

Altibase® Application Development

# ODBC User's Manual

Release 7.1 (July 5, 2017)



Altibase® Application Development ODBC User's Manual  
Release 7.1  
Copyright © 2001~2017 Altibase Corp. All rights reserved.

This manual contains proprietary information of Altibase Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright patent and other intellectual property law. Reverse engineering of the software is prohibited. All trademarks, registered or otherwise, are the property of their respective owners.

Altibase Corp.  
10F, Daerung PostTower II,  
306, Digital-ro, Guro-gu, Seoul 08378, Korea  
Telephone: +82-2-2082-1000 Fax: 82-2-2082-1099  
Homepage: <http://www.altibase.com>

# Contents

|  |           |
|--|-----------|
| <b>Preface</b> .....   | <b>5</b>  |
| About This Manual .....  | 6         |
| Audience .....   | 6         |
| Software Environment.....  | 6         |
| Organization.....  | 6         |
| Documentation Conventions.....                                     | 6         |
| On-line Manuals.....   | 7         |
| Altibase Welcomes Your Comments.....                               | 7         |
| <br>   |           |
| <b>1 Introduction</b> .....  | <b>9</b>  |
| 1.1 Introduction .....   | 10        |
| 1.2 Data Types .....   | 11        |
| 1.3 ODBC API.....  | 12        |
| 1.3.1 ODBC Conformance Level.....                                  | 12        |
| <br>   |           |
| <b>2 Installing and Configuring the Altibase ODBC Driver</b> ..... | <b>17</b> |
| 2.1 Installing the Altibase ODBC Driver.....                       | 18        |
| 2.1.1 Unix-like Operating Systems .....                            | 18        |
| 2.2 Configuration .....  | 19        |
| 2.2.1 Unix-like Operating Systems .....                            | 19        |
| 2.2.2 Adding the DSN.....  | 19        |
| <br>   |           |
| <b>3 ODBC Programming</b> .....                                    | <b>21</b> |
| 3.1 Connection String .....  | 22        |
| 3.2 Basic Programming Examples .....                               | 23        |
| 3.2.1 Example .....  | 23        |
| 3.2.2 Execution Result .....                                       | 27        |
| 3.3 Example of Using LOB.....                                      | 28        |



# Preface

---

# About This Manual

This manual describes how to install and configure the Altibase ODBC Driver on Microsoft Windows platforms.

## Audience

This manual has been prepared for the following Altibase users:

- database managers
- performance managers
- database users
- application developers
- technical support workers

It is recommended that those reading this manual possess the following background knowledge:

- basic knowledge in the use of computers, operating systems, and operating system utilities
- experience in using relational databases and an understanding of database concepts
- computer programming experience
- experience in database server, operating system or network administration

## Software Environment

This manual has been prepared assuming that Altibase 7.1 is used as the database server.

## Organization

This manual is organized as follows:

- [Chapter1: Introduction](#)
- [Chapter2: Installing and Configuring the Altibase ODBC Driver](#)
- [Chapter3: ODBC Programming](#)

## Documentation Conventions

This section describes the conventions used in this manual. Understanding these conventions will make it easier to find information in this manual and other manuals in the series.

| Rules         | Meaning  |
|---------------|--|
| <i>Italic</i> | Indicates book title, emphasis, or placeholder variables for which particular values are supplied by user. |
| monospace     | Indicates commands within a paragraph, code in examples, etc.  |

## On-line Manuals

Online versions of our manuals (PDF or HTML) are available from Altibase's Customer Support site (<http://altibase.com/support-center/>).

## Altibase Welcomes Your Comments

Please let us know what you like or dislike about our manuals. To help us with future versions of our manuals, please tell us about any corrections or classifications that you would find useful.

Include the following information :

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

When you need an immediate assistance regarding technical issues, please contact Altibase's Customer Support site (<http://altibase.com/support-center/>).

Thank you. We appreciate your feedback and suggestions.





# 1 Introduction

---

This chapter introduces the Altibase ODBC driver.

## 1.1 Introduction

ODBC(Open Database Connectivity) is a standard programming language interface developed by Microsoft, to access a database. For an ODBC application (which is written using the ODBC API) to access a database, a separate module or driver provided by the database to be accessed is required, apart from the ODBC software (e.g., ODBC Driver Manager). Altibase provides the Altibase ODBC driver, which conforms to ODBC 3.51 specifications, for this purpose.

As the Altibase ODBC driver is built upon Altibase CLI, you can refer to the *Altibase CLI User's Manual* for further information on the internal procedures of the Altibase ODBC driver or for more specific support.

## 1.2 Data Types

For further information on the SQL data types supported by Altibase and their mapping relationships with ODBC data types, please refer to "B. Appendix: Data Types" in *Altibase CLI User's Manual*.

## 1.3 ODBC API

For further information on the ODBC API, please refer to the ODBC API Reference (<http://msdn.microsoft.com/en-us/library/windows/desktop/ms714562%28v=vs.85%29.aspx>).

### 1.3.1 ODBC Conformance Level

Based on the conformance of ODBC functions, this section describes three types of functions: those that are currently supported for Altibase ODBC, those that are soon to be supported, and those that are not supported.

The purpose of evaluating the ODBC conformance level is to provide to the application, information of which functionality of the ODBC driver to use. ODBC conformance is currently categorized into the three levels: Core, Level 1 and Level 2. To satisfy the conformance level of a function, the driver must meet all the conditions required for that level.

The conformance levels listed in the following table are in compliance with ODBC 3.x. These differ from the conformance levels of ODBC 2.x; Level 1 for ODBC 2.x is equivalent to Core for ODBC 3.x.

The Altibase ODBC driver conforms to ODBC 3.51 specifications. The following table shows the conformance level of each ODBC function, and whether or not it is supported by the Altibase ODBC driver.

| Function Name     | Level  | Support Status | Future Support | Remarks |
|-------------------|--------|----------------|----------------|---------|
| SQLAllocHandle    | Core   | O              |                |         |
| SQLBindCol        | Core   | O              |                |         |
| SQLBindParameter  | Core   | O              |                |         |
| SQLBrowseConnect  | Level1 | X              | X              |         |
| SQLBulkOperations | Level1 | O              |                |         |
| SQLCancel         | Core   | O              |                |         |
| OSQLCloseCursor   | Core   | O              |                |         |
| SQLColAttribute   | Core   | O              |                |         |

| Function Name       | Level  | Support Status | Future Support | Remarks  |
|---------------------|--------|----------------|----------------|--|
| SQLColumnPrivileges | Level2 | X              | X              | Column privileges are not supported by Altibase. |
| SQLColumns          | Core   | O              |                |  |
| SQLConnect          | Core   | O              |                |  |
| SQLCopyDesc         | Core   | X              | O              |  |
| SQLDescribeCol      | Core   | O              |                |  |
| SQLDescribeParam    | Level2 | O              |                | Not fully supported                              |
| SQLDisconnect       | Core   | O              |                |  |
| SQLDriverConnect    | Core   | O              |                |  |
| SQLEndTran          | Core   | O              |                |  |
| SQLExecDirect       | Core   | O              |                |  |
| SQLExecute          | Core   | O              |                |  |
| SQLFetch            | Core   | O              |                |  |
| SQLFetchScroll      | Core   | O              |                |  |
| SQLForeignKeys      | Level2 | O              |                |  |
| SQLFreeHandle       | Core   | O              |                |  |
| SQLFreeStmt         | Core   | O              |                |  |
| SQLGetConnectAttr   | Core   | O              |                |  |
| SQLGetCursorName    | Core   | O              |                |  |
| SQLGetData          | Core   | O              |                |  |
| SQLGetDescField     | Core   | O              |                | ODBC 3.0   |
| SQLGetDescRec       | Core   | O              |                | ODBC 3.0   |

| Function Name       | Level  | Support Status | Future Support | Remarks  |
|---------------------|--------|----------------|----------------|----------|
| SQLGetDiagField     | Core   | O              |                | ODBC 3.0 |
| SQLGetDiagRec       | Core   | O              |                | ODBC 3.0 |
| SQLGetEnvAttr       | Core   | O              |                |          |
| SQLGetFunctions     | Core   | O              |                |          |
| SQLGetInfo          | Core   | O              |                |          |
| SQLGetStmtAttr      | Core   | O              |                |          |
| SQLGetTypeInfo      | Core   | O              |                |          |
| SQLMoreResults      | Level1 | O              |                |          |
| SQLNativeSql        | Core   | O              |                |          |
| SQLNumParams        | Core   | O              |                |          |
| SQLNumResultCols    | Core   | O              |                |          |
| SQLParamData        | Core   | O              |                |          |
| SQLPrepare          | Core   | O              |                |          |
| SQLPrimaryKeys      | Level1 | O              |                |          |
| SQLProcedureColumns | Level1 | O              |                |          |
| SQLProcedures       | Level1 | O              |                |          |
| SQLPutData          | Core   | O              |                |          |
| SQLRowCount         | Core   | O              |                |          |
| SQLSetConnectAttr   | Core   | O              |                |          |
| SQLSetCursorName    | Core   | O              |                |          |
| SQLSetDescField     | Core   | O              |                | ODBC 3.0 |
| SQLSetDescRec       | Core   | O              |                | ODBC 3.0 |

| Function Name      | Level  | Support Status | Future Support | Remarks |
|--------------------|--------|----------------|----------------|---------|
| SQLSetEnvAttr      | Core   | O              |                |         |
| SQLSetPos          | Level1 | O              |                |         |
| SQLSetStmtAttr     | Core   | O              |                |         |
| SQLSpecialColumns  | Core   | O              |                |         |
| SQLStatistics      | Core   | O              |                |         |
| SQLTablePrivileges | Level2 | O              |                |         |
| SQLTables          | Core   | O              |                |         |





# 2 Installing and Configuring the Altibase ODBC Driver

---

This chapter offers explanations on how to install and configure the Altibase ODBC driver.

## 2.1 Installing the Altibase ODBC Driver

This section describes how to install the Altibase ODBC driver.

### 2.1.1 Unix-like Operating Systems

For Unix-like operating systems, the Altibase ODBC driver is installed when the Altibase server or client package is installed. For further information on how to install the Altibase server or client package, please refer to the Installation Guide.

If you install the 64-bit package, the following 32-bit and 64-bit ODBC drivers are installed to `$ALTIBASE_HOME/lib`.

`libaltibase_odbc-64bit-ul32.so`: SQLLEN size is 32 bits

`libaltibase_odbc-64bit-ul64.so`: SQLLEN size is 64 bits

The size of the SQLLEN type is defined by 64-bit ODBC Driver Managers as 64 bits. However, as unixODBC (one of the ODBC Driver Managers available for use in Unix-like operating systems) defines the size of the SQLLEN type to 32 or 64 bits, depending on the version and compile option, Altibase provides two drivers to offer a wide range of support. If you use unixODBC, it is recommended that you select the driver that matches the SQLLEN size.

If you install the 32-bit package, the following file is installed to `$ALTIBASE_HOME/lib`.

`libaltibase_odbc.so`

The above is identical for HP, except that the filename extension is `sl`.

## 2.2 Configuration

This section describes how to configure the ODBC driver.

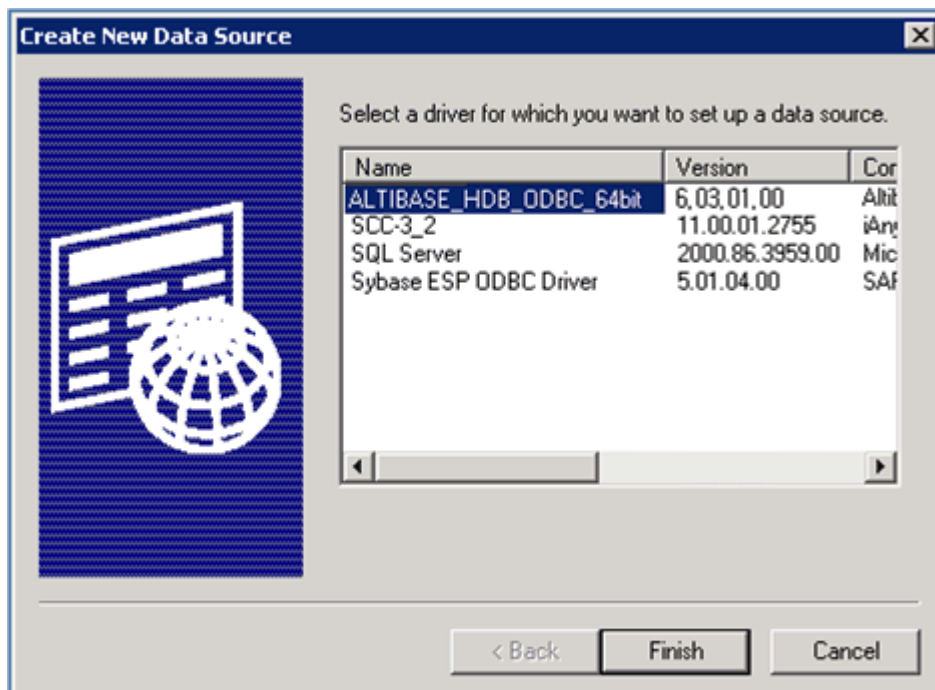
### 2.2.1 Unix-like Operating Systems

To use the ODBC driver in Unix, you must first install the ODBC Driver Manager. The unixODBC Driver Manager and iODBC Driver Manager are ODBC driver managers available for use in Unix. For further information on each driver manager, please refer to the following links.

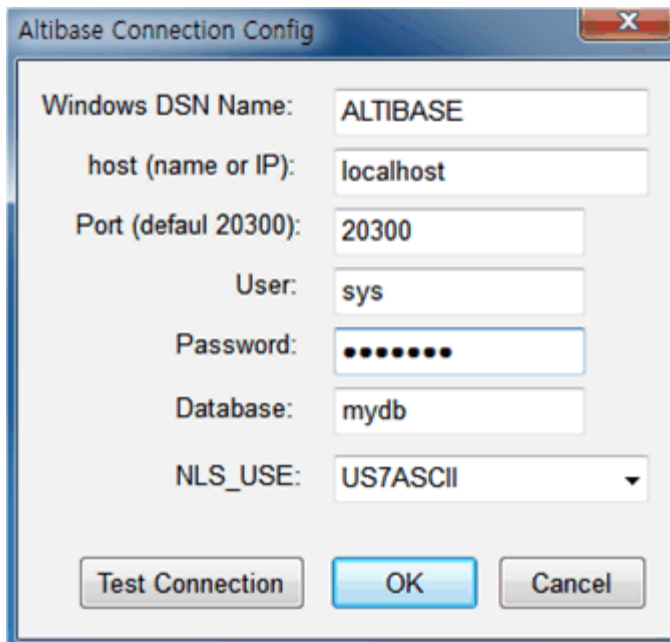
- <http://www.unixodbc.org/>
- <http://www.iodbc.org/>

### 2.2.2 Adding the DSN

For an ODBC application to obtain access to the database, the DSN of the database must be added. The following dialog box appears when you open the [Data Sources(ODBC)] panel, click the User DSN tab or System DSN tab and click "Add". Select the Altibase ODBC driver and click "Finish"



When the "Altibase Connection Config" window appears, enter the following.



- Windows DSN Name: enter the data source name.
- host (name or IP): enter the host name or IP address of the computer on which the Altibase server to be accessed is running.
- Port (default 20300): enter the listening port number of the Altibase server. You can check the PORT\_NO value from the altibase.properties file or the value of the ALTIBASE\_PORT\_NO environment variable.
- User: enter the database user name.
- Password: enter the database user password.
- Database: enter the database name.
- NLS\_USE: enter the client character set.

You can check whether or not the ODBC driver is successfully connected to the database by clicking "Test Connection". Once you click "OK", you will be able to see the data source added to the DSN tab in the name you have just entered.

# 3 ODBC Programming

---

This chapter describes with examples, how to write ODBC applications using the Altibase ODBC Driver.

## 3.1 Connection String

When writing an ODBC application using Altibase ODBC, you can use a connection string, instead of a DSN. A connection string consists of the following attributes.

| Attributes     | Description  |
|----------------|--|
| DRIVER         | The ODBC driver name. This can be checked in the ODBC data source administrator window.                |
| User           | The database user name.  |
| Password       | The database user password.  |
| Server         | The IP address of the Altibase server to be connected.   |
| PORT           | The listening port number of the Altibase server.  |
| NLS_USE        | The client character set.  |
| LongDataCompat | ON or OFF.<br>When using LOB (e.g., BLOB), it is recommended to set this to ON.<br>The default is OFF. |

The following is an example of a connection string made up of the above attributes.

```
"DRIVER=ALTIBASE_HDB_ODBC_64bit;User=SYS;Password=xxx;Server=127.0.0.1;PORT=20300;NLS_USE=US7ASCII;LongDataCompat=OFF"
```

## 3.2 Basic Programming Examples

Below are an example code of an ODBC application connecting to an Altibase server and its execution result.

### 3.2.1 Example

```
/* test_odbc.cpp */
#include <windows.h>
#include <sql.h>
#include <sqlext.h>
#include <stdio.h>
#include <stdlib.h>

#define SQL_LEN 1000
#define MSG_LEN 1024

SQLHENV     henv;
SQLHDBC     hdbc;
SQLHSTMT    hstmt;
SQLRETURN   retcode;

void execute_err(SQLHSTMT stat, char* q)
{
    printf("Error : %s\n",q);
    SQLINTEGER errNo;
    SQLSMALLINT msgLength;
    SQLTCHAR errMsg[MSG_LEN];

    if (SQL_SUCCESS == SQLError ( henv, hdbc, stat, NULL, &errNo, errMsg, MSG_LEN, &msgLength ))
    {
        printf(" Error : # %lld, %s\n", errNo, errMsg);
    }

    SQLFreeStmt(stat, SQL_DROP);
    if (SQL_ERROR == SQLDisconnect(hdbc))
    {
        printf("disconnect error\n");
    }

    SQLFreeConnect(hdbc);
    SQLFreeEnv(henv);
}
```

```

exit (1);
}

void main()
{
    char    *DSN, *DBNAME, *USERNAME, *PASSWD, *PORTNO;
    char    query[SQL_LEN], name[21];
    int     age;

    SQLCHAR constr[100];
    SQLINTEGER len;
    DSN = "ALTIBASE"; // Domain Server Name

    /* Allocate memory for the Environment */
    if(SQLAllocEnv(&henv) == SQL_ERROR)
    {
        printf("AllocEnv error!!\n");
        exit(1);
    }

    /* Allocate memory for a Connection */
    if(SQLAllocConnect(henv, &hdbc) == SQL_ERROR)
    {
        printf("AllocDbc error!!\n");
        SQLINTEGER errNo;
        SQLSMALLINT msgLength;
        SQLTCHAR errMsg[MSG_LEN];

        if (SQL_SUCCESS == SQLError ( henv, NULL, NULL, NULL, &errNo,
            errMsg, MSG_LEN, &msgLength ))
        {
            printf(" Error : # %lld, %s\n", errNo, errMsg);
        }
        exit(1);
    }

    /* Establish the Connection */
    sprintf((char*)constr,
        "DSN=%s", DSN);

    if ( SQLDriverConnect(hdbc, NULL, constr, SQL_NTS, NULL, 0, NULL,
        SQL_DRIVER_COMPLETE))
    {

```



```

printf("DBNAME = %s\n", DBNAME);
printf("USERNAME = %s\n", USERNAME);
printf("Connection error!!\n");
SQLINTEGER errNo;
SQLSMALLINT msgLength;
SQLTCHAR errMsg[MSG_LEN];

if (SQL_SUCCESS == SQLError ( henv, hdbc, NULL, NULL, &errNo,
errMsg, MSG_LEN, &msgLength ))
{
    printf(" Error : # %lld, %s\n", errNo, errMsg);
}

    SQLFreeConnect(hdbc);
    SQLFreeEnv(henv);
    exit(1);
}
printf("connected...\n");

/* Allocate memory for the statement */
if ( SQLAllocStmt(hdbc, &hstmt) == SQL_ERROR )
{
    printf("AllocStmt error!!\n");
    SQLDisconnect(hdbc);
    SQLFreeConnect(hdbc);
    SQLFreeEnv(henv);
    exit(1);
}

/* Execute the query */
sprintf(query,"DROP TABLE TEST001");
SQLExecDirect(hstmt,(SQLTCHAR*)query, SQL_NTS);

sprintf(query,"CREATE TABLE TEST001 ( name varchar(20), age number(3) )");
if (SQL_ERROR == SQLExecDirect(hstmt,(SQLTCHAR*)query, SQL_NTS))
{
    execute_err(hstmt, query);
}

/* Prepare the statement and bind the variable*/
sprintf(query,"INSERT INTO TEST001 VALUES( ?, ? )");
if (SQL_ERROR == SQLPrepare(hstmt, (SQLTCHAR*)query, SQL_NTS))
{
    execute_err(hstmt, query);
}

```

```

    }

    if (SQL_ERROR == SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT,
SQL_C_CHAR, SQL_CHAR, 0, 0, name,
19, &len))
    {
        printf("SQLBindParameter error!!! ==> %s \n",query);
        exit(1);
    }

    if (SQL_ERROR == SQLBindParameter(hstmt, 2, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_NUMERIC, 0, 0,
&age, 0, &len))
    {
        printf("SQLBindParameter error!!! ==> %s \n",query);
        exit(1);
    }

    /* Execute the prepared statement */
    sprintf(name, "김민석");
    age = 28;
    if (SQL_ERROR == SQLExecute(hstmt))
    {
        execute_err(hstmt, query);
    }

    sprintf(name, "홍길동");
    age = 25;
    if (SQL_ERROR == SQLExecute(hstmt))
    {
        execute_err(hstmt, query);
    }

    sprintf(name, "아무개");
    age = 34;
    if (SQL_ERROR == SQLExecute(hstmt))
    {
        execute_err(hstmt, query);
    }

    sprintf(query,"SELECT * FROM TEST001");
    if (SQL_ERROR == SQLExecDirect(hstmt,(SQLTCHAR*)query, SQL_NTS))
    {
        execute_err(hstmt, query);
    }

```

```

    }

    /* Store the SELECT result value to the variable */
    if (SQL_ERROR == SQLBindCol(hstmt, 1, SQL_C_CHAR, name, 21, &len))
    {
        printf("SQLBindCol error!!!\n");
        exit(1);
    }

    if (SQL_ERROR == SQLBindCol(hstmt, 2, SQL_C_SLONG, &age, 0, &len))
    {
        printf("SQLBindCol error!!!\n");
        exit(1);
    }

    while (SQLFetch(hstmt) == SQL_SUCCESS)
    // Print the result value to screen */
    {
        printf("Name : %5s, Age : %5ld\n", name, age);
    }

    /* Free all handles and disconnect */
    SQLFreeStmt(hstmt, SQL_DROP);
    SQLDisconnect(hdbc);
    SQLFreeConnect(hdbc);
    SQLFreeEnv(henv);
}

```

### 3.2.2 Execution Result

If you run the exe file generated after compiling in Visual C++, you will get the following result.

```

connected...
Name : 王 明 杰, Age : 28
Name : 李 明 杰, Age : 25
Name : 刘 明 杰, Age : 24
Press any key to continue...

```

### 3.3 Example of Using LOB

This section describes with examples, how to handle LOB data using the Altibase ODBC driver.

The Altibase LOB Locator requires that LOB data is handled in a session in non-autocommit mode. For further information, please refer to Chapter 3. LOB Interface in the *Altibase CLI User's Manual*.

The LongDataCompat attribute must also be set to ON in the connection string as shown below.

```
"DSN=ALTIBASE;LongDataCompat=ON"
```

or

```
"DRIVER=ALTIBASE_HDB_ODBC_64bit;User=SYS;Password=xxx;Server=127.0.0.1;PORT=20300;NLS_US  
E=US7ASCII;LongDataCompat=ON"
```

The following is an example of inserting BLOB data into a table and querying it in C#.

```
FileStream fs = new FileStream("c:\\test.dat", FileMode.Open, FileAccess.Read);  
Byte[] blob = new byte[fs.Length];  
fs.Read(blob, 0, System.Convert.ToInt32(fs.Length));  
fs.Close();  
  
OdbcTransaction tx = cn.BeginTransaction();  
cmd.Transaction = tx;  
  
cmd.CommandText = "INSERT INTO T1 (C1, C2) VALUES (?, ?)";  
cmd.Parameters.Add("C1", OdbcType.Int);  
cmd.Parameters.Add("C2", OdbcType.Binary);  
  
cmd.Parameters[0].Value = 1;  
cmd.Parameters[1].Value = blob;  
  
cmd.ExecuteNonQuery();  
tx.Commit();  
  
// BLOB SELECT  
cmd.CommandText = "SELECT binary_length(C2), C2 FROM T1";  
  
tx = cn.BeginTransaction();  
cmd.Transaction = tx;  
OdbcDataReader dr = cmd.ExecuteReader();  
int len;
```

```
while (dr.Read())
{
    len = dr.GetInt32(0);
    Byte[] ff = new Byte[len];
    dr.GetBytes(1, 0, ff, 0, len);

    fs = new FileStream("c:\\test.dat", FileMode.CreateNew, FileAccess.Write);
    fs.Write(ff, 0, len);
    fs.Close();
}
```



# Index

| <b>A</b>                        |    | <b>E</b>                                 |    |
|---------------------------------|----|--|----|
| Adding the DSN.....             | 19 | Example of Using LOB.....                | 28 |
| <b>B</b>                        |    | <b>I</b>                                 |    |
| Basic Programming Examples..... | 23 | Installing the Altibase ODBC Driver..... | 18 |
| <b>C</b>                        |    | Introduction.....                        | 10 |
| Configuration.....              | 19 | <b>O</b>                                 |    |
| Connection String.....          | 22 | ODBC API.....                            | 12 |
| <b>D</b>                        |    | ODBC Conformance Level.....              | 12 |
| Data Types.....                 | 11 | <b>U</b>                                 |    |
|                                 |    | Unix-like Operating Systems.....         | 18 |