Hearts of Iron 4 Wiki

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

Conditions (also known as **Triggers**) are used to specify when the execution of effects occurs. When a trigger returns true, then the execution will continue, otherwise it stops. Several triggers, including the default AND-trigger, allow nesting to create more complex expression.

The list of triggers may be outdated. A complete, but unsorted, list of triggers can be found in /Hearts of Iron IV/documentation/triggers_documentation.html or /Hearts of Iron IV/documentation/triggers_documentation.md. As of 1.12.3, this file was last updated in the version 1.11.4.

Operators[编辑 | 编辑源代码]

Every trigger uses one of the three operators: The equality sign = or the comparison signs > and <.

The equality sign can either mean strict equality (Unlike previous games in the series where it checked for being equal or greater than) or can serve as a way to introduce a block of additional triggers, as in `has_opinion = { target = GRE value < -10 }`. The comparison signs serves as strict comparison: `has_political_power > 100` will not be true if the country has exactly 100 political power, for instance.

Not all triggers support the equality sign or comparison signs. See the details for each trigger in the notes for it. If not stated otherwise, the trigger only supports the equality sign.

Context scopes[编辑 | 编辑源代码]

There are three general scopes all triggers operate in:

country

state

character

They give context to a trigger by stating what the trigger is checking against.

Each country acts as a sub-scope of the general country scope, i.e. `GER = { }` will check only against Germany, whereas `any_country` will check against all countries. Likewise for states and characters.

Triggers won't work in scopes they are not assigned to. Country triggers will not work for states or vice-versa.

Scopes[编辑 | 编辑源代码]

Main article: Scopes

These don't serve as triggers, but rather as scopes that change for whom the triggers are being checked. Each one also serves as an AND statement.

Trigger scopes[编辑 | 编辑源代码]

Trigger scopes:
折叠

Name	Usage	Target type	Example	Description	Version Added
all_country	Always usable	Country	`all_country = { … }`	Checks if all countries meet the triggers.	1.0
any_country	Always usable	Country	`any_country = { … }`	Checks if any country meets the triggers.	1.0
all_other_country	Within country scope only	Country	`all_other_country = { … }`	Checks if all countries other than the one where this scope is located meet the triggers.	1.0
any_other_country	Within country scope only	Country	`any_other_country = { … }`	Checks if any country other than the one where this scope is located meets the triggers.	1.0

all_country_with_original_tag	Always usable	Country	all_country_with_original_tag = {
			original_tag_to_check = TAG #required
			... #triggers to check
			}
				Checks if all countries originating from the specified country, including the dynamic countries created for civil wars and other purposes, meet the triggers. `original_tag_to_check = TAG` is used to specify the original tag.	1.9

any_country_with_original_tag	Always usable	Country	any_country_with_original_tag = {
			original_tag_to_check = TAG #required
			... #triggers to check
			}
				Checks if any country originating from the specified country, including the dynamic countries created for civil wars and other purposes, meets the triggers. `original_tag_to_check = TAG` is used to specify the original tag.	1.9

all_neighbor_country	Within country scope only	Country	`all_neighbor_country = { … }`	Checks if all countries that border the one where this scope is located meet the triggers.	1.0
any_neighbor_country	Within country scope only	Country	`any_neighbor_country = { … }`	Checks if any country that borders the one where this scope is located meets the triggers.	1.0
any_home_area_neighbor_country	Within country scope only	Country	`any_home_area_neighbor_country = { … }`	Checks if any country that borders the one where this scope is located, as well as being in its home area - meaning a direct land connection between the capitals of countries - meets the triggers.	1.0
all_guaranteed_country	Within country scope only	Country	`all_guaranteed_country = { … }`	Checks if all countries that are guaranteed by the one where this scope is located meet the triggers.	1.9
any_guaranteed_country	Within country scope only	Country	`any_guaranteed_country = { … }`	Checks if any country that is guaranteed by the one where this scope is located meets the triggers.	1.9

Name	Usage	Target type	Example	Description	Version Added
all_allied_country	Within country scope only	Country	`all_allied_country = { … }`	Checks if all countries that are allied with the one where this scope is located - meaning that they are either a subject of the country, its overlord, or that they share a faction - meet the triggers. Does not include the country itself.	1.9
any_allied_country	Within country scope only	Country	`any_allied_country = { … }`	Checks if any country that is allied with the one where this scope is located - meaning that they are either a subject of the country, its overlord, or that they share a faction - meets the triggers. Does not include the country itself.	1.9
all_occupied_country	Within country scope only	Country	`all_occupied_country = { … }`	Checks if all countries that are occupied by the one where this scope is located - meaning that the occupied country has core states controlled by the occupier country - meet the triggers.	1.9
any_occupied_country	Within country scope only	Country	`any_occupied_country = { … }`	Checks if any country that is occupied by the one where this scope is located - meaning that the occupied country has core states controlled by the occupier country - meets the triggers.	1.9
all_enemy_country	Within country scope only	Country	`all_enemy_country = { … }`	Checks if all countries that are at war with the one where this scope is located meet the triggers.	1.9
any_enemy_country	Within country scope only	Country	`any_enemy_country = { … }`	Checks if any country that are at war with the one where this scope is located meets the triggers.	1.9
all_subject_countries	Within country scope only	Country	`all_subject_countries = { … }`	Checks if all countries that are a subject of the one where this scope is located meet the triggers. Notice the plural spelling in the scope.	1.11
any_subject_country	Within country scope only	Country	`any_subject_country = { … }`	Checks if any country that is a subject of the one where this scope is located meets the triggers.	1.11
any_country_with_core	Within state scope only	Country	`any_country_with_core = { … }`	Checks if any country that has the current scope as a core state meets the triggers. **Does not have an equivalent for other effect/trigger scope types.**	1.12
all_state	Always usable	State	`all_state = { … }`	Check if all states meet the triggers.	1.0
any_state	Always usable	State	`any_state = { … }`	Check if any state meets the triggers.	1.0
all_neighbor_state	Within state scope only	State	`all_neighbor_state = { … }`	Check if all states that are neighbour to the one where this scope is located meet the triggers.	1.0
any_neighbor_state	Within state scope only	State	`any_neighbor_state = { … }`	Check if any state that is neighbour to the one where this scope is located meets the triggers.	1.0
all_owned_state	Within country scope only	State	`all_owned_state = { … }`	Check if all states that are owned by the country where this scope is located meet the triggers.	1.0
any_owned_state	Within country scope only	State	`any_owned_state = { … }`	Check if any state that is owned by the country where this scope is located meets the triggers.	1.0
all_core_state	Within country scope only	State	`all_core_state = { … }`	Check if any state that is cored by the country where this scope is located meets the triggers.	1.11
any_core_state	Within country scope only	State	`any_core_state = { … }`	Check if all states that are cored by the country where this scope is located meet the triggers.	1.11
all_controlled_state	Within country scope only	State	`all_controlled_state = { … }`	Check if all states that are controlled by the country where this scope is located meet the triggers.	1.9
any_controlled_state	Within country scope only	State	`any_controlled_state = { … }`	Check if any state that is controlled by the country where this scope is located meets the triggers.	1.9
all_unit_leader	Within country scope only	Unit Leader	`all_unit_leader = { … }`	Checks if all unit leaders (corps commanders, field marshals, admirals) that are employed by the country where this scope is located meet the triggers.	1.5
any_unit_leader	Within country scope only	Unit Leader	`any_unit_leader = { … }`	Checks if any unit leader (corps commander, field marshal, admiral) that is employed by the country where this scope is located meets the triggers.	1.5
all_army_leader	Within country scope only	Unit Leader	`all_army_leader = { … }`	Checks if all army leaders that are employed by the country where this scope is located meet the triggers.	1.5
any_army_leader	Within country scope only	Unit Leader	`any_army_leader = { … }`	Checks if any army leader that is employed by the country where this scope is located meets the triggers.	1.5
all_navy_leader	Within country scope only	Unit Leader	`all_navy_leader = { … }`	Checks if all navy leaders that are employed by the country where this scope is located meet the triggers.	1.5
any_navy_leader	Within country scope only	Unit Leader	`any_navy_leader = { … }`	Checks if any navy leader that is employed by the country where this scope is located meets the triggers.	1.5
all_operative_leader	Within country scope or operations only	Operative	`all_operative_leader = { … }`	Checks if all operatives that are employed by the country where this scope is located meet the triggers.	1.9
any_operative_leader	Within country scope or operations only	Operative	`any_operative_leader = { … }`	Checks if any operative that is employed by the country where this scope is located meets the triggers.	1.9
all_character	Within country scope only	Character	`all_character = { … }`	Checks if all characters that are recruited by the country where this scope is located meet the triggers.	1.11
any_character	Within country scope only	Character	`any_character = { … }`	Checks if any character that is recruited by the country where this scope is located meets the triggers.	1.11
any_country_division	Within country scope only	Division	`any_country_division = { … }`	Checks if any division owned by the current country meets the triggers.	1.12
any_state_division	Within state scope only	Division	`any_state_division = { … }`	Checks if any division within the current state meets the triggers.	1.12

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	`SOV = { country_event = my_event.1 }`	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	`123 = { transfer_state_to = SCO }`	The state defined by this id.	✓	1.0
<character>	Always usable	Character scope	`ENG_theodore_makhno = { set_nationality = UKR }`	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	ENG = {
			FRA = {
			GER = {
			declare_war_on = {
			target = ROOT
			type = annex_everything
			}
			}
			}
			} #GER declares war on ENG (if there is no scope before ENG)
				Targets the root node of the block, an inherent property of each block. Most commonly, this is the default scope: for example, ROOT within a national focus will always refer to the country doing the focus and ROOT within a event 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	set_temp_variable = { target_country = THIS.id }	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

PREV	Always usable	Depends on usage	FRA = {
			random_country = {
			GER = {
			declare_war_on = {
			target = PREV
			type = annex_everything
			}
			}
			}
			} #Germany declares war on random_country
				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	declare_war_on = {
			target = FROM
			type = annex_everything
			}
			
			FROM = {
			load_oob = defend_ourselves
			}
				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	`overlord = { … }`	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	`faction_leader = { add_to_faction = FROM }`	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	`owner = { add_ideas = owns_this_state }`	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	controller = {
			ROOT = {
			create_wargoal = {
			target = PREV
			type = take_state_focus
			generator = { 123 }
			}
			}
			}
				The controller of the current state. Subject to the 'invalid event target' error.	X	1.0

capital_scope	Within country scope only	State scope	`capital_scope = { … }`	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	`event_target:my_event_target = { … }`	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	`var:my_variable = { … }` `add_to_faction = my_variable` or `add_to_faction = var:my_variable`	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.

Flow control tools[编辑 | 编辑源代码]

Main article: Scopes#Flow control tools

These are triggers that serve as more of a way to establish a connection in how triggers are evaluated. Each one serves as a trigger scope with additional arguments and can be used regardless of scope.

Flow control tools:
折叠

Name	Additional parameters	Examples	Description	Notes	Version Added

AND	None	AND = {
		original_tag = GER
		has_stability > 0.5
		}
			Returns false if any sub-trigger returns false, true otherwise. Evaluation stops at the first false sub-trigger.	Only necessary within OR statements and NOT statements, as everything else is implicitly AND.	1.0

OR	None	OR = {
		original_tag = ENG
		original_tag = USA
		}
			Returns true if any sub-trigger returns true, false otherwise. Evaluation stops at the first true sub-trigger.		1.0

NOT	None	NOT = {
		has_stability > 0.5
		has_war_support > 0.5
		}
			Returns false if any sub-trigger returns true, true otherwise. Evaluation stops at the first true sub-trigger.	Equivalent to a NOR rather than a NAND.	1.0

count_triggers	`amount = <int>`
	The amount of triggers that need to be fulfilled.
		count_triggers = {
		amount = 2
		10 = { state_population = 100000 }
		11 = { state_population = 100000 }
		12 = { state_population = 100000 }
		}
			Sums the results of all sub-triggers (false=0, true=1) and returns true if the sum is at least `amount`.		1.5

if	`limit = <trigger block>`
	`else_if = <if-trigger>`
	Alternative condition. Optional.
	`else = <AND-trigger>`
	Final alternative condition. Optional.
		if = {
		limit = {
		has_dlc = "Poland: United and Ready"
		}
		has_political_power > 100
		}
		else_if = {
		limit = {
		has_dlc = "Waking the Tiger"
		}
		has_war_support > 0.5
		}
		else = {
		always = no
		}
			If `limit` is true, the sub-triggers are evaluated like an AND-trigger. If `limit` is false, `else_if` blocks are tried in sequence and finally `else` (if present). *Otherwise true is returned.*	Both nested (As in inside of `if = { ... }`) and unnested (As in the example) `else` and `else_if` exist. In case of overlap, unnested is preferred.
				Can be useful for tooltip management: same as in effects, if the limit is unmet, nothing appears. If it is met, then the limit is hidden while the condition outside of the limit appears in the tooltip.
					1.0

hidden_trigger	None	hidden_trigger = {
		country_exists = GER
		}
			Hides the triggers from the tooltip shown to the player.	Also serves as an AND statement.	1.0

custom_trigger_tooltip	`tooltip = <string>`
	The localisation key to use.
		custom_trigger_tooltip = {
		tooltip = sunrise_invasion_tt
		any_state = {
		is_owned_by = JAP
		is_on_continent = europe
		is_coastal = yes
		}
		}
			Hides the triggers from the tooltip shown to the player and instead uses the specified localisation key.	Also serves as an AND statement. When required to be false (as with NOT), the game will use the same localisation key but with `_NOT` appended (as in `sunrise_invasion_tt_NOT` in the example). If it doesn't exist, the same tooltip as for being true will get re-used, possibly causing player confusion.	1.0

Any scope[编辑 | 编辑源代码]

These triggers do not require any particular scope.

General[编辑 | 编辑源代码]

General any-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
always	<bool> Boolean.	always = yes	Always returns true or false. Useful for debugging.		1.0
has_global_flag	<string> The flag to check.	has_global_flag = my_flag	Checks if the specified flag has been set.		1.0

has_global_flag	flag = <string>
	The flag to check.
	value = <int>
	The flag value to check for. Optional.
	date = <date>
	The flag creation date to check for. Optional.
	days = <int>
	The duration the flag existed for. Optional.
		has_global_flag = {
		flag = my_flag
		days > 30
		date > 1936.6.1
		value > 0
		}
			Compares the specified flag's last set date, days since last set, and/or value.	If not set, the value comparison is >0. value is limited between -32768 and 32767.	1.0

has_dlc	<string> The DLC name to check for.	has_dlc = "Waking the Tiger"	Checks if the specified DLC is enabled.		1.0
has_start_date	<date> The date to check for.	has_start_date > 1950.01.01	Checks if the specified date was the start date used for the current game.	Year.Month.Day Must use either > or < operators.	1.0
date	<date> The date to check for.	date < 1950.01.01	Checks if the specified date against the current date.	Year.Month.Day Must use either > or < operators.	1.0
difficulty	<int> The difficulty value.	difficulty > 0	checks if the specified difficulty against the current difficulty.	Must use either > or < operators.	1.0
has_any_custom_difficulty_setting	<bool> Boolean.	has_any_custom_difficulty_setting = yes	Checks if any custom difficulty setting is changed from their default value.	Custom difficulty in this case refers to /Hearts of Iron IV/common /difficulty_settings/*.txt, used in base game to strengthen a specific country.	1.0
has_custom_difficulty_setting	<string> The setting to check.	has_custom_difficulty_setting = custom_diff_strong_sov	Checks if the specified custom difficulty setting is changed from the default value.	Custom difficulty in this case refers to /Hearts of Iron IV/common /difficulty_settings/*.txt, used in base game to strengthen a specific country.	1.0
game_rules_allow_achievements	<bool> Boolean.	game_rules_allow_achievements = yes	Checks if all of the active game rule options allow achievements.		1.9
country_exists	<scope> / <variable> The country to check.	country_exists = GER	Checks if the specified country currently exists in game.		1.0
is_ironman	<bool> Boolean.	is_ironman = yes	Checks if the current game is running in Ironman mode.		1.0
is_historical_focus_on	<bool> Boolean.	is_historical_focus_on = yes	Checks if the current game is running with Historical Focuses on.		1.0
is_tutorial	<bool> Boolean.	is_tutorial = yes	Checks if the current game is running in Tutorial mode.		1.0
is_debug	<bool> Boolean.	is_debug = yes	Checks if game is in debug mode (launched with -debug argument).		1.9
threat	<float> The amount to check for.	threat > 0.5	Checks if World Tension is above the specified amount.	Must use either > or < operators.	1.0
has_game_rule	<string> The game rule to check for. <string> / <bool> The option to check.	has_game_rule = { rule = GER_can_remilitarize_rhineland option = yes }	Checks if a game rule is set to a particular option.		1.5

Variables[编辑 | 编辑源代码]

Variable-related triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_variable	<variable> The variable to check.	has_variable = my_var	Checks if the specified variable exists for the current scope.		1.5

Name	Parameters	Examples	Description	Notes	Version Added

check_variable	`var = <variable>`
	The variable to check.
	`value = <float> / <variable>`
	The value to check for.
	`compare = <type>`
	The type of comparison. Optional, can use < or > instead.
		check_variable = {
		var = my_var
		value = 10
		compare = greater_than_or_equals
		}
		check_variable = {
		my_var > 10
		}
			Check the specified variable for the current scope.	Possible compare types:
				• less_than
				• less_than_or_equals
				• greater_than
				• greater_than_or_equals
				• equals
				• not_equals
					1.5

Remember that variables need to refer to the scope they were set in. This means you can't check a country variable in a state scope without scoping the variable.

For example, to get the country variable whilst in a state scope, you'd do the following:

```
<country> = {
  <state> = {
    limit = {
      check_variable = { from.my_country_var > 0.0 }
    }
  }
}
```

See Variables for more information.

Debugging[编辑 | 编辑源代码]

These are usable as both effects and triggers and are used for debugging, with the player never seeing them.

Debugging-helpful triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
log	`<string>` What to log. Supports dynamic localisation.	log = "Added [?temp_add] to [THIS.GetTag]'s variable [?THIS.varvalue]"	Appends an entry into the game.log and, if open, the console when evaluating the trigger.	game.log is stored within /Hearts of Iron IV/logs/ in the user directory	1.5

print_variables	`print_global = <bool>`
	Print global variables. Defaults to `no`.
	`var_list = <list>`
	The variables to print. Defaults to all variables.
	`file = <string>`The file path to log to. Defaults to "variable_dump". Does not include the `.log` extension.
	`text = <string>`Text to prepend. Defaults to "No Header".
	`append = <bool>`Whether to append to the file instead of overwrite. Defaults to `yes`
		print_variables = {
		var_list = { myvar1 myvar2 }
		file = "my_dump_file"
		text = "my header"
		}
			Dumps the specified variables from the current scope and optionally the global scope into a log file with the specified name.	The log will be within /Hearts of Iron IV/logs/variable_dumps/ in the user directory. See also debugging variables.	1.5

Country scope[编辑 | 编辑源代码]

Can be used in **country** scope.

General[编辑 | 编辑源代码]

General country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
exists	`<bool>` Boolean.	exists = yes	Checks if the current scope exists in game.		1.0
tag	`<scope> / <variable>` The country to check.	tag = GER tag = var:my_country	Checks if the current scope is the specified country.		1.0
original_tag	`<scope> / <variable>` The country to check.	original_tag = GER original_tag = var:my_country	Checks if the current scope originates from the specified country.	Dynamic countries originating from the specified country (alongside the specified country itself), via start_civil_war or create_dynamic_country, will have this trigger true.	1.0
is_ai	`<bool>` Boolean.	is_ai = yes	Checks if the current scope is AI.		1.0

has_collaboration	`target = <country>`
	The country to check.
	`value <> <decimal>`
	The value of the collaboration on the 0-1 scale.
		has_collaboration = {
		target = GER
		value > 0.5
		}
			Checks if the current scope has a collaboration level in the target scope.	The target is occupied by the current scope. Must use < or > in the value argument.	1.9

Name	Parameters	Examples	Description	Notes	Version Added
has_country_flag	<string> The flag to check.	has_country_flag = my_flag	Checks if the current scope has the specified flag.		1.0

has_country_flag	flag = <string>
	The flag to check.
	value = <int>
	The flag value to check for. Optional.
	date = <date>
	The flag creation date to check for. Optional.
	days = <int>
	The duration the flag existed for. Optional.
		has_country_flag = {
		flag = my_flag
		days > 30
		date > 1936.6.1
		value > 0
		}
			Compares the specified flag's last set date, days since last set, and/or value.	If not set, the value comparison is >0. value is limited between -32768 and 32767.	1.0

has_cosmetic_tag	<string> The cosmetic tag to check.	has_cosmetic_tag = SOV_custom	Checks if the current scope has the specified cosmetic tag active.		1.5
has_event_target	<event target> The event target to check.	has_event_target = my_var	Checks if the current scope is assigned as the specified event target.		1.0
has_decision	<string> The decision to check.	has_decision = my_decision	Checks if the current scope has the specified decision activated.		1.5

has_dynamic_modifier	modifier = <string>
	The dynamic_modifier to check.
	scope = <scope>
	The country to check. Optional, if the original modifier has been targeted.
		has_dynamic_modifier = {
		modifier = my_dynamic_modifier
		scope = GER
		}
			Checks if the current scope has the specified dynamic modifier activated.		1.6

has_active_mission	<string> The mission to check.	has_active_mission = my_mission	Checks if the current scope has the specified mission active.		1.5
has_focus_tree	<string> The focus tree to check.	has_focus_tree = soviet_tree	Checks if the current scope has the specified focus tree.		1.3
has_completed_focus	<string> The focus to check.	has_completed_focus = my_focus	Checks if the current scope has the specified focus completed.		1.0

focus_progress	focus = <string>
	The focus to check.
	progress = <string>
	The progress to check for.
		focus_progress = {
		focus = my_focus
		progress > 0.5
		}
			Checks if the specified focus has been completed the specified percent for the current scope.	Must use either > or < operators for progress.	1.0

has_country_custom_difficulty_setting	<bool> Boolean.	has_country_custom_difficulty_setting = yes	Checks if the any custom difficulty setting targeting the current scope is changed from the default value.	Custom difficulty in this case refers to /Hearts of Iron IV/common/difficulty_settings /*.txt, used in base game to strengthen a specific country.	1.0
has_terrain	<terrain> Terrain.	has_terrain = urban	Checks if the current scope has any provinces of the specified terrain.	Only can be used in country scope.	1.11
is_dynamic_country	<bool> Boolean.	is_dynamic_country = yes	Checks if the current scope is a dynamic country.	Dynamic countries include those generated in civil wars as well as those generated with the create_dynamic_country effect, such as collaboration governments.	1.11
num_of_supply_nodes	<int> The amount to check for.	num_of_supply_nodes > 10	Checks if the current scope has the specified amount of supply nodes under control.	Can only use < or > operators.	1.11
has_completed_custom_achievement	mod = <mod ID> The mod where the achievement is from. achievement = <achievement ID> The name of the	has_completed_custom_achievement = { mod = my_mod_unique_id achievement = my_achievement_token	Checks if the player controlling the current scope has completed the	The achievement (including the ID of the mod it's from) is defined within /Hearts of Iron IV/common/achievements/*.txt files[1]. The achievement could be	1.12.5

Name	Parameters	Examples	Description	Notes	Version Added
	achievement.	}	specified custom achievement.	completed during a previous session, not necessarily the current one. If the mod defining the achievement is not loaded, the trigger evaluates as false.	

Politics[编辑 | 编辑源代码]

Political country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
<ideology>	`<ideology> = <float> / <variable>` The amount of the ideology to check for.	fascism > 0.5 democratic > party_popularity@communism	Checks if the current scope has popularity of the specified ideology above the specified amount.		1.0
has_political_power	`<float>` The amount to check for.	has_political_power > 100	Checks if the current scope has the specified amount of political power.	Must use either > or < operators.	1.0
political_power_daily	`<float> / <variable>` The amount to check for.	political_power_daily > 1	Checks if the current scope has the specified amount of daily political power gain.	Must use either > or < operators.	1.5
political_power_growth	`<float> / <variable>` The amount to check for.	political_power_growth > 1	Checks if the current scope has the specified amount of daily political power gain.	Must use either > or < operators.	1.5
command_power	`<float> / <variable>` The amount to check for.	command_power > 1	Checks if the current scope has the specified amount of command power.	Must use either > or < operators.	1.5
command_power_daily	`<float> / <variable>` The amount to check for.	command_power_daily > 1	Checks if the current scope has the specified amount of daily command power gain.	Must use either > or < operators.	1.5
has_war_support	`<float> / <variable>` The amount to check for.	has_war_support > 0.5	Checks if the current scope has the specified percentage of War Support.	Must use either > or < operators.	1.5
has_stability	`<float> / <variable>` The amount to check for.	has_stability > 0.5	Checks if the current scope has the specified percentage of Stability.	Must use either > or < operators.	1.5

has_government	`<ideology>`
	The ideology group to check for.
	OR
	`<country>`
	The country to compare with.
		has_government = fascism
		has_government = ROOT
			Checks if the ruling party of the current scope meets the requirements of being either the specified ideology group or having the same ideology group as the specified country.		1.0

has_elections	`<bool>` Boolean.	has_elections = yes	Checks if the current scope holds elections.		1.0
is_staging_coup	`<bool>` Boolean.	is_staging_coup = yes	Checks if the current scope is staging a coup.		1.3
is_target_of_coup	`<bool>` Boolean.	is_target_of_coup = yes	Checks if the current scope is the target of a coup.		1.0
has_civil_war	`<bool>` Boolean.	has_civil_war = yes	Checks if the current scope has a civil war active.		1.0
civilwar_target	`<scope>` The target country.	civilwar_target = GER	Checks if the specified country is a target of a civil war.		1.0

has_manpower_for_recruit_change_to	`value = <float>`
	The amount to check for.
	`group = <group>`
	The group to check for.
		has_manpower_for_recruit_change_to = {
		value > 0.05
		group = mobilization_laws
		}
			Checks if the current scope has the specified amount of manpower for changing the specified idea group.	**Must** use either > or < operators as = operator checks for the **exact** value	1.0

has_rule	`<string>` The rule to check for.	has_rule = can_create_factions	Checks if the current scope has the specified country rule.		1.6
has_casualties_war_support	`<float> / <variable>` The amount to check for.	has_casualties_war_support < 0	Checks if the current scope has the specified percentage of war support from own combat casualties.	Must use either > or < operators.	1.12
has_convoys_war_support	`<float> / <variable>` The amount to check for.	has_convoys_war_support < 0	Checks if the current scope has the specified percentage of war support from own convoys sunk.	Must use either > or < operators.	1.12

Name	Parameters	Examples	Description	Notes	Version Added
has_bombing_war_support	<float> / <variable> The amount to check for.	has_bombing_war_support < 0	Checks if the current scope has the specified percentage of war support from own states bombed by the enemy.	Must use either > or < operators.	1.12

Balance of power[编辑 | 编辑源代码]

Balances of power are stored within /Hearts of Iron IV/common/bop/*.txt files.

Balance of power-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added

has_power_balance	id = <bop ID>
	The balance to check for.
		has_power_balance = {
		id = TAG_my_bop
		}
			Checks if the current scope has the specified balance of power active.		1.12

has_any_power_balance	<bool> Boolean.	has_any_power_balance = yes	Checks if the current scope has any balance of power active.		1.12

power_balance_value	id = <bop ID>
	The balance to check in.
	value = <float>
	The value to check for.
		power_balance_value = {
		id = TAG_my_bop
		value > 0.7
		}
			Checks if the current scope has the specified value within the balance of power.	Either =, >, or < operators are allowed.	1.12

power_balance_daily_change	id = <bop ID>
	The balance to check in.
	value = <float>
	The value to check for.
		power_balance_daily_change = {
		id = TAG_my_bop
		value < -0.01
		}
			Checks if the current scope's balance of power changes each day by the specified value.	Either =, >, or < operators are allowed.	1.12

power_balance_weekly_change	id = <bop ID>
	The balance to check in.
	value = <float>
	The value to check for.
		power_balance_weekly_change = {
		id = TAG_my_bop
		value < -0.01
		}
			Checks if the current scope's balance of power changes each week by the specified value.	Either =, >, or < operators are allowed.	1.12

is_power_balance_in_range	id = <bop ID>
	The balance to check in.
	range = <range ID>
	The range to check for.
		is_power_balance_in_range = {
		id = TAG_my_bop
		range = TAG_my_bop_right_range
		}
			Checks if the current scope's balance of power value lies within the specified range.	Ranges are defined within the balance of power. Can use either =, >, and < operators. In case of > or <, the comparison is 'strict', i.e. excluding the range itself.	1.12

is_power_balance_side_active	id = <bop ID>
	The balance to check in.
	side = <side ID>
	The side to check.
		is_power_balance_side_active = {
		id = TAG_my_bop
		side = TAG_my_bop_right_range
		}
			Checks if the current scope's balance of power value makes a side active.	Sides are defined within the balance of power.	1.12

has_power_balance_modifier	id = <bop ID>
	The balance to check in.
	modifier = <modifier ID>
	The static modifier.
		has_power_balance_modifier = {
		id = TAG_my_bop
		modifier = TAG_my_bop_modifier
		}
			Checks if the current scope's balance of power value activates a modifier.	BoP modifiers are defined within /Hearts of Iron IV/common/modifiers/*.txt files, while they're activated in the balance of power definition.	1.12

Buildings[编辑 | 编辑源代码]

Building-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
<building>	<building> = <int> The amount of the specified building to to check for.	arms_factory > 10	Checks if the current scope has the specified amount of the specified building.	Must use either > or < operators.	1.0
num_of_military_factories	<int> The amount to check for.	num_of_military_factories > 10	Checks if the current scope has the specified amount of military factories.	Must use either > or < operators.	1.0
num_of_civilian_factories	<int> The amount to check for.	num_of_civilian_factories > 10	Checks if the current scope has the specified amount of civilian factories.	Must use either > or < operators.	1.0
num_of_naval_factories	<int> The amount to check for.	num_of_naval_factories > 10	Checks if the current scope has the specified amount of dockyards.	Must use either > or < operators.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
num_of_available_military_factories	<int> The amount to check for.	num_of_available_military_factories > 10	Checks if the current scope has the specified amount of available military factories.	Must use either > or < operators.	1.0
num_of_available_civilian_factories	<int> The amount to check for.	num_of_available_civilian_factories > 10	Checks if the current scope has the specified amount of available civilian factories.	Must use either > or < operators.	1.0
num_of_available_naval_factories	<int> The amount to check for.	num_of_available_naval_factories > 10	Checks if the current scope has the specified amount of available dockyards.	Must use either > or < operators.	1.0
num_of_factories	<int> The amount to check for.	num_of_factories > 10	Checks if the current scope has the specified amount of military, civilian or dockyard factories.	Must use either > or < operators.	1.0
num_of_controlled_factories	<int> The amount to check for.	num_of_controlled_factories > 10	Checks if the current scope has the specified amount of military, civilian or dockyard factories under control.	Must use either > or < operators.	1.11
num_of_owned_factories	<int> The amount to check for.	num_of_owned_factories > 10	Checks if the current scope has the specified amount of military, civilian or dockyard factories under owned states.	Must use either > or < operators.	1.11
num_of_civilian_factories_available_for_projects	<int> The amount to check for.	num_of_civilian_factories_available_for_projects > 10	Checks if the current scope has the specified amount of civilian factories usable for projects.	Must use either > or < operators.	1.5

ic_ratio	tag = <scope>
	The country to check.
	ratio = <float>
	The ratio to check for.
		ic_ratio = {
		tag = GER
		ratio > 0.5
		}
			Checks if the current scope has the specified ratio of factories with the target country.	Must use either > or < operators for ratio.	1.0

has_damaged_buildings	<bool> Boolean.	has_damaged_buildings = yes	Checks if the current scope has any damanged buildings in their states.		1.0

has_built	type = <building>
	The building to check for.
	value = <int>
	The amount to check for.
		has_built = {
		type = arms_factory
		value > 10
		}
			Checks if the current scope has built the specified building the specified number of times.	Must use either > or < operators for value.	1.0

Technology[编辑 | 编辑源代码**]**

Technology-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_tech	<string> The technology to check for.	has_tech = my_technology	Checks if the current scope has the specified technology.		1.0
is_researching_technology	<string> The technology to check for.	is_researching_technology = my_tech	Checks if the current scope is currently researching the specified technology.		1.0
can_research	<string> The technology to check for.	can_research = my_tech	Checks if the current scope can start researching the specified technology.		1.0
original_research_slots	<int> The amount to check for.	original_research_slots > 3	Checks if the current scope had the specified amount of slots at game start.	Must use either > or < operators.	1.0
amount_research_slots	<int> The amount to check for.	amount_research_slots > 3	Checks if the current scope has the specified amount of research slots.	Must use either > or < operators.	1.3
is_in_tech_sharing_group	<string> The group to check for.	is_in_tech_sharing_group = us_research	Checks if the current scope is in the specified technology sharing group.		1.3
num_tech_sharing_groups	<int> The amount to check for.	num_tech_sharing_groups > 3	Checks if the current scope is in the specified amount of technology sharing groups.	Must use either > or < operators.	1.3

has_tech_bonus	technology = <string>
	The technology to check for. Optional.
	category = <string>
		has_tech_bonus = {
		technology = my_tech
		}
		has_tech_bonus = {
			Checks if the current scope has a technology bonus in the specified category, or for the specific technology.		1.3

Name	Parameters	Examples	Description	Notes	Version Added
	The category to check for. Optional.	category = my_category }			
land_doctrine_level	<int> The amount to check for.	land_doctrine_level > 2	Checks if the current scope has the specified amount of land doctrine technologies.	Must use either > or < operators.	1.0
num_researched_technologies	<int> The amount to check for.	num_researched_technologies > 10	Checks how many technologies the target has researched.	Must use either > or < operators.	1.3

Ideas[编辑 | 编辑源代码**]**

Idea-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_idea	<string> The idea to check for.	has_idea = my_idea	Checks if the current scope has the specified idea.		
has_idea_with_trait	<string> The trait to check for.	has_idea_with_trait = my_trait	Checks if the current scope has any ideas with the specified trait.		1.0

has_allowed_idea_with_traits	idea = <string>
	The trait to check for.
	limit = <int>
	The amount to check for.
	characters = <bool>
	If set, will only run this on characters.
	ignore = { <ideas> }
	If set, ignores the ideas inside. Optional.
		has_available_idea_with_traits = {
		idea = my_trait
		limit = 1
		ignore = { generic_head_of_intelligence }
		}
			Checks if the current scope has the specified amount of ideas with the specified trait.	ignore = idea_name works for 1 idea.	1.9.1

has_available_idea_with_traits	idea = <string>
	The trait to check for.
	limit = <int>
	The amount to check for.
	characters = <bool>
	If set, will only run this on characters.
	ignore = { <ideas> }
	If set, ignores the ideas inside. Optional.
		has_available_idea_with_traits = {
		idea = my_trait
		limit = 1
		ignore = { generic_head_of_intelligence }
		}
			Checks if the current scope has the specified amount of ideas with the specified trait.	ignore = idea_name works for 1 idea.	1.0

amount_taken_ideas	amount = <int>
	The amount to check for.
	slots = { <string> }
	The slot type.
		amount_taken_ideas = {
		amount > 3
		slots = {
		political_advisor
		}
		}
			Checks if the current scope has the specified amount of ideas of the specified slot type. Excludes spirits, hidden ideas, and laws.	Slots types are found in /Hearts of Iron IV/common/idea_tags /*.txt.	1.4

Diplomacy[编辑 | 编辑源代码**]**

Diplomatic country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
is_major	<bool> Boolean.	is_major = yes	Checks if the current scope is considered a Major.		1.0
is_ally_with	<scope> / <variable> The country to check for.	is_ally_with = GER is_ally_with = var:country	Checks if the current scope is an ally (Faction members or subject-master relation).		1.0
is_in_faction_with	<scope> / <variable> The country to check for.	is_in_faction_with = GER is_in_faction_with = var:country	Checks if the current scope is in a faction with the specified country.		1.0
is_in_faction	<bool> Boolean.	is_in_faction = yes	Checks if the current scope is in a faction.		1.0
is_faction_leader	<bool> Boolean.	is_faction_leader = yes	Checks if the current scope is the leader of a faction.		1.0
is_spymaster	<bool> Boolean.	is_spymaster = yes	Checks if the current scope is the spymaster of a faction.		1.9
num_faction_members	<int> The amount to check for.	num_faction_members > 1	Checks if the faction of the current scope has the specified amount of members.	Must use either > or < operators.	1.0
has_non_aggression_pact_with	<scope> / <variable> The country to check for.	has_non_aggression_pact_with = GER	Checks if the current scope has a non-aggression pact with the specified country.		1.0

Name	Parameters	Examples	Description	Notes	Version Added
is_guaranteed_by	<scope> / <variable> The country to check for.	is_guaranteed_by = GER	Checks if the current scope has been guaranteed by the specified country.		1.0
has_guaranteed	<scope> / <variable> The country to check for.	has_guaranteed = GER	Checks if the current scope has guaranteed the specified country.		1.0
has_military_access_to	<scope> / <variable> The country to check for.	has_military_access_to = GER	Checks if the current scope has military access to the specified country.		1.0
gives_military_access_to	<scope> / <variable> The country to check for.	gives_military_access_to = GER	Checks if the current scope gives military to the specified country.		1.0
is_neighbor_of	<scope> / <variable> The country to check for.	is_neighbor_of = GER	Checks if the current scope is a neighbor of the specified country.		1.0
is_owner_neighbor_of	<scope> / <variable> The country to check for.	is_owner_neighbor_of = GER	Checks if the current scope is a neighbor of the specified country with their core territory only.		1.0
is_puppet_of	<scope> / <variable> The country to check for.	is_puppet_of = GER	Checks if the current scope is a puppet of the specified country.	A "puppet" is an autonomous state that has `is_puppet = yes` in its definition within /Hearts of Iron IV/common /autonomous_states/. For any subject type, see is_subject_of.	1.0
is_subject_of	<scope> / <variable> The country to check for.	is_subject_of = GER	Checks if the current scope is a subject of the specified scope.		1.0
is_puppet	<bool> Boolean.	is_puppet = yes	Returns true if the current country is a puppet.	A "puppet" is an autonomous state that has `is_puppet = yes` in its definition within /Hearts of Iron IV/common /autonomous_states/. For any subject type, see is_subject.	1.0
is_subject	<bool> Boolean.	is_subject = yes	Checks if the current scope is a subject.		1.0
has_subject	<bool> Boolean.	has_subject = GRE	Checks if the country has for subject the given country.		1.0
num_subjects	<int> The amount to check for.	num_subjects > 3	Checks if the current scope has the specified amount of subjects.	Must use either > or < operators.	1.3
has_autonomy_state	<string> The autonomy state to check for.	has_autonomy_state = autonomy_dominion	Checks if the current scope is in the specified autonomous state.		1.0
compare_autonomy_state	<string> The autonomy state to check for.	compare_autonomy_state > autonomy_dominion	Checks if the current scope's autonomy state `min_freedom_level` is less or greater than that of the specified autonomy state. The special value "autonomy_free" compares as greater than any autonomy state. If the current scope is not a subject, it is treated as greater than any autonomy state (including "autonomy_free"). With `=`, checks if the current scope is in the specified autonomous state.		1.0
compare_autonomy_progress_ratio	<float> The amount to check for.	compare_autonomy_progress_ratio > 0.5	Checks if the current scope autonomy progress is at the specified ratio. If the current scope is not a subject, the ratio is 1.		1.3
has_opinion_modifier	<string> The opinion modifier to check for.	has_opinion_modifier = my_modifier	Checks if the current scope has the specified opinion modifier.		1.0
has_opinion	`target = <scope>` The country to check for. `value = <float>` The amount to check for.	has_opinion = { target = GER value > 50 }	Checks if the current scope has the specified opinion with the specified country.	Must use either > or < operators.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
has_relation_modifier	target = <scope> The country to check for. modifier = <modifier> The modifier to check for.	has_relation_modifier = { target = GER modifier = my_modifier }	Checks if the current scope has the specified relation modifier with the specified country.		1.0
has_legitimacy	<int> Amount to check.	has_legitimacy > 50	Checks how much legitimacy the current government in exile has.	Must use either > or < operators. Legitimacy ranges from 0 to 100.	1.6
is_exile_host	<bool> Boolean.	is_exile_host = yes	Checks if the current country is hosting an exile.		1.6
is_hosting_exile	<tag> Country.	is_hosting_exile = POL	Checks if the current country is hosting a specific exile.		1.6
is_government_in_exile	<bool> Boolean.	is_government_in_exile = yes	Checks if the current country is exiled in a different country.		1.6
is_exiled_in	<tag> Country to be exiled in.	is_exiled_in = POL	Checks if the current country is exiled in a specific country.		1.6
received_expeditionary_forces	sender = <tag> Country which sent forces. value <> <int> Amount of forces.	received_expeditionary_forces = { sender = POL value > 10 }	Checks if the current country received X units in expeditions from the specified country.		1.6
can_declare_war_on	<tag> Country to check.	can_declare_war_on = POL	Checks if the current scope is able to declare war on the specified country.		1.9
foreign_manpower	<int> Amount to check.	foreign_manpower > 10000	Checks how much foreign manpower we have received for garrisoning.	Must use either > or < operators.	1.9
is_embargoed_by	<scope> Amount to check.	is_embargoed_by = USA	Checks if the current scope is embargoed by the specified country.		1.12
is_embargoing	<scope> Amount to check.	is_embargoing = CUB	Checks if the current scope is embargoing the specified country.		1.12

War[编辑 | 编辑源代码**]**

War-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_war	<bool> Boolean.	has_war = yes	Checks if the current scope is at war.		1.0
has_war_with	<scope> / <variable> The country to check for.	has_war_with = GER has_war_with = var:country	Checks if the current scope is at war with the specified country.		1.0
has_offensive_war_with	<scope> / <variable> The country to check for.	has_offensive_war_with = GER	Checks if the current scope is in an offensive war against the specified country.		1.0
has_defensive_war_with	<scope> / <variable> The country to check for.	has_defensive_war_with = GER	Checks if the current scope is in an defensive war against the specified country.		1.0
has_offensive_war	<bool> Boolean.	has_offensive_war = yes	Checks if the current scope is in an offensive war.		1.0
has_defensive_war	<bool> Boolean.	has_defensive_war = yes	Checks if the current scope is in a defensive war.		1.0
has_war_together_with	<scope> / <variable> The country to check for.	has_war_together_with = GER	Checks if the current scope is in a war alongside the specified country.		1.0
has_war_with_major	<bool> Boolean.	has_war_with_major = yes	Checks if the current scope is at war with any other country that is considered major.		1.12
has_war_with_wargoal_against	target = <scope> / <variable> The country to check for. type = <wargoal> The wargoal to check for. Optional.	has_war_with_wargoal_against = { target = ENG type = independence_wargoal }	Checks if the current scope is at war with the specified country with the specified wargoal being active.	Wargoals are stored within /Hearts of Iron IV/common/wargoals/*.txt files. If no wargoal is specified, checks for *any* wargoal. Joining an ally in their war does not count as a wargoal.	1.12

Name	Parameters	Examples	Description	Notes	Version Added
surrender_progress	`<float>` / `<variable>` The amount to check for.	surrender_progress > 0.1	Checks if the current scope has the specified amount of surrender progress.	Must use either > or < operators.	1.0
any_war_score	`<float>` The amount to check for.	any_war_score > 10	Checks if the current scope has the specified amount of **war progress** (not war participation)[2] in any war.	Must use either > or < operators.	1.0
has_capitulated	`<bool>` Boolean.	has_capitulated = yes	Checks if the current scope has capitulated.		1.0
days_since_capitulated	`<int>` Amount of days.	days_since_capitulated > 10	Checks the amount of days since the target last capitulated.	If the target never capitulated, the amount of days is extremely large. Recommended to combine with has_capitulated.	1.9
has_border_war_with	`<scope>` / `<variable>` The country to check for.	has_border_war_with = GER	Checks if the current scope has a border war with the specified country.		1.5

has_border_war_between	`attacker = <scope> / <variable>` The state to check for. `defender = <scope> / <variable>` The state to check for.	has_border_war_between = {
		attacker = 1
		defender = 2
		}
			Checks if there is a border war between the two specified states.		1.5

has_border_war	`<bool>` Boolean.	has_border_war = yes	Checks if the current scope has a border war active.		1.5
has_added_tension_amount	`<float>` / `<variable>` The amount to check for.	has_added_tension_amount > 10	Checks if the current scope has caused the specified amount of World Tension.	Must use either > or < operators.	1.0
has_wargoal_against	`<scope>` / `<variable>` The country to check for.	has_wargoal_against = GER	Checks if the current scope has any wargoal against the specified country.		1.0

has_wargoal_against	`target = <scope> / <variable>` The country to check for. `type = <string>` The type of wargoal to check for.	has_wargoal_against = {
		target = FROM
		type = take_state
		}
			Checks if the current scope has a specific wargoal type against the specified country.		1.8

is_justifying_wargoal_against	`<scope>` / `<variable>` The country to check for.	is_justifying_wargoal_against = GER	Checks if the current scope is justifying a wargoal against the specified country.		1.0
has_annex_war_goal	`<scope>` / `<variable>` The country to check for.	has_annex_war_goal = GER	Checks if the current scope has the Annex wargoal against the specified country.		1.0
any_claim	`<bool>` Boolean.	any_claim = yes	Checks if the current scope has any claims on another country.		1.0
is_in_peace_conference	`<bool>` Boolean.	is_in_peace_conference = yes	Checks if the current scope is in a peace conference.	Please test this in-game for 1.12.	1.0
controls_province	`<id>` The province to check for.	controls_province = 1239	Checks if the current scope has control of the specified province.		1.9

State[编辑 | 编辑源代码]

These are state-related triggers in the country scope, not state-scoped triggers.

State-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
controls_state	`<scope> / <variable>` The state to check for.	controls_state = 39 controls_state = var:state	Checks if the current scope has control of the specified state.		1.0
owns_state	`<scope> / <variable>` The state to check for.	owns_state = 39	Checks if the current scope owns the specified state.		1.0
num_of_controlled_states	`<int>` The amount to check for.	num_of_controlled_states > 5	Checks if the current scope has the specified amount of controlled states.	Must use either > or < operators.	1.0
num_occupied_states	`<int>` The amount to check for.	num_occupied_states > 5	Checks if the current scope has the specified amount of occupied states.	Must use either > or < operators.	1.0
has_full_control_of_state	`<scope> / <variable>` The state to check for.	has_full_control_of_state = 39	Checks if the current scope has total control (100% occupation) of the specified state.		1.3

Name	Parameters	Examples	Description	Notes	Version Added

core_compliance	occupied_country_tag = <TAG>
	The country for which to check compliance.
	value = <int>
	The value to check for.
		core_compliance = {
		occupied_country_tag = ITA
		value > 10
		}
			Compares the average compliance of core states of the specified country within controlled states of the current scope.	Must use either > or < operators for value.	1.9

core_resistance	occupied_country_tag = <TAG>
	The country for which to check resistance.
	value = <int>
	The value to check for.
		core_resistance = {
		occupied_country_tag = ITA
		value > 10
		}
			Compares the average resistance of core states of the specified country within controlled states of the current scope.	Must use either > or < operators for value.	1.9

garrison_manpower_need	<int> Amount to check.	garrison_manpower_need > 10000	Checks how much garrison manpower we need for resistance in controlled states.	Must use either > or < operators.	1.9
has_core_occupation_modifier	occupied_country_tag = <scope> / <variable> The country to check. modifier = <token>The modifier to check.	has_core_occupation_modifier = { occupied_country_tag = ITA modifier = token }	Checks if the current scope has an occupation modifier for resistance/compliance that applies to our occupied states of a specified country.		1.9

occupation_law	<law ID>
	The law to check.
		POL = { POL = { occupation_law = brutally_oppressive_occupation } } # Checks POL's default occupation law
		HOL = { BEL = { occupation_law = foreign_civilian_oversight } } # Checks HOL's occupation law over BEL
			Checks the occupation law that's either the default or applied over a specific country.	Checks PREV's occupation law over the current country. If they're the same scope, checks the default occupation law.	1.12

Military[编辑 | 编辑源代码]

Military-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_army_experience	<float> / <variable> The amount to check for.	has_army_experience > 10 has_army_experience > var:number	Checks if the current scope has the specified amount of Army experience.	**Must** use either > or < operators as = operator checks for the **exact** value	1.3
has_air_experience	<float> / <variable> The amount to check for.	has_air_experience > 10	Checks if the current scope has the specified amount of Air experience.	**Must** use either > or < operators as = operator checks for the **exact** value	1.3
has_navy_experience	<float> / <variable> The amount to check for.	has_navy_experience < 10	Checks if the current scope has the specified amount of Navy experience.	**Must** use either > or < operators as = operator checks for the **exact** value	1.3
has_manpower	<float> / <variable> The amount to check for.	has_manpower > 1000	Checks if the current scope has the specified amount of manpower.	**Must** use either > or < operators as = operator checks for the **exact** value	1.0

has_army_manpower	size = <int>
	The amount to check for.
		has_army_manpower = {
		size > 1000
		}
			Checks if the current scope has an army using the specified amount of manpower.	Must use either > or < operators.	1.0

manpower_per_military_factory	<float> The amount to check for.	manpower_per_military_factory > 1000	Checks if the current scope has the specified manpower times their number of military factories.	Must use either > or < operators.	1.0
conscription_ratio	<float> / <variable> The ratio to compare with.	conscription_ratio < 0.2	Checks if the current scope has the specified conscription ratio currently, not to be mixed up with the target conscription ratio.	**Must** use either > or < operators as = operator checks for the **exact** value	1.9
current_conscription_amount	<float> / <variable> The amount to compare with.	current_conscription_amount > 2000	Checks if the current scope has already conscripted that much manpower.	**Must** use either > or < operators as = operator checks for the **exact** value	1.9

Name	Parameters	Examples	Description	Notes	Version Added
target_conscription_amount	<float> / <variable> The amount to compare with.	target_conscription_amount > 2000	Checks if the current scope is targeting to conscript that much manpower.	**Must** use either > or < operators as = operator checks for the **exact** value	1.9
num_divisions	<int> The amount to check for.	num_divisions > 5	Checks if the current scope has the specified amount of divisions.	Must use either > or < operators.	1.3
num_of_nukes	<int> The amount to check for.	num_of_nukes > 5	Checks if the current scope has the specified amount of nukes.	Must use either > or < operators.	1.0
casualties	<int> The amount to check for.	casualties > 10000	Checks if the current scope has suffered the specified amount of casualties.	Must use either > or < operators.	1.0
casualties_k	<int> The amount to check for.	casualties_k > 10	Checks if the current scope has suffered the specified amount of casualties in thousands.	Must use either > or < operators.	1.0

casualties_inflicted_by	opponent = <tag>
	The tag that inflicted the casualties.
	thousands <> <int>
	The amount of casualties in thousands.
		casualties_inflicted_by = {
		opponent = POL
		thousands > 10
		}
			Checks if the current scope has suffered the specified amount of casualties in thousands from a specific country.	Must use either > or < operators for thousands.	1.6

amount_manpower_in_deployment_queue	<float> The amount to check for.	amount_manpower_in_deployment_queue > 1000	Checks if the current scope has the specified amount of manpower in their deployment queue.	Must use either > or < operators.	1.5
has_attache_from	<scope> / <variable> The country to check for.	has_attache_from = GER	Checks if the current scope has an attache from the specified scope.		1.5
has_attache	<bool> Boolean.	has_attache = yes	Checks if the current scope has an attache.		1.5
is_lend_leasing	<scope> / <variable> The country to check for.	is_lend_leasing = GER	Checks if the current scope is lend leasing to the specified scope.		1.0
has_template	<string> The name of the template.	has_template = "Infantry Division"	Checks if the current scope has a division template of the specified name.		1.0
has_template_majority_unit	<string> The unit to check for.	has_template_majority_unit = infantry	Checks if the current scope has a division template composed mostly of the specified unit.		1.0
has_template_containing_unit	<string> The name of the unit.	has_template_containing_unit = light_armor	Checks if the current scope has a division template contained any of the specified unit.		1.0

strength_ratio	tag = <scope>
	The country to check for.
	ratio <> <float>
	The ratio to check for.
		strength_ratio = {
		tag = GER
		ratio > 1
		}
			Checks if the current scope has the specified strength ratio against the specified country. The ratio is the number of fielded divisions of the current scope divided by those of tag (or 1 if tag has no	Must use > or < in the ratio.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
			divisions). The ratio gets increased by 10% if the current scope has a stronger air forces.[3]		
naval_strength_ratio	`tag = <scope>` The country to check for. `ratio <> <float>` The ratio to check for.	naval_strength_ratio = { tag = GER ratio <> 1 }	Checks if the current scope has the specified naval strength ratio against the specified country.	Must use > or < in the ratio.	1.0
naval_strength_comparison	`other = <scope>` The country to check for. `tooltip = <string>` The ratio to check for. Optional. `ratio <> <float>` The ratio to check for. `sub_unit_def_weights = { ... }` The weight to assign to each unit. Optional.	naval_strength_comparison = { other = POL tooltip = my_loc_key_tt ratio > 1 sub_unit_def_weights = { carrier = 1 submarine = 2 } }	Checks if the current scope has the specified naval strength ratio against the specified country.	Must use > or < in the ratio. If sub_unit_def_weights is unset, each unit is assumed to have 1 weight. If sub_unit_def_weights is set, only specified units will be counted towards strength. Units are defined in /Hearts of Iron IV/common/units /*.txt.	1.6
alliance_strength_ratio	`<float> / <variable>` The ratio to check for.	alliance_strength_ratio > 0.5	Checks if the current scope and allies has an army strength higher than the specified ratio against estimated enemy strength.	Must use either > or < operators.	1.0
alliance_naval_strength_ratio	`<float> / <variable>` The ratio to check for.	alliance_naval_strength_ratio > 0.5	Checks if the current scope and allies has an naval strength ratio higher than the specified ratio against estimated enemy strength.	Must use either > or < operators.	1.0
enemies_strength_ratio	`<float> / <variable>` The ratio to check for.	enemies_strength_ratio > 0.5	Checks if the estimated enemy army strength ratio is higher than the specified ratio.	Must use either > or < operators.	1.0
enemies_naval_strength_ratio	`<float> / <variable>` The ratio to check for.	enemies_naval_strength_ratio > 0.5	Checks if the estimated enemy naval strength ratio is higher than the specified ratio.	Must use either > or < operators.	1.0
has_army_size	`size = <float>` The amount to check for. `type = <string>` The battalion type to check for. Divisions that are majority made up of battalions in that type count. Optional, counts all divisions by default.	has_army_size = { size > 10 type = armor }	Checks if the current scope has the specified number of divisions, or of a specified type of division.	Battalion types are defined within /Hearts of Iron IV/common/units /*.txt files. Must use either > or < operators for size.	1.0
has_navy_size	`size = <float>/<variable>` The amount to check for. `type = <string>` The type to check for. Optional. `archetype = <string>` The ship archetype to check for. Optional.	has_navy_size = { size > 10 type = capital_ship archetype = ship_hull_heavy }	Checks if the current scope has the specified number of ships, or of a specified type of ship.	Ship types are defined within /Hearts of Iron IV/common/units/*.txt files. Must use either > or < operators for size. Ship archetypes are found in /Hearts of Iron IV/common/units /equipment/*.txt files.	1.0
has_deployed_air_force_size	`size = <float>` The amount to check for. `type = <string>` The type to check for. Optional.	has_deployed_air_force_size = { size > 10 type = cas }	Checks if the current scope has the specified number of aircraft, or of a specified type of aircraft.	Airwing types are defined within /Hearts of Iron IV/common/units/*.txt files. Must use either > or < operators for size.	1.0

Name	Parameters	Examples	Description	Notes	Version Added

divisions_in_state	size = <float>
	The amount to check for.
	type = <string>
	The battalion type to check for. Divisions that are majority made up of battalions in that type count. Optional, counts all divisions by default.
	unit = <string>
	The exact battalion to check for. Divisions that are majority made up of that battalions count. Optional, counts all divisions by default.
	state = <scope> / <variable>
	The state to check in.
		divisions_in_state = {
		type = armor
		size > 10
		state = 49
		}
			Checks if the specified state contains the specified amount of divisions.	Battalions and their types are defined within /Hearts of Iron IV/common/units/*.txt files. Must use either > or < operators for size.	1.0

army_manpower_in_state	amount <> <float>
	The amount to check for.
	type = <string>
	The type to check for. Optional.
	state = <scope> / <variable>
	The state to check in.
		army_manpower_in_state = {
		type = support
		amount > 10000
		state = 49
		}
			Checks if the specified state contains the specified amount of army manpower within the state.	Battalion types are defined within /Hearts of Iron IV/common/units /*.txt files. Must use either > or < operators for size.	1.6

divisions_in_border_state	size = <float>
	The amount to check for.
	type = <string>
	The battalion type to check for. Divisions that are majority made up of battalions in that type count. Optional, counts all divisions by default.
	state = <scope> / <variable>
	The state to check in.
	border_state = <scope> / <variable>
	The border state to check in.
		divisions_in_border_state = {
		type = infantry
		size > 10
		state = 49
		border_state = var:state
		}
			Checks if the border provinces between the specified state and border state contain the specified amount of divisions.	Battalion types are defined within /Hearts of Iron IV/common/units /*.txt files. Must use either > or < operators for size.	1.5

num_divisions_in_states	count = <int>
	The amount to check for.
	states = { <int> <...> <int> }
	The states to check in.
	types = { <string> <...> <string> }
	The battalion types to check for. Divisions that are majority made up of battalions in that type count. Optional, counts all divisions by default.
	exclude = { <string> <...> <string> }
	The sub-units to exclude from the search. Divisions that are majority made up of specified battalions are excluded. Optional, excludes no divisions by default.
		num_divisions_in_states = {
		count > 24
		states = { 550 559 271 }
		exclude = { irregular_infantry }
		}
			Checks if the specified states contain enough divisions of the specified types.	Battalions and their types are defined within /Hearts of Iron IV/common/units/*.txt files. Can use either =, >, or < operators for count. The tooltip does not specify the states the check runs for nor the filtered types.	1.12

num_battalions_in_states	count = <int>
	The amount to check for.
	states = { <int> <...> <int> }
	The states to check in.
	types = { <string> <...> <string> }
	The battalion types to check for.
	exclude = { <string> <...> <string> }
	The sub-units to exclude from the search.
		num_battalions_in_states = {
		count > 24
		states = { 550 559 271 }
		exclude = { irregular_infantry }
		}
			Checks if the specified states contain enough battalions (or sub-units) of the specified types.	Divisions and their types are defined within /Hearts of Iron IV/common/units/*.txt files. Can use either =, >, or < operators for count. The tooltip does not specify the states the check runs for nor the filtered types.	1.12

ships_in_state_ports	size = <float>
	The amount to check for.
	type = <string>
	The type to check for. Optional.
	state = <scope> /
		ships_in_state_ports = {
		type = capital_ship
		size > 10
		state = 49
		}
			Checks if the specified state contains the specified amount of ships, or of ships of the	Ship types are defined within /Hearts of Iron IV/common/units/*.txt files. Must use either > or < operators for size.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
	<variable> The state to check in.		specified type.		

num_planes_stationed_in_regions	value = <float>
	The amount to check for.
	regions = { <id> <...> <id> }
	The regions to check in.
		num_planes_stationed_in_regions = {
		value > 10
		regions = { 123 321 }
		}
			Checks if the current scope has the specified number of aircraft stationed within strategic regions.	Must use either =, >, or < operators for value.	1.12

has_volunteers_amount_from	tag = <scope>
	The country to check for.
	count = <int>
	The amount to check for.
		has_volunteers_amount_from = {
		tag = GER
		count > 10
		}
			Checks if the current scope has recieved volunteers from the specified country of the specified amounts.	Must use either > or < operators for count.	1.0

convoy_threat	<float> The threat to compate with.	convoy_threat > 0.5	Checks how much the convoys are threatened.	Must use either > or < operators for count. Threat is always between 0 and 1.	1.6

has_mined	target = <tag>
	The country the coast of which is mined.
	value <> <int>
	The amount of mines to compare with.
		has_mined = {
		target = POL
		value > 1000
		}
			Checks if the current scope has X mines on the coast of the specified country.	Must use either > or < operators for value.	1.6

has_mines	region = <ID>
	The strategic region that contains the mines.
	amount = <int>
	The amount of mines to compare with.
		has_mined = {
		target = POL
		amount = 1000
		}
			Checks if the current scope has at least X mines within the specified strategic region.		1.6

mine_threat	<float> The threat to compate with.	mine_threat < 0.6	Checks how dangerous enemy mines are.	Must use either > or < operators for count. Threat is always between 0 and 1.	1.6

Equipment[编辑 | 编辑源代码**]**

Equipment-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added

stockpile_ratio	archetype = <string>
	The equipment archetype to check for.
	ratio = <float>
	The ratio of equipment to check for.
		stockpile_ratio = {
		archetype = infantry_equipment
		ratio > 0.5
		}
			Checks if the current scope has stockpiled the specified equipment to the specified ratio against fielded equipment of the same type.	Must use either > or < operators for ratio.
				For the convoy equipment which is not fielded as other equipments, ratio shall be not a percentage but a direct amount (for instance 256 convoys)
					1.5

has_equipment	<equipment> = <int> / <variable>
	The equipment to check for, and the amount to check for.
		has_equipment = {
		infantry_equipment_1 > 10
		}
			Checks if the current scope has the specified equipment to the specified amount.	Must use either > or < operators.	1.0

has_any_license	<bool> Boolean.	has_any_license = yes	Checks if the current scope has any licenses from other countries.		1.0
is_licensing_any_to	<scope> The country to check for.	is_licensing_any_to = GER	Checks if the current scope is licensing to the specified scope.		1.0

is_licensing_to	target = <scope>
	The country to check for.
	archetype = <string>
	The equipment archetype to check for. Optional.
	Equipment scope
	type = <string>
	The equipment to check for. Optional.
	version = <int>
	The variant id of the equipment. Optional.
		is_licensing_to = {
		target = GER
		archetype = infantry_equipment
		}
		is_licensing_to = {
		target = GER
		equipment = {
		type = light_tank_equipment
		version = 1
		}
		}
			Checks if the current scope is licensing the specified equipment to the specified country.		1.0

has_license	from = <scope>
	The country to check for.
	archetype = <string>
	The equipment archetype to check for. Optional.
	Equipment scope
	type = <string>
	The equipment to check for. Optional.
		has_license = {
		from = GER
		archetype = infantry_equipment
		}
		has_license = {
		from = GER
		equipment = {
		type = light_tank_equipment
			Checks if the current scope has a license for the specified equipment from the specified country.		1.0

Name	Parameters	Examples	Description	Notes	Version Added
	version = <int> The variant id of the equipment. Optional.	version = 1 } }			
fuel_ratio	<float> The ratio to check with.	fuel_ratio > 0.4	Checks the fuel ratio of the country.	Must use either < or > operators.	1.6
has_fuel	<int> The amount to compare with.	has_fuel > 400	Checks the fuel amount of the country.	Must use either < or > operators.	1.6
has_design_based_on	<archetype> The equipment archetype.	has_design_based_on = light_tank_chassis	Checks if the country has a builtable non-obsolete design based on the specified equipment archetype.	Equipment archetypes can be seen in /Hearts of Iron IV/common/units /equipment/*.	1.11

Intelligence[编辑 | 编辑源代码]

Intelligence-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added

estimated_intel_max_piercing	tag = <scope>
	The country to check for.
	value = <float>
	The amount to check for.
		estimated_intel_max_piercing = {
		tag = GER
		value > 0.5
		}
			Checks if the specified scope has the specified amount of piercing based on the current scope's intel.	Must use either > or < operators for value.	1.0

estimated_intel_max_armor	tag = <scope>
	The country to check for.
	value = <float>
	The amount to check for.
		estimated_intel_max_armor = {
		tag = GER
		value > 0.5
		}
			Checks if the specified scope has the specified amount of armor based on the current scope's intel.	Must use either > or < operators for value.	1.0

compare_intel_with	target = <tag>
	The target to compare with.
	civilian_intel <>= <float>
	Comparison of civilian intel.
	army_intel <>= <float>
	Comparison of army intel.
	navy_intel <>= <float>
	Comparison of navy intel.
	airforce_intel <>= <float>
	Comparison of airforce intel.
		compare_intel_with = {
		target = POL
		civilian_intel > 0.5
		army_intel = 0
		navy_intel < 0
		}
			Compares intel between 2 countries.	Can use < (in which case the current country has x less intel), >, and = (in which case it must be equal).	1.9

intel_level_over	target = <tag>
	The target to compare with.
	civilian_intel <>= <float>
	Comparison of civilian intel.
	army_intel <>= <float>
	Comparison of army intel.
	navy_intel <>= <float>
	Comparison of navy intel.
	airforce_intel <>= <float>
	Comparison of airforce intel.
		intel_level_over = {
		target = POL
		civilian_intel > 0.5
		army_intel = 0
		navy_intel < 0
		}
			Checks the intel level from the current country over a specified country.	Can use < (in which case the current country has x less intel), >, and = (in which case it must be equal).	1.9

has_intelligence_agency	<boolean> The intelligence agency to check.	has_intelligence_agency = yes	Checks if the current scope has an intelligence agency.		1.9

network_national_coverage	target = <tag>
	The country which is checked.
	value <> <float>
	The value of network.
		network_national_coverage = {
		target = POL
		value < 70
		}
			Checks network national coverage over a specific country.	Must use < or > for value.	

network_strength	target = <tag>
	The country which is checked.
	state = <id>
	The state which is checked.
	value <> <float>
	The strength of network.
		network_strength = {
		target = POL
		value < 70
		}
			Checks network national coverage over a specific country.	Must use < or > for value. Can use either or both of target and state.	1.9

has_done_agency_upgrade	<string> The agency upgrade to check.	has_done_agency_upgrade = upgrade_army_department	Checks if the current scope has the specified agency upgrade (to its highest level).		1.9
agency_upgrade_number	<int> The amount of agency upgrades to check for.	agency_upgrade_number > 4	Checks the number of upgrades done in the current scope's intelligence agency.	Must use either > or < operators.	1.9

Name	Parameters	Examples	Description	Notes	Version Added

decryption_progress	target = <tag>
	The country to compare with.
	value <> <float>
	The value to compare.
		decryption_progress = {
		target = POL
		value < 0.5
		}
			Checks the decryption progress towards a country.	Must use either > or < operators for value.	1.9

has_captured_operative	<tag>/<bool> Country whose operative was captured/Whether an operative was captured.	has_captured_operative = POL has_captured_operative = yes	Checks if the current scope has captured an operative.		1.9

has_finished_collecting_for_operation	target = <tag>
	Country towards whom the operation is targeted.
	operation = <token>
	The operation which current scope is planning against the target.
		has_finished_collecting_for_operation = {
		target = POL
		operation = operation_infiltrate_armed_forces_navy
		}
			Checks if the current scope has finished collecting resources for an operation.		1.9

is_preparing_operation	target = <tag>
	Country towards whom the operation is targeted.
	operation = <token>
	The operation which current scope is planning against the target. Optional.
		is_preparing_operation = {
		target = POL
		operation = operation_infiltrate_armed_forces_navy
		}
			Checks if the current scope is preparing an operation against the specified country.		1.9

is_running_operation	target = <tag>
	Country towards whom the operation is targeted.
	operation = <token>
	The operation which current scope is planning against the target. Optional.
		is_running_operation = {
		target = POL
		operation = operation_infiltrate_armed_forces_navy
		}
			Checks if the current scope is running an operation against the specified country.		1.9

num_finished_operations	target = <tag>
	Country towards whom the operation is targeted.
	operation = <token>
	The operation which current scope is planning against the target. Optional.
		num_finished_operations = {
		target = POL
		operation = operation_infiltrate_armed_forces_navy
		}
			Checks how many finished operations the current scope had against the specified country.		1.9

has_operation_token	tag = <tag>
	Country towards whom the operation is targeted.
	token = <token>
	The operation token.
		has_operation_token = {
		tag = POL
		token = token_name
		}
			Checks if the current scope has an operation token against an another country.		1.9

is_active_decryption_bonuses_enabled	<tag> The country towards which the bonus is enabled.	is_active_decryption_bonuses_enabled = POL	Checks if the current scope has any decryption bonuses towards the specified country.		1.9
is_cryptology_department_active	<bool> Boolean.	is_cryptology_department_active = yes	Checks if the current scope has a cryptology department active.		1.9
is_decrypting	<tag> The country which is decrypted.	is_decrypting = POL	Checks if the current scope is decrypting a certain country.		1.9
is_fully_decrypted	<tag> The country which is decrypted.	is_fully_decrypted = POL	Checks if the current scope has fully decrypted a certain country.		1.9
num_fake_intel_divisions	<int> Amount of divisions.	num_fake_intel_divisions > 10	Checks the amount of fake intel divisions.	Must use either < or >.	1.9
num_free_operative_slots	<int> Amount of slots.	num_free_operative_slots > 2	Checks the amount of free operative slots.	Must use either < or >.	1.9
num_operative_slots	<int> Amount of slots.	num_operative_slots > 2	Checks the amount of operative slots.	Must use either < or >.	1.9
num_of_operatives	<int> Amount of operatives.	num_of_operatives > 2	Checks the amount of operatives.	Must use either < or >.	1.9

AI[编辑 | 编辑源代码**]**

AI-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
ai_irrationality	<int> The amount to check for.	ai_irrationality > 10	Checks if the current scope AI has the specified irrationality.	Must use either > or < operators.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
ai_liberate_desire	target = <scope> The country to check for. count = <float> The amount to check for.	ai_liberate_desire = { target = GER count > 1 }	Checks if the current scope AI has the specified liberation desire towards the specified country.	Must use either > or < operators for count.	1.0
ai_has_role_division	<string> The role to check for.	ai_has_role_division = infantry	Checks if the current scope AI has a division with the specified role.	Roles are defined in /Hearts of Iron IV/common /ai_templates/*.txt	1.0
ai_has_role_template	<string> The role to check for.	ai_has_role_template = armor	Checks if the current scope AI has a division template with the specified role.	Roles are defined in /Hearts of Iron IV/common /ai_templates/*.txt	1.0
ai_wants_divisions	<int> The amount to check for.	ai_wants_divisions > 10	Checks if the current scope AI desires the specified amount of divisions.	Must use either > or < operators.	1.0
has_template_ai_majority_unit	<string> The unit to check for.	has_template_ai_majority_unit = infantry	Checks if the current scope AI has a division template mostly made up of the specified unit.		1.0

Characters[编辑 | 编辑源代码]

These are character-related triggers, not character-scoped triggers.

Character-related country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
can_be_country_leader	<character> The character to check.	can_be_country_leader = POL_character_test	Checks if the specified character has a country leader role, active or not, and can utilise it in this country.		1.11
has_character	<string> The character to check.	has_character = my_character	Checks if the current scope has the specified character recruited. The character does NOT need to be in power.		1.11
has_country_leader	ruling_only = <bool>(default = yes) Limit check to ruling only. character = <character_token> (recommended criteria) The character to check for. Optional. name = <string> The name to check for. Optional. id = <int> The id to check for. Optional.	has_country_leader = { id = 10 } has_country_leader = { character = SPR_niceto_alcala_zamora ruling_only = yes } has_country_leader = { name = "John Smith" ruling_only = yes }	Checks if the current scope has the specified country leader.		1.3
has_country_leader_ideology	<ideology> Checks the ideology of the active country leader	has_country_leader_ideology = nazism	Checks if the current scope's active country leader has the specified ideology.		1.11
has_country_leader_with_trait	<string> The trait to check.	has_country_leader_with_trait = champion_of_peace_1	Checks if the leader of the country has a specific trait.		1.6
is_female	<bool> Boolean.	is_female = yes	Checks if the current country leader is female.		1.9
has_unit_leader	<int> The id to check for.	has_unit_leader = 1	Checks if the current scope has a unit leader with the specified id.		1.0

Peace conferences[编辑 | 编辑源代码]

These are not exactly peace conference-related triggers, but **those that can only be used within peace conferences**.

Peace conference-only country-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
pc_is_winner	<bool> Boolean.	pc_is_winner = yes	Checks if the current scope is a winner within the peace conference.		1.12
pc_is_on_winning_side	<bool> Boolean.	pc_is_on_winning_side = yes	Checks if the current scope is on the winning side within the peace conference.		1.12
pc_is_loser	<bool> Boolean.	pc_is_loser = yes	Checks if the current scope is a loser within the peace conference.		1.12
pc_is_untouched_loser	<bool> Boolean.	pc_is_untouched_loser = yes	Checks if the current scope is an untouched loser within the peace conference.		1.12
pc_is_on_same_side_as	<scope> Country to check for.	pc_is_on_same_side_as = BHR	Checks if the current scope is on the same side of the peace conference as the specified country.		1.12
pc_is_liberated	<bool> Boolean.	pc_is_liberated = yes	Checks if the current scope has been liberated within the peace conference.		1.12

Name	Parameters	Examples	Description	Notes	Version Added
pc_is_liberated_by	<scope> Country to check for.	pc_is_liberated_by = BHR	Checks if the current scope has been liberated within the peace conference by the specified country.		1.12
pc_is_puppeted	<bool> Boolean.	pc_is_puppeted = yes	Checks if the current scope has been puppeted within the peace conference.		1.12
pc_is_puppeted_by	<scope> Country to check for.	pc_is_puppeted_by = BHR	Checks if the current scope has been puppeted within the peace conference by the specified country.		1.12
pc_is_forced_government	<bool> Boolean.	pc_is_forced_government = yes	Checks if the current scope has had an enforced government change within the peace conference.		1.12
pc_is_forced_government_by	<scope> Country to check for.	pc_is_forced_government_by = BHR	Checks if the current scope has had an enforced government change within the peace conference demanded by the specified country.		1.12
pc_is_forced_government_to	<ideology group> Ideology group to check for.	pc_is_forced_government_to = democratic	Checks if the current scope has had an enforced government change to the specified ideology group.		1.12
pc_total_score	<decimal> Scope to check for.	pc_total_score > 2400	Checks if the current scope has the specified amount in total score within the peace conference.	Can only be used for the winning countries.	1.12
pc_current_score	<decimal> Scope to check for.	pc_current_score > 100	Checks if the current scope has the specified amount in current score within the peace conference.	Can only be used for the winning countries.	1.12

State scope[编辑 | 编辑源代码]

Can be used in **state** scope.

General[编辑 | 编辑源代码]

General state-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
state	<scope> / <variable> The state to check for.	state = 10 state = var:state	Checks if the current scope is the specified state.		1.0
region	<int> The strategic region id to check for.	region = 10	Checks if the current scope is a state in the specified strategic region.		1.0
<building>	<building> = <int> The amount of the specified building to to check for.	arms_factory > 10	Checks if the current scope has the specified amount of the specified building.	Must use either > or < operators.	1.0

free_building_slots	building = <string>
	The building to check for.
	size = <int>
	The amount to check for.
	include_locked = <bool>
	Whether to include locked slots.
		free_building_slots = {
		building = arms_factory
		size > 10
		include_locked = yes
		}
			Checks if the current scope has available slots for the specified amount of buildings.	Must use either > or < operators for size.	1.0

non_damaged_building_level	building = <string>
	The building to check for.
	level = <int>
	The amount to check for.
		non_damaged_building_level = {
		building = arms_factory
		level > 4
		}
			Checks if the current scope has the specified amount of the specified buildings that are undamaged.	Must use either > or < operators for level.	1.9

any_province_building_level	building = <string>
	The building to check for.
	limit = <int>
	The amount to check for.
	Province scope
	id = <int>
	The province to check for.
	limit_to_border = <bool>
	Whether to limit check to border provinces.
		any_province_building_level = {
		province = {
		id = 445
		id = 494
		limit_to_border = yes
		}
		building = bunker
		level < 5
		}
			Checks if the current scope has the specified provincal building at the specified amount in the specified provinces.	Must use either > or < operators for level.	1.0

has_state_flag	<string> The flag to check for.	has_state_flag = my_flag	Checks if the current scope has the specified flag.		1.0

Name	Parameters	Examples	Description	Notes	Version Added

has_state_flag	flag = <string>
	The flag to check.
	value = <int>
	The flag value to check for. Optional.
	date = <date>
	The flag creation date to check for. Optional.
	days = <int>
	The duration the flag existed for. Optional.
		has_state_flag = {
		flag = my_flag
		days > 30
		date > 1936.6.1
		value > 0
		}
			Compares the specified flag's last set date, days since last set, and/or value.	If not set, the value comparison is >0. value is limited between -32768 and 32767.	1.0

state_population	<float> The amount to check for.	state_population > 10000	Checks if the current scope has the specified state population.	Must use either > or < operators.	1.0
state_population_k	<float> The amount to check for.	state_population_k > 10	Checks if the current scope has the specified state population in thousands.	Must use either > or < operators.	1.0
is_capital	<bool> Boolean.	is_capital = yes	Checks if the current scope is a capital.		1.5
is_controlled_by	<scope> / <variable> The country to check for.	is_controlled_by = GER	Checks if the current scope is controlled by the specified country.		1.0
is_fully_controlled_by	<scope> / <variable> The country to check for.	is_fully_controlled_by = GER	Checks if the current scope is fully controlled by the specified country.		1.5
is_owned_by	<scope> / <variable> The country to check for.	is_owned_by = GER	Checks if the current scope is owned by the specified country.		1.0
is_claimed_by	<scope> / <variable> The country to check for.	is_claimed_by = GER	Checks if the current scope is claimed by the specified country.		1.0
is_core_of	<scope> / <variable> The country to check for.	is_core_of = GER	Checks if the current scope is a core of the specified country.		1.0
is_owned_and_controlled_by	<scope> / <variable> The country to check for.	is_owned_and_controlled_by = GER	Checks if the current scope is owned and controlled by the specified country.		1.0
is_demilitarized_zone	<bool> Boolean.	is_demilitarized_zone = yes	Checks if the current scope is a demilitarized zone.		1.0
is_border_conflict	<bool> Boolean.	is_border_conflict = yes	Checks if the current scope is part of a border war.		1.0
is_in_home_area	<bool> Boolean.	is_in_home_area = yes	Checks if the current scope is connected to the capital state over land. The scope needs to be owned as well for the statement for it to be true.		1.0
is_coastal	<bool> Boolean.	is_coastal = yes	Checks if the current scope is a coastal state.		1.0
is_island_state	<bool> Boolean.	is_island_state = yes	Checks if the current scope is a coastal state with no adjacent land states.		1.0
is_on_continent	<string> The continent to check for.	is_on_continent = europe	Checks if the current scope is on the specified continent.	Continents are found in /Hearts of Iron IV/map /continent.txt.	1.0
impassable	<bool> Boolean.	impassable = yes	Checks if the current scope is impassable.		1.9.1
has_state_category	<string> The category to check for.	has_state_category = rural	Checks if the current scope has the specified category.	State categories are found in /Hearts of Iron IV/common /state_category/*.txt.	1.0
state_strategic_value	<int> The amount to check for.	state_strategic_value > 10	Checks if the current scope has the specified strategic value.	Must use either > or < operators.	1.5
state_and_terrain_strategic_value	<int> The amount to check for.	state_and_terrain_strategic_value > 10	Checks if the current scope has the specified state and terrain strategic value.	Must use either > or < operators.	1.5

num_owned_neighbour_states	owner = <scope>
	The country to check for.
	count = <int>
	The amount to check for.
		num_owned_neighbour_states = {
		owner = GER
		count > 2
		}
			Checks if the current scope has the specified amount of neighbor states belonging to the specified country.	Must use either > or < operators for count.	1.0

distance_to	distance = <float>
	The distance to check for.
	target = <scope>
	The state to compare against.
		distance_to = {
		value > 1000
		target = 49
		}
			Checks if the current scope is at the specified distance from the specified state.	Must use either > or < operators for distance.	1.0

ships_in_area	area = <int> The strategic region to check for.	ships_in_area = { area = 104 size > 14 }	Checks if the current scope has the specified amount	Must use either > or < operators for count.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
	size = <int> The amount to check for.		of ships in the specified strategic region.		

has_resources_amount	resource = <string>
	The resource to check for.
	amount = <int>
	The amount to check for.
	delivered = yes
	If specified, checks the amount after the modifiers are applied rather than the base resource value.
		has_resources_amount = {
		resource = oil
		amount > 10
		delivered = yes
		}
			Checks if the current scope has the specified amount of the specified resource.	Must use either > or < operators for amount.	1.3

has_resources_rights	state = <scope>
	The state to check in. Mandatory if used in country scope.
	receiver = <int>
	The receiver of the resource rights. Mandatory if used in state scope.
	resources = {
	<resource> <...>
	<resource> }
	Resources to check for. Optional, defaults to any if unset.
		has_resources_rights = { # Used in country scope
		state = 123
		resources = { oil steel }
		}
		has_resources_rights = { # Used in state scope
		receiver = POL
		}
			Checks if there are any resource rights with the specified parameters.	Can be used in either state or country scope.	1.12

has_resources_in_country	resource = <string>
	The resource to check for.
	amount = <int>
	The amount to check for.
		has_resources_in_country = {
		resource = oil
		amount > 10
		}
			Checks if the current scope has the specified amount of the specified resource in reserve.	Must use either > or < operators for amount. 'In reserve' means that it's not spent on equipment production or exports.	1.12

days_since_last_strategic_bombing	<int> The amount to compare with.	days_since_last_strategic_bombing < 10	Checks how many days have passed since the last strategic bombing of the state.	Must use either > or < operators.	1.6

has_railway_connection	<scope> / <variable>
	The states to check.
	<id>
	The provinces to check. Optional.
		has_railway_connection = {
		start_state = 10
		target_state = 90
		}
		has_railway_connection = {
		start_province = 402
		target_province = 9400
		}
			Returns true if the states are connected by a railway. Can also check provinces.		1.11

can_build_railway	<scope> / <variable>
	The states to check.
	<id>
	The provinces to check. Optional.
		can_build_railway = {
		start_state = 10
		target_state = 90
		}
		can_build_railway = {
		start_province = 402
		target_province = 9400
		}
			Returns true if a railway can be built between states. Can also check for provinces.		1.11

has_railway_level	<scope> / <variable>
	The states to check.
	<int>
	Railway level.
		has_railway_level = {
		state = 114
		level = 5
		}
			Checks if the specified state has a railway line in it with the specified level.		1.11

pc_does_state_stack_demilitarized	<bool> Boolean.	pc_does_state_stack_demilitarized = yes	Checks if the current scope was demilitarised during a current or previously-ended peace conference.		1.12
pc_does_state_stack_dismantled	<bool> Boolean.	pc_does_state_stack_dismantled = yes	Checks if the current scope was dismantled during a current or previously-ended peace conference.		1.12
pc_is_state_claimed_by	<scope> Country to check for.	pc_is_state_claimed_by = BHR	Checks if the current scope was claimed by the specified country during the peace conference.	**Can only be used within peace conferences.**	1.12

Resistance and Compliance[编辑 | 编辑源代码]

Resistance-related state-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
compliance	<int> The amount to compare with.	compliance > 50	Compares the compliance value of the current scope with the given value.	Must use either > or < operators.	1.9

Name	Parameters	Examples	Description	Notes	Version Added
compliance_speed	<int> The amount to compare with.	compliance_speed > 50	Compares the compliance speed of the current scope with the given value.	Must use either > or < operators.	1.9
has_active_resistance	<bool> Boolean.	has_active_resistance = yes	Checks if the current scope has non-zero resistance.		1.9
has_resistance	<bool> Boolean.	has_resistance = yes	Checks if the current scope has resistance.		1.9
resistance	<int> The amount to compare with.	resistance > 50	Compares the resistance value of the current scope with the given value.	Must use either > or < operators.	1.9
resistance_speed	<int> The amount to compare with.	resistance_speed > 50	Compares the resistance speed of the current scope with the given value.	Must use either > or < operators.	1.9
resistance_target	<int> The amount to compare with.	resistance_target > 50	Compares the target resistance value of the current scope with the given value.	Must use either > or < operators.	1.9
has_occupation_modifier	<token> The occupation modifier to check.	has_occupation_modifier = modifier_name	Checks if the current scope has an occupation modifier, changing resistance/compliance.		1.9
occupation_law	<token> The occupation law to check.	occupation_law = law_name	Checks if the current scope has an occupation law.		1.9
occupied_country_tag	<tag> The occupation tag to check.	occupied_country_tag = POL	Checks which country creates resistance.		1.9

Character scope[编辑 | 编辑源代码]

Can be used in **Character** scope.

General[编辑 | 编辑源代码]

General character-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
is_character	<scope> Character ID.	is_character = POL_test_character	Checks if the current character's token matches up with the specified one.		1.11
can_be_country_leader	<bool> Boolean.	can_be_country_leader = yes	Checks if the character in the current scope has a country leader role, active or non-active.		1.11
is_country_leader	<bool> Boolean.	is_country_leader = yes	Checks if the character in the current scope is the active country leader.		1.11
is_unit_leader	<bool> Boolean.	is_unit_leader = yes	Checks if the character in the current scope has an active unit leader (Army/Navy leader) role.		1.11
is_advisor	<bool> Boolean.	is_advisor = yes	Checks if the character in the current scope has an advisor role (includes advisors/theorists/high command).		1.11
is_air_chief	<bool> Boolean.	is_air_chief = yes	Checks if the character in the current scope is selected as an air chief.	Prior to 1.12, checked if the character had a role within the slot, regardless of being selected.	1.11
is_army_chief	<bool> Boolean.	is_army_chief = yes	Checks if the character in the current scope is selected as an army chief.	Prior to 1.12, checked if the character had a role within the slot, regardless of being selected.	1.11
is_army_leader	<bool> Boolean.	is_army_leader = yes	Checks if the character in the current scope has an army leader (General/Field Marshal) role.		1.11
is_navy_chief	<bool> Boolean.	is_navy_chief = yes	Checks if the character in the current scope is selected as a navy chief.	Prior to 1.12, checked if the character had a role within the slot, regardless of being selected.	1.11
is_navy_leader	<bool> Boolean.	is_navy_leader = yes	Checks if the character in the current scope has an navy leader (Admiral) role.		1.11
is_high_command	<bool> Boolean.	is_high_command = yes	Checks if the character in the current scope is selected as high command.	Prior to 1.12, checked if the character had a role within the slot, regardless of being selected.	1.11
is_corps_commander	<bool> Boolean.	is_corps_commander = yes	Checks if the character in the current scope is a corps commander.		1.11
is_operative	<bool> Boolean.	is_operative = yes	Checks if the character in the current scope is an operative.		1.11
is_political_advisor	<bool> Boolean.	is_political_advisor = yes	Checks if the character in the current scope is selected as a political advisor.	Prior to 1.12, checked if the character had a role within the slot, regardless of being selected.	1.11

Name	Parameters	Examples	Description	Notes	Version Added
is_theorist	`<bool>` Boolean.	is_theorist = yes	Checks if the character in the current scope is selected as a theorist.	Prior to 1.12, checked if the character had a role within the slot, regardless of being selected.	1.11
is_character_slot	`<string>` The advisor slot to check.	is_character_slot = political_advisor	Checks if the character in the current scope has a role within the specified character slot	Character slots are defined within /Hearts of Iron IV/common/idea_tags /*.txt.	1.11
has_air_ledger	`<bool>` Boolean.	has_air_ledger = yes	Checks if the character in the current scope has an air ledger.		1.11
has_army_ledger	`<bool>` Boolean.	has_army_ledger = yes	Checks if the character in the current scope has an army ledger.		1.11
has_navy_ledger	`<bool>` Boolean.	has_navy_ledger = yes	Checks if the character in the current scope has an navy ledger.		1.11
has_character_flag	`<string>` The flag to check for.	has_character_flag = my_flag	Checks if the current scope has the specified flag.		1.11

has_character_flag	`flag = <string>`
	The flag to check.
	`value = <int>`
	The flag value to check for. Optional.
	`date = <date>`
	The flag creation date to check for. Optional.
	`days = <int>`
	The duration the flag existed for. Optional.
		has_character_flag = {
		flag = my_flag
		days > 30
		date > 1936.6.1
		value > 0
		}
			Compares the specified flag's last set date, days since last set, and/or value.	If not set, the value comparison is `>0`. `value` is limited between -32768 and 32767.	1.11

has_trait	`<trait>` The trait to check for.	has_trait = really_good_boss	Checks if the current scope has the specified trait.		1.5
has_id	`<int>` The id to check for.	has_id = 1	Checks if the current character has the specificed ID.		1.5
not_already_hired_except_as	`<slot>` The slot to check in.	not_already_hired_except_as = political_advisor	Checks if the current character is not hired, with the exception of the specified slot.		1.5

Country leaders[编辑 | 编辑源代码]

These triggers are to be used specifically for country leaders.

Country leader-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_ideology	`<ideology>` The sub-ideology to check for.	has_ideology = liberalism	Checks if the current character has the specificed sub-ideology assigned.		1.11
has_ideology_group	`<ideology>` The ideology to check for.	has_ideology_group = democratic	Checks if the current character has the specificed ideology assigned.		1.11

Unit leaders[编辑 | 编辑源代码]

These triggers are to be used specifically for unit leaders, i.e. generals and admirals.

Unit leader-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_unit_leader_flag	`<string>` The flag to check for.	has_unit_leader_flag = my_flag	Checks if the current scope has the specified flag.	Deprecated. Use has_character_flag instead.	1.5

has_unit_leader_flag	`flag = <string>`
	The flag to check.
	`value = <int>`
	The flag value to check for. Optional.
	`date = <date>`
	The flag creation date to check for. Optional.
	`days = <int>`
	The duration the flag existed for. Optional.
		has_unit_leader_flag = {
		flag = my_flag
		days > 30
		date > 1936.6.1
		value > 0
		}
			Compares the specified flag's last set date, days since last set, and/or value.	Deprecated. Use has_character_flag instead. If not set, the value comparison is `>0`. `value` is limited between -32768 and 32767.	1.5

Name	Parameters	Examples	Description	Notes	Version Added
is_leading_army	<bool> Boolean.	is_leading_army = yes	Checks if the current scope is leading a single army.		1.5
is_leading_army_group	<bool> Boolean.	is_leading_army_group = yes	Checks if the current scope is leading an army group.		1.5
is_leading_volunteer_group	<tag> Country tag.	is_leading_volunteer_group = POL	Checks if the current scope is leading a volunteer army within the specified country.	If the target country is in a civil war, this will only be valid for one side.	1.11
is_leading_volunteer_group_with_original_country	<tag> Country tag.	is_leading_volunteer_group_with_original_country = POL	Checks if the current scope is leading a volunteer army within a country of the specified original tag.	If the target country is in a civil war, this will only be valid for each side.	1.11
is_field_marshal	<bool> Boolean.	is_field_marshal = yes	Checks if the current scope is a Field Marshal.		1.5
is_assigned	<bool> Boolean.	is_assigned = yes	Checks if the current scope is an assigned unit leader.		1.5
can_select_trait	<string> The trait to check for.	can_select_trait = offensive_doctrine	Checks if the current scope can select the specified trait.		1.5
has_ability	<string> The ability to check for.	has_ability = glider_planes	Checks if the current scope has the specified unit leader ability.		1.5
skill	<int> The amount to check for.	skill > 1	Checks if the current scope has a Skill above the specified amount.	Must use either > or < operators.	1.5
skill_advantage	<int> The amount to check for.	skill_advantage > 1	Checks if the current scope has a Skill advantage above the specified amount in against an enemy unit leader whilst in combat.	Must use either > or < operators.	1.5
planning_skill_level	<int> The amount to check for.	planning_skill_level > 1	Checks if the current scope has a Planning skill above the specified amount.	Must use either > or < operators.	1.5
logistics_skill_level	<int> The amount to check for.	logistics_skill_level > 1	Checks if the current scope has a Logistics skill above the specified amount.	Must use either > or < operators.	1.5
defense_skill_level	<int> The amount to check for.	defense_skill_level > 1	Checks if the current scope has a Defense skill above the specified amount.	Must use either > or < operators.	1.5
attack_skill_level	<int> The amount to check for.	attack_skill_level > 1	Checks if the current scope has a Attack skill above the specified amount.	Must use either > or < operators.	1.5
average_stats	<int> The amount to check for.	average_stats > 5	Checks if the current scope has an average skill	Must use either > or < operators.	1.5

Name	Parameters	Examples	Description	Notes	Version Added
			above the specified amount.		
is_border_war	<bool> Boolean.	is_border_war = yes	Checks if the current socpe is in a border war.		1.5
num_units	<int> The amount to check for.	num_units > 5	Checks if the current scope is commanding the specified amount of divisions.	Must use either > or < operators.	1.5
is_exiled_leader	<bool> Boolean.	is_exiled_leader = yes	Checks if the current scope is a general from an exiled country.		1.6
is_exiled_leader_from	<tag> Country.	is_exiled_leader_from = POL	Checks if the current scope is a general from the specified exiled country.		1.6
is_female	<bool> Boolean.	is_female = yes	Checks if the current scope is female.	Works for aces.	1.9

Operatives[编辑 | 编辑源代码]

These triggers only work for operatives.

Operative-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
has_nationality	<tag> The nationality to check.	has_nationality = POL	Checks if the current operative has the nationality.		1.9
is_operative_captured	<bool> Boolean.	is_operative_captured = yes	Checks if the current scope is captured.		1.9
operative_leader_mission	<token> Mission.	operative_leader_mission = mission_name	Checks if the current scope is on the given mission.		1.9
operative_leader_operation	<token> Operation.	operative_leader_operation = operation_name	Checks if the current scope is on the given operation.		1.9

Combat[编辑 | 编辑源代码]

These triggers are used within the combatant scope, which is checked in abilities, combat tactics, and unit leader traits.

Combatant-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
hardness	<float> The amount to check for.	hardness > 0.5	Checks if the current scope has the specified amount of hardness.	Must use either > or < operators.	1.0
armor	<float> The amount to check for.	armor > 0.5	Checks if the current scope has the specified amount of armor units.	Must use either > or < operators.	1.0
dig_in	<float> The amount to check for.	dig_in > 0.5	Checks if the current scope has the specified amount of Dig In bonus.	Must use either > or < operators.	1.0
min_planning	<float> The amount to check for.	min_planning > 0.5	Checks if the current scope has the specified amount of planning.	Must use either > or < operators.	1.0
fastest_unit	<float> The speed in km/h to check for.	fastest_unit > 12	Checks if the current scope has a unit with the specified speed.	Must use either > or < operators.	1.0
temperature	<float> The temperature in celsius to check for.	temperature > 20	Checks if the current scope is in a province with a temperature above the specified amount.	Must use either > or < operators.	1.0
reserves	<float> The amount to check for.	reserves > 10	Checks if the current scope has the specified amount of reserves waiting.	Must use either > or < operators.	1.0
has_cavalry_ratio	<float> The ratio to check for.	has_cavalry_ratio > 0.5	Checks if the current scope has the specified ratio of cavalry in their division composition.	Must use either > or < operators.	1.0

Name	Parameters	Examples	Description	Notes	Version Added
has_combat_modifier	<string> The modifier to check for.	has_combat_modifier = river_crossing	Checks if the current scope has the specified combat modifier.		1.0
is_fighting_in_terrain	<string> The terrain to check for.	is_fighting_in_terrain = desert	Checks if the current scope is fighting in the specified terrain.		1.0
is_fighting_in_weather	<string> The weather to check for. **OR** { <string> <...> <string> } The weather to check for in an OR statement.	is_fighting_in_weather = sandstorm is_fighting_in_weather = { rain_light rain_heavy }	Checks if the current scope is fighting in the specified weather.		1.0
phase	<bool> Boolean.	phase = yes	Checks if the current scope is in phase.		1.0
recon_advantage	<bool> Boolean.	recon_advantage > 0	Checks if the current scope has x recon advantage.		1.0
night	<bool> Boolean.	night = yes	Checks if the current scope is fighting at night.		1.0
frontage_full	<bool> Boolean.	frontage_full = yes	Checks if the current scope has a full combat width.		1.0
has_flanked_opponent	<bool> Boolean.	has_flanked_opponent = yes	Checks if the current scope has flanked their opponent.		1.0
has_max_planning	<bool> Boolean.	has_max_planning = yes	Checks if the current scope has the maximum planning bonus.		1.0
has_reserves	<bool> Boolean.	has_reserves = yes	Checks if the current scope has any reserves waiting.		1.0
is_amphibious_invasion	<bool> Boolean.	is_amphibious_invasion = yes	Checks if the current scope is performing an amphibious invasion.		1.0
is_attacker	<bool> Boolean.	is_attacker = yes	Checks if the current scope is attacking.		1.0
is_defender	<bool> Boolean.	is_defender = yes	Checks if the current scope is defending.		1.0
is_winning	<bool> Boolean.	is_winning = yes	Checks if the current scope is winning their battle.		1.0
is_fighting_air_units	<bool> Boolean.	is_fighting_air_units = yes	Checks if the current scope is fighting air units.		1.0
less_combat_width_than_opponent	<bool> Boolean.	less_combat_width_than_opponent = yes	Checks if the current scope is fighting with less combat width than their opponent.		1.0
has_carrier_airwings_on_mission	<bool> Boolean.	has_carrier_airwings_on_mission = yes	Checks if the current scope has carrier airwings on a mission.		1.0
has_carrier_airwings_in_own_combat	<bool> Boolean.	has_carrier_airwings_in_own_combat = yes	Checks if the current scope has carrier airwings in their own combat.		1.0

Division scope[编辑 | 编辑源代码]

Can be used in **Division** scope.

Division-scoped triggers:
折叠

Name	Parameters	Examples	Description	Notes	Version Added
division_has_majority_template	<battalion> Battalion to check for.	division_has_majority_template = light_armor	Checks if the current scope is majority made up of the specified battalion.	Battalions are defined within /Hearts of Iron IV/common/units/*.txt files.	1.12
division_has_battalion_in_template	<battalion> Battalion to check for.	division_has_battalion_in_template = light_armor	Checks if the current scope has any battalions of the type in the template.	Battalions are defined within /Hearts of Iron IV/common/units/*.txt files.	1.12
unit_strength	<battalion> Battalion to check for.	unit_strength < 0.3	Checks the current strength of the unit on the scale from 0 to 1.	Must use either the < or > operator.	1.12
unit_organisation	<battalion> Battalion to check for.	unit_organisation < 0.3	Checks the current organisation of the unit on the scale from 0 to 1.	Must use either the < or > operator.	1.12
is_unit_template_reserves	<bool> Boolean.	is_unit_template_reserves = yes	Checks if the current division has the supply priority set to 'Reserves', i.e. the lowest priority.	Must use either the < or > operator.	1.12

Meta triggers[编辑 | 编辑源代码]

Meta triggers are a system added with 1.6 with Man the Guns[4] used in the exact same manner as meta effects: in order to tie a trigger block to a dynamic localisation entry.

This is usually used in conjunction with scripted localisation for non-numerical checks. This also can serve in order to place a variable check where one is not possible. Meta triggers work in any scope that supports triggers, including combat scope.

The following arguments go inside of a scripted trigger:

`text = { ... }` is a trigger block used by the meta trigger. In here, any string that must be made dynamic is marked with the square brackets on each side as `[EXAMPLE]`.

`EXAMPLE = "..."` is a string that will serve as the dynamic replacement for `[EXAMPLE]` within `text = { ... }`. This accepts dynamic localisation to a limited degree, such as getting a variable's value with `[?var_name]` or some namespaces like `[ROOT.GetTag]`.

`debug = yes`, if added, will add the final output in the user directory's /Hearts of Iron IV/logs/game.log file for debugging purposes.

For example, the following can be used with has_equipment to make it depend on a variable's value:

```
meta_trigger = {
  text = {
    has_equipment = { [EQUIPMENT_ARCHETYPE] > 100 }
  }
  EQUIPMENT_ARCHETYPE = "[GetEquipmentArchetype]"
}
```

In this case, `GetEquipmentArchetype` is scripted localisation. An example definition of scripted localisation in this case is as such:

```
defined_text = {
  name = GetEquipmentArchetype
  text = {
    trigger = {
      check_variable = { v = 0 }
    }
    localization_key = artillery_equipment
  }
  text = {
    localization_key = infantry_equipment
  }
}
```

Scripted triggers[编辑 | 编辑源代码]

Scripted triggers serve a similar purpose to functions in that they can be defined in /Hearts of Iron IV/common/scripted_triggers/*.txt and then used elsewhere as a shortened version. Alongside that, the game has certain base game scripted triggers that are checked directly in the game's code determining triggers for functions that cannot be changed.

A scripted trigger is defined simply as

```
scripted_trigger_name = {
    <triggers>
}
```

This example can be used as a trigger in regular code as `scripted_trigger_name = yes` or `scripted_trigger_name = no`.

Following scripted triggers, as of 1.12, are checked directly in the game's code:

diplomacy_scripted_triggers.txt decides on triggers that decide when a diplomatic option is available to send, where ROOT is the country for which the triggers are checked and FROM is the target country towards which the diplomatic option is oriented. In order to find the internal name of a diplomatic function that's not in the file, /Hearts of Iron IV/localisation/diplomacy_l_english.yml. As an example, `DIPLOMACY_WAR_TITLE:0 "Declare war"` means that the 'Declare war' diplomatic action has the ID of DIPLOMACY_WAR.

00_diplo_action_valid_triggers.txt decides when a diplomatic action is visible.

00_resistance_initiate_triggers.txt decides when resistance can be initiated. Alongside the default, global, trigger which applies to every state, state-specific ones can be added using the template of

```
should_initiate_resistance_123_321 = {
    if = {
        limit = {
            tag = ENG
        }
        is_core_of = FRA
    }
    else = {
        should_initiate_resistance = yes
    }
}
```

This example will activate the resistance on states 123 and 321 for England if it's a core of France, otherwise normal rules will be followed. Do note that this will completely overwrite the default triggers on resistance initiation on the specified states.

Useful scripted triggers[编辑 | 编辑源代码]

These scripted triggers are defined in base game and might be useful to implement in the mod to cut down on the amount of code. As scripted triggers, all of these use a boolean value as argument.

Scripted triggers:
折叠

Name	Scope	Example	Description	Notes
can_ROOT_get_wargoal_on_THIS	Country	can_ROOT_get_wargoal_on_THIS = yes	Checks if ROOT can obtain a wargoal on the current scope.	To evaluate as true, the current scope must exist, not share a faction with ROOT, and not be a subject of ROOT.
is_free_or_subject_of_root	Country	is_free_or_subject_of_root = yes	Checks if the current scope is either independent or a subject of ROOT.	
has_same_ideology	Country	has_same_ideology = yes	Checks if the current scope has the same ideology as ROOT.	Needs modifying if there are custom ideologies. Equivalent to `has_government = ROOT` for base game ideologies.
is_enemy_ideology	Country	is_enemy_ideology = yes	Checks if the current scope has an ideology that is considered enemy to ROOT's.	Communism, Democracy, and Fascism are considered enemy to each other.
has_ROOT_at_least_1_div_in_current_state_scope	State	has_ROOT_at_least_1_div_in_current_state_scope = yes	Checks if ROOT has at least one division in the current scope.	
controls_or_subject_of	State	controls_or_subject_of = yes	Checks if the current state is controlled by ROOT or a subject of ROOT.	
is_controlled_by_ROOT_or_ally	State	is_controlled_by_ROOT_or_ally = yes	Checks if the current state is controlled by ROOT, a subject of ROOT, or a country in the same faction as ROOT.	
owns_or_subject_of	State	owns_or_subject_of = yes	Checks if the current scope is owned by ROOT or a subject of ROOT.	

References[编辑 | 编辑源代码]

↑ [Modding] Achievement for mods
Tutorial to write achievements files in your mod

↑ How does any_war_score work?

↑ forum:1356228/#post-26361808

↑ HOI4 Dev Diary - Modding and Traits