

# VisiWinNET 2005

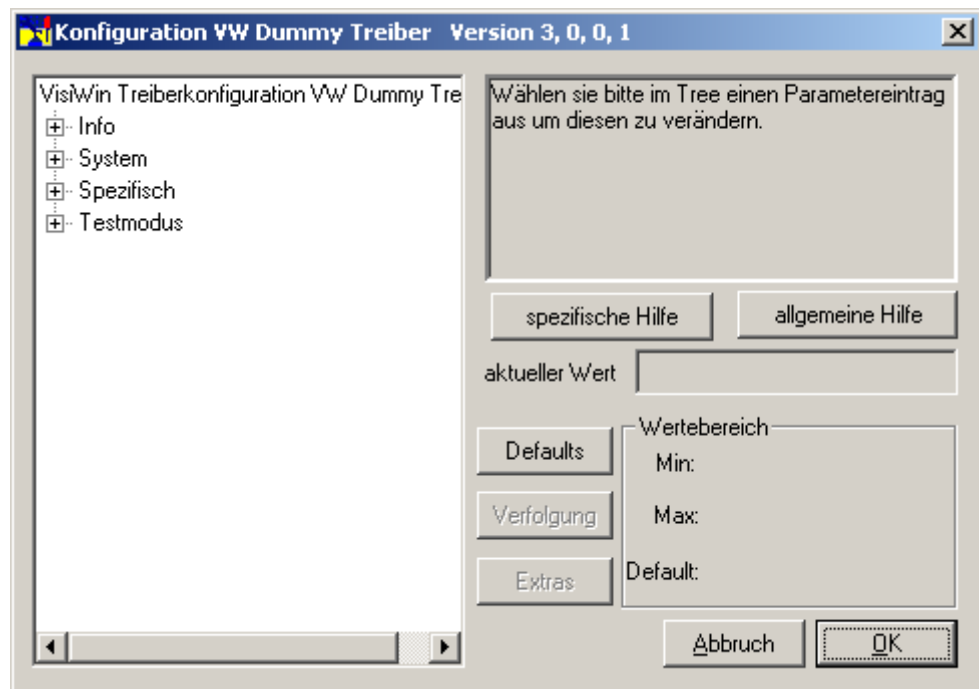
## VisiWin Drivers



- VisiWin
- VisiWinNET 2005
  - Common
  - Class Library
  - Systems
  - Tools
- Technical Informations
- Inosoft OPCServer
  - Basics and helping tools
  - Protocols

# VisiWinNET 2005

## VisiWin Drivers



Typical VisiWin driver configuration dialog







The contents of this manual must not otherwise be used without explicit written consent from INOSOFT GmbH.

We have checked the contents of this manual for compliance with the described software. Discrepancies can, however, not be ruled out. For this reason we cannot guarantee full compliance. The contents of the manual are subject to regular checking for necessary updates/amendments. Such amendments will be made in the subsequent edition.

Suggestions for improvement are welcome.

## Legend

In order to point out particular paragraphs the following symbols are used in the INOSOFT documentations:

	<b>Attention</b>	Passages with this sign should be read – and observed – with particular attention.
	<b>Hint</b>	Important paragraph “additional information”
	<b>Tip</b>	Many roads lead to Rome; here a shortcut is to be found.
	<b>In work</b>	Functions that are in preparation or already implemented but not yet prepared for documentation.
	<b>Example execute</b>	Instructions to be carried out in an example
	<b>Observe result</b>	Results to be observed with carrying out the exemplary instructions

© / ™ / ®

Windows®, Windows NT®, Windows 2000®, Windows XP® are registered trademarks of the Microsoft company.

Further product names marked ® are trademarks of the appropriate manufacturers.

INOSOFT GmbH created on

VisiWinNET Version: from 6.04.000

created on 08.06.2010

# Contents

<b>1 Preamble</b> .....	<b>1</b>
<b>2 VisiWin-Driver</b> .....	<b>2</b>
2.1 VisiWin drivers with the VisiWinNET variable kernel.....	2
2.1.1 Process variable .....	3
2.1.2 Structures.....	3
2.1.3 Scan structure.....	3
2.1.4 Image structure .....	4
<b>3 Setup</b> .....	<b>4</b>
3.1 Setup.....	4
<b>4 Project inclusion</b> .....	<b>6</b>
4.1 Project inclusion in VisiWinNET.....	6
<b>5 Configuration</b> .....	<b>8</b>
5.1 Configuration start .....	8
5.1.1 Start through the development environment.....	8
5.1.2 Direct configuration start .....	9
5.1.3 Start through the VisiWin-Driver.....	9
5.2 Configuration operating .....	10
5.2.1 Standard Index Card .....	10
5.2.2 Parameter Area Enhanced.....	11
5.2.3 Parameter Area Data Format.....	13
5.2.4 Parameter Area Specific.....	14
5.2.5 Parameter Area Traces .....	14
5.2.6 Parameter Area Data Monitoring .....	15
5.2.7 Freely editable Entries .....	16
<b>6 The VisiWin-Driver</b> .....	<b>17</b>
6.1 The VisiWin-Driver-Program .....	17
6.2 VisiWin-Driver start .....	17
6.2.1 Start through visualization .....	17
6.2.2 Immediate VisiWin-Driver start [Test mode] .....	17
6.3 VisiWin-Driver operation .....	19
6.3.1 Window surface .....	19
6.4 View and send data.....	21
6.4.1 Menu [Options].....	22
6.5 Data exchange with the application process database .....	24
6.6 The diagnosis dialog.....	26
6.7 Communication optimization .....	27

# 1 Preamble

## About this manual

This manual contains specific information on VisiWin drivers. It describes the properties that all VisiWin drivers possess. Beside this document an individual document exists for every driver that refers to the protocol-specific particularities.

## Questions and Problems

For technical questions and problems please contact your responsible INOSOFT agent or the INOSOFT GmbH Support under +49 (5221) 16 66 02 or email: [Support@INOSOFT.com](mailto:Support@INOSOFT.com)

Frequent questions and problems are dealt with on our homepage under [www.inosoft.com](http://www.inosoft.com)

There you will also find a support area for direct contact with our Main Office.

## 2 VisiWin-Driver

VisiWin drivers are communication components that control the data exchange between the variable kernel and a PLC. This documentation provides an overview of the operation of the VisiWin drivers. All information describes mainly the connection of the VisiWin drivers with VisiWinNET.

The document describes the characteristics that are identical with all VisiWin drivers. Parameters or particularities that depend on the individual protocol are described in the specific documentation for every individual VisiWin driver.

The contents of this document are to provide important assistance with the installation, the integration in a VisiWinNET project, the configuration of the VisiWin driver, and finally the optimization of the communication. In addition background information is provided to help understand the interaction between the VisiWin driver and the application at runtime.



In the past it became increasingly obvious that especially with the installation and implementation various problems arose. Often unexpected problems can occur, particularly at the interface between software and hardware. Therefore we want to point out here, that it is definitely useful to test the communication at the project start to avoid unnecessary trouble at the building site. Further please note that communication with a complete (large-scaled) database will not behave as a test database, which includes only few objects. [Also see optimization].

A diagram depicting the integration of the VisiWin drivers can be found in chapter Data Access/Introduction to the Data Access.

### 2.1 VisiWin drivers with the VisiWinNET variable kernel

In the development history of VisiWin the VisiWin drivers represented (prior to the development of the OPC standard) the optimum solution for a flexible and fast interface for the link with different communication partners (PLCs). Other visualization systems followed similar concepts, too. The big disadvantage with this was that the wheel had to be reinvented over and over again, meaning that all visualization systems, including of course VisiWin, had to invent their own communication components with great effort instead of utilizing readily usable ones from other manufacturers. Even in the future there will be specialized ways of communication where the VisiWin driver will be the optimum solution.

The development of the OPC standard has made it possible to link communication components from other manufacturers with one's own product. In order to utilize the advantages that this system offers it was necessary to integrate OPC as a central interface into VisiWinNET, too.

However, to also use the approximately 100 VisiWin drivers developed so far, and not to lose the opportunity to still use this optimized software interface in the future, there was a demand to combine these communication components with the OPC specification. This led to numerous difficulties as some functionalities that exist with the VisiWin drivers are not specified under OPC.

This is further explained below. At the same time solutions are introduced for optimum use of the VisiWinNET variable kernel with the VisiWin drivers.

### 2.1.1 Process variable

A process variable will be called Item. This assignment is not quite right as the Item includes further sectors after the OPC-Specification:

- Value (Value). Here, the actual core image of the PLC is located. Therefore this is comparable with the so far familiar VisiWin process variable.
- Time stamp (Timestamp). This part notifies the moment of the last process variable value update. The value is set to the moment of the process variable reading from the VisiWin Driver storage area, by the Kernel. This is different with the OPC-Server, which sets the time stamp to the moment directly after the process variable value was read from the PLC.
- Quality (Quality). Through this property the user will get the information if the process variable value, read from the PLC, is valid. If there is a read error with the VisiWin Driver for example, in the data area of the process variable location, quality is set to "BAD".
- Data type (Data type). Defines the data type in which the process variable value will be interpreted. I.e.:VT\_UI2 is an unsigned 16 Bit value (Range from 0 to 65535)

### 2.1.2 Structures

Structures largely reduce the complexity of protocolling between visualization and control. In the visualization several (or many) process values are merged by structuring definitions in a datablock that is exchanged with the control in one single read/write access. The breakup into individual elements (process values) that are required in the visualization is effected in the variable kernel.

VisiWinNET offers the following way: First a structure is to be defined. In this structure the elements are now defined along the lines of a VisiWin32 driver object. In the process elements can again be structures. Now an process variable is to be created that has the structure as a data type. In the VisiWinNET control elements access is now possible to the elements of the structure.

### 2.1.3 Scan structure

Scan structures offer the opportunity to control the exchange of individual data or datablocks programmatically or via the visualization surface when required.

To optimize the data flow of the communication to the PLC "Groups" can be defined in VisiWinNET. These groups contain any process variables. At runtime the groups can be activated or deactivated so that the process variables are only updated when the user wants them to be. If no groups are created all defined process variables are contained in a "Standard" group. Process variables are only read when they are currently required in the running application. The variable kernel has an automatism to detect this. Thus, groups are only active when an element of them is required in the application.

### 2.1.4 Image structure

Image structures are a special case in the scan structure. When a screen template is loaded the communication of individual process values depicted in the image is to be started.

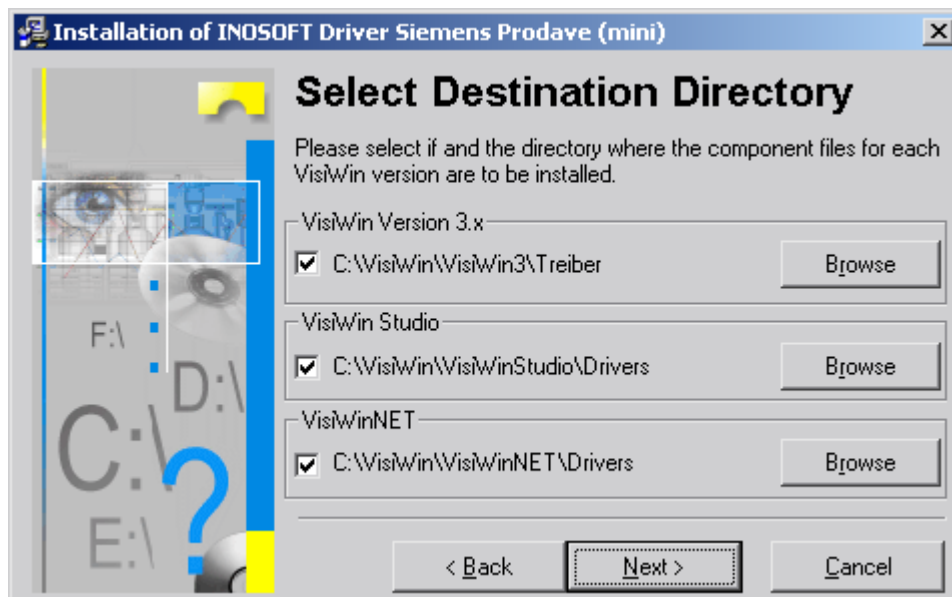
For this, too, OPC groups can be used. Groups are to be defined for this purpose that contain the process variables that are to be updated by the forms when addressed. The OPC group is then to be activated in the "Formload" event, and then deactivated in the "Formunload" event.

## 3 Setup

### 3.1 Setup

The set up is provided as SETUP.EXE. The program can be initialized easily through the Explorer or also through the "Windows-Start menu".

The set up tool will search for the different VisiWin versions. For every installed Release, the VisiWin-Driver path can be selected, if necessary.



With set up, following files will be copied from the SETUP.EXE to the target directory:

- |                               |  |
|-------------------------------|--|
| <b>&lt;Driver&gt;.exe</b>     | VisiWin-Driver program   |
| <b>&lt;Driver&gt;k.txt</b>    | Text file, in which the accessory VisiWin-Driver parameter sets are specified. The configuration module requires this file.                  |
| <b>&lt;Driver&gt;.dep</b>     | File, which defines the VisiWin-Driver interdependences. This is required for the VisiWinNET run time generation.                            |
| <b>&lt;Driver&gt;.chm</b>     | Help file, in which the VisiWin-Driver log-specific parameter sets are specified. Here, some VisiWin-Drivers still use a WORD document file. |
| <b>Possible further files</b> | I.e. DLL's... These files can own other target directories as well.  |

All VisiWin-Drivers are entered to the System Registry. With VisiWinNET, the entries can be found under following path:

[HKEY\_LOCAL\_MACHINE\SOFTWARE\INOSOFT GmbH\VisiWin\5.0\Drivers\<Driver>]



A VisiWin-Driver can be set up only, if a VisiWin-Product (VW32, VWStudio/VWNET) was installed on the system before.



If it happens to be, that there is no option to select the VisiWin-Driver from the VisiWinNET development environment, therefore is not available in the option list, you have to check first if the VisiWin-Driver was entered to the System Registry. If not, the set up must be processed again.

## 4 Project inclusion

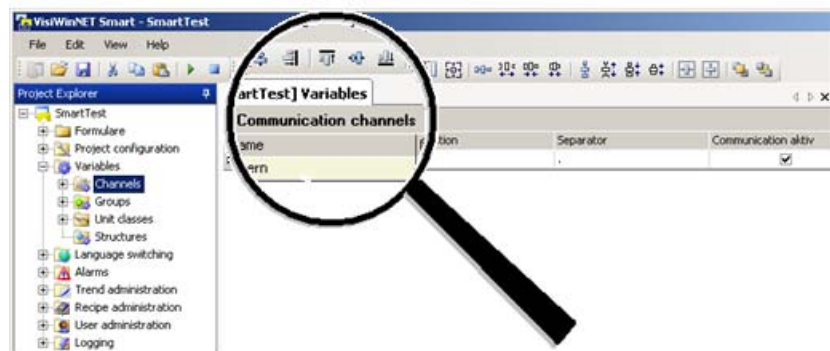
### 4.1 Project inclusion in VisiWinNET



The "Variables" node in the VisiWinNET project explorer is to be expanded. Through a click on the "Channels" node beneath the table editor of the variable editor is to be opened.




If the "Channels" node is highlighted in the Project Explorer all communication channels of the project are listed in the table editor.



In the table editor the context menu is to be opened (click with r.h. mouse button), and the "New" entry to be selected.

This opens the "Add channel" dialog. Here basic properties of the new channel are determined:

- Name: A freely chosen name for the channel. The name must be unequivocal within the communication channels of the project. References to process variables in the applications incorporate this name (example: "Ch1.w0" nominates the "w0" variable in the communication channel with the name "Ch1").
- Channel type: either OPC or Driver. This setting is to be set to "Driver".
- OPC Server / Driver: through the  button the dialog for the selection of a communication component is to be opened. Here the desired driver is to be selected.

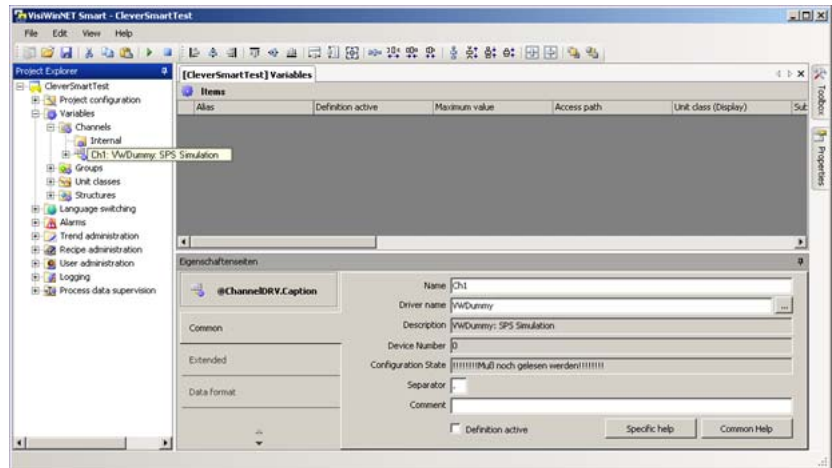
After closing the dialog click on the "Channels" node in the project explorer.



Highlighting the "Channels" node has the effect of the entered data being stored in the project database.

The new communication channel is now represented in the Project Explorer by a node.

If the node is highlighted the VisiWinNET properties page is automatically displayed. Here the full parameter set of the communication channel is provided for editing.



## 5 Configuration

### 5.1 Configuration start

There are three different ways to start the VisiWin-Driver configuration.

#### 5.1.1 Start through the development environment

This is the "normal" procedure to configure a VisiWin driver:

After a new communication channel has been added in VisiWinNET, and the communication component determined the configuration data are entered through the VisiWinNET properties page. For this, first the communication channel is to be marked in the project explorer.



The following index cards contain settings that belong to the basic volume of functions of the VisiWin drivers:

Index Card

Settings

#### **Enhanced**

- Communication parameterization (see Data Access manual)
- Information on the driver
- Main cycle and cycle interim
- Test mode

#### **Data Format**

Settings for special data type conversions

#### **Traces**

Monitoring of events in the driver and saving as a log file.

#### **Data Monitoring**

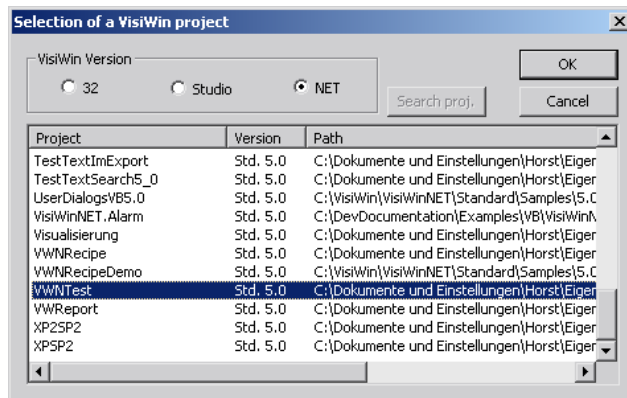
Determines variables whose values in the driver are to be monitored, and written to the log file.

The "Specific" index card contains the individual settings of a VisiWin driver. Further information on these settings can be found in the specific driver help. This is accessed through the "specific help" button on the "Standard" index card.

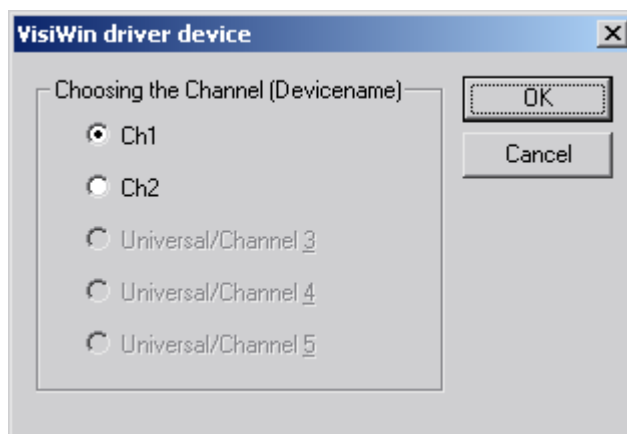
### 5.1.2 Direct configuration start

As a global configuration tool the "VisiWinNET.DrvConfig.exe" file is provided in the "Common" subdirectory of the VisiWinNET installation path.

After the start through the Windows Explorer or the "Windows Start Menu" the following dialog appears:



This dialog shows the selection of the VisiWin version and the VisiWinNET project.



This dialog only appears if more than one driver is used in the project as a communication component. This means that here the channel to be configured is to select.

### 5.1.3 Start through the VisiWin-Driver

Through menu selection [Options\Configuration] of the running VisiWin-Driver the configuration will be started (also see: VisiWin-Driver menu).

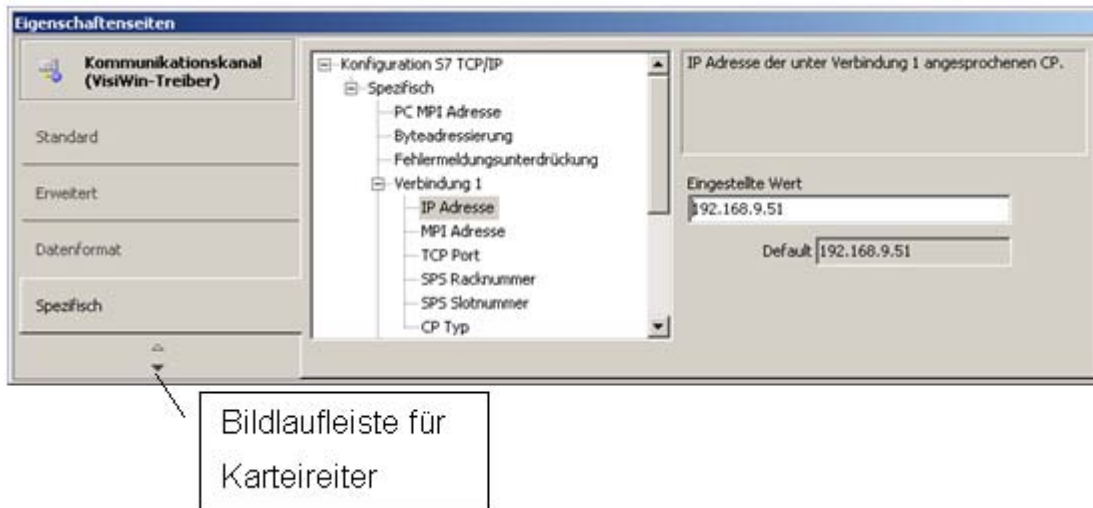


This menu can only be used if the VisiWin-Driver was previously configured from the development interface and the window type "visible" was selected (Parameter value 2). Besides, the configuration together with the accessory TXT file must be set up on the system (concerning the run time configuration).

If the VisiWin-Driver configuration was changed here, it must be initiated again, for new parameter acceptance.

## 5.2 Configuration operating

The surface of the configuration contains several index cards that can be activated through the tabs at the r.h. margin:



If not all tabs are visible because the dialog is too small navigation to the invisible index cards is effected through the scroll bar of the tabs.

### 5.2.1 Standard Index Card



Some of the settings listed here cannot be changed if the configurator is accessed outside the development environment.

The "Standard" index card contains configuration settings for the integration into VisiWinNET:

Setting	Description
<b>Name</b>	Specifies the name of the communication channel. When linking with control elements for example the name of the communication channel is put in front of the variable names. It has no significance for the driver function.
<b>Driver name</b>	Specifies the name of the used driver.
<b>Description</b>	Specifies a short text for the driver.
<b>Interface</b>	Specifies the universal interface of the driver in the project. The universal interface is automatically allocated when the driver is added to a project. It unequivocally identifies the driver instance if multiple communication channels are created in the project that use the same driver.

<b>Separator</b>	Specifies the separator character that separates the variable display in the editor into namespaces.
<b>Comment</b>	Specifies a freely selectable comment.
<b>Definition active</b>	Specifies whether at runtime the communication component (the driver program) is started.

In addition the index card contains two buttons:

<b>Common Help</b>	Opens this document. Here the properties that every VisiWin driver contains are described.
<b>Specific Help</b>	Opens the driver-specific document. The installation of a VisiWin driver always carries with it an online help (<DriverName>.CHM) in which the specific configuration settings, the address structure and special error messages are described.

## 5.2.2 Parameter Area Enhanced

Setting	Description						
<b>Allow synchronized communication</b>	<p>The "Allow synchronous communication" parameter determines whether the variable kernel uses synchronous access with a VisiWin driver. This is of particular importance if:</p> <ul style="list-style-type: none"> <li>the reading or writing access is set to "synchronous" in a group (<input type="checkbox"/> "Reading access/Writing access" parameter)</li> <li>the "Do not read start values synchronously" parameter of the communication component is set to "False".</li> </ul>						
	<table> <thead> <tr> <th>Setting</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>True</b></td> <td>Synchronous access is used.</td> </tr> <tr> <td><b>False</b></td> <td>Synchronous access is carried out as asynchronous access. This is the performance that VisiWinNET provided as standard prior to the 4.07 version.</td> </tr> </tbody> </table>	Setting	Description	<b>True</b>	Synchronous access is used.	<b>False</b>	Synchronous access is carried out as asynchronous access. This is the performance that VisiWinNET provided as standard prior to the 4.07 version.
Setting	Description						
<b>True</b>	Synchronous access is used.						
<b>False</b>	Synchronous access is carried out as asynchronous access. This is the performance that VisiWinNET provided as standard prior to the 4.07 version.						
<b>Don't read start values synchronously</b>	The "Do not read start values synchronously" parameter determines the initializing performance with the first data request. Normally the variable kernel reads with the first request of a process variable value that value synchronously from the communication interface. With this failed accesses and faulty data display in the application are avoided. In certain cases this can, however, lead to errors or unwanted delays.						
<b>Actual driver version</b>	This information can also be obtained by having the properties of the VisiWin driver displayed with the explorer through the r.h. mouse button. The format through the explorer is – contrary to the triple format displayed here (x.yy.zzz) – in quadruple format (x.y.a.z). The current value can only be displayed, and not changed.						

## Window style

The window style of the communication driver. There are five alternatives:

**0-Window not visible** The VisiWin driver has no visible window. It does neither appear as an icon in the task bar, nor is it assessable trough the task manager.

This setting makes sense for the final runtime on the target computer as the user should not be able to access the VisiWin driver through Windows to e.g. end it. Besides, the least computer performance is required with this setting.

**1-Window only as an icon** The window is only visible as an icon in the task bar.

**2-Window visible** The window of the VisiWin driver is visible. This does now allow to display the data areas that are exchanged with the PLC or change them through the VisiWin driver!

**3-Window with debug info** The window is visible, and data areas can be displayed, and changed through the VisiWin driver.

**4-Manual start of the VisiWin driver** The **VisiWin** driver is started. The communication must, however, be started separately through the menu [**File**→**Start**].



If the VisiWin driver has been started manually and not through WisiWin and if a window style smaller than 3 has been set this value is not adopted. Instead, the style is set to 3 to make sure that the drover is operable.



This setting should not be selected if the VisiWin driver is started through VisiWin as the data communication is not started automatically.

## Main cycle

This value is specified either is milliseconds or 10 milliseconds (see info field of the configurator).



For the first communication test the default value should be retained. The chapter "Optimization of the communication" contains further information on this parameter.

## time between cycles

This value should normally always be kept on the default value 1!



Only in exceptional cases is this parameter to be changed. The parameter must **not be set to 0!** The chapter "Optimization of the communication" contains further information on this parameter, too.

**Test address/ Test count**

VisiWin drivers contain a test mode. Here the driver is directly started without an application. Through the test address data can be exchanged with the control.

The address string is to be entered into the test address. This string must satisfy the conventions that are described in the specific driver documentation for the used protocol. A possible example for the AS511 (S5-PG) Siemens protocol could be

01:20:10

This stands for: COM1:data component 20:from word 10.

The test length describes the number of words that the VisiWin driver is to read from the PLC for the area selected under **Test Address**.



Even without an application? Still, the test mode only works if a VisiWin project has already been created on the workstation in which the VisiWin driver has been integrated and parameterized.

**5.2.3 Parameter Area Data Format**

Here it is defined whether data types are to be converted between VisiWin driver and variable kernel. Whether a conversion is to take place depends on the PLC type. If a data format conversion is set here this is not effected by the VisiWin driver but the variable kernel as this conversion depends on the data type of the process variable. The VisiWin driver only knows the data block but not its separation into the separate process variables.

**Floating point conversion (IEEE)**

This parameter is particularly valid for Siemens S5 PLCs. These use their own floating point format. If the floating point values are not displayed correctly this parameter is to be selected (value=1). Often this parameter must be set with floating point values together with the word swapping.

**Word swapping floats**

The high and low words of floating point values are to be swapped.

**Word swapping longs**

The high and low words of long word values are to be swapped.

**Texts**

The high and low bytes of process variables of the text data type are to be swapped.

**Motorola format conversion**

Many PLCs do not use the Intel but the Motorola data format. This means that the present data are swapped byte-wise as well as word-wise.

Byte-swapping is effected by the VisiWin driver itself. Where applicable this is to be set with some protocols under the "Specific" chapter.

## 5.2.4 Parameter Area Specific

The parameters displayed here are specific to VisiWin drivers, and not mentioned in this specification. Instead, they are described in the driver-specific documentation.

## 5.2.5 Parameter Area Traces

These parameters serve to configure the trace function. There are different trace levels that can be activated independently from each other. In addition, the way of handling the file in which the traces are recorded is configurable.

The name of the file is mostly identical with or similar to the EXE name of the VisiWin driver. The ending is always "LOG". Normally the trace file is in the applications directory. If this path cannot be found by the driver, for example if the VisiWin driver is started in test mode, the file is stored in the Root directory on drive C:\.

In the default setting all traces are deactivated.

<b>Safe logging</b>	Safe logging opens and closes the file for any access. This means a considerably higher strain on the system but has the advantage that the information in the file cannot be lost in the event of a crash. Normally the file can remain opened until the program is ended.
<b>Send trace texts to Spy</b>	If the texts are not to be saved in a file they can instead be sent to the "VisiSpy". "VisiSpy" is not included in the standard installation of a driver. Contact INOSOFT GmbH.
<b>Internal functions</b>	Internal function processes are logged. Attention: this means a very high data volume.
<b>Logging functions</b>	Function processes from the logging level of the VisiWin driver are logged. Attention: this means a very high data volume.
<b>Internal errors</b>	Internal function errors of the VisiWin driver are logged.
<b>Logging errors</b>	Logging errors from the communication level of the VisiWin driver are logged.
<b>Read commands</b>	Logs all orders that copy data from the PLC to the application. Attention: this can mean a very high data volume.
<b>Write commands</b>	Logs all orders that copy data from the application to the PLC.
<b>Windows messages</b>	Logs all Windows messages that serve the communication between VisiWin driver and variable kernel.
<b>Info</b>	Logs status and diagnosis information of the VisiWin driver. Currently not yet implemented.
<b>Specific messages</b>	Here trace output particularly programmed for customers can be activated.

<b>Writing data</b>	The first 32 data bits of writing orders are logged. The value is displayed hexadecimal in four digits without consideration of a data format.
<b>List infos</b>	Information on the write and read list that serves the communication between VisiWin driver and variable kernel. Attention: this can mean a very high data volume.
<b>File size</b>	Specifies the size of the files in Kbytes. The specification "0" means that the file can assume an infinite size. Once the specified volume has been reached the file is compressed. In the process, however, the last 25 percent of the content of the old file are always retained. (Attention: the system becomes instable when there is not enough space left on a drive.)
<b>log mode at 'infinite'</b>	If the file size has not been restricted through the previous parameter the file can be split by hours or days.  In the process the old file is saved in the same path under a file name reflecting the current date or time.
<b>Data tracing</b>	Logs self-defined files. Through the "Data tracing" index card addresses can be specified that are monitored for value changes.

## 5.2.6 Parameter Area Data Monitoring

<b>Data tracing</b>	Contains a list of data blocks to be recorded in the trace output.  For this the "Data tracing" option must be active on the "Traces" index card.  Through the "New" button a new line is generated to specify the address to be logged. A click on the new line allows editing its contents. The following information is expected as input:
<b>Address</b>	Driver-specific address specification
<b>Byte offset</b>	Offset in byte, identifying the start byte of the data area to be logged
<b>Byte count</b>	Length of the data area in byte

### 5.2.7 Freely editable Entries

Nearly all parameters are established with the development of a VisiWin driver. There are only rare exceptions that demand to add new parameters and their values to the firm parameters when the VisiWin driver is configured. The following dialog box provides the facility to do this. It can be opened through the [**Extras**] button in the configurator. This button is, however, only active with some VisiWin drivers.

With the access of the dialog all parameters not firmly programmed are entered into a list. By clicking on an entry in the **Entry** or **Value** columns that entry can be changed.

To set a list field to the edit mode first select the line in which a parameter is to be changed. Then click in the entry or value that is to be edited. There is a short timespan until the list field changes to the edit mode.



- OK** Adopts the entries and their values into the configuration of the VisiWin driver for the current project. Subsequently the dialog is ended.
- Abort** Ends the dialog without adopting parameters and values.
- New Entry** Adds a new line to the list. Entry and values must then be re-set through the selection of the list field.
- Delete** Deletes the selected line from the list.

## 6 The VisiWin-Driver

### 6.1 The VisiWin-Driver-Program

The VisiWin-Driver transmits data to the process database of the application, respectively reverse, through a therefore defined device. With it, the device can i.e. be a serial port as well as a PC plug-in card, connected with the PLC.

### 6.2 VisiWin-Driver start

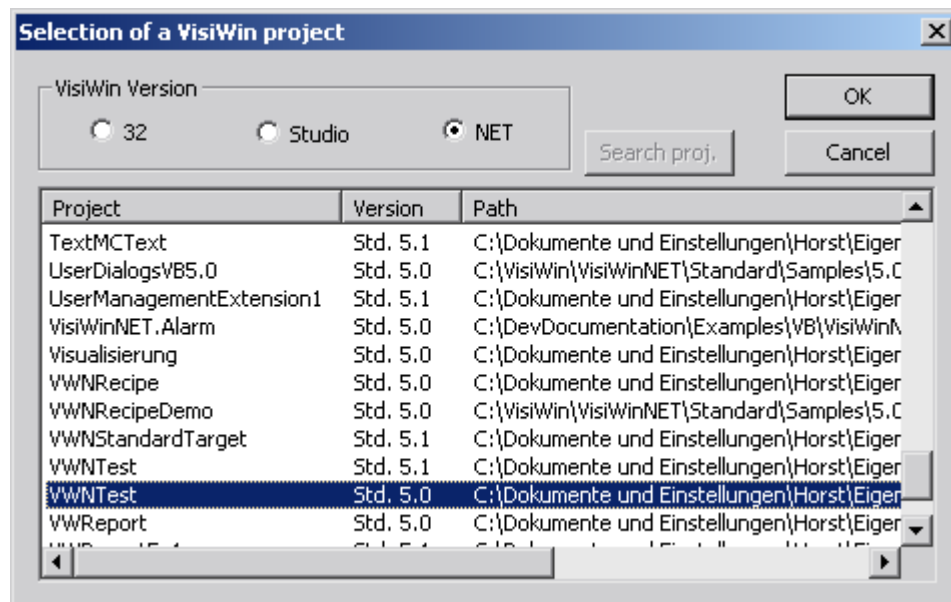
There are two different ways to start a VisiWin Driver.

#### 6.2.1 Start through visualization

If a VisiWin-Driver is included in a visualization application, it will be started together with the application start.

#### 6.2.2 Immediate VisiWin-Driver start [Test mode]

If a project is configured with a VisiWin-Driver once, the VisiWin-Driver can also be started directly, either through the Explorer or the Windows Start menu. The VisiWin-Driver does not access to the process database of the application but runs in test mode then and reads the test address cyclical. The VisiWin driver obtains its parameterization from the configuration that belongs to the selected application.



When starting the VisiWin driver manually the VisiWin version and subsequently the project must be selected.



If in the selected project the same driver is used by more than one communication channel the communication channel to be started must be selected.

**OK** Entries will be accepted. The VisiWin-Driver will be started.

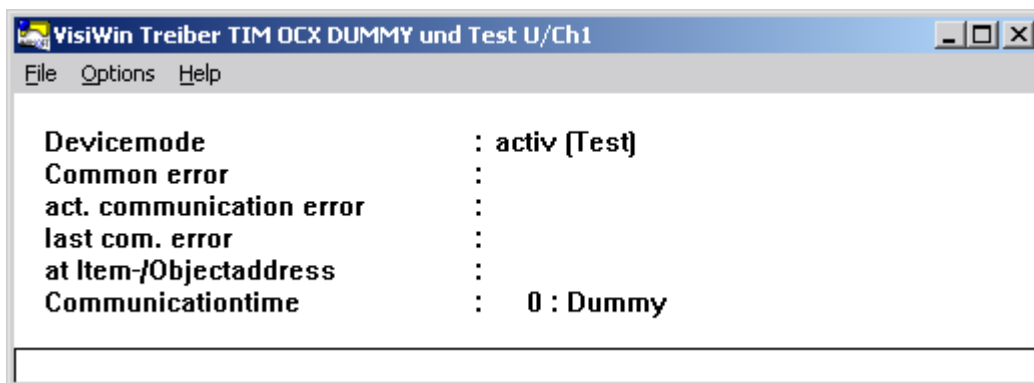
**Cancel** Dialog will be closed without start of the VisiWin-Driver.

## 6.3 VisiWin-Driver operation

### 6.3.1 Window surface

If the parameter window style in the VisiWin-Driver configuration was set to min. 2-windows visible, following window surface will be displayed after the start.

In the title bar the name of the VisiWin-Driver, as well as the utilized universal interface (Channel) will be presented. The output in the window is updated cyclically.



<b>Driver mode</b>	Active- the VisiWin-Driver runs; inactive- the VisiWin-Driver was started manually with parameter Window style 4 or else was not able to process the log initialization. In the last case an error message will be displayed. Behind the output a (Test) is displayed if the driver is running in test mode, i.e. with the test address.
<b>General error</b>	If errors occur, not caused by the log but through an incorrect internal run in the VisiWin-Driver program, a message including the respective error will be displayed here.
<b>Current communication error</b>	If a write or read order created a log dependent error, a respective error message will be displayed here. If the last communication did not produce an error anymore this text line will be deleted.
<b>Last communication error</b>	The last occurred communication error will be registered in this line together with time seal and the error causing structure address.
<b>With object address</b>	Here, the structure address under which the last communication error occurred will be represented separately again for further information.
<b>Current communication time</b>	The time in milliseconds that the last read or write command has taken is displayed here. What has to be taken into consideration is that Windows 95/98 systems work on 1 millisecond spans, Windows XP/Vista mostly on 10 milliseconds. Some drivers reduce this to one millisecond even with Windows XP/Vista to obtain optimum performance.

### 6.3.1.1 Menus

In the menu bar all VisiWin-Driver available instructions are subdivided in groups and assigned to various menus.

#### 6.3.1.2 Menu [File]

##### Start

If the VisiWin-Driver was parameterized with the window style 4-manual Start, the communication can be started here.

##### Stop

Stop communication.

##### Close

The communication program will be closed.



The menu levels under [File] should be used only if the VisiWin-Driver was started directly. If the VisiWin-Driver was started through an application, start, stop and close will be handled by the application.

#### 6.3.1.3 Menu [Options]

##### Configuration

If the configuration and the accessory parameter text file are available, the VisiWin-Driver configuration will be started. .



If parameter sets are changed here, they will be accepted only after the VisiWin-Driver restart. The application databank does not need to be retranslated.

If the configuration is not installed yet, here a dialog box will be displayed, in which the configuration parameter sets can be viewed. A parameter change cannot be processed here.

##### Diagnosis

Opens a dialog in which further information on communications is displayed. See diagnosis dialog.

##### Dump

If the VisiWin-Driver was started with the window style 3-Window with "Debug information", will be displayed through this menu, through which the read data areas can be viewed and changed.

##### View Log

Opens the diagnosis window of the driver. Here the loggings activated in the configuration under "Traces" are displayed.

#### 6.3.1.4 Menu [Help]

##### Specific Help

If the log dependent documentation is listed in the VisiWin-Driver set up directory, this document will be opened. If the documentation is not listed there, nothing will happen here.

##### General Help

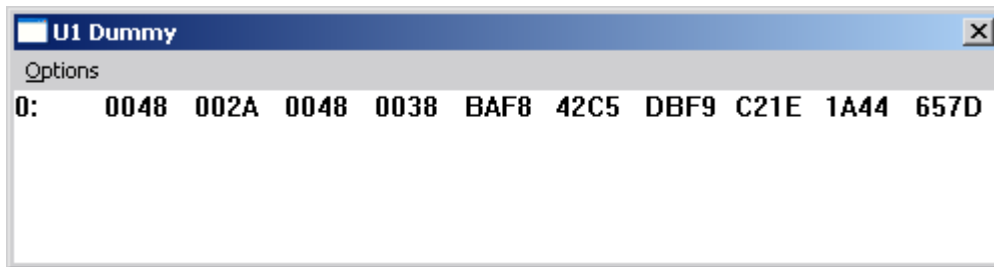
This help document will be opened.

##### About VisiWin-Driver

A dialog box with Release number, Copyright and further VisiWin-Driver information will be displayed.

## 6.4 View and send data

After menu call [Options\dump] a window will be opened, in which the net data of the first process database data range will be displayed as storage dump. If the VisiWin-Driver runs in test mode, the test address data range will be displayed.



Opening the window, the entire data range displayed, is assign to the value 0. Just when the displayed data range was read out through another communication cycle, the window content will be updated.

The title bar consists of the universal interface number (Channel), here 1, and the displayed structure address name (here DUMMY), ten (10) sign bit values are displayed per line. The first word offset net value of each line will be written separated through colon before the line.

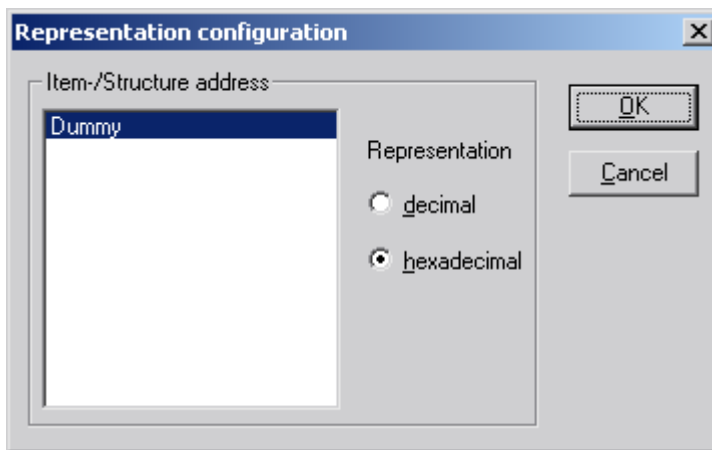


The window content structure does not apply to process variables, defined in a structure, but displays a core image only. A data conversion, set under the configuration subjects [System\Dateformat] is not yet considered here.

## 6.4.1 Menu [Options]

### 6.4.1.1 Menu [Options]

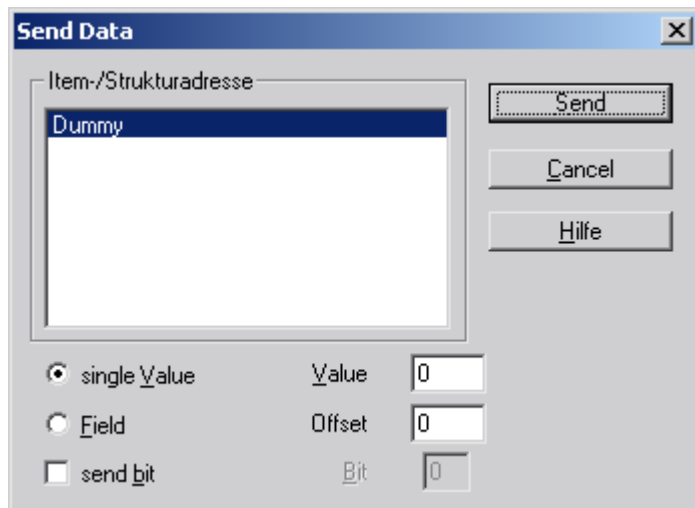
Opens a dialog box in which the representation of the data output window can be switched from decimal to hexadecimal. Furthermore, the structure address to be displayed can be selected here.



With close of the dialog box windows, the entire data range of the data output window displayed is assigned to the value 0. Just when the data range displayed, was read out through another communication cycle, the window content will be updated.

### 6.4.1.2 Menu [Send data]

The following dialog box enables direct value transfer to the PLC through the VisiWin-Driver.



<b>Variable/ Structure address</b>	Structure address selection to which the data is to be sent.
<b>Single value</b>	Either a word (16Bit) or a single Bit is to be sent.
<b>Field</b>	A field with several words is to be sent. The control field Value is renamed to Start value there, the control field Bit is renamed to number and the control field send Bit is renamed to count.
<b>Send Bit</b>	Dieses Feld (de)-aktiviert das Wert Kontrollfeld und (de)-aktiviert das Kontrollfeld Bit.
<b>Count</b>	With selection of Field the control field means, if all values to be written are to be described with the value from Start value or else if each word around the value 1 is to be incremented.
<b>Value</b>	Sets the value to be sent.
<b>Start value</b>	With selection of Field the first word value will be entered here.
<b>Offset</b>	Defines the word offset of the entry, selected under structure address, as of which the writing shall start.
<b>Bit</b>	With send Bit, the bit number to be set, is written here.
<b>Number</b>	With selection of Field, this parameter describes the number of words to be sent.
<b>Send</b>	Transfers the transmission statement to the VisiWin-Driver log level.
<b>Cancel</b>	Dialog box will be closed.
<b>Help</b>	Opens the help document.

### 6.4.1.3 Menu [End]

The data output window will be closed.

## 6.5 Data exchange with the application process database

This chapter describes the internal order of events between the application process database and the VisiWin-Driver log level.

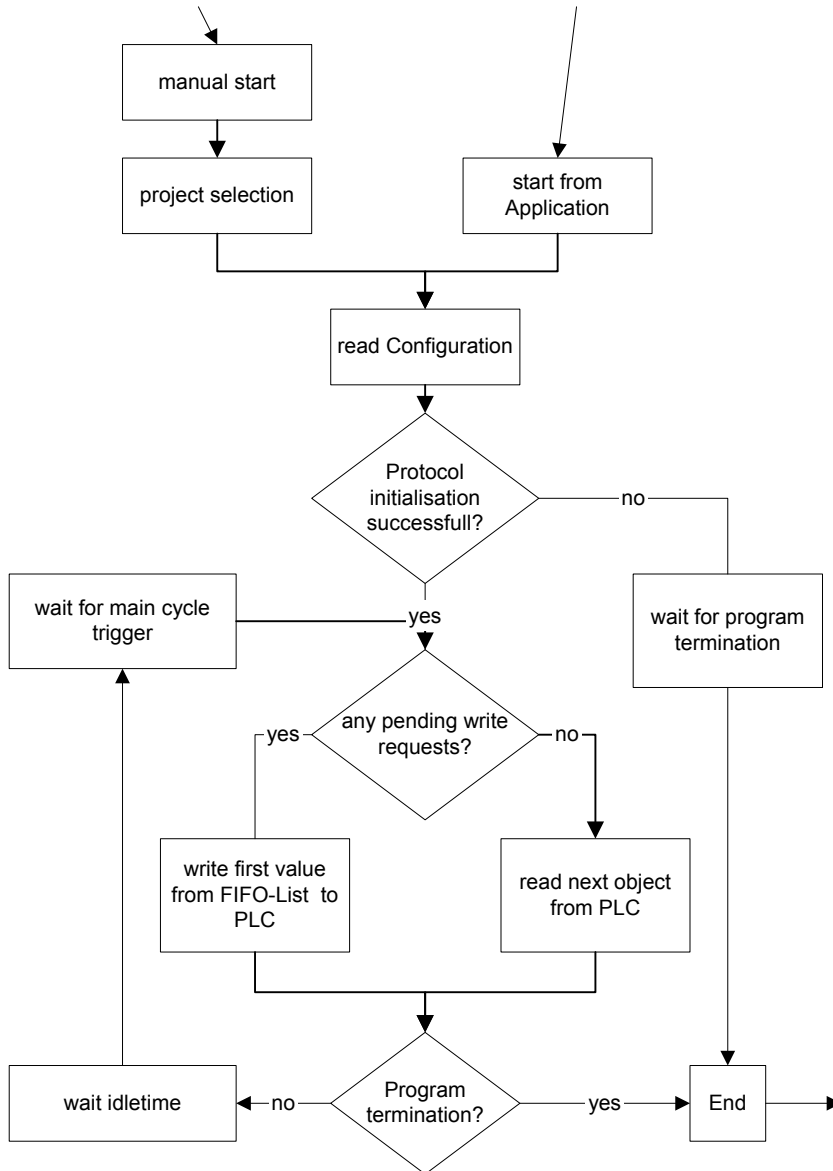
Between process database and VisiWin-Driver there are two different data areas (memory based files). The first file includes all structure objects, defined in a project database and created as process variable (respectively, process variables which contain a single data type only). The file has a fixed size, dependent on the number of objects. The size cannot be affected. The second area contains process variables, which values are to be written to the PLC. Normally this area is blank. Only if the process variable value changed on the visualization side, an object will be created in this file. This file has a standard size, which is dependent on the number of objects in the database. The smallest size offers room for 500 objects. It can possibly be necessary to enhance this number.



I.e. with a low-speed communication (serial with low baud rate) or if the cycle times of process database and VisiWin-Driver are coordinated badly and many single words are sent to the PLC (with recipes), it is possible, that the VisiWin-Driver will not be able to process the orders fast enough. This results in the queuing of write orders. In this case, a system error message will appear within the alarm control process variables, including theVisiWinDriver EXE name.

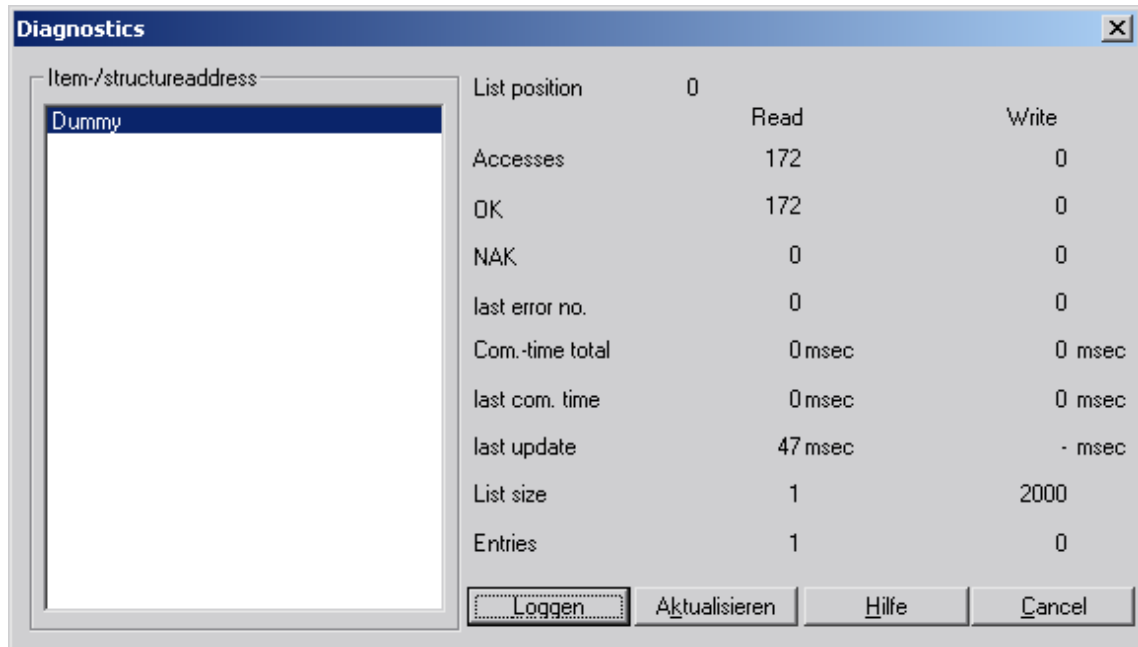
If this happens, the number of list entries is to be extended in the runtime configuration dialog. First though, you should try to optimize the communication. Further information is provided in the chapter "Optimization of the communication".

The VisiWin-Driver order of events can be displayed simplified with the following diagram:



## 6.6 The diagnosis dialog

The following dialog is opened via the Options/Diagnosis menu item.



In the Variable/Structure addresses collection all objects are listed that are currently known to the driver. By selecting an object via mouse or keyboard operation its data are displayed in the r.h. area of the dialog. As well as the read column where all displayed values relate to data read from the PLC there is a write column whose data display the write commands to the PLC.

<b>List position</b>	Indicates the position of the selected object in the appropriate list (internal information).
<b>Accesses*</b>	Indicates the number of all read and write commands since the start of the VisiWin driver.
<b>OK*</b>	Number of commands completed without communication error.
<b>NAK*</b>	Number of commands completed with communication error.
<b>Last error ID number</b>	ID number of last communication error.
<b>Communication time total*</b>	Addition of all errorless communication times. This is not 100 percent precise as the time resolution comes into it, corrupting the result.
<b>Last communication time</b>	Last communication time, in milliseconds, of an errorless command.
<b>Last update</b>	The last object update was x milliseconds ago. Recorded is the time the object was accessed or the Update button was activated.
<b>List size</b>	Size of the appropriate list (internal information).
<b>Entries</b>	Number of entries in the appropriate list.

\* After the VisiWin driver has been running for longer duration an overflow of the 32-bit value can occur.

<b>Logging</b>	If any trace option is activated the currently displayed values are recorded in the logfile.
<b>Update</b>	Updates the data for the selected object.
<b>Help</b>	Opens this document.
<b>Abort</b>	Closes the dialog box.

## 6.7 Communication optimization

The communication optimization depends on many factors and therefore cannot be solved through a "magic formula". The following list contains tips, which should be considered already with the creation of the process database. Some of these process variables will also affect the PLC-programming and therefore should be discussed with the PLC-programmer.

- It is always useful to minimize the data quantity, being read cyclic. For that purpose image and scan structures are to be used if possible. Groups serve to control, and thus minimize, the data flow.
- This results in single process variables being combined to structure elements by their utilization in the application. The structures should then be combined in groups from the update rate side to control the data flow to the PLC.
- With structures, which data must not necessarily be updated very fast, the scan cycle can be raised. If i.e. the scan cycle is set to 7, the structure will only be read from the VisiWin-Driver every 7\*time cycle of the runtime (in milliseconds). If this option is used the variable must be defined in a group whose update cycle can be manipulated.
- Data, which is written to the PLC only, should be combined in structures, which own the attribute write only.
- Structures should not include too many unused areas. Thus, data capacity can be decreased.
- In general it can be said that structure size is to be utilized completely if possible. Many small structures increase the administrative effort and worsen the relation between the transferred net data and the log-overhead. Here, it is always very difficult to make a compromise on the previous point (Structure size). If it is more useful now to create a structure with unused area and transfer blocks therefore, or else to divide the structure, depends on the log type as well. If the PLC reaction time is bad, or else the log itself is fast, it is advisable to put up with a gap within the transferred data area. Otherwise the data area should rather be divided.

- The Variables Kernel is quite fast in the process database processing, but still useful values should be set with its cycle. Normally cycle times between 200 and 500 milliseconds last out to reach refresh rates from 2 to 5 times per second. Under Windows XP/Vista the system utilization of the VW32 kernels or the VisiWinNET Manager can be monitored via the task manager (Ctrl-Alt-Del). The system should be utilized by a maximum rate of 20 percent. If the utilization rate is higher the cycle time of the VW32 kernel or the scan cycle rate for the groups are set too high. This in turn distinctly minimizes the VisiWin driver's performance.
- The VisiWin-Driver should always be set faster than the variables kernel (min factor 3). If there is no idea of the PLC communication rapidity, its window can be observed with the VisiWin-Driver operation. The communication time of the last processed structure will be output in the lowest line. If you look at the times for some time, you will realize that they are quite constant sometimes, but some other time they are subject to strong fluctuation. In principle it is wise to proceed from the lowest value. If it is 15 milliseconds i.e., the VisiWin-Driver main cycle is to be set on a value between 10 and 15 milliseconds. Smaller values will not speed up the VisiWin-Driver, but enlarge the system utilization!

Via the task manager (Ctrl-Alt-Del) the running processes can be monitored. The CPU utilization rate of the VisiWin driver should be below 30 percent (normally distinctly less, approx. 0-5 percent). Should this not be the case the Interim Cycle parameter should be slightly increased. This leaves more CPU time for other running applications. A value of more than 10 milliseconds should, however, not be set!