



***BAZA - Driver for Access to Database  
User's Manual***

Doc. No. ENP4008  
Version: 29-08-2005

**ASKOM**<sup>®</sup> and **asix**<sup>®</sup> are registered trademarks of ASKOM Spółka z o.o., Gliwice. Other brand names, trademarks, and registered trademarks are the property of their respective holders.

All rights reserved including the right of reproduction in whole or in part in any form. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without prior written permission from the ASKOM.

ASKOM sp. z o. o. shall not be liable for any damages arising out of the use of information included in the publication content.

Copyright © 2005, ASKOM Sp. z o. o., Gliwice



ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,  
tel. +48 (0) 32 3018100, fax +48 (0) 32 3018101,  
<http://www.askom.com.pl>, e-mail: [office@askom.com.pl](mailto:office@askom.com.pl)

# 1. BAZA - Driver for Access to Database

---

## 1.1. Driver Use

The BAZA driver allows to import data into the **asix** system from databases. The access to database is realized on the basis of the ADO technology. BAZA makes all the data stored in databases available to the **asix** system. The received data may (but they needn't) be stamped with a status and time. The driver also allows reading from other sources like an Excel spreadsheet. If data are stamped with a time, the driver allows to complete historical data in ASPAD archives. When data stored in a database are not stamped with a time, the data newly received by the driver are stamped with a current time. If a datum is not stamped with a status, the status *proper datum* will be assigned to it.

## 1.2. Declaration of Transmission Channel

The full syntax of declaration of transmission channel operating with the BAZA driver:

*logical\_name*=BAZA, *database*

where:

*database*

- field that determines a database.

It can be:

- file name; the name has to allow to distinguish it from the database name i.e. it has to include the "." or "\" character; when declaring a file name as *database*, it is assumed that it is a Microsoft Jet database (Microsoft.Jet.OLEDB.4.0);
- database name; in such case the database is assumed to be serviced by an SQL Server on a local station (SQLOLEDB);
- connection string put in quotation marks; this specification form allows to determine an arbitrary database, i.e. to determine the following parameters: database server localization (e.g. remote computer), user name, password, timeout of establishing a connection etc.; this form allows to specify a database as a DSN name;
- section name, put in square brackets, in which elements of the connection string are placed; this form is used in case of a long connection string.

### EXAMPLE

An exemplary declaration of the transmission channel:

```
[ASMEN]
```

```
....
```

```
;Microsoft Jet database:
```

```
Measurements1 = BAZA, c:\Pomiary.mdb
```

```
;Database defined by DSN data source (of computer or user)
```

```
Measurements2 = "DSN=Pomiary"
```

```
;Database defined by file DSN data source
```

```

Measurements3 = "FILEDSN=C:\BAZA\Pomiary.dsn"

;Database defined by UDL file (Microsoft Data Link)
Measurements4 = Baza,"File Name=C:\BAZA\Pomiary.UDL"

;SQL database named „Pomiary" on local computer:
Measurements5 = BAZA, Pomiary

;SQL database named "Pomiary" on computer "Emisja"
Measurements6 = BAZA,"Provider=SQLOLEDB.1;Data Source=Emisja;Initial Catalog =
Poiary;Integrated Security=SSPI;"

;Parameters of SQL database named „Pomiary" in separate section
Measurements7 = BAZA,[BAZA-POMIARY]

[BAZA-POMIARY]
Provider = SQLOLEDB.1
Data Source = Emisja
Initial Catalog = Pomiary
Integrated Security = SSPI

```

### 1.3. Addressing the Variables

The syntax of symbolic address which is used for the variables belonging to the BAZA driver channel is presented below:

$$\text{array\_declaration}[\text{value\_field}[\text{time\_field}][\text{status\_field}]]$$

where:

*array\_declaration* - expression that determines an array in the database (set of records);

*value\_field* - field name (column) containing the datum value;

*time\_field* - field name (column) containing the data time (Data/Time type);

*status\_field* - field name (column) containing the data status (numerical type – OPC status).

*Value\_field* may be omitted if *array\_declaration* determines the array containing one column.

If *time\_field* is omitted, a current time is taken.

If *status\_field* is omitted, the *proper data* status is assigned to the variable value.

*Array\_declaration* can be:

- name of the array placed in the database;
- query, put in apostrophes or round brackets, sending to the database by the driver in order to data readout;
- symbolic name in the  $\$(name)$  form. The name determines the query the syntax of which is defined in the [BAZA] section in an initialization file.

In the most simple and typical case the array is determined by its name. If, for example, the database contains the array named `Pomiary` containing `Temperature`, `Pressure`, `Time` and `Status` columns, then variable addresses would have the following form:

Pomiary.Temperature.Time.Status  
 Pomiary.Pressure.Time.Status

### EXAMPLE

When the more complex rule defining the records is necessary, other address forms may be used, i.e. with query text, e.g.:

(SELECT \* FROM Pomiary WHERE .....). Temperature.Time.Status

The queries have to be constructed to determine a record set ordered decreasingly according to the time. When reading current data, the driver modifies the query to read the newest record (the driver adds TOP 1 fraze). When reading historical data (only when address contains *time* field), the driver adds or modifies the WHERE fraze to receive data from the specified time range.

The other form of using the queries is use of a query name. The query is defined in the [BAZA] section in an initialization file. The name use allows:

- shortening the address in case of using many variables with the same query but different value fields;
- avoidance of errors in case of necessity of using the characters (which are interpreted by ASMEN in a different manner) in the query;
- optimization of a query number, i.e. if the array of many variables is the result of the query, then the driver sends only one query instead of one by one for each variable.

An exemplary declaration of query name use:

Defition of the variable address:

Name: Temperature

Address: \$(QUERY1).Temperature.Time.Status

Name: Pressure

Address: \$(QUERY1).Pressure.Time.Status

Initialization file:

[BAZA]

Query1 = SELECT \* FROM Pomiary WHERE .....

### NOTICE

*The quotation marks (") should NOT be used in the address.*

## 1.4. Driver Configuration

The BAZA driver may be configured using the [BAZA] section placed in the application initialization file or sections having the same name as channel names in the definition of channels in the [ASMEN] section. The parameters placed in the [BAZA] section apply to all driver channels. The parameters placed in other sections refer to a specified channel. If the parameter is defined both in the [BAZA] section and in the channel section, then the parameter referring to a specified channel has a higher priority.

***No\_TOP =YES/NO***

Meaning - if the item has value NO, then the driver puts in the SQL query the fraze TOP limiting a number of read records. Some databases don't allow using the fraze TOP – then one should declare the value YES.

Default value - NO.

***Log =log\_file***

Meaning - defines a file to which all the diagnostic messages of the driver and the information about the content of telegrams received by the driver will be written.

Default value - lack.

***Max\_history=number***

Meaning - determines a time period from the current moment backwards, for which historical data, saved in the station memory, will be read.

Default value - 30.

Parameter:

*number* - number in days.

***Record\_optimize=number***

Meaning - the parameter refers to variables determined by the name of the array. If the item has the value YES, then only one (common for all variables) SQL query, causing readout of the record containing only the fields that occur in variable addresses, will be formulated for all variables placed in the array. If the parameter has the value NO, all fields in array will be read.

Default value - yes.

***Order =yes/no***

Meaning - if the parameter has the value YES, the driver will formulate an SQL query in such way that read records will be ordered in according to the time fields. If the parameters has the value NO, the records will not be ordered.

Default value - yes.

***History\_records =number***

Meaning - the parameter determines the maximal number of records read from the database once during historical data reading. The

parameter has a sense only when *No\_TOP* parameter takes the value *NO*.  
Default value - 1000.  
Parameters:  
    *number* - number of records.



*UTC =yes/no*

Meaning - determines whether the time written in the database is an UTC time (Universal Time Coordinate or Greenwich Mean Time). If the parameter has the value *YES*, the time is an UTC time, if the parameter has the value *NO* – the time is a local time.

Default value - *no*.

## 1.5. Optimization of Field Number in a Record

The driver formulates one SQL query, common for all variables contained in the same array, causing readout of record including only the fields that occur in variable addresses. If the address even of one variable includes the name of the field that doesn't occur in the array, then reading of all array variables ends with an error. To determine which of variables has an incorrect name, one should declare *Record\_optimize=No* - thanks to such solution the error will refer only to wrongly declared variables. If the error refers to the time field name, one should additionally set the *Order=No* parameter - but absence of ordering may cause reading incorrect data.



<b>1. BAZA - DRIVER FOR ACCESS TO DATABASE .....</b>	<b>3</b>
1.1. DRIVER USE.....	3
1.2. DECLARATION OF TRANSMISSION CHANNEL .....	3
1.3. ADDRESSING THE VARIABLES.....	4
1.4. DRIVER CONFIGURATION.....	5
1.5. OPTIMIZATION OF FIELD NUMBER IN A RECORD.....	7