MUSHclient 实用型教程

一次很无意的网络上乱逛，主要是为了找一些ＭＵＤ上的资料。不小心，发现了北侠，很喜欢这里，想留下来。既然决定留下来了，就应该把自己视为北侠的一份子，呵，换句话说，就应该为北侠做点什么。

mush 我也是从进了北侠开始学习研究的，在这个过程中，发现学习资料很难找，要么就是很高深的那种，要么就是不着边际的那种。自己也是花了不少力气才入的门。

一直想整一个比较详细的实用型 mush 教程，属于那种一看就明白，一看就知道怎么用的那种。也是那种高手不愿意写，初手写不了的那种。今天，终于开始写了，也算是对北侠的一种回报吧，呵……希望大家能够喜欢。

先说明一点，我个人的英文水平非常差，呃——差到了，软件一般只使用中文版和汉化版的。所以，所有的例子都以汉化版的 mush 4.18 为准。

还有一点，我个人以前用过很长一段时间的 php，本来 mush 也是支持 php 的，但这里好象没有什么人使用 php，而我在实际使用过程中，也的确发现 Lua 对于 mush 的契合性更好，所以，本教程中的实例都将以 Lua 为代表介绍。

最后，如果可能的话，教程中所有实例我都会使用真实有效的，希望对大家有帮助。

不多说了，我们开始学习吧。

基于网络上现在还没有一份比较完整的介绍 mush 使用的教程，所以写了这个教程。本文适合人群为。１、以前玩过 mud，至少使用过 Zmud，觉得 Zmud 功能不足够使用者；２、以前没有玩过 mud，但对各种软件的配置使用有一定基础者；３、现在正在玩 mud，对 mush 有一定兴趣，但被具较为麻烦的配置所困者４、看着 mush 机器人眼红却又不会用的无奈者（呵，这个是我乱说的）。

在写本文的过程中，参考了北侠论坛上的大量资料，并且对论坛上一些常见问题和奇怪问题，在文中做了统一说明，感谢北侠的 ddid，bigpswd，zgbl，没有你们的支持，这篇文字不可能出现（特别是关于正则部分，现在回想出来，居然写跟正则有关的教程，真的不是疯就是傻。不过，写完了回头看看，还不错嘛，有点小得意）。

祝大家使用 mush 顺利，祝大家玩 mud 开心。

小

目录

何在 mush 中避免这种问题。

3.一些常用设定（新人必看，老人必看）

想在使用 mush 的时候，跟 Zmud 一样得心应手吗？

4.自动连接（新人选看，老人选看）

自动连接，永远的话题，mush 中，有着更多的选择。

5.别名的使用（新人选看，老人选看）

很多mush新手头疼的地方,发现许多人放弃mush有一部分原因是因为这个。

6.触发的使用（新人选看，老人选看）

没有触发的 mud，是不完整的 mud，学会做触发，是玩好 mud 的第一步。mush 中的触发，比 Zmud 更为强大，同时也不是那么难学，为了让这一部分更容易被人看懂，小刀写了一个全新的正则教程（死了许多脑细胞），希望这个正则教程能够让你丢掉对正则的害怕。

7.抓取变量送到脚本（新人可不看，老人必看）

里面有小刀自认为很有针对性的代码，Lua 的、Js 的，php 的，如果能真正理解了，做出不一定强大但绝对有效的机器人是不在话下的。

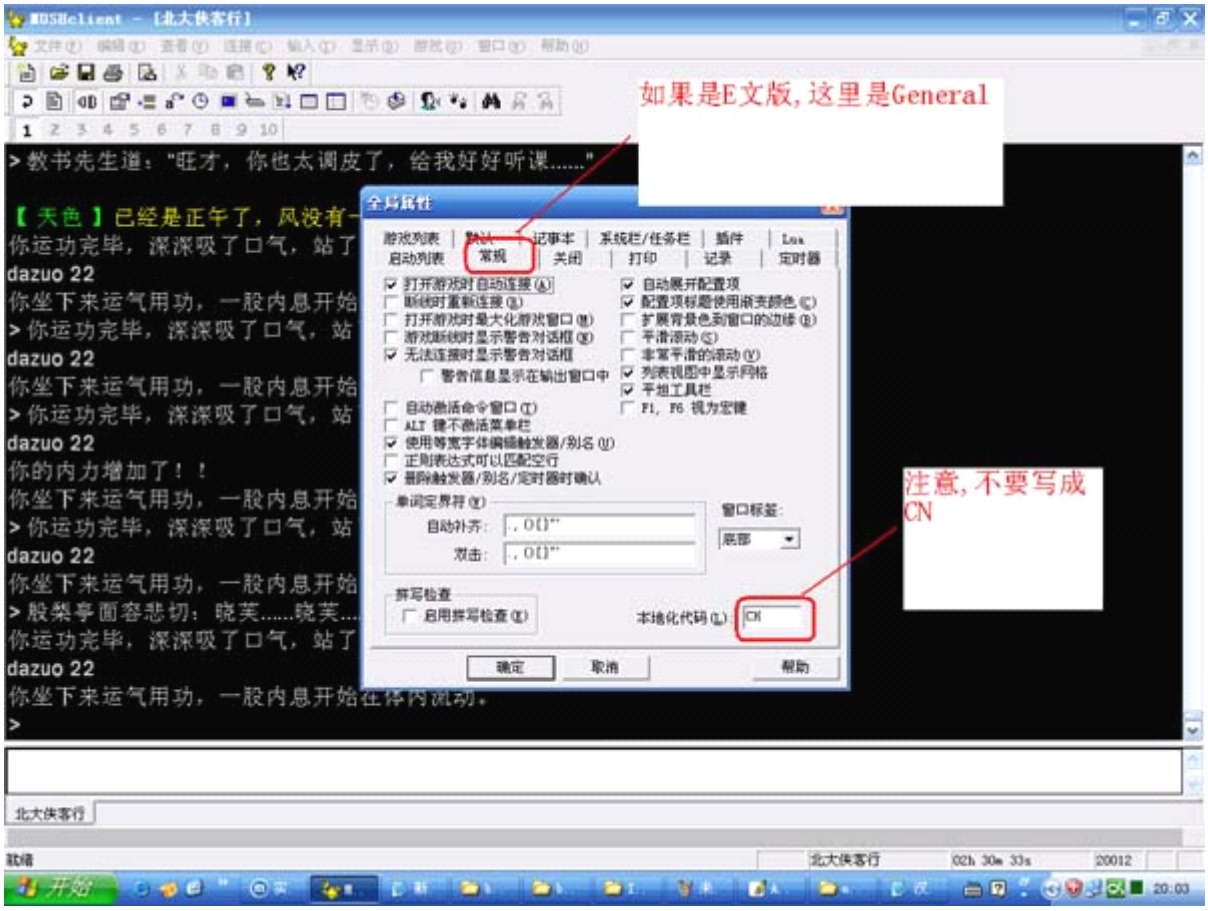
8.插件（新人必看，老人必看）

如果充分利用别人的成果？在这一部分内容里，有比较详细的讲述。

mush 在北侠就有下载，不过，版本相对旧了一些
http://www.pkuxkx.net/download/soft.php?id=6
我所使用的是 mush 4.18 汉化版,大家可以找一个这个版本的下载,应该差不多。

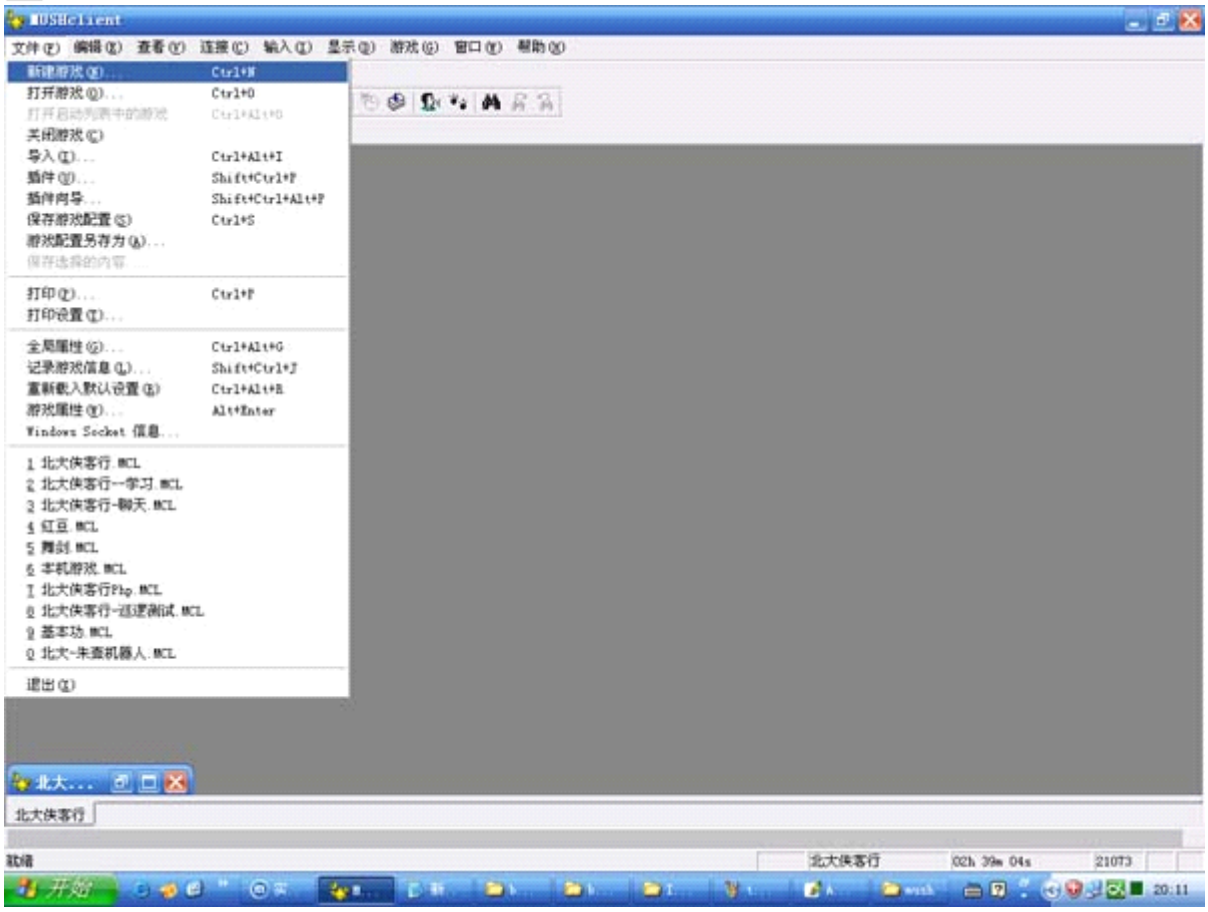
安装的过程就不用说了吧？相信大家都可以装上去，下面说一下汉化的过程。

把压缩包中的文件复制到 locale 目录下，然后运行 MUSHclient，按 Ctrl+Alt+G，在“General（常规）”对话框的右下角把语言代码改成 CH，保存后退出 MUSHclient，下次运行就可以看到效果了。



005.jpg (83.68 KB)
2009-11-3 08:06 PM

安装成功后，我们就要开始建立一个新的游戏。点击菜单－－文件－－新建游戏

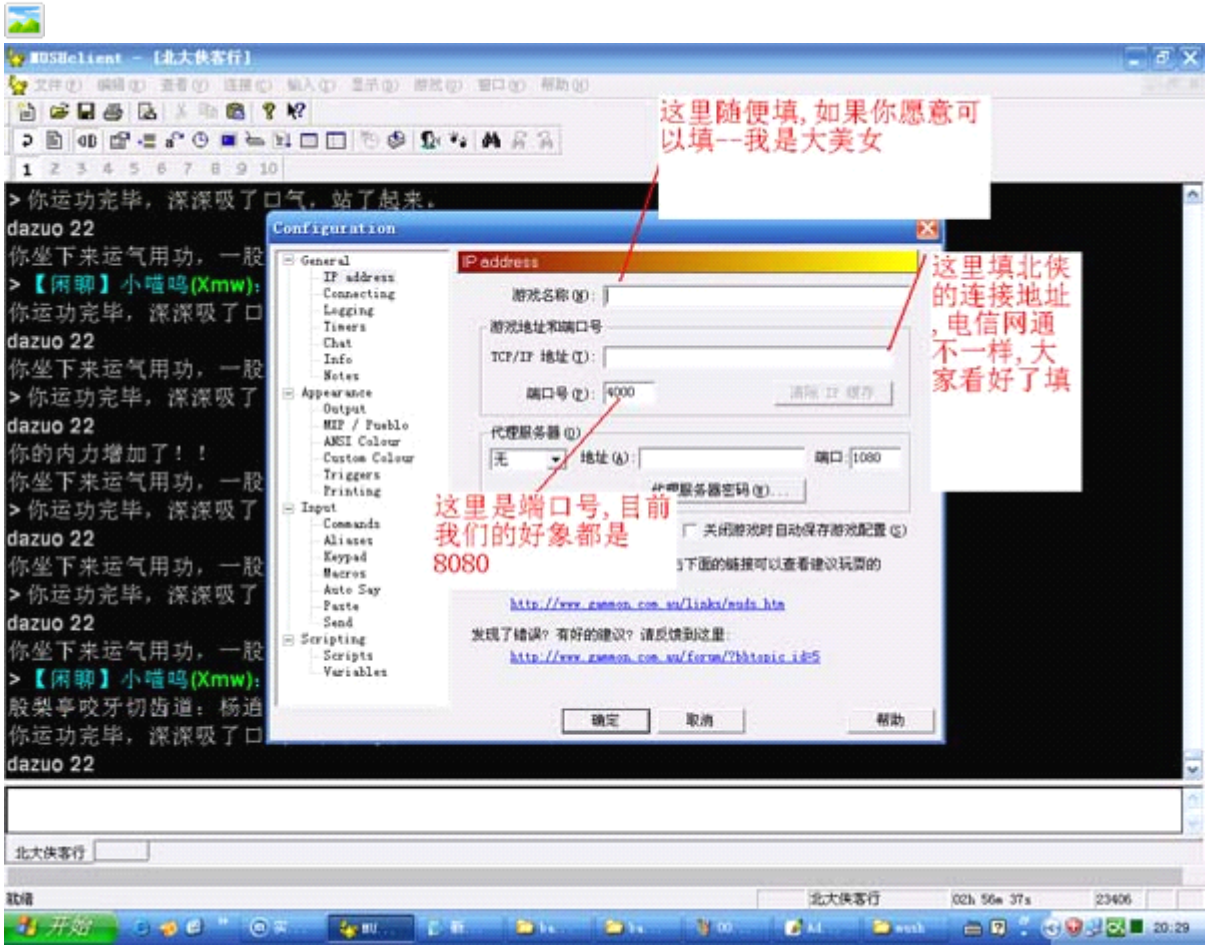


006.jpg (49.29 KB)
2009-11-3 08:14 PM

有些版本的 mush 安装完后，你点新建游戏，会出现下面这个提示，这其实是里面内置了一个游戏，我们不管它，直接点击否就可以了

temp.jpg (28.85 KB)
2009-11-3 08:21 PM

然后，我们就可以后面下面这个画面，按照北侠的连接填好后，就可以进入北侠了。



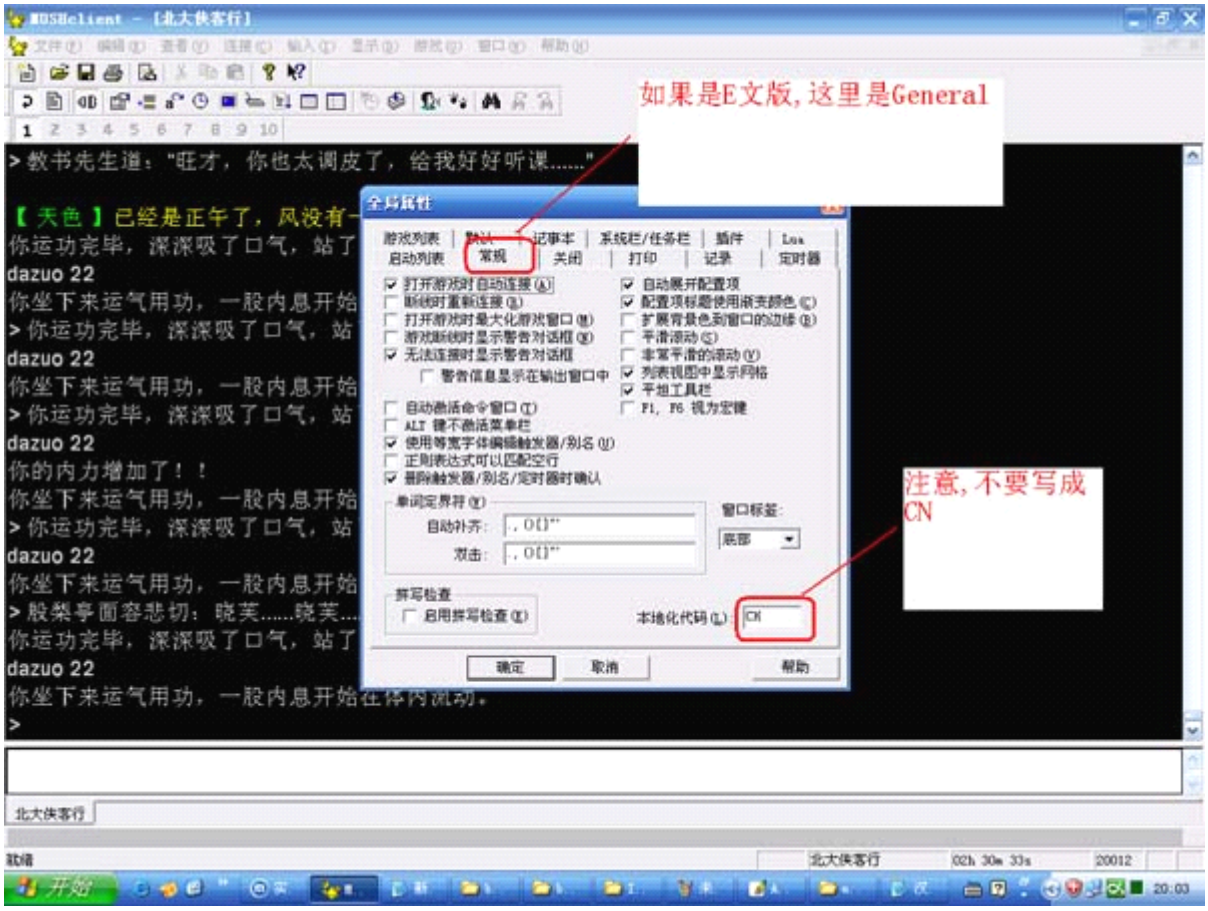
2. 一些资料的保存及注意事项

在 Zmud 时代，不知道，你有没有遇见过这样的问题——有时候，明明修改好的触发，做好的别名，第二次打开 Zmud 的时候，居然会不见！真的好奇怪……如果发生了这种事，除了生气，郁闷，你还能怎么办？

为了让这么要命的事情不在发生，考虑了半天后，还是决定加上这部分内容，如果不喜欢，可以直接跳过。建议新手还是看一下，也许会省你以后许多事情。

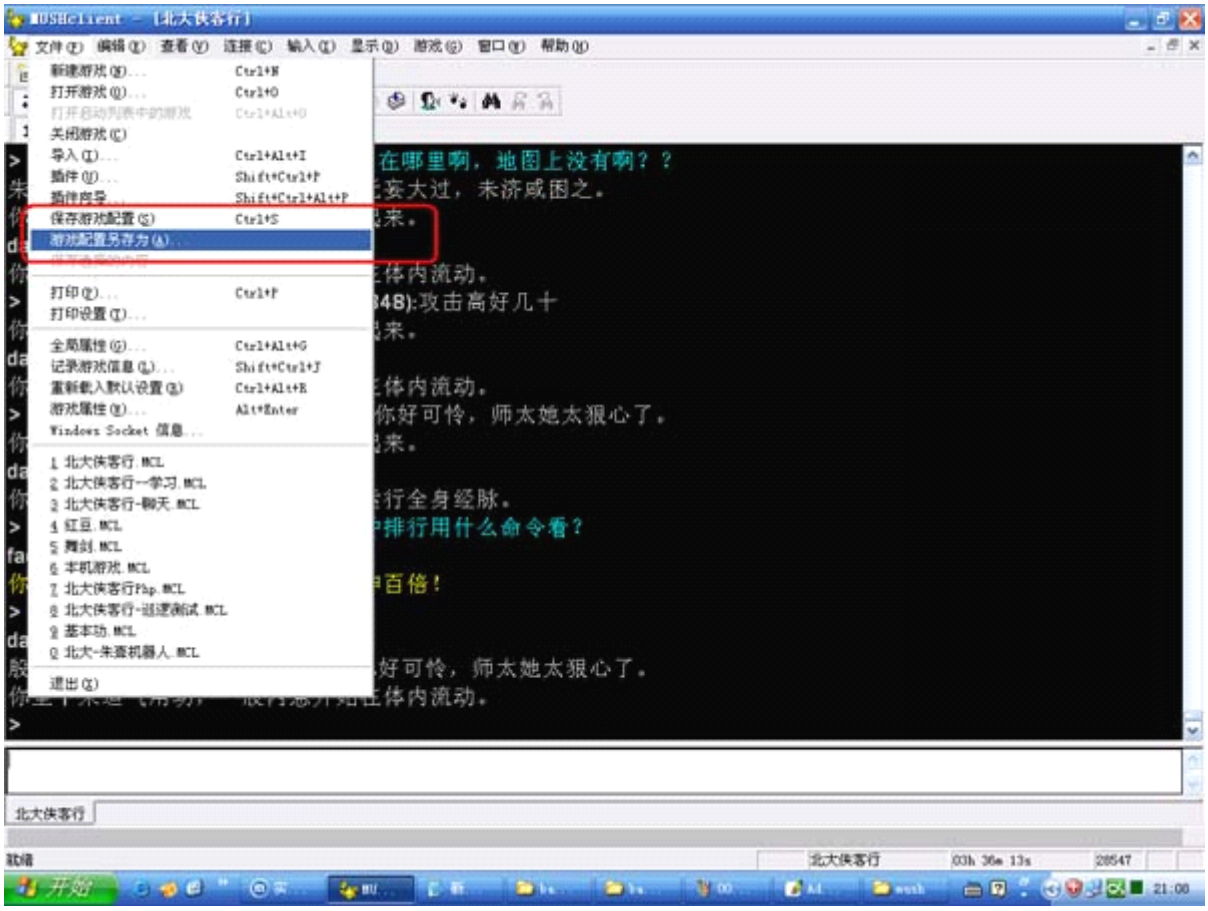
mush 中参数的保存分成两个部分，一个是 mush 本身的参数，也称全局属性，这是你的这个 mush 目录下所有游戏共享的，这里的设置，只要你改变，你所有游戏里的设置都会跟着改变。具体的位置是：点击菜单－－文件－－全局属性

出来的图是下面这个样子的



还有一个是游戏参数，每个在 mush 下的游戏人物（估且这么说吧）都有一个配置文件，这里面的内容都是不一样的，一般来说，每个新建人物，都需要重新设置。建议新手设置完后，一定要进行保存，保存的位置是在：点击菜单－－文件－－保存游戏配置（或者　游戏配置另存为）。

具体见下图。顺便说一下，该设置文件，一般是保存在安装目录的 worlds 目录下，扩展名为 mcl。

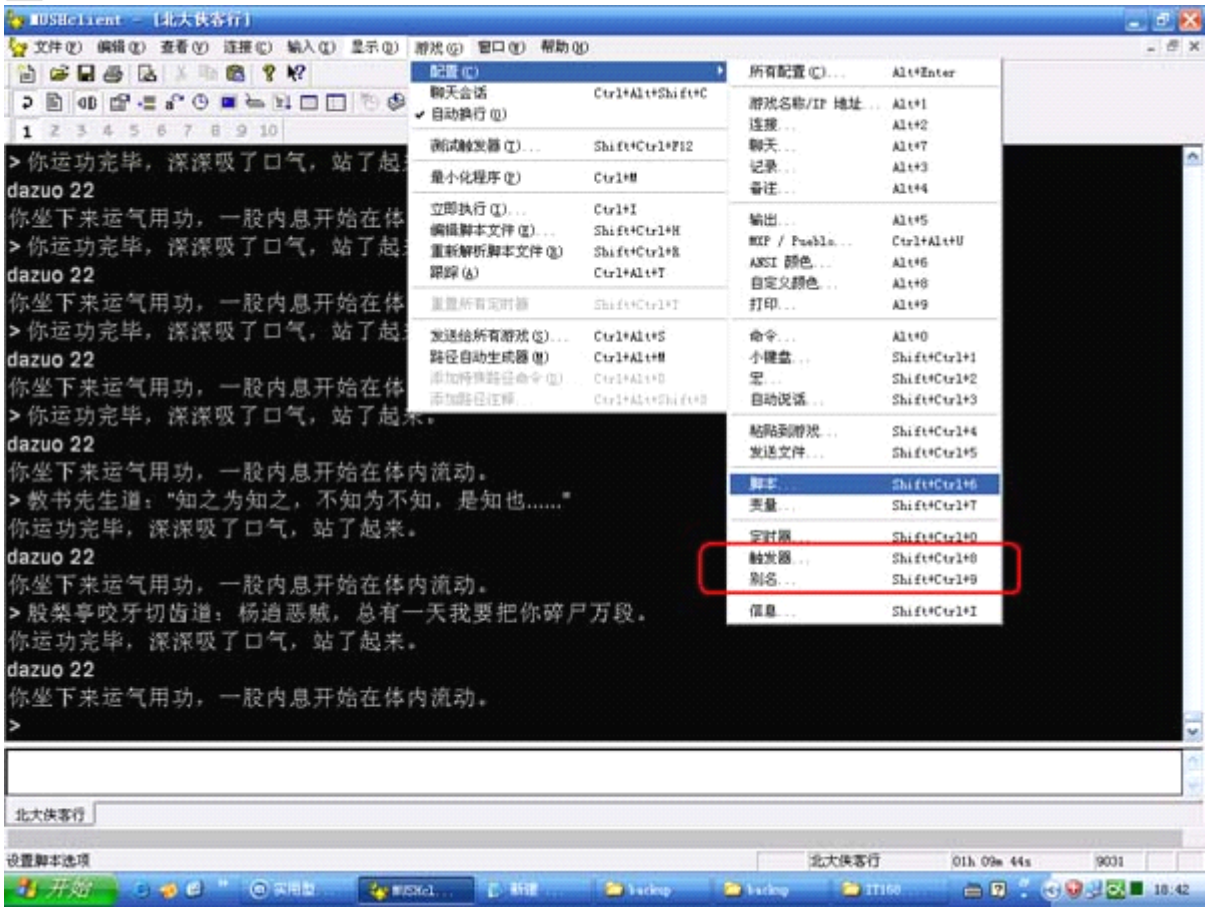


008.jpg (64.25 KB)
2009-11-3 09:12 PM

但，仅是保存游戏配置文件还不够的，因为，我们还有两个东西需要保存，相比之下，这两个东西才是我们更关心的。当然，聪明的你，一定可以猜到了，一个是别名文件，一个是触发文件。

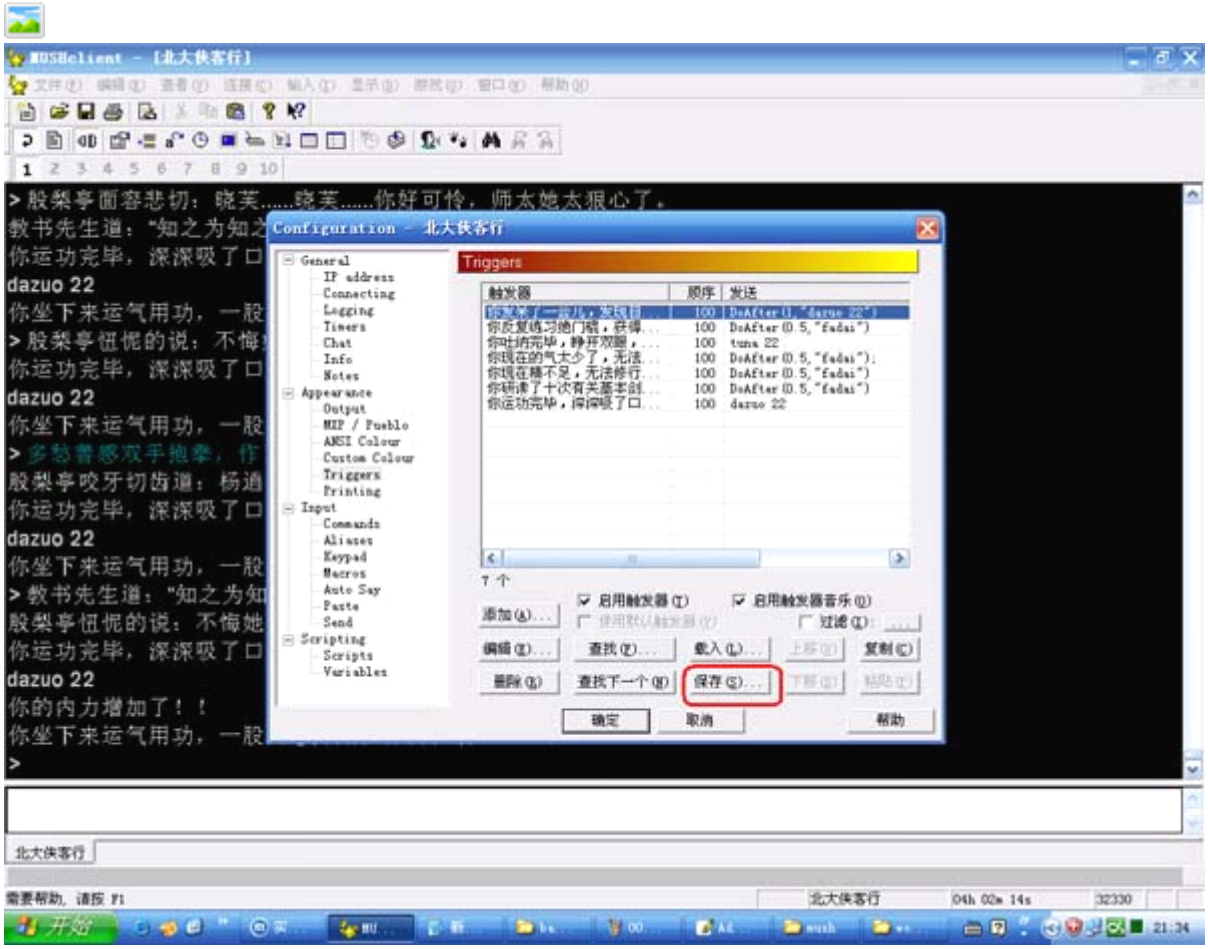
这两个文件的入口，分别是：点击菜单——游戏——配置——别名（触发器）

具体见下图。



009.jpg (86.32 KB)
2009-11-3 09:28 PM

以触发器为例，讲一下，如何保存触发器文件。请大家看下面的图。触发文件和别名文件一般是保存在安装目录的 worlds 目录下，触发文件的扩展名为 mct，别名文件的扩展名为 mca。



010.jpg (79.97 KB)
2009-11-3 09:37 PM

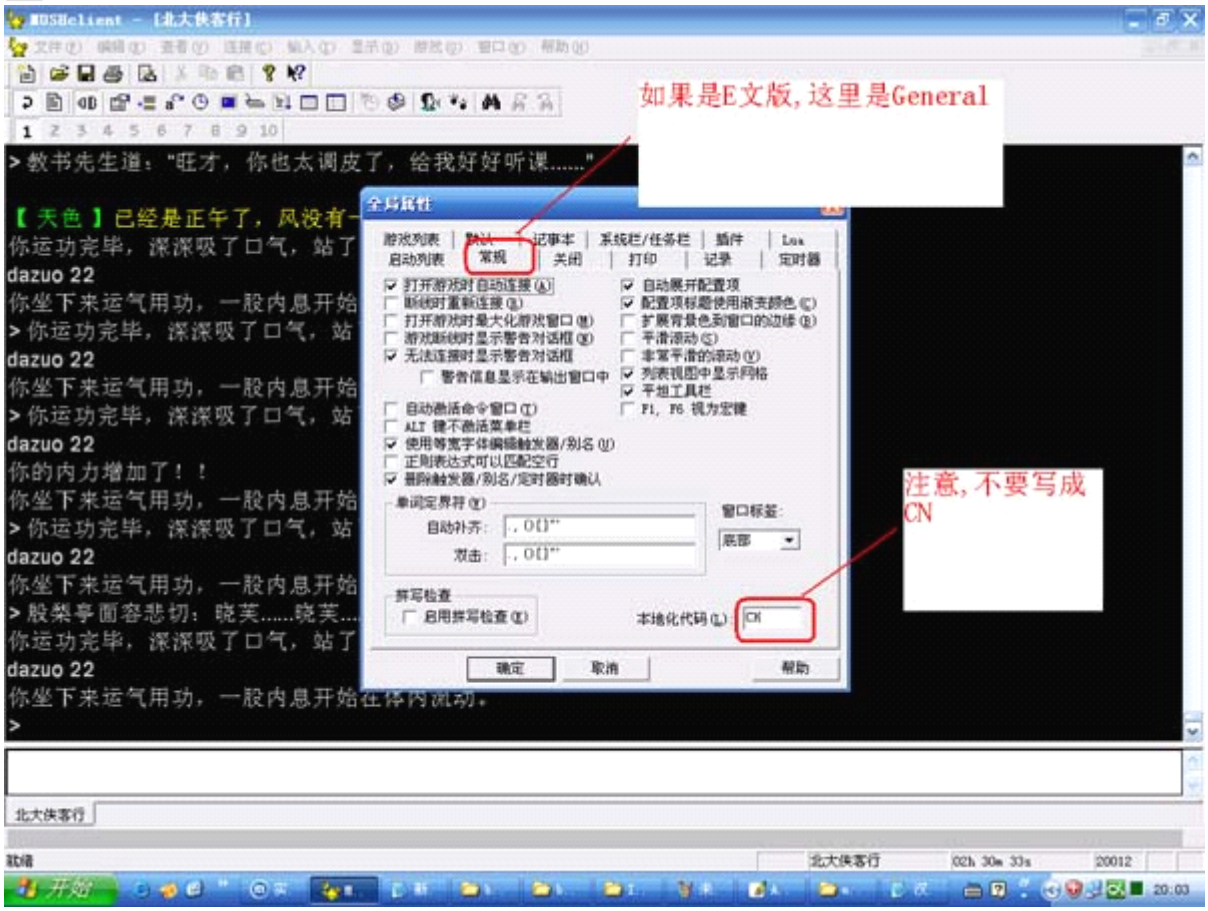
另外，请注意，使用别人提供的机器人的时候，不要忘了载入对方提供的游戏配置文件，触发文件，别名文件（如果对方有提供的话），不然的话，会导致机器人出错，或者无法使用。

3.一些常用设定

mush 的设定种类繁多，功能也繁多。作为初级教程，这部分的内容有一个原则——按照 Zmud 的习惯来——也就是说，Zmud 是什么样的，教程中就安排 mush 也设置成什么样。

首先，我们来看全局设置。

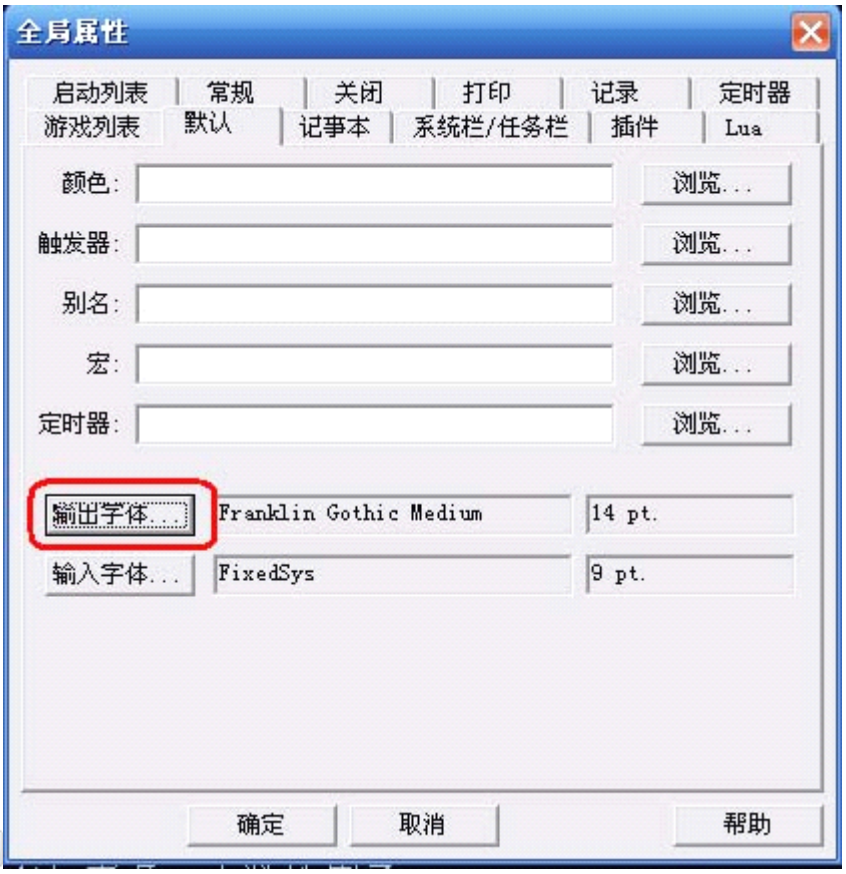
点击菜单——文件——全局属性，打开全局设置菜单，如下图：



005.jpg (83.68 KB)
2009-11-8 07:42 PM

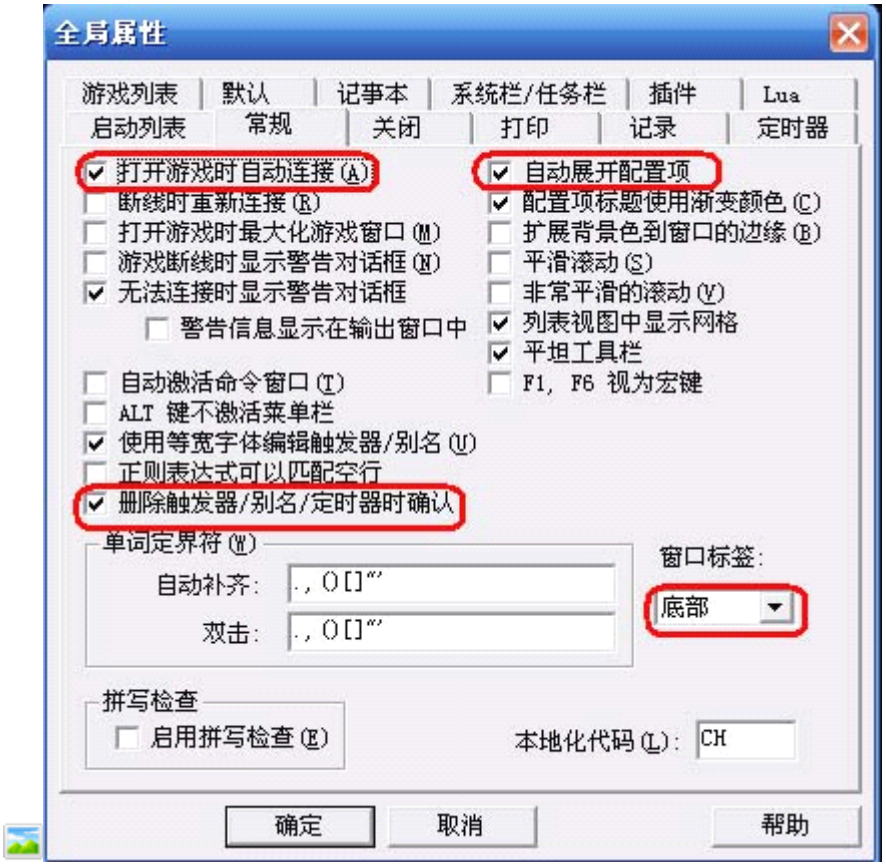
第一，我们选择——默认，在这里，可以设置屏幕上的字体等选项（见下图，

点击红色标出部分，可以设置屏幕字体）：



011.jpg (61.34 KB)
2009-11-4 07:09 PM

第二，我们选择－－常规（见下图，内容较多，一条一条说）



012.jpg (91.38 KB)
2009-11-4 07:21 PM

打开游戏时自动连接

打开游戏时自动连接到游戏服务器。

自动展开配置项

一个游戏有很多配置项，每个配置项又有一些子项，当打开游戏配置窗口时，如果选中了这项，这些配置项就会自动展开。

删除触发器 / 别名 / 定时器时确认

以防止不小心的误操作。

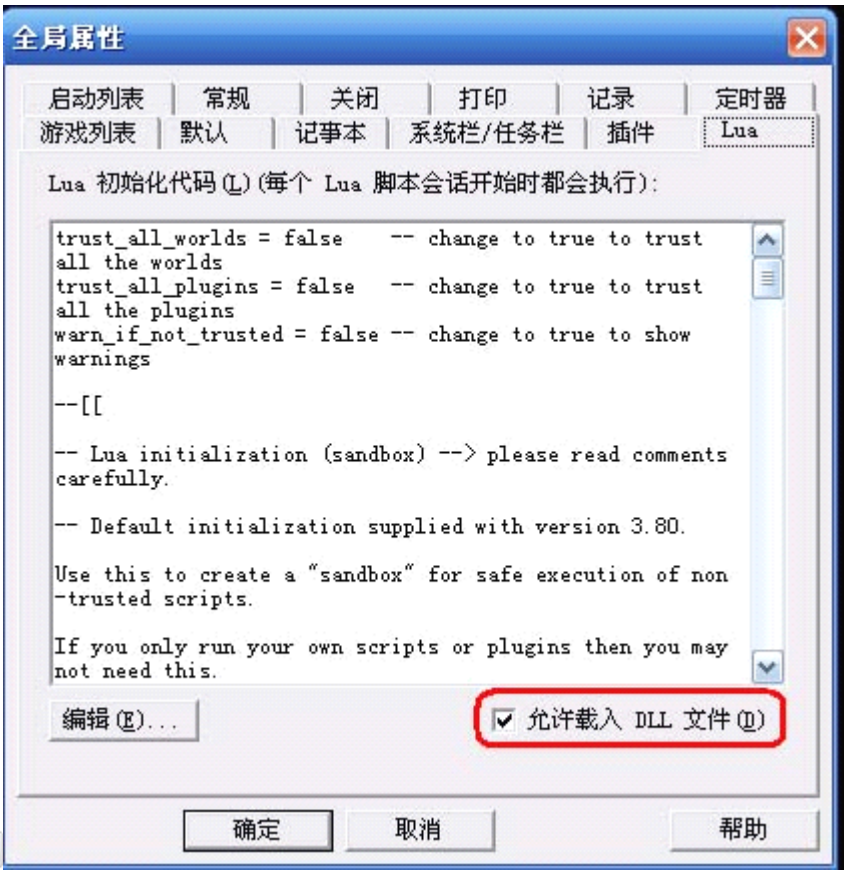
窗口标签（选择底部）

设置是否显示窗口标签栏。当打开的窗口比较多时，通过标签栏切换窗口是很方便的，你可以隐藏它，或者显示在窗口顶端或底端。如果标签对应的窗口是游戏窗口，且不是活动窗口，当接收到了新的信息，标签上还会显示新消息的行

数。如果窗口是记事本窗口，且内容已经被改变但未保存，标签上会以星号标识出来。

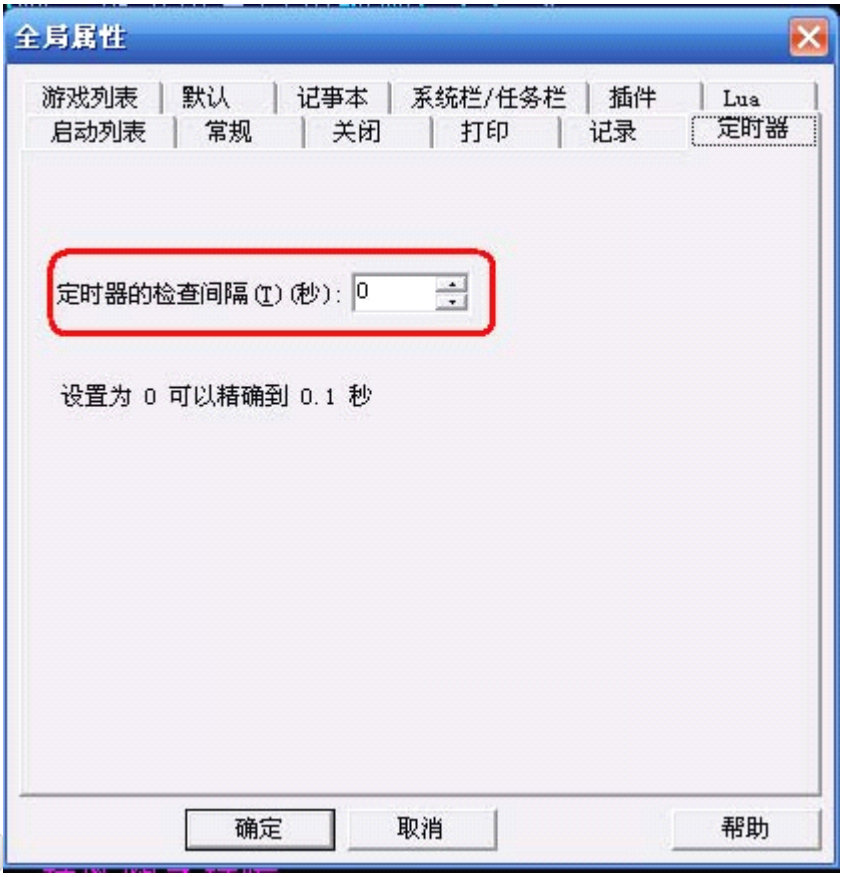
这个选项会在下次启动 MUSHclient 时生效。

第三，我们选择——Lua（下图红色标出部分勾上，主要是为了以后写脚本方便）



013.jpg (79.59 KB)
2009-11-4 07:35 PM

第四，我们选择——定时器（把下图中红色标出部分，改为 0 秒，这主要是为了几个函数需要这么设置）

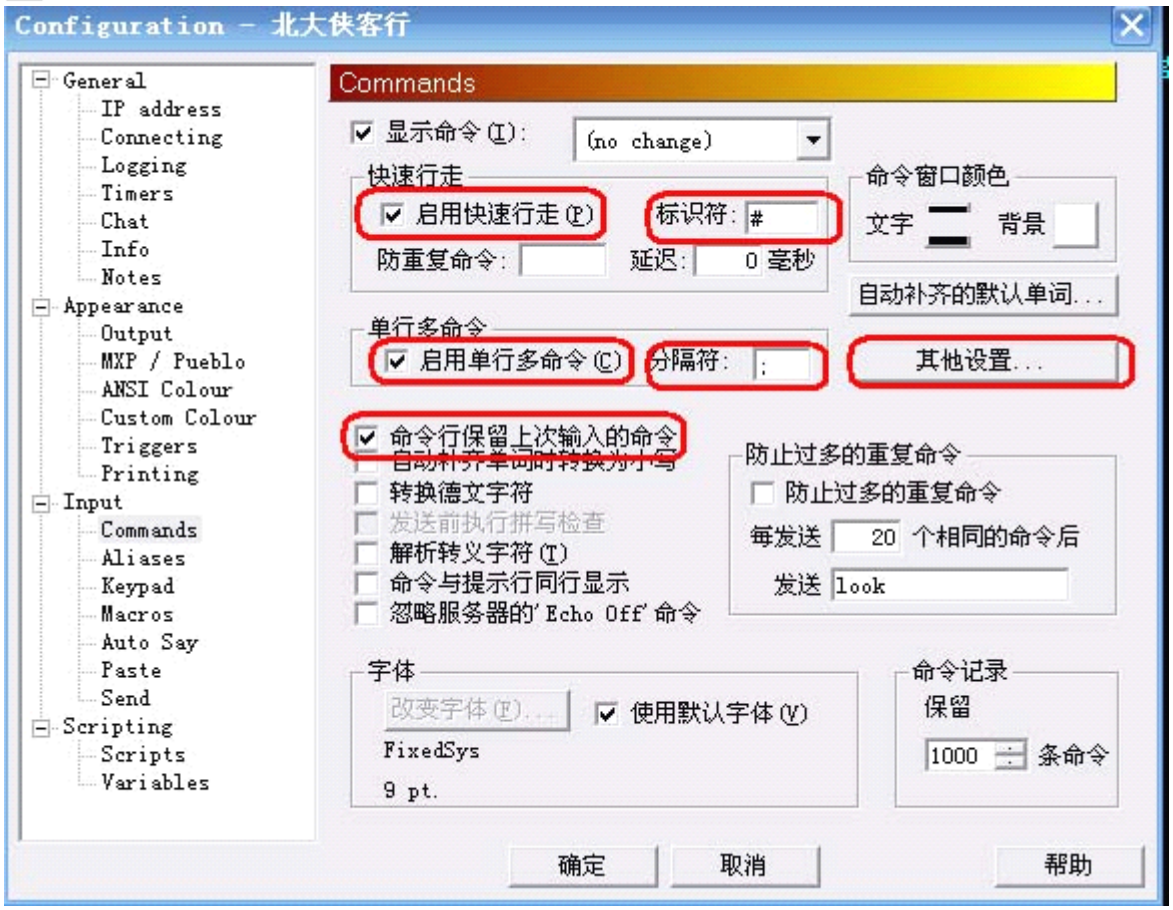


014.jpg (48.17 KB)
2009-11-4 07:39 PM

好了，到此为止，全局属性都设好了。下面我们开始设置游戏属性。

－－－－－－－－－－－－－－　我是喝醉酒的分割线　－－－－－－－－－－－－－－－－－－－－－－

第一，我们选择－－Commands（见下图，内容较多，一条一条说）



015-1.jpg (97.32 KB)
2009-11-4 07:56 PM

快速行走

快速行走是一种从一个地点快速移动到另一个地点的方法。一旦你在配置对话框中设置了快速行走的标识符，并启用了快速行走功能，你就可以在命令窗口中输入这个标识符 + 快速行走路径来快速移动角色。例如：

#4n3esuwd

这个路径命令会让你想北走 4 步，东走 3 步，南，上，西，下

这里跟 Zmud 是有不同的，Zmud 里，上面的命令应该是#4 n ;#3 e;s;u;w;d。请大家记住，Zmud 里的#是表示下面的命令重复几次的意思，mush 里的#（你也可以设置成别的符号），表示——下面开启快速行走。

启用单行多命令

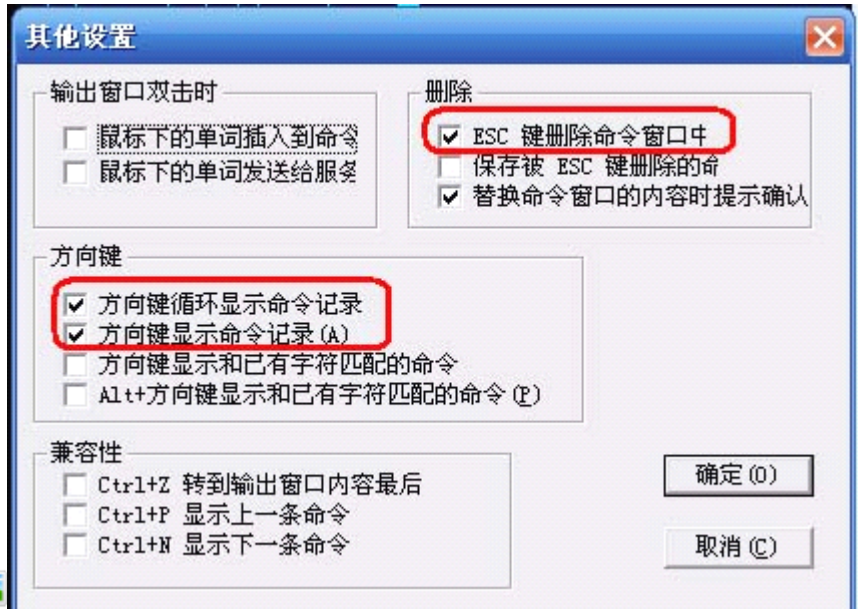
选择了这个以后，你可以在 mush 输入窗口里输入一大堆如 buy jitui;eat jitui;eat jitui 这样的命令。而不是只能输一个命令了。

命令行保留上次输入的命令

这个选择主要是为了跟 Zmud 保持风格的统一。

其他设置

这个看下面的图。



015-2.jpg (61.26 KB)
2009-11-4 08:22 PM

这三个红色标出部分只是为了跟 Zmud 风格统一，应该没什么好讲的吧？

有人问起，命令输入以后，按第一个字母无法回到历史的那个命令，在这里补充一下，大家把方向键显示和已有字符匹配的命令，那个勾也勾上，就可以。效果嘛，跟 Zmud 一样。

好了，到这里，mush 最基本的设置都讲完了，这么一番设置下来，mush 不说用的很顺手，也和 Zmud 比较接近了。

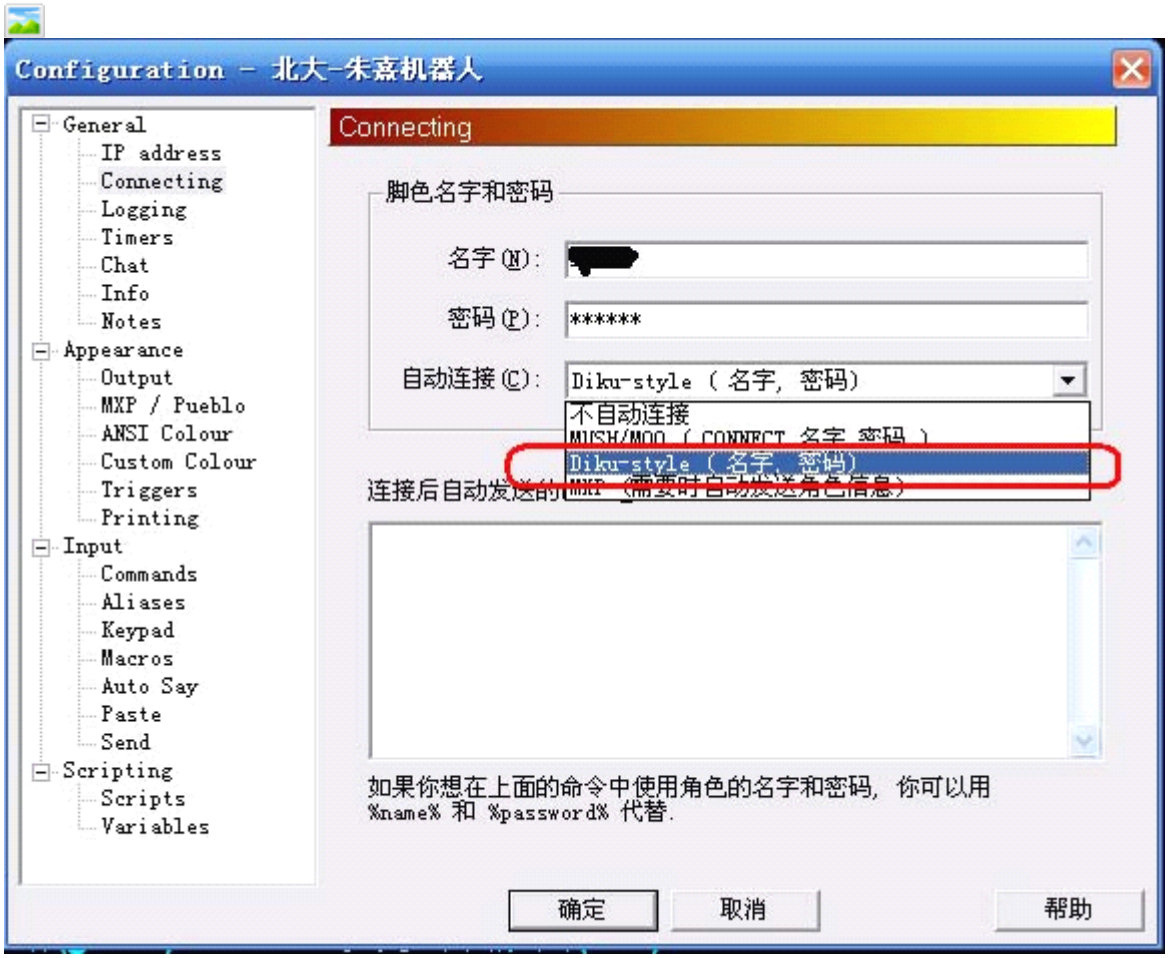
4. 自动连接

自动连接，具有很广泛的作用，这种废话就不多说了，我们整点干的。

在 mush 中，自动连接可以有 3 种表现方式：第一种是处理直接登陆；第二种是处理有些 mud 在登陆前先让你输点什么(最常见的是,让你输入 gb or gb5)；第三种是使用变量换值的方法，好处是，当你的机器人完成所有任务的时候，可以改变用户名和密码的值。这样，就不会一直拚命连进来了。

好了，我们来讲一下，第一种和第二种连接方式。呃……如果，你问我第三种，我一时没研究出来，等我研究出来了，会再更新了。

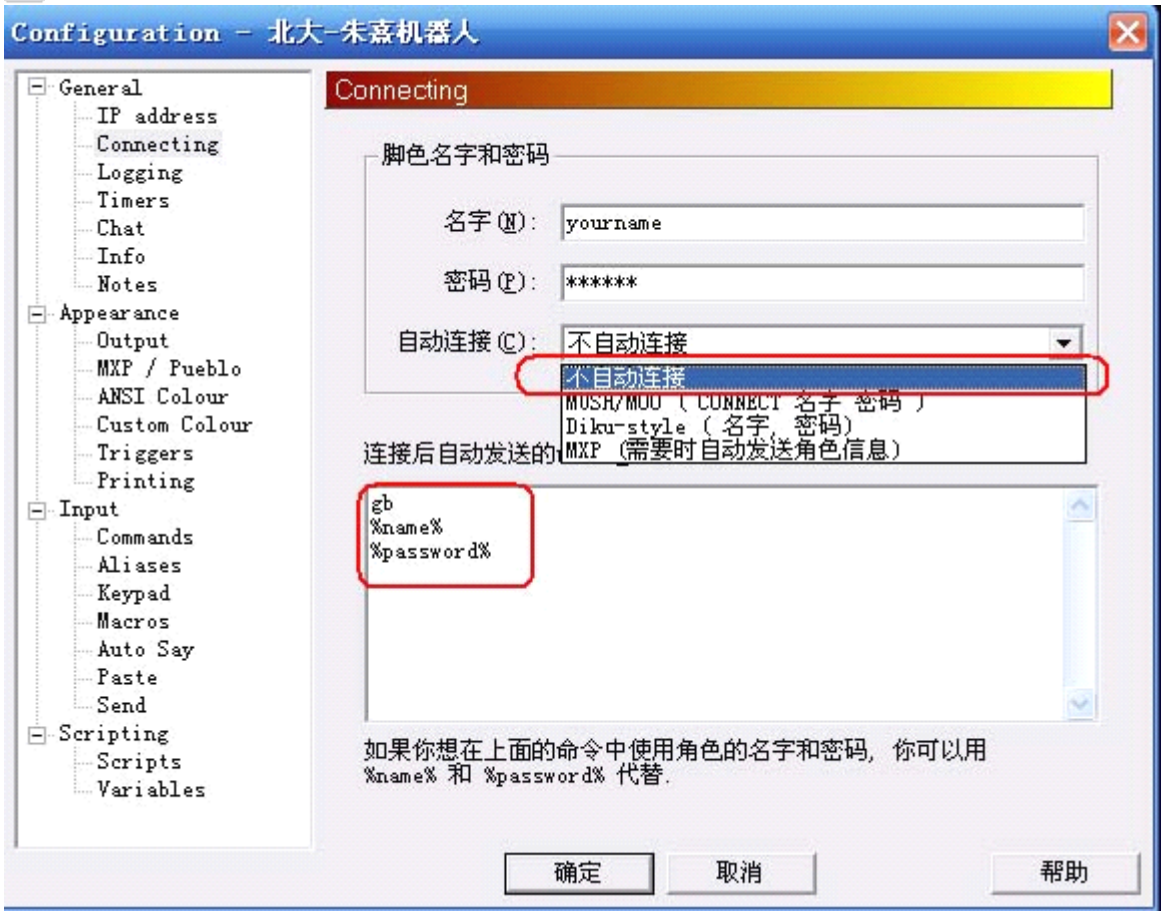
点击菜单－－游戏－－配置－－连接，如下图：



022.jpg (78.08 KB)
2009-11-7 09:50 PM

填上用户名，密码后，选择红色标记部分，再点确定，下次连线的时候，就可以自动连接了，很简单吧。

好了，我们讲第二种情况，同样见下图：



023.jpg (80.5 KB)
2009-11-7 09:56 PM

呃，下面就是昭着红色标记部分修改就是了。其中，用户名用%name%代替，密码用%password%代替，这主要是为了安全。

好了，自动连接暂时就讲这么多了。

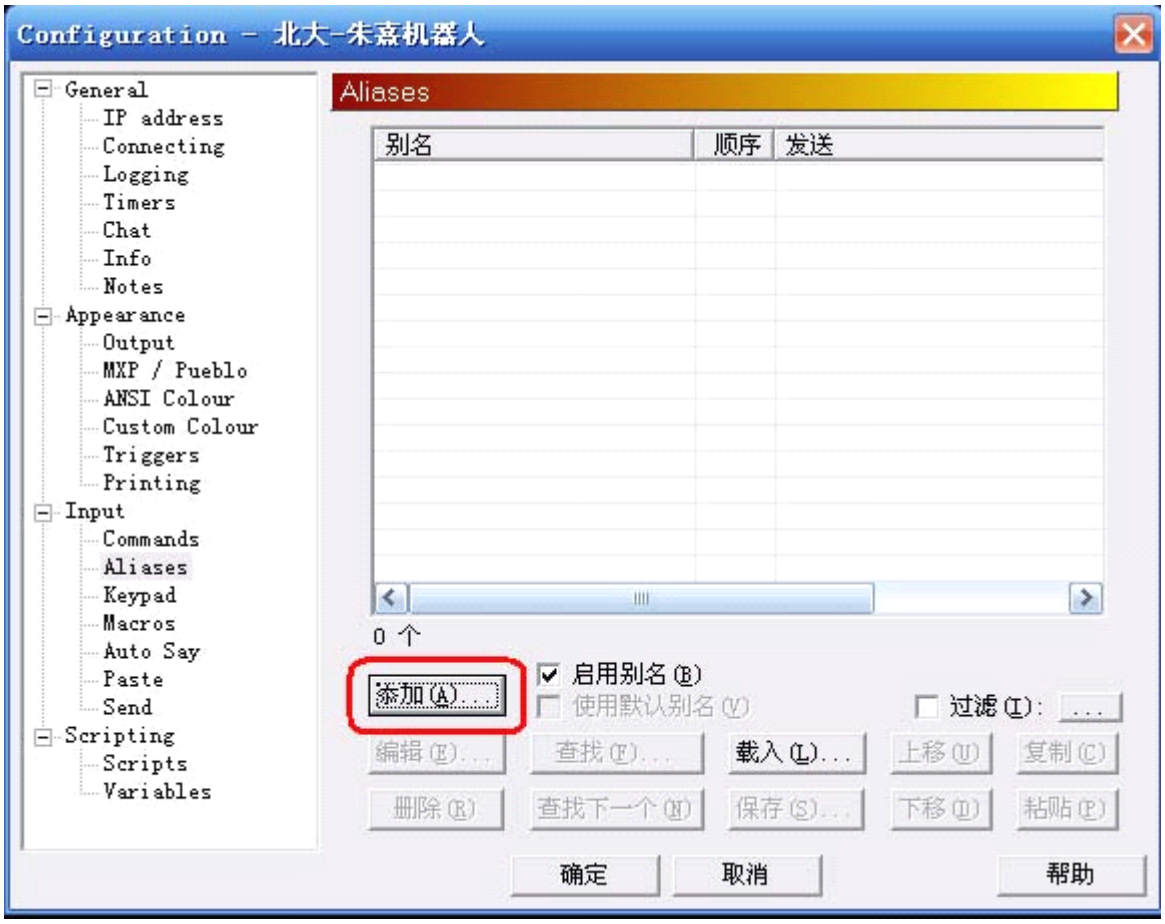
5.别名的使用

别名，也就是 alias，是我们玩 mud 的一大利器了。通过别名，我们可以做到，快速度行走，快速度拣东西，快速度杀怪，快速度泡ＭＭ（呃……这个是调节气氛来的。不过，你 mud 玩的好，自然泡到 MM 的成功率也会增加）。

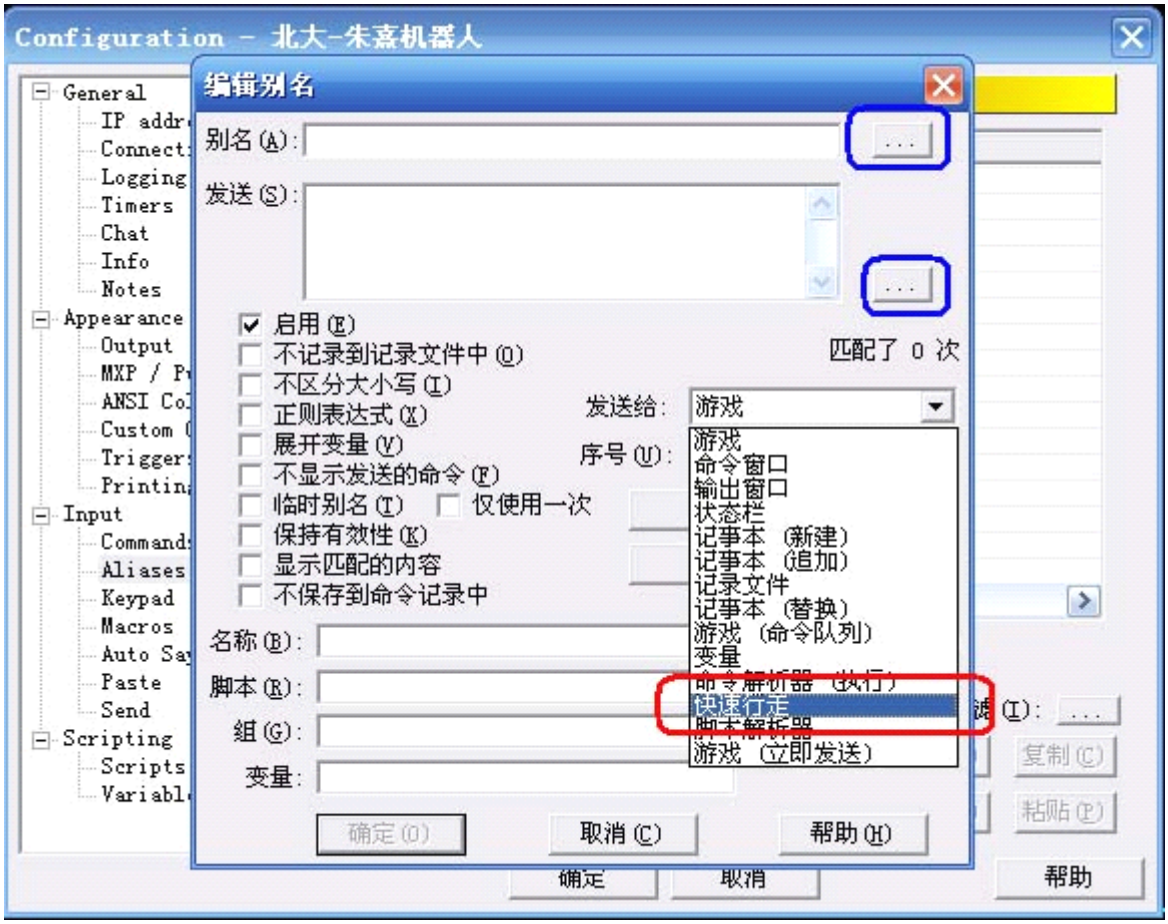
我们以前用惯的 Zmud 中的别名，跟 mush 中的别名是大不一样的。下面，我们来慢慢学习。

（１）mush 中别名的建立

点击菜单－－游戏－－配置－－别名，可以进入别名设置画面（怎么点的，就不抓图了，如果看到这里连这个都不会，那就应该反省了），具体如下图：



点击红色标志部分可以添加一条新的别名，如下图。



蓝色标志部分，是点击可以展开输入更多内容的。

红色标志部分，一般是选快速度行走，也可以选游戏，具体的区别，我们下面就讲到。

好了，剩下的，大家都应该会填了，我们继续下一步。

（2）mush 中快速行走与 mush 中单行多命令的区别与联系

这句话好象有点绕的感觉哦。个人认为，这是一个比较容易混淆的地方，希望能尽自己的努力，让大家弄的更明白一点。

两者的联系

快速行走和单行多命令都可以同时向 mud 服务器发送一连串的命令，达到快速行动的目的

两者的区别

单行多命令的表现方式为 n;s;ask zhu about question……，但这一串命令，在快速度行走中是不能被通过的。也就是说，在 Zmud 里，把一串类似 n;s;ask zhu about question 命令直接放进别名里，是行不通的了。

（３）mush 中，快速行走的使用（这部分比较复杂，建议慢慢看）

（a）方向命令

默认可以识别的方向命令有：N:north，S:south，E:east，W:west，U:up，D:down。也就是说，大家可以用 n 来代表 north，mush 是认的（这个应该没问题哦，大家一直在用）。

（b）动作命令

默认可以识别的动作命令有：C:close，O  pen，L:lock，K:unlock。也就是说，大家可以用 o 来代表 open，mush 也是认的（这个其实没什么用，我们一般不会就用一个 open 命令）。

（c）多次执行的语法

在一个命令的前面还可以添加一个 1 - 99 的数字，用来表示执行这个命令多少次。例如，“5s”意味着向南走 5 步。

路径中可以包含空格，但是会被忽略。虽然可以不写上空格，但为了阅读方便，建议还是把空格加上，后面会给出具体的例子。

（d）特殊路径命令

如果需要执行其它命令（例如，ne 为 向东北方走一步），你必须把这些命令放在括号中。否则 ne 就会被解析为先向被走一步，再向东走一步。假设你想向北、西、东北各走一步，然后买个酒袋，其路径命令为：

n w (ne) (buy jiudai)

（４）Zmud 中的别名，mush 中的快速度行走，mush 中的单行多命令的具体演示

好了，学习了不少了，现在我们来做个实战练习。

假设，我们现在在扬州客栈，我们希望走到阿庆嫂那里接 murong 任务。

Zmud 下

w;#3 s;w;ask fu about job

mush 单行多命令

w;# 3s; w;# (ask fu about job) 或者 w;s;s;s;w;# (ask fu about job)

mush 快速度行走

w 3s w (ask fu about job)

呃，现在大家应该很清楚这三者之间的区别和联系了吧？很显然，快速度行走功能最差，不建议大量使用（其实，也没可能大量使用，快速行走都是在命令窗口输入的，一般来讲，很少会有人一直在命令窗口不停的输入）。

余下的两种方式都很不错。但，经过这么一比较，一个新的问题又产生了——我以前是用 Zmud 的，现在准备换成 mush 了，那我以前的路径怎么转换呢？我们往下看。

（５）Zmud 和 mush 的路径之间的转换

把 Zmud 的路径变成 mush 的路径，有这样几种方法。

（a）手工转换，这是按照我上面给出的格式，一个一个的转，优点是可以锻炼自己的耐心、毅力，缺点是比较累，比较麻烦。

（b）北侠的机器人版，有位高人写了一个转换程序，只要把 Zmud 的路径输入，可以自动得出 mush 路径。优点是比较方便，缺点是一条一条拷贝，也挺累的。

（c）如果你使用 php 作为你的脚本语言的话，我提供了一个函数，可以直接把 zmud 转换为 mush 路径，优点是比较方便，缺点是——php for mush 配置比较麻烦，当然如果你喜欢 php 可以参考我的另一篇帖子，里面有比较详细的配置方法 。

（d）呃，或者，你可以直接求北侠的几位高手，写几个 for Lua, for Js 等等的函数，这样，大家都有的用，不过，好象也比较麻烦。

（e）还有一个方法，北侠提供了一个 mush 的路径插件，一般该有的路径都有了，我们会在后面介绍如何使用插件，就拿这个做例子。

好了，到这里为止，所有关于别名、路径的内容都讲完了，如果你都明白了，那至少在 mud 里飞快的跑来跑去，是不成问题了。

下面要讲触发了，还要涉及到正则，很要命。难倒是不难，主要是怎么讲清楚、讲透彻，是个问题。

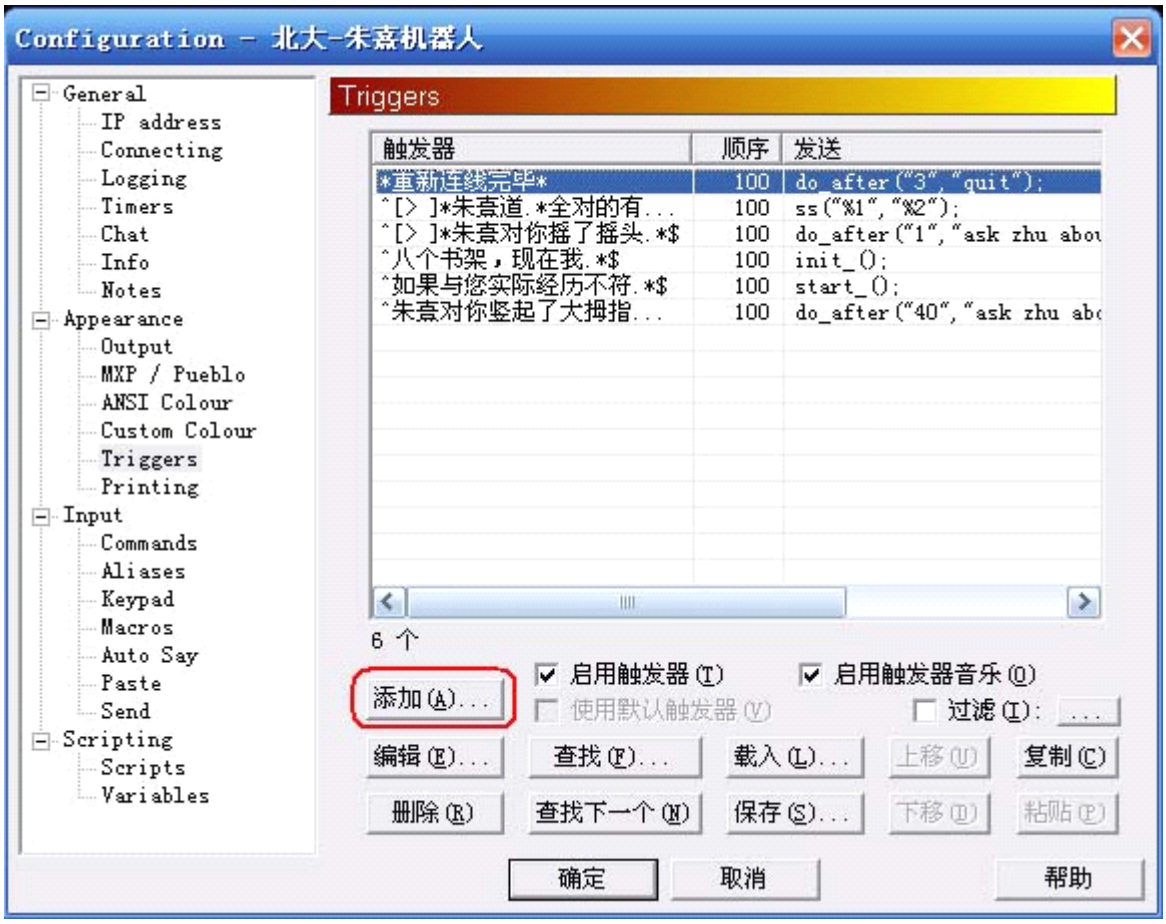
6.触发的使用

mud 中的机器人，８０％（或者更高的比例）的工作，是由触发来完成的。和 Zmud 相比，mush 中的触发的功能变得更为强大，更为复杂，更为优秀，也更为的不好学，其难学的程度，不亚于泡一个非常出色的，可以打９５分以上（满分１００）的漂亮ＭＭ……呃，还是调气气氛。

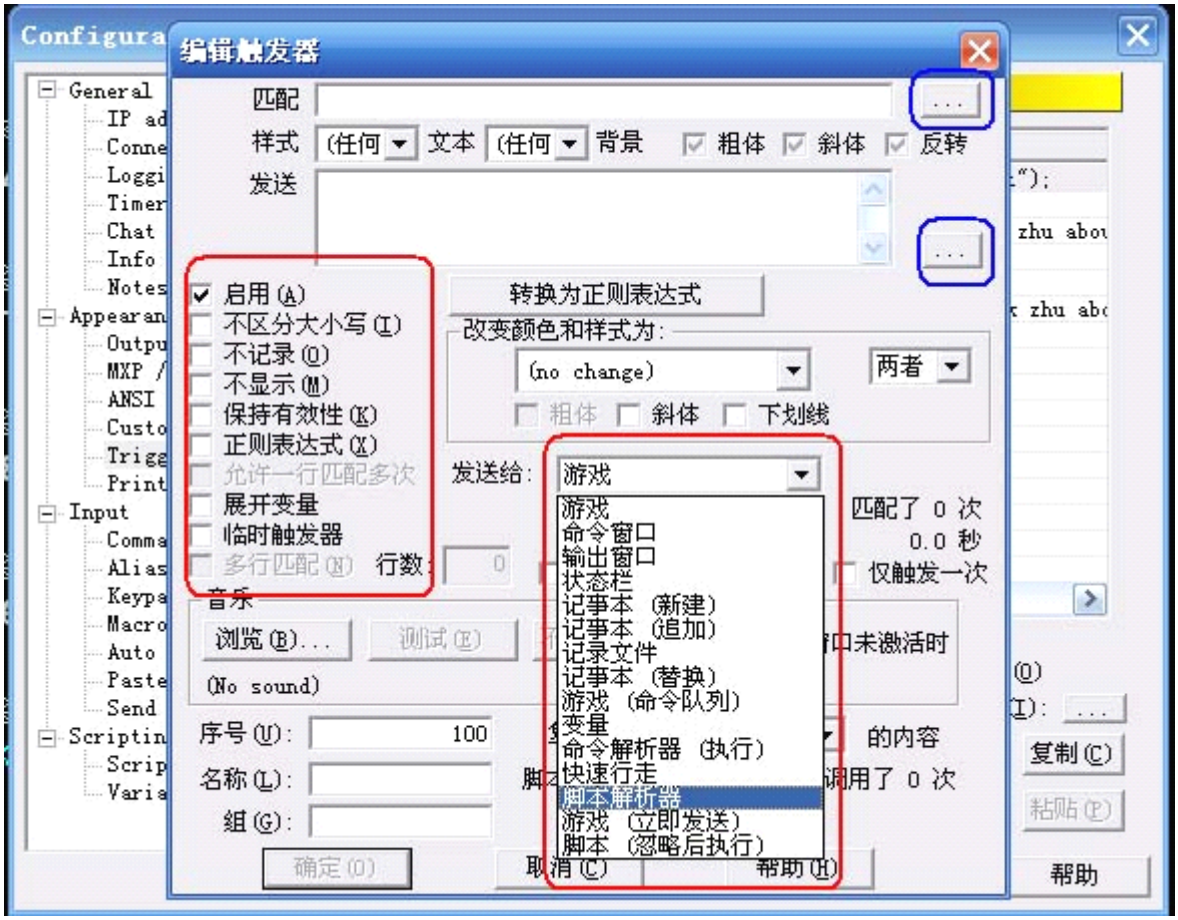
其实，mush 的触发并不难学，包括 mush 的脚本也并不难学。讲到底，电脑只是一种非常愚蠢且认死理的东西，你只要把他的脾气摸清楚了，他就会乖乖的给你干活。下面，我们就来学习触发吧。

（１）mush 中触发的建立

点击菜单－－游戏－－配置－－触发器，可以进入触发设置画面（怎么点的，就不抓图了，如果看到这里连这个都不会，那就应该反省了），具体如下图：



点击红色标志部分可以添加一条新的触发，如下图。



蓝色标志部分，是点击可以展开输入更多内容的。

红色标志部分，大家有没有发现，这次红色的框特别大。没错，这次要把几乎所有的项目都讲一下，所以全画出来了。

（２）上图中各项的说明

（a）匹配

触发这个触发器的文字，可以使用正则表达式。

（b）样式

这里的设置可以让触发器只匹配特定颜色或者字体样式的文字。字体样式检查框有三种状态：

不选择（例如,被匹配行不是粗体）

选择（例如,被匹配行是必须是粗体）

灰色（例如,被匹配行是不是粗体都可以）

（c）发送

触发器发送的命令。你可以使用 %1 引用第一个通配符的内容，%2 引用第二个通配符的内容,最多可以引用到第九个。%0 则引用整个被匹配的文字。%C 可以引用剪贴板中的内容。要在发送命令中使用“%”，你必须用“%%”代替。

注意：如果发送的内容是脚本命令，你应该根据不同的脚本语言添加相应的引号（脚本文件中则不用），如果该加引号的不加则会出现语法错误。具体的例子，我们后面会讲到，这里只要知道一下就可以了。

（d）启用

允许使用这个触发器。

（e）不区分大小写

匹配触发器的时候不区分大小写。

（f）不记录到记录文件中

不在记录文件中记录被匹配的行。

（g）不显示匹配行

不在输出窗口显示被匹配的行。

（h）保持有效性

默认情况下，当一行文字和一个触发器匹配后，这行文字不会再尝试匹配其它的触发器，也就是说一行文字最多只能匹配一个触发器。如果选择了这项，这行文字就会在所有触发器中尝试匹配。这样就可以让一行文字同时匹配多个触发器。

（i）正则表达式

选择这项后，触发器将会被作为一个正则表达式处理。

（j）允许一行匹配多次

如果选择了这项，同一行文字可以多次匹配这个触发器。例如，假如一行文字中有多个你想匹配的单词，如果不选这项，只有第一个匹配的单词会触发这个触发器。反之，则每个单词都会匹配这个触发器。

（k）展开变量

发送命令时展开命令中的变量。命令中的变量以“@”开始，后面跟着变量的名称。选中这项后，如果要在发送命令中使用“@”字符，你必须用“@@”代替。

（l）临时触发器

如果选择了这项，这个触发器不会保存到配置文件中。也就是说它仅在当前游戏会话中有效，游戏关闭时这个触发器会被自动删除 。

（m）多行匹配

触发器默认只能匹配一行，选择这项后就可以匹配多行，同时你还必须设置这个触发器匹配的行数。这里的行是逻辑行，是服务器传送过来的行，而不是你在输出窗口看到的行。也就是说如果一行的文字太长，在输出窗口呈两行或者更多行显示，它仍然算一行。

使用多行匹配必须选择“正则表达式”选项。

多行匹配的局限性：

＊触发器匹配的行数越多，性能就越低。

＊要使用多行触发器，必须先启用“正则表达式”，因为普通方式无法匹配一个换行符。

＊因为实现方式的不同，使用了多行触发器就不能启用其它一些功能(例如，改变颜色，不在输出窗口中显示匹配行，不纪录到记录文件中）。

＊即使前一个触发器（普通触发器，设置了“保持有效性”）选择了不显示匹配行，这一行信息仍然会被后面的多行触发器使用。

（o）转换为正则表达式

点击这个按钮可以把普通触发器转换为正则表达式的形式。当你需要把普通触发器修改为一个正则表达式的时候，使用这个按钮可以帮你节约一点时间。

（p）改变颜色和样式为：

对于被匹配的行，你可以通过这里改变它的颜色和字体样式。

一个普通的触发器（没有使用正则表达式）总是会改变整个行的颜色或字体样式。

（q）通配符内容转换为小写

在使用通配符前把通配符中的英文大写字母转化为小写字母。

（r）发送给

选择把触发器发送的内容发送到哪里去。默认是直接发送给服务器。但是你也可以选择其它地方，例如输出窗口，记事本窗口，或者选择“脚本解析器”，把发送的内容作为脚本来执行。同样，如果你选择了发送给“命令解析器 (执行)”，发送内容就会交给 MUSHclient 的命令解析器来执行，就像你直接在命令输出窗口中输入它们一样。换句话说，此时你可以在发送内容中包含脚本，别名，快速行走等功能。

你可以选择发送到的地方有以下几种：

＊游戏 （MUD 服务器）

＊命令窗口 （显示在命令窗口中）

＊输出窗口 （显示在输出窗口中，就像使用 Note 函数显示注释一样)

＊状态栏 （显示在状态栏上）

＊记事本 - 新建 （创建一个新的记事本窗口并把发送内容添加进去）

＊记事本 - 追加 （把发送内容添加到现有记事本窗口的最后面）

＊记录文件 （把发送内容直接添加到记录文件中）

＊记事本 - 替换 （把现有记事本中窗口的内容用发送内容替换掉）

＊游戏 - 命令队列 （把发送内容添加到命令队列中，队列的时间间隔可以在“命令”配置对话框中设置）

＊变量 （把发送内容赋值给一个变量，必须在变量框中输入变量的名

称）

＊命令解析器 - 执行 （通过命令解析器执行，这是发送内容中可以包含别名、快速行走等等）

＊快速行走 （把发送内容作为快速行走的路径来执行）

＊脚本解析器 （把发送内容作为脚本命令来执行）

＊游戏 - 立即发送 （把发送内容立即发送给服务器，不管当前队列中是否有命令）

（s）音乐

当触发器被触发时播放预先设置好的音乐。

（t）仅在窗口未激活时

设置只有游戏窗口不是当前活动窗口时才播放音乐。

（u）复制通配符 n 中的内容

把选择的通配符内容自动复制到剪贴板上。

（v）序号

在匹配触发器的时候，MUSHclient 会根据序号的大小依次做匹配比较，小的序号先匹配，大的序号后匹配。因此，改变触发器序号的大小可以改变它的优先级。

（w）名称

可以给每个触发器一个名称，方便在脚本中使用。注意名称必须是唯一的，不能有重复。

（x）脚本

如果使用了脚本文件（在“脚本”配置对话框中设置），你可以在这里填写这个文件中的一个函数名称，触发器被匹配时会自动调用这个函数。这个必须函数必须包含以下 3 个参数：name，line，wildcards。具体例子见后面。

（y）组

具有相同组名称的触发器、别名和定时器会自动归为一组。使用组的好处是可以使用相应的函数启用或者禁用组中的所有项。

（z）变量

如果你设置了把发送内容赋值给一个变量，这里就应该填写这个变量的名称。

好了，到这里，触发器本身的相关内容都讲完了，如果大家不明白的，也只有再仔细的多看几遍，下面就是和正则相结合的实例部分了。

－－－－－－－－－－－－－－－－ 我是痛苦的分割线 －－－－－－－－－－－－－－－－－－－－－－－－

我深深的了解到，逃避——是没有用的；正则——是跑不掉的。就象 ddid 所说的一样，讲触发，怎么可以不讲正则呢。这正则，是那么好讲的吗？唉，苍天呀，大地呀，掉个漂亮ＭＭ下来，安慰我受伤的心吧……

呵，转换一下心情。

经过仔细考虑，我决定，选择一个 mud 里比较典型的东西，结合正则与触发一起来讲。当然，就这么一些篇幅里，想要教会大家使用正则几乎是不可能的。

但，我可以把正则的思路告诉给大家。如果让正则来根据你的想法来工作，再加上实例和比较详细的说明，我会尽量把说明述说的白一些（也就是说，尽量少用计算机词汇，用人的词汇——怎么这么怪怪的），如果说，在这里大家能对正则有了一些了解，甚至有了一些兴趣，那我会非常荣幸。

我的例子是——当我们敲入 hp 的时候，会有显示我们的自身状态。我想达到这样的效果，输入 hp，然后就自动把这几样值报出来：当前气血，最大气血，受伤程度，当前内力，最大内力。当然，还有别的内容也可以一一报出来，但作为一个实例，没有必要写的太全面，如果各位有兴趣，可以自己试着写一下。

还有两个说明，请大家在学习这一段的时候，一定要记得：

１、如果你以前会 Zmud 里的抓取变量，请你先暂时全部忘掉，差别是很大的。

２、如果我给出的正则代码跟你所知道的不一样，这很正常，１００个人本身就可以有１００种正则写法解决同样的问题。

如果都没问题了，那我们开始吧。

首先，我们把所谓正则的正统定义扔一边去不管，静下心来想一下，如果有一天，我们告诉电脑——你到大街上去，到东边的杂货店，帮我买一个干粮回来。

电脑会怎么样？１００％电脑会傻掉，或者，干不了这活。因为，你给的条件太模糊，不明确，而电脑只会（或者说，目前只会）执行明确的命令。

好了，我们换一个说法——你到大街上去，往东走，５０米后，看见一个电线杆，然后向右拐，有一个杂货店，进去，执行 buy jiudai 命令。

如果这样说，电脑基本上就可以完成你命令了。

好了，我们来分析一下，为什么第二次电脑就可以完成任务。

第二次，电脑有了明确的参照物（呃，这么说，能明白吗？）或者说，你给了电脑一个到了哪里就停，这样一个标志性的东西的存在。

电脑就开始这样操作——进了大街，什么都不管，就直直走，直到看见电线杆，在看见电线杆之前所有的，都被忽略掉了。（实际操作中，你可以选择忽略或者不忽略）

呃……扯了不少了，不知道大家对正则有没有一点点一丝丝的概念了。简单来说，正则很聪明，他会把你所想要忽略的忽略掉，想要得到的保存好。但，前提是，你得说到让他能听懂才可以。

结合我学习和使用正则的经验来看，一下子要学会很复杂的正则是不可能的，但一点一点从比较简单的做起，那反而比较容易。但一般的正则的教程往往

没考虑到这些。所以，我在这里使用了一个新的概念——电线杆——正则不是复杂吗？没关系，我们一点一点来啃，先挑一段容易的竖根杆子。杆子外的，我们不管，等杆子里的弄明白了再说。

好了，下面我们可以正式开始了。

首先，我们看一个北侠中 hp 一下后气血、内功的显示。

【 气血 】 503 / 573 [97%] 【 内力 】 855 / 835 (+ 0)

我们的总的目标，是把５０３、５７３、９７、８５５、８３５这五个值都抓下来。根据我上面讲的理论，我们先考虑怎么来抓第一个５０３，也就是当前气血值。

所谓杆子，在正则的世界里，我们应该理解为标志性的字符，换句话说，这个字符（或者这一组字符）在你所需要分析和判断的字符中，是唯一的。这种唯一性，是最容易被电脑认可的（因为电脑比较笨）。

现在，很明显，上面的【 气血 】 和【 内力 】 具有非常好的唯一性。我们现在先用【 气血 】，把第一根电线杆，就竖在了【 气血 】的上面。

我们现在要做的是——告诉电脑，在【 气血 】这个东西的后面，有一组数字，是我所需要的。

那，这样写行吗？很坦白的告诉大家——不行。（表砸偶）那，为什么呢？大话西游里，紫霞仙子有说过这样一句，我猜中了开头，却没有猜中结局。电脑很笨的，你就告诉他，到哪里开始找什么，却不告诉他，哪里是结束，他就会乱来的。

那，正确的说法是什么呢？——告诉电脑，在【 气血 】这个东西的后面（这是开始了），是一些空格（这个等下讲），在那些空格的后面，是我所需要的数字，在数字的后面，又是一些空格（这是结束，在这里又立一根结束的电线杆）再后面，我就不管了，你爱干什么就干什么吧。

先说一下，上面突然冒出来的空格，是什么回事。对于我们来讲【 气血 】和５０３之间是没有什么存在的，但对于电脑来讲，这两者之间是存在有一些空格的，如果不把这个空格告诉电脑，他是又会犯傻的。

好了，我们开始把刚才的话转成电脑语言。

^[>]*【 气血 】\s+(\d+)\s+.*$

把上面这串东西输入到你的触发器的匹配，然后在发送中输入 say %1，试一下，是不是可以得到你的当前气血值。

唉……万事开头难呀。现在我们来一点一点解释上面那串鬼画符似的东西，是怎么回事。先说一下，我不会列出一张总表，让大家对照着看。我们基于碰到什么，就讲什么的原则，当然，第一次讲到了，第二次再遇到，就不会详细讲了。为了让大家能够看的清楚，我用红蓝双色把上面的代码分别标出了，代码是连在一起用，但作用是分别累加的。呃，可以这么理解，上面我说的每一句话，就是一小段代码的具体功能，加在一起，就达到了那个效果。

告诉电脑——^[>]*——任何一样东西，总是有个开始，^代表的意思是从一行的最开头开始起，嗯，这么做的目的，是防止你做机器人的时候，别人在你前面 say 一些东西，把你的机器人破坏掉；虽然北侠的 hp 后的状态，在【 气血 】

前是没有什么东西的，但有时候因为网络的原因，会有一个或者几个>出现，[>]*代表的意思是有 0 个或者几个>符号，*的意思是前面的那个东西出现 0 至无穷多次。是不是觉得有点乱，没关系，硬记住就可以，后面我们会继续提到。

在【 气血 】这个东西的后面——【 气血 】——这个应该很好理解，就不解释了。

是一些空格——\s+——在正则中，\s（注意，是小写的 s）是表示空格（这个只能硬记，或者查表），但我们不知道确切有几个空格，所以，我们需要有一个符号来表示，前面字符的数量不确定，在这里，我们选择了+。也许，有人要问，为什么这里是＋，上面是＊。在这里简单说明一下，＊代表的意思是——我不确定这东西是否会出现，也不确定这东西会出现几次，所以＊代表的是 0 ～N 次的数量（就是说，可以没有，可以有许多，mud 中的>的确也是这种情况没错吧。）＋代表的意思是——我确定这东西肯定出现至少一次，但不确定会出现几次。现在明白＊、＋的区别了吧？

在那些空格的后面，是我所需要的数字——(\d+)——在正则中，\d（注意，是小写的 d）代表数字，＋号我们刚才已经说了，是指这里可以出现至少一个到多个数字，()的作用和 Zmud 里是一样的，表示要取这部分的值。

在数字的后面，又是一些空格——\s+——这跟上面是一样的。

再后面，我就不管了，你爱干什么就干什么吧——.*$——.*的意思，是代表所有一切字符，$的意思是一行的结束。这两者加在一起的意思就是，对后面的东西不做要求了，随便吧。

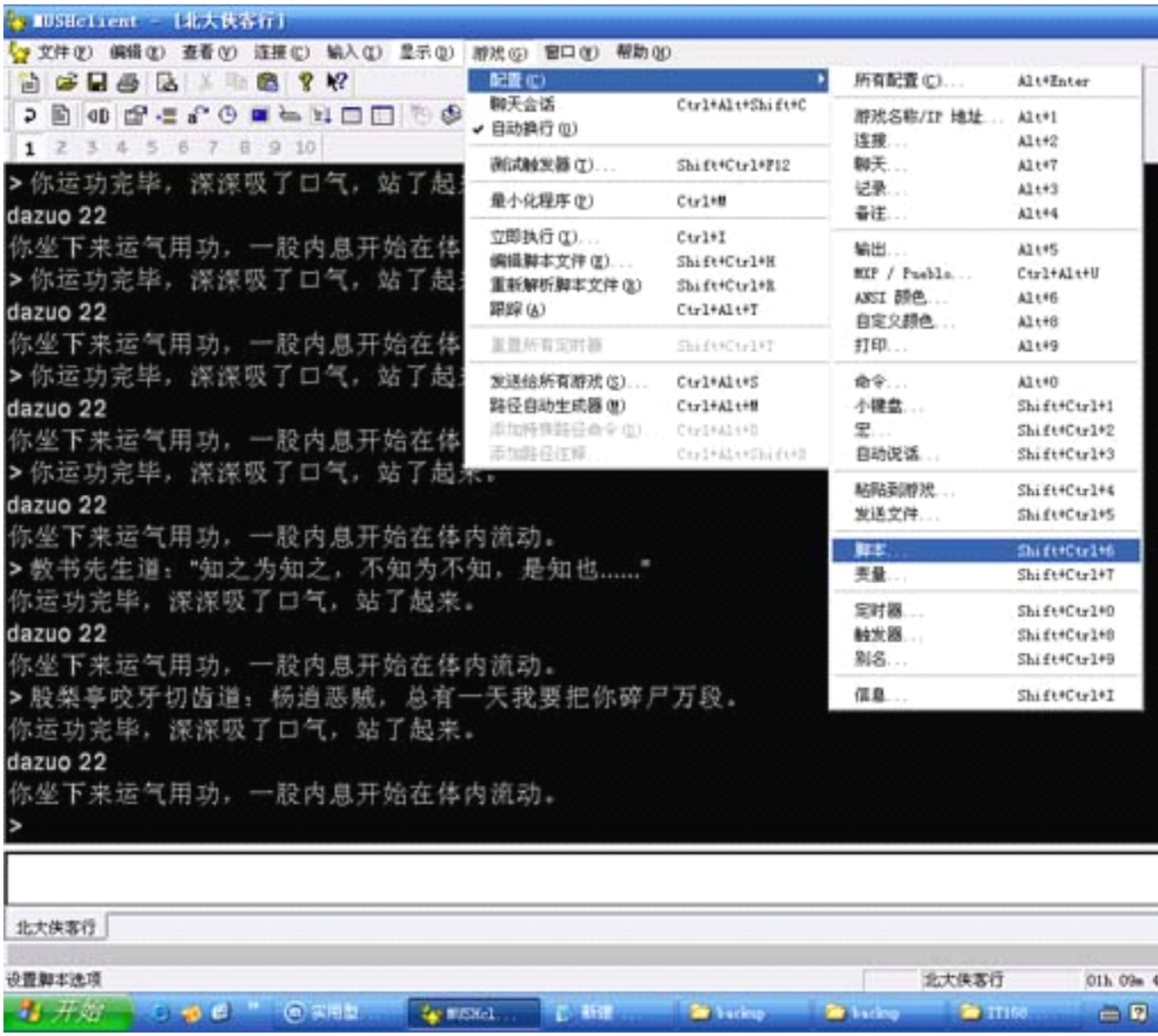
我想，经过上面这么详细的一点一点的分析下，大家如果都能看明白的话，至少说，对正则是应该不会太怕了。我写这么一大篇的目的，也只是希望让没有接触或者不敢接触正则的朋友，至少知道正则的脾气是怎么样的。至于具体的正则的学习，不是我这篇小小的文章能够教的了，大家在北侠搜“正则”，已经有人推荐了几篇文章，我有看过，也挺不错的。至少看起来比较专业。

下面，给出我所写的抓这 5 个值的答案，大家试着用我上面讲的东西分析一下。

【 气血 】 503 / 573 [97%] 【 内力 】 855 / 835 (+ 0)
在匹配里输入^[>]*\s*【 气血 】\s+(\d+)\s+/\s+(\d+)\s+\[\s*(\d+)\%]\s+【 内力 】\s+(\d+)\s+./\s+(\d+)\s+.*$
在发送里输入 say %1 %2 %3 %4 %5

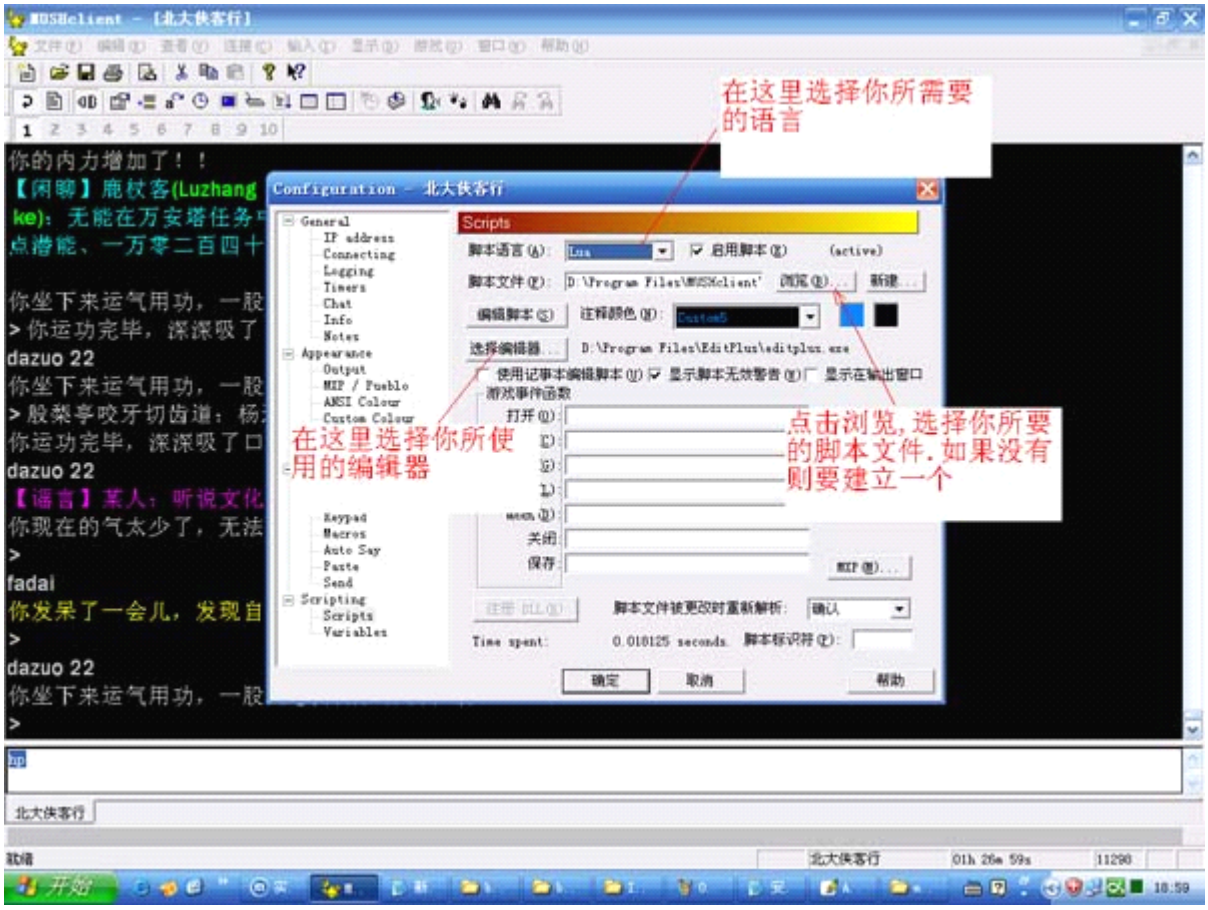
好了，触发就到这里了。

7.抓取变量送到脚本
首先,我们要设定我们所使用的脚本语言,并且建立一个脚本文件



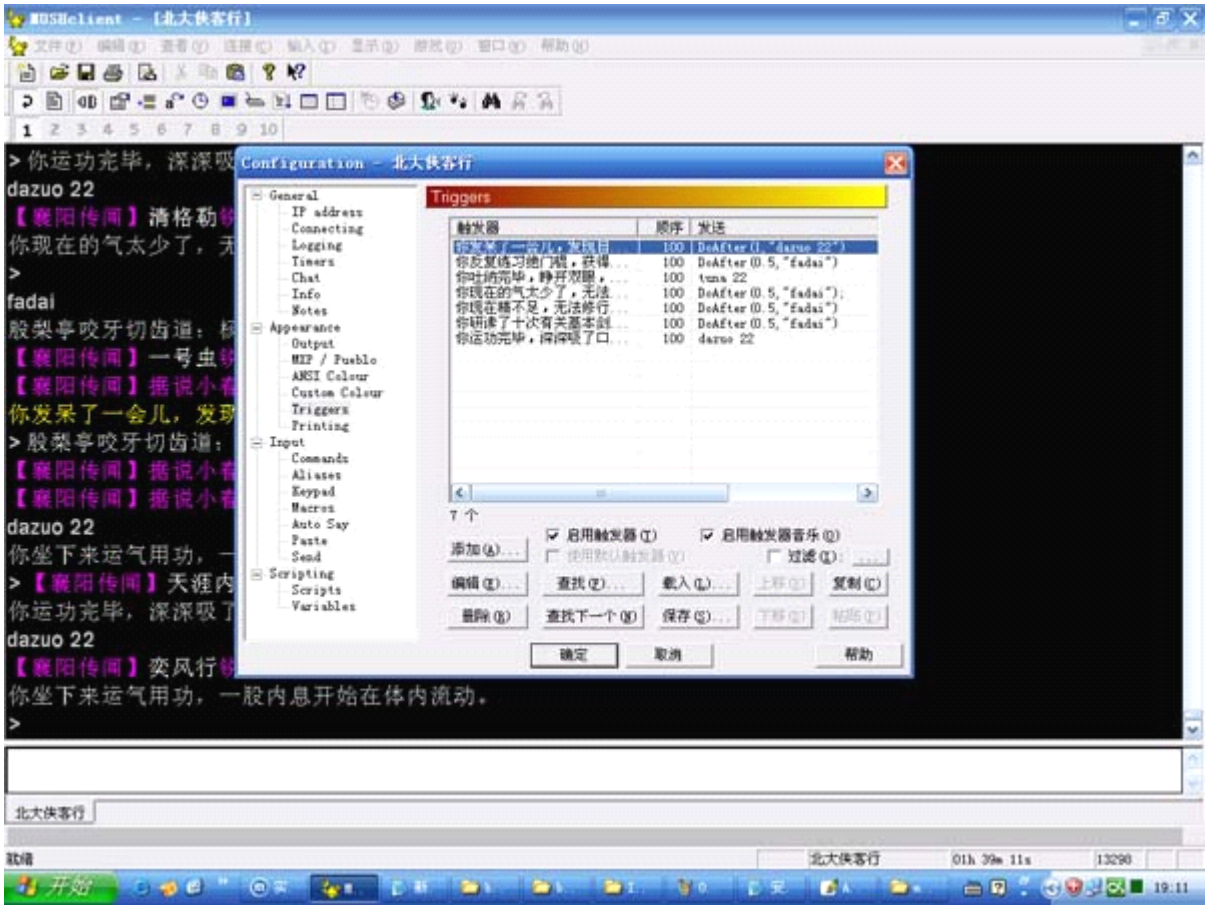
点击游戏-->配置-->脚本,我们可以看见下面的画面

在下面的画面里,我们进行脚本语言的选择和脚本文件的选择

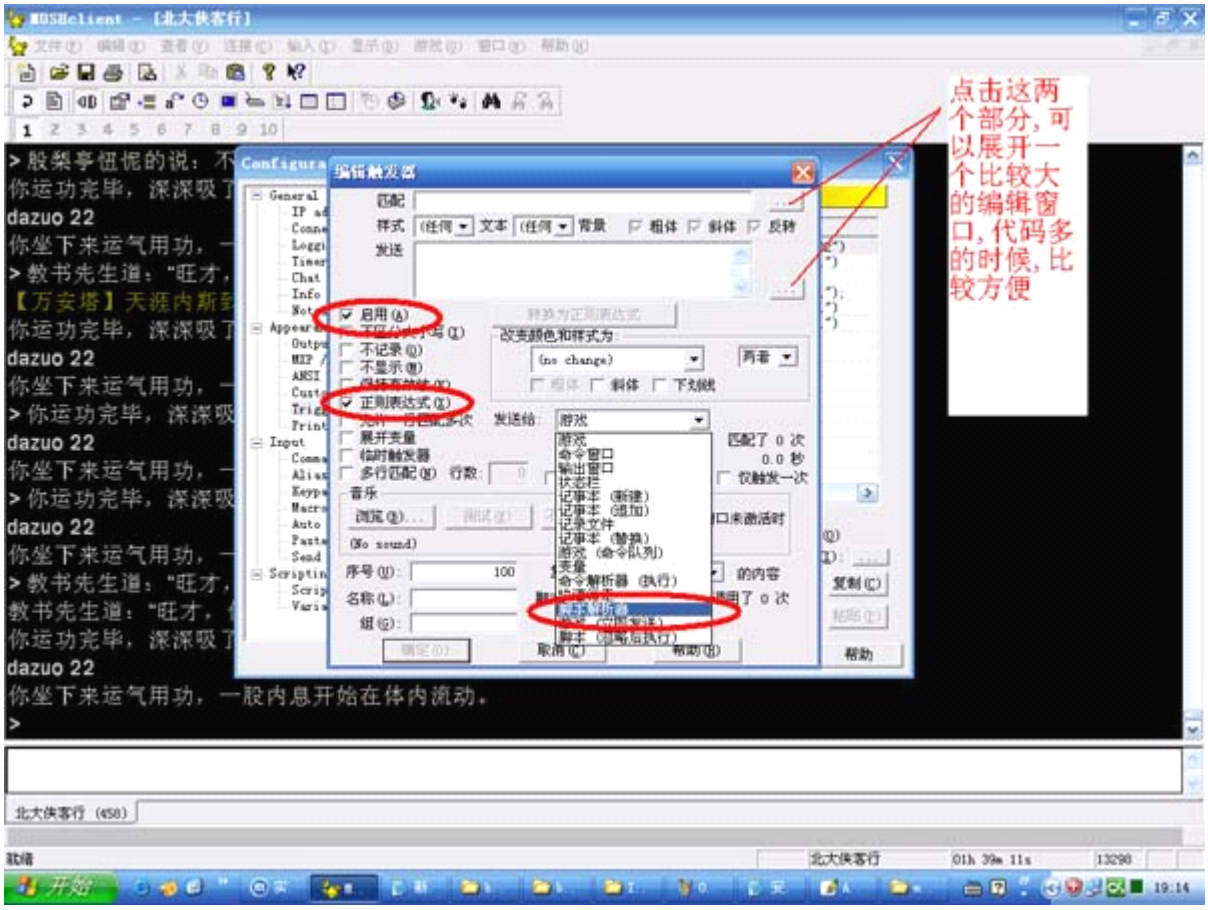


选择好脚本语言后，我们就可以开始想办法把抓取的内容送入脚本了

首先，我们点击游戏-->配置-->触发器，（这张图看上面就可以了）
打开触发器的设置画面（这图很简单，没什么好说的）



这张图请注意我加红色圈的部分,一定要勾上,不然的话,调试不可能成功,"发送给",一定要选"脚本解析器"
这张图的"匹配"是放正则,"发送",则是把内容输入到脚本



写一个最简单的做为例子(注:这例子是随手写的,没有经过验证)

"匹配"部分:你买下了一个(.+)
"发送"部分:mybuy("%1")

在脚本里,就有这么一个函数

```
function mybuy( sth )
    if        sth == "酒袋" then he jiudai
    elseif  sth == "鸡腿" then eat jitui
    elseif  sth == "干粮" then eat liang
    else    sth == "漂亮 MM" then kiss mm  --这个是我乱说的.呵,调节一下气氛
    end
end
```

因为这里好象没人懂 PHP,借用了 Lua 的函数形式,其实也都差不多

－－－－－－－－－－－－　我是潇洒的分割线　－－－－－－－－－－－－－－－

－－－－－－

经过再三考虑，还是决定再次丰富这部分内容。其实，任何一种语言都不可怕，可怕的是，你不敢以平常心去面对他。

就我个人学习 php 的心得来讲，一段正确，简洁，有针对性的小程序，比云山雾海说一大堆都来的有效。

所以，我想针对 mush 所支持的语言，尽可能做到每种语言都提供一小段标准的代码，以方便大家学习。

首先要说明的是，所有的代码都能完成同样的效果，只是所用的语言，你没有必要看完所有代码，挑你所想学的那种看就可以了。

其次说一下代码的思路。我们都知道 mud 再复杂的情况（不考虑一些ＢＴ弄的特别东西），要抓取的无非是两样，一个是字符，一个数字，其中数字还有可能涉及到计算。任何复杂的问题，也不过就是这三样的分解组合罢了。

第三个说一下代码的效果——在当前说一句话，可以随意说，不过，要求这句当中有一个数字（比如说：大家好好 11 所有人），这样就可以了。然后，代码会以 Note 方式输出三行，第一行是数字内容，第二行是数字后字符的内容，第三行为数字＋1 的内容。

如，刚才说的——大家好好 11 所有人。

则输出：

第一行：数值抓取结果 = 11
第二行：字符抓取结果 = 所有人
第三行：数值计算结果 = 12

基于实用性原则，本教程仅提供 Lua、Javascirpt、Php 三种代码范本。

Lua 代码

匹配部分：^\D+(\d+)(.*)$
发送部分：getSay(%1,"%2")
代码部分：

```
function getSay( getInt, getStr)
        Note( "数值抓取结果 = " .. getInt );
        getInt = getInt + 1;
        Note( "字符抓取结果 = " .. getStr );
        Note( "数值计算结果 = " .. getInt );
end
```

注意事项：

１、在发送部分的％２要加""，％１（内为数值）不要加，％２（内为字符）一定要加。

２、虽然 Lua 程序代码可以不使用;作为结束符，但个人建议还是加上。因为 Lua 是可以一行内放多个语句的，在这种情况下，是一定要加上;的。

Javascirpt 代码

代码部分：

```
function getsay(get_int, get_str)
{
        world.Note ("数值抓取结果 = " + get_int);
        var getint = parseInt(get_int) + 1;
        world.Note( "字符抓取结果 = " + get_str );
        world.Note( "数值计算结果 = " + getint.toString() );
}
```

感谢 ddid 提供代码

php 代码

匹配部分：^\D+(\d+)(.*)$

发送部分：getSay(%1, "%2");

代码部分：

```
function getSay( $getInt, $getStr )
{
        $world->Note( "数值抓取结果 = " . "$getInt" );
        $getInt = $getInt + 1;
        $world->Note( "字符抓取结果 = " . "$getStr" );
        $world->Note( "数值计算结果 = " . "$getInt" );
}
```

注意事项：

1、在发送部分的%2要加""，%1（内为数值）不要加，%2（内为字符）一定要加。

2、在发送部分 getSay(%1, "%2");最后是有一个；的，千万不要忘。

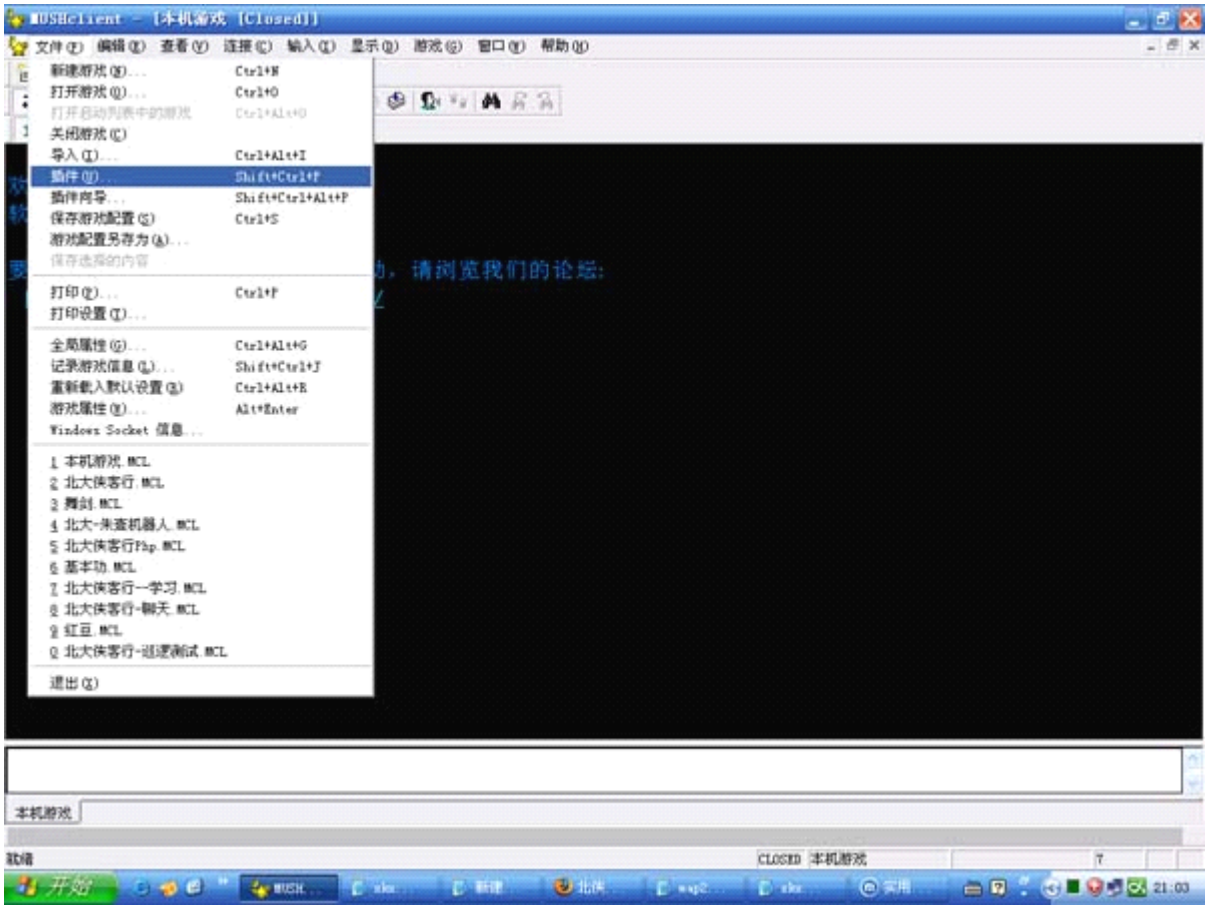
8.插件

所谓插件，大致可以理解为暗黑 2 中套装，别人把一些功能都配好了，封装好了，你只要拿过来直接用就可以了。目前在 mush 中，最有用，最受欢迎的，应该是路径插件了。

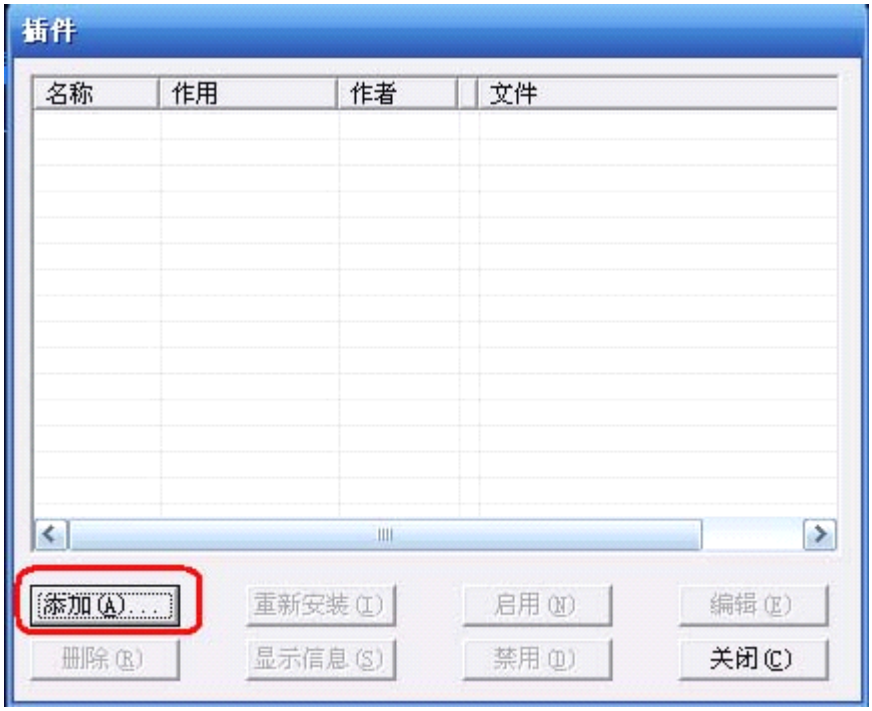
我在上面“别名的使用”这部分内容中，提到过北侠的路径插件，我们就拿这个来学习吧。

首先，下载。地址在这里 http://pkuxkx.net/forum/thread-8059-1-2.html

然后，我们点击菜单——文件——插件，如下图：


MUSHclient - [本机游戏 [Closed]]
文件(F) 编辑(E) 查看(V) 连接(C) 输入(I) 显示(D) 游戏(G) 窗口(W) 帮助(H)
新建游戏(N)... Ctrl+N
打开游戏(O)... Ctrl+O
关闭游戏(C)
导入(I)... Ctrl+Alt+I
插件(P)... Shift+Ctrl+P
插件向导... Shift+Ctrl+Alt+P
保存游戏配置(S) Ctrl+S
游戏配置另存为(A)...
打印(P)... Ctrl+P
打印设置(I)...
全局属性(G)... Ctrl+Alt+G
记录游戏信息(L)... Shift+Ctrl+J
重新载入默认设置(R) Ctrl+Alt+R
游戏属性(E)... Alt+Enter
Windows Socket 信息...
1 本机游戏.MCL
2 北大侠客行.MCL
3 舞剑.MCL
4 北大-朱查机器人.MCL
5 北大侠客行Php.MCL
6 基本功.MCL
7 北大侠客行-学习.MCL
8 北大侠客行-聊天.MCL
9 红豆.MCL
0 北大侠客行-逻辑测试.MCL
退出(X)
，请浏览我们的论坛:
本机游戏
就绪
CLOSED 本机游戏
开始

点击以后，显示如下图：


插件
名称
作用
作者
文件
添加(A)...
重新安装(I)
启用(N)
编辑(E)
删除(R)
显示信息(S)
禁用(D)
关闭(C)

点击红色标记，再选择到下载的文件，就可以使用该插件了。（以这个路径为例，你直接输入 suzhou 就跑到苏州去了，你是看不见插件的 suzhou 在哪里的。除非打开原文件看。）

其次，说一些注意点。细心一点的同学，可能已经发现了，这个插件载入后，找不到可以修改的地方。这就是插件的特点，或者说是局限性了。使用很方便，修改很麻烦。至于用不用，怎么用，就是大家自己考虑了。

好了，看到这里，实用型 mush 教程，已经完全结束了，如果上面所有内容都掌握的话，你就已经不是一个 mush 初哥了。那么，最后祝大家挖泥愉快。