

Hearts of Iron 4 Wiki

This is a community maintained wiki. If you spot a mistake then you are welcome to fix it.

Effects (also known as Commands) are used in order to affect the game dynamically from within a specific scope. They are a one-time change to the current condition of the game, **without the ability to have a lasting effect**. Instead, [modifiers](#) are used to have a continuous, everlasting effect on the game's condition that can be represented with a number. Effect blocks cannot be used to apply modifiers directly, however they can add something that can apply modifiers, most commonly with [add ideas](#).

Effects are used throughout the game in numerous scopes, most commonly [National focuses](#), [Events](#) and [Decisions](#).

Note that certain effects may take a value from a variable, i.e. `add_manpower = var:my_var` This is noted by **<variable>** in an effect's parameters. See [Variables](#) for information on the variable effects.

The list of effects may be outdated. A complete, but unsorted, list of effects can be found in `/Hearts of Iron IV/documentation/effects_documentation.html` or `/Hearts of Iron IV/documentation/effects_documentation.md`.

Scopes[\[编辑 | 编辑源代码\]](#)

Main article: [Scopes](#)

These don't serve as effects, but rather as scopes that change for whom the effects are being executed.

Effect scopes[\[编辑 | 编辑源代码\]](#)

Effect scopes: <div>折叠</div>					
Name	Usage	Target type	Example	Description	Version Added
every_possible_country	Always usable	Country	<code>every_possible_country = { ... }</code>	Executes children effects on every country that meets the limit, including those that do not exist.	1.11
every_country	Always usable	Country	<code>every_country = { ... }</code>	Executes contained effects on every country that meets the limit.	1.0
random_country	Always usable	Country	<code>random_country = { ... }</code>	Executes contained effects on a random country that meets the limit.	1.0
every_other_country	Within country scope only	Country	<code>every_other_country = { ... }</code>	Executes contained effects on every country that meets the limit and is not the same country as the one this is contained in.	1.0
random_other_country	Within country scope only	Country	<code>random_other_country = { ... }</code>	Executes contained effects on a random country that meets the limit and is not the same country as the one this is contained in.	1.0
every_country_with_original_tag	Always usable	Country	<code>every_country_with_original_tag = { original_tag_to_check = TAG #required ... #effects to run }</code>	Executes contained effects on every country that meets the limit and has the specified original tag.	1.9
random_country_with_original_tag	Always usable	Country	<code>random_country_with_original_tag = { original_tag_to_check = TAG #required ... #effects to run }</code>	Executes contained effects on a random country that meets the limit and has the specified original tag.	
every_neighbor_country	Within country scope only	Country	<code>every_neighbor_country = { ... }</code>	Executes contained effects on every country that meets the limit and borders the country this is contained in.	1.0
random_neighbor_country	Within country scope only	Country	<code>random_neighbor_country = { ... }</code>	Executes contained effects on a random country that meets the limit and borders the country this is contained in.	1.0
every_occupied_country	Within country scope only	Country	<code>every_occupied_country = { ... }</code>	Executes contained effects on every country that meets the limit and has any core states controlled by the country this is contained in.	1.9
random_occupied_country	Within country scope only	Country	<code>random_occupied_country = { ... }</code>	Executes contained effects on a random country that meets the limit and has any core states controlled by the country this is contained in.	1.9
every_enemy_country	Within country scope only	Country	<code>every_enemy_country = { ... }</code>	Executes contained effects on every country that meets the limit and is at war with the country this is contained in.	1.0
random_enemy_country	Within country scope only	Country	<code>random_enemy_country = { ... }</code>	Executes contained effects on a random country that meets the limit and is at war with the country this is contained in.	1.0
every_subject_country	Within country scope only	Country	<code>every_subject_country = { ... }</code>	Executes contained effects on every country that meets the limit and is a subject of the country this is contained in.	1.11
random_subject_country	Within country scope only	Country	<code>random_subject_country = { ... }</code>	Executes contained effects on a random country that meets the limit and is a subject of the country this is contained in.	1.11
every_state	Always usable	State/s	<code>every_state = { ... }</code>	Executes contained effects on every state that meets the limit.	1.0
random_state	Always usable	State	<code>random_state = { prioritize = { 123 321 } #optional ... #effects to run }</code>	Executes contained effects on a random state that meets the limit.	1.0
every_neighbor_state	Within state scope only	State	<code>every_neighbor_state = { ... }</code>	Executes contained effects on every state that meets the limit and neighbours the state this is contained in.	1.0
random_neighbor_state	Within state scope only	State	<code>random_neighbor_state = { ... }</code>	Executes contained effects on a random state that meets the limit and neighbours the state this is contained in. Does not support prioritizing .	1.0
every_owned_state	Within country scope only	State	<code>every_owned_state = { ... }</code>	Executes contained effects on every state that meets the limit and is owned by the country this is contained in.	1.0
random_owned_state	Within country scope only	State	<code>random_owned_state = { prioritize = { 123 321 } #optional ... #effects to run }</code>	Executes contained effects on a random state that meets the limit and is owned by the country this is contained in.	1.0
every_core_state	Within country scope only	State	<code>every_core_state = { ... }</code>	Executes contained effects on every state that meets the limit and is a core of the country this is contained in.	1.11
random_core_state	Within country scope only	State	<code>random_core_state = { prioritize = { 123 321 } #optional ... #effects to run }</code>	Executes contained effects on a random state that meets the limit and is a core of the country this is contained in.	1.11

Name	Usage	Target type	Example	Description	Version Added
every_controlled_state	Within country scope only	State	<code>every_controlled_state = { ... }</code>	Executes contained effects on every state that meets the limit and is controlled by the country this is contained in.	1.9
random_controlled_state	Within country scope only	State	<code>random_controlled_state = { prioritize = { 123 321 } #optional ... #effects to run }</code>	Executes contained effects on a random state that meets the limit and is controlled by the country this is contained in.	1.9
random_owned_controlled_state	Within country scope only	State	<code>random_owned_controlled_state = { prioritize = { 123 321 } #optional ... #effects to run }</code>	Executes contained effects on a random state that meets the limit and is owned and controlled by the country this is contained in.	1.3
every_unit_leader	Within country scope only	Unit Leader	<code>every_unit_leader = { ... }</code>	Executes contained effects on every unit leader (corps commanders, field marshals, admirals) that meets the limit and is recruited by the country this is contained in.	1.5
random_unit_leader	Within country scope only	Unit Leader	<code>random_unit_leader = { ... }</code>	Executes contained effects on a random unit leader (corps commanders, field marshals, admirals) that meets the limit and is recruited by the country this is contained in.	1.5
every_army_leader	Within country scope only	Unit Leader	<code>every_unit_leader = { ... }</code>	Executes contained effects on every army leader that meets the limit and is recruited by the country this is contained in.	1.5
random_army_leader	Within country scope only	Unit Leader	<code>random_army_leader = { ... }</code>	Executes contained effects on a random army leader that meets the limit and is recruited by the country this is contained in.	1.5
global_every_army_leader	Always usable	Unit Leader	<code>global_every_army_leader = { ... }</code>	Executes contained effects on every army leader that meets the limit. Preferable to use <code>every_army_leader</code> unless necessary to use <code>global_every_army_leader</code> .	1.5
every_navy_leader	Within country scope only	Unit Leader	<code>every_navy_leader = { ... }</code>	Executes contained effects on every navy leader that meets the limit and is recruited by the country this is contained in.	1.5
random_navy_leader	Within country scope only	Unit Leader	<code>random_navy_leader = { ... }</code>	Executes contained effects on a random navy leader that meets the limit and is recruited by the country this is contained in.	1.5
every_operative	Within country scope or operations only	Operative	<code>every_operative = { ... }</code>	Executes contained effects on every operative that meets the limit and is recruited by the country this is contained in.	1.9
random_operative	Within country scope or operations only	Operative	<code>random_operative = { ... }</code>	Executes contained effects on a random operative that meets the limit and is recruited by the country this is contained in.	1.9
every_character	Within country scope only	Character	<code>every_character = { ... }</code>	Executes contained effects on every character that meets the limit and is recruited by the country this is contained in.	1.11
random_character	Within country scope only	Character	<code>random_character = { ... }</code>	Executes contained effects on a random character that meets the limit and is recruited by the country this is contained in.	1.11
every_country_division	Within country scope only	Division	<code>every_country_division = { ... }</code>	Executes contained effects on every division that meets the limit and is owned by the current country.	1.12
random_country_division	Within country scope only	Division	<code>random_country_division = { ... }</code>	Executes contained effects on a random division that meets the limit and is owned by the current country.	1.12
every_state_division	Within state scope only	Division	<code>every_state_division = { ... }</code>	Executes contained effects on every division that meets the limit and is located within the current state.	1.12
random_state_division	Within state scope only	Division	<code>random_state_division = { ... }</code>	Executes contained effects on a random division that meets the limit and is located within the current state.	1.12

NOTE: Some of these scopes may have no countries/states that match the criteria

Effects with scopes[[编辑](#) | [编辑源代码](#)]

Effects that change the scope include the following:

[start_civil_war](#), which changes it to the rebelling dynamic country.

[create_dynamic_country](#), which changes it to the newly-created dynamic country.

Dual scopes[[编辑](#) | [编辑源代码](#)]

The following scopes can be used either as effect or trigger scopes; some can also be used as the right side of some effects and triggers as a target. If usage as a target is possible, it's marked within the table.

Several dual scopes may have a scope that varies depending on where it's used, such as variables, which can be set to anything.

Dual scopes: 折叠							
Name	Usage	Target type	Example	Description	Usable as target	Version Added	
TAG	Always usable	Country scope	<code>SOV = { country_event = my_event.1 }</code>	The country defined by the tag or tag alias. Tag aliases are defined in /Hearts of Iron IV/common/country_tag_aliases, as a way to refer to a specific country (such as a side in a civil war) in addition to its actual tag.	✓	1.0	
<state_id>	Always usable	State scope	<code>123 = { transfer_state_to = SCO }</code>	The state defined by this id.	✓	1.0	
<character>	Always usable	Character scope	<code>ENG_theodore_makhno = { set_nationality = UKR }</code>	On game versions prior to 1.12.8, the character must be already recruited by the country this is scoped from.	✓	1.11	
ROOT	Always usable	Depends on usage	<code>ENG = { FRA = { GER = { declare_war_on = { target = ROOT type = annex_everything } } } } } #GER declares war on ENG (if there is no scope before ENG)</code>	Targets the root node of the block, an inherent property of each block. Most commonly, this is the default scope: for example, ROOT <i>within a national focus</i> will always refer to the country doing the focus and ROOT <i>within a event</i> will always refer to the country getting the event. However, some blocks do distinguish between the default scope and ROOT, such as certain scripted GUI contexts or certain on actions . If a block doesn't have ROOT defined (such as on_startup in on actions), then it is impossible to use it.	✓	1.0	
THIS	Always usable	Depends on usage	<code>set_temp_variable = { target_country = THIS.id }</code>	Targets the current scope where it's used. For example, when used in every_state , it will refer to the state that's currently being evaluated. Primarily useful for variables or for built-in localisation commands . There is little to no usage outside of these two cases.	✓	1.0	

Name	Usage	Target type	Example	Description	Usable as target	Version Added
PREV	Always usable	Depends on usage	<pre>FRA = { random_country = { GER = { declare_war_on = { target = PREV type = annex_everything } } } } #Germany declares war on random_country</pre>	Targets the scope that the current scope is contained in. Can have additional applications where the assumed default scope differs from the ROOT, such as in state events or some on_actions. Can be chained indefinitely as PREV.PREV. Commonly results in broken-looking tooltips: what's shown to the player doesn't always correlate with reality.	✓	1.0
FROM	Always usable	Depends on usage	<pre>declare_war_on = { target = FROM type = annex_everything } FROM = { load_oob = defend_ourselves }</pre>	Can be chained indefinitely as FROM.FROM. Used to target various hardcoded scopes inherent to the block, often a secondary scope in addition to ROOT. For example: In events , this refers to the country that sent the event (i.e. if the event was fired using an effect , then it's the ROOT scope where it was fired). In targeted decisions or diplomacy scripted triggers , this refers to the scope that is targeted.	✓	1.0
overlord	Within country scope only	Country scope	<pre>overlord = { ... }</pre>	The overlord of the country if it is a subject. Subject to the 'invalid event target' error.	X	1.3
faction_leader	Within country scope only	Country scope	<pre>faction_leader = { add_to_faction = FROM }</pre>	Faction leader of the faction the country is a part of. Subject to the 'invalid event target' error.	X	1.10.1
owner	Within state or combatant scope only	Country scope	<pre>owner = { add_ideas = owns_this_state }</pre>	In state scope, the country that owns the state. In combatant scope, the country that owns the divisions. In character scope, the country that has recruited the character. Subject to the 'invalid event target' error when used for a state.	X	1.0
controller	Within state scope only	Country scope	<pre>controller = { ROOT = { create_wargoal = { target = PREV type = take_state_focus generator = { 123 } } } }</pre>	The controller of the current state. Subject to the 'invalid event target' error.	X	1.0
capital_scope	Within country scope only	State scope	<pre>capital_scope = { ... }</pre>	The state where the capital of the current country is located in. Subject to the 'invalid event target' error in rare cases.	X	1.0
event_target: <event_target_key>	Always usable	Depends on usage	<pre>event_target:my_event_target = { ... }</pre>	Saved event target or global event target , with no space after the colon. Subject to the 'invalid event target' error.	✓	1.0
var:<variable>	Always usable	Depends on usage	<pre>var:my_variable = { ... } add_to_faction = my_variable OR add_to_faction = var:my_variable</pre>	Variable set to a scope. When used as a target rather than a scope, the var: can be omitted in most cases.	✓	1.5

Invalid event target[[编辑](#) | [编辑源代码](#)]

See also: [Event targets](#)

In regards to some dual scopes, a possible logged error to get while using them is "Invalid event target", as in `common/national_focus/generic.txt:690: controller: invalid event target: controller`, while the scope being used is not necessarily an event target. This refers to the scope not having any defined target in the context that it is used, i.e. it is impossible to select any single target when it is used. In case of `controller = { ... }` as in the example, this means that the scope is checked or executed in a state that isn't controlled by any country. Such states are rather unstable and can cause crashes easily (such as if evaluated for an air mission by AI or if doing almost any effect on them), so this error occurring should never happen.

In practice, this gets skipped over entirely when evaluating the effects or triggers: none of the effects would be executed; as a trigger it'll not come up as either true or false. However, since this can be checked every tick, leaving it as is can result in cluttering the error log. To avoid this, it's possible to use the if statement in [effects](#) or [triggers](#) in such a manner that the dual scope would only be generated when needed, such as by checking that the country is indeed a subject before checking the overlord.

Any scope[[编辑](#) | [编辑源代码](#)]

Can be used in **country**, **state** or **character** scopes.

General[[编辑](#) | [编辑源代码](#)]

General any-scoped effects:
[折叠](#)

Name	Parameters	Examples	Description	Notes	Version Added
add_dynamic_modifier	<pre>modifier = <modifier_string> The name of the Modifier. scope = <scope> If you specify it, your dynamic modifier will be scoped to this scope. Optional. days = x The modifiers will be removed after x days have passed. Optional.</pre>	<pre>add_dynamic_modifier = { modifier = example_dynamic_modifier scope = GER days = 14 }</pre>	Adds a dynamic modifier to the specified scope (the default scope is ROOT). It will be updated daily, unless forced to update early by <code>force_update_dynamic_modifier</code> effect.	Examples can be found in <code>/Hearts of Iron IV/common/dynamic_modifiers/*.txt</code> . Any modifiers that use variables within of the dynamic modifier will not show up in the tooltip of this effect, while those that are set to a static value will.	1.6
remove_dynamic_modifier	<pre>modifier = <modifier_string> The name of the Modifier.</pre>	<pre>remove_dynamic_modifier = { modifier = sabotaged_ressources }</pre>	Removes a dynamic modifier from the current scope	Examples can be found in <code>/Hearts of Iron IV/common/dynamic_modifiers/*.txt</code>	1.6
force_update_dynamic_modifier	<pre><bool> Boolean.</pre>	<pre>force_update_dynamic_modifier = yes</pre>	Forces an update to the effects given by variables within dynamic modifiers.	An update is done daily by default; this can be used if the applied values need to be changed urgently, such as if modifiers are checked or used later in the effect block.	1.6
set_global_flag	<pre><flag> An unique string to identify the global flag with. OR flag = <flag></pre>	<pre>set_global_flag = my_flag set_global_flag = { flag = my_flag days = 123 }</pre>	Defines a global flag.	No tooltip is shown. The flag in this effect is used in the meaning of 'boolean flag'. used to store information.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
	The flag to set. days = <int> Sets the flag to last for the specified amount of days. Optional. value = <int> The new value of the flag on the scale from -2 147 483 648 to 2 147 483 647.	<pre>value = 1 }</pre>			
play_song	<song title from .asset> A music file located in the music folder and .asset	play_song = "general_peace_1"	Plays an audio track	The song must be defined in a music station in order to work. More information can be found in the Music modding page. If you wish to simply play a sound, the sound_effect effect should be used instead.	1.9.3
clr_global_flag	<flag> The unique string of a global flag to clear.	clr_global_flag = my_flag	Clears a defined global flag.	No tooltip is shown	1.0
modify_global_flag	flag = <flag> The global flag to modify. value = <value> The value to set it to: 0 or 1	modify_global_flag = { flag = my_flag value = 1 }	Modifies a global flag.	Use variables instead.	1.0
custom_effect_tooltip	<string> A localized string to display in the tooltip.	custom_effect_tooltip = my_tooltip_tt	Displays a localized key in the effect tooltip.		1.0
log	<string> An string to in the game.log	log = "myVariable: [?myVariable]"	Displays a string in the user directory's /Hearts of Iron IV/logs/game.log file when executed, as well as showing up in the console if it is open when the logging effect was executed.	Accepts all localisation commands (e.g. [Root.GetName], [GetDateText], etc)	1.5
save_event_target_as	<string> An unique string to identify the event target with.	capital_scope = { save_event_target_as = my_state }	Saves the current scope as a key. Is cleared once execution ends (i.e. end of event).	Use event_target:<key> to access the scope. Do not use in Scripted GUIs.	1.0
save_global_event_target_as	<string> An unique string to identify the global event target with.	random_other_country = { save_global_event_target_as = my_country }	Saves the current scope as a key. Persists after execution until cleared via effect.	Use event_target:<key> to access the scope. Do not use in Scripted GUIs.	1.0
clear_global_event_target	<string> The unique string of the global event target to clear.	clear_global_event_target = my_country	Clears a specific global event target.		1.0
clear_global_event_targets	yes Boolean.	clear_global_event_targets = yes	Clears all global event targets.		1.0
sound_effect	<string> A sound reference from an .asset file.	sound_effect = "boom"	Plays the specified sound once.	The sound effect must be properly defined in /Hearts of Iron IV/sound/ More info can be found in the Sound modding article.	1.0
randomize_weather	<int> A seed integer.	randomize_weather = 12345	Randomizes the weather with the specified seed.		1.0
set_province_name	id = <id> The id of the province to be changed. name = <string> The name to change the province to.	set_province_name = { id = 325 name = LOC_KEY } set_province_name = { id = 325 name = "New Name" }	Changes the specified province/victory point's name to the specified name.	Localisation keys are to be defined in /Hearts of Iron IV/localisation /*_1_<language>.yaml	1.3
reset_province_name	<id> The id of the province to reset.	reset_province_name = 325	Resets the specified province's name.		1.3
damage_units	province = <id> Province where to damage units. state = <id> State where to damage units. region = <id> Strategic region where to damage units. limit = { <triggers> } Will only delete units if the triggers within are met for the country that owns the units. damage = <fraction> The percentage of damage done to units. org_damage = <fraction> The percentage of damage done to units to organisation in particular. str_damage = <fraction> The percentage of damage done to units to strength in particular. ratio = <yes> Will damage a ratio damage to total organisation/strength of unit if set. template = <string> If specified, requires the template name to match. army = <bool> Will damage the army units. navy = <bool> Will damage the navy units.	damage_units = { province = 42 state = 5 region = 5 limit = { has_country_flag = TAG_test } damage = 0.5 org_damage = 0.5 str_damage = 0.5 ratio = yes template = "template_name" army = no navy = yes }	Damages units in the specified area.		1.11
create_entity	entity = <gfx_entry> The entity to spawn, defined within /Hearts of Iron IV/gfx/entities/*.asset files. id = int	create_entity = { entity = entity_name id = 123 var = var_name x = 42	Creates an entity.	Uses the the same coordinate system that the map uses . A positive change in rotation results in counter-clockwise rotation, a full 360 degrees rotation is	1.11

Name	Parameters	Examples	Description	Notes	Version Added
	A number ID which can be referred to by other effects. Optional. var = <variable> If provided, the id of the entity will be stored using this variable. Optional. x = <int> The X position of the entity. y = <int> The Y position of the entity. z = <int> The Z position of the entity. province = <int> The province the middle of which to use as the entity's position. state = <int> The state the middle of which to use as the entity's position. rotation = <decimal> The rotation of the entity in radians. scale = <decimal> The size of the entity. min_zoom = <decimal> Minimum zoom level needed to be able to see the entity. visible = <scripted_trigger> The scripted trigger that must be met for a country for it to see the entity.	y = 21 z = 3 province = 123 state = 42 rotation = 1.2 scale = 10.0 min_zoom = 100.0 visible = scripted_trigger_name }		approximately 6.28 radians. For comparison, default minimum zoom level (closest to the map) is 50 units, while default maximum zoom level is 3000 units.	
destroy_entity	<id> The ID of the entity to destroy.	destroy_entity = 123	Deletes an entity	IDs are set by the create_entity effect .	1.11
set_entity_movement	id = <ID> The ID of the entity to modify. ratio = <int> Distance between starting position and target position where the entity is to be placed. rotation = <int> The rotation to apply <i>after</i> the positioning. start and target arguments: x = <int> The X position of the point. y = <int> The Y position of the point. z = <int> The Z position of the point. province = <int> The province the middle of which to use as the point. state = <int> The state the middle of which to use as the point.	set_entity_movement = { id = 123 start = { x = 42 y = 21 z = 3 } target = { province = 124 } ratio = 0.5 rotation = 1.2 }	Sets the position and rotation of an entity using two coordinates.	IDs are set by the create_entity effect . Uses the the same coordinate system that the map uses . A positive change in rotation results in counter-clockwise rotation, a full 360 degrees rotation is approximately 6.28 radians.	1.11
set_entity_position	id = <id> x = <int> y = <int> z = <int> province = <int> state = <int>	set_entity_position = { id = 123 x = 42 y = 21 z = 3 province = 123 state = 42 }	Sets the position of an existing entity	IDs are set by the create_entity effect . Uses the the same coordinate system that the map uses .	1.11
set_entity_rotation	id = <ID> The ID of the entity to modify. rotation = <decimal> The new angle in radians.	set_entity_rotation = { id = 123 rotation = 0.23 }	Sets the currently-facing angle of an existing entity.	IDs are set by the create_entity effect . A positive change results in counter-clockwise rotation, a full 360 degrees rotation is approximately 6.28 radians.	1.11
set_entity_scale	id = <ID> The ID of the entity to modify. scale = <decimal> The scale to change the entity to.	set_entity_scale = { id = 123 scale = 5.0 }	Sets the size of an existing entity.	IDs are set by the create_entity effect .	1.11
set_entity_animation	id = <int> The ID of the entity to modify. animation = <animation_type> The animation entry to apply.	set_entity_animation = { id = 123 animation = "shoot_lasers" }	Sets the animation of a specified entity.	IDs are set by the create_entity effect . Animations are defined within the /Hearts of Iron IV/gfx/models/**/*.*.asset files.	1.11
build_railway	level = <int> Defaults to 1 build_only_on_allied = <bool> No by default, if yes and in a country scope, it will only build on allied territories for the country scoped. fallback = <bool> Defaults to no, if yes each option will try to fallback to the next available one. path = { <list of provinces> } start_province = <int> target_province = <int> start_state = <int> target_state = <int> If using start state/target	build_railway = { level = 1 build_only_on_allied = yes controller_priority = { base = 1 modifier = { tag = MAN add = 2 } } fallback = yes path = { 42 10 20 30 40 84 } start_province = 42 target_province = 84 } build_railway = { level = 1	Adds a railway level between two provinces or along a predefined path.		1.11

Name	Parameters	Examples	Description	Notes	Version Added
	state, the game will pick the provinces with the best supply available. If using state province/target province, the game will link those two provinces.	<pre>build_only_on_allied = yes controller_priority = { base = 1 modifier = { tag = MAN add = 2 } } fallback = yes path = { 50 10 20 30 40 100 } start_state = 50 target_state = 100 }</pre>			

Border wars[[编辑](#) | [编辑源代码](#)]

Border war-related any-scoped effects:

折叠

Name	Parameters	Examples	Description	Notes	Version Added
start_border_war	<pre>change_state_after_war = <bool> Whether the state changes hands after the war. Attacker or Defender scope state = <id> / <variable> The state the side is fighting on. num_provinces = <id> The number of provinces used in the state. on_win = <id> The event to fire for the side on a win. on_lose = <id> The event to fire for the side on a loss. on_cancel = <id> The event to fire for the side on a draw. modifier = <decimal> The modifier on combat. Defaults to 0. dig_in_factor = <decimal> The modifier applied to dig-in bonuses. Defaults to 1. terrain_factor = <decimal> The modifier applied to terrain bonuses. Defaults to 1.</pre>	<pre>start_border_war = { change_state_after_war = no attacker = { state = 527 num_provinces = 4 on_win = japan_border_conflict.2 on_lose = japan_border_conflict.3 on_cancel = japan_border_conflict.4 modifier = 0.1 dig_in_factor = 0 terrain_factor = 0 } defender = { state = 408 num_provinces = 4 on_win = japan_border_conflict.3 on_lose = japan_border_conflict.2 on_cancel = japan_border_conflict.4 } }</pre>	Starts a border war for the specified attacker and defender. The participating countries are the owners of the specified states.		1.5
set_border_war_data	<pre>attacker = <id> / <variable> The attacker state. defender = <id> / <variable> The defender state. attacker_modifier = <id> / <variable> The modifier applied to attacker strength. defender_modifier = <id> / <variable> The modifier applied to attacker strength. combat_width = <id> / <variable> The combat width used in the border war battle.</pre>	<pre>set_border_war_data = { attacker = 527 defender = 408 defender_modifier = 0.15 combat_width = 100 }</pre>	Sets the bonuses or penalties for the attacker and defender in an on-going border war. Used after start_border_war .		1.5
cancel_border_war	<pre>attacker = <id> / <variable> The attacker state. defender = <id> / <variable> The defender state. dont_fire_events = <bool> Stops the events from start_border_war from firing.</pre>	<pre>cancel_border_war = { dont_fire_events = yes defender = 408 attacker = 527 }</pre>	Cancels an on-going border war without a winner.		1.5
finalize_border_war	<pre>attacker = <id> / <variable> The attacker state. defender = <id> / <variable> The defender state. attacker_win = <bool> Makes the attacker the winner. defender_win = <bool> Makes the defender the winner.</pre>	<pre>finalize_border_war = { attacker_win = yes attacker = 527 defender = 408 }</pre>	Ends an on-going border war.		1.5

Country scope[[编辑](#) | [编辑源代码](#)]

The effects here must be used within a **country** scope.

General[[编辑](#) | [编辑源代码](#)]

General country-scoped effects:

折叠

Name	Parameters	Examples	Description	Notes	Version Added
set_country_flag	<pre><flag> An unique string to identify the country flag with. OR flag = <flag> The flag to set. days = <int> Sets the flag to last for the specified amount of days. Optional. value = <int> The new value of the flag on the scale from -2 147 483 648 to 2 147 483 647.</pre>	<pre>set_country_flag = my_flag set_country_flag = { flag = my_flag days = 123 value = 1 }</pre>	Defines a country flag.	No tooltip is shown. The flag in this effect is used in the meaning of 'boolean flag', used to store information. In order to change the flag that represents the country, see cosmetic tags.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
clr_country_flag	<flag> The unique string of a country flag to clear.	clr_country_flag = my_flag	Clears a defined country flag.		1.0
modify_country_flag	flag = <flag> The country flag to modify. value = <value> The value to set it to: 0 or 1	modify_country_flag = { flag = my_flag value = 1 }	Modifies a country flag.	Use variables instead.	1.3
country_event	id = <event> The event to fire. days = <int> / <variable> Fires the event in the specified number of days. Optional. hours = <int> / <variable> Fires the event in the specified number of hours. Optional. random_hours = <int> / <variable> Adds a random number (between 0 and random_hours, inclusive) of hours to the scheduled fire time. Optional. random_days = <int> / <variable> Adds a random number (between 0 and random_days, inclusive) of days to the scheduled fire time. Optional.	country_event = { id = my_event.1 days = 10 random_hours = 12 random_days = 10 } country_event = my_event.1	Fires the specified event for the current country.	Where triggers do not need to be repeatedly checked random can be a performance light alternative to mean_time_to_happen for scheduling events. Shortened variant exists if the event's ID is used instead of arguments.	1.0
news_event	id = <event> The event to fire. days = <int> / <variable> Fires the event in the specified number of days. Optional. hours = <int> / <variable> Fires the event in the specified number of hours. Optional. random_hours = <int> / <variable> Adds a random number (between 0 and random_hours, inclusive) of hours to the scheduled fire time. Optional. random_days = <int> / <variable> Adds a random number (between 0 and random_days, inclusive) of days to the scheduled fire time. Optional.	news_event = { id = my_event.1 days = 10 random_hours = 12 random_days = 10 } news_event = my_event.1	Fires the specified news event for the current country.	The news event uses a different interface to the country event. Where triggers do not need to be repeatedly checked random can be a performance light alternative to mean_time_to_happen for scheduling events. Shortened variant exists if the event's ID is used instead of arguments.	1.0
set_cosmetic_tag	<string> The cosmetic tag to switch to.	set_cosmetic_tag = SAF_SOV_communism	Makes the current scope use the specified cosmetic tag, changing name and flag.		1.3
drop_cosmetic_tag	<bool> Boolean.	drop_cosmetic_tag = yes	Makes the current scope drop the current cosmetic tag they are using.		1.3
set_rule	<rule> Boolean. desc = <localisation key> The localisation used as the rule's description.	set_rule = { desc = TAG_my_rule_description can_create_factions = yes }	Toggles the special game rules for the current scope.	Possible rules: <ul style="list-style-type: none">can_boost_other_ideologiescan_create_factionscan_declare_war_on_same_ideologycan_declare_war_without_wargoal_when_in_warcan_decline_call_to_warcan_force_governmentcan_generate_female_acescan_guarantee_other_ideologiescan_join_factionscan_join_factions_not_allowed_diplomacycan_join_opposite_factionscan_lower_tensioncan_not_declare_warcan_occupy_non_warcan_only_justify_war_on_threat_countrycan_puppetcan_send_volunteerscan_use_kamikaze_pilotsunits_deployed_to_overlord (subjects only)	1.0

Name	Parameters	Examples	Description	Notes	Version Added
set_party_rule	ideology = <ideology group> Ideology group of the party. desc = <localisation key> A description used for the rule. Optional, defaults to being the same as default. <rule> = <bool> Rule's new value.	set_party_rule = { ideology = democratic desc = TAG_my_rule_description can_create_factions = yes }	Toggles the special game rules for the current scope's political party.		1.12
set_relation_rule	target = <scope> Target of the rule. desc = <localisation key> A description used for the rule. Optional, defaults to being the same as default. <rule> = <bool> Rule's new value.	set_relation_rule = { target = SOV desc = TAG_my_rule_description can_not_declare_war = yes }	Toggles the special game rules for the current scope in diplomacy towards the specified country only.		1.12
scoped_sound_effect	<string> A sound reference from an .asset file.	scoped_sound_effect = "boom"	Plays the specified sound once only for the current country.	The sound effect must be properly defined in /Hearts of Iron IV/sound/ More info can be found in the Sound modding article.	1.6
scoped_play_song	<song title from .asset> A music file located in the music folder and .asset	scoped_play_song = "general_peace_1"	Plays an audio track for the specified country only.	The song must be defined in a music station in order to work. More information can be found in the Music modding page. If you wish to simply play a sound, the scoped_sound_effect effect should be used instead.	1.9.3
goto_province	<id> The id of the province go to.	goto_province = 325	Moves the camera position over the specified province.		1.0
goto_state	<scope> / <variable> The id of the state go to.	goto_state = 1 goto_state = var:some_state	Moves the camera position over the specified state.		1.0
change_tag_from	<scope> / <variable> The country to change from.	change_tag_from = ROOT change_tag_from = var:from.country	Switches the player to the current scope from the target scope.	The country the player becomes needs to be the scope in which the command is used.	1.0
reserve_dynamic_country	<bool>	reserve_dynamic_country = yes	Reserves the dynamic country, making sure that it does not get recycled for civil war even if it does not exist.	Usually used in combination with create_dynamic_country .	1.9
force_update_map_mode	limit = { ... } Triggers required for the map mode to refresh. Optional. mapmode = <id> The ID of the custom map mode.	force_update_map_mode = { limit = { is_ai = no } mapmode = my_map_mode }	Forcefully refreshes the specified mapmode for the player, rather than waiting for a daily update.	Map modes are defined in /Hearts of Iron IV/common/map_modes /*.txt	1.11
add_ai_strategy	type = <type> The type of strategy. id = <scope> What country the strategy is against. value = <int> The weighting added by the strategy.	add_ai_strategy = { type = alliance id = GER value = 200 }	Sets an AI strategy for the current scope.	See AI Modding for more details.	1.0
create_dynamic_country	original_tag = <tag> The original tag to be used by the country. copy_tag = <tag> If specified, copies stuff from this tag rather than the original tag. <effects> Effects that will be executed on the new dynamic country.	create_dynamic_country = { original_tag = POL copy_tag = SOV add_political_power = 100 transfer_state = 123 }	Creates a new dynamic country, akin to ones used in civil wars.	The reserve_dynamic_country effect can be used if the dynamic country does not yet exist in order to ensure that it does not get overwritten by other creations of dynamic countries. If this is not done, the dynamic country will immediately stop existing if no states are transferred in the same scope. Every state of the original country immediately gets set as a dynamic country's core: if that's unneeded, the cores would need to be removed after creation.	1.9

States[[编辑](#) | [编辑源代码](#)]

These effects in particular are country-scoped effects that are related to states rather than effects within the state scope.

State-related country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
add_state_core	<scope> / <variable> The state to add core to.	add_state_core = 345	Adds a core for the current scope to the specified state.		1.0
remove_state_core	<scope> / <variable> The state to remove core from.	remove_state_core = 345	Removes the core of the current scope from the specified state.		1.0
set_capital	<scope> / <variable> The state to make capital.	set_capital = {state = 345} set_capital = { state = 345 remember_old_capital = no }	Makes the specified state the current scope's capital state.	Syntax has been changed in 1.11. It was "set_capital = 345" Old capital is remembered, if not specified otherwise.	1.0
add_state_claim	<scope> / <variable> The state to add a claim to.	add_state_claim = 345	Adds a claim for the current scope on the specified state.		1.0

Name	Parameters	Examples	Description	Notes	Version Added
remove_state_claim	<code><scope></code> / <code><variable></code> The state to remove the claim from.	<code>remove_state_claim = 345</code>	Removes a claim of the current scope from the specified state.		1.0
set_state_owner	<code><scope></code> / <code><variable></code> The state to change ownership of.	<code>set_state_owner = 345</code>	Makes the current scope the owner of the specified state.	This can fail to carry over the control, so it's recommended to instead use transfer_state unless transferring the ownership without transferring over the control.	1.0
set_state_controller	<code><scope></code> / <code><variable></code> The state to change controller of.	<code>set_state_controller = 345</code>	Makes the current scope the controller of the specified state.		1.0
transfer_state	<code><scope></code> / <code><variable></code> The state to change owner and controller of.	<code>transfer_state = 345</code>	Makes the current scope the owner and controller of the specified state.	transfer_state_to exists as a state-scoped variant.	1.0
set_province_controller	<code><id></code> The province to change controller of.	<code>set_province_controller = 2999</code>	Changes the controller of the specified province to the current scope.		1.0

Mana[\[编辑 | 编辑源代码\]](#)

Mana in this usage means political power, stability, war support, and other values in the topbar. Fuel is, instead, in the [resources section](#), while convoys can be added/removed with [add_equipment_to_stockpile](#).

Mana-related country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
add_political_power	<code><int></code> / <code><variable></code> The amount to add.	<code>add_political_power = 100</code> <code>add_political_power = var:my_var</code>	Adds the specified amount of political power to the current scope.		1.0
set_political_power	<code><int></code> / <code><variable></code> The amount to add.	<code>set_political_power = 100</code>	Sets the specified amount of political power for the current scope.		1.0
add_stability	<code><int></code> / <code><variable></code> The amount to add.	<code>add_stability = 0.1</code>	Adds to the current stability value for the current scope.	Stability values are between 0 and 1.	1.5
set_stability	<code><int></code> / <code><variable></code> The amount to add.	<code>set_stability = 0.5</code>	Sets the current stability value for the current scope.	Stability values are between 0 and 1.	1.5
add_war_support	<code><int></code> / <code><variable></code> The amount to add.	<code>add_war_support = 0.1</code>	Adds to the current war support value for the current scope.	War Support values are between 0 and 1.	1.5
set_war_support	<code><int></code> / <code><variable></code> The amount to set.	<code>set_war_support = 0.5</code>	Sets the current war support value for the current scope.	War Support values are between 0 and 1.	1.5
add_command_power	<code><int></code> / <code><variable></code> The amount to add.	<code>add_command_power = 100</code>	Adds the specified amount of command power to the current scope.		1.5
add_manpower	<code><int></code> / <code><variable></code> The amount to add.	<code>add_manpower = 100000</code> <code>add_manpower = var:my_var</code>	Adds the specified amount of manpower to the current scope.		1.0
army_experience	<code><float></code> / <code><variable></code> The amount to add.	<code>army_experience = 10</code>	Adds the specified amount of army experience to the current scope.		1.0
navy_experience	<code><float></code> / <code><variable></code> The amount to add.	<code>navy_experience = 10</code>	Adds the specified amount of navy experience to the current scope.		1.0
air_experience	<code><float></code> / <code><variable></code> The amount to add.	<code>air_experience = 10</code>	Adds the specified amount of air experience to the current scope.		1.0

Politics[\[编辑 | 编辑源代码\]](#)

Political country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
hold_election	<code><scope></code> The country to hold an election for.	<code>hold_election = ROOT</code>	Executes the events in the on_new_term_election on action for the current scope.		1.0
add_popularity	<code>ideology = <ideology></code> The party to change. <code>popularity = <int></code> / <code><variable></code> The amount of popularity to change.	<code>add_popularity = {</code> <code>ideology = fascism</code> <code>popularity = -0.5</code> <code>}</code>	Adjusts the popularity for the specified party in the current scope.	Values used are 0 to 1.	1.0
set_politics	<code>ruling_party = <ideology></code> The party to set. <code>elections_allowed = <bool></code> Whether elections are allowed. Optional. <code>last_election = <date></code> When the last election was. Optional. <code>election_frequency = <int></code> How often in months an election occurs. Optional. <code>long_name = <string></code> The long name of the country's new ruling party, appearing when hovering over it. Optional. <code>name = <string></code> The name of the country's new ruling party. Optional.	<code>set_politics = {</code> <code>ruling_party = democratic</code> <code>elections_allowed = no</code> <code>last_election = "1935.12.17"</code> <code>election_frequency = 48</code> <code>long_name = TAG_party_long</code> <code>name = TAG_party</code> <code>}</code>	Sets the political party setup for the current scope.	Previously included the "parties" parameter, which has been deprecated by the "set_popularities" command. The "parties" parameter no longer works as of version 1.7	1.0 (Updated 1.7)

Name	Parameters	Examples	Description	Notes	Version Added
set_popularities	<code><ideology> = <int>/<variable></code> The popularity to set.	<code>set_popularities = { democratic = 50 neutrality = 15 fascism = 30 communism = 5 }</code>	Sets the political party popularities for the current scope.	The popularities must add up to 100, otherwise the command will have no effect.	1.7
set_political_party	<code>ideology = <ideology></code> The party to change. <code>popularity = <int></code> The amount of popularity to set.	<code>set_political_party = { ideology = fascism popularity = 50 }</code>	Sets the popularity for the specified political party in the current scope.		1.0
set_party_name	<code>ideology = <ideology></code> The party to change. <code>long_name = <string></code> The new full name for the party. <code>name = <string></code> The new short name for the party.	<code>set_party_name = { ideology = neutrality long_name = GER_neutrality_party_kaiserreich_long name = GER_neutrality_party_kaiserreich }</code>	Changes the name of the specified political party for the current scope.		1.0

Balance of power[[编辑](#) | [编辑源代码](#)]

Balance of power is defined in `/Hearts of Iron IV/common/bop/*.txt` files.

Balance of power-related country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
set_power_balance	<code>id = <BoP ID></code> Balance of power to set/modify. <code>left_side = <BoP side ID></code> The left side of the BoP. <code>right_side = <BoP side ID></code> The right side of the BoP. <code>set_default = <bool></code> Whether the BoP is considered default. Defaults to false. <code>set_value = <decimal></code> The new value of the BoP. Optional, defaults to not changing the value.	<code>set_power_balance = { id = my_bop left_side = my_bop_left_side right_side = my_bop_right_side }</code>	Sets a new balance of power or edits the existing one.	Necessary for a balance of power to appear.	1.12
remove_power_balance	<code>id = <BoP ID></code> Balance of power to modify.	<code>remove_power_balance = { id = my_bop }</code>	Removes the balance of power in entirety.		1.12
add_power_balance_value	<code>id = <BoP ID></code> Balance of power to modify. <code>value = <decimal></code> The value to add. <code>tooltip_side = <BoP side ID></code> The side to show in the tooltip. Optional.	<code>add_power_balance_value = { id = my_bop value = -0.1 tooltip_side = my_bop_side }</code>	Pushes the balance of power towards one side.		1.12
add_power_balance_modifier	<code>id = <BoP ID></code> Balance of power to modify. <code>modifier = <static modifier></code> The static modifier to apply.	<code>add_power_balance_modifier = { id = my_bop modifier = my_static_modifier }</code>	Applies a balance of power modifier.		1.12
remove_power_balance_modifier	<code>id = <BoP ID></code> Balance of power to modify. <code>modifier = <static modifier></code> The static modifier to apply.	<code>remove_power_balance_modifier = { id = my_bop modifier = my_static_modifier }</code>	Cancels a balance of power modifier.		1.12
remove_all_power_balance_modifiers	<code>id = <BoP ID></code> Balance of power to modify.	<code>remove_all_power_balance_modifiers = { id = my_bop }</code>	Cancels all balance of power modifiers.		1.12
set_power_balance_gfx	<code>id = <BoP ID></code> Balance of power to modify. <code>side = <BoP side ID></code> The side whose GFX to change. <code>gfx = <sprite></code> The sprite to change the GFX to.	<code>set_power_balance_gfx = { id = my_bop side = my_bop_side gfx = GFX_my_bop_side_new }</code>	Changes the appearance of one of the sides within the balance of power.	Sprites are defined within <code>/Hearts of Iron IV/interface/*.gfx</code> files.	1.12

Diplomacy[[编辑](#) | [编辑源代码](#)]

Diplomatic country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
set_major	<code><bool></code> Boolean.	<code>set_major = yes</code>	Makes the current scope a major country.		1.0
create_faction	<code><string></code> The name of the faction.	<code>create_faction = MY_FACTION_NAME</code>	Creates a faction with the specified name for the current scope. The current scope and any subjects automatically join the faction.		1.0
add_to_faction	<code><TAG></code> The TAG of the nation to add to the faction of the current scope.	<code>FACTION_LEADER_TAG = { add_to_faction = TAG_TO_ADD }</code>	Adds the TAG_TO_ADD to the faction of the FACTION_LEADER_TAG.		1.0
leave_faction	<code><bool></code> Boolean.	<code>leave_faction = yes</code>	Removes the current scope from the faction they are part of.		1.5
remove_from_faction	<code><TAG></code> The target country.	<code>remove_from_faction = TAG_TO_REMOVE</code>	Removes the specified scope from the faction led by the current scope.		1.0

Name	Parameters	Examples	Description	Notes	Version Added
dismantle_faction	<bool> Boolean.	dismantle_faction = yes	Dismantles the faction of the current scope.		1.0
set_faction_name	Sets a faction name as the loc name.	set_faction_name = SOME_LOC_KEY	Changes faction names.		1.6
set_faction_leader	<bool>	set_faction_leader = yes	Sets the current country as the faction leader.		1.0
set_faction_spymaster	<bool>	set_faction_spymaster = yes	Sets the current country as the faction spymaster.		1.9
release	<scope> The target country.	release = GER	Releases the specified non-existent country as a free nation within the current country's owned states.	The effect does nothing if the country exists. All states that are cored by the specified country will be given to it. If the current country has a core on a state transferred to the released country, the core will be lost. If looking to make a subject into an independent nation, use set_autonomy. States that are owned but not controlled will be transferred to the released country, but won't be controlled by it.	1.0
release_on_controlled	<scope> The target country.	release_on_controlled = GER	Releases the specified non-existent country as a free nation within the current country's controlled states.	The effect does nothing if the country exists. All states that are cored by the specified country will be given to it. If the current country has a core on a state transferred to the released country, the core will be lost.	1.9.1
release_puppet	<scope> The target country.	release_puppet = GER	Releases the specified non-existent country as a puppet of the current scope within the current country's owned states.	The effect does nothing if the country exists. All states that are cored by the specified country will be given to it. If the current country has a core on a state transferred to the released country, the core will be lost. States that are owned but not controlled will be transferred to the released country, but won't be controlled by it.	1.0
release_puppet_on_controlled	<scope> The target country.	release_puppet_on_controlled = GER	Releases the specified non-existent country as a puppet of the current scope within the current country's controlled states.	The effect does nothing if the country exists. All states that are cored by the specified country will be given to it. If the current country has a core on a state transferred to the released country, the core will be lost.	1.9.1
release_autonomy	target = <scope> / <variable> The subject country. autonomy_state = <type> The type of autonomy state to set. freedom_level = <float> The new freedom level value. Optional.	release_autonomy = { target = VIN autonomy_state = autonomy_puppet freedom_level = 0.5 }	Releases the specified non-existent country as a subject of the specified autonomy of the current scope within the current country's owned states.	The effect does nothing if the country exists. All states that are cored by the specified country will be given to it. If the current country has a core on a state transferred to the released country, the core will be lost. States that are owned but not controlled will be transferred to the released country, but won't be controlled by it. The autonomy states are found in /Hearts of Iron IV/common/autonomous_states/*.txt.	1.3
give_guarantee	<scope> The target country.	give_guarantee = GER	The current scope guarantees the target country.	diplomatic_relation effect can be used to remove it.	1.0
give_military_access	<scope> The target country.	give_military_access = GER	The current scope grants military access to the target country.	diplomatic_relation effect can be used to remove it.	1.0
recall_attache	<scope> The target country with an attache.	recall_attache = GER	Recalls the current scope's attaché from the specified country.		1.5
diplomatic_relation	country = <scope> The target country to alter the relationship with ROOT. relation = <type> The relation to change. active = <bool> Whether the relation is started or broken.	diplomatic_relation = { country = SOV relation = guarantee active = no }	Used to define a diplomatic relation between the current scope and target scope country.	Possible relations: <ul style="list-style-type: none">• non_aggression_pact• guarantee• puppet• military_access• docking_rights	1.0
add_opinion_modifier	target = <scope> The target country. modifier = <modifier> The opinion modifier to add.	add_opinion_modifier = { target = GER modifier = faction_traitor }	The current scope gains the specified opinion modifier towards the target scope . Can also be used to modify trade relations by adding 'trade = yes' in the opinion <modifier> in /Hearts of Iron IV/common/opinion_modifiers/*.txt. If used with a trade opinion_modifier the behaviour is reversed, meaning that the target gains the trade opinion towards the current scope .	Opinion modifiers are found in /Hearts of Iron IV/common/opinion_modifiers/*.txt.	1.0
remove_opinion_modifier	target = <scope> The target country. modifier = <modifier> The opinion modifier to remove.	remove_opinion_modifier = { target = GER modifier = faction_traitor }	The current scope loses the specified opinion modifier towards the target scope .	Opinion modifiers are found in /Hearts of Iron IV/common/opinion_modifiers/*.txt.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
reverse_add_opinion_modifier	target = <scope> The target country. modifier = <modifier> The opinion modifier to add.	reverse_add_opinion_modifier = { target = GER modifier = faction_traitor }	The target scope gains the specified opinion modifier towards the current scope .	Opinion modifiers are found in /Hearts of Iron IV/common/opinion_modifiers/*.txt. Useful for when you don't know what the current scope will be.	1.0
add_relation_modifier	target = <scope> The target country. modifier = <modifier> The relation modifier to add.	add_relation_modifier = { target = SWE modifier = HUN_dynastic_ties_license }	The current scope gains the specified relation modifier towards the target scope .	Relation modifiers are found in /Hearts of Iron IV/common/modifiers/*.txt files, used to apply a targeted modifier with a non-static target. To change the diplomatic opinion of a country, see add opinion modifier .	1.4
remove_relation_modifier	target = <scope> The target country. modifier = <modifier> The relation modifier to remove.	remove_relation_modifier = { target = SWE modifier = HUN_dynastic_ties_license }	The current scope loses the specified relation modifier for towards the target scope .	Relation modifiers are found in /Hearts of Iron IV/common/modifiers/*.txt, used to apply a targeted modifier with a non-static target. To change the diplomatic opinion of a country, see remove opinion modifier .	1.4
add_collaboration	target = <scope> The target country. value = <0-1> How much collaboration to add.	add_collaboration = { target = TAG value = 0.3 }	Adds collaboration in TAG with the scoped country.		1.9
set_collaboration	target = <scope> The target country. value = <0-1> How much collaboration will be set.	set_collaboration = { target = TAG value = 0.3 }	Sets the collaboration in TAG with the scoped country.		1.9
recall_volunteers_from	<tag> The target country.	recall_volunteers_from = SPR	Recalls volunteers sent to the specified country back to the current country.		1.9
set_occupation_law	<law ID> The new occupation law enacted by the previous scope OR default_law.	USA = { GER = { set_occupation_law = foreign_civilian_oversight } } # Changes USA's occupation law for GER. USA = { USA = { set_occupation_law = default_law } } # Changes the USA's default occupation law to the default.	Sets the occupation law of the country.	PREV will be the country for whom the occupation law will be changed. If PREV is not a country, nothing changes. If PREV is the same country, changes the default occupation law. If PREV is different, default_law resets the country-specific law to the global default, otherwise it resets the default law to the occupation law with starting_law = yes in definition.	1.12
set_occupation_law_where_available	<law ID> The new occupation law enacted by the previous scope OR default_law.	USA = { GER = { set_occupation_law_where_available = foreign_civilian_oversight } } # Changes USA's occupation law for GER where possible. USA = { USA = { set_occupation_law_where_available = default_law } } # Changes the USA's default occupation law to the default where possible.	Sets the occupation law of the country.	Identical to set_occupation_law , except if the law is impossible to set, tries again at every smaller sub-set: if default is impossible, tries every single individual occupied country; if the country's law is impossible to change, tries every single state within the country.	1.12
send_embargo	<tag> The target country.	send_embargo = ITA	Embargos the target country.		1.12
break_embargo	<tag> The target country.	break_embargo = ITA	Stops embargoing the target country.		1.12

Autonomy[[编辑](#) | [编辑源代码](#)]

Autonomy-related country-scoped effects:

折叠

Name	Parameters	Examples	Description	Notes	Version Added
puppet	<scope> The target country. OR target = <scope> The target country. end_wars = <bool> Whether the target country will peace out in all of its non-civil wars it's participating in. Defaults to true. end_civil_wars = <bool> Whether the target country will peace out in all of its civil wars it's participating in. Defaults to true.	puppet = GER puppet = { target = ITA end_wars = no }	Makes the specified country a subject of the current scope.	The autonomous state picked is one which contains default = yes and where allowed = { ... } is fulfilled within the /Hearts of Iron IV/common/autonomous_states/ definition, rather than necessarily being autonomy_puppet. Results in a crash-to-desktop if the game is unable to find any such autonomous states.	1.0
end_puppet	<scope> The target country.	end_puppet = GER	Removes the subject status between the target and the current scope.	Must be used within the overlord's scope.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
add_autonomy_ratio	value = <float> The freedom score to add. localization = <string> The localization key for the modifier.	add_autonomy_ratio = { value = 0.1 localization = AST_adopt_westminster }	Adds a freedom score ratio modifier to the current scope.	Used in the subject's scope.	1.3
add_autonomy_score	value = <float> The freedom score to add. localization = <string> The localization key for the modifier.	add_autonomy_score = { value = 10 localization = EXAMPLE }	Adds an exact freedom score modifier to the current scope.	Used in the subject's scope.	1.3
set_autonomy	target = <scope> / <variable> The subject country. autonomy_state = <type> The type of autonomy state to set. freedom_level = <float> The new freedom level value. Optional. end_wars = <yes/no> Will end any wars the subject is involved in. end_civil_wars = <yes/no> Will end any civil wars the subject is subject to	set_autonomy = { target = AST autonomy_state = autonomy_free end_wars = no end_civil_wars = no }	Sets the autonomy level for the specified country, including independence .	The autonomy_free state will free the subject, however this effect has to be executed within the scope of the target country's current overlord for this to have effect. The autonomy states are found in /Hearts of Iron IV/common/autonomous_states/*.txt files.	1.3

Governments in exile[[编辑](#) | [编辑源代码](#)]

Government in exile-related country-scoped effects:

Name	Parameters	Examples	Description	Notes	Version Added
add_legitimacy	Adds legitimacy to a government in exile.	add_legitimacy = 10	Adds legitimacy.		1.6
set_legitimacy	Sets the legitimacy of governments in exile.	set_legitimacy = 10	Sets legitimacy.		1.6
become_exiled_in	Makes a country a government in exile in a set country, with a set starting legitimacy.	become_exiled_in = { target = <Host tag> legitimacy = <0-100> (starting legitimacy, optional) }	Creates a government in exile.	Must be fired from ROOT, the country that should be exiled, or a TAG specification must be used.	1.6
end_exile	Ends a government in exile.	end_exile = yes	Ends a government in exile.		1.6

War[[编辑](#) | [编辑源代码](#)]

War-related country-scoped effects:

Name	Parameters	Examples	Description	Notes	Version Added
add_threat	<int> The amount to change by.	add_threat = 10	Adjusts the level of World Tension.		1.0
add_named_threat	threat = <int> The amount to change by. name = <string> The localization string.	add_named_threat = { threat = 5 name = GER_rhineland }	Adjusts the level of World Tension and adds an entry in the World Tension tooltip.		1.0
annex_country	target = <scope> Which country to annex. transfer_troops = yes Whether to transfer the troops of the annexed country.	annex_country = { target = GER transfer_troops = yes }	Annex the specified country for the current scope.	Without transferring troops, the annexed country's divisions' equipment is lost.	1.0
add_to_war	targeted_alliance = <scope> The country to assist. enemy = <scope> The country attacking the ally. hostility_reason = <string> Localization for the reason for joining. Optional.	add_to_war = { targeted_alliance = PREV enemy = HUN hostility_reason = asked_to_join }	Forces the current scope to join the war of the specified ally against the specified enemy.		1.0
declare_war_on	target = <scope> / <variable> The country to attack. type = <wargoal> The wargoal to declare with. generator = { <state id> } The states to supply the wargoal (i.e. take_state_focus).	declare_war_on = { target = GER type = annex_everything }	Makes the current scope declare war on the specified country with the specified wargoal.	Wargoals are found in /Hearts of Iron IV/common/wargoals/*.txt. See also add_civil_war_target in order to assign a war between different countries to be a civil war.	1.0
white_peace	<scope> / <variable> The scope to white peace. OR tag = <scope> / <variable> The scope to white peace. message = <localisation key> The reason for peace showing up in the pop-up.	white_peace = GER white_peace = { tag = GER message = my_peace_tt }	Makes the current scope white peace the specified scope.		1.0
start_peace_conference	tag = <scope> / <variable> The scope to peace with. score_factor = <decimal> / <variable> The fraction of the total score awarded to the winners compared to regular victory. message = <localisation key> The reason for peace showing up in the pop-up.	start_peace_conference = { tag = GER score_factor = 0.4 message = my_peace_tt }	Makes the current scope start a peace conference with the specified scope on the other side.	Current scope is the winner, target and its subjects are the losers. Can only be used if at war with the target.	1.12
set_truce	target = <scope> The scope to truce with. days = <int> The duration of the truce.	set_truce = { target = GER days = 90 }	Makes the current scope truce with the specified scope.		1.0

Name	Parameters	Examples	Description	Notes	Version Added
create_wargoal	target = <scope> / <variable> The country to target. type = <wargoal> The wargoal to generate. generator = { <state id> } The states to supply the wargoal (i.e. take_state_focus).	create_wargoal = { type = puppet_wargoal_focus target = ROOT }	Grants the current scope a wargoal against the specified country.		1.0
remove_wargoal	target = <scope> / <variable> The country to target. type = <wargoal> The wargoal to remove. "all" will remove all wargoals.	remove_wargoal = { type = all target = ROOT }	Removes wargoals from the current scope to the specified country.		1.10.2
start_civil_war	ideology = <ideology> The ideology of the breakaway country. ruling_party = <ideology> The ruling party of the original, player-led country. Optional. size = <float> The size of the breakaway country and the fraction of the original stockpile and military units it will receive by default. Optional, defaults to 0.5. army_ratio = <float> The size of the land army that the breakaway country gets. Optional, defaults to being the same as size. navy_ratio = <float> The size of the naval forces that the breakaway country gets. Optional, defaults to being the same as size. air_ratio = <float> The size of the airforce that the breakaway country gets. Optional, defaults to being the same as size. capital = <state> The capital state of the breakaway country. Optional. states = { <state> } The states included in the breakway country. Optional, defaults to random states based off size. all will result in all states that meet the filter going to the breakaway. states_filter = { <triggers> } A trigger block checked for the state that must be met to be transferred to the breakaway. Optional. keep_unit_leaders = { <unit leader id> } List of unit leaders to be kept by their legacy_id. Optional. keep_unit_leaders_trigger = { <triggers> } Trigger block checked for every unit leader that forces them to be kept if they meet the triggers. The default scope is the unit leader, ROOT is the country receiving the unit leader, while FROM is the original owner of the unit leader. Optional. keep_political_leader = <bool> Controls if the promoted party leader (i.e. the one that'd take power if the country were to be switched to that ideology group) of the revolting ideology group will be kept by the country or join the revolt, yes resulting in the former. Optional, defaults to false. keep_political_party_members = <bool> Controls if non-promoted party leaders of the revolting ideology group will be kept by the country or join the revolt, yes resulting in the former. Optional, defaults to false. keep_all_characters = yes If true, the revolter will have no characters from the original country transferred to them. Optional, defaults to false. <effects> An effect block executed for the breakaway country.	start_civil_war = { ruling_party = communism # Original country's ideology changes to communism ideology = ROOT # Breakaway gets old ideology of ROOT size = 0.8 capital = 282 states = { 282 533 536 555 529 530 528 } keep_unit_leaders = { 750 751 752 } keep_political_leader = yes keep_political_party_members = yes } start_civil_war = { ideology = democratic size = 0.1 states = all states_filter = { is_on_continent = europe is_capital = no } set_country_flag = TAG_my_country_tag_alias_trigger # Sets a country flag that gets used in a country tag alias. } (See country tag aliases) start_civil_war = { ideology = neutrality size = 0.1 army_ratio = 0.5 navy_ratio = 0 air_ratio = 1 keep_unit_leaders_trigger = { has_trait = my_trait_name } keep_all_characters = yes PREV = { # Original country TAG_airforce_leader = { # Character set_nationality = PREV.PREV # Transfers to breakaway } } promote_character = TAG_airforce_leader } (See usage for PREV and PREV.PREV)	Starts a civil war for the current scope with the specified parameters.	states = all would include every single state controlled by the country. If the country's current capital state is set as one of the states that the revolt can gain, it won't fire. set_capital can be used to change the capital beforehand, with On_actions#on_civil_war_end being used to set it back to the default after the civil war ends. A civil war started via this effect cannot have more than two sides. For adding more sides, this can be simulated by setting an existing war (typically originating from a dynamic country created via create_dynamic_country) as a civil war via add_civil_war_target .	1.0
add_civil_war_target	<country> - The country to set as the target.	add_civil_war_target = TAG	Sets that the war between ROOT and TAG is a civil war, resulting in the victory being the annexation of the other side and setting world tension limits on intervention.	ROOT and TAG must already be at war with each other for the effect to take place.	1.9
transfer_units_fraction	target = <country> The country which should receive the units from the current scope. size = <float> The size of the breakaway country and the fraction of the original stockpile and military units it will receive by default. Optional, defaults to 0.5. army_ratio = <float> The size of the land army that the breakaway country gets. Optional, defaults to being the same as size. navy_ratio = <float> The size of the naval forces that the breakaway country gets. Optional, defaults to being the same as size. air_ratio = <float> The size of the airforce that the breakaway country gets. Optional,	transfer_units_fraction= { target = SPD size = 0.5 stockpile_ratio = 0.8 army_ratio = 0.8 navy_ratio = 0.5 air_ratio = 0.5 keep_unit_leaders_trigger = { has_trait = trait_SPA_nationalist_sympathies } }	Transfers a fraction of the military to a target, including units (either type: land, navy, or air), equipment, and unit leaders.		1.9

Name	Parameters	Examples	Description	Notes	Version Added
	defaults to being the same as size. keep_unit_leaders = { <unit leader id> } List of unit leaders to be kept by their legacy_id. Optional. keep_unit_leaders_trigger = { <triggers> } Trigger block checked for every unit leader that forces them to be kept if they meet the triggers. The default scope is the unit leader, ROOT is the country receiving the unit leader, while FROM is the original owner of the unit leader. Optional.				
add_nuclear_bombs	Adds nuclear bomb to TAG's stockpile.	add_nuclear_bombs = 100	Adds specified number of nukes to the country's stockpile	Needs the Nuke tech to use.	1.6
launch_nuke	province = <ID> The specific province to nuke. state = <ID> The state to nuke. controller = <TAG> Prioritises provinces controlled by this country. use_nuke = <boolean> Whether a nuke should be deducted from the country's stockpile. Defaults to false.	launch_nuke = { province = 1234 } launch_nuke = { state = 42 controller = GER use_nuke = yes }	Nukes the specified province or a province in the needed state. If a state is set rather than the specific province, first prioritises the country set in controller, then prioritises the countries at war with the current scope, and then countries that are neutral.	If set to use a nuke, then requires at least one nuclear bomb in the stockpile.	1.6

Resources[[编辑](#) | [编辑源代码](#)]

Resource-related country-scoped effects: <div>折叠</div>					
Name	Parameters	Examples	Description	Notes	Version Added
add_resource	type = <resource> The resource to add. amount = <int> The amount of resource to add. state = <id>Which state to add the resource to. Variables can be used.	add_resource = { type = oil amount = 50 state = 88 }	Adds the specified resource in the specified amount to the specified state.	Can also be used in state scope.	1.0
create_import	resource = <resource> The resource to import. amount = <int> The amount of resource to import. exporter = <id>Which country exports the resource.	create_import = { resource = steel amount = 100 exporter = GER }	Creates an import for the current scope with the specified resource and from the specified exporter.		1.0
give_resource_rights	receiver = <tag> The country that would get the resource rights. state = <state> The state where the resource rights are located. resources = { <resource> <...> <resource> } The resources to which give resource rights to. Optional, defaults to all.	give_resource_rights = { receiver = ENG state = 291 } give_resource_rights = { receiver = POL state = 321 resources = { oil } }	Gives all the resources of a state to the target country	The resource rights will only be provided as long as the current country controls the state with resource rights.	1.6
remove_resource_rights	<state> The state to remove current country's resource rights from.	ENG = { remove_resource_rights = 477 }	Removes given resource rights		1.6
add_fuel	<int> The fuel amount	add_fuel = 400	Adds fuel to the current country.		1.6
set_fuel	<int> Fuel amount.	set_fuel = 400	Sets country's current fuel amount.		1.6
set_fuel_ratio	<decimal> The needed ratio of fuel.	set_fuel_ratio = 0.5	Set country's current fuel ratio relative to its capacity.		1.6

Buildings[[编辑](#) | [编辑源代码](#)]

Building-related country-scoped effects: <div>折叠</div>					
Name	Parameters	Examples	Description	Notes	Version Added
add_offsite_building	type = <building> The building to add. level = <level> / <variable> The maximum level to add.	add_offsite_building = { type = arms_factory level = 1 }	Adds an off-map (offmap) building for the current scope that produces its effects without being present in a state.		1.5
modify_building_resources	building = <building> The building to modify. resource = <resource> The resource to add. amount = <amount> The amount of resource to add.	modify_building_resources = { building = synthetic_refinery resource = oil amount = 1 }	Modifies the resource output of the specified building for the current scope.		1.5

National focuses[[编辑](#) | [编辑源代码](#)]

National focus-related country-scoped effects: <div>折叠</div>					
Name	Parameters	Examples	Description	Notes	Version Added
load_focus_tree	<string> The focus tree to load. Alternatively, as an	load_focus_tree = china_communist_focus load_focus_tree = { tree =	Loads a new focus tree for the current scope, retaining any shared focuses.		1.5

Name	Parameters	Examples	Description	Notes	Version Added
	effect block: tree = <string> keep_completed = yes/no	british_focus keep_completed = yes }			
unlock_national_focus	<focus> The focus to unlock.	unlock_national_focus = my_focus	Bypasses the specified focus for the current scope (marks as complete without firing complete_effect of the focus).		1.0
complete_national_focus	<focus> The focus to complete.	complete_national_focus = my_focus	Completes the specified focus for the current scope.		1.0
uncomplete_national_focus	focus = <focus> uncomplete_children = <bool> Defaults "no". Optional. refund_political_power = <bool> Defaults "no". Optional.	uncomplete_national_focus = { focus = GER_oppose_hitler uncomplete_children = yes refund_political_power = no }	Removes a focus from list of completed focus, and potentially all focuses requiring it as a prerequisite. If the focus has one, the 'on_uncomplete' effect will be executed on each uncompleted focus.		1.11
mark_focus_tree_layout_dirty	<bool> Boolean.	mark_focus_tree_layout_dirty = yes	Refreshes the focus tree for the specified country, restarting the checks in allow_branch and position offsets for focuses.	If put within a focus' completion reward, the focus will not be marked as complete at the time the effect is executed, leading to has_completed_focus checks specifying that focus in particular to be marked as false. This can be bypassed by putting an effect within a hidden event fired immediately within the focus or by reloading the same focus tree with load_focus_tree set to keep completed focuses, marking the focus as complete, before using the effect.	1.9

Decisions[[编辑](#) | [编辑源代码](#)]

Decision-related country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
activate_decision	<decision> The decision to activate.	activate_decision = my_decision	Activates the specified decision for the current scope, ignoring triggers for the decision.	Decisions are found in /Hearts of Iron IV/common/decisions/*.txt	1.0
activate_targeted_decision	target = <scope> The country to target. decision = <decision> The decision to activate.	activate_targeted_decision = { target = GER decision = my_decision }	Activates the specified targeted decision for the specified target for the current scope.	Decisions are found in /Hearts of Iron IV/common/decisions/*.txt	1.5
remove_targeted_decision	<decision> The decision to remove.	remove_targeted_decision = { target = FROM decision = my_decision }	Removes the specified targeted decision for the current scope.	Decisions are found in /Hearts of Iron IV/common/decisions/*.txt	1.5
unlock_decision_tooltip	<decision> The decision to display.	unlock_decision_tooltip = my_decision	Displays a special tooltip for the specified decision in the effect tooltip.	Decisions are found in /Hearts of Iron IV/common/decisions/*.txt	1.5
unlock_decision_category_tooltip	<category> The decision category to display.	unlock_decision_category_tooltip = my_category	Displays a special tooltip for the specified decision category in the effect tooltip.	Decision categories are found in /Hearts of Iron IV/common/decisions/categories/*.txt	1.5
add_days_remove	decision = <decision> The decision to remove days from. days = <int> / <variable> The number of days to remove from the mission.	add_days_remove = { decision = decision_here days = 30 }	Adds the number of days to a decision's days_remove.	Decisions are found in /Hearts of Iron IV/common/decisions/*.txt	1.9
remove_decision	Allows to remove specified decision without running remove_effect.	remove_decision = GER_MEPO	Removes a decision.		1.6
remove_decision_on_cooldown	<decision> The decision that is to be removed.	remove_decision_on_cooldown = TAG_my_decision	If the decision is on cooldown, it gets removed, in order to reactivate or remove completely.		1.11

Missions[[编辑](#) | [编辑源代码](#)]

Mission-related country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
activate_mission	<mission> The mission to activate.	activate_mission = my_mission	Activates the specified mission for the current scope, ignoring any triggers for the decision.	Missions are found in /Hearts of Iron IV/common/decisions/*.txt	1.5
activate_mission_tooltip	<mission> The mission to display.	activate_mission_tooltip = my_mission	Displays a special tooltip for the specified mission in the effect tooltip.	Missions are found in /Hearts of Iron IV/common/decisions/*.txt	1.5
remove_mission	<mission> The mission to remove.	remove_mission = my_mission	Removes the specified mission for the current scope.	Missions are found in /Hearts of Iron IV/common/decisions/*.txt	1.5
add_days_mission_timeout	mission = <mission> The mission to add days to. days = <int> / <variable> The number of days to add to the mission.	add_days_mission_timeout = { mission = my_mission days = 20 }	Adds the number of days to the specified mission.	Missions are found in /Hearts of Iron IV/common/decisions/*.txt	1.9

Technologies and doctrines[[编辑](#) | [编辑源代码](#)]

Technology-related country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
add_research_slot	<int> The number of slots to add or remove.	add_research_slot = 1	Adjusts the number of research slots the		1.0

Name	Parameters	Examples	Description	Notes	Version Added
			current scope has. Can remove slots with negatives.		
set_research_slots	<int> The number of slots to set.	set_research_slots = 4	Sets the number of research slots the current scope has.		1.0
add_tech_bonus	bonus = <float> The bonus to technology given, default 0. uses = <int> The amount of times the bonus can be used, default 1. ahead_reduction = <float> The cost reduction if ahead of time, default 0. category = <string> Which technology category the bonus applies to. Multiple can be defined. technology = <string> Which technology the bonus applies to. Multiple can be defined. name = <string> Tooltip shown in research tabs, optional.	add_tech_bonus = { bonus = 0.5 uses = 1 category = radar_tech }	Grants a research bonus to the current scope with the specified parameters.	Research bonus categories are defined in /Hearts of Iron IV/common/technology_tags/*.txt files, while technologies are defined in /Hearts of Iron IV/common/technologies/*.txt files.	1.0
set_technology	<technology> = <int> The technology to add. popup = no To not show the popup after adding technology	set_technology = { suicide_craft = 1 }	Grants the specified technology to the current scope.	A value of 1 sets the technology. A value of 0 removes the technology, but if it is a researchable technology, the duration it takes to research isn't reset, meaning it can be researched in 1 day. Technologies that are mutually exclusive with other technologies can not be removed by this effect. Technologies are defined in /Hearts of Iron IV/common/technologies/*.txt files.	1.0
add_to_tech_sharing_group	<string> The group to add the current scope to.	add_to_tech_sharing_group = us_research	Adds the current scope to the specified technology sharing group.	Technology sharing groups are found in Hearts of Iron IV\common\technology_sharing*.txt	1.3
remove_from_tech_sharing_group	<string> The group to remove the current scope from.	remove_from_tech_sharing_group = us_research	Removes the current scope from the specified technology sharing group.	Technology sharing groups are found in Hearts of Iron IV\common\technology_sharing*.txt	1.3
modify_tech_sharing_bonus	id = <string> The group to modify. bonus = <float> The new bonus.	modify_tech_sharing_bonus = { id = us_research bonus = 0.5 }	Modifies the specified technology sharing group.	Technology sharing groups are found in Hearts of Iron IV\common\technology_sharing*.txt	1.3
inherit_technology	<tag> The country to inherit technology from.	inherit_technology = CAN	Makes the current country's researched technologies be copied from the specified country.	Useful when making a country independent.	1.6
add_doctrine_cost_reduction	name = <name> Tooltip showing why the doctrine has reduced cost in the doctrine menu. Optional. cost_reduction = <fraction> Percentage of cost reduced. uses = <integer> Number of times the cost reduction can be used. category = <doctrine category> Which doctrine category the cost reduction will apply to. (Ex: cat_mobile_warfare,land_doctrine ,air_doctrine.)	add_doctrine_cost_reduction = { cost_reduction = 0.5 uses = 2 category = land_doctrine }	Gives bonuses of reducing land doctrine cost to current scope.	Doctrines are defined in /Hearts of Iron IV/common/technologies/*.txt files. For a general doctrine cost reduction, see "<land/air/naval>_doctrine_cost_factor" in Modifiers .	1.11

Ideas[[编辑](#) | [编辑源代码](#)]

This includes national spirits, laws, designers, and advisors (using the idea_token)

Idea-related country-scoped effects:

Name	Parameters	Examples	Description	Notes	Version Added
add_ideas	<idea> The idea to add.	add_ideas = my_idea add_ideas = { my_idea_1 my_idea_2 }	Adds the specified ideas to the current scope.	Can be used as a scope to add multiple at once.	1.0
add_timed_idea	idea = <idea> The idea to add. days = <int> / <variable> The number of days to add the idea for.	add_timed_idea = { idea = my_idea days = 180 }	Adds the specified ideas to the current scope for the specified number of days.		1.0
modify_timed_idea	idea = <idea> The idea to modify. days = <int> / <variable> The number of days to add to the idea.	modify_timed_idea = { idea = my_idea days = 60 }	Extends the duration of the timed idea by the specified amount.		1.0
swap_ideas	add_idea = <idea> The idea to add. remove_idea = <idea>	swap_ideas = { remove_idea = my_idea_1 add_idea = my_idea_2 }	Switches two ideas with a tooltip displaying any modifier differences between them.	If the ideas have the same name in the localisation, it will show up as modifying the idea rather than swapping them.	1.3

Name	Parameters	Examples	Description	Notes	Version Added
	The idea to remove.	}		The add will occur before the removal of the old idea.	
remove_ideas	<idea> The idea to remove.	remove_ideas = my_idea remove_ideas = { my_idea_1 my_idea_2 }	Removes the specified idea from the current scope.	Can be used as a scope to remove multiple at once.	1.0
remove_ideas_with_trait	<trait> The trait to target.	remove_ideas_with_trait = motorized_equipment_manufacturer	Removes all ideas for the current scope that use the specified trait.		1.0
show_ideas_tooltip	<idea> The idea to display.	show_ideas_tooltip = my_idea	Displays the specified idea in the tooltip for the current effect scope. Does not add the idea.		1.0

Units[[编辑](#) | [编辑源代码](#)]

Unit-related country-scoped effects:

折叠

Name	Parameters	Examples	Description	Notes	Version Added
load_oob	<oob> The filename of the order of battle to load, without the .txt extension.	load_oob = "GER_default"	Loads the specified order of battle for the current scope, applying the effects within. The filename with the .txt extension omitted is used as the effect's target.	Orders of battle are stored within /Hearts of Iron IV/history/units/*.txt. Primarily used to spawn divisions at specified locations.	1.0
division_template	name The name of the division. regiments = { <unit> = { x = 0 y = 0 } } support = { <unit> = { x = 0 y = 0 } } The composition of the division. Sub-units are defined in /Hearts of Iron IV/common/units/*.txt files. division_names_group = <group> The division names group that the template will use, deciding on the automatically-generated names of any new divisions built using that template. Optional, assigns one automatically if omitted. These are defined within /Hearts of Iron IV/common/units/names_divisions/*.txt files. is_locked = <bool> Whether the division is locked to modification and deletion. Optional. force_allow_recruiting = <bool> Whether the locked template can have units deployed using it without allowing editing. Optional, only has an effect in locked templates. division_cap = <int> The maximum amount of divisions that this template may have; requires the template to be locked. Optional. priority = <int> The priority the template receives in receiving supplies. Goes from 0 to 2. Optional, 1 by default. template_counter = <int> The icon used by the division as an integer. Optional, defaults to the icon of the most common sub-unit within. The icons are defined as sprites within any /Hearts of Iron IV/interface/*.gfx file (By default subuniticons.gfx) with the pattern of GFX_div_tmpl_<int>_large and GFX_div_tmpl_<int>_small. override_model = <entity> Enforces the entity used by the units using this template to be the specified one. Optional.	division_template = { name = "Test" is_locked = yes division_cap = 3 division_names_group = USA_INF_01 priority = 0 template_counter = 0 regiments = { infantry = { x = 0 y = 0 } infantry = { x = 0 y = 1 } infantry = { x = 0 y = 2 } infantry = { x = 0 y = 3 } } support = { military_police = { x = 0 y = 0 } } }	Creates and adds the specified division template to the current scope.	The x and y attributes represent the rows and columns in the division designer and start from 0. No tooltip is shown.	1.0
add_units_to_division_template	template_name = <string> The template to change. Optional if used in division scope. regiments = { <unit> = <column> } support = { <unit> = <column> } The units to add to the template. Sub-units are defined in /Hearts of Iron IV/common/units/*.txt files.	add_units_to_division_template = { template_name = "Test" regiments = { infantry = 2 infantry = 2 } support = { military_police = 0 } }	Adds the specified brigades to first available slots of specified columns to the template (if possible).	Columns go left-to-right starting with 0. Can also be used in division scope.	1.0
set_division_template_lock	division_template = <string> The name of the division template. is_locked = <bool> Whether the division is locked or not.	set_division_template_lock = { division_template = "Infantry Division" is_locked = yes }	Toggles the locked status on a division template for the current scope, which prevents editing or deletion.		1.5
country_lock_all_division_template	<bool> Boolean.	country_lock_all_division_template = yes	Locks all division templates for the current scope.	Used to prevent training, disbanding, and editing units.	1.9

Name	Parameters	Examples	Description	Notes	Version Added
set_division_force_allow_recruiting	<code>division_template = <string></code> Template to modify. <code>force_allow_recruiting = <bool></code> Whether to allow or disallow recruiting. Defaults to true if unset.	<code>set_division_force_allow_recruiting = { division_template = "My locked template" }</code>	Changes whether it's possible to recruit divisions of a locked template without unlocking the template.		1.12
set_division_template_cap	<code>division_template = <string></code> The name of the division template. <code>division_cap = <int></code> The division cap.	<code>set_division_template_cap = { division_template = "Swiss Citizen Militia" division_cap = SWI_militia_division_cap }</code>	Sets the cap of a division template. The template has to be locked first.		1.12
clear_division_template_cap	<code>division_template = <string></code> The name of the division template.	<code>clear_division_template_cap = { division_template = "Swiss Citizen Militia" }</code>	Clears the cap on the template, allowing it to have an unlimited amount of divisions.		1.12
delete_unit_template_and_units	<code>division_template = <string></code> The name of the division template.	<code>delete_unit_template_and_units = { division_template = "Infantry Division" disband = yes #will refund equipment and manpower }</code>	Deletes the specified division template and all units using it for the current scope.		1.5
delete_unit	<code>state = <number id></code> The id number of the state the unit must be in. <code>template = <string></code> The template the units must use to be deleted. <code>id = <int></code> The id given to the unit if created via the <code>create_unit</code> effect. <code>disband = <bool></code> If true, will refund equipment and manpower.	<code>delete_unit = { state = 787 disband = yes #will refund equipment and manpower } delete_unit = { template = "Infantry Division" }</code>	Deletes all units that meet the filters.	No tooltip is generated. <code>delete_units</code> can be used if deleting all units of a specific template.	1.5
delete_units	<code>division_template = <string></code> The template the units must use to be deleted. <code>disband = <bool></code> If true, will refund equipment and manpower.	<code>delete_units = { division_template = "Infantry Division" disband = yes }</code>	Deletes all units with a certain template.	Generates a tooltip, unlike <code>delete_unit</code> . Mandatory to specify a <code>division_template</code> .	1.9
create_railway_gun	<code>equipment = <type></code> Equipment type used by the railway gun. <code>name = <string></code> The name used by the railway gun. Optional. <code>location = <province></code> Location where the railway gun is created. Assumes the capital by default.	<code>create_railway_gun = { equipment = railway_gun_equipment_1 name = TAG_new_railway_gun location = 12406 }</code>	Creates a railway gun.		1.11
teleport_railway_guns_to_deploy_province	<code><boolean></code>	<code>teleport_railway_guns_to_deploy_province = yes</code>	Teleports all railway guns to the province where they get deployed.		1.11

Equipment[[编辑](#) | [编辑源代码](#)]

Equipment-related country-scoped effects: 折叠

Name	Parameters	Examples	Description	Notes	Version Added
set_equipment_fraction	<code><float></code> / <code><variable></code> The fraction of equipment to remove.	<code>set_equipment_fraction = 0.5</code>	Reduces the overall equipment stockpile by the specified fraction.	This should not be used in civil wars to simulate stockpile splitting. <code>start_civil_war</code> automatically divides stockpiles according to the respective size.	1.0
add_equipment_to_stockpile	<code>type = <equipment></code> The equipment to add. Either types and archetypes are accepted. <code>amount = <int></code> / <code><variable></code> The amount to add. <code>producer = <scope></code> / <code><variable></code> Defines who produced the equipment. Optional, defaults to the current scope. <code>variant_name = <string></code> The equipment variant to add. Mandatory if a variant needs to be created to produce the equipment, optional otherwise.	<code>add_equipment_to_stockpile = { type = infantry_equipment amount = -100 producer = GER } add_equipment_to_stockpile = { type = medium_tank_chassis_1 amount = 100 variant_name = "Panzer III" }</code>	Edits the equipment stockpile of the current scope, adds or removes equipment of a specified type or archetype.	With negative numbers, optionally specifying a producer will ensure only equipment with that producer gets removed. The equipment must be unlocked by the producer for the effect to succeed.	1.0
send_equipment	<code>type = <equipment></code> The equipment to add. Can be archetype. <code>amount = <int></code> / <code><variable></code> The amount to add. <code>target = <scope></code> / <code><variable></code> Which country receives the equipment.	<code>send_equipment = { equipment = infantry_equipment amount = 100 target = GER }</code>	Sends the specified amount of equipment to the specified target, removing said equipment from the current scope.	Cannot remove equipment into negatives, in which case equipment will not be received by the target in entirety.	1.0
send_equipment_fraction	<code>value = <0-1></code> How much equipment to send. <code>target = <scope></code> / <code><variable></code> Which country receives the equipment.	<code>send_equipment_fraction = { value = 0.3 target = GER }</code>	Sends the specified fraction of equipment to the specified target, removing said equipment from the current scope.		1.9

Name	Parameters	Examples	Description	Notes	Version Added
create_production_license	<code>target = <scope></code> Which country receives the license. <code>cost_factor = <float></code> Modifies the production cost. Equipment scope <code>type = <equipment></code> The equipment the country is licensed to produce. Must be an non-archetype equipment. <code>version = <int></code> The version indicates which variant should be licensed. The default is 0, meaning the base variant. <code>new_prioritised = <boolean></code> Whether new equipment is prioritised or not. Yes by default.	<code>create_production_license = {</code> <code>target = HUN</code> <code>equipment = {</code> <code>type = fighter_equipment_1</code> <code>version = 0</code> <code>new_prioritised = no</code> } <code>cost_factor = 0</code> }	Grants the specified country a license to produce the specified equipment from the current scope.		1.4
create_equipment_variant	<code>name = <string></code> The name of the variant. <code>type = <equipment></code> The equipment type the variant is of. <code>parent_version = <int></code> Ordering for multiple variants of the same equipment. 0 is the oldest, 1 is the second-oldest, etc. Optional, 0 by default. <code>obsolete = <bool></code> Whether the equipment variant is flagged as obsolete within the GUI and for AI. Optional, no by default. <code>name_group = <name group></code> The name group used for equipment. Stored in /Hearts of Iron IV/common/units /names_ships. Optional, can only be defined for ships. <code>role_icon_index = <int>/auto</code> Index of the role icon that will be used, as an integer. If set to "auto", will pick automatically. If set to 0, will be unset. Optional, only can be defined for ships. <code>model = <model name></code> Model that will be used by the equipment on the world map. Optional. <code>icon = <sprite></code> The icon that will be used by equipment. Stored as a spriteType within /Hearts of Iron IV/interface /*.gfx. Optional. Upgrade scope <code><upgrade> = <amount></code> The upgrades configuration for the variant. Module scope <code><slot> = <module></code> The modules configuration for the variant.	<code>create_equipment_variant = {</code> <code>name = "Vetehinen</code> Class" <code>type = ship_hull_submarine_1</code> <code>name_group = FIN_SS_HISTORICAL</code> <code>role_icon_index = 1</code> <code>modules = {</code> <code>fixed_ship_torpedo_slot = ship_torpedo_sub_1</code> <code>fixed_ship_engine_slot = sub_ship_engine_1</code> <code>rear_1_custom_slot = ship_mine_layer_sub</code> } } <code>create_equipment_variant = {</code> <code>name = "He 112"</code> <code>type = fighter_equipment_0</code> <code>obsolete = yes</code> <code>upgrades = {</code> <code>plane_gun_upgrade = 1</code> <code>plane_range_upgrade = 1</code> } } <code>create_equipment_variant = {</code> <code>name = "Light Tank Mk. IV"</code> <code>type = light_tank_chassis_1</code> <code>parent_version = 1</code> <code>modules = {</code> <code>main_armament_slot = tank_heavy_machine_gun</code> } <code>upgrades = {</code> <code>tank_nsb_engine_upgrade = 2</code> } <code>icon = "GFX_ENG_basic_light_tank_medium"</code> <code>model = ENG_MKIV_light_tank_entity</code> }	Creates the specified equipment variant for the current scope.	Role icons for ships are defined in /Hearts of Iron IV/gfx/army_icons/army_icons.txt. Upgrades are defined within /Hearts of Iron IV/common/units/equipment/upgrades/*.txt. Equipment types, including module slots for them, are defined within /Hearts of Iron IV/common/units/equipment/*.txt. Equipment modules are defined within /Hearts of Iron IV/common/units/equipment/modules/*.txt.	1.0
add_equipment_production	<code>amount = <int></code> The amount to produce. Optional. <code>requested_factories = <int></code> The number of factories to assigned initially. Optional. <code>progress = <float></code> The initial production progress. Optional. <code>efficiency = <float></code> The initial production efficiency. Optional. Equipment scope <code>type = <equipment></code> The name of the equipment to produce. <code>creator = <scope></code> The country which is producing the equipment. Used if root scope isn't producer. Optional. <code>version_name = <string></code> The name of the variant to produce. Optional.	<code>add_equipment_production = {</code> <code>equipment = {</code> <code>type = light_cruiser_2</code> } <code>requested_factories = 1</code> <code>progress = 0.95</code> <code>amount = 1</code> }	Starts a production line for the specified equipment for the current scope.		1.0

Military[\[编辑\]](#) [| 编辑源代码\]](#)

Military-related country-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
destroy_ships	<code>type = <ship></code> The type of ship to destroy. <code>count = <int></code> or all The amount to destroy.	<code>destroy_ships = {</code> <code>type = destroyer</code> <code>count = all</code> }	Destroys the specified type and amount of ships controlled by the current scope.		1.5
transfer_navy	<code>target = <scope></code> The target country.	<code>transfer_navy = {</code> <code>target = GER</code> }	Transfers the current scope navy to the specified country.		1.5
transfer_ship	<code>type = <ship></code> The type of ship to transfer. <code>target = <scope></code> The target country. <code>prefer_name = <string></code> Name of ship in origin navy that will preferably be transferred to target navy. Optional. <code>exclude_refitting = <yes/no></close></code> Determines whether ships that are being refitted will be transferred. Optional.	<code>transfer_ship = {</code> <code>prefer_name = "HMS Achilles"</code> <code>type = light_cruiser</code> <code>target = NZL</code> <code>exclude_refitting = no</code> }	Transfers the specified type of ship from the current scope to the specified country.		1.4
create_ship	<code>type = <ship></code> The type of ship to create. <code>equipment_variant = <string></code> The equipment variant to use.	<code>FRA = {</code> <code>create_ship = {</code> <code>type =</code>	Create a ship from another country and assign it to the reserve fleet. If not set, it will be the scoped country.		1.9

Name	Parameters	Examples	Description	Notes	Version Added
	<code>creator = <scope></code> The country that created this ship. Optional. <code>name = <string></code> Name of the ship. Optional.	<code>ship_hull_submarine_1</code> <code>equipment_variant = "S Class"</code> <code>creator = ENG</code> <code>name = "My ship name"</code> <code>}</code> <code>}</code>			
add_mines	Add mines to a strategic region for the current country.	<code>add_mines = { region = 42 amount = 100 }</code>	Add mines to a strategic region.		1.6
add_ace	<code>name = <string></code> The name of the ace. <code>surname = <string></code> The surname of the ace. <code>callsign = <string></code> The callsign of the ace. <code>type = <type></code> The ace type. <code>is_female = <bool></code> The gender of the ace.	<code>add_ace = {</code> <code> name = "Amelia"</code> <code> surname = "Earhart"</code> <code> callsign = "Revenant"</code> <code> type = fighter_genius</code> <code> is_female = yes</code> <code>}</code>	Adds an ace for the current scope.	Ace types found in /Hearts of Iron IV/common/aces/*.txt.	1.0

Intelligence[[编辑](#) | [编辑源代码](#)]

Intelligence-related country-scoped effects:

折叠

Name	Parameters	Examples	Description	Notes	Version Added
create_intelligence_agency	Allows to create automatically an intelligence agency	<code>create_intelligence_agency = {</code> <code> name = "A.G.E.N.C.Y"</code> <code> icon = GFX_intelligence_agency_logo_agency</code> <code>}</code>	Creates an Intelligence Agency.		1.9
upgrade_intelligence_agency	Allows to unlock automatically an intelligence agency upgrade	<code>upgrade_intelligence_agency = upgrade_form_department</code> <code>upgrade_intelligence_agency = <upgrade></code>	Unlocks an Intelligence Agency Upgrade.	Upgrades can be found in common/intelligence_agency_upgrades	1.9
add_decryption	<code>target = <tag></code> Towards which country to add decryption. <code>amount = <int></code> How much decryption to add in flat numbers. <code>ratio = <0-1></code> How much decryption ratio to add.	<code>add_decryption = {</code> <code> target = GER</code> <code> amount = 300</code> <code>}</code> <code>add_decryption = {</code> <code> target = GER</code> <code> ratio = 0.5</code> <code>}</code>	Adds decryption towards the target country	target and ratio arguments are mutually exclusive.	1.9
add_intel	<code>target = <tag></code> Towards which country to add intelligence. <code>civilian_intel = <int></code> How much civilian intel to add. <code>army_intel = <int></code> How much army intel to add. <code>navy_intel = <int></code> How much navy intel to add. <code>airforce_intel = <int></code> How much airforce intel to add.	<code>add_intel = {</code> <code> target = GER</code> <code> civilian_intel = 3</code> <code> army_intel = 2</code> <code> navy_intel = 1</code> <code> airforce_intel = 2</code> <code>}</code>	Adds the specified amount of intel towards the specified country.	If an intel argument is left out, 0 is assumed.	1.9
add_operation_token	<code>tag = <tag></code> Towards which country to add a token on. <code>token = <id></code> Which token to add.	<code>add_operation_token = {</code> <code> tag = GER</code> <code> token = token_test</code> <code>}</code>	Adds an operation token towards the country, allowing access to more intel or applying a targeted modifier.	Operation tokens are defined in /Hearts of Iron IV/common/operation_tokens/*.	1.9
remove_operation_token	<code>tag = <tag></code> Towards which country to remove a token from. <code>token = <id></code> Which token to remove.	<code>remove_operation_token = {</code> <code> tag = GER</code> <code> token = token_test</code> <code>}</code>	Removes an operation token from the country.	Operation tokens are defined in /Hearts of Iron IV/common/operation_tokens/*.	1.9
capture_operative	<code>operative = <tag></code> Which operative to capture. <code>ignore_death_chance = <bool></code> Whether to ignore the death chance on capture (no by default).	<code>capture_operative = {</code> <code> operative = PREV</code> <code> ignore_death_chance = yes</code> <code>}</code> <code>capture_operative = PREV</code>	Captures the specified operative.	Operatives can be referred to by using tags that refer to scopes	1.9
create_operative_leader	<code>bypass_recruitment = <bool></code> Whether the operative is directly added to the list of available operatives or needs to be recruited. <code>available_to_spy_master = <bool></code> Whether the operative can be recruited by the spy master. <code>bypass_recruitment</code> should be set to no. <code>portrait_tag_override = <bool></code> If selecting a random portrait, create one that is from the specified country rather than the current country. <code>name = <string></code> The name of the operative. <code>GFX = <string></code> The graphical reference of the picture of the leader, defined as a sprite within any /Hearts of Iron IV/interface/*.gfx file. <code>nationalities = { <tag> }</code> The nationalities of the operative. <code>traits = { <trait> }</code>	<code>create_operative_leader = {</code> <code> name = "Jacques Duclos"</code> <code> GFX = GFX_portrait_jacques_duclos</code> <code> traits = { operative_infiltrator operative_natural_orator }</code> <code> bypass_recruitment = no</code> <code> available_to_spy_master = yes</code> <code> nationalities = { FRA POL }</code> <code>}</code>	Creates an operative for the current scope with the specified attributes.	Traits are found in /Hearts of Iron IV/common/unit_leader/*.txt. All arguments aside from <code>bypass_recruitment</code> are optional. Must use a <code>sprite</code> type for the portrait , a direct link as in "gfx/leaders/TAG/filename.dds" will not work.	1.9

Name	Parameters	Examples	Description	Notes	Version Added
	The traits the leader spawns with.				
free_operative	<tag> The operative to be freed.	free_operative = PREV	Frees the specifes operative.	Operatives can be referred to by using tags that refer to scopes	1.9
free_random_operative	captured_by = <tag> The country that captured the operative. all = <bool> Whether to free all operatives or not (Defaults to no).	free_random_operative = { captured_by = POL all = yes }	Frees one random captured operative or all of them.		1.9
kill_operative	operative = <tag> The operative that is killed.	kill_operative = { operative = PREV } kill_operative = PREV	Kills the targeted operative.	Operatives can be referred to by using tags that refer to scopes	1.9
turn_operative	operative = <tag> The operative that is turned.	turn_operative = { operative = PREV } turn_operative = PREV	Turns the targeted operative against their own country, transferring them to the current country.	Operatives can be referred to by using tags that refer to scopes . This counts as the operative dying and will trigger the corresponding On action . Logs an error if used against your own operative.	1.9
steal_random_tech_bonus	category = <category name> The category to steal from. See /Hearts of Iron IV/common/technology_tags/* for list. folder = naval_folder The folder to steal from. See /Hearts of Iron IV/common/technology_tags/* for list. ahead_reduction = <float> The reduction to the ahead of time penalty. bonus = <float> The bonus to research speed. base_bonus = <float> The backup bonus if no tech is available. instant = <bool> Whether to instantly give a tech instead of a bonus or not. No by default. dynamic = <bool> Changes between instant and non-instant based on type. No by default. name = <localisation key> The name of the bonus. target = <tag> The country to steal from. uses = <int> How many times the bonus can be used.	steal_random_tech_bonus = { category = air_equipment folder = naval_folder ahead_reduction = 0.8 bonus = 1.2 base_bonus = 1.1 dynamic = yes name = LOC_KEY target = POL uses = 2 }	Steals a random tech bonus from the specified country.	If a country does not have a tech to be stolen, a random bonus will be applied by using base_bonus as a base.	1.9

Characters[[编辑](#) | [编辑源代码](#)]

Character-related country-scoped effects:

折叠

Name	Parameters	Examples	Description	Notes	Version Added
set_nationality	target_country = <scope> / <variable> The target country. character = <character> The charactor to transfer.	set_nationality = { target_country = TZN character = OMA_sultan }	Switches the specified character to the specified country.	If you wish to change the nationality of a specific character, and the country getting the effect doesn't have the character recruited already, use the every_possible_country = { limit = { has_character = ID } random_character = { limit = { is_character = ID } set_nationality = TAG } } command to call them up. Only necessary in 1.11 and beyond.	1.11
retire_character	<character>	retire_character = GER_Character_Token	Retires the character, removing every role they hold and making them disappear from the game.	Country scope only. The character cannot be re-recruited after retiring.	1.11
set_character_name	character = <character> The character to modify. name = <localisation key> The new name of the character.	set_character_name = { character = my_character name = my_name }	Sets the new name for the target character.	Can also be used in character scope.	1.11
character_list_tooltip	limit = { <triggers> } Triggers that must be fulfilled to show up in the list.	character_list_tooltip = { limit = { has_character_flag = SOV_targeted_for_purge_flag } }	Displays a list of every character meeting the specified limitation and recruited by the current country.		1.11
add_trait	character = <character> The character to modify. slot = <slot> Slot of the character. Necessary for advisors. ideology = <sub-ideology> Ideology type of the character. Necessary for country leaders. trait = <trait> The trait to add.	add_trait = { character = TAG_jane_smith slot = political_advisor trait = really_good_boss } add_trait = { character = TAG_my_leader ideology = liberalism trait = field_of_gar }	Adds the specified country leader trait to the character.	Can also be used in character scope. Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon. The character slot can be the character's name or id. Using name is recommended because 1.11 made id obsolete.	1.11
remove_trait	character = <character> The character to modify. slot = <slot> Slot of the character. Necessary for advisors. ideology = <sub-ideology>	remove_trait = { character = TAG_jane_smith slot = political_advisor trait = really_good_boss }	Removes the specified trait from the character.	Can also be used in character scope. Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon. The character slot can be the character's name or id. Using name is recommended because 1.11 made id obsolete.	1.11

Name	Parameters	Examples	Description	Notes	Version Added
	Ideology type of the character. Necessary for country leaders. trait = <trait> The trait to remove.	<pre>remove_trait = { character = TAG_my_leader ideology = liberalism trait = field_of_gar }</pre>			

Unit leaders[[编辑](#) | [编辑源代码](#)]

Unit leader-related country-scoped effects: 折叠					
Name	Parameters	Examples	Description	Notes	Version Added
create_corps_commander	<pre>name = <string> The name of the leader. picture = <string> The graphical reference of the picture of the leader. skill = <int> The skill of the leader. attack_skill = <int> The attack skill of the leader. defense_skill = <int> The defense skill of the leader. planning_skill = <int> The planning skill of the leader. logistics_skill = <int> The logistics skill of the leader. traits = { <trait> } The traits the leader spawns with. female = <bool> The gender of the leader. legacy_id = <int> The legacy ID used for the unit leader. Optional.</pre>	<pre>create_corps_commander = { name = "Jean de Lattre de Tassigny" picture = "Portrait_France_Jean_de_Lattre_de_Tassigny.dds" traits = { trickster brilliant_strategist } skill = 4 attack_skill = 4 defense_skill = 2 planning_skill = 4 logistics_skill = 3 }</pre>	Creates a commander for the current scope with the specified attributes.	Traits are found in /Hearts of Iron IV/common/unit_leader/*.txt. Deprecated , recommended to use add corps commander role instead when possible. The created corps commander will not be able to have a portrait if assigned to be a minister via officer corps, causing errors.	1.0
create_field_marshal	<pre>name = <string> The name of the leader. picture = <string> The graphical reference of the picture of the leader. skill = <int> The skill of the leader. attack_skill = <int> The attack skill of the leader. defense_skill = <int> The defense skill of the leader. planning_skill = <int> The planning skill of the leader. logistics_skill = <int> The logistics skill of the leader. traits = { <trait> } The traits the leader spawns with. female = <bool> The gender of the leader. legacy_id = <int> The legacy ID used for the unit leader. Optional.</pre>	<pre>create_field_marshal = { name = "Maurice Gamelin" picture = "Portrait_France_Maurice_Gamelin.dds" traits = { defensive_doctrine } skill = 2 attack_skill = 1 defense_skill = 3 planning_skill = 2 logistics_skill = 1 }</pre>	Creates a field marshal for the current scope with the specified attributes.	Traits are found in /Hearts of Iron IV/common/unit_leader/*.txt. Deprecated, recommended to use add field marshal role instead when possible. The created field marshal will not be able to have a portrait if assigned to be a minister via officer corps, causing errors.	1.0
create_navy_leader	<pre>name = <string> The name of the leader. picture = <string> The graphical reference of the picture of the leader. skill = <int> The skill of the leader. traits = { <trait> } The traits the leader spawns with. female = <bool> The gender of the leader. legacy_id = <int> The legacy ID</pre>	<pre>create_navy_leader = { name = "François Darlan" picture = "Portrait_France_Francois_Darlan.dds" traits = { superior_tactician } skill = 3 }</pre>	Creates a naval leader for the current scope with the specified attributes.	Traits are found in /Hearts of Iron IV/common/unit_leader/*.txt. Deprecated, recommended to use add naval commander role instead when possible.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
	used for the unit leader. Optional.				
remove_unit_leader	<id> The id of the unit leader.	remove_unit_leader = 70	Removes the specified unit leader by their legacy ID.	Does not work with the character ID. Instead, remove_unit_leader_role within the scope of the character is recommended when possible.	1.0
add_corps_commander_role	character = <character> The character to modify. <...> Army leader role definition	add_corps_commander_role = { Character = GER_Character_token skill = 4 attack_skill = 2 defense_skill = 3 planning_skill = 3 logistics_skill = 5 }	Sets the specified character to also act as a corps commander.	Can also be used in character scope.	1.11
add_field_marshal_role	character = <character> The character to modify. <...> Army leader role definition	add_field_marshal_role = { character = GER_Character_token skill = 4 attack_skill = 2 defense_skill = 3 planning_skill = 3 logistics_skill = 5 }	Sets the specified character to also act as a field marshal.	Can also be used in character scope.	1.11
add_naval_commander_role	character = <character> The character to modify. <...> Navy leader role definition	add_naval_commander_role = { Character = GER_Character_token skill = 4 attack_skill = 2 defense_skill = 3 planning_skill = 3 logistics_skill = 5 }	Sets the specified character to also act as an admiral.	Can also be used in character scope.	1.11
show_unit_leaders_tooltip	<character> The character whose name is to be shown.	show_unit_leaders_tooltip = TAG_my_leader	Shows the name of the specified character as a tooltip.		1.11

Country leaders[[编辑](#) | [编辑源代码](#)]

Country leader-related country-scoped effects:

Name	Parameters	Examples	Description	Notes	Version Added
create_country_leader	name = <string> The name of the leader. desc = <string> The description of the leader. picture = <spriteType> The graphical reference to the leader portrait. expire = <string> When the leader dies in history. ideology = <string> The sub-ideology of the country leader. Does not accept regular ideologies. female = <bool> The gender of the leader. Traits scope <trait> The trait to add. Can add multiple.	create_country_leader = { name = AFG_mohammed_zahir_shah desc = "POLITICS_MOHAMMED_ZAHIR_SHAH_DESC" picture = GFX_AFG_mohammed_zahir_shah expire = "1965.1.1" ideology = despotism traits = { } }		The portrait uses a spriteType, defined within /Hearts of Iron IV/interface/*.gfx. Sub-ideologies are defined in /Hearts of Iron IV/common/ideologies. Deprecated. Recommended to use add_country_leader_role instead when possible.	1.0
add_country_leader_role	character = <character> The character to modify. country_leader = { ... } Country leader role definition promote_leader = <bool> Will promote the leader to be the leader of the assigned party. Optional, defaults to false.	add_country_leader_role = { character = GER_character_token promote_leader = yes country_leader = { ideology = fascism_type expire = "1965.1.1" traits = { war_industrialist } } }	Sets the specified character to also act as a country leader, promoting to the party leader if specified.	Can also be used in character scope.	1.11
promote_character	<character> The character to promote. OR character = <character> The character to promote. ideology = <ideology type> The ideology type used by the country leader role.	promote_character = GER_erwin_rommel promote_character = { character = GER_erwin_rommel ideology = nazism }	Promotes a character to the leader of their political party.	Can also be used in character scope. If the character has multiple country leader roles, specifying the ideology type is mandatory. Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon.	1.11
remove_country_leader_role	character = <character> The character to modify. ideology = <string> The ideology type of	remove_country_leader_role = { character = GER_Character_Token ideology = socialism }	Removes a country leader role from a character.	Can also be used in character scope. Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon.	1.11

Name	Parameters	Examples	Description	Notes	Version Added
	the character.				
kill_ideology_leader	<ideology> Ideology.	kill_ideology_leader = communism	Kills the country leader of the designated ideology for the current scope.		1.9
retire_ideology_leader	<ideology> Ideology.	retire_ideology_leader = fascism	Retires and removes the country leader of the ideology party for the current scope.		1.9
kill_country_leader	<bool> Boolean.	kill_country_leader = yes	Kills the country leader for the current scope.		1.0
retire_country_leader	<bool> Boolean.	retire_country_leader = yes	Retires and removes the country leader as head of their party for the current scope.		1.0
set_country_leader_ideology	<government> The government to set.	set_country_leader_ideology = socialism	Changes the country leader's government type for the current scope.		1.0
set_country_leader_description	ideology = <ideology> The ideology of the country leader, optional. desc = <localisation key> The new description.	set_country_leader_description = { ideology = neutrality desc = LOC_KEY }	Changes the country leader's description.	Must use a localisation key from any /Hearts of Iron IV/localisation/*.yml file, putting the description in quotes will not work. Localisation for more info	1.9.1
set_country_leader_name	ideology = <ideology> The ideology of the country leader, optional. name = <localisation key> The new name.	set_country_leader_name = { ideology = neutrality name = LOC_KEY }	Changes the country leader's name.		1.9.1
set_country_leader_portrait	ideology = <ideology> The ideology of the country leader, optional. portrait = <sprite name> The new portrait.	set_country_leader_portrait = { ideology = neutrality portrait = GFX_IMAGE_NAME }	Changes the country leader's portrait.	The portrait must be defined in /Hearts of Iron IV/interface/*.gfx	1.9.1
add_country_leader_trait	<trait> The trait to add.	add_country_leader_trait = nationalist_symbol	Adds the specified trait to the current country's country leader.	Traits are found in /Hearts of Iron IV/common/country_leader/*.txt files.	1.0
remove_country_leader_trait	<trait> The trait to remove.	remove_country_leader_trait = nationalist_symbol	Removes the specified trait from the current scope's country leader.	Traits are found in /Hearts of Iron IV/common/country_leader/*.txt files.	1.0
swap_ruler_traits	Similar to swap_ideas. Removes one trait and adds another.	swap_ruler_traits = { remove = <trait> add = <trait> }	Swaps traits.	Use swap_country_leader_traits in character scope.	1.6

Advisors[\[编辑 | 编辑源代码\]](#)

Advisor-related country-scoped effects:

Name	Parameters	Examples	Description	Notes	Version Added
activate_advisor	<character> The character to activate.	activate_advisor = GER_character_token_air_chief	Hires an advisor, placing them into their respective slot.		1.11
deactivate_advisor	<character> The character to deactivate.	deactivate_advisor = GER_character_token_air_chief	Dismisses an advisor from their respective slot, leaving it empty.		1.11
add_advisor_role	character = <character> The character to modify. advisor = { ... } Advisor role definition activate = <bool> Will activate the advisor (add them directly when the command is run to the countries government). Optional, defaults to false.	add_advisor_role = { character = GER_Character_token activate = yes advisor = { slot = air_chief cost = 50 idea_token = GER_character_token_air_chief traits = { air_chief_ground_support_2 } } }	Sets the specified character to also act as an advisor, activating if specified.	Can also be used in character scope. Trigger and effect blocks (such as <code>allowed</code> and <code>on_add</code>) cannot be added within advisor definitions created this way.	1.11
remove_advisor_role	character = <character> Specifies the character if the effect is executed in country scope. slot = <int> The slot where to remove the advisor slot from.	remove_advisor_role = { character = "SOV_genrikh_yagoda" slot = political_advisor }	Removes the specified advisor role from the character.	Can also be used in character scope.	1.11

History[\[编辑 | 编辑源代码\]](#)

These effects can **only be used within history files**, failing when used outside. However, they're considered effects anyway rather than history arguments, as they can be used in if statements.

Effects to be used in country history files: 折叠					
Name	Parameters	Examples	Description	Notes	Version Added
recruit_character	<character>	recruit_character = GER_Character_token	Initially assigns the specified character to the current country.		1.11
generate_character	token_base = <string> Mandatory, acts as the character token. name = <localisation key> The name used for the character. Generates a random name if not set.	generate_character = { token_base = army_chief_defensive_1 name = funny_name advisor = { slot = air_chief cost = 50 idea_token = GER_character_token_air_chief } traits = { air_chief_ground_support_2 } allowed = { always = yes } }	Generates a character for current country.		1.11
set_oob	<order of battle> The name of the file used for the order of battle without the .txt extension.	set_oob = BHR_1936	Sets the order of battle to be used for the current country's divisions, overriding every other non-naval and non-air order of battle.	Orders of battle are defined in /Hearts of Iron IV/history /units/*.txt files.	1.0
set_naval_oob	<order of battle> The name of the file used for the order of battle without the .txt extension.	set_naval_oob = BHR_1936_naval_legacy	Sets the order of battle to be used for the current country's divisions, overriding every other naval order of battle.	Orders of battle are defined in /Hearts of Iron IV/history /units/*.txt files.	1.0
set_air_oob	<order of battle> The name of the file used for the order of battle without the .txt extension.	set_air_oob = ITA_1936_air_bba	Sets the order of battle to be used for the current country's divisions, overriding every other air order of battle.	Orders of battle are defined in /Hearts of Iron IV/history /units/*.txt files.	1.12
set_keyed_oob	key = <string> The key used for the file. name = <order of battle> The name of the file used for the order of battle without the .txt extension.	set_keyed_oob = { key = naval name = BHR_1936_mtg }	Sets the order of battle to be used for the current country's divisions, overriding every other keyed order of battle that uses the same key.	Orders of battle are defined in /Hearts of Iron IV/history /units/*.txt files.	1.0

State scope[[编辑](#) | [编辑源代码](#)]

The effects here must be used within a **state** scope.

General[[编辑](#) | [编辑源代码](#)]

General state-scoped effects: 折叠					
Name	Parameters	Examples	Description	Notes	Version Added
state_event	id = <event> The event to fire. days = <int> / <variable> Fires the event in the specified number of days. Optional. hours = <int> / <variable> Fires the event in the specified number of hours. Optional. random = <int> / <variable> Adds a random number (between 0 and random, inclusive) of hours to the scheduled fire time. Optional. random_days = <int> / <variable> Adds a random number (between 0 and random_days, inclusive) of days to the scheduled fire time. Optional.	state_event = { id = my_event.1 days = 10 random = 50 random_days = 10 trigger_for = controller }	Fires the specified event for the current state.	Where triggers do not need to be repeatedly checked random can be a performance light alternative to mean_time_to_happen for scheduling events. Using days = <event> / <variable> or hours may still be bugged and will not fire the event.	1.0
set_state_flag	<flag> An unique string to identify the state flag with. OR flag = <flag> The flag to set. days = <int> Sets the flag to last for the specified amount of days. Optional. value = <int> The new value of the flag on the scale from -2 147 483 648 to 2 147 483 647.	set_state_flag = my_flag set_state_flag = { flag = my_flag days = 123 value = 1 }	Defines a state flag.	No tooltip is shown. The flag in this effect is used in the meaning of 'boolean flag', used to store information.	1.0
clr_state_flag	<flag> The unique string of a state flag to clear.	clr_state_flag = my_flag	Clears a defined state flag.	No tooltip is shown.	1.0
modify_state_flag	flag = <flag> The state flag to modify. value = <value> The value to set it to: 0 or 1	modify_state_flag = { flag = my_flag value = 1 }	Modifies a state flag.	Use variables instead.	1.3
set_state_name	<string> Defines the new name.	set_state_name = "Funland"	Changes the current state's name to the specified name.		1.3
reset_state_name	<bool> Boolean.	reset_state_name = yes	Resets any changes to the current state's name.		1.3
add_claim_by	<scope> / <variable> The country to add the claim for.	add_claim_by = SOV	Adds a claim for the specified country on the current scope.		1.0
remove_claim_by	<scope> / <variable> The country to remove the claim for.	remove_claim_by = SOV	Removes a claim by the specified country on the current scope.		1.0
add_core_of	<scope> / <variable> The country to add the core for.	add_core_of = SOV	Adds a core for the specified country on the current scope.		1.0

Name	Parameters	Examples	Description	Notes	Version Added
remove_core_of	<scope> / <variable> The country to remove the core for.	remove_core_of = SOV	Removes a core for the specified country on the current scope.		1.0
set_demilitarized_zone	<bool> Boolean.	set_demilitarized_zone = yes	Makes the current scope a demilitarized zone.		1.0
set_state_category	<category> The category to change to.	set_state_category = rural	Changes the current state category to the specified category.	Categories are found in /Hearts of Iron IV/common /state_category/*.txt	1.3
add_state_modifier	Modifier scope <modifier> = <float> Adds a modifier to the state.	add_state_modifier = { modifier = { local_resources = 2.0 } }	Adds a modifier to the current state.		1.3
add_manpower	<int> / <variable> The amount to add.	add_manpower = 10000	Adds the specified amount of manpower to the current state.		1.0
add_resource	type = <resource> The resource to add. amount = <int> / <variable> The amount to add.	add_resource = { type = oil amount = 100 }	Adds the specified resource in the specified amount to the current state.	Can also be used in country scope.	1.0
set_border_war	<bool> Boolean.	set_border_war = yes	Enables Border War status for the current state.		1.0
create_unit	division = <division string> The division string. owner = <scope> The owner of the division. prioritize_location = <province> If possible, this province within the state gets used. Optional. allow_spawning_on_enemy_provs = yes Allows the units to be created on provinces owned by the enemy. Defaults to false. The following arguments go within division = "": name = \"<string>\" The name of the division. division_template = \"<string>\" The template to be used by the division. start_experience_factor = <double> Experience of the division, with 0 being none and 1 being full training. Defaults to 1. start_equipment_factor = <double> Equipment stockpile of the division. Defaults to 1. start_manpower_factor = <double> Manpower stockpile of the division. Defaults to 1.	create_unit = { division = "name = \"Infantry Division\" division_template = \"Infantry Division\" start_experience_factor = 0.5\" owner = GER }	Adds the specified division to the current state.	The division string must be on one line . A linebreak in the middle of <code>division = "..."</code> will break the effect and result in no units being spawned. Can only be used within a state scope , such as capital scope . The effect will do nothing when put into a country's scope.	1.3
teleport_armies	limit = { <triggers> } The condition that must be true for the owner of the armies for them to teleport. to_state_array = <array> The state array the armies will get teleported to. to_province = <ID> The province the armies will get teleported to. to_state = <ID> The state the armies will get teleported to.	teleport_armies = { limit = { has_war_together_with = ROOT } to_state_array = owned_controlled_states }	Teleports all armies in the specified state if the owner of the armies meets the condition.	Only define one of to_state_array, to_state, or to_province. If none is specified, it defaults to the capital.	1.9
add_province_modifier	static_modifiers = { <modifiers> } The list of modifiers. province = <id>The province to apply the modifiers to.provinces = {}Scope for selecting multiple provinces. The following arguments have to go inside it: id = <id>The ID of the province. Multiple can be specified. all_provinces = yesSelects all provinces to which the limitations apply. The following arguments require it: limit_to_coastal = yes Limits the selection of provinces to only coastal ones. limit_to_border = yes Limits the selection of provinces to only ones bordering a different country. limit_to_naval_base = yes Limits the selection of provinces to only ones that have a naval base. limit_to_victory_point = yes Limits the selection of provinces to only ones that have a victory point, or a city, in them.	add_province_modifier = { static_modifiers = { mod_modifier_1 mod_modifier_2 } province = 1234 } add_province_modifier = { static_modifiers = { mod_modifier_1 mod_modifier_2 } province = { id = 1234 id = 4321 } } add_province_modifier = { static_modifiers = { mod_modifier_1 mod_modifier_2 } province = { all_provinces = yes limit_to_coastal = yes limit_to_border = yes limit_to_naval_base = yes limit_to_victory_point = yes } } }	Adds a province modifier to the specified provinces in this state.	Province modifiers are defined in /Hearts of Iron IV/common /modifiers/*.txt	1.6
remove_province_modifier	static_modifiers = { <modifiers> } The list of modifiers. province = <id>The province to apply the modifiers to.provinces = {}Scope for selecting multiple provinces. The following arguments have to go inside it: id = <id>The ID of the province. Multiple can be specified. all_provinces = yesSelects all provinces to which the limitations apply. The following arguments require it: limit_to_coastal = yes Limits the selection of provinces to only coastal ones. limit_to_border = yes Limits the selection of	remove_province_modifier = { static_modifiers = { mod_modifier_1 mod_modifier_2 } province = 1234 } remove_province_modifier = { static_modifiers = { mod_modifier_1 mod_modifier_2 } province = { id = 1234 id = 4321 } }	Removes a province modifier to the specified provinces in this state.	Province modifiers are defined in /Hearts of Iron IV/common /modifiers/*.txt	1.6

Name	Parameters	Examples	Description	Notes	Version Added
	provinces to only ones bordering a different country. limit_to_naval_base = yes Limits the selection of provinces to only ones that have a naval base. limit_to_victory_point = yes Limits the selection of provinces to only ones that have a victory point, or a city, in them.	<pre>} remove_province_modifier = { static_modifiers = { mod_modifier_1 mod_modifier_2 } province = { all_provinces = yes limit_to_coastal = yes limit_to_border = yes limit_to_naval_base = yes limit_to_victory_point = yes } }</pre>			
add_victory_points	Add victory points to a province	<pre>add_victory_points = { province = 1234 value = 10 }</pre>	Adds victory points to a province.	Accepts negative values	1.10
set_victory_points	Set the victory points of a province	<pre>set_victory_points = { province = 1234 value = 10 }</pre>	Sets the number of victory point in a province.	Accepts negative values	1.10
set_state_province_controller	<pre>controller = <tag> The new controller of the province. limit = { <triggers> } The triggers that must be fulfilled by the province's current controller to be transferred to the new controller.</pre>	<pre>set_state_province_controller = { controller = POL limit = { OR = { tag = GER is_in_faction_with = GER } } }</pre>	Changes the controller of all provinces within that state controlled by countries that meet triggers to the specified country.		1.9
transfer_state_to	<pre><country> Country to transfer the state to.</pre>	<pre>transfer_state_to = JAM</pre>	Sets owner and controller of the state to the given country		1.11
set_state_owner_to	<pre><country> Country to set the owner (but not the controller) of the state to.</pre>	<pre>set_state_owner_to = JAM</pre>	Sets the owner of the state to the given country	Use transfer_state_to unless the control specifically shouldn't be given.	1.11
set_state_controller_to	<pre><country> Country to set the controller (but not the owner) of the state to.</pre>	<pre>set_state_controller_to = ITA</pre>	Sets the controller of the state to the given country		1.11

Buildings[[编辑](#) | [编辑源代码](#)]

Building-related state-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
add_extra_state_shared_building_slots	<pre><int> / <variable> The amount of slots to add or remove.</pre>	<pre>add_extra_state_shared_building_slots = 2</pre>	Changes the number of shared building slots for the current state.	Shared buildings slots being the ones used for multiple building types, such as military or civilian factories. This is in contrast to non-shared slots, such as those used by radio stations or air bases, which only can be changed globally with technologies.	1.0
add_building_construction	<pre>type = <string> The building to add. level = <int> / <variable> The level to set the building to. instant_build = <bool> Defines whether the buildings are instantly built. province = <id> Defines the exact province to add provincial buildings in. Can be used as a scope. Province scope all_provinces = <bool> Affect all provinces within the state that meet each limit. Used in the province scope. id = <id> Affect the specified province ID. Used in the province scope, will apply for each building if inserted multiple times. limit_to_coastal = <bool> Affect only coastal provinces within the selection. Used in the province scope. limit_to_naval_base = <bool> Affect only provinces that have naval bases built. Used in the province scope. limit_to_border = <bool> Affect only provinces that lie on a border between countries. Used in the province scope. limit_to_victory_point = <int><bool> Affect only provinces that meet the victory point amount prerequisite. If yes is used in place of a number, any amount of victory points works. Used in the province scope. level = <int> Affect only provinces with</pre>	<pre>add_building_construction = { type = arms_factory level = 5 instant_build = yes } add_building_construction = { type = bunker level = 10 instant_build = yes province = { all_provinces = yes limit_to_border = yes limit_to_victory_point > 1 } } add_building_construction = { type = bunker level = 1 instant_build = yes province = 2999 }</pre>	Starts construction in the current state for the specified building.	For provincial buildings, must be done within the scope of the state that contains the province even if done on a specific province. If the controller country doesn't have an order of battle assigned within the history file, the buildings will not show up within the production menu until a recalculation of buildings, such as by changing consumer goods or reloading a savefile. Can only be used within a state scope , such as random owned controlled state . The effect will do nothing when put into a country's scope.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
	buildings level below, at or above the specified level. Used in the province scope.				
set_building_level	<div>type = <string> The building to add. level = <int> / <variable> The level to set the building to. instant_build = <bool> Defines whether the buildings are instantly built. province = <id> Defines the exact province to add provincial buildings in. Can be used as a scope. Province scope all_provinces = <bool> Affect all provinces. Used in the province scope. limit_to_coastal = <bool> Affect coastal provinces. Used in the province scope. limit_to_naval_base = <bool> Affect naval base provinces. Used in the province scope. limit_to_border = <bool> Affect border provinces. Used in the province scope. level = <int> Affect only provinces with buildings level below, at or above the specified level. Used in the province scope. id = <id> Affect the specified province. Used in the province scope.</div>	<div>set_building_level = { type = infrastructure level = 10 instant_build = yes } set_building_level = { type = bunker level = 3 province = { all_provinces = yes limit_to_border = yes level < 3 } }</div>	Sets the specified building to the current state (or provinces within the state).	The province scope is used for provincial level buildings. You can limit the construction to victory points using :limit_to_victory_point > 5 (only build province buildings on province with VP over 5) limit_to_victory_point = yes (only build province buildings on province with VP) For provincial buildings, must be done within the scope of the state that contains the province even if done on a specific province.	1.4
damage_building	<div>type = <building> The building to damage. damage = <float> The amount of damage to inflict. province = <id> The province to target for provincial buildings.</div>	<div>damage_building = { type = infrastructure damage = 1 }</div>	Damages the specified building in the current state.	The health of buildings is determined by the value attribute in a building's definition. This is multiplied by their level to get their total health.	1.3
remove_building	<div>type = <building> The building to remove. level = <int> / <variable> The levels to remove.</div>	<div>remove_building = { type = arms_factory level = 5 }</div>	Removes the specified building in the current state. For shared buildings level determines the amount, whereas for the others it is the actual level.		1.0

Resistance and compliance[[编辑](#) | [编辑源代码](#)]

Resistance-related state-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
add_compliance	<int> / <variable> The amount to add.	add_compliance = 30	Adds compliance to the specified state.		1.9
add_resistance	<int> / <variable> The amount to add.	add_resistance = 30	Adds resistance to the specified state.		1.9
add_resistance_target	<int> / <variable> The amount to add.	add_resistance_target = 30	Increases resistance target in the specified state.		1.9
add_resistance_target	<div>id = <int> The ID of the target increase. amount = <int>/<variable> The amount to increase the resistance target by. occupied = <scope> Will only apply the increase if the the occupied country is the specified scope. occupier = <scope> Will only apply the increase if the the occupier is the specified scope. days = <int>/<variable> If set, the resistance target will only be increased for the specified amount of days. tooltip = <string> The tooltip to show in the resistance target tooltip.</div>	<div>add_resistance_target = { id = 123 amount = 30 occupied = ENG occupier = GER days = 365 tooltip = my_localisation_key }</div>	Increases resistance target in the specified state.		1.9
cancel_resistance	<bool> Boolean.	cancel_resistance = yes	Cancels resistance activity for the current state.		1.9
force_disable_resistance	<scope> The target country.	force_disable_resistance = GER	Disables resistance for the scoped state when the occupier is the specified country.		1.9
force_disable_resistance	<div>clear = <bool> If set to yes, will clear resistance. occupier = <scope> Resistance will be disabled if the occupier is the specified scope.</div>	<div>force_disable_resistance = { clear = yes occupier = GER occupied = ENG }</div>	Disables resistance for the scoped state when the occupier is the specified country.		1.9

Name	Parameters	Examples	Description	Notes	Version Added
	occupied = <scope> Resistance will be disabled if the occupied country is the specified scope.				
force_enable_resistance	<scope> The target country.	force_enable_resistance = GER	Enables resistance for the scoped state when the occupier is the specified country.	Does not start resistance by itself, only removes the checks forcefully disabling it. Use with start_resistance in order to immediately start resistance.	1.9
force_enable_resistance	clear = <bool> If set to yes, will clear resistance. occupier = <scope> Resistance will be enabled if the occupier is the specified scope. occupied = <scope> Resistance will be enabled if the occupied country is the specified scope.	force_enable_resistance = { clear = yes occupier = GER occupied = ENG }	Enables resistance for the scoped state when the occupier is the specified country.	Does not start resistance by itself, only removes the checks forcefully disabling it. Use with start_resistance in order to immediately start resistance.	1.9
remove_resistance_target	<int> / <variable> The id of the resistance target to remove. (Must be set with add_resistance_target)	remove_resistance_target = 30	Removes a set resistance target increase in the specified state.	Has no tooltip.	1.9
set_compliance	<int> / <variable> The amount to set the compliance to.	set_compliance = 30	Sets compliance in the specified state.		1.9
set_resistance	<int> / <variable> The amount to set the resistance to.	set_resistance = 30	Sets resistance in the specified state.		1.9
start_resistance	<bool>/<scope> Whether to start resistance or not. If set to a scope, starts resistance for the specified scope.	start_resistance = POL start_resistance = yes	Starts resistance in the specified state.	If used on a state that normally can't start resistance, use alongside with force_enable_resistance .	1.9
set_garrison_strength	<0-1> The new garrison strength.	set_garrison_strength = 0.5	Sets the strength of the garrison in the specified state.		1.9
set_occupation_law	<law ID> The new occupation law enacted by the previous scope OR default_law.	GER = { every_controlled_state = { set_occupation_law = military_governor_occupation } } # Changes GER's occupation law for every controlled state.	Sets the occupation law of the state.	PREV will be the country for whom the occupation law will be changed. If PREV is not a country, nothing changes. If PREV doesn't occupy the state, nothing happens until it does. If using <code>default_law</code> , resets to the law set by the country's occupation.	1.12

Character scope[[编辑](#) | [编辑源代码](#)]

The effects here must be used within a **character** scope.

General[[编辑](#) | [编辑源代码](#)]

General character-scoped effects:

[折叠](#)

Name	Parameters	Examples	Description	Notes	Version Added
set_character_flag	<flag> An unique string to identify the character flag with. OR flag = <flag> The flag to set. days = <int> Sets the flag to last for the specified amount of days. Optional. value = <int> The new value of the flag on the scale from -2 147 483 648 to 2 147 483 647.	set_character_flag = my_flag set_character_flag = { flag = my_flag days = 123 value = 1 }	Defines a character flag.	No tooltip is shown. The flag in this effect is used in the meaning of 'boolean flag'. used to store information.	1.11
set_character_name	<localisation key> The name to use.	set_character_name = GER_my_cool_flag	Changes the character's name to the specified localisation key's value.		1.11
modify_character_flag	flag = <flag> The country flag to modify. value = <value> The value to set it to: 0 or 1	modify_character_flag = { flag = GER_my_cool_flag value = 5 }	Modifies an existing character flag	Use variables instead.	1.11
clr_character_flag		clr_character_flag = <bool>	Clears a character flag		1.11
retire	<bool> Boolean>	retire = yes	Retires the current character (removing them).		1.5
set_nationality	<scope> / <variable> The target country.	set_nationality = GER	Switches the current character to the specified country, giving them the character.	If you wish to change the nationality of a specific character, and the country getting the effect doesn't have the character recruited already, use the every_possible_country = { limit = { has_character = ID } random_character = { limit = { is_character = ID } set_nationality = TAG } } command to call them up. Only necessary in 1.11 and beyond.	1.5

Name	Parameters	Examples	Description	Notes	Version Added
set_portraits	<code>character = <character></code> The character name. Optional if in character scope. Army scope: <code>small = <sprite></code> The sprite used as an advisor. <code>large = <sprite></code> The sprite used as a general. Character scope: <code>large = <sprite></code> The sprite used as a country leader.	<code>set_portraits = { character = my_character army = { small ="MySmallCharacterGFX" } civilian = { large ="MyLargeCharacterGFX" } }</code>	Changes the specified portraits of a character.	Sprites are defined within /Hearts of Iron IV/interface/*.gfx files.	1.11
add_trait	<code>slot = <slot></code> Slot of the character. Necessary for advisors. <code>ideology = <sub-ideology></code> Ideology type of the character. Necessary for country leaders. <code>trait = <trait></code> The trait to add.	<code>add_trait = { slot = political_advisor trait = really_good_boss } add_trait = { ideology = liberalism trait = field_of_gar }</code>	Adds the specified country leader trait to the character.	Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon. The character slot can be the character's name or id. Using name is recommended because 1.11 made id obsolete.	1.11
remove_trait	<code>slot = <slot></code> Slot of the character. Necessary for advisors. <code>ideology = <sub-ideology></code> Ideology type of the character. Necessary for country leaders. <code>trait = <trait></code> The trait to remove.	<code>remove_trait = { slot = political_advisor trait = really_good_boss } remove_trait = { ideology = liberalism trait = field_of_gar }</code>	Removes the specified trait from the character.	Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon. The character slot can be the character's name or id. Using name is recommended because 1.11 made id obsolete.	1.11
add_corps_commander_role	<code><...></code> Army leader role definition	<code>add_corps_commander_role = { skill = 4 attack_skill = 2 defense_skill = 3 planning_skill = 3 logistics_skill = 5 }</code>	Sets the specified character to also act as a corps commander.		1.11
add_field_marshall_role	<code><...></code> Army leader role definition	<code>add_field_marshall_role = { skill = 4 attack_skill = 2 defense_skill = 3 planning_skill = 3 logistics_skill = 5 }</code>	Sets the specified character to also act as a field marshal.		1.11
add_naval_commander_role	<code><...></code> Navy leader role definition	<code>add_naval_commander_role = { skill = 4 attack_skill = 2 defense_skill = 3 planning_skill = 3 logistics_skill = 5 }</code>	Sets the specified character to also act as an admiral.		1.11
add_country_leader_role	<code>character = <character></code> The character to modify. <code>country_leader = { ... }</code> Country leader role definition <code>promote_leader = <bool></code> Will promote the leader to be the leader of the assigned party. Optional, defaults to false.	<code>add_country_leader_role = { character = GER_character_token promote_leader = yes country_leader = { ideology = fascism_type expire = "1965.1.1.1" traits = { war_industrialist } } }</code>	Sets the specified character to also act as a country leader, promoting to the party leader if specified.		1.11
promote_character	<code><bool></code> Boolean. OR <code><ideology type></code> The ideology type used by the country leader role.	<code>promote_character = yes</code> <code>promote_character = liberalism</code>	Promotes a character to the leader of their political party.	If the character has multiple country leader roles, specifying the ideology type is mandatory. Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon.	1.11
remove_country_leader_role	<code>ideology = <string></code> The ideology type of the character.	<code>remove_country_leader_role = { ideology = socialism }</code>	Removes a country leader role from a character.	Ideology type refers to a sub-type of an ideology group assigned to characters, commonly referred to as sub-ideologies in community jargon.	1.11
add_advisor_role	<code>advisor = { ... }</code> Advisor role definition <code>activate = <bool></code> Will activate the advisor (add them directly when the command is run to the countries government). Optional, defaults to false.	<code>add_advisor_role = { activate = yes advisor = { slot = air_chief cost = 50 idea_token = GER_character_token_air_chief traits = { air_chief_ground_support_2 } } }</code>	Sets the specified character to also act as an advisor, activating if specified.	Trigger and effect blocks (such as <code>allowed</code> and <code>on_add</code>) cannot be added within advisor definitions created this way.	1.11
remove_advisor_role	<code>slot = <int></code> The slot where to remove the advisor slot from.	<code>remove_advisor_role = { slot = political_advisor }</code>	Removes the specified advisor role from the character.		1.11

Unit leaders[[编辑](#) | [编辑源代码](#)]

These can only be used with characters of the unit leader type.

General unit leader-scoped effects:
[折叠](#)

Name	Parameters	Examples	Description	Notes	Version Added
------	------------	----------	-------------	-------	---------------

Name	Parameters	Examples	Description	Notes	Version Added
unit_leader_event	<code>id = <event></code> The event to fire. <code>days = <int> / <variable></code> Fires the event in the specified number of days. Optional. <code>hours = <int> / <variable></code> Fires the event in the specified number of hours. Optional. <code>random = <int> / <variable></code> Adds a random number (between <i>0</i> and <i>random</i> , inclusive) of hours to the scheduled fire time. Optional. <code>random_days = <int> / <variable></code> Adds a random number (between <i>0</i> and <i>random_days</i> , inclusive) of days to the scheduled fire time. Optional.	<code>unit_leader_event = {</code> <code>id = my_event.1</code> <code>days = 10</code> <code>random = 50</code> <code>random_days = 10</code> }	Fires the specified event for the owner of the current unit leader.	Uses a special interface displaying the current unit leader portrait. Where triggers do not need to be repeatedly checked <code>random</code> can be a performance light alternative to <code>mean_time_to_happen</code> for scheduling events.	1.5
set_unit_leader_flag	<code><flag></code> An unique string to identify the unit leader flag with.	<code>set_unit_leader_flag = my_flag</code>	Defines a unit leader flag.	Deprecated. Use <code>set_character_flag</code> instead. No tooltip is shown.	1.5
clr_unit_leader_flag	<code><flag></code> The unique string of a unit leader flag to clear.	<code>clr_unit_leader_flag = my_flag</code>	Clears a defined unit leader flag.	Deprecated. Use <code>clr_character_flag</code> instead. No tooltip is shown.	1.5
modify_unit_leader_flag	<code>flag = <flag></code> The unit leader flag to modify. <code>value = <value></code> The value to set it to: 0 or 1	<code>modify_unit_leader_flag = {</code> <code>flag = my_flag</code> <code>value = 1</code> }	Modifies a unit leader flag.	Deprecated. Use <code>modify_character_flag</code> instead.	1.5
promote_leader	<code><bool></code> Boolean	<code>promote_leader = yes</code>	Promotes the current unit leader to Field Marshal (if Commander).		1.5
demote_leader	<code><bool></code> Boolean	<code>demote_leader = yes</code>	Demotes the current unit leader to Commander (if Field Marshal).		1.5
add_unit_leader_trait	<code><trait></code> The trait to add.	<code>add_unit_leader_trait = old_guard</code>	Adds the specified trait to the current unit leader.	Traits are found in <code>/Hearts of Iron IV/common/unit_leader/*.txt</code> files.	1.0
remove_unit_leader_trait	<code><trait></code> The trait to remove.	<code>remove_unit_leader_trait = old_guard</code>	Removes the specified trait from the current unit leader.	Traits are found in <code>/Hearts of Iron IV/common/unit_leader/*.txt</code> files.	1.0
add_random_trait	<code><trait></code> The trait to add.	<code>add_random_trait = {</code> <code>old_guard</code> <code>brilliant_strategist</code> <code>inflexible_strategist</code> }	Adds a random trait from the list to the character.	Traits are found in <code>/Hearts of Iron IV/common/unit_leader/*.txt</code> files.	1.5
add_timed_unit_leader_trait	<code><trait></code> The trait to add. <code>days = <int></code> The duration of the trait.	<code>add_timed_unit_leader_trait = {</code> <code>trait = wounded</code> <code>days = 90</code> }	Adds the specified trait to the current unit leader for the specified duration.	Traits are found in <code>/Hearts of Iron IV/common/unit_leader/*.txt</code> files.	1.5
replace_unit_leader_trait				Traits are found in <code>/Hearts of Iron IV/common/unit_leader/*.txt</code> files.	1.5
remove_exile_tag	Remove the exile tag on an army leader, making them no longer be considered exile leaders.	<code>remove_exile_tag = yes</code>	Removes a leaders exile tag.		1.6
gain_xp	<code><int></code>	<code>gain_xp = 5</code>	Adds experience to the current unit leader, promoting to the next skill level if applicable.	Cannot be used with negatives.	1.9
remove_unit_leader	<code><bool></code>	<code>remove_unit_leader = yes</code>	Removes the current unit leader.		1.0
remove_unit_leader_role	<code><bool></code> Boolean.	<code>remove_unit_leader_role = yes</code>	Removes every unit leader role from the character		1.11

Country leaders[[编辑](#) | [编辑源代码](#)]

These can only be used with characters of the country leader type.

Country leader-scoped effects:

折叠

Name	Parameters	Examples	Description	Notes	Version Added
add_country_leader_trait	<code><trait></code> The trait to add. OR: <code>ideology = <sub-ideology></code> The sub-ideology of the country leader role to which the trait is added. <code>trait = <trait></code> The trait to add.	<code>add_country_leader_trait = nationalist_symbol</code> <code>add_country_leader_trait = {</code> <code>ideology = marxism</code> <code>trait = anti_communist</code> }	Adds the specified trait to the current character.	Traits are found in <code>/Hearts of Iron IV/common/country_leader/*.txt</code> files. <i>The former only if the character has one country leader role.</i>	1.11
remove_country_leader_trait	<code><trait></code> The trait to remove. OR: <code>ideology = <sub-ideology></code> The sub-ideology of the country leader role to which the trait is added. <code>trait = <trait></code> The trait to remove.	<code>remove_country_leader_trait = nationalist_symbol</code> <code>remove_country_leader_trait = {</code> <code>ideology = marxism</code> <code>trait = anti_communist</code> }	Removes the specified trait from the current character.	Traits are found in <code>/Hearts of Iron IV/common/country_leader/*.txt</code> files. <i>The former only if the character has one country leader role.</i>	1.11
swap_country_leader_traits	<code>remove = <trait></code> Trait to remove <code>add = <trait></code> Trait to add <code>ideology = <sub-ideology></code> Sub-ideology of the leader where to swap traits.	<code>swap_country_leader_traits = {</code> <code>remove = nationalist_symbol</code> <code>add = anti_communist</code> <code>ideology = marxism</code> }	Swaps traits of the current character.	Use swap_ruler_traits in country scope.	1.11

Combat[[编辑](#) | [编辑源代码](#)]

Combat-related unit leader-scoped effects: <div>折叠</div>					
Name	Parameters	Examples	Description	Notes	Version Added
supply_units	<int> / <variable> The amount of hours of supply.	supply_units = 24	Adds the specified amount of hours of supply to troops led by the current unit leader.		1.5
add_max_trait	<int> The amount to add.	add_max_trait = 1	Adds the specified amount of assignable trait slots to the current unit leader.		1.5
add_skill_level	<int> The skill to add.	add_skill_level = 1	Adds skill to the current unit leader.		1.5
add_logistics	<int> How many skill levels to add.	add_logistics = 1	Adds logistics skill to the current unit leader.		1.5
add_planning	<int> How many skill levels to add.	add_planning = 1	Adds planning skill to the current unit leader.		1.5
add_defense	<int> How many skill levels to add.	add_defense = 1	Adds defense skill to the current unit leader.		1.5
add_attack	<int> How many skill levels to add.	add_attack = 1	Adds attack skill to the current unit leader.		1.5
add_coordination	<int> How many skill levels to add.	add_coordination = 1	Adds coordination skill to the current navy leader.		1.5
add_maneuver	<int> How many skill levels to add.	add_maneuver = 1	Adds maneuver skill to the current navy leader.		1.5
add_temporary_buff_to_units	combat_offense = <float> The bonus to grant. Optional. combat_breakthrough = <float> The bonus to grant. Optional. combat_defense = <float> The bonus to grant. Optional. combat_entrenchment = <float> The bonus to grant. Optional. org_damage_multiplier = <float> The bonus to grant. Optional. str_damage_multiplier = <float> The bonus to grant. Optional. war_support_reduction_on_damage = <float> The bonus to grant. Optional. cannot_retreat_while_attacking = <float> The bonus to grant. Optional. cannot_retreat_while_defending = <float> The bonus to grant. Optional. days = <int> The duration of the buff. Optional. tooltip = <string> The tooltip to display for the buff.	add_temporary_buff_to_units = { combat_offense = 0.25 combat_breakthrough = 0.25 org_damage_multiplier = -1.0 str_damage_multiplier = 0.25 war_support_reduction_on_damage = 0.2 cannot_retreat_while_attacking = 1.0 days = 7 tooltip = ABILITY_FORCE_ATTACK_TOOLTIP } days = 7 tooltip = ABILITY_FORCE_ATTACK_TOOLTIP }	Adds the specified combat buff to the current unit leader.		1.5

Operatives[[编辑](#) | [编辑源代码](#)]

Operative-scoped effects: <div>折叠</div>					
Name	Parameters	Examples	Description	Notes	Version Added
add_nationality	<tag> The country to set the nationality to.	add_nationality = GER	Adds the nationality to the current operative.		1.9
capture_operative	captured_by = <tag> By which country to get captured. ignore_death_chance = <bool> Whether to ignore the death chance on capture (no by default).	capture_operative = { captured_by = POL ignore_death_chance = yes }	Makes the current operative be captured by a specific country.		1.9
force_operative_leader_into_hiding	<bool>	force_operative_leader_into_hiding = yes	Forces the current operative into hiding.		1.9
free_operative	captured_by = <tag> The country that captured the operative.	free_operative = { captured_by = POL }	Frees the current operative.		1.9
harm_operative_leader	<int> How much to harm the operative.	harm_operative_leader = 12	Harms the current operative.	The value is subject to modifiers.	1.9
kill_operative	killed_by = <tag> The country that'll kill the operative.	kill_operative = { killed_by = POL }	Kills the current operative.		1.9
turn_operative	turned_by = <tag> The country to which the operative defects.	turn_operative = { turned_by = PREV }	Turns the current operative against their own country, transferring them to the specified country.	This counts as the operative dying and will trigger the corresponding On action . Logs an error if used against your own operative.	1.9
operative_leader_event	id = <event> The event to fire. days = <int> / <variable> Fires the event in the specified number of days. Optional. hours = <int> / <variable> Fires the event in the specified number of hours. Optional. random = <int> / <variable> Adds a random number (between 0 and random, inclusive) of hours to the scheduled fire time. Optional. random_days = <int> / <variable> Adds a random number (between 0 and random_days, inclusive) of days to the scheduled fire time. Optional. originator = <tag> The originator of the event. Optional, defaults to owner of operative. recipient = <tag> The recipient of the event. Optional, defaults to owner of operative.	operative_leader_event = { id = my_event.1 originator = POL recipient = GER days = 10 random = 50 random_days = 10 set_from = ENG set_root = SOV set_from_from = FRA } id = my_event.1 originator = POL recipient = GER days = 10 random = 50 random_days = 10 set_from = ENG set_root = SOV set_from_from = FRA }	Fires the specified event for the operative.	Uses a special interface displaying the current operative portrait. Where triggers do not need to be repeatedly checked random can be a performance light alternative to mean_time_to_happen for scheduling events.	1.9

Name	Parameters	Examples	Description	Notes	Version Added
	<div>set_from = <tag> Sets the scope of FROM in scripted localization. Optional. set_from_from = <tag> Sets the scope of FROM.FROM in scripted localization. Optional. set_root = <tag> Sets the scope of ROOT in scripted localization. Optional.</div>				

Division scope[[编辑](#) | [编辑源代码](#)]

The effects here must be used within a **division** scope.

Division-scoped effects:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
destroy_unit	<bool> Boolean.	destroy_unit = yes	Destroys the currently-scoped division.		1.12
add_history_entry	key = <localisation key> The name of the entry. subject = "<string>" Logged entry. Never shown to the player. allow = <bool> Whether a medal can be awarded to the division over the history entry.	add_history_entry = { key = my_history_entry subject = "Test entry" allow = no }	Creates an entry within the command history of a decision.		1.12
change_division_template	<string> The name of the division.	change_division_template = "New template"	Changes the template of the division to the specified one.		1.12
add_random_valid_trait_from_unit	<character> Character to grant the trait to.	add_random_valid_trait_from_unit = FROM	Adds a random valid unit trait to a unit leader.	Only possible to use if the division scope is the same as the ROOT scope.	1.12
add_unit_medal_to_latest_entry	unit_medals = <medal ID> The medal to add.	add_unit_medal_to_latest_entry = { unit_medals = my_medal }	Adds the specified medal to the latest entry within the unit's history.		1.12
add_divisional_commander_xp	<decimal> Experience to add.	add_divisional_commander_xp = 10	Adds the specified amount of experience to the divisional commander.		1.12
reseed_division_commander	<int> The seed to use.	reseed_division_commander = 760	Re-randomises the division commander using the given seed.	Does not have a tooltip.	1.12

Other scopes[[编辑](#) | [编辑源代码](#)]

The effects here must be used within a scope that's specified within the notes.

Otherwise-scoped effects:
折叠



Name	Parameters	Examples	Description	Notes	Version Added
execute_operation_coordinated_strike	amount = <int> How many times the operation will get executed within the days set in the operation.	execute_operation_coordinated_strike = { amount = 12 }	All prepared Port Strike and Strategic Bombing in the target region will execute multiple times without air defence being able to intercept them.	Can only be used within operations.	1.9

Flow control[[编辑](#) | [编辑源代码](#)]

These scopes are used within effect scopes to control the execution of effects.

If statements[[编辑](#) | [编辑源代码](#)]

An [if statement](#) allows an execution of effects to only be done if certain [triggers](#) are met. This is done with the `if = { ... }` block. This serves as an effect block with an additional `limit = { ... }` argument, which is a [trigger block](#) where each trigger must be met for the scope being checked.

For example, the following will add 10%  [Stability](#) to the country this is executed on if it has positive  [Political Power](#) and below 90% stability:

```
if = {  
  limit = {  
    has_political_power > 0  
    stability < 0.9  
  }  
  add_stability = 0.1  
}
```

If the limit is not met, then none of the effects inside will be executed. If it is, then each one will be. If the limit is omitted, it defaults to being always true.

Optionally, `else_if = { ... }` (with `limit = { ... }` serving in a similar fashion) and `else = { ... }` can be added. If the initial limit within `if = { ... }` is false, it moves on to the next `else_if = { ... }`, checking the limit there. If the limit there is false, then it moves on to the next one, until hitting an end or an `else = { ... }`.

Two variants exist: nested and unnested. In the first case, the `else_if` or `else` is put directly inside of the preceding `if` or `else_if`, while in the second case it's put *right after*. In case of overlap, unnested if statements are preferred. Here is an example using unnested if statements:

```
if = {  
  limit = {  
    stability < 0.3 # If stability is below 30%, add 30%.  
  }  
  add_stability = 0.3  
}  
else_if = {  
  limit = {  
    stability < 0.6 # Otherwise, if it's below 60% (i.e. 30-59%), add 20%
```

```
}
    add_stability = 0.2
}
else = {
    add_stability = 0.1 # If there's 60-100% stability, add 10%
}
```

Within the tooltip, only effects that would be executed are shown. The effects within an unfulfilled if statement (or an `else/else_if` that's not read due to the if statement being met) will be hidden from the player, and so will the trigger. In order to avoid player confusion, [custom effect tooltips can be used to tell the player what this effect block would do](#), such as being used within an `else`.

Random effects[[编辑](#) | [编辑源代码](#)]

If you want an effect to have a random chance to be done or have nothing happen otherwise, the `random = { ... }` block is the simplest way to accomplish that:

```
random = {
    chance = 80
    add_stability = 0.4
    add_war_support = 0.3
}
```

This in particular will have an 80% chance to add 40% stability and 30% war support and, accordingly, a 20% chance to do nothing. The chance here is on the scale from 0 to 100.

If you want the game to choose between effect blocks, `random_list` can be used instead. For example, if you wanted an effect to randomly given the player one out of four bonuses, you'd do the following:

```
random_list = {
    10 = {
        add_stability = 0.5
    }
    10 = {
        add_manpower = 10000
    }
    10 = {
        add_war_support = 0.5
    }
    10 = {
        army_experience = 100
    }
}
```

The number is not the chance, but the weight for each option, as they don't have to add up to 100 or any number. An option with the weight of 20 is twice as likely to be picked as the option with the chance of 10, for instance. In total, the probability for an option to be picked is equal to the weight of the option divided by the sum of all weights.

It is also possible to use modifiers (akin to [MTTH blocks](#)) to affect the weight of each possible random effect or to use [variables](#) as chances.

```
random_list = {
    30 = {
        modifier = {
            factor = 1.3
            has_country_flag = inward_perfect_flag
        }
        add_stability = 0.5
    }
    25 = {
        add_manpower = 10000
    }
    20 = {
        add_war_support = 0.5
    }
    my_variable = { # Taking "my_variable" as the variable's name, both "var:my_variable" and "my_variable" are valid options, left up to the developer's preference.
        army_experience = 100
    }
}
```

If the country flag `inward_perfect_flag` is set, it'll multiply the above chance of 30 by 1.3 to get 39. Meanwhile, `my_variable` will take the value of the according temp variable or the current scope's variable as the weight of the option.

Note that if you want to create a repeatable decision including a random list, by default the same decision will pick the same random result every time it is triggered in a game. You can reverse this behaviour by including the following line in the decision block:

```
fixed_random_seed = no
```

This is only for decisions. Elsewhere, random seed is unfixed by default, making this argument unnecessary to set to "no".

Tooltip manipulation[[编辑](#) | [编辑源代码](#)]

See also: [Localisation](#)

The "tooltip" in this case refers to the text shown to the player in-game that explains what the effect block changes within the game, such as "+50 🏰 [Political Power](#)".

There are 3 ways to edit the tooltip within an effect block:

`hidden_effect = { ... }` is used in order to hide the effects within, making their execution not get shown to the player.

`effect_tooltip = { ... }` is, instead, used in order to put the effects into the tooltip without actually executing them.

`custom_effect_tooltip = my_localisation_key` is used in order to put an arbitrary paragraph of text as an effect that will get executed.

For example, this sample [focus' completion reward](#) utilises all three:

```
completion_reward = {
    hidden_effect = {
```

```
        every_subject_country = { country_event = my_event.1 }
    }
    custom_effect_tooltip = send_event_to_subjects_tt
    effect_tooltip = {
        add_political_power = 100
    }
    custom_effect_tooltip = reject_war_tt
}
```

In this case, `send_event_to_subjects_tt` and `reject_war_tt` are localisation keys defined within any `/Hearts of Iron IV/localisation/english/*_1_english.yml` file encoded with UTF-8-BOM, assuming the English language.

```
send_event_to_subjects_tt: "Sends a demand to our every subject.\nIf they agree, we get the following for each subject."
reject_war_tt: "If they reject the demand, we gain a wargoal against them."
```

In-game, this will appear as such:

Effect:

Sends a demand to our every subject.

If they agree, we get the following for each subject:

Political Power: +**100**

If they reject the demand, we gain a wargoal against them.

Noticably, the effect that fires the country event gets hidden from the tooltip. After completing the focus, the only thing that happens is that every subject country receives an event with the ID of `my_event.1`, the country does not immediately gain 100 political power.

Meta effects[[编辑](#) | [编辑源代码](#)]

Meta effects allow you to use non-dynamic effects (the ones that do not accept modifiers and can only use static tokens or constant values) as if they were accepting variables.

```
add_equipment_to_stockpile = {
    type = infantry_equipment_2
    amount = eq_amount
}
```

In the effect shown above, amount of equipment added is dynamic and can be set using the variable `"eq_amount"`. However, this effect does not let you use a variable as equipment type. You can not store `"infantry_equipment_2"` in a variable and use it here.

However, meta effects will let you use variables and scripted localization within them to build effects as if they were texts and run them. Let's make previous effect accept equipment type and equipment level as variables stored in `"eq_type"` and `"eq_level"`.

```
set_variable = { eq_type = 1 } # Sets the equipment type to "1", which determines the equipment given using scripted localisation, included below
```

```
set_variable = { eq_amount = 10 } # Sets the amount of equipment given to 10
```

```
set_variable = { eq_level = 2 } # Sets the equipment level to 2, which is used directly in the meta effect, no scripted localisation required
```

```
meta_effect = { # The actual meta effect. This can go anywhere you need it: in a decision, in a scripted effect, in a scripted GUI click effect, etc...
    text = {
        add_equipment_to_stockpile = {
            type = [EQ_TYPE].[EQ_LEVEL]
            amount = eq_amount
        }
    }
    EQ_LEVEL = "[?eq_level|.0]" # Gets the "eq_level" variable and saves it as "EQ_LEVEL" for the meta effect to use
    EQ_TYPE = "[This.GetEquipmentName]" # Gets the equipment type from scripted localisation, included below, based on the "eq_type" variable, and saves it as "EQ_TYPE" for the meta effect to use
}
```

```
# The scripted localization for the "eq_type" variable, which goes in a scripted localisation file
```

```
defined_text = { # Since the "eq_type" variable in this example is equal to 1, the equipment given by the effect is "artillery_equipment"
```

```
    name = GetEquipmentName
    text = {
        trigger = {
            check_variable = { eq_type = 0 }
        }
        localization_key = "infantry_equipment"
    }
    text = {
        trigger = {
            check_variable = { eq_type = 1 }
        }
        localization_key = "artillery_equipment"
    }
}
```

As you can see, we have created a `meta_effect` that takes two arguments. These arguments will be used replacing the parameters `[EQ_TYPE]` and `[EQ_LEVEL]` inside the meta effect. `EQ_LEVEL` will be replaced by `[?eq_level|.0]` which is the integer value of `eq_level` (in this case 2.000 becomes 2). `EQ_TYPE` is a bit more complicated, it is being replaced by a scripted localization. This scripted localization will check `eq_type` variable and depending on its value it will return the key token for the equipment. If it is 0, it will return `"infantry_equipment"`. If it is 1, it will return `"artillery_equipment"`.

So the final result is `[EQ_TYPE]` is being replaced by `"artillery_equipment"` and `[EQ_LEVEL]` is being replaced by `"2"` and in the end our effect will be built as:

```
add_equipment_to_stockpile = {
    type = artillery_equipment_2
    amount = eq_amount
}
```

which will give you 10 `artillery_equipment_2`.

`debug = yes` can be added to meta effects. Which will print the final effect to `game.log` when the effect is executed and make debugging easier.

Scripted effects[编辑 | 编辑源代码]

Scripted effects serve a similar purpose to [functions](#) in that they can be defined in `/Hearts of Iron IV/common/scripted_effects/*.txt` and then used elsewhere as a shortened version. Alongside that, the game allows the creation of custom console commands, which are scripted effects.

A scripted effect is defined simply as

```
scripted_effect_name = {  
    <effects>  
}
```

This example can be used as an effect in regular code as `scripted_effect_name = yes`.

Scripted effects can be accessed in console by typing `e scripted_effect_name` to run them.

To create a custom console command, the scripted effect's name should begin with `d_`. The console command itself does not include `d_`, so `d_test_command` would be run in console as `test_command`

In custom console commands, the country running the command is `FROM`, while `ROOT` is the selected country, state, or character. Anything entered after the console command, separated by spaces like `test_command 123 321 GER` is added to the 'args' temp [array](#). An example of a scripted effect which will transfer every state entered as an argument to the country that runs the console command is

```
d_transfer_states = {  
    for_each_scope_loop = {  
        array = args  
        FROM = {  
            transfer_state = PREV  
        }  
    }  
}
```

used like `transfer_states 123 321`