
宾语自动化参数说明

- 宾语自动化参数说明
 - 目标
 - 其他文档
 - info 宾语介绍
 - * info 宾语中基本参数介绍
 - * info 宾语的其他说明
 - * info 中的参数类型
 - tree_info
 - * tree_info 必填参数
 - * tree_info 可选参数
 - * tree_info 可被引用的其他参数
 - object_info
 - * object_info 必填参数
 - * object_info 可选参数
 - dictionary_info
 - * dictionary_info 必填参数
 - * dictionary_info 可选参数
 - inadd_info
 - * inadd_info 必选参数
 - * inadd_info 可选参数
 - mtext_info
 - * mtext_info 必选参数
 - * mtext_info 可选参数
 - building_info
 - * building_info 必填参数
 - * building_info 可选参数
 - * building_info 可被引用的其他参数
 - teamDetect_info
 - * teamDetect 必选参数
 - * teamDetect 可选参数
 - * teamDetect_info 可被引用的其他参数
 - numDetect_info
 - * numDetect 必选参数

-
- ★ numDetect 可选参数
 - ★ numDetect_info 可被引用的其他参数
 - multiText_info
 - ★ multiText 必选参数
 - ★ multiText 可选参数
 - multiRemove_info
 - ★ multiRemove 必选参数
 - ★ multiRemove 可选参数
 - multiAdd_info
 - ★ multiAdd 必选参数
 - ★ multiAdd 可选参数
 - flash_info
 - ★ flash_info 必选参数
 - ★ flash_info 可选参数
 - ★ flash_info 可被引用的其他参数
 - step_info
 - ★ step_info 必选参数
 - ★ step_info 可选参数
 - ★ step_info 可被引用的其他参数
 - idcheck_info
 - ★ idcheck_info 必填参数
 - ★ idcheck_info 可选参数
 - ★ idcheck_info 可被引用的其他参数
 - time_info
 - ★ time_info 必填参数
 - ★ time_info 可选参数
 - 使用
 - ★ 命令
 - ★ 命令参数
 - ★ 其他特性
 - ★ 注意事项
 - 所有时间单位
 - team 设置
 - 造成混淆的宾语
 - 自动化后的地图处理
-

-
- [标记问题](#)
 - [字符问题](#)
 - [unit/team](#)

目标

该命令行工具旨在简化城夺地图的制作。对于需要在不同位置放置相同的一个或几个宾语时，但仅仅一些细节不同的时候，可以使用该工具简化操作。

只需要按照命令行教程的格式在地图文件中添加额外的 info 宾语（需要填写大量参数）和标记宾语（仅有名称需要填写, 类型不需要填写），标记宾语就会根据 info 宾语自动生成想要的宾语。

具体来说，程序会识别标记宾语名称的前缀 (prefix)，来找到指定的 info 宾语，并根据该 info 宾语的内容按照约定导入相关属性。相关属性可指定标记宾语的必填参数和可选参数，标记宾语可以依照 info 宾语的格式 (args 和 opargs) 填写参数。

其他文档

如果想循序渐进学习宾语自动化，请阅读[宾语自动化引导教程](#)。

[宾语自动化地图示例](#)是一份地图文件，可供实际作图是复制粘贴和参考。还有一份该地图示例用法的具体说明，[宾语自动化示例说明](#)。

info 宾语介绍

info 宾语是执行一切自动转换的基础，地图作者需要将生成的不同组宾语中相同的参数放入 info 宾语，并将不同的参数设置为在标记宾语中才会填写的参数。还有许多参数可以参与各种不同的功能，后面会有详细的介绍。

info 宾语的名字必须以 xxx_info 开头，后面可以跟任意字符。后方可以存在“ ,d”，表明添加了删除标记，该 info 宾语将不能参与宾语生成。如果存在“ ,D”，该 info 宾语不仅不能参与宾语生成，本身无论如何也将会被彻底删除。

info 宾语中基本参数介绍

prefix: 填写标记宾语的前缀（或者其他 info 宾语的引用），在标记宾语名称开头加入 prefix 即证明该标记宾语使用该 info 宾语的格式。

isprefixseg: 标记宾语中，前缀之后加入必填参数时，是否跟一个“ .”。

args: 声明标记宾语的必填参数。比如“ aunit,str;num,str”，这一串表明标记宾语必须填入 2 个参数，第一个是代入 aunit，第二个代入 num。参数名后面目前支持 str 和 bool 格式。bool 格式下，可以输入“ true” 或“ false”。标

记宾语在前缀之后添加 supplyDepot.1 意思是 aunit 为 supplyDepot, num 为 1。如果 info 宾语的 prefix 为 e, 那么标记宾语将填 esupplyDepot.1。如果 isprefixseg 标记了, 那么标记宾语将填 e.supplyDepot.1。

opargs: 声明标记宾语的选填参数。比如“ u,aunit,str,supplyDepot;n,num,str,2”, 这一串表明 aunit 和 num 是可以选填的 (第一个参数必须只有一个字母)。想填写 aunit 为 supplyDepot 时需要输入“ ,usupplyDepot”。想填写 num 为 1 时需要输入“ ,n1”。注意, 即使没有填写 num, num 默认为 2。如果出现比如“ ,uNone” 时, 该项将被无视。也就是说 None 参数无法导入。如果格式为 bool。那么, 如果在选填参数默认或者 info 参数中没有该项, 只要填写了该选项, 不需要额外输入参数, 认为是“ true”。如果有该项, 那么填写该选项后将会取逆 (相反)。

,d是默认存在的选项, 加入即证明该标记宾语被打上了删除标记。

,D是默认存在的选项, 加入即证明该标记宾语将会被彻底删除。

参数事实上可以不添加info有的参数, 可以自创一个参数, 但在实际参数中引用。比如args设置为“`buildingname,str`”, 而info中写aunit:{buildingname}, inaddunit:{buildingname}。可以实现复用。

与id前缀有关的选项、附属宾语导入prefix均不能有{ }, 不会进行翻译。

cite_name: 标记宾语的引用, 如果没有, 无法被引用。比如在某 info 宾语中设置 args 为“ cite_name,str”, 在使用该 info 宾语格式的标记宾语中名称为“ e.td”, 其中 e 是前缀。在 cite_name 后面加“ .” 再添加属性名称, 即可实现引用。比如 td.aunit 表明将会引用该标记宾语的其中 aunit 的内容。

标志宾语引用不得发生重复。

不得出现循环引用, 文件中靠前的宾语不得引用靠后的宾语。

引用过程不能直接引用已有复杂参数 (类型不为str和bool)

特别的, 引用是有限制的。仅有info中的参数, args,opargs,brace中的参数, 产生的id会被引用。因此, 如果想要自创参数被引用, 必须将其写进brace中。而dictionary_info没有这个限制, 无需特别写进brace。可引用的其他参数也可被引用。

brace: 字符串数组。在最后会将数组中的所有键进行字符串翻译, 以为外部引用提供良好的代入环境。比如, 设置brace为“ a”, a为“ {setidTeam0_0}”, cite_name为“ m1”。这样外部出现 m1.a 时, 相当于得到了队伍检测的第一个 id。

info 宾语的其他说明

无参数表明将不支持 args 和 opargs。附属 info 宾语表示该 info 宾语不能独立指定相关标记宾语, 只能被其他 info 宾语引用自己的 prefix 导入参数, 并且不支持 brace。

info 中的参数类型

请注意，is 前缀的需要改变其数据类型为 bool（文本文件中的形式为 type= "bool" value= "true" 或者 value= "true"）。

存在一些需要输入多个数据的属性，这些属性格式总结如下：

字符串数组：字符串之间用 ' , ' 隔开

数字数组：数字之间用 ' ' 隔开

字符串二维数组：第一维用 " ; " 隔开，第二维用 " , " 隔开

数字二维数组：第一维用 " , " 隔开，第二维用 " " 隔开

info 宾语内的属性也可以互相引用。假如 info 宾语内有一个属性为 aunit:supplyDepot。那么，设置另一个属性为 inaddunit:{aunit} 即可实现引用（引用自己标记宾语的属性）。

tree_info

这是一个可以产生其他标志宾语的 info 宾语。产生的新标志宾语将会继续产生新宾语。会形成类似一棵树的结构。

tree_info 必填参数

prefix: 标记宾语引用前缀。（[info 宾语默认参数](#)）

name: 字符串二维数组。用; 分割开来新产生标志宾语的 name，建议大量使用引用。

tree_info 可选参数

isprefixseg: 可选，默认为否。（[info 宾语默认参数](#)）

args: 可选，默认没有，必填参数。（[info 宾语默认参数](#)）

opargs: 可选，默认没有，可选参数。（[info 宾语默认参数](#)）

cite_name: 可选，默认没有，标志宾语标记。（[info 宾语默认参数](#)）

brace: 可选，默认没有，外部引用翻译列表。（[info 宾语默认参数](#)）

idprefix: 可选，字符串二维数组，默认没有。每一列第一个表示要获取的 id 前缀，第二个表示要获取的 id 前缀的数量。这些 id 应当被用于 cite_name。

exist: 可选，字符串数组，默认全部为 true。当只有一项时，该计算结果对所有分支均适用。每一项都将进行计算，结果为 true 时，对应位置的标志宾语将正常部署。结果为 false 或者为其他结果时，对应位置宾语将不能正常部署。

offset: 可选, 数字二维数组, 默认" 0 0", 表示不同分支标记宾语偏移。当只有一个坐标时, 该偏移对所有分支均适用。不同分支之间用逗号隔开, 一个分支有 x 和 y 两个坐标, 中间空格隔开。填写例子" 0 -20,-20 0"。

offsetsize: 可选, 数字二维数组, 默认" 0 0", 表示不同分支标记宾语大小改变。当只有一个坐标时, 该偏移对所有分支均适用。不同分支之间用逗号隔开, 一个阵营有 x 和 y 两个坐标, 中间空格隔开。填写例子" 0 40,40 0"。

tree_info 可被引用的其他参数

idprefix{i}_{j} 是检测的 id。i 是第几个 id 前缀产生的 id, j 是该 id 前缀获得的第几个 id。

object_info

这是一个任意生成一个宾语的 info 宾语。可进行时间修正。

object_info 必填参数

prefix: 标记宾语引用前缀。(info 宾语默认参数)

objectType: 必选, object 的类型。

object_info 可选参数

isprefixseg: 可选, 默认为否。(info 宾语默认参数)

args: 可选, 默认没有, 必填参数。(info 宾语默认参数)

opargs: 可选, 默认没有, 可选参数。(info 宾语默认参数)

cite_name: 可选, 默认没有, 标志宾语标记。(info 宾语默认参数)

brace: 可选, 默认没有, 外部引用翻译列表。(info 宾语默认参数)

name: 可选, 默认为" ", 宾语名字。

offset: 可选, 数字数组, 默认为" 0 0", 宾语偏移。

offsetsize: 可选, 数字数组, 默认为" 0 0", 宾语大小改变。

time_prefix: 可选, 默认不存在。将会导入对应 time_info 的数据, 进行时间修正。

所有可能的宾语参数都能添加, 标志宾语将会按程序生成一个对应的宾语。当宾语参数结果为空或为" None" 时不产生。

dictionary_info

这是一个可以书写键值对以供引用的 info 宾语，任何键值对均可写入。info 宾语写完后，再写一个标志宾语就可以进行引用了。

dictionary_info 必填参数

prefix: 标记宾语引用前缀。(info 宾语默认参数)

cite_name: 第一个必填参数，标志宾语标记。(info 宾语默认参数)

brace: 可选，默认没有，外部引用翻译列表。(info 宾语默认参数)

dictionary_info 可选参数

isprefixseg: 可选，默认为是。(info 宾语默认参数)

还有任意键值对。

inadd_info

初始单位添加宾语，是附属宾语，且无参数。

inadd_info 必选参数

prefix: 其他 info 宾语引用标记。(info 宾语默认参数)

inaddteam: 当 isinadd 为是时必选，表示单位添加的队伍。

inadd_info 可选参数

isinadd: 可选，默认为是，在 building_info 中默认为否，是否添加城市的初始刷新。

inaddwarmup: 当 isinadd 为是时可选，默认为 {addWarmup}，表示单位添加的 warmup。为 0s 时，不产生 warmup 属性。

inaddunit: 当 isinadd 为是时可选，默认 {aunit}，表示单位添加的类型。

inaddspawnnum: 当 isinadd 为是时可选，默认为 1，表示单位添加的数量。

inaddisshowOnMap: 当 isinadd 为是时可选，默认为否，启用时，初始城市生成时小地图显示。

inaddname: 当 isinadd 为是时可选，默认为“”，建筑初始添加宾语名字。

inaddoffset: 当 isinadd 为是时可选，数字数组，默认为“0 0”，建筑初始添加宾语偏移。

inaddoffsetsize: 当 isinadd 为是时可选，数字数组，默认为“0 0”，建筑初始添加宾语大小改变。

inaddisinitialunit: 可选，默认为否。启用时，建筑刷新将使用“unit”，而不是“spawnUnits”。除去“unit”和“team”以外的选项将不会出现。并将该宾语加入 unitObject 层。请确保 unitObject 层已经设置。

mtext_info

建筑文本显示，是附属宾语，且无参数。

mtext_info 必选参数

mtext: 当 ismtext 为是时必选，表示城市文本的内容。

mtext_info 可选参数

ismtext: 可选，默认为是，在 building_info/idcheck_info 中默认为否，是否启用该宾语。

mcolor: 当 ismtext 为是时可选，表示城市文本的颜色。

mtextsize: 当 ismtext 为是时可选，表示城市文本的字号。

mname: 当 ismtext 为是时可选，默认为“”，文本宾语名字。

moffset: 当 ismtext 为是时可选，数字数组，默认为“0 0”，文本宾语偏移。

moffsetsize: 当 ismtext 为是时可选，数字数组，默认为“0 0”，文本宾语大小改变。

building_info

这是一个建筑添加 info 宾语。可以生成自动刷新建筑的宾语。可以添加初始城市 (inadd)，确定城市初始阵营，附属宾语为 inadd_info。城市可以提供文本生成 (mtext)，显示城市名称，附属宾语为 mtext_info。因此，inadd_info 和 mtext_info 的所有必填参数和选填参数都在 building_info 中存在。可进行时间修正。

building_info 必填参数

prefix: 标记宾语引用前缀。([info 宾语默认参数](#))

idprefix: 表示所使用的 id 前缀，城市 id 在 idprefix 后将自动按照 1,2...顺序延伸，请确保其他 id 没有此前缀。

detectReset: 建筑检测的 resetActivationAfter。

auunit: 建筑单位类型。

building_info 可选参数

isprefixseg: 可选，默认为否。(info 宾语默认参数)

args: 可选，默认没有，必填参数。(info 宾语默认参数)

opargs: 可选，默认没有，可选参数。(info 宾语默认参数)

cite_name: 可选，默认没有，标志宾语标记。(info 宾语默认参数)

brace: 可选，默认没有，外部引用翻译列表。(info 宾语默认参数)

addWarmup: 可选，建筑添加的 warmup。为 0s 或 addWarmup 不存在时，不产生 warmup 属性。

addReset: 可选，建筑添加的 resetActivationAfter。

spawnnum: 可选，默认为 1，建筑添加的数量。

minUnits: 可选，默认没有，建筑检测的 minUnits。

maxUnits: 可选，默认没有，建筑检测的 maxUnits。

team: 可选，默认为-1，建筑添加的队伍。为-1 时，建筑检测没有 team。不为-1 时，建筑检测有 team。

isonlybuilding: 可选，默认为否，是否启用 onlybuilding 来检测建筑，如果启用，则不使用 unitType 检测。

isshowOnMap: 可选，默认为否，启用时，建筑生成时小地图显示。

acti: 可选，字符串数组，默认没有。建筑添加宾语添加的额外 activatedBy。

deacti: 可选，字符串数组，默认没有。建筑添加宾语添加的额外 deactivatedBy。

isdetectdeacti: 检测宾语是否抑制建筑添加宾语。

auunitbrace: 可选，字符串，默认为" "。将在建筑添加后面添加内容 (而不在建筑检测添加)。例如" (techLevel=2)"

addname: 可选，默认为" "，建筑添加宾语名字。

addoffset: 可选，数字数组，默认为" 0 0"，建筑添加宾语偏移。

addoffsetsize: 可选，数字数组，默认为" 0 0"，建筑添加宾语大小改变。

detectname: 可选，默认为" "，建筑检测宾语名字。

detectoffset: 可选，数字数组，默认为" -10 0"，建筑检测宾语偏移。

detectoffsetsize: 可选，数字数组，默认为" 20 0"，建筑检测宾语大小改变。

inadd_prefix: 可选，默认不存在。将会导入对应 inadd_info 的数据，建立初始建筑。

mtext_prefix: 可选，默认不存在。将会导入对应 mtext_info 的数据，生成建筑名字。

time_prefix: 可选，默认不存在。将会导入对应 time_info 的数据，进行时间修正。

building_info 可被引用的其他参数

idprefix0 是建筑检测的 id。

teamDetect_info

这是一个队伍检测 info 宾语。可以生成许多检测不同队伍单位宾语。

teamDetect 必选参数

prefix: 标记宾语引用前缀。(info 宾语默认参数)

reset: 必选，表示检测宾语的刷新周期。

setTeam: 数字二维数组，表示队伍分组。同阵营内部用空格隔开，阵营之间用逗号隔开。如果为-3，表示检测与本检测 min 和 max 互补的检测，且无队伍。填写例子“ 0 2 4 6 8 10 12 14 16 18,1 3 5 7 9 11 13 15 17,-3 -2 -1”。不允许小于等于-4。

setidTeam: 字符串二维数组，表示不同组的 id 前缀。不同组之间用“；”隔开，一个阵营可以有一个或多个 id。每一个组的第一个 id 相同时，表明不同组是一个阵营。填写例子“ A_city,B_city”。

teamDetect 可选参数

isprefixseg: 可选，默认为否。(info 宾语默认参数)

args: 可选，默认没有，必填参数。(info 宾语默认参数)

opargs: 可选，默认没有，可选参数。(info 宾语默认参数)

cite_name: 可选，默认没有，标志宾语标记。(info 宾语默认参数)

brace: 可选，默认没有，外部引用翻译列表。(info 宾语默认参数)

aunit: 可选，检测单位类型，用于 unitType。

minUnits: 可选，默认为 1，建筑检测的 minUnits。如果为 0，那么没有 minUnits。不允许小于 0。

maxUnits: 可选，默认没有该参数，建筑检测的 maxUnits。不允许小于 0。必须大于等于 minUnits。

name: 可选，字符串数组，默认没有，表示不同阵营宾语的名称。每个阵营仅会显示一个，不同阵营的名称之间用逗号隔开。填写例子“ setidTeam0_0,setidTeam1_0”。

offset: 可选, 数字二维数组, 默认" 0 0", 表示不同阵营宾语偏移。当只有一个坐标时, 该偏移对所有阵营均适用。不同阵营之间用逗号隔开, 一个阵营有 x 和 y 两个坐标, 中间空格隔开。填写例子" 0 -20,-20 0"。

offsetsize: 可选, 数字二维数组, 默认" 0 0", 表示不同阵营宾语大小改变。当只有一个坐标时, 该偏移对所有阵营均适用。不同阵营之间用逗号隔开, 一个阵营有 x 和 y 两个坐标, 中间空格隔开。填写例子" 0 40,40 0"。

neutralindex: 可选, 字符串, 默认-1, 最后一组 id 和 team。一个数字, 对引用 setidTeam_id_dep 和 team-toid_dep 有影响。

basicoffset: 可选, 字符串数组, 默认" -10 10", 表示 basic 宾语偏移 (setidTeam 中同一组出现不止一个)。

basicoffsetsize: 可选, 字符串数组, 默认" 20 0", 表示 basic 宾语大小偏移 (setidTeam 中同一组出现不止一个)。

此外, 还有大量 only, 将会原样添加到检测宾语中(如果有的话)。它们是: onlyIdle, onlyBuildings, onlyMainBuildings, onlyEmptyQu

teamDetect_info 可被引用的其他参数

setidTeam{i}_{j}_0 是生成的 id。i 是第几组, j 是该组内的第几个 id。只有 j = 0 的 id 前缀对之后的变量有用 (每组的第一个 id)。

setidTeam_id 是生成 id 的列表。

setidTeam_id_dep 是生成对应 id 补集 (不包括 neutralindex 位置) 的列表。

setidTeam_id_depn 是生成对应 id 补集的列表。

teamtoi 是队伍到 id 索引的字典。

teamtoid 是队伍到 id 的字典。

teamtoid_dep 是队伍到对应 id 的补集 (不包括 neutralindex 位置) 的字典。

teamtoid_depn 是队伍到对应 id 的补集的字典。

numDetect_info

这是一个多个单位检测 info 宾语。可以生成许多检测不同数量单位的宾语。

numDetect 必选参数

prefix: 标记宾语引用前缀。 ([info 宾语默认参数](#))

reset: 必选, 表示检测宾语的刷新周期。

team: 必选, 表示检测宾语的检测队伍

setNum: 数字二维数组, 表示不同数量检测的分组。使用相同 id 的检测内部用空格隔开 (一组之内, 两两分开, 第一个做 minUnits, 第二个做 maxUnits), 使用不同 id 的检测之间用逗号隔开。填写例子" 0 2 6 8,3 5" (检测到 0-2,6-8 个单位激活第一个 id, 检测到 3-5 个单位激活第二个 id) (如果 minUnits 小于等于 0, 则不添加, 如果 maxUnits 大于等于 65536, 则不添加)。

setidNum: 字符串数组, 表示不同数量检测的 id 前缀。不同检测 id 之间用逗号隔开, 一个阵营仅有一个 id。填写例子" A_city,B_city"。

numDetect 可选参数

isprefixseg: 可选, 默认为否。([info 宾语默认参数](#))

args: 可选, 默认没有, 必填参数。([info 宾语默认参数](#))

opargs: 可选, 默认没有, 可选参数。([info 宾语默认参数](#))

cite_name: 可选, 默认没有, 标志宾语标记。([info 宾语默认参数](#))

brace: 可选, 默认没有, 外部引用翻译列表。([info 宾语默认参数](#))

aunit: 可选, 检测单位类型, 用于 unitType。

name: 可选, 字符串数组, 默认为" ", 表示不同组宾语的名称。每个组仅会显示一个, 不同组的名称之间用逗号隔开。

offset: 可选, 数字二维数组, 默认" 0 0", 表示不同组宾语偏移。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子" 0 -20,-20 0"。

offsetsize: 可选, 数字二维数组, 默认" 0 0", 表示不同组宾语大小改变。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子" 0 40,40 0"。

此外, 还有大量 only, 将会原样添加到检测宾语中(如果有的话)。它们是: onlyIdle, onlyBuildings, onlyMainBuildings, onlyEmptyQu

numDetect_info 可被引用的其他参数

setidNum0_0,setidNum1_0...是生成的 id。

multiText_info

这是一个产生多个文本的 info 宾语。会根据规则自动产生一系列文本宾语。

multiText 必选参数

prefix: 标记宾语引用前缀。(info 宾语默认参数)

text: 必选, 字符串数组。文本宾语文本内容。如果仅有一个, 那么该 text 将会应用于所有产生文本宾语中。text 将尽力全部加入文本宾语组中。

multiText 可选参数

isprefixseg: 可选, 默认为否。(info 宾语默认参数)

args: 可选, 默认没有, 必填参数。(info 宾语默认参数)

opargs: 可选, 默认没有, 可选参数。(info 宾语默认参数)

brace: 可选, 默认没有, 外部引用翻译列表。(info 宾语默认参数)

reset: 可选, 默认为 1s, 表示文本宾语的刷新周期。

acti: 可选, 字符串二维数组, 默认没有。文本宾语的激活来源被";"分隔, 将会添加进入。文本宾语数量保证将 acti 均加入。

deacti: 可选, 字符串二维数组, 默认没有。文本宾语的抑制来源被";"分隔, 将会添加进入。文本宾语数量保证将 deacti 均加入。

teamDetect_cite: 可选, 默认没有, teamDetect 标记宾语引用。将会将 teamDetect 的 id 逐个加入激活中。如果 isdefaultText 选择, 将继续所有 teamDetect 的 id 加入下一个的抑制中。文本宾语数量保证 teamDetect_cite 充分发挥作用。

isdefaultText: 可选, 默认为否, teamDetect_cite 是否包含默认文本。具体见上文 teamDetect_cite。

numDetect_cite: 可选, 默认没有, numDetect 标记宾语引用。将会将 numDetect 的 id 逐个加入激活中。文本宾语数量保证 numDetect_cite 充分发挥作用。

textsize: 可选, 字符串数组, 默认没有。文本宾语文本的大小。如果仅有一个, 那么该 textsize 将会应用于所有产生文本宾语中。textsize 将尽力全部加入文本宾语组中。

color: 可选, 字符串数组, 默认没有。文本宾语文本的颜色。如果仅有一个, 那么该 color 将会应用于所有产生文本宾语中。color 将尽力全部加入文本宾语组中。

name: 可选, 字符串数组, 默认为"", 表示不同阵营宾语的名称。每个阵营仅会显示一个, 不同阵营的名称之间用逗号隔开。

offset: 可选, 数字二维数组, 默认"0 0", 表示不同组宾语偏移。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子"0 -20,-20 0"。

offsetsize: 可选, 数字二维数组, 默认"0 0", 表示不同组宾语大小改变。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子"0 40,40 0"。

multiRemove_info

这是一个产生多个删除宾语的 info 宾语。类似于 multiText。可进行时间修正。

multiRemove 必选参数

prefix: 标记宾语引用前缀。([info 宾语默认参数](#))

multiRemove 可选参数

isprefixseg: 可选，默认为否。([info 宾语默认参数](#))

args: 可选，默认没有，必填参数。([info 宾语默认参数](#))

opargs: 可选，默认没有，可选参数。([info 宾语默认参数](#))

brace: 可选，默认没有，外部引用翻译列表。([info 宾语默认参数](#))

acti: 可选，字符串二维数组，默认没有。删除宾语的激活来源被”；“分隔，将会添加进入。删除宾语数量保证将 acti 均加入。

deacti: 可选，字符串二维数组，默认没有。删除宾语的抑制来源被”；“分隔，将会添加进入。删除宾语数量保证将 deacti 均加入。

teamDetect_cite: 可选，默认没有，teamDetect 标记宾语引用。将会将 teamDetect 的 id 按照 Remove 的 team 加入激活中。如果没有 team，或者 team 不在 teamDetect 中出现，则不加入 id。

numDetect_cite: 可选，默认没有，numDetect 标记宾语引用。将会将 numDetect 的 id 逐个加入激活中。删除宾语数量保证 numDetect_cite 充分发挥作用。

team: 可选，字符串数组，默认没有。删除宾语队伍。如果仅有一个，那么 team 将会应用于所有删除宾语中。删除宾语数量保证 team 充分发挥作用。

warmup: 可选，字符串数组，默认没有。删除宾语的 warmup。如果仅有一个，那么 warmup 将会应用于所有删除宾语中。warmup 将尽力全部加入删除宾语组中。

reset: 可选，字符串数组，默认没有。删除宾语的 resetActivationAfter。如果仅有一个，那么该 reset 将会应用于所有删除宾语中。reset 将尽力全部加入删除宾语组中。

delay: 可选，字符串数组，默认没有。删除宾语的 delay。如果仅有一个，那么 delay 将会应用于所有删除宾语中。delay 将尽力全部加入删除宾语组中。

repeat: 可选，字符串数组，默认没有。删除宾语的 repeatDelay。如果仅有一个，那么该 repeat 将会应用于所有删除宾语中。repeat 将尽力全部加入删除宾语组中。

name: 可选, 字符串数组, 默认为“ ”, 表示不同组宾语的名称。每个组宾语仅会显示一个, 不同组宾语的名称之间用逗号隔开。

offset: 可选, 数字二维数组, 默认“ 0 0”, 表示不同组宾语偏移。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子“ 0 -20,-20 0”。

offsetsize: 可选, 数字二维数组, 默认“ 0 0”, 表示不同组宾语大小改变。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子“ 0 40,40 0”。

time_prefix: 可选, 默认不存在。将会导入对应 time_info 的数据, 进行时间修正。

multiAdd_info

这是一个产生多个单位添加宾语的 info 宾语。类似于 multiText。可进行时间修正。

multiAdd 必选参数

prefix: 标记宾语引用前缀。([info 宾语默认参数](#))

spawnUnits: 必选, 字符串二维数组, 默认没有。添加宾语的单位。不同添加宾语的 spawnUnits 被“ ”隔开。如果仅有一个, 那么 spawnUnits 将会应用于所有添加宾语中。添加宾语数量保证 spawnUnits 充分发挥作用。

team: 可选, 字符串数组, 默认没有。添加宾语队伍。如果仅有一个, 那么 team 将会应用于所有添加宾语中。添加宾语数量保证 team 充分发挥作用。

multiAdd 可选参数

isprefixseg: 可选, 默认为否。([info 宾语默认参数](#))

args: 可选, 默认没有, 必填参数。([info 宾语默认参数](#))

opargs: 可选, 默认没有, 可选参数。([info 宾语默认参数](#))

brace: 可选, 默认没有, 外部引用翻译列表。([info 宾语默认参数](#))

acti: 可选, 字符串二维数组, 默认没有。添加宾语的激活来源被“ ”分隔, 将会添加进入。添加宾语数量保证将 acti 均加入。

deacti: 可选, 字符串二维数组, 默认没有。添加宾语的抑制来源被“ ”分隔, 将会添加进入。添加宾语数量保证将 deacti 均加入。

teamDetect_cite: 可选, 默认没有, teamDetect 标记宾语引用。将会将 teamDetect 的 id 按照 Add 的 team 加入激活中。如果没有 team, 或者 team 不在 teamDetect 中出现, 则不加入 id。

numDetect_cite: 可选, 默认没有, numDetect 标记宾语引用。将会将 numDetect 的 id 逐个加入激活中。添加宾语数量保证 numDetect_cite 充分发挥作用。

warmup: 可选, 字符串数组, 默认没有。添加宾语的 warmup。如果仅有一个, 那么 warmup 将会应用于所有添加宾语中。warmup 将尽力全部加入添加宾语组中。

reset: 可选, 字符串数组, 默认没有。添加宾语的 resetActivationAfter。如果仅有一个, 那么该 reset 将会应用于所有添加宾语中。reset 将尽力全部加入添加宾语组中。

delay: 可选, 字符串数组, 默认没有。添加宾语的 delay。如果仅有一个, 那么 delay 将会应用于所有添加宾语中。delay 将尽力全部加入添加宾语组中。

repeat: 可选, 字符串数组, 默认没有。添加宾语的 repeatDelay。如果仅有一个, 那么该 repeat 将会应用于所有添加宾语中。repeat 将尽力全部加入添加宾语组中。

name: 可选, 字符串数组, 默认为“ ”, 表示不同组宾语的名称。每个组宾语仅会显示一个, 不同组宾语的名称之间用逗号隔开。

offset: 可选, 数字二维数组, 默认“ 0 0”, 表示不同组宾语偏移。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子“ 0 -20,-20 0”。

offsetsize: 可选, 数字二维数组, 默认“ 0 0”, 表示不同组宾语大小改变。当只有一个坐标时, 该偏移对所有组均适用。不同组之间用逗号隔开, 一个组有 x 和 y 两个坐标, 中间空格隔开。填写例子“ 0 40,40 0”。

time_prefix: 可选, 默认不存在。将会导入对应 time_info 的数据, 进行时间修正。

flash_info

这是一个振荡器, 为了解决 unitAdd 在 deactivatedBy 由激活到不激活时, 异常触发刷兵的 bug。该振荡器需要一片空间, 可以作为控制台一部分。具体来说, 将振荡器检测 id 加入 unitAdd 的 deactivatedBy 中, 就能在振荡器激活变为非激活时, 直接产生刷新。不得将振荡器放置在玩家可以接触到的区域。

具体来说, 如果 initialtime: “5s,15s” , periodtime: “5s,20s” 。那么, 检测 id 激活停止将会在 5s,15s,20s,35s,40s,45s...无限完成。

可进行时间修正。

flash_info 必选参数

prefix: 标记宾语引用前缀。([info 宾语默认参数](#))

idprefix: 表示振荡器输出的 id 前缀, 检测 id 在 idprefix 后将自动按照 1,2...顺序延伸, 请确保其他 id 没有此前缀。

initialtime: 必选, 字符串数组。检测 id 停止激活的若干初始时间。时间应当不断增加, 最后一个 initial 为周期开始的时间。

flash_info 可选参数

isprefixseg: 可选，默认为否。(info 宾语默认参数)

args: 可选，默认没有，必填参数。(info 宾语默认参数)

opargs: 可选，默认没有，可选参数。(info 宾语默认参数)

cite_name: 可选，默认没有，标志宾语标记。(info 宾语默认参数)

brace: 可选，默认没有，外部引用翻译列表。(info 宾语默认参数)

detectReset: 可选，默认为“0.25s”，振荡器检测的 resetActivationAfter。

periodtime: 可选，字符串数组，默认没有。检测 id 停止激活的若干周期时间（在周期中停止激活的相位）。最后一个 periodtime 为周期时间。

initialacti: 可选，二维字符串数组。初始添加的 acti。不同组被“；”分割。如果仅有一个，那么 acti 将会应用于所有添加宾语中。acti 将尽力全部加入添加宾语组中。

initialdeacti: 可选，二维字符串数组。初始添加的 deacti。不同组被“；”分割。如果仅有一个，那么 ddeacti 将会应用于所有添加宾语中。deacti 将尽力全部加入添加宾语组中。

periodacti: 可选，二维字符串数组。周期添加的 acti。不同组被“；”分割。如果仅有一个，那么 acti 将会应用于所有添加宾语中。acti 将尽力全部加入添加宾语组中。

perioddeacti: 可选，二维字符串数组。周期添加的 deacti。不同组被“；”分割。如果仅有一个，那么 ddeacti 将会应用于所有添加宾语中。deacti 将尽力全部加入添加宾语组中。

time_prefix: 可选，默认不存在。将会导入对应 time_info 的数据，进行时间修正。

flash_info 可被引用的其他参数

idprefix0 是振荡器输出的 id。

step_info

这是一个方波器。可进行时间修正。

step_info 必选参数

prefix: 标记宾语引用前缀。(info 宾语默认参数)

idprefix: 表示器输出的 id 前缀，城市 id 在 idprefix 后将自动按照 1,2...顺序延伸，请确保其他 id 没有此前缀。

steptime: 必选，字符串数组。添加宾语和删除宾语作用的时间。开头应当为 0s。以避免出现 bug。

iaactiend: 必选。当 isactiend 为 true 时，检测 id 最后为激活状态。之前每经过一个 steptime，激活状态都会改变。

step_info 可选参数

isprefixseg: 可选，默认为否。(info 宾语默认参数)

args: 可选，默认没有，必填参数。(info 宾语默认参数)

opargs: 可选，默认没有，可选参数。(info 宾语默认参数)

cite_name: 可选，默认没有，标志宾语标记。(info 宾语默认参数)

brace: 可选，默认没有，外部引用翻译列表。(info 宾语默认参数)

detectReset: 可选，默认为“ 0.25s”，振荡器检测的 resetActivationAfter。

aunit: 可选，建筑单位类型，默认为高射炮“ antiAirTurretFlak”。

spawnnum: 可选，默认为 1，添加的数量。

team: 可选，默认为-2，添加的队伍。

stepacti: 可选，二维字符串数组。初始添加的 acti。不同组被“；”分割。如果仅有一个，那么 acti 将会应用于所有改变宾语中。acti 将尽力全部加入宾语组中。

stepdeacti: 可选，二维字符串数组。初始添加的 deacti。不同组被“；”分割。如果仅有一个，那么 ddeacti 将会应用于所有改变宾语中。deacti 将尽力全部加入宾语组中。

time_prefix: 可选，默认不存在。将会导入对应 time_info 的数据，进行时间修正。

step_info 可被引用的其他参数

idprefix0 是振荡器输出的 id。

idcheck_info

这是一个 id 探测 info 宾语。可以根据 id 是否激活决定原地是否会刷新建筑。可以提供文本生成 (mtext)，显示名称，附属宾语为 mtext_info。因此，mtext_info 的所有必填参数和选填参数都在 idcheck_info 中存在。可以进行时间修正。

idcheck_info 必填参数

prefix: 标记宾语引用前缀。([info 宾语默认参数](#))

idprefix: 表示所使用的 id 前缀, 城市 id 在 idprefix 后将自动按照 1,2,...顺序延伸, 请确保其他 id 没有此前缀。

otherid: 检测的 id。

idcheck_info 可选参数

isprefixseg: 可选, 默认为否。([info 宾语默认参数](#))

args: 可选, 默认没有, 必填参数。([info 宾语默认参数](#))

opargs: 可选, 默认没有, 可选参数。([info 宾语默认参数](#))

cite_name: 可选, 默认没有, 标志宾语标记。([info 宾语默认参数](#))

brace: 可选, 默认没有, 外部引用翻译列表。([info 宾语默认参数](#))

addWarmup: 可选, 添加的 warmup。

addReset: 可选, 添加的 resetActivationAfter, 默认 0.25s。

detectReset: 可选, 检测的 resetActivationAfter, 默认 0.25s。

removeReset: 可选, 删除的 resetActivationAfter, 默认 0.25s。

aunit: 可选, 建筑单位类型, 默认为高射炮" antiAirTurretFlak"。

spawnnum: 可选, 默认为 1, 添加的数量。

team: 可选, 默认为-2, 添加的队伍。

addname: 可选, 默认为" ", 添加宾语名字。

detectname: 可选, 默认为" ", 检测宾语名字。

removename: 可选, 默认为" ", 删除宾语名字。

offset: 可选, 数字数组, 默认为" -10 0", 宾语偏移。

offsetsize: 可选, 数字数组, 默认为" 20 0", 宾语大小改变。

mtext_prefix: 可选, 默认不存在。将会导入对应 mtext_info 的数据, 生成建筑名字。

time_prefix: 可选, 默认不存在。将会导入对应 time_info 的数据, 进行时间修正。

idcheck_info 可被引用的其他参数

idprefix0 是建筑检测的 id。

time_info

这是一个附属 info，如果其他 info 引用了该 time_info，可以修改其他宾语的相关时间变量。

time_info 必填参数

prefix: 其他 info 宾语引用前缀。([info 宾语默认参数](#))

timeratio: 时间修正乘数。默认会对所在 info 的所有 resetActivationAfter 和 repeatDelay 相关的时间进行修正。

time_info 可选参数

istime: 可选，默认为是，在其他引用的 info 中默认为否，是否进行时间修正。

isincorrectwarmup: 可选，默认为否。如果为是，将会把该 info 的 warmup 和 delay 的时间也进行修正。

使用

命令

打开终端，输入如下指令：

```
1 objectgroupauto {map_file}
```

该指令读入地图路径，将转换结果自动输到原路径。

命令参数

```
1 -o --output 输出到不同路径（建议，防止覆盖后无法悔改）。
2
3 -r --reset 选择进行id的重置。不添加-r命令可以保证已有的id不会改变(从其他地方使用id不会断线错位)。
4
5 -d(--delete) -D(--DeleteAllSym) --DeleteAll:
6
7 默认情况下，带,d的info宾语或者标记宾语自动产生的宾语将会被删除（并且不产生），",d"后缀删除后仍然可以重新产生宾语。[正常使用]
8
9 -d --delete 命令会同时把带,d的info宾语或者标记宾语(也包括其city_info带,d)删除，删除标记没有反悔机会。[删除决定去掉的标记宾语/info宾语]
10
```

```
11 -D --DeleteAllSym命令在-d命令的基础上，将所有info宾语和标记宾语删除，只
    留下使用宾语。[发布地图]
12
13 --DeleteAll 命令会将所有info宾语、标记宾语和自动产生的宾语均删除，彻底
    除去使用痕迹。[取消使用该宾语格式]
14
15 --resetid 命令将会将宾语ID重置为自然数列。应当添加，否则ID空洞会越来越
    多，尤其是同一地图频繁进行自动化操作。不添加对实际地图没有影响。
16
17 -v --verbose 显示运行信息
18
19 -c --citytrans表明除了info宾语，标记宾语，以及标记宾语产生的宾语，其他
    宾语也可以使用引用来获得参数。
20
21 --ignorewarning 发生warning时，会继续执行，而不会退出程序。
22
23 -y --isyes 自动同意程序的一切(y/n)请求
24
25 --check 不忽略程序中的一些检测，但是会对效率产生不利影响。
26
27 --language 后仅允许跟"ch"(中文)/"eg"(英文)。语言设置将会被储存，之后使
    用延续上一次的修改。
28
29 --infopath命令约定了.py文件路径，--infovar命令约定了变量名，随后程序会
    根据该约定变量来自动转换。
30 默认会使用command/auto/_data中的auto_func_arg作为约定变量。可以自行修改
    约定变量来个性化自动转换模式。infopath中必须是一个python包。内部必须
    有一个config.json文件。
```

约定变量

宾语自动化约定变量参数说明

其他特性

会检测 info 宾语和标记宾语的参数正确性，错误时返回错误信息。

会自动将 info 和标记宾语放到文件最后，保证在 Tiled 中 info 和标记宾语可以直接被点击到。

注意事项

所有时间单位 时间必须以 s 结尾。时间不得为负数。时间列表间隔应当大于等于 0.25s，否则相当于一个时间。初始时间不得为 0s，至少为 0.25s。

team 设置 和铁锈一样，team 设置是从 0 开始的，-1 中立，-2 敌对。

造成混淆的宾语 所有 type 一栏空着的宾语，但与 info 不产生联系。

自动化后的地图处理 如果要清除一组宾语，不得擅自删除。请在标志宾语后面添加,d（暂时清除，不产生新宾语）或者,D（无论如何都会删除）。再运行自动化程序。不经过以上步骤的删除，可能造成不良后果。

标记问题 可能与关键词发生冲突，冲突时会有很多种形式，目前尚未严格处理。可以换一个标记。非常推荐使用现成的例子，例子是经过检验的。

字符问题 引用字符串仅支持英文字符和中文字符，其他语言不支持。

unit/team 铁锈检测要求，凡是出现 unit 选项的宾语必须有 team 选项。