**c++数据库访问OTL和DTL**

2010-10-07 20:13

OTL :http://otl.sourceforge.net/

DTL:http://dtemplatelib.sourceforge.net/

OTL例子：<http://otl.sourceforge.net/otl3_examples.htm>

OTL是一个纯C++的通用数据库连接模板库，可以支持各种当下流行的数据库，如Oracle，Sybase, MySQL, PostgreSQL, EnterpriseDB, SQLite, MS ACCESS, Firebird等等.它是一个跨平台类库，在MS Windows, Linux/Unix/Mac OS X 都可以使用。

OTL使用简单, 只要头文件中包含有: #include "otlv4.h" 就可，实际上整个OTL就一个.H的文件，使用起来极为的方便。

OTL使用方法：

1、首先指定要连接的数据库类型，OTL用宏定义来指定要连接的数据库类型。OTL会根据这个宏定义来初始化数据库连接的环境。

相关的宏定义列表 <http://otl.sourceforge.net/otl3_compile.htm>

如： #define OTL\_ORA8I 表示连接Oracle 8i 数据库。

------------------------------------------------------------------------------------

使用otl访问sql server

#include <iostream>  
using namespace std;  
  
#include <stdio.h>

#define OTL\_ODBC\_MSSQL\_2008 // Compile OTL 4.0/ODBC for MS SQL 2008  
#define [OTL\_MAP\_SQL\_GUID\_TO\_CHAR](http://otl.sourceforge.net/otl3_compile.htm#OTL_MAP_SQL_GUID_TO_CHAR)  
#include <otlv4.h> // include the OTL 4.0 header file  
  
[otl\_connect](http://otl.sourceforge.net/otl3_connect_class.htm) db; // connect object  
  
void insert()  
// insert rows into table  
{   
 [otl\_stream](http://otl.sourceforge.net/otl3_stream_class.htm) o(5, // buffer size  
 "insert into test\_tab values(:f1<int>,newid())",   
 // SQL statement  
 db // connect object  
 );  
  
 for(int i=1;i<=13;++i)  
 o<<i;  
}  
  
void select()  
{   
 [otl\_stream](http://otl.sourceforge.net/otl3_stream_class.htm) i(5, // buffer size  
 "select \* from test\_tab",  
 // SELECT statement  
 db // connect object  
 );   
 // create select stream  
  
 [otl\_stream](http://otl.sourceforge.net/otl3_stream_class.htm) o(7, // buffer size  
 "insert into test\_tab2 values(:f1<int>,:f2<char[37]>)",  
 // INSERT statement  
 db // connect object  
 );   
 // create insert stream  
 o.set\_commit(0); // turnin off the otl\_stream's autocommit flag  
   
 int f1;  
 char f2[37];  
  
 while(!i.eof()){ // while not end-of-data  
 i>>f1>>f2;  
 cout<<"f1="<<f1<<", f2="<<f2<<endl;  
 o<<f1<<f2;  
 }  
  
 o.flush(); // flushing the otl\_stream's buffer  
 db.commit(); // committing transaction  
  
}  
  
int main()  
{  
 [otl\_connect::otl\_initialize](http://otl.sourceforge.net/otl3_connect_class.htm)(); // initialize ODBC environment  
 try{  
  
 db.rlogon("UID=scott;PWD=tiger;DSN=mssql2008"); // connect to ODBC  
  
 [otl\_cursor::direct\_exec](http://otl.sourceforge.net/otl3_const_sql.htm) (  
 db,  
 "drop table test\_tab",  
 otl\_exception::disabled // disable OTL exceptions  
 ); // drop table  
  
 [otl\_cursor::direct\_exec](http://otl.sourceforge.net/otl3_const_sql.htm) (  
 db,  
 "create table test\_tab(f1 int, f2 uniqueidentifier)"  
 ); // create table  
  
 [otl\_cursor::direct\_exec](http://otl.sourceforge.net/otl3_const_sql.htm) (  
 db,  
 "drop table test\_tab2",  
 otl\_exception::disabled // disable OTL exceptions  
 ); // drop table  
  
 [otl\_cursor::direct\_exec](http://otl.sourceforge.net/otl3_const_sql.htm) (  
 db,  
 "create table test\_tab2(f1 int, f2 uniqueidentifier)"  
 ); // create table  
  
  
 insert(); // insert records into table  
 select(); // select records from test\_tab and insert them into test\_tab2  
  
 }  
  
 catch([otl\_exception](http://otl.sourceforge.net/otl3_exception_class.htm)& p){ // intercept OTL exceptions  
 cerr<<p.msg<<endl; // print out error message  
 cerr<<p.stm\_text<<endl; // print out SQL that caused the error  
 cerr<<p.sqlstate<<endl; // print out SQLSTATE message  
 cerr<<p.var\_info<<endl; // print out the variable that caused the error  
 }  
  
 db.logoff(); // disconnect from ODBC  
  
 return 0;  
  
}