在LR中用它自带的FTP协议去录制FTP工具(例如cuteFTP等)的下载或上传时,虽然可以看到LR的录制条有捕获到东西,但是脚本里却什么都没有,这时录制无效时就需要自己手动去写脚本了.

下面是脚本,里面加了详细的注释,大家可以看看.因为只是示范,所以里面没有加集合点和事务,也没有做参数化.

**[plain]** [view plain](http://blog.csdn.net/sasafeng/article/details/8012154) [copy](http://blog.csdn.net/sasafeng/article/details/8012154)

1. Action()
2. {
3. //定义FTP对象
4. FTP ftp1 = 0;
6. //建立连接并登录
7. //这其中需要修改的部分是username:password@IPAddress
8. //username:password 不做多的说明,大家都知道
9. //IPAddress:这里可以写IP地址,也可以写成域名,如www.test.com
10. ftp\_logon\_ex(&ftp1,"FtpLogon",
11. "URL=ftp://username:password@IPAddress",
12. LAST);
13. //下载FTP上的文件到本地指定目录
14. //SOURCE\_PATH:是指FTP上文件对应的目录,一定要指定一个确定存在的文件对象
15. //TARGET\_PATH:下载的目的地址.这里要确定该路径存在,同时要指定文件名(可以和FTP上的文件名不一致).
16. //性能测试时,一般会把源文件名参数化,目标文件名使用源文件名的参数+唯一序列号,这样方便验证下载是否成功
17. //MODE:下载的模式,有二进制和ASCII这2种方式,系统默认的是二进制,避免有时下载的文件被损坏
18. ftp\_get\_ex(&ftp1,"Get\_Files",
19. "SOURCE\_PATH=/2项目组活动/0710.rar",
20. "TARGET\_PATH=e:\\Tools\\Dowload\\07101.rar",
21. "MODE=BINARY",
22. ENDITEM ,
23. LAST);
25. //上传FTP上的文件到本地指定目录
26. //PASSIVE:是指由客户端主动去连接服务端.
27. //一般来说,电脑都有防火墙,没有单独设置过,防火墙是不允许服务器端来主动连接客户端的,所以这里一般注明是客户端连接服务端
28. //更多关于PASSIVE的解释大家可以去google下
29. ftp\_put\_ex(&ftp1,"Put\_Files",
30. "SOURCE\_PATH=e:\\Tools\\Dowload\\07101.rar",
31. "TARGET\_PATH=/2项目组活动/07102.rar",
32. "MODE=BINARY",
33. "PASSIVE=TRUE",
34. ENDITEM ,
35. LAST);
37. lr\_think\_time(5);
38. //退出登录
39. ftp\_logout\_ex(&ftp1);
41. return 0;
42. }

具体步骤如下：

1、  首先要了解loadrunner中几个FTP函数

在loadrunner联机帮助文档的索引中，输入FTP，此时你会看到不下50个与ftp有关的函数，想了一下为了解决使用FTP文件上传这个问题，只需要一下几个步骤就可以了：

第一、   与ftp服务器建立连接

第二、   传输文件

第三、   关闭连接

所以我现在只需要关注与这三个步骤有关的函数就可以了

经过挑选终于找到了一下几个函数：

第一、   与ftp服务器建立连接的函数

       ftp\_logon\_ex (FTP \*ppftp, char \*transaction, char \*url, LAST);

       这个函数主要的功能是建立与ftp服务器的连接，其中的参数含义如下：

       Transaction：为这个连接起一个名字，在这里随便起

       url：[ftp://username:password@mailserver:port](ftp://username:password@mailserverport/)，指定连接用户名、密码、服务器地址、端口

第二、   传输文件的函数

       ftp\_put\_ex (FTP \*ppftp, char \*transaction, char \*item\_list, LAST);

       这个函数主要的功能是指定把本地的某个文件上传到服务器的某个目录下

       transaction：为这个操作起一个名字，在这里随便起

       item\_list：其中包括

SOURCE\_PATH,指定本地上传文件的路径

TARGET\_PATH制定要上传到服务器的路径

ENDITEM - Marks the end of the list.（不好翻译呵呵）

第三、   关闭连接的函数

       ftp\_logout\_ex (FTP \*ppftp);

       这个函数的主要功能是断开FTP链接

2、  模拟文件上传过程

知道了上面的函数我们把代码写一下：

**[plain]** [view plain](http://blog.csdn.net/sasafeng/article/details/8012154) [copy](http://blog.csdn.net/sasafeng/article/details/8012154)

1. Action()
3. {
5. unsigned long \*ftp\_session=NULL;
6. ftp\_logon\_ex(&ftp\_session,"ftp\_logon","URL=ftp://192.168.0.70:21",LAST);
7. ftp\_put\_ex(&ftp\_session, "Ftp\_Put",
9. "SOURCE\_PATH=d:/huruihai.txt",
11. "TARGET\_PATH=/coreftplite/huruihai.txt", ENDITEM,            LAST);
13. //释放FTP连接
15. ftp\_logout\_ex(&ftp\_session);
17. return 0;
19. }

运行一次，发现d:\huruihai.txt文件已经成功上传到FTP服务器E:\tools\coreftplite目录下，OK大功告成！

注意：FTP设置时设置的根目录为E:\tools\

好了，基本的结构写出来了，执行脚本，发现可以把指定的文件上传到制定的目录下了，接下来就根据自己实际需要强化脚本就可以了，参数化上传文件、加入事务、集合点，监测上传文件的时间和服务器的性能，强化脚本在这里就不多说了。

在这个过程中遇到了如下几个问题：

A、 在生命ftp连接句柄时，我看网上的一些资料中介绍声明称FTP ftp1 = 0 ；

但是这样写会出现语法错误，当我把它修改成unsigned long \*ftp\_session=NULL;后就没有问题了，其中原因不明

B、 当使用SSH加密后，使用loadrunner就无法连接成功，这个问题也没搞清楚原因，出现的错误提示是Logon Session failed SSH-1.99-OpenSSH\_3.9p1

C、 为什么loadrunner能捕捉到事件，但没有生成脚本？