Country creation - Hearts of Iron 4 Wiki

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

The guide below presumes that you understand the basics of how to create a mod. This page should also help modders understand how to change some more specific aspects of a country.

Quick check-list[编辑 | 编辑源代码]

Define a unique 3-character country tag in any <yourmod>/common/country_tags/*.txt file.

Create a file in the <yourmod>/common/countries folder with the same name as the one that's linked in the country tag definition.

Define the country leader and optionally advisors and unit leaders in <yourmod>/common/characters

Create a .txt file in the <yourmod>/history/countries folder, with the first three letters of the filename being the country's tag, and fill it with the starting historical information.

Create an OOB file in <yourmod>/history/units and load it within the previous country history file. **Buildings will not be assigned to the country properly otherwise.**

Add the country's name to any <yourmod>/localisation/english/*_l_english.yml file that uses the UTF-8-BOM encoding[a].

Add a large, medium and small flag for each ideology to the <yourmod>/gfx/flags folder.

Optionally add the country to the game by assigning it states within the <yourmod>/history/states/*.txt files.

Country tag[编辑 | 编辑源代码]

Countries need a 3-character country tag. This is how a country is referred to in all of the game's script. All country tags must be unique. A country must not share a tag with any other country.

To create a tag, add a new file to <yourmod>/common/country_tags (or edit an existing file, though creating a new one is preferable due to making the mod easier to make compatible to newer game versions) and add a line similar to `SCO = "countries/Scotland.txt"`, where `countries/Scotland.txt` is the address of the country file that will be created/edited, located in <yourmod>/common/countries

There are some country tags that must be avoided, as they will cause in-game confusion whether something is meant to be a reference to a country or not. Their mere existence can cause other code to break. A list of them is the following:

Tag	Reason to avoid
NOT	Used as a flow control tool that comes up as true when any of the triggers within is false.
AND	Used as a flow control tool that comes up as true when all of the triggers within are true.
TAG	Used as a trigger that checks the country being chosen.
OOB	Used within country history files as an argument deciding the loaded order of battle file, deciding the amount of divisions and where they are located.
LOG	Used as an effect or a trigger that writes the specified argument within the user directory's /Hearts of Iron IV/logs/game.log file or in console if it is open.
NUM	Used for arrays in order to count the amount of elements within them. This will break the resistance system as the scripted trigger responsible for deciding when resistance must be enabled utilises this.
RED	Used within custom map modes as a temporary variable deciding how red the state should be. This will make every custom map mode always have 0 as the red value.
Numbers	Numbers meaning an entirely numeric country tag such as `123`, although the game prevents tags from *beginning* with numbers. While numbers can be used within country tags (such as `D01`), if it's entirely numeric there might be confusion whether, for example, `123` would refer to a state ID or a country tag.

A file can have `dynamic_tags = yes` within of itself, which marks every country defined in that file afterwards to be marked as a dynamic country. A dynamic country is one that doesn't have explicit data defined, but instead copies it on the fly from an original country that it branches off. This is most commonly used within civil wars, with the effect to start one auto-generating one. **If the mod doesn't have enough dynamic countries defined, the game will crash** if there is a sufficient amount of non-dynamic countries, with the last read file being marked as map/cities.txt or a savegame (Though that crash is not necessarily caused by the lack of dynamic countries).

Country file[编辑 | 编辑源代码]

Create a new file in the <yourmod>/common/countries folder. It is recommended one copies the file of a very similar country, then rename and edit it.

Graphical culture defines the set of graphics used for units, while the 2D graphical culture defines portraits used for aces and generic advisors. A list can be found in /Hearts of Iron IV/common /graphicalculturetype.txt. For Scotland this would be the following:

graphical_culture = commonwealth_gfx
graphical_culture_2d = commonwealth_2d

`color = { 2 10 222 }` is the country's default colour on the political map mode (overridden by a colors.txt entry if one exists. **Commonly, this doesn't work, making the colors.txt entry mandatory**). It uses the RGB color model on the scale from 0 to 255.

Additionally, an entry can be made in <yourmod>/common/countries/colors.txt in order to specify more country colours. The file **has to be called colors.txt**, and so it will overwrite the base game file if the entry is added. An entry in the file is formatted as the following:

SCO = {
   color = rgb { 2 10 222 }
   color_ui = rgb { 255 255 155 }
}

`color` will overwrite the country's colour on the political map, while `color_ui` will be used externally, such as in the map generated with the `^Ctrl + F10` hotkey or the history viewer tool in /Hearts of Iron IV/tools/history_viewer/. Colors may also be specified in the HSV color model, in which case `rgb` will need to be replaced with `hsv`. The `color_ui` is also used in order to colour in the background of the country's division counter, if the option is set to be country-specific. If a country doesn't have a color_ui entry, it defaults to a grey colour.
In HSV, hue, saturation, and value are each on the scale from 0 to 1, using decimals to show the value.

After reading the colour from the file, the saturation and value of the colour will be multiplied by 0.6 and 0.8 respectively by default[1]. This means the colour appearing in-game will be slightly darker and less saturated (closer to a shade of grey) than defined in the file. In order to have a country colour in HSV have a higher saturation or value than 0.6 or 0.8 respectively, a value larger than 1 can be used as long as it'll end up between 0 and 1 after the application of modifiers. For example, a pure red country colour can be simulated in HSV with `color = hsv { 0 1.66 1.25 }`. After applying the offsets, this will result in a saturation of 0.996 and a value of 1, being as close to pure red as possible.
For RGB, this calculation is harder: "Saturation" defines the difference between the colour with the largest value and the others (changing it does not change the largest colour's value), while "Value" manifests in multiplying every colour's value by that. For replicating a desired value, let V_{M} represent the maximum value of colour. As such, each colour's value, originally V, by default should be $\frac{5}{12}\left(5V-2V_{M}\right)$. With arbitrary Value/Saturation modifiers (represented with M_{V} and M_{S}), it's $\frac{1}{M_{V}}\left(\frac{V}{M_{S}}+V_{M}*\frac{M_{S}-1}{M_{S}}\right)$. For example, pure green (i.e. what `rgb { 0 255 0 }` intends), would be represented well with `rgb { -107 425 -107 }` plugging numbers into that formula. Indeed, entering this value into the game, the result is exactly pure green, far closer to it than `rgb { 0 255 0 }` or `rgb { 0 425 0 }` are.
These modifiers do not get applied when using the `tag_color` console command.

Country history[编辑 | 编辑源代码**]**

Create a file in the <yourmod>/history/countries folder. Alternatively, copy, rename, and edit a similar file. The filename typically follows the format of `TAG - country name.txt`, e.g. "SCO - Scotland.txt". The important aspect is that the **first three letters of the filename must be the same as the country's tag** in order to be loaded, without taking capitalisation into account. Everything after the tag is purely a comment and can be made into anything: "SCO - Bahrain.txt" will work just as well for SCO as would "SCO_qatar.txt".
This file contains information about the starting attributes of the country for each scenario, such as the capital and the ruling party. They serve as an effect block, so most effects can be used within them. **These files get executed in the order of the countries in the country_tags file,** which may come into play for subjects, as puppeting changes party popularity.

A country may have more than one history file, in which case each one will be used.
As a note, **any unnecessary closing bracket will stop the execution of the file prematurely**, leading anything afterwards to not be read. If nothing in the history file after a certain point applies, this is likely the cause.

`capital = 121` defines the capital state. The province with the most victory points within that state will get used as the capital. In case of overlap, the lowest province ID out of the possible choices gets chosen. In this case, 121 is the state Lothian, which contains Edinburgh as the largest victory point. To reiterate, **this must be a state ID, not a province ID.**

Order of battle[编辑 | 编辑源代码**]**

> *See also: [Division modding](#)*

Orders of battle are used for the starting military information: templates, divisions that use them, as well as the starting navy and airforce. Traditionally, the starting equipment production is put there within instant_effect while equipment variants are within country history, however instant_effect in orders of battle is an effect block, similarly to country history, so it's not necessary to divide them in this manner. As a byproduct of how the engine works, **countries with no order of battle will have broken buildings**: any building [constructed via an effect](#) will not apply to the country immediately, rendering them impossible to use until the consumer goods percentage changes or a savefile gets reloaded. Orders of battle can also contain information on what national focus the country is *currently* doing, however any already-completed focuses must be in country history.

There are several arguments/effects that can be used in country history. Each one is constructed in the manner of `oob = "filename"`, which would link to the <yourmod>/history/units/filename.txt file. The .txt extension is omitted from the filename necessary to be put in. **The filename of the order of battle is irrelevant as long as it matches up with the one loaded within the country history file.** These are what can be used:

`oob = "SCO_1936"` is the traditional way to load a land order of battle, assigning it to the country to be loaded before the game's start. However, *this is not an effect but an argument in country history*, meaning that it can't be used within other effect blocks (e.g. if statements).

`set_oob = "SCO_1936"` is identical to the prior way: assigning a land order of battle to the country to be loaded before the game's start. However, *this is an effect*, meaning it can indeed be used in if statements. This is required to use within if statements, such as if different orders of battle are used depending on the DLC, however this can be used outside of if statements.

`set_naval_oob = "SCO_1936_naval_legacy"` and `set_air_oob = "SCO_1936_air_bba"` are similar to the previous one, however are instead used for the specified branches of the military. These will overwrite any other previously-defined order of battle of the same type.

`load_oob = "my_template_shared"` is an effect and **is not recommended to be used in history**. It, instead, loads the order of battle *immediately*, which can lead to errors if it's put before technologies necessary to load the order and can lead to crashes if the order contains any divisions. However, this can be used to load orders of battle *after* the first-defined orders. This can be used for compatibility reasons as to not overwrite any unnecessary files or to re-use the same file containing templates, starting effects, or starting focus information across different countries to be loaded after the base order of battle.

Naval and air units are divided into two separate orders of battle. This is because [Man the GunsMan the Guns](#) changes the naval system and [By Blood AloneBy Blood Alone](#) changes the air system enough that any orders of battle created for the base game will not be compatible with the DLC or vise-versa. At times, the land order of battle can also be divided into multiple, most commonly if there are any references to tank equipment within `force_equipment_variants` or starting equipment production. For each one of these, the same [if statements are used](#):

```
if = {
    limit = { has_dlc = "Man the Guns" }
    set_naval_oob = "SCO_1936_naval_mtg"
}
if = {
    limit = { NOT = { has_dlc = "Man the Guns" } }
    set_naval_oob = "SCO_1936_naval_legacy"
}
if = {
    limit = { has_dlc = "By Blood Alone" }
    set_air_oob = "SCO_1936_air_bba"
}
if = {
    limit = { NOT = { has_dlc = "By Blood Alone" } }
    set_air_oob = "SCO_1936_air_legacy"
}
```

Note that `else` only works in its nested form in history files.

Effects[编辑 | 编辑源代码**]**

> *Main article: [Effects](#)*

These are [effects](#) which are traditionally put in history files of the country to be executed on startup, meaning that they can be used elsewhere as well. **Any other [effect within the list](#) can be used as well.**

`set_research_slots = 3` defines the amount of research slots a country has access to at the start. Normally, major powers have 4 slots, minor countries in North America and Europe have 3, and minor countries in the rest of the world have 2. Defaults to 2 slots if not specified.

`set_stability = 0.7` defines the [StabilityStability](#) that a country will have at the start of the game, defined on the scale from 0 to 1. The prior example of 0.7 will lead to the country starting with 70% stability.

`set_war_support = 0.5` defines the [War supportWar support](#) that a country will have at the start of the game. Similar to stability, it is defined on the scale from 0 to 1, the prior example of 0.5 being 50%.

```
add_ideas = {
    henschel
    GER_autarky_idea
    #laws #Remember that the sharp sign is used to mark comments.
    war_economy
    extensive_conscription
}
```

This is the list of ideas that the country starts with in the game, including national spirits, design companies, and laws. Ideas are defined in /Hearts of Iron IV/common/ideas. If starting laws aren't defined, then the country will start with the laws of Volunteer OnlyVolunteer Only, Export FocusExport Focus, and Civilian EconomyCivilian Economy.

```
set_technology = {
    infantry_weapons = 1
    infantry_weapons1 = 1
    gw_artillery = 1
    interwar_antiair = 1
    fuel_silos = 1
    basic_train = 1
}
```

This defines the starting technologies of the country. A list of technologies can be found in /Hearts of Iron IV/common/technologies/. It is notable that, similarly to orders of battle, several technologies depend on DLC, such as the Man the GunsMan the Guns naval tech and No Step BackNo Step Back tank tech. In that case, an if statement needs to be used in order to set only the technologies that the country has access to, such as, using it in a conjunction with the prior naval oob:

```
if = {
    limit = { has_dlc = "Man the Guns" }
    set_technology = {
        basic_naval_mines = 1
        submarine_mine_laying = 1
        early_ship_hull_light = 1
        early_ship_hull_submarine = 1
        basic_ship_hull_submarine = 1
        basic_battery = 1
        basic_torpedo = 1
        basic_depth_charges = 1
    }
    set_naval_oob = "SCO_1936_naval_mtg"
}
if = {
    limit = { NOT = { has_dlc = "Man the Guns" } }
    set_technology = {
        early_submarine = 1
        early_destroyer = 1
    }
    set_naval_oob = "SCO_1936_naval_legacy"
}
```

`recruit_character = TAG_character_name` is used to recruit the specified character. This includes country leaders, commanders, admirals, and advisors. Country leaders that are recruited *after* setting political information can fail to appear, so it's recommended to recruit them before setting any political information. If multiple country leaders fall under the same political party, the game will place the first-recruited one as its leader.

[编辑 | 编辑源代码]

In order to make the country start as a subject, the following can be put in the history file:

```
ENG = {
    if = {
        limit = {
            has_dlc = "Together for Victory"
        }
        set_autonomy = {
            target = SCO
            autonomous_state = autonomy_integrated_puppet
        }
    }
    else = {
        puppet = SCO
    }
}
```

Since the history files get loaded in a specific order, it is preferable to put it in the subject's history file, as to avoid the case where the overlord's file gets loaded after the subject's, which'll make the puppeting reset the politics of the subject. For this reason, set_autonomy must also be set before the political party and popularities of the subject. The if statement is there to ensure that the integrated puppet is only there with Together for VictoryTogether for Victory, as that autonomous state is not available otherwise. A list of autonomous states can be found in /Hearts of Iron IV/common/autonomous_states/*.txt files.

A faction can be created with `create_faction = localisation_key`. The localisation key will be used to define the name of the faction, defined in any <yourmod>/localisation/english /*_l_english.yml file.
Members are added to the faction with `add_to_faction = IRE` within the faction leader's history file. Although puppets usually get added to the faction automatically, it is recommended to add them manually as well.

In order to use technology sharing groups, `add_to_tech_sharing_group = my_tech_group` can be used in each country's history file. This requires the Together for VictoryTogether for Victory DLC in order to work properly. Technology groups are defined in /Hearts of Iron IV/common/technology_sharing/*.txt files. Additional details on the technology groups can be found on the dedicated article.

[编辑 | 编辑源代码]

```
set_popularities = {
    democratic = 80
    communism = 10
    fascism = 10
}
```

This defines the starting party popularities, on the scale from 0 to 100. If an ideology is left out, then it will be assumed to have 0% popularity. **If it doesn't add up to 100, then the popularities will fail to be set.**

```
set_politics = {
    ruling_party = neutrality
    last_election = "1932.11.8"
    election_frequency = 48 # Every 4 years
    elections_allowed = yes
}
```

This defines the starting political situation of the country, with the ruling party and details on elections. Notably, the election frequency is defined in months.

Variants[编辑 | 编辑源代码]

Special variants on equipment available for this country from the start can be put in, defining the starting equipment upgrades or modules. Since this can depend on DLC, for ships, tanks, and airplanes, a DLC check can be used as the following:

```
if = {
    limit = { has_dlc = "Man the Guns" }
    create_equipment_variant = {
        name = "Celtic Series"
        type = ship_hull_submarine_1              # See list in common/units/equipment
        name_group = SCO_SS_HISTORICAL            # Defines names used for ships of this variant, defined in common/units/names_ships/
        modules = {
            fixed_ship_torpedo_slot = ship_torpedo_sub_1    # Module slots for equipment types are defined in common/units/equipment
            fixed_ship_engine_slot = sub_ship_engine_1      # While modules themselves are defined in common/units/equipment/modules
            rear_1_custom_slot = empty
        }
        obsolete = yes                     # Marks as obsolete, changing the UI.
    }
}
if = {
    limit = { NOT = { has_dlc = "Man the Guns" } }
    create_equipment_variant = {
        name = "Celtic Series"
        type = submarine_1                  # Note different equipment type due to no MtG.
        upgrades = {
            ship_reliability_upgrade = 1         # See list in common/units/equipment/upgrades
            sub_engine_upgrade = 1
            sub_stealth_upgrade = 1
            sub_torpedo_upgrade = 1
        }
        obsolete = yes
    }
}
```

Adding history for other bookmarks[编辑 | 编辑源代码]

If you want to edit something about your country, but only want it applied after a specific date (as in 1939 bookmark for example), a block with a date (in the YYYY.MM.DD format) can be used as this:

```
1939.1.1 = {
    oob = "SCO_1939"
    set_technology = {
        atomic_research = 1
        nuclear_reactor = 1
        nukes = 1
    }
}
```

In particular, this would *replace* the used order of battle with /Hearts of Iron IV/history/units/SCO_1939.txt and *add* atomic_research, nuclear_reactor, and nukes technologies to the ones the country already has researched.
Any effect or history-specific argument such as oob can be used within a date-stamp. This will only apply on the start dates that are **strictly** later than the defined timestamp.

Useful effects[编辑 | 编辑源代码]

While any effects can be used, same as in regular history, the following are specifically useful for later start dates:

In order to simulate completion of the national focus tree, the two effects are used: unlock_national_focus = focus_name, which bypasses the focus without granting effects, and complete_national_focus = focus_name, which fully completes the focus, granting the effects.

The diplomatic_relation effect can be used in order to cancel previously applied diplomatic relations, such as guarantees or military access. Ideas would be removed with the remove_ideas effect, applied in the exact same manner as add_ideas

In order to replace the country leader, the promote_character effect can be used as promote_character = TAG_character_name. However, make sure that the character is recruited beforehand and has a country leader role within their definition. Recruitment is allowed within a date-block.

Full file example[编辑 | 编辑源代码]

> 展开The text in this section has been collapsed by default.

Flag[编辑 | 编辑源代码]

The flags representing a country are stored in /Hearts of Iron IV/gfx/flags and its subfolders. Each country needs at least 3 flags: the regular flag is used when viewing the country in the diplomacy menu or in the top-left corner of the screen, while the medium and small flags are used for divisions and in localisation; the flags get automatically assigned to countries based on the flag's filename. Each flag needs to be a TGA-file with the name of the country tag and, optionally, the ideology group that the flag is assigned to. **The flag file format is uncompressed 32-bit ARGB.** In this case, "32-bit", "32 bits per pixel (bpp)", and "32 bitdepth" have the exact same meaning. In some editors where the bitdepth isn't settable when exporting but rather static without allowing the user to draw anything that the bitdepth doesn't allow, 32-bit means the RGB image mode alongside an alpha channel used for transparency present within the image. If it is 24-bit (meaning there is no alpha channel), the file will load but throw an error stating that it'll slow down the game and that it's recommended to use 32bpp.

If the origin point is set to top left rather than bottom left, the image will appear flipped upside down. This can be common within online converters. Instead, it's better to use a more configurable image editor in order to convert files to TGA, such as Paint.net, GIMP, or Photoshop, with Photoshop requiring a plug-in to work with TGA.

There are two ways to name a flag:

Ideology group-specific, such as `SCO_neutrality.tga`. These will have the highest priority as the starting flags (Assuming no cosmetic tags are set).

Non-specific, such as `SCO.tga`. If there's an ideology group-specific flag for the party ruling over the country, that one will be chosen, and this one is used as a backup in case it's not present.

Either one is optional, however it's recommended for a country to have some sort of a flag regardless which ideology group it has (Meaning either that there's a non-specific flag optionally alongside some ideology group-specific ones or that each ideology group in the mod has a specific flag for this country)

	Size requirement
Standard	82x52 pixels
Medium	41x26 pixels
Small	10x7 pixels

For the purpose of this mode, place them in <yourmod>/gfx/flags, each flag size into the corresponding folder.

Localisation[编辑 | 编辑源代码]

See also: Localisation

Localisation, such as the country name, is defined in the <yourmod>/localisation/english/ folder for English. It is preferable to use a new file in the folder, as to not overwrite any base game files. The newly-created file will have to end with `_l_english.yml` in its filename (**This is a lowercase L, not an uppercase i**) for it to be loaded properly. Additionally, **it has to use the UTF-8-BOM text encoding**. Exact details on how to change the encoding depend on the text editor used.

Within the file, localisation can be added accordingly:

```
l_english:
 SCO: "Scotland"                #Non-ideology name, primarily used for collaboration governments and as a fallback if an ideology group-specific one doesn't exist.
 SCO_DEF: "Scotland"
 SCO_ADJ: "Scottish"
 SCO_democratic: "Republic of Scotland"
 SCO_democratic_DEF: "the Republic of Scotland"
 SCO_democratic_ADJ: "Scottish"
 <...>
```

The localisation key follows the formatting of `TAG_ideology`, where TAG is to be replaced with the country's tag and ideology is to be replaced with the ideology group. The _DEF and _ADJ suffixes may be added. The suffixes serve the following purpose:

No suffix: The primary name for the country. Shows up on the world map and with the [TAG.GetName] localisation command

_ADJ: The adjective form for the country. Used with the [TAG.GetAdjective] localisation command.

_DEF: The definition form for the country. Used with the [TAG.GetNameDef] localisation command and when hovering over the state on the world map as the owner/controller. In base game, it's most commonly the same as the country's regular name but with "the " (Beginning with a lowercase letter as [TAG.GetNameDefCap] exists for otherwise) inserted in the beginning *if needed*. Due to widespread use of this localisation command within generic events and decisions, this is essentially mandatory to define.

By changing each instance of English to the corresponding language, including the filename, the folder, and the first line of the file, it is also possible to make the name for different languages, including and limited to Brazilian Portuguese (l_braz_por instead of l_english), French, German, Polish, Russian, Spanish, and Japanese.
Additionally, there are other cosmetic versions of the country's name that can apply automatically that can be mixed together:

```
 SCO_liberalism: "Federal Republic of Scotland"        # Applies when the country has a specific ideology type, commonly called sub-ideologies in community jargon.
 SCO_subject: "$OVERLORDADJ$ Scotland"                 # Applies when the country is a subject of any other country.
 SCO_IRE_subject: "Alba"              # Applies when the country is a subject of a specific another country.
 SCO_autonomy_dominion: "Dominion of Scotland"         # Applies when the country has the specified autonomy type with any other country as overlord.
 SCO_IRE_democratic_autonomy_integrated_puppet: "Most Northern Ireland" # Example of mixing these together. Must be in the country-ideology-autonomy order.
 COUNTRY_SCO_autonomy_dominion: "Comhairle nan $NONIDEOLOGY$"        # The first tag can be replaced with COUNTRY to apply to a generic country.
```

IDs of autonomous states can be found within /Hearts of Iron IV/common/autonomous_states/*.txt files, while the names of ideology types can be found in /Hearts of Iron IV/common/ideologies /*.txt files.
The country names are applied with the following priority:

Autonomous state with a specific country as overlord

Autonomous state

Subject of a specific country

Subject

Ideology type

Ideology group

TAG

COUNTRY

For example, `COUNTRY_SCO_autonomy_dominion`, due to being on the top of the list, will apply towards *every* country that's a dominion of Scotland, unless a more specific mix of a Scottish-controlled dominion applies (e.g. `ENG_SCO_autonomy_dominion` or `COUNTRY_SCO_democratic_autonomy_dominion`).

Inside of subject names, nested strings can be used that correspond to the overlord's name and the country's non-ideology name. Despite the fact that the latter doesn't seem to be subject-related, it cannot be used in names of independent countries. These include:

`$NONIDEOLOGY$`, `$NONIDEOLOGYADJ$`, and `$NONIDEOLOGYDEF$` for the non-ideology name.

`$OVERLORD$`, `$OVERLORDADJ$`, and `$OVERLORDNAMEDEF$` (Note the NAME) for the overlord's name.

This can *only* be used in country names, regardless if it's an automatically-applying name or a cosmetic tag.

Starting characters[编辑 | 编辑源代码]

Main article: Character modding

Characters are defined in any file in the /Hearts of Iron IV/common/characters/ folder. An example of a file within that folder, with just a country leader, is the following:

```
characters = {
  SCO_ronald_mcdonald = {
    name = SCO_ronald_mcdonald
    portraits = {
      civilian = {
        large = GFX_SCO_ronald_mcdonald
      }
    }
    country_leader = {
      ideology = socialism
      traits = { scary_clown }
    }
  }
}
```

The `ideology` defined for the country leader must be an ideology type, rather than an ideology group such as democraticdemocratic. A list of ideology types across groups, alongside more information, can be found on the dedicated character modding article.

The picture used is a spriteType, defined in any /Hearts of Iron IV/interface/*.gfx file. It is recommended to use a new file for this purpose instead of overwriting the base game files. An example of how a new file can look like is the following:

```
spriteTypes = {
  spriteType = {
    name = GFX_SCO_ronald_mcdonald
    texturefile = gfx/leaders/SCO/ronald_mcdonald.dds
  }
}
```

The name picked for the character depending on the language set when launching the game is defined within /Hearts of Iron IV/localisation/. In particular, for English, this would be in any /Hearts of Iron IV/localisation/english/*_l_english.yml file as this: `SCO_ronald_mcdonald:0 "Ronald McDonald"`. The description when hovering over the character can be added by appending `_desc` to the character's name, as `SCO_ronald_mcdonald_desc:0 "Insert description here"`.

In order to appear, a character **must** be recruited in the history file of the country. This is done with the `recruit_character` effect, as the following:

recruit_character = SCO_ronald_mcdonald

If multiple country leaders fall under the same political party, the game will place the first-recruited one as its leader. Since `set_politics = { ... }` forces a leader of the specified ideology group to come to power, any characters recruited after a `set_politics` block will not immediately become country leader since the country already has one: **recruitment of the intended country leader must be before setting politics**.

Corps commanders, field marshals, admirals, and advisors are also regarded as characters in the in-game code, and are added similarly: defined in the characters file and then recruited in /Hearts of Iron IV/history/countries/TAG*.txt

Order of battle[编辑 | 编辑源代码]

> *Main article: Division modding*

If the country will exist at the start, an order of battle is necessary for the country to have an army. Create a new file in /Hearts of Iron IV/history/units or copy one from a similar country. The filename, not counting the .txt file extension, must be the same as in the history file of the country in `oob = "SCO_1936"`. Specific names of different types of units and their support companies can be found in /Hearts of Iron IV/common/units.

Division template[编辑 | 编辑源代码]

First the default division template will be defined, simply set up what regiments will fill the country's divisions:

```
division_template = {
  name = "Blueskirt Division"

  regiments = {
    infantry = { x = 0 y = 0 }
    infantry = { x = 0 y = 1 }
    infantry = { x = 0 y = 2 }
    infantry = { x = 1 y = 0 }
    infantry = { x = 1 y = 1 }
    infantry = { x = 1 y = 2 }
    infantry = { x = 2 y = 0 }
    infantry = { x = 2 y = 1 }
    infantry = { x = 2 y = 2 }
    infantry = { x = 3 y = 0 }
    infantry = { x = 3 y = 1 }
  }
}
```

As can be seen, this division exists solely out of infantry. Example of more advanced unit template from Germany, which will use German infantry names:

```
division_template = {
  name = "Infanterie-Division"
  division_names_group = GER_Inf_01        # Defined in common/units/names_divisions/

  regiments = {
    infantry = { x = 0 y = 0 }
    infantry = { x = 0 y = 1 }
    infantry = { x = 0 y = 2 }
    infantry = { x = 1 y = 0 }
    infantry = { x = 1 y = 1 }
    infantry = { x = 1 y = 2 }
    infantry = { x = 2 y = 0 }
    infantry = { x = 2 y = 1 }
```

```
      infantry = { x = 2 y = 2 }
    }
    support = {
      engineer = { x = 0 y = 0 }
      artillery = { x = 0 y = 1 }
    }
}
```

Division placement[编辑 | 编辑源代码**]**

Division can be placed as follows:

```
units = {
    division= {
      name = "1st Blueskirt Division"
      location = 9392 # Edinburgh
      division_template = "Blueskirt Division"
      start_experience_factor = 0.2   #optional, otherwise starts with 0 experience
      start_equipment_factor = 0.3    #optional, otherwise starts with equipment factor of 1 (maximum)
    }
}
```

The name is simply the name the game shows for the division, the location is where the division is on the map, the template links to the template defined above and start_experience_ and _equipment_factors define how much experience and equipment the division has. Alternatively you can use the automatic naming using a predefined list of names.

```
    division= {
      division_name = {
          is_name_ordered = yes
          name_order = 35
      }
      location = 6488
      division_template = "Infanterie-Division"
      start_experience_factor = 0.1                      #optional, otherwise starts with 0 experience
      force_equipment_variants = { infantry_equipment_0 = { owner = "GER" } }      #optional, replaces equipment with a specific variant (for example to give obsolete equipment to garrison units)
    }
```

Air wings[编辑 | 编辑源代码**]**

Air wings are not placed into a specific province, but rather a state.

```
air_wings = {
    121 = {    # Edinburgh
      tac_bomber_equipment_0 = { owner = "SCO" amount = 72 }
          start_experience_factor = 0.2    #optional, otherwise starts with 0 expereince
      fighter_equipment_1 = { owner = "SCO"  version_name = "Scotfire" amount = 150 }      #version_name is optional without bba
          start_experience_factor = 0.5
    }
}
```

version_name defines a specific variant of that type of equipment to be used. It first needs to be defined in the same file as an instant effect or in the country history file.

Randomly-generated names (optional)[编辑 | 编辑源代码**]**

> *Main article: [Namelist modding](#)*

Ministers without a specific name specified, randomly-generated unit leaders, and aces pull from a name list, specific to each country. This name list is specified in /Hearts of Iron IV/common /names/*.txt. An entry in it looks like this:

```
TAG = {
    male = {
      names = {
          Name "Name with multiple words"
      }
    }
    female = {
      names = {
          Name "Name with multiple words"
      }
    }
    surnames = {
      Surname "Surname with multiple words"
    }
    callsigns = {
      Callsign "Callsign with multiple words"
    }
}
```

This will assign these names to TAG. While a name with one word can be outside of quotes, one with multiple words must use them to let the game know what's a part of the name and what isn't. The game will randomly pull a name, or the first name, and a surname, or the last name, each time it has to generate one. Callsigns are used for aces.

Randomly-generated portraits (optional)[编辑 | 编辑源代码**]**

> *Main article: [Portrait modding](#)*

When the game automatically generates a country or a unit leader, a portrait is randomly chosen from a pool that depends on country or continent, if there's not a country-specific one. This pool is located in /Hearts of Iron IV/portraits/*.txt An example entry looks like

```
TAG = {
```

```
navy = {
    male = {
        "<file path>"
    }
    female = {
        "<file path>"
    }
}

army = {
    male = {
        "<file path>"
    }
    female = {
        "<file path>"
    }
}

operative = {
    male = {
        "<file path>"
    }
    female = {
        "<file path>"
    }
}

political = {
    <ideology> = {
        male = {
            "<file path>"
        }
        female = {
            "<file path>"
        }
    }
}
}
```

This will assign these portraits to TAG. A proper file path looks like `"gfx/leaders/leader_unknown.dds"`

Adding the country to the game[编辑 | 编辑源代码]

The country will exist as long as an entry in /Hearts of Iron IV/common/country_tags/*.txt exists. However, it will not exist on the map at first.

If desiring to make it appear on the game's start, whether fully owning the states or just having them as cores, the states themselves need to be edited to be owned by the country at the game's start, stored in /Hearts of Iron IV/history/states/*.txt files.

If the country is only to appear in the middle of the game, effects can be used within any effect block, such as a focus reward or an event option. In particular, add_state_core and transfer_state, within that country's scope can be used to, respectively, core the specified states and change their owner and controller to the needed country.

Notes and references[编辑 | 编辑源代码]

^ **a:** A similar process applies for other languages that the game supports, with the filename ending with the internal name of the language. The exact folder path is actually irrelevant, as long as the file is contained within the localisation folder or any of its subfolders at any depth level.

↑ `NDefines.NGraphics.COUNTRY_COLOR_SATURATION_MODIFIER = 0.6` and `NDefines.NGraphics.COUNTRY_COLOR_BRIGHTNESS_MODIFIER = 0.8` in Defines