

# Age of Empires Personality Types

September, 1997

© Microsoft Corporation, 1997

Age of Empires uses a fairly complex set of rules to run its computer player personalities. These rules are moderated and controlled by a set of variables called strategic numbers. The strategic numbers can be set for a specific scenario (by the scenario author). For random games, the strategic numbers are set up via specialized code modules (that look at the map type, map size, etc.).

## File names

These rules are initialized out of files with a .per extension. These files reside in the Data folder where you installed Age of Empires. They must have a .per extension to be recognized by the scenario builder and the AI loading mechanism.

## How the files interact with the game

A default set of strategic numbers will be initialized in the code. As such, one could run the game with no .per files. Due to the code-based default system, user-level .per files need not be complete; they only need to contain the changes to the previous layer.

As an example, if a scenario uses a specific .per file called sample.per, there will be two layers of strategic number initialization. The hardcoded default system will be set first. Then, any initializations contained in sample.per will be done.

## File content

The .per files simply contain a list of number pairs. The first number is keyed to the strategic number, the second is the value for that strategic number. The pairs do not need to be ordered. There are minimal facilities for bounds checking during the read process, but there are no checks for values that cause bad AI behavior or game slowdowns.

## File parsing and syntax

The .per files have a simple, line-based parser. Lines must be < 255 characters. Blank lines are allowed. Comments are denoted by a double forward slash "//". Comments should appear at the beginning of lines that are to be considered comment lines or following all of the keywords or values on the line:

```
//ExampleDEFAULT0      34      //SNPercentCivilianExplorers1      34
//SNPercentCivilianBuilders
//
END
//
```

The numeric pairs may exist in any order within the file. Repeated key numbers will overwrite the first value with the second value. The file must end with the word `END` on a line by itself.

## Strategic number description

The following is a list of the strategic numbers. The number on the left is the key for the strategic number on the right (this is the key that must be referenced in the VC file). Below each pair is a description of what that strategic number represents.

### CIVILIAN NUMBERS

0 SNPercentCivilianExplorers

Controls the normal, formula-based percentage of civilian explorers allocated. Must be >= 0.

1 SNPercentCivilianBuilders

Controls the normal, formula-based percentage of builders allocated. Must be >= 0.

2 SNPercentCivilianGatherers

Controls the normal, formula-based percentage of gatherers allocated. Must be >= 0.

3        `SNCapCivilianExplorers`

Caps the number of civilian explorers allocated.

Factored in after the percentage is calculated. Ignored when set to -1. Must be  $\geq -1$ .

4        `SNCapCivilianBuilders`

Caps the number of builders allocated. Factored in after the percentage is calculated. Ignored when set to -1. Must be  $\geq -1$ .

5        `SNCapCivilianGatherers`

Caps the number of gatherers allocated. Factored in after the percentage is calculated. Ignored when set to -1. Must be  $\geq -1$ .

18       `SNTotalNumberExplorers`

Caps the total number of explorers/explorer groups allocated. Factored in after the percentage of civilian explorers is calculated and the soldier groups are set up. Ignored when set to -1.

35       `SNMinimumCivilianExplorers`

Sets a minimum number of civilian explorers. Must be  $\geq 0$ .

117      `SNFoodGathererPercentage`

The required percentage of food gatherers. Must be  $\geq 0$  and  $\leq 100$ . This is applied before the normal calculation formula takes effect.

118      `SNGoldGathererPercentage`

The required percentage of gold gatherers. Must be  $\geq 0$  and  $\leq 100$ . This is applied before the normal calculation formula takes effect.

119      `SNStoneGathererPercentage`

The required percentage of stone gatherers. Must be  $\geq 0$  and  $\leq 100$ . This is applied before the normal calculation formula takes effect.

120      `SNWoodGathererPercentage`

The required percentage of wood gatherers. Must be  $\geq 0$  and  $\leq 100$ . This is applied before the normal calculation formula takes effect.

147      `SNMostNeededResourceLookAhead`

The number of build list objects the computer player will use to look ahead to tabulate the resources required. Only used for the dynamic gathering percentages. Must be  $\geq 0$ .

145      `SNNumberEnemyObjectsRequired`

The count of the number of enemy objects the computer player must see before dropping the number of civilian explorers down to the minimum level. Number must be  $\geq 0$ .

148      `SNRetaskGatherAmount`

The minimum amount that a gatherer must gather before the computer player allows him to be retasked to another resource type. Some code may override this. Must be  $\geq 0$ .

149      `SNMaxRetaskGatherAmount`

The maximum amount that a gatherer can be told to gather before being allowed to be retasked. Some code may override this. Must be  $\geq 0$ .

167      `SNInitialExplorationRequired`

The percentage of the map that must be explored by a computer player before any building is allowed. Must be  $\geq 0$  and  $\leq 100$ .

203      `SNUseByTypeMaxGathering`

Controls whether or not logical, type-specific gatherer requirements are placed on the quantity of resources gatherers must collect before being allowed to be retasked. Must be 0 or 1.

#### SNPercentHalfExploration

The percentage of map exploration that allows the computer player to task down to half the number of explorers. Must be  $\geq 0$ .

#### 204 SNMinimumElephantHuntGroupSize

The number of civilians a computer player must collect before allowing elephants to be hunted for food. Must be  $\geq 1$ .

### GROUP-RELATED NUMBERS

#### 40 SNGroupFillMethod

Sets the method by which a computer player fills a group of units. There are two concepts here: single group fill and level group fill. In single group fill, all available units are put into the first non-minimally full group. Once that group is full, the next group is entirely filled before the third group is considered. In level group fill, all groups are filled at the same time (one unit is placed in the first group, the next in the second, etc.). The single group fill fills up to the minimum in each group, then goes to the level group fill. Both methods respect the maximum group sizes. 0 keys the single group fill, and 1 keys the level group fill. Must be 0 or 1.

#### 75 SNGroupCommanderSelectionMethod

Sets the method by which group commanders are selected. 0 selects the unit with the most hit points. 1 selects the unit with the fewest hit points. 2 selects the unit with the most range. The commander is set once during a group's creation and is only reset when the commander dies. Must be  $\geq 0$  and  $\leq 2$ .

#### 76 SNConsecutiveIdleUnitLimit

Sets the number of consecutive seconds that pass before a group is set to idle if all of its units are idle. This is only used during attack and retreat phases. Must be  $\geq 0$ .

#### 143 SNTaskUngroupedSoldiers

Controls whether or not ungrouped computer player soldiers get tasked to spread out and guard the computer player's general town area. Must be 0 or 1.

### ATTACK GROUP NUMBERS

#### 36 SNNumberAttackGroups

Sets the desired number of land-based attack groups. Must be  $\geq 0$ .

#### 16 SNMinimumAttackGroupSize

Sets the minimum size of land-based attack groups. A group must meet its minimum size as one of the tasking prerequisites. Must be  $\geq 0$ .

#### 26 SNMaximumAttackGroupSize

Sets the maximum size of land-based attack groups. Must be  $\geq 0$  and  $\geq$  SNMinimumAttackGroupSize.

#### 93 SNScaleMinimumAttackGroupSize

The scaling factor for the minimum attack group size. Added to SNMinimumAttackGroupSize when the tactical AI does its scaling.

#### 94 SNScaleMaximumAttackGroupSize

The scaling factor for the maximum attack group size. Added to SNMinimumAttackGroupSize when the tactical AI does its scaling.

#### 98 SNAttackGroupSizeRandomness

The randomness factor in the attack group size. This sets a cap on the amount of randomness in the minimum attack group size. The randomness factor is set once (when the group is created) and will be between 0 and this number.

41      `SNAttackGroupGatherSpacing`

Controls the relative proximity (to the group gather point) that grouped units must be in before the group is considered gathered. Must be  $\geq 1$ .

30      `SNPercentHealthRetreat`

Sets the percentage of hit points that a group can lose (relative to what it started the attack with) before retreating. Must be  $\geq 1$  and  $\leq 100$ .

31      `SNPercentDeathRetreat`

Sets the percentage of units that a group can let die (relative to what it started the attack with) before retreating. Must be  $\geq 1$  and  $\leq 100$ .

91      `SNPercentUnitHealthRetreat` Sets the percentage of hit points that a unit can lose (relative to what it started the attack with) before retreating. Must be  $\geq 1$  and  $\leq 100$ .

95      `SNScalePercentHealthRetreat`

The scaling factor for the percent health retreat. Added to `SNPercentHealthRetreat` when the tactical AI does its scaling. Must be  $\geq -100$  and  $\leq 100$ .

96      `SNScalePercentDeathRetreat`

The scaling factor for the percent health retreat. Added to `SNPercentDeathRetreat` when the tactical AI does its scaling. Must be  $\geq -100$  and  $\leq 100$ .

97      `SNScalePercentUnitHealthRetreat`

The scaling factor for the percent health retreat. Added to `SNPercentUnitHealthRetreat` when the tactical AI does its scaling. Must be  $\geq -100$  and  $\leq 100$ .

121     `SNDefendImportantGroupLeaders`

Controls whether or not important attack group leaders are defended by the other group units. A value of 1 has the members defend the leader. A value of 0 does not.

131     `SNGroupLeaderDefenseDistance` Sets the defense distance for defenders of an important attack group leader. Must be  $\geq 1$ .

### MISCELLANEOUS ATTACK NUMBERS

46      `SNAttackSeparationTime`

Sets the amount of time that must pass between computer player attacks. Must be  $\geq 0$ .

47      `SNAttackCoordination`

Selects the type of coordination between computer player attacks. 0 means no coordination. 1 means that one group may attack at a time. 2 means that multiple groups may attack at the same time. Must be  $\geq 0$  and  $\leq 2$ .

19      `SNPercentEnemySightedResponse`

Sets the percentage of idle troops that will respond to another unit being attacked. Must be  $\geq 0$  and  $\leq 100$ .

20      `SNEnemySightedResponseDistance`

Sets the distance inside of which units will be candidates for response to an enemy attack. Must be  $\geq 0$  and  $\leq 144$ .

48      `SNAttackResponseSeparationTime`

Sets the amount of time that must pass before units respond to a subsequent enemy attack distress call. Must be  $\geq 0$ .

49      `SNRetreatAfterTargetDestroyed`

Selects what happens when a target is destroyed during an attack. 0 means that the attack group will never retreat and will recenter upon their current position. 1 means that the group will retreat if no other target is reachable. 2 means that the group will always retreat when the target is destroyed. 3 means that the group will go into extermination mode; they will explore unexplored territory and attack any enemies units they uncover. Must be  $\geq 0$  and  $\leq 3$ .

135      `SNBlotExplorationMap`

This controls whether or not the computer player re-explores previously explored regions. A value of 1 has the computer player re-explore, a value of 0 does not.

136      `SNBlotSize`

This controls the size of the area that a computer player marks for re-exploration. Must be  $> 0$  and  $<$  the map size.

71      `SNLockAttackAndAttackResponse`

This treats the `SNAttackSeparationTime` and `SNAttackResponseSeparationTime` as the same numbers. Must be either 0 or 1.

100      `SNMaximumGAIAAttackResponse`

The maximum number of civilians that respond to another civilian getting attacked by a Gaia animal. Must be  $\geq 0$ .

102      `SNAttackSeparationTimeRandomness`

The amount of randomness incorporated into the attack separation time. Must be  $\geq 0$  and  $<$  `SNAttackSeparationTime`.

103      `SNAttackIntelligence`

Specifies whether the intelligent attack system is used. The intelligent attack system tries to avoid enemy units when attacking and tries to attack from different sides. When used with the `SNAttackCoordination` set to 2, this can create multifront attacks. Must be 0 (to turn off) and 1 (to turn on).

104      `SNInitialAttackDelay`

The forced, initial delay before any computer player attacks (in seconds). Must be  $\geq 0$ .

134      `SNInitialAttackDelayType`

The type of initial attack delay. A value of 1 denotes a delay ended by the build list. A value of 2 uses the `SNInitialAttackDelay` timeout. A value of 3 allows the computer player to attack after he has been attacked by a non-Gaia player. A value of 0 allows any of the three occurrences to enable attacks.

### DEFEND GROUP NUMBERS

38      `SNNumberDefendGroups`

Sets the desired number of land-based defend groups. Must be  $\geq 0$ .

25      `SNMinimumDefendGroupSize`

Sets the minimum size of land-based defend groups. A group must meet its minimum size as one of the tasking prerequisites. Must be  $\geq 0$ .

28      `SNMaximumDefendGroupSize`

Sets the maximum size of land-based defend groups. Must be  $\geq 0$  and  $\geq$  `SNMinimumDefendGroupSize`.

50      `SNGoldDefendPriority`

Sets the priority of defending gold. A priority of 0 means that gold will not be defended. A priority of 1 means that gold has the highest defend priority. Must be  $\geq 0$  and  $\leq 7$ .

51      `SNStoneDefendPriority`

Sets the priority of defending stone. Must be  $\geq 0$  and  $\leq 7$ .

52      `SNForageDefendPriority`

Sets the priority of defending forage sites. Must be  $\geq 0$  and  $\leq 7$ .

54      `SNRuinsDefendPriority`

Sets the priority of defending Ruins. Must be  $\geq 0$  and  $\leq 7$ .

55      `SNArtifactDefendPriority`

Sets the priority of defending Artifacts. Must be  $\geq 0$  and  $\leq 7$ .

56      `SNTownDefendPriority`

Sets the priority of defending the computer player's town. Must be  $\geq 0$  and  $\leq 7$ .

57      `SNDefenseDistance`

Sets the distance at which items (town excluded) are defended. Must be  $\geq 0$ .

22      `SNSentryDistance`

Sets the distance at which the town is defended. Must be  $\geq 0$ .

72      `SNSentryDistanceVariation`

Sets the amount of allowable variation in the defense distances. Must be  $\geq 0$ .

92      `SNDefendOverlapDistance`

Sets the amount of influence that a defend group has. Defend groups will be assigned so that their circles of influence do not overlap. Must be  $\geq 0$ .

### EXPLORE GROUP NUMBERS

42      `SNNumberExploreGroups`

Sets the desired number of land-based soldier exploration groups. Must be  $\geq 0$ .

43      `SNMinimumExploreGroupSize` Sets the minimum size of land-based soldier exploration groups. A group must meet its minimum size as one of the tasking prerequisites. Must be  $\geq 0$ .

44      `SNMaximumExploreGroupSize`

Sets the maximum size of land-based soldier exploration groups. Must be  $\geq 0$  and  $\geq$  `SNMinimumExploreGroupSize`.

### BOAT ATTACK GROUP NUMBERS

58      `SNNumberBoatAttackGroups`

Sets the desired number of boat attack groups. Must be  $\geq 0$ .

59      `SNMinimumBoatAttackGroupSize`

Sets the minimum size of boat attack groups. A group must meet its minimum size as one of the tasking prerequisites. Must be  $\geq 0$ .

60      `SNMaximumBoatAttackGroupSize`

Sets the maximum size of boat attack groups. Must be  $\geq 0$  and  $\geq$  `SNMinimumBoatAttackGroupSize`.

### BOAT DEFEND GROUP NUMBERS

67      `SNNumberBoatDefendGroups`

Sets the desired number of boat defend groups. Must be  $\geq 0$ .

68      `SNMinimumBoatDefendGroupSize`

Sets the minimum size of boat defend groups. Must be  $\geq 0$ .

69      `SNMaximumBoatDefendGroupSize`

Sets the maximum size of boat defend groups. Must be  $\geq 0$  and  $\geq$  `SNMinimumBoatDefendGroupSize`.

70       `SNDockDefendPriority`

Sets the priority of defending a Dock. 0 does not protect Docks, 1 does. Must be either 0 or 1.

### **BOAT EXPLORE GROUP NUMBERS**

61      `SNNumberBoatExploreGroups`

Sets the desired number of boat exploration groups. Must be  $\geq 0$ .

62      `SNMinimumBoatExploreGroupSize`

Sets the minimum size of boat exploration groups. Must be  $\geq 0$ .

63      `SNMaximumBoatExploreGroupSize`

Sets the maximum size of boat exploration groups. Must be  $\geq 0$  and  $\geq$  `SNMinimumBoatExploreGroupSize`.

64      `SNDesiredNumberTradeEscorts`

Sets the desired number of warboat escorts for tradeboats. Must be  $\geq 0$ .

65      `SNDesiredNumberFishEscorts`

Sets the desired number of warboat escorts for fishing boats. Must be  $\geq 0$ .

66      `SNDesiredNumberTransportEscorts`

Sets the desired number of warboat escorts for transports. Must be  $\geq 0$ .

### **TOWN BUILDING NUMBERS**

73      `SNMinimumTownSize`

Sets the minimum size of a computer player town. Must be  $\geq 0$ .

74      `SNMaximumTownSize`

Sets the maximum size of a computer player town. Must be  $\geq 0$  and  $\geq$  `SNMinimumTownSize`.

86      `SNStoragePitMaxDistance`

Sets the maximum distance that Storage Pits may be placed from a Town Center. Must be  $\geq 0$ .

87      `SNGranaryMaxDistance`

Sets the maximum distance that Granaries may be placed from a Town Center. Must be  $\geq 0$ .

84      `SNNumberWallGates`

Sets the number of gates that must be left in the wall around a computer player's town. Must be  $\geq 0$ .

85      `SNSizeWallGates`

Sets the size (in tiles) of the gates in the wall around a computer player's town. Must be  $\geq 0$ .

112     `SNMinimumWaterBodySizeForDock`

The minimum number of tiles (in surface area) that a body of water must be for a Dock to be placed on it. Must be  $\geq 10$ .

205     SNAutoBuildDropsites

Controls whether or not the computer player decides how and when to build storage pits and granaries. Must be 0 or 1.

206     SNAutoBuildFarms

Controls whether or not the computer player can use extra wood to build Farms once all of the build list buildings are constructed. Must be 0 or 1.

207     SNAutoBuildTowers

Controls whether or not the computer player can use extra stone to build towers. Must be 0 or 1.

208     SNAutoBuildDocks

Controls whether or not the computer player decides how and when to build Docks. Must be 0 or 1.

209     SNAutoBuildFishingBoats

Controls whether or not the computer player decides how and when to build fishing boats. Must be 0 or 1.

210     SNAutoBuildTransports

Controls whether or not the computer player decides how and when to build transports. Must be 0 or 1.

223     SNAutoBuildWarships

Determines if the computer player is allowed to decide how and when to build warships. Must be 0 or 1.

212     SNDesiredNumberDocks

How many Docks the computer player wants in a given game. Must be  $\geq 0$ .

213     SNDesiredNumberFishingBoats

How many fishing boats the computer player wants in a given game. Must be  $\geq 0$ .

214     SNDesiredNumberTransports

How many transports the computer player wants in a given game. Must be  $\geq 0$ .

224     SNDesiredNumberWarships

How many warships the computer player wants in a given game. Must be  $\geq 0$ .

174     SNMaximumHousesBeforeDropsites

The maximum number of Houses that can be built before a dropsite is built. Must be  $\geq 0$ .

175     SNSpecificBuildItemToBuild

A specific build item that should be inserted into the computer player's list. Must be a valid build ID.

176     SNSpecificBuildItemTime

The time (in minutes) that the SNSpecificBuildItemToBuild should be inserted into. Must be  $\geq 0$ .

177     SNUnskippableItemType

Allows the computer player to not skip a particular item type during building. Must be a valid build ID.

150     SNMaxStoragePits

The maximum number of storage pits a computer player can build in one game. Must be  $\geq 0$ .

151     SNMaxGranaries

The maximum number of granaries a computer player can build in one game. Must be  $\geq 0$ .

152     SNHouseOverage



The number of Houses that a computer player will autobuild over the amount needed to support 50 units. Must be  $\geq 0$ .

155    `SNBuildPlanDivisions`

The number of divisions a computer player will place in its build list (used to calculate gathering percentages). Must be  $\geq 1$ .

160    `SNMaxBuildPlanGathererPercentage`

The maximum percentage of gatherers that a computer player will task based on of the pregame requirements of the build plan. Must be  $\geq 0$  and  $\leq 100$ .

161    `SNRequiredFirstBuilding`

Controls what building a computer player must place first. 0: No restriction, 1: Either a Storage Pit or a granary, 2: A Storage Pit, 3: A Granary, 4: Both a Storage Pit and a Granary. Exceptions consist of a Town Center and `SNMaximumHousesBeforeDropsites`.

163    `SNFoodDropsiteDistance`

The maximum number of tiles a computer player likes to walk to drop off its food. Must be  $\geq 3$ .

164    `SNWoodDropsiteDistance`

The maximum number of tiles a computer player likes to walk to drop off its wood. Must be  $\geq 3$ .

165    `SNStoneDropsiteDistance`

The maximum number of tiles a computer player likes to walk to drop off its stone. Must be  $\geq 3$ .

166    `SNGoldDropsiteDistance`

The maximum number of tiles a computer player likes to walk to drop off its gold. Must be  $\geq 3$ .

220    `SNMaxFarms`

Caps the number of Farms a computer player will build. Must be  $\geq 0$ .

222    `SNMaxTowers`

Caps the number of towers a computer player will build. Must be  $\geq 0$ .

202    `SNMinimumDropsiteBuffer`

Controls how far away a computer player will keep dropsites in relation to enemy Town Centers. Must be 0 or 1.

168    `SNRandomPlacementFactor`

A number that gets added into the placement of the computer player to randomize building placement (off of the calculated ideal). Must be  $\geq 0$ .

169    `SNMinimumForestTiles`

The minimum number of forest tiles that a computer player must uncover before placing its first storage pit. Must be  $\geq 0$ .

180    `SNAutoBuildHouses`

Controls whether the computer player can decide to build its houses by itself. Must be 0 or 1.

181    `SNUpgradeToToolAgeASAP`

Controls whether or not the computer player will abandon all to upgrade to the Tool Age as soon as it becomes available for research. Must be 0 or 1.

182    `SNUpgradeToBronzeAgeASAP`

Controls whether or not the computer player will abandon all to upgrade to the Bronze Age as soon as it becomes available for research. Must be 0 or 1.

183    `SNUpgradeToIronAgeASAP`

Controls whether or not the computer player will abandon all to upgrade to the Iron Age as soon as it becomes available for research. Must be 0 or 1.

### TARGET EVALUATION NUMBERS

77      `SNTargetEvaluationDistance`

Sets the multiplier used for the distance rating in computer player target evaluation. Must be  $\geq 0$ .

78      `SNTargetEvaluationHitpoints`

Sets the multiplier used for the hitpoint rating in computer player target evaluation. Must be  $\geq 0$ .

79      `SNTargetEvaluationDamageCapability`

Sets the multiplier used for the damage capability rating in computer player target evaluation. Must be  $\geq 0$ .

80      `SNTargetEvaluationKills`

Sets the multiplier used for the kill rating in computer player target evaluation. Must be  $\geq 0$ .

81      `SNTargetEvaluationAllyProximity`

Sets the multiplier used for the ally proximity (number of allies in range) rating in computer player target evaluation. Must be  $\geq 0$ .

82      `SNTargetEvaluationROF`

Sets the multiplier used for the rate of fire rating in computer player target evaluation. Must be  $\geq 0$ .

83      `SNTargetEvaluationRandomness`

Sets the multiplier used for the randomness factor in computer player target evaluation. Must be  $\geq 0$ .

89      `SNTargetEvaluationAttackAttempts`

Sets the multiplier used for the attack attempts rating in computer player target evaluation. Must be  $\geq 0$ .

90      `SNTargetEvaluationRange`

Sets the multiplier used for the range rating in computer player target evaluation. Must be  $\geq 0$ .

106     `SNSpecialAttackType1`

Sets the type of object (first slot) that the computer player particularly wants to attack. Must be a valid master object ID or  $-1$  if no special attack type is desired.

109     `SNSpecialAttackInfluence1`

Sets the multiplier used for the special attack type 1 rating in computer player target evaluation. Must be  $> 0$  to influence the computer player toward attacking the special type 1,  $< 0$  to influence the computer player away from attacking the special type 1.

107     `SNSpecialAttackType2`

Sets the type of object (second slot) that the computer player particularly wants to attack. Must be a valid master object id or  $-1$  if no special attack type is desired.

110     `SNSpecialAttackInfluence2`

Sets the multiplier used for the special attack type 2 rating in computer player target evaluation. Must be  $> 0$  to influence the computer player toward attacking the special type 2,  $< 0$  to influence the computer player away from attacking the special type 2.

108     `SNSpecialAttackType3`

Sets the type of object (third slot) that the computer player particularly wants to attack. Must be a valid master object ID or  $-1$  if no special attack type is desired.

111 SNSpecialAttackInfluence3

Sets the multiplier used for the special attack type 3 rating in computer player target evaluation. Must be > 0 to influence the computer player toward attacking the special type 3, < 0 to influence the computer player away from attacking the special type 3.

122 SNTargetEvaluationContinent

Sets the additive value used for the targets on the same continent as the attack group commander. Must be > 0 to influence the computer player toward attacking the units on the same continent or 0 for no special influence.

123 SNTargetEvaluationSiegeWeapon

Sets the additive value used for influencing siege weapons to attack stationary targets (and influencing non-siege weapons not to attack those stationary targets). Must be > 0 to influence the computer player to use siege weapons to attack stationary targets or 0 for no special influence.

144 SNTargetEvaluationBoat

Sets the additive value used for influencing land units to attack or not attack boats. Must be > 0 to influence land units to attack boats, 0 for no special influence, and less than 0 for aversion.

184 SNTargetEvaluationTimeKillRatio

The amount of influence the time to kill a target has in deciding what to attack. Must be >= 0.

185 SNTargetEvaluationInProgress

The amount of influence of continuing to attack a target already under attack. Must be >= 0.

### COMPUTER PLAYER TRIBUTE NUMBERS

124 SNTributeAmount

Sets the required amount of gold that must be tributed to the computer player. If this is 0, no computer player tributing is monitored for the special system (thus, the normal tribute system is in effect). When this value is > 0, the computer player will expect some gold in tribute.

125 SNTributeChatFrequency

Sets the frequency (in seconds) of the computer player's chat messages asking for tribute. Must be >= 0.

126 SNTributeChatRandomness

Sets the randomness (in seconds) of the computer player's chat messages asking for tribute. Must be >= 0 and < SNTributeChatFrequency.

127 SNTributeTimeout

Sets the amount of time (in seconds) within which the tribute must take place. Must be >= 0.

128 SNTributePlayer

Sets the player ID# of the player the computer player will target for the tribute request. Must be a valid player number for the game.

129 SNTributeSuccessOutcome

Controls what happens when the tribute request is fulfilled within the allotted time. If set to 0, nothing happens. If set to 1, the computer player will ally when the tribute amount is paid.

130 SNTributeFailureOutcome

Controls what happens when the tribute request is not fulfilled within the allotted time. If set to 0, nothing happens. If set to 1, the computer player will go to war when the tribute amount is not paid.

132 SNTributePersistence

Controls whether or not the interactive tribute system operates once or forever. A value of 1 makes it last the entire game (i.e. the computer player will continue to demand `SNtributeAmount` of gold for the entire game, at intervals roughly equivalent to `SNtributeChatFrequency` from the outcome evaluation). A value of 0 makes the interactive tribute occur one time.

133 `SNtributeRevokeOnAttack`

Controls whether or not the computer player rescinds the favorable tribute outcome when the `SNtributePlayer` attacks the computer player. A value of 1 has the computer player rescind, a value of 0 does not.

### MISCELLANEOUS NUMBERS

23 `SNartifactReturnDistance`

Sets the distance that Artifacts must be within to be considered returned to the Town Center. Must be  $\geq 0$ .

29 `SNminimumPeaceLikeLevel`

Sets the level at which computer players must like another player before allying with that player. Must be  $\geq 0$  and  $\leq 100$ .

32 `SNpercentExplorationRequired`

Sets the minimum amount of exploration that a computer player must have accomplished before being allowed to retask civilian explorers. Must be  $\geq 0$  and  $\leq 100$ .

34 `SNzeroPriorityDistance`

Sets the distance at which a computer player's order for a unit is set to a priority of 0. Must be  $\geq 0$  and  $\leq 144$ .

88 `SNtacticalUpdateFrequency`

Sets the number of seconds that pass between each tactical AI update. Must be  $\geq 0$ .

99 `SNscalingFrequency`

Sets the number of minutes that pass between each scaling inside the tactical AI. Must be  $\geq 0$ .

101 `SNbuildFrequency`

Sets the number of tactical AI updates that pass between each training or research attempt. Must be  $\geq 0$ .

105 `SNsaveScenarioInformation`

Controls whether the learning information is saved at the end of the scenario for a given computer player. Must be 0 (to turn off) or 1 (to turn on).

114 `SNnumberBuildAttemptsBeforeSkip`

The maximum number of build attempts a build plan can go through before being put into skip mode. Must be  $\geq 1$ .

115 `SNmaxSkipsPerAttempt`

The maximum number of unbuilt items that can be skipped during any build plan processing before giving up (for being too far ahead of the current position in the plan). Must be  $\geq 1$ .

170 `SNminimumFood`

The minimum amount of food a computer player likes to keep on hand. Must be  $\geq 0$ .

171 `SNminimumWood`

The minimum amount of wood a computer player likes to keep on hand. Must be  $\geq 0$ .

172 `SNminimumStone`

The minimum amount of stone a computer player likes to keep on hand. Must be  $\geq 0$ .

173     SNMinimumGold

The minimum amount of gold a computer player likes to keep on hand. Must be  $\geq 0$ .

190     SNMaximumFood

Controls the maximum amount of food the computer player likes to have on hand. Must be  $\geq 0$ .

191     SNMaximumWood

Controls the maximum amount of wood the computer player likes to have on hand. Must be  $\geq 0$ .

192     SNMaximumStone

Controls the maximum amount of stone the computer player likes to have on hand. Must be  $\geq 0$ .

193     SNMaximumGold

Controls the maximum amount of gold the computer player likes to have on hand. Must be  $\geq 0$ .

215     SNAllowDiplomacyChangeOnAllyAttack

Controls whether or not the computer player can change his alliance when attacked by an ally. Must be 0 or 1.

216     SNMinimumAmountForTrading

Controls how much of a resource a computer player must have before using it for trading. Must be  $\geq 0$ .

217     SNAllowDiplomacyChangeOnTribute

Controls whether or not the computer player will allow his diplomacy to change when he receives tribute from a player. Must be 0 or 1.

221     SNHitsBeforeAllianceChange

Sets the number of times a computer player will allow his units to be hit by an ally before allowing his diplomacy to be changed. Must be  $\geq 0$ .

178     SNAttackDiplomacyImpact

The impact (positive or negative) that a computer player injects into his diplomacy system when attacked. Must be  $\geq 0$  and  $\leq 100$ .

218     SNEasiestReactionPercentage

Sets the effective reaction percentage (of normal LOS) a computer player unit will use in single player Easiest level scenario or campaign games. Must be  $\geq 0$  and  $\leq 100$ .

219     SNEasierReactionPercentage

Sets the effective reaction percentage (of normal LOS) a computer player unit will use in single player easier scenario or campaign games. Must be  $\geq 0$  and  $\leq 100$ .

201     SNTrackPlayerHistory

Decides whether or not a human player's tendencies are tracked or not. Must be 0 or 1.

188     SNAttackWinningPlayer

Controls whether or not the computer player will attack the winning player (if there is more than one to choose from). Must be 0 or 1.

194     SNCoopShareInformation

Controls whether or not allied computer players share information about what they uncover (this is not like Writing; instead, it's analogous to two humans chatting). Must be 0 or 1.

195     SNAttackWinningPlayerFactor

The influence the `SNAttackWinningPlayer` will have on deciding who to attack if it's set to 1. Must be  $\geq 0$  and  $\leq 100$ .

196     SNCoopShareAttacking

Controls whether allied computer players can attack to defend each other. Must be 0 or 1.

197     SNCoopShareAttackingInterval

Controls how often this computer player can ask another for help (in seconds). Must be  $\geq 0$ .

198     SNPercentageExploreExterminators

Determines how many of the computer player's soldier explore groups are set as extermination groups. Must be  $\geq 0$  and  $\leq 100$ .

186     SNCoopDemandTributeInterval

Controls how often the computer player may demand tribute from his computer player allies (in seconds). Must be  $\geq 0$ .

187     SNCoopDemandTributeMaximum

Controls the maximum amount a computer player may demand from his computer player allies at any one time. Must be  $\geq 1$ .