Table	Supplies the contents of an HTML table with all events. The structure of each row is as follows: <tr><td>NUMBER</td><td>TIME</td><td>DATE</td><td>EVENT</td></tr></td> <b>Note:</b> The NUMBER, TIME, DATE and EVENT texts specified in the format are replaced with the corresponding value of each event.

The following variables can be used for direct access to the data of certain events in the diagnostic buffer:

Table 5-15 Variables of the DiagBuffer group - direct access

Variable	Description
DiagBuffer.EventCnt	Number of events currently in the diagnostic buffer
DiagBuffer.Time_1 to DiagBuffer.Time_n	Time of each event
DiagBuffer.Date_1 to DiagBuffer.Date_n	Date of each event
DiagBuffer.Text_1 to DiagBuffer.Text_n	Text of each event <b>Note:</b> If the event text number and their parameters cannot be resolved, the number and parameters are output in the same as format as DiagBuffer.HEX_1 to DiagBuffer.HEX_n.
DiagBuffer.HEX_1 to DiagBuffer.HEX_n	Binary dump of each event in HEX format. The value of these variables is a string of 20 hexadecimal characters (without separator).

### 5.3.8 ActiveTraces group

#### Variables of the ActiveTraces group

The ActiveTraces group returns the number of active traces and a list of the active traces. Access to all variables is read-only.

Table 5-16 Variables of the ActiveTraces group

Variable	Description
ActiveTraces.TraceCnt	Number of active traces
ActiveTraces.TableHead	Supplies the header of an HTML table with all active traces. The contents are: <tr><th>Name</th><th>State</th></tr>
ActiveTraces.Table	Supplies the contents of an HTML table with all active traces. The structure of each row is as follows: <tr><td>NAME</td><td>STATE</td></tr>  <b>Note:</b> The NAME and STATE placeholders specified in the format are replaced with the corresponding value of each trace.

### 5.3.9 Comparison with the device diagnostics of SIMOTION SCOUT

#### Comparison with device diagnostics in SIMOTION SCOUT

The variables described in this chapter are based on the view of the device diagnostics in SIMOTION SCOUT. The following figures show the connection between the "SIMOTION diagnostics" variables and the device diagnostics in SIMOTION SCOUT.

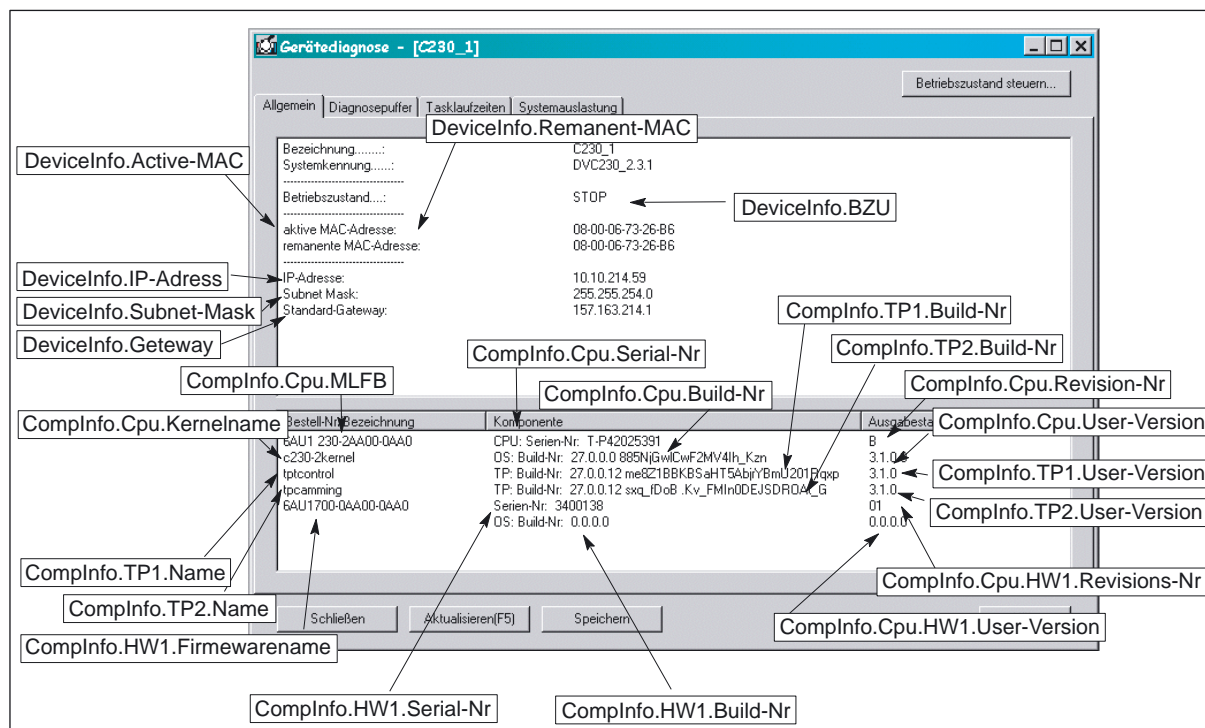


Figure 5-1 "General" device diagnostics

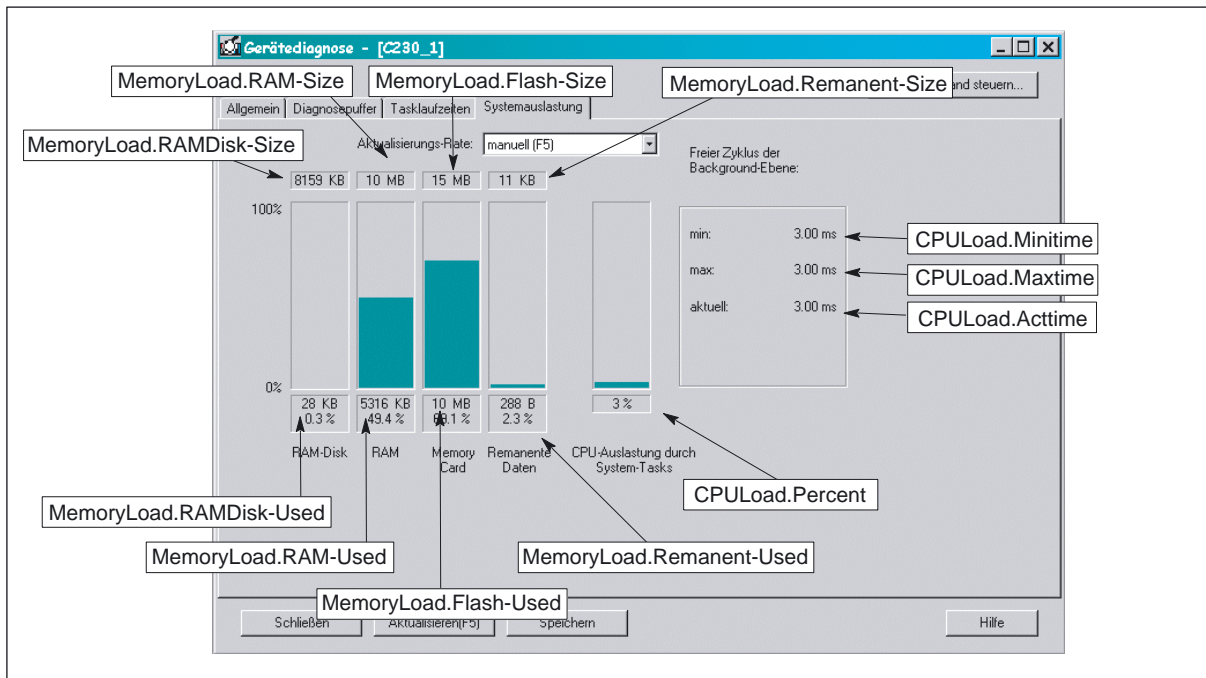


Figure 5-2 "System load" device diagnostics

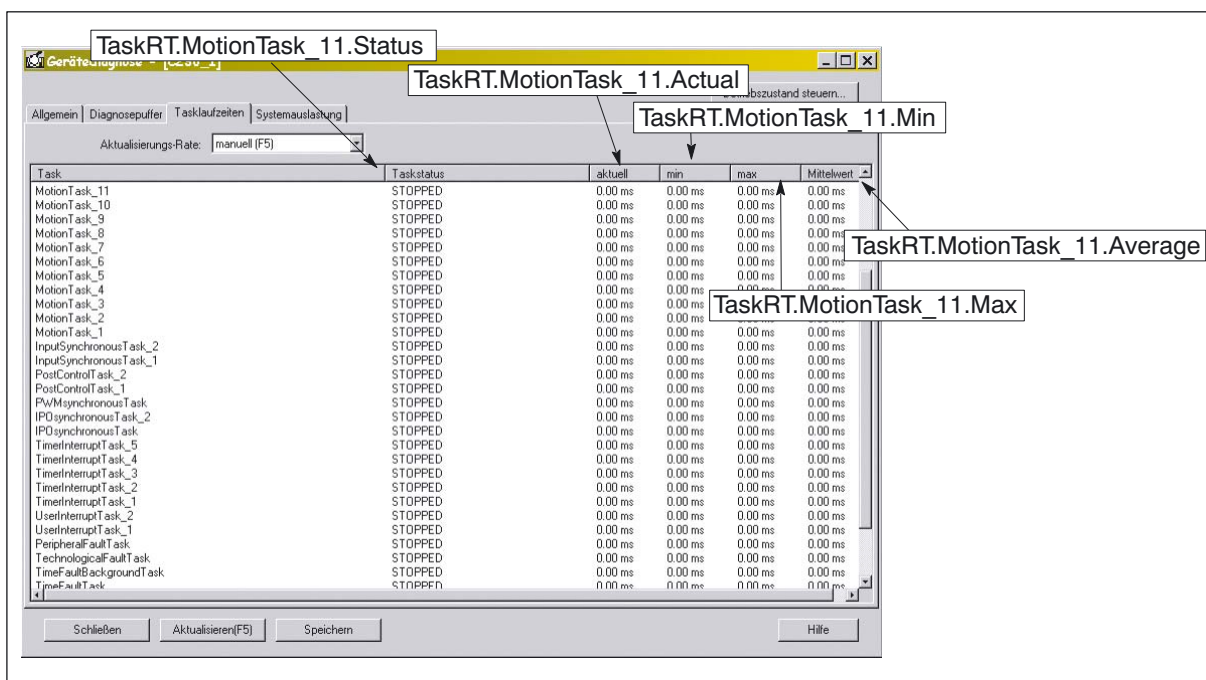


Figure 5-3 "Task runtimes" device diagnostics

## 5.4 MemPool

The "MemPool" provider gives you an insight into the internal memory management. The variables of MemPool refer to the memory management for each project. The information provider by MemPool can be forwarded to the support department if necessary.

All variables are of the string type. Therefore, numerical values are converted into strings by the variable provider.

---

**Note**

Upper and lower case characters must be adhered to!

---

### Variables of the MemPool provider

The memory pools are generated at runtime. A name is allocated to each memory pool. The MemPool provider presents two types of variable:

#### General variables

These provide information about all existing memory pools.

Table 5-17 General variables

Variable	Description
MemPool/Count	Number of memory pools
MemPool/TotalSize	Total size of all pools
MemPool/XML	Data of all MemPools as XML text

#### Memory pool-specific variables

These provide memory information about each memory pool.

Table 5-18 Memory pool-specific variables

Variable	Description
MemPool/<poolname>/TotalSize	Total size of pool
MemPool/<poolname>/TotalCount	Number of elements (memory blocks)
MemPool/<poolname>/MaxUsed	Maximum number of elements (memory blocks) used
MemPool/<poolname>/CurrentUsed	Number of elements (memory blocks) currently used

# Secure Socket Layer

## Introduction

The Secure Socket Layer protocol (SSL) enables encrypted data transfer between a client and SIMOTION. HTTPS access between the browser and the SIMOTION CPU is based on the Secure Socket Layer protocol.

Encrypted access to SIMOTION can take place via SIMOTION IT OPC-XML-DA or SIMOTION IT DIAG.

This section tells you which steps you need to follow to enable encrypted data communication between a client and SIMOTION. The possibilities are as follows:

1. You have a Certification Authority (CA) in your organization and the necessary key files are available. In this case, "Transferring key files to SIMOTION" is the relevant section.
2. You do not have a CA in your organization. In this case, you will need to create the key files yourself. An example of how to do this is described in the "Creating key files" section. When you have created the key files, they will need to be transferred to SIMOTION.

---

### Note

HTTPS connections are supported in SIMOTION V3.2 and higher.

---

## 6.1 Key files

### Encryption methods

You need two key files for the encryption method used by the Secure Socket Layer protocol. You need a public certificate and a private key. The pair of keys is created individually for each SIMOTION CPU. This ensures that the address requested matches the SIMOTION device accessed during HTTPS communication.

---

#### Note

Encrypted access to SIMOTION is only possible using the name specified when the key is created (name/IP address).

---

You can find further information about Secure Socket Layer certificates at <http://www.verisign.de>.

### Default upon delivery

To ensure that you can access SIMOTION by HTTPS with the supplied version of SIMOTION IT DIAG (standard diagnostics pages), the system is supplied with two key files already integrated in the WEBCFG.XML file.

When you attempt HTTPS access using the key files supplied with the system, you will be warned that the certificate is unknown and that the current address of the CPU does not match the name of the CPU in the certificate.

---

#### Note

Encrypted HTTPS access with OPC XML-DA is not possible using these standard certificates.

---



## 6.2 Creating key files

### Overview

---

**Note**

HTTPS connections are supported in SIMOTION V3.2 and higher.

---

If no Certification Authority (CA) is available in your organization, we recommend that you follow the steps described in the following section. The OpenSSL tool is used to create the key files.

The following steps must be followed in order to create key files.

No.	Working step	Note
1.	Install the development environment	Only do this if a tool is not available for creating the CA
2.	Generate the OpenSSL tool	You only need to do this if a tool is not available for creating the CA
3.	Modify the OpenSSL configuration	
4.	Create a Certification Authority (CA)	You only need to do this if you are using SSL with your own Certification Authority
5.	Creating a server certificate request	You only need to do this if you are using SSL with your own Certification Authority
6.	Sign the server certificate request	You only need to do this if you are using SSL with your own Certification Authority
7.	Generate the SIMOTION files	You only need to do this if you are using SSL with your own Certification Authority
8.	Import the certificate to the browser	You only need to do this if you are using SSL with your own Certification Authority

When the key pairs have been successfully created, you will need to copy them onto the SIMOTION memory card. HTTPS access is available when the SIMOTION CPU is booted.

## 6.2.1 Install the development environment

### Software components

The following section gives you an overview of the software components needed in order to create the key pair.

---

**Note**

An installed version of Microsoft Visual Studio 6 is necessary in order to create the OpenSSL application.

---

In certain cases, you may be able to use the existing software in your organization in place of the specified software components. The software described here, which can be obtained from the specified sources, is suitable for the example key creation described in this manual.

Tool/source	Note
<b>Winzip</b> <a href="http://www.winzip.com/downauto.cgi?winzip90.exe">http://www.winzip.com/downauto.cgi?winzip90.exe</a> (free downloadable evaluation version)	if no tool is available for unpacking tar.gz.
<b>ActivePerl</b> <a href="http://downloads.activestate.com/ActivePerl/Windows/5.8/ActivePerl-5.8.3.809-MSWin32-x86.msi">http://downloads.activestate.com/ActivePerl/Windows/5.8/ActivePerl-5.8.3.809-MSWin32-x86.msi</a> (free downloadable version)	if Perl is not installed
<b>Assembler</b> <a href="http://www.kernel.org/pub/software/devel/nasm/binaries/win32/">http://www.kernel.org/pub/software/devel/nasm/binaries/win32/</a> (free downloadable version)	if no assembler is installed  <b>Note:</b> In the example, NASMW.EXE was saved in "C:\Program Files\Microsoft Visual Studio\Common\Tools".
<b>OpenSSL</b> <a href="http://www.openssl.org/source/openssl-0.9.7d.tar.gz">http://www.openssl.org/source/openssl-0.9.7d.tar.gz</a> (free downloadable version)	<b>Note:</b> Saved in the example in "My Documents\VisualStudio\openssl-0.9.7d"

## 6.2.2 Generating the OpenSSL tool

### Proceed as follows

Please follow the steps below to generate the OpenSSL tool:

1. Change to the "My Documents\VisualStudio\openssl-0.9.7d" directory
2. Open the DOS prompt and enter the following commands in succession:
  - wperl Configure VC-WIN32
  - "C:\Program Files\Microsoft Visual Studio\VC98\Bin\VCVARS32.BAT"
  - ms\do\_nasm
  - nmake -f ms\ntdll.mak
3. Please follow the steps below to test the OpenSSL tool:
  - From the DOS prompt, change to directory "out32dll". To do this, enter the following command:  
cd out32dll
  - Then run the following command:  
..\ms\test

If the OpenSSL tool was generated successfully, the last line of the text will read "passed all tests".

### 6.2.3 Modifying the OpenSSL configuration

Proceed as follows

---

**Note**

An installed version of the Perl interpreter is necessary in order to use the OpenSSL tool.

---

1. First, copy the following files into a separate directory. "My Documents\OpenSSL" was used in the example.

```
openssl-0.9.7d\apps\CA.pl
openssl-0.9.7d\apps\openssl.cnf
openssl-0.9.7d\out32dll\openssl.exe
openssl-0.9.7d\out32dll\libeay32.dll
openssl-0.9.7d\out32dll\ssleay32.dll
```

2. Open the CA.pl file in an editor of your choice and make the following changes and entries:

```
# $SSLEAY_CONFIG=$ENV{"SSLEAY_CONFIG"};
$SSLEAY_CONFIG="-config openssl.cnf";
$DAYS="-days 7305";
```

The settings, e.g. DAYS, are examples and can be changed if necessary.

3. Open the "openssl.cnf" file in an editor of your choice and make the following changes and entries: The settings, e.g. default\_days, are examples and can be changed if necessary. Customize the data to match the details of your organization.

```
unique_subject = no
default_days = 7305
countryName_default = DE
stateOrProvinceName_default = Germany
localityName_default = Erlangen
0.organizationName_default = Siemens AG
organizationalUnitName_default = A&D MC
commonName_default = SIMOTION
emailAddress_default =
```

## **6.2.4 Create a Certification Authority (CA)**

The certificate created for each SIMOTION CPU must be capable of being authenticated by a Certification Authority. The following options are available:

1. You can submit the CPU certificates to a generally recognized Certification Authority (e.g. <http://www.verisign.com>) for signature subject to charge.
2. You can create a dedicated Certification Authority for your organization. Each machine manufacturer needs to take this action only once.

---

### **Note**

If you use SSL with your own Certification Authority, you will need to prepare your PCs for communication with the SIMOTION CPU. To do this, import the certificate of the CA using your standard browser.

---

### **Proceed as follows**

To create the Certification Authority, proceed as follows:

1. Open the DOS prompt and change to the OpenSSL directory,  
e.g. `cd My Documents\OpenSSL`
2. Run the following command:  
`perl CA.pl -newca`
3. Various questions will appear during the execution of the Perl script. Accept the default answers, except in the case of "pass phrase". To confirm the default values, press the Enter key.

In "pass phrase", enter a password to protect the CA key against misuse.

### 6.2.5 Creating a server certificate request

#### Proceed as follows

1. Open the DOS prompt and change to the OpenSSL directory,  
e.g. `cd My Documents\OpenSSL`
2. Run the following command:  
`perl CA.pl -newreq-nodes`
3. Various questions will appear during the execution of the Perl script.  
Accept the default answers, except in the case of the "common name" and the "optional company name". To confirm the default values, press the Enter key.

common name	Enter the final address of the SIMOTION CPU. If the SIMOTION CPU is registered on a DNS or if it uses local HOSTS files, you will need to specify the full name of the CPU. Otherwise, use the IP address of the CPU. In the example, a DNS is used and the name is: <code>d435.erlf.siemens.de</code> <b>Note</b> Problem-free HTTPS access to the SIMOTION CPU will be possible later on only if this specified name is used.
optional company name	If you use a generally recognized Certification Authority, enter your own organization's name as the "optional company name".

### 6.2.6 Sign the server certificate request

#### Proceed as follows

Proceed as follows to sign the server certificate request:

1. Open the DOS prompt and change to the OpenSSL directory,  
e.g. `cd My Documents\OpenSSL`
2. Run the following command:  
`perl CA.pl -signreq`
3. For "pass phrase", enter the password you specified in "pass phrase" when creating the Certification Authority. Confirm the signature by pressing "y" twice.

## 6.2.7 Generate the SIMOTION files

The key and certificate you created must be converted to a format which can be read by SIMOTION. Proceed as follows:

### Proceed as follows

1. Open the "newcert.pem" file in an available editor (e.g. Notepad).
2. Copy the section at the end of the file from "-----BEGIN CERTIFICATE-----" to "-----END CERTIFICATE-----"
3. Create a new document in the editor and paste the copied text. Save the new file as "MWSSLCer.pem".

---

#### Note

If you are using Notepad to edit the file, select "All Files" as the file type in the Save dialog. This prevents the file from being saved with the ".txt" extension.

---

4. Open the "newreq.pem" file in your editor.
5. Copy the section at the start of the file from "-----BEGIN RSA PRIVATE KEY-----" to "-----END RSA PRIVATE KEY-----"
6. Create a new document in the editor and paste the copied text. Save the new file as "MWSSLKey.pem".

---

#### Note

If you are using Notepad to edit the file, select "All Files" as the file type in the Save dialog. This prevents the file from being saved with the ".txt" extension.

---

### 6.2.8 Importing the certificate to the browser

If you use SSL with your own Certification Authority, you will need to prepare your PCs for communication with the SIMOTION CPU. To do this, the "SIMOTION.cer" certificate must be included in the list of root certificates.

#### Proceed as follows

You first need to edit the cacert.pem file as follows:

1. Copy the "cacert.pem" file (in the example in "My Documents\OpenSSL\demoCA").
2. Insert the copied file e.g. in "My Documents\OpenSSL".
3. Rename the file to "SIMOTION.cer".

The table below shows you to import the certificate (SIMOTION.cer) into your browser.

Browser	Proceed as follows
Internet Explorer 6.0	<ol style="list-style-type: none"> <li>1. Click "Internet Options" in the "Tools" menu.</li> <li>2. In the window that appears, click the "Content" tab to open it and select "Certificates".</li> <li>3. Click "Import Certificate" in the next window. A wizard guides you through the import. Accept the default settings in response to the questions.</li> </ol>
Konqueror 3.2.1	<ol style="list-style-type: none"> <li>1. Open the "Settings" menu and click "Configure Konqueror".</li> <li>2. Scroll down on the left of the window until the "Crypto" icon appears and then click on it.</li> <li>3. Click the "SSL Signers" tab to open it and activate the "Import" button.</li> <li>4. Insert the "SIMOTION.cer" file.</li> </ol>
Netscape 7.1	<ol style="list-style-type: none"> <li>1. Open the "Edit" menu and click "Preferences".</li> <li>2. In the "Privacy &amp; Security" category of the window that appears, click "Certificates".</li> <li>3. Click "Manage Certificates" to launch the manager.</li> <li>4. Click the "Authorities" tab to open it and activate the "Import" button.</li> <li>5. Enter the file name "SIMOTION.cer" in the dialog box.</li> </ol>
Mozilla 1.4	<ol style="list-style-type: none"> <li>1. Open the "Edit" menu and click "Preferences".</li> <li>2. In the "Privacy &amp; Security" category of the window that appears, click "Certificates".</li> <li>3. Click "Manage Certificates" to launch the manager.</li> <li>4. Click the "Authorities" tab and "Import".</li> <li>5. Accept the "SIMOTION.cer" file. Activate all settings.</li> </ol>



## 6.3 Transferring the key files to SIMOTION

To enable OPC XML DA access to SIMOTION via HTTPS, you will need to transfer the key files "MWSSLCer.pem" and "MWSSLKey.pem" to the memory card of the SIMOTION CPU.

In order to copy the key files onto the memory card you need a memory card reader/writer.

The procedure for transferring key files to the SIMOTION P350 is described separately.

### Proceed as follows

1. Insert the SIMOTION CPU memory card in the card reader/writer.
2. Copy the "MWSSLCer.pem" and "MWSSLKey.pem" files on the memory card to directory \USER\SIMOTION\HMICFG. Please create the directory if it does not already exist.
3. Insert the memory card in the SIMOTION CPU and switch the CPU on.

OPC XML DA access via HTTPS is available when the SIMOTION CPU is booted.

### Procedure for P350

1. Shut down the SIMOTION P.
2. Copy the "MWSSLCer.pem" and "MWSSLKey.pem" files on your P350 to directory F:\SIMOTION\USER\CARD\USER\SIMOTION\HMICFG (path in a default installation).
3. Start the SIMOTION P.

OPC XML DA access via HTTPS is available when the SIMOTION CPU is booted.



## File access via FTP

### Introduction

The "File access via FTP" function makes it possible to access specific files on the memory card. An FTP client can be, for example, Windows Explorer.

### Condition

In the WEBCFG.xml file, a user must be in the "FTPUser" group in order to log on to FTP. The WEBCFG.xml file is generated during the first power-up.

### Excerpt from the WEBCFG.xml file

In the excerpt from the WEBCFG.xml displayed below, the "FTPUser" group was assigned to the "simotion" user.

```
<UserDataBase>
  <FILE NAME="UserDataBase.xml">
    <![CDATA[
      <?xml version="1.0" encoding="UTF-8"?>
      <UserDataBase>
        <USER NAME="anonymous"
          PASSWORD="anonymous">
          <DESCRIPTION>Anonymous</DESCRIP
            TION>
          <GROUP NAME="Anyone"/>
          <GROUP NAME="OPC_XML"/>
        </USER>
        <USER NAME="internal"
          PASSWORD="internal">
          <DESCRIPTION>Internal
            user</DESCRIPTION>
          <GROUP NAME="Anyone"/>
        </USER>
        <USER NAME="simotion"
          PASSWORD="simotion">
          <DESCRIPTION>Default
            User</DESCRIPTION>
          <GROUP NAME="Administrator"/>
          <GROUP NAME="FTPUser"/>
        </USER>
      </CDATA>
    </FILE>
  </UserDataBase>
```

```

<GROUP NAME="Anyone" />
<GROUP NAME="OPC_XML" />
    </USER>
</UserDataBase>
]]>
</FILE>
</UserDataBase>

```



---

**Warning**

Caution while accessing system files.

---

## Appendix

### A.1 List of abbreviations

#### Abbreviations

AKZ	Higher-level designation of item
OKZ	Location designation
HTTPS	Secure HTTP
SSI	Server Side Include
SSL	Secure Socket Layer
CA	Certification Authority
YDB	SIMOTION data block



# Index

## B

Browse, 2-13

## C

CancelTrace, 3-7  
Client application, 2-10  
Communication package, 2-1  
ComplInfo group, 5-6

## E

Encrypted data transfer, 6-1

## G

GetProperties, 2-13  
GetStatus, 2-13, 3-7

## H

HTML pages  
    Active Traces, 4-12  
    Device Info, 4-7  
    Diag Buffer, 4-10  
    Home page, 4-5, 4-6  
    IP Config, 4-13  
    Mem Pool, 4-11  
    Settings, 4-8  
    Task runtime, 4-9  
    User-defined HTML pages, 4-21  
HTTPS, 6-1

## I

InitializeWebTrace, 3-10  
ItemName, 5-2  
ItemPath, 5-2

## O

OPC, 2-1  
OPC XML, 2-1  
OPC XML server interface, 2-13, 2-15

## R

Read, 2-13  
ReadData, 3-11

## S

Secure Socket Layer, 6-1  
Server Side Includes, 4-21  
Settings, 4-14  
    Operating mode, 4-16  
    System time, 4-16  
    Time zone, 4-16  
SOAP, 1-1, 2-1  
Standard pages, 4-5  
StartTrace, 3-7  
StopTrace, 3-7  
Subscribe, 2-14  
SubscriptionCancel, 2-14  
SubscriptionPolledRefresh, 2-14

## T

Task names, 5-9  
TaskRT group, 5-9  
TraceDataCycleEnum, 3-6  
TraceStateEnum, 3-6

## V

Variables groups, 5-4  
Variables providers  
    MemPool, 5-16  
    SIMOTION, 5-2  
    SIMOTION diagnostics, 5-4

**W**

Write, 2-14