



XLibEditor使用说明及开源许可

翼王 E-mail: eawtest@163.com

功能简介：

XLibEditor是个简单易用的泛用型数据库编辑器，可以轻松的制定的数据库的模板，快速创建游戏或者其他应用需要的数据库，方便大量不同类型数据的录入，支持导出为通用的PLIST文件格式和Cocoa专用的文件格式。

编辑器支持多种数据类型，并可以实现不同类别数据的引用、自动预览加载的图片路径、多语言字串等多种功能。

使用流程：

- 1.在模板编辑器中制作对应数据库的模板
- 2.在主界面中创建和编辑对应的数据

使用范例：

在模板编辑器中分出道具、怪物类别，分别给道具和怪物设置名称、攻击力等属性，到主界面中给道具列表下创建“铁剑”资源并编辑对应的属性，导出为数据库放入游戏中使用。

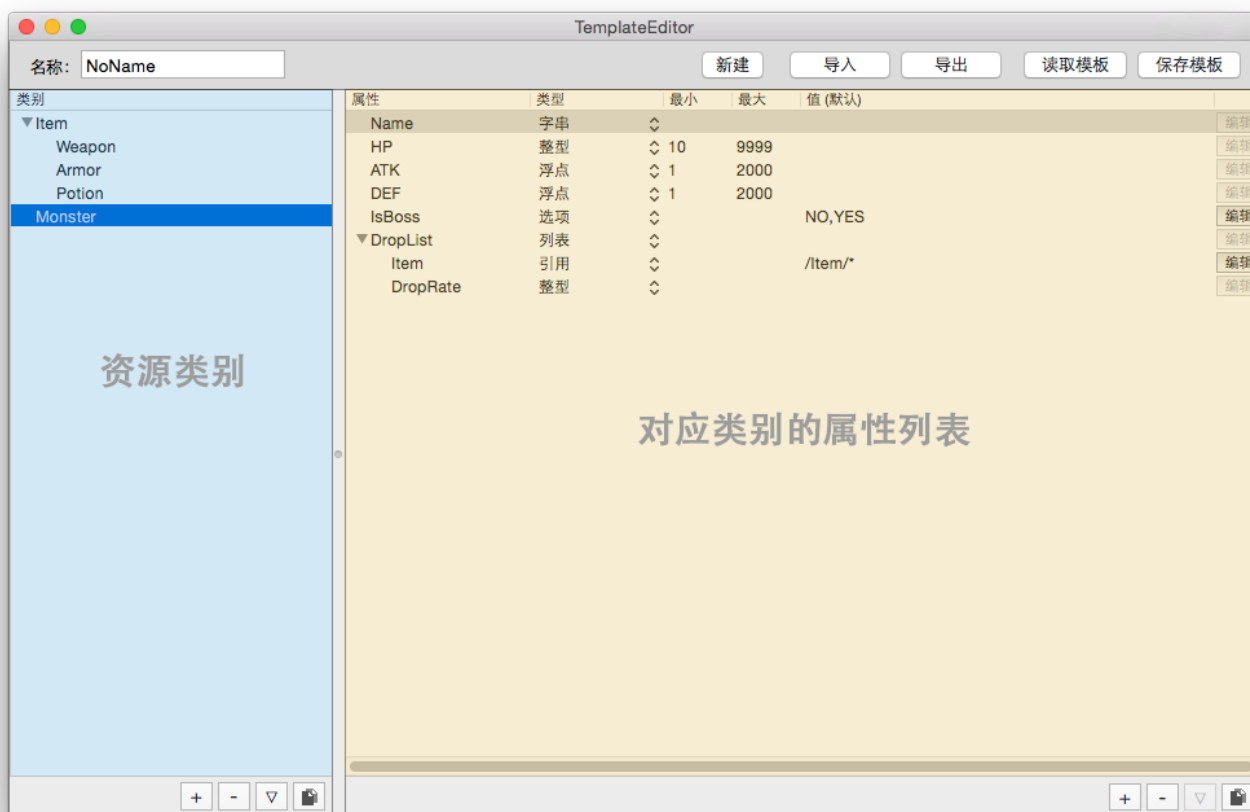
（本说明由模板编辑、数据库编辑、文件格式、开源许可四个部分组成）

===== 模板编辑 =====

因为模板生成一次即可重复使用，同时开发中多数时间是处理数据库，所以模板编辑器默认是隐藏的，可以在程序菜单中打开



模板编辑器主界面：



1.资源类别编辑区域



资源类别 使用树形结构来管理，下方按钮的功能为



*只需要拖动类别即可更改资源位置和在树形图中的位置

2.资源类别属性编辑区域

属性	类型	最小	最大	值 (默认)	
Name	字符串	↕			编辑
HP	整型	↕ 10	9999		编辑
ATK	浮点	↕ 1	2000		编辑
DEF	浮点	↕ 1	2000		编辑
IsBoss	选项	↕		NO,YES	编辑
▼ DropList	列表	↕			编辑
Item	引用	↕		/Item/*	编辑
DropRate	整型	↕			编辑

对应功能：

属性：该类别具有的属性（属性名称）

类型：该属性的数据类型

- 整型：和C语言中的整型一致，记录整数
- 浮点：和C语言中的浮点一致，记录带有小数点的数据
- 关键字：记录文字，不支持多语言
- 字符串：记录文字，支持多语言
- 图片：可以弹出文件选择框来设置图片资源的路径，并且在主编辑器中提供预览功能，支持多语言
- 选项：可以自行编辑选项列表，记录的数据为选中的项目的序号（例如可

以给装备或者道具设置“元素”属性，通过选择来设置对应数据，防止错误录入)

-引用：可以引用另外一个类别中的资源，数据库中记录的是引用资源的完整路径（/父类别/子类别.../最终子类别.资源名，参考下图）；

-分类：允许添加子属性，用途是将属性归类

-列表：参考下图，用于建立有重复子属性的列表，图中是怪物的掉落概率列表

模板：

▼ DropList	列表	↕		编辑
Item	引用	↕	/Item/*	编辑
DropRate	整型	↕		编辑

实际使用效果：

▼ DropList		↕		编辑
▼ 0		↕		编辑
Item		↕	/Item/Potion.Apple	编辑
DropRate		↕	15	编辑
▼ 1		↕		编辑
Item		↕	/Item/Weapon.wood Sword	编辑
DropRate		↕	2	编辑

最小：该属性允许的最小值（当属性类型为“列表”时设置允许的最少子项个数）

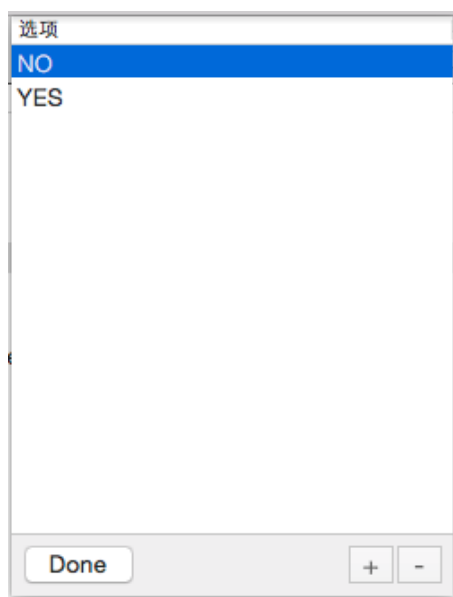
最大：该属性允许的最大值（当属性类型为“列表”时设轩子项个数上限）

值：当为数字或者字串类型时，该项为初始值。为“选项”时显示选项列表，为“引用”时显示引用类别的路径。

***双击对应栏目或者点右边的编辑选项即可修改对应值，底下的编辑按钮和类别编辑区域类似，分别为添加属性、删除属性、新建子属性、复制属性。**

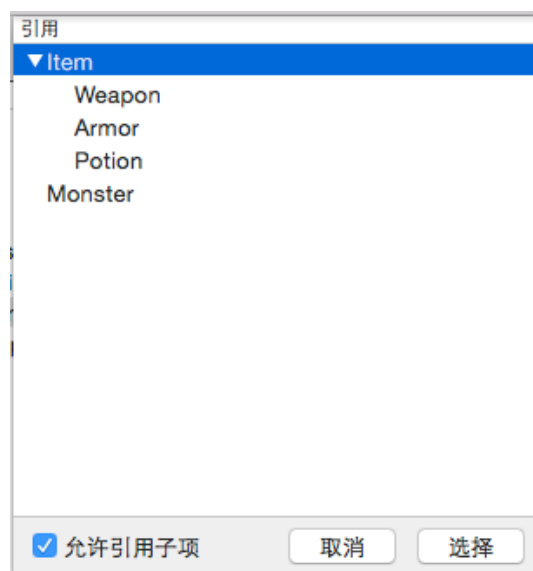
*其中选项编辑界面：

（直接加减选项和双击选项名即可进行编辑）



*引用编辑界面：

（不勾选底下允许引用子项时，编辑数据库时只能引用指定类别下的资源，勾选则可以选择该类别及其任意子类别下的资源）



3.模板文件编辑区域



名称：设定该模板的名字

新建：放弃当前正在编辑的模板，新建一个空模板

导入：从文件导入模板

导出：将模板导入到文件

读取模板：从编辑器中读取模板

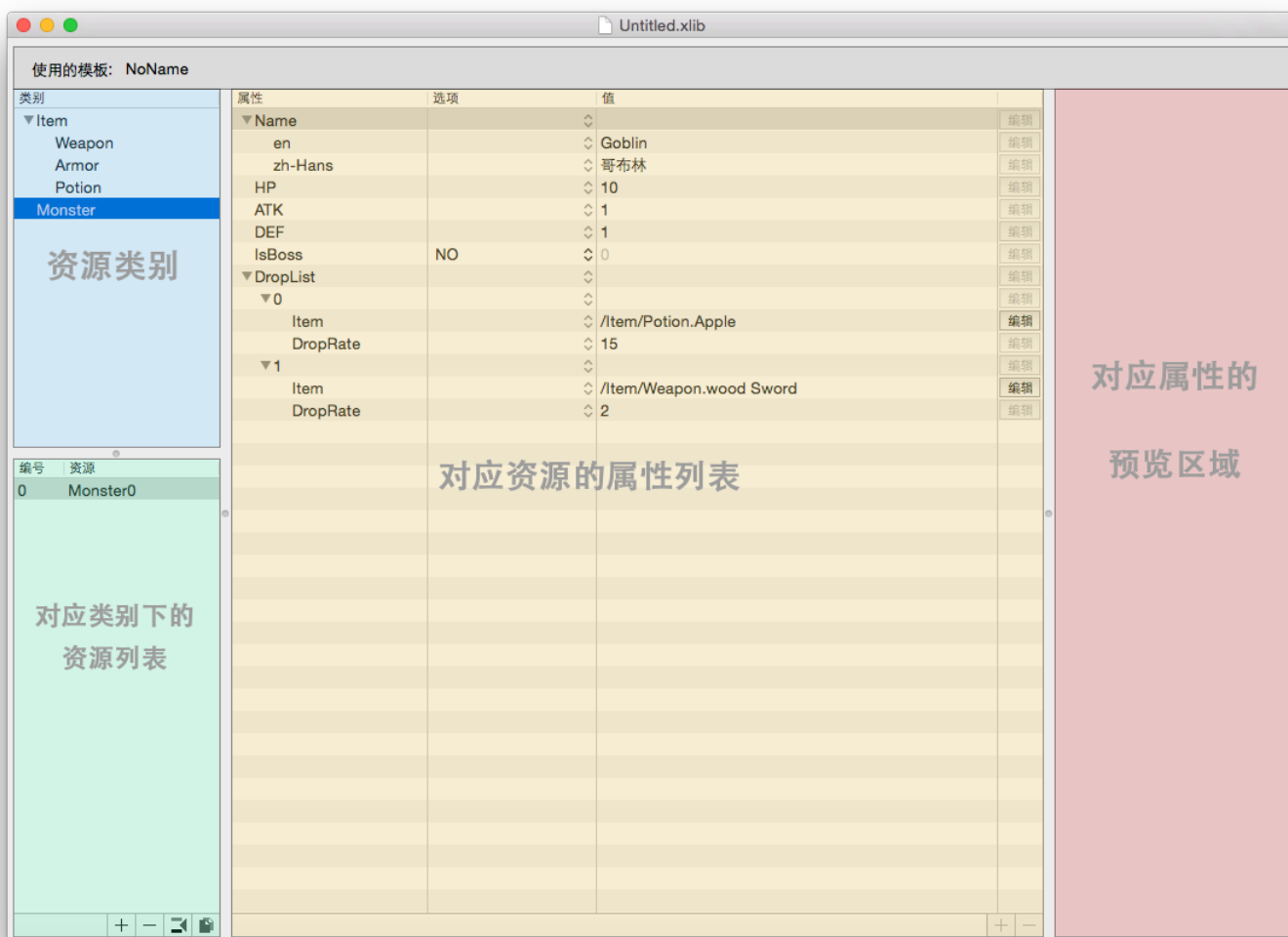
保存模板：将模板保存到编辑器中，并对已打开的数据库文件应用该模板

*编辑器使用的模板实际上是保存在user default里面，因为开发进度和时间

（TOU）精力（LAN）的原因，目前还不支持保存多个模板，当模板和数据库不符时，会自动清除数据库中 新模板中没有的属性，并添加新属性和设置好初始值。可能会在以后的版本加入更灵活的模板管理。发生错误保存模板的时候不要保存数据库文件即可。

===== 数据库编辑 =====

主界面：



1.左上方“资源类别”区域

该区域中没有增删拖动等编辑选项，显示的是数据库的树形结构，选择对应类型即可编辑这个类别中的资源。

2.资源列表

编号	资源
0	Monster0

编号：资源在该类别中的序号

资源：显示该资源的名称

下方编辑按钮分别为增加资料、删除资源、插入资源、复制资源。拖动即可更改资源顺序，双击资源的名称即可进入修改。

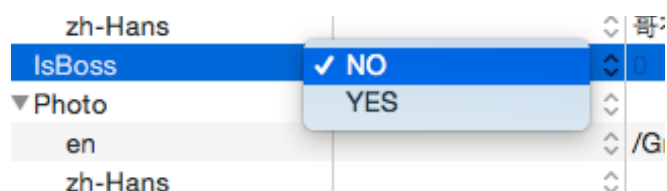
*无须担心更改资源名称或者顺序时，其他资源对他的引用会失效，引用会自动跟踪定位到对应的资源。但是如果在模板中更改了类别的名称，则会失效。

3.属性编辑区域

属性：显示该资源所具有的全部属性

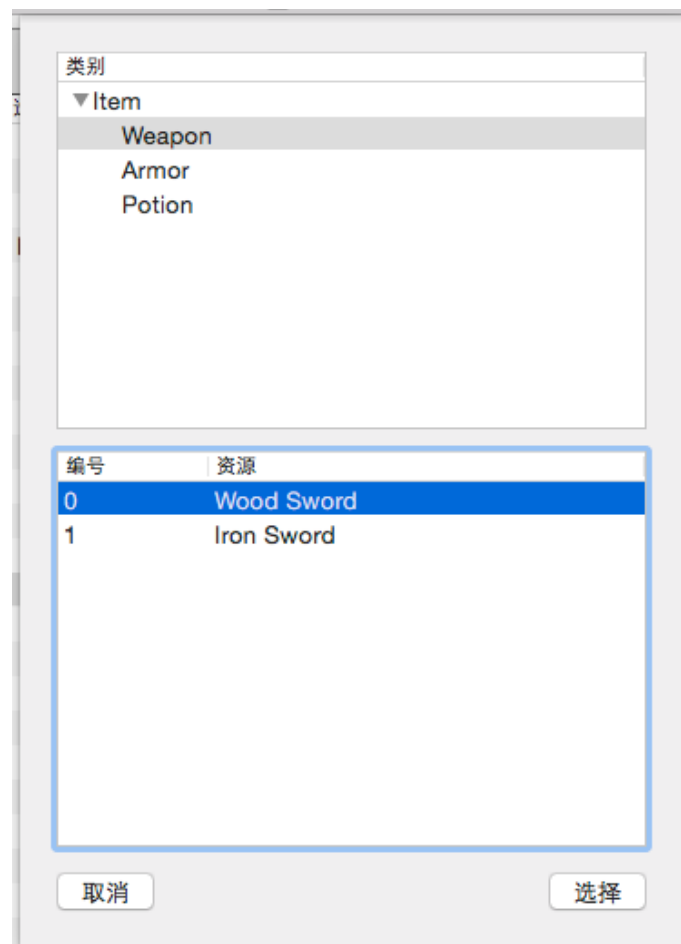
[illegible]

选项：之前在模板中设置的选项列表，选择后右的边的“值”栏会显示对应的数值，当该属性不支持选项时会被禁用。

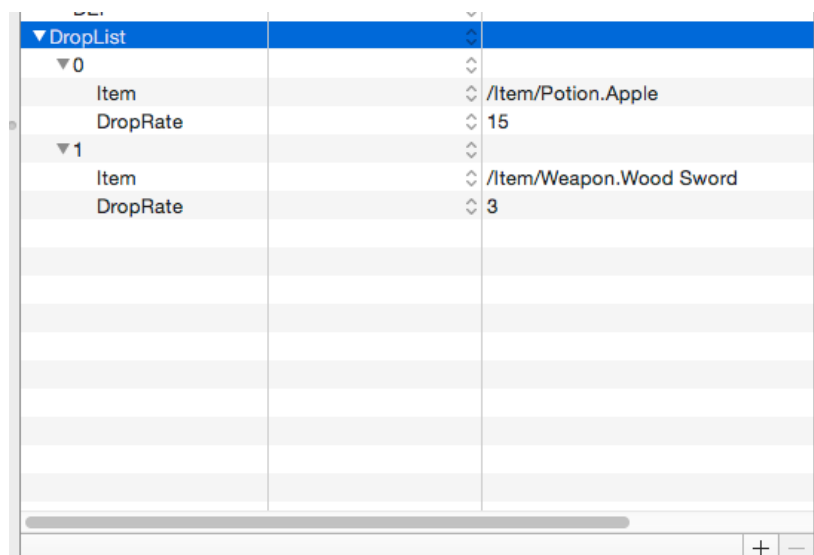


值：显示该属性对应的值，可以直接双击编辑，对于不想要的照片或者引用直接将值删掉即可。

编辑：对于引用和照片类别，会弹出对应的辅助框帮助选择。



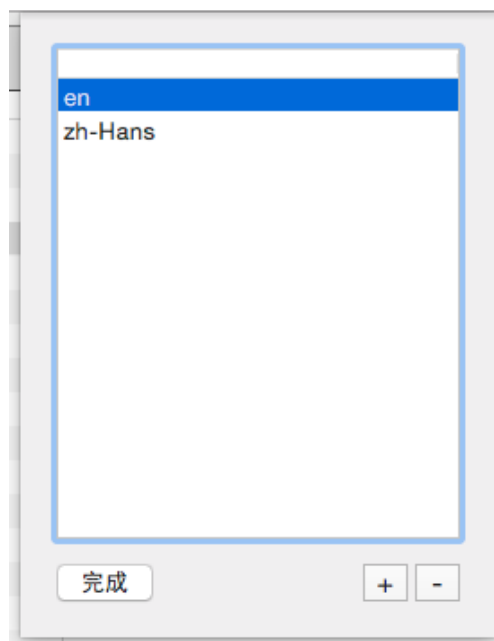
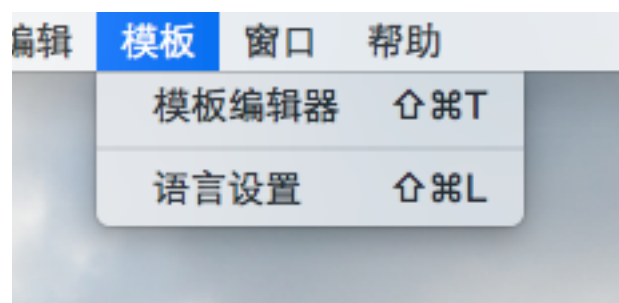
最右下角的 + - 按钮可以对列表项进行增删



*考虑到游戏开发可能会面向多个国家或者语言，编辑器中加入了多语言的支持，同一个字串下会自动根据语言列表建立对应的子项：

▼ Name		
en		Goblin
zh-Hans		哥布林
zh-Hant		哥布林

语言列表可以在程序菜单中打开设置：



===== 文件格式 =====

编辑器目前支持两种格式，一种是cocoa自动的NSKeyedArchiver打包，另一种是PLIST（实际上就是XML）。因为是开源的，所以也可以自行修改添加其他的格式支持。

NSKeyedArchiver打包内部是NSArray和NSDictionary的嵌套结构，在mac和ios程序中都可以直接支持。数据结构可以做一个简单的数据库，在mac中导入并nslog出来查看。也可以另存为PLIST版本，用PLIST / XML编辑器来查看。

导入代码可以用：

```
NSString *dataPath = @"your file path";
NSData *data = [NSData dataWithContentsOfFile:dataPath];
NSKeyedUnarchiver *Unarchiver = [[NSKeyedUnarchiver alloc]
initWithReadingWithData:data];

NSString *templateName = [Unarchiver decodeObjectForKey:@"templateName"];
NSArray *languageList = [Unarchiver decodeObjectForKey:@"languageList"];
NSArray *dataArchive = [Unarchiver decodeObjectForKey:@"data"];
[Unarchiver finishDecoding];
```

也可以从源代码中抽取对应的类和读写代码，可以读取文件并还原回为编辑器中的树形结构（即objective-c对象组成的结构），根据需要对类别进行精简或者直接照用。资源的类是BEResource，属性的类是BEProperty，将打包还原回对象的方法在BEDataController类里面。不过这个方法需要模板来加载，要去掉模板需要花点时间整理代码，所以按照NSArray和NSDictionary这种方式来读取会更简单一些。

其中这种格式根据模板文件和数据库文件会分为不同的后续，xlib和template，防止加载错误。

第二种PLIST，Mac和iOS也是由cocoa原生支持的，其他平台都可以找到相应的PLIST / XML解析库。

模板文件和数据库文件都可以导出为PLIST，因为内部结构不同，所以错误打开时也不会发生BUG（理论上）。

Mac和IOS导入PLIST可以用

```
NSString *dataPath = @"your file path";
NSData *data = [NSData dataWithContentsOfFile:dataPath];

NSPropertyListFormat format;
NSDictionary *dictionary = [NSPropertyListSerialization
    dictionaryWithData:data
    options:NSPropertyListMutableContainersAndLeaves
    format:&format
    error:nil];

NSString *templateName = [dictionary objectForKey:@"TemplateName"];
NSArray *languageList = [dictionary objectForKey:@"languageList"];
NSArray *dataArchive = [dictionary objectForKey:@"data"];
```

另外这个程序包里面了带上了苹果官方的AES256加密类扩展

（NSData+AES256），不过我不确定其他平台是否能同样解压回来，以及时间（TOU）精力（LAN）的关系，并没有写在程序之中，有需要的可以在文档保存的地方调用：

```
NSData *EncryptedData = [data AES256EncryptWithKey:key];
```

和

```
NSData *decryptedData = [data AES256DecryptWithKey:key];
```

即可完成数据库的加密和解密。

另外资源引用方面采用了路径的方法，因为可以简单地适配多级类别和跟踪等功能，根据路径只要解析字符串就可以查到目标资源，工程中有对应的代码，可以作为参考。另外记得设定名称的时候不要使用/ .之类在路径中使用的符号哦~。

===== 开源许可 =====

Copyright (c) 2015, Wing Xu
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.