

Manual for mapconfig v1.2

This manual provides information about how to make your own Reanimation configuration file for your map. With this file, it's possible to customize a few things on the mod. Open "rea_mapconfig.lua" from the copied "Reanimation_mod" folder with notepad. Let's see what we can change:

-- Town Conversion

reaTownConversion: defines Town Conversion. It can be 0 or 1. If 0, it means none of the players can convert any town on the map.

reaTownConversionCreatures: defines how many interior dwelling creatures can be converted. The value is in percent. For example if 50, it means half of the creatures will be converted. The script takes notice of the weekly growth of the creatures.

reaTownConversionCostGlobal: defines the cost of the town conversion for all races for a town that is on lvl1. Resources in succession: Wood, Ore, Mercury, Crystal, Sulfur, Gem, Gold

reaTownConversionCostAcademy, reaTownConversionCostDungeon, etc.: defines the cost of the town conversion for the specified race for a town that is on lvl1. The cost will be added to reaTownConversionCostGlobal.

reaTownConversionCostLevel: defines how much the price will grow for each level of the town. The value is in percent.

reaTownConversionDays: defines how many days will pass until the conversion for a town that is on lvl1.

reaTownConversionDaysLevel: defines how many days will pass until the conversion for each level of the town. The value will be added to reaTownConversionDays.

reaTownConversionLost: defines that the player can lose the conversion(1) or not(0) when another player captures the town before the conversion process ends.

reaTownConversionLostCost: defines how many resources will get back the player when he loses the conversion. The value is in percent.

reaTownConversionDisableDays: defines how many days will pass until a player can launch a new conversion process after a town that is on lvl1 has been converted.

reaTownConversionDisableDaysLevel: defines how many days will pass until a player can launch a new conversion process after the town has been converted, for each level of the town. The value will be added to reaTownConversionDisableDays.

-- Dwelling Conversion

reaDwellingConversionCostGlobal, reaDwellingConversionCostAcademy, etc.: Same as for town conversion.

reaDwellingConversionCostLevel: defines how much the price will grow for each level of a dwelling. In this case, military posts are fourth level dwellings. The value is in percent.

reaDwellingConversionDisableMode: defines that the dwelling's conversion won't be allowed for the current week, after a successful conversion. It can be 0, 1 or 2. If 0, it means the script won't disable the conversion of the dwelling. If 1, then none of the players can convert the dwelling, if it is outside of an area. If 2, then it doesn't matter if the dwelling is in an area or not.

-- Areas

reaAreas: defines Area creation. It can be 0, 1, 2 or 3. If 0, it means areas are not allowed. Only mode 1 is available for mapscripts.

reaTownAreaRadius: defines town area radius in tiles. It is useless in area mode 1.

reaAreaOfControl: defines Area of Control. It can be 0 or 1. If 0, it means area of control is not allowed.

-- Town Gate

reaTownGate: defines Town Gate function. It can be 0 or 1. If 0, it means town gate is not allowed.

reaTownGateMode: defines which town gate mode will be used. It can be 1, 2 or 3.

Mode 1: teleports the hero from a town to a selected town.

Mode 2: teleports the hero from a town's area to a selected town.

Mode 3: teleports the hero from a random location to a selected town.

reaTownGateToAllies: defines that the player can launch a town gate to ally towns (1) or not (0).

reaTownGateTownRequirements: defines that the town gate function has some requirements for the town. It can be 0, 1 or 2.

If 0, it means no requirements for the town. If 1, it means the requirements belong to the starter town, if 2, then to the target town.

reaTownGateTownLevel: defines town level requirement of the town gate function. It can be 0.

reaTownGateOrbs: defines how many orbs will be used for a town gate. It can be 0.

reaTownGateOrbsStart: defines how many orbs will be generated for every town at the beginning of the map. It can be 0.

reaTownGateOrbsMax: defines how many orbs can be stored by a town.

reaTownGateOrbsNext: defines when the town gets the next orb. The value is in days. It can be 0.

reaTownGateHeroLevel: defines hero level requirement of the town gate function. It can be 0.

reaTownGateTownPortalSpell: defines that the town gate function requires Town Portal Spell from the hero (1) or not (0).

reaTownGateMana: defines mana requirement of the town gate function. It can be 0.

reaTownGateManaAll: defines that the hero lose all of his/her mana (1) or not (0) after a town gate.

reaTownGateMovementPoints: defines movement points requirement of the town gate function. It can be 0.

reaTownGateMovementPointsAll: defines that the hero lose all of his/her movement points (1) or not (0) after a town gate.

reaTownGateDisableDays: defines how many days will pass until the player can launch a new town gate, after a successful town gate.

reaDisableRaces: defines available races in town taverns. It can be 0 or 1. If 1, it means every player can only hire heroes from his own race at towns.

After the changes, save your file and close it. It is possible to make another mapconfig file, to force the mod to use the values in that (useful if the player can choose from customized configuration files).

To do this, make a copy of "rea_mapconfig.lua", and rename the new file to "rea_mapconfig_forced.lua". Open this new file and delete everything from it, that you don't want to force to the mod. It is also possible to use only the "forced" config without the mapconfig, or none of them.