BAPAS®-DB

Realtime SQL Database System

- ODBC-Driver for ODBC 3.0 -

- English -

Release 1.8

Reg. No. 3.4.7.16.1

Date: 2008-01-02

werum
SOFTWARE & SYSTEMS

BAPAS-DB
Realtime SQL Database System
- ODBC-Driver for ODBC 3.0 -
Release 1.8
Reg. No. 3.4.7.16.1
Date: 2008-01-02

This document is subject to change without notice.

Werum Software & Systems AG
Wulf-Werum-Strasse 3
21337 Lueneburg, Germany
Tel. +49 (0) 4131/8900-0
Fax +49 (0) 4131/8900-20
Info@werum.com
www.werum.com

Document History

Version	Date	Comments
1.6	2003-03-14	Extended version
1.7	2003-07-04	Updated version
1.8	2008-01-02	Updated version for installation under Windows XP

Validity

This manual applies to

- BAPAS-DB Version 8, all releases including and higher than V8.13n (Windows NT, Windows 95)

  and

- BAPAS-DB Version 9, all releases including and higher than V9.07n (Windows NT, Windows 95)

Contents

1. Overview

This manual describes the BAPAS-DB ODBC driver for ODBC Version 3.0.

An ODBC driver is required to access data stocks in ODBC applications (e.g. EXCEL, ACCESS, C programs).

The layer model below illustrates the scheme of how a BAPAS-DB database is addressed in an ODBC application:

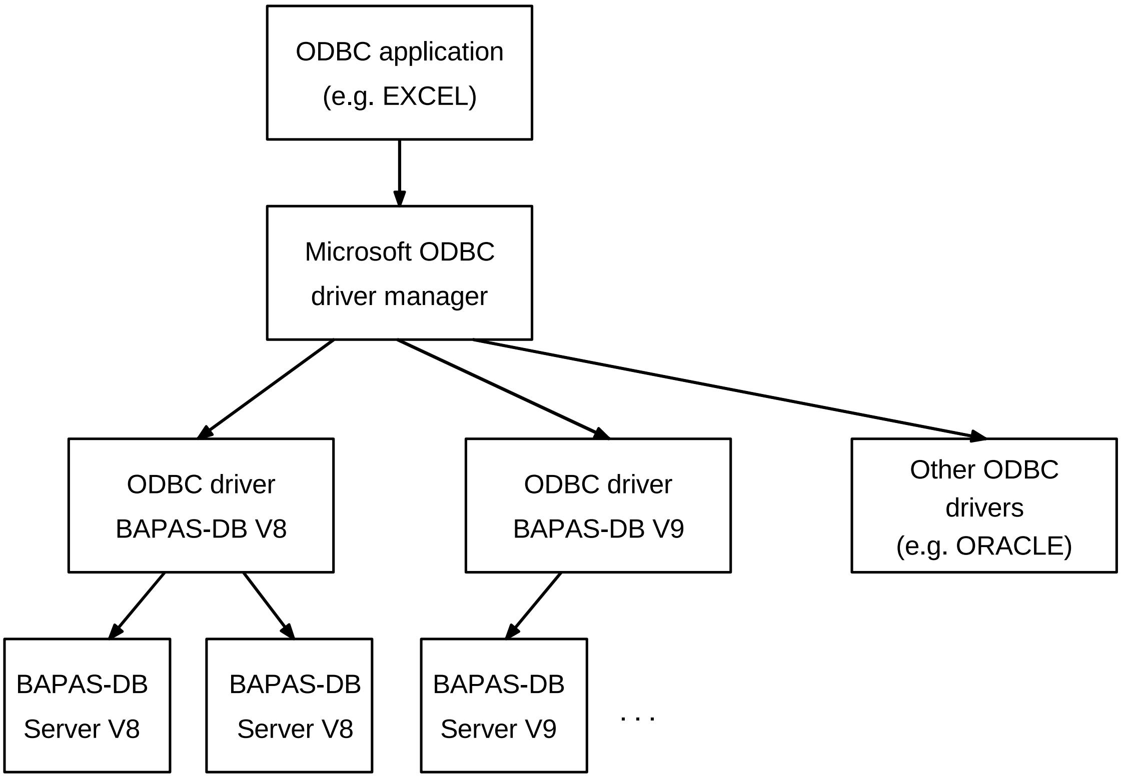


Figure 1: Layer Model of BAPAS-DB Addressing

It is possible to install more than one BAPAS-DB ODBC driver (e.g. for different BAPAS-DB versions). Every BAPAS-DB ODBC driver is able to address several BAPAS-DB servers.

An ODBC application runs under *Windows NT*, *Windows 2000*, or *Windows XP* alike. BAPAS-DB servers run under *Windows*, *Sun Solaris*, or *Linux*. Suitable server operating systems are *OS/2*, *Windows*, and various *Unix* operating systems. ODBC drivers use TCP/IP for communication with the BAPAS-DB servers.

2. Prerequisites

The BAPAS-DB ODBC driver of ODBC 3.0 is available for the operating systems *Windows NT*, *Windows 2000*, and *Windows XP*. To operate it, a TCP/IP card and the corresponding driver software are required.

The software programs Werum used for testing are:

Operating system	Communication	Driver system
Windows NT Windows 2000 Windows XP	ComX2	*PC-NFS* for *Windows NT* by *Sun*

The BAPAS-DB ODBC driver was developed for *Microsoft ODBC* Version 3.0, the associated *Driver Manager* by *Microsoft* is included in the consignment. Also part of the consignment is an *ODBC Administrator*. As far as required, the driver manager and the ODBC administrator are copied to the Windows system directory by the setup program.

3. Scope of Delivery

Delivery is made on disks, on CD, or by eMail in the form of a self-extracting ZIP archive called `bapodbc.<OS>.exe`, with `<OS>` in this name standing for the relevant operating system. Delivery includes an ODBC driver for a BAPAS-DB version.

The files supplied in a consignment among others are:

• `install.bat`	Job to install the ODBC driver
• `odbc.inf`	Configuration file to set up the driver
• `drvstp32.exe`	Setup program
• `bav8c2xt.dll`	ODBC driver for BAPAS-DB Version 8, ComX2, TCP/IP

or

• `bav9c2xt.dll`	ODBC driver for BAPAS-DB Version 9, ComX2, TCP/IP

The `\clnt` subdirectory contains the environment for a BAPAS-DB client. It includes the files:

• `bapcl2x.dll`	BAPAS-DB client DLL
• `comxxd.exe`	Communication domain
• `comxxdinfo.exe`	Program to execute the communication information function
• `comxxsc.dll`	Communication DLL for the BAPAS-DB client
• `dbadmrem.exe`	To set up the user administration (upon creation of user tables using the `\ddl\DLPASS` script)
• `dbcheck.exe`	Program to check the communication line to the server
• `dberror.exe`	Program to output error texts
• `dbsrvccom.exe`	Program to set up the communication demon as *Windows NT* service
• `dynsql2x.dll`	SQL interpreter DLL

4. Installation of the ODBC Driver

4.1 Prerequisites (Logon at Operating System)

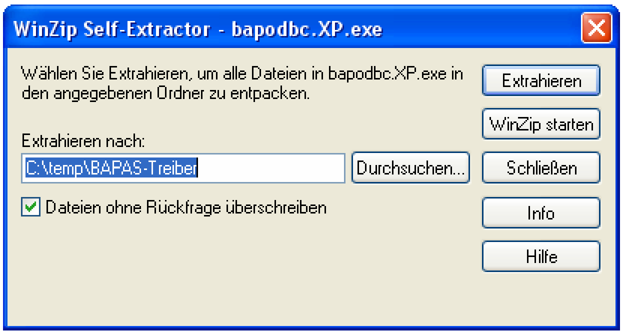
Prerequisite for ODBC driver setup is that you log on as an administrator or as a user with the writing access to registry and system.

1. Log on at the operating system as user having administrator rights.

4.2 Setting up the BAPAS ODBC Driver

The consignment of a BAPAS-DB version includes an ODBC driver. The ODBC driver is filed on the delivery CD packed in a self-extracting ZIP archive called `bapodbc.exe`.

1. Double-click the `bapodbc.exe` ZIP archive to have the file automatically extracted to a temporary directory on the hard disk.



WinZip Self-Extractor - bapodbc.XP.exe

Wählen Sie Extrahieren, um alle Dateien in bapodbc.XP.exe in den angegebenen Ordner zu entpacken.

Extrahieren nach: C:\temp\BAPAS-Treiber

☑ Dateien ohne Rückfrage überschreiben

Figure 2: Self-Extractor Dialog

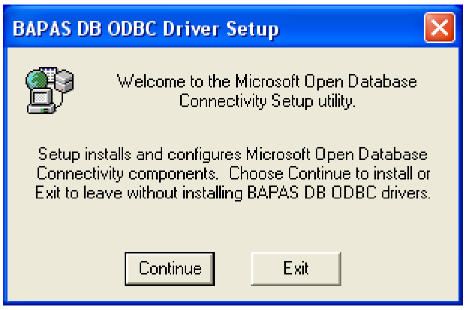
The temporary directory specified in this dialog is where you can find the required files and directories.

2. Open a DOS prompt, change to the directory the files were extracted to, and enter the following call to start the setup process:

```
install [/disk <x:>]
```

With the `/disk <x:>` option you specify the disk partition to install the BAPAS-DB client to. Replace `<x:>` by the required installation partition. If no definition is given for this option, disk `c:` is used (default).

The setup program is called automatically.



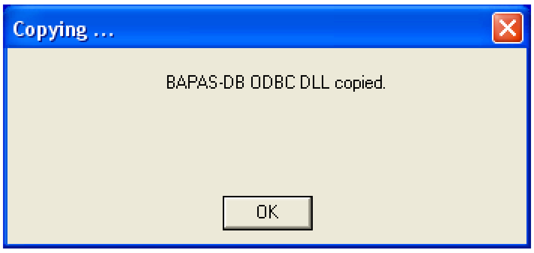
BAPAS DB ODBC Driver Setup

Welcome to the Microsoft Open Database Connectivity Setup utility.

Setup installs and configures Microsoft Open Database Connectivity components. Choose Continue to install or Exit to leave without installing BAPAS DB ODBC drivers.

Figure 3: BAPAS-DB ODBC Driver Setup

3. Click **Continue** to continue the setup process.

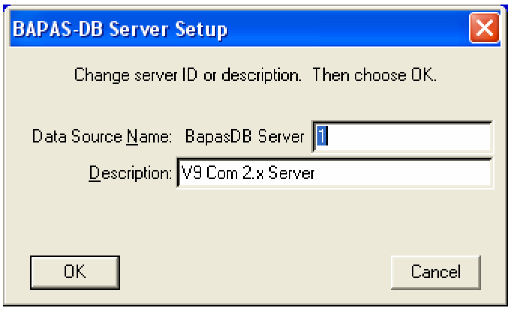


Copying ...

BAPAS-DB ODBC DLL copied.

Figure 4: Information on copied DLL

4. Click **OK** to confirm the copy operation before you enter the ID or name of the BAPAS-DB server in the next dialog.



BAPAS-DB Server Setup

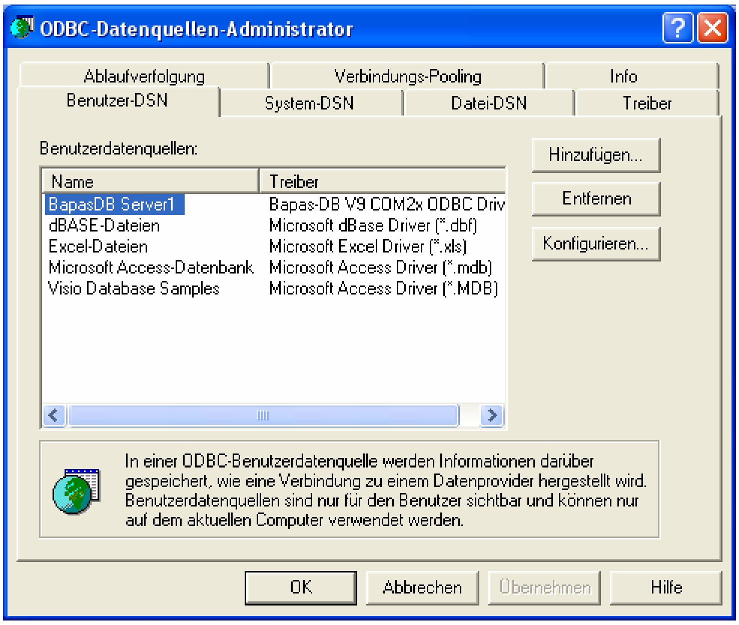
Change server ID or description. Then choose OK.

Data Source Name: BapasDB Server 1

Description: V9 Com 2.x Server

Figure 5: Query of BAPAS-DB Server

5. Enter any required explaining text in the **Description** input field and confirm with **OK**.



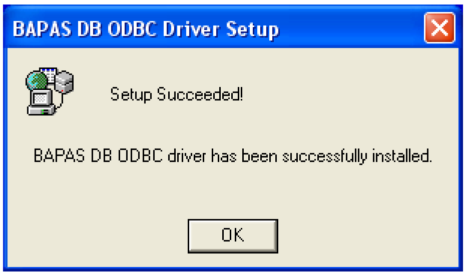
Benutzerdatenquellen:

Name	Treiber
BapasDB Server1	Bapas-DB V9 COM2x ODBC Driv
dBASE-Dateien	Microsoft dBase Driver (*.dbf)
Excel-Dateien	Microsoft Excel Driver (*.xls)
Microsoft Access-Datenbank	Microsoft Access Driver (*.mdb)
Visio Database Samples	Microsoft Access Driver (*.MDB)

In einer ODBC-Benutzerdatenquelle werden Informationen darüber gespeichert, wie eine Verbindung zu einem Datenprovider hergestellt wird. Benutzerdatenquellen sind nur für den Benutzer sichtbar und können nur auf dem aktuellen Computer verwendet werden.

Figure 6: ODBC Data Source Administrator

6. Select the required data source in the **User Data Sources** list and confirm with **OK**.



Setup Succeeded!

BAPAS DB ODBC driver has been successfully installed.

Figure 7: Finish Setup

7. Click **OK** to finish the setup process.

The directories created in the setup process are:

Directory	Contents
<x>:\users\bapasdb\pro	DLL and Exe files
<x>:\users\bapasdb\ddl	DDL scripts for user administration

Table 1: Created Directories and their Contents

4.3 Setting the Environment Variables

1. Add the following entry to the **PATH** user variable:

   ```
   <x>:\users\bapasdb\pro
   ```

Note: (Alternatively, you may copy the DLLs from `<x>:\users\bapasdb\pro` to `%Systemroot%\system32`.)

2. Create the system variables **COMXXDHOST** and **BAPASODBC_PRIO** with the values:

System variable	**Value**
COMXXDHOST	IP address of the system running the demon
BAPASODBC_PRIO	255

Table 2: System Variables to be Created

3. After you have changed the user/system variables, log back on at the system again to make the changes take effect.

4.4 Port Setting

1. Add the following two lines to the `%Systemroot%\system32\drivers\etc\services` file:

   ```
   comxxd 51904/tcp
   comxxd 51904/udp
   ```

4.5 Communication Testing (Optional)

To check the communication, the server must be running and it must be known in the network.

Should the server not yet be known, enter its name in the `hosts` file. In case of *Windows XP*, this file is located in the directory:

```
C:WINDOWS/system32/drivers/etc/
```

Note: Otherwise, use the IP address. That means, the **COMXXDHOST** system variable must have the server's IP address as value.

1. Make the following entry to check that communication with the server works properly:

   ```
   <x>:\users\bapasdb\pro\dbcheck [<server_id>]
   ```

The server ID is an optional statement. If no number is specified, ID=1 will be used. The server will give the following reply:

"Server is alive"

Note: Should error no. 126 occur, no client DLL could be found. This error indicates that `<x>:\users\bapasdb\pro` was not properly entered as value in the **PATH** user variable.

Once you have successfully tested the communication, you may use programs like *MS ACCESS* to access the required data source. To do so, establish a connection to the BAPAS-DB server under *MS ACCESS* and enter the server ID to select the required database. It is not necessary to state user name and password. The only databases to which access is permissible by default are the **10MinRec** database (server ID: 11) and the **Stationbook** database (server ID: 10).

Note: If it is not possible to set up a connection to the required database, you should check that access to this database is allowed in the first place. Whether or not access is allowed and the number of access operations allowed in parallel, is defined by the value of the `ODBC` entry in the `SERVERID` file. This file can be found in every database directory on DAVIS-SHIP systems. In case of the stationbook, for example, it is filed under `$DSHIP_Stationbook/db.`

4.6 Changing/Adding BAPAS-DB Servers

1. To add further BAPAS-DB servers or to change the data of an existing server, use the **ODBC Data Sources Administrator** dialog. Under *Windows XP*, the menu item to open it is: **Start > System Control > Administration > Data Sources (ODBC)**.

5. Configuration of the ODBC Driver

There are various environment variables that can be used to control the ODBC driver's behavior. Available variables are:

BAPASODBC_NULLS

NULL values are used to indicate that a record component does not contain a defined value. So far, there are no NULL values in BAPAS-DB. Programs using the ODBC driver, such as *ACCESS*, however, do work with NULL values. This is the reason why the BAPAS-DB ODBC driver was provided with the option to specify a value for every DDL data type to be interpreted as NULL value.

So, if it is certain that an integer record component of an application can positively not assume a value `-32768`, this value may be defined as NULL value of the `FIXED(15)` data type. When the data record is inserted and a record component of the `FIXED(15)` type contains a NULL value, the ODBC driver will put in the value `-32768`. In reading a data record, the ODBC driver checks the contents of the record components. Should a record component of the `FIXED(15)` type contain the value `-32768`, a value NULL is passed on to *ACCESS*.

The values to be interpreted as NULL values and the data types they apply to are defined in a table to be drawn up by the administrator. This table must be filed in a file called `ODBCNULL.CFG`. The **BAPASODBC_NULLS** environment variable specifies the directory in which the `ODBCNULL.CFG` file is stored. If **BAPASODBC_NULLS** is not set or there is no `ODBCNULL.CFG` file in the specified directory, NULL values will not be interpreted.

A text editor may be used to draw up the table. The structure of the table is this:

Data type	Value
	...
Data type	Value

Any required data type of the DDL is allowed to be used. Hexadecimal and decimal representation of the values is allowed for all types. For `CHAR` and `OPAQUE` types it is also possible to specify a single character enclosed in inverted commas. (The record component will be filled up to its individual length with the specified character.) Not allowed are the characters "`\0`" and "`\n`". In addition to this, decimal point and E-representation are allowed to represent FLOAT values.

Example:

FIXED(15	-32768
FIXED(31)	0x80000000
BIT(16)	0x8000
BIT(32)	0x80000000
CHAR	"@"
CTIME	0

NULL values will not be interpreted for any data types that are not specified in this table.

BAPASODBC_PRIO

The priority by which clients are processed on the BAPAS-DB server is passed on to the server by the client. The priority by which an ODBC application is processed on the server can be set by means of the **BAPASODBC_PRIO** environment variable. It may take a value in the range from 1 to 255 (1 being the highest priority).

If the **BAPASODBC_PRIO** environment variable is not set, "*normal*" priority class is used (corresponds to numerical priority value *156*).

BAPASODBC_STRINGS

Record components of the `CHAR(n)` type are filed in their full length n in BAPAS-DB. In accessing the data records, it may be useful to cut off blanks at the end of a character string for transfer to the calling application (e.g. *ACCESS*). The **BAPASODBC_STRINGS** environment variable makes it possible to cut such blanks. When an environment variable contains a value starting with "*CUT*", any blanks at the end of a character string will be cut off in the transfer operation.

BAPASODBC_SUBST

BAPAS-DB permits the use of underscores in table names. There are tools, however, which do have a problem with underscores in table names, such as *ACCESS*. Should you use *ACCESS* to access an existing and set up BAPAS-DB database, you may replace underscores in table names in the ODBC driver by a minus sign. Prerequisite is that the **BAPASOBDC_SUBST** environment variable is defined and contains a *YES* value.

BAPASODBC_TEST

Should errors occur in using the ODBC driver, you may engage debuggers to help localizing them. The **BAPASODBC_TEST** environment variable determines the test directory. Please refer to chapter 6.4 for an additional description on how to switch debuggers on and off.

BAPASODBC_TXN

To have database jobs executed without any transactions in BAPAS-DB set the **BAPASODBC_TXN** environment variable to "*NO*". Please refer to the *Transaction Handling* section further below in this manual for a detailed description.

BAPASODBC_SCROLL

Setting the **BAPASODBC_SCROLL** environment variable to "*NO*" deactivates the functions **SQLExtendedFetch** and **SQLSetPos**. That means, the ODBC driver acts as if the functions were not implemented at all. This environment variable is available in the versions 8.13r and 9.07r and higher.

Attention: Some of the environment variables mentioned above have been existing for quite some time, even before BAPAS-DB releases 8.11 or 9.05. But they did have different names then. The old names are still accepted, but should not be used in new applications anymore. The environment variables concerned are:

New name	Old name
BAPASODBC_PRIO	ODBCPRIO
BAPASODBC_TEST	ODBCTST
BAPASODBC_TXT	BAPASTXN

6. Configuration of the ODBC User

6.1 User and Access Rights

Various users can be created in BAPAS-DB along with their user names and passwords. For every individual user the DB files he may access can be specified as well as the privileges required to access them.

Details on how to create a user and define the DB files a user may access with specific privileges are available in the "*User Administration*" manual, Reg. No. 3.2.26.

Should no user administration be set up on the DB server, the ODBC user may use any required user name to read DB files but has no write access.

6.2 Limiting the Number of ODBC Users

To limit the number of ODBC clients logged on at the DB server at the same time, there is a specific start parameter to be set when starting the DB server (use a command line or the `SERVERID` parameter file).

odbc <n>

Set in file:	`odbc = n`	

Set in command line:	`-odbc n`
	`/odbc n`
		or

Default:	`32767`	

Example:

The `SERVERID` file (on the server's start directory) contains the lines:

```
ID          = 1
ODBC = 3
```

The server starts with server ID 1 and admits no more than three ODBC clients. A fourth ODBC client would be rejected with an error 283 ("*Too many ODBC clients*").

6.3 Priority of an ODBC Application

The client passes the priority by which clients are processed on the BAPAS-DB server on to the server. The priority by which an ODBC application is processed on the server can be set by means of the **BAPASODBC_PRIO** environment variable. It may take a value in the range from 1 to 255 (1 being the highest priority). If the **BAPASODBC_PRIO** environment variable is not set, a "*normal*" priority class is used (corresponds to numerical priority value *156*).

6.4 Engaging Debuggers

Should errors occur in using the ODBC driver, you may engage debuggers to help localizing them.

First of all, you have to define a directory to file the test settings and the debugger log file. The directory is determined by the **BAPASODBC_TEST** environment variable. When the environment variable fails to be set, `C:\` is used as default test directory.

A file called `flags.tst` is created in the test directory. This file must contain a bit bar defining the required test settings. We recommend using the bit bar:

```
0x0000606b
```

Possible ways to create the `flags.tst` file are:

```
echo 0x0000606b > %BAPASODBC_TEST%\flags.tst        or
```

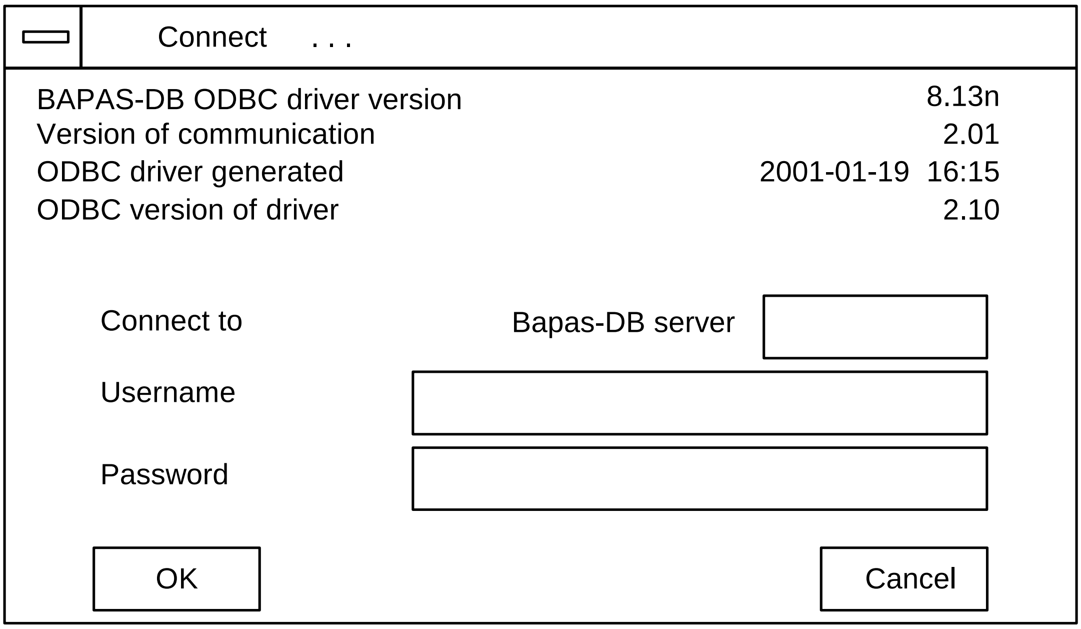
```
echo 0x0000606b > c:\flags.tst
```

When you use one of these methods to create the `flags.tst` file, information of an engaged debugger will be written into the `odbcbap.log` log file in the test directory. The debuggers output is attached to an existing log file, if any. Consequently, it is advisable to delete the log file from time to time.

Debuggers can be deactivated again either by deleting the `flags.tst` file or by inserting a new line containing only 0-bits to be the first line in this file (0x00000000). You may leave the second line with the test bits where it is as only the first line will be evaluated.

7. Processing of BAPAS-DB Tables using ODBC Applications

When setting up the connection, the ODBC driver will show a window that may look like this:



Connect . . .

BAPAS-DB ODBC driver version 8.13n
Version of communication 2.01
ODBC driver generated 2001-01-19 16:15
ODBC version of driver 2.10

Figure 8: Display of ODBC Driver for Connection Setup

(The display layout may vary slightly in the different releases of BAPAS-DB.)

On your entering user name and password and clicking the **OK** button, the connection will be set up. Should errors occur, please check:

- that communication demon and server are running
- that the communication was properly configured on the PC
- that the communication version indicated in the connect window matches the server's communication version
- that entered user name and password are correct

General information:

As *ACCESS* does allow the use of points and square brackets in names, the names of structured record components will be generated in the form:

structure ^ selector (arrayindex)

Array indices always start with 0 (zero).

7.1 Addressing BAPAS-DB Tables with EXCEL

Prerequisite for an import of database tables is that **QUERY** is installed (cf. *EXCEL* documentation).

Addressing a BAPAS-DB table includes the following steps:

- Select the **Import Data** item of the **Data** menu. *EXCEL* now loads the **QUERY** component. The first thing **QUERY** will do is listing all available data sources. Should the required BAPAS-DB server fail to be on the list, you may click the **Others** button to select a different data source.
- Click at the required BAPAS-DB server in the list to select it.
- In the connect window you may change the server ID, if required, enter your user name and password, and click the **OK** button.
- If the connection can be set up successfully, all available DB tables are displayed. Click at the table you want to edit to select it. Selection of several tables for reading access is supported as of BAPAS-DB Version 9.02.
- Close the selection dialog. You may now use the **Data records** item of the **Criteria** menu to define queries. The data is displayed in table form. You may use the **File** menu to transfer it to *EXCEL* for further processing.

As far as the ODBC add-in is installed, you may also use the **SQL.REQUEST** function of *EXCEL* to query the data (cf. *EXCEL* documentation).

The function has the parameters:

1.	Connection string in the form	”DSN = BapasDB Server1; UID = username;
		PWD = password”

2	Output field for the complete connection string	
3.	Number between 1 and 4, controls the connect dialog display	
4.	SQL statement	
	TRUE if you want column headers to be displayed, otherwise FALSE	

Notes: **QUERY** does not always indicate that it is busy (e.g. when loading a large volume of data). A mouse click performed during a loading operation may lead to error number 7026 ("*Previous function not yet finished*"). In this case, you have to restart the query.

Closing a query or quitting **QUERY** during a loading operation may result in a program crash!

Restrictions:

QUERY can display only 255 columns.

No more than 4096 data records can be imported in *EXCEL*.

The SQL statement in **SQL.REQUEST** must not exceed the length of 255 characters; the output is limited to about 6550 fields.

7.2 Addressing with ACCESS

Access operations to a BAPAS-DB table include the following steps:

- Create a new *ACCESS* database or open an existing one.
- Select the **External data, link tables** item of the **File** menu.
- Select **ODBC databases ()** under **File type**.
- Select the **PC data source**.
- Select the required server in the list of data sources.
- Enter user name and password in the connect dialog and click the **OK** button.
- Click at the required DB file to select it and close the selection dialog. You may now edit the selected file.

7.3 Transaction Handling

If you want DB files to be processed by transactions in BAPAS-DB, you have to create them as such in the DDL. By default, BAPAS-DB works without transactions. Opening a transaction is explicitly done by the application (declaring a transaction). Another possible way is to change the default behavior (**SET TRANSACTION MODE**): In this case, the corresponding task will work with implicit transactions.

When an ODBC application executes modifying jobs on a database (e.g. *EXCEL* using the **QUERY** interface), the jobs are probably packed in transactions (e.g. in the **QUERY** interface). Should the DB files to be modified fail to be created with **RECOVERY = TRANSACT** in the BAPAS-DB DDL, the changes will be rejected with:

Error 87 – File not declared for transaction

In most cases, though, there is no point in creating the DB files with **RECOVERY = TRANSACT** in the BAPAS-DB DDL just because of the ODBC interface. For this reason, there is the option to use the environment variable

BAPASODBC_TXN

to tell the ODBC driver that it is generally not supposed to work with transactions. If **BAPASODBC_TXN** contains a "NO" value, the ODBC driver will by default not use any transactions. If **BAPASODBC_TXN** is not defined or has a value unequal “NO”, the transaction jobs will be passed on by the ODBC driver.

Annex: List of Supported ODBC Functions and Options

Supported ODBC 2.x functions are:

Function	Level	Restrictions / remarks
SQLAllocConnect	core	-
SQLAllocEnv	core	-
SQLAllocStmt	core	-
SQLBindCol	core	-
SQLBindParameter	Level 1	-
SQLCancel	core	Since BAPAS-DB does not support asynchronous jobs, this function is empty.
SQLColAttributes	core	All attributes of ODBC 2.x are supported.
SQLColumns	Level 1	-
SQLConnect	core	-
SQLDescribeCol	core	-
SQLDisconnect	core	-
SQLDriverConnect	core	-
SQLError	core	-
SQLExecDirect	core	For restrictions as to possible SQL statements, see BAPAS-DB SQL manual.
SQLExecute	core	-
SQLExtendedFetch	Level 2	As of releases 8.13p and 9.07p, respectively; not available for catalogue functions
SQLFetch	core	-
SQLFreeConnect	core	-
SQLFreeEnv	core	-
SQLFreeStmt	core	-
SQLGetConnectOption	Level 1	For options, see below
SQLGetCursorName	core	-
SQLGetData	Level 1	-
SQLGetFunctions	Level 1	-
SQLGetInfo	Level 1	All information types of ODBC 2.x are supported.
SQLGetStmtOption	Level 1	For options, see below
SQLGetTypeInfo	Level 1	Supported types are: **SQL_CHAR, SQL_BINARY,** **SQL_INTEGER, SQL_SMALLINT,** **SQL_FLOAT, SQL_REAL, SQL_DOUBLE,** **SQL_DATE, SQL_TIME, SQL_TIMESTAMP**
SQLNumParams	core	-

Function	Level	Restrictions / remarks
SQLNumResultCols	core	-
SQLParamData	Level 1	-
SQLPrepare	core	For restrictions as to possible SQL statements, see BAPAS-DB SQL manual.
SQLProcedures	Level 2	A following **SQLFetch** will always supply an error 100 (*not found*)
SQLPutData	Level 1	-
SQLRowCount	core	The number of data records is only known for **INSERT**, **UPDATE**, and **DELETE** statements and the correct number will only be supplied after one of these statements.
SQLSetConnectOption	Level 1	For options, see below
SQLSetCursorName	core	-
SQLSetPos	Level 2	As of releases 8.13p and 9.07p, respectively; not available for catalogue functions; **SQL_UPDATE** and **SQL_ADD** operations are not implemented
SQLSetStmtOption	core	For options, see below
SQLSpecialColumns	Level 1	-
SQLStatistics	Level 1	-
SQLTables	Level 1	-
SQLTransact	core	-

Connection options (**SQLGetConnectOption, SQLSetConnectOption**):

Option	Supported	Comments
SQL_ACCESS_MODE	Yes	-
SQL_AUTOCOMMIT	Yes	-
SQL_CURRENT_QUALIFIER	Yes	Supplies the name of the current log database
SQL_LOGIN_TIMEOUT	No	-
SQL_PACKET_SIZE	No	-
SQL_QUIET_MODE	Yes	-
SQL_TRANSLATE_DLL	No	-
SQL_TRANSLATE_OPTION	No	-
SQL_TXN_ISOLATION	Yes	If transactions are deactivated (cf. chapter 7.3) **SQL_TXN_READ_UNCOMMITTED** is the only setting that is supported; otherwise, it is **SQL_TXN_READ_COMMITTED**.

Statement options (**SQLGetStmtOption, SQLSetStmtOption**):

Option	Supported	Comments
SQL_QUERY_TIMEOUT	No	-
SQL_NOSCAN	Yes	-
SQL_MAX_LENGTH	Yes	-
SQL_ASYNC_ENABLE	No	-
SQL_MAX_ROWS	Yes	-
SQL_CONCURRENCY	Yes	As of releases 8.13p and 9.07p, respectively: **SQL_CONCUR_READ_ONLY and SQL_CONCUR_VALUES**
SQL_RETRIEVE_DATA	Yes	Does not have any effect
SQL_ROWSET_SIZE	Yes	As of releases 8.13p and 9.07p, respectively:
SQL_BIND_TYPE	Yes	As of releases 8.13p and 9.07p, respectively: **SQL_BIND_BY_COLUMN** only
SQL_CURSOR_TYPE	Yes	As of releases 8.13p and 9.07p, respectively: **SQL_CURSOR_FORWARD_ONLY** and **SQL_CURSOR_KEYSET_DRIVEN**
SQL_KEYSET_SIZE	Yes	As of releases 8.13p and 9.07p, respectively
SQL_SIMULATE_CURSOR	No	-
SQL_USE_BOOKMARKS	No	-
SQL_GET_BOOKMARK	No	-
SQL_ROW_NUMBER	No	-