



MIGHT & MAGIC
HEROES VI
Triggered AI Scripting Guide

Version 1.0
November 4, 2011

© 2011 Ubisoft Entertainment. Developed by Black Hole Entertainment.

Introduction

The present guide will teach you how to create complex behaviors for the AI players.

Important note: this guide is for **Advanced users only**. In order to use the functions described below, you should have mastered the functions presented in the [Quickstart Guide](#) and the [Storytelling Tools Guide](#).

1 Table of Contents

1	Table of Contents	2
2	AI Triggers	4
2.1	AI and AI triggers in general	4
2.1.1	Skirmish AI Basics	4
2.2	Triggered AI – Control Hero	4
2.2.1	Name (strName)	4
2.2.2	Player (iPlayerId)	4
2.2.3	Source entity (pSource)	4
2.2.4	Pack Handling (dwEnemySuperiorityHandlingType)	5
2.2.5	Keep Pursuing Goal (dwKeepPursuingGoal)	6
2.2.6	Make Hero a Skirmish Hero when completed all orders (dwMakeSkirmishHeroIfDone)	6
2.2.7	Army Composition (ArmyComposition)	7
2.2.8	Orders	9
2.2.9	Rules	9
2.3	Triggered AI – Keep Guards	9
2.3.1	Player (iPlayerId)	9
2.3.2	Entity (pCastle)	9
2.3.3	Army Composition	9
2.4	Orders in general	10
2.4.1	Priorities	10
2.4.2	Objectives and parsing orders	10
2.4.3	Army Fill Up Phase	11
2.4.4	Army Unload	11
2.4.5	Targets set in Arrays	12
2.4.6	Enemy Superiority Handling (dwEnemySuperiorityHandlingType inside orders)	12
2.4.7	Use Waypoints (pUseWaypoints branch of orders)	13
2.5	Orders	13
2.5.1	Attack Target	13
2.5.2	Attack / Defend Region	14
2.5.3	Attack in Area	15
2.5.4	Attack Player	16
2.5.5	Explore Location	16
2.5.6	Explore Area	18
2.5.7	Defend Town/Fort	18
2.5.8	Refill Army to Percentage of Max Army Power	20
2.5.9	Move and Stay	20
2.5.10	Interact with Entity	21
2.6	Rules	22

2.6.1	Rules in general	22
2.6.2	Set Forbidden Regions	22
2.6.3	Use the same hero	23
2.6.4	Use any hero	23
2.6.5	Set Hero Aggression Limit (Set Free Target Limit)	23
2.6.6	Set Free Interact Limit	24
2.6.7	Delay Objectives	24
2.6.8	Set Avoid Limits	25
2.6.9	Disable Town Portal	26
2.6.10	Prefer Garrison Slot	26
2.7	Army Rules	26
2.7.1	Army Rules in general	26
2.7.2	Default Army Rules	26
2.7.3	Disable recruit of entity-type.....	27
2.7.4	Keep Army Composition.....	27
2.7.5	Army Composition Ratios	27
2.8	Other Triggered AI functions	28
2.8.1	Additional Triggered AI attributes	28
2.8.2	Triggered AI modifying Triggers.....	28
2.8.3	Turning on Skirmish AI after Triggered commands	30
2.8.4	AI Interact restrictions	30

2 AI Triggers

2.1 AI and AI triggers in general

2.1.1 Skirmish AI Basics

When the user chooses a computer controller for a player slot, this slot is considered as a Skirmish AI by default. Skirmish AI is the computer opponent in multiplayer or skirmish games. The Skirmish AI does not need to be controlled; it acts at its best to win the game. However, there are some ways to control or guide the Skirmish AI:

- In the game lobby, the computer player can be set to different difficulties.
- There are some trigger functions that control the Skirmish AI (as well as the Triggered AI). However, most trigger functions can control only the Triggered AI.

Skirmish AIs can be turned on/off during the course of the game with the “AdventureAI_ToggleSkirmishAI” trigger function. When the Skirmish AI is turned off, it becomes a Triggered AI. The Triggered AI’s benefit is that the user has more control over it, but its drawback is that it does not do anything by default; the user has to command it with AI trigger functions.

2.2 Triggered AI – Control Hero

The basic function to control a Hero is: “**AdventureAI/AdventureAI_ControlHero**”. Each “Control Hero” function is attached to one and only one hero of the AI. It has several parameters:

2.2.1 Name (strName)

This is the identifier (name) of this function. It can be used to turn the Control Hero off/on or to add/remove orders and rules to it.

2.2.2 Player (iPlayerId)

Defines which player slot will be controlled. Human player slots are not affected, when an AI function is used on them. Multiple “Control Hero” commands can exist for the same player carrying out different orders; a hero is always attached to these orders.

Note – Enter “0” for Player 1, “1” for Player 2, “2” for Player 3, etc.

2.2.3 Source entity (pSource)

Defines what the source of the Control Hero is.

- It can be set to a hero (use the Scenario/ScenarioEntity or the Scenario/Variable/GetEntityVariable function): the selected hero will be used to execute the orders. If the source hero is already executing the orders of another Control Hero, then this new command will replace the old one.
- It can be set to a town (use the Scenario/ScenarioEntity or the Scenario/Variable/GetEntityVariable function): the computer will assign a hero

without Control Hero command to the town (preferring heroes located in the town) and execute the orders. If there is no such hero available then the computer tries to buy/resurrect a hero in the town. If the hall of heroes is missing in the town then the computer attempts to build it first.

- It can be set to nothing: the computer will choose a Hero without Control Hero command for the source entity. If there are no free heroes, then it chooses the optimal town to buy a new hero to execute the orders. The computer will either buy a hero in a town that has hall of heroes built or it will choose the closest town (for its task) that has no hall of heroes, builds it and then buys a hero.

It is important that the computer has the same rules regarding the recruit of heroes: additional heroes cost more gold.

Location of buying back a dead or retreated Hero

The AI will buy back its fallen or retreated heroes in the following priority:

1. In the town where the Hero recruited creatures for the last time
2. In the town which has the highest priority among the Hero's defend town orders
3. In the town which is closest to the secondary target of its highest priority order

2.2.4 Pack Handling (dwEnemySuperiorityHandlingType)

With this parameter, the user can set the behavior of the AI. The user can set one of the following behaviors:

- "Smart": this is the default setting
- "Percentage": Attack enemies that are at least X% weaker than own army power
- "Amount": Attack enemies that are at least X army power weaker than own army power
- "Disabled"

Pack handling is a function to control the hero's route to its goal. When there are packs or enemy heroes in the way to carry out an order, this function is used. Only the first pack in the path is checked. For example if there is a pack on the shortest way with an army power of 850 and the pack handling is set to "Attack enemies that are at least 20% weaker than own army power", then the hero with 1000 army power will not go through that pack and tries to find another way. It is possible that there is no such way and the hero is at maximum army power; this makes the AI skip its actual objective in the order and do the next one. For more information on army power limits, see [2.2.7 Army Composition](#). For more information on order objectives and priorities, see [2.4.2 Objectives and parsing orders](#).

Smart means the AI decides on its own which the optimal route for its goal is. The AI simulates all battles in advance and chooses the shortest one that will keep its army power above the minimum.

Disabled means that the hero will go on the shortest route attacking everything on its way no matter how strong they are.

Note – Use negative values for “Percentage” or “Amount” to attack stronger enemies than own army power.

Avoiding Strong Enemy Heroes

The Triggered AI has a built in function to avoid enemy heroes that it cannot defeat. “Avoid areas” are created around all enemy heroes with a range of one turn movement. When the AI Hero executes an order, the order’s superiority handling is used inside these areas: the AI will not end his turn inside these areas if the superiority handling of the order does not allow it. The AI hero can go through these areas, but he needs to end his turn on a safe square. Towns and forts within “Avoid areas” are considered as safe spots.

When the AI hero starts his turn while he is already in an unsafe square, he will try to run away to a safe square considering his orders: he will try to execute the order with the highest priority and end its turn on a safe square. When there is no such square, because the enemy hero has much more movement points, the AI Hero’s “avoidance” rule becomes disabled for this turn and the hero straight up executes its highest priority order.

It is possible to fine-tune or disable this function with the [Set Avoid Limits](#) rule.

2.2.5 Keep Pursuing Goal (dwKeepPursuingGoal)

This is a True/False parameter. The default setting is False. In short:

Control Heroes that keep pursuing their goal persist through heroes, while Control Heroes whose “Keep Pursuing Goal” parameter is “False” die with the hero, and the function doesn’t even start when no hero can be bought or used.

False case: In the first turn of the AI after the “Control Hero” function is activated the computer tries to attach a hero to the function based on the “Source entity” settings. If the computer cannot attach any hero, the “Control Hero” function perishes. Also, when a hero dies with “Keep Pursuing Goal” “False”, the function perishes.

True case: The function will never perish and the computer waits until it can attach a hero to the “Control Hero” based on the “Source entity” settings. Also, when the attached hero dies, the computer will do the following:

- If the source entity is a hero, then the AI will always resurrect this hero (as if the [Use the same hero](#) rule would be used), but in any town (the closest to his objective is preferred). There is a [Use any hero](#) rule to disable this.
- If the source entity is a town, then the AI will select, hire or resurrect a hero in this town. If the town is owned by another player, the Control Hero won’t work until the computer captures it.
- If the source entity is set to “none”, then the AI will select, hire or resurrect a hero in any town (the closest to his objective is preferred).

2.2.6 Make Hero a Skirmish Hero when completed all orders (dwMakeSkirmishHeroIfDone)

This is a True/False parameter. The default setting is False.

False case: when the hero completed all primary objectives of his orders and there are no secondary objectives, then he will stay still and won’t move until any of his objectives

become active again, or until he is given a new order. Some orders can never be completed, for example the “Defend target” order.

True case: when all the hero’s primary objectives are in completed state, then he will turn into a hero of a Skirmish AI instead of checking and executing the secondary objectives. (Disabled orders are not checked.) Anytime his primary objectives switch from completed state to something else, or when he is given a new order, he will turn back to a hero of a Triggered AI and starts to complete his orders. See chapter [2.4.2 Objectives and parsing orders](#) for more information.

2.2.7 Army Composition (ArmyComposition)

With this setting, the user can configure the minimum and maximum army power of the hero. This parameter has the following sub-parameters:

Name (strName)

This is the identifier (name) of this function. It can be used to modify the army composition with other trigger functions.

Minimum Army Power (MinPower)

The hero goes to the closest town or fort and won’t start carrying out his orders until he can reinforce his army to this value. Whenever his army power drops below this amount, he will stop his orders and goes to the closest town or fort to fill up.

Maximum Army Power (MaxPower)

The hero won’t reinforce his army above this value. If the player sets this value lower than the Minpower value, then the value is automatically interpreted as MinPower+1. When the hero has initially more creatures than the set Maximum army power, he won’t disband the extra creatures, but he can deliver them to a town or fort when he is in Fill up phase. For more information on the Fill up phase, see [2.4.3 Army Fill Up Phase](#).

Automatic Increase (AutoWeeklyIncreaseType)

With this setting the user can set up the weekly growth of the Minimum and Maximum Army Powers for the Control Hero. It has the following options:

-
- Disabled (default) – The minimum and the maximum power won’t increase. AutoWeeklyIncreaseArg1 and 2 have no effect this case.
- Depending on Difficulty – The minimum and maximum power is automatically increased with a percentage based on the game’s difficulty level:
 - Easy: growth is 7%
 - Medium: growth is 9%
 - Hard: growth is 11%
 - Custom: the growth depends on the strength level of the AI (easy: 7%, medium: 9%, hard: 11%).

AutoWeeklyIncreaseArg1 and 2 have no effect this case.

- **Specific Value** – The user can specify two values: “Add to Minimum Army Power” (AutoWeeklyIncreaseArg1) and “Add to Maximum Army Power” (AutoWeeklyIncreaseArg2), which can be negative values as well. At the first day of each week the Minimum Army Power and Maximum Army Power of the Control Hero are modified by adding the appropriate values.
- **Percentage** – Entering “X” into “AutoWeeklyIncreaseArg1” will increase both Minimum and Maximum Army Power of the Control Hero by X%. Enter negative X to decrease them by X%.

AutoWeeklyIncreaseArg2 has no effect this case.

Triggers

Use the following trigger functions to modify the Army Composition of an existing Control Hero:

- Scenario/AdventureAI/AdventureAI_ModifyArmyPower
- AdventureAI_ModifyAutoWeeklyIncrease)

Refill percentage (RefillPercentage)

This parameter is inside the Army Composition part of the Control Hero or Keep Guards (see: [2.3 Triggered AI – Keep Guards](#)) functions. It can be set from 0 to 100 and its default value is 100.

In Control Hero: when the hero fills up his army, he cannot get more creatures per day than the set percentage of the Control Hero’s maximum army power. For example if the maximum army power is 1000 and the Refill percentage is 10%, he can get creatures worth of 100 army power per day.

When the hero reaches minimum army power and could buy more creatures than the set percentage, he will not start to do his orders until he reaches the maximum army power or until he cannot buy more creatures (due to resource or creature shortage): so he will wait a turn.

In Keep Guards: The town or fort can buy the set percentage of maximum army power per day.

Army Rules (pArmyRules)

The user can add several army rules that determine the army composition of the AI. Elements must be created inside the pArmyRules array to add army rules to the army composition. See [2.7 Army Rules](#) for more information.

Army Composition in “Keep Guards” functions

There is a similar Army Composition parameter inside the function called the “Keep Guards” that can be used for controlling the armies of towns and forts. The difference between them is that Keep Guards’ Army composition is lower in priority than the Control Hero’s Army Composition. When there is a hero in a town and both the hero and the town need to fill up their armies, the following happens:

1. The hero fills up his army to his minimum army power value.
2. The town fills up its army to its minimum army power value.

3. The hero fills up his army to his maximum army power value.
4. The town fills up its army to its maximum army power value.

The hero can transfer creatures from the town to himself only if the town's army power won't drop below its minimum army power.

Siege Split Behavior

When there are more than 7 different types of creatures in a town, or when there are 2 heroes in the town and the AI player is attacked, it will split its forces into two: one battle outside the walls and one siege battle after that (if the first one results in a defeat). When this happens the AI will try to concentrate its forces (up to his maximum army power) into the garrison slot for the second battle.

See [2.6.10 Prefer Garrison Slot](#) for further information.

2.2.8 Orders

The user can add several orders that determine the goals of the AI (see [2.5 Orders](#) for more information). Both rules and orders can be put in the pRules branch of the Control Hero as new elements in the array.

2.2.9 Rules

The user can add several rules that can guide or restrict the AI (see [2.6 Rules](#) for more information). Both rules and orders can be put in the pRules branch of the Control Hero as new elements in the array.

2.3 Triggered AI – Keep Guards

The basic function to control an army garrisoned inside a Town or Fort is: **“AdventureAI/AdventureAI_KeepGuards”**. It has several parameters:

2.3.1 Player (iPlayerId)

Defines which player slot will be controlled. Human player slots are not affected, when an AI function is used on them. A town or fort can have multiple “Keep Guards” commands attached to it, but only one for each AI player.

Note – Enter “0” for Player 1, “1” for Player 2, “2” for Player 3, etc.

2.3.2 Entity (pCastle)

An entity can be selected, but the function will work only with town or fort entities. The selected entity will be controlled. Entities of other players can also be selected and the orders and rules will be applied to them when they change owner to the selected player. For example AI Player 1 and AI Player 2 can have different Keep Guards function set for the same town.

2.3.3 Army Composition

This parameter is the same as the Control Hero's Army Composition parameter (see [2.2.7 Army Composition](#)).

When a town has more creatures than the Maximum Army power set for it, they will be kept (the AI won't disband any of them). Instead, the AI Player's heroes will reinforce their army from the excess amount of creatures.

When there are two heroes in the town and both heroes have Control Hero functions with Army Composition settings, the town's Army Composition rule is temporarily disabled. It is reactivated as soon as one of them exits the town.

2.4 Orders in general

2.4.1 Priorities

Orders are used to specify the actions and goals of the Control Hero. A Control Hero function without orders will do nothing.

Control Hero functions can have multiple orders. They have priorities; the lower the number, the higher their priority is. The highest priority order is attempted to be executed first. If it cannot be executed then the AI attempts to execute the order with the second highest priority in the priority list, and so on. When there are two orders with the same priority value (for example with a priority of 4), then "sub-priorities" are randomly created between them (e.g. 4.1 and 4.2). These "sub-priorities" are first randomized when the Control Hero is started and they are re-randomized each time the hero dies.

2.4.2 Objectives and parsing orders

Orders can have secondary objectives that will be executed only after all orders' primary objective is completed or skipped. The objective parsing order is the following:

1. Primary objective of order with priority 0
2. Primary objective of order with priority 1
3. Primary objective of order with priority 2
4. ...
5. Secondary objective of order with priority 0
6. Secondary objective of order with priority 1
7. Secondary objective of order with priority 2
8. ...

The AI checks each objective one after the other and sets their states. An objective can be in only one of the following states:

- Active – The topmost objective that meets the activation conditions turns to active. This is the objective the hero will do or currently does. All objectives below it are set to inactive state.
- Completed – When the objective's completion conditions are met, the objective is set to completed state and the next objective gets checked. A completed objective can turn into an active objective later. For example when the objective is to kill an enemy hero and he gets resurrected.

- Skipped – When the objective’s skip conditions are met, the objective is set to skipped state and the next objective gets checked. For example if there is no threat in the area of the town that is to be defended, then the AI skips this objective.
- Inactive – All objectives below the currently active one are inactive.

Secondary objectives cannot be skipped or completed; when the AI gets to the first secondary objective (because all primary objectives were skipped or completed), then the hero will do this until any of the primary objectives become active again.

A completed or skipped order’s secondary objective will still be executed when the parsing gets to it. In case an order does not have a secondary objective, then the next order’s secondary objective is parsed. When a Control Hero does not have any secondary objectives and all primary objectives are completed or skipped, then the hero will stay in his place.

Orders can be turned off or removed from a hero. When parsing an order that is turned off, it is counted as an order that is always in completed state and its secondary objective is skipped as well.

For exact information on each order’s primary and secondary objective and their state conditions, see the detailed information of the orders below ([2.5 Orders](#)).

2.4.3 Army Fill Up Phase

The hero will go to the closest place (town or fort) where he can fill up his army under the following conditions:

- When his army power drops below the minimum army power, he immediately stops his objective and goes to the closest place for filling up at least until the minimum. He continues his objectives only after the minimum power is met.
- When his army power is between the minimum and the maximum and all his primary objectives are skipped or completed, then he goes to the closest place to fill up his army, and after that he checks the primary objectives again. It is not required to fill up until the maximum to carry on.

Additionally each time the hero is in a town or fort and his army power is between the minimum and the maximum, he fills up his army. This can happen during a defend (town) order, during an attack (town) order, etc.

2.4.4 Army Unload

There are some cases when the hero needs to get rid of some creatures in his army and unload them to towns or forts:

- When the hero’s army grows above its maximum army power value due to various reasons (he visits a dwelling and creatures join him, or when he gets creatures by trigger functions).
- When a creature inside his army is restricted by army rules ([2.7 Army Rules](#)).

When one of the above conditions is met, the hero won’t go right away to a town or fort to unload the creatures. He will unload them only if he enters a town or fort by his regular

actions (like at Fill Up Phase, Defend order, etc.). The hero unloads these creatures to a town or fort only if:

- the town/fort has free slots for them
- the hero will stay above his minimum army power

Also, the hero will go to the closest place to unload the creatures when all his primary objectives are skipped or completed. It is not required to unload everything or to find another place and unload the rest of the creatures to carry on.

2.4.5 Targets set in Arrays

Some orders have a parameter that can be set in an array to be able to easily create several orders with the same parameters except for the target and the priority. For example when an [Attack Target](#) order is set with three target entity parameter and with a priority of 4, it is the same as if three different orders would be created for the targets with “sub-priorities” of 4.1, 4.2 and 4.3. The “sub-priorities” are randomized the first time when the control hero starts to function and each time the hero dies.

2.4.6 Enemy Superiority Handling (`dwEnemySuperiorityHandlingType` inside orders)

This is an order parameter that is almost the same as the Control Hero’s Pack Handling parameter, but Enemy Superiority Handling is applied only to the target(s) of the order. If it is enabled and the target is stronger than the allowed, then the hero will skip the objective. The user can set one of the following behaviors:

- “Smart”: this is the default setting
- “Use Pack Handling”
- “Percentage”: Attack enemies that are at least X% weaker than own army power
- “Amount”: Attack enemies that are at least X army power weaker than own army power
- “Disabled”

With using both Pack Handling and Enemy Superiority Handling, it is possible to set up different behaviors for the target and the route. For example making the Hero not to attack neutral packs on its way (find the shortest unblocked way) to the target, but attack the target regardless of its power.

Smart means the AI decides on its own, based on Skirmish AI target handling. The AI simulates the battles in advance. In case the target can be defeated by the simulation, the AI will attack it.

Choosing the Use pack handling option makes the Superiority Handling the same as in the Control Hero’s Pack Handling setting.

Note – Use negative values for “Percentage” or “Amount” to attack stronger enemies than own army power.

2.4.7 Use Waypoints (pUseWaypoints branch of orders)

This is a True/False order parameter (dwEnabled); its default state is False. With this parameter the user can specify waypoints the hero must follow before executing the objective, be it either primary or secondary. Waypoints can be set with locations in an array. The size of the locations does not matter, the hero will go to its center. Each time an objective becomes active (and the “dwOnce” parameter is false), the hero goes through the given waypoints using the Control Hero’s pack handling settings and then goes for the target. Pack Handling is checked only for the first enemy through the waypoints. It has the following sub-parameters:

- Method (dwJoin) – A parameter that can be set to “true” or “false”. “False” means the hero will start his path from the first location and goes through all one by one. “True” means the hero will find the closest point of the “road” made from the locations and proceeds from there.
- Only once (dwOnce) – A parameter that can be set to “True” or “False”, default is “False”. If it is set to “True” and the Hero completed the path once by reaching the last region, then the “Use Waypoints” parameter turns automatically to “False”, meaning the waypoints will not be used again.

When the hero goes through the waypoints, there are several cases when doing them can be aborted:

- At the beginning of each turn, the AI parses through all objectives and it is possible that an objective with higher priority becomes active.
- After each battle the army power of the hero might change so it is possible that the hero cannot continue due to superiority or pack handling.
- After doing all waypoints, the AI parses through all objectives again (in case an order with higher priority becomes active in the process).

2.5 Orders

2.5.1 Attack Target

Scenario/AdventureAI/Rule/AdventureAI_Rule_AttackTarget

Primary Objective

This orders the hero to attack or capture the specific target. When parsing the objective, it becomes:

- **Completed**, when the target entity is:
 - dead
 - captured or owned by the AI (in case it is a building)
 - not existing on the map
 - owned by an ally of the AI
 - not reachable for the hero because there is no path to the target

- in forbidden regions (see [2.6.2 Set Forbidden Regions](#))
- **Skipped**, when the target entity is:
 - not reachable for the hero because of the Pack Handling restrictions
 - restricted by the order's Superiority Handling parameter (see below)
- **Active** again, when the target entity:
 - is resurrected
 - is captured or owned by the enemy of the AI (in case it is a building)
 - appears on the map
 - becomes reachable for the hero

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Entity (Array) (pTarget) – This is the target for the AI to attack or capture. See [2.4.5 Targets set in Arrays](#) for more information on arrayed parameters.
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.2 Attack / Defend Region

Scenario/AdventureAI/Rule/AdventureAI_Rule_AttackRegion

Primary Objective

This orders the hero to attack enemy heroes and capture enemy towns/forts in the specific region. When parsing the objective, it becomes:

- **Completed**, when there are no possible target entities, because all of them are:
 - dead
 - captured or owned by the AI (in case it is a building)
 - owned by an ally of the AI
 - not reachable for the hero because there is no path to the target
 - in forbidden regions (see [2.6.2 Set Forbidden Regions](#))
- **Skipped**, when all possible target entities are:
 - not reachable for the hero because of the Pack Handling restrictions

- restricted by the order's Superiority Handling parameter (see below).
- **Active** again, when an entity in the region becomes a possible target, for example:
 - by resurrection
 - a town/fort inside is captured or owned by the enemy of the AI
 - by placing one with triggers
 - by becoming reachable for the hero

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Region (Array) – See [2.4.5 Targets set in Arrays](#) for more information on arrayed parameters.
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.3 Attack in Area

Scenario/AdventureAI/Rule/AdventureAI_Rule_AttackArea

Primary Objective

This order works the same as [2.5.2 Attack / Defend Region](#), but in a control area specified by an area control building. When parsing the objective, it changes state the same way as described in the “Attack in Region” order.

Secondary Objective

The secondary objective orders the hero to go to the area control building of the area and wait there for possible enemy targets.

Parameters

- Name (strName)
- Priority (dwPriority)
- Area Control Entity (Array) (pTarget) – The user can select an area control building (a town or fort). This determines the target area. See [2.4.5 Targets set in Arrays](#) for more information on arrayed parameters.
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.4 Attack Player

Scenario/AdventureAI/Rule/AdventureAI_Rule_AttackPlayer

Primary Objective

This order works like several [Attack Target](#) orders: one for each Hero, Town and Fort of the specified player. When the player captures a town or fort, or when he recruits new heroes, new “hidden” Attack Target orders are created automatically. The priorities of these “hidden” orders are based on the length of the path from the AI hero: the closer the target is, the smaller (and thus higher) its priority is.

These “hidden” orders can also be skipped or completed the same way as the Attack Target order.

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Player (pTarget)
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.5 Explore Location

Scenario/AdventureAI/Rule/AdventureAI_Rule_ExploreRegion

Primary Objective

This orders the hero to explore the specific location: he picks up resources and items, visits adventure map buildings, flags mines, dwellings, etc. The hero clears neutral packs if needed and allowed by the Enemy Superiority Handling. This order does not make the hero attack other players or area control buildings (even neutrals) in the location.

Entities that are always targets for the explore order:

- Objects that can be picked up
- Buildings providing permanent bonuses (e.g. Armorsmith)

Entities that are targets only when the AI Player is the owner of the area control building of the explored location:

- Buildings that can be captured (except Towns and Forts, e.g. Lighthouse)

If they have no area control building, they are targets for the explore order.

Entities that are no targets for the explore order:

- Buildings providing temporary bonuses (“for next battle” and “for this week”)

Use [2.6.6 Set Free Interact Limit](#) to make the hero visit these entities.

When parsing the objective, it becomes:

- **Completed**, when there are no possible target entities, because all of them are:
 - picked up
 - visited
 - owned by an ally of the AI
 - in forbidden regions (see [2.6.2 Set Forbidden Regions](#))
 - restricted by trigger functions or in their properties to be visited by the AI
 - captured once by the AI
 - not reachable for the hero because there is no path to the target
- **Skipped**, when all possible target entities are:
 - not reachable for the hero because of the Pack Handling restrictions
 - restricted by the order's Superiority Handling parameter (see below).
- **Active** again, when an entity in the location becomes a possible target, for example:
 - when the AI gets ownership of its area control building.
 - by placing an entity with triggers that can be explored
 - by becoming reachable for the hero
 - by disabling forbidden regions
 - by disabling restrictions with triggers

Buildings providing temporary bonuses (e.g. Rally Flag) need not be visited for the order to be completed.

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Location (Array) (pTarget) – See [2.4.5 Targets set in Arrays](#) for more information on arrayed parameters. This is the target location in where the AI will explore.
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – It is applied to neutral packs that guard the target entities. Packs that do not guard anything (but block the way) are handled with Pack Handling. For more information on Enemy Superiority Handling, see [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.6 Explore Area

Scenario/AdventureAI/Rule/AdventureAI_Rule_ExploreArea

Primary Objective

This order works the same as [2.5.5 Explore Location](#), but in a control area specified by an area control building. When parsing the objective, it changes state the same way as described in the “Explore Location” order.

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Entity (Array) (pTarget) – The user has to select an area control building to determine the area in where the AI will explore. See [2.4.5 Targets set in Arrays](#) for more information on arrayed parameters.
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.7 Defend Town/Fort

Scenario/AdventureAI/Rule/AdventureAI_Rule_DefendTarget

Primary Objective

This orders the hero to defend a town or fort: he enters the building and waits when there is a threat in the defend area. A Defend Area can be set for the order and the hero acts depending on the order’s Enemy Superiority Setting:

- If an enemy hero enters the Defend Area and it is not restricted by the Superiority Handling, then the hero leaves the town or fort and attacks him.
- However if there is at least one enemy hero in the Defend Area and it is restricted by the Superiority Handling, then the hero stays in the town or fort, behind its walls.

When the town/fort that is to be defended is captured by the enemy of the AI, the order works as if an [Attack Target](#) order, targeting the town/fort keeping its Enemy Superiority Handling parameter; the hero has to re-capture the town/fort before the order changes back to the original defend order.

When parsing the objective, it becomes:

- **Skipped**, when:
 - there are no enemy heroes in the Defend Area
 - the target town/fort is not reachable for the hero (either because of the Pack Handling restrictions or because there is no path to the target)

- both the visiting and the garrison slot of the town/fort is occupied by the AI's heroes
- the target town/fort is owned by an ally of the AI
- the target town/fort is restricted by trigger functions or in their properties to be visited by the AI (in case it is captured by the enemy of the AI)
- the town/fort is in forbidden regions (see [2.6.2 Set Forbidden Regions](#))
- **Active** again, when:
 - an enemy hero enters the Defend Area
 - the town/fort that is to be protected becomes captured by the enemy of the AI (the order changes to an Attack Target order, as described in the primary objective)
 - the town/fort becomes reachable for the hero

This order never changes to completed state.

Secondary Objective

The secondary objective orders the hero to enter the town or fort that is to be defended.

When the town/fort that is to be defended is captured by the enemy of the AI Player, the order works as if an [Attack Target](#) order, targeting the town/fort, without checking the Superiority Handling (as if it would be disabled).

Parameters

- Name (strName)
- Priority (dwPriority)
- Entity (Array) (pTarget) – This is the target town or fort the AI must defend. See [2.4.5 Targets set in Arrays](#) for more information on arrayed parameters.
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Defend Area (dwDefendType) – The area that can be set by choosing one of the following 2 options:

Turns: The user enters a positive number. The Defend Area is the area the AI hero can reach within the set amount of turns like he started moving from the entrance of the town/fort.

Adventure map squares: The user enters a positive number. The Defend Area is the area the hero can reach from the set amount of “square points” like he started moving from the entrance of the town/fort, where moving horizontally or vertically costs 1, moving diagonally costs 1.41 “square points”

Chase extension rule applies: When the hero started moving towards the threat, he can leave the Defend Area, but he will abandon chasing his target and returns to the town/fort when the threat leaves the Defend Area extended by 33%.

- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.8 Refill Army to Percentage of Max Army Power

Scenario/AdventureAI/Rule/AdventureAI_Rule_RefillArmy

Primary Objective

This orders the hero to refill his army to a percentage of his maximum army power value. When parsing the objective, it becomes:

- **Completed**, when the hero's army power is above the required amount
- **Skipped**, when:
 - the hero cannot reach any of his towns or forts (either because of the Pack Handling restrictions or because there is no path to them)
 - the AI does not have any towns or forts
- **Active** again, when:
 - the army power of the hero drops below the required amount
 - the hero can reach a town or fort

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Percentage (pPercentage)
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.9 Move and Stay

Scenario/AdventureAI/Rule/AdventureAI_Rule_MoveAndStay

Primary Objective

This orders the hero to move to a location and stay there for a set amount of turns. The hero will try to go as close to the center of the location as he can. When parsing the objective, it becomes:

- **Completed**, when the hero is in the location and stood there for the set amount of turns
- **Skipped**, when all squares of the location are:
 - forbidden (see [2.6.2 Set Forbidden Regions](#))
 - not passable

- not reachable for the hero because of the Pack Handling restrictions
- not reachable for the hero because there is no path to them
- **Active** again, when a square in the location becomes passable, for example:
 - by disabling forbidden regions
 - by becoming reachable for the hero

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Destination (Array) (pTarget) – It's sub-parameters are:
 - Location (pLocation)
 - Stay Duration (iDuration) – An Integer can be set here. The hero will stay in the location for the set duration (measured in turns). “-1” means he will stay there forever (or until a higher priority order becomes active again).
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#).
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.5.10 Interact with Entity

Scenario/AdventureAI/Rule/AdventureAI_Rule_InteractWithEntity

Primary Objective

This orders the hero to interact with a specified entity. It can be any entity, for example resources, adventure map buildings, mines, dwellings, towns, forts, heroes, etc... The interaction will happen regardless of the owner of the target: if the target is an enemy hero, then it will be attacked, if the target is a fort owned by the AI Player, then the hero will enter the fort. When parsing the objective, it becomes:

- **Completed**, when the hero has interacted with the target. Once this order becomes completed, it won't change to any other states.
- **Skipped**, when the target entity is:
 - dead
 - in forbidden regions (see [2.6.2 Set Forbidden Regions](#))
 - not existing on the map
 - not reachable for the hero because there is no path to the target
 - not reachable for the hero because of the Pack Handling restrictions

- restricted by the order's Superiority Handling parameter (see below)
- **Active** again, when the target entity:
 - is resurrected
 - appears on the map
 - becomes reachable for the hero

Secondary Objective

This order does not have a secondary objective.

Parameters

- Name (strName)
- Priority (dwPriority)
- Entity (Array) (pTarget) – This is the interact target. See [2.4.5 Targets set in Arrays](#) for more information on arrayed parameters.
- Enemy Superiority Handling (dwEnemySuperiorityHandlingType) – See [2.4.6 Enemy Superiority Handling](#). This applies only if the target is an enemy.
- Use Waypoints (pUseWaypoints branch) – See [2.4.7 Use Waypoints](#).

2.6 Rules

2.6.1 Rules in general

Rules are guidelines or restrictions in general. As orders, rules can also be turned off/on, removed/added and their priority can be changed. If a rule has no priority, then it functions until turned off or removed.

The list of rules is the following:

2.6.2 Set Forbidden Regions

Scenario/AdventureAI/Rule/AdventureAI_Rule_AddForbiddenRegions

The AI hero cannot enter the specified regions. This rule makes an order completed even if there are possible targets for the order, but they are inside a forbidden region. For example if the hero picked up and visited everything in the given area through an explore area order, except for some resources inside a forbidden region, then the objective becomes completed. The objective can turn back to active state if the “Set Forbidden Regions” rule gets disabled.

Parameters

- Name (strName)
- Forbidden regions (pRegions) – The user can set a region or multiple regions in the array.

2.6.3 Use the same hero

Scenario/AdventureAI/Rule/AdventureAI_Rule_UseSameHero

When the “Keep Pursuing Goal” parameter of the “Control Hero” is “True” and the hero carrying out the orders dies or retreats, this is a rule to tell the computer to revive or re-hire the same hero. Otherwise (when this rule is not added and “Keep Pursuing Goal” is “True”), the computer can buy new heroes or it can select other heroes without “Control Hero” to continue carrying out the orders (unless the Source of the Control Hero function is a specific hero).

Parameters

- Name (strName)

2.6.4 Use any hero

Scenario/AdventureAI/Rule/AdventureAI_Rule_UseAnyHero

This rule is effective only when all the following conditions are met:

- The “Keep Pursuing Goal” parameter of the “Control Hero” is “True”.
- The hero carrying out the orders dies or retreats.
- The Source of the Control Hero function is a specific hero.

This rule tells the computer that it can use any hero after the initial one dies. Otherwise (when this rule is not added) and “Keep Pursuing Goal” is “True”, the computer can only use the hero that was specified as the source.

Parameters

- Name (strName)

2.6.5 Set Hero Aggression Limit (Set Free Target Limit)

Scenario/AdventureAI/Rule/AdventureAI_Rule_SetFreeTargetLimit

This rule allows the hero to leave his path to attack enemy (including neutral) heroes, towns and forts which are in range of X squares. This range can be set with X squares or X turns (X can be a real number).

Parameters

- Name (strName)
- Priority (dwPriority) – The rule functions only if its priority is higher or equal the objective the hero is currently doing.
- Free Target Limits (dwTargetLimitType) – With this parameter, the user can set the free target limit. The user can select one of the following behavior:
 - “Percent”: Treat enemy heroes that are at least X% weaker than own army power as attack targets
 - “Amount”: Treat enemy heroes that are at least X army power weaker than own army power as attack targets

- Range (dwDefendRangeType) – Same parameter as in [2.6.6 Set Free Interact Limit](#).

Multiple Set Free Target Limit rules can be added to the same Control Hero only if their priorities are different, because only one of these rules can be in effect when the AI hero executes his orders. The closest higher or equal priority Set Free Target Limit rule will be active compared to the currently executed order.

2.6.6 Set Free Interact Limit

Scenario/AdventureAI/Rule/AdventureAI_Rule_SetFreeInteractLimit

This rule allows the hero to leave his path to visit some objects, which are in range of X squares. This range can be set with X squares or X turns (X can be a real number).

The hero can visit the following type of entities with this rule:

- Objects that can be picked up
- Buildings providing temporary bonuses (“for next battle” and “for this week”)
- Buildings providing permanent bonuses
- Buildings that can be captured

Parameters

- Name (strName)
- Priority (dwPriority) – The rule functions only if its priority is higher or equal the objective the hero is currently doing.
- Enemy Superiority Handling (dwTargetLimitType) – It is applied to neutral packs that guard the target entities. See [2.4.6 Enemy Superiority Handling](#).
- Range (dwDefendRangeType) – Can be set with X turns or X squares.
- Black Mask multiplier (iBlackMaskModifier) – (default 50) Range is multiplied with this percentage-based factor when the AI Hero is under the Black Mask for the Player. For example if this value is 50 and the Range is 6, then the free interact range becomes 3 under the Black Mask.

Note – Using multiple Set Free Interact Limit rules in the same Control Hero is currently not supported.

2.6.7 Delay Objectives

Scenario/AdventureAI/Rule/AdventureAI_Rule_DelayObjectives

A countdown timer starts, when the AI hero dies. The hero won't do any of his objectives that are lower or equal in priority than this rule's priority until the timer expires. He will still fill up his army and wait in the town where he was resurrected (or do other objectives that are higher in priority than the delay rule's priority).

Important: When the control hero is activated first with this rule, the rule will have no effect until the hero dies!

The countdown timer restarts each time the hero dies.

Parameters

- Name (strName)
- Priority (dwPriorityTreshold) – Orders that are lower or equal in priority will be delayed until the timer expires.
- Delay Timer (dwDelaytime) – This is the countdown timer in turns.

2.6.8 Set Avoid Limits

Scenario/AdventureAI/Rule/AdventureAI_Rule_SetAvoidLimit

This rule is used to fine-tune or disable the “Avoiding Strong Enemy Heroes” default function of [Pack Handling](#). The user can set up multiple ranges around enemy heroes with army power handling values. For example the user can set up a range of 1 turn and 20% army power handling and another one with 1.2 turn and 10% army power handling. This means the AI hero won’t stop on a square that an enemy hero can reach within 1 turn and the enemy hero has more army power than 80% of the AI hero’s army. The second range means the AI hero won’t stop on a square that an enemy hero can reach within 1.2 turn and the enemy hero has more army power than 90% of the AI hero’s army.

To disable the “Avoiding Strong Enemy Heroes” default function, the user can set up Avoid Limits of 0 turn or 0 square or a disabled army power handling (see below).

Parameters

- Name (strName)
- Array (pAvoidData) – This is an Array in which new Elements can be created. Each element should be an AdventureAI_AvoidData that can be found in the Scenario/AdventureAI folder. Each AdventureAI_AvoidData has the following parameters:
 - Range Type (dwDefendType) – Can be set to “Turn” or “Square”.
 - Range (fDefendAmount) – This parameter can be used to set up ranges of 1.2 turns for example. If Range Type is set to Squares, then this number is rounded to the top.
 - Army Power Handling (dwEnemySuperiorityHandlingType) – Almost the same as Enemy Superiority Handling. The user can set one of the following behaviors:
 - “Smart”: if the enemy hero as an attacker could defeat the AI hero, then he is avoided
 - “Amount”: Avoid heroes that are stronger than own army power decreased by X
 - “Percent”: Avoid heroes that are stronger than own army power decreased by X%
 - “Disabled”

2.6.9 Disable Town Portal

Scenario/AdventureAI/Rule/AdventureAI_Rule_DisableTownPortal

With this rule the user can specify towns that cannot be the target of basic and advanced town portal spells for the AI. When the Control Hero uses the advanced town portal, it will teleport to the closest town that is not disabled.

Parameters

- Name (strName) – The identifier of the rule. The rule can be enabled/disabled with this name.
- Entity (Array) (pEntities) – Towns set with this parameter cannot be the target of the town portal spells (neither the basic, nor the advanced version of the spell). (Use the Scenario/ScenarioEntity or the Scenario/Variable/GetEntityVariable function on each Element.)

2.6.10 Prefer Garrison Slot

Scenario/AdventureAI/Rule/AdventureAI_Rule_PreferGarrisonSlot

If a Siege Split case happens (See [2.2.7 Army Composition](#)), then the hero with this rule will prefer to stay in the garrison slot. The hero gains priority on army composition: his army power fills up to the maximum limit first, and then the spare creatures will be given to the hero in the visiting slot or sent out to the first battle without walls.

Note – When both heroes in the town have this rule, the garrison hero is chosen randomly.

Parameters

- Name (strName)

2.7 Army Rules

2.7.1 Army Rules in general

Army Rules are guidelines or restrictions for the heroes' army composition. As orders, the army rules can also be turned off/on or removed/added. Elements must be created inside the pArmyRules array to add army rules to the army composition.

2.7.2 Default Army Rules

There are some default army rules for both the Skirmish and the Triggered AI that work for all armies and are always active (therefore it is not required to add them to each Control Hero or Keep Guards triggers). These default army rules are the following:

- A maximum of 4 large (2x2) Creatures are allowed in the army.
- The AI prefers buying Creatures from its own (starting) faction.
- The AI tries to build a diverse army. For example if it has 1000 Shark Warriors and 1000 Kappas in the Creature Pool, and he doesn't have enough resources to buy them all, it won't build an army with 500 Shark Warriors and 0 Kappas, unless it has a strict army rule for that.

2.7.3 Disable recruit of entity-type

Scenario/AdventureAI/ArmyCompositionRule/AdventureAI_ArmyCompositionRule_DisableUnit

This Army Rule forbids the Triggered AI to have the specified entity-type in its army.

Parameters

- Name (strName)
- Entity Type (Type)
- Include other version (bUpgrades) – True(default)/False. If it's set to True and a basic entity is selected, then its upgrade version will be disabled as well. Also, if it's set to True and an upgrade entity is selected, then its basic version will be disabled as well. If it's set to False, then its other version will not be disabled.

2.7.4 Keep Army Composition

Scenario/AdventureAI/ArmyCompositionRule/AdventureAI_ArmyCompositionRule_Keep

With this rule, the Player can order the AI to keep the initial army composition.

Parameters

- Name (strName)
- dwKeepType – Initial:

When the rule is added to the Control Hero or Keep Guards function, the AI stores the ratio of creatures that are inside its army (for example 3 ghouls, 4 skeletons, 6 ghosts) and it will keep this ratio until the rule is disabled. There is a ~10% allowance of new unit types or variance of ratios.

dwKeepType – Actual is not used.

2.7.5 Army Composition Ratios

Scenario/AdventureAI/ArmyCompositionRule/AdventureAI_ArmyCompositionRule_Ratios

When the AI buys a new Creature to its army, it considers these ratios first. The ratios are converted to percentages (i.e. 4 Core, 6 Elite, 10 Champion to 20% Core 30% Elite, 50% Champion), and then the AI starts to buy Creatures in a way that these percentages are allowed to change by ~10%. If it is not possible for the AI to buy any creatures (because the ratios are too restrictive or there are too few types of creatures in its pool) within the allowed range, the hero will wait and won't start any of his objectives.

- Name (strName)
- Army Composition Type (dwCompositionType) – The user can select from "Army Power" or "Stack Number". This determines how the army fill up works based on the "Ratio" values.

50% Core and 50% Elite in Army Power Army Composition Type translates to "Fill up your army in a way that 50% of its army power shall be Core and 50% of its army power shall be Elite creatures" (i.e. 50 Kappas and 14 Kenshis). In the other

hand, the same rule in Stack Number Army Composition Type translates to “Fill up your army in a way that 50% of the creatures shall be Core and 50% of the creatures shall be Elite creatures” (i.e. 22 Kappas and 22 Kenshis).

- Fill up slots with any creatures after duration (iFillTime) – The AI will ignore the ratios when this duration expires and fills up his army with any creatures. The computer still needs to have enough resources to buy the Creatures.
- Army composition – The user can set the following ratios:
 - Tier Ratios – These determine the tier composition of the army.
 - Core (iTier1) – The user can set a positive integer.
 - Elite (iTier2) – The user can set a positive integer.
 - Champion (iTier3) – The user can set a positive integer.
 - Attack Type Ratios – These determine the attack type composition of the army.
 - Melee (iMelee) – The user can set a positive integer.
 - Ranged (iRanged) – The user can set a positive integer.
 - Affinity Ratios – These determine the affinity composition of the army.
 - Might (iMight) – The user can set a positive integer.
 - Magic (iMagic) – The user can set a positive integer.

A Creature has Might affinity if the school of its damage is Might, otherwise its affinity is Magic.

2.8 Other Triggered AI functions

2.8.1 Additional Triggered AI attributes

When the Skirmish AI is turned off for a Player slot, the computer player will still do the following:

- Constructs buildings in its turn if it can, considering its orders and needs. For example if the source of a “Control Hero” function is a town that has no hall of heroes, then the computer will build it.
- Converts its towns.
- Recruits creatures in its towns.
- Upgrades creatures.
- Revives its heroes if the “Keep Pursuing Goal” parameter of the “Control Hero” is “True”.

2.8.2 Triggered AI modifying Triggers

These trigger actions affect only Triggered AIs:

- **“Modify Automatic Increase of Triggered**

- AI”**(Scenario/AdventureAI/AdventureAI_ModifyAutoWeeklyIncrease)

- Name (strName) – Enter the name of the Control Hero or Keep Guards or the name of their Army Composition.
 - Modify to
 - Disabled
 - Difficulty
 - Specific
 - Percentage

This trigger sets the given Control Hero’s or Keep Guards’ “Automatic Increase” parameter inside the “Army Composition” parameter to the specified types and values. For more information, see [2.2.7 Army Composition](#).

- **“Modify Army Power**

- Min/Max”**(Scenario/AdventureAI/AdventureAI_ModifyArmyPower)

- Name (strName) – Enter the name of the Control Hero or Keep Guards or the name of their Army Composition.
 - Add to MinPower (fMinPower) – Entered number will be added to the Army Power Min. Negative values can be used as well for subtraction.
 - Add to MaxPower (fMaxPower) – Entered number will be added to the Army Power Max. Negative values can be used as well for subtraction.

This trigger adds to the MinPower and the MaxPower parameters inside the “Army Composition” parameter of the Control Hero or the Keep Guards. For more information, see [2.2.7 Army Composition](#).

- **“Add New Rule or Order”** (Scenario/AdventureAI/AdventureAI_AddNewRule)

- Name (strOwnerName) - Enter the name of the Control Hero or Keep Guards to add a new order or rule. Enter the name of the Army Composition to add a new Army Rule.
 - New Rules List (pNewRules) – The user can select a rule or multiple rules with the Array function.

This trigger adds new rules to the specified Control Hero or Keep Guards.

- **“Remove Rule or Order”** (Scenario/AdventureAI/AdventureAI_RemoveRule)

- Name (strOwnerName) - Enter the name of the Control Hero or Keep Guards if you want to remove an order or rule. Enter the name of the Army Composition if you want to remove an Army Rule.
 - Rule Name (strRuleName) – Enter the name of the rule to be removed.

This trigger removes the rule from the Control Hero or Keep Guards.

- **“Enable/Disable Control Hero or Keep Guards”**(Scenario/AdventureAI/AdventureAI_SetPersistentEnabled)
 - Name (strPersistentName) - Enter the name of the Control Hero or Keep Guards.
 - Enable/Disable (dwEnabled) – False/True parameter. When set to false, the selected Control Hero or Keep Guards will be disabled. When set to true, the selected Control Hero or Keep Guards will be enabled.

This trigger enables or disables the whole Control Hero or Keep Guards.

- **“Enable/Disable Rule or Order”**(Scenario/AdventureAI/AdventureAI_SetRuleEnabled)
 - Name (strRuleName) - Enter the name of the rule or order.
 - Enable/Disable (dwEnabled) – False/True parameter. When set to false, the selected rule or order will be disabled. When set to true, the selected rule or order will be enabled.

This trigger enables or disables a rule, order or army rule inside a Control Hero or Keep Guards.

- **“Set Rule or Order Priority”** (Scenario/AdventureAI/AdventureAI_SetRulePriority)
 - Name (strRuleName) - Enter the name of the rule or order.
 - Priority (dwPriority) – Enter the new priority.

This trigger changes the priority of a rule or order inside a Control Hero or Keep Guards.

2.8.3 Turning on Skirmish AI after Triggered commands

The Skirmish AI can be turned on for a player slot even after it was turned off and several “Control Hero” and “Keep Guards” functions were used. The following events happen when the Skirmish AI is turned off while there are Triggered AI commands:

- All “Control Hero” and “Keep Guards” functions stop working and the Skirmish AI takes control over the heroes and towns.
- When the Skirmish AI is turned off again, the “Control Hero” and “Keep Guards” functions become active again.

2.8.4 AI Interact restrictions

There are three ways to restrict the AI to interact with entities like resources or adventure map buildings:

- In the Scenario Editor’s Properties window, there is a setting for each entity called “AI Interact”. If this setting is set to “No” for the current difficulty, then the AI (both Skirmish and Triggered) cannot interact with it (cannot use it, visit it, pick it up, trade with it, etc.).

- With trigger functions, the user can overwrite this property setting for each entity in each difficulty setting. This is the Scenario/AdventureAI/AdventureAI_AllowEntityInteract trigger.
- There is a [Set Forbidden Regions](#) AI rule to prevent the Triggered AI to enter an array of locations. This restriction can only be used in Control Hero functions therefore it has no effect on heroes controlled by the Skirmish AI.