OPC/ADO-Bridge

User's manual

OPC/ADO-Bridge User's manual

This manual is a product of iXTS Software GmbH, Germany

iXTS Software GmbH
Hohenadlstrasse 4
85737 Ismaning
Germany
Phone: +49 89 95 84 08 - 0
Fax: +49 89 95 84 08 -19
http://www.iXTS.de

Trademarks

Technical Support

Should you have any questions about the use of this product, please contact us by e-mail

support@iXTS.de

Confidentiality

Contents

Introduction

What is the OPC/ADO-Bridge?

A broad variety of industrial systems and devices, in especially *Supervisory Control and Data Acquisition* systems (SCADA), *Data Collection Systems* (DCS) and *Programmable Logic Controllers* (PLCs) support the standardized OPC interface to share their data among each other.

On the other hand companies often store their data in enterprise databases, which usually can be accessed using the ADO technology, as e.g. an Oracle database, a Microsoft SQL Server or any ODBC compliant database.

The *OPC/ADO-Bridge* allows the connection of these two different enterprise sub systems. Data can be retrieved automatically from OPC servers and stored in an ADO database.

There are two main uses for the *OPC/ADO-Bridge*.

- Archive process values from an OPC server to an ADO database to create a process historian. The data can then be analyzed using any tool which can operate on ADO databases.
- Exchange data with other systems which do not support the OPC interface, but can access the database (using ADO or any other way)

Who should use this manual

This manual is intended for all users of *OPC/ADO-Bridge*. It explains how to install and configure the software, as well as how to perform common tasks.

Overview of the manual

This manual consists of several chapters and is structured as follows:

Introduction: This introductory chapter

Getting started: Setting up the *OPC/ADO-Bridge* in a couple of minutes

System description: A short technical description of the *OPC/ADO-Bridge*

User Interface: A detailed description of the user interface, including all dialogs and menus.

Diagnostics: Makes familiar with the logging features of the *OPC/ADO-Bridge*

OPC Compliance

The *OPC/ADO-Bridge* can interact with OPC Servers compliant to the *OPC DA 2.05 Specification* that support all required interfaces and additionally the *IOPCBrowserServerAddressSpace* interface.

References

To learn more about OPC visit the website of the OPC Foundation at: http://www.opcfoundation.org/ The *OPC Data Access Custom Interface Specification* Version 2.05 is also available from the OPC foundation.

To learn more about ADO and its possibilities pleas visit the Microsoft website at http://www.microsoft.com/ or the Microsoft Developer Network at http://msdn.microsoft.com/

Getting started

System requirements

The *OPC/ADO-Bridge* hat both Software and Hardware minimum system requirements. These requirements must be met in order for the server to work properly.

Software requirements

The *OPC/ADO-Bridge* requires, at a minimum, the following software:

- Micosoft Windows 2000, XP or 2003
- Internet Explorer 4.01 Service Pack 1 or better
- OPC DA 2.0x server applications
- *Microsoft ActiveX Data Objects 2.5 Library (ADO)* and *Microsoft ADO Extension for DDL and Security 2.5 Library (ADOX)*. Both components are normally pre-installed on all systems running Windows 2000 or higher, but are also included in the *Microsoft Data Access Components package 2.5 (MDAC)* which can be obtained from http://msdn.microsoft.com/data/mdac/downloads

Hardware requirements

The *OPC/ADO-Bridge* requires, at a minimum, the following hardware:

- Intel Pentium II 500 MHz
- 128 MB RAM
- 20 MB fee hard disk space

Installation

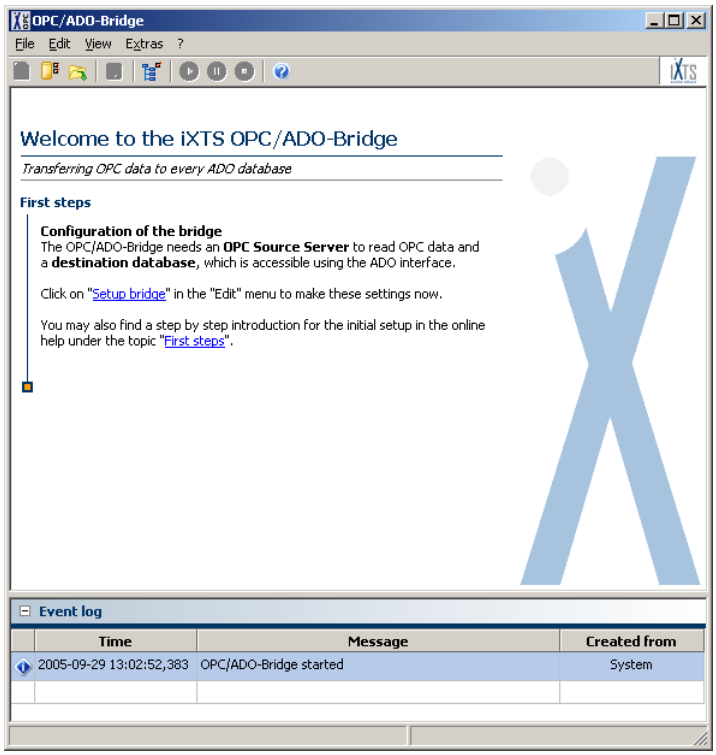
Once you are certain that your system meets the system requirements, you are ready to install the *OPC/ADO-Bridge*.

To install the *OPC/ADO-Bridge* double-click **setup.exe** on your *OPC/ADO-Bridge* media and follow the instructions on the screen.

The setup includes the *OPC Core components* from the OPC Foundation, which are necessary for any OPC application to work properly. If these components are *not* installed on your system, the *OPC/ADO-Bridge* will not work correctly (neither any other standard OPC application).

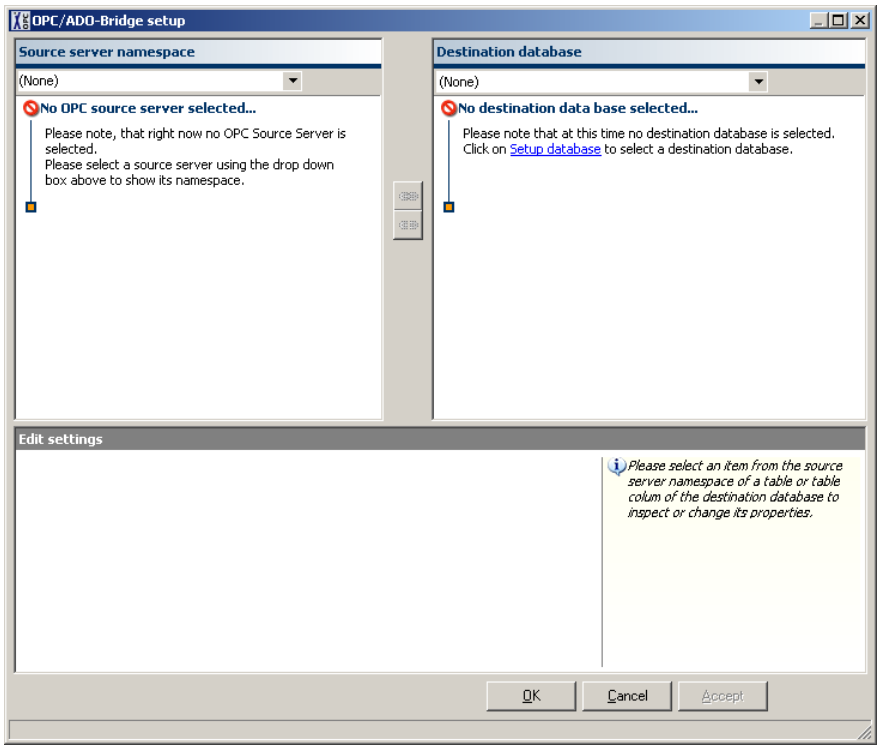
First steps

After installing and starting the *OPC/ADO-Bridge* for the first time you will see the following screen.



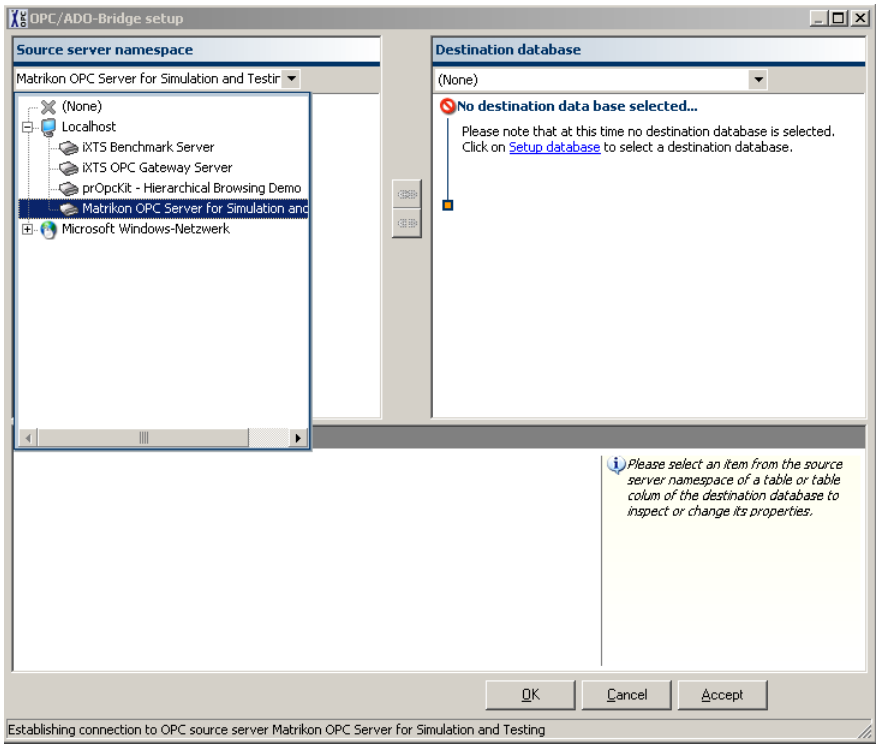
To start working with the bridge an OPC Source Server and an ADO destination database need to be defined and the OPC elements which should be stored in the database must be selected.

To select the OPC server as well as the ADO database and make all necessary setting select **Setup bridge** from the **Edit** menu or click to the corresponding link in the Welcome panel or on the  icon in the toolbar. The following dialog will appear:

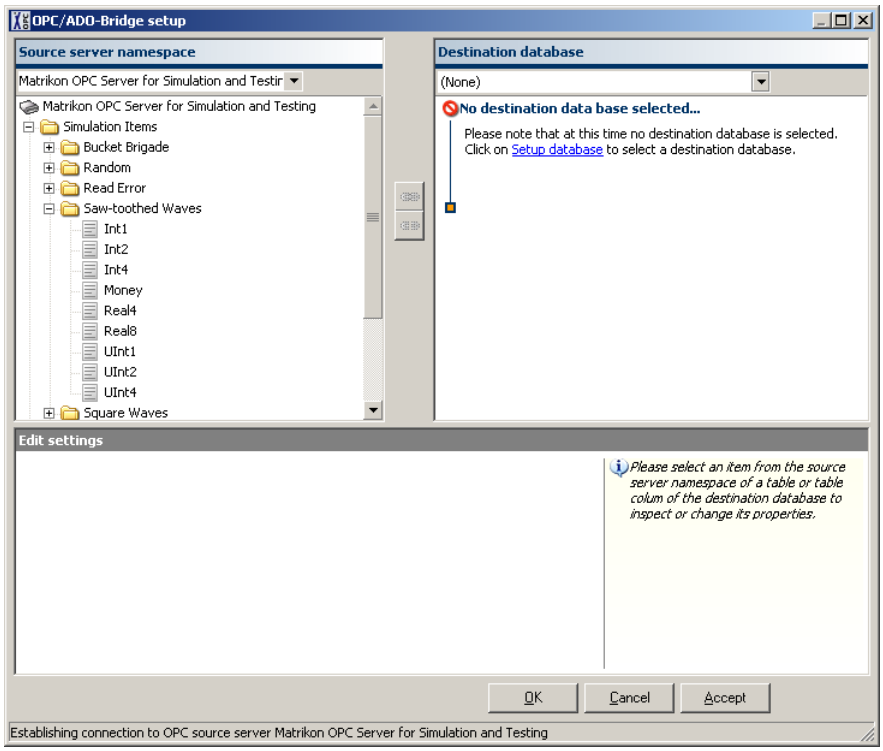


Selecting the OPC source server

Now first an OPC source server must be selected using the dropdown list in the upper left side of the dialog.

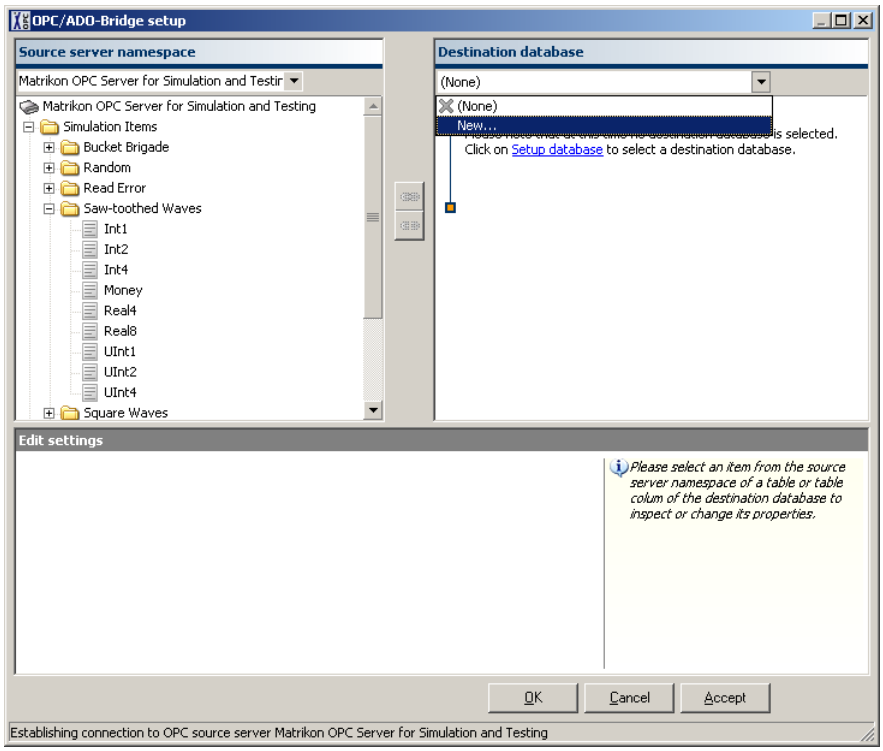


When a source server has been selected its namespace will be retrieved and filled into the tree view on the left side of the dialog.



Selecting the destination ADO database

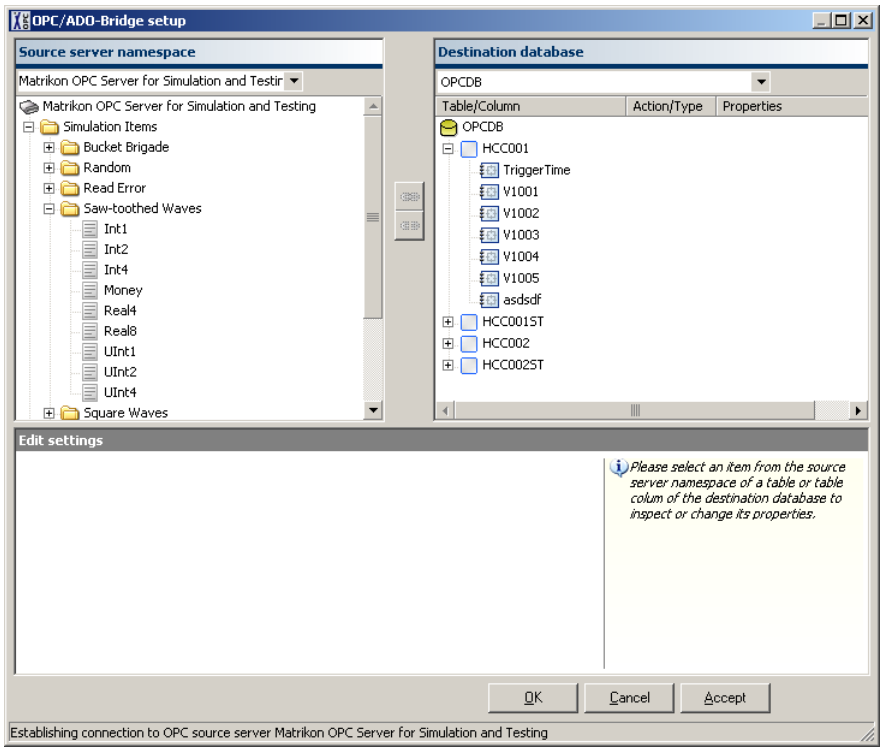
Now the ADO database, which should store the data of the OPC server, must be defined. This can be done by selecting the **New** item in the drop down list on the upper right side



This will open the standard Windows dialog for setting up the properties of an ADO data link and allows the selection of an OLE DB Provider (e.g. for an Oracle Database, a Microsoft SQL Server or an ODBC compliant database) together with the respective database. More information about

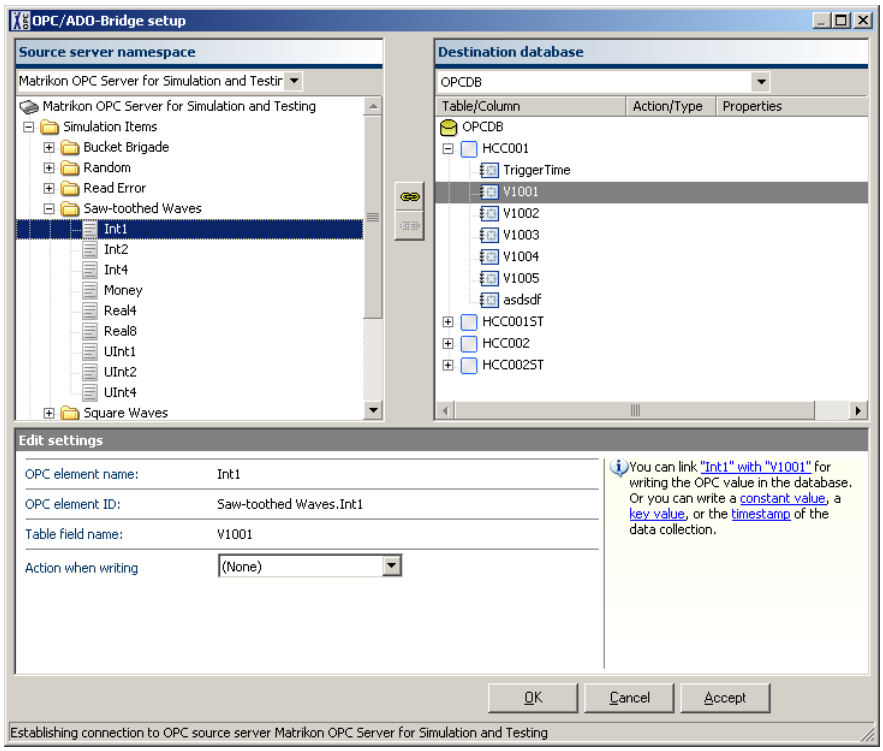
how to set up this properties can be found in the Microsoft Data Link Help which is available from within the dialog by clicking the Help button.

When the database is selected and the connection has been established successfully the database structure is diplayed in form of a tree on the right hand of the dialog.



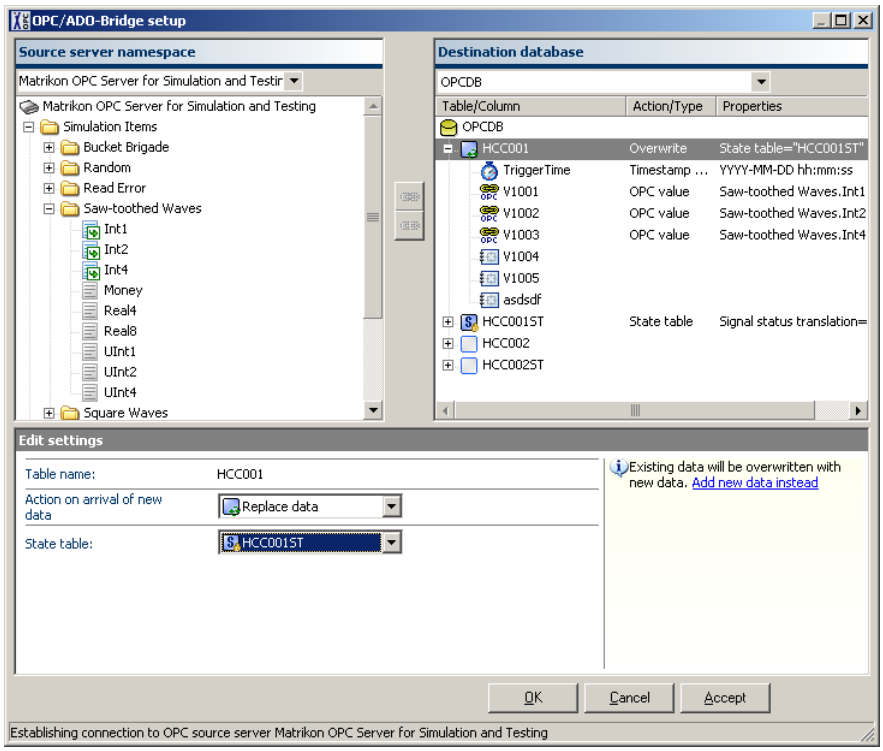
Configuring the data transfer

The main purpose of the *OPC/ADO-Bridge* is transferring data from an OPC server to an ADO destination database, so the first step in configuring the data transfer should be the setup of links between OPC items and database columns (or fields respectively). Establishing a link is a very easy task, as only the source item in the OPC tree and the corresponding table column in the database must be selected and button with the link symbol must be pressed. The linked table column will then store all data received from the OPC server item.

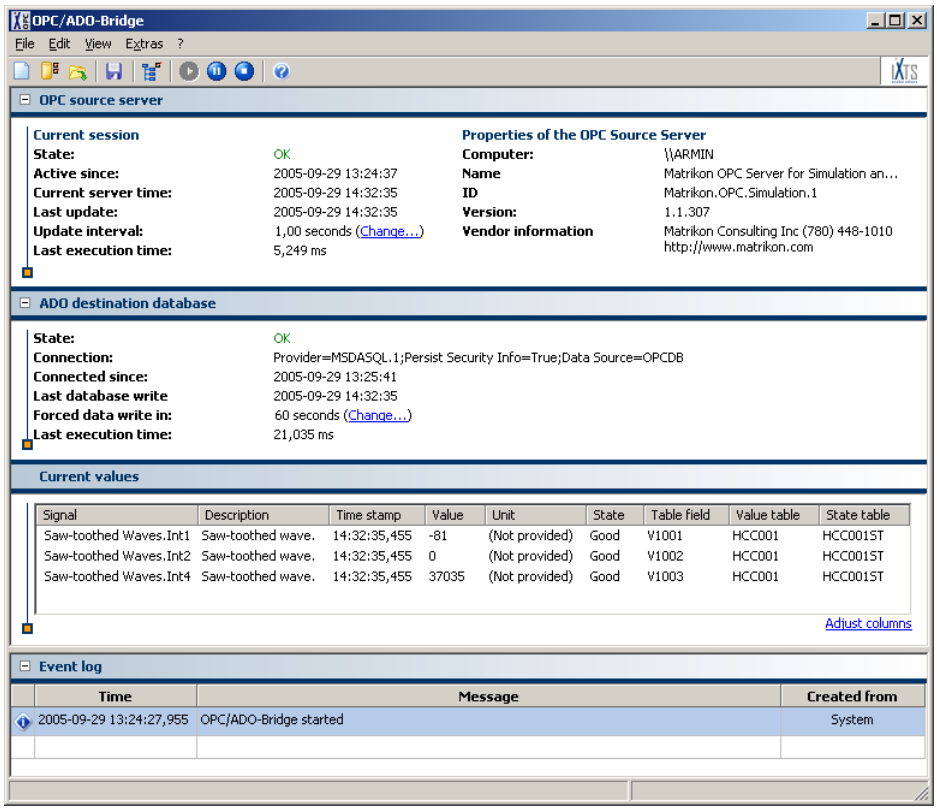


This procedure needs to be repeated for each OPC item that should be stored in the database. Apart of the OPC data also the timestamp of the transmission can be stored by selecting the respective column in the database and changing the **Action when writing** to **Timestamp of data collection**.

To store also the quality information of the received OPC data a state table must be assigned to the table that contains the OPC values (value table). This can be done by first selecting the table itself and the changing its state table property to the desired state table. The state table must contain at least for all used columns in the value table a corresponding column with the same name in the state table, otherwise the assignment as a state table will fail.



After confirming all changes and closing the dialog the bridge is setup completely and the data transmission can start. The main window should now show some information about the OPC source server, the ADO destination database and the transmitted data.



Saving the configuration

To store the configuration for the next time the *OPC/ADO-Bridge* will be started, select **Save** from the **File** menu.

The configuration will be loaded automatically the next time when the *OPC/ADO-Bridge* is started. To change this behavior please refer to the chapter *General options*.

Congratulation. The configuration is completed and the bridge is now ready to be used.

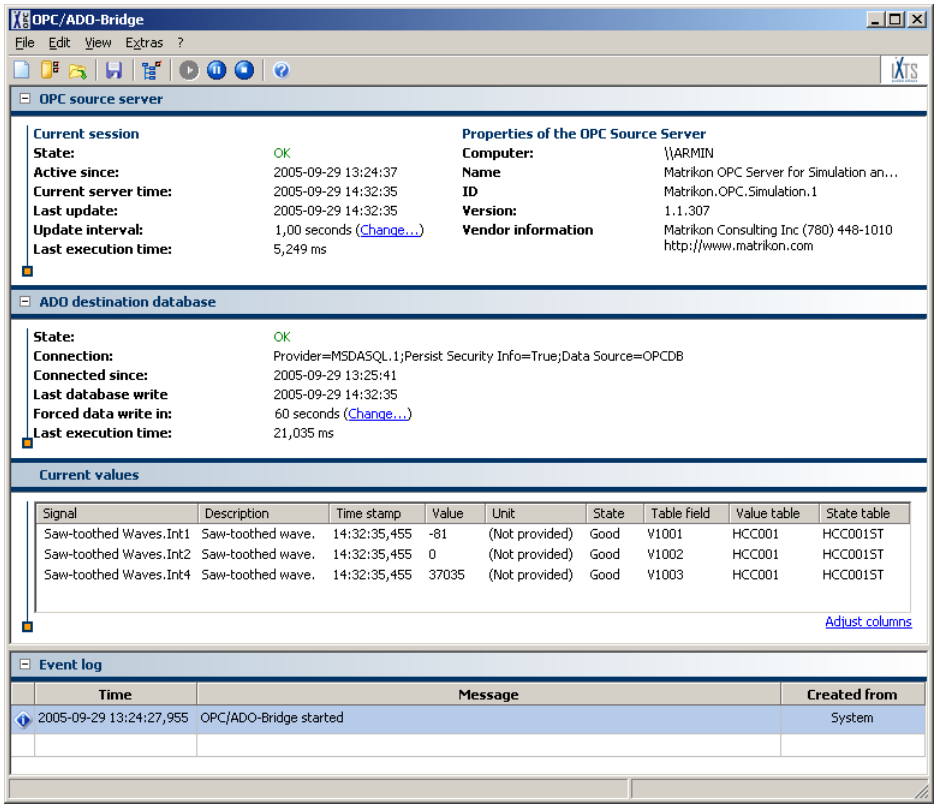
User Interface

The main window

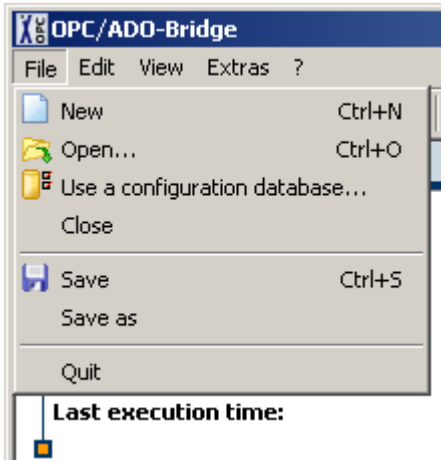
The main window of the *OPC/ADO-Bridge* is divided in four sections.

- OPC source server and session information
- ADO destination database and session information
- Last values retrieved from the OPC server
- Event log where all error, warning or other information is logged. The amount of information shown here depends on the respective settings in the environment options. Please refer to the section *Setting the environment options* to learn more about this.

All sections except the current values section can be collapsed using the commands in the **View** menu, or by clicking on the ⊟-Sign in the upper left corner of the respective section.



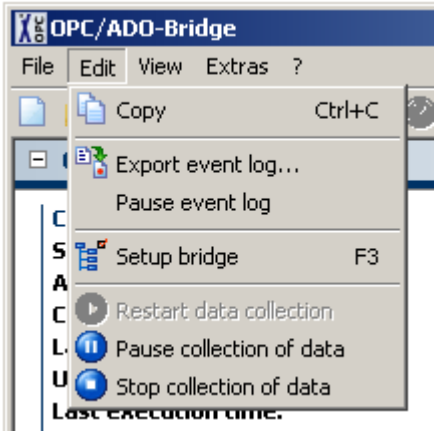
The File Menu



The following commands are available in the **File** menu:

Command	Description
New	Clear the configuration and start with a new one.
Open	Load a previously saved configuration from disk.
Use a configuration database	Load a configuration from a database or create a link to a configuration in a database
Save/Save as	Save the current configuration to a file.
Quit	Shut down the bridge and quit the application.

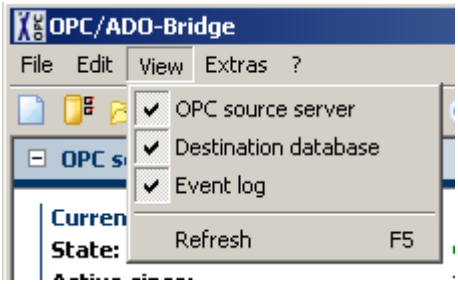
The Edit Menu



The following commands are available in the **Edit** menu:

Command	Description
Copy	Copy the selected entries of the event log to the clipboard. Only available when the event log is displayed and not empty.
Export event log	Exports the currently displayed event log to a file. **Note**: This command is intended to be used to save only the currently displayed events to a file. To save events automatically the auto-store function should be used. Please refer to the section *Setting the environment options*
Pause event log	Pause the event log to prevent updates while navigating in the event log. **Caution**: Pausing the event log may result in losing events, as they are not gathered from the application while the event log is paused
Setup bridge	Opens the setup bridge dialog. Please refer to section *Setting up the bridge* for detailed information
Restart data collection	Restarts the data collection after it was paused or stopped (see below)
Pause data collection	Pauses the data collection from the OPC server. Pausing means that the OPC group is deactivated, so the OPC server will no longer raise callbacks. The connection to the server remains established.
Stop data collection	Stops the data collection from the OPC server. The bridge will release all opened groups on the server and disconnect completely from the server.

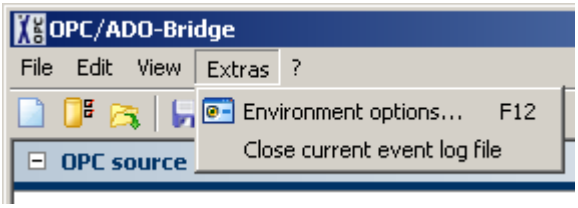
The View Menu



The following commands are available in the **View** menu:

Command	Description
OPC source server	Collapse or expand the OPC source server panel
Destination database	Collapse or expand the Destination database panel
Event log	Collapse or expand the Event log panel
Refresh	Gather the current values from the OPC source server and refresh the shown information

The Extras menu

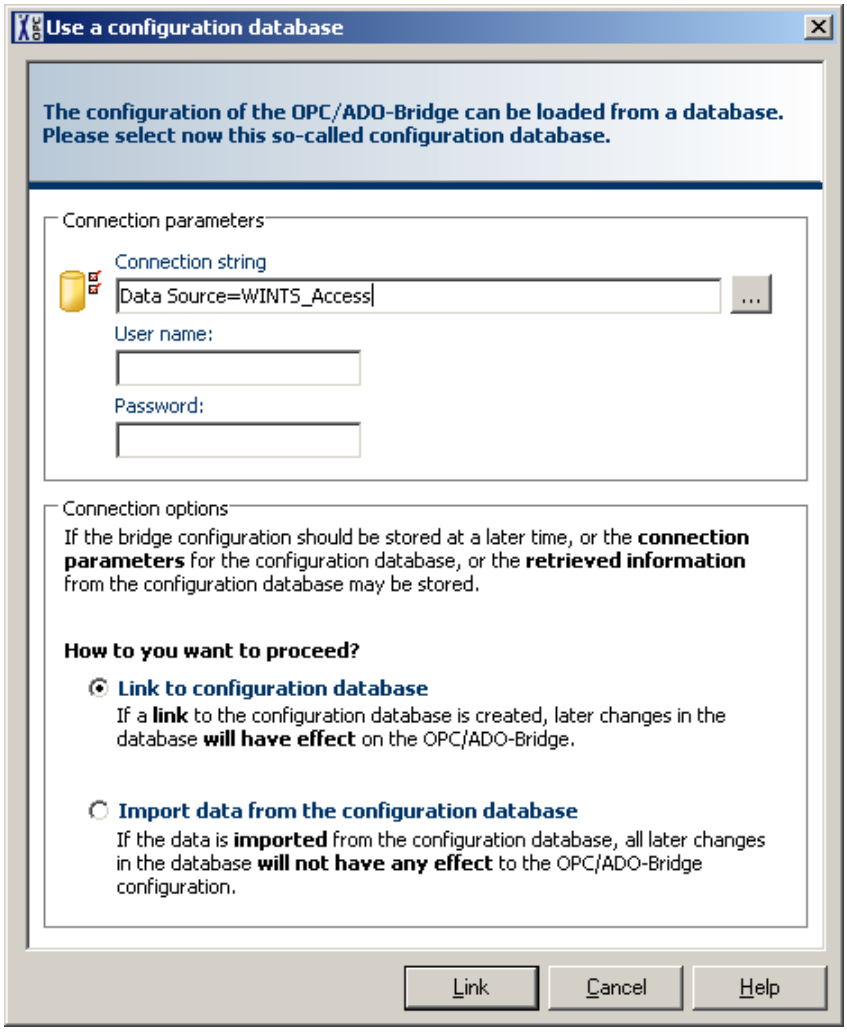


The following commands are available in the **Extras** menu:

Command	Description
Environment options	Open the environment option dialog to set startup options, event logging and default values. Please refer to section *Setting the environment options* for detailed information.
Close current event log file	Close the current event log file. This option is only useful in conjunction with the automatic event log saving to hard disk to finalize the currently opened file and start a new one.

Using a configuration database

As a special option the OPC/ADO-Bridge can also be configured by using a configuration database. In this case all information necessary to setup the bridging functionality between an OPC Server and an ADO database will be retrieved from another ADO database (or from different tables within the same database). To make use of this option, the command **Use a configuration database** in the **File** menu must be selected which will open the following dialog:



The database containing the configuration information must be accessible using ADO and the connection string must be entered in the first line of the dialog. Alternatively the connection string can be assembled using the Microsoft Windows connection string wizard by clicking on the button right beside the input box.

The database can be used in two different ways:

1. **Linked**: This means that the connection information for the database is stored in the local configuration. When the configuration is saved to a file in fact only the link information is saved and when the file will be opened again, the link information is retrieved and the actual OPC/ADO-Bridge configuration is again retrieved from the database.

Using this option allows the dynamic configuration of the OPC/ADO-Bridge by changing the database entries. **Caution**: Changes in the database will have no effect until the respective file containing the database link information is reloaded into the OPC/ADO-Bridge.

2. **Import**: Another possibility is to import the information from the configuration database. Doing this means, that once the configuration information is read from the database the connection information for the database is discarded. When the configuration is saved to a file, the actual configuration will be saved and reloading the file will result in exactly this configuration whether the database have been changed in the meantime or not. This option is useful when the database is only available for a short time or when changes in the database should not automatically affect the OPC/ADO-Bridge configuration.

After selecting one of these two options the dialog can be closed using the **Link/Import** button and the configuration will be load from the database.

Database Format

For a successful configuration import from the database, the database must meet the following requirements:

1. The database must contain at least 3 tables named '**Tab Viper#Parameter**', '**Tab Viper#IO_Data2VM600**' and '**Tab KKS**'
2. The table '**Tab Viper#Parameter**' contains the global configuration parameters and and must have at least the columns '**KanalID**' (Longint), '**ID**' (Longint) and '**Value**' (Text). While for all records concerning the OPC/ADO-Bridge configuration the field 'KanalID' must have the value '0', the values for 'ID' can vary between 1 and 24 whereby the respective 'Value' field contains the actual configuration information.
3. The table '**Tab Viper#IO_Data2VM600**' contains the main information regarding the question which OPC items should be retrieved from the OPC Server, and in which field they should be stored. It must have at least the columns '**Enabled**' (Boolean), '**KKSID**' (Longint), '**MachineID**' (Longint) and '**PointID**' (Longint). Each record with the 'Enabled'-field set to TRUE indicates an active bridge item, with its OPC source item determined by the 'KKSID', the destination table name determined by the 'MachineID' and the destination table field determined by the 'PointID'.
4. The table '**Tab KKS**' is used to translate the numerical KKSID from the 'Tab Viper#IO_Data2VM600' table into an alphanumeric ID string which is then used as the OPC ID at the OPC server.

Example:

The following tables form an exemplary configuration database and may help to understand how to use the configuration database.

'Tab Viper#Parameter'

Id	Value	Comment
1	1	OPC Update interval [s]
4	VIPERAutoDSN	DSN for ODBC destination database
10	DBA_MCURR	Prefix used together with the MachineID to determine the value table name in the destination database
11	DBA_MCURRSTAT	Prefix used together with the MachineID to determine the status table name in the destination database
12	V	Prefix used together with the PointID to determine the fieldname in the value and status table where the OPC values should be stored, Suffix: PointID
13	TriggerTime	Field name for the transmission time stamp
14	YYYY-MM-DD hh:mm:ss	Formatstring for the transmission time stamp
15	True	Overwrite mode enabled (new data will overwrite old data instead of appending new data)
16	60	Forced Data Write time (The current data will be stored at least every 60 Seconds even if they haven't changed)
17	True	Replace bad OPC values (OPC values will be replaced if quality is BAD)
18	-10000	Substitute value for OPC values with quality = BAD
20	localhost	Name or IP of the compute containing the OPC source server
21	OPCServer.WinCC	ID of the OPC source server
22	GOOD	Quality-Translation GOOD
23	UNCERTAIN	Quality-Translation UNCERTAIN
24	BAD	Quality-Translation BAD

'Tab Viper#IO_Data2VM600'

Enabled	KKSID	MachineID	PointID
TRUE	10	1000	100002
FALSE	21	1000	100003
TRUE	30	2000	200005

'Tab KKS'

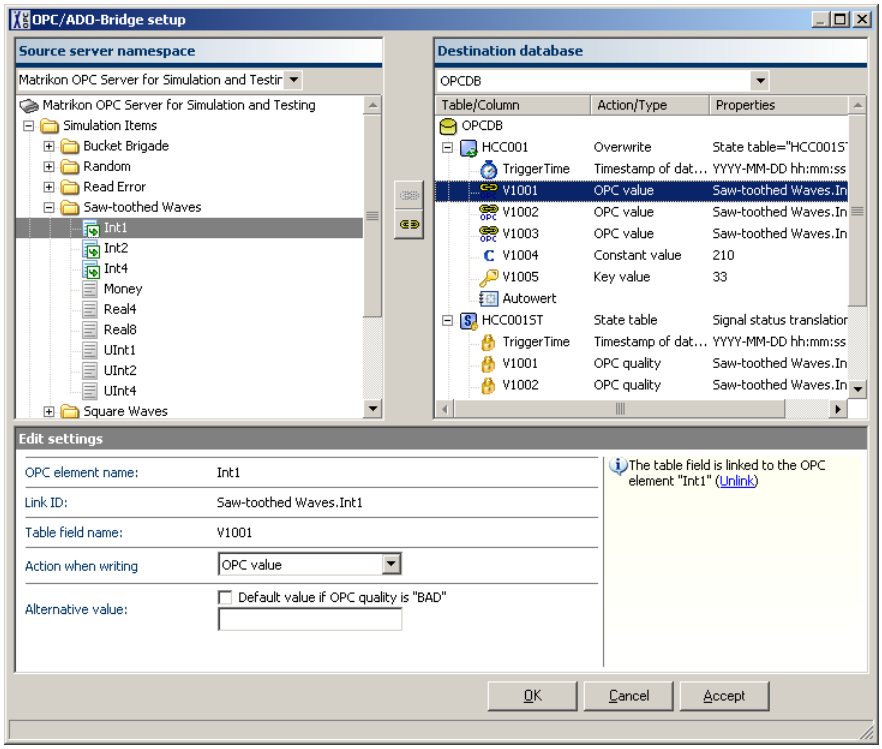
ID	KKS
10	xxMBY10FT010XQ01
21	CKA01EP211XT05
30	CKA01EP211XT04

Resulting OPC/ADO-bridge items

OPC-ID	Destination table	Destination field
xxMBY10FT010XQ01	DBA_MCURR1000	V100002
CKA01EP211XT04	DBA_MCURR2000	V200005

Setting up the bridge

The bridge configuration can be changed using the OPC/ADO-Bridge setup dialog, which can be opened using the **Setup bridge** command in the **Edit** menu.



The setup dialog is split up into three regions. In the upper left area the OPC source server and its namespace is shown. To the right the structure of the selected database is shown. More information for the currently selected items are shown at the bottom of the dialog.

OPC source server namespace

The OPC source server can be selected using the drop-down-list in the upper left area. After selecting a source server, the *OPC/ADO-Bridge* will try to start the source server and fetch its namespace to display it in the tree view below the list box. When the namespace is available any OPC item can be linked to a table column in the database either by using drag'n'drop or the link button between the two tree views.

Already linked items are marked with the icon, the link can be removed by using the unlink button.

ADO database structure

The ADO database which should store the data received from the OPC server can be selected using the drop-down-list in the upper right area. After selecting the ADO database the *OPC/ADO-Bridge* will try to open the database and retrieve its internal structure including table names and columns. The structure is then displayed in a tree view below the list box.

Each table is displayed with one of the following icon:


The table is not used and will not be altered by the *OPC/ADO-Bridge*


The table contains at least one column/field that will be altered by the *OPC/ADO-Bridge*. New data will replace existing data (SQL command *Update*)


The table contains at least one column/field that will be altered by the *OPC/ADO-Bridge*. New data will be added to the table (SQL command *INsert*)


The table is a status table which is linked to another table in the database.

The columns/fields within a table also have different icons depending on what kind of data will be stored in it


The column/field is not used and will not be altered by the *OPC/ADO-Bridge*


The column/field is linked with an OPC item and will store the data received for this item


The column/field will store the timestamp of the last data transmission


The column/field is used as a key field to select a specific record within a table that contains more than one record.


The column/field will store a constant value


The column/field is part of a linked status table (see above) and its properties can therefore no be changed, it is locked.

Item properties

When an item is selected in the OPC Source Server namespace or in the ADO Database structure (or both if two items are linked), the item properties and settings are displayed in the bottom area of the dialog. Most of the properties are self-explaining, so only a few a little bit more “sophisticated” properties should be mentioned here:

When the OPC Source Server  is selected, the **Update interval** can be changed. This interval will be used to set the *Update Rate* in the corresponding OPC Group on the OPC Server.

When the ADO Database  is selected, the **Maximum write interval** can be changed. After this interval a forced write to the database will be executed even if the source data has not been changed. This may be used like a keep alive signal.

When a state table  is selected the **Translation table** for the signal quality can be edited. The translation table allows to replace the numeric OPC values for the signal qualities (‘Good’, ‘Uncertain’, ‘Bad’) with user defined strings or numbers. If no translation table is provided, the numeric value will be written instead.

When a time stamp field  is selected, the format string for the time might be changed. This can be necessary, because different databases are expecting time data with different formats. When the **Save as UTC** checkbox is checked, the time will be converted to the Universal time coordinated which ignores the geographic information and is therefore independent from the local time zone.

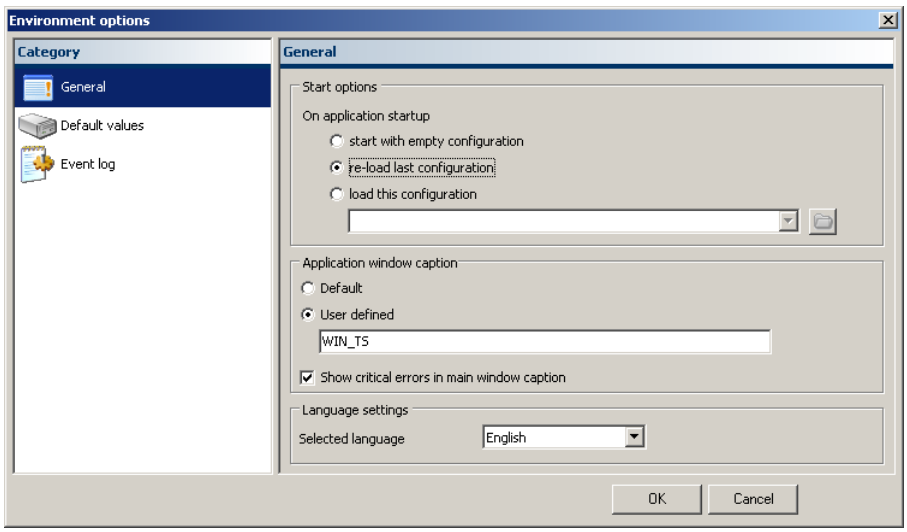
If the data received from an OPC source server for a linked field/column has a bad quality it can be replaced by another value by checking the **Default value if OPC quality is "BAD"** checkbox and providing a substitute value for this case.

After setting up the server configuration click on **Accept** to keep the dialog open or on **OK** to close it.

Setting the environment options

By setting the environment options the overall behavior of the *OPC/ADO-Bridge* can be adjusted to the personal needs. The settings are divided into three categories.

General options

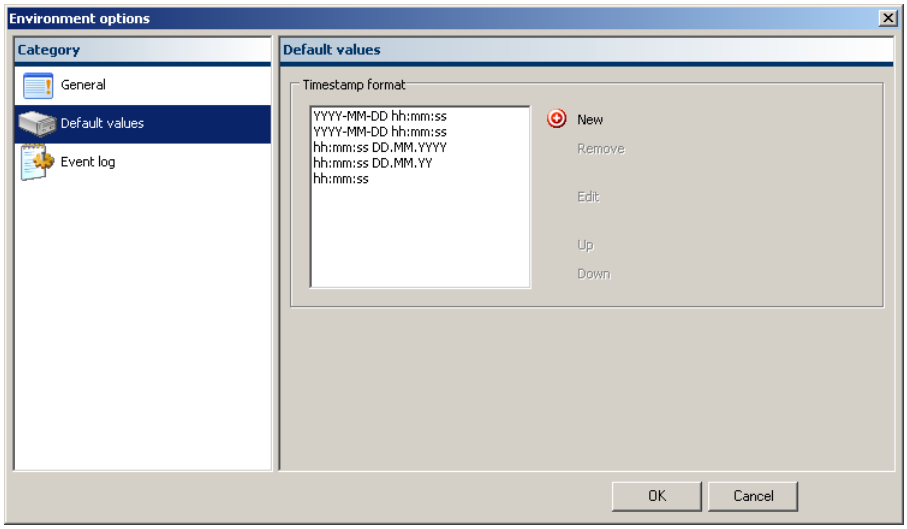


The general options include the start options, which control the application behavior regarding the server configuration at startup. Available options are to **start with empty configuration**, which under normal circumstances doesn't make much sense. The default setting is **re-load last configuration** which means that the application will try to re-load the last used configuration from the previous run. Maybe it is also desirable to define a fixed configuration file which always will be opened without regarding what was opened before. In this case use the option **load this configuration**.

For automation and to ease the status monitoring of the *OPC/ADO-Bridge* application it may be desired to set the **Application window caption** to an individual value and/or to include the current application status in the window caption by checking the **Show critical errors in main window caption** checkbox.

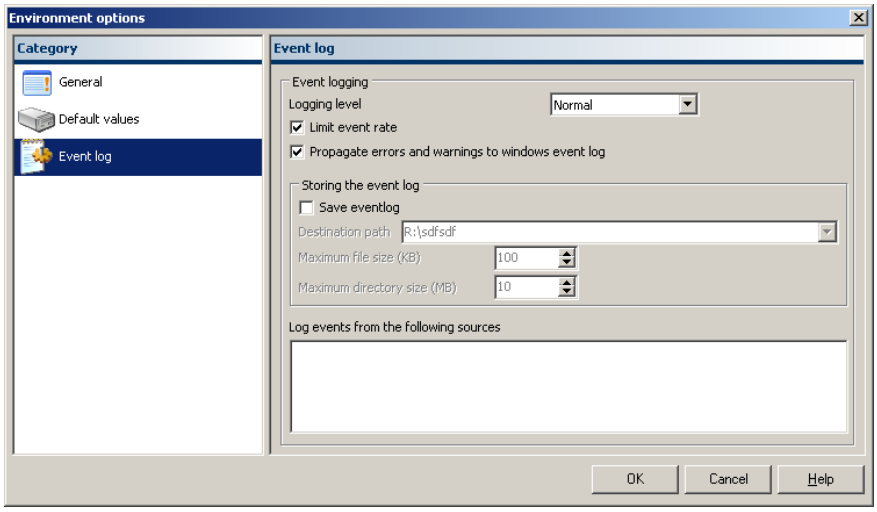
By changing the language settings the language for the applications GUI can be set. Available languages are German and English.

Default values



In the Default values tab the **timestamp formats** can be edited, which are offered in the drop down list for the timestamp format of the database fields which should receive the time stamp of the data collection.

Event log



The event log is used to log all relevant information, including errors and warnings which occur during the applications lifetime. Depending on the situation a more or less communicative system may be desirable. This can be controlled by setting the **logging level**. Six different levels are available: *Nothing, Very little, Little, Normal, Extended* and *All*. The default value is normal which means that all important information is logged without flooding the log with irrelevant details. This setting should normally not be changed, except for diagnostic purposes. See section *Logging* to learn more about this.

The **Limit event rate** checkbox is by default checked, which means that in case that an excessive amount of events is created by the system (e.g. due to a malfunction or due to a high OPC load in conjunction with the logging level set to *All*) the rate is limited by dropping events. This option is included to prevent the system from “hanging” due to a overload caused by consecutive repaint operations. This option should only be unchecked if under certain circumstances the dropping of events is not desired. This may lead to the described consequences.

The meaning of **Propagate errors and warnings to windows event log** is quite self-explaining. This option is checked by default.

If desired the logged events can also be stored on hard disk by checking the **Save event log** checkbox. In this case a directory for the log files must be selected and a maximum file and directory size must be given. When the maximum file size is reached the current file is closed and a new file is created. When the maximum directory size is reached the oldest log file will be deleted automatically.

Another possibility to limit the number of logged events, especially when the logging level is set to *All* is given by the option **Log events from the following sources**. By un-checking some of the listed event sources all events from these sources will be dropped. The list is dynamically created, so at the first start up, the list will be empty. When the first message from a new source is received this sender will be added to the list. By default all senders are enabled and changing this setting should only be necessary for diagnostic purposes. See section *Logging* to learn more about how to use this feature

Diagnostics

Logging

The *OPC/ADO-Bridge* generates a variety of so-called events which contain information about the current program state, occurred error or other details. These events are always logged within the application in the event-log and may also be written automatically to the hard disk.. In case of a problem with the application the log files can be extremely valuable for troubleshooting. By default the automatic saving to hard disk is *disabled* and must be enabled when necessary. Also by default the application logs not very much information, i.e. that the *logging level* is set to normal. Both settings can be changed in the *Event log* tab of the *Environment options* dialog.

The higher the log level, the more information is recorded. However, the performance of the *OPC/ADO-Bridge* may decrease at higher log levels. The recommended logging level is *Normal*. In most cases the *Extended* logging level should be sufficient to troubleshoot the problem. If not the *All* logging level will record all information available within the application.

In general, all errors and other information of immediate importance to the user are logged when the *Little*, or *Very little* logging level is selected, all warnings and other information of not immediate but still significant importance to the user when the N*ormal* logging level is selected. Information of moderate importance to the user will be logged using the *Extended* logging level, and further information especially for the support personnel will be recorded if the logging level is set to *All*.

Especially in higher logging levels a lot of events may be produced so the application load may increase significantly. To avoid this effect, the *OPC/ADO-Bridge* allows to restrict which software classes are allowed to send events to the event log. To configure this feature a deeper understanding of the software architecture is necessary, so under normal circumstances all classes should remain enabled as event sources.