

## OTL 编程配置

—作者: shuishanga

### 一、参考资料:

<http://blog.csdn.net/sherlockhua/article/details/4353531>

<http://www.xuebuyuan.com/1123237.html>

<http://blog.csdn.net/qianmianyuan/article/details/8891002>

### 二、配置原因

在编译 OTL 的程序时,需要使用其相应的数据库 API。这就要求程序在编译时链接 lib 库文件,不同的数据库对应的 lib 文件所在位置各不相同。

OTL 使用起来也很简单,使用不同的数据库连接,主要是根据程序开始的“宏定义”来指定的。OTL 是首先根据这个宏定义来初始化数据库连接环境。OTL 中用来区分连接方式的宏定义主要有下面这些:

OTL\_ORA7, OTL\_ORA8, OTL\_ODBC, OTL\_DB2\_CLI, OTL\_ODBC\_MYSQL...

不同的宏对应的数据库 API

下面表格分别是在 windows 与 Unix 下的数据库 API 所需要的头文件及 lib 文件所在的位置列表:

API	API header files for Windows	API libraries for Windows
OCI7	In <ORACLE_HOME>/oci/include	<ORACLE_HOME>/oci/lib/<compiler_specific>/ociw32.lib
OCI8	In <ORACLE_HOME>/oci/include	<ORACLE_HOME>/oci/lib/<compiler_specific>/oci.lib
OCI8i	In <ORACLE_HOME>/oci/include	<ORACLE_HOME>/oci/lib/<compiler_specific>/oci.lib
OCI9i	In <ORACLE_HOME>/oci/include	<ORACLE_HOME>/oci/lib/<compiler_specific>/oci.lib
OCI10g	In <ORACLE_HOME>/oci/include	<ORACLE_HOME>/oci/lib/<compiler_specific>/oci.lib
ODBC	Normally, in one of the C++ compiler system directories, no need to include explicitly.	Normally, in one of the C++ compiler system directories: odbc32.lib
DB2 CLI	In <DB2_HOME>/include	<DB2_HOME>/lib/db2api.lib <DB2_HOME>/lib/db2cli.lib

API	API header files for Unix	API libraries for Unix
OCI7	-I\$(ORACLE_HOME)/rdbms/demo -I\$(ORACLE_HOME)/rdbms/public	-L\$(ORACLE_HOME)/lib/ -lclntsh
OCI8	-I\$(ORACLE_HOME)/rdbms/demo -I\$(ORACLE_HOME)/rdbms/public	-L\$(ORACLE_HOME)/lib/ -lclntsh
OCI8i	-I\$(ORACLE_HOME)/rdbms/demo -I\$(ORACLE_HOME)/rdbms/public	-L\$(ORACLE_HOME)/lib/ -lclntsh
OCI9i	-I\$(ORACLE_HOME)/rdbms/demo -I\$(ORACLE_HOME)/rdbms/public	-L\$(ORACLE_HOME)/lib/ -lclntsh
OCI10g	-I\$(ORACLE_HOME)/rdbms/demo -I\$(ORACLE_HOME)/rdbms/public	-L\$(ORACLE_HOME)/lib/ -lclntsh
ODBC	ODBC bridge specific	ODBC bridge specific
DB2 CLI	-I<DB2_HOME>/sqlib/include	-L<DB2_HOME>/sqlib/lib -ldb2

### 三、VS2008 中具体配置

配置实例环境： 操作系统： win7 使用数据库： oracle10g

编程环境： VS2008

#### 1) 引入头文件

在 VS 中依次选择“工具→选项→项目和解决方案→VC++目录”，在右上方组合框中选择“包含文件”，把 oci 目录包含进来(如图 1)。例如我的是：  
D:\oracle\product\10.2.0\db\_1\oci\include。

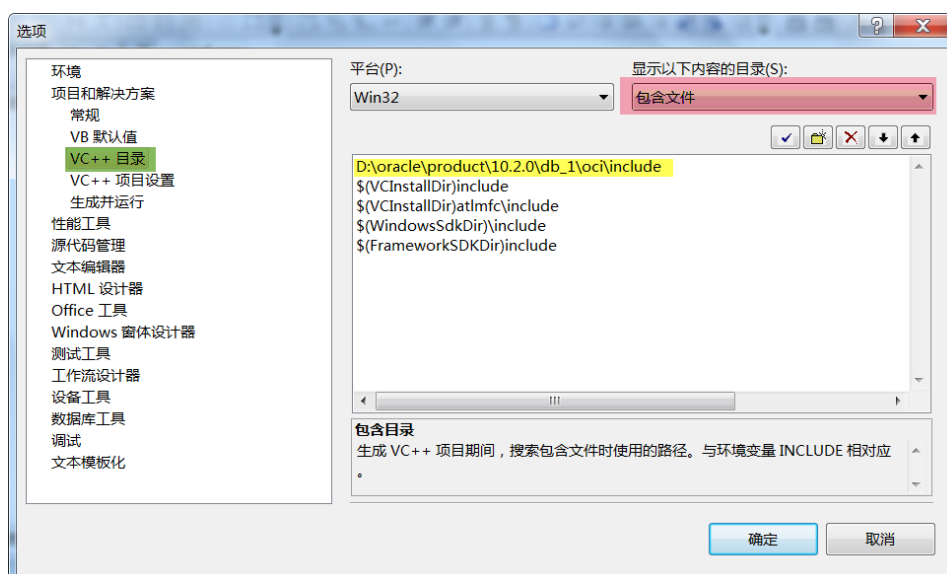


图 1 头文件引入

图 2 中给出了 oracle10g 中的 API 头文件。

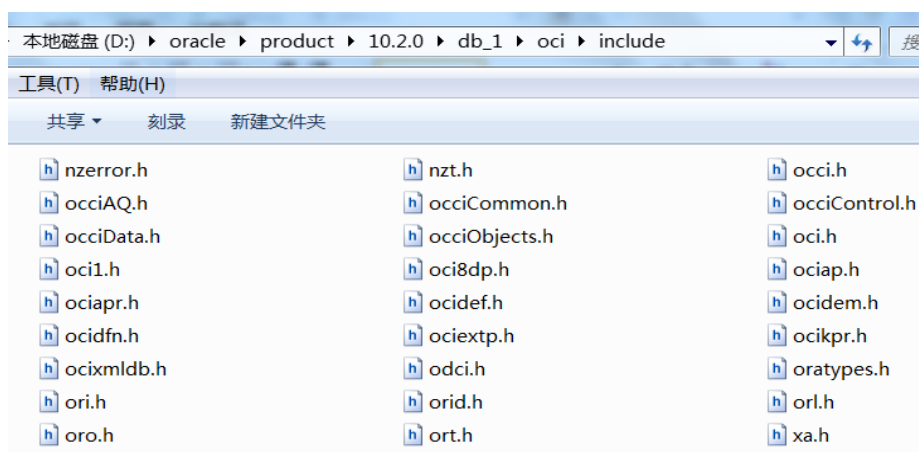


图 2 oracle10g 中的头文件

注：对于一个 OTL 程序而言，当未引入数据库 API 所需要的头文件时一般会出现“无法打开‘oci.h’的错误”，如下图 3：

```
fatal error C1083: 无法打开包括文件:“oci.h” : No such file or directory
as\as\Debug\BuildLog.htm
^ =====
```

图 3 未引入头文件错误信息

## 2) 引入库文件

由上面表格可以知道，oracle10g 所需要的库文件为：oci.lib。

在 VS 中依次选择“工具→选项→项目和解决方案→VC++目录”，在右上方组合框中选择“库文件”，添加库文件目录(如图 4)。例如我的是：  
D:\oracle\product\10.2.0\db\_1\oci\lib\msvc。

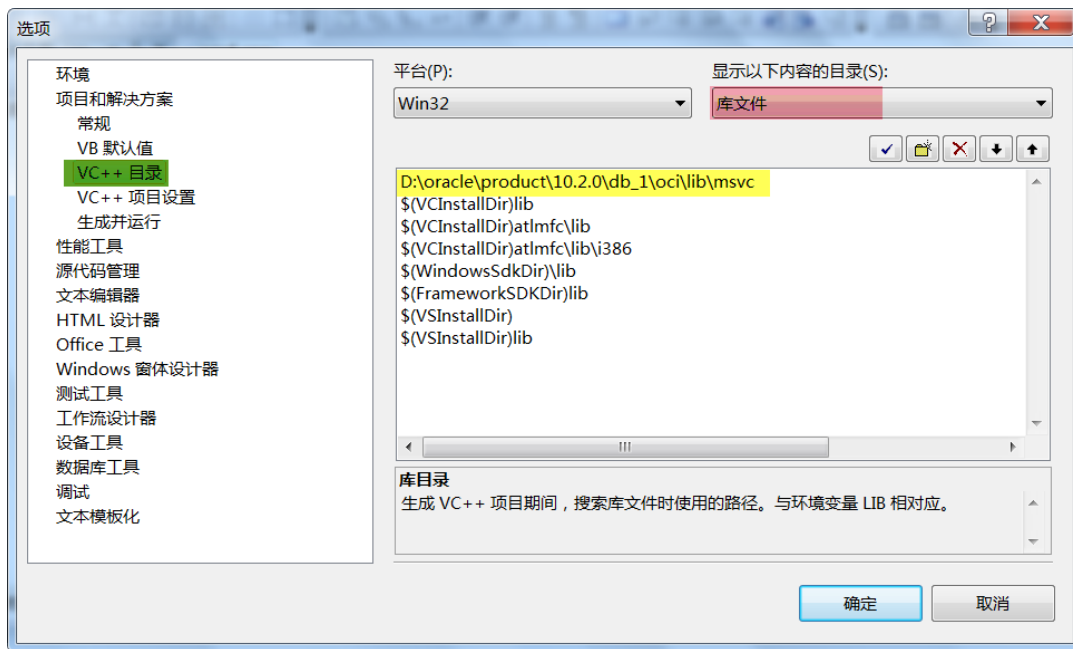


图 4 库文件引入

注意：这时并未配置完成，此时运行程序会出现“无法解析外部符号\*\*\*”的错误，如图 5 所示

```

1>----- 已启动生成: 项目: as, 配置: Debug Win32 -----
1>正在链接...
1>assd.obj : error LNK2019: 无法解析的外部符号 _OCIBindByName, 该符号在函数 "public:
int __thiscall otl_ref_cursor::first(void)" (?first@otl_ref_cursor@@QAEHXZ) 中被引用
1>assd.obj : error LNK2019: 无法解析的外部符号 _OCIAttrGet, 该符号在函数 "public:
unsigned int __thiscall otl_cur::rpc(void)" (?rpc@otl_cur@@QAEIXZ) 中被引用
1>assd.obj : error LNK2019: 无法解析的外部符号 _OCIStmtFetch, 该符号在函数 "public: int
__thiscall otl_cur::fetch(int, int &)" (?fetch@otl_cur@@QAEHHAH@Z) 中被引用
1>assd.obj : error LNK2019: 无法解析的外部符号 _OCIHandleFree, 该符号在函数 "public:
int __thiscall otl_conn::server_detach(void)" (?server_detach@otl_conn@@QAEHXZ) 中被引用
1>assd.obj : error LNK2019: 无法解析的外部符号 _OCIServerDetach, 该符号在函数 "public:
int __thiscall otl_conn::server_detach(void)" (?server_detach@otl_conn@@QAEHXZ) 中被引用

```

图 5 错误信息

### 3) 添加依赖项

在工程左侧项目上，右键单击选择“项目属性→配置属性”在弹出的属性页对话框中“连接器→输入”，在附加依赖项中填入“oci.lib”，如下图 6 所示。这时完成配置完毕。

若此步不配置，也可以直接在代码中添加

```
#pragma comment(lib, "oci.lib")
```

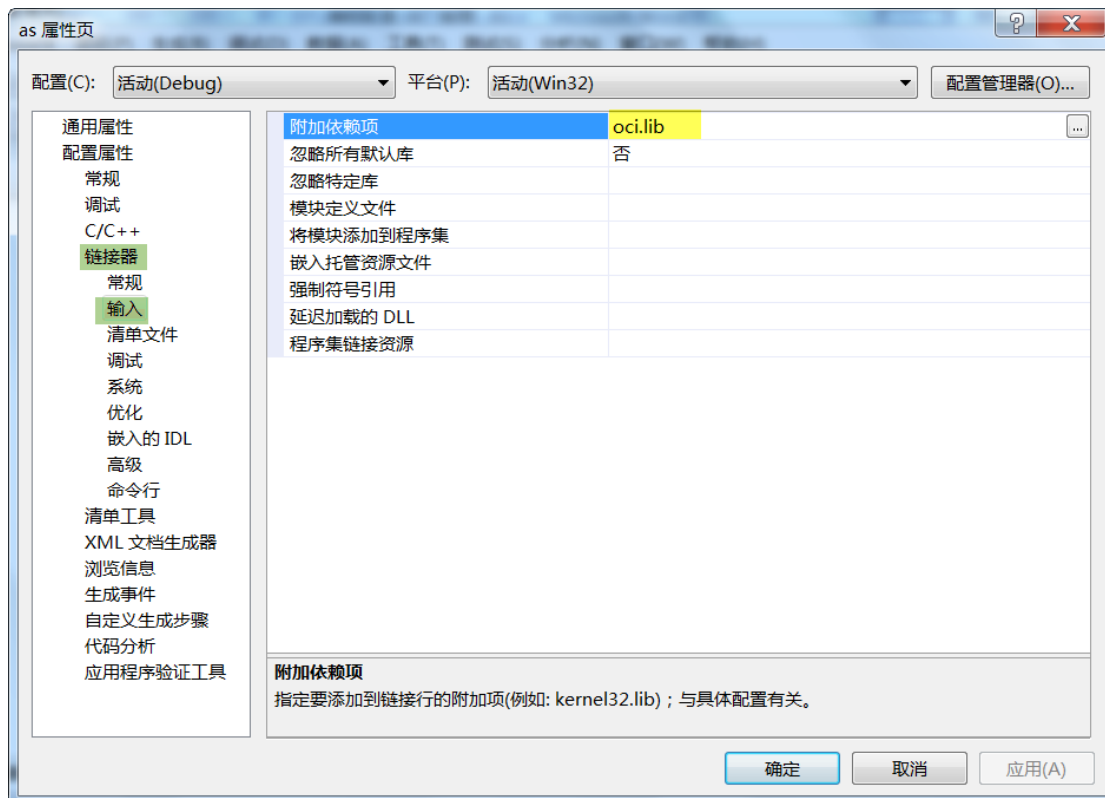


图 6 添加依赖性

#### 四、VS2010/2012 中具体配置

在 VS2010/2012 中，设置思路同 VS2008 是一样的，只不过在“工具->选项->项目和解决方案->VC++ 目录”不再提供设置，如下图：

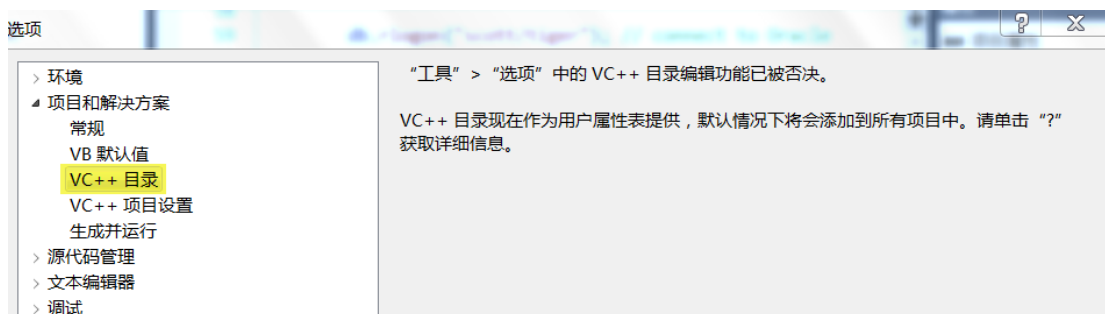


图 7 VS2010/2012 中 VC++目录

打开“属性管理器”视图，找到“[Microsoft.Cpp.Win32.user](#)”，双击打开，找到“[VC++ 目录](#)”

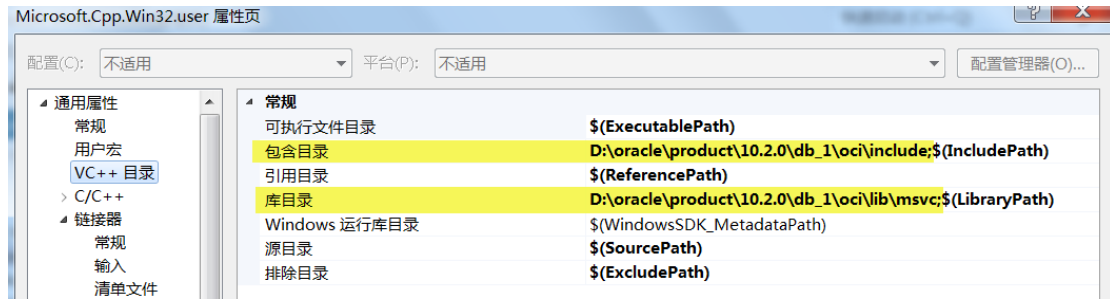


图 8 VS2010/2012 中头文件和库文件的引入