

# IP-S7-LINK

## RFC 1006

### for

# PC – SIMATIC S7

S7 200 – 1200 – 300 – 400

MS-Windows 2000/NT/XP/Vista/7, Linux

C/C++, c#, VB, VB.net, Delphi, Excel, Access, PHP and others



Version 1.47 (25.11.2011)

**Requirements:**

<b>operating systems</b>	MS-Windows 95 ,98, NT, 2000, XP,Vista, Windows 7 Linux
<b>programming languages</b>	C, C++, Delphi, VisualBasic, C-Sharp, VB.net, Access, Excel, PHP
<b>Hardware</b>	PC with installed TCP/IP-protocol and ethernet card
<b>PLC</b>	Simatic S7 200/300/400 with CP 243, CP343-1, CP-443-1, or LeanCP, PLC with ProfiNet S7 1200, or S7-LAN

**Delivery:**

The following files can included:

<b>main folder</b>	
Ips7lnk.pdf	german manual
ips7lnk-en.pdf	english manual
Version.htm	release information

<b>folder ´win32´</b>	<b>Files Windows 32 - Bit</b>
IPS7LNK.DLL	Driver DLL
IPS7LNK.LIB	Lib-File to be linked with C++

<b>Folder ´x64´</b>	<b>Files Windows 64 – Bit</b>
IPS7LNK.DLL	Driver DLL
IPS7LNK.LIB	Lib-File to be linked with C++

<b>Folder ´CPP´</b>	<b>Files for Visual CPP C++</b>
IPS7LNK.H	Header-File for C / C++
IPS7LNK.DLL	Driver DLL
IPS7LNK.LIB	Lib-File to be linked with C++
IPS7DEMO.DSP	Project file for Visual C++ V 6.00
IPS7DEMO.CPP	sample program ´C++´
IPS7DEMO.EXE	EXE-File der CPP-Demo

<b>Folder ´Delphi´</b>	<b>Files for Delphi</b>
IPS7LNK.PAS	Delphi-Header TPU im Source code
IPS7LNK.DLL	Driver DLL

<b>Folder 'Delphi'</b>	<b>Files for Delphi</b>
IPS7DEMO.exe	EXE-File of Delphi demo
IPS7DEMO.cfg IPS7DEMO.dof IPS7DEMO.dpr IPS7DEMO.res IPS7LNK.dcu main.dcu main.dfm main.pas OEM.BMP	Delphi - Project files

Folder ' VisualBasic'	Files for Visual Basic
IPS7LNK.DLL	Driver DLL, <b>Note: For Visual Basic and Excel, copy this file to windows system folder!</b>
IPS7LNK.BAS	Header / Module file for Visual Basic
IPS7DEMO.EXE	EXE-File of VB demo
IPS7DEMO.FRM IPS7DEMO.FRX IPS7DEMO.VBP IPS7DEMO.VBW	Visual Basic project files

Folder Excel	Files for Excel
IPS7LNK.DLL	Driver DLL, <b>Note: For Visual Basic and Excel copy these files to windows system folder!</b>
IPS7LNK.BAS	Header/Module file for VisualBasic
IPS7DEMO.XLS	Excel-File with macro for demo

Folder ' PHP'	Files for PHP
ips7lnk_php.so	Module of the PHP extension, copy ths file to the extension folder of your PHP installation

Folder ' DotNet'	Files for. Net
DemoCSharp	Folder demo program C#
DemoVB.net	Folder demo program VB.net
NetFiles	Folder with assemblies for the different frameworks (2.0/3.0/3.5)
ips7lnk.chm	Helo file / manual for .net Assemblies

**Installation:**

**Windows: Copy the DLL to the program folder of your application or to the Windows system directory.**

**Linux: link the .o-file to your application:**

For .net see the documentation in ips7lnk.chm-file. For .net (C# and VB.net) all functiones are embedded in a class. There exist assemblies for frameworks 2.0/3.0/3.5. These assemblies are using the ,ips7lnk.dll'. Take care to copy ,ips7lnk.dll' to your program folder or to the Windows system folder. Put the corresponding assembly ips7lnknet.dll as reference to your application. So all functions are available. Copy als this file to your program folder or to the Windows system folder

**PHP:****Copy the extension ips7lnk\_php.so to the extension folder of your PHP installation**

Take care that the extension is loaded. Do in PHP.ini or in your program by loading dynamically.

for php.ini: extension = ips7lnk\_php.so

in your program: `ld ('ips7lnk_php.so');`

**How it works:**

IP-S7-LINK is a DLL for MS-Windows (95/98/2000/NT/XP/Vista/7) or a library or an extension or „.o-file“ for Linux (C,C++,PHP). With IP-S7-LINK you can connect your PC with industrial ethernet of Siemens SIMATIC S7 systems from. „Easy to use“- functions make it very easy to communicate with the PLC with C, C++, Delphi, Visual Basic, Excel or PHP. For connection to the PLC you need only the ip address , number of rack and slot of the PLC. So you can read and write Flags, In, Out, Timer, Counter and data blocks of the PLC.

**Description in detail:**

Please note: The functions are using the standard socket interface, with the result that the function returns only after performing the task to the caller. For asynchronous operation, call these functions easily from a separate thread on which to communicate to the system.

The following functions are available:

**Functions for initialization:**

<i>Function</i>	<i>Function (PHP)</i>	<i>Description / Purpose</i>
<b>IPS7Open</b>	<b>ips7_open</b>	to initialize the connection, there is only memory prepared. The tcp/ip connections is to be started with the first call to the read or write functions. <b>The connection is done over the OP-Channel of the PLC.</b>
<b>IPS7OpenPG</b>	<b>ips7_openpg</b>	Version 1.17 to initialize the connection, there is only memory prepared. The tcp/ip connections is to be started with the first call to the read or write functions. <b>The connection is done over the PG-Channel of the PLC.</b>
<b>IPS7OpenS7200</b>	<b>ips7_opens7200</b>	Version 1.21 to initialize the connection, there is only memory prepared. The tcp/ip connections is to be started with the first call to the read or write functions. <b>The connection is done to a S7-200 PLC.</b>

**Calling parameters:**

<b>N o.</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
1	32-Bit Pointer to C-String	string	IPAdr	IP address of PLC format: xxx.xxx.xxx.xxx. e.g.: "192.169.0.100"
2	32-Bit Value unsigned	long	Rack	Number of racks ,in which the PLC-CPU is plugged in. The numbering starts at „0“. Usually: 0
3	32-Bit Value unsigned	long	Slot	Number of slot in which CPU is plugged in. Numbering starts at „1“, usually „2“.Don't care with S7-200 systems.

<b>N o.</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
4	32-Bit Value unsigned	long	RxTimeout	timeout in milliseconds waiting for TCP/IP packed from the PLC „0“ means default setting = 500 ms
5	32-Bit Value unsigned	long	TxTimeout	timeout in milliseconds transmitting a TCP/IP packed to the PLC. “0” means default setting= 500 ms
6	32-Bit Value unsigned	long	ConTimeout	Timeout in milliseconds waiting for connection to the PLC “0 „ means default setting: = 5000 ms ( 5sec.) Must be changed if necessary.

Function	Function (PHP)	Description / Purpose
<b>IPS7OpenEx</b>	<b>ips7_openex</b>	Version 1.23 to initialize the connection, there is only memory prepared. The tcp/ip connections is to be started with the first call to the read or write functions. <b>With parameter „AccessMode“ you can select the connection channel or the PLC type (S7 200/300/400) or you can make connection over a S7-Subnet.</b>

**calling parameters:**

N o.	Data type	Data type (PHP)	Name	Function
1	32-Bit Pointer to C-String	string	IPAdr	IP address of PLC format: xxx.xxx.xxx.xxx. sample: 192.169.0.100
2	32-Bit Value unsigned	long	Rack	Number of racks ,in which the PLC-CPU is plugged in. The numbering starts at „0“. Usually: 0
3	32-Bit Value unsigned	long	Slot	Number of slot in which CPU is plugged in. Numbering starts at „1“, usually „2“.Don't care with S7-200 systems.
4	32-Bit Value unsigned	long	SubNetId	Subnet-ID, if you want to connect over S7-Subnet. In Step-S7 software the address ist displayed as e.g.: 0035 – 0001 So give 0x00350001 as parameter. This paramter is only used for AccessMode 10 or 11. For other AccessModes don't care.
5	32-Bit Value unsigned	long	DstMPIAdr	Destination MPI address, if you want to connect over S7-Subnet This paramter is only used for AccessMode 10 or 11. For other AccessModes don't care.

<b>N o.</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
6	32-Bit Value unsigned	long	AccessMode	Mode of access 0 = connect over OP-channel with rack and slot number 1 =connect over PG-channel with rack and slot number 2 = connect to S7-200 over inserted TCP/IP-CP of the PLC 10 = connect over OP-channel over subnet connected to the PLC which ist addressed with rack and slot number, subnet-ID and DstMPI address. 11 = connect over PG-channel over subnet connected to the PLC which ist addressed with rack and slot number, subnet-ID and DstMPI address..
7	32-Bit Value unsigned	long	RxTimeout	Timeout in milliseconds receiving TCP/IP-packet from the PLC. 0 means default value: = 500 ms
8	32-Bit Value unsigned	long	TxTimeout	Timeout in milliseconds sending TCP/IP-packet to the PLC. 0 means default value: = 500 ms
9	32-Bit Value unsigned	long	ConTimeout	Timeout in milliseconds establishing the IP connection to the PLC. 0 means default value: = 5000 ms ( 5sec.) increase if necessary.

**Return value of the „open functions“:**

The return value of the „open-functions“ is a signed 32-bit value with the meaning:

<b>Value</b>	<b>Description of error</b>	<b>meaning</b>
>= 0	Everything okay	Return value ist the reference number for this connection. This calue ist to be use to call the other functions with the input parameter „Ref“.
-2	No resource available.	maximum of the available connections is reached.
-10	AccessMode invalid (from 1.23)	the value passed for AccessMode is invalid. see IPS7OpenEx

<i>Function</i>	<i>Function</i>	<i>Description / Purpose</i>
<b>IPS7Connect</b>	<b>ips7_connect</b>	Executes an explicit connection to the PLC. Since Version 1.35! This function does only connect to PLC without reading or writing. Usefull if you want to check the connection before the first call tor read or write.

**Calling parameters:**

<b>Nr</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
1	32-Bit Value unsigned	long	Ref	Reference of connection generated with IPS7Open/IPS7OpenEx. Used to internally identify the connection.

**Return value:**

IPS7Connect returns a signed 32-Bit Value. The meaning is:

<b>Value</b>	<b>Description of error</b>
1	connection established
<= 0	connetion was not established. For detail see the return value of the read/write-functions.

<i>Function</i>	<i>Function</i>	<i>Description / Purpose</i>
<b>IPS7GetConnectStatus</b>	<b>ips7_getconnectstatus</b>	check the status of TCP/IP connection to the PLC.

**Calling parameters:**

<b>Nr</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
1	32-Bit Value unsigned	long	Ref	Reference of connection generated with IPS7Open/IPS7OpenEx. Used to internally identify the connection.

**Return value:**

IPS7Connect returns signed 32-bit value. The meaning is:

<b>Value</b>	<b>Description</b>
1	connection established.
<= 0	connection disconnected. Call read/write functions again or all call IPS7Connect to connect.

<i>Function</i>	<i>Function</i>	<i>Description / Purpose</i>
<b>IPS7SetKeepAlive</b>	<b>ips7_setkeepalive</b>	Since V 1.35! To set individual TCP/IP KeepAlive times. Use only if you want to change the default TCP/IP settings. Call this function immediately after calling "Open"-function.

**Calling parameters:**

<b>Nr</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
1	32-Bit Value unsigned	long	Ref	Reference of connection generated with IPS7Open/IPS7OpenEx. Used to internally identify the connection.
2	32-Bit Value unsigned	long	AliveTime	If there is no traffic in "AliveTime" (ms) for the TCP/IP-connection so a KeepAlive-packet is sent to check the connection. If an error happens the IP-Stack retries the in "AliveInterval" (ms) the next KeepAlive-packet. This is repeated several times within "AliveInterval". (Windows 6 times). If not successfully the connection is closed.
3	32-Bit Value unsigned	long	AliveInterval	Interval in ms the KeepAlive packet to be repeated. The time gets active if an error happened transmitting or receiving a KeepAlive-packet.

**Return value:**

IPS7SetKeepAlive returns a signed 32-Bit Value. The meaning is:

<b>Value</b>	<b>Description</b>
0	setting the value was successfully.
< 0	KeepAlive-time couldn't be set.

<i>Function</i>	<i>Function</i>	<i>Description / Purpose</i>
<b>IPS7Close</b>	<b>ips7_close</b>	Used to deinitialize the connection. The TCP/IP connection is closed, allocated memory is released.

**Calling parameters:**

<b>Nr</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
1	32-Bit Value unsigned	long	Ref	Reference of connection generated with IPS7Open/IPS7OpenEx. Used to internally identify the connection.

**Return value:**

The Functions returns a signed 32-Bit value. The meaning is:

<b>Value</b>	<b>Description of error</b>	<b>meaning / action</b>
0	everything okay	Connection closed, allocated memory released
-3	With the reference number passed, no open function was called successfully.	Call any Open function!
-99 -30 -31	reference numbe invalid only PHP only PHP	----- number or type of passed parmeters in valid the internal conversion of data not successfully, e.g. a string was passed but long was expected.

## Functions for reading and writing

Function	Function (PHP)	Description / Purpose
<b>IPS7RdW</b>	<b>ips7_rdw</b>	reading word by word (16-bit) from PLC (I, O, F, DB )
<b>IPS7RdPlcW</b>	<b>ips7_rdplcw</b>	reading word by word (16-bit) from PLC (I, O, F, DB ) Note the start address is calculated in PLC-address mode (from 1.17) so you can access to odd start addresses in the PLC
<b>IPS7RdB</b>	<b>ips7_rdb</b>	reading byte by byte (8 bit) from PLC (I, O, F, DB )
<b>IPS7WrW</b>	<b>ips7_wrw</b>	writing word by word (16 bit) to PLC (I, O, F, DB, C)
<b>IPS7WrPlcW</b>	<b>ips7_wrplcw</b>	writing word by word (16 bit) to PLC (I, O, F, DB, C) Note the start address is calculated in PLC-address mode (from 1.17) so you can access to odd start addresses in the PLC
<b>IPS7WrB</b>	<b>ips7_wrb</b>	writing byte by byte (8 bit) to PLC (I, O, F, DB, C)
<b>IPS7RdDW</b>	<b>ips7_rddw</b>	reading dword by dword (32-bit) from PLC (I, O, F, DB, T)
<b>IPS7WrDW</b>	<b>ips7_wrdw</b>	writing dword by dword (32 bit) to PLC (E, A, F, DB, T)
<b>IPS7RdReal</b>	<b>ips7_rdreal</b>	reading floating point values (I, O, F, DB )
<b>IPS7WrReal</b>	<b>ips7_wrreal</b>	writing floating point values (I, O, F, DB )

**Before reading/writing word by word to the PLC check if you want to use the starting address like PLC-programming-syntax or if you want to use the arithmetical correct starting address.** You have to select the right function IPS7RdPlcW/IPS7WrPlcW or IPS7WrW/IPWrPlcW For details see **New in Version 1.17!**

### **Explanation for PHP**

In PHP variables are in principle not assigned to a specific data type. The determination of the data type processes the extension module.

Basically:: The destination variable respectively source variable representing the read/write buffer (=parameter no. 6) have to be passed by reference. You have to use the „&“ - character.

z.B. \$Res = ips7\_rdplcw (\$Ref, ord ("M"), 0, 0, 2, &\$Value);

Value[0] is MW0 (Flag word)

Value[1] is MW2 (Flag word)

In PHP there do not exist 16-bit-values, instead all 16-bit-data from PLC are saved as long in PHP.

The word-functions will save the result to an array of long. If you read only one single item, so the result is saved to one single long (If the variable passed isn't yet an array. Reading and writing of words (16-bit) is basically processed as a signed value. That is the value is interpreted as a 16-bit integer. If the value should be processed as unsigned 16-bit integer, so

you have to use the optional parameter 7 (bSigned).

The byte functions basically save the result as string. If you want to save the result an array of longs set bLong = 1.

**Calling parameters:**

**The read/write functions have the same calling parameters:**

<b>Nr</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
1	32-Bit Value unsigned	long	Ref	Reference of connection generated with IPS7Open/IPS7OpenEx. Used to internally identify the connection.
2	32-Bit Value unsigned	long	Typ	<p>Selecting the memory area of the PLC to be processed(DB, In, Out,Flags...):</p> <p>'D' = 68 dec. Datablocks (DB)</p> <p>'E' = 69, dec. In (Inputs)</p> <p>'A' = 65 dec. Out (Outputs)</p> <p>'M' = 77 dec. Flags (in german „Merker“)</p> <p>'T' = 84 dec. Timer (access only as double word, 32 bit)</p> <p>Timers of the PLC are saved with base of time and value in BCD – format. IP-S7-LINK automatically converts the format an 32 bit value in units of milliseconds. The resolution is 10 ms. Writing time values, the drivers will select the best time base in the PLC. Note: Depending of the size of the value an the selected time base, there will be resolution errors. Allowed values from 0 to 9990000 ms</p> <p>'Z' = 90 dec. Counter (german Zähler), supported ony word functions</p> <p>In the PLC counters are also saved in BCD format. Values for counter are running frome 0 to 999. The driver converts automatically to and from decimal fromat.</p>
3	32-Bit Value unsigned	32-Bit Value unsigned	DBNr	Number of data block, only used with type 'D'. Otherwise use value „0“
4	32-Bit Value unsigned	32-Bit Value unsigned	Ab	<p>Number of the first byte/word to be read from /written to.</p> <p>With word operations (IPS7RdW/IPS7WrW) = start word.</p> <p>With byte/word (IPS7RdPLCW/IPS7WrPLCW) dword/floating point functions= start byte.</p> <p>Accessing timer or counter this is the number of the first element to be processed.</p>

<b>Nr</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
5	32-Bit Value unsigned	32-Bit Value unsigned	Anz	Count of items (byte, words, double words, floating points, timer counter), to be read or written.
6	32-Bit Pointer	mixed	Buffer	<p>Pointer / address of the destination/source buffer in your pc program. Calling word functions this is a pointer to 16-bit values, calling byte functions this is a pointer to 8-bit values. Calling double word functions this is a pointer to 32-bit values. Calling floating point values this is a pointer to double values.</p> <p><b>Note for PHP:</b></p> <p>In PHP pass the refernz of the variable: calling:  ips7_rdplcw  ips7_rdw  ips7_rddw  ips7_rdreal</p> <p>Call e.g.: Result = ips7_rdplcw (\$Ref, ord ("M"), 0, 6,5, &amp;\$W);</p> <p>Reading more than one single item the is converted in an arry of longs or doubles. Reading only one single Value the variable will be converted in long/double if this variable were no array before.</p> <p>If this variable is yet an array and reading an one single item. The result is saved in the first element of the array.</p> <p>Double words (ips7_rddw) are basically processed as signed values.</p>

<b>Nr</b>	<b>Data type</b>	<b>Data type (PHP)</b>	<b>Name</b>	<b>Function</b>
7	-----	long (optional)	bSigned with ipss7_rdplcw ips7_rdw ipss7_wrplcw ips7_wrdw  bLong with ips7_rdb	<p><b>Note for PHP:</b></p> <p><b>ipss7_rdplcw</b> <b>ips7_rdw</b> <b>ipss7_wrplcw</b> <b>ips7_wrdw</b></p> <p>Optionally you can select the value is processed as signed or unsigned 16-bit integer. If this parameter is not passed the value will be processed as 16-bit signed integer. Passing the parameter the rule takes place:</p> <p>0 = unsigned 1 = signed</p> <p>To convert a single the variable after calling you can use the functions <b>ips7_i2w</b> ord <b>ips7_w2i</b>.</p> <p>For details look at these functions.</p> <p><b>ips7_rdb</b></p> <p>Ips7_rdb saves the result as string. If you like to save the values as an array set bLong = 1 so the result saved as an array of longs.</p>

## Functions reading / writing setting/resetting bit

Function	Function (PHP)	Description / Purpose
<b>IPS7RdBit</b>	<b>ips7_rdbit</b>	read one Bin from PLC (I,O,F,DB )
<b>IPS7SetBit</b>	<b>ips7_setbit</b>	write one Bit to PLC (I,O,F,DB)
<b>IPS7ResetBit</b>	<b>ips7_resetbit</b>	reset one Bit to PLC (I,O,F,DB)

### Calling parameters:

#### Bit read/write functions have the same input parameters:

Nr	Data type	Data type (PHP)	Name	Function
1	32-Bit Value unsigned	long	Ref	Reference of connection generated with IPS7Open/IPS7OpenEx. Used to internally identify the connection.
2	32-Bit Value unsigned	long	Typ	Selecting the memory area of the PLC to be processed(DB, In, Out,Flags...): 'D' = 68 dec. Datablocks (DB) 'E' = 69, dec. In (Inputs) 'A' = 65 dec. Out (Outputs) 'M' = 77 dec. Flags (in german „Merker“)
3	32-Bit Value unsigned	32-Bit Value unsigned	DBNr	Number of data block, only used with type 'D'. Otherwise use value „0“
4	32-Bit Value unsigned	32-Bit Value unsigned	Ab	Number of the byte address e.g. M 10.0 pass 10.
5	32-Bit Value unsigned	long	Bit	Number of bit to read/write 0 .. 7. E.g. M 5.4 pass 4
6	32-Bit Adresse	mixed	Buffer	<b>This parameter is only for IPS7RdBit.</b> Address/pointer to buffer in your pc program. If Bit is 1 value is otherwise value is 0. e.g.: read M 6.5 BYTE W; IPS7RdBit (Ref, 'M', 0, 6,5, &W); <b>Note for PHP:</b> In PHP pass the reference of the varaibale: ips7_rdbit (Ref, ord ("M"), 0, 6,5, & \$W); The variable is automatically converted to "long".

**Function for optimized reading of various data areas with one single call (new since V 1.35)**

<i>Function</i>	<i>Function (PHP)</i>	<i>Description / Purpose</i>
<b><i>IPS7RdMulti</i></b>	<b><i>ips7_rdmulti</i></b>	Processes a mixed read job. The jobs will be filled in a list / array of records / structs of type IPS7_RQ_MULTI and passed to function. The function sorts and bundles the jobs, and processes those optimized. <b>NOTE!</b> Reading order can be different from the order of the passed list.
<b><i>IPS7RdMultiC alcPacketCnt</i></b>	<b><i>ips7_rdmulticalcpack etcnt</i></b>	The return value is the number of required communication packages that are needed for all „multi-read“ requests. If return value is 1 so all the required variables can be read in one protocol. Please note the return value: > = 0: the number of packets <0: An error occurred as below

**IPS7RdMulti** automatically performs a data transfer between PC and S7. The conversion is comparable to the "Data type-casting" in the language. The bit width must be equal to or greater than the data types in the PC than that in the PLC. The caller has to provide a sufficiently large target area. Which means: If for example Two 16-bit value (total 32 bits) are read and converted to double value (64 bit) in the PC. The size of the buffer in your PC program has to be minimum two double values (a total of 128 bits). The conversion looks like in the example:

Value in PLC	Value in PC
6	6.0
1	1.0

The choice of PLC and PC data types is in the request job. Encoded using constants as described in the following

## The structure of the request records / structs for IPS7RdMult

Name	Typ	Description / Purpose																														
<i>DataArea</i>	<i>32-Bit unsigned</i>	<p>Selecting the memory area of the PLC to be processed(DB, In, Out,Flags...):</p> <p>'D' = 68 dec. Datablocks (DB)  'E' = 69, dec. In (Inputs)  'A' = 65 dec. Out (Outputs)  'M' = 77 dec. Flags (in german „Merker“)  'T' = 84 dec. Timer (access only as double word, 32 bit)</p> <p>Timers of the PLC are saved with base of time and value in BCD – format. IP-S7-LINK automatically converts the format an 32 bit value in units of milliseconds. The resolution is 10 ms. Writing time values, the drivers will select the best time base in the PLC. Note: Depending of the size of the value an the selected time base, there will be resolution errors. Allowed values from 0 to 9990000 ms</p> <p>'Z' = 90 dec. Counter (german Zähler), supported ony word functions  In the PLC counters are also saved in BCD format. Values for counter are running frome 0 to 999. The driver converts automatically to and from decimal fromat.</p>																														
<i>DataType</i>	<i>32-Bit unsigned</i>	<p>Data type in PLC</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> <th>PLC-Data type / count of bits</th> </tr> </thead> <tbody> <tr> <td><i>IPS7_BIT</i></td> <td>0</td> <td>one Bit / Boolean</td> </tr> <tr> <td><i>IPS7_BYTE</i></td> <td>1</td> <td>Byte (8 Bit)</td> </tr> <tr> <td><i>IPS7_WORD</i></td> <td>2</td> <td>WORD 16 Bit unsigned integer</td> </tr> <tr> <td><i>IPS7_INT</i></td> <td>3</td> <td>INT 16 Bit signed</td> </tr> <tr> <td><i>IPS7_DWORD</i></td> <td>4</td> <td>DWORD 32 Bit unsigned integer</td> </tr> <tr> <td><i>IPS7_DINT</i></td> <td>5</td> <td>long 32 Bit signed integer</td> </tr> <tr> <td><i>IPS7_REAL</i></td> <td>6</td> <td>S7Real</td> </tr> <tr> <td><i>IPS7_TIMER</i></td> <td>7</td> <td>Timer in S7 PLC</td> </tr> <tr> <td><i>IPS7_COUNTER</i></td> <td>8</td> <td>counterr in S7 PLC</td> </tr> </tbody> </table>	Name	Value	PLC-Data type / count of bits	<i>IPS7_BIT</i>	0	one Bit / Boolean	<i>IPS7_BYTE</i>	1	Byte (8 Bit)	<i>IPS7_WORD</i>	2	WORD 16 Bit unsigned integer	<i>IPS7_INT</i>	3	INT 16 Bit signed	<i>IPS7_DWORD</i>	4	DWORD 32 Bit unsigned integer	<i>IPS7_DINT</i>	5	long 32 Bit signed integer	<i>IPS7_REAL</i>	6	S7Real	<i>IPS7_TIMER</i>	7	Timer in S7 PLC	<i>IPS7_COUNTER</i>	8	counterr in S7 PLC
Name	Value	PLC-Data type / count of bits																														
<i>IPS7_BIT</i>	0	one Bit / Boolean																														
<i>IPS7_BYTE</i>	1	Byte (8 Bit)																														
<i>IPS7_WORD</i>	2	WORD 16 Bit unsigned integer																														
<i>IPS7_INT</i>	3	INT 16 Bit signed																														
<i>IPS7_DWORD</i>	4	DWORD 32 Bit unsigned integer																														
<i>IPS7_DINT</i>	5	long 32 Bit signed integer																														
<i>IPS7_REAL</i>	6	S7Real																														
<i>IPS7_TIMER</i>	7	Timer in S7 PLC																														
<i>IPS7_COUNTER</i>	8	counterr in S7 PLC																														
<i>DBNr</i>	<i>32-Bit unsigned</i>	Data block number, this is only used for type 'D '. Otherwise there is the value "0"																														
<i>Cnt</i>	<i>32-Bit unsigned</i>	Number of data elements that are to be read / written																														
<i>Start</i>	<i>32-Bit unsigned</i>	Start byte in the PLC.																														
<i>StartBit</i>	<i>32-Bit unsigned</i>	The number of the first bit in the PLC, value (0 - 7), is used only for bit access, 0 otherwise																														

Name	Typ	Description / Purpose																								
PCDatyType	32-Bit unsigned	Data type im PC																								
		<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> <th>Data type in der PLC</th> </tr> </thead> <tbody> <tr> <td>PC_BYTE</td> <td>0</td> <td>Byte (8 Bit)</td> </tr> <tr> <td>PC_WORD16</td> <td>1</td> <td>16 Bit unsigned integer</td> </tr> <tr> <td>PC_INT16</td> <td>2</td> <td>16 Bit signed integer</td> </tr> <tr> <td>PC_WORD32</td> <td>3</td> <td>32 Bit unsigned integer</td> </tr> <tr> <td>PC_INT32</td> <td>4</td> <td>32 Bit signed integer</td> </tr> <tr> <td>PC_FLOAT</td> <td>5</td> <td>32 Bit float in PC</td> </tr> <tr> <td>PC_DOUBLE</td> <td>6</td> <td>64-Bit double in PC</td> </tr> </tbody> </table>	Name	Value	Data type in der PLC	PC_BYTE	0	Byte (8 Bit)	PC_WORD16	1	16 Bit unsigned integer	PC_INT16	2	16 Bit signed integer	PC_WORD32	3	32 Bit unsigned integer	PC_INT32	4	32 Bit signed integer	PC_FLOAT	5	32 Bit float in PC	PC_DOUBLE	6	64-Bit double in PC
		Name	Value	Data type in der PLC																						
		PC_BYTE	0	Byte (8 Bit)																						
		PC_WORD16	1	16 Bit unsigned integer																						
		PC_INT16	2	16 Bit signed integer																						
		PC_WORD32	3	32 Bit unsigned integer																						
		PC_INT32	4	32 Bit signed integer																						
PC_FLOAT	5	32 Bit float in PC																								
PC_DOUBLE	6	64-Bit double in PC																								
Result	32-Bit unsigned	Result after processing the job. The individual value can be found under the description: <b>Return value for Read/Write functions or Result in record IPS7RdMulti</b>																								
UserData_0	32-Bit unsigned	Free for the user to saveing additional information like pointers or handles																								
UserData_1	32-Bit unsigned	Free for the user to saveing additional information like pointers or handles.																								
Data	32-Bit Pointer	Pointer / address of the data buffer in the PC program.																								
pUserData	32-Bit Pointer	Free for the user to saveing additional information like pointers or handles.																								

### Example for IPS7RdMulti in C /C++

```
// small function initializing of one request
void InitRq (IPS7_RQ_MULTI *pRq, long DataArea, long DataType, long PcDataArea,
            long DBNr, long Start, long StartBit, long Cnt, void *pData)
{
    pRq->DataArea = DataArea;
    pRq->DataType = DataType;
    pRq->DBNr = DBNr;
    pRq->Cnt = Cnt;
    pRq->Start = Start;
    pRq->StartBit = StartBit;
    pRq->PcDataArea = PcDataArea;
    pRq->Data = pData;
}

void DemoRdMulti (int Ref)
{
    int EBits[64];
    BYTE EBytes[64];
    WORD MWords[32];
    WORD DB10Words [150];
    double DB10WordsAsDouble [150];
    float DB20RealAsFloat [60];
    LONG32 TimerAsInt [10];
    int Cnt = 10;
    int Res;
```

```

IPS7_RQ_MULTI Rq[10] ; // Max. 10 jobs;
memset (Rq, 0, sizeof (Rq));

Cnt = 0;

// read from E 4.0 32 Bit
InitRq (&Rq[Cnt++], 'E', IPS7_BIT, PC_BYTE, 0, 4, 0, 32, EBits);

// read from EBO 20 Bytes and save starting at [20]
InitRq (&Rq[Cnt++], 'E', IPS7_BYTE, PC_BYTE, 0, 0, 0, 20, &EBytes[20]);

//read from MB 20 10 words
InitRq (&Rq[Cnt++], 'M', IPS7_WORD, PC_WORD16, 0, 20, 0, 1, MWords);

//read DB10 starting at byte 0, 150 words
InitRq (&Rq[Cnt++], 'D', IPS7_WORD, PC_WORD16, 10, 0, 0, 150, DB10Words);

//read DB20 starting at byte 0, 150 words, save as doubles to PC
InitRq (&Rq[Cnt++], 'D', IPS7_WORD, PC_DOUBLE, 10, 0, 0, 150, DB10WordsAsDouble);

//read DB20 starting at byte 6, 60 Real-Values, save as float value in PC
InitRq (&Rq[Cnt++], 'D', IPS7_REAL, PC_FLOAT, 10, 6, 0, 60, DB20RealAsFloat);

//read starting at timer 5, 10 timers, save as 32 bit integer
InitRq (&Rq[Cnt++], 'T', IP,S7_TIMER, PC_WORD32, 0, 5, 0, 10, TimerAsInt);

Res = IPS7RdMulti(Ref, Rq, Cnt);

for (int i = 0; i < Cnt; i++)
{
    if (Rq[i].Result == 0) // okay
    {
        //.. process data here    }
    else
    {
        // process errors
    }
}
}

```

**Return value for Read/Write functions / result im Record IPS7RdMulti:**

The functions returns a signed 32-bit value as result:

Value	Description of error	reaction / Action
0	Everything okay	process data
2	Block or data area does not exist, for example access to DB, which is not available, or too small.	Verify that the desired data area exists in the PLC.
-1	Timeout, PLC not reachable	Start further read/write operations. The driver automatically repairs the connection. May you hav to increase the timout values (receive/connect timeout).
-2	The maximum number (256) of the possible connections is reached	Close unused connections

<b>Value</b>	<b>Description of error</b>	<b>reaction / Action</b>
-3	Can occur at Close. With the specified reference has not performed Open..	Check your source code whether the variable for the reference has not been overwritten. Or it was executed for this reference already a „ Close“
-5	general error	Check whether network is properly installed in the PC: TCP / IP enabled? Winsocket installed?
-6	Target CPU not found	Rack or number of slot invalid. No connection to these slot available. Check configuration in CP and PLC
-7	socket error	call IPS7GetSockErr and process error value
-8	memory error	requested memory in the PC is not available
-9	overrange	e.g.. timer > 9990000 ms
-10	Desired data type is not allowed or is not supported.	Check that the code for data type is valid
-11	The specified PC data type do not correspond to the specified PLC data area	e.g : This can occur if you want to access counter and the PC is specified as data type BYTE. Solution: Change the data type in the PC
-20	The specified memory in the PC is too small (eg array is too small), can only occur. net or PHP.	Enlarge data area in PC
-21	Only. net! For this instance of the class Open was already called.	call close.
-31	<b>Only with MultiRead:</b> PC and S7 data type are in invalid relation e.g. PC = BYTE = Word PLC	adjust PC -data area
-32	<b>Only with MultiRead:</b> S7 returns incorrect number of data for the specified data type	
-88	<b>Only with MultiRead:</b> The task was not yet processed.	
-99	invalid reference number	Did you call IPS7Open?
4660	demo time expired	order a license

### Additional functions for PHP

`long ips7_w2i(mixed Buffer, long Count);`  
converts unsigned 16-Bit - value to signed 16-Bit value

`long ips7_i2w(mixed Buffer, long Count);`  
converts signed 16-Bit - value to unsigned 16-Bit value

<i>Parameter</i>	<i>Description / Purpose</i>
<b>Buffer</b>	reference to array of long or the long value to be converted.
<b>Count</b>	count of values

### New in Version 1.17 !

**Diference S7RdPlcW <-> IPS7RdW and S7WrPlcW <-> S7WrW**

The PC and the PLC have different addressing modes. In S7, the memory is byte-oriented. To address from the perspective of the PLC programmer with the MB0 and MB1 MW 0, but with the MW1 MB1 and MB2. You see that MW0 and MW1 in MB1 overlap.

Prior to the release before V 1.17, it was only possible to access just start with the word addresses functions. From V 1.17 the functions S7RdPlcW and S7WrPlcW can access to odd start addresses. Would you now read MW 1, as the PLC programmer sees the call on you.

**IPS7RdPlcW (Ref, 'M', 0, 1, 1, WordBuffer);**

#### **!!! Note at word operation with S7RdW and S7WrW!**

Example of flags. This applies also for inputs and outputs data words

The word addressing in the PLC assigned each of the following bytes.

<b>word address</b>	<b>assigned bytes</b>
MW0	MB 0 and MB 1
MW1	MB 1 and MB 2
MW2	MB 2 and MB 3

You see, that there may be when using odd word addresses in double occupancy. Therefore, the word functions (IPS7RdW and IPS7WrW) only support access to even word addresses. This means that the start-word-no in the driver is always multiplied by 2. This method allows for the simple image of the PLC memory to the PC. So a word step in the PC are 16 bits in the PC and 16 bit in the PLC

Example:

`WORD Buf[64];`

Result of calling **IPS7RdW (Ref, Typ, DBNr, 0, 5, Buf):**

<b>PC</b>	<b>PLC</b>
Buf[0]	DW 0
Buf[1]	DW 2
Buf[2]	DW 4

You have to divide the start word number by two to make right access in the PC. This is also necessary for data blocks!! --> odd word addresses of PLC could not be read/write word by word.

If you like to access to odd addresses use **IPS7RdPlcW / IPS7WrW** functions.

**program samples:**

a) Reading words read DB 10, start at DW0, 10 words:

*Calling from C/C++:*

```
unsigned short int WordBuffer[512];
IPS7RdW (Ref, 'D', 10, 0, 10, WordBuffer);
```

*Calling from Delphi:*

```
WordBuffer array [0..511] of Word;
IPS7RdW (Ref, LongWord ('D'), 10, 0, 10, @WordBuffer[0]);
```

*Calling from Visual Basic:*

```
Dim WordBuffer (0..511) as Word;
IPS7RdW (Ref, 68, 10, 0, 10, WordBuffer(0))
result after calling:
```

<b>PC</b>	<b>=</b>	<b>PLC</b>
WordBuffer[0]	=	DB10.DBW0
WordBuffer[1]	=	DB10.DBW2
WordBuffer[2]	=	DW10.DBW4

b) reading starting at MB 0, 10 bytes

*Calling from C/C++:*

```
unsigned char ByteBuffer[512];
IPS7RdB (Ref, 'M', 0, 0, 10, ByteBuffer);
```

*Calling from Delphi:*

```
ByteBuffer array [0..511] of Byte;
IPS7RdB (Ref, 'M', 0, 0, 10, @ByteBuffer[0]);
```

Calling from Visual Basic:

Dim ByteBuffer (0 to 511) as Byte;

IPS7RdB (Ref, 77, 0, 0, 10, ByteBuffer(0));

result after calling:

PC		PLC
ByteBuffer[0]	=	MB 0
ByteBuffer[1]	=	MB 1

Function	Function (PHP)	Description / Purpose
<b>IPS7GetSockErr</b>	<b>ips7_getsockerr</b>	get the last socket error. For Windows check the list in the appendix. In Linux the numbers are declared in the systemfile errno.h

Calling parameters:

Nr	Data type	Data type (PHP)	Name	Function
1	32-Bit Value unsigned	long	Ref	Reference of connection generated with IPS7Open/IPS7OpenEx. Used to internally identify the connection.

Return value:

The return value is a signed 32-bit value:

Value	Description of error	Meaning / Reaktion
0	everything okay	no error
else	last socket error	For Windows see the list in the appendix. For Linux see system file errno.h.

The declarations of the functions can be found in the header files:

for C/C++	ips7lnk.h
for Delphi	ips7lnk.pas
for VB	ips7lnk.bas

For .net (C#, VB.net) include the assembly located in the folder „NetFiles“. Additional help for .net can be found in the help file ips7lnk.chm.

## Winsock error list

Return code/value	Description
WSA_INVALID_HANDLE 6	<p>Specified event object handle is invalid.</p> <p>An application attempts to use an event object, but the specified handle is not valid. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.</p>
WSA_NOT_ENOUGH_MEMORY 8	<p>Insufficient memory available.</p> <p>An application used a Windows Sockets function that directly maps to a Windows function. The Windows function is indicating a lack of required memory resources. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.</p>
WSA_INVALID_PARAMETER 87	<p>One or more parameters are invalid.</p> <p>An application used a Windows Sockets function which directly maps to a Windows function. The Windows function is indicating a problem with one or more parameters. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.</p>
WSA_OPERATION_ABORTED 995	<p>Overlapped operation aborted.</p> <p>An overlapped operation was canceled due to the closure of the socket, or the execution of the SIO_FLUSH command in <a href="#">WSAIoctl</a>. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.</p>
WSA_IO_INCOMPLETE 996	<p>Overlapped I/O event object not in signaled state.</p> <p>The application has tried to determine the status of an overlapped operation which is not yet completed. Applications that use <a href="#">WSAGetOverlappedResult</a> (with the <i>fWait</i> flag set to FALSE) in a polling mode to determine when an overlapped operation has completed, get this error code until the operation is complete. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.</p>

Return code/value	Description
WSA_IO_PENDING 997	<p>Overlapped operations will complete later.</p> <p>The application has initiated an overlapped operation that cannot be completed immediately. A completion indication will be given later when the operation has been completed. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.</p>
WSAEINTR 10004	<p>Interrupted function call.</p> <p>A blocking operation was interrupted by a call to <a href="#">WSACancelBlockingCall</a>.</p>
WSAEBADF 10009	<p>File handle is not valid.</p> <p>The file handle supplied is not valid.</p>
WSAEACCES 10013	<p>Permission denied.</p> <p>An attempt was made to access a socket in a way forbidden by its access permissions. An example is using a broadcast address for <a href="#">sendto</a> without broadcast permission being set using <a href="#">setsockopt</a>(SO_BROADCAST).</p> <p>Another possible reason for the WSAEACCES error is that when the <a href="#">bind</a> function is called (on Windows NT 4.0 with SP4 and later), another application, service, or kernel mode driver is bound to the same address with exclusive access. Such exclusive access is a new feature of Windows NT 4.0 with SP4 and later, and is implemented by using the <a href="#">SO_EXCLUSIVEADDRUSE</a> option.</p>
WSAEFAULT 10014	<p>Bad address.</p> <p>The system detected an invalid pointer address in attempting to use a pointer argument of a call. This error occurs if an application passes an invalid pointer value, or if the length of the buffer is too small. For instance, if the length of an argument, which is a <a href="#">sockaddr</a> structure, is smaller than the sizeof(sockaddr).</p>
WSAEINVAL 10022	<p>Invalid argument.</p> <p>Some invalid argument was supplied (for example, specifying an invalid level to the <a href="#">setsockopt</a> function). In some instances, it also</p>

Return code/value	Description
	<p>refers to the current state of the socket—for instance, calling <b>accept</b> on a socket that is not listening.</p>
<p>WSAEMFILE 10024</p>	<p>Too many open files.</p> <p>Too many open sockets. Each implementation may have a maximum number of socket handles available, either globally, per process, or per thread.</p>
<p>WSAEWOULDBLOCK 10035</p>	<p>Resource temporarily unavailable.</p> <p>This error is returned from operations on nonblocking sockets that cannot be completed immediately, for example <b>recv</b> when no data is queued to be read from the socket. It is a nonfatal error, and the operation should be retried later. It is normal for WSAEWOULDBLOCK to be reported as the result from calling <b>connect</b> on a nonblocking SOCK_STREAM socket, since some time must elapse for the connection to be established.</p>
<p>WSAEINPROGRESS 10036</p>	<p>Operation now in progress.</p> <p>A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per- task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the WSAEINPROGRESS error.</p>
<p>WSAEALREADY 10037</p>	<p>Operation already in progress.</p> <p>An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling <b>connect</b> a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request (<b>WSAAsyncGetXbyY</b>) that has already been canceled or completed.</p>
<p>WSAENOTSOCK 10038</p>	<p>Socket operation on nonsocket.</p> <p>An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for <b>select</b>, a member of an <b>fd_set</b> was not valid.</p>
<p>WSAEDESTADDRREQ 10039</p>	<p>Destination address required.</p>

Return code/value	Description
	<p>A required address was omitted from an operation on a socket. For example, this error is returned if <b>sendto</b> is called with the remote address of ADDR_ANY.</p>
WSAEMSGSIZE 10040	<p>Message too long.</p> <p>A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself.</p>
WSAEPROTOTYPE 10041	<p>Protocol wrong type for socket.</p> <p>A protocol was specified in the <b>socket</b> function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM.</p>
WSAENOPROTOOPT 10042	<p>Bad protocol option.</p> <p>An unknown, invalid or unsupported option or level was specified in a <b>getsockopt</b> or <b>setsockopt</b> call.</p>
WSAEPROTONOSUPPORT 10043	<p>Protocol not supported.</p> <p>The requested protocol has not been configured into the system, or no implementation for it exists. For example, a <b>socket</b> call requests a SOCK_DGRAM socket, but specifies a stream protocol.</p>
WSAESOCKTNOSUPPORT 10044	<p>Socket type not supported.</p> <p>The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a <b>socket</b> call, and the implementation does not support SOCK_RAW sockets at all.</p>
WSAEOPNOTSUPP 10045	<p>Operation not supported.</p> <p>The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation is trying to accept a connection on a datagram socket.</p>
WSAEPFNOSUPPORT	<p>Protocol family not supported.</p>

Return code/value	Description
10046	<p>The protocol family has not been configured into the system or no implementation for it exists. This message has a slightly different meaning from WSAEAFNOSUPPORT. However, it is interchangeable in most cases, and all Windows Sockets functions that return one of these messages also specify WSAEAFNOSUPPORT.</p>
WSAEAFNOSUPPORT 10047	<p>Address family not supported by protocol family.</p> <p>An address incompatible with the requested protocol was used. All sockets are created with an associated address family (that is, AF_INET for Internet Protocols) and a generic protocol type (that is, SOCK_STREAM). This error is returned if an incorrect protocol is explicitly requested in the <b>socket</b> call, or if an address of the wrong family is used for a socket, for example, in <b>sendto</b>.</p>
WSAEADDRINUSE 10048	<p>Address already in use.</p> <p>Typically, only one usage of each socket address (protocol/IP address/port) is permitted. This error occurs if an application attempts to <b>bind</b> a socket to an IP address/port that has already been used for an existing socket, or a socket that was not closed properly, or one that is still in the process of closing. For server applications that need to <b>bind</b> multiple sockets to the same port number, consider using <b>setsockopt</b> (SO_REUSEADDR). Client applications usually need not call <b>bind</b> at all—<b>connect</b> chooses an unused port automatically. When <b>bind</b> is called with a wildcard address (involving ADDR_ANY), a WSAEADDRINUSE error could be delayed until the specific address is committed. This could happen with a call to another function later, including <b>connect</b>, <b>listen</b>, <b>WSAConnect</b>, or <b>WSAJoinLeaf</b>.</p>
WSAEADDRNOTAVAIL 10049	<p>Cannot assign requested address.</p> <p>The requested address is not valid in its context. This normally results from an attempt to <b>bind</b> to an address that is not valid for the local computer. This can also result from <b>connect</b>, <b>sendto</b>, <b>WSAConnect</b>, <b>WSAJoinLeaf</b>, or <b>WSASendTo</b> when the remote address or port is not valid for a remote computer (for example, address or port 0).</p>
WSAENETDOWN 10050	<p>Network is down.</p> <p>A socket operation encountered a dead network. This could indicate a</p>

Return code/value	Description
	<p>serious failure of the network system (that is, the protocol stack that the Windows Sockets DLL runs over), the network interface, or the local network itself.</p>
<p>WSAENETUNREACH 10051</p>	<p>Network is unreachable.</p> <p>A socket operation was attempted to an unreachable network. This usually means the local software knows no route to reach the remote host.</p>
<p>WSAENETRESET 10052</p>	<p>Network dropped connection on reset.</p> <p>The connection has been broken due to keep-alive activity detecting a failure while the operation was in progress. It can also be returned by <a href="#">setsockopt</a> if an attempt is made to set <b>SO_KEEPALIVE</b> on a connection that has already failed.</p>
<p>WSAECONNABORTED 10053</p>	<p>Software caused connection abort.</p> <p>An established connection was aborted by the software in your host computer, possibly due to a data transmission time-out or protocol error.</p>
<p>WSAECONNRESET 10054</p>	<p>Connection reset by peer.</p> <p>An existing connection was forcibly closed by the remote host. This normally results if the peer application on the remote host is suddenly stopped, the host is rebooted, the host or remote network interface is disabled, or the remote host uses a hard close (see <a href="#">setsockopt</a> for more information on the SO_LINGER option on the remote socket). This error may also result if a connection was broken due to keep-alive activity detecting a failure while one or more operations are in progress. Operations that were in progress fail with WSAENETRESET. Subsequent operations fail with WSAECONNRESET.</p>
<p>WSAENOBUFS 10055</p>	<p>No buffer space available.</p> <p>An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.</p>
<p>WSAEISCONN 10056</p>	<p>Socket is already connected.</p> <p>A connect request was made on an already-connected socket. Some</p>

Return code/value	Description
	<p>implementations also return this error if <b>sendto</b> is called on a connected SOCK_DGRAM socket (for SOCK_STREAM sockets, the <i>to</i> parameter in <b>sendto</b> is ignored) although other implementations treat this as a legal occurrence.</p>
WSAENOTCONN 10057	<p>Socket is not connected.</p> <p>A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using <b>sendto</b>) no address was supplied. Any other type of operation might also return this error—for example, <b>setsockopt</b> setting <b>SO_KEEPALIVE</b> if the connection has been reset.</p>
WSAESHUTDOWN 10058	<p>Cannot send after socket shutdown.</p> <p>A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous <b>shutdown</b> call. By calling <b>shutdown</b> a partial close of a socket is requested, which is a signal that sending or receiving, or both have been discontinued.</p>
WSAETOOMANYREFS 10059	<p>Too many references.</p> <p>Too many references to some kernel object.</p>
WSAETIMEDOUT 10060	<p>Connection timed out.</p> <p>A connection attempt failed because the connected party did not properly respond after a period of time, or the established connection failed because the connected host has failed to respond.</p>
WSAECONNREFUSED 10061	<p>Connection refused.</p> <p>No connection could be made because the target computer actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host—that is, one with no server application running.</p>
WSAELOOP 10062	<p>Cannot translate name.</p> <p>Cannot translate a name.</p>
WSAENAMETOOLONG 10063	<p>Name too long.</p>

Return code/value	Description
	A name component or a name was too long.
WSAEHOSTDOWN 10064	Host is down.  A socket operation failed because the destination host is down. A socket operation encountered a dead host. Networking activity on the local host has not been initiated. These conditions are more likely to be indicated by the error WSAETIMEDOUT.
WSAEHOSTUNREACH 10065	No route to host.  A socket operation was attempted to an unreachable host. See WSAENETUNREACH.
WSAENOTEMPTY 10066	Directory not empty.  Cannot remove a directory that is not empty.
WSAEPROCLIM 10067	Too many processes.  A Windows Sockets implementation may have a limit on the number of applications that can use it simultaneously. <b>WSAStartup</b> may fail with this error if the limit has been reached.
WSAEUSERS 10068	User quota exceeded.  Ran out of user quota.
WSAEDQUOT 10069	Disk quota exceeded.  Ran out of disk quota.
WSAESTALE 10070	Stale file handle reference.  The file handle reference is no longer available.
WSAEREMOTE 10071	Item is remote.  The item is not available locally.
WSASYSNOTREADY 10091	Network subsystem is unavailable.  This error is returned by <b>WSAStartup</b> if the Windows Sockets implementation cannot function at this time because the underlying system it uses to provide network services is currently unavailable.

Return code/value	Description
	<p>Users should check:</p> <ul style="list-style-type: none"> <li>• That the appropriate Windows Sockets DLL file is in the current path.</li> <li>• That they are not trying to use more than one Windows Sockets implementation simultaneously. If there is more than one Winsock DLL on your system, be sure the first one in the path is appropriate for the network subsystem currently loaded.</li> <li>• The Windows Sockets implementation documentation to be sure all necessary components are currently installed and configured correctly.</li> </ul>
WSAVERNOTSUPPORTED 10092	<p>Winsock.dll version out of range.</p> <p>The current Windows Sockets implementation does not support the Windows Sockets specification version requested by the application. Check that no old Windows Sockets DLL files are being accessed.</p>
WSANOTINITIALISED 10093	<p>Successful WSASStartup not yet performed.</p> <p>Either the application has not called <b>WSASStartup</b> or <b>WSASStartup</b> failed. The application may be accessing a socket that the current active task does not own (that is, trying to share a socket between tasks), or <b>WSACleanup</b> has been called too many times.</p>
WSAEDISCON 10101	<p>Graceful shutdown in progress.</p> <p>Returned by <b>WSARecv</b> and <b>WSARecvFrom</b> to indicate that the remote party has initiated a graceful shutdown sequence.</p>
WSAENOMORE 10102	<p>No more results.</p> <p>No more results can be returned by the <b>WSALookupServiceNext</b> function.</p>
WSAECANCELLED 10103	<p>Call has been canceled.</p> <p>A call to the <b>WSALookupServiceEnd</b> function was made while this call was still processing. The call has been canceled.</p>
WSAEINVALIDPROCTABLE 10104	<p>Procedure call table is invalid.</p> <p>The service provider procedure call table is invalid. A service provider returned a bogus procedure table to Ws2_32.dll. This is usually caused</p>

Return code/value	Description
	by one or more of the function pointers being NULL.
WSAEINVALIDPROVIDER 10105	<p>Service provider is invalid.</p> <p>The requested service provider is invalid. This error is returned by the <a href="#">WSCGetProviderInfo</a> and <a href="#">WSCGetProviderInfo32</a> functions if the protocol entry specified could not be found. This error is also returned if the service provider returned a version number other than 2.0.</p>
WSAEPROVIDERFAILEDINIT 10106	<p>Service provider failed to initialize.</p> <p>The requested service provider could not be loaded or initialized. This error is returned if either a service provider's DLL could not be loaded (<a href="#">LoadLibrary</a> failed) or the provider's <a href="#">WSPStartup</a> or <a href="#">NSPStartup</a> function failed.</p>
WSASYSCALLFAILURE 10107	<p>System call failure.</p> <p>A system call that should never fail has failed. This is a generic error code, returned under various conditions.</p> <p>Returned when a system call that should never fail does fail. For example, if a call to <a href="#">WaitForMultipleEvents</a> fails or one of the registry functions fails trying to manipulate the protocol/namespace catalogs.</p> <p>Returned when a provider does not return SUCCESS and does not provide an extended error code. Can indicate a service provider implementation error.</p>
WSASERVICE_NOT_FOUND 10108	<p>Service not found.</p> <p>No such service is known. The service cannot be found in the specified name space.</p>
WSATYPE_NOT_FOUND 10109	<p>Class type not found.</p> <p>The specified class was not found.</p>
WSA_E_NO_MORE 10110	<p>No more results.</p> <p>No more results can be returned by the <a href="#">WSALookupServiceNext</a> function.</p>

Return code/value	Description
WSA_E_CANCELLED 10111	<p>Call was canceled.</p> <p>A call to the <a href="#">WSALookupServiceEnd</a> function was made while this call was still processing. The call has been canceled.</p>
WSAEREFUSED 10112	<p>Database query was refused.</p> <p>A database query failed because it was actively refused.</p>
WSAHOST_NOT_FOUND 11001	<p>Host not found.</p> <p>No such host is known. The name is not an official host name or alias, or it cannot be found in the database(s) being queried. This error may also be returned for protocol and service queries, and means that the specified name could not be found in the relevant database.</p>
WSATRY_AGAIN 11002	<p>Nonauthoritative host not found.</p> <p>This is usually a temporary error during host name resolution and means that the local server did not receive a response from an authoritative server. A retry at some time later may be successful.</p>
WSANO_RECOVERY 11003	<p>This is a nonrecoverable error.</p> <p>This indicates that some sort of nonrecoverable error occurred during a database lookup. This may be because the database files (for example, BSD-compatible HOSTS, SERVICES, or PROTOCOLS files) could not be found, or a DNS request was returned by the server with a severe error.</p>
WSANO_DATA 11004	<p>Valid name, no data record of requested type.</p> <p>The requested name is valid and was found in the database, but it does not have the correct associated data being resolved for. The usual example for this is a host name-to-address translation attempt (using <a href="#">gethostbyname</a> or <a href="#">WSAAsyncGetHostByName</a>) which uses the DNS (Domain Name Server). An MX record is returned but no A record—indicating the host itself exists, but is not directly reachable.</p>
WSA_QOS_RECEIVERS 11005	<p>QOS receivers.</p> <p>At least one QOS reserve has arrived.</p>
WSA_QOS_SENDERS	QOS senders.

Return code/value	Description
11006	At least one QOS send path has arrived.
WSA_QOS_NO_SENDERS 11007	No QOS senders.  There are no QOS senders.
WSA_QOS_NO_RECEIVERS 11008	QOS no receivers.  There are no QOS receivers.
WSA_QOS_REQUEST_CONFIRMED 11009	QOS request confirmed.  The QOS reserve request has been confirmed.
WSA_QOS_ADMISSION_FAILURE 11010	QOS admission error.  A QOS error occurred due to lack of resources.
WSA_QOS_POLICY_FAILURE 11011	QOS policy failure.  The QOS request was rejected because the policy system couldn't allocate the requested resource within the existing policy.
WSA_QOS_BAD_STYLE 11012	QOS bad style.  An unknown or conflicting QOS style was encountered.
WSA_QOS_BAD_OBJECT 11013	QOS bad object.  A problem was encountered with some part of the filterspec or the provider-specific buffer in general.
WSA_QOS_TRAFFIC_CTRL_ERROR 11014	QOS traffic control error.  An error with the underlying traffic control (TC) API as the generic QOS request was converted for local enforcement by the TC API. This could be due to an out of memory error or to an internal QOS provider error.
WSA_QOS_GENERIC_ERROR 11015	QOS generic error.  A general QOS error.
WSA_QOS_ESERVICETYPE 11016	QOS service type error.  An invalid or unrecognized service type was found in the QOS

Return code/value	Description
	flowspec.
WSA_QOS_EFLOWSPEC 11017	QOS flowspec error.  An invalid or inconsistent flowspec was found in the QOS structure.
WSA_QOS_EPROVSPECBUF 11018	Invalid QOS provider buffer.  An invalid QOS provider-specific buffer.
WSA_QOS_EFILTERSTYLE 11019	Invalid QOS filter style.  An invalid QOS filter style was used.
WSA_QOS_EFILTERTYPE 11020	Invalid QOS filter type.  An invalid QOS filter type was used.
WSA_QOS_EFILTERCOUNT 11021	Incorrect QOS filter count.  An incorrect number of QOS FILTERSPECS were specified in the FLOWDESCRIPTOR.
WSA_QOS_EOBJLENGTH 11022	Invalid QOS object length.  An object with an invalid ObjectLength field was specified in the QOS provider-specific buffer.
WSA_QOS_EFLOWCOUNT 11023	Incorrect QOS flow count.  An incorrect number of flow descriptors was specified in the QOS structure.
WSA_QOS_EUNKOWNPSOBJ 11024	Unrecognized QOS object.  An unrecognized object was found in the QOS provider-specific buffer.
WSA_QOS_EPOLICYOBJ 11025	Invalid QOS policy object.  An invalid policy object was found in the QOS provider-specific buffer.
WSA_QOS_EFLOWDESC 11026	Invalid QOS flow descriptor.  An invalid QOS flow descriptor was found in the flow descriptor list.
WSA_QOS_EPSFLOWSPEC	Invalid QOS provider-specific flowspec.

Return code/value	Description
11027	<p>An invalid or inconsistent flowspec was found in the QOS provider-specific buffer.</p>
WSA_QOS_EPSFILTERSPEC 11028	<p>Invalid QOS provider-specific filterspec.</p> <p>An invalid FILTERSPEC was found in the QOS provider-specific buffer.</p>
WSA_QOS_ESDMODEOBJ 11029	<p>Invalid QOS shape discard mode object.</p> <p>An invalid shape discard mode object was found in the QOS provider-specific buffer.</p>
WSA_QOS_ESHAPERATEOBJ 11030	<p>Invalid QOS shaping rate object.</p> <p>An invalid shaping rate object was found in the QOS provider-specific buffer.</p>
WSA_QOS_RESERVED_PETYPE 11031	<p>Reserved policy QOS element type.</p> <p>A reserved policy element was found in the QOS provider-specific buffer.</p>