

ALTIBASE® HDB™ Application Development

ODBC User's Manual

Release 6.5.1.

May 28, 2015



ALTIBASE HDB Application Development ODBC User's Manual
Release 6.5.1.
Copyright © 2001~2015 Altibase Corporation. 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 Corporation
10F, Daerung PostTower II, 182-13,
Guro-dong Guro-gu Seoul, 152-847, South Korea
Telephone: +82-2-2082-1000 Fax: 82-2-2082-1099
Homepage: <http://www.altibase.com>

Contents

Preface	v
About This Manual	vi
Audience.....	vi
Software Environment.....	vi
Organization.....	vi
Documentation Conventions	vi
On-line Manuals.....	vii
Altibase Welcomes Your Comments	vii
1. Introduction	1
1.1 Introduction.....	2
1.2 Data Types.....	3
1.3 ODBC API	4
1.3.1 ODBC Conformance Level	4
2. Installing and Configuring the ALTIBASE HDB ODBC Driver.....	7
2.1 Installing the ALTIBASE HDB ODBC Driver	8
2.1.1 Unix-like Operating Systems.....	8
2.1.2 Windows.....	8
2.2 Configuration	9
2.2.1 Unix-like Operating Systems.....	9
2.2.2 Windows.....	9
2.2.3 Adding the DSN.....	9
3. ODBC Programming	13
3.1 Connection String	14
3.2 Basic Programming Examples.....	15
3.2.1 Example.....	15
3.2.2 Execution Result.....	18
3.3 Visual C++ Example	19
3.4 Visual C# Example.....	20
3.5 Visual Basic Example.....	21
3.6 Example of Using LOB	22

Preface

About This Manual

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

Audience

This manual has been prepared for the following ALTIBASE HDB 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 HDB 6 is used as the database server.

Organization

This manual is organized as follows:

- [Chapter1: Introduction](#)
- [Chapter2: Installing and Configuring the ALTIBASE HDB 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.
<code>monospace</code>	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://support.altibase.com/>).

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://support.altibase.com/>).

Thank you. We appreciate your feedback and suggestions.

About This Manual

1 Introduction

This chapter introduces the ALTIBASE HDB 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 HDB ODBC driver, which conforms to ODBC 3.51 specifications, for this purpose.

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

1.2 Data Types

For further information on the SQL data types supported by ALTIBASE HDB and their mapping relationships with ODBC data types, please refer to “B. Appendix: Data Types” in *ALTIBASE HDB 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 HDB 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 HDB 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 HDB 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		
SQLColumnPrivileges	Level2	X	X	Column privileges are not supported by ALTIBASE HDB.
SQLColumns	Core	O		
SQLConnect	Core	O		
SQLCopyDesc	Core	X	O	
SQLDescribeCol	Core	O		

Function Name	Level	Support Status	Future Support	Remarks
SQLDescribeParam	Level2	0		Not fully supported
SQLDisconnect	Core	0		
SQLDriverConnect	Core	0		
SQLEndTran	Core	0		
SQLExecDirect	Core	0		
SQLExecute	Core	0		
SQLFetch	Core	0		
SQLFetchScroll	Core	0		
SQLForeignKeys	Level2	0		
SQLFreeHandle	Core	0		
SQLFreeStmt	Core	0		
SQLGetConnectAttr	Core	0		
SQLGetCursorName	Core	0		
SQLGetData	Core	0		
SQLGetDescField	Core	0		ODBC 3.0
SQLGetDescRec	Core	0		ODBC 3.0
SQLGetDiagField	Core	0		ODBC 3.0
SQLGetDiagRec	Core	0		ODBC 3.0
SQLGetEnvAttr	Core	0		
SQLGetFunctions	Core	0		
SQLGetInfo	Core	0		
SQLGetStmtAttr	Core	0		
SQLGetTypeInfo	Core	0		
SQLMoreResults	Level1	0		
SQLNativeSql	Core	0		
SQLNumParams	Core	0		
SQLNumResultCols	Core	0		
SQLParamData	Core	0		
SQLPrepare	Core	0		
SQLPrimaryKeys	Level1	0		

1.3 ODBC API

Function Name	Level	Support Status	Future Support	Remarks
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
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 HDB ODBC Driver**

This chapter offers explanations on how to install and configure the ALTIBASE HDB ODBC driver in Unix and Windows.

2.1 Installing the ALTIBASE HDB ODBC Driver

This section describes how to install the ALTIBASE HDB ODBC driver in Unix and Windows.

2.1.1 Unix-like Operating Systems

For Unix-like operating systems, the ALTIBASE HDB ODBC driver is installed when the ALTIBASE HDB server or client package is installed. For further information on how to install the ALTIBASE HDB 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 HDB 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`.

2.1.2 Windows

For Windows also, the ALTIBASE HDB ODBC driver is installed when the ALTIBASE HDB server or client package is installed. For further information on how to install the ALTIBASE HDB server or client package, please refer to the Installation Guide.

In Windows, you can install the ODBC driver file without installing the ALTIBASE HDB server or client package. If you download and run the ODBC driver installation file from <http://support.altibase.com>, the ALTIBASE HDB ODBC driver's DLL is installed to the system folder (for example, `C:\Windows\system32` for Windows7). For further information, please refer to the Windows ODBC Driver Installer User's Guide.

The ALTIBASE HDB ODBC driver's DLL file name for Windows is `altiodbc.dll`.

2.2 Configuration

This section describes how to configure the ODBC driver in Unix and Windows.

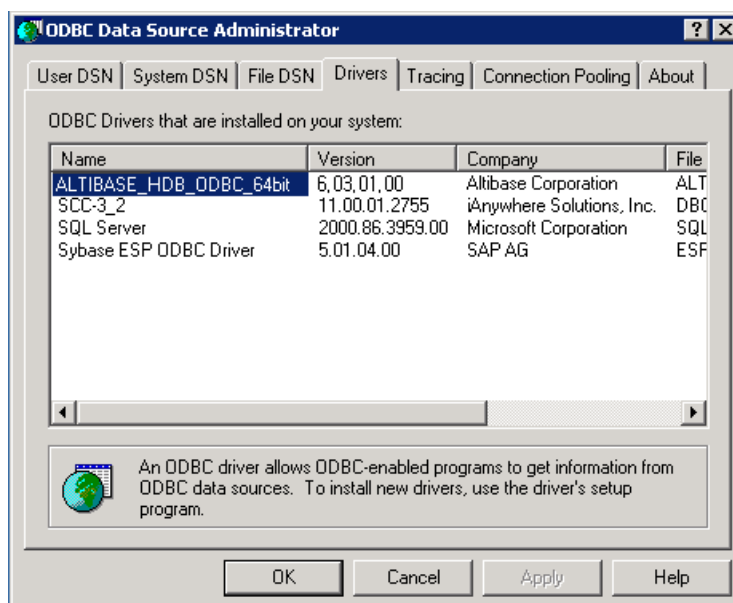
2.2.1 Unix-like Operating Systems

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 Windows

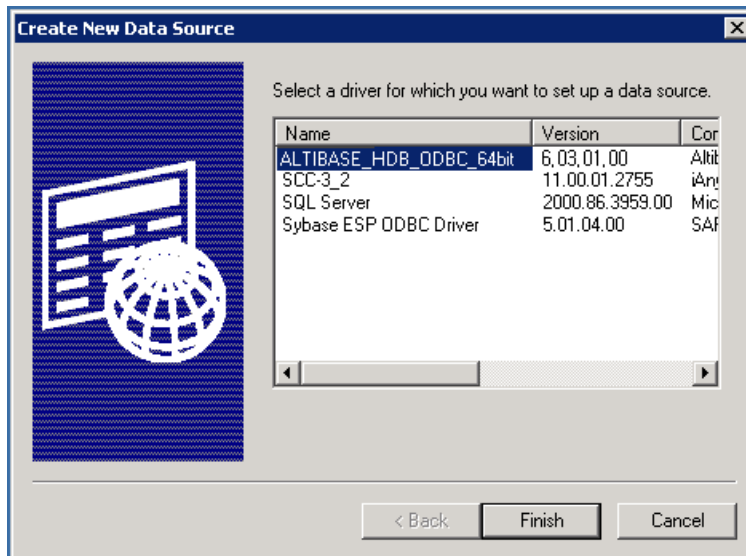
If the installation was successful, open [Control Panel] > [System and Security](Windows 7) > [Administrative Tools] > [Data Sources (ODBC)] and you will be able to see the ALTIBASE HDB ODBC driver in the Drivers tab.



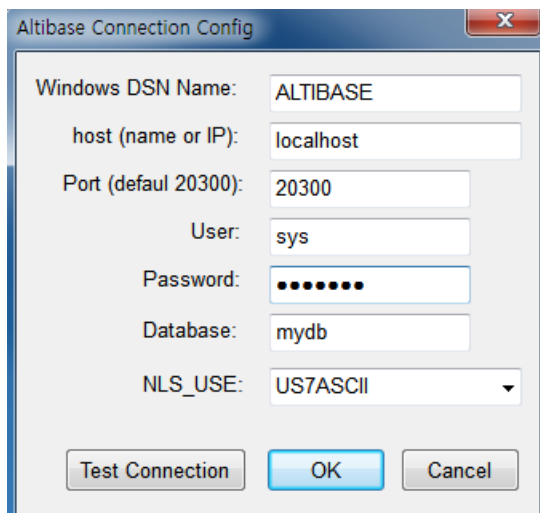
2.2.3 Adding the DSN

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 HDB ODBC driver and click "Finish"

2.2 Configuration



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 HDB server to be accessed is running.
- Port (default 20300): enter the listening port number of the ALTIBASE HDB 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.

2.2 Configuration

3 ODBC Programming

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

3.1 Connection String

When writing an ODBC application using ALTIBASE HDB 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 HDB server to be connected.
PORT	The listening port number of the ALTIBASE HDB server.
NLS_USE	The client character set.
LongData-Compat	ON or OFF. When using LOB (e.g., BLOB), it is recommended to set this to ON. 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 HDB 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");
    }

```

3.2 Basic Programming Examples

```
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);
}
}
```


3.2 Basic Programming Examples

```
/* 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, "??");
age = 28;
if (SQL_ERROR == SQLExecute(hstmt))
{
    execute_err(hstmt, query);
}

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

sprintf(name, "??");
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);
}
```

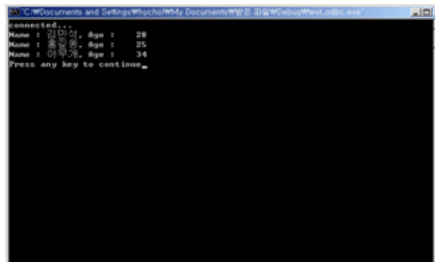
3.2 Basic Programming Examples

```
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.



```
C:\Documents and Settings\Whoa\My Documents\WP2_31\Working\test\test.exe
Name : [redacted], Age : 28
Name : [redacted], Age : 25
Name : [redacted], Age : 34
Press any key to continue...
```

3.3 Visual C++ Example

```
#include <Afx.h>
#include <Afxdb.h>
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    CDatabase db;

    try
    {
        db.OpenEx(_T("DSN=ALTIBASE"), CDatabase::noOdbcDialog);
        AfxMessageBox (_T("Connect OK"));
    } catch (CDBException *e)
    {
        AfxMessageBox(e->m_strError);
    }
    return 0;
}
```

The following is an example of using ADO in VC++.

```
#include "stdafx.h"
#include "windows.h"

#import "C:\Program Files\Common Files\System\ado\msado15.dll" \
no_namespace rename("EOF", "adoEOF")

int _tmain(int argc, _TCHAR* argv[])
{
    _ConnectionPtr pCon = NULL;

    try
    {
        CoInitialize(NULL);

        pCon.CreateInstance (__uuidof(Connection));
        pCon->Mode = adModeReadWrite;
        pCon->Open("DSN=ALTIBASE",
            "", "", adConnectUnspecified);

        MessageBox(NULL, (LPCSTR)"Successfully Connected.", (LPCSTR)"OK", 0);
        CoUninitialize();
    }
    catch (_com_error &ce)
    {
        MessageBox(NULL, ce.ErrorMessage(), (LPCSTR)"Error", 0);
    }
    return 0;
}
```

3.4 Visual C# Example

```
Sub Main()  
  
    Dim cn As Odbc.OdbcConnection  
    Dim cmd As Odbc.OdbcCommand  
    Dim dr As Odbc.OdbcDataReader  
    cn = New Odbc.OdbcConnection  
    cmd = New Odbc.OdbcCommand  
  
    cn.ConnectionString = "DSN=ALTIBASE"  
  
    Try  
        cn.Open()  
        Console.WriteLine("Successfully Connected.")  
        cmd.Connection = cn  
  
        cmd.CommandText = "SELECT TO_CHAR(SYSDATE,  
                            'YYYY-MM-DD HH:MI:SS')  
                            FROM DUAL"  
  
        dr = cmd.ExecuteReader()  
        While (dr.Read())  
            Console.WriteLine(dr.GetString(0))  
        End While  
    Catch ex As Odbc.OdbcException  
        Console.WriteLine("The connection has an error." + ex.Message)  
    End Try  
  
    Console.ReadLine()  
End Sub
```

3.5 Visual Basic Example

```
Sub Main()  
  
    Dim cn As Odbc.OdbcConnection  
    Dim cmd As Odbc.OdbcCommand  
    Dim dr As Odbc.OdbcDataReader  
    cn = New Odbc.OdbcConnection  
    cmd = New Odbc.OdbcCommand  
  
    cn.ConnectionString = "DSN=ALTIBASE"  
  
    Try  
        cn.Open()  
        Console.WriteLine("Successfully Connected.")  
        cmd.Connection = cn  
  
        cmd.CommandText = "SELECT TO_CHAR(SYSDATE,  
                            'YYYY-MM-DD HH:MI:SS')  
                            FROM DUAL"  
  
        dr = cmd.ExecuteReader()  
        While (dr.Read())  
            Console.WriteLine(dr.GetString(0))  
        End While  
    Catch ex As Odbc.OdbcException  
        Console.WriteLine("The connection has an error" + ex.Message)  
    End Try  
  
    Console.ReadLine()  
End Sub
```

3.6 Example of Using LOB

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

The ALTIBASE HDB 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 HDB 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;Pass-  
word=xxx;Server=127.0.0.1;PORT=20300;NLS_USE=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, FileAc-  
cess.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, FileAc-  
cess.Write);  
    fs.Write(ff, 0, len);  
    fs.Close();  
}
```

Index

A

Adding the DSN 9

B

Basic Programming Examples 15

C

Configuration 9

Connection String 14

D

Data Types 3

E

Example of Using LOB 22

I

Installing the Altibase ODBC Driver 8

Introduction 2

O

ODBC API 4

ODBC Conformance Level 4

U

Unix-like Operating Systems 8

V

Visual Basic Example 21

Visual C# Example 20, 21

Visual C++ Example 19

W

Windows 8