

Dominions 6 Event Modding Manual

Illwinter Game Design (ver 6.11)

Introduction	2
Event Codes	2
Basic Event Modding	2
Event Requirements	2
Generic Requirements	2
Nation Requirement	3
Treasury Requirements	3
Province Requirements	3
Site Requirements	4
Dominion Requirements	5
Scale Requirements	5
Monster Requirements	6
Mage Requirements	6
Target Requirements	7
Event Code Requirements	9
Enchantment Requirements	9
Event Variable Requirements	9
Misc Requirements	9
Event Effects	9
Basic Effects	10
Treasure Effects	10
Scale Effects	11
Province Effects	11
Monster Effects	12
Magic Pathboost Effects	14
Global Event Effects	15
Special Event Commands	16
Event Code Effects	16
Event Variables	17
Misc Event Effects	17
Examples	17
Basic Event	17
Complex Event	17
Event Chain	17
Simple Global Event	18
Advanced Global Event	18

Introduction

Modding events works very similar to modding other things in Dominions and you enter the events commands in the same file as the other mod commands in your mod and the same minimum requirements apply. You should know the basics of how to mod nations and units before doing the more advanced event modding.

Event Codes

Events can create numeric codes tied to the province they occur in. These codes can then be requirements, enabling specific follow-up events.

If the same event code is used by different events (for example in two different mods), this will scramble event chains and create potentially severe bugs and other strangeness. To limit this problem, please use only negative event numbers -300 to -5000. Avoid all other event codes except 0, which is the default in all provinces and can be used to reset the event chain in a province. This range of event codes should probably be split between different mods, so that conflicting code can be avoided. If multiple event mods are used in a game, they should be carefully checked for compatibility first.

Basic Event Modding

#clearallevents

This command removes all events from the game. You usually don't want to use this command.

#newevent

This command creates a new event. All subsequent commands will modify this event until modding the event is finished with the #end command.

#end

Always use this command at the end of modifying an event.

#rarity <nbr>

Sets the rarity of the event. Normal events should use number 1, 2, -1, -2, 11 or 12. The other numbers are always events that are best used for creating global enchantment effects or special dominion effects.

The always events will occur every month in every province unless they are limited somehow. The normal always event uses rarity 0 and only one rarity 0 event can occur in each province per month. Rarity 5 is like rarity 0, except that any number of rarity 5 events can occur in a single province each month. Usually rarity 5 should only be used for global enchantment effects.

The global events are planned 7 months before they are executed, except for number 13 that happens immediately.

This delay is necessary because fortune tellers should be able to foresee them. Otherwise the events will appear by random chance. Event rarities are listed in the table below. Certain event effects can modify the event calendar.

Nbr	Rarity
0	always
1	common bad
2	uncommon bad
5	always (unlimited)
-1	common good
-2	uncommon good
10	always global
11	common global
12	uncommon global
13	always immediate global

Event Requirements

These commands set the requirements needed for an event to happen.

Generic Requirements

#req_rare <percent>

The event is extra rare and only has a <percent> chance of being valid when rolled. An always event can be combined with #req_rare to set its chance of happening (per province).

#req_unique <value>

The event can only happen <value> times during the course of the game.

#req_story < 0 | 1 >

Event only happens when story events are enabled. 1 = yes, 0 = no.

#req_indepok < 0 | 1 >

Event can also happen to independents. 1 = yes, 0 = no.

#req_era <value>

Can only happen in this era. 1 = early, 2 = mid, 3 = late

#req_noera <value>

Cannot happen in this era.

#req_turn <value>

The event cannot happen before this turn.

#req_maxturn <value>

The event cannot happen after this turn.

#req_pregame <value>

The event can happen before the first turn, right after the game has been created, but it will never occur otherwise. The message from this events will show up on the first turn of the game. There are no events like this in the standard game (unless you count the two introduction messages), but it can be used to create special

start conditions for mods. These events must have an 'always' rarity as no normal events will occur before the first turn.

#req_season <0 - 4>

Event requires this season to happen.

0 = spring

1 = summer

2 = fall

3 = winter

Season numbers

#req_noseason <0 - 4>

The event cannot happen in this season.

#req_month <0 - 11>

Month of the year required. E.g. 4 = mid summer, 10 = mid winter.

#req_ai <0 | 1>

The event can only occur for AI players (0=cannot).

Nation Requirement

#req_nation <nation number>

This nation must be in play for the event to happen.

#req_nonation <nation number>

If this nation is in play the event cannot occur.

#req_fornation <nation number>

The event can only happen for this nation. This command can be used multiple times for the same event to enable it for multiple nations.

#req_notfornation <nation number>

This nation cannot receive the event. The old variant "req_notnation" can also be used for the same effect.

#req_notforally <nation number>

This nation and its allies cannot receive the event. In a non-disciple game this will have the same effect as req_notfornation.

Treasury Requirements

#req_gem <type>

There must be at least one gem of the required type in the gem treasury. Gem types are listed in the table below.

Nbr	Gem Type
0	Fire
1	Air
2	Water
3	Earth
4	Astral
5	Death
6	Nature
7	Glamour
8	Blood slave

Gem Types

#req_gold <type>

The nation must have at least this much gold in the treasury.

Province Requirements

#req_capital < 0 | 1 >

Event can only happen in capital provinces. This includes conquered capital provinces. 1 = only in capitals, 0 = never in capitals

#req_nearbycapital < 0 | 1 >

Event can only happen in capital provinces or provinces next to a capital. This includes conquered capital provinces. 1 = only near capitals, 0 = never near capitals

#req_owncapital < 0 | 1 >

Event can only happen in event owner's own capital province. 1 = only in event owner's capital, 0 = never in event owner's capital

#req_poptype <value>

The province must have the correct poptype.

#req_notpoptype <value>

The province must not have this poptype.

#req_pop0ok

The event can also happen in provinces with no population. Standard events are limited to provinces with at least 50 population (5 dekapop).

#req_maxpop <dekapop>

There may not be more than this amount of population in the province. One dekapop = 10 population in game.

#req_minpop <dekapop>

There must be at least this much population in the province.

#req_mindef <value>

There must be at least <value> PD in the province for the event to happen. The value varies with +/- 25%.

#req_maxdef <value>

There may not be more than <value> PD in the province for the event to happen. The value varies with +/- 25%.

#req_minunrest <value>

There must be at least <value> unrest in the province for the event to happen. The value varies with +/- 25%.

#req_maxunrest <value>

There cannot be more than <value> unrest in the province for the event to happen. The value varies with +/- 25%.

#req_lab < 0 | 1 >

There must be a lab in the province. 1 = yes, 0 = no

#req_temple < 0 | 1 >

There must be a temple in the province. 1 = yes, 0 = no

#req_fort < 0 | 1 >

There must be a fort in the province. 1 = yes, 0 = no

#req_land < 0 | 1 >

Event requires land province. 1 = land, 0 = sea.

#req_plane < planenr >

Event can only occur on this plane. 0 = default plane (demiurgia), 1+ = other map planes, -1 = cave plane, -2 = nexus plane. If no custom map with extra planes is used, 1 will be the same as the cave plane and 2 will be the same as the nexus plane.

#req_coast < 0 | 1 >

Event requires the province to have coastal terrain. 1 = must be coast, 0 = cannot be coast.

#req_mountain < 0 | 1 >

Event requires the province to have mountain or border mountain terrain. 1 = must be mountain, 0 = cannot be mountain.

#req_forest < 0 | 1 >

Event requires the province to have forest terrain. 1 = must be forest, 0 = cannot be forest.

#req_farm < 0 | 1 >

Event requires the province to have farm terrain. 1 = must be farm, 0 = cannot be farm.

#req_swamp < 0 | 1 >

Event requires the province to have swamp. 1 = must be swamp, 0 = cannot be swamp.

#req_waste < 0 | 1 >

Event requires the province to have waste terrain. 1 = must be waste, 0 = cannot be waste.

#req_cave < 0 | 1 >

Event requires the province to have cave terrain. 1 = must be cave, 0 = cannot be cave.

#req_freshwater < 0 | 1 >

Event requires the province to have fresh water. 1 = must have fresh water, 0 = cannot have fresh water.

#req_mincorpses <dead>

There must be at least this number of unburied corpses in the province.

#req_maxcorpses <dead>

There mustn't be more than this number of unburied corpses in the province.

#req_nativesoil

Event must be on land for most nations or underwater if it regards an underwater nation.

#req_void < 0 | 1 >

Event can only take effect on the void plane (0 = cannot).

#req_voidok < 0 | 1 >

Event can also take effect on the void plane (0 = cannot, which is also the default).

Site Requirements

For site requirement commands with the attribute of < 0 | 1 >, the sitename of the site in question should be written within brackets at the end of the event message text. For example:

```
#msg "A clutch of eggs were discovered in the volcano.
[Volcano]"
```

The spelling must be exactly like the site is spelled in the game. A 0 instead of a 1 as value indicates that the requirement is inverted, e.g. the site must not be present and discovered.

#req_freesites <value>

The province must have at least <value> free site slots for the event to happen. Remember to use this requirement if the event creates a site.

#req_nositenbr <site number>

There cannot be this site in the province. If there is, the event cannot happen. This command only accepts the site number as an attribute.

#req_foundsite < 0 | 1 >

Event requires "sitename" to be in the province and discovered.

#req_hiddensite < 0 | 1 >

Event requires "sitename" to be in the province and not discovered.

#req_site < 0 | 1 >

Event requires "sitename" to be in the province. It can be discovered or hidden.

#req_nearbysite < 0 | 1 >

Event requires "sitename" to be in the province or in adjacent provinces.

#req_claimedthrone < 0 | 1 >

The throne "sitename" must be claimed by someone. Combine with #req_site to target the throne province.

#req_unclaimedthrone < 0 | 1 >

The throne "sitename" must be present in the game and unclaimed. Combine with #req_site to target the throne province.

#req_thronesite < 0 | 1 >

Event requires a throne (any type) to be in the province.

#req_nearbythrone < 0 | 1 >

Event requires a throne to be in the province or in adjacent provinces.

Dominion Requirements

#req_fullowner <nation number>

The event requires this nation to control the province, any fort (if present) and to have positive dominion in the province.

#req_domowner <nation number>

The event requires this nation to have dominion in the province.

#req_mydominion < 0 | 1 >

Event requires province owner's dominion. 0 = must not have owner's dominion.

#req_dominion <0 - 10>

The province must have the owner's dominion at <value> candles or more.

#req_maxdominion <-10 - 10>

The province must have a dominion strength of no more than <value> for the event to happen. Negative values can be used and mean that enemy dominion of that strength or stronger is required for the event to happen.

#req_domchance <0 - 100>

Requires a dominion and the higher dominion the higher the chance of this event happening. The owner of the dominion, friendly or not, does not matter for this command. The event will have dominion value * this value % chance of happening (if it should have happened without this command that is). This command can be used to create global enchantments that have the chance of something happening scale with dominion strength.

#req_godismnr "name" | <number>

Requires the event owner's Pretender God (or Disciple) to be a specific monster type. If a pretender can change between different shapes, this command will trigger if it matches anyone of the possible shapes. This command can be used multiple times, in which case only one of the monster types is required.

#req_godisnotmnr "name" | <number>

Requires the event owner's Pretender God (or Disciple) to not be this specific monster type. This command can be used multiple times, in which case the god mustn't be any of these types.

#req_godawake <0 - 1>

1 = Requires the event owner's Pretender God (or Disciple) to be alive and on the map somewhere.

#req_pretismnr "name" | <number>

Requires the event owner's Pretender God to be a specific monster type. If the event owner is a disiple player, the team's pretender will be used instead. This command can be used multiple times, in which case only one of the monster types is required.

#req_pretawake <0 - 1>

1 = Requires the event owner's Pretender God to be alive and on the map somewhere. If the event owner is a disiple player, the team's pretender will be used instead.

Scale Requirements

Events with scale requirements can only trigger if the province has the required scale at the required level. The value for scale requirements can range from -3 to 3. When a negative number is used as an argument, the event can happen in the opposite scale as well.

For example, #req_growth -1 would allow the event to happen if the province has a Growth / Death scale of Death 1, neutral or any level of Growth, but cannot happen if the scale value is Death 2 or Death 3.

#req_chaos <value>

Turmoil scale required.

#req_lazy <value>

Sloth scale required.

#req_cold <value>

Cold scale required.

#req_death <value>

Death scale required.

#req_unluck <value>

Misfortune scale required.

#req_unmagic <value>

Drain scale required.

#req_order <value>

Order scale required.

#req_prod <value>

Production scale required.

#req_heat <value>

Heat scale required.

#req_growth <value>

Growth scale required.

#req_luck <value>

Luck scale required.

#req_magic <value>

Magic scale required.

Monster Requirements

The argument of "name" | <number> means monster name or monster number.

#req_commander < 0 | 1 >

There must be a non-sneaking commander present for the event to happen. 1 = yes, 0 = no.

#req_monster "name" | <number>

This monster must be present in the province.

#req_2monsters "name" | <number>

This monster must be present in the province and there must be at least 2 of them.

#req_5monsters "name" | <number>

This monster must be present in the province and there must be at least 5 of them.

#req_nomonster "name" | <number>

This monster cannot be present in the province.

#req_mnr "name" | <number>

This monster must be present somewhere in the world or the event cannot happen. The command can be used multiple times and then all of the monsters must be in the world.

#req_nomnr "name" | <number>

This monster must not be present in the world or the event cannot happen. This requirement is best used with unique monsters or national heroes. This command can be used multiple times and then none of the monsters may be in the world.

For example, #req_nomnr 563 would require the Queen of Storms to not have been summoned yet. If she has already been summoned, the event can no longer happen. If she is killed, the event can happen again until she is summoned once more.

#req_deadmnr "name" | <number>

This monster must be present or be among the dead monsters or the event cannot happen. Unique monsters and heroes stay among the dead monsters when they are killed, ordinary monsters are purged from memory when they are killed. Use this command to see if a unique monster has ever been summoned to the game.

#req_mintroops <value>

There must be at least this many troops in the province for the event to happen.

#req_maxtroops <value>

There may not be more than this many troops in the province for the event to happen.

#req_humanoidres

There must be a humanoid researcher present in the province for the event to happen.

#req_researcher

There must be a researcher present in the province for the event to happen.

#req_preach <value>

Event has a chance of <value> per priest level of happening when a priest is present and preaching the teachings of his god.

Mage Requirements

#req_pathfire <level>

There must be a Fire mage of at least this level in the province.

#req_pathair <level>

There must be an Air mage of at least this level in the province.

#req_pathwater <level>

There must be a Water mage of at least this level in the province.

#req_pathearth <level>

There must be an Earth mage of at least this level in the province.

#req_pathastral <level>

There must be an Astral mage of at least this level in the province.

#req_pathdeath <level>

There must be a Death mage of at least this level in the province.

#req_pathnature <level>

There must be a Nature mage of at least this level in the province.

#req_pathglamour <level>

There must be a Glamour mage of at least this level in the province.

#req_pathblood <level>

There must be a Blood mage of at least this level in the province.

#req_pathholy <level>

There must be a Priest of at least this level in the province.

#req_nopathfire <level>

There cannot be a Fire mage of this level or higher in the province.

#req_nopathair <level>

There cannot be an Air mage of this level or higher in the province.

#req_nopathwater <level>

There cannot be a Water mage of this level or higher in the province.

#req_nopatheart <level>

There cannot be an Earth mage of this level or higher in the province.

#req_nopathastral <level>

There cannot be an Astral mage of this level or higher in the province.

#req_nopathdeath <level>

There cannot be a Death mage of this level or higher in the province.

#req_nopathnature <level>

There cannot be a Nature mage of this level or higher in the province.

#req_nopathglamour <level>

There cannot be a Glamour mage of this level or higher in the province.

#req_nopathblood <level>

There cannot be a Blood mage of this level or higher in the province.

#req_nopathholy <level>

There cannot be a Priest of this level or higher in the province.

#req_nopathall <level>

There cannot be a mage or priest of any kind of this level or higher in the province.

Target Requirements

A single commander must fulfill all target requirements for the event to occur. Some commands may be used multiple times with different values (e.g. Monster types), in which case one of the conditions being present fulfills that requirement.

#req_targmnr "name" | <number>

Target commander must be this monster type. This command can be used multiple times.

#req_targomnr "name" | <number>

Target commander mustn't be this monster type. This command can be used multiple times.

#req_targgod < 0 - 2 >

Target must be god. 1 = must be pretender or disciple, 2 = must be pretender, 0 = cannot be pretender or disciple.

#req_targprophet < 0 | 1 >

Target must be a prophet. 1 = must be a prophet, 0 = cannot be a prophet.

#req_targhumanoid < 0 | 1 >

Sets item slot requirement for target. 1 = must have hands, 0 = must not have hands.

#req_targsight < 0 | 1 >

Target must have eyes and be able to see with them. 1 = must have sight, 0 = cannot have sight.

#req_targmale < 0 | 1 >

Sets target's gender requirement. 1 = must be male, 0 = must be female.

#req_targmanygems <path number>

Target must carry many gems (>d20) of this type. Use path number 53 for all type of gems and 56 for all but blood.

#req_targpath1 <path number>

Must have level 1+ in target magic path.

#req_targpath2 <path number>

Must have level 2+ in target magic path.

#req_targpath3 <path number>

Must have level 3+ in target magic path.

#req_targpath4 <path number>

Must have level 4+ in target magic path.

#req_targnopath1 <path number>

Must have level 0 in target magic path.

#req_targnopath2 <path number>

Must have level 0-1 in target magic path.

#req_targnopath3 <path number>

Must have level 0-2 in target magic path.

#req_targnopath4 <path number>

Must have level 0-3 in target magic path.

Nbr	Magic Path
0	Fire
1	Air
2	Water
3	Earth
4	Astral
5	Death
6	Nature
7	Glamour
8	Blood
9	Priest
50	Random
51	Elemental
52	Sorcery
53	All (not priest)

Magic path numbers

#req_targaff <affliction number>

Target must have this affliction. If you have multiple afflictions added together, then the target must have all of these afflictions.

#req_targnoaff <affliction number>

Target mustn't have this affliction. If you have multiple afflictions added together, then the target mustn't have all of these afflictions.

#req_targorder <order number>

Target must be performing these orders. This command can be used multiple times, in which case one of the conditions being present is sufficient to fulfill the requirement. Target orders are listed in the table below.

Nbr	Order
0	Defend
2	Sneak
3	Patrol
4	Research
6	Preach
7	Site search
8	Blood hunt
14	Call God
17	Summon allies
18	Build lab
19	Build temple
20	Build fort
44	Demolish fort
45	Demolish lab
50	Enter site (except scry)
52	Void Summon
100	Investigate
101	Continue
102	Accept
103	Decline
105	Attack
106	Diplomacy
107	Subterfuge
108	Magic

Order numbers

#req_targitem "item name" | <item number>

Target must have this magic item.

#req_targnoitem "item name" | <item number>

Target must not have this magic item.

#req_targundead <0 | 1>

Target must (not) be an undead.

#req_targdemon <0 | 1>

Target must (not) be a demon.

#req_targanimal <0 | 1>

Target must (not) be an animal.

#req_targinanimate <0 | 1>

Target must (not) be an inanimate being.

#req_targmindless <0 | 1>

Target must (not) be mindless.

#req_targimmobile <0 | 1>

Target must (not) be immobile.

#req_targmagicbeing <0 | 1>

Target must (not) be a magic being.

#req_targowner <nation number>

Target must owned by this nation. Negative nation number cannot be used and the command can only be used once per event.

#req_targforeignok

Indicates that the target can be owned by someone else than the owner of the land. The default (without this command) is that all req_targ... commands only target commanders that are owned by the player that owns the land where the event occurs.

#req_targminsize <size>

Target must be this size or larger.

#req_targmaxsize <size>

Target must be this size or smaller.

#req_targinsane <0 | 1>

Target must (not) be insane and uncontrollable right now.

#req_targnotowner <nation number>

Target must not be owned by this nation.

#req_targnoorder <order>

Target must not be performing this order.

#req_targminmorale <morale>

Target cannot have lower morale than this

#req_targmaxmorale <morale>

Target cannot have higher morale than this

#req_targally <nation number>

Target must be owned by this nation or one of its allies

#req_targnotally <nation number>

Target must not be owned by this nation or any of its allies

#req_targseductions <value>

Target must have succeeded with at least this many seductions (dream seduction and corruption also counts).

#req_targminkills <value>

Target must have killed at least this many people.

#req_targmaxkills <value>

Target must have killed at most this many people.

Event Code Requirements

These commands set event code requirements that must be fulfilled for the event to take place. They are generally used in event chains where the event requires another event to have taken place previously.

There is a risk of incompatible mods if the same event codes are used across different mods!

Please use only negative event codes from -300 and downwards, or it may interfere with the events in the vanilla game (except for 0, which is default).

#req_code <event code>

Province must have this event code. Can be used multiple times for an event. If the command is used multiple times, one of the codes being present is sufficient to fulfill the requirement. Use #req_code 0 for events that generate event codes. Otherwise they may break ongoing event chains.

#req_notcode <event code>

Province must not have this event code. Can be used multiple times for an event.

#req_anycode <event code>

Event code must be present somewhere in the world. Can be used multiple times for an event. If the command is used multiple times, one of the codes being present is sufficient to fulfill the requirement.

#req_notanycode <event code>

Event code must not be present in the world. Can be used multiple times for an event. If the command is used multiple times, none of the codes may be present.

#req_nearbycode <event code>

Event code must be present in a neighboring province.

#req_nearowncode <event code>

Current province or neighboring province owned by the same player must have this event code.

Enchantment Requirements

These commands are intended for global enchantment creation.

#req_permonth <nbr>

This event can only happen <nbr> times per month.

#req_noench <ench nbr>

This enchantment cannot be active.

#req_ench <ench nbr>

This enchantment must be active.

#req_myench <ench nbr>

This enchantment must be active and owned by the event owner.

#req_friendlyench <ench nbr>

This enchantment must be active and friendly.

#req_hostileench <ench nbr>

This enchantment must be active and hostile.

#req_enchdom <ench nbr>

This enchantment must be active and in its owner's dominion.

#req_enchtarget <ench nbr>

This global enchantment must be active and have its origin in the current province.

#req_enchnearby <ench nbr>

This global enchantment must be active and have its origin in the current or a neighboring province.

Event Variable Requirements

These commands can be used to check a set of global variables called "event variables" that can be altered and checked by events. The event variable number from 0 to 9999 and can contain any integer number. Event var number -1 can also be used, it will be changed to the number of the nation receiving the event. So by using the -1 event var it is e.g. possible to create an event that can occur a limited amount of times per player.

#req_varpos <event var>

The event variable must have a value of 1 or higher.

#req_varneg <event var>

The event variable must have a value of -1 or lower.

#req_varzero <event var>

The event variable must have a value of exactly zero.

#req_varone <event var>

The event variable must have a value of exactly one.

Misc Requirements

#req_arenadone <0 | 1>

The will require that an arena battle has taken place (been resolved) earlier this turn. If set to zero an arena battle must not have taken place.

#req_worlditem "item name" | <item number>

This item must exist somewhere in the world.

#req_noworlditem "item name" | <item number>

This item must not exist anywhere in the world.

Event Effects

Basic Effects

#nation <nation number>

Sets the nation that owns the event. The default owner is independents. -1 = random enemy, -2 = province owner.

The effects of the event are applied in the province where the event occurs and by the nation who owns the event. In an event with an attack, it can be used to set the attacking troops to be non-independents.

#nationench <nation number>

Sets the event owner to the nation that has this global enchantment active. This command is useful to create global enchantments that spawns units owned by the caster of the global enchantment.

#msg "<message text>"

Message text for the event. For many event modding commands that need item or site names to be determined for the event to function, the item or site name to be added to the end of the text of this command in brackets. For example:

```
#msg "An adventurous young mage is searching for the fabled Singing Stones. [Singing Stones]"
```

In this example the event probably contains a requirement for the magic site Singing Stones to be in the province or an adjacent province and the command formulation requires the name of the site to be inserted into the message text of the event.

There are also many special tags that can be inserted in the message text to print certain variables, e.g. the name of the land or the name of the player's pretender. See the list below for the possible tags that can be used.

```
##landname##  
##godname##      (god name + 1 random title)  
##fullgodname## (god name + all titles)  
##disname##      (use name of disciple if current player  
is one, otherwise name of god)  
##goddisname##   (50% chance of using name of disciple if  
current player is a disciple)  
##targname##     (name of targeted commander) (not for  
global events)  
##fulltargname## (name + unit type of targeted commander)  
(not for global events)  
##targhis##      (his/her depending on targeted  
commander) (not for global events)  
##natname##      (name of nation, e.g. Vanheim, Aaysia)  
##profname##     (name of prophet/disciple)
```

#header <type>

Usually all messages start with "An unexpected event has occurred in...". Use this command to change that. Type 1 = no unexpected, 2 = no header. If you use 2 (no header) the first line

in the message will be used as a header, the header should end with two newlines before the rest of the message.

#notext

Event does not generate a message.

#extramsmsg <nation number>

This nation will also get the event message.

#nolog

Event does not appear in the chronicle (event history) of the province where it occurs.

Treasure Effects

All of these commands affect the gold, gems or magic items in the event owner's treasury.

#magicitem <rarity>

The nation that owns the event gains a magic item that is placed directly into the lab. Item rarity is listed in the table below. When rarity 9 is used, add the itemname in brackets at the end of the event message text defined by the #msg command.

Rarity	Constr Level
0	1
1	1 - 3
2	1 - 5
3	1 - 7
4	5 - 7
9	(named item)

Item Rarity

#gold <value>

The nation that owns the event gains <value> gold. Use negative numbers for gold loss. The value varies with +/- 50%.

#exactgold <value>

The nation that owns the event gains <value> gold. Use negative numbers for gold loss.

#forcegold <value>

The nation that owns the event gains <value> gold. Use negative numbers for gold loss. The value varies with +/- 50%. This event may occur event if income cannot be traced to a friendly fort.

#forceexactgold <value>

The nation that owns the event gains <value> gold. Use negative numbers for gold loss. This event may occur event if income cannot be traced to a friendly fort.

Scale Effects

Scale effects alter the scales of a province. Each `#incscale` command functions as the equivalent `#decscale` of its opposite.

`#incscale <scale>`

Increases the specified scale in the province by one step.

Nbr	Scale	Opposite
0	Turmoil	Order
1	Sloth	Productivity
2	Cold	Heat
3	Death	Growth
4	Misfortune	Luck
5	Drain	Magic

Scale Numbers

`#incscale2 <scale>`

Increases the specified scale in the province by two steps.

`#incscale3 <scale>`

Increases the specified scale in the province by three steps.

`#1d3vis <gem type>`

Gain 1d3 gems of specified type. Event will not occur if income cannot be traced to a friendly lab.

`#1d6vis <gem type>`

Gain 1d6 gems of specified type. Event will not occur if income cannot be traced to a friendly lab.

`#2d4vis <gem type>`

Gain 2d4 gems of specified type. Event will not occur if income cannot be traced to a friendly lab.

`#2d6vis <gem type>`

Gain 2d6 gems of specified type. Event will not occur if income cannot be traced to a friendly lab.

`#3d6vis <gem type>`

Gain 3d6 gems of specified type. Event will not occur if income cannot be traced to a friendly lab.

`#4d6vis <gem type>`

Gain 4d6 gems of specified type. Event will not occur if income cannot be traced to a friendly lab.

`#force1d3vis <gem type>`

Gain 1d3 gems of specified type. Event may occur even if income cannot be traced to a friendly lab.

`#force1d6vis <gem type>`

Gain 1d6 gems of specified type. Event may occur even if income cannot be traced to a friendly lab.

`#force2d4vis <gem type>`

Gain 2d4 gems of specified type. Event may occur even if income cannot be traced to a friendly lab.

`#force2d6vis <gem type>`

Gain 2d6 gems of specified type. Event may occur even if income cannot be traced to a friendly lab.

`#force3d6vis <gem type>`

Gain 3d6 gems of specified type. Event may occur even if income cannot be traced to a friendly lab.

`#force4d6vis <gem type>`

Gain 4d6 gems of specified type. Event may occur even if income cannot be traced to a friendly lab.

`#gemloss <gem type>`

Lose 3d6 gems of specified type. Gem type 53 can be used for all types, then 1d6 will be lost of each type of gem. Use gem type 56 to exclude blood slaves.

`#gemlosssmall <gem type>`

Like `#gemloss`, but 2d6 gems instead.

`#gemlosslarge <gem type>`

Like `#gemloss`, but 4d6 gems instead.

`#decscale <scale>`

Decreases the specified scale in the province by one step.

`#decscale2 <scale>`

Decreases the specified scale in the province by two steps.

`#decscale3 <scale>`

Decreases the specified scale in the province by three steps.

Province Effects

These effects affect the target province directly.

`#landgold <value>`

Permanent increase to province gold income. Can be negative to reduce income.

`#landprod <value>`

Permanent increase to province resources. Can be negative to cause reduce resources.

`#taxboost <percent>`

Tax income increase in the province for one turn. Use negative numbers for a tax decrease. A value of -100 means no gold income for that turn.

`#defence <value>`

Increases Province Defense by `<value>`

`#kill <percent>`

Kills off a percentage of the province population.

`#killpop <dekapop>`

Kills off a fixed number of province population. One dekapop equals 10 inhabitants. The exact amount varies with +/- 25%.

#incpop <dekapop>

Province population increases by this amount +/- 25%.

#incorporates <amount>

Number of unburied corpses increases by this amount +/- 25%.
The value can be negative to remove corpses.

#emigration <percent>

A percentage of the province population emigrates elsewhere.

#unrest <value>

Unrest in province increased by <value>. Can be negative to reduce unrest.

#incdom <value>

Increases dominion in the province by <value> candles.

#fort <fort number>

Builds a fort of the specified type in the province.

#temple < 0 | 1 >

The event either creates or destroys a temple. 1 = temple gained, 0 = temple destroyed.

#lab < 0 | 1 >

The event either creates or destroys a lab. 1 = new lab, 0 = lab destroyed.

#revealsite

Reveals the site called "sitename" in the province. "sitename" must be added to the event message text in brackets.

#addsite <site number>

Adds a new, already discovered site to the province. Use -1 for site number to place "sitename" instead. "sitename" must be added to event message text in brackets.

#removesite <site number>

Removes a site from the province. Use -1 for site number to place "sitename" instead. "sitename" must be added to event message text in brackets. The site will be removed regardless of being found or not.

#hiddensite <site number>

Adds a new, hidden site to the province. Use -1 for site number to place "sitename" instead. "sitename" must be added to event message text in brackets.

#visitors

The province is attacked by Bogus the Troll and his Companions.

#newdom <value>

Changes dominion to that of the event owner and sets it to the level of 1d(value) candles.

#revolt

Province revolts and turns independent.

#revealprov

The name of the province is revealed to the entire world.

#claimthrone

Any throne in the province is claimed by the event owner.

#setpoptype <value>

Sets the poptype in a province. The poptype will affect what kind of units that can be recruited in the province.

#addgeo <terrain bitmask>

Adds to the terrains in the province. A number of 128 would add the forest terrain to the province.

#remgeo <terrain bitmask>

Removes from the terrains in the province. A number of 128 would remove the forest terrain should there be any.

Monster Effects

The argument of "name" | <number> means monster name or monster number.

#assassin "name" | <number>

An assassin of this type attacks a random commander in the province. If a target requirement for the event is set, the assassin attacks that commander.

#assowner <nation number>

The following assassins created by this event will be owned by this nation.

#assfollower1 "name" | <number>

Assassins from following event commands will receive 1 follower of this type.

#assfollower2 "name" | <number>

Assassins from following event commands will receive 2 followers of this type.

#assfollower3 "name" | <number>

Assassins from following event commands will receive 3 followers of this type.

#assfollower1d3 "name" | <number>

Assassins from following event commands will receive 1d3 followers of this type.

#stealthcom "name" | <number>

The nations that owns the event gains a sneaking commander of the specified type in the province.

#com "name" | <number>

The nation that owns the event gains a commander of the specified type.

#2com "name" | <number>

The nation that owns the event gains two commanders of the specified type.

#4com "name" | <number>

The nation that owns the event gains four commanders of the specified type.

#5com "name" | <number>

The nation that owns the event gains five commanders of the specified type.

#tempunits < 0 | 1>

Units gained after this command are only temporary and disappear after the event has taken place, like the longdead horsemen from the Ghost Riders spell. 1 = temporary units, 0 = permanent units.

#1unit "name" | <number>

The nation that owns the event gains one unit.

#1d3units "name" | <number>

The nation that owns the event gains 1d3 units.

#2d3units "name" | <number>

The nation that owns the event gains 2d3 units.

#3d3units "name" | <number>

The nation that owns the event gains 3d3 units.

#4d3units "name" | <number>

The nation that owns the event gains 4d3 units.

#1d6units "name" | <number>

The nation that owns the event gains 1d6 units.

#2d6units "name" | <number>

The nation that owns the event gains 2d6 units.

#3d6units "name" | <number>

The nation that owns the event gains 3d6 units.

#4d6units "name" | <number>

The nation that owns the event gains 4d6 units.

#5d6units "name" | <number>

The nation that owns the event gains 5d6 units.

#6d6units "name" | <number>

The nation that owns the event gains 6d6 units.

#7d6units "name" | <number>

The nation that owns the event gains 7d6 units.

#8d6units "name" | <number>

The nation that owns the event gains 8d6 units.

#9d6units "name" | <number>

The nation that owns the event gains 9d6 units.

#10d6units "name" | <number>

The nation that owns the event gains 10d6 units.

#11d6units "name" | <number>

The nation that owns the event gains 11d6 units.

#12d6units "name" | <number>

The nation that owns the event gains 12d6 units.

#13d6units "name" | <number>

The nation that owns the event gains 13d6 units.

#14d6units "name" | <number>

The nation that owns the event gains 14d6 units.

#15d6units "name" | <number>

The nation that owns the event gains 15d6 units.

#16d6units "name" | <number>

The nation that owns the event gains 16d6 units.

#strikeunits <damage>

All units in the province are hit with an attack.

#killmon "name" | <number>

Kills one non-commander unit of the specified type in the province.

#kill2d6mon "name" | <number>

Kills 2d6 non-commander units of the specified type in the province.

#killcom "name" | <number>

Kills a commander of the specified type in the province.

#curse <percent>

Every unit in the province has this chance of being cursed.

#disease <percent>

Every unit in the province has this chance of becoming diseased.

#researchaff <affliction bitmask>

A researcher suffers an affliction.

#gainaff <affliction bitmask>

One commander determined by target requirements gains the specified affliction. Afflictions are listed in the table below. The bitmask is expressed as powers of 2. Add together several bitmasks if more than one affliction is desired.

2^x	Affliction
0	disease
1	curse
3	plague
5	curse of stones
6	entangle
7	rage
8	decay
9	burn
10	destroyed armor
11	rusty armor
12	blind

13	bleed
14	earth grip
16	fire bonds
17	false fetters
18	limp
19	lost eye
20	weakness
21	battle fright
22	mute
23	chest wound
24	crippled
25	feeble minded
26	never healing wound
27	slime
28	frozen
29	webbed
30	arm loss

Affliction Bitmasks

#gainmark

One commander determined by target requirements gains one level of horror mark.

#healaff <nbr>

One commander determined by target requirements gets nbr of affliction healing chances. One healing chance can remove at most a single permanent battle affliction, and sometimes no affliction is healed.

#banished <value>

One commander determined by target requirements is banished to another plane. -11 = The Void, -12 = Inferno, -13 = Kokytos.

#addequip <equipment level>

Adds magic items to a commander. Equipment level is from 1 to 4 and corresponds to item level from trinkets to very powerful items. Use the value 9 for equipment level and a bracketed "item name" in the #msg text to add a specific magic item. An item is not guaranteed at equipment level 1. The command also gives the commander gems appropriate to his magic paths.

#transform "name" | <number>

One commander determined by target requirements is transformed into another monster, negative numbers can be used for montags.

#forcetransform "name" | <number>

One commander determined by target requirements is transformed into another monster, negative numbers can be used for montags. This command can even transform gods and other units that are immune to polymorph.

#remount

The commander determined by target requirements regains his mount, should he have lost it.

#cleartarg

Commander set by target requirements is cleared. Commands after this will use default target.

#poison <dmg>

One commander determined by target requirements is poisoned. The poison effect is armor negating but the damage will vary a bit anyway because of the usual +2d6 -2d6 calculations.

#killtarg

One commander determined by target requirements is killed. The mount is also killed if any.

#xp <experience points>

One commander determined by target requirements will receive additional XP. The exact amount of XP gained is +/- 25%.

#setxp <experience points>

One commander determined by target requirements will get its XP set to this amount. The exact amount of XP is +/- 25%.

#addseductions <value>

One commander determined by target requirements will get this number added to the number of seductions she has successfully performed. The resulting value will never go below zero.

#addkills <value>

One commander determined by target requirements will get this number added to the number of kills he has. The resulting value will never go below zero.

Magic Pathboost Effects

When the target of these commands is set by the target requirements of the event, the monster name or monster number will have no effect (you can set it to -1). If no target requirements have been set, monster name or number is used to determine the target.

#fireboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Fire magic.

#airboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Air magic.

#waterboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Water magic.

#earthboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Earth magic.

#astralboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Astral magic.

#deathboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Death magic.

#natureboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Nature magic.

#glamourboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Glamour magic.

#bloodboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Blood magic.

#holyboost "name" | <number>

One commander determined by target requirements or the monster number may gain +1 Priest level.

#pathboost <path>

The mage who gains this event has the specified path level increased by 1, to a maximum of 5. The target of this event is set by the target requirements of the event. Path must be a single magic path, nbr 0 - 9.

Nbr	Path
0	Fire
1	Air
2	Water
3	Earth
4	Astral
5	Death
6	Nature
7	Glamour
8	Blood
9	Holy

Global Event Effects

These event effects affect every province in the world when the event happens.

#worldincscale <scale>

Increases the specified scale in every province in the world by one step.

#worldincscale2 <scale>

Increases the specified scale in every province in the world by two steps.

#worldincscale3 <scale>

Increases the specified scale in every province in the world by three steps.

#worlddecscale <scale>

Decreases the specified scale in every province in the world by one step.

#worlddecscale2 <scale>

Decreases the specified scale in every province in the world by two steps.

#worlddecscale3 <scale>

Decreases the specified scale in every province in the world by three steps.

Nbr	Scale	Opposite
0	Turmoil	Order
1	Sloth	Productivity
2	Cold	Heat
3	Death	Growth
4	Misfortune	Luck
5	Drain	Magic

Scale Numbers**#worldunrest <value>**

Unrest in every province in the world is increased by <value>. Can be negative to reduce unrest.

#worldincdom <value>

Increases dominion in every province in the world by <value> candles.

#worldritrebat <school>

The casting cost of rituals of this school is reduced by 20% for this turn. School numbers are listed in the table below.

Worldritrebat can only be used on normal global events (rarity 11 and 12).

Nbr	School
0	Conjuration
1	Alteration
2	Evocation
3	Construction
4	Enchantment
5	Thaumaturgy
6	Blood
7	Divine

Magic Schools**#worlddarkness**

The entire world is covered in darkness for this turn.

#worldcurse <percent>

Every unit in the world has a <percent> chance of being cursed.

#worlddisease <percent>

Every unit in the world has a <percent> chance of becoming diseased.

#worldmark <percent>

Every unit in the world has a <percent> chance of being horromarked.

#worldheal <percent>

Each unit in the world has a <percent> chance of having one affliction healed.

#worldage <years>

Everyone in the world ages this many years. Negative numbers reduce age.

#linger <nbr of turns>

Use this command for global events whose effects last for more than one turn.

Special Event Commands

These commands are used to construct event chains and special events.

#flagland < 0 | 1 >

Sets or removes an event flag from a province. The event flag is displayed as an image of a book on the map. 1 = set flag, 0 = remove flag.

Requires an event code in the province. If the event code is reset to 0, the flag automatically disappears.

#delay <value>

Succeeding event occurs <value> turns after this event. The succeeding event will not occur normally.

#delay25 <value>

Succeeding event occurs <value> +/- 25% turns after this event. The succeeding event will not occur normally.

#delay50 <value>

Succeeding event occurs <value> +/- 50% turns after this event. The succeeding event will not occur normally.

#delayskip <percent>

Use this command right after a delay command. This is the chance that the next next event will be used instead of the next one, when determining what delayed event to execute. The next next event will not occur normally.

#order <order bitmask>

Set special event orders that are enabled for the province (investigate, accept, decline etc). If several orders are needed to be available, add together all the mask values for the orders. The nil order resets all order options. See table below.

Mask	Event Order
0	Nil
1	Investigate
2	Continue
4	Accept
8	Decline
16	Withdraw
32	Attack
64	Diplomacy
128	Subterfuge
256	Magic

Event Code Effects

#code <event code>

Sets the event code of the current province (when planned for globals)

#code2 <event code>

Sets the event code of the current province (when executed for globals)

#resetcode <event code>

Removes this event code from the entire world.

#purgecalendar <attr>

Removes an event determined by attr from the global event calendar.

1 = the event that had been added after this event would be affected.

0 = current event

-1 = the previous event.

The event calendar contains the planned global events that will occur in the coming 7 months.

#purgedelayed <attr>

Removes an event from the delayed event list (same attr as previous command). Delayed events are events added by the #delay command.

#id <event identifier>

Sets an event identifier. Used for spells that create anonymous random events.

#codedelay <event code>

Sets the event code of the current province, but not immediately. It will be set after all events have been executed this turn.

#codedelay2 <event code>

Sets the event code of the current province, but not immediately. It will be set after all events have been executed on the next turn.

#resetcodedelay <event code>

Removes this event code from the entire world, but not immediately. It will be removed after all events have been executed this turn.

#resetcodedelay2 <event code>

Removes this event code from the entire world, but not immediately. It will be removed after all events have been executed on the next turn.

Event Variables

These commands can be used to manipulate a set of global variables called "event variables" that can be altered and checked by events. The event variable number from 0 to 9999 and can contain any integer number. All event variables start with a value of zero. Event var number -1 can also be used, it will be changed to the number of the nation receiving the event. So by using the -1 event var it is e.g. possible to create an event that can occur a limited amount of times per player. Note that using the -1 variable means that variables up to 499 will be used.

#clearvar <event var>

Sets an event var (0 to 9999) to zero.

#incvar <event var>

Adds one to an event var.

#decvar <event var>

Decreases one from an event var.

#invvar <event var>

Inverts the value of an event var (result = -event var).

#togglevar <event var>

Toggles the value of an event var (result = 1-event var). Use to toggle between 1 and 0.

Misc Event Effects

#arena

Opens the Area Death Match for competitors. Players will be able to send heroes to the arena this turn.

#resolvearena1

The area battles will be resolved and the winner rewarded. This event (or #resolvearena2) must occur the turn after the #arena event occurred.

#resolvearena2

Same as #resolvearena1, but it will be an arena where the use of magic is prohibited.

#addascension <AP>

Adds to the number of ascension points required to win the game. This command has no effect if the thrones win condition is not in effect. AP can be negative to make ascension easier.

#minascension <AP>

Ensures the number of ascension points required to win the game is at least this many.

Examples

Basic Event

This is an example event that reveals a hidden site in the province.

```
#newevent
#rarity -2
#req_hiddensite 1
#msg "A famous prospector has discovered previously
unknown deposits of precious jewels. [Hidden Gem
Deposits]"
#revealsite
#end
```

Complex Event

This is a complex event that requires the unique site The Sea of the Tears of Men in the province (hidden or discovered) as well as a humanoid mage with W2D3 or greater set to site search. The event is an uncommon good event for the province owner and can only happen once during a game. If the event triggers, the mage has a chance of getting +1 Water and +1 Death magic and the recipient of the event also gains a Kokythiad commander.

```
#newevent
#rarity -2
#nation -2
#req_unique 1
#req_site 1
#req_targhumanoid 1
#req_targpath2 2
#req_targpath3 5
#req_targorder 7
#msg "During a pilgrimage to the River Styx, one of
your mages was swept away by the waters and out into the
Sea of the Tears of Men. He would have died but for
catching the eye of a passing Kokythiad who rescued him
and nursed him back to health. With the help of the
Kokythiad, he has now found his way back and brought her
with him as his wife. [The Sea of the Tears of Men]"
#waterboost -1
#deathboost -1
#com 1477
#end
```

Event Chain

This event chain contains two events that follow one after another.

```

#newevent
#rarity 2
#req_land 1
#req_mountain 1
#req_code 0
#code -509
#flagland 1
#decscale3 5
#incscale2 3
#incunrest 20
#delay 1
#magicitem 9
#msg "Heavy rains have washed away a clearcut hillside, revealing a great stone carved with disquieting runes. At the foot of the stone a skull staff had been planted, but the local lord removed it at once and sent it to the treasury. At night the runes began to glow with a bilious green light. The local hedge witches and soothsayers have reported their cantrips and remedies to be more powerful since, but plants and trees in the vicinity of the stone have begun to wither and die and the circle of devastation is spreading outward. [Skull Staff]"
#end

#newevent
#rarity 0
#req_code -509
#nation 2 -- special independents
#com "Mound Fiend"
#com "Bane Lord"
#com "Wraith Lord"
#2d6units "Ghost"
#3d6units "Wight"
#code 0
#msg "At the dark of the moon the runes on the stone uncovered from the hillside flared brightly and the slab moved aside, revealing the entrance to a crypt where a triumvirate of ancient evil had been locked away. With the skull staff removed, the chains of magic holding the evil contained had unraveled and now the denizens of the prison have attacked the province!"
#end

```

Note how the second event in the chain resets the event code in the province back to 0 to prevent this event reoccurring and paving the way for other event chains.

Simple Global Event

This creates a new global enchantment ritual. The global enchantment will active a worldwide event.

```

#newspell
#name "Illvaettitime"
#descr "my modded end of the world global"
#school 4
#researchlevel 1
#path 0 6
#pathlevel 2
#fatiguecost 500
#effect 10081
#damage 200
#end

-- The following event will occur whenever there is a global nbr 200 (illvaettitime) in play and disease 10% of all units

#newevent
#rarity 13
#msg "Illvaettitime strikes again"
#req_ench 200
#req_permonth 1
#worldldisease 10
#end

```

Advanced Global Event

This is a more advanced example of a global enchantment event.

```

#newspell
#name "Illvaettitime II"
#descr "This global enchantment grants the owner 500 gold per month and creates wolf attacks in forest and mountain provinces owned by the enemy. Wolf attacks will only occur inside your dominion."
#school 4
#researchlevel 1
#path 0 6
#pathlevel 2
#fatiguecost 500
#effect 10081
#damage 201
#end

-- This event occurs whenever there is a global nbr 201 (illvaettitime II) in play and gives the owner 500 gold

#newevent
#rarity 5
#msg "Illvaettitime II gives gold"
#notext
#req_myench 201
#req_permonth 1
#gold 500
#end

```

```
-- This event may create (50% chance) a small wolf  
attack in a mountain when global nbr 201 is in effect
```

```
#newevent  
#rarity 5  
#msg "Illvaettitime II spawned wolves here"  
#notext  
#req_hostileench 201  
#req_rare 50  
#req_mountain 1  
#req_enchdom 201  
#req_indepok  
#nationench 201  
#com "Wolf"  
#2d6units "Wolf"  
#end
```

```
-- This event may create (35% chance) a large wolf  
attack in a forest when global nbr 201 is up
```

```
#newevent  
#rarity 5  
#msg "Illvaettitime II spawned wolves here"  
#notext  
#req_hostileench 201  
#req_rare 35  
#req_forest 1  
#req_enchdom 201  
#req_indepok  
#nationench 201  
#com "Wolf"  
#5d6units "Wolf"  
#end
```