Idea modding - Hearts of Iron 4 Wiki

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

Ideas are a static way to apply modifiers to a country, notably including national spirits, laws, designers, officer corps spirits, and hidden ideas. Ideas themselves are defined in /Hearts of Iron IV/common/ideas/*.txt, while idea categories are defined in /Hearts of Iron IV/common/idea_tags/*.txt

National spirit creation[编辑 | 编辑源代码]

Spirits are created in /Hearts of Iron IV/common/ideas/*.txt. An example of a file with empty spirits, without modifiers yet still applicable if added via `add_ideas = idea_name`, is

```
ideas = {
  country = {
    my_idea_1 = {
    }
    my_idea_2 = {
    }
  }
}
```

In this case, `ideas = {}` encompasses each idea in the file, necessary to include due to the engine consraints. Meanwhile, `country = {}` is an idea category, which is why the game would recognise `my_idea_1` and `my_idea_2` as spirits rather than laws or designers, which are also ideas.

Localisation[编辑 | 编辑源代码]

Localisation for ideas is defined in any /Hearts of Iron IV/localisation/english/*_l_english.yml file **encoded in the UTF-8-BOM formatting**, assuming the English language. If an idea does not have a name, the game uses the country's name-list in /Hearts of Iron IV/common/names to create a random name. This, however, does not apply to national spirits, where it uses the idea's ID. An idea can have both a name and a description that appears when hovering over it. Their localisation entries are defined as such, taking ZZZ_example_idea as an example:

```
 ZZZ_example_idea:0 "Idea's name"
 ZZZ_example_idea_desc:0 "Idea's description"
```

Picture[编辑 | 编辑源代码]

折叠General sprite overview

For loading GFX, the game uses the sprite system. Sprites are code definitions that attach a name to an image file, as well as optionally adding additional information, such as animation, the amount of frames, the way that the image will be loaded, and so on. This means **placing an image into the gfx folder isn't enough for it to work**, a sprite has to use that image file as well. Sprites are defined in any /Hearts of Iron IV/interface/*.gfx file (this is separate from gfx/interface/), opened with a text editor. To create a new .gfx file, a text file can be created and renamed to change the extension (on Windows, the Windows Explorer needs to show the extensions, which it doesn't by default). In particular, sprites are defined within a `spriteTypes = { ... }` block, as to separate from fonts and map arrows also defined in that folder, while the simplest sprite with the least mandatory properties is a `spriteType = { ... }`. The simplest sprite definition looks like the following:

```
spriteTypes = {
  spriteType = {
    name = GFX_first_sprite                       # In some cases, beginning with GFX_ is mandatory for it to work.
    texturefile = gfx/interface/folder/filename.dds # The folder and filename don't matter, as long as they are correct
  }                                  # Only the forward slash '/' (can be doubled as '//') can be used to separate folders.
  spriteType = {                           # The image doesn't have to be .dds, as .tga and .png are acceptable.
    name = GFX_second_sprite
    texturefile = gfx/interface/folder2/filename2.dds
    noOfFrames = 2
  }
}
```

In this case, this creates a sprite with the name of `GFX_first_sprite` and attaches the /Hearts of Iron IV/gfx/interface/folder/filename.dds image to it, and a second sprite similarly. The second sprite will be split into 2 frames: this is decided by having the left half of the image as the first frame and the right half as the second frame (more frames would further split the image horizontally). This doesn't make the sprite animated, just turns on the option to switch between the two halves as needed. `GFX_second_sprite:1` serves as a reference to the first frame, and GUI can be set up to change the shown frame depending on context, such as with radio stations.
In order to add animation, a frameAnimatedSpriteType is used.

It's never mandatory to copy a base game file to change a sprite. If there are duplicate definitions of a sprite with the same name in different files, the game will prioritise the one that would be evaluated later, based on the filename, and the older sprite will be ignored in entirety. This can be ensured by beginning the replacement file's name with a symbol late in the ASCII character table. Typically the lowercase letter 'z' is used for this purpose. For example, to change the amount of frames in `GFX_idea_traits_strip` to 10, it is possible to define a sprite with that name with 10 frames in the mod's modname/interface/zz_replace.gfx file instead of copying over the base game file.
Since most .gfx files define integral parts of the user interface, copying them over can lead to the mod's loaded files missing sprites upon a major game update, which would appear in-game as the default image, which is the error dog by default. As to ease the burden of needing to check the interface files, it's best to never copy over .gfx files, unless more additions would be actively harmful to the mod, such as with interface/subuniticons.gfx

By default, an idea uses a sprite that is the same as the idea's name, but with `GFX_idea_` prepended in the beginning. For example, an idea with the name of `my_idea_1` will use the sprite with the name of `GFX_idea_my_idea_1`.

This is possible to change using the `picture = my_picture` attribute within the idea. **This inserts the prefix of `GFX_idea_`** and so the sprite should be defined accordingly: in this case, the sprite with the name of `GFX_idea_my_picture` will be used, rather than `GFX_my_picture`.
This means that, most of the time, `picture = GFX_idea_my_picture` or `picture = idea_my_picture` are erroneous, leading to the default picture Unknown.png showing up since the idea links to a non-existing spriteType.

If the `texturefile` within the sprite is incorrect, the idea will be invisible in the country politics view, yet still apply the modifiers. Ensure that the idea is stored with the same folder path location, the same filename (including the extension), and that the right folder separator is used.

It is also possible to make the picture depend on the `graphical_culture_2d` of the country, defined in the /Hearts of Iron IV/common/countries/*.txt file. This is done by appending the name after the picture's name, separated by an underscore. For example, a sprite with the name of `GFX_idea_my_picture_middle_eastern_2d` in its definition will show up instead of `GFX_idea_my_picture` for countries that have the `middle_eastern_2d` 2D graphical culture.

Implementation[编辑 | 编辑源代码]

A spirit can only be added manually, using an effect. `add_ideas = idea_name` within an effect block (Such a focus completion reward), while `remove_ideas = idea_name` will remove it.

In order to swap two ideas and have the game show the modifiers, `swap_ideas` is used. **There is no way to directly modify an idea**, however in case the 2 swapped spirits have the same name in localisation, this will show up in-game as modifying the idea. `swap_ideas` is used as the following:

```
swap_ideas = {
  remove_idea = old_idea
  add_idea = new_idea
}
```

In order for a spirit to last for a period of time, the `add_timed_idea` effect is used as such:

```
add_timed_idea = {
    idea = my_timed_idea
    days = 365
}
```

modify_timed_idea can be used to extend or speed up the timer as needed.

In order for a country to start with the idea, the file in /Hearts of Iron IV/history/countries/ for that country serves as an effect block that decides the starting historical information. Using add_ideas, usually in the expanded form to assign multiple ideas at once, in there will enforce it starting with the idea. In order for the idea to show up in country selection, the bookmark is edited to include the idea within its `ideas = { ... }` block for the country's entry.

Modifiers[编辑 | 编辑源代码]

Modifiers are applied continuously as long as the spirit is applied to a country. Alongside modifiers, the spirit can also add a bonus to a technology category or modify an equipment archetype. The spirit can only apply modifiers towards the country that has the spirit, there is no way to scope into a different country. **Each of these attributes is entirely separate within the idea** and so each one should be located directly inside of the idea. For example, defining `research_bonus` inside of `modifier = { ... }` is erroneous, since `research_bonus` is an attribute of the idea rather than a modifier.

`modifier = { ... }` stores the regular modifiers, applied to the country with the spirit. A modifier block can list multiple modifiers, and negatives are also allowed. **Variables do not work in this section**: instead, use dynamic modifiers. A typical example of a modifier block is the following:

```
modifier = {
    political_power_cost = 0.1
    stability_factor = -0.2
}
```

`targeted_modifier = { ... }` is for using modifiers targeted towards a different country. The target is specified as `tag = ABC`, where ABC represents the target's tag. These are still applied towards the country with the spirit, but their effect is targeted towards a different country. For example, the following block will give the country with the idea a 10% attack bonus against Flag of Afghanistan Afghanistan:

```
targeted_modifier = {
    tag = AFG
    attack_bonus_against = 0.1
}
```

`research_bonus = { ... }` grants the country a boost to researching a specific technology category. An unsorted list of technology categories can be found in /Hearts of Iron IV/common/technology_tags/ or, if desiring to know which technologies exactly are assigned to each category, individual technologies can be checked in /Hearts of Iron IV/common/technologies/. The following example would provide a +10% bonus to researching destroyers and a -20% bonus to artillery:

```
research_bonus = {
    dd_tech = 0.1
    artillery = -0.2
}
```

`equipment_bonus = { ... }` applies the bonuses towards an equipment archetype or a type of equipment archetypes. By default, the bonus is not granted immediately, but rather requiring to research a new tech first, which is how it's done within designer ideas. This can be prevented by adding `instant = yes` inside of the equipment archetype within the equipment bonus. Equipment archetypes are defined in /Hearts of Iron IV/common/units/equipment/*.txt, which is where you can also find what can be applied to them. The name of the block within the /Hearts of Iron IV/common/units/equipment/*.txt file would be the archetype, while `type = { ... }` or `type = archetype_type` provides the types that the equipment archetype has. The following example will modify the cost to produce equipment within the `artillery` type (including artillery, rocket artillery, and tank artillery) by -20%, applying immediately, as well as increasing soft attack of any infantry equipment researched after the spirit was added by 10%:

```
equipment_bonus = {
    artillery = {
        instant = yes
        build_cost_ic = -0.2
    }
    infantry_equipment = {
        soft_attack = 0.1
    }
}
```

`rule` modifies the set of rules that decide what the country is allowed to do. The possible rules include `can_be_called_to_war`, `can_boost_other_ideologies`, `can_create_factions`, `can_declare_war_on_same_ideology`, `can_declare_war_without_wargoal_when_in_war`, `can_decline_call_to_war`, `can_force_government`, `can_generate_female_aces`, `can_guarantee_other_ideologies`, `can_join_factions`, `can_join_factions_not_allowed_diplomacy`, `can_join_opposite_factions`, `can_lower_tension`, `can_not_declare_war`, `can_occupy_non_war`, `can_only_justify_war_on_threat_country`, `can_puppet`, `can_send_volunteers`, `can_use_kamikaze_pilots`, and `units_deployed_to_overlord`.

```
rule = {
    can_join_factions = no
    can_send_volunteers = yes
}
```

Effects[编辑 | 编辑源代码]

A spirit can be set to apply effects when added or removed. This is done, respectively, with `on_add` and `on_remove` effect blocks, looking like the following example:

```
on_add = {
    add_stability = 0.1
}
on_remove = {
    add_political_power = -50
}
```

The effects will only get executed if the idea is added after the game has already started: if done via a history file or a bookmark's `effect = { ... }` block, the effects will not be executed. In that case, it's needed to replicate the effects given when adding the idea within the same effect block that adds the idea.

Cancellation[编辑 | 编辑源代码]

A spirit can be set to automatically cancel itself once a set of triggers is met. This is done with the `cancel` block. A cancellation will also trigger the `on_remove` effect block in the idea if one is present. This will look like the following:

```
cancel = {
    has_political_power > 50
```

}

Additional arguments[编辑 | 编辑源代码]

Another trigger block that can go into an idea is `allowed_civil_war = { ... }`. When starting a civil war, this is evaluated for each side, and the spirit will only appear for sides where this is true. For instance, a `has_government = democratic` will ensure that only the DemocraticDemocratic side in a civil war will obtain the spirit. **By default, always false, leading to the spirits disappearing when a civil war starts.** Setting it to be always true can be preferred over the default, making it appear for both sides.

`do_effect = { ... }` will ensure that any modifiers (including research and equipment bonuses) will apply only if the triggers are true. If this trigger block is false, the idea will not be removed, but it will not do anything.

`name = <localisation_key>` **name** assigns the idea's name to use a different localisation key. This can be useful if you plan to modify an idea by swapping it multiple times to not create a localisation entry for each one of them

Full idea file example[编辑 | 编辑源代码]

```
ideas = {
  country = { # Necessary for the game to consider them spirits rather than a different idea type.
    my_spirit = {
      picture = my_picture      # Using GFX_idea_my_picture
      modifier = {
        training_time_factor = -0.1
      }
      targeted_modifier = {
        tag = QAT
        defense_bonus_against = 0.2
      }
      research_bonus = {
        infantry = 0.1
      }
      equipment_bonus = {
        infantry_equipment = {
          instant = yes
          defense = 0.2
        }
      }
      rule = {
        can_join_factions = no
      }
    }
    my_spirit_2 = {      # Using GFX_idea_my_spirit_2
      allowed_civil_war = {
        has_government = democratic      # Only appear in a civil war for the democratic side.
      }
      cancel = {
        democratic > 0.6
      }
      on_add = {
        add_popularity = {
          ideology = democratic
          popularity = 0.2
        }
      }
      on_remove = {
        if = {
          limit = {
            democratic > 0.6
          }
          start_civil_war = {
            ideology = democratic
            size = 0.6
          }
        }
      }
      do_effect = {
        NOT = {
          has_government = democratic
        }
      }
      modifier = {
        democratic_drift = 0.005
      }
    }
  }
}
```

Other idea categories[编辑 | 编辑源代码]

Non-spirit ideas can use everything that can be within spirits, however there are more arguments that make sense to be added in them that do not do anything in spirits or don't have any reason to be added.

Hidden ideas[编辑 | 编辑源代码]

Hidden idea are exactly the same as regular spirits in every aspect in regards to creation, except for the fact that they have to be defined within the `hidden_ideas` idea category instead of using country. This will ensure that the idea will be hidden and not show up in the spirit container. However, it can still make sense to create localisation for them: If a hidden idea cancels automatically, the pop-up will show up, and hovering over certain elements (Such as the stability counter or research speed) will show each idea that modifies it.

Idea category arguments[编辑 | 编辑源代码]

These arguments are used within the idea category within the /Hearts of Iron IV/common/ideas/*.txt file.

`designer = yes` marks the entire category as designer ideas. This is primarily used for AI.
`law = yes` marks the entire category as laws. This is primarily used for AI.
`use_list_view = yes` marks the entire category to use a list view for selecting the idea, akin to how the base game treats laws.

These will look in the idea file like the following:

```
ideas = {
  my_law_category = {
    law = yes
    use_list_view = yes
    my_law_1 = {
    }
    my_law_2 = {
    }
  }
}
```

Additional arguments[编辑 | 编辑源代码]

This assumes that the idea is of the type that can be selected in-game, such as a law or a designer.

`allowed` is a trigger block that checks only at the game's start or when loading a save, primarily used to restrict an idea to a country (As `tag = BHR` or `original_tag = POL`) and/or a DLC (As `has_dlc = "One Step Back"`). If an idea's allowed is unfulfilled, it will never appear within the selection unless it becomes true on the save being reloaded; however, manual assignment via `add_ideas` bypasses the check. If left out, assumes to be always allowed. **This only checks once!**

`allowed_to_remove` is a trigger block that details when exactly you can remove the idea, changing it to a different one in the category. This is checked continuously, unlike `allowed`.

`visible` is a trigger block that continuously checks every frame if allowed was met, required to make the idea be visible in the decision selection screen. It is preferable to put country or DLC checks into allowed instead.

`available` is a trigger block that continuously checks every frame if visible was met, required to be possible to actually take the idea. If false, the idea will remain visible, but will be greyed out and be impossible to take. This is applied on top of the political power cost.

`cost = 123` is the price in political power it takes to add the idea. This becomes 150 political power if not set[a].

`removal_cost = 123` is the price in political power it takes to remove the idea. If set to -1, the idea cannot be removed manually. Defaults to 0 if not set.

`level = 2` is used in ideas to create the escalating price, akin to the recruitment laws. If you have the law with the level of 1, changing to level 3 will require the cost of the idea with the level 2 in addition to the level 3, and vise-versa.

`traits` is a block of traits that are assigned to this idea, defined in /Hearts of Iron IV/common/country_leader/*.txt, also granting a static modifier and showing up near the idea.

`ledger` decides which intelligence ledger the idea will be assigned to. This is primarily a leftover now applying to the character system, but this can still be used in officer corps spirits. Possible values are `army`, `air`, `navy`, `military` (Appearing on each of the prior ledgers), `civilian`, `all`, and `hidden`.

Examples[编辑 | 编辑源代码]

```
my_law = {
  allowed_to_remove = {
    num_of_civilian_factories > 10
  }
  removal_cost = 100
  cost = 100
  level = 3
  modifier = {
    production_speed_industrial_complex_factor = 0.2
  }
}
```

```
my_designer = {      # No cost is defined, defaults to 150
  allowed = {
    tag = QAT
  }
  visible = {
    has_tech = infantry_weapons1
  }
  available = {
    has_equipment = {
      infantry_equipment > 1000
    }
  }
  traits = { infantry_equipment_manufacturer }
}
```

Categories[编辑 | 编辑源代码]

Idea categories are defined in any /Hearts of Iron IV/common/idea_tags/*.txt file. All categories will be contained in the `idea_categories` top element and can contain the following arguments:

`slot` is a proper idea category slot. This is what shows up in the GUI in the politics view menu, and this is what ideas have to have as their category in their definition.

`character_slot` is a character slot. Ideas can also be defined to use this slot and it will show up in the GUI like a regular idea slot, however.

`cost` is the default price in political power to add an idea within one of the slots of this category. This is applied if the idea doesn't have one.

`removal_cost` is the default price in political power to remove an idea within one of the slots of this category.

`ledger` is the intelligence ledger defined to ideas in this category, unless overridden within the slot, idea, or character. Possible values are `army`, `air`, `navy`, `military` (Appearing on each of the prior ledgers), `civilian`, `all`, and `hidden`. Additionally, `invalid` can be used, forcing the ledger to be defined one level lower.

`hidden` is a boolean value making this category not show up in the GUI. Optional, defaults to no.

`politics_tab` is a boolean value making this category show up in the politics tab. Optional, defaults to yes.

This will look like the following:

```
idea_categories = {
  my_category = {
    slot = my_slot
    character_slot = my_character_slot
    cost = 30
    removal_cost = 10
    ledger = hidden
  }
}
```

The category will then be used by the idea via the slot (which allows mapping an idea to a category)
The file also includes slot_ledgers, which distributes idea slots to intelligence ledgers individually, which will take priority over the idea category's ledger. An example of it being used is

```
slot_ledgers = {
    XXX_idea_slot = civilian
    ZZZ_idea_slot = invalid
}
```

Invalid will require specification in each idea individually.

GFX and GUI[编辑 | 编辑源代码]

See also: Interface modding

Each idea category represents a row in the country politics view, while each slot represents a slot under that row.
The `country_politics_idea_category_entry` container window within the /Hearts of Iron IV/interface/countrypoliticsview.gui file decides the user inteface information about each row. The most important thing here is the gridbox, as it decides the following:

The size of each idea slot within the category, by default 80 by width and 64 by height. This is used for distances between slots.

The max amount of slots, **by default 7 horizontally and 1 vertically.** If not adjusted, it'd be impossible to have more than 7 idea slots in a category.

The format of the gridbox, deciding in which direction the idea slots are added.

The idea category's icon, shown on the left of the category's row, is decided by the GFX_idea_categories sprite. The sprite is split into multiple frames horizontally, and each category is assigned a part of it depending on its order defined in /Hearts of Iron IV/common/idea_tags/*.txt. The sprite is defined in /Hearts of Iron IV/interface/countrypoliticsview.gfx and its definition in the base game consists of the following:

```
    spriteType = {
        name = "GFX_idea_categories"
        texturefile = "gfx/interface/idea_categories.dds"
        noOfFrames = 6
    }
```

When adding a new category, make sure to update the sprite and the amount of frames accordingly.

An icon can be defined for idea slots, which will be used for them if there is no idea assigned to that slot. This definition in any /Hearts of Iron IV/interface/*.gfx file, taking XXX_idea_slot as an example, is done the following way:

```
    spriteType = {
        name = GFX_idea_slot_XXX_idea_slot
        textureFile = gfx/interface/filename.dds
    }
```

In other words, the sprite must have the same name as the idea slot but with `GFX_idea_slot_` prepended in the beginning.

Modifying cost of a slot[编辑 | 编辑源代码]

The `<idea slot>_cost_factor` modifier can be used to modify the price in political power for adding ideas or characters in this slot. However, there is a restriction on that modifier. In order for it to work, the idea slot needs to have any ideas or characters defined beforehand.

The files in the /Hearts of Iron IV/common/ideas/*.txt folder are loaded in the order of the unicode character IDs, which put capital letters before underscores, which are put before lowercase letters. This means that a file with the filename of `TAG.txt` is loaded before `_economy.txt` or `_manpower.txt`, which would get loaded before `country.txt`. Due to the above quirk, this means that an idea with the modifier affecting a price of a law, for instance, `economy_cost_factor = -0.10`, **will throw an error if the filename starts with an uppercase character**, however it will still work in practice.

There are two primary ways to fix this. The first one is to simply change the filename of the country's idea file, changing it from, for instance, `GER.txt` to`germany.txt`. In case you have to keep the filename the same to overwrite a file, a new file can be created instead. Alternatively, assuming that you have a replace_path to the ideas folder, you could rename the files storing the laws to begin with a character with a smaller ID, for instance, to `00_economy.txt`, which would get loaded prior to `TAG.txt`.

Additionally, this also means that there have to be ideas defined within character slots for the modifier to work, such as `political_advisor_cost_factor = 0.1`. This is because characters are loaded later than ideas or country leader traits, so they're not loaded yet when evaluating the modifier, causing the error. This can be bypassed by creating ideas for the character slots in /Hearts of Iron IV/common/ideas/*.txt files. These will never appear for the countries as these are character slots rather than idea slots, but an idea that was loaded beforehand will correct the error.

Notes[编辑 | 编辑源代码]

^ **a:** Depends on the value in the /Hearts of Iron IV/common/idea_tags/*.txt file, which is always 150 in base game