技 术 文 件

技术文件名称：IKEMBX00
数据库模块
使用说明
技术文件编号：
版　　本：　　0.1

拟 制 _____
审 核 _____
会 签 _____
_____
_____
标准化 _____
批 准 _____

中科虹霸科技有限公司

版本变更记录

文件编号	版本号	拟制人/修改人	拟制/修改日期	更改理由	主要更改内容 （写要点即可）
	0.1	何颖醛	2014-01-26	无	创建文档
	0.2	何颖醛	2014-02-25	修改文字上的小错误	

注 1：每次更改归档文件（指归档发布数据库）时，需填写此表。

注 2：文件第一次归档时，“更改理由”、“主要更改内容”栏写“无”。

目录

1 引言

1.1 编写目的

介绍 IKEMBX00 项目中使用的数据库模块的使用方法，以调用数据库模块实现需要的功能。

1.2 适用范围

适用于 IKEMBX00 项目数据库需求的开发和管理。

1.3 说明

文中的一些操作过程或代码是参考了网上一些文章，并在实际使用中总结的。如有错误，请尽快提出，以免影响使用。

2 术语、定义与缩略语

2.1 术语、定义

术语、定义见下表

术语/定义	说明
客户端	IKEMBX00 设备，总是作为客户端出现
服务器端	计算机上，与客户端进行网络通讯的程序

图表 1 术语、定义

2.2 缩略语

缩略语见下表。

缩略语	原文	中文含义
SOCI **SOCI** (['sokɪ])	Simple Open (Database) Call Interface	简明数据库访问接口 原为 Simple Oracle Call Interface

图表 2 缩略语

3 SOCI 简介

3.1 SOCI 特性

SOCI 是对数据库操作的封装库，采用标准 C++语言。

跨平台，可在 Linux 及 Windows 环境下使用。

目前支持数据库系统包括：

- DB2
- Firebird
- MySQL
- ODBC (通用访问方法)
- Oracle
- PostgreSQL
- SQLite3

使用较为简单方便，在所有数据库系统下，操作方式几乎一样。

不是并发安全的，但提供连接池方式解决部分问题。

开放源码。

网站为：http://soci.sourceforge.net/index.html，可了解更详细的信息。

http://slid.es/mloskot/soci，有 PPT 简介。

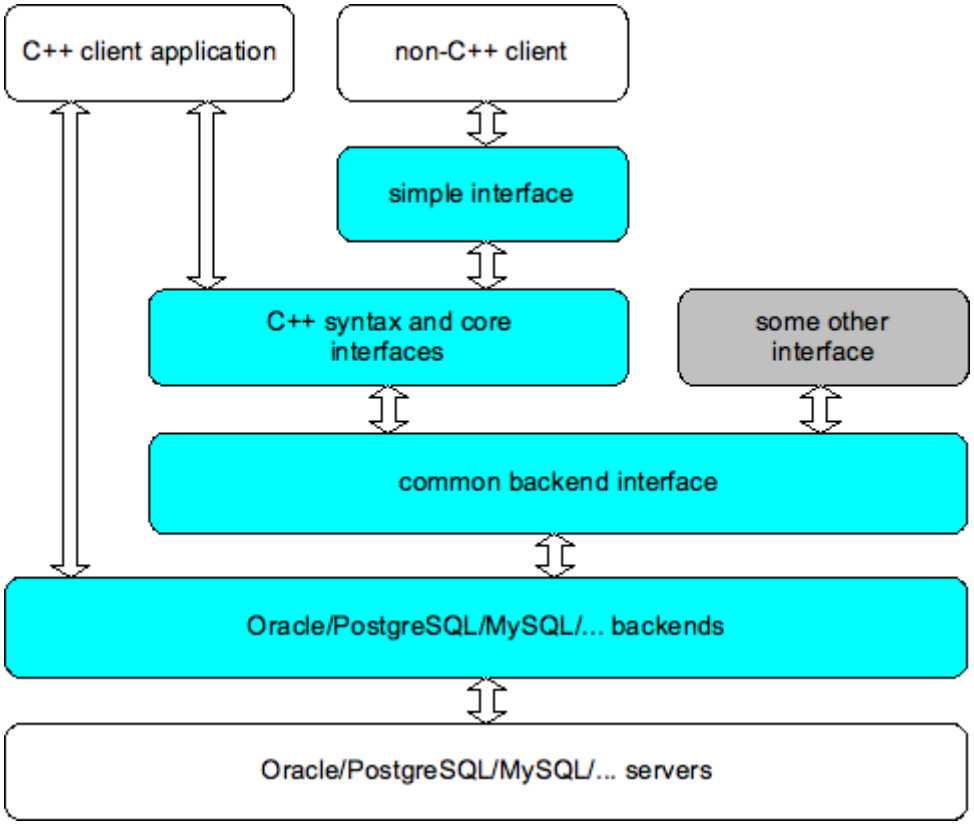
3.2 目前支持的特性

	Oracle	PostgreSQL	MySQL	SQLite3	Firebird	ODBC	DB2
Binding by Name	YES	YES (>=8.0)	YES	YES	YES	YES	YES
Dynamic Binding	YES	YES	YES	YES	YES	YES	

Bulk Operations	YES	YES	YES	YES	YES	YES	YES
Transactions	YES	YES	YES (with servers that support them, usually >= 4.0)	YES	YES	YES	YES
BLOB Data Type	YES	YES	MySQL's BLOB type is mapped to std::string	YES	YES	NO	NO
RowID Data Type	YES	YES	NO	NO	NO	NO	NO
Nested Statements	YES	NO	NO	NO	NO	NO	YES
Stored Procedures	YES	YES	NO (but stored functions, YES)	NO	YES	NO	YES

3.3 SOCI 结构

下图描述了 SOCI 的库结构：



图表 3 SOCI 库结构

图中涂了颜色的模块属于 SOCI 库提供的部分。

3.4 SOCI 缺点

代码更新不是非常及时，可以看出来从 2008-07 到 2011-10 期间没有更新，2013 年有三次更新。

3.5 与 SOCI 类似的其他选择

3.5.1 OTL

OTL 也是一个用 C++语言的，通用数据库连接模板库，支持 Oracle，Sybase， MySQL, PostgreSQL， EnterpriseDB， SQLite， MS ACCESS， Firebird。也是一个跨平台类库，在 Windows 和 Linux 下均可使用。也不是并发安全的。

可在http://otl.sourceforge.net/home.htm找到介绍及文档。OTL 与 SOCI 比较类似。

其好处是有很丰富的示例代码http://otl.sourceforge.net/otl3_examples.htm；最近更新很快。

但其代码的组成样子看起来不是很美观，所以没有选 OTL。

3.5.2 LiteSQL

也是对数据库操作的 C++封装，可以跨平台，支持 SQLite3、PostgreSQL、MySQL 和 Oracle。

不同之处是 LiteSQL 是个 ORM 框架，且不支持 Access，SQLServer。所以没有选用 LiteSQL。

对 LiteSQL 的介绍请参考http://sourceforge.net/apps/trac/litesql/。

3.6 为什么选择 SOCI

3.6.1 满足需求

IKEMBX00 项目在客户端部分，使用 Linux 系统，可能会用到 PostgreSQL 和 SQLite3 数据库；在服务器部分，使用 Windows 系统，可能会用到 PostgreSQL、SQLite3 和 SQLServer 数据库。

选择 SOCI，以上的这些需求都可以满足。

3.6.2 使用简单

SOCI 封装较为简单，使用方法简易明了。

由于 SOCI 提供的统一的接口，不论跨平台，还是使用不同种类的数据库系统，代码的编写方式都是类似的，节约学习成本。

3.6.3 简单代码示例

（以下代码来自http://slid.es/mloskot/soci）

连接数据库示例：

```
using namespace soci;

session sql([BACKEND], [CONNECTION STRING]);
```

其中的[BACKEND]　　指的是使用哪种数据库，[CONNECTION STRING]指的是连接字符串。

具体来说，比如用户’skippy’连接 test 数据库，数据库类型为 postgresql：

```
string connstr("dbname=test user=skippy");

// uses factory object provided

session sql(postgresql, connstr);
```

实际连接的语句可以有几种

删除 employee 表中，一条 id 为 7 的记录示例：

```
session sql;

string table_name("employee");

int id = 7;

sql << "DELETE FROM "

    << table_name

    << " WHERE id="

    << id;
```

查询记录示例：

```
string name; // output variable

int id = 7;  // input variable

sql << "SELECT name FROM person
```

```
<< WHERE id=:i",
into(name),
use(id);
```

4 SOCI 在 Ubuntu 系统的安装

请访问：http://soci.sourceforge.net/doc/3.2/installation.html获得更详细信息。

4.1 下载 SOCI

从 Git 仓库克隆 SOCI 代码：

```
git clone git://github.com/SOCI/soci.git
```

（克隆前，需要现在 Ubuntu 系统中安装 Git，可参见“Git 服务器使用说明.docx”）

例如，在 home 目录下建了一个 gitTest 目录，并以 gitTest 为当前目录，运行上述指令。如果正确执行，结果是在 gitTest 目录下，自动创建了一个 soci 目录，SOCI 的代码都已克隆到此目录下。

4.2 编译及安装 SOCI

SOCI 的库分为两大部分，一部分是 SOCI 的核心(core)库，一部分是每种数据库系统的后端(backend)库。核心库必须编译；后端库不需要全部编译，而是根据使用需求，分别编译。在这里，针对 IKEMBX00 的需求，写出编译 PostgreSQL 和 SQLite3 的过程。

4.2.1 编译 SOCI 核心库

（假定现在已经在克隆后的 soci 目录下）

```
$ cd build
```

（切换到 build 目录）

```
$ cmake -G “Unix Makefiles” -DWITH_BOOST=OFF -DWITH_ORACLE=OFF ../src
```

（注意大小写，按照上面的格式。

这个过程实际上是为后面 SOCI 的编译做配置。这个过程中自动发现当前 Ubuntu 系统

中已经安装了什么数据库系统。发现了什么数据库系统，就会配置相应的数据库系统。因此，这一步执行完毕后，请检查一下输出。在输出比较靠后的位置，查询到的数据库系统会有一行绿色的文字，例如用绿色显示”PostgreSQL – SOCI backend for PostgreSQL database engine”，表明发现了系统中安装的 PostgreSQL，并在配置文件中配置了相应内容。

如果希望使用某种数据库系统，但没有在后面发现相应的绿色文字，请查看后面关于安装数据库系统的章节。目前提供 PostgreSQL 和 SQLite3 的安装过程。）

```
$ make
```

（编译 SOCI 核心库连接）

```
$ sudo make install
```

（把核心库连接及头文件拷贝到/usr 目录下。如果权限不够，不能拷贝，因此加上 sudo）

```
$ cd ../src/core
```

（切换到核心文件所在目录）

```
$ make -f Makefile.basic
```

（编译核心库。一般来说，编译第一次时，编译到 backend_loader.cpp 时，可能会报错，提示找不到 soci_backends_config.h。实际上此文件存在，只不过编译时不能找到它的位置而已。解决办法是打开 backend_loader.cpp 文件，找到其中的

#include “soci_backend_config.h”

这样一行，修改为

#include “../../build/core/soci_backend_config.h”

保存该文件后重新编译，就可以通过了。）

至此，核心库编译完毕，之后根据需要编译后端库。本文给出编译 PostgreSQL 和 SQLite3 的过程。

4.2.2 编译 PostgreSQL 后端库

（假定现在已经在 src/core 目录下）

```
$ cd ../backends/postgresql
```

（切换到 PostgreSQL 后端库代码目录）

```
$ make -f Makefile.basic
```

（编译后端库。一般来说，编译第一次时，会提示 libpg-fe.h 找不到。这是 PostgreSQL 的库，在/usr/include/postgresql 目录下。

因此，修改 Makefile.basic，找到

PGSQLINCLUDEDIR= -I/usr/include，后面加上/postgresql，使这一行变为

PGSQLINCLUDEDIR= -I/usr/include/postgresql

即可编译通过。）

4.2.3 编译 SQLite3 后端库

（假定现在已经在 backends/postgresql 目录下）

```
$ cd ../sqlite3
```

（切换到 sqlite 后端库代码目录）

```
$ make -f Makefile.basic
```

（编译。一般第一次即可编译成功）

5 使用 SOCI 及示例

5.1 使用 SOCI

使用 SOCI，头文件查找路径需要有：

/usr/local/include/soci（都必须的）

/usr/include/postgresql（postgresql 需要的）

头文件需要有：

#include <soci.h> （都必须的）

#include <postgresql/soci-postgresql.h> （postgresql 需要的）

#include <sqlite3/soci-sqlite3.h> （sqlite3 需要的）

库文件需要有：

soci-core （核心库）

soci-postgresql （postgresql 需要的）

sqlite3 （sqlite3 需要的）

dl （？？不知道是什么）

5.2 示例

SOCI 的使用示例在 IKEMBX00 项目 Git 库中，文件位置为：

Code/databaseLib/examples

目前示例包括：

1)连接本地和远程的 postgresql 数据库；

2)postgresql 数据库数据读写；

3)SQLServer 数据库读（Windows 环境下）；

4)SQLite 数据库读；

5)从 postgresql 数据库读出数据写入到 SQLite 数据库；

示例的编译方法是，在每个具体示例所在目录下，输入

```
$ make
```

即编译形成可执行文件，可执行文件在示例所在的目录下。

6 数据库在 Ubuntu 系统的安装

6.1 安装 PostgreSQL

6.1.1 安装

```
$ sudo apt-get install -y postgresql-9.1 postgresql-client-9.1 postgresql-contrib-9.1 postgresql-server-dev-9.1
```

（下载并安装以上这些模块。如果提示哪个模块由于什么原因不能安装什么的，就把它们拆开来，一个一个地下载并安装）

```
$ sudo -u postgres psql
```

（进入用户 postgres 的 psql 提示符，之后要修改用户 postgres 在 postgresql 数据库系统中的密码。因为刚安装后，postgresql 系统中有个缺省用户 postgres，但是这个用户是无密码的。所以安装完毕后，一定要设置密码。

上面那行输入回车后，正常情况，会进入 psql 提示符，提示符的形式为“postgres=#”）

修改密码有两种方式，建议使用第一种方式，因为可以进行密码验证。

第一种方式：

在 psql 提示符后面输入

```
\password
```

（注意是反斜杠，且文字为小写）

```
Enter new password:
```

```
Enter it again:
```

（以上为系统输出提示，要求输入两遍相同的密码）

第二种方式：

在 psql 提示否后面输入

```
ALTER USER postgres WITH  PASSWORD '123456';
```

（以上是在 psql 提示符后输入的内容，表示把用户 postgres 的密码修改为 123456，注意使用半角单引号把密码括起来）

不管用以上哪种方式，密码修改完成后，输入

```
\q
```

（以上是在 psql 提示符后输入的内容，表示退出 psql）

可以考虑安装 pgAdmin，是对 PostgreSQL 的图形化管理工具，比较方便。通过软件中心安装即可。

6.1.2 修改配置

PostgreSQL 安装完成后，需要修改配置。第一是为了配合 SOCI 编程，第二是为了能远程访问。

6.1.2.1 修改配置文件 pg_hba.conf

```
$ sudo vi  /etc/postgresql/9.1/main/pg_hba.conf
```

（编辑配置文件 pg_hba.conf）

1、找到 local all postgres peer 这一行，把 peer 修改为 trust，即这一行改为：

local all postgres trust

如果在数据库使用中出现问题，也可以尝试把 trust、peer 等都修改为 md5，表示需要密码。

（不改这个地方，以后编程连接本地数据库时，总会出现错误“FATAL: Peer authentication failed for user ‘posgres’”）

2、在文件最后增加一行

host all all 0.0.0.0/0 md5

（为了能够远程连接本机的 PostgreSQL 数据库）

3、保存文件并退出

6.1.2.2 修改配置文件 postgresql.conf

```
$ sudo vi /etc/postgresql/9.1/main/postgresql.conf
```

（编辑配置文件 postgresql.conf）

找到 listen_addresses=”…”的一行，有可能第一次安装后这里的文字为：

listen_addresses=”localhost”。把它修改为

listen_addresses=”*”

（”localhost”表示只在本机监听；”*”表示在所有地址监听；”localhost, 10.2.1.117”表示在本机和 10.2.1.117 的地址上监听）

6.1.2.3 重启 PostgreSQL 服务

以下操作选择任意一个即可。

```
$ sudo /etc/init.d/postgresql restart
```

（重启）

还有一种重启命令为

```
$ sudo service postgresql restart
```

（重启，两种重启是一样的效果）

```
$ sudo /etc/init.d/postgresql reload
```

（重新装载，估计是重新装载配置）

按照上面的方式修改配置文件后，必须要重启或重新装载，新的配置才能生效

6.2 安装 SQLite3

6.2.1 安装

```
$ sudo apt-get install sqlite sqlite3
```

（安装 SQLite3）

```
$ sudo  apt-get  install  libsqlite3-dev
```

（安装需要的工具包）

可考虑安装 sqlitebrowser 图形工具界面。

6.2.2 检查安装结果

```
$ sqlite3 test.db
```

（进入 SQLite 提示行模式，提示符为’sqlite>’）

在提示符后输入

.database

（检查数据库）

.exit

（退出 SQLite 提示行模式）

退出后，在当前目录下有文件 test.db。该文件无用，可删除。