

1.12 宏命令基础教学

前言

因被老 G 的公益行为所感动，并致 12 年前的青春，在官服决定开放怀旧服之际，决定朝花夕拾写下本《宏基础教学》来支持公益与情怀。同时感谢所有在本文档编著中帮助过的作者的朋友以及所有参考文档的作者们！

“枪是无罪的，有罪的是扣动扳机的人”。写此教程目的是因为 60 年代繁琐的技能列表，为了完善游戏体验，帮助 PvE 玩家解放双手降低错误而写。我不期望我的宏被用在大号虐杀小号、群号围殴单号上。当然宏并不是万能的，码农们都知道代码不可能没有 bug。我们要做的是尽可能在完善的条件下达到最顺畅的技能输出。一般情况下，宏将 3 个以内的技能捆绑一起并通过相应的判断条件有限释放。会有人可否一键全技能，答案虽然是肯定的，但需要臃肿的代码量来支持，复杂的条件判断及循环增加了巨大的开发和维护成本，不一定能带来流畅的体验。所以请根据实际情况在最性价比的情况下编写实用的宏命令。

因本教学手册编写仓促，还有很多的 API 函数未列式解释。其余相关扩展待日后更新，如有需要还请自行验证。同时希望大家对不足之处提出宝贵建议指正，也可将未解释部分提供解析内容，非常感谢！

变更记录

版本号	更新日期	描述
1.0	20171220	初稿定稿。确定手册基础架构与内容。
1.01	20180105	增加装备栏、背包、光标函数定义。删除 CleverMacro 插件。

目录

- 插件准备
- Lua 语句
 - Lua 简介
 - 语法测试环境配置
 - 编译软件部署
 - Lua 编译器与编译软件关联
- Lua 基础语法
 - 数据类型
 - 变量
 - 循环
 - 流控制
 - 运算符
 - 函数
 - 动作条定义
 - 姿态定义
 - 动作条
 - 增益 Buff/减益 Debuff
 - 宠物
 - 法术技能
 - 单位
 - 针对目标
- 实例分析
- 备注与附件

一、 插件准备

“工欲善其事，必先利其器”。众所周知，早期的 wow 版本中玻璃渣开放了大量的 API，为玩家带来了各种便利。但又因为某些原因这些 api 并不能非常好的应用到一起。所以推荐几款非常不错的宏插件，它们可以让这些 API 结合的无比强大。

为避免涉及文档著作权等问题，这里只简略表述其功能，并附上附件。详情请自行搜索。

- 1) supermacro 超级宏——支持 2000 字节的宏语句。官方开放的宏长度只有 250 字节，足足翻了 8 倍。

附件：

- 2) Buff&Debuff 函数——将原有的 buff/debuff 判断语句精简到只需要一个函数上。大大节省宏长度

附件：

- 3) CleverMacro——支持 WOW2.0 的 api 使用在 1.12 版本上。2.0 开放的 API 请自行搜索。

附件：**目前因无限刷屏，暂时禁用。**

图样，红框表示“Buff&Debuff 函数”应用：



二、 Lua 语句

2.1 Lua 简介

Lua 是一种轻量小巧的脚本语言，用标准 C 语言编写并以源代码形式开放，其设计目的是为了嵌入应用程序中，从而为应用程序提供灵活的扩展和定制功能。

顺带提下《魔兽世界编程宝典--World of Warcraft Addons 完全参考手册》，这本书详细编写了 WOW 中所包含的宏命令编写及插件制作，有兴趣的同学可以自行学习。附件中提供译本下载。

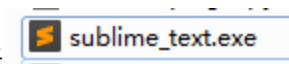
本教程针对初学者，综合上述内容并精简。提供基础 Lua 语法及技能类 API 知识。进阶学习者可参照附件文档或自行搜索深入内容。

2.2 语法测试环境配置

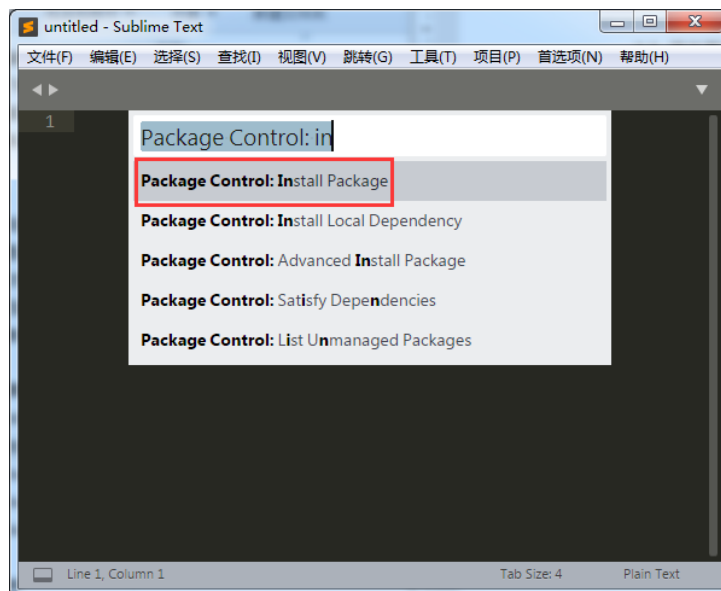
2.2.1 编译软件部署

感谢“BL-ZS-怪兽”同学提供建议使用 Sublime_Text3 来编译 Lua 语法验证语句是否存在错误。

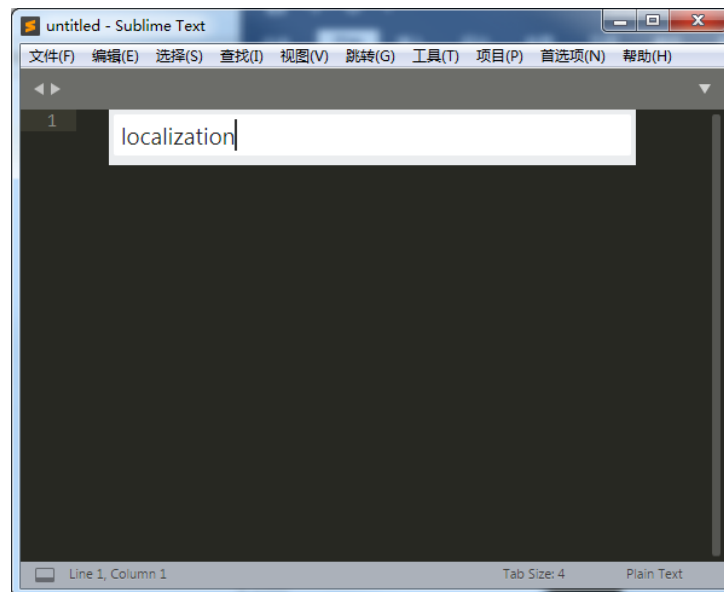
下载地址：



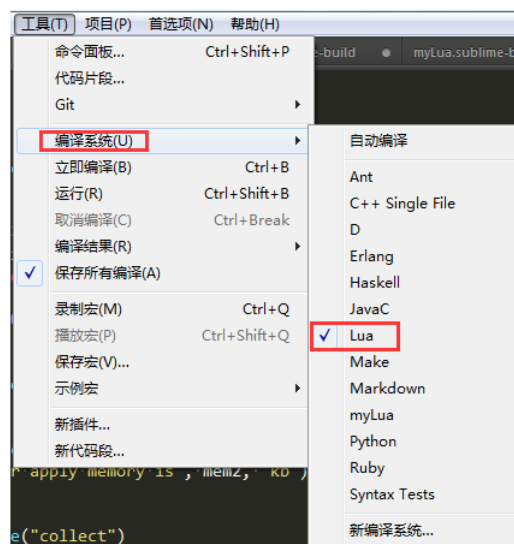
解压即可，双击打开，弹出界面如下（作者已经安装相关插件，原始是英文界面。以下操作均以中文称呼，请自行翻译）：



点击工具栏“工具”，选择“命令面板”，选择上图红框，在新窗口中输入“localization”，会弹出中文插件，选择安装即可（作者已安装故不再显示）。安装完毕后自动更新至中文界面。



重复“install package”操作，输入 LuaFormat，并自行安装。安装完毕后可以再“工具”-“编译系统”中查看是否有 Lua 编译确认安装成功。

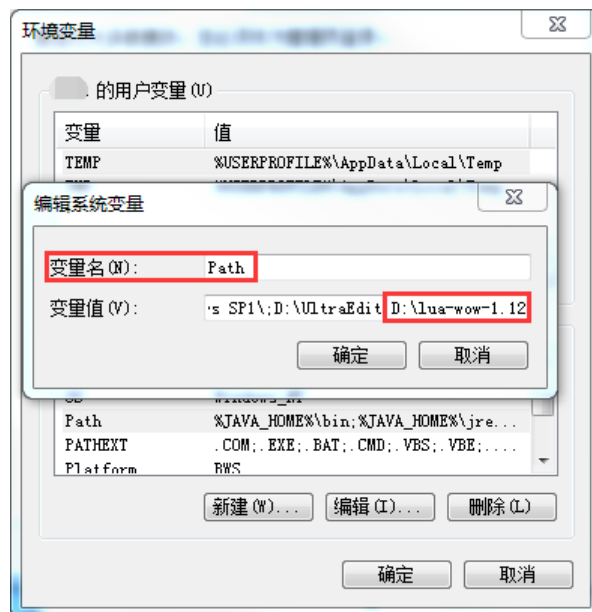


2.2.2 Lua 编译器与编译软件关联

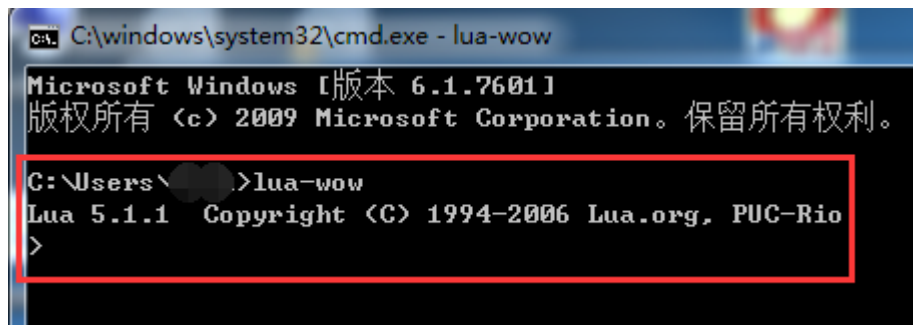
Lua 编译器，类似 Java 的 JDK，用以系统识别其语言。这类往上有很多教材，不作累述。

附件提供下载：

需配置环境变量。如下图在 Path 变量下增加编译器路径，确认退出即可。



通过 cmd 命令提示符窗口，验证 Lua 编译器部署成功。

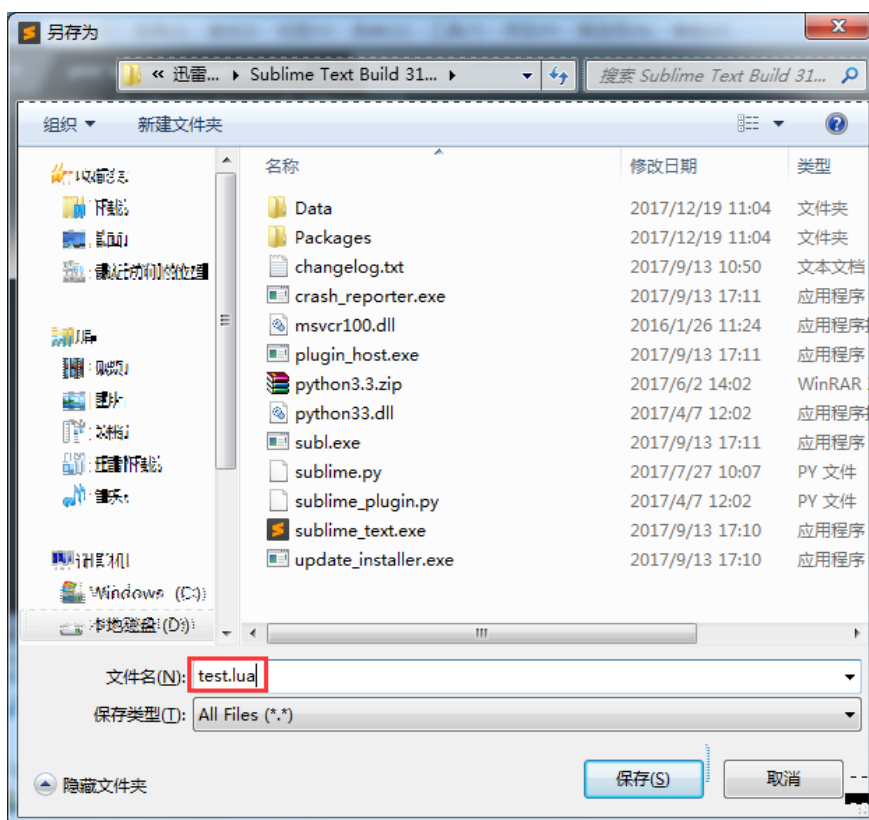


打开 sublime_text3 编译工具，菜单中的“工具->编译系统->新编译系统”来新建一个编译系统配置文件，在配置文件中使使用下面的配置代码覆盖：

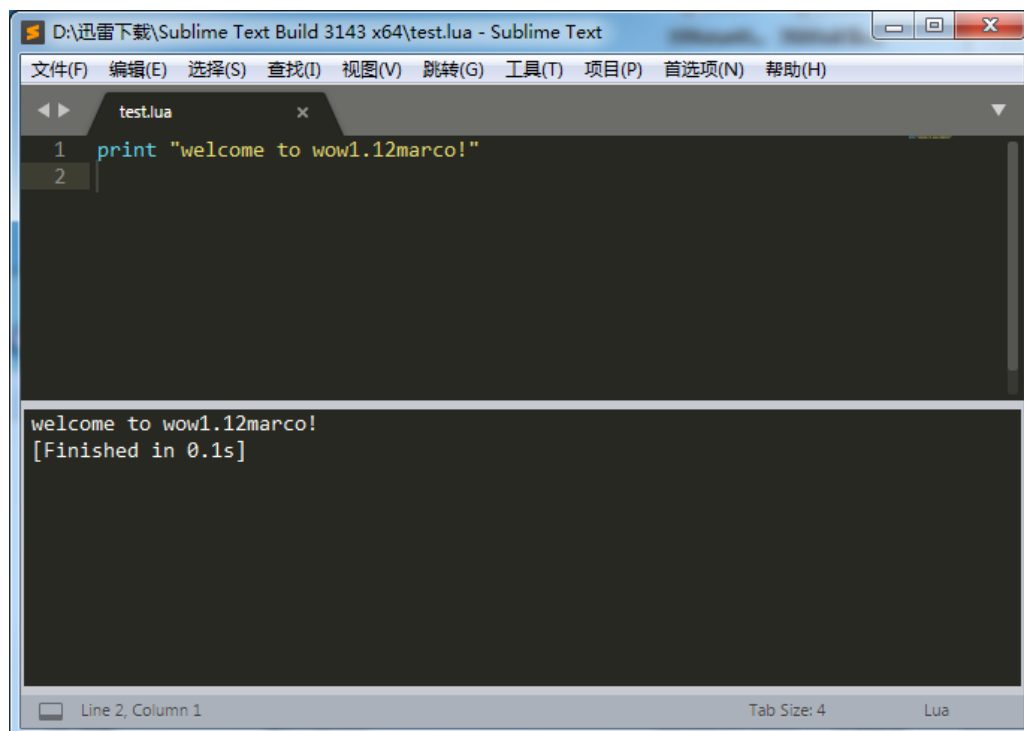
```
{
"cmd": ["$lua 编译器路径/$lua 文件名.exe", "$file"],
"file_regex": "^(?:lua:)?[\\t ](...*?):([0-9]*):?([0-9]*)",
"selector": "source.lua"
}
```

注意红色部分需要根据实际环境修改调整（即环境变量配置路径，并增加编译器文件名），斜杠符号为“/”，非 window 默认斜杠。保存退出。

新建文件，输入 Lua 语句，保存时注意以“.lua”后缀名结尾，如下图。



保存后，界面中按“F7”运行，下框正常打印，表示关联部署正常。



2.3 Lua 基础语法

切入正题，本节开始正式讨论 Lua 语法。以宏常用的数据类型、变量、循环、流控制运算符几个点分别讲

解。

2.3.1 数据类型

常见宏命令中的数据类型：

数据类型	描述
nil	仅此一值，表示无效值（在条件表达式中相当于 false）。
boolean	包含两个值：false 和 true，非正即负。
number	表示双精度类型的实浮点数，简单理解数字。如：1、2、3、4 等。
string	字符串，有双引号或单引号表示。如：“魔兽世界”，'魔兽争霸'。
function	函数，这里即 WOW-API。如：IsUsableAction()。

还有一种数据类型，即：数组，是值相同数据类型的元素按一定顺序排列的集合，可以是一维数组和多维数组。如：{“魔兽世界”，“魔兽争霸”}，表示 2 个字符串类型的数组。
另关于字符串的详细定义使用及转义字符，请自行搜索学习。

2.3.2 变量

变量在使用前，必须在代码中进行声明，即创建该变量。编译程序执行代码之前编译器需要知道如何给语句变量开辟存储区，用于存储变量的值。
Lua 变量有三种类型：全局变量、局部变量、表中的域。Lua 中的变量全是全局变量，那怕是语句块或是函数里，除非用 local 显式声明为局部变量。
局部变量的作用域为从声明位置开始到所在语句块结束。变量的默认值均为 nil。
而在宏命令中，最常见是局部变量，开头以 local 定义，后面一串即为所声明的变量与其赋值。
上面代码中，分别定义 5 个局部变量。其中 a, b 分别为 2 个数组，a 表示数字类型的数组；b 表示字符串类型数组。C 表示字符串。i 和 j 则为赋值，即默认值 nil。

2.3.3 循环

一组被重复执行的语句称之为循环体，能否继续重复，决定循环的终止条件。
循环结构是在一定条件下反复执行某段程序的流程结构，被反复执行的程序被称为循环体。循环语句是由循环体及循环的终止条件两部分组成的。

常见循环类型

循环类型	描述
while	在条件为 true 时，让程序重复地执行某些语句。执行语句前会先检查条件是否为 true
for	重复执行指定语句，重复次数可在 for 语句中控制。

循环控制

循环控制语句用于控制程序的流程， 以实现程序的各种结构方式。

控制语句	描述
break	退出当前循环或语句，并开始脚本执行紧接着的语句。

语法实例：


```
for i=$初始次数,$循环次数,$增长次数 do
    <执行题>
end
```

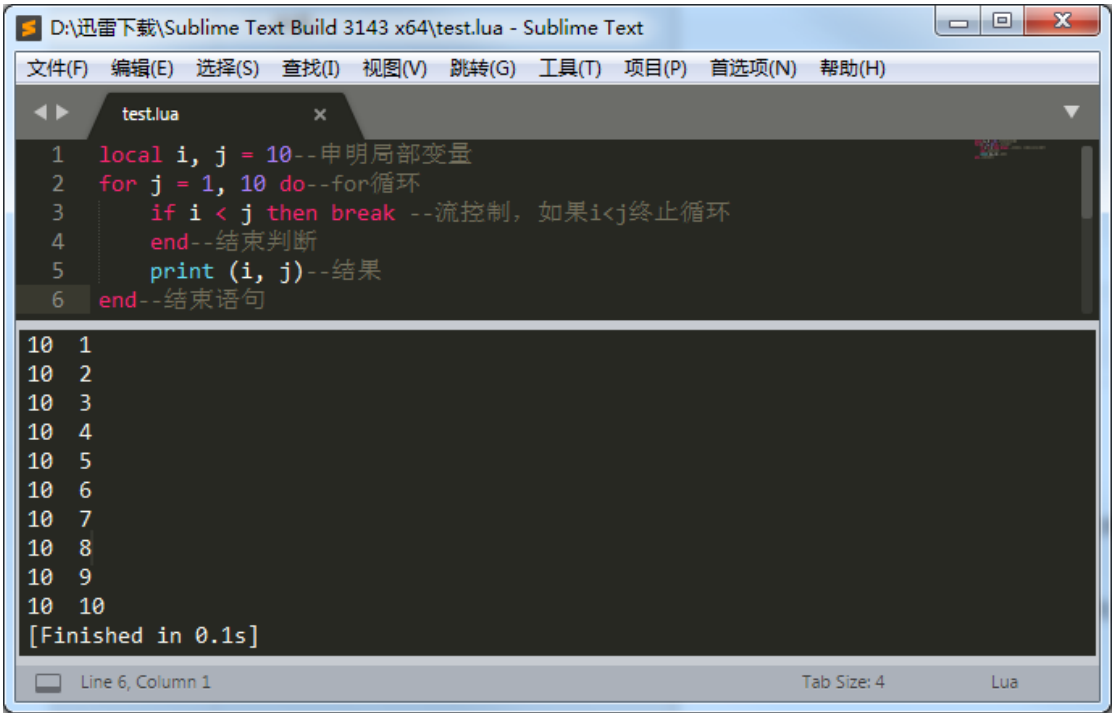
2.3.4 流控制

流程控制语句通过程序设定一个或多个条件语句来设定。在条件为 true 时执行指定程序代码，在条件为 false 时执行其他指定代码。

常见流程控制语句

语句	表述
if	if 语句 由一个布尔表达式作为条件判断，其后紧跟其他语句组成。
if……else	if 语句 可以与 else 语句搭配使用，在 if 条件表达式为 false 时执行 else 语句代码
if……elseif	表示 if 可能有两种以上情况的判断。需要注意与 if……else if 区分，else if 表示子条件；而 elseif 表示另一种情况。

语法实例，结合循环控制语句与运算符：



2.3.5 运算符

运算符，大家并不陌生。不同语言之间大同小异。主要分 4 类：算术运算符、关系运算符、逻辑运算符、其他运算符。

操作运算符：

运算符	描述	实例
-----	----	----

+	加	
-	减、负号	
*	乘	
/	除	
%	取余	11%10=1
^	乘幂	10^2=100

关系运算符：

运算符	描述	实例
==	等于，检测两个值是否相等，相等返回 true，否则返回 false	10==20 为 false
~=	不等于，检测两个值是否相等，相等返回 false，否则返回 true	10~=20 为 true
>	大于，如果左边的值大于右边的值，返回 true，否则返回 false	10>20 为 false
<	小于，如果左边的值大于右边的值，返回 false，否则返回 true	10<20 为 true
>=	大于等于，如果左边的值大于等于右边的值，返回 true，否则返回 false	10>=20 为 false
<=	小于等于，如果左边的值小于等于右边的值，返回 true，否则返回 false	10<=20 为 true

逻辑运算符：

运算符	描述	实例
and	逻辑与操作符。若 A 为 false，则返回 A，否则返回 B。	True and false 为 false
or	逻辑或操作符。若 A 为 true，则返回 A，否则返回 B。	True or false 为 true
not	逻辑非操作符。与逻辑运算结果相反，如果条件为 true，逻辑非为 false。	Not(True and false) 为 false

其他运算符

运算符	描述	实例
..	连接两个字符串	a..b，其中 a 为 "Hello"，b 为 "World"，输出结果为 "Hello World"。
#	一元运算符，返回字符串长度	#"Hello" 返回 5

运算符优先级

^
not - (unary)
* /
+ -
..

< > <= >= ~= ==
and
or
从高到低顺序。

2.3.6 函数

函数，绝大多数指的是玻璃渣给 wow 提供的 API 接口。因能力有限，本教程仅提供下面几种 API 常用接口，并简单描述其所需传参及返回值：动作条（Action）、Buff/Debuff、宠物技能（Pet）、法术（Spell）、单位（Unit）、目标（Targeting）清单，更详细的 API 函数请通过：
http://wowwiki.wikia.com/index.php/World_of_Warcraft_API 访问查询。

另外有些函数前有“Protected”标识，其标识受保护，主要定义为 API 或插件函数不可调用。
先将一些传参定义解释下，便于更好的理解相关的 API 含义。

动作条定义

Slots 即动作条 id（目前看与 index 表示一致）。从 1.11 开始，魔兽世界允许 120 个动作条 ID，编号从 1-120。ID 1-72 用于六个默认栏（见下面的按钮范围）。某些特殊姿态使用 ID 73-120。

默认动作条 id：

主动作条 1：1 - 12
主动作条 2：13 - 24
右动作条 1：25 - 36
右动作条 2：37 - 48
右下动作条：49 - 60
左下动作条：61 - 72

战士姿态动作条

战斗姿态动作条：73 - 84
防御姿态动作条：85 - 96
狂暴姿态动作条：97 - 108

德鲁伊姿态动作条

猫形态：73 - 84
潜行形态：85 - 96
熊形态：97 - 108
咕咕卖萌：109 - 120

盗贼姿态动作条

潜行形态：73 - 84

牧师姿态动作条

暗影形态：73 - 84

姿态定义

在 CastShapeshiftForm 函数中使用，定义不同职业的姿态或光环。
战士：1 战斗姿态、2 防御姿态、3 狂暴姿态。
德鲁伊：1 熊形态、2 海狗形态、3 猫形态、4 旅行形态、5 咕咕形态、6 树形态。
盗贼：1 潜行状态。
神圣牧师：
圣骑士：1 牺牲光环、2 惩戒光环、3 虔诚光环、4 暗影抗光环、5 冰抗光环、6 火抗光环、7 十字军光环。

人物装备界面定义

GetInventorySlotInfo()	slotId
AmmoSlot	0
HeadSlot	1
NeckSlot	2
ShoulderSlot	3
ShirtSlot	4
ChestSlot	5
WaistSlot	6
LegsSlot	7
FeetSlot	8
WristSlot	9
HandsSlot	10
Finger0Slot	11
Finger1Slot	12
Trinket0Slot	13
Trinket1Slot	14
BackSlot	15
MainHandSlot	16
SecondaryHandSlot	17
RangedSlot	18
TabardSlot	19
Bag0Slot	20
Bag1Slot	21
Bag2Slot	22
Bag3Slot	23

动作条

ActionHasRange(slot)

传参：动作条 id，单数字或数字数组。

返回：布尔类型“是”或“否”

作用：确定指定的操作是否是范围限制。

备注：

GetActionCooldown(slot)

传参：动作条 id，单数字或数字数组。

返回：返回指定动作栏冷却时间。

作用：判断动作条中技能是否已冷却（等于 0）。

备注：

IsActionInRange(slot,[unit])

传参：第一位参数为动作条 id，单数字或数字数组；第二位参数为单位。

返回：1 = 是，0 = 否，nil = 不适用。

作用：判断动作条技能是否在使用范围内。

备注：

IsAttackAction(slot)

传参：动作条 id，单数字或数字数组。

返回：是返回 1，否返回 0，nil 为不适用。

作用：判断动作条技能是不是攻击技能。

备注：

IsUsableAction(slot)

传参：动作条 id，单数字或数字数组。

返回：1 或 0。1 表示可用，0 表示不可用或没有足够蓝。

作用：判断动作条中技能是否可用。

备注：

PetHasActionBar()

传参：

返回：

作用：确定玩家是否拥有带有动作栏的宠物。

备注：

StopAttack()

传参：

返回：

作用：如果当前处于攻击状态，则关闭自动攻击。

备注：

IsCurrentAction (actionSlot)

传参：动作条 id

返回：1 标志正在释放，否则 nil。

作用：确认某个动作条正在运行。

备注：

PROTECTED UseAction(slot[, checkCursor[, onSelf]]) - This instructs the interface to use the action associated with the specified ID, optionally on the player, regardless of target. (protected 2.0)

传参： 第一传参：数字型的动作条编号；第二传参：3 个值 (0,1, nil)；第三传参：3 个值 (0,1, nil)。

返回：

作用： 执行制定的操作。

备注：

增益 Buff/减益 Debuff

UnitAura("unit", index or "buffName" [,filter]) - Returns information about a buff/debuff of a certain unit. 返回关于某个单位的 buff / debuff 的信息。

传参：第一位参数为目标（字符串类型），如“player”或“target”；第二位参数代表需要游历的数（数字类型），最大 40；第三位为过滤条件（布尔值），这个可以是“HELPFUL”，“HARMFUL”，“PLAYER”，“RAID”，“CANCELABLE”，“NOT_CANCELABLE”中的任何一个。也可以指定由|分隔的多个过滤器或空格字符链接多个过滤器（例如“有益的 RAID”或“有益的 RAID”==有用的增益，你可以在你的 RAID 上施放）。默认情况下 UnitAura 具有“HELPFUL”作为一个隐式过滤器你不能同时返回有用和有害的。对于 UnitBuff / UnitDebuff，“HELPFUL”或“HARMFUL”都没有意义，将被忽略。

返回：返回的类型很多，如“技能名”、“法术效果等级”、“建议效果次数”、“持续时间”、“debuff”类型等

作用：查询单位增益和减益情况。

备注：

UnitBuff("unit", index or "buffName" [,castable])

传参：同上。

返回：同上。

作用：同上。

备注：单位中也有描述。

UnitDebuff("unit", index or "buffName" [,removable])

传参：同上。多一个若玩家可移除 debuff 返回 1。

返回：同上。

作用：同上。

备注：单位中也有描述。使用方法：name 名称，rank 等级，icon 图标，count 计数，debuffType 减益类型，duration 持续时间，expirationTime 到期时间，unitCaster, isStealable, shouldConsolidate, spellId 法术 id=UnitDebuff (“unit”，index 或[“name”，“rank”] [, “filter”])

宠物

PROTECTED CastPetAction(index, [unitid])

传参：宠物技能 id 索引值，unitid 表示可选的宠物 id

返回：

作用：施放相应的宠物技能。

备注：《U r not perpare》版本可用。

GetPetActionCooldown(index)

传参：与玩家冷却函数一致。

返回：与玩家冷却函数一致。

作用：在指定的宠物动作条插槽中返回宠物动作的冷却时间信息。

备注：《U r not perpare》版本可用。

PROTECTED PetAggressiveMode()

传参：

返回：

作用：设置你的宠物为“进攻”模式。

备注：《U r not perpare》版本可用。

PROTECTED PetAttack()

传参：

返回：

作用：使你的宝宝攻击目标。

备注：

IsPetAttackActive() - Returns true if the pet is currently attacking. 如果宠物目前正在攻击，则返回 true。

传参：

返回：攻击状态 true，否则 false

作用：判断宠物当前是否正在攻击。

备注：

PetStopAttack() - Stop the attack of the pet. 停止宠物的攻击。

传参：

返回：

作用：使你的宝宝停止攻击。

备注：

PROTECTED PetDefensiveMode()

传参：

返回：

作用：设置你的宠物为“防御”模式。

备注：《U r not perpare》版本可用。

PROTECTED PetPassiveMode()

传参：

返回：

作用：设置你的宠物为“被动”模式。

备注：《U r not perpare》版本可用。

法术技能

“spellID”是技能书中的一个索引 id。从上到下依次递增，然后从左到右，然后是类别之间。当玩家学习新的“技能”和“专业”时，法术 ID 将会改变。

PROTECTED CastShapeshiftForm(index)

传参：姿态参数。

返回：

作用：切换职业姿态。

备注：《U r not perpare》版本可用。

GetShapeshiftFormInfo(index)

传参：姿态值，参见姿态定义。

返回：根据参数按顺序返回：

icon（图标位置和名称）——用于该按钮的图标的位置和名称

name（名称）——返回“姿态或光环”的名称（可选）

active（激活状态）——如果此“姿态或光环”当前处于活动状态，则返回 1，否则返回 0（可选）。

castable（可释放）——确定如果可释放。

spellId（法术 ID）——法术/光环的 id。

上面 5 个返回不同的结果，具体依据其表述作用。若需要返回靠后顺序的值，需要先为前面的参数留空位声明，如：_,...a=GetShapeshiftFormInfo(1)，即返回 active 值。

作用：检索当前姿态或光环信息。

备注：见下图例子

Example

```
icon = GetShapeshiftFormInfo(1);
icon, name = GetShapeshiftFormInfo(1);
icon, name, active = GetShapeshiftFormInfo(1);
icon, name, active, castable = GetShapeshiftFormInfo(1);
icon, name, active, castable, spellId = GetShapeshiftFormInfo(1);
```

PROTECTED CastSpell(spellID, "bookType")

传参：spellID 即法术 ID。bookType 技能书类型。

返回：

作用：制定某一类型技能数下的某一法术释放。

备注：《U r not perpare》版本可用。

PROTECTED CastSpellByName("name"[, onSelf])

传参：法术名称，不加等级默认最高等级；第 2 位定义目标，自己、友方、敌方；

返回：

作用：对定义目标释放制定法术。

备注：文档标注 TBC 可用，但实测 1.12 兼容。

GetSpellBookItemInfo(spellID, "bookType")

传参：可以单传参法术名（字符串类型），也可以是双传参：技能书索引号，目标；

返回：法术类型（字符串类型，已知值：“SPELL”，“PETACTION”，“FUTURESPELL”，“FLYOUT”）；或者法术 id。无效则为 nil。

作用：返回玩家法术书中法术类型和法术 ID。

备注：

GetSpellBookItemName(spellID, "bookType")

传参：可以单传参法术 id（与上一条区别，数字类型），也可以是双传参：技能书索引号，目标；

返回：法术类型（字符串类型，已知值：“SPELL”，“PETACTION”，“FUTURESPELL”，“FLYOUT”）；或者法术 id。无效则为 nil。

作用：返回玩家法术书中的法术名称和法术等级。

备注：

GetSpellCooldown(spellName | spellID, "bookType")

传参：可以是字符串类型的“法术名”；可以是数字类型的“法术 id”或“技能栏编号”；也可以是法术类型，支持数组传参。

返回：数字类型“开始时间（以秒为单位）”或“持续时间”；0 则表示法术可用；

作用：判断法术是否可用。

备注：

GetTotemInfo(slot)

传参：数字类型的图腾编号：火=1、土=2、水=3、气=4。

返回：是否包中有图腾试剂；当前激活的图腾名称；图腾开始时间；图腾持续时间；

作用：判断关于图腾的信息。

备注：

IsAttackSpell(spell)

传参：法术名称

返回：1 为攻击法术

作用：判断是否是攻击法术

备注：

IsPassiveSpell(spellID, "bookType")

传参：法术书索引号，从 1 开始递增；法术书类型。

返回：1 位被动，否则为 0

作用：判断法术书中的技能是否是被动技能。

备注：

IsSpellInRange("spellName", [unit])

传参：法术 id；法术类型；法术名；目标；

返回：0 不再射程内；1 射程内；nil 为无效目标；

作用：判断目标是否有效，且是否在射程内。

备注：

IsUsableSpell(spell)

传参：法术 id；法术类型；法术名；目标；

返回：1 为可用；否则为 0，法术未学、无法力、未达到条件等均为 0；布尔值，法力值过低无法释放为 1，否则为 0。

作用：法术是否可用。

备注：

UI QueryCastSequence("sequence")

传参：字符串类型，法术、物品的 id 或索引号。如：reset=1 技能 1，技能 2，技能 3；

返回：

作用：按限定时间，并顺序释放制定法术。等同于 CastSequence。

备注：

SpellHasRange()

传参：法术 id，法术名。

返回：

作用：判断目标是否在法术范围内。

备注：未验证。

SpellsTargeting()

传参：

返回：布尔值，如果法术已对指定目标使用，为 true，否则为 false。

作用：判断技能是否已对制定目标使用。

备注：TBC2.0.1 以后不可用。

PROTECTED SpellStopCasting()

传参：

返回：nil 未不操作；1 表示停止施法。

作用：停止当前施法。

备注：不适用有施法时间技能。

Unit 单位

这些是对一个或多个目标单位起作用的功能。单位由 UnitIds 标识。

PROTECTED AssistUnit("unit")

传参：目标，即：“player”或“target”。可以使用任何定义的 UnitId 值。也可以使用 AssistUnit (pet) 来协助宠物。

返回：nil

作用：指示你的角色帮助指定的单位。

备注：

PROTECTED FocusUnit("unit")

传参：

返回：

作用：设置目标为焦点。

备注：已关闭，但可/RunMacroText("/focus")仍可使用。

PROTECTED ClearFocus()消除您可能设置的任何焦点。

传参：

返回：

作用：消除您可能设置的任何焦点。

备注：已关闭，但/ clearfocus 仍然有效。

SpellCanTargetUnit("unit")

传参：目标，即：“player”或“target”。

返回：布尔值

作用：判断法术是否可以对目标释放。可以返回 true，否则为否。

备注：同时监控距离，超出则无效。

PROTECTED SpellTargetUnit("unit") - Casts the spell awaiting target selection on the specified unit. 在指定单位施放等待目标选择的法术。

传参：目标，即：“player”或“target”。

返回：

作用：

备注：

PROTECTED TargetUnit("unit")

传参：目标，即：“player”或“target”。

返回：nil

作用：选择指定的单位作为当前目标。

备注：

UnitCanAttack("unit", "otherUnit")

传参：第一个单位 unit 为发起攻击方，第二个单位 otherUnit 为被攻击方。

返回：布尔值 true 为可攻击，否则为 false。

作用：判断 2 单位之间能否攻击。

备注：

UnitCanCooperate("unit", "otherUnit")

传参：第一个单位 unit 为发起攻击方，第二个单位 otherUnit 为被攻击方。

返回：布尔值 true 为可合作，否则为 false。

作用：判断 2 单位是否可合作。

备注：

UnitClass("unit")

传参：目标，即：“player”或“target”。

返回：根据客户端语言返回职业名称。或者发挥职业索引值（数字与含义）：0 空、1 战士、2 圣骑、3 猎人、4 盗贼、5 牧师、6 死骑、7 萨满、8 法师、9 术士、10 武僧、11 德鲁伊、12 恶魔猎手。

作用：判断指定目标职业。

备注：请根据版本判断。

UnitClassification("unit")

传参：目标，即：“player”或“target”。

返回：字符串，世界 boss、稀有精英、精英、稀有、普通等

作用：判断指定目标性质。

备注：

UnitCreatureFamily("unit")

传参：目标，即：“player”或“target”。

返回：字符串，如熊、狼、蜘蛛等。

作用：判断目标类型。

备注：

UnitCreatureType("unit")

传参：目标，即：“player”或“target”。

返回：字符串，如人形、龙、野兽等。

作用：判断目标属性。

备注：

UnitExists("unit")

传参：目标，即：“player”或“target”。

返回：布尔类型，返回 1 则存在；否则为 nil

作用：判断目标是否存在。

备注：

UnitHealth("unit")

传参：目标，即：“player”或“target”。

返回：数字类型，制定目标当前生命值百分比；若为 0 则表示不存在。

作用：返回指定单位的当前生命值百分比。

备注：

UnitHealthMax("unit")

传参：目标，即：“player”或“target”。

返回：数字类型，制定目标最大生命值百分比；若为 0 则表示不存在。

作用：返回指定单位的最大生命值百分比。

备注：

UnitIsDead("unit")

传参：目标，即：“player”或“target”。

返回：布尔值，单位死亡返回 1；否则为 0

作用：判断目标是否死亡。

备注：与 UnitIsDeadOrGhost("unit")类似。

UnitIsEnemy("unit", "otherUnit")

传参：第一位制定目标，第二位进行比较单位，如（player，target）比较玩家与目标是否敌对。

返回：布尔值。

作用：判断指定的单位是否是敌人，是返回 true；否返回 false。

备注：

UnitMana("unit")

传参：目标，即：“player”或“target”。

返回：当前目标当前法力/能量/愤怒值。

作用：判断目标当前法力/能量/愤怒值。

备注：3.0.2 版本后弃用，由 UnitPower 取代

UnitManaMax("unit")

传参：目标，即：“player”或“target”。

返回：当前目标最大法力/能量/愤怒值。

作用：判断目标最大法力/能量/愤怒值。

备注：3.0.2 版本后弃用，由 UnitPowerMax 取代

UnitSex("unit") - Returns a code indicating the gender of the specified unit, if known. (1=unknown, 2=male, 3=female) (updated 1.11) 如果已知，则返回指示指定单位的性别的代码。（1 =未知，2 =男，3 =女）（更新 1.11）

传参：目标，即：“player”或“target”。

返回：

作用：

备注：

UnitPowerType("unit")

传参：目标，即：“player”或“target”。

返回：第一字段为字符类型的数值，第二字段为字符串类型的类型名称。

作用：返回对应于指定单位的力量类型（例如法力，怒气或能量）的数字。

备注：返回样例 powerType, powerTypeString = UnitPowerType(UnitId);

UnitAffectingCombat("unit") - Determine if the unit is in combat or has aggro. returns nil if "false" and 1 if "true".

传参：目标，即：“player”或“target”。

返回：

作用：确定制定目标是否在战斗状态，如果是返回 1，否则返回 nil。

备注：

Targeting 针对目标

PROTECTED AssistUnit("unit")

传参：目标。

返回：nil

作用：协助制定单位或目标。

备注：

PROTECTED AttackTarget()

传参：

返回：

作用：不在近战攻击状态则自动攻击；否则停止攻击。

备注：猎人自动远程攻击使用"/startattack"。

PROTECTED ClearTarget()

传参：

返回：

作用：取消当前选定的目标。

备注：

PROTECTED TargetLastEnemy()

传参：

返回：

作用：选择最后一个目标敌人作为当前目标。

备注：

PROTECTED TargetLastTarget()

传参：

返回：

作用：选择最后一个目标作为当前目标。

备注：

PROTECTED TargetNearestEnemy([reverseFlag])

传参：

返回：

作用：选择最近的敌人作为当前目标。

备注：可理解为“tab”键。

PROTECTED TargetNearestEnemyPlayer([reverseFlag])

传参：

返回：

作用：选择最近的敌方玩家作为当前目标。

备注：

PROTECTED TargetNearestFriend([reverseFlag])

传参：

返回：

作用：选择最近的友军单位作为当前目标。

备注：

PROTECTED TargetNearestFriendPlayer([reverseFlag])

传参：

返回：

作用：选择最近的友方玩家作为当前目标。

备注：

PROTECTED TargetNearestPartyMember()

传参：

返回：

作用：选择最近的公会员作为当前目标。

备注：

PROTECTED TargetNearestRaidMember()

传参：

返回：

作用：选择最近的 Raid 成员作为当前目标。

备注：

背包

GetContainerItemLink

传参：

返回：bag #， slot # 中项目的 itemLink。

作用：

备注：

PickupInventoryItem(invSlot)

传参：

返回：

作用：将物品放于人物装备 x 位置中。

备注：

PickupContainerItem (bagID,slot) -

传参：

返回：

作用：将物品放于 x 背包 x 位置中。

备注：

光标

CursorHasItem

传参：

返回：布尔值，判断有无项目

作用：如果光标当前持有一个项目，则返回 true；否则为 0

备注：判断身上装备有无，多用于武器栏判断。

特质

GetComboPoints ()

传参：

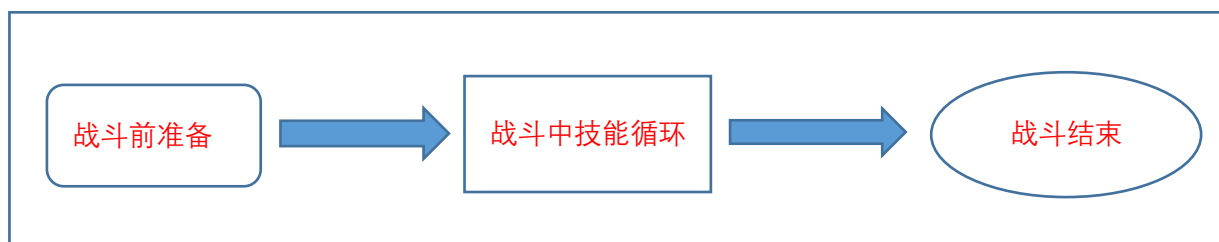
返回：

作用：获得当前连击点数。

备注：

4、实例分析

首先，对于宏编写的中心思想可按战斗流程区分：战斗前准备→战斗中技能循环→战斗结束。



即可以把宏命令分为“战斗前准备宏”、“战斗中技能循环宏”、“战斗结束”三大类。如战士考虑战斗前的姿态判断是否冲锋，进战斗后的状态切换释放技能。再比如战斗结束，猎人假死脱战则最为典型。

本节整理汇总网上的一些宏加以注释，并根据 9 种职业分别列式其叫有针对性的宏，来提供大家一些思路和见解从而更深入的了解宏命令的运作和编写。

授人以鱼不如授人以渔

需要指出，所有宏为了代码工整及注释（重复注释不过多标识）调整了格式，游戏内请删掉回车符及注释。同时部分宏没有验证，还请同学们指出错误，便于修改。

战士

在当前姿态中判断“狂暴之怒”是否可用，如果可用则切换姿态并释放；否则则防御姿态，多用于 MT 在 boss 释放恐惧时开技能。

```
/script
G=GetActionCooldown;--声明 G 为技能冷却函数
if(G(70)<1)then CastShapeshiftForm(3);--当键位 70 的冷却小于 1 秒，则姿态切换狂暴姿态
end;--结束判断
if(IsUsableAction(70)==1)then CastSpellByName("狂暴之怒");--当键位 70 可用，则释放狂暴之怒
end;--结束判断
if(G(70)>1)then CastShapeshiftForm(2);--如果键位 70 冷却大于 1 秒，则姿态切换回防御
end;--结束判断
```

打断宏，在狂暴姿态就“拳击”，战斗或者防御姿态就装备盾牌然后“盾击”。注意此处使用了 UseAction 来关联其他动作。

```

/script
local c,p,u=CastSpellByName,PickupInventoryItem,UseAction;--声明 c、p、u 三个局部变量对应 3 个函数
_,a=GetShapeshiftFormInfo(3);--声明狂暴姿态
if not UnitAffectingCombat("player") then u(36);--如果玩家本身不再战斗中，则使用动作条 36 关联动作。其目的为了使语句继续执行下去，所以这里 36 可以是个坐骑。
else if a then c("拳击");--子条件，如果狂暴姿态则拳击。
else u(27);--子条件否则，使用关联动作条 27，即动作条 27 是个换装动作。
c("盾击");--释放盾击。
end;end;

```

群拉宏，判断当前姿态是不是战斗，是释放“雷霆”；否则“旋风斩”。

```

/script
local c=CastSpellName;
_,a=GetShapeshiftFormInfo(1);
if a then c("雷霆一击");
else c("旋风斩");
end;

```

圣骑

当目标生命值小于 25%，释放圣疗并团队喊话，否则使用圣光术；

```

/script
local a,b="圣疗术","圣光术";
if UnitHealth("target") < 25 then CastSpellByName("a");
else CastSpellByName("b") and SendChatMessage("对<%T>使用了[圣疗]","RAID");
end;end;

```

惩戒骑若点出命令圣印后的技能循环：十字军圣印、审判、命令圣印，同时并循环，可以做成一键宏重复按 1 个键。

猎人

近战一键宏，以“摔绊”>“猫鼬撕咬”>“猛禽一击”的优先顺序释放。当然如果点出“反击”，则调整 a、b、j 三个变量的申明，动作条增加反击按钮即可，正题语句框架不变。（需要 Buff\Debuff 判断函数）

```

/script
local a,b,c,d,i,j={9,10},{“猫鼬撕咬”,“猛禽一击”,“摔绊”,“猛禽一击”};--申明 6 个变量。
if IsBuffActive(“摔绊”,“target”) then for j=1,2 do
if IsUsableAction(a[j]) and GetActionCooldown(a[j])==0 then CastSpellByName(b[j]);--如果目标身上
有摔绊 buff，那么判断动作条 9 和 10 是否可用且以冷却，如果可以则按顺序优先释放。
else CastSpellByName(d) ; -- 否则释放猛禽一击。
end;
end;
else CastSpellByName(c);--如果目标没有摔绊则释放摔绊。
end;

```

姿态切换，以“灵猴”切换“雄鹰”为例，反复按即来回切换。

```

/script
local c,i,b,f=CastSpellByName;
for i=1,16 do b=UnitBuff(“player”,i);--循环自身 buff。
if b and strfind(b,“Monk”) then f=1;--判断是否存在灵猴，有 f=1。
break;--跳出循环
end;
end;
if f then c(“雄鹰守护”);--如果 f=1，则切换为雄鹰。
else c(“灵猴守护”);--否则灵猴。
end;

```

若使用“Buff\Debuff 判断函数”，则代码如下，可以看出代码量骤减。

```

/script
local c=CastSpellByName;
if IsBuffActive(“灵猴守护”,“player”) then c(“雄鹰守护”);--如果 f=1，则切换为雄鹰。
else c(“灵猴守护”);--否则灵猴。
end;

```

起手攻击宏

```

/script
local c =CastSpellByName;
if IsBuffActive(“雄鹰守护”,“player”) then c(“猎人印记”);
PetAttack();
else c(“雄鹰守护”);
end;

```

抓贼宏，判断目标存不存在，有则猎人印记，否则照明弹。可以与起手攻击宏绑定一键。

```
/script
if UnitExists("target") then CastSpellByName("猎人印记");
else CastSpellByName("照明弹");
end;
```

智能钉刺，判断玩家或 npc，针对玩家属性释放钉刺。

```
/script
local c,s,t=UnitClass("target"),"毒蛇钉刺",UnitPowerType("target");--申明 3 个变量。
if UnitIsPlayer("target") then if t==0 then s="蝰蛇钉刺";--判断目标是否是玩家且有蓝职业释放蝰蛇。
if c=="圣骑士" or c=="德鲁伊" then s=s.."(等级 1)";--判断玩家职业是不是圣骑士或德鲁伊，是则释放毒蛇钉刺 1 级。
end;
else s="毒蝎钉刺";--不是则毒蝎。
end;
end;
CastSpellByName(s);--非玩家，释放满级毒蛇
```

收宝宝，假死，冰冻陷阱，一键宏。

```
/script
PetFollow();--优先收宝宝。
if UnitAffectingCombat("player") then CastSpellByName("假死"); --判断战斗状态，是则假死。
else CastSpellByName("冰冻陷阱");--非战斗状态释放陷阱。
end;
```

萨满

不才，无可推荐。

盗贼

战斗天赋一键宏，5 星输出技能可根据实际调整。

```

/script
local p,c,b,i,f=GetComboPoints(),CastSpellByName;
for i=1,16 do b=UnitBuff("player",i);
if b and strfind(b,"SliceDice") then f=1;
break;
end;
end;
if p>0 and not f then c("切割"); --当战斗点数大于 0，目标身上无切割就切割。
elseif p==5 and f then c("割裂"); --当战斗点数=5，目标有切割则割裂。
else c("邪恶攻击"); --以上均非则邪恶攻击。
end;

```

小德

一键变熊后冲锋

```

/script
local c,a,_=CastSpellByName;
_,a=GetShapeshiftFormInfo(1);
if a then c("野性冲锋");
else c("巨熊形态");
end;

```

术士

一键 3DOT 宏，需要注意当前版本无法判断目标身上的 duff 是队友还是自身释放，若目标已有 dot 则无法释放对应 dot

```

/script
local n,s,k,j,d,f={"Abomi","Sarge","Immol"},{"腐蚀术","痛苦诅咒","献祭"};
for k=1,3 do f=nil;
for j=1,16 do d=UnitDebuff("target",j);
if d and strfind(d,n[k]) then f=1;
break;
end;
end;
if not f then CastSpellByName(s[k]);
break;
end;
end;

```

燃烧军团最强连招

```
/script
local g=GetActionCooldown($死亡缠绕编号);
if g==0 then CastSpellByName("死亡缠绕");
else CastSpellByName("恐惧术");
end;
```

搭配判断目标生物类型宏，再特定时候更有效率。

```
/script
local t=UnitCreatureType("target");
if t=="恶魔" or t=="元素生物" then CastSpellByName("放逐术");
else CastSpellByName("恐惧术");
end;
```

恶魔支配可用情况下的恶魔术快速召唤蓝胖子。

```
/script
local c=CastSpellByName;
if UnitCreatureFamily("pet")=="虚空行者" then c("牺牲");
else c("恶魔支配");
SpellStopCasting();
c("召唤虚空行者");
end;
```

恶魔猎犬智能吞噬魔法。优先自身 debuff，再目标 buff。

```
/script
local i,d,s,t,_;
for i=1,16 do d,_=UnitDebuff("player",i);
if d and t=="Magic" then s=1;
break;end;end;
CastSpellByName('吞噬魔法',s);
```

法师

冰箱宏。冰箱可用则用，不能则急速冷却在冰箱。

```
/script
local c,i,b,f,d,_=CastSpellByName;
for i=1,16 do b=UnitBuff("player",i);
if b and strsub(b,5)="Frost" then f=1;
break;
end;
end;
if not f then _,d=GetActionCooldown(你冰箱编号);
if d>2 then c("急速冷却");
SpellStopCasting();
end;
c("寒冰屏障");
end;
```

一键自动 buff

```
/script
local n,s,k,j,b,f={"tArm","ISen","hMag"},{"冰甲术","奥术智慧","魔法抑制"};
for k=1,3 do f=nil;
for j=1,16 do b=UnitBuff("player",j);
if b and strfind(b,n[k]) then f=1;
break;
end;
end;
if not f then CastSpellByName(s[k]);
break;
end;
end;
```

队友敌对后变羊喊话

```
/script
local n,p,i,t=4,"party";
if UnitInRaid("player") then n=40;
p="raid";
end;
for i=1,n do t=p..i;
if UnitCanAttack("player",t) then TargetUnit(t);
CastSpellByName("变形术");
SendChatMessage("%t 被控制，我把他变羊了~",p);
break;
end;
end;
```

指向变羊

```
/script
if UnitExists("mouseover")then TargetUnit("mouseover");
CastSpellByName("变形术");
TargetLastTarget();
else CastSpellByName("变形术");
end;
```

牧师

驱散自身 debuff，否则对目标驱散。

```
/script
local t;
if UnitDebuff("player",1,1) then t=1;
end;
CastSpellByName("驱散魔法",t);
```

一键 buff


```
/script
local n,s,k,j,b,f={"dFor","rFir","iSha"},{"真言术： 韧","心灵之火","防护暗影"};
for k=1,3 do f=nil;
for j=1,16 do b=UnitBuff("player",j);
if b and strfind(b,n[k]) then f=1;
break;
end;
end;
if not f then CastSpellByName(s[k]);
break;
end;
end;
```

非暗影形态使用治疗魔法动作条；暗影形态使用攻击魔法动作条。

```
/script
local i,b,f,p;
for i=1,16 do b=UnitBuff("player",i);
if b and strfind(b,"wform") then f=1;
break;
end;
end;
if f then p=1;
else p=2;
end;
CastSpellByName("暗影形态");
CURRENT_ACTIONBAR_PAGE=p; --依据返回的状态定义动作条编号
ChangeActionBarPage();
```

其他

重置副本

```
/run ResetInstances()
```

视角距离控制

```
/console CameradistanceMaxFactor 50
```

铁皮手雷

```
/script
local b,s,l;
if PlayerFrame.inCombat then AttackTarget(); --如果玩家有选取圈那么攻击目标（选取圈就没了）；
end;
SpellStopCasting(); --停止施法；
for b=0,4 do for s=1,18 do l=GetContainerItemLink(b,s); --b=0 到 4， s=1 到 18， l=（存在消耗品于
背包 0 到 4 的 1 到 18 格里）（这里说一下， 背包默认编号是 0 到 4， 背包内格子是 1 到 18）；
if l and strfind(l,"铁皮手雷") then UseContainerItem(b,s); --如果在背包 0 到 4 的 1 到 18 格里有消耗
品“铁皮手雷”， 那么使用这个物品；
break;
end;
```

5、附件与备注

nga 扩展阅读：<http://bbs.ngacn.cc/read.php?tid=2329182>

插件下载：<http://60addons.com/topic/124-1>

百度 N 服吧：<https://tieba.baidu.com/f?kw=nostalrius&ie=utf-8&tp=0>

NFUWOW：<http://60.nfuwow.com/>

WOW-API:http://wowwiki.wikia.com/index.php/World_of_Warcraft_API