

QKeyMapper 参数映射键完整速查表

本文件列出了 QKeyMapper 中所有需要填写参数的映射键，按常用程度分组，每个映射键附带格式说明和使用示例。

一、文本发送类

SendText(文本)

直接向当前前台窗口发送文本字符串。

格式：SendText(要发送的文字)

示例：SendText>Hello World)

示例：SendText(你好世界)

特殊参数： 支持剪贴板占位符

格式：SendText({{CLIPBOARD_TEXT}})

说明：读取当前 Windows 系统剪贴板中的字符串并发送

PasteText(文本)

通过剪贴板粘贴方式向当前前台窗口发送文本，速度快但依赖程序对粘贴快捷键的支持。

格式：PasteText(要粘贴的文字)

示例：PasteText>Hello World)

特殊参数： 同样支持 {{CLIPBOARD_TEXT}}

格式：PasteText({{CLIPBOARD_TEXT}})

说明：将系统剪贴板中已复制的内容用粘贴方式发送

💡 **PasteText 粘贴模式：** 可在"映射项设定"窗口中选择 `Shift+Insert` 或 `Ctrl+V` 两种粘贴模式，默认 `Shift+Insert`。

二、鼠标坐标点击类

以下映射键用于在指定坐标位置执行鼠标点击。坐标可以是屏幕坐标或窗口内坐标。

屏幕坐标点击

鼠标指针移动到屏幕指定坐标位置后点击（实际移动鼠标指针）。

格式： `Mouse-L(x,y)` ← 鼠标左键点击屏幕坐标 `(x,y)`

`Mouse-R(x,y)` ← 鼠标右键点击

`Mouse-M(x,y)` ← 鼠标中键点击

`Mouse-X1(x,y)` ← 鼠标侧键1点击

`Mouse-X2(x,y)` ← 鼠标侧键2点击

示例： `Mouse-L(500,100)` ← 鼠标左键点击屏幕坐标 `x=500, y=100`

示例： `Mouse-R(1920,1080)` ← 鼠标右键点击屏幕坐标 `x=1920, y=1080`

窗口内坐标点击

鼠标指针移动到当前前台匹配窗口内的相对坐标位置后点击。

格式： `Mouse-L:W(x,y)` ← 鼠标左键点击窗口内坐标 `(x,y)`

`Mouse-R:W(x,y)` ← 鼠标右键点击窗口内坐标

`Mouse-M:W(x,y)` ← 鼠标中键点击窗口内坐标

`Mouse-X1:W(x,y)` ← 鼠标侧键1点击窗口内坐标

`Mouse-X2:W(x,y)` ← 鼠标侧键2点击窗口内坐标

示例： `Mouse-L:W(500,100)` ← 鼠标左键点击当前匹配窗口内坐标 `x=500, y=100`

PostMessage 后台发送（:BG 修饰符）

添加 `:BG` 后，鼠标事件通过 `PostMessage` 直接发送给窗口，**屏幕上的鼠标指针不动**。配合"发送到同名窗口"勾选框，可发送给后台窗口。

- 格式：Mouse-L:BG(x,y)

Mouse-L:W:BG(x,y)

Mouse-R:BG(x,y)

Mouse-R:W:BG(x,y)

Mouse-M:BG(x,y)

Mouse-M:W:BG(x,y)

Mouse-X1:BG(x,y)

Mouse-X1:W:BG(x,y)

Mouse-X2:BG(x,y)

Mouse-X2:W:BG(x,y)
- ← 屏幕坐标，后台发送

← 窗口坐标，后台发送
- 示例：Mouse-L:BG(500,100)

示例：Mouse-L:W:BG(500,100)
- ← 后台发送屏幕坐标点击

← 后台发送窗口内坐标点击

坐标拾取辅助按键名

以下划线后缀（_WindowPoint、_ScreenPoint）结尾的按键名是映射按键选择列表中显示的选项。选择这些按键进行添加时，软件会引导你通过坐标拾取器获取坐标，然后自动将对应的标准格式写入映射表：

按键列表中的名称	操作方式	映射表中自动插入为
Mouse-L_WindowPoint	L-Alt + 鼠标左键点击窗口内目标位置	Mouse-L:W(x,y)
Mouse-R_WindowPoint	同上	Mouse-R:W(x,y)
Mouse-M_WindowPoint	同上	Mouse-M:W(x,y)
Mouse-X1_WindowPoint	同上	Mouse-X1:W(x,y)
Mouse-X2_WindowPoint	同上	Mouse-X2:W(x,y)
Mouse-L_ScreenPoint	L-Ctrl + 鼠标左键点击屏幕目标位置	Mouse-L(x,y)
Mouse-R_ScreenPoint	同上	Mouse-R(x,y)
Mouse-M_ScreenPoint	同上	Mouse-M(x,y)
Mouse-X1_ScreenPoint	同上	Mouse-X1(x,y)
Mouse-X2_ScreenPoint	同上	Mouse-X2(x,y)
Mouse-Move_WindowPoint	同上	Mouse-Move:W(x,y)
Mouse-Move_ScreenPoint	同上	Mouse-Move(x,y)

按键列表中的名称	操作方式	映射表中自动插入为
Mouse-Move_Relative	在主窗口"参数"文本框中手动输入偏移量 dx,dy ， 如： 6,10	Mouse-Move:R(dx,dy)

💡 通过映射按键列表选择添加的坐标映射键，可以在映射表中右键菜单中选择编辑手动修改坐标。

三、鼠标移动类

Mouse-Move(x,y)

将鼠标指针移动到指定屏幕坐标位置。

- 格式：Mouse-Move(x,y)
- Mouse-Move:W(x,y)
- Mouse-Move:BG(x,y)
- Mouse-Move:W:BG(x,y)
- 示例：Mouse-Move(800,600) ← 移动到屏幕坐标 (800,600)
- 示例：Mouse-Move:W(500,400) ← 移动到窗口内坐标 (500,400)

Mouse-Move:R(delta_x,delta_y)

鼠标指针基于当前位置进行相对位移（不移动到绝对坐标，而是移动指定的偏移量）。

- 格式：Mouse-Move:R(delta_x,delta_y)
- 示例：Mouse-Move:R(6,10) ← 指针 x+6, y+10（右移6，下移10）
- 示例：Mouse-Move:R(-6,-10) ← 指针 x-6, y-10（左移6，上移10）
- 示例：Mouse-Move:R(0,8) ← 指针 x不变, y+8（垂直下移8）
- 示例：Mouse-Move:R(-2,0) ← 指针 x-2, y不变（水平左移2）

💡 按键列表中的 Mouse-Move_Relative 选择后需在文本框中输入偏移量，软件自动插入为 Mouse-Move:R(dx,dy) 。

Mouse-Move_WindowPoint 和 Mouse-Move_ScreenPoint 同理，分别对应 Mouse-Move:W(x,y) 和 Mouse-Move(x,y) 。

Mouse-PosSave

保存当前鼠标指针位置。

格式：Mouse-PosSave

Mouse-PosRestore

鼠标指针恢复到之前保存的位置。

格式：Mouse-PosRestore

组合使用示例：

Mouse-PosSave»Mouse-L(300,200)»Mouse-PosRestore

说明：先保存当前位置 → 左键点击 (300,200) → 恢复之前位置

四、运行命令类

Run(命令)

运行指定的程序或命令。

格式：Run(可执行文件路径 [参数])

示例：Run(C:\Windows\notepad.exe)

示例：Run(C:\Windows\system32\calc.exe)

示例：Run(E:\Game\StartGame.exe -windowed)

可选附加参数：

参数	说明
Wait=true	等待被启动的程序执行完毕后才继续后续映射键
Wait=false	不等待（默认行为）

参数	说明
WorkingDir="路径"	指定工作目录
ShowOption=Max	最大化启动
ShowOption=Min	最小化启动
ShowOption=Hide	隐藏启动

示例：Run(E:\Game.exe WorkingDir="E:\Game" ShowOption=Max)

示例：Run(C:\script.bat Wait=true)

系统动作（命令开头指定）：

动作	说明
find	查找
explore	用资源管理器打开
open	打开文件
edit	用默认编辑器编辑
openas	选择程序打开
print	打印
properties	显示文件属性
runas	管理员权限运行

示例：Run(explore C:\MyGame)

示例：Run(properties C:\MyGame\game.exe)

示例：Run(runas C:\MyGame\patcher.exe)

五、音量控制类

SetVolume(数值)

设置当前播放设备音量。

格式: SetVolume(数值)

示例: SetVolume(10) ← 音量设置为10%

示例: SetVolume(+5) ← 增加5%

示例: SetVolume(-20) ← 减少20%

示例: SetVolume(30.8) ← 精确设置为30.8%

示例: SetVolume(Mute) ← 切换静音状态

示例: SetVolume(MuteOn) ← 设置为静音

示例: SetVolume(MuteOff) ← 取消静音

SetVolume🔊(数值)

功能同上，但调整音量时会显示提示信息。

格式: SetVolume🔊(数值)

示例: SetVolume🔊(+10)

SetMicVolume(数值)

设置当前麦克风设备音量。

格式: SetMicVolume(数值)

示例: SetMicVolume(50.5) ← 设置为50.5%

示例: SetMicVolume(+10) ← 增加10%

示例: SetMicVolume(-5.5) ← 减少5.5%

示例: SetMicVolume(Mute) ← 切换静音

示例: SetMicVolume(MuteOn) ← 设置为静音

示例: SetMicVolume(MuteOff) ← 取消静音

SetMicVolume🎤(数值)

功能同上，带提示信息显示。

格式: SetMicVolume🔧(数值)

六、循环与条件类

Repeat{...}

将 {} 内的映射按键循环发送指定次数。

格式: Repeat{映射内容}x循环次数

示例: Repeat{A🔧50}x5

示例: Repeat{B+C}x3

示例: A🔧50»Repeat{C+D»E🔧50}x2»F

注意:

- 循环次数范围 1~99999
- 也可在"映射项设定"窗口中勾选"按键按下时循环"或"循环次数"来实现循环, 无需手动写 Repeat{}

OnlyOnce{...}

{ } 内的映射按键仅在按键序列首次发送时执行一次, 后续循环中跳过。

格式: OnlyOnce{映射内容}

示例: OnlyOnce{Mouse-L🔧50}»A🔧50»B🔧50

带重复次数:

格式: OnlyOnce{映射内容}xN

说明: xN 表示 OnlyOnce 部分执行次数

七、按键序列控制类

KeySequenceBreak

打断所有当前正在执行的按键序列发送。

格式：KeySequenceBreak

或 KeySequenceBreak(原始按键)

示例：KeySequenceBreak ← 打断所有运行中的按键序列

示例：KeySequenceBreak(A) ← 仅打断A原始按键对应的运行中按键序列

KeySequenceToggle

单按键切换所有按键序列的暂停/继续状态。

格式：KeySequenceToggle

KeySequencePause

暂停所有当前正在执行的按键序列发送。

格式：KeySequencePause

KeySequenceContinue

继续所有之前被暂停的按键序列发送。

格式：KeySequenceContinue

KeySequenceToggle(原始按键)

单按键切换特定原始按键对应映射项按键序列的暂停/继续。

格式：KeySequenceToggle(原始按键)

示例：KeySequenceToggle(A)

KeySequencePause(原始按键)

暂停特定原始按键对应映射项的按键序列发送。

格式：KeySequencePause(原始按键)

示例：KeySequencePause(A)

KeySequenceContinue(原始按键)

继续特定原始按键对应映射项的按键序列发送。

格式：KeySequenceContinue(原始按键)

示例：KeySequenceContinue(A)

八、解锁类

Unlock(原始按键)

解锁指定原始按键的锁定状态。

格式：Unlock(原始按键名)

示例：A->B (B键勾选锁定)

C->Unlock(A)

说明：A键按下后B键持续按下（锁定状态），按下C键可解锁A键，释放B键

双击/长按触发解锁：

格式：Unlock(原始按键X) ← 双击原始按键时触发解锁

Unlock(原始按键⊕数值) ← 长按指定毫秒后触发解锁

示例：Unlock(AX) ← 双击A键时解锁

示例：Unlock(A⊕500) ← 长按A键500毫秒时解锁

💡 配合"映射项设定"中的"禁用原始按键解锁"勾选框，锁定后的按键只能通过 Unlock() 或停止映射来解除锁定。

九、映射表切换类

SwitchTab(映射表Tab名)

映射过程中切换到指定名称的映射表 Tab。

格式：SwitchTab(Tab名称)

示例：SwitchTab(Tab1)

示例：SwitchTab(战斗模式)

SwitchTab (映射表Tab名)

切换映射表并记录保存此 Tab 名，下次切换到此设定时会记住此 Tab。

格式：SwitchTab (Tab名称)

示例：SwitchTab (Tab2)

十、宏调用类

Macro(宏名)

发送当前设定项中已添加的宏。宏在"映射宏列表"→"宏"标签页中编辑。

格式：Macro(宏名)

示例：Macro(宏A)

示例：Macro(连招一套)x3 ← 发送3次

UniversalMacro(通用宏名)

发送通用宏（所有设定项共用）。通用宏在"映射宏列表"→"通用宏"标签页中编辑。

格式：UniversalMacro(通用宏名)

示例：UniversalMacro(通用宏B)

示例：UniversalMacro(通用宏B)x5 ← 发送5次

💡 宏内容编写格式与映射按键内容相同。

十一、悬浮按钮控制类

ShowFButton(原始按键)

显示指定原始按键对应的悬浮按钮。

格式：ShowFButton(原始按键)

示例：ShowFButton(A)

说明：显示A原始按键对应的悬浮按钮

HideFButton(原始按键)

隐藏指定原始按键对应的悬浮按钮。

格式：HideFButton(原始按键)

示例：C->HideFButton(A)

说明：C键按下时隐藏A原始按键对应的悬浮按钮

💡 可配合锁定功能让特定悬浮按钮切换隐藏/显示状态。

十二、虚拟按钮类

VButton{标签}

添加一个虚拟按钮，鼠标点击后发送此按钮对应的映射按键内容。虚拟按钮会显示在虚拟按钮面板中。

格式：VButton{按钮标签}

示例：VButton{Button1}

示例：VButton{开火}

示例：VButton{一键连招}

💡 按钮的布局、颜色、尺寸等可在"虚拟按钮面板设定"中自定义。
勾选"启用悬浮按钮"后，虚拟按钮将以悬浮按钮形式单独显示，不在虚拟按钮面板中显示。

十三、手柄触摸板控制类

GamepadTouchpadOn

启用手柄触摸板转鼠标功能。

格式：GamepadTouchpadOn

GamepadTouchpadOff

禁用手柄触摸板转鼠标功能。

格式：GamepadTouchpadOff

GamepadTouchpadToggle

切换手柄触摸板转鼠标功能的启用/禁用状态。

格式：GamepadTouchpadToggle

可选后缀：

后缀	说明	示例
🎮	切换时显示提示通知	GamepadTouchpadToggle🎮
@N	指定玩家编号（0~9）	GamepadTouchpadToggle@0
🎮@N	通知 + 玩家编号	GamepadTouchpadOn🎮@1

- 示例：GamepadTouchpadToggle🎮
- 示例：GamepadTouchpadOn@1
- 示例：GamepadTouchpadOff🎮@2
- ← 切换并显示通知
- ← 启用1号玩家的触摸板
- ← 禁用2号玩家的触摸板并提示

💡 不指定 @N 时作用于所有已连接的手柄。

十四、虚拟手柄轻推值类

以下虚拟手柄按键可添加 [数值] 后缀指定轻推/轻按的按压值，范围 0~255。

格式：vJoy-按键名[数值]

示例：vJoy-LS-Up[150] ← 左摇杆上推，轻推值为150

示例：vJoy-Key12(RT)[100] ← 右扳机轻按值为100

支持的按键列表：

vJoy-Key11(LT)

vJoy-Key12(RT)

vJoy-LS-Up

vJoy-LS-Down

vJoy-LS-Left

vJoy-LS-Right

vJoy-RS-Up

vJoy-RS-Down

vJoy-RS-Left

vJoy-RS-Right

💡 数值为 255 时按键效果等同于不加后缀的按到底效果。

注意：按键序列中的 vJoy- 普通虚拟手柄按键未指定 ⌚ 时，默认按下保持 20 毫秒。

十五、虚拟手柄编号后缀

vJoy-按键@编号

为虚拟手柄按键指定映射到哪个虚拟手柄设备。编号范围 0~3。

格式：vJoy-按键名@编号

示例：vJoy-Key1(A/x)@0 ← 映射到0号虚拟手柄的A键

示例：vJoy-LS-Up@1 ← 映射到1号虚拟手柄的左摇杆上推

💡 不添加 @编号 时，默认映射到 0 号虚拟手柄。

十六、虚拟手柄扳机限定类

vJoy-LT-Max[数值]

限定虚拟手柄左扳机键（vJoy-Key11）的最大按压值。

格式：vJoy-LT-Max[数值]

示例：vJoy-LT-Max[150] ← 左扳机按压值最大限制为150（范围0~255）

vJoy-RT-Max[数值]

限定虚拟手柄右扳机键（vJoy-Key12）的最大按压值。

格式：vJoy-RT-Max[数值]

示例：vJoy-RT-Max[50] ← 右扳机按压值最大限制为50

💡 限定范围 0~255，不使用时默认值 255（按到底）。

十七、虚拟手柄摇杆半径限定类

vJoy-LS-Radius[数值]

限定虚拟手柄左摇杆的偏移半径范围。

格式：vJoy-LS-Radius[数值]

vJoy-LS-Radius[U=数值,D=数值,L=数值,R=数值]

示例：vJoy-LS-Radius[150] ← 左摇杆整体半径限定为150

示例：vJoy-LS-Radius[U=200,D=0,L=150,R=150] ← 上方向最大200，下方向禁止偏移0，左右各150

vJoy-RS-Radius[数值]

限定虚拟手柄右摇杆的偏移半径范围。

格式: vJoy-RS-Radius[数值]

vJoy-RS-Radius[U=数值,D=数值,L=数值,R=数值]

示例: vJoy-RS-Radius[U=120,R=180]

← 上方向最大120, 右方向最大180

其他未指定方向默认255

示例: vJoy-RS-Radius[150,U=100,R=200]

← 整体半径150, 上覆盖为100, 右覆盖为200

💡 参数说明: U (上)、D (下)、L (左)、R (右), 数值范围 0~255, 未指定的方向使用默认值255。

十八、虚拟手柄摇杆偏移移动类

vJoy-LS-Move[X=数值,Y=数值]

设置虚拟手柄左摇杆的偏移位置 (绝对偏移或相对偏移)。

绝对偏移 (X/Y 相对于摇杆中心):

格式: vJoy-LS-Move[X=数值,Y=数值]

示例: vJoy-LS-Move[X=-60,Y=100]

← 左摇杆水平左偏60, 垂直下偏100

示例: vJoy-LS-Move[X=0,Y=80]

← 左摇杆水平居中, 垂直下偏80

示例: vJoy-LS-Move[X=200,Y=0]

← 左摇杆水平右偏200, 垂直居中

相对偏移 (RX/RX 基于当前摇杆位置):

格式: vJoy-LS-Move[RX=数值,RY=数值]

示例: vJoy-LS-Move[RX=6,RY=10]

← 基于当前位置, 水平右偏6, 垂直下偏10

示例: vJoy-LS-Move[RX=0,RY=8]

← 基于当前位置, 垂直下偏8 (水平不变)

混合偏移:

示例: vJoy-LS-Move[RX=6,Y=200]

← 水平基于当前位置右偏6, 垂直绝对覆盖为下偏200

vJoy-RS-Move[X=数值,Y=数值]

设置虚拟手柄右摇杆的偏移位置, 用法同上。

示例（绝对偏移）：

vJoy-RS-Move[X=-60,Y=-100] ← 右摇杆水平左偏60，垂直上偏100

vJoy-RS-Move[X=200,Y=0] ← 右摇杆水平右偏200，垂直居中

示例（相对偏移）：

vJoy-RS-Move[RX=-6,RY=-10] ← 基于当前位置，水平左偏6，垂直上偏10

vJoy-RS-Move[RX=-2,RY=0] ← 基于当前位置，水平左偏2（垂直不变）

示例（混合偏移）：

vJoy-RS-Move[X=-200,RY=-10] ← 水平绝对覆盖为左偏200，垂直基于当前位置上移10

vJoy-Mouse2LS / vJoy-Mouse2RS

将鼠标移动映射为虚拟手柄左/右摇杆偏移。这两个键作为**原始按键**使用——将其设为映射项中的原始按键，按住时鼠标移动会转换为对应摇杆偏移。

格式：vJoy-Mouse2LS ← 鼠标→左摇杆
vJoy-Mouse2RS ← 鼠标→右摇杆

💡 它们不带 @编号 后缀，始终作用于 0 号虚拟手柄。

十九、物理手柄编号后缀

Joy-按键@编号

为物理手柄原始按键指定区分玩家编号。编号范围 0~9。

格式：Joy-按键名@编号

示例：Joy-Key1(A/x)@0 ← 0号物理手柄的A键

示例：Joy-LS-Up@1 ← 1号物理手柄的左摇杆上推

💡 不添加 @编号 时，映射来自所有物理手柄的输入。

二十、Forza 自动映射键

以下映射键用于 Forza 系列赛车游戏的自动油门/刹车控制。

格式: vJoy-Key11(LT)_BRAKE[INIT=数值,THR=数值] ← LT作刹车自动控制
vJoy-Key11(LT)_ACCEL[INIT=数值,THR=数值] ← LT作油门自动控制
vJoy-Key12(RT)_BRAKE[INIT=数值,THR=数值] ← RT作刹车自动控制
vJoy-Key12(RT)_ACCEL[INIT=数值,THR=数值] ← RT作油门自动控制

参数说明:

参数	含义	取值范围
INIT	扳机初始按压值	0~255
THR	触发阈值（游戏内数值）	0~10000（支持最多5位小数）

示例: vJoy-Key11(LT)_BRAKE[INIT=150,THR=85.5] ← LT初始值150，当游戏数据达到85.5时触发刹车

示例: vJoy-Key12(RT)_ACCEL[INIT=100,THR=200] ← RT初始值100，阈值200时触发油门

 两个参数至少填写一个。该功能依赖 Forza 游戏数据读取，仅在支持的游戏上有效。

完整格式速查索引

映射键	参数格式	参数含义
SendText(文本)	文本 或 {{CLIPBOARD_TEXT}}	要发送的字符串 / 剪贴板内容
PasteText(文本)	文本 或 {{CLIPBOARD_TEXT}}	要粘贴的字符串 / 剪贴板内容
Mouse-L(x,y)	(x坐标,y坐标)	屏幕坐标，可选 :W（窗口坐标）、:BG（后台发送）
Mouse-L:W(x,y)	(x坐标,y坐标)	窗口内相对坐标
Mouse-L:BG(x,y)	(x坐标,y坐标)	后台发送不移动鼠标指针

映射键	参数格式	参数含义
Mouse-Move(x,y)	(x坐标,y坐标)	移动到的目标坐标，可选 :W、:BG
Mouse-Move:R(dx,dy)	(delta_x,delta_y)	相对偏移量
Mouse-PosSave	无	保存指针位置
Mouse-PosRestore	无	恢复指针位置
Run(命令 参数)	命令+参数	可执行文件路径及参数，可选 Wait、WorkingDir、ShowOption
SetVolume(数值)	数值/+/-/Mute	音量数值或增减量
SetVolume🔊(数值)	数值/+/-/Mute	音量数值或增减量（带提示）
SetMicVolume(数值)	数值/+/-/Mute	麦克风音量数值或增减量
SetMicVolume🎤(数值)	数值/+/-/Mute	麦克风音量数值或增减量（带提示）
Repeat{内容}xN	{映射键}x次数	循环发送的映射键内容和循环次数
OnlyOnce{内容}	{映射键}	仅首次发送的映射键内容
KeySequenceBreak(原始按键)	按键名(可选)	要打断的原始按键 (不填则打断全部)
KeySequenceToggle(原始按键)	按键名	切换暂停/继续的原始按键
KeySequencePause(原始按键)	按键名	暂停的原始按键
KeySequenceContinue(原始按键)	按键名	继续的原始按键
Unlock(原始按键)	按键名，可选 ✕/🕒 N 后缀	要解锁的原始按键，支持双击/ 长按触发
SwitchTab(Tab名)	名称	要切换到的映射表Tab名称
SwitchTab📄(Tab名)	名称	要切换到的映射表Tab名称 (并保存)
Macro(宏名)	名称	宏的名称，可选 xN 重复次数
UniversalMacro(宏名)	名称	通用宏的名称，可选 xN 重复次数
ShowFButton(原始按键)	按键名	要显示悬浮按钮对应的原始按键

映射键	参数格式	参数含义
HideFButton(原始按键)	按键名	要隐藏悬浮按钮对应的原始按键
VButton{标签}	{标签文本}	虚拟按钮上显示的标签文字
GamepadTouchpadOn	可选 🎮 、 @N	启用手柄触摸板（可指定玩家编号）
GamepadTouchpadOff	同上	禁用手柄触摸板
GamepadTouchpadToggle	同上	切换手柄触摸板
vJoy-按键[数值]	[数值]	轻推值，0~255
vJoy-按键@编号	@编号	虚拟手柄设备编号 0~3
vJoy-LT-Max[数值]	[数值]	最大按压值，0~255
vJoy-RT-Max[数值]	[数值]	最大按压值，0~255
vJoy-LS-Radius[数值]	[数值] 或 [U=,D=,L=,R=]	半径限定值或各方向独立限定
vJoy-RS-Radius[数值]	同上	同上
vJoy-LS-Move[X=,Y=]	[X=,Y=] 或 [RX=,RY=]	绝对或相对偏移坐标
vJoy-RS-Move[X=,Y=]	同上	同上
vJoy-Mouse2LS	无	鼠标→左摇杆映射（原始按键，作0号手柄）
vJoy-Mouse2RS	无	鼠标→右摇杆映射（原始按键，作0号手柄）
Joy-按键@编号	@编号	物理手柄玩家编号 0~9
vJoy-Key1x(LT)_BRAKE[INIT=,THR=]	[INIT=数值,THR=数值]	Forza 自动刹车/油门



以上文档基于 QKeyMapper 源码整理，具体功能表现以实际软件版本为准。