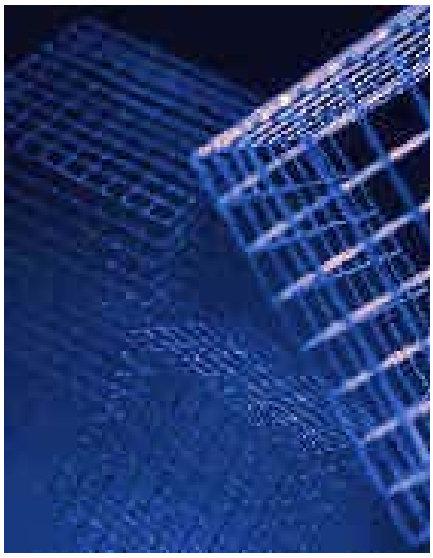
ODBC Manual: SAP DB



Copyright

© Copyright 2003 SAP AG.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation.

For more information on the GNU Free Documentaton License see http://www.gnu.org/copyleft/fdl.html#SEC4.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Typographic Conventions

Type Style	Description
Example text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, titles of graphics and tables.
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.
`EXAMPLE TEXT`	Keys on the keyboard, for example, function keys (such as `F2`) or the `ENTER` key.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

ODBC Manual: SAP DB 5
Supported ODBC Versions 5
Properties of the SAP DB ODBC Driver 5
Supported Data Types 6
SQL Data Types 6
C Data Types 6
Conversion of SQL and C Data Types 7
Processing SQL Statements 7
Database Procedures 8
Cursor Functions 8
ODBC-API Compatibility 8
Core API 9
Level 1 API 10
Remark for SQLGetConnectAttr 11
Remark for SQLGetFunctions 11
Remark for SQLGetInfo 14
Level 2 API 19
Unicode Support 20
SAP DB ODBC Driver Under Microsoft Windows 21
Installation Under Microsoft Windows 21
Creating Data Sources Under Microsoft Windows 23
Options for Data Sources (Microsoft Windows) 23
SQL Mode (Microsoft Windows) 24
Isolation Level (Microsoft Windows) 24
Logging SQL Statements (Microsoft Windows) 25
Installed Files (Microsoft Windows 95 and following) 26
SAP DB ODBC Driver Under UNIX/Linux 26
Installation Under UNIX/Linux 26
Integrating the SAP DB ODBC Driver 27
Creating and Changing Data Sources Under UNIX/Linux 28
Options for Data Sources (UNIX/Linux) 28
SQL Mode (UNIX/Linux) 29
Isolation Level (UNIX/Linux) 29
Logging SQL Statements (UNIX/Linux) 30
Installed Files (UNIX/Linux) 31

ODBC Manual: SAP DB

This manual describes the basics and features of the SAP DB ODBC driver. The explanations apply to SAP DB database systems as of version 7.3.01.

To understand this documentation, you require a basic knowledge of the C programming language, knowledge of SQL, and a general knowledge of database systems.

Use

The SAP DB ODBC driver allows access to the SAP DB database system. Data is accessed using the Structured Query Language (SQL).

Depending on the operating system, the SAP DB ODBC driver is both 32 bit- and 64 bit-compatible. It runs on Linux, Microsoft Windows as of Microsoft Windows 95 and all UNIX systems that are supported by SAP DB.

If the ODBC application is operated on the same host as the database instance, communication takes place using shared memory. Remote access of the ODBC application to the database instance is performed using TCP/IP.

You can download the SAP DB ODBC driver free of charge from the SAP DB homepage *http://www.sapdb.org*.

For a complete description of the ODBC functions and an ODBC syntax reference, see the Microsoft homepage (*http://www.microsoft.com/data/ODBC*) in the ODBC Software Development Kit (SDK) under *ODBC Programmer's Reference*.

For general information about the SAP DB database system, see The SAP DB Database System and on the SAP DB Homepage at http://www.sapdb.org.

Supported ODBC Versions

The SAP DB ODBC driver fulfils the ODBC 3.51 interface specification.

When developing your own applications, note that these differentiate between ODBC 2.x and ODBC 3.x versions. You should therefore use the function *SQLSetEnvAttr* to set the *SQL_ATTR_ODBC_VERSION* before you request a connection handle with *SQLAllocHandle*.

Properties of the SAP DB ODBC Driver

The functionality of the SAP DB ODBC driver is described in more detail for certain areas:

Supported Data Types [Page 6]

Conversion of SQL and C Data Types [Page 7]

Processing SQL Statements [Page 7]

ODBC-API Compatibility [Page 8]

Unicode Support [Page 20]

Supported Data Types

The ODBC interface differentiates between the data types made available by the database system (SQL data types) and the data types used in the ODBC application (C data types).

SQL Data Types [Page 6]

C Data Types [Page 6]

SQL Data Types

Every database system defines its own SQL data types. An ODBC driver processes only those data types that are defined by the associated database system.

You can use the function *SQLGetTypeInfo* to determine how an ODBC driver maps the SQL data types of the database system to the ODBC supported data types and to its own driver-specific data types, as well as the specifications for length, decimal places (scale), and the number of valid digits (precision).

For more information about the data types defined for the SAP DB database system, see the *Reference Manual: SAP DB 7.4*, data type section.

SAP DB data types that are supported by the SAP DB ODBC Driver

Character	CHAR, VARCHAR, LONG, CHAR BYTE, VARCHAR BYTE, LONG BYTE
Numeric	DECIMAL, FIXED, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE PRECISION
Date	DATE, TIME, TIMESTAMP
Other	BOOLEAN, CHAR EBCDIC, VARCHAR EBCDIC

C Data Types

ODBC C data types represent those data types of C variables in which the application data intended for communication with the database is processed.

Relationship Between SQL and C Data Types

fCType	ODBC C Typedef	C Type
SQL_C_CHAR	SQLCHAR *	unsigned char *
SQL_C_SSHORT	SQLSMALLINT	short int
SQL_C_SLONG	SQLINTEGER	long int (32 bit)
SQL_C_USHORT	SQLUSMALLINT	unsigned short int
SQL_C_ULONG	SQLUINTEGER	unsigned long int (32 bit)
SQL_C_FLOAT	SQLREAL	float
SQL_C_DOUBLE	SQLDOUBLE	double

SQL_C_BIGINT	SQLBIGINT	_int64, long (64 bit)
SQL_C_UBIGINT	SQLUBIGINT	unsigned _int64, unsigned long (64 bit)
SQL_C_BINARY	SQLCHAR*	unsigned char*
SQL_C_DATE	SQL_DATE_STRUCT	struct tagDATE_STRUCT { SQLSMALLINT year; SQLSMALLINT month; SQLSMALLINT day;}
SQL_C_TIME	SQL_TIME_STRUCT	struct tagTIME_STRUCT { SQLSMALLINT hour; SQLSMALLINT minute; SQLSMALLINT second;}
SQL_C_TIMESTAMP	SQL_TIMESTAMP_STRUCT	struct tagTIMESTAMP_STRUCT { SQLSMALLINT year; SQLSMALLINT month; SQLSMALLINT day; SQLSMALLINT hour; SQLSMALLINT minute; SQLSMALLINT second; SQLUINTEGER fraction;}
SQL_C_BIT	SQLCHAR	unsigned char

Conversion of SQL and C Data Types

The SAP DB ODBC driver allows the conversion for all supported data types [Page 6].

See also:

Microsoft Homepage (http://www.microsoft.com/data/ODBC) in the ODBC Software Development Kit (SDK) under *ODBC Programmer's Reference*, *Converting Data from SQL to C Data Types* and *Converting Data from C to SQL Data Types* sections.

Processing SQL Statements

The SAP DB ODBC driver supports the complete ODBC-SQL syntax.

For detailed information about the ODBC function calls, see the Microsoft Homepage (http://www.microsoft.com/data/ODBC) in the ODBC Software Development Kit (SDK) under *ODBC Programmer's Reference, Chapter 8: SQL Statements, Escape Sequences in ODBC*, *Scalar Functions* section.

Numeric functions	ABS, ACOS, ASIN, ATAN, ATAN2, CEILING, COS, COT,

	DEGREES, EXP, FLOOR, LOG, MOD, SIGN, SIN, SQRT, TAN, TRUNCATE, PI, RADIANS, ROUND
String functions	CONCAT, LCASE, LEFT, LENGTH, LOCATE, LTRIM, REPLACE, RIGHT, RTRIM, SOUNDEX, SUBSTRING, UCASE,
System functions	DBNAME, IFNULL, USERNAME
Time and date functions	CURDATE, CURTIME, DAYNAME, DAYOFMONTH, DAYOFWEEK, DAYOFYEAR, HOUR, MINUTE, MONTH, MONTHNAME, NOW, SECOND, WEEK, YEAR

```
SELECT {fn MONTH(NOW())} FROM dual
```

The SAP DB ODBC driver has extensions to the usual SQL syntax in the following areas:

Database Procedures [Page 8]

Cursor Functions [Page 8]

Database Procedures

Database procedures can be processed as SQL statements.

See also: *Reference Manual: SAP DB*, Database Procedure section.

The ODBC syntax for procedures does not support return values of functions. Therefore, a specification in the format
`{ ? = call ...}`
is invalid.

Cursor Functions

The SAP DB ODBC driver supports three types of positionable cursors: static, dynamic, and keyset-driven. These differ in the extent to which changes to the database data by the current session or the sessions of other users are visible.

As SAP DB can lock row-by-row, a user can change data in a table even if another user has also opened parts of this table. The prerequisite for this is that the rows to be changed are outside the part of the table that the other user has opened.
Locks for individual pages or entire tables lead to significantly longer waits than row locks in operation with multiple concurrent users.

Using the function *SQLSetPos*, an application can execute positioning INSERT, UPDATE, and DELETE statements.

ODBC-API Compatibility

In the SAP DB ODBC driver, all functions can be executed both under Microsoft Windows and under UNIX/Linux.

The ODBC API is structured into the areas core, level 1, and level 2.

There is an overview of the functions that are defined for the individual API compatibility levels below.

Core API [Page 9]

Level 1 API [Page 10]

Level 2 API [Page 19]

Core API

Overview of the functions that are defined in the SAP DB ODBC driver for the core API

Function	Remarks
SQLAllocConnect	-
SQLAllocEnv	-
SQLAllocStmt	-
SQLBindCol	If the length specification `pcbValueMax` is smaller than the length of a bound LONG column, the SAP DB ODBC driver returns the value `SQL_NO_TOTAL` for `pcbValue`. In all other cases, `pcbValue` specifies the length of the copied bytes. You can then fetch other parts of this LONG column with the *SQLGetData* function.
SQLCancel	-
SQLColAttributes	-
SQLConnect	-
SQLDescribeCol	-
SQLDisconnect	-
SQLError	-
SQLExecDirect	-
SQLExecute	-
SQLFetch	-
SQLFreeConnect	-
SQLFreeEnv	-
SQLFreeStmt	-
SQLGetCursorName	Case-sensitive, maximum length 18 characters Take this into account, for example, with the statement SELECT FOR UPDATE, as identifiers that are not enclosed in quotation marks are automatically converted into uppercase letters. The SAP DB ODBC driver constructs default values for cursor names in accordance with the following schema: `SQL_CURSOR_nnnn`, where `nnnn` corresponds to an internal counter. `SQL_CUR_nnnnnnn`, if `n` >= 10000

SQLNumResultCols	-
SQLPrepare	-
SQLRowCount	Specifies the number of affected rows for all SELECT, INSERT, DELETE statements, and so on. If no result set was created, the value `-1` is displayed for the undeterminable number of affected rows.
SQLSetCursorName	-
SQLTransact	-

Level 1 API

Overview of the functions that are defined in the SAP DB ODBC driver for the Level 1 API

Function	Remark
SQLBindParameter	If you want to process more than one LONG column with an INSERT, UPDATE, or DELETE statement, and bind the parameters with the attribute *SQL_DATA_AT_EXECUTE*, note that, for the *SQLPutData* function, it is only possible to assign the NULL value to one LONG column, at most. If you want to set more than one LONG column to the NULL value with only one statement, you must bind the columns in the length specification (`pcbValue`) using the attribute *SQL_NULL_DATA*. Otherwise, you receive the message `szSqlState = S1000 SQLCODE =-22002.`
SQLColumns	-
SQLDriverConnect	Keyword for the name of the database server: SERVERNODE Keyword for the name of the database instance: SERVERDB
SQLGetConnectAttr	Remark [Page 11]
SQLGetData	-
SQLGetFunctions	Remark [Page 11]
SQLGetInfo	Remark [Page 14]
SQLGetStmtAttr	-
SQLGetTypeInfo	You can determine the data types supported by the SAP DB database system using the function *SQLGetTypeInfo*. The following data types are supported: CHAR, VARCHAR, DECIMAL, FIXED, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE PRECISION, DATE, TIME, TIMESTAMP, LONG, CHAR() BYTE, VARCHAR() BYTE, LONG BYTE, BOOLEAN If the database is configured so that Unicode characters can be processed, SAP DB also supports the following data types: CHAR() UNICODE, VARCHAR() UNICODE, LONG UNICODE
SQLParamData	For LONG columns that contain NULL values, see remark for *SQLBindParameter*

SQLPutData	For LONG columns that contain NULL values, see remark for *SQLBindParameter*
SQLSetConnectAttr	*See SQLGetConnectAttr*
SQLSetParam	-
SQLSetStmtAttr	-
SQLSpecialColumns	-
SQLStatistics	-
SQLTables	The SAP DB ODBC driver supports the following table types (`szTableType`): `ALIAS, RESULT, SNAPSHOT, SYSTEM, SYSTEM TABLE, TABLE, UNLOADED, VIEW`

Remark for SQLGetConnectAttr

List of all supported attributes and return values of the SAP DB ODBC driver

Attributes	Return Values
SQL_ ATTR_ACCESS_MODE	pvParam = `SQL_MODE_READ_WRITE`
SQL_ ATTR_AUTOCOMMIT	pvParam = `SQL_AUTOCOMMIT_ON`
SQL_ATTR_CONNECTION_DEAD	
SQL_ATTR_CURRENT_CATALOG	
SQL_ ATTR_LOGIN_TIMEOUT	pvParam = `15`
SQL_ATTR_ODBC_CURSORS	
SQL_ATTR_QUIET_MODE	
SQL_ ATTR_OPT_TRACE	pvParam = `SQL_OPT_TRACE_OFF =0`
SQL_ ATTR_OPT_TRACEFILE	(The filename set with *SQLSetConnectAttr*, default value: `\SQL.LOG`)
SQL_ ATTR_TRANSLATE_LIB	
SQL_ ATTR_TRANSLATE_OPTION	pvParam = `0`
SQL_ ATTR_TXN_ISOLATION	pvParam = `SQL_TXN_REPEATABLE_READ`
SQL_ MODE = SQL_CONNECT_OPT_START + 2	pvParam = `(1 = INTERNAL, 2 = DB2, 3 = ANSI, 4 = ORACLE)`

Remark for SQLGetFunctions

If the function ID **`SQL_API_ODBC3_ALL_FUNCTION`** is specified, the macro SQL_FUNC_EXISTS returns the result `TRUE` for the following values:

SQL_API_SQLALLOCCONNECT

SQL_API_SQLALLOCENV

SQL_API_SQLALLOCHANDLE
SQL_API_SQLALLOCHANDLESTD
SQL_API_SQLALLOCSTMT
SQL_API_SQLBINDCOL
SQL_API_SQLBINDPARAM
SQL_API_SQLBINDPARAMETER
SQL_API_SQLBROWSECONNECT
SQL_API_SQLBULKOPERATIONS
SQL_API_SQLCANCEL
SQL_API_SQLCLOSECURSOR
SQL_API_SQLCOLATTRIBUTES
SQL_API_SQLCOLUMNPRIVILEGES
SQL_API_SQLCOLUMNS
SQL_API_SQLCONNECT
SQL_API_SQLCOPYDESC
SQL_API_SQLDATASOURCES
SQL_API_SQLDESCRIBECOL
SQL_API_SQLDESCRIBEPARAM
SQL_API_SQLDISCONNECT
SQL_API_SQLDRIVERCONNECT
SQL_API_SQLDRIVERS
SQL_API_SQLENDTRAN
SQL_API_SQLERROR
SQL_API_SQLEXECDIRECT
SQL_API_SQLEXECUTE
SQL_API_SQLEXTENDEDFETCH
SQL_API_SQLFETCH
SQL_API_SQLFETCHSCROLL
SQL_API_SQLFOREIGNKEYS
SQL_API_SQLFREECONNECT
SQL_API_SQLFREEENV
SQL_API_SQLFREEHANDLE
SQL_API_SQLFREESTMT
SQL_API_SQLGETCONNECTATTR
SQL_API_SQLGETCONNECTOPTION
SQL_API_SQLGETCURSORNAME
SQL_API_SQLGETDATA
SQL_API_SQLGETDESCFIELD
SQL_API_SQLGETDESCREC
SQL_API_SQLGETDIAGFIELD

SQL_API_SQLGETDIAGREC
SQL_API_SQLGETENVATTR
SQL_API_SQLGETFUNCTIONS
SQL_API_SQLGETINFO
SQL_API_SQLGETSTMTATTR
SQL_API_SQLGETSTMTOPTION
SQL_API_SQLGETTYPEINFO
SQL_API_SQLMORERESULTS
SQL_API_SQLNATIVESQL
SQL_API_SQLNUMPARAMS
SQL_API_SQLNUMRESULTCOLS
SQL_API_SQLPARAMDATA
SQL_API_SQLPARAMOPTIONS
SQL_API_SQLPREPARE
SQL_API_SQLPRIMARYKEYS
SQL_API_SQLPROCEDURECOLUMNS
SQL_API_SQLPROCEDURES
SQL_API_SQLPUTDATA
SQL_API_SQLROWCOUNT
SQL_API_SQLSETCONNECTATTR
SQL_API_SQLSETCONNECTOPTION
SQL_API_SQLSETCURSORNAME
SQL_API_SQLSETDESCFIELD
SQL_API_SQLSETDESCREC
SQL_API_SQLSETENVATTR
SQL_API_SQLSETPARAM
SQL_API_SQLSETPOS
SQL_API_SQLSETSCROLLOPTIONS
SQL_API_SQLSETSTMTATTR
SQL_API_SQLSETSTMTOPTION
SQL_API_SQLSPECIALCOLUMNS
SQL_API_SQLSTATISTICS
SQL_API_SQLTABLEPRIVILEGES
SQL_API_SQLTABLES
SQL_API_SQLTRANSACT

Remark for SQLGetInfo

List of all attributes and return values supported by the SAP DB ODBC driver

Attributes	Return Values
SQL_ACCESSIBLE_PROCEDURES	rgbInfoValue = "N"
SQL_ACCESSIBLE_TABLES	rgbInfoValue = "N"
SQL_ACTIVE_CONNECTIONS	rgbInfoValue = 0
SQL_ACTIVE_STATEMENTS	rgbInfoValue = 32767
SQL_ALTER_TABLE	rgbInfoValue = SQL_AT_ADD_COLUMN | SQL_AT_DROP_COLUMN | SQL_AT_ADD_COLUMN_DEFAULT | SQL_AT_ADD_CONSTRAINT | SQL_AT_ADD_TABLE_CONSTRAINT | SQL_AT_DROP_COLUMN_CASCADE | SQL_AT_DROP_COLUMN_DEAFAULT | SQL_AT_SET_COLUMN_DEFAULT
SQL_BOOKMARK_PERSISTENCE	rgbInfoValue = SQL_BP_SCROLL | SQL_BP_UPDATE
SQL_CATALOG_LOCATION	rgbInfoValue = 1 (SQL_CL_START)
SQL_CATALOG_NAME_SEPARATOR	rgbInfoValue = "."
SQL_CATALOG_TERM	rgbInfoValue = "Qualifier"
SQL_CATALOG_USAGE	rgbInfoValue = 0x00000000
SQL_COLUMN_ALIAS	rgbInfoValue = "Y"
SQL_CONCAT_NULL_BEHAVIOR	rgbInfoValue = 0 (SQL_CB_NULL)
SQL_CONVERT_BIGINT	
SQL_CONVERT_BINARY	
SQL_CONVERT_BIT	
SQL_CONVERT_CHAR	
SQL_CONVERT_DATE	
SQL_CONVERT_DECIMAL	
SQL_CONVERT_DOUBLE	
SQL_CONVERT_FLOAT	
SQL_CONVERT_FUNCTIONS	rgbInfoValue = 0x00000000L
SQL_CONVERT_INTEGER	
SQL_CONVERT_LONGVARBINARY	rgbInfoValue = 0x00000000L
SQL_CONVERT_LONGVARCHAR	
SQL_CONVERT_NUMERIC	
SQL_CONVERT_REAL	
SQL_CONVERT_SMALLINT	

SQL_CONVERT_TIME	
SQL_CONVERT_TIMESTAMP	
SQL_CONVERT_TINYINT	
SQL_CONVERT_VARBINARY	
SQL_CONVERT_VARCHAR	
SQL_CORRELATION_NAME	rgbInfoValue = `2 (SQL_CN_ANY)`
SQL_CURSOR_COMMIT_BEHAVIOR	rgbInfoValue = `SQL_CB_PRESERVE` | `SQL_CB_DELETED` (SQLMode = ANSI)
SQL_CURSOR_ROLLBACK_BEHAVIOR	rgbInfoValue = `SQL_CB_PRESERVE` | `SQL_CB_DELETED` (SQLMode = ANSI)
SQL_DATA_SOURCE_NAME	rgbInfoValue = (Name of the opened data source)
SQL_DATA_SOURCE_READ_ONLY	rgbInfoValue = `"N"`
SQL_DBMS_NAME	rgbInfoValue = `"SAP DB"`
SQL_DBMS_VER	rgbInfoValue = `"/07.03.nnn"`
SQL_DEFAULT_TXN_ISOLATION	rgbInfoValue = `SQL_TXN_READ_COMMITTED`
SQL_DRIVER_HDBC	rgbInfoValue = (Handle of the driver logon)
SQL_DRIVER_HENV	rgbInfoValue = (Handle of the driver environment)
SQL_DRIVER_HLIB	rgbInfoValue = (Handle of the library)
SQL_DRIVER_HSTMT	rgbInfoValue = (Handle of the driver statement)
SQL_DRIVER_NAME	rgbInfoValue = `"sqlod32.dll"`, `"libsqlod.so"` (UNIX)
SQL_DRIVER_ODBC_VER	rgbInfoValue = `"03.51"`
SQL_DRIVER_ODBC_VER	rgbInfoValue = (Version of the Driver Manager > 3.51)
SQL_DRIVER_VER	rgbInfoValue = `"7.<minor_version>.<correction_level>"`
SQL_EXPRESSIONS_IN_ORDERBY	rgbInfoValue = `"N"`
SQL_FETCH_DIRECTION	rgbInfoValue = `SQL_FD_FETCH_NEXT` | `SQL_FD_FETCH_FIRST` | `SQL_FD_FETCH_LAST` | `SQL_FD_FETCH_PRIOR` | `SQL_FD_FETCH_ABSOLUTE` | `SQL_FD_FETCH_RELATIVE` | `SQL_FD_FETCH_BOOKMARK`
SQL_FILE_USAGE	rgbInfoValue = `0 (SQL_FILE_NOT_SUPPORTED)`

SQL_GETDATA_EXTENSIONS	rgbInfoValue = SQL_GD_ANY_COLUMN | SQL_GD_ANY_ORDER | SQL_GD_ANY_BOUND | SQL_GD_ANY_BLOCK
SQL_GROUP_BY	rgbInfoValue = SQL_GB_GROUP_BY_CONTAINS_SELECT
SQL_IDENTIFIER_CASE	rgbInfoValue = 1 (SQL_IC_UPPER)
SQL_IDENTIFIER_QUOTE_CHAR	rgbInfoValue = "\""
SQL_INTEGRITY	rgbInfoValue = "N"
SQL_KEYWORDS	rgbInfoValue = (List of key words, dependent on SQLMode)
SQL_LIKE_ESCAPE_CLAUSE	rgbInfoValue = "Y"
SQL_LOCK_TYPES	rgbInfoValue = 1 SQL_LCK_NO_CHANGE
SQL_MAX_BINARY_LITERAL_LEN	rgbInfoValue = 0
SQL_MAX_CATALOG_NAME_LEN	rgbInfoValue = 0
SQL_MAX_CHAR_LITERAL_LEN	rgbInfoValue = 0
SQL_MAX_COLUMN_NAME_LEN	rgbInfoValue = 32
SQL_MAX_COLUMNS_IN_GROUP_BY	rgbInfoValue = 128
SQL_MAX_COLUMNS_IN_INDEX	rgbInfoValue = 16
SQL_MAX_COLUMNS_IN_ORDER_BY	rgbInfoValue = 128
SQL_MAX_COLUMNS_IN_SELECT	rgbInfoValue = 1023
SQL_MAX_COLUMNS_IN_TABLE	rgbInfoValue = 1023
SQL_MAX_CONCURRENT_ACTIVITIES	rgbInfoValue = 32767
SQL_MAX_CURSOR_NAME_LEN	rgbInfoValue = 32
SQL_MAX_DRIVER_CONNECTIONS	rgbInfoValue = 0
SQL_MAX_INDEX_SIZE	rgbInfoValue = 1024
SQL_MAX_OWNER_NAME_LEN	rgbInfoValue = 32
SQL_MAX_PROCEDURE_NAME_LEN	rgbInfoValue = 32
SQL_MAX_QUALIFIER_NAME_LEN	rgbInfoValue = 0
SQL_MAX_ROW_SIZE	rgbInfoValue = 0
SQL_MAX_ROW_SIZE_INCLUDES_LONG	rgbInfoValue = "N"
SQL_MAX_SCHEMA_NAME_LEN	rgbInfoValue = 32
SQL_MAX_STATEMENT_LEN	rgbInfoValue = 0
SQL_MAX_TABLE_NAME_LEN	rgbInfoValue = 32
SQL_MAX_TABLES_IN_SELECT	rgbInfoValue = 64
SQL_MAX_USER_NAME_LEN	rgbInfoValue = 32
SQL_MULT_RESULT_SETS	rgbInfoValue = "Y"
SQL_MULTIPLE_ACTIVE_TXN	rgbInfoValue = "Y"

SQL_NEED_LONG_DATA_LEN	rgbInfoValue = "N"
SQL_NON_NULLABLE_COLUMN	rgbInfoValue = 1 (SQL_NNC_NON_NULL)
SQL_NULL_COLLATION	rgbInfoValue = 1 (SQL_NC_LOW)
SQL_NUMERIC_FUNCTIONS	rgbInfoValue = SQL_FN_NUM_ABS | SQL_FN_NUM_ACOS | SQL_FN_NUM_ASIN | SQL_FN_NUM_ATAN | SQL_FN_NUM_ATAN2 | SQL_FN_NUM_CEILING | SQL_FN_NUM_COS | SQL_FN_NUM_COT | SQL_FN_NUM_EXP | SQL_FN_NUM_FLOOR | SQL_FN_NUM_LOG | SQL_FN_NUM_MOD | SQL_FN_NUM_SIGN | SQL_FN_NUM_SIN | SQL_FN_NUM_SQRT | SQL_FN_NUM_TAN | SQL_FN_NUM_PI | SQL_FN_NUM_DEGREES | SQL_FN_NUM_RADIANS | SQL_FN_NUM_ROUND
SQL_ODBC_API_CONFORMANCE	rgbInfoValue = 2 (SQL_OAC_LEVEL2)
SQL_ODBC_SAG_CLI_CONFORMANCE	rgbInfoValue = 1 (SQL_OSCC_COMPLIANT)
SQL_ODBC_SQL_CONFORMANCE	rgbInfoValue = 2 (SQL_OSC_EXTENDED)
SQL_ODBC_SQL_OPT_IEF	rgbInfoValue = "N"
SQL_ODBC_VER	rgbInfoValue = "03.520000"
SQL_ORDER_BY_COLUMNS_IN_SELECT	rgbInfoValue = "N"
SQL_OUTER_JOINS	rgbInfoValue = "Y"
SQL_OWNER_TERM	rgbInfoValue = "Owner"
SQL_OWNER_USAGE	rgbInfoValue = SQL_OU_DML_STATEMENTS | SQL_OU_PROCEDURE_INVOCATION | SQL_OU_TABLE_DEFINITION | SQL_OU_INDEX_DEFINITION | SQL_OU_PRIVILEGE_DEFINITION
SQL_POS_OPERATIONS	rgbInfoValue = SQL_POS_UPDATE | SQL_POS_DELETE | SQL_POS_ADD (only allowed with dynamic cursors) | SQL_POS_POSITION | SQL_POS_REFRESH
SQL_POSITIONED_STATEMENTS	rgbInfoValue = SQL_PS_POSITIONED_DELETE | SQL_PS_POSITIONED_UPDATE | SQL_PS_SELECT_FOR_UPDATE
SQL_PROCEDURE_TERM	rgbInfoValue = "stored procedure"

SQL_PROCEDURES	rgbInfoValue = "Y"
SQL_QUALIFIER_LOCATION	rgbInfoValue = 1 (SQL_CL_START)
SQL_QUALIFIER_NAME_SEPARATOR	rgbInfoValue = "."
SQL_QUALIFIER_TERM	rgbInfoValue = "Qualifier"
SQL_QUALIFIER_USAGE	rgbInfoValue = 0x00000000
SQL_QUOTED_IDENTIFIER_CASE	rgbInfoValue = 3 (SQL_IC_SENSITIVE)
SQL_ROW_UPDATES	rgbInfoValue = "Y"
SQL_SCHEMA_TERM	rgbInfoValue = "Owner"
SQL_SCHEMA_USAGE	rgbInfoValue = SQL_OU_DML_STATEMENTS | SQL_OU_PROCEDURE_INVOCATION | SQL_OU_TABLE_DEFINITION | SQL_OU_INDEX_DEFINITION | SQL_OU_PRIVILEGE_DEFINITION
SQL_SCROLL_CONCURRENCY	rgbInfoValue = SQL_SCCO_READ_ONLY | SQL_SCCO_LOCK | SQL_SCCO_OPT_ROWVER | SQL_SCCO_OPT_VALUES
SQL_SCROLL_OPTIONS	rgbInfoValue = SQL_SO_FORWARD_ONLY | SQL_SO_KEYSET_DRIVEN | SQL_SO_DYNAMIC | SQL_SO_STATIC
SQL_SEARCH_PATTERN_ESCAPE	rgbInfoValue = "\"
SQL_SERVER_NAME	rgbInfoValue = (Name of the database server)
SQL_SPECIAL_CHARACTERS	rgbInfoValue = "#@$"
SQL_STATIC_SENSITIVITY	rgbInfoValue = 0x00000000
SQL_STRING_FUNCTIONS	rgbInfoValue = SQL_FN_STR_LEFT | SQL_FN_STR_LTRIM | SQL_FN_STR_LENGTH | SQL_FN_STR_LCASE | SQL_FN_STR_REPLACE | SQL_FN_STR_RIGHT | SQL_FN_STR_RTRIM | SQL_FN_STR_SUBSTRING | SQL_FN_STR_UCASE | SQL_FN_STR_SOUNDEX
SQL_SUBQUERIES	rgbInfoValue = SQL_SQ_COMPARISON | SQL_SQ_EXISTS | SQL_SQ_IN | SQL_SQ_CORRELATED_SUBQUERIES
SQL_SYSTEM_FUNCTIONS	rgbInfoValue = SQL_FN_SYS_USERNAME | SQL_FN_SYS_DBNAME | SQL_FN_SYS_IFNULL
SQL_TABLE_TERM	rgbInfoValue = "Table"

SQL_TIMEDATE_ADD_INTERVALS	rgbInfoValue = `0x00000000`
SQL_TIMEDATE_DIFF_INTERVALS	rgbInfoValue = `0x00000000`

SQL_TIMEDATE_FUNCTIONS	rgbInfoValue =
	`SQL_FN_TD_NOW`
	`| SQL_FN_TD_CURDATE`
	`| SQL_FN_TD_DAYOFMONTH`
	`| SQL_FN_TD_DAYOFWEEK`
	`| SQL_FN_TD_DAYOFYEAR`
	`| SQL_FN_TD_MONTH`
	`| SQL_FN_TD_WEEK`
	`| SQL_FN_TD_YEAR`
	`| SQL_FN_TD_CURTIME`
	`| SQL_FN_TD_HOUR`
	`| SQL_FN_TD_MINUTE`
	`| SQL_FN_TD_SECOND`
	`| SQL_FN_TD_DAYNAME`
	`| SQL_FN_TD_MONTHNAME`

SQL_TXN_CAPABLE	rgbInfoValue = `2 (SQL_TC_ALL)`

SQL_TXN_ISOLATION_OPTION	rgbInfoValue =
	`SQL_TXN_READ_UNCOMMITTED`
	`| SQL_TXN_READ_COMMITTED`
	`| SQL_TXN_REPEATABLE_READ`
	`| SQL_TXN_SERIALIZABLE`

SQL_UNION	rgbInfoValue =
	`SQL_U_UNION`
	`| SQL_U_UNION_ALL`

SQL_USER_NAME	rgbInfoValue = (Name of the active user)

Level 2 API

Overview of the functions that are defined in the SAP DB ODBC driver for the Level 2 API

Function	Remark
SQLBrowseConnect	
SQLColumnPrivileges	
SQLDataSources (*)	
SQLDescribeParam	
SQLExtendedFetch	You can only insert rows using the function with the dynamic cursor model (`SQL_CURSOR_DYNAMIC`). This is not possible with any other cursor models. UPDATE and DELETE statements are possible both with the static (`SQL_CURSOR_STATIC`) and with the keyset-driven cursor (`SQL_CURSOR_KEYSETDRIVEN`) models.
SQLFetchScroll	
SQLForeignKeys	
SQLMoreResults	The function calls *SQLFreeStmt* with the attribute *SQL_CLOSE* to close

	the previous result set and then returns `SQL_NO_DATA_FOUND`.
SQLNativeSql	
SQLNumParams	
SQLParamOptions	
SQLPrimaryKeys	
SQLProcedureColumns	
SQLProcedures	
SQLSetPos	All attributes are permissible. The attribute *SQL_ADD* is only possible in the dynamic cursor model (`SQL_CURSOR_DYNAMIC`). Only the attribute *SQL_LOCK_NO_CHANGE* is permissible for the fLock argument.
SQLSetScrollOptions	
SQLTablePrivileges	

Unicode Support

The SAP DB ODBC driver supports Unicode. This means that the functions listed below are also available, irrespective of whether the data source supports Unicode or not.

Under Microsoft Windows, the Driver Manager basically uses the Unicode variants of the ODBC API. If, for example, the *SQLConnect* function is called within the application, *SQLConnectW* is called in the ODBC driver. The conversions between ASCII and Unicode are performed by the Driver Manager.

Overview of the functions that are defined in the SAP DB ODBC driver for access to Unicode data

SQLBrowseConnectW

SQLColAttributesW

SQLColAttributeW

SQLColumnPrivilegesW

SQLColumnsW

SQLConnectW

SQLDataSourcesW

SQLDescribeColW

SQLDriverConnectW

SQLDriversW

SQLErrorW

SQLExecDirectW

SQLForeignKeysW

SQLGetConnectAttrW

SQLGetConnectOptionW

SQLGetCursorNameW

SQLGetDescFieldW

SQLGetDescRecW

SQLGetDiagFieldW

SQLGetDiagRecW

SQLGetInfoW

SQLGetStmtAttrW

SQLGetTypeInfoW

SQLNativeSqlW

SQLPrepareW

SQLPrimaryKeysW

SQLProcedureColumnsW

SQLProceduresW

SQLSetConnectAttrW

SQLSetConnectOptionW

SQLSetCursorNameW

SQLSetDescFieldW

SQLSetStmtAttrW

SQLSpecialColumnsW

SQLStatisticsW

SQLTablePrivilegesW

SQLTablesW

SAP DB ODBC Driver Under Microsoft Windows

Installation Under Microsoft Windows [Page 21]

Creating Data Sources Under Microsoft Windows [Page 23]

Installation Under Microsoft Windows

Use

You have various options for installing the SAP DB ODBC driver under Microsoft Windows:

- As part of the SAP DB server package
- As a separate software package, such as for purely client installations

These options differ only in the default value of the system for storing the installed software.

If you install the SAP DB ODBC driver as part of the server package, the system default value for the installation path up to database version 7.3 is the `<dependent_data_path>`, as of database version 7.4, the `<independent_data_path>`.

If you install the SAP DB ODBC driver as a separate software package, the default value for the system is the installation path `C:\Program Files\SAP DB\ODBC7x`, where `x` stands for the particular version of the SAP DB ODBC driver.

As of version 7.4, the ODBC installation package is included not only in the server package, but also in the Web Tools package and in the DB Analyzer package. The combination of the software packages corresponds to different installation profiles. You can install the entire software package; that is, including ODBC software, or select only the ODBC installation package from the software package.

See also: *Installation Guidelines: SAP DB 7.4*, Installation Profile section

Procedure

Check whether a SAP DB Server is already installed on your server. If this is the case, the SAP DB ODBC driver is also already installed, and no further steps are necessary. Otherwise, proceed as follows:

Open the InstallShield Wizard for the SAP DB ODBC driver by double clicking the file `odbc7x-setup.exe`.

The default value for the installation path is `C:\Program Files\SAPDB\ODBC7x`. You can change this installation path.

Follow the instructions in the Wizard.

Result

Information about all installed ODBC drivers is stored by the system in the file `HKEY_LOCAL_MACHINE/SOFTWARE/ODBC/ODBCINST.INI`. A driver-specific key is also created for each ODBC driver, and is also stored in this file.

The following entry is stored in the registry for the SAP DB ODBC driver during installation:

```
HKEY_LOCAL_MACHINE
     SOFTWARE\ODBC\ODBCINST.INI
          SAP DB 7.4
```

The key `SAP DB 7.4` is created.

The names of the ODBC driver DLLS and the name of the setup DLL are stored under the driver-specific key. The ODBC Administrator tool uses this information to load the corresponding DLL when creating new data sources.

Under the key SAP DB 7.4, there is the following entry:

```
HKEY_CURRENT_USER
     SOFTWARE\ODBC\ODBCINST.INI\SAP DB 7.4
        Driver=C:\SAP DB\pgm\sqlod32.dll
        Setup=C:\SAP DB\pgm\sqlsp32.dll
```

Creating Data Sources Under Microsoft Windows

Use

You can create data sources using the ODBC Administrator tool.

The specifications for the data sources are stored by the system in the file `HKEY_CURRENT_USER/Software/ODBC/ODBC.INI`.

Procedure

1. Choose *Start* → *Settings* → *Control Panel* → *Administrative Tools* →*Data Sources (ODBC)*. This opens the ODBC Data Source Administrator.
2. Choose *Add*.
3. From the displayed list of the installed ODBC drivers, choose the entry `SAP DB` and confirm your choice by choosing *Finish*.
4. Make the following entries:

Data Source Name	Name of the data source. The data source is identified using this name. Use this name to simplify logging on to the database server. The name of the data source may not contain any special characters.
Description	Description of the data source
Database Name	Name of the database instance
Database Server	Name of the database server.If the database instance and the application are on the same server, you do not need to fill out this field.

1. If you want to specify options for the new data source, choose *Settings*.
   Proceed with the entry of Options for the Data Source [Page 23].
   If you do not want to specify any options, choose *OK*.
   Your entries become effective when the next ODBC connection is created.

Options for Data Sources (Microsoft Windows)

You can specify various options to influence the functioning of the SAP DB ODBC driver.
These options become active the next time that you log on to the database server.

The following options are possible:

Option	Remarks
SQL Mode [Page 24]	Select the database-specific SQL mode
Isolation Level [Page 24]	Overrides the default value of the SAP DB ODBC driver for the lock operation type (the default value is `Committed`).
Trace On [Page 25]	Activates or deactivates the driver log. If you have set this option, the SAP DB ODBC driver creates a log of the database session. All SQL statements that the application transfers to the database system are recorded in the log. A large number of file operations are performed if you activate logging. This affects the execution speed of the application. You should therefore deactivate logging again after your analysis. This driver logging is not identical to the Driver Manager log,

	which you can activate and deactivate in the ODBC Data Source Administrator on the *Tracing* tab page.

SQL Mode (Microsoft Windows)

Use

If you specify the option *SQL Mode*, you can use the SAP DB ODBC driver in a different SQL mode. This means that the driver can process not only the ODBC and SAP DB-specific SQL syntax, but also the ORACLE or DB2 syntax, for example.

Specifying this option can simplify the customization of SQL statements during the migration from ODBC applications from other database systems.

For a description of the SAP DB-specific SQL mode `INTERNAL`, see the *Reference Manual: SAP DB 7.4*, SQL Mode section.

Procedure

1. Start the ODBC Data Source Administrator by choosing *Start* → *Settings* → *Control Panel* → *Administrative Tools* → *Data Sources*.
2. Choose *Add*.
3. Choose the SAP DB ODBC driver and then *Finish*.
4. Specify the name of the data source, the name of the database instance, and the name of the database server. Choose *Options*.
5. In the `SQL Mode` field, choose the desired SQL mode.
6. Confirm the windows until the ODBC Data Source Administrator closes.

Result

The next time that you log on to the database server using this data source, the SAP DB ODBC driver functions in the selected SQL mode.

Isolation Level (Microsoft Windows)

Use

SAP DB allows concurrent transactions on database objects. To do this, it uses a lock concept that creates locks on rows and tables to contain individual transactions. The lock operation type plays an important role in this. This is determined by defining an isolation level.

Use the *Isolation Level* option if you want to change the system default value for the lock operation type (Committed). The value defined in this way then applies for all connections of the data source.

The set lock operation type can be overridden by the application using the function *SQLSetConnectAttr* and queried with *SQLGetConnectAttr*.

Overview of the possible lock operation types

	Dirty Read	Non-repeatable Read	Phantom Read
`Uncommitted`	x	x	x
`Committed`		x	x
`Repeatable`			x
`Serializable`			

The lock operation type `Uncommitted` corresponds to the weakest isolation level (0), while `Serializable` corresponds to the strongest isolation level (4).

Procedure

1. Start the ODBC Data Source Administrator by choosing *Start* → *Settings* → *Control Panel* → *Administrative Tools* → *Data Sources*.
2. Choose *Add*.
3. Choose the SAP DB ODBC driver and then *Finish*.
4. Specify the name of the data source, the name of the database instance, and the name of the database server. Choose *Options*.
5. In the `Isolation Level` field, select the desired lock operation type.
6. Confirm the windows until the ODBC Data Source Administrator closes.

Result

The next time that you log on to the database server using this data source, the SAP DB ODBC driver functions with the selected lock operation type.

Logging SQL Statements (Microsoft Windows)

Use

Select the *Trace On* option for a data source, if you want to log the SQL statements that are transferred from the application to the database system. You can set this option for existing data sources or when creating a new data source.

The following data is recorded:

- SQL statement
- Execution time
- Start and end of a session
- CONNECT parameters
- Input and output parameters

Procedure

1. Start the ODBC Data Source Administrator by choosing *Start* → *Settings* → *Control Panel* → *Administrative Tools* → *Data Sources*.
2. Choose *Add*.
3. Choose the SAP DB ODBC driver and then *Finish*.

4. Specify the name of the data source, the name of the database instance, and the name of the database server, and choose *Options*.
5. Select the `Trace On` field or choose a file using *Browse* to which the log should be written.
6. Confirm the windows until the ODBC Data Source Administrator closes.

Result

The next time that you log on to the database server using this data source, the system logs all SQL statements that are transferred by the application to the database system.

Installed Files (Microsoft Windows 95 and following)

Overview of the Files created during installation and their functions

The files are specified relative to the respective installation directory.
(See also: Installation Under Microsoft Windows [Page 21])

Filename	Description
`sql*.dll`	Libraries of the SAP DB ODBC driver
`odbcreg.exe`	Program for entering the driver information in the registry

SAP DB ODBC Driver Under UNIX/Linux

Installation Under UNIX/Linux [Page 26]

Creating Data Sources Under UNIX/Linux [Page 28]

Installation Under UNIX/Linux

You have various options for installing the SAP DB ODBC driver:

- As part of the SAP DB server package

  The software for the SAP DB ODBC driver is stored under the `<dependent_data_path>` up to database version 7.3 and under the `<independent_data_path>` as of database version 7.4.

- As part of the rpm package for the call interfaces (only for Linux)

  The installation program creates the directory `/opt/sapdb/interfaces/odbc` and stores the SAP DB ODBC driver files there.

- Using a separate tgz package (ODBC Patch)

  You can decide the path under which the tgz package unpacks and the software is installed yourself.

As of version 7.4, the ODBC installation package is included not only in the server package, but also in the Web Tools package. The combination of the software packages corresponds to different installation profiles. You can install the entire software package; that is, including ODBC software, or select only the ODBC installation package from the software package.

See also: *Installation Guidelines: SAP DB 7.4*, Installation Profile section

You can use the SAP DB ODBC driver under UNIX/Linux both with and without a Driver Manager (Integrating the SAP DB ODBC Driver [Page 27]).

Integrating the SAP DB ODBC Driver

Use

You can use the SAP DB ODBC driver under UNIX/Linux both with and without a Driver Manager.

By entering link commands, you connect the static library `libsqlod.a` or the dynamic library `libsqlod.so` with the application. Ensure with dynamic links that the dynamic library `libsqlod.so` is found by the system at execution time (see, for example, Idconfig, LD_LIBRARY_PATH).

Procedure

Use with a Driver Manager

Using a Driver Manager, such as iODBC or unixODBC simplifies the exchange of ODBC drivers and allows the concurrent use of different ODBC drivers. The relevant documentation describes how you integrate an ODBC driver into the Driver Manager.

Use without a Driver Manager

If you are not using a Driver Manager, you must use an editor to enter the data for the SAP DB ODBC driver manually, either in the file ~/`.odbc.ini` or the file `/var/spool/sql/ini/odbc.ini`.

Result

The system first uses the valid `.odbc.ini` file for the respective user from his or her home directory during the assignment of data sources. If no suitable data source is found there, the system evaluates the file `/var/spool/sql/ini/odbc.ini`.

Using the ODBC Driver without a Driver Manager

Compiler command:

Compile the ODBC application `myapp.c` with appropriate ODBC header files

```
cc -c myapp.c -I<ODBC_include>
```

Link command for static library:

Static linking of the application `myapp`

```
cc -o myapp myapp.o <ODBC_lib>/libsqlod.a
```

Link command for dynamic library:

Dynamic linking of the application `myapp`

```
cc -o myapp myapp.o -L<ODBC_lib> -lsqlod
```

Creating and Changing Data Sources Under UNIX/Linux

Use

Under the UNIX and Linux operating systems, data sources are stored system-specifically in the directory `/var/spool/sql/config/` in the file `odbc.ini` or user-specifically in the file ~/`.odbc.ini`. To create data sources, edit the desired file appropriately.

Procedure

1. Open an editor.
2. Open the desired file.
   Specify the name of the data source, the database server, and the database instance in accordance with the following structure:
   You can also specify Options for Data Sources [Page 28], if you want to change the system default values.

   ```
   [<data_source_name>]
   ServerDB=<database_name>
   ServerNode=<name_of_database_server>
   SQLMode=<INTERNAL|DB2|ANSI|ORACLE>
   IsolationLevel=<Uncommitted|Committed|Repeatable|Serializable>
   TraceFileName=<trace_file_path_and_name>
   ```

   If you want to specify multiple data sources, repeat the group of specifications above in accordance with the number of data sources.
3. Save your entries.
   Your entries become effective when the next ODBC connection is created.

Options for Data Sources (UNIX/Linux)

You can specify various options to influence the functioning of the SAP DB ODBC driver. These options become active the next time that you log on to the database server.

The following options are possible:

Option	Remark
SQLMode [Page 29]	Select the database-specific SQL mode (default value is `INTERNAL`)
IsolationLevel [Page 29]	Overrides the default value of the SAP DB ODBC driver for the isolation level (the default value is `Committed`).
TraceFileName [Page 30]	Activates or deactivates the driver log for logging the SQL statements (Default value: No log is written). If you have set this option, the SAP DB ODBC driver creates a log of the database session. All SQL statements that the application sends to

	the database system are recorded in the log. A large number of file operations are performed if you activate logging. This affects the execution speed of the application. You should therefore deactivate logging again after your analysis.

SQL Mode (UNIX/Linux)

Use

By changing the option `SQLMode`, you can use the SAP DB ODBC driver in a different SQL mode. This means that the driver can process not only the ODBC and SAP DB-specific SQL syntax, but also the ORACLE or DB2 syntax, for example.

The system default value is `INTERNAL` (see also: *Reference Manual: SAP DB 7.4*, SQL Mode section).

If you migrate ODBC applications from other database systems, specifying this option can simplify the customization of the SQL statements.

Syntax

```
SQLMode =
  INTERNAL
| DB2
| ANSI
| ORACLE
```

For a description of the SAP DB-specific SQL mode `INTERNAL`, see the *Reference Manual: SAP DB 7.4*, SQL Mode section.

Procedure

To change the `SQLMode` option, follow the procedure described under Creating and Changing Data Sources Under UNIX/Linux [Page 28].

The value defined in this way then applies for all connections of the data source.

Isolation Level (UNIX/Linux)

Use

SAP DB allows concurrent transactions on database objects. To do this, it uses a lock concept that creates locks on rows and tables to contain individual transactions. The lock operation type plays an important role in this. This is determined by defining an isolation level.

The system default value for the lock operation type is `Committed`.

The set lock operation type can be overridden by the application using the function *SQLSetConnectAttr* and queried with *SQLGetConnectAttr*.

Syntax

```
IsolationLevel =
  Uncommitted
| Committed
| Repeatable
| Serializable
```

Overview of the possible lock operation types

	Dirty Read	Non-repeatable Read	Phantom Read
Uncommitted	x	x	x
Committed		x	x
Repeatable			x
Serializable			

The lock operation type `Uncommitted` corresponds to the weakest isolation level (0), while `Serializable` corresponds to the strongest isolation level (4).

Procedure

To change the `IsolationLevel` option, follow the procedure described under Creating and Changing Data Sources Under UNIX/Linux [Page 28].

The value defined in this way then applies for all connections of the data source.

Logging SQL Statements (UNIX/Linux)

Use

The SQL statements transferred by the application to the database instance can be logged in a file. The following data is recorded:

- SQL statement
- Execution time
- Start and end of a session
- CONNECT parameters
- Input and output parameters

Only one application can write to each log file.

If this option is not set, no log is written (system default value).

Syntax

```
TraceFileName = <file_name>

<file_name> ::= [<path_name>/]file_identifier
```

Procedure

To change the *`TraceFileName`* option, follow the procedure described under Creating and Changing Data Sources Under UNIX/Linux [Page 28].

Installed Files (UNIX/Linux)

Overview of the Files created during installation and their functions

The files are specified relative to the respective installation directory.
(See also: Installation Under UNIX/Linux [Page 26])

Filename	Description
`libsqlod.a, libsqlod.so`	SAP DB ODBC driver library
`incl/sql.h`	ODBC driver header file (Core API)
`incl/sqlext.h`	ODBC driver header file (Level 1 API and Level 2 API)
`incl/sqltypes.h`	ODBC driver header file (Data types)
`incl/WINDOWS.H`	Header file for the UNIX/Linux platforms
`demo/eng/ODBC/sqlexamp.c`	C example program for an ODBC application
`demo/eng/ODBC/sqladhoc.c`	C example program for an ODBC application
`demo/eng/ODBC/Makefile`	Make file for generating a demo program