

AI Instruction Summary

Note: Only the Rate control-table differentiates between the two races. In the control program and build-table it is up to your taste what variable names you use (for example `footmen` and `grunts` compile to the same value). For the sake of clarity I use the human names in the descriptions.

For some variables there are also other spellings allowed. Look at the file `aidefs.inc` in the WarDraft installation directory for those.

Control program variables:

Legal values are 0-255.

<code>Size_land</code>		set size of land attack group(s)
<code>size_water</code>		set size of naval attack group(s)
<code>size_air</code>		set size of air attack group(s)
<code>ptys_land</code>		set number of land attack parties
<code>ptys_water</code>		set number of naval attack parties
<code>ptys_air</code>		set number of air attack parties
<code>peasants</code>	<code>peons</code>	set number of wanted peasants
<code>footmen</code>	<code>grunts</code>	set number of wanted footmen
<code>archers</code>	<code>axethrowers</code>	set number of wanted archers
<code>ballistas</code>	<code>catapults</code>	set number of wanted ballistas
<code>knights</code>	<code>ogres</code>	set number of wanted knights
<code>tankers</code>		set number of wanted tankers
<code>destroyers</code>		set number of wanted destroyers
<code>transports</code>		set number of wanted transports
<code>battleships</code>	<code>juggernaughts</code>	set number of wanted battleships
<code>submarines</code>	<code>turtles</code>	set number of wanted submarines
<code>mages</code>	<code>deathknights</code>	set number of wanted mages
<code>flyers</code>	<code>zeppelins</code>	set number of wanted flying machines
<code>demosquads</code>	<code>sappers</code>	set number of wanted demolition squads
<code>gryphons</code>	<code>dragons</code>	set number of wanted gryphon riders

Control program commands

<code>reset_land</code>		no land attacks
<code>reset_water</code>		no naval attacks
<code>reset_air</code>		no air attacks
<code>attk_land</code>		start land attack
<code>attk_water</code>		start naval attack
<code>attk_air</code>		start air attack
<code>strat_off</code>		? (unknown)
<code>strat_on</code>		? (unknown)
<code>aggr_low</code>		? (unknown)
<code>aggr_high</code>		? (unknown)
<code>wait_keep</code>	<code>wait_stronghold</code>	wait until you have at least 1 keep
<code>wait_castle</code>	<code>wait_fortress</code>	wait until you have at least 1 castle
<code>wait_worker</code>	<code>wait_peasants</code> <code>wait_peons</code>	wait until number of wanted peasants are trained
<code>wait_land</code>		wait until land attack groups(s) are complete
<code>wait_water</code>		wait until naval attack groups(s) are complete
<code>wait_air</code>		wait until air attack groups(s) are complete
<code>wait_7</code>		? (unknown)
<code>sleep <time></code>		suspend program for <time> ticks
<code>goto <label></code>		jump to <label>. Should be last command.
<code>build <item></code>		builds the specified <item>. See below.
<code>upgrade</code> <code>upgrd</code> <code>learn</code> <code>make</code> <code>do <item></code>		same as build

Items

These are the names you can use in conjunction with a `build` command (or `item-assignment`). Again, these are race-independent, so for example `stables` and `ogremound` compile to the same value.

<code>farm</code>		Farm
<code>barracks</code>		Barracks
<code>church</code>	<code>altar</code>	Church
<code>tower</code>		Scout Tower
<code>stables</code>	<code>ogremound</code>	Stables
<code>inventor</code>	<code>alchemist</code>	Gnomish Inventor
<code>aviary</code>	<code>roost</code>	Gryphon Aviary
<code>shipyard</code>		Shipyard
<code>townhall</code>	<code>greathall</code>	Town Hall
<code>lumbermill</code>		Lumber mill
<code>foundry</code>		Foundry
<code>magetower</code>	<code>temple</code>	Mage Tower
<code>blacksmith</code>		Blacksmith
<code>refinery</code>		Refinery
<code>oilwell</code>		Oil well
<code>up_arrow_1</code>	<code>up_axe_1</code>	Arrow upgrade (1/2)
<code>up_arrow_2</code>	<code>up_axe_2</code>	Arrow upgrade (2/2)
<code>up_rangers</code>	<code>up_berserkers</code>	upgrade Archers to Rangers
<code>up_rangers_A</code>	<code>up_berserkers_A</code>	Ranger upgrade A
<code>up_rangers_B</code>	<code>up_berserkers_B</code>	Ranger upgrade B
<code>up_rangers_C</code>	<code>up_berserkers_C</code>	Ranger upgrade C
<code>up_footmen_A1</code>	<code>up_grunts_A1</code>	Footman, Knight upgrade A (1/2)
<code>up_footmen_A2</code>	<code>up_grunts_A2</code>	Footman, Knight upgrade A (2/2)
<code>up_footmen_B1</code>	<code>up_grunts_B1</code>	Footman, Knight upgrade B (1/2)
<code>up_footmen_B2</code>	<code>up_grunts_B2</code>	Footman, Knight upgrade B (2/2)
<code>up_ballistas_1</code>	<code>up_catapults_1</code>	Ballista upgrade (1/2)
<code>up_ballistas_2</code>	<code>up_catapults_2</code>	Ballista upgrade (1/2)
<code>up_shipcannons_1</code>		Ship cannons upgrade (1/2)
<code>up_shipcannons_2</code>		Ship cannons upgrade (2/2)
<code>up_shiparmor_1</code>		Ship armor upgrade (1/2)
<code>up_shiparmor_2</code>		Ship armor upgrade (2/2)
<code>up_paladins</code>	<code>up_ogremages</code>	upgrade Knights to Paladins
<code>sp_paladin_A</code>	<code>sp_ogremage_A</code>	Paladin spell A
<code>sp_paladin_B</code>	<code>sp_ogremage_B</code>	Paladin spell B
<code>sp_mage_A</code>	<code>sp_deathknight_A</code>	Mage spell A
<code>sp_mage_B</code>	<code>sp_deathknight_B</code>	Mage spell B
<code>sp_mage_C</code>	<code>sp_deathknight_C</code>	Mage spell C
<code>sp_mage_D</code>	<code>sp_deathknight_D</code>	Mage spell D
<code>sp_mage_E</code>	<code>sp_deathknight_E</code>	Mage spell E
<code>up_keep</code>	<code>up_stronghold</code>	upgrade Townhall to Keep
<code>up_castle</code>	<code>up_fortress</code>	upgrade Keep to Castle

Rates

These are the build-rates for the rate control-table. The rate control table is introduced with the keyword **rates**:. If you don't specify a rate the compiler will set it to a default of 100 ticks.

Note that the rate control-table is **race-dependent**. So if you don't set the Orcish and human rates to the same values, the computer behaves differently when controlling either Orcs or Humans.

<code>rate 0</code>	? (always disabled)
<code>rate 1</code>	? (always disabled)
<code>rt_Hbarracksfolk</code>	Hum: Footmen, Archers, Knights, Ballistas
<code>rt_Obarracksfolk</code>	Orc: Grunts, Axethrowers, Ogres, Catapults
<code>rt_paladins</code>	Hum: Knight to Paladin upgrade, all Paladin spells
<code>rt_ogremages</code>	Orc: Ogre to Ogre-Mage upgrade, all Ogre-Mage spells
<code>rt_Hguardtower</code>	Hum: Scout Tower to Guard Tower upgrade

rt_Oguardtower	Orc: Scout Tower to Guard Tower upgrade
rate 8	?
rate 9	?
rt_flyers	Hum: Flying Machines
rt_zeppelins	Orc: Zeppelins
rt_gryphons	Hum: Gryphon Riders
rt_dragons	Orc: Dragons
rt_Hships	Hum: Ships
rt_Oships	Orc: Ships
rt_peasants	Hum: Peasants
rt_peons	Orc: Peons
rt_up_archers	Hum: all Arrow and Ranger upgrades
rt_up_axethrowers	Orc: all Axe and Berserker upgrades
rt_Hup_shipcannons	Hum: all Ship Cannons, Ship Armor upgrades
rt_Oup_shipcannons	Orc: all Ship Cannons, Ship Armor upgrades
rt_mages	Hum: Mages, all Mage spells
rt_deathknights	Orc: Death Knights, all Death Knight spells
rt_Hup_weapons	Hum: all Sword, Shield and Ballista upgrades
rt_Oup_weapons	Orc: all Sword, Shield and Catapult upgrades
rate 26	? (always disabled)
rate 27	? (always disabled)
rate 28	? (always disabled)
rate 29	? (always disabled)
rt_up_castle	Hum: Keep to Castle upgrade
rt_up_fortress	Orc: Stronghold to Fortress upgrade
rt_castlepeasants	Hum: "Castle-Peasants"
rt_fortresspeons	Orc: "Fortress-Peons"
rate 34	? (always disabled)

The Item-table

This usually not used. Use *build* commands instead.

If you prefer it complicated however, you can also use an item-table and *maxitem*-assignments. Don't use any *build* commands if you do this.

The Item-table is introduced by the keyword **Items**:. There is only one command that can be used:

```
item <xx> = <item>
```

where <xx> is the number and <item> is an item-variable (see above). You should pay attention that the lowest number is 1 and that there are no "holes" in numbers.

Since the AI Editor will never create item-tables when auto-creating source-code from an existing AI, I'll give a short example here. We will build a townhall, lumbermill, blacksmith and barracks, train some units and then just do nothing:

```
#include aidefs.inc

Program:
    maxitem = 0          ; build nothing
    ...                ; standard initialisation block
    peons = 5           ; we want 5 Peons
    maxitem = 4         ; build the first 4 items (in this example this is all we have)
    grunts = 3         ; want 3 Grunts
    axethrowers = 2    ; and 2 Axethrowers

Loop:
    sleep 1000
    goto Loop

Items:                  ; here we define the 4 items
    item 1 = townhall
    item 2 = lumbermill
    item 3 = blacksmith
    item 4 = barracks

Rates:                  ; let's make them train fast
    rt_Hbarracksfolk = 1
    rt_Obarracksfolk = 1
    rt_peasants      = 1
    rt_peons         = 1
```

