OPC Driver
User's Manual

Doc. No. ENP4045
Version: 29-08-2005

1. OPC Driver

1.1. Driver Use

The OPC driver is used for data exchange between **asix** system computers and any industrial PLC or SCADA program that has an access to a data server compatible with the OPC 1.0 or OPC 2.05 specification. (OPC specification is available at www.opcfoundation.org).

The OPC driver handles operations of read/write from/into a controller of:
1. simple variables containing scalar values like:
   a. Byte - 8-bit unsigned number;
   b. Word - signed or unsigned word (16-bit number);
   c. Double Word - signed or unsigned double word (32-bit number);
   d. Single Precision - 32-bit real number;
2. single dimension table variables consisting of simple variables (*see: 1*);
3. text variables.

1.2. Changes in OPC Driver - Version 2.0

- Adding the OPC server service compatible with the OPC 2.05 specification.
- Adding the possibility of specifying the access path for variables in the transmission channel.
- Adding the possibility of blocking the operation of writing to the OPC server.
- Driver extension with read/write operation of text / table variables.
- Elimination of large OPC server loading reducing considerably its capacity in case when a variable archiving period was too long in relation to their sampling period.

1.3. Declaration of Transmission Channel

The definition of transmission channel using the OPC driver is as follows:

<*Channel_name*> = UniDriver, OPC, <*Options*>

The basic and the only obligatory OPC driver option is the OPC server identifier given in the following form:

ProgId = <*OPC server identifier*>

EXAMPLE

An examplary declaration of the channel recalling to the OPC server registered uder the *Matrikon.OPC.Simulation* identifier:

Matrikon = UniDriver, OPC, ProgId=Matrikon.OPC.Simulation

All the driver options are described in the table below (see: ***Table 1***).

Table 1. OPC Driver Options Used in the Channel Declaration.

Option	Range and Value	Description	Default Value
ProgId	<tekst>	OPC server identifier under which the server is registered in the Windows system.	Lack
ReadOnly	Yes | No	If the option has the value *Yes*, then all write operations to the OPC server are blocked.	No
ItemsAlwaysActive	Yes | No	The option should have the value *Yes* (with the exception of case when variables are not archived in the **asix** system and not used in scripts and the number of variables read at the same moment should be minimize because of a limited transmission band).	Yes
OPCVersion	0 | 1 | 2	If the option have the value 1 or 2, then communication with the OPC server is realized always according to the OPC specification of the version – properly 1.0 or 2.05. If the option have the value 0, then the driver tries to use both versions (beginning with 2.05).	0

An examplary declaration of the channel using all options (it should be written in the application INI file in one line):

```
Matrikon = UniDriver, OPC, ProgId=Matrikon.OPC.Simulation, ReadOnly=yes,
    ItemsAlwaysActive=yes, OPCVersion=2
```

1.4. Defining the Process Variables

Defining the variables in the **asix** system is described in *asix4, User Manual*, in chapter *Asmen – Communication Manager* (*see: Declaration of Process Variables*). When defining the variable (recalling to an OPC server), the variable identifier from the OPC server variable base should be declared as the variable address. The syntax of the identifier depends on the specific OPC server and is described in its documentation. If the identifier contains small letters, comma or white space, then the identifier (the whole one) should be put in quotation marks.

1.5. Type Maching

While initializing the variable in the OPC server, the OPC driver sends a requested type of the variable to it. This type is defined according to the conversion function that has been assigned to the variable in the **asix** VariableBase. If the variable type requested by the OPC driver differs from the type declared in the OPC variable base, then the OPC server usually accepts the requested type and performs the proper conversion during the data transmission. If the OPC server is not able to perform the proper conversion, then an error is notified at the moment of variable initialization. This fact is also signaled in the list of system messages (in *'Control Panel'*) as the message "*Incorrect variable type*". The variable the initialization of which finished with an error will have a bad status.

If the error connected with the variable type misfit in the **asix** system and the OPC server occurs, then the other analogical conversion function (but operating with the variable type handled by the OPC server) should be used. The variable types used by conversion functions are described in the **asix** system documentation, in chapter *ASMEN – Communication Manager/Conversion Functions* (the type of variables requested by the OPC server is named *type of PLC variable*).

The variable type misfit most often occurs when the conversion functions that operate on unsigned numbers are used. It is because some OPC servers don't handle such numbers at all. In such case, it is impossible to use the following functions: TIME, CIRCLE1, COUNTER, MASK, FACTOR, FACTOR_DW, NEGBIT, NEGBIT_DW, NOTHING, NOTHING_BYTE, NOTHING_DD, NOTHING_DW, ON/OFF, SHIFT_L, SHIFT_R, SLIODER, SLIDER1, SLIDER1_FP.

There are three conversion functions handling any type of values being received from the controller: GRADIENT, AVERAGE, TABLE. For these functions, the OPC driver always requests from the OPC server sending the variable values in the form of floating-point numbers – because the floating-point number is always these functions' output type.

1.6. Manager of Logical Channels for the OPC Driver

Manager of logical channels enables easy editing the definition of transmission channels using the OPC server through simple setting options in the dialog window, specially prepared for the OPC driver. The text edition of transmission channels using other drivers than the OPC driver is also possible.

The manager is placed in the *ChannelsManager.exe* file in the directory of the **asix** system (by default c:\asix). More detailed information you can find in Manager of the logical channels for the OPC driver.

1.7. Communication Testing

The OPC driver enters on the message list of *'Control Panel'* information on important events like driver loading, driver connection to the OPC server, variable initialization performance. The information on possible errors at the driver initialization and operation stage is also entered on the list.

Detailed information on errors and diagnostic information is placed in the OPC log file. The driver log file is named *UniDriver,<current_date>.log* and is placed in the **asix** system directory by default. There are two groups of options defining the kind of information written to the log file. The first group options are placed in the **[UniDriver]** section:

☑ ***LogPath =record_track***

Meaning - the log file is placed in the defined directory.
Default value - by default, the log file will be created in the **asix** directory.
Parameter:
record_track - path to the directory in which the log file will be created.

ShowLogConsole = YES/NO

Meaning - if the option has the value YES, then the diagnostic window is displayed and all the information being written to the log appears in this window.

Default value - NO.

TracedNames = variable_list

Meaning - for each variable, which name is on the list, the information on its value, quality and stamp will be written to the log (during the variable processing).

Default value - lack.

Parameter:

variable_list - list of the variable names in the **asix** system, delimited with commas.

The second group of options is placed in the section with the same name as the channel name to which they apply (see: ***Table 2***).

Table 2. Options Placed in the Section with the Same Name as the Channel Name for the OPC Driver.

Option	Variable Range	Description
LogOnDataChangeStat	Yes | No	The information on the number of variables sent each time by the OPC server, during the refreshing and processing of the data sent by the driver, is placed in the log.
LogOnDataChangeItemInfo	Yes | No	The names of variables sent each time by the OPC server during the refreshing are placed in the log.
LogAddItemSucceeded	Yes | No	The information on successful end of the operation of writing to the OPC server is placed in the log.
LogWriteSucceeded	Yes | No	The information on successful end of the operation of writing to the OPC server is placed in the log.
TracedNames	< list of the variable names in the **asix** system, delimited with commas >	For each variable, which name is on the list, the information on its value, quality and stamp will be written to the log (during the variable processing).

All the options concerning the communication test may be changed during the **asix** system operation. After the modification and writing the application initialization file on the disc,

the new option values will be retrieved form the log by the OPC driver and will begin to have an effect on the range of information to be written to the log.

1.8. Channel Definition Updating for the OPC 1.0 Driver

The channel definition has the following form for the previous OPC driver version:

<*Channel_name*> = OPC, <*OPC server Progi*>

To convert the definition to the current obligatory form, one should change the text *"OPC,"* (with the comma) into *"UniDriver, OPC, ProgId =".*

EXAMPLE

Matrikon = OPC, Matrikon.OPC.Simulation

The current form:

Matrikon = UniDriver, OPC, ProgId = Matrikon.OPC.Simulation

2. List of Tables