# 性能分析工具从xlua移植到tolua的说明

## 步骤：

1. 下载xlua的源代码，tolua\_runtime的源代码，以及 tolua工程代码
2. 复制 xLua\build\perflib.c文件到tolua\_runtime目录下
3. 复制 xLua\build\xlua.c文件中的部分代码到 tolua\_runtime\tolua.c中。如下：

(tolua.c)

Line 52[追加] static int hook\_index = -1;

static const char \*const hooknames[] = {"call", "return", "line", "count", "tail return"};

Line 1821[追加] static void hook(lua\_State \*L, lua\_Debug \*ar)

{

…（此处需复制xlua.c相关代码）

}

static int profiler\_set\_hook(lua\_State \*L)

{

…（此处需复制xlua.c相关代码）

}

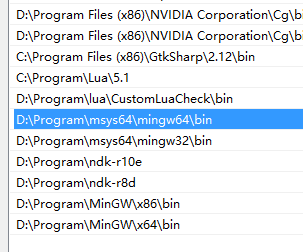
Line 2227[追加] { "sethook", profiler\_set\_hook},

1. 复制 xLua\Assets\XLua\Resources\perf 目录下的两txt文件到tolua\Assets\Lua\perf 目录下，并删除后缀名txt，为lua文件，修改profiler.lua中第8行xlua.sethook为tolua.sethook
2. 修改 tolua\_runtime\build\_xxx.sh文件，比如：

（build\_win32.sh）

Line 21[追加] perflib.c \

1. 重新编译 tolua\_runtime，生成成功后，拷贝到 tolua\Assets\相应目录下，注意 环境变量要设计成相应的mingw版本，比如要编译64位：



1. 修改tolua\Assets\ToLua\Core\LuaDLL.cs和tolua\Assets\ToLua\Misc\LuaClient.cs下文件，如下：

（LuaDLL.cs）

Line 229[追加] [DllImport(LUADLL, CallingConvention = CallingConvention.Cdecl)]

public static extern int luaopen\_perflib(IntPtr L);

（LuaClient.cs）

Line 67[追加] luaState.OpenLibs(LuaDLL.luaopen\_perflib);

1. 按照tolua\Assets\XLua性能分析工具.doc 文档测试与使用。