



统一数据框架 DSCI

使用手册

(2.0.0.0)

北京思特奇信息技术股份有限公司
二〇一六年十二月

目 录

1. 概述	4
2. 安装与配置	5
2.1. 安装	5
2.2. 配置	5
3. 函数接口	7
3.1. 编译说明	7
3.2. C 接口	7
3.2.1. DSCI_ver()	7
3.2.2. DSCI_init()	8
3.2.3. DSCI_init_direct()	8
3.2.4. DSCI_set_auto_commit()	8
3.2.5. DSCI_set_log_mode()	9
3.2.6. DSCI_end_trans()	9
3.2.7. DSCI_query()	9
3.2.8. DSCI_query_f()	9
3.2.9. DSCI_prepare_sql()	10
3.2.10. DSCI_prepare_sql_f()	10
3.2.11. DSCI_bind_param()	10
3.2.12. DSCI_bind_blob_param()	11
3.2.13. DSCI_bind_clob_param()	11
3.2.14. DSCI_query_p()	12
3.2.15. DSCI_free_sql()	12
3.2.16. DSCI_num_rows()	12
3.2.17. DSCI_quit()	12
3.2.18. DSCI_ecode()	12
3.2.19. DSCI_error()	13
3.2.20. DSCI_set_Buffer_flag()	13
3.2.21. DSCI_num_fields()	13
3.2.22. DSCI_fetch_row()	13
3.2.23. DSCI_fetch_row_s()	14
3.2.24. DSCI_lob_size()	14
3.2.1. DSCI_lob_getp	14
3.2.2. DSCI_lob_get()	14
3.2.1. DSCI_lob_isclob	15
3.3. C++类接口	15
3.3.1. 工厂方法: DSCI_MakeConnection	15

3.3.2. DSCIConnection 类	15
3.3.3. DSCIStatement 类.....	16
3.3.4. DSCIPreparedStatement 类.....	16
3.3.5. DSCIResultSet 类	16
 4. 附录 A（DSCI 错误代码定义）	 18
 5. 附录 B（编程示例）	 19

1. 概述

本通用接口实现对内存数据库（ProMDB/Timesten/Altibase）、分布式内存数据库（ProDMDB）、磁盘数据库(Oracle/Mysql) 等的接口访问的统一封装，应用程序通过 API 函数调用访问各类数据，屏蔽应用程序对数据存储模式的依赖，使应用处理层与数据管理层实现有效分离。应用程序无需关心具体每张表的数据存放在什么数据库上，只需要发送标准的 SQL 语句，即可得到对应的操作结果。使得应用程序的设计可以完全抛开不同数据库处理设计，达到程序设计与数据库无关的目的，所有数据库的操作 API 相同，简化了应用代码，方便维护、实现和数据的移植。

2. 安装与配置

2.1. 安装

将安装包 DSCI.tar 解压到某个目录，比如：/home/work/DSCI

在安装用户的 profile 文件中添加

➤ 非 IBM 主机：

```
export DSCI_PATH=你的安装路径
export LD_LIBRARY_PATH=$DSCI_PATH/lib:$LD_LIBRARY_PATH
```

➤ IBM 主机：

```
export DSCI_PATH=你的安装路径
export LIBPATH=$DSCI_PATH/lib:$LIBPATH
```

重新登录，cd \$DSCI_PATH/src，执行：sh make_all 完成编译安装。

2.2. 配置

通用接口库有自己专用的配置信息，用于配置各类数据库信息，各表的存储位置等。
函数内部根据表名确定应该到哪个库上去执行 SQL 语句

1. 数据库配置(\$DSCI_PATH/cfg/db.cfg)：

参数	说明
[SYS]	
DEFAULT_DB=DB02	如果某张表没有配置，则认为此表在默认数据库中
[DB??]	配置每个数据库的信息，序号从 01 开始，确保连续
login = ./cfg/login.db	本数据库的用户名密码的加密文件，请使用 \$DSCI_PATH/lib/DSCI_tools -p 生成；参数支持环境变量，例如 \${DSCI_PATH}/cfg/login.db
Server=bilddb	配置数据库名： Oracle ：填 tnsnames 中配置的连接串 iMDB ：填 net.cfg 文件中配置的连接串 DMDB ：填 net.cfg 文件中配置的连接串 Altibase ：填 IP 地址:端口号:连接类型:altibase_home 的目录： 例如 172.21.35.214:20300:2:/alt02/altibase_home 其中连接类型 1:IPC 方式；2:unix 域方式；3: socket 方式 MySQL ：填库名称:IP 地址:端口号：例如 mysql:172.21.35.214:3306

Type=oracle	数据库类型，目前可填：oracle、imdb、dmdb、altibase、mysql
-------------	--------------------------------------------

示例：

```
[SYS]
DEFAULT_DB = DB01    #配置默认数据库

[DB01] #序号从 01 开始，确保连续
login = ./cfg/login.db
Server=bilddb
Type=oracle

[DB02]
.....
```

2. 表信息配置(\$DSCI_PATH/cfg/table.cfg):

[TAB_DB]	配置每个表存放在哪个库上
mytest=DB01 tab2=DB02 bbb=DB01	等号左边是表名，右边为数据库名，注意，数据库名必须为 db.cfg 中存在的库。

示例：

```
[TAB_DB]
mytest=DB01
tab2=DB02
bbb=DB01
```

说明，如果某张表没有配置，则认为此表在默认数据库中（对应 db.cfg 中的 DEFAULT_DB 参数）。

3. 函数接口

3.1. 编译说明

■ C 语言接口

库文件: \$DSCI_PATH /lib/libdsci.so

头文件: \$DSCI_PATH /include/dsci.h

INCLUDE 中增加 -I\${DSCI_PATH}/include

LIB 中增加 -L\${DSCI_PATH}/lib/ -ldsci

■ C++ 接口

库文件: \$DSCI_PATH /lib/libdscpp.so

头文件: \$DSCI_PATH /include/dscpp.h

INCLUDE 中增加 -I\${DSCI_PATH}/include

LIB 中增加 -L\${DSCI_PATH}/lib/ -ldscpp

3.2. C接口

结构体	说明
DSCI	表示一个连接数据库的句柄, 被用于几乎所有的 DSC 函数, 每一个连接都需要预先定义一个这样的变量。
DSCI_ROW	字符串数组。用于获取数据行信息。select 时使用
DSCI_SQL	存放解析后的 SQL 信息, SQL 语句只有先解析再使用, 才可能获得较高的性能。写程序时, 需要事先看看本程序总共需要用到多少个 SQL, 然后就定义相应个这样的变量, 以便后续使用。

3.2.1. DSCI_ver()

函数原型	void DSCI_ver(char *version, char *r_date)
功能说明	查看 DSCI 接口函数库的版本号
参数说明	char *version : 版本号 char *r_date : 发布日期

返回值	无
备注	

3.2.2. DSCI_init()

函数原型	int DSCI_init(char *errStr, DSCI *pDSC)
功能说明	初始化 DSCI 结构体； 注意，如果采用多线程连接，需要每个线程对应一个 DSCI 结构体，并分别初始化。
参数说明	char *errStr: 如果出错，错误信息存放在此参数中； DSCI *pDSC: DSCI 结构体指针。
返回值	成功：返回 0，失败：返回-1
备注	对于每一个 DSCI 变量，此函数只在程序开始时执行一次即可

3.2.3. DSCI_init_direct()

函数原型	int DSCI_init_direct(char *errStr, DSCI *pDSC, char *dbStr)
功能说明	不使用配置文件，直接在参数中指定数据库用户名密码类型等信息。 初始化 DSCI 结构体； 注意，如果采用多线程连接，需要每个线程对应一个 DSCI 结构体，并分别初始化。
参数说明	char *errStr: 如果出错，错误信息存放在此参数中； DSCI *pDSC: DSCI 结构体指针。 char *dbStr: 数据库信息，格式为：“用户名/密码@数据库类型:数据库名”，其中数据库类型参见 db.cfg 中的 Type 配置，数据库名参见 db.cfg 中的 Server 配置，例如： user1/mypass@oracle:billdb user1/mypass@altibase: 172.21.35.214:20300:2:/altio2/altibase_home
返回值	成功：返回 0，失败：返回-1
备注	如果应用程序只访问一个数据库，可以不配置 db.cfg 和 table.cfg，直接在 dbStr 参数中指定数据库信息即可，如果使用了本函数，则不再需要调用 DSCI_init，具体示例参见：\$DSCI_PATH/demo/onedb/的示例程序。

3.2.4. DSCI_set_auto_commit()

函数原型	int DSCI_set_auto_commit(DSCI *pDSC, int flag)
功能说明	设置连接是否自动提交
参数说明	DSCI *pDSC: DSCI 结构体指针 int flag: 0 非自动提交，1 自动提交。
返回值	成功：返回 0，失败：返回-1
备注	如果设成自动提交，每条 SQL 成功执行后，都将自动提交，不需要再执行 commit 操作。

3.2.5. DSCI_set_log_mode()

函数原型	void DSCI_set_log_mode(DSCI *pDSC, const char *inMode)
功能说明	设置日志模式
参数说明	DSCI *pDSC: DSCI 结构体指针 char *inMode: on 写日志, off 不写日志, sync 实时写磁盘。
返回值	无
备注	

3.2.6. DSCI_end_trans()

函数原型	int DSCI_end_trans(DSCI *pDSC, int flag)
功能说明	提交或者回滚当前事务
参数说明	DSCI *pDSC: DSCI 结构体指针 int flag: 0 提交, 1 回滚
返回值	成功: 0, 失败: -1
备注	

3.2.7. DSCI_query()

函数原型	int DSCI_query(DSCI *pDSC, const char *sqlStr)
功能说明	向数据库发送 SQL 语句请求, 并接收返回结果;
参数说明	DSCI *pDSC: DSCI 指针 const char *sqlStr: SQL 语句(如, SELECT, INSERT 等)
返回值	成功: 返回 0, 失败: 返回-1
备注	此函数仅适用于执行那些操作不频繁的 SQL, 如果是频繁的操作, 不要使用此函数。

3.2.8. DSCI_query_f()

函数原型	int DSCI_query_f(DSCI *pDSC, const char *sqlStr, int dbNo)
功能说明	向数据库发送 SQL 语句请求, 并接收返回结果;
参数说明	DSCI *pDSC: DSCI 指针 const char *sqlStr: SQL 语句(如, SELECT, INSERT 等) int dbNo: 指定数据库序号, 从 1 开始, 1 对应 DB01, 2 对应 DB02, 依次类推, 表示直接根据此序号到对应的数据库上去执行, 不再读取表配置信息; 如果此参数为 0, 则等同于 DSCI_query 函数。
返回值	成功: 返回 0, 失败: 返回-1

备注	此函数仅适用于执行那些操作不频繁的 SQL，如果是频繁的操作，不要使用此函数。
----	-----------------------------------------

3.2.9. DSCI_prepare_sql()

函数原型	int DSCI_prepare_sql(DSCI *pDSC, DSCI_SQL *pSQL, const char *sqlStr)
功能说明	预解析 SQL 语句
参数说明	DSCI *pDSC: DSCI 指针 DSCI_SQL *pSQL: 存放 SQL 解析结果,以便后续的 DSCI_query_p 使用。 const char *sqlStr: SQL 语句(如, SELECT, INSERT 等),可带参数,例如: update tab set a=a+? where b=? and c=?
返回值	成功: 返回 0, 失败: 返回-1
备注	注意, 1、在首次对“*pSQL”做预解析时,请务必先对此变量做 memset ; memset 只能做一次,如果断线重连,重新预解析时,则不能 memset 。 2、如果其他函数报错,并且错误码大于 0,则说明数据库连接断了,那么先前所有做过预解析的 SQL 语句都会失效,此时需要重新调用 DSCI_prepare_sql 进行 SQL 预解析。

3.2.10. DSCI_prepare_sql_f()

函数原型	int DSCI_prepare_sql_f(DSCI *pDSC, DSCI_SQL *pSQL, const char *sqlStr, int dbNo)
功能说明	预解析 SQL 语句
参数说明	DSCI *pDSC: DSCI 指针 DSCI_SQL *pSQL: 存放 SQL 解析结果,以便后续的 DSCI_query_p 使用。 const char *sqlStr: SQL 语句(如, SELECT, INSERT 等),可带参数,例如: update tab set a=a+? where b=? and c=? int dbNo: 指定数据库序号,从 1 开始,1 对应 DB01,2 对应 DB02,依次类推,表示直接根据此序号到对应的数据库上去执行,不再读取表配置信息;如果此参数为 0,则等同于 DSCI_prepare_sql 函数。
返回值	成功: 返回 0, 失败: 返回-1
备注	注意事项, 同 DSCI_prepare_sql

3.2.11. DSCI_bind_param()

函数原型	int DSCI_bind_param(DSCI *pDSC, DSCI_SQL *pSQL, int param_id, char data_type, void *param)
功能说明	为预解析的 SQL 绑定参数,以便后续执行

参数说明	DSCI *pDSC: DSCI 指针 DSCI_SQL *pSQL: 事先已经解析好的 SQL 信息 int param_id: 参数序号, 从 0 开始算, 如 0, 1, 2, char data_type: 参数类型 void *param: 参数值对应的指针
返回值	成功: 返回 0, 失败: 返回-1
备注	解析后的 SQL 有多少个参数, 则本函数需要调用多少次。 参数类型包括: DSCI_TYPE_CHAR, DSCI_TYPE_DATE, DSCI_TYPE_INT, DSCI_TYPE_LONG, DSCI_TYPE_FLOAT, DSCI_TYPE_DOUBLE。 如果是日期类型, 使用 14 位字符串, 如 20080808000000

3.2.12. DSCI_bind_blob_param()

函数原型	int DSCI_bind_blob_param(DSCI *pDSC, DSCI_SQL *pSQL, int param_id, void *param, int size)
功能说明	如果 SQL 中某个字段为 BLOB 字段, 则调用此函数绑定变量
参数说明	DSCI *pDSC: DSCI 指针 DSCI_SQL *pSQL: 事先已经解析好的 SQL 信息 int param_id: 参数序号, 从 0 开始算, 如 0, 1, 2, void *param: 参数值对应的指针 int size: 待绑定变量的字节数
返回值	成功: 返回 0, 失败: 返回-1
备注	

3.2.13. DSCI_bind_clob_param()

函数原型	int DSCI_bind_clob_param(DSCI *pDSC, DSCI_SQL *pSQL, int param_id, void *param)
功能说明	如果 SQL 中某个字段为 CLOB 字段, 则调用此函数绑定变量
参数说明	DSCI *pDSC: DSCI 指针 DSCI_SQL *pSQL: 事先已经解析好的 SQL 信息 int param_id: 参数序号, 从 0 开始算, 如 0, 1, 2, void *param: 参数值对应的指针
返回值	成功: 返回 0, 失败: 返回-1
备注	本函数仅支持 imdb 和 dmdb

3.2.14. DSCI_query_p()

函数原型	int DSCI_query_p(DSCI *pDSC, DSCI_SQL *pSQL)
功能说明	执行解析后（并且参数已绑定）的 SQL 命令
参数说明	DSCI *pDSC: DSCI 指针 DSCI_SQL *pSQL: 事先已经解析好的 SQL 命令
返回值	成功: 返回 0, 失败: 返回-1
备注	

3.2.15. DSCI_free_sql()

函数原型	void DSCI_free_sql(DSCI_SQL *pSQL)
功能说明	释放预解析的 SQL
参数说明	DSCI_SQL *pSQL: 存放 SQL 解析结果的指针
返回值	无
备注	

3.2.16. DSCI_num_rows()

函数原型	int DSCI_num_rows(DSCI *pDSC)
功能说明	获取 SQL 操作成功后的记录数。
参数说明	DSCI *pDSC: DSCI 指针
返回值	返回值: 增删改操作成功的记录数, 或者 select 查询到的记录总数（仅在缓冲模式下有效）
备注	

3.2.17. DSCI_quit()

函数原型	void DSCI_quit(DSCI *pDSC)
功能说明	退出数据库, 释放连接。
参数说明	DSCI *pDSC: DSCI 指针
返回值	无

3.2.18. DSCI_ecode()

函数原型	int DSCI_ecode(DSCI *pDSC)
功能说明	如果 SQL 执行失败, 通过本函数获取错误码。
参数说明	DSCI *pDSC: DSCI 指针

返回值	错误码
-----	-----

3.2.19. DSCI_error()

函数原型	char *DSCI_error(DSCI *pDSC)
功能说明	展如果 SQL 执行失败，通过本函数获取错误信息(错误编号、DSC 错误信息、系统错误信息等)。
参数说明	DSCI *pDSC: DSCI 指针
返回值	错误信息字符串

3.2.20. DSCI_set_Buffer_flag()

函数原型	void DSCI_set_Buffer_flag(DSCI *pDSC, int flag)
功能说明	设置是否采用 缓冲模式 ，用于 SELECT 操作；数据库连接成功后默认是非缓冲模式。 缓冲模式 ，是将 select 结果集数据驻留在客户端内存缓冲区内。适用一次获取所有结果集数据。
参数说明	DSCI *pDSC: DSCI 指针 int flag: 1 是缓冲模式， 0 非缓冲模式
返回值	无
备注	如果需要此种模式，在 DSCI_init 后，执行一次本函数即可。在此模式下，执行 select 操作后，可以使用 DSCI_num_rows 函数获取 select 查询到的记录总数。

3.2.21. DSCI_num_fields()

函数原型	int DSCI_num_fields(DSCI *pDSC)
功能说明	获取查询结果集字段个数。
参数说明	DSCI *pDSC: DSCI 指针
返回值	select 的结果集字段个数
备注	

3.2.22. DSCI_fetch_row()

函数原型	DSCI_ROW DSCI_fetch_row(DSCI *pDSC)
功能说明	从结果集中，读取一条查询记录
参数说明	DSCI *pDSC: DSCI 指针
返回值	成功：返回 (DSCI_ROW) 指针 到达记录集尾部（无记录）：返回 NULL

备注	select 执行成功后，即可循环调用此函数，获取每一条查询结果，如果返回 NULL，则说明后续没有记录了。
----	--------------------------------------------------------

3.2.23. DSCI_fetch_row_s()

函数原型	DSCI_ROW DSCI_fetch_row_s(DSCI *pDSC, DSCI_SQL *pSQL)
功能说明	从结果集中，读取一条查询记录
参数说明	DSCI *pDSC: DSCI 指针 DSCI_SQL *pSQL: 事先已经解析好的 SQL 命令
返回值	成功: 返回 (DSCI_ROW) 指针 到达记录集尾部 (无记录): 返回 NULL
适用范围	SELECT 查询语句
备注	select 执行成功后，即可循环调用此函数，获取每一条查询结果，如果返回 NULL，则说明后续没有记录了。 如果是多个 select 语句嵌套查询，请务必使用本函数。

3.2.24. DSCI_lob_size()

函数原型	int DSCI_lob_size(char *pData)
功能说明	获取 LOB 字段的字节长度
参数说明	char *pData: select 得到的二进制信息流的指针
返回值	返回值: lob 字段存储信息的长度，如果为 0，说明是空值
备注	执行 DSCI_fetch_row 后，如果某个字段是 lob 字段，则使用此函数获取此字段的长度，例如: len = DSCI_lob_size(v_row[i]);

3.2.1. DSCI_lob_getp

函数原型	char *DSCI_lob_getp(char *p_DATA)
功能说明	获取 LOB 字段实际内容的首指针
参数说明	char *p_DATA: select 得到的二进制信息流的指针
返回值	返回值: LOB 字段实际内容的首指针。
适用范围	执行 DSCI_fetch_row 后，如果某个字段是 lob 字段，则使用此函数获取此字段内容的首指针，例如: char *p = DSCI_lob_getp(v_row[i]);

3.2.2. DSCI_lob_get()

函数原型	int DSCI_lob_get(void *outP, char *pData)
功能说明	获取 LOB 字段的内容及字节长度
参数说明	void *outP: 待复制的目标结构体指针

	char *pData: select 得到的二进制信息流的指针
返回值	返回值: lob 字段存储信息的长度, 如果为 0, 说明是空值
备注	执行 DSCI_fetch_row 后, 如果某个字段是 lob 字段, 则使用此函数获取此字段的内容, 例如: len = DSCI_lob_get(&myP, v_row[i]);

3.2.1. DSCI_lob_isclob

函数原型	int DSCI_lob_isclob(char *p_DATA)
功能说明	获取 LOB 字段是否为 CLOB
参数说明	char *p_DATA: select 得到的二进制信息流的指针
返回值	返回值: 是否为 CLOB, 1 是, 0 否。
适用范围	执行 DSCI_fetch_row 后, 如果某个字段是 lob 字段, 则使用此函数查看是否是 CLOB, 例如: flag = DSCI_lob_isclob(v_row[i]);

3.3. C++类接口

3.3.1. 工厂方法: DSCI_MakeConnection

为了实现各数据库的操作只跟接口定义相关, 不受接口升级影响, 创建数据库连接必须使用本工厂方法。

函数原型	DSCConnection *DSCI_MakeConnection(char *errStr) DSCConnection* DSCI_MakeConnection(char *errStr, char *dbStr)
功能说明	创建数据库连接
参数说明	char *errStr: 如果出错, 本参数存放具体的出错信息 char *dbStr: 如果指定了本参数, 则不使用配置文件, 直接在参数中指定数据库用名密码类型等信息, 本参数具体格式参见 DSCI_init_direct 函数的描述。
返回值	返回值: 连接指针, 如果出错, 返回 NULL
备注	

3.3.2. DSCIConnection类

用于控制数据库事务连接。

成员函数	说明
createStatement	创建 DSCIStatement, 用于直接执行 SQL 语句, 如果失败, 返回 NULL
prepareStatement	创建 DSCIPreparedStatement, 用于预解析指定的 SQL 语句, 如果失败, 返回 NULL

	如果指定 dbNo, 则不读配置, 直接到指定库上解析 SQL
setAutocommit	设置是否自动提交 1 是, 0 否; 成功:返回 0, 失败:返回-1
setBufferflag	设置是否采用 select 缓冲模式 1 是, 0 否
setLogmode	设置写日志模式 on 写日志, off 不写日志, sync 实时写磁盘
getErrCode	获取错误码
getErrMsg	获取错误信息
commit	提交
rollback	回滚
release	销毁

3.3.3. DSCIStatement类

用于直接执行 SQL 语句。

成员函数	说明
executeQuery	直接执行查询操作, 如果指定 dbNo, 则不读配置, 直接到指定库上执行, 如果出错, 返回 NULL
executeUpdate	直接执行增删改操作, 如果指定 dbNo, 则不读配置, 直接到指定库上执行 成功, 返回操作影响到的记录数(>=0), 失败, 返回-1;
release	销毁

3.3.4. DSCIPreparedStatement类

用于预解析 SQL 语句。

成员函数	说明
bind	绑定变量, paramid 为参数序号, 从 0 开始算, 如 0,1,2..... 成功:返回 0, 失败:返回-1
executeQuery	直接执行查询操作, 如果失败, 返回 NULL
executeUpdate	直接执行增删改操作 成功, 返回操作影响到的记录数(>=0), 失败, 返回-1;
release	销毁

3.3.5. DSCIResultSet类

用于存放 select 查询结果。

成员函数	说明
------	----

next	获取下一条记录，true:有记录；false:无记录
getString getInt getLong getDouble getBlobSize getBlob	取出指定字段的值，字段序号从 0 开始，0,1,2.....
getFieldCount	获取字段个数
getRecCount	获取总记录条数，仅在缓冲模式下有意义

4. 附录A（DSCI错误代码定义）

错误代码可以通过 DSCI_ecode 函数获得，分为两类：大于 0 表示客户端自身产生的错误（连接异常等）；小于 0 表示服务端返回的错误（一般指 SQL 执行失败，连接是正常的，具体的错误代码参见各种数据库的手册，各类数据库对应的错误代码是不统一的）。

为了便于编程，下列错误代码，针对各类数据库都是统一的：

服务端错误代码	
-2000	SQL 执行失败
-2100	主键约束

5. 附录B（编程示例）

请参考\$DSCI_PATH/demo 目录下的各个应用示例。

samples	简单示例
lob	操作 BLOB 字段
cpp	C++应用
capability	测试效率