

Database  
Management  
System

# LINTER®

Version 5.9

ODBC

Relational Expert Systems

---



# Table of Contents

<b>Introduction</b> .....	<b>4</b>
<b>Driver Functions</b> .....	<b>5</b>
<b>ODBC Architecture</b> .....	<b>6</b>
Application .....	6
Driver manager.....	6
Driver.....	6
Data source .....	7
<b>Input and Output Data</b> .....	<b>8</b>
SQLAllocHandle .....	8
SQLBindCol.....	8
SQLBindParameter .....	8
SQLBrowseConnect.....	9
SQLBulkOperations.....	9
SQLCancel .....	9
SQLCloseCursor .....	10
SQLColAttribute.....	10
SQLColumnPrivileges .....	10
SQLColumns .....	10
SQLConnect.....	11
SQLCopyDesc.....	11
SQLDescribeCol.....	11
SQLDescribeParam .....	11
SQLDisconnect.....	12
SQLDriverConnect .....	12
SQLEndTran.....	13
SQLError.....	13
SQLExecDirect.....	13
SQLExecute .....	13
SQLFetch.....	14
SQLFetchScroll .....	14
SQLForeignKeys .....	14
SQLFreeHandle .....	15

---

SQLGetConnectAttr.....	15
SQLGetCursorName .....	15
SQLGetData .....	16
SQLGetDescField.....	16
SQLGetDescRec.....	16
SQLGetDiagField.....	16
SQLGetDiagRec.....	16
SQLGetEnvAttr.....	17
SQLGetFunctions .....	17
SQLGetInfo .....	17
SQLGetStmtAttr.....	17
SQLGetTypeInfo.....	18
SQLMoreResults .....	18
SQLNativeSql .....	18
SQLNumParams .....	19
SQLNumResultCols .....	19
SQLParamData .....	19
SQLPrepare.....	19
SQLProcedureColumns .....	20
SQL Procedures .....	20
SQLPutData.....	20
SQLRowCount.....	20
SQLSetConnectAttr .....	21
SQLSetCursorName.....	22
SQLSetDescField .....	22
SQLSetDescRec.....	22
SQLSetPos .....	22
SQLSetScrollOptions.....	23
SQLSetStmtAttr .....	23
SQLSpecialColumns .....	24
SQLStatistics .....	24
SQLTablePrivileges .....	24
SQLTables .....	25
<b>Installation Instructions.....</b>	<b>26</b>
Driver Manager Installation.....	26
Data Source Configuration .....	26

## Introduction

This document describes Linter's ODBC driver. The driver generally conforms to Microsoft's ODBC 3.0 standard for Windows 95/NT. For each function, there is a description of Linter's compatibility with Microsoft's standard specification. Linter's implementation specifics are given for each function.

Additionally, installation and driver data source configuration are described.

The Open Data Base Connectivity (ODBC) interface provides relational database control system access for applications using SQL as the data processing language.

The ODBC interface provides maximum application portability from one DBMS to another, minimizing re-coding. This is achieved through the implementation of two components in the interface: the ODBC driver manager and the ODBC driver.

The ODBC driver manager supplies a unified interface for all applications that access the database. It is achieved by means of a full, complex set of functions through which the applications can perform all the tasks related to a DBMS - data addition and deletion queries, stored procedure execution, and extraction of information from a data source. The ODBC administrator interface remains unchanged, regardless of the DBMS the application is accessing.

The second component, the ODBC driver, on the other hand, depends entirely on the DBMS it is working in. The driver manager converts the application queries into the internal DBMS language. Every DBMS supporting the ODBC technology must provide an ODBC driver for application developers.

## Driver Functions

The Linter's ODBC driver provides the following functions:

- Application access to DBMS Linter;
- SQL query execution based on the X/Open and SQL Access Group (SAG) SQL CAE 1992 specifications;
- Receiving SQL query processing results;
- Returning query processing termination codes;
- Support for standard data types;
- Formation of static and dynamic SQL expressions; and
- Sending and receiving data values in the format specified by the application.

## ODBC Architecture

The ODBC architecture consists of four components:

Application	Performs application tasks, calls the ODBC functions, sends SQL expressions, and receives results.
Driver manager	Loads drivers on application request.
Driver	Processes ODBC function calls according to DBMS specifics and returns the results to the application. If necessary, the driver modifies queries according to DBMS criteria.
Data Source	Contains user defined control information. The data source information is used by the ODBC interface to access a specific DBMS using the appropriate operating system and network platform tools.

## Application

An application using the ODBC interface performs the following functions:

- Requests a connection (session) with a data source;
- Sends SQL queries to the data source;
- Defines the storage area and format for SQL query results;
- Queries data;
- Handles errors;
- If necessary, alerts the user to the presence of errors;
- Performs action commit or rollback in transaction mode;
- Closes the connection to the data source.

## Driver manager

Microsoft's driver manager is a dynamic link library (DLL). The manager's main function is to load drivers. Additionally, it performs the following functions:

- Insures compatibility between the data source name and the DLL driver, using the ODBC.INI file or the system registry;
- Processes several ODBC initialization calls;
- Provides access to all ODBC functions in every driver;
- Performs parameter and call order control for ODBC functions.

## Driver

The ODBC driver is also a DLL. It implements ODBC functions and accesses the data source.

The driver manager loads the DBMS Linter driver (or any other ODBC driver) when an application calls the `SQLBrowseConnect`, `SQLConnect` or `SQLDriverConnect` functions.

On application call, the driver performs the following functions:

- Creates a connection to the data source;
- Sends queries to the data source;
- Converts data into different formats, if necessary;
- Returns query results to the application;
- Converts error codes into standard form and returns them to the application;
- Defines and manipulates cursors. This operation is hidden from the application until the application explicitly requests access based on a cursor name.

## **Data source**

A data source is a definition uniting the DBMS, the operating system, and the network into a single concept.

For example, the application may have access to:

- DBMS Linter launched under Windows NT and running on a Novell NetWare network;
- Local Xbase file. In this case the network and remote operating system do not participate in communication.

In both instances, the location and path to the necessary data are described in the data source.

# Input and Output Data

A complete description of the ODBC interface is found in ODBC 3.0. Programmers Reference. MS Windows and Windows NT Standard. This section describes the ODBC interface in relation to DBMS Linter.

## SQLAllocHandle

### Use

Allocates a region in memory for connection, environment, cursor, or descriptor handles.

### Details

The function's argument values correspond to ODBC 3.0 specs.

SQLAllocEnv, SQLAllocConnect, and SQLAllocStmt function calls are allowed.

For Win32, an explicit DLL load is performed when calling the SQLAllocHandle function for the environment handle SQLAllocEnv. If the libraries can not be found, the driver will inform the user of this fact via a message window.

## SQLBindCol

### Use

Binds the buffer and data type with a response column, including:

- The buffer, which will hold the response column value;
- Buffer size;
- The real size of data located in the buffer;
- Conversion data type.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLBindParameter

### Use

Binds the buffer to the SQL expression parameter.

Argument values:

<u>Argument</u>	<u>Legal Values</u>	<u>Illegal Values</u>
Input	SQL_PARAM_IN PUT	
Output	SQL_PARAM_OUT PUT	
Type	SQL_PARAM_INPUT_OUTPUT	
Parameter type	SQL_CHAR	SQL BIGINT

<u>Argument</u>	<u>Legal Values</u>	<u>Illegal Values</u>
	SQL_DECIMAL	SQL_BINARY
	SQL_DOUBLE	SQL_BIT
	SQL_FLOAT	SQL_DATE
	SQL_INTEGER	SQL_LONGVARCHAR
	SQL_LONGVARBINARY	SQL_TIME
	SQL_NUMERIC	SQL_TINYINT
	SQL_REAL	SQL_VARBINARY
	SQL_SMALLINT	SQL_VARCHAR
	SQL_DATA_AT_EXEC and SQL_LEN_DATA_AT_EXEC (both legal for BLOB data type)	SQL_DATA_AT_EXEC and SQL_LEN_DATA_AT_EXEC (both illegal for non-BLOB data types)
	SQL_TIMESTAMP	
	SQL_NTS	
	SQL_NULL_DATA	

The driver does not support parameter arrays.

## SQLBrowseConnect

### Use

Provides the means to list the necessary attributes and values or establishing a connection to the data source.

### Details

Function not implemented.

## SQLBulkOperations

### Use

Allows operation with blocks of data: packet addition of data or edit/delete operations on previously selected and marked entries).

### Details

Function not implemented.

## SQLCancel

### Use

Interrupts expression evaluation.

**Details**

Function not implemented.

## SQLCloseCursor

**Use**

Closes a previously opened cursor and cancels the data selection process.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

The SQLCloseCursor function call is analogous to SQLFreeStmt call with an SQLCLOSE parameter.

## SQLColAttribute

**Use**

Returns a column description (except for bookmark columns). The description is returned as an alphanumeric string with a 32-bit or 16-bit value.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLColumnPrivileges

**Use**

Returns a list of columns and their privileges for the selected table.

**Details**

Function not implemented.

## SQLColumns

**Use**

Returns a column list for the selected table.

Specific arguments:

<u>Argument</u>	<u>Value</u>
CatalogName	Ignored
NameLength1	Ignored

Allowed values for other arguments correspond to the ODBC 3.0 spec.

Specific returned values:

<u>Column</u>	<u>Value</u>
TABLECAT	NULL

Allowed values for other columns correspond to the ODBC 3.0 spec.

## SQLConnect

### Use

Creates a connection to the data source.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLCopyDesc

### Use

Copies data from one descriptor into another.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLDescribeCol

### Use

Returns a result column description (name, type, length, precision, ability to receive a NULL value).

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLDescribeParam

### Use

Returns a translated SQL expression parameter description.

Argument values:

<u>Argument</u>	<u>Legal Values</u>	<u>Illegal Values</u>
DataTypePtr	SQL CHAR	SQL BIGINT
	SQL DECIMAL	SQL BINARY
	SQL DOUBLE	SQL BIT
	SQL FLOAT	SQLDATE
	SQL INTEGER	SQL_LONGVARCHAR

<u>Argument</u>	<u>Legal Values</u>	<u>Illegal Values</u>
	SQL LONGVARBINARY	SQL TIME
	SQL NUMERIC	SQL TINYINT
	SQL REAL	SQL VARBINARY
	SQL SMALLINT	SQL VARCHAR
	SQL TIMESTAMP	

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLDisconnect

**Use**

Closes the connection bound to the selected connection identifier.

**Details**

The function's argument values correspond to ODBC 3.0 spec.

## SQLDriverConnect

**Use**

Loads the driver and creates a connection to the data source using additional information and displaying an additional parameter dialogue window.

**Details**

Supports connection parameter transfer through:

- Connection string containing the data source name, one or several user identifiers, one or several passwords, and other information requested by the data source;
- Dialogue window.

Argument values:

<u>Argument</u>	<u>Allowed values</u>
SzConnStrIn	Format string: DSN=<dsn>;UID=<uid>;PWD=<pwd>; SERVER=<server>;TXN=<txn>; where: <ul style="list-style-type: none"> <li>• dsn – data source name;</li> <li>• uid – user name;</li> <li>• pwd – user password;</li> <li>• server – remote linter server;</li> <li>• txn – transaction processing mode (Optimistic, Pessimistic, Autocommit).</li> </ul>

Allowed values for other arguments correspond to the ODBC 3.0 spec.

## SQLEndTran

### Use

Calls the commit or rollback of all the active operations in all the cursors bound to the selected connection identifier. The function may also call commit or rollback on all connections bound to the selected environment identifier.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLError

### Use

Returns an error or status information.

Argument values:

<u>Argument</u>	<u>Required Value</u>
Henv	One of the arguments must be non-NULL
Hdbc	
Hstmt	

Allowed values for other arguments correspond to the ODBC 3.0 spec.

## SQLExecDirect

### Use

Executes the defined SQL statement using the current buffer values bound to parameters.

Argument values:

<u>Argument</u>	<u>Allowed values</u>
StatementText	Linter supported SQL statement

Allowed values for other arguments correspond to the ODBC 3.0 spec.

## SQLExecute

### Use

Evaluates a previously translated SQL-statement using the current buffer values bound to parameters.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

**SQLFetch****Use**

Moves the current position in the reply onto the next line. The driver fills buffers bound to reply columns with data.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

**SQLFetchScroll****Use**

Moves the cursor to the selected entry in the row set and returns the values of all bound columns. Row sets may be specified relative to the current position or absolutely from the start of the row set.

Argument values:

<u>Argument</u>	<u>Legal Values</u>	<u>Illegal Values</u>
FetchOrientation	SQL_FETCH_NEXT SQL_FETCH_FIRST SQL_FETCH_LAST SQL_FETCH_PRIOR SQL_FETCH_ABSOLUTE SQL_FETCH_RELATIVE	SQL_FETCH_BOOKMARK

Allowed values for other arguments correspond to the ODBC 3.0.

**SQLForeignKeys****Use**

Returns:

- A listing of external keys for the selected table (table columns referencing primary keys of other tables);
- A listing of external keys referencing primary keys in the selected table.

The driver returns the listing as an array of strings.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

Argument values:

<u>Argument</u>	<u>Value</u>
CatalogName	Ignored
NameLength1	Ignored

## SQLFreeHandle

### Use

Destroys the connection, environment, statement, or descriptor handle and frees the allocated memory.

### Details

The function's argument values correspond to ODBC 3.0 specs.

The SQLFreeEnv, SQLFreeConnect, and SQLFreeStmt functions may be used.

## SQLGetConnectAttr

### Use

Returns the current connection parameters.

Argument values:

<u>Argument</u>	<u>Legal Values</u>	<u>Illegal Values</u>
Attribute	SQL_ACCESS_MODE	SQL_ODBC_CURSORS
	SQL_AUTOCOMMIT	SQL_OPT_TRACE
	SQL_LOGIN_TIMEOUT	SQL_OPT_TRACEFILE
	SQL_CURRENT_QUALIFIER (ODBC 3.0)	SQL_PACKET_SIZE
	SQL_ATTR_ASYNC_ENABLE	SQL_QUIET_MODE
	SQL_ATTR_AUTO_IPD	SQL_TRANSLATE_DLL
	SQL_ATTR_CONNECTION_TIMEOUT	SQL_TRANSLATE_OPTION
	SQL_ATTR_METADATA_ID	SQL_TXN_ISOLATION

## SQLGetCursorName

### Use

Returns the cursor name bound to selected operator.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetData

**Use**

Returns the data from one column of result row set.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetDescField

**Use**

Returns the value or properties of the selected field of the descriptor entry.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetDescRec

**Use**

Returns the values or properties of the selected field set of the descriptor entry. The properties include the field name, field data type, NULL value allowance, and scale and precision of numeric values.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetDiagField

**Use**

Returns the current value of the selected field of the diagnostic bock (bound to selected descriptor) which contains the error description, warnings, or status information.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetDiagRec

**Use**

Returns the current values of the field selection set of the diagnostic bock (bound to selected descriptor) which contain error descriptions, warnings, or status information. Unlike SQLGetDiagField, which returns only one diagnostic field, SQLGetDiagRec returns several diagnostic bock fields including termination codes, internal error codes, and diagnostic message text.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetEnvAttr

**Use**

Returns the current environment descriptor parameters.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetFunctions

**Use**

Returns information on ODBC functions that the driver supports.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetInfo

**Use**

Returns information on the driver and data source that are bound to the connection identifier.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

## SQLGetStmtAttr

**Use**

Returns the current parameters of the statement descriptor.

Argument values:

<u>Argument</u>	<u>Illegal values</u>
Attribute	SQL_GET_BOOKMARK
	SQL_ATTR_FETCH_BOOKMARK_PTR

Allowed values for other arguments correspond to the ODBC 3.0 spec.

## SQLGetTypeInfo

### Use

Returns information on data types supported by the data source. The driver returns the information in the form of an array of strings.

Argument values:

<u>Argument</u>	<u>Legal Values</u>	<u>Illegal Values</u>
DataType	SQL_CHAR	SQL_BIGINT
	SQL_DECIMAL	SQL_BINARY
	SQL_DOUBLE	SQL_BIT
	SQL_FLOAT	SQL_DATE
	SQL_INTEGER	SQL_LONGVARCHAR
	SQL_LONGVARIABLE	SQL_TIME
	SQL_NUMERIC	SQL_TINYINT
	SQL_REAL	SQL_VARBINARY
	SQL_SMALLINT	SQL_VARCHAR
	SQL_TIMESTAMP	
	SQL_ALL TYPES	

Allowed values for other arguments correspond to the ODBC 3.0 spec.

## SQLMoreResults

### Use

Determines whether accessible data is present in a statement after the SELECT, UPDATE, INSERT or DELETE operations, and initializes the process of receiving results.

### Details

Function not implemented.

## SQLNativeSql

### Use

Returns the SQL expression as modified by the driver. The SQL expression is not executed.

### Details

The function's argument values correspond to ODBC 3.0 specs. In this version of the driver, only stored procedure calls of the type `{[?]=}call procedure_name [[([parameter][],[parameter]]...)] }` escape sequence are evaluated.

## SQLNumParams

### Use

Returns the number of parameters in a translated SQL expression.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLNumResultCols

### Use

Returns the number of reply columns after translation or execution of SQL SELECT expressions.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLParamData

### Use

Is used in conjunction with SQLPutData for entering parameter values during statement execution.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLPrepare

### Use

Translates the SQL expression before execution.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLPrimaryKeys

### Use

Returns a listing of columns that compose the primary table key. The driver returns the information in the form of an array of strings.

Argument values:

<u>Argument</u>	<u>Value</u>
CatalogName	Ignored
NameLength1	Ignored

Allowed values for other arguments correspond to the ODBC 3.0 spec.

Reply values:

<u>Column</u>	<u>Value</u>
TABLE_QUALIFIER	NULL
KEY_SEQ	1
PKNAME	COLUMN_NAME

Allowed values for other columns correspond to the ODBC 3.0 spec.

## SQLProcedureColumns

### Use

Returns the listing of input and output parameters of the selected stored procedure. The driver returns the information in the form of an array of strings.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQL Procedures

### Use

Returns a listing of stored procedures. The driver returns the information as an array of strings.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLPutData

### Use

Allows the application to enter data into the parameter during statement execution. This function is usually used for sending data into a long length column (of the type `SQL_LONGBINARY` or `SQL_LONGVARCHAR`).

### Details

This function may only be used with parameters of the BLOB data type.

## SQLRowCount

### Use

Returns the number of strings evaluated by the UPDATE, INSERT or DELETE statements or by the SQLSetPos function.

**Details**

The function's argument values correspond to ODBC 3.0 specs.

**SQLSetConnectAttr****Use**

Sets the connection parameters.

Illegal argument values:

<u>Argument</u>	<u>Illegal Values</u>
Attribute	SQL_CURRENT_QUALIFIER
	SQL_ODBC_CURSORS
	SQL_OPT_TRACE
	SQL_OPT_TRACEFILE
	SQL_PACKET_SIZE
	SQL_QUIET_MODE
	SQL_TRANSLATE_DLL
	SQL_TRANSLATE_OPTION
	SQL_TXN_ISOLATION

Legal argument values:

<u>Argument</u>	<u>Legal Values</u>
Attribute	SQL_ACCESS_MODE
	SQL_MODE_READ_ONLY
	SQL_MODE_READ_WRITE
	SQL_AUTOCOMMIT
	SQL_LOGIN_TIMEOUT (ODBC 3.0 Linter specific)
	SQL_ATTR_LINTER_SET_SAVEPOINT
	SQL_ATTR_LINTER_COMMIT_SAVEPOINT
	SQL_ATTR_LINTER_ROLLBACK_SAVEPOINT

For the Attribute argument values:

- SQL\_ATTR\_LINTER\_SETSAVEPOINT,
- SQL\_ATTR\_LINTER\_COMMIT\_SAVEPOINT,
- SQL\_ATTR\_LINTER\_ROLLBACK\_SAVEPOINT

The ValuePtr argument must be of the SQLCHAR type.

Specification of non-standard values:

- `SQL_ATTR_LINTER_SET_SAVEPOINT` sets a named point for later possible partial rollbacks;
- `SQL_ATTR_LINTER_COMMIT_SAVEPOINT` savepoint changes in the DB prior to the intermediate savepoint point;
- `SQL_ATTR_LINTER_ROLLBACK_SAVEPOINT` makes a change rollback for the changes prior to the intermediate savepoint point.

The ValuePtr must contain the name of the intermediate savepoint point.

Allowed values for other arguments correspond to the ODBC 3.0 spec.

## SQLSetCursorName

### Use

Sets the name of the cursor bound to the selected statement.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLSetDescField

### Use

Sets the selected field values in the entry descriptor.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLSetDescRec

### Use

Sets the values in the selection set of fields of the entry descriptor according to the description located in the input data buffer.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLSetPos

### Use

Sets the cursor position in the reply and allows the application to re-read, update, delete or add and entry.

Argument values:

<u>Argument</u>	<u>Allowed values</u>
Attribute	SQL_REFRESH
	SQL_UPDATE
	SQL_DELETE
	SQL_ADD
	SQL_POSITION

Allowed values for other arguments correspond to the ODBC 3.0 spec.

## SQLSetScrollOptions

### Use

Sets the cursor behavior mode bound to selected statement.

### Details

Function not implemented.

## SQLSetStmtAttr

### Use

Sets the current parameters of the selected statement.

Illegal argument values:

<u>Argument</u>	<u>Illegal values</u>
Attribute	SQL_USE_BOOKMARK
	SQL_ASYNC_ENABLE=On
	SQL_CONCURRENCY
	SQL_CURSOR_TYPE
	SQL_KEYSET_SIZE
	SQL_MAX_LENGTH
	SQL_MAX_ROWS
	SQL_NOSCAN
	SQL_QUERY_TIMEOUT

Specifications for the following legal argument values are set out in section "SQLSetDescField".

<u>Argument</u>	<u>Legal values</u>
Attribute	SQL_ATTR_LINTER_SET_SAVEPOINT SQL_ATTR_LINTER_COMMIT_SAVEPOINT SQL_ATTR_LINTER_ROLLBACK_SAVEPOINT

Legal values for other arguments correspond to the ODBC 3.0 spec.

## SQLSpecialColumns

### Use

Returns the following information on columns of the selected table:

- the optimal number of columns uniquely identifying a table entry;
- columns that are changed automatically due to any change to the row values.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLStatistics

### Use

Returns statistical information on the table and indexes connected to the table. The driver returns the information in the forms of an array of strings.

### Details

The function's argument values correspond to ODBC 3.0 specs.

## SQLTablePrivileges

### Use

Returns a listing of tables and the privileges connected to every table. The driver returns the information in the form of an array of strings.

Argument values:

<u>Argument</u>	<u>Value</u>
CatalogName	Ignored
NameLength	Ignored

Legal values for other arguments correspond to the ODBC 3.0 spec.

Details of the reply: legal values for other arguments correspond to the ODBC 3.0 spec.

## SQLTables

### Use

Returns a listing of tables stored in the data source. The driver returns the information in the form of an array of strings.

Argument values:

<u>Argument</u>	<u>Value</u>
CatalogName	Ignored
NameLength1	Ignored
TableType	VIEW SYNONYM TABLE SYSTEM TABLE

Allowed values for other arguments correspond to the ODBC 3.0

# Installation Instructions

This section describes the installation of the ODBC driver manager and the Linter driver and configuration of the data source.

## Driver Manager Installation

To install the ODBC driver manager and the DBMS Linter ODBC driver itself, launch SETUP.EXE located in the distribution package of the ODBC driver. Prior to installation, the program will check for earlier installed ODBC driver components. If such components are found, the program will prompt the user concerning their future.

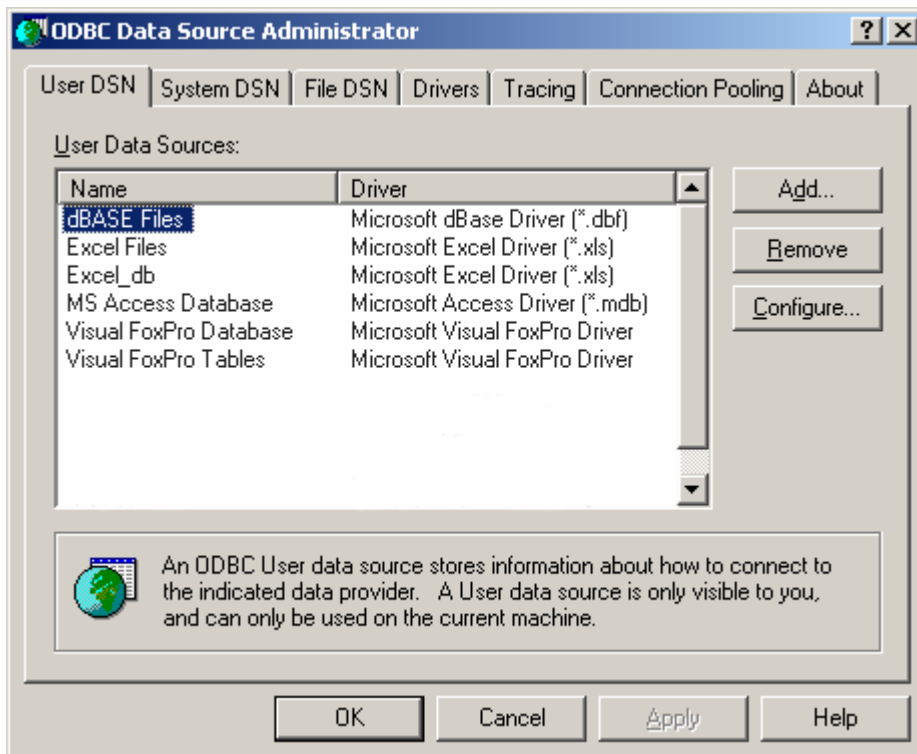
The following options are possible:

- Yes                      Replace the existing component with the one from the distribution package.
- No                        Do not replace the component.
- No For All              Do not replace any of the existing components. If this option is selected, there will be no further questions.

## Data Source Configuration

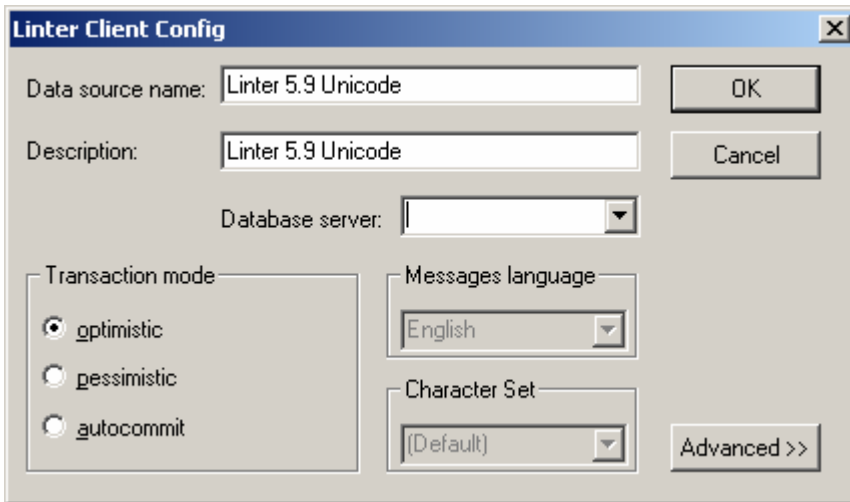
To configure the ODBC data source:

1. Open the Control Panel and double click the ODBC icon. The data source configuration window will be displayed.



Screen 1 – Data Source Administration Window

2. Click on the User DSN tab.
3. In the User Data Sources window, choose the name of the needed source (in the Name column) from the list of existing data sources. For DBMS Linter this source must show Linter n.n Driver (in the Driver column).
4. Click the Configure button. The configuration window will then be displayed.



Screen 2 – Linter ODBC Setup Window

5. Enter the required configuration parameters:

<u>Field</u>	<u>Content</u>
Data source name	Name of the data source.
Description	Data source description.
Database server	Name of the remote server if the connection will be to a server other than the default.
Default concurrency control	The default transaction method.
Messages language	Desired message language.

6. After all fields have been selected, click the Ok button to return to the Data Sources Administration window.
7. Click the Ok button there and the data source configuration is complete.