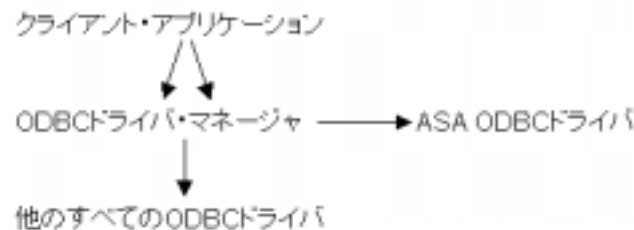


ODBC for Linux

当社のソフトウェアには、Adaptive Server Anywhere ODBC ドライバが付属しています。ASA ODBC ドライバはドライバ・マネージャと同等の機能を果たすため、ほとんどのアプリケーションは実際のドライバ・マネージャを必要としません (UNIX のドライバ・マネージャは入手が困難な場合や高価な場合があります)。ASA ODBC ドライバは SQLDriverConnect と SQLConnect を扱いますが、別のドライバはロードしません。

ODBC ドライバを使用する状況としては、次の 2 つが考えられます。

- SQL Anywhere ODBC ドライバとサード・パーティの ODBC ドライバ・マネージャ (たとえば unixODBC)
- SQL Anywhere ODBC ドライバのみ (ODBC ドライバ・マネージャを使わずに Adaptive Server Anywhere ODBC ドライバを使用する場合は、Adaptive Server Anywhere データ・ソースにしかアクセスできない)



ODBC ドライバ・マネージャと ODBC ドライバのリンク

libodbc.so (libodbc.so.1) を使用する ODBC アプリケーションを使用している場合は、単に libodbc.so と libodbcinst.so に対する symlink を作成して、その両方が /opt/SYBSasa6/lib/dbodbc6.so.1 を指し示すようにします。

例:

```
ln -s /opt/SYBSasa6/lib/dbodbc6.so.1 libodbc.so
ln -s /opt/SYBSasa6/lib/dbodbc6.so.1 libodbcinst.so
```

カスタムの ODBC アプリケーションを作成する場合は、アプリケーションを dbodbc6.so に直接リンクできます。

ODBC ドライバ・マネージャが検出されなかった場合、Adaptive Server Anywhere は .odbc.ini をデータ・ソース情報のために使用します。Adaptive Server Anywhere は、次の中で最初に見つかったファイルを使用します。

```
$ODBCINI
$ODBC_INI
$ODBCHOME/.odbc.ini
$HOME/.odbc.ini
~/.odbc.ini [according to pwuid info]
```

*** 注: SQLDrivers は実装されていません。

サンプル .odbcinst.ini ファイル

.odbcinst.ini の例

```
-----
[ODBCDrivers]
Sybase SQL Anywhere=Installed
[Sybase SQL Anywhere]
Driver=/opt/SYBSsa6/lib/dbodbc6_r.so
```

サンプル .odbc.ini ファイル

.odbc.ini の例

```
-----
[ODBC Data Sources]
ASAny 6.0 Sample=asademo.db
[ASAny 6.0 Sample ]
Driver=/opt/SYBSasa6/lib/dbodbc6_r.so
UID=dba
PWD=sql
DatabaseName=asademo
```

***注: どちらの ini ファイルもピリオドで始まっていることに注意してください。これは、ファイルを隠すことによって、エンドユーザが誤って削除する可能性を低くするための処置です。

リモート接続用のサンプル .odbc.ini ファイルのフォーマット

クライアントが Linux にあり、サーバが別のマシンにあるリモート接続の場合は、少なくとも "EngineName" 接続パラメータと "CommLinks" 接続パラメータが .odbc.ini ファイルに含まれている必要があります。

次にエントリの例を示します。

```
[SQLAnywhere]
Driver=/opt/SYBSasa6/lib/libodbc.so
EngineName=rickb
DatabaseName=sample
CommLinks=tcPIP
```

odbc.c サンプルのコンパイルと実行

/opt/SYBSasa6/lib ディレクトリに移動します。

libodbc.so と libodbcinst.so に対する symlink を作成して、その両方が /opt/SYBSasa6/lib/dbodbc6.so.1 を指し示すようにします。

例:

```
ln -s /opt/SYBSasa6/lib/dbodbc6.so.1 libodbc.so
ln -s /opt/SYBSasa6/lib/dbodbc6.so.1 libodbcinst.so
cd into the /opt/SYBSasa6/cxmp directr:
/opt/SYBSasa6/cxmp
```

makeall スクリプトに従って環境変数をいくつか設定します。

COMPILER — 使用するコンパイラ。デフォルトは gnu。

ASANY — ASA6 インストールの場所。

(デフォルト :/opt/SYBSasa6)

ODBC — ODBC ドライバ・マネージャのインストールの場所。

(デフォルト :/opt/odbc)

PLATFORM — Linux で実行する場合は、これを LINUX として定義する。

次のように入力します。

```
make -f makeall
```

実行するには、次のように入力します。

```
./odbc
```

query_db の例

ODBC 接続の基本的なテストを行って確認メッセージを出力する例をもう 1 つ示します。

```
#include <stdio.h>
#include <stdlib.h>
#include "/opt/SYBSasa6/include/odbc.h"
HENV Environment;
HDBC Connection;

HSTMT Statement;

static unsigned char SelectStmt[] = {"SELECT 'Connected to: ENG=' ||property('name') || ',
DBN=' ||db_property('name')"};

static void CheckReturn()
{
    SQLSMALLINT handletype;
    SQLHANDLE handle;
    unsigned char sqlstate[ 6 ];
```

```

unsigned char msg[ 256 ];

if( Statement != NULL )
handletype = SQL_HANDLE_STMT;
handle = Statement;
} else if( Connection != NULL )
handletype = SQL_HANDLE_DBC;
handle = Connection;
} else
handletype = SQL_HANDLE_ENV;
handle = Environment;
}
SQLGetDiagRec( handletype, handle, 1, sqlstate, NULL, msg, sizeof( msg ), NULL );
printf( "SQL error %s -- %s¥n", sqlstate, msg );
exit( 1 );
}
int main( int argc, char * argv[] )
/*****/
{
int res;
unsigned char * connect_name = (unsigned char *)"test";
unsigned char * connect_userid = (unsigned char *)"DBA";
unsigned char * connect_password = (unsigned char *)"SQL";
SQLCHAR cname[256];
SQLINTEGER cnamesize;
if( SQLAllocEnv( &Environment ) != SQL_SUCCESS )
printf( "Unable to allocate env¥n" );
exit( 0 );
}
if( SQLAllocConnect( Environment, &Connection ) != SQL_SUCCESS )
printf( "Unable to allocate connection¥n" );
SQLFreeEnv( Environment );
exit( 0 );
}
if( argc > 1 )
connect_name = (unsigned char *)argv[1];
if( argc > 2 )
connect_userid = (unsigned char *)argv[2];
if( argc > 3 )
connect_password = (unsigned char *)argv[3];
}
}
}
printf( "connection parameters: DSN='%s', UID='%s', PWD='%s'¥n", connect_name,
connect_userid, connect_password);

```

```

res = SQLConnect( Connection,
connect_name, SQL_NTS,
connect_userid, SQL_NTS,
connect_password, SQL_NTS );
if( res != SQL_SUCCESS )
printf( "Unable to open data source (ret=%d)\n", res );
CheckReturn();
exit( 0 );
}
if( SQLAllocStmt( Connection, &Statement ) != SQL_SUCCESS )
printf( "Unable to allocate statement\n" );
CheckReturn();
exit( 0 );
}
if( SQLExecDirect( Statement, SelectStmt, SQL_NTS )
!= SQL_SUCCESS )
printf( "Unable to execute statement\n" );
CheckReturn();
exit( 0 );
}
res = SQLFetch(Statement);
if( res != SQL_SUCCESS && res != SQL_SUCCESS_WITH_INFO )
printf( "Unable to fetch row\n" );
CheckReturn();
exit( 0 );
}
if( SQLGetData(Statement, 1, SQL_C_CHAR, cname, 255, &namesize)
!= SQL_SUCCESS)
printf( "Unable to get data\n" );
CheckReturn();
exit( 0 );
}
printf( "%s\n", cname );
SQLDisconnect( Connection );
SQLFreeConnect( Connection );
SQLFreeEnv( Environment );
printf( "Done.\n" );
exit( 0 );
}

```

このサンプルをコンパイルするには、次を実行します。

```

g++ -Wall -I/opt/SYBSasa6/include/ -L/opt/SYBSasa6/lib/ -DUNIX -DODBC_UNIX -lodbc -o
query_db query_db.c

```

また、次のように入力して、このサンプルを実行します。

```

./query_db

```

出力は次のようになります。

```
connection parameters: DSN='test', UID='DBA', PWD='SQL'
```

```
Connected to: ENG=rickb, DBN=asademo
```

```
Done.
```