



Protocol Manual

## **EtherNet/IP Slave (Adapter)**

Hilscher Gesellschaft für Systemautomation mbH  
Rheinstraße 15  
D-65795 Hattersheim  
Germany

Tel. +49 (0) 6190 / 9907 - 0

Fax. +49 (0) 6190 / 9907 - 50

Sales: +49 (0) 6190 / 9907 - 0

Hotline and Support: +49 (0) 6190 / 9907 - 99

Sales Email: [sales@hilscher.com](mailto:sales@hilscher.com)

Hotline and Support Email: [hotline@hilscher.com](mailto:hotline@hilscher.com)

Web: <http://www.hilscher.com>

---

Index	Date	Version	Chapter	Revision
1	08.02.05	2.000	all	Created

**Although this protocol implementation has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this program for any purpose not confirmed by us in writing.**

**Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this program or its documentation shall be limited to cases of intent.**

**We reserve the right to modify our products and their specifications at any time in as far as this contribute to technical progress. The version of the manual supplied with the program applies.**

---

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
<b>2</b>	<b>SPECIFICATION AND PROTOCOL FUNDAMENTALS.....</b>	<b>5</b>
<b>2.1</b>	<b>Object Modeling .....</b>	<b>5</b>
2.1.1	Object Description .....	6
2.1.1.1	Identity Object .....	6
2.1.1.2	Message Router .....	6
2.1.1.3	Assembly Object .....	7
2.1.1.4	Connection Manager .....	8
2.1.1.5	TCP/IP Interface Object .....	8
2.1.1.6	EtherNet Link Object .....	9
2.1.1.7	Application Specific Objects .....	10
<b>3</b>	<b>CONFIGURATION AND START-UP THE DEVICE .....</b>	<b>11</b>
<b>3.1</b>	<b>Start-Up with Database (Configuration).....</b>	<b>11</b>
3.1.1	Table PROJECT .....	11
3.1.2	Table DEV_PROFILE.....	12
3.1.3	Table IO_LENGTH.....	13
3.1.4	Table IP_SETUP.....	14
3.1.5	Table ETHERNET .....	18
<b>3.2</b>	<b>Download of the configuration .....</b>	<b>19</b>
3.2.1	Via Serial interface .....	19
3.2.2	Via Dual-port Memory .....	19
<b>4</b>	<b>ERROR CODES.....</b>	<b>20</b>
<b>4.1</b>	<b>Initialization Errors.....</b>	<b>20</b>
<b>4.2</b>	<b>Runtime Errors .....</b>	<b>21</b>
<b>4.3</b>	<b>System Errors.....</b>	<b>23</b>

## 1 Introduction

This description documents the implementation of the EtherNet/IP Slave (Adapter) on our devices. It must be noted, however, that the range of functions described may be limited depending on the device or utilization case. This is especially the case in the protocol conversions where a lesser functions range is usually used!

## **2 Specification and Protocol Fundamentals**

The implementation is based on EtherNet/IP Specification Release 1.0.

### **2.1 Object Modeling**

The DEVICE is modeled as a collection of objects. Object modeling organizes related data and procedures into one entity: the object. An object is a collection of related services and attributes. Services are procedures an object performs. Attributes are characteristics of objects represented by values or variables. Typically, attributes provide status information or govern the operation of an object. An object's behavior is an indication of how the object responds to particular events.

The following objects are present in the DEVICE and available from the link. For details refer to the EtherNet/IP specification.

## 2.1.1 Object Description

### 2.1.1.1 Identity Object

(Class Code 01h)

#### Description:

The Identity Object provides identification and general information about the device. The first instance identifies the whole device. It is used for electronic keying and by applications wishing to determine what nodes are on the network.

#### Supported Services:

Instance	Name	Attribute ID	Name	Supported Service
0	Class	1	Revision	Get Attribute All
		2	Max. Instance	
		3	Max. Class Attrib.	
		4	Max. Instance Attrib.	
1	Instance Attributes	1	Vendor ID	Get Attribute All
		2	Device Type	
		3	Product Code	
		4	Major Revision	
			Minor Revision	
		5	Status	
		6	Serial Number	
7	Product Name			
1				Reset

### 2.1.1.2 Message Router

(Class Code: 02h)

#### Description:

The Message Router Object provides a messaging connection point through which a client may address a service to any object class or instance residing in the physical device.

#### Supported Services:

There are no services supported by the Message Router Object.

### 2.1.1.3 Assembly Object

(Class Code 04h)

#### Description:

The Assembly Object binds attributes of multiple objects, which allows data to or from each object to be sent or received over a single connection. Assembly Objects can be used to bind input data or output data. The terms "input" and "output" are defined from the network's point of view. An output will produce data on the network and an input will consume data from the network.

#### Supported Services

Instance	Name	Attribute ID	Name	Supported Service
0	Class	1	Revision	Get Attribute Single
		2	Max. Instance	Get Attribute Single
101/1 (*)	In Data	3	Data	Get Attribute Single Set Attribute Single
100/2 (*)	Out Data	3	Data	Get Attribute Single Set Attribute Single

(\*) 101 and 100 are used if the EtherNet/IP slave is configured for ProductType 12, 1 and 2 for other product types

### 2.1.1.4 Connection Manager

(Class Code 06h)

#### Description:

The Connection Manager Object is used to manage the establishment and maintenance of communication connections.

#### Supported Services

There are no services supported by the Connection Manager Object.

### 2.1.1.5 TCP/IP Interface Object

(Class Code F5h)

#### Description:

The TCP/IP Interface Object provides the mechanism to configure a device's TCP/IP network interface. Examples of configurable items include the device's IP Address, Network Mask, and Gateway Address.

#### Supported Services:

Instance	Name	Attribute ID	Name	Supported Service
0	Class	1	Revision	Get Attribute All Get Attribute Single
		2	Max. Instance	Get Attribute All Get Attribute Single
		3	Number of Instance	Get Attribute All Get Attribute Single
1	Instance Attributes	1	Status	Get Attribute All Get Attribute Single
		2	Configuration Capability	Get Attribute All Get Attribute Single
		3	Configuration Control	Get Attribute All Get Attribute Single
		4	Path Size	Get Attribute All
			Path	Get Attribute Single
		5	IP Address	Get Attribute All Get Attribute Single
			Network Mask	
			Gateway Address	
			Name Server	
			Name Server 2	
6	Domain Name	Get Attribute All		
	Host Name			

### 2.1.1.6 EtherNet Link Object

(Class Code F6h)

#### Description:

The Ethernet Link Object maintains link-specific counters and status information for an Ethernet communications interface. A request to access instance 1 of the Ethernet Link Object refers to the instance associated with the communications interface over which the request was received.

#### Supported Services:

Instance	Name	Attribute ID	Name	Supported Service
0	Class	1	Revision	Get Attribute All Get Attribute Single
		2	Max. Instance	Get Attribute All Get Attribute Single
		3	Number of Instance	Get Attribute All Get Attribute Single
1	Instance Attributes	1	Interface Speed	Get Attribute All Get Attribute Single
		2	Interface Flags	Get Attribute All Get Attribute Single
		3	Physical Address	Get Attribute All Get Attribute Single
		4	Interface Counters	Get Attribute All Get Attribute Single Get And Clear Attribute
		5	Media Counters	Get Attribute All Get Attribute Single Get And Clear Attribute

### 2.1.1.7 Application Specific Objects

(Class Code 64h .. C7h)

#### Description:

The Application specific objects provide access to data which are not defined by the EtherNet/IP task. The definition of these data depends on the implementation of the User Application.

#### Supported Services:

Instance	Name	Attribute ID	Name	Supported Service
0	Class	1 .. 65535	User Specific	Get Attribute All Get Attribute Single Set Attribute All Set Attribute Single
1 .. 255	Instance Attribute	1 .. 65535	User Specific	Get Attribute All Get Attribute Single Set Attribute All Set Attribute Single

### 3 Configuration and Start-Up the Device

There are two strategies to start-up the device. The first and most common way is using a static database, which is described in this manual. The other way is to put the start-up parameter into the dual-port memory and then to perform a warm-start command, which is described in the protocol interface manual EtherNet/IP Slave (Adapter).

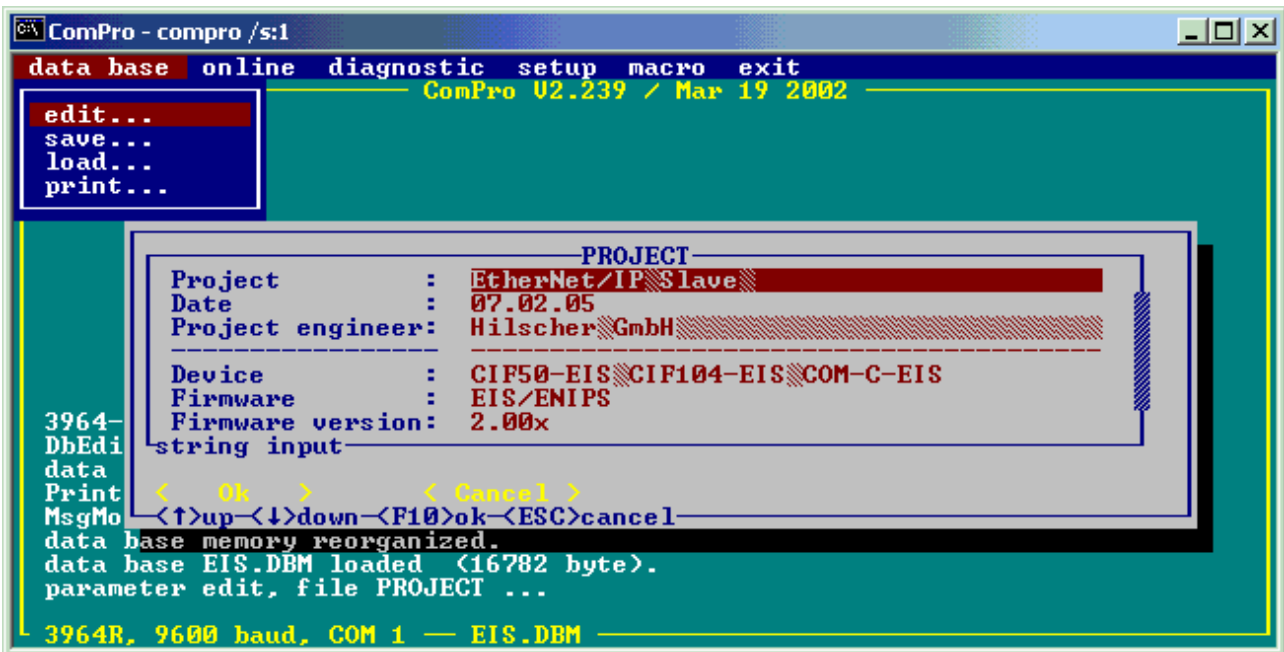
#### 3.1 Start-Up with Database (Configuration)

Normally the device will be configured by a static database. This database comes from the DOS-based configuration program **ComPro**. The database file consists of 4 relevant tables as described below:

At writing time there is no SyCon System Configurator available.

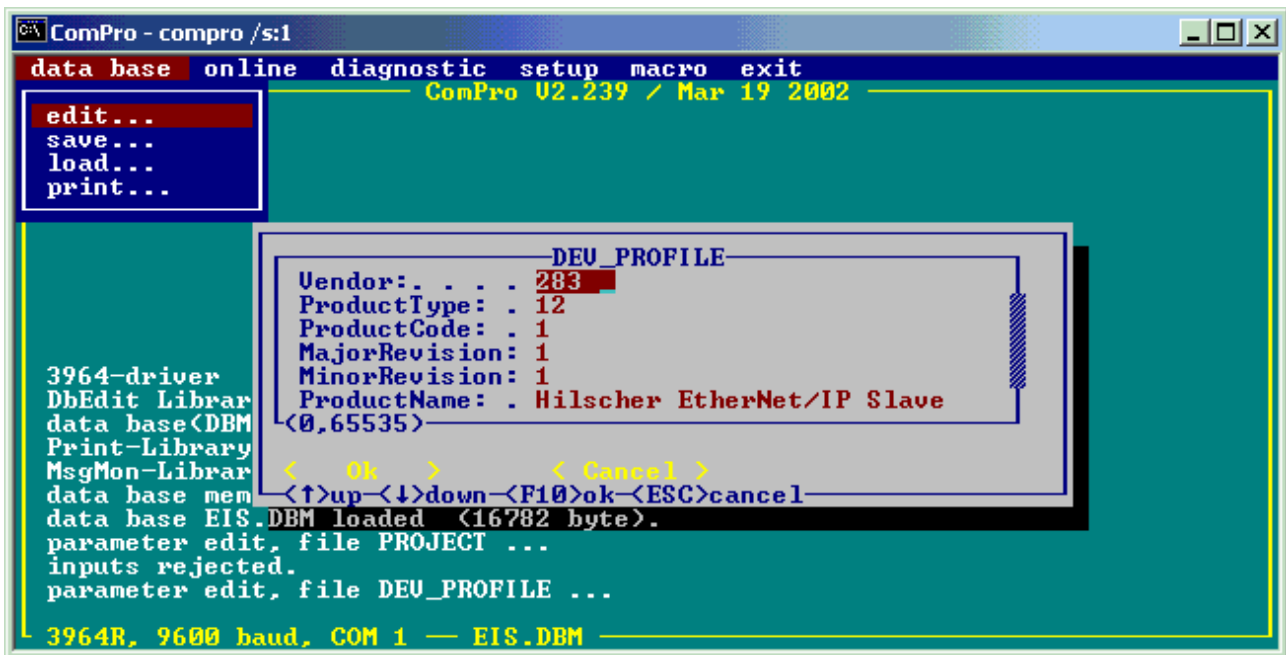
##### 3.1.1 Table PROJECT

In the table PROJECT the parameter **Project**, **Date** and **Project engineer** can be set by the user. This is for documentation purposes only. The parameters **Device**, **Firmware** and **Firmware version** are read only and can't be changed by the user.



### 3.1.2 Table DEV\_PROFILE

This table represents the contents of the device's ID object and its user specific entries, like VendorID, ProductType and ProductCode, etc.

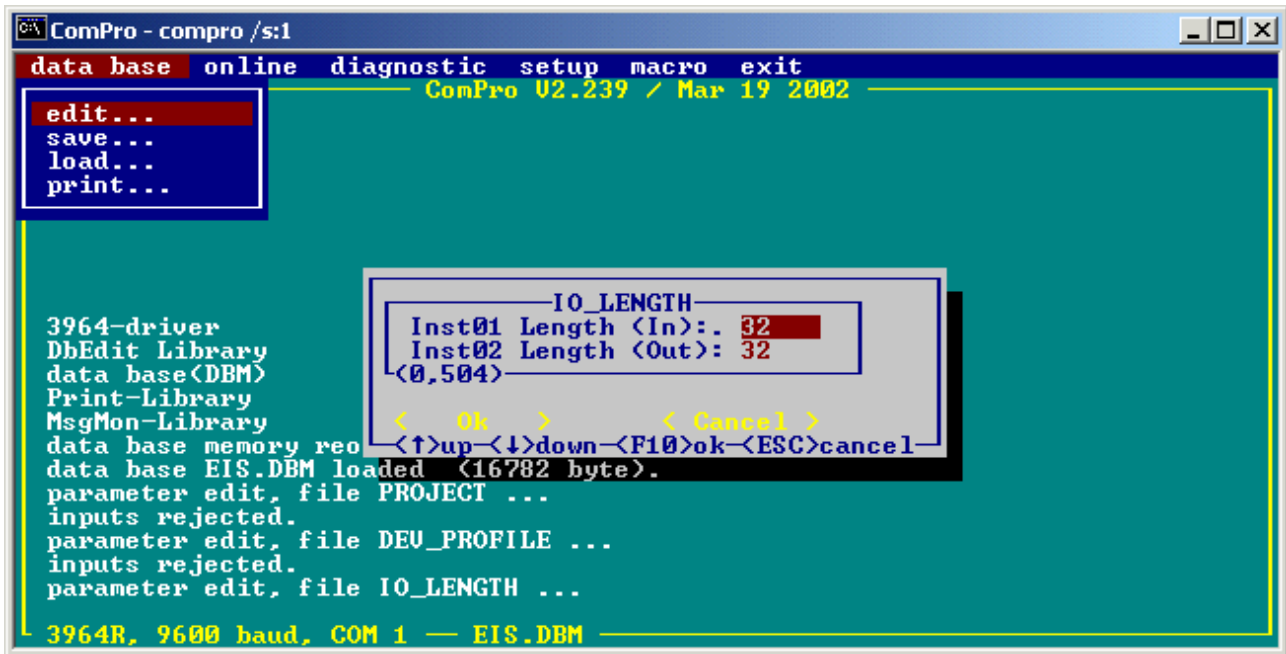


Parameter	Default	Description
Vendor	283	VendorID. If your company does not have a VendorID on its own, you can use the VendorID of Hilscher GmbH (283 decimal).
ProductType	12	Sets the value for product type, 12 is for 'Communications Adapter'
ProductCode	1	Sets the value for the product code
Major Revision	1	Sets the value for major revision
Minor Revision	1	Sets the value for minor revision
ProductName	Hilscher EtherNet/IP Slave	Sets the value for the product name

For the specific meaning of the other parameter refer to the EtherNet/IP specification.

### 3.1.3 Table IO\_LENGTH

This table contains the length of the cyclic IO data channel. The length in the table corresponds to the length for the device's Assembly instances. During the process to establishing a Class 1 connection the EtherNet/IP Slave (Adapter) device compares the values here with the values coming with the ForwardOpen request.



At the moment only the Assembly instance 01 and 02 are supported. The other instances are reserved for further extensions.

Parameter	Range	Default	Description
Inst01 Length (IN)	0 .. 504	32	Set the length of input data (Slave send data). The value is counted in bytes.
Inst02 Length (OUT)	0 .. 504	32	Set the length of output data (Slave receive data)

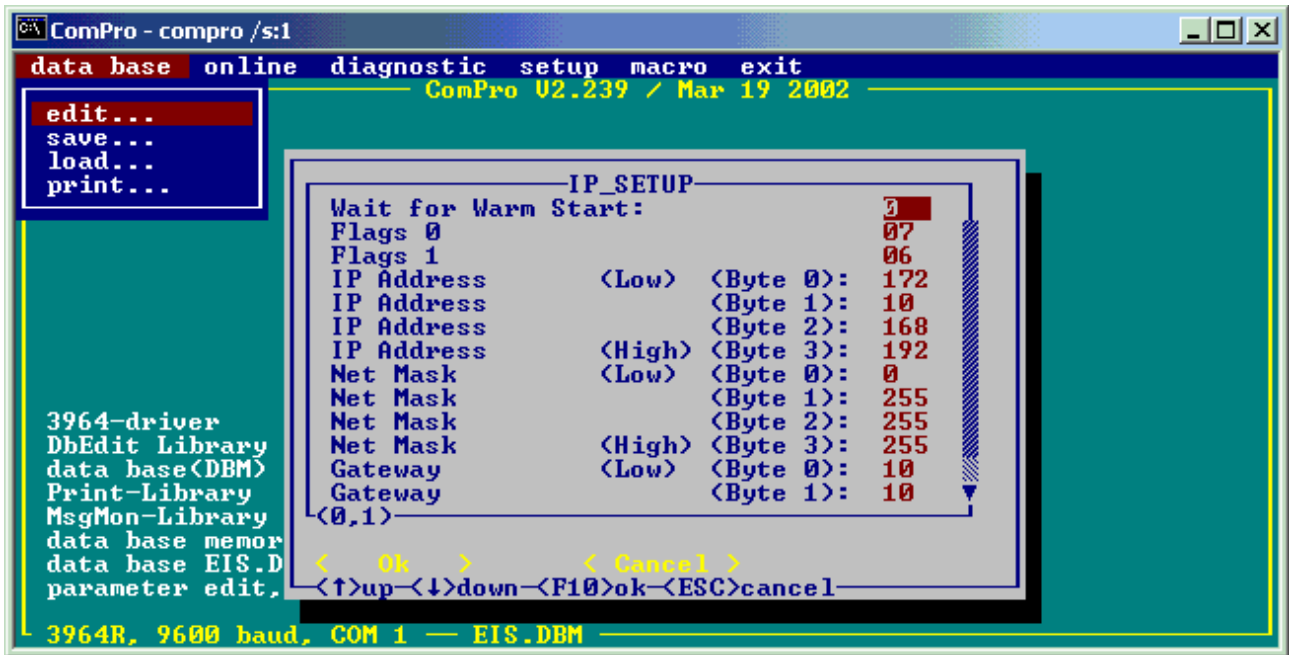
Input data and output data are seen from the viewpoint of the EtherNet/IP Master (Scanner):

- Input data are received by the Master (Scanner) and send from the Slave (Adapter)
- Output data are send by the Master (Scanner) and received from the Slave (Adapter)

**Note:** The value for Inst01 Length (In) is the number of bytes for cyclic IO data for the EtherNet/IP Slave device. The EtherNet/IP Master (Scanner) typically adds 4 bytes of input data for status information. The value for Inst01 at the EtherNet/IP Master (Scanner) configuration has to set to **Inst01 Length (In) plus 4**.

### 3.1.4 Table IP\_SETUP

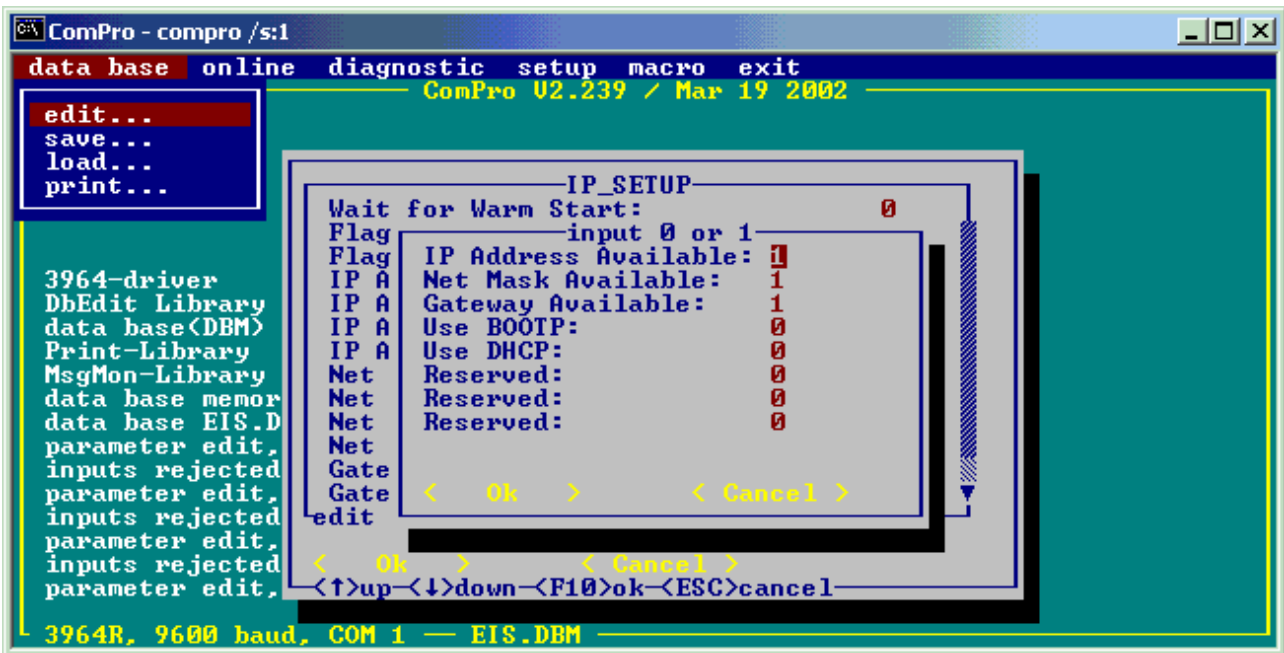
This table holds parameter to configure the TCP/IP interface, such as the local IP address, NetMask and the Gateway's IP address.



For the description see next page please.

Parameter	Range	Default	Description
Wait for Warm Start	0 .. 1	0	0 = The configuration file is used from the device 1 = The device will wait for the parameters from the application program transferred with a warm-start command
Flags 0	1 .. 31	07	Defines the way how the device gets the values for: IP address, net mask and gateway address. The default is the device gets these values from the settings described below. If BOOTP or DHCP is needed, then this can be defined via Flags 0. Details see below.
Flags 1	02, 06, 0A, 12, 1A	06	Enables (default) or disables the auto negotiation. If no auto negotiation is enables, then half or full duplex respectively 10 or 100 MBit/s can be set. Details see below.
IP Address	0 .. 255 0 .. 255 0 .. 255 0 .. 255	192 168 10 172	Sets the IP address of the device. The IP address A.B.C.D is assigned A to IP Address byte 3 B to IP Address byte 2 C to IP Address byte 1 D to IP Address byte 0 The figure above shows IP Address 192.168.10.172
NetMask	0 .. 255 0 .. 255 0 .. 255 0 .. 255	255 255 255 0	Sets the Net Mask of the device. The Net Mask A.B.C.D is assigned A to Net Mask byte 3 B to Net Mask byte 2 C to Net Mask byte 1 D to Net Mask byte 0 The figure above shows IP Address 255.255.255.0
Gateway	0 .. 255 0 .. 255 0 .. 255 0 .. 255	192 168 10 10	Sets the IP address of the device. The IP address A.B.C.D is assigned A to Gateway byte 3 B to Gateway byte 2 C to Gateway byte 1 D to Gateway byte 0 The figure above shows IP Address 192.168.10.10

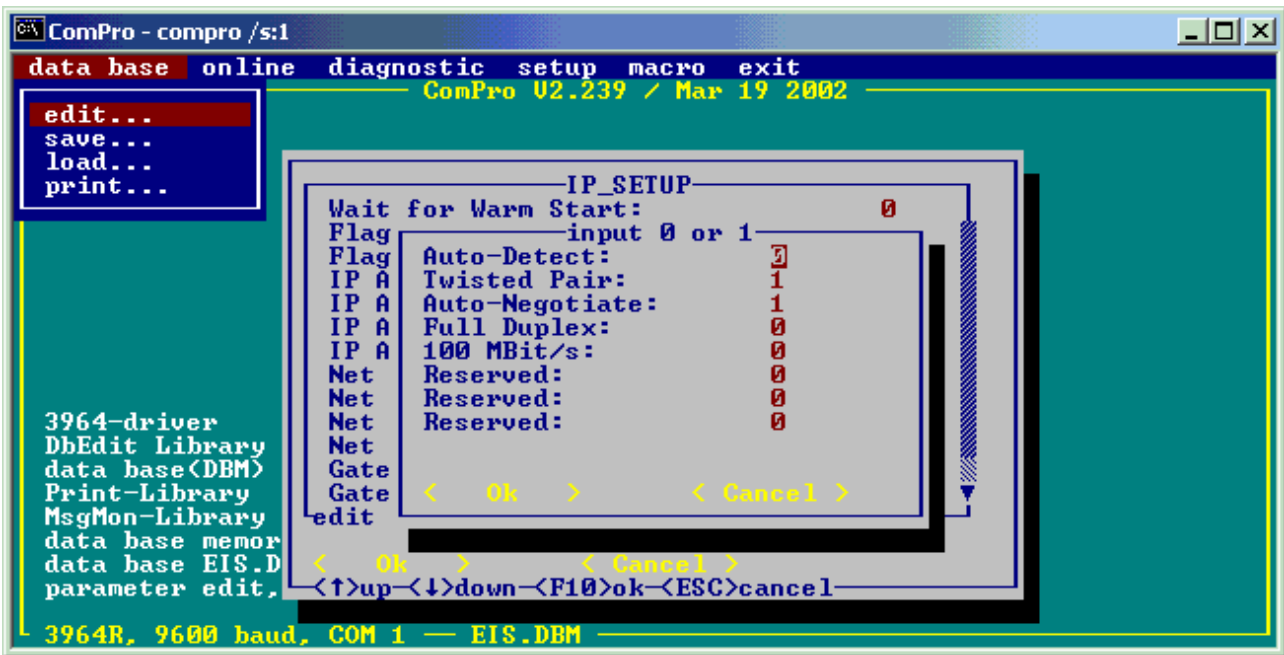
The following picture shows the settings for **Flag 0**.



Via the settings in **Flag 0** the user defines the way how the device gets the values for: **IP address**, **net mask** and **gateway address**.

Please note, that there exists a fallback procedure between the different configuration methods, if more than one is enabled in the **Flags 0**. If enabled, the device will first try to configure using **DHCP**. If **DHCP** configuration fails, the device will fallback to **BOOTP**, if this is enabled. In case of a **BOOTP** failure, the values found in the **IP address**, **Net Mask** and **Gateway** address will be used, if enabled in **Flags 0**. If none of the configuration mechanisms succeeds, the device will report an error.

The following picture shows the settings for **Flag 1**.

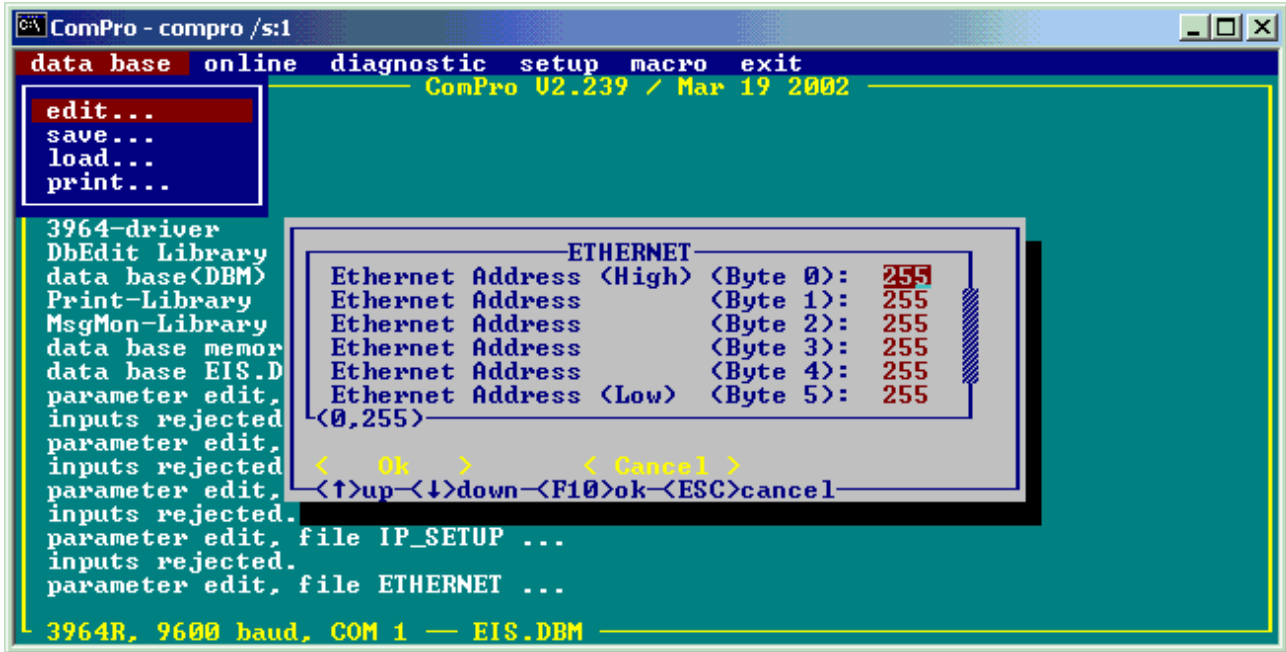


Enables (default) or disables the AutoNegotiation. If no AutoNegotiation is enables, then Half or Full Duplex respectively 10 or 100 MBit/s can be set.

Parameter	Range	Default	Description
Auto-Detect	0	0	Do not change. Use value 0 (Zero).
Twisted-Pair	1	1	Do not change. Use value 1 (One).
Auto-Negotiate	0, 1	1	0 = AutoNegotiation is disabled. Then the settings for 1 = AutoNegotiation is enabled.
Full Duplex	0, 1	0	Sets Full or Half duplex, if auto negotiation is disabled 0 = Half Duplex 1 = Full Duplex
100 MBit/s	0, 1	0	Sets the transmission speed, if auto negotiation is disabled 0 = 10 MBit/s 1 = 100 MBit/s

### 3.1.5 Table ETHERNET

The entries in this table can be used to overwrite the device's default Ethernet address (MAC address). If all bytes (Byte 0 ... Byte 6) are set to FF, then the default MAC address is used by the device (recommended).



For details refer to the Protocol Interface manual for TCP/IP, UDP/IP.

## 3.2 Download of the configuration

### 3.2.1 Via Serial interface

The configuration can be downloaded into the device via the serial diagnostic interface of the device.

ComPro communicates with the device via a serial connection. In this case a COM port of the PC must be connected with the diagnostic interface of the device via a diagnostic cable.

**Note:** The pin assignment of the diagnostic cable is described in the hardware documentation of the device manufacturer.

Start ComPro with

```
compro /s:1
```

respectively

```
compro /s:2
```

to use it on COM 1 or COM 2.

This communication is used if ComPro has to access the device via its serial diagnostic interface. The following two options are possible:

Option 1: ComPro runs on a remote PC (e.g. a notebook).

Option 2: ComPro runs on the same PC on which the EtherNet/IP slave is situated. Then the application can use the dual-port memory to access the device and the diagnostic interface can be used at the same time to communicate with the device (diagnostic data).

To download the configuration into the device select the menu **online > database > download** in ComPro and press the **Enter** key. A security question appears. Confirm this question by pressing the **Enter** key again to download the configuration.

### 3.2.2 Via Dual-port Memory

ComPro communicates with the device via the Dual-port memory of the device. This communication is utilized when ComPro is used on the same PC on which the device is installed.

**Note:** The CIF Device Driver has to be installed and it must have access to the device.

Start the **CIF Device Driver Test Program** with the menu **Start > Programs > CIF Device Driver > CIF Device Driver Test**.

Then use the menu **Board Select > Board 0** or **Board 1** or **Board 2** or **Board 3**.

Select **Driver Function > DevDownload > Configuration Download** and select the DBM file and press the button **Open**. This downloads the configuration file into the device into its FLASH memory.

## 4 Error Codes

### 4.1 Initialization Errors

Error Number	Description
0	<u>No Error / Success</u> No error detected.
50	<u>No Global Data</u> The global data area can not be obtained.
51	<u>No Hardware Info</u> The procedure to obtain the hardware information structure returns an error.
52	<u>No Token</u> No token can be obtained.
53	<u>Data Length Out of Range</u> The configured data lengths exceeds the max. allowed number (504 byte).
54	<u>Invalid Data Area</u> The data area index is invalid.
55	<u>Memory Allocation Failed</u> The allocation of memory failed (internal error).
56	<u>Improper State</u> The internal state machine receives an invalid command in the current state.
57	<u>EIP Not Started</u> The Ethernet/IP Firmware is not started
58	<u>OS Not Running</u> The internal operation system is not running
59	<u>Unknown Error</u> The detected error is none of the above defined codes.

## 4.2 Runtime Errors

Error Number	Description
0	<u>No Error / Success</u> No error detected.
110	<u>Watchdog Failure</u> A watchdog failure occurred.
111	<u>Unknown Command</u> The detected command is none of the defined.
112	<u>No Socket</u> There is currently no socket available.
113	<u>Bad Socket</u> The detected socket number is not used for the current connection.
114	<u>Socket Overflow</u> The number of required sockets exceeds the number of available sockets.
115	<u>Connection Overflow</u> The number of allowed connection was reached.
116	<u>Found No Connection Endpoint Description</u> The requested Connection Endpoint description could not be found.
117	<u>Bad Service</u> The detected service code is none of the defined.
118	<u>Unknown Mode</u> The detected mode is none of the defined.
119	<u>Abort Request</u> The Firmware detected an Abort.Request.
120	<u>Connection Error</u> The requested connection is not established yet.
121	<u>Connection Closed</u> The connection to a client was shut down by some reasons.
122	<u>No Data to Send</u> The task found an error while passing data to the IP task.
123	<u>Invalid Connection Type</u> The requested connection type is none of the defined.
124	<u>No Resources</u> Insufficient memory resources. The device runs out of memory.
125	<u>Bad Data</u> Poorly formed or incorrect data in the portion of the encapsulation message.
126	<u>Bad Length</u> The device received a message of invalid length.
127	<u>Bad State</u> The request is invalid at the current state.
128	<u>Unsupported Revision</u> The device detects an unsupported encapsulation revision.
129	<u>Bad Target ID</u> The detected target ID is invalid.

130	<u>Bad Session ID</u> The detected session ID is invalid.
131	<u>Invalid Tag Index</u>
132	<u>No Transport</u> No valid transport could be obtained for the current request.
133	<u>Invalid Transport ID</u> The detected transport ID is invalid.
134	<u>Bad Client Address</u> The detected client address is invalid.
135	<u>Unhanded Command</u> The device detects an invalid or unsupported encapsulation command.
136	<u>Register Session Failed</u> The Register Session request failed by some reason.
137	<u>Unregister Session Failed</u> The Unregister Session request failed by some reason.
138	<u>Bad Class/Instance</u> The detected Class/Instance is unknown.
139	<u>No Multicast Resources Available</u> The max. number of configurable Multicast addresses was reached.
140	<u>Multicast Address Allocation Failed</u> The allocation of a multicast address had no success.
141	<u>Join Multicast Group Failed</u> The attempt to join a multicast group failed.
142	<u>Not Registered</u> There is on connection established until now
143	<u>Set Socket Options Failed</u>
144	<u>Connection Timeout</u> The connection get a timeout
145	<u>No Such Entry</u> Connection Entry not found
146	<u>Wrong State</u> The execution of the command is not possible in the current state.

### 4.3 System Errors

Error Number	Description
0	<u>No Error / Success</u> No error detected.
105	Timeout occurred
202	No segment available
220	Pointer Error
221	Unexpected Error