XML2SQL工具说明文档

|  |  |  |
| --- | --- | --- |
| V1.0 | 20160825 | 王雨顺 |
|  |  |  |
|  |  |  |

目录

[一、编译 3](#_Toc459888868)

[1. xml2sql.env文件 3](#_Toc459888869)

[2. Makefile文件 3](#_Toc459888870)

[3. x2s\_config.xml文件 4](#_Toc459888871)

[4. x2smain.h文件 4](#_Toc459888872)

[二、接口 4](#_Toc459888873)

[1. x2s\_execute 4](#_Toc459888874)

[2. db\_open 4](#_Toc459888875)

[3. db\_close 4](#_Toc459888876)

[4. db\_get\_value\_str 4](#_Toc459888877)

[5. db\_set\_value\_str 5](#_Toc459888878)

[6. db\_get\_value\_int 5](#_Toc459888879)

[7. db\_set\_value\_int 5](#_Toc459888880)

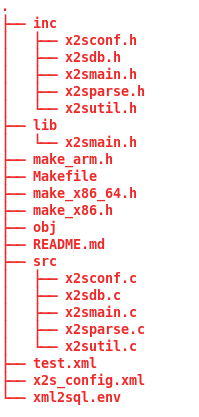
[8. db\_insert\_item 5](#_Toc459888881)

[9. db\_delete\_item 5](#_Toc459888882)

[三、TODO 6](#_Toc459888883)

# 一、编译

源代码的目录结构如下图：



下面介绍几个重要文件

## 1. xml2sql.env文件

本文件用于编译平台控制，现有内容如下表：

|  |
| --- |
| export XML2SQL\_BASE=`pwd`  #export PLATFORM=ARM  #export PLATFORM=X86  **export PLATFORM=X86\_64** |

对于要使用的平台去掉 **#** 注释符就好，然后执行 ***source xml2sql.env*** 用来导出配置好的环境变量。

## 2. Makefile文件

本文件用来控制工程的编译过程，目前工程可以生成三种不同的目标文件：

a). 执行***make exe*** 生成可执行文件

b). 执行 ***make liba*** 生成静态库文件

c). 执行 ***make libso*** 生成动态库文件

生成的库文件以及导出头文件位于工程的lib文件夹中

## 3. x2s\_config.xml文件

这是项目的配置文件，里面可以配置数据库的名字，数据库表的名字，以后将要解析的xml文件列表，还有debug开关的配置。

## 4. x2smain.h文件

这是生成库文件的时候相对应的导出文件

# 二、接口

接口部分主要介绍库导出文件x2smain.h中的接口，所有导出函数的返回值均为X2S\_SUCCESS/X2S\_FAIL

## 1. x2s\_execute

原型：int x2s\_execute();

描述：用于分析提取xml文件信息，并将相关信息存入sqlite数据库

## 2. db\_open

原型：int db\_open();

描述：打开sqlite数据库，操作数据库之前必须执行此函数

## 3. db\_close

原型：int db\_close();

描述：关闭sqlite数据库，操作数据库完成之后必须执行此函数

## 4. db\_get\_value\_str

原型：int db\_get\_value\_str(const char \*key, char \*value, int len);

参数：

key：键值

value：用于存储根据键值获取到的相对应的值

len：value的buffer大小

描述：通过提供的键值，从数据库中获取相对应的值

## 5. db\_set\_value\_str

原型：int db\_set\_value\_str(const char \*key, char \*value, int len);

参数：

key：键值

value：用于设置的值

len：value的内容大小

描述：设置制定键值的值

## 6. db\_get\_value\_int

原型：int db\_get\_value\_int(const char \*key, int \*value);

参数：

key：键值

value：用于存储根据键值获取到的相对应的值

描述：通过提供的键值，从数据库中获取相对应的值，并转成整数形式

## 7. db\_set\_value\_int

原型：int db\_set\_value\_int(const char \*key, int value);

参数：

key：键值

value：用于设置的值

描述：设置制定键值的值，设置值是整数形式

## 8. db\_insert\_item

原型：int db\_insert\_item(int id, const char \*name, const char \*value, int type);

参数：

id：插入条目的id

name：键值

value：用于设置的值

type：设置值的类型（目前未使用）

描述：向数据库插入一条新的记录

## 9. db\_delete\_item

原型：int db\_delete\_item(const char \*name);

参数：

name：键值

描述：从数据库删除一条记录

# 三、TODO

1. 数据库记录中的type字段目前还没有使用，以后可以根据需求更改

2. 在导出库的时候可能需要将x2smain.c中的main函数删除，本来想在Makefile中加以控制，目前还没有实现

3. 项目没有经过详细测试