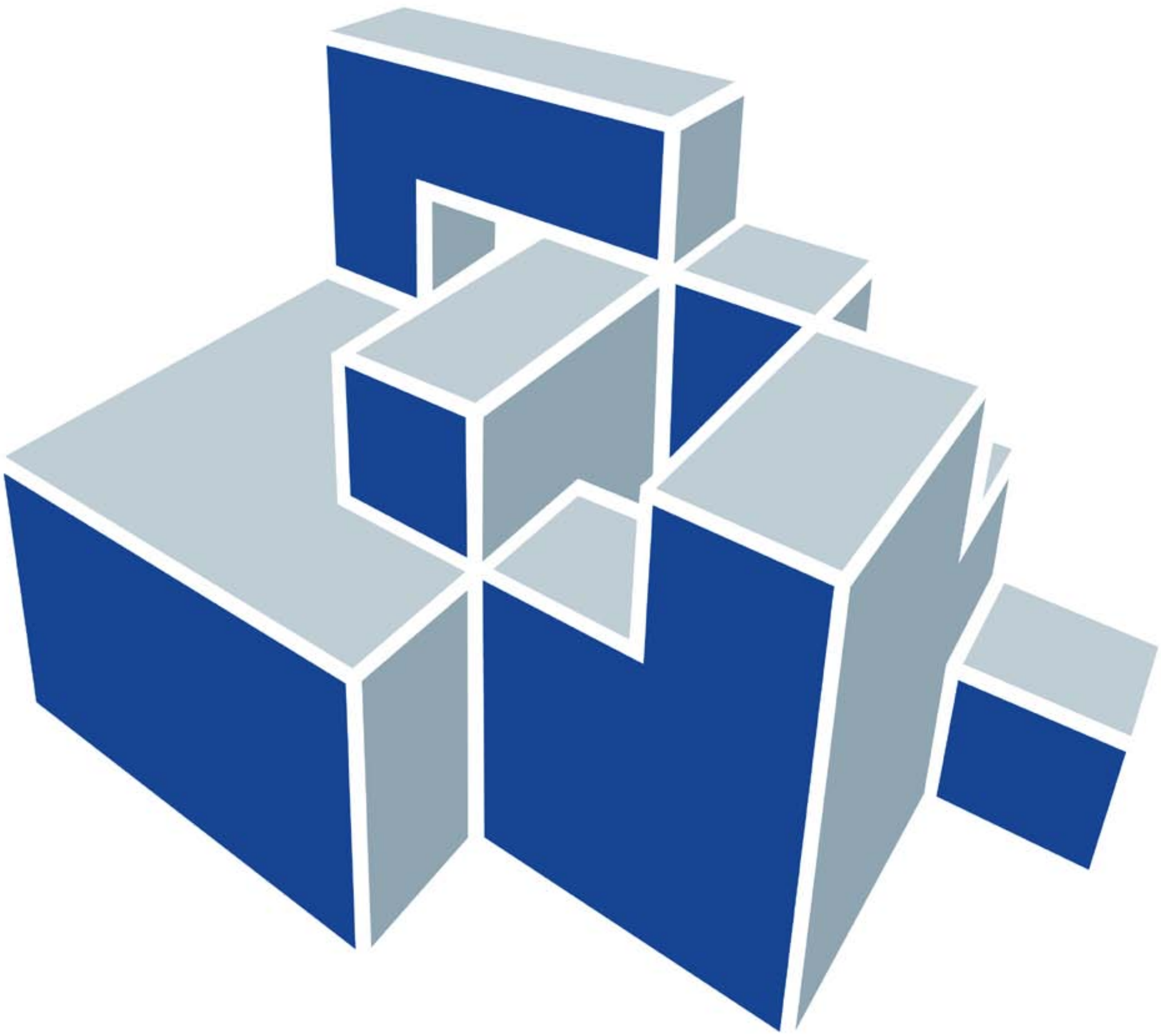
SIEMENS

SIMIT 7

Shared Memory Gateway

User Manual



SIEMENS

Edition

January 2013

Siemens offers simulation software to plan, simulate and optimize plants and machines. The simulation- and optimization-results are only non-binding suggestions for the user. The quality of the simulation and optimizing results depend on the correctness and the completeness of the input data. Therefore, the input data and the results have to be validated by the user.

Siemens AG
Industry Sector
Industry Automation Division
Process Automation

SIMIT HB-V7SHM-2013-01-en

Content

List of Figures

List of Tables

1 PREFACE

1.1 Target group

This manual is intended for anyone who uses the SIMIT simulation system. It describes how to use and handle a shared memory gateway (SHM) in SIMIT.

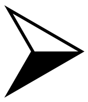
In addition to thorough knowledge of the use of personal computers and the Microsoft Windows user interface, it assumes knowledge of SIMIT. An understanding of the aspects common to all gateways, as described in the SIMIT gateways manual, is also necessary. As the SHM gateway is very similar to a Simatic gateway in SIMIT, a knowledge of the common features of Simatic gateways as described in the gateways manual is also assumed.

1.2 Contents

Section 2 describes how the SHM gateway works and section 3 contains a description of how to configure an SHM gateway.

1.3 Symbols

Particularly important information is highlighted in the text as follows:

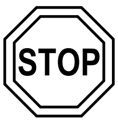


NOTE

Notes contain important supplementary information about the documentation contents. They also highlight those properties of the system or operator input to which we want to draw particular attention.

CAUTION

This means that the system will not respond as described if the specified precautionary measures are not applied.



WARNING

This means that the system may suffer irreparable damage or that data may be lost if the relevant precautionary measures are not applied.

2 HOW THE SHM GATEWAY WORKS

The shared memory gateway, or SHM gateway for short, can be used to exchange signals from SIMIT with any other application via a shared memory (SHM). This gateway provides a universal, high-performance signal interface to SIMIT.

In this gateway, as is usual in SIMIT gateways, input signals are signals written by SIMIT to the memory and output signals are signals read by SIMIT from the memory (see Figure 2-1).

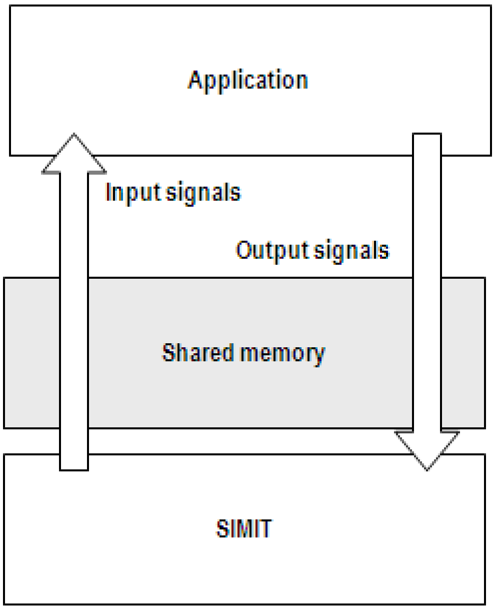


Figure 2-1: Direction of action of signals

2.1 Accessing the memory

If several mutually independent processes access the same memory, this access has to be synchronised in order to ensure the consistency of the values. For that reason a mutex is used in the SHM gateway as a synchronisation object.

In each simulation cycle SIMIT writes and reads all the input and output signals defined in the SHM gateway and blocks the mutex while it does so. All other applications that access this memory should do the same.

NOTE

Every application that is connected to SIMIT via an SHM gateway should keep the time for which the mutex is blocked as short as possible so as to avoid blocking access to the shared memory by SIMIT and other applications unnecessarily.

2.2 Structure of the memory

The memory is divided into a header and a data area. The header size is at least 8 bytes. The first four bytes of the header contain the size of the entire memory, the next four bytes the size of the header (see Figure 2-2). Both values are in little-endian format, as is usual in Microsoft Windows.

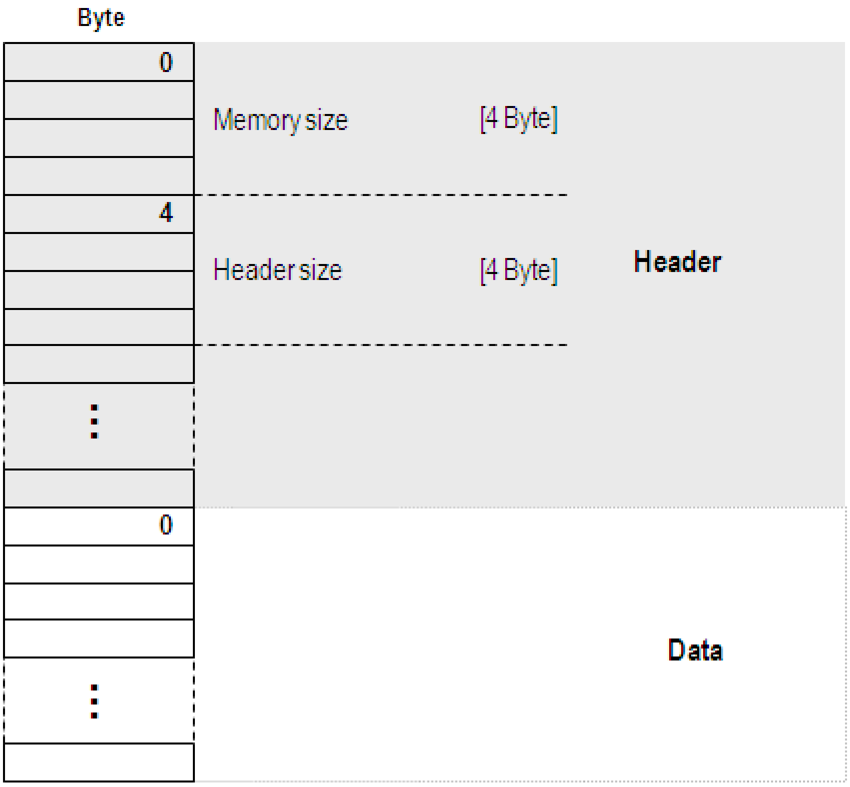


Figure 2-2: Structure of the memory

2.2.1 Structure of the data area

SIMIT addresses the data area in bytes similarly to the way in which Simatic automation systems address the I/O area: every signal from the SHM gateway is linked to a unique address in the data area. However, as a signal in an address in the data area can only be defined as an input signal or output signal, it is not permissible to assign the same address to both an input and an output signal. Therefore a signal occupies one, two or four bytes of the address in the data area, depending on its data type (see also section 3.2).

- One byte for BOOL and BYTE data types
- Two bytes for WORD and INT data types, and
- Four bytes for DWORD, DINT and REAL data types

An example of addressing for different data types is shown in Figure 2-3.

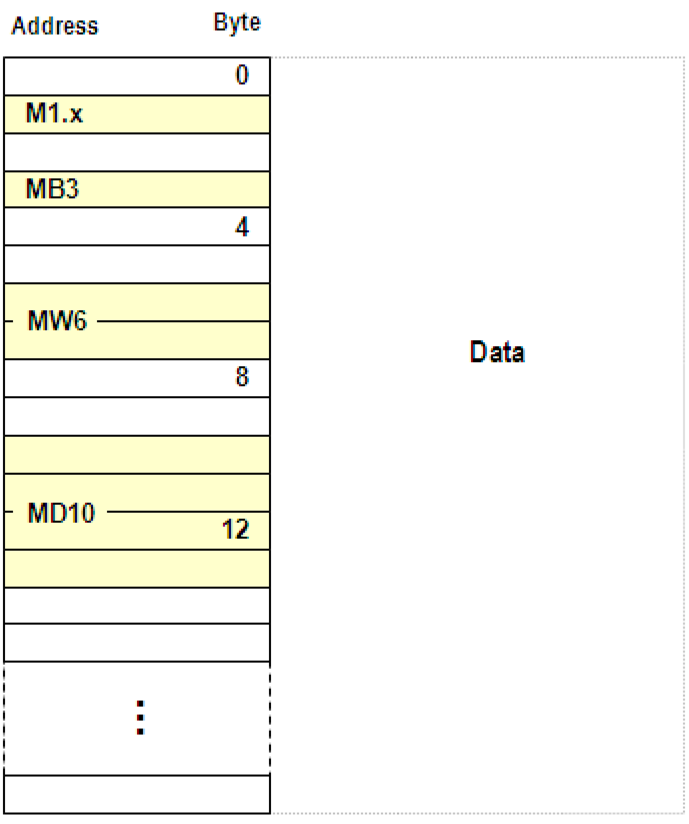


Figure 2-3: Addressing of the data area

2.2.2 Structure of the header

The minimum header size is 8 bytes. The first four bytes of the header contain the size of the entire memory, the next four bytes the size of the header (see Figure 2-4).

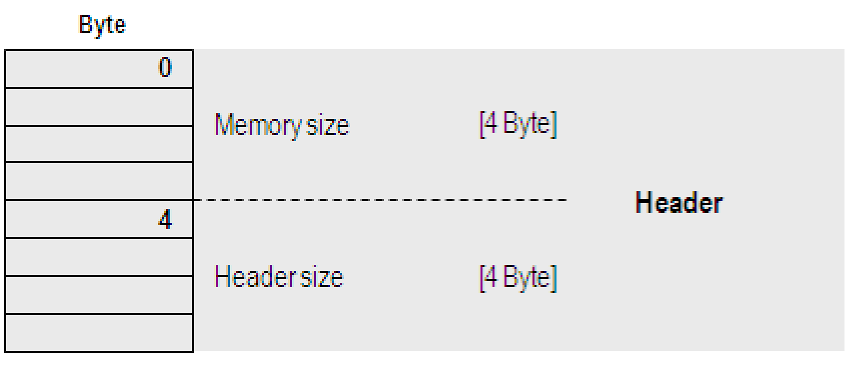


Figure 2-4: Minimum header structure

If SIMIT creates the shared memory, SIMIT can add additional variables and a list of signals to the header if required (see Figure 2-5). This information can then be used by applications to configure their access to the data area.

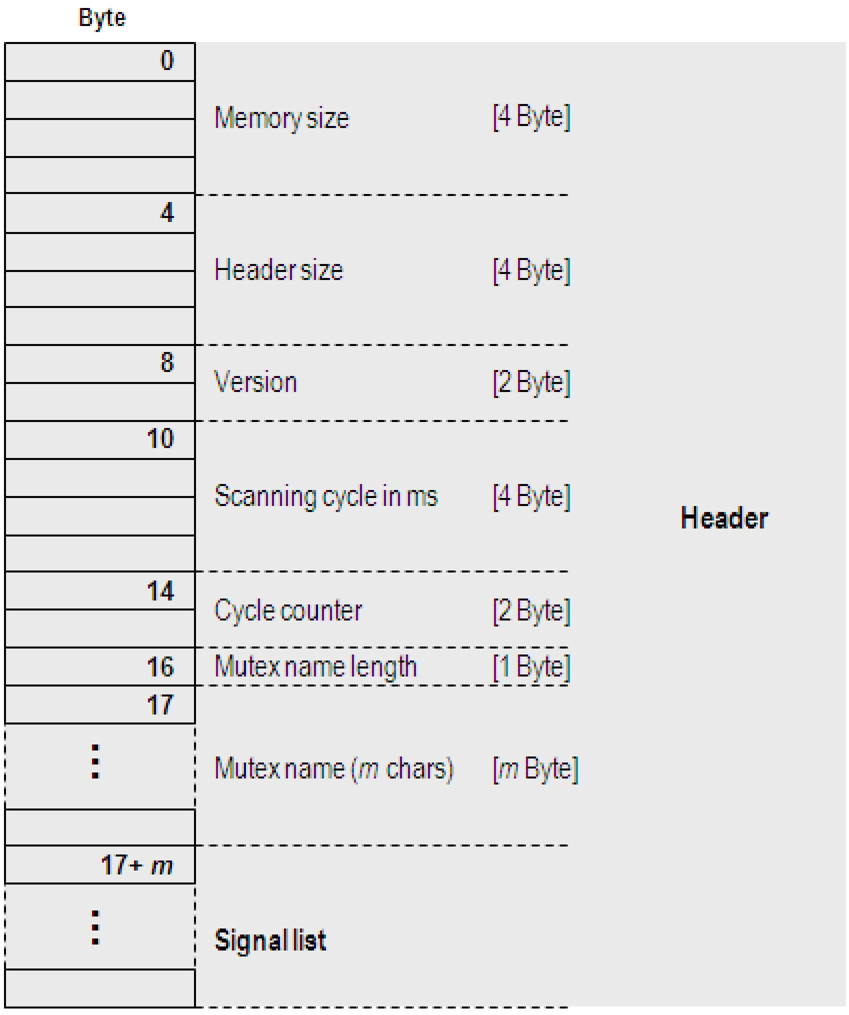


Figure 2-5: Extended header structure

The version identifies the memory structure. As the structure defined here always has the version code zero, you will always find the value zero entered there. If the structure of the memory area is altered, the version is changed accordingly.

The predefined scanning cycle for the SHM gateway is entered in milliseconds (ms) as a whole number. The value of the cycle counter is incremented by one in each cycle of the SHM gateway.

The length of the mutex name, i.e. the number *m* of characters in the mutex name, is stored in byte 16 of the header. The mutex name is stored in the adjacent header area from byte 17 onwards.

The signal list created from byte 17+*m* onwards in the header provides information about the signals in the data area. The following variables are specified for each signal:

- The length of the signal name, i.e. the number of characters in the signal name
- The signal name
- The signal address, i.e. the offset of the signal in the data area
- The I/O identifier, i.e. the identifier of whether the signal is an input signal (identifier *0*) or an output signal (identifier *1*), and
- The type identifier, to identify the data type (see Table 2-1)

Type identifier	Meaning (data type)
0	BOOL, bit address 0
1	BOOL, bit address 1
...	...
7	BOOL, bit address 7
8	BYTE
9	WORD
10	INT
11	DWORD
12	DINT
13	REAL

Table 2-1: Signal type identifiers

The structure of the signal list in the header is shown in Figure 2-6.

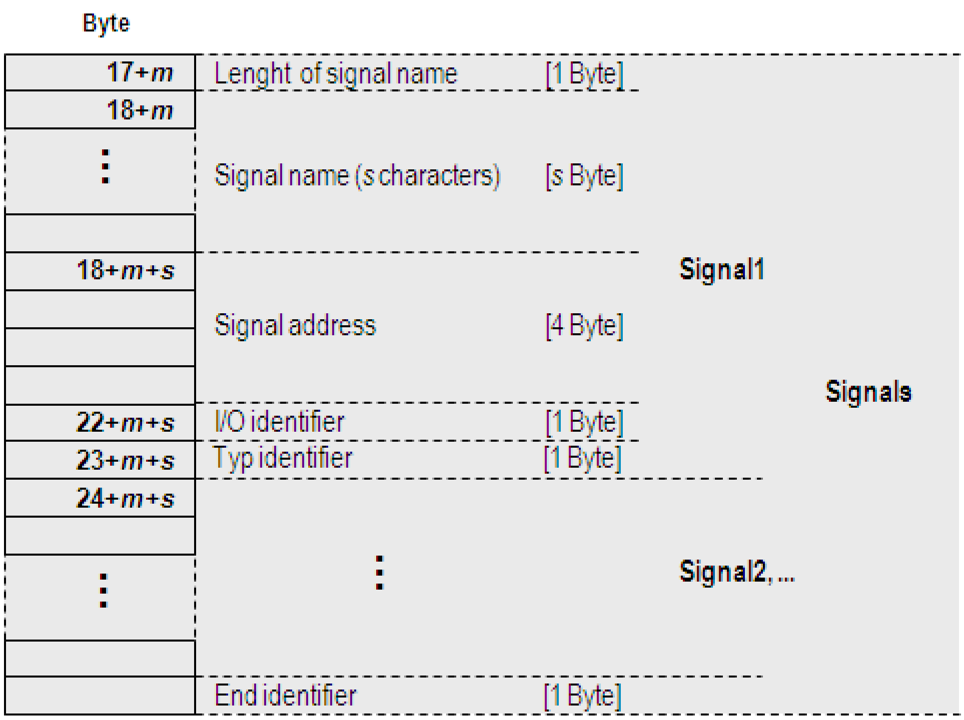


Figure 2-6: Structure of the signal list in the header

The signal list ends with an end identifier of the value zero.

2.3 Creating the memory

The shared memory can be created either by SIMIT or by another application that is connected to SIMIT via the memory. The way in which SIMIT behaves when a simulation containing an SHM gateway is started is adapted accordingly.

If the shared memory was created by an application, i.e. it already exists when the simulation is started in SIMIT, then SIMIT simply opens the memory. However, SIMIT only connects to the memory if the size of the data area corresponds to the size of the address area defined by the input/output signals in the SHM gateway. Otherwise an error message (see Figure 2-7) is displayed and SIMIT does not connect to the memory.

If the shared memory does not exist when SIMIT starts the simulation, then it is created by SIMIT. SIMIT enters the size of the entire memory and the header in the header. The size of the data area is determined by the highest address of the signals defined in the SHM gateway. As described in section 2.2 above, SIMIT can also enter additional variables and a list of signals in the header if required.

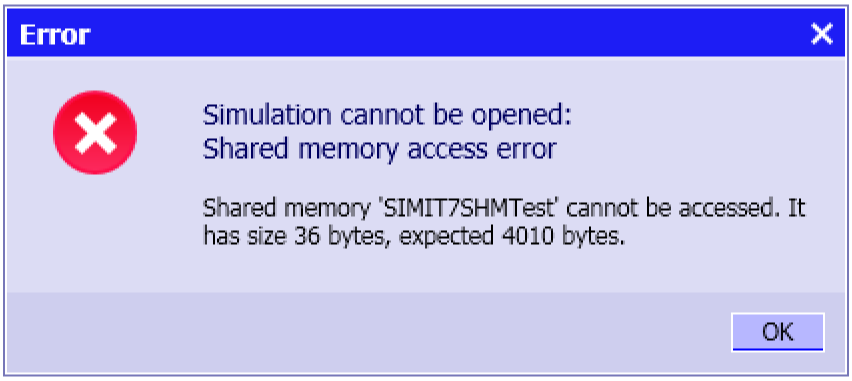


Figure 2-7: Error message reporting that the data area is the wrong size

3 CONFIGURING THE SHM GATEWAY

This section describes how to create and configure an SHM gateway in SIMIT.

3.1 Creating an SHM gateway

To add an SHM gateway to your simulation project, use the *New Gateway* item in the project tree to create a gateway. Select *SHM* as the gateway type (Figure 3-1).

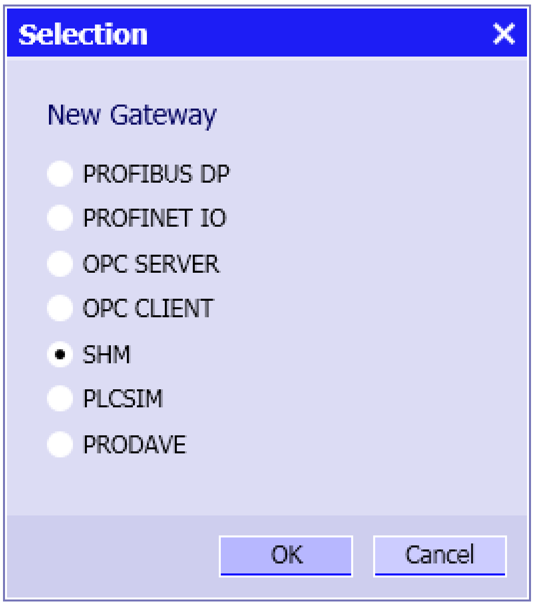


Figure 3-1: Creating an SHM gateway

You can choose any name for the SHM gateway in your SIMIT project. To accept the suggested name, simply hit the Enter key. The gateway editor then opens automatically. You can also open the gateway editor at a later date by double-clicking the gateway in the project tree or via the context menu (Figure 3-2).

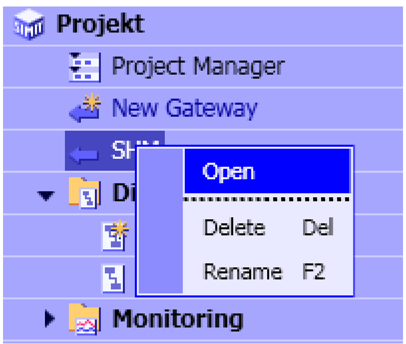
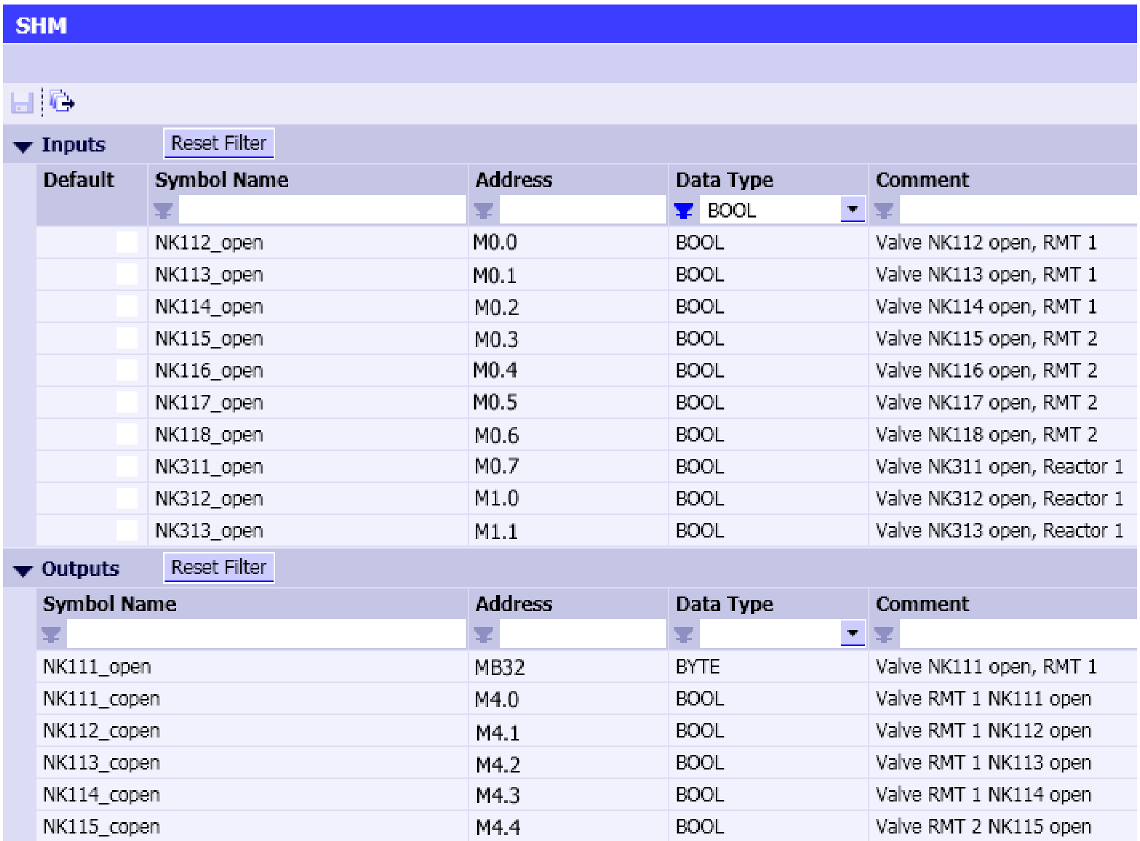


Figure 3-2: Opening the SHM gateway for editing

3.2 Configuring the signals in the SHM gateway

The input/output signals of the SHM gateway can either be entered manually in the SHM gateway editor or configured by importing a signal list as described in section 4.7 of the gateways manual. Figure 3-3 shows the open SHM gateway editor.



Default	Symbol Name	Address	Data Type	Comment
			BOOL	
	NK112_open	M0.0	BOOL	Valve NK112 open, RMT 1
	NK113_open	M0.1	BOOL	Valve NK113 open, RMT 1
	NK114_open	M0.2	BOOL	Valve NK114 open, RMT 1
	NK115_open	M0.3	BOOL	Valve NK115 open, RMT 2
	NK116_open	M0.4	BOOL	Valve NK116 open, RMT 2
	NK117_open	M0.5	BOOL	Valve NK117 open, RMT 2
	NK118_open	M0.6	BOOL	Valve NK118 open, RMT 2
	NK311_open	M0.7	BOOL	Valve NK311 open, Reactor 1
	NK312_open	M1.0	BOOL	Valve NK312 open, Reactor 1
	NK313_open	M1.1	BOOL	Valve NK313 open, Reactor 1

Symbol Name	Address	Data Type	Comment
NK111_open	MB32	BYTE	Valve NK111 open, RMT 1
NK111_copen	M4.0	BOOL	Valve RMT 1 NK111 open
NK112_copen	M4.1	BOOL	Valve RMT 1 NK112 open
NK113_copen	M4.2	BOOL	Valve RMT 1 NK113 open
NK114_copen	M4.3	BOOL	Valve RMT 1 NK114 open
NK115_copen	M4.4	BOOL	Valve RMT 2 NK115 open

Figure 3-3: SHM gateway editor

SIMIT reads the output signals from and writes the input signals to the shared memory. Access takes place cyclically in the cycle defined for the gateway. One byte of the memory must be clearly assigned to either an input or an output signal. In addition, each input signal must be clearly mapped in the data area. The data areas of different input signals must not overlap.

Figure 3-4 shows an example of two input signals, MD3 and MW6, which overlap in byte 6. In the consistency check by SIMIT, overlapping signals are reported as an error. Figure 2-7 shows a typical error message when a binary signal (M0.0 in this example) is in the data area of another signal (MB0 in this example).

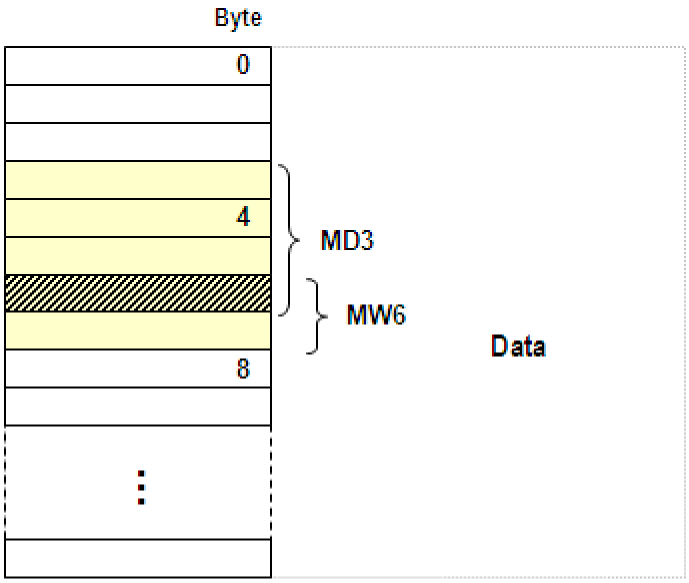


Figure 3-4: Overlapping input signals in the data area

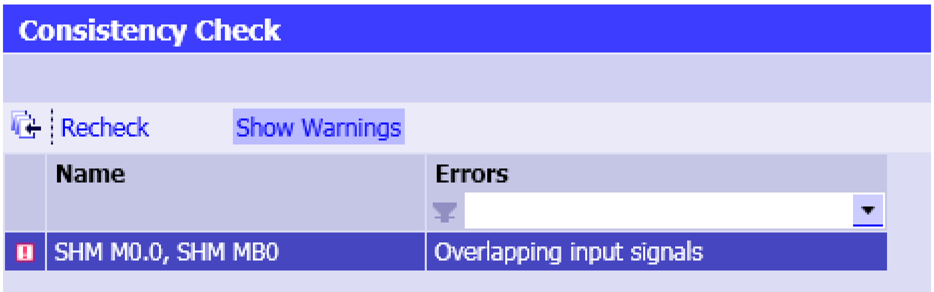


Figure 3-5: Error message reporting overlapping signals in the consistency check

---

CAUTION

As the smallest addressable unit of the memory is a byte, a byte can only be assigned to either an input signal or an output signal. Therefore binary signals with the same byte address can only be either all binary input signals or all binary output signals.

In addition, write access to the memory by SIMIT is in bytes rather than in bits. A bit in a byte of the memory for which no binary input signal is defined in the SHM gateway is set to zero during write access to the memory by SIMIT.

The address designation of a signal begins with "M" for memory and contains the data type and the address. Based on the notation for addresses in Simatic automation systems, the data types listed in Table 3-1 can be used.

Data type	Size	Notation	Value range
BOOL	1 bit	M<byte>.<bit>	True/False
BYTE	1 byte (8 bits)	MB<byte>	0 ... 255 or -128 ... +127
WORD	2 bytes	MW<byte>	0 ... 65,535
INT	2 bytes	MW<byte>	-32,768 ... 32,767
DWORD	4 bytes	MD<byte>	0 ... 4,294,967,295
DINT	4 bytes	MD<byte>	-2,147,483,648 ... 2,147,483,647
REAL	4 bytes	MD<byte>	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$

Table 3-1: Definition of data types

The following applies to the mapping of integer signals in SIMIT simulation projects to data types in the memory:

The least significant byte (LSB) of the signal is used for the BYTE data type. The WORD and DWORD data types are unsigned and the INT and DINT data types are signed. The values are limited to the value ranges specified in Table 3-1.

The structure of a floating-point number is governed by the IEEE Standard for Binary Floating Point Arithmetic (ANSI/IEEE Std 754-1985).

3.3 Signal properties

The properties of a signal are shown in the individual columns of the editor window and in the Properties dialog of the gateway editor (see Figure 3-6).

NK112_open

Property	Value
Symbol Name	NK112_Setting
Address	MW100
Data Type	WORD
Comment	Valve NK112 open, RMT 1

Figure 3-6: Signal properties in the Properties dialog

The following signal properties are defined:

- **Symbol name**

  The signal is identified by this name in SIMIT.

- **Address**

  Signal values are stored in the data area of the shared memory under this byte address. The address zero corresponds to the first byte after the header.

- **Data type**

  A signal's data type determines the space it occupies in the data area and how the values stored there should be interpreted: as a logical value, as a signed or unsigned integer or as a floating-point number.

- **Comment**

  The comment is used to document the signal. It is not evaluated by SIMIT.

3.4 Properties of the SHM gateway

The properties of the SHM gateway can be defined in the Properties dialog (Figure 3-7) of the gateway editor:

SHM	
Property	**Value**
Cycle	2
Shared Memory Name	SIMITSHM
Mutex Name	SIMITSHMMutex
Signal description in header	☐
Header Size	8
Big/Little Endian	little

Figure 3-7: Properties of the SHM gateway

- **Cycle**

  Like any SIMIT gateway, the SHM gateway is processed in fixed cycles. Specify which of the eight possible time slices of the SIMIT project the gateway should be assigned to.

- **Shared Memory Name**

  Enter the name with which the shared memory can be addressed.

- **Mutex Name**

  Enter the name of the mutex for synchronising access to the shared memory.

- **Signal description in header**

  This option allows you to choose whether SIMIT should created an extended header (see section 2.2.2).

- **Header Size**

  Enter the size of the header in bytes. The minimum value is 8 bytes. When SIMIT creates the shared memory it creates a header of the specified size.

NOTE

If you activate the "Signal description in header" option, the size of the header is determined by SIMIT from the input/output signals in the gateway. The "Header size" property is not editable in that case.

- **Big/Little Endian**

  This property determines the byte order in which values of the WORD, INT, DWORD and DINT data types are coded in the data area (Table 3-2).

Setting	Byte order
Big endian	The most significant byte is stored first, i.e. at the lowest memory address.
Little endian	The least significant byte is stored first, i.e. at the lowest memory address.

Table 3-2: Byte order

3.5 Importing and exporting signals

Gateway signals can be exported and imported as a text file with a ".txt" extension. The text file contains five columns separated by tabs (see Table 3-3). Additional columns are permitted by the import operation but are ignored by SIMIT.

Column	Designation	Description
1	Symbol	Symbolic name of the signal (optional)
2	I/O	Labelled *E*, *A*, *EB*, *AB*, *EW*, *AW*, *ED*, *AD* or in corresponding international nomenclature (I/Q)
3	Address	Absolute address of the signal, e.g. *0.0* or *512*
4	Type	Signal data type: *BOOL*, *BYTE*, *WORD, DWORD, INT, DINT or REAL*
5	Comment	Text as comment (optional)

Table 3-3: Signal table format

To specify whether a signal is an input or output signal, signals are labelled in column 2 with "I" for "input" or "Q" for "output" rather than "M". "E" and "A" are also permitted instead of "I" and "Q".

Simatic symbol tables can also be imported in asc, seq and xlsx format as well as txt format. In that case in particular, however, note that in the SHM gateway the addresses for inputs and outputs have to be unique, i.e. an input signal and an output signal may not have the same address.

CAUTION

If you edit the signal table in Excel, ensure that all cells are formatted as "text" so that Excel does not make any unintended format conversions.