Altibase FAQ | Development & API

ODBC

32-bit ODBC installation on 64-bit Windows

Overview

This document describes how to set up 32-bit ODBC driver on Windows Server 2003 64-bit systems.

Version

~ Altibase 6.5.1 (**Note:** Starting from version 7.1, 32-bit ODBC drivers are no longer supported.)

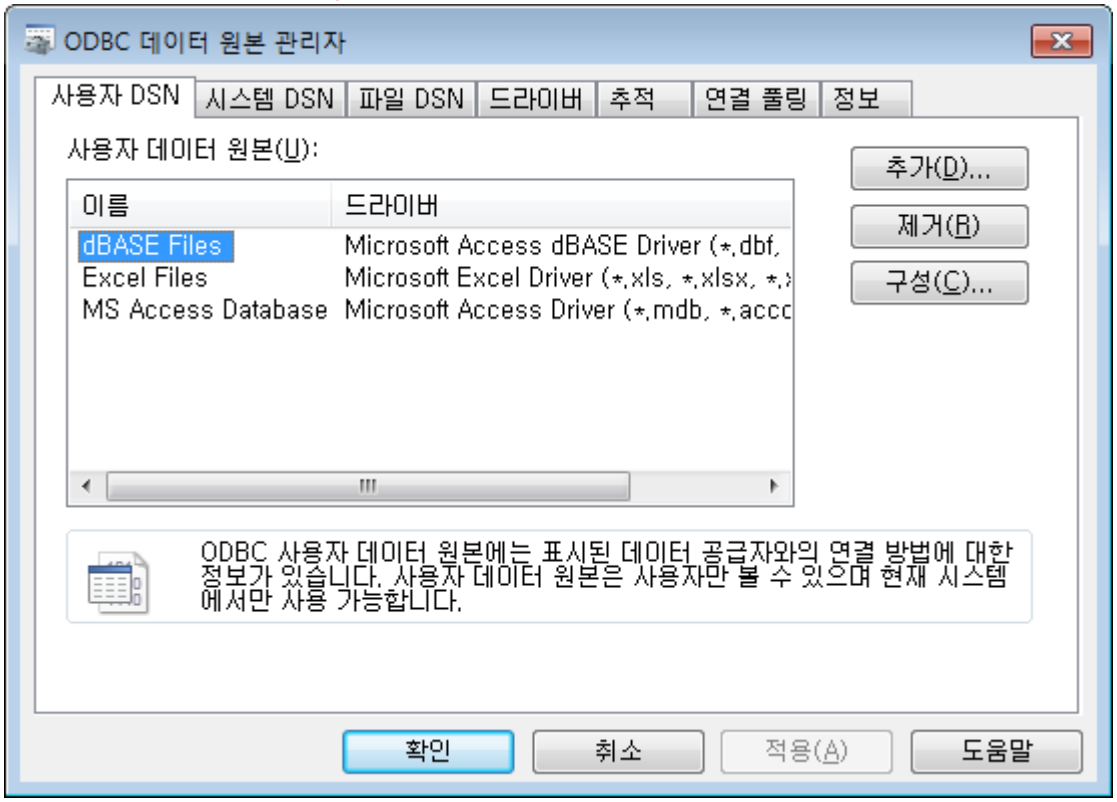
Altibase ODBC installation

1. Download the Windows Altibase 32-bit client installation file or the Altibase 32-bit ODBC installation file.
   Download: http://support.altibase.com/en/product
   Client installation file name example: altibase-HDB-client-x.x.x.x.x-WINDOWS-X86-32bit-release.exe
   ODBC installation file name example: altibase-HDB-ODBC-x.x.x.x.x-WINDOWS-X86-32bit-release.exe
2. Install the Windows Altibase 32-bit client.
   For the installation method, refer to 2. Product Installation Using Package Installer -> Altibase Client Product Installation in Windows in the Installation Guide manual.
   Manual download: https://manual.altibase.com

© Altibase Corp.

ODBC Setting

1. Run the data source manager for 32-bit ODBC.
   Double-click C:\windows\sysWOW64\odbcad32.exe.

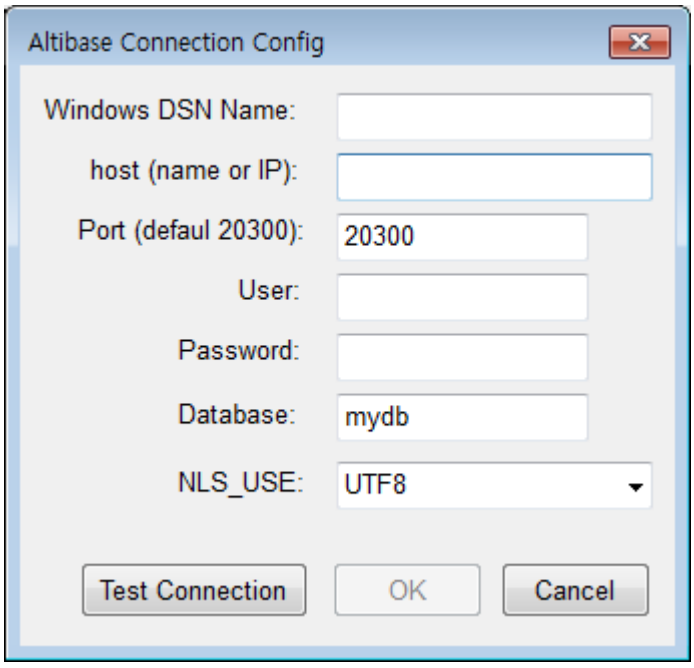


2. Create a new data source
   Click the 'Add' button in the user 'DSN tab' of the ODBC Data Source Administrator window to open the Create New Data Source window.
   Select the Altibase 32-bit ODBC driver and click 'Finish'.



© Altibase Corp.

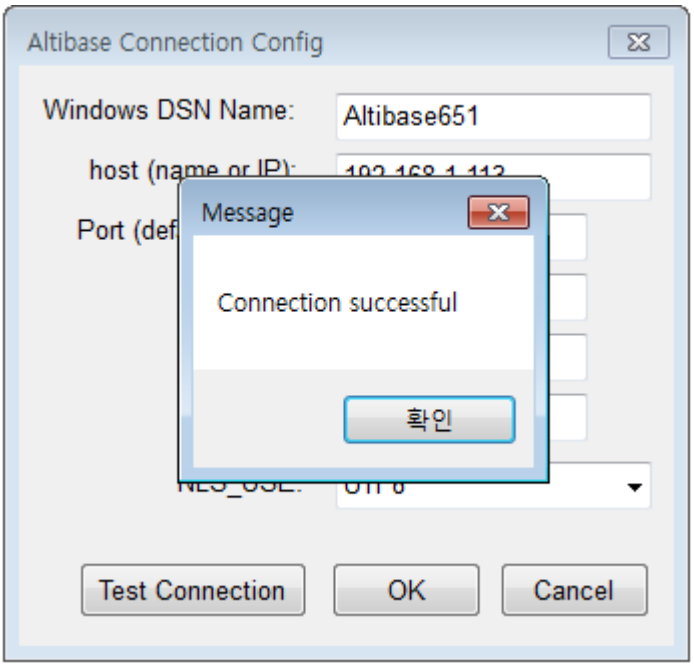
3. When the 'Altibase Connection Config' window appears, enter the Altibase server connection information.



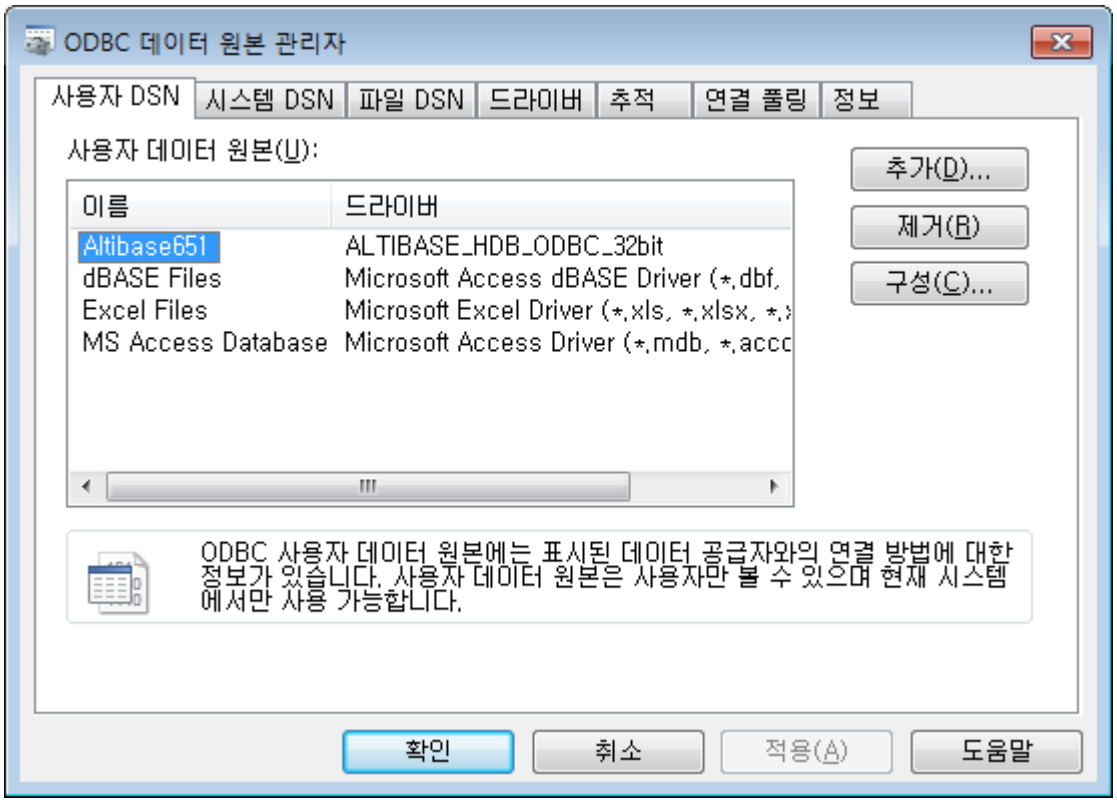
● Windows DSN Name
Enter a custom name for the data source.
● host (name or IP)
Enter the IP of the Altibase server.
● Port (default 20300)
Enter the service port of the Altibase server
Check the value of PORT_NO in the altibase.properties file (Altibase server properties file) or the value of the ALTIBASE_PORT_NO environment variable on the Altibase server.
● User
Enter the database user name.
● Password
Enter the database user password.
● Database
Enter the database name.
The database name can be checked with SELECT DB_NAME FROM V$DATABASE;
● NLS_USE
Enter the Altibase server character set.
The Altibase server character set can be checked with SELECT NLS_CHARACTERSET FROM V$NLS_PARAMETERS;

© Altibase Corp.

4. Click the Test Connection button to check the connection.



5. Check the Altibase DSN added to the User DSN tab.



ODBC function, SQLFreeStmt

Overview

This document describes how to use the ODBC function and SQLFreeStmt.

© Altibase Corp.

Version

- This document is written based on Altibase version 6.3.1
- For additional information or updates, please leave a request at http://support.altibase.com/en/ or in a comment section on this page.

SQLFreeStmt options

The following four options are described.

Item	Description
SQL_CLOSE	Close the cursor associated with stmt and discard all pending results. The application can reopen this cursor later by executing the SELECT statement again using the same or different variables. If no cursor is open, this option has no effect on the application.
SQL_DROP	The resources associated with the input statement handle are dropped and the handle is invalidated. If there is an open cursor, it is closed and all pending results are discarded.
SQL_UNBIND	All rows bound by the previous SQLBindCol() call on this statement handle are dropped.
SQL_RESET_PARAMS	All parameters set by the previous SQLBindParameter() call on this statement handle are dropped. The association between the application variable or file reference and the SQL statement parameter marker in the statement handle is broken.

Among these, SQL_CLOSE and SQL_DROP are mainly used.

1. SQLFreeStmt(stmt, SQL_DROP):
   This means that the resources for the prepared stmt (statement handle) are completely freed.
   That is, it only calls when that stmt is not being reused.
2. SQLFreeStmt(stmt, SQL_CLOSE):
   If all the retrieved data have not been fetched, an error will occur if you execute SQLExecute again without SQLFreeStmt(SQL_CLOSE).
   If SQLFreeStmt(stmt, SQL_CLOSE) is used, the existing stmt can be used without doing SQLAllocStmt().

Integrating with unix_odbc

Overview

This document is for integrating Altibase with UNIX_ODBC.

Version

All the versions of Altibase

© Altibase Corp.

UNIX ODBC type and URL

unix_odbc is an open source project, and unixODBC is mainly used.

1. unixODBC can be downloaded from the link below:
   - http://unixodbc.org/

INSTALL

```
$ mkdir $HOME/ODBC_HOME
$ gzip -d unixODBC-2.3.2.tar.gz
$ tar -xvf unixODBC-2.3.2.tar
$ cd unixODBC-2.3.2
$ cd $ALTIBASE_HOME/lib
$ file libaltibase_odbc-64bit-ul64.so
libaltibase_odbc-64bit-ul64.so : ELF 64-bit LSB shared object, AMD x86-64, version 1 (SYSV), not stripped$

./configure -prefix=/home/wonsik/ODBC_HOME --enable-gui=no --enable-threads=yes
```

> **Tip**
>
> **Option description**
>
> prefix: The path to install unixODBC
> enable-gui: Whether to build GUI ODBC Administrator
> enable-threads: The default is "yes" if thread-support is found on the machine. All modern Linuxes will have pthreads support in glibc so it is probably best to leave this to default.
>
> If the enable-threads option is set to no, the following error may occur when testing the connection with isql.
>
> $ ./isql Altiodbc
> ./isql: symbol lookup error: /home/wonsik/altibase_home/lib/ libaltibase_odbc-64bit-ul64.so: undefined symbol: pthread_sigmask
>
> $./isql Altiodbc
> ./isql: symbol lookup error: /home/wonsik/altibase_home/lib/libaltibase_odbc-64bit-ul64.so: undefined symbol: pthread_sigmask

```
$ make
$ make install
$ cd ODBC_HOME
$ dltest $ALTIBASE_HOME/lib/libaltibase_odbc-64bit-ul64.so
SUCCESS: Loaded /home/wonsik/altibase_home/lib/libaltibase_odbc-64bit-ul64.so

$ cat odbc.ini
[ODBC Data Sources]
Altiodbc = Altibase ODBC Driver

[Altiodbc]
Driver = /home/wonsik/altibase_home/lib/libaltibase_odbc-64bit-ul64.so
Description = Sample Altibase DSN
UserName = SYS
Password = MANAGER
ServerType = Altibase
```

© Altibase Corp.

```
Server = 127.0.0.1
User = SYS
Port = 40501
Database = mydb
FetchBufferSize = 64
ReadOnly = no
TraceFile = /home/wonsik/ODBC_HOME/odbc.trace
Trace = 0
Trace option means 1 (debug), 0 (normal)

$ export ODBCINI=/home/wonsik/odbc.ini

$ cd $HOME/ODBC_HOME/bin

$ odbc_config --ulen
-DSIZEOF_SQLULEN=8

$ ./isql Altiodbc
+---------------------------------------+
¦ Connected!                            ¦
¦                                       ¦
¦ sql-statement                         ¦
¦ help [tablename]                      ¦
¦ quit                                  ¦
¦                                       ¦
+---------------------------------------+
SQL>
```

PHP, PERL Integration

PHP, PERL, etc. support unix_odbc, so it can be integrated with Altibase in the above way.

PHP Integration Guide

1. Setting related environment variables

```
Be sure to set the following variables in advance.

export CFLAGS=-DBUILD_LEGACY_64_BIT_MODE=1

export ODBCINI=odbc.ini File location path

export LD_LIBRARY_PATH=$APACHE_HOME/modules:unixODBC installation path/lib:PHP installation path/lib
```

2. Check SQLLEN, SQLULEN

```
Move to the bin directory of the unixODBC installation path and use the odbcinst command to check whether
SQLLEN size and SQLULEN size are 4.

In case of 4, it is set to 32bit, and in case of 8, it is set to 64bit.

$ ./odbcinst -j

unixODBC 2.3.2

DRIVERS............: /home/altibase/phptest/unixodbc/etc/odbcinst.ini

SYSTEM DATA SOURCES: /home/altibase/phptest/unixodbc/etc/odbc.ini

FILE DATA SOURCES..: /home/altibase/phptest/unixodbc/etc/ODBCDataSources

USER DATA SOURCES..: /home/altibase/phptest/unixodbc/etc/odbc.ini
```

© Altibase Corp.

```
SQLULEN Size.......: 4

SQLLEN Size........: 4   <-If 4, 32 bit. If 8, 64bit.

SQLSETPOSIROW Size.: 2
```

3. Modify unixODBC's odbc.ini file

```
It can be set to process lob data by adding LongDataCompat=ON to the odbc.ini file.
```

4. PHP compilation and installation operation

```
$> ./configure \
--prefix=/user/web/php \
--with-config-file-path=/user/web/php/conf \
--with-apxs2=/user/web/apache2/bin/apxs \
--with-unixODBC=/user/web/odbc \
--with-mcrypt \
--with-mhash \
--with-openssl \
--with-kerberos \
--with-curl \
--enable-dom \
--with-iconv=/usr/local \
--with-xmlrpc \
--enable-libxml \
--with-libxml-dir=/usr/lib64 \
--with-gd \
--with-freetype-dir=/usr \
--with-jpeg-dir=/usr/lib64 \
--with-png-dir=/usr/lib64 \
--with-zlib-dir=/usr/lib64 \
--enable-mbstring \
--enable-shmop \
--enable-sockets \
--enable-sigchild \
--enable-soap \
--enable-zip \
--enable-ftp \
--enable-sysvmsg \
--enable-sysvsem \
--enable-sysvshm \
--with-regex=php \
--with-pcre-regex \
--enable-mbregex \
--enable-dba=shared \
--enable-mod-charset \
--with-xml \
--with-pcre \
--with-gdbm \
--with-dbm



$> make

$> make install
```

5. php.ini settings

```
memory_limit=-1
```

© Altibase Corp.

```
odbc.defaultbinmode=1

odbc.defaultlrl=1048576
```

© Altibase Corp.