

xlang 动态链接库开发与使用

文档库地址 <https://xlang.link/documents/index.html>

xlang 在版本 3.0 中新增了共享库与动态链接库，本文档主要讲解**共享库**和**动态链接库**的使用。

1.**共享库**，扩展名 slx (意为 shared library for xlang), slx 共享库是 xlang 可执行程序集，以供各程序集模块之间的功能调用，支持 xlang 支持的所有平台，支持静态加载和动态加载两种方式，如同 linux 系统中的 so 文件或者 windows 系统的 dll 文件，可以在编译时通过动态链接库进行加载，也可以在运行时通过 `_system_.loadLibrary("xxx.slx");` 的方式进行加载；

共享库的入口程序格式为:

```
void _entry(int hmodule, int ver){
```

```
}
```

参数 hmodule 为当前模块的 id，可以通过 `_system_.getModulePath` 来获取模块的文件地址，

参数 ver 为 x 虚拟机版本，用以解决未来版本发生变更时可能产生的问题。

共享库的初始化顺序为 1 全局->2 入口

程序集初始化顺序为后加载的模块最先初始化，先加载的后初始化，主模块最后执行。

例加载顺序为 MainModule -> Module1.slx -> Module2.slx -> Module3.slx;

则初始化顺序为 Module3(全局\静态对象) -> Module2(全局\静态对象) -> Module1(全局\静态对象)->MainModule(全局\静态对象) -> Module3(入口) -> Module2(入口) -> Module1(入口)->MainModule(入口);

2.**动态链接库**，扩展名为 xl (xlang linker)，该库用于 xlang 程序集之间的链接描述（如同 windows 系统中的 DLL 中导出函数产生的 lib 或者 linux 中的.a 文件）。

该链接库用来导出程序集中需要导出的对象，并对程序集(主模块 <-> 共享库、共享库 <-> 共享库) 进行调用描述、编译链接和 IntelliSense 的参考。

与传统的 Dll 或者 SO 不同点在于该链接库可以支持从 slx 中反向调用主模块程序集中的功能。

注意：为了限制模块之间的耦合度，仅能导出 class 、package，不可导出变量、常量、全局方法。

示例:

一. 正向链接 (即从主模块调用 slx 中的功能)。

1. 编译共享库.

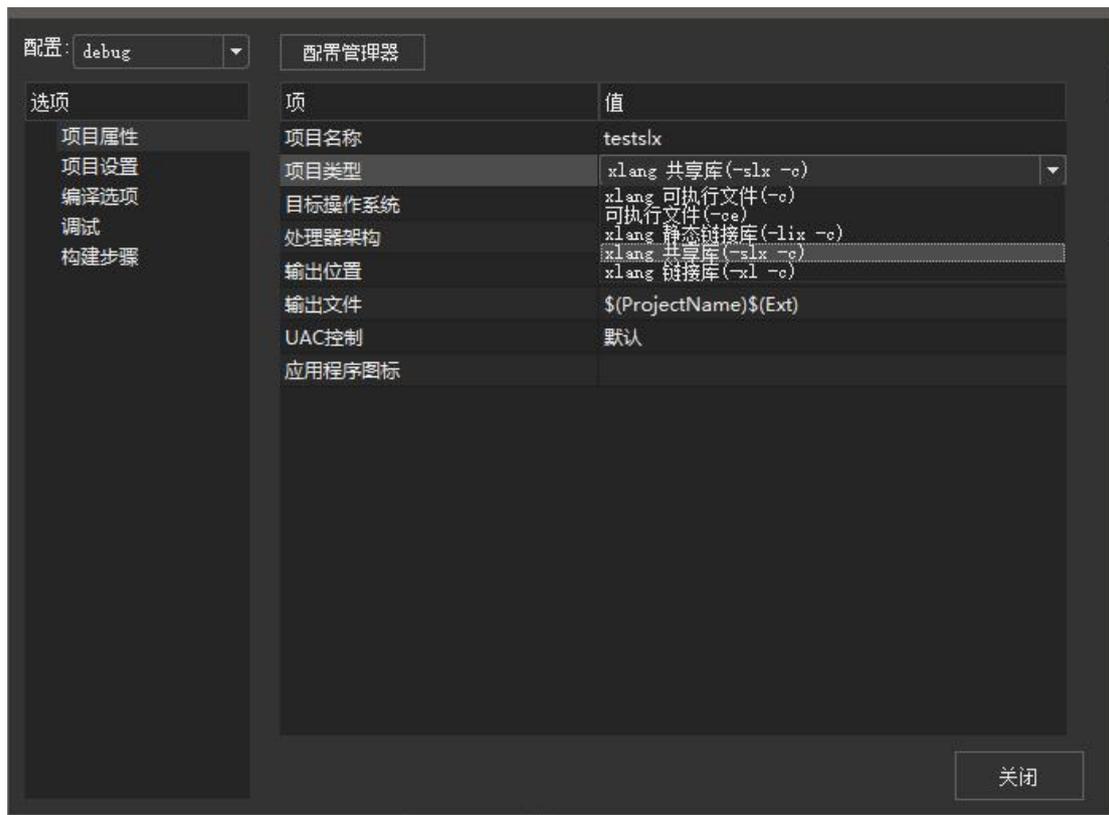
建立 xlang 共享库项目 testslx, 编写如下代码

```
class MyClass{
    static void output(String text){
        _system_.output(text);
    }
};

void _entry(int hm,int ver){

}
```

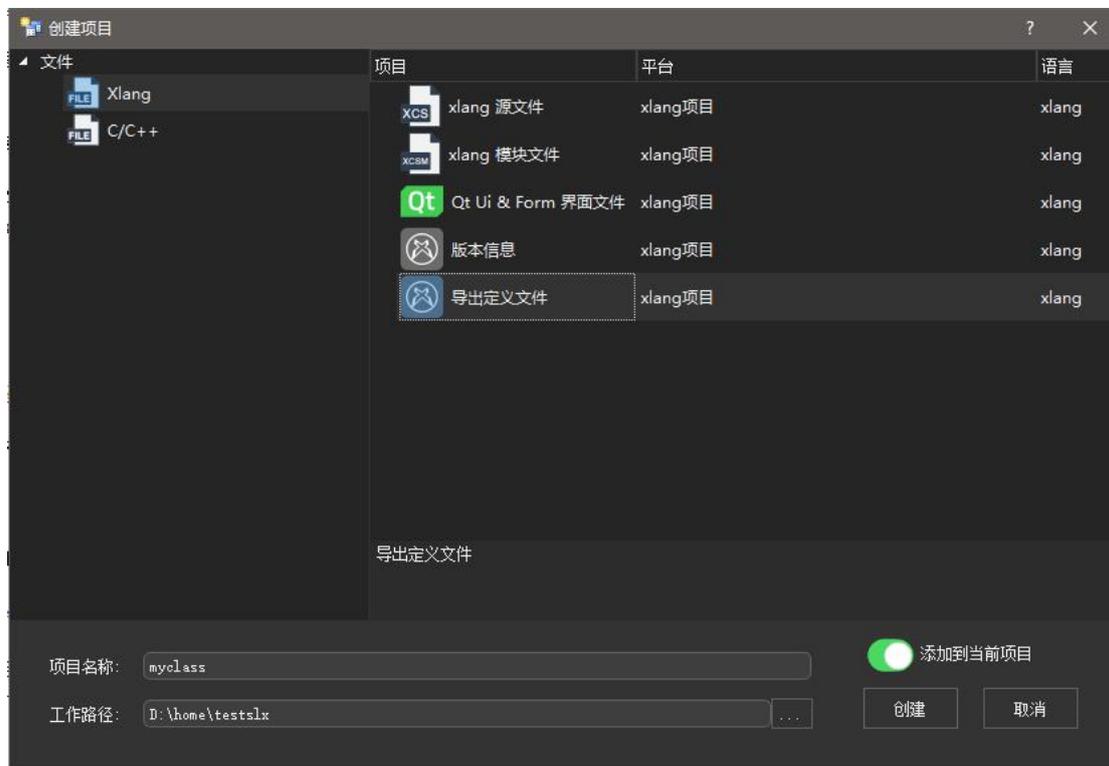
修改项目类型为共享库:



编译生成 testslx.slx

2.生成链接库.

点击新建添加 xdef 文件:



文件内容:

```
{  
  "library": "testslx.slx",  
  "exports": ["MyClass"],  
  "hidden": []  
}
```

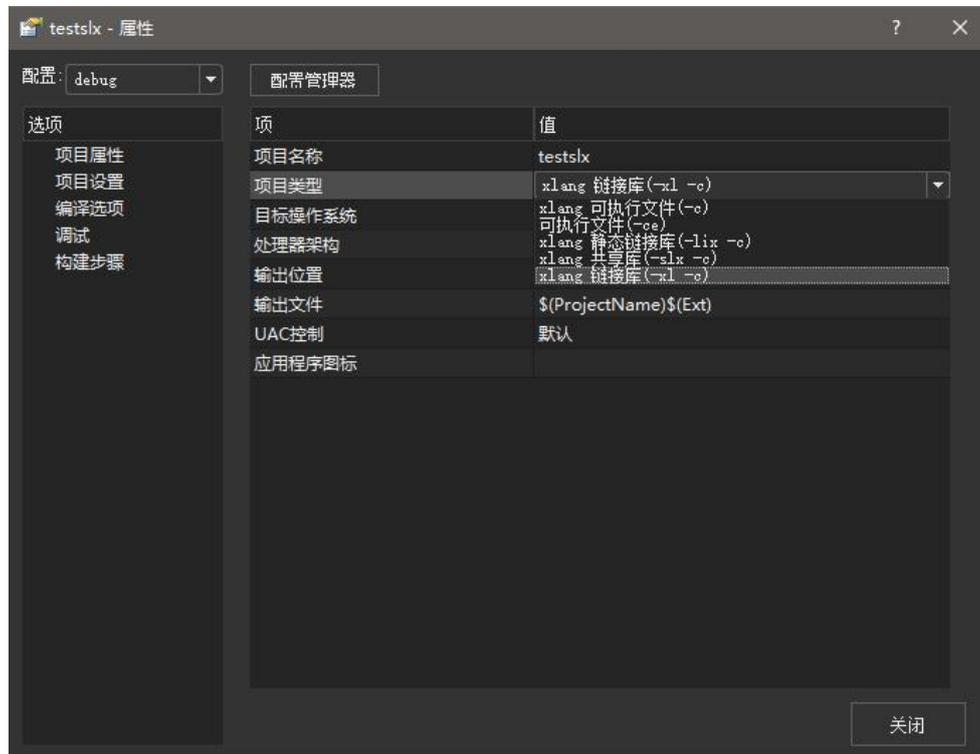
该文件为 json 格式文件,

library 字段指明该共享库的名称, 程序启动时将由系统自动加载该模块.

exports 指明需要导出的对象名称。(如不指定默认全部导出)。

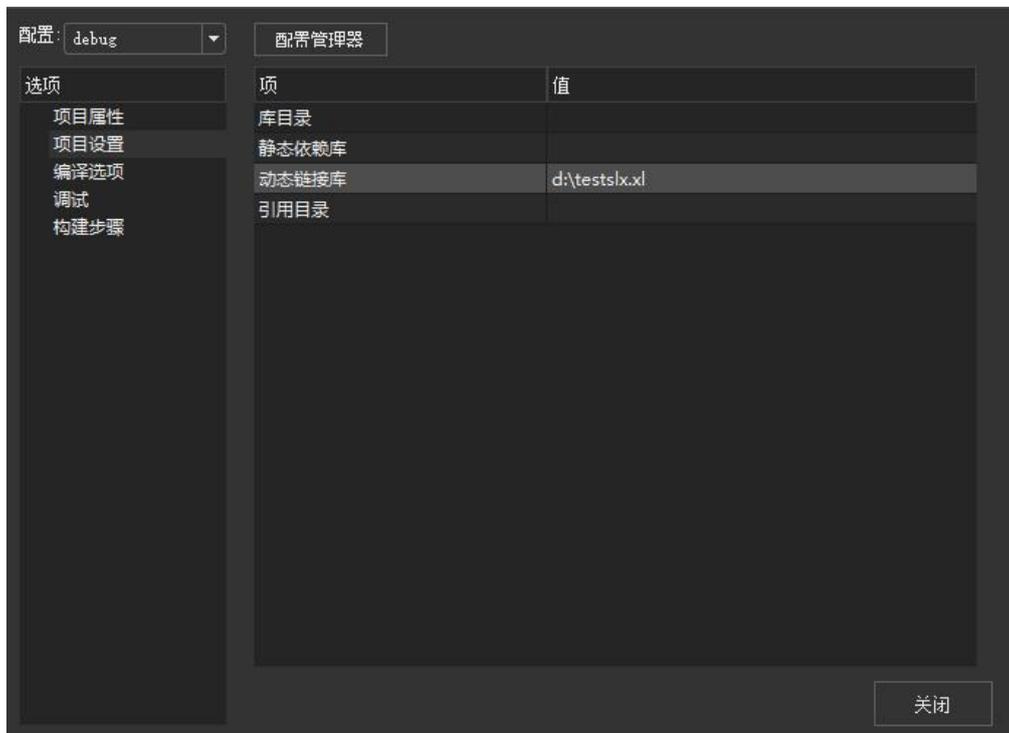
hidden 指定需要隐藏的对象名称(在 library 不指定的情况下, 该选项将隐藏不想导出的对象)。

修改项目类型为动态链接库:



编译生成 test.xl 文件。

3.在需要使用此共享库的项目中添加链接库.



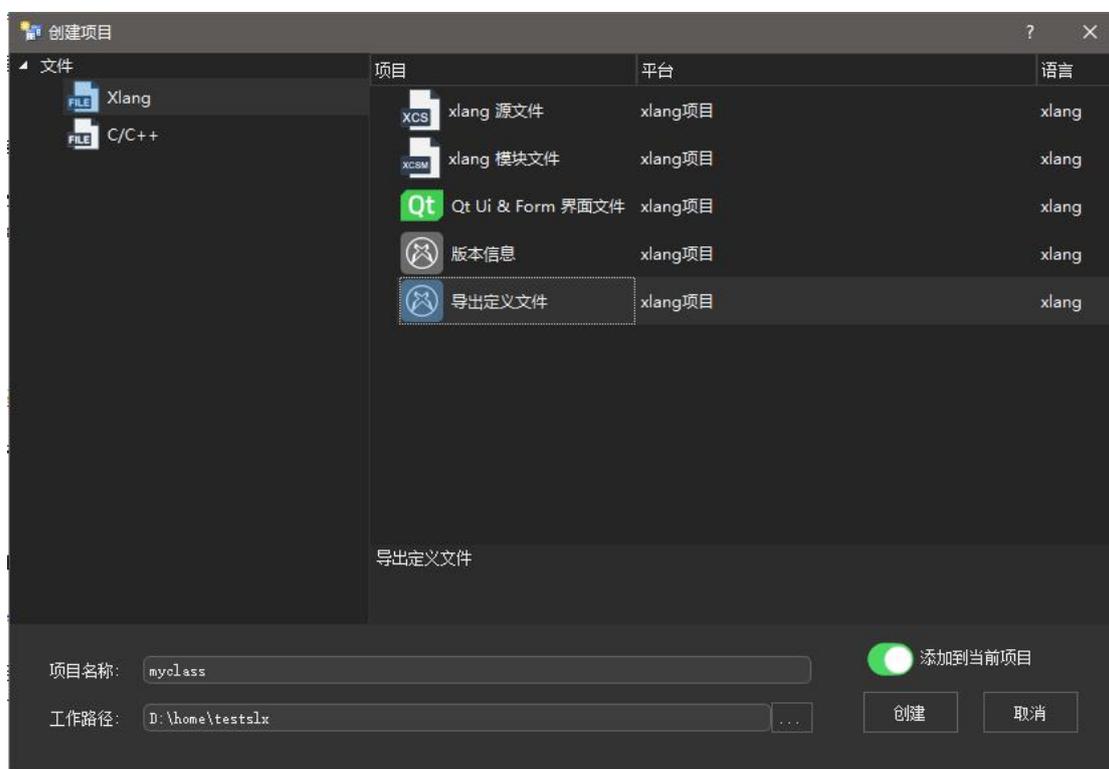
然后在需要使用的地方直接调用 `MyClass.output("test");` 编译生成即可;

二. 反向链接 (slx 调用主模块功能)。

1.主模块中有代码:

```
class MyClass {
    static void output(String text){
        _system_.output(text);
    }
};
```

2.点击新建添加 xdef 文件:



文件内容:

```
{
    "exports": ["MyClass"]
}
```

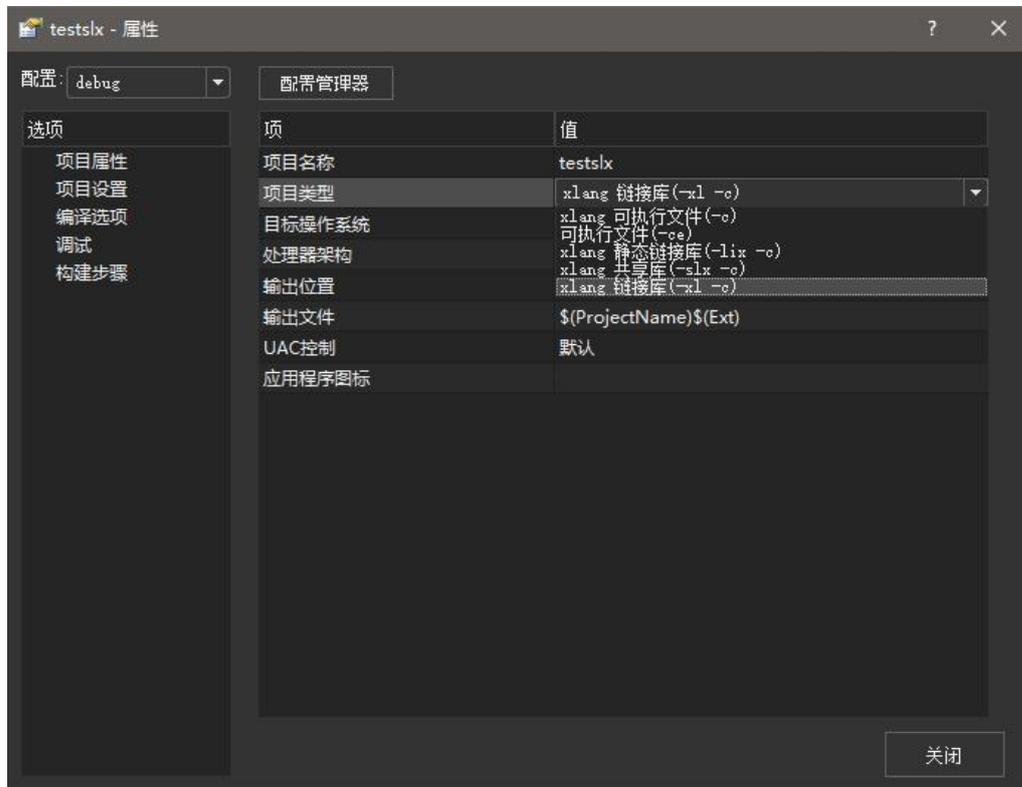
注意反向链接不需要指定 library。

exports 指明需要导出的对象名称。(如不指定默认全部导出)。

hidden 指定需要隐藏的对象名称(在 library 不指定的情况下, 该选项将隐藏不想导出的对

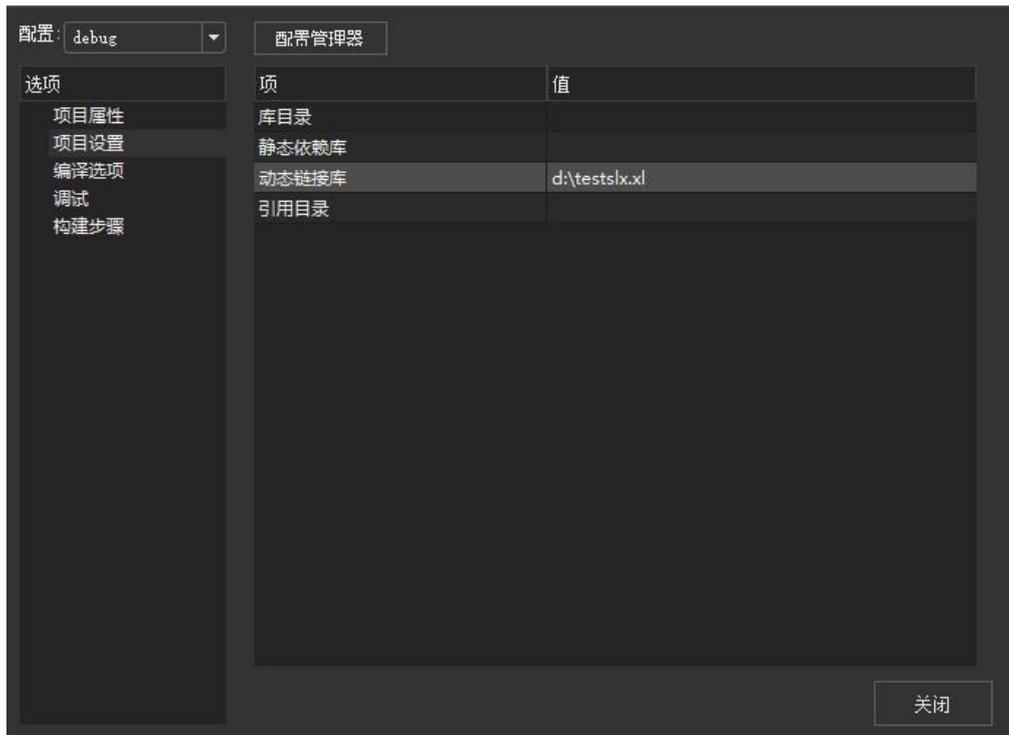
象)。

3.修改项目类型为动态链接库:



编译生成 test.xl 文件。

4.在需要使用此功能的共享库项目中添加链接库.



在主模块代码中使用 `loadLibrary("shared.slx")`; 加载动态库或者使用 xdef 文件将动态库链接到主模块中(方法为在共享库中添加 xdef 文件, 指明 library 名称, 如无导出对象则无需 exports 字段)

在需要使用的地方直接调用 `MyClass.output("test")`; 编译生成即可;

有任何疑问请进入  加入QQ群 [QQ 群: 591392649](https://www.qq.com/join/591392649), 进行交流探讨.