

Real Alternative DBMS ALTIBASE, Since 1999  
**WINDOWS환경의 ALTIBASE ODBC 개발 가이드**

2010. 09



Copyright © 2000~2013 ALTIBASE Corporation. All Rights Reserved.



---

## 목차

개요 .....	4
ODBC DRIVER 설정사항 .....	5
ODBC Driver 다운로드.....	5
ODBC Driver 설치.....	6
제어판의 ODBC 설정.....	6
ODBC DRIVER 개발 가이드.....	9
ODBC 연결문자열.....	9
Visual C++ 예제소스.....	9
Visual C# 예제소스.....	10
Visual Basic 예제소스.....	10
LOB 사용 시 주의사항.....	11

---

## 개요

본 문서는 WINDOWS환경의 ALTIBASE 개발환경에서 ODBC Driver를 이용한 각종 개발툴의 설정방법등을 가이드한다. ALTIBASE 버전은 5.3 을 기반으로 하며 문서를 위해 사용된 WINDOWS의 버전은 XP이다.

연관되어 다음과 같은 문서를 참고하는 것을 권장한다.

1. WINDOWS ADO.NET 환경의 ALTIBASE 개발 가이드
2. WINDOWS OLEDB 환경의 ALTIBASE 개발 가이드

## ODBC Driver 설정사항

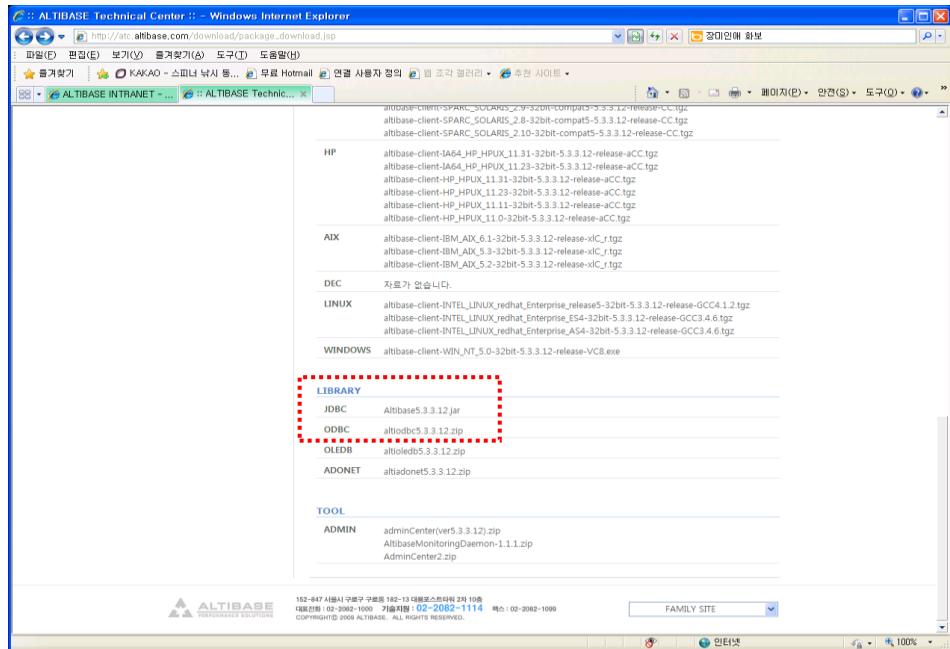
개발하기 전에 먼저 ALTIBASE ODBC환경의 개발을 위해서는 ALTIBASE가 제공하는 ODBC Driver를 설치해야 한다. 현재 <http://atc.altibase.com>에서 다운로드 받을 수 있으며 별도의 설치과정 없이 DLL파일을 제공 받을 수 있다.

## ODBC Driver 다운로드

<http://atc.altibase.com> 에 접속하여 “제품 다운로드” → “LIBRARY” 부분으로 이동하면 ODBC 파일을 내려 받을 수 있다. ALTIBASE는 제품버전 별로 ODBC Driver를 제공하고 있는데 만일, 서버패키지와 동일한 버전을 얻고자 할 경우는 “CLIENT” 부분에서 동일한 버전을 사용하면 된다.

(웹 사이트에 없는 이전 버전인 경우 [support@altibase.com](mailto:support@altibase.com)으로 문의하도록 한다.)





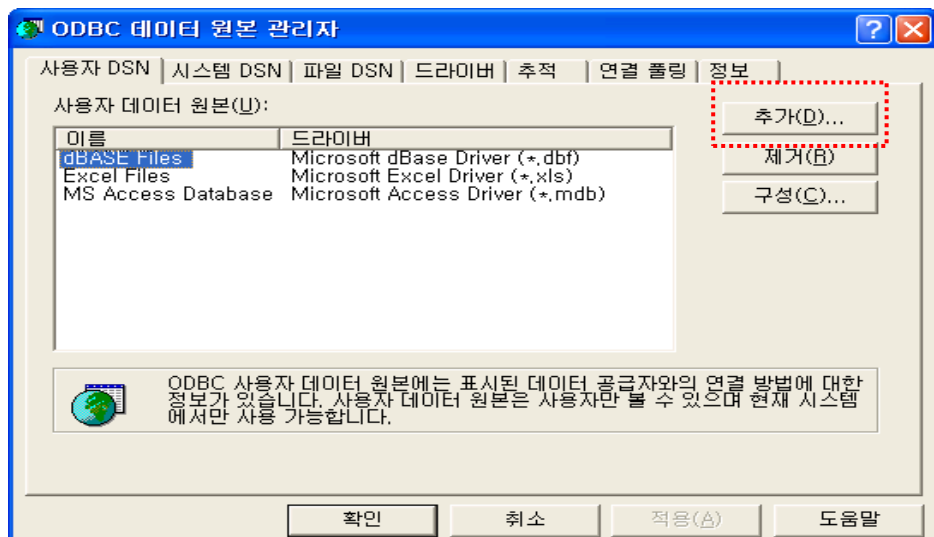
(본 문서의 작성 시점에는 altiodbc5.3.3.12.zip 파일이 최신으로 업로드 되어 있다.)

## ODBC Driver 설치

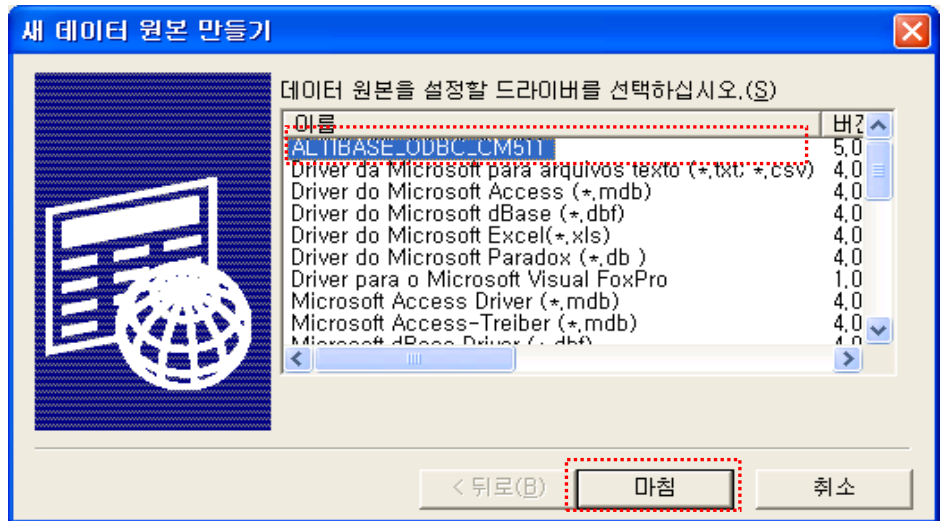
처음 설치할 경우라면 WINDOWS CLIENT 패키지를 다운로드 받아 설치하도록 한다. 설치파일에서는 별도의 복잡한 진행과정이 없어 본 문서에서는 별도로 설명하지 않는다.

## 제어판의 ODBC 설정

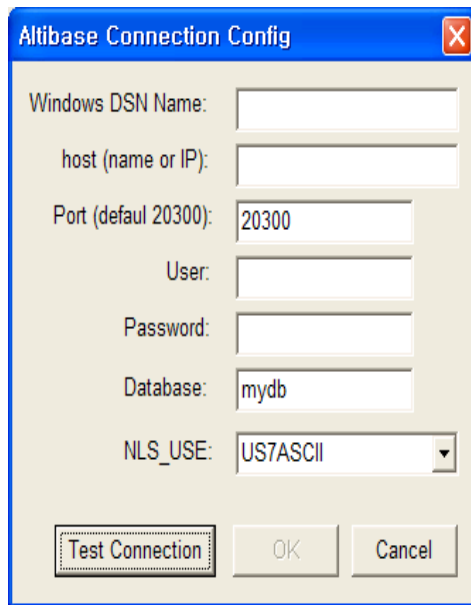
ALTIBASE ODBC Driver의 등록을 위해서는 “시작” → “제어판” → “관리도구” → “데이터 원본 (ODBC)” 을 클릭하여 ODBC 관리화면을 구동시킨다.



화면과 같이 “추가” 버튼을 클릭한다.



새로운 창이 뜬 후 “ALTBASE\_ODBC\_CM511” 이라는 이름이 확인이 된다면 ODBC Driver는 정상적으로 설치되었다고 볼 수 있다. 해당 Driver를 선택 한 후 “마침” 버튼을 클릭한다.

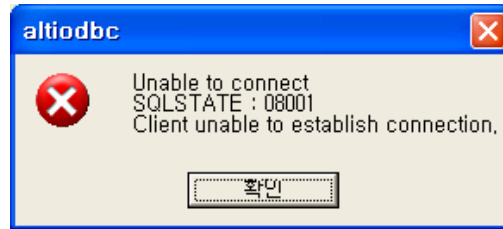


위와 같은 DB접속 설정화면이 나오면 다음과 같은 값을 입력하도록 한다.

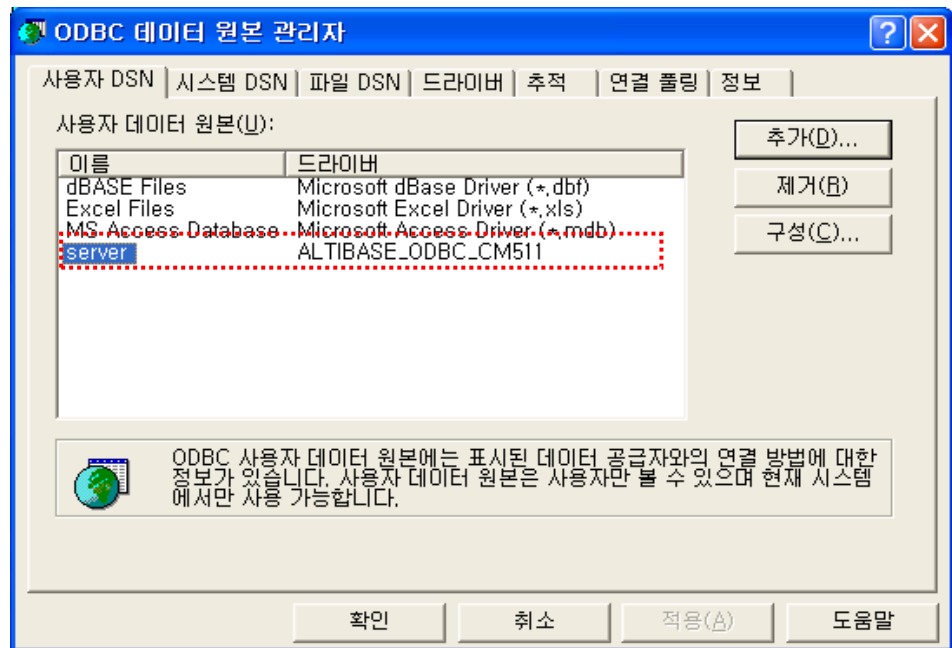
항목	설명	예
Windows DSN Name	다른 DSN Name과 구분되는 고유한 명칭	SERVER1
Host (name or IP)	ALTIBASE DB Server가 위치한 IP정보	192.168.1.35
Port (default 20300)	ALTIBASE 엔진이 Listen 하는 Port정보	20300
User	DB상의 계정	sys
Password	DB계정의 비밀번호	manager
Database	DB생성 시점에 만든 고유의 DB_NAME 이름	mydb
NLS_USE	DB 생성 시점에 입력한 문자셋 정보	MS949

정상적으로 입력 후 “Test Connection” 버튼을 클릭하면 ODBC Driver를 통해 ALTIBASE 서버로 정상 접속이 되는지를 체크해 볼 수 있다. (ALTIBASE DBMS가 구동된 상태이어야 함.)

오류가 날 경우 네트워크 상태에 다르지만 일정 시간이 소요된 후 다음과 같은 에러 경고창이 나타난다.



정상적으로 설정을 하였다면 아래와 같이 사용자가 새롭게 추가한 ODBC 항목이 목록에 추가되었음을 확인할 수 있다. 제거나 구성정보의 변경은 해당 화면의 “제거”, “구성” 버튼을 클릭하여 적용이 가능하다.





---

## ODBC Driver 개발 가이드

ODBC Driver를 통해 ALTIBASE에 접속할 경우는 별도의 소스 변환은 크게 필요하지 않으며 연결 부분에 ODBC 연결문자열등의 변경이 필요하다. 앞서 설명한 프로젝트의 별도 설정은 필요하지 않으며 WINDOWS에서 필요한 ODBC driver를 연동할 수 있는 헤더(ex. windows.h, sql.h, sqlext.h, afxdb.h)와 라이브러리(ex. odbc32.lib)들을 적절하게 사용하여 컴파일하면 된다.

---

### ODBC 연결문자열

ODBC Driver를 이용한 프로그램에서는 다음과 같이 연결문자열을 사용한다. 빨간색으로 표기된 부분은 키워드로 사용되는 것이며 파란색으로 표기된 부분은 사용자가 설치한 서버의 접속정보와 ODBC버전에 맞게 값을 변경하여 사용하면 된다.

<b>DRIVER=ALTIBASE_ODBC_CM511;user=sys;password=manager;</b> <b>Server=127.0.0.1;PORT=20300;NLS_USE=MS949;LongDataCompat=on</b>	
DRIVER	ODBC 관리도구에서 확인한 ALTIBASE Driver 이름
user	DB 사용자 명
password	DB 사용자의 패스워드
Server	접속할 DB서버의 IP
PORT	DB의 Listen Port
NLS_USE	문자셋
LongDataCompat	ON/OFF (BLOB등의 대용량 데이터를 사용할 경우 on으로 설정)

아래 단락에서는 Visual C++/Basic/C#에서의 예제소스를 사용하고 있으며 “ODBC 연결문자열” 이라고 표현한 곳에는 위의 표에 언급된 내용을 사용자 환경에 맞게 변경하여 사용하면 된다.

---

### Visual C++ 예제소스

Visual C++에서의 간단한 연결 예는 아래와 같다.

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

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

    try
    {
        db.OpenEx(_T("ODBC연결문자열"), CDatabase::noOdbcDialog);
        AfxMessageBox (_T("Connect OK"));
    } catch (CDBException *e)
    {
    }
}
```

```
AfxMessageBox(e->m_strError);
}
return 0;
}
```

---

## Visual C# 예제소스

Visual C#에서의 간단한 연결 예는 아래와 같다.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Data.Odbc;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            OdbcConnection cn = new OdbcConnection();

            try
            {
                cn.ConnectionString = "ODBC연결문자열";
                cn.Open();
                Console.WriteLine("connect ok");
            }
            catch (OdbcException ex)
            {
                Console.WriteLine(ex.Message);
            }
            Console.ReadLine();
        }
    }
}
```

---

## Visual Basic 예제소스

Visual Basic 에서의 예제는 다음과 같으며 아래 예제는 DB에 접속하여 현재 날짜와 시간을 가져오는 소스로 구현되어 있다. (ADO객체를 사용하여도 동일하다.)

```
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 = "ODBC 연결문자열"

Try
  cn.Open()
  Console.WriteLine("정상적으로 연결되었습니다")
  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("연결에 오류가있습니다." + ex.Message)
End Try

Console.ReadLine()

End Sub

```

## LOB 사용 시 주의사항

ALTIBASE의 경우는 LOB 데이터타입을 사용할 경우 반드시 DB연결 세션의 속성을 Non-AutoCommit으로 변경한 후 사용이 가능하다. 그렇지 않을 경우 조회 시점에 LOB 데이터타입에 대해 Null을 가지고 오거나 삽입/변경 시점에 다음과 같은 에러를 발생하게 된다.

Connection is in autocommit mode. One can not operate on LOB datas with autocommit mode on.

다음에서 C# 소스를 통해 BLOB 데이터타입을 테이블에 삽입/조회하는 예제를 살펴해보도록 한다. (여타의 개발언어에서도 ALTIBASE LOB 사용 시 트랜잭션의 관리를 해야 한다는 사실을 반드시 고려해야 한다.)

BLOB INSERT의 예제는 DB연결이 된 상태에서 앞서 설명한 바와 같이 세션의 속성을 Non-AutoCommit으로 변경을 하여 (C#의 경우 트랜잭션의 개시) 데이터를 삽입하는 예제이다. 사용된 "blob" 변수는 Byte[] 타입으로 선언하였다.

BLOB SELECT의 예제에서는 BLOB타입의 길이를 알아내기 위해 ALTIBASE가 제공하는 BINARY\_LENGTH라는 함수를 통해 Byte[]변수를 선언하고 해당 변수에 데이터를 저장 후 파일로 생성하는 예제로 구성한 경우이다.

```

// BLOB INSERT
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();
}

```



#### **알티베이스㈜**

서울특별시 구로구 구로3동 182-13  
대림포스트 2차 1008호  
02-2082-1000  
<http://www.altibase.com>

#### **대전사무소**

대전광역시 서구 둔산동 921  
주은리더스텔 901호  
042-489-0330

#### **기술지원본부**

서울특별시 구로구 구로3동 182-13  
대림포스트 2차 908호  
02-2082-1000

#### **기술지원센터**

02-2082-1114  
<http://support.altibase.com>

Copyright © 2000~2013 ALTIBASE Corporation. All Rights Reserved.

이 문서는 정보 제공을 목적으로 제공되며, 사전에 예고 없이 변경될 수 있습니다. 이 문서는 오류가 있을 수 있으며, 상업적 또는 특정 목적에 부합하는 명시적, 묵시적인 책임이 일체 없습니다. 이 문서에 포함된 ALTIBASE 제품의 특징이나 기능의 개발, 발표 등의 시기는 ALTIBASE 재량입니다. ALTIBASE는 이 문서에 대하여 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산을 보유할 수 있습니다.