mxml 库说明书

版本	日期	作者	说明
1.0	2016-7-19		1、mxml

1. 主要功能

模块主要功能：

1、 负责 xml 文件的创建

2. XML 文件优缺点

1、 优点：

A：格式统一，符合标准；

B：容易与其他系统进行远程交互，数据共享比较方便；

2、 缺点：

A：XML 文件庞大，文件格式复杂，传输占带宽；

B：服务器端和客户端都需要花费大量代码来解析 XML，导致服务器端和客户端代码变得异常复杂且不易维护；

C：客户端不同浏览器之间解析 XML 的方式不一致，需要重复编写很多代码；

D: 服务器端和客户端解析 XML 花费较多的资源和时间。

3. 输入

1、 无

4. 输出

1、 无

5. 目录结构

Objs 目录：存放编译时产生的.o 等文件，是一个临时文件

.c/.h 文件：模块的源文件

Makefile 文件：编译所需文件

Build.sh 文件：一键编译文件，产生的静态库文件 libmxml_x64.a 将被放到上级目录的 lib 文件中

依赖库文件： 无

6. 流程

一、部分函数详解

1、添加一个节点到树种

```
    void mxmlAdd (
mxml_node_t *parent,
int where,
mxml_node_t *child,
mxml_node_t *node
);
```

2、删除一个节点和它的所有的子节点

```
    void mxmlDelete (
mxml_node_t *node
);
```

3、获取一个参数

```
    const char *mxmlElementGetAttr (
mxml_node_t *node,
const char *name
);
```

4、设置属性

```
    void mxmlElementSetAttr (
mxml_node_t *node,
const char *name,
const char *value
);
```

5、设置一个 XML 元素属性使用一个格式化的值

```
    void mxmlElementSetAttrf (
mxml_node_t *node,
const char *name,
const char *format,
...
);
```

6、添加一个回调函数将 XML 实体转换为 Unicode 编码字符

```
    int mxmlEntityAddCallback (void);
```

7、获取一个字符值对应的 XML 实体名字

```
    const char *mxmlEntityGetName (
int val
);
```

8、获取一个代表到一个 XML 命名实体的字符

```
int mxmlEntityGetValue (
const char *name
);
```

9、删除一个 XML 实体回调

```
void mxmlEntityRemoveCallback (void);
```

10、搜索一个命名的 XML 元素

```
mxml_node_t *mxmlFindElement (
mxml_node_t *node,
mxml_node_t *top,
const char *name,
const char *attr,
const char *value,
int descend
);
```

11、删除一个索引

```
void mxmlIndexDelete (
mxml_index_t *ind
);
```

12、返回索引中的下一个节点

```
mxml_node_t *mxmlIndexEnum (
 mxml_index_t *ind
 );
```

13、搜索下一个匹配节点

```
mxml_node_t *mxmlIndexFind (
mxml_index_t *ind,
const char *element,
const char *value
);
```

14、创建一个新的索引

```
mxml_index_t *mxmlIndexNew (
mxml_node_t *node,
const char *element,
const char *attr
);
```

16、重新设置索引中的枚举/搜索指针并且返回索引中的第一个节点

```
mxml_node_t *mxmlIndexReset (
```

```
 mxml_index_t *ind
 );
```

17、载入一个文件描述到一个 XML 节点树

```
      mxml_node_t *mxmlLoadFd (
mxml_node_t *top,
int fd,
mxml_load_cb_t cb
);
```

18、载入一个文件到一个 XML 节点树

```
      mxml_node_t *mxmlLoadFile (
 mxml_node_t *top,
 FILE *fp,
 mxml_load_cb_t cb
 );
```

19、载入一个文件到一个 XML 节点树

```
      mxml_node_t *mxmlLoadString (
 mxml_node_t *top,
 const char *s,
 mxml_load_cb_t cb
 );
```

20、创建一个新的 CDATA 节点（Mini-XML2.3）

```
       mxml_node_t *mxmlNewCDATA (
 mxml_node_t *parent,
 const char *data
 );
```

21、创建一个新的用户自定义数据节点（Mini-XML2.1）

```
      mxml_node_t *mxmlNewCustom (
 mxml_node_t *parent,
 void *data,
 mxml_custom_destroy_cb_t destroy
 );
```

22、创建一个新的 XML 元素节点

```
      mxml_node_t *mxmlNewElement (
mxml_node_t *parent,
const char *name
);
```

23、创建一个新的整数节点

```
mxml_node_t *mxmlNewInteger (
mxml_node_t *parent,
int integer
);
```

24、创建一个新的不透明字符串节点

```
mxml_node_t *mxmlNewOpaque (
mxml_node_t *parent,
const char *opaque
);
```

25、创建一个浮点数节点

```
mxml_node_t *mxmlNewReal (
mxml_node_t *parent,
double real
);
```

26、创建新的文本分段节点

```
mxml_node_t *mxmlNewText (
mxml_node_t *parent,
int whitespace,
const char *string
);
```

27、创建一个新的格式化文本分段节点

```
mxml_node_t *mxmlNewTextf (
mxml_node_t *parent,
int whitespace,
const char *format,
...
);
```

28、创建一个新的 XML 文档树（Mini-XML2.3）

```
mxml_node_t *mxmlNewXML (
const char *version
);
```

29、释放一个节点（Mini-XML2.3）

```
int mxmlRelease (
mxml_node_t *node
);
```

30、移除一个节点从它的父节点中

```
void mxmlRemove (
```

```
mxml_node_t *node
);
```

31、保留一个节点

```
    int mxmlRetain (
mxml_node_t *node
);
```

32、使用 SAX 回调从一个文件描述符中加载数据到一个 XML 节点树（Mini-XML2.3）

```
    mxml_node_t *mxmlSAXLoadFd (
 mxml_node_t *top,
 int fd,
 mxml_load_cb_t cb,
 mxml_sax_cb_t sax_cb,
 void *sax_data
 );
```

33、使用 SAX 回调一个文件中加载数据到一个 XML 节点树（Mini-XML2.3）

```
    mxml_node_t *mxmlSAXLoadFile (
 mxml_node_t *top,
 FILE *fp,
 mxml_load_cb_t cb,
 mxml_sax_cb_t sax_cb,
 void *sax_data
 );
```

34、使用 SAX 回调从一个字符串中加载数据到一个 XML 节点树

```
    mxml_node_t *mxmlSAXLoadString (
 mxml_node_t *top,
 const char *s,
 mxml_load_cb_t cb,
 mxml_sax_cb_t sax_cb,
 void *sax_data
 );
```

35、保存一个 XML 节点树到一个内部分配的字符串

```
    char *mxmlSaveAllocString (
mxml_node_t *node,
mxml_save_cb_t cb
);
```

36、保存一个 XML 节点树到一个文件描述符

```
    int mxmlSaveFd (
```

```
 mxml_node_t *node,
 int fd,
 mxml_save_cb_t cb
 );
```

37、保存一个 XML 节点树到一个文件

```
        int mxmlSaveFile (
 mxml_node_t *node,
 FILE *fp,
 mxml_save_cb_t cb
 );
```

38、保存一个 XML 节点树到一个字符串

```
       int mxmlSaveString (
mxml_node_t *node,
char *buffer,
int bufsize,
mxml_save_cb_t cb
);
```

39、设置一个 CDATA 元素节点的名称（Mini-XML2.3）

```
       int mxmlSetCDATA (
mxml_node_t *node,
const char *data
);
```

40、对一个用户自定义数据节点设置数据和销毁回调函数（Mini-XML2.1）

```
       int mxmlSetCustom (
mxml_node_t *node,
void *data,
mxml_custom_destroy_cb_t destroy
);
```

41、设置对于自定义数据的处理回调函数

```
      void mxmlSetCustomHandlers (
mxml_custom_load_cb_t load,
mxml_custom_save_cb_t save
);
```

42、设置 XML 元素节点的名字

```
      int mxmlSetElement (
mxml_node_t *node,
const char *name
);
```

43、设置错误信息回调函数

```
    void mxmlSetErrorCallback (
mxml_error_cb_t cb
);
```

44、设置一个不透明字符串节点的值

```
    int mxmlSetOpaque (
mxml_node_t *node,
const char *opaque
);
```

45、设置一个浮点数节点的值

```
    int mxmlSetReal (
mxml_node_t *node,
double real
);
```

46、设置一个文本节点的值

```
    int mxmlSetText (
mxml_node_t *node,
int whitespace,
const char *string
);
```

47、设置一个文本节点的值为一个格式化的字符串

```
    int mxmlSetTextf (
mxml_node_t *node,
int whitespace,
const char *format,
...
);
```

48、设置在保存 XML 数据时的自动折行位置

```
    void mxmlSetWrapMargin (
int column
);
```

49、遍历到 XML 树种的下一个逻辑节点

```
    mxml_node_t *mxmlWalkNext (
mxml_node_t *node,
mxml_node_t *top,
int descend
);
```

50、遍历到 XML 树中的上一个逻辑节点

```
mxml_node_t *mxmlWalkPrev (
mxml_node_t *node,
mxml_node_t *top,
int descend
);
```

二、数据类型

1、XML 元素节点的属性值：mxml_attr_t

2、自定义数据销毁回调函数原型：mxml_custom_destroy_cb_t

3、自定义数据加载回调函数原型：mxml_custom_load_cb_t

4、自定义数据保存回调函数原型：mxml_custom_save_cb_t

5、自定义 XML 类型值：mxml_custom_t（Mini-XML 2.1）

6、XML 元素值：mxml_element_t

7、错误回调函数原型：mxml_error_cb_t

8、XML 节点索引：mxml_index_t

9、加载回调函数：mxml_load_cb_t

10、XML 节点：mxml_node_t

11、保存回调函数：mxml_save_cb_t

12、SAX 回调函数：mxml_sax_cb_t

13、SAX 事件类型.：mxml_sax_event_t

14、XML 文本节点值：mxml_text_t

15、XML 节点值：mxml_value_t

7. 鸣谢

感谢之前对这个模块有过贡献的同事们。