Connecting to MS SQL 6.x+ via Openlink/PHP/ODBC mini−HOWTO

Table of Contents

Connecting to MS SQL 6.x+ via Openlink/PHP/ODBC mini−HOWTO

Zili Zhang, `silen@silen.net`

3.0, 1999−07−15

How to connect to MS SQL 6.x+ database server via ODBC functions of PHP3(3.0.1x or above) compiled with Openlink drivers under Linux.

---

1. Introduction

This document describes how to connect to MS SQL 6.x+ database server via ODBC functions of PHP3(3.0.1x or above) compiled with Openlink drivers under Linux.

In the Unix world, people may hate to be invoked with any M$ related software. However, in the real world, the boss may require you to store data in a MS SQL database on NT and run web application on Linux. What to do? Quit the job or sit down to read this document? If you choose latter, I will give you a detailed installation guide so that you can keep the job for a while. It is a How−to guide, not a Why−so guide. So don't ask me why abc instead of cba. I don't know either.

PHP is becoming more and more popular in web programmers, mainly because it can be configured to connect to various databases like Oracle, MySQL, Solid and so on. But for a MS SQL server, the problem is different. Though you can use PHP's Sybase−ct support features to directly connect to MSSQL, many people (at least me) prefer to connect via ODBC.

Utilizing Openlink ODBC middleware, you can use those ODBC_xxxx functions to connect to all the database Openlink has a driver for. You have to install Openlink Linux client and Server middleware and re−compile PHP to support ODBC functions. In the end, I made a script sample for reference.

1.1 Copyright

1.2 Disclaimer

The following document is offered in good faith as comprising only safe configuration and procedures. No responsibility is accepted by the author for any loss or damage caused in any way to any person or equipment, as a direct or indirect consequence of following these instructions.

This document was derived from the plain text entry found on http://www.silen.net/openlink−php−odbc.txt.

---

2. Openlink

This step is a little bit complicated. You have to done works on both sides of your Linux client machine and NT server.

2.1 On the Client

- From ftp://www.openlinksw.com/, download `install.sh` and `likoxglc.taz` (for a libc6 system) or `likoxxxx.taz` (for a libc5 system).
- `mkdir /usr/local/openlink`
- copy `install.sh` and `likoxglc.taz` into `/usr/local/openlink`
- `cd /usr/local/openlink`
- `sh install.sh`, the install script will ask you the owner and group of the program. It will extract things to `odbcsdk` directory under `/usr/local/openlink` and copy a `.odbc.ini` into the owner's home directory.

2.2 On the Server

- From ftp://www.openlinksw.com/, download `ntadm65x.zip` onto your NT server.
- `unzip ntadm65x.zip`
- `cd disk1` – directory of where you unzip the package.
- execute `setup` and follow the instructions to install the Openlink middleware.
- remember to start Openlink request broker from start menu or service control panel.

2.3 Configuration Before Test

- keep the `.odbc.ini` file in your home directory.
- copy the `udbc.ini` from the bin directory of Openlink middleware install directory to the `/etc` directory of client.
- customize `/etc/udbc.ini`. In [dsn_sql6] section, change the host, database, username and password entries to fit your server settings. Here is part of my `/etc/udbc.ini`:

```
[dsn_sql6]
Host            = 10.0.0.1
ServerType      = sql6
;ServerOptions  =
Database        = pubs
;FetchBufferSize = 30
UserName        = sa
Password        = xxxxxxx
```

- add to your environment `LD_LIBRARY_PATH='/usr/local/openlink/odbcsdk/lib'` and export it. In csh shell, type: `setenv LD_LIBRARY_PATH /usr/local/openlink/odbcsdk/lib`

2.4 Text with `odbctest`

- `cd /usr/local/openlink/odbcsdk/examples`
- `./odbctest`
- type: `dsn=dsn_sql6`
- when 'sql>' appears, you can execute your sql clauses to test the connection.

---

3. PHP

Now we have Openlink and can go to PHP compilation. Note for this version, PHP 3.0.10 or above is required to make things work.

- From http://www.php.net, download `php−3.0.11.tar.gz`
- Perform the following steps:

```
gzip -dc php-3.0.11.tar.gz|tar -xof -

cd php-3.0.11

./configure --with-openlink (--with-mysql --with-gd=/usr/local/gd1.3 --enable-track-vars)

    NOTE: My configuration is to run PHP as CGI mode, support mysql,
          as well. Your configuration may be different.

make --silent
```

```
        NOTE: Don't mind if there are warning messages.

make install
```

These will install php executable into `/usr/local/bin`. Before you execute php, copy the library files under `/usr/local/openlink/odbcsdk/lib` into `/usr/lib` to make it easier for php to find openlink libraries (I know there are better methods).

---

4. Example

Please refers to PHP manual for ODBC functions. Here is my example code odbc.php3: (Note that odbc_num_rows() will not return anything, so you must repeately call odbc_fetch_row() to get that number.)

```
<?
/* some environment variables, you can test to comment them out to see
 * if things still work.
 */
putenv("LD_LIBRARY_PATH=/usr/local/openlink/odbcsdk/lib");
putenv("UDBCINI=/etc/udbc.ini");
putenv("ODBCINI=/root/.odbc.ini");
putenv("DebugFile=/tmp/udbc.out");      // debug trace output

$dsn="DSN=dsn_main";    // note 'DSN=' is required
$user="sa";
$password="xxxxxxx";

$sql="SELECT * FROM titles";

/* directly execute mode                        */
if ($conn_id=odbc_connect("$dsn",$user,$password)){
        echo "connected to DSN: $dsn<br><br>";
        if($result_id=odbc_do($conn_id, $sql)) {
                echo "executing '$sql'<br><br>";
                        $num_fields=odbc_num_fields($result_id);
                        if($num_fields>0){
                                echo "Number of fields:
$num_fields<br>";
                                for($i=1;$i<=$num_fields;$i++){

$field_name[$i-1]=odbc_field_name($result_id,$i);
                                }
                                $num_rows=0;
                                while(odbc_fetch_row($result_id)){
                                        for($i=1;$i<=$num_fields;$i++){

$result[$num_rows][$field_name[$i-1]]=odbc_result($result_id,$i);
                                        }
                                        $num_rows++;
                                }
                                echo "Number of rows: $num_rows<br>";
                        }else{
                                echo "not a field returned. <br><br>";
                        }
                echo "Results:<br>";
                for($i=0;$i<sizeof($result);$i++){
                        while(list($key,$value)=each($result[$i])){
                                echo "$i:$key=$value<br>";
```

```
                        }
                }
                echo "freeing result<br><br>";
                odbc_free_result($result_id);
        }else{
                echo "can not execute '$sql'<BR><BR>";
        }
        echo "closing connection $conn_id";
        odbc_close($conn_id);
}else{
        echo "can not connect to DSN: $dsn<br><br>";
}
?>
```

5. FAQ

- *Does this exact procedure work for MSSQL 7.0?*

  Yes, Openlink will upgrade their software to 7.0, but I don't know when.

- *I followed this document step by step to install the openlink drivers but I stopped at the "install ntadm65x.zip" step. After I installed ntadm65x I tried to start the OpenLink Request Broker and got the error:*

  ```
  unable to open the service control manager <5>
  press RETURN to exit oplrqb
  ```

  Could you please tell me how I can fix that problem?

  This should be a problem with NT itself. try restart the computer and then start openlink service from Control Panel/service manager.