《钢铁雄心 IV》 1.15.x 版本与 "Götterdämmerung"扩展包迁移和 DLC 更新指南

作者：千禧黎明的 AngriestBird，马国戡乱史的 BiscuitCookie，56 之路的 SpicyAlfredo 等

《迁移指南》由霜泽图书馆 ParaWikis 汉化组汉化
汉化：秋起.，-[U|S|A]-William，littlekirovy，罗彧，Sergey-Taboritsky 等

本文档详细介绍了如何将你的 mod 从钢铁雄心 IV1.14.x 版本迁移到 1.15.x 版本。它还将包括有关新的机制、更改、特性和其他有关新的 DLC 和钢铁雄心 IV 补丁内容的信息。

首要迁移问题:

直接的 CTD 问题:

- 地区文件文件名中的变音字符需要更改为英文字符。
- core.gui 文件中需要新的 right_vertical_slider 容器。

潜在的 CTD 问题:

如果在修复了现有的崩溃问题后您的模组仍然有崩溃问题，或者在修改后出现新的崩溃问题，请检查以下可能的崩溃问题:

- 如果在 common/ideas 中覆盖了‘_economic.txt’文件，您现在可能需要在可用的 idea 代码块中包含这些新的 ELSE 语句，如果 IF 语句在 OR 语句中，如下例所示:

```
OR = {
    has_idea = war_economy
    has_idea = tot_economic_mobilisation
    if = {
        limit = {
            original_tag = GER
        }
        has_idea = totaler_krieg_economy
    }
    else = {
        hidden_trigger = { always = no } #NEED
    }
}
```

- 如果在 common\units 中覆盖了 air.txt 文件，您可能需要在单位属性中添加对“mega_carrier_air_wing_size = x” 的引用。
- 如果在 common\units 中覆盖了 air.txt 文件，您可能需要在此文件中添加对新导弹和特殊项目技术的引用。
- 目前在修改突袭时，游戏需要至少一个针对设施的突袭。如果您不希望有任何针对设施的突袭，只需在您的突袭文件之一中创建这个空白突袭:

```
types = {

    stop_crashing_my_game = {

        target_type = {
            building = {
                tags = facility
            }
        }
    }

}
```

- 覆盖的定义可能需要更新，请参阅下面的定义以了解变更。

地图/建筑方面更改:

- 地图文件 rocketsites.txt 与 airports.txt 如今已经弃用。你需要为全部建筑创建各自的实体模型，随后将其置于地区中。如果你需要移除建筑，可以使用 Nudger 工具。
- 新增了几种新建筑以及一个新的建筑定义词条，称为生成点（Spawn Points）。
- **生成点**需要放置于空军基地与火箭发射中心的连接处以正常起效。
- 现在可以通过在建造 UI 中启用以开启自动生成建筑模型的功能。

- 如果“Random All States”已经启用，地标生成“landmark_spawn”与水坝生成“dam_spawn”位置不一定会成功自动填充。在此情况下，需要使用 Nudger 工具手动在想要生成的地区单独选择相应建筑类型。

其他 GUI 变更:

重复的容器名称现在会在 error.log 中报告。检查这些并确保没有问题非常重要。这仅在它们位于同一文件/屏幕时相关。嵌套的相似名称容器不会出错。

根据更改，您可能需要解决模组 UI 中的问题的文件:

- airselectionview.gui
  - rocket_selection_view 现在作为支持新导弹系统的容器。不使用将遇到崩溃。
- Countrytechtreeview.gui
  - “research_groups_grid”在“technology_unit_statlist_item_equipment”中需要删除，因为如果它在您之前的 GUI 中存在将导致现版本游戏崩溃。
  - 新容器和其他更新也存在于此容器中
- mapicons.gui
  - 需要添加突袭过滤器元素以确保突袭动态过滤器的功能
- naviesview.gui
  - 航行中补给现在是一个功能。这需要在 UI 中添加一个新按钮，请务必检查此处。
- stateview.gui
  - 添加了一个新的地区建筑修正部分
  - 添加了一个新的动态修正部分，以帮助修复奇怪的推离
- unitview.gui
  - divisions_summary_item 从 unitview.gui 移到了它自己的文件中
- alerts.gfx
  - 如果您在标准警报条中有任何自定义警报，这些警报已扩展了 8 种更多的警报类型

脚本/代码更改:

脚本本地化更改

- 内联本地化在 GUI 和其他一些屏幕中不再有效。它必须附有一个本地化字符串到脚本 loc localization_key 才能正常工作。

  来自 v1.14.x 的旧示例:

```
   text = { trigger = { check_variable = { pos_array^1 = 12 } }
        localization_key = "[?party_pop_array^12|%1]"
    }
```

v1.15.x 的新版本运作方式:

```
   text = { trigger = { check_variable = { pos_array^0 = 14 } }
        localization_key = num_partypos_1_14
    }
```

效果变更

- **custom_effect_toolip** 现在允许绑定本地化
- **add_province_modifier** 现可通过设置 days =添加带有一定时长的修正，接受整数与整值变量
- **remove_building** 现在接受 tags = <buildingtag> 或 tags = { <buildingtag> } 字段，标签在建筑文件中设置。
- **create_unit** 现可为师指挥官添加 divisional_commander_xp = <int?>以在生成时给予指挥官一定经验
- **damage_building** 还添加了 tags = 字段和 repair_speed_modifier = <float>，这允许设置修复速度，直到完全修复。
- **force_update_dynamic_modifier**, **remove_dynamic_modifier** 支持 special_project 作用域
- **launch_nuke** 允许选择 nuke_type = 使用的核弹类型（例如 nuclear_bomb、thermonuclear_bomb 等）。

- 绑定本地化案例:

本地化所用代码:

```
add_pink_fluffy_unicorn = 1

custom_effect_tooltip = {
    localization_key = ADD_UNICORN # Root look key
    UNICORN_AMOUNT = "1"
}

Loc Entry:
ADD_UNICORN:0 "Adds $UNICORN_AMOUNT$ unicorn to your rainbow parade"
```

```
Resulting in the following:
"Adds 1 unicorn to your rainbow parade"
```

```
custom_override_tooltip = {
    tooltip = {
        localization_key = MY_TOOLTIP # Root look key
        IMPORTANT_QUESTION = { # ID IMPORTANT_QUESTION in MY_TOOLTIP will get
value:
            localization_key = MEANING_OF_LIFE # Root loc key in
IMPORTANT_QUESTION
            ANSWER = "42" # ID ANSWER in IMPORTANT_QUESTION will get value 42
        }
        JUST_AS_IMPORTANT = OR_NOT # ID JUST_AS_IMPORTANT in MY_TOOLTIP will get
value OR_NOT
    }
    <other effects>
}
```

触发条件变更:

- `custom_trigger_tooltip` 似乎已被 custom_override_tooltip 取代，并作为别名保留，因此现在也支持可绑定的 loc(本地化)。

脚本常量:

- v1.15.x 的新补丁引入了一个名为“脚本常量”的新概念，可以在 common/script_constants 下找到。

- 从功能上讲，它们是一种定义简单和复杂脚本值的方法，这些值可以在脚本中的不同文件中重复使用。只有记录支持脚本常量的条目才能在其脚本中使用它。

- 我们可以用它在一个区域设置一个值，然后从 script_constants 中的一个真实来源操纵该值。

示例常量定义:

```
sp_military_xp_gain = {
```

```
    schema = {
            any_key = yes
            data = int
    }
    low = 15
    medium = 30
    high = 50
}
```

脚本中的示例用法:

```
country_effects = {
    army_experience = constant:sp_military_xp_gain.medium
    FROM = { set_project_flag = sp_land_generic_reward_army_xp_flag }
}
```

任何支持作用域变量的地方都将支持脚本常量，通过在前面加上 constant:<yourconstantformathere>前缀。add_design_template_bonus 和 add_breakthrough_progress 直接支持常量，无需前缀。
在特殊项目中，prototype_time 和 complexity 也直接支持常量。

DLC 专属效果:

DLC 专属效果需要拥有 DLC 以解锁，主要围绕突袭行动与特殊工程的自定义内容，并包含少量通用内容。

DLC 专属内容:

- 地标建筑
    - 该内容无 DLC 仍然可用，但其原版游戏内的内容需要拥有 DLC 以解锁。
- 自定义突袭行动与特殊工程
    - 包含突袭大本钟、埃菲尔铁塔等建筑
    - 特殊工程包含大部分自定义新单位，包括陆地巡洋舰、巡洋潜艇等。
- 改进核武器系统

    - 该调整与导弹系统相关
- 新导弹系统
    - 旧版导弹 Guided_missile_equipment 仍然免费开放。新导弹类型，如洲际导弹、地空导弹、中程导弹全部需要 DLC 解锁，同时其发射任务也会要求拥有 DLC。
- 被重置的新国策树（非史实德国，奥地利，匈牙利，比利时，比属刚果）

免费更新内容:

该部分列出了可用于模组的全部非 DLC 更新内容

- 特殊工程（代码与通用部分）
    - 特殊工程对模组制作完全开放，除非涉及 DLC 锁定内容如巡洋潜艇、冰航母、新导弹或其他此类工程。
- 突袭行动（代码与通用部分）
    - 突袭行动对模组制作完全开放，除非涉及 DLC 锁定内容如新导弹。
- 地标建筑系统（非原版游戏）
    - 原版地标建筑系统对 DLC 开放并能够使用。然而，除非你拥有 DLC 或已经得到许可否则不能够使用原版的地标建筑（其主要原因是模型造成的版权问题）
- 改进 AI 突破行为
    - AI 改进对任何模组均开放。
- 改进 AI 决策行为
    - 此为 AI 对为重要决议保存政治点数的行为。这里并不需要进行模组迁移，同时也不需要修改。

新增效果:

add_breakthrough_point

作用域为国家（country）

```
Add breakthrough points to one specialization or all for a country scope.
ex:
    add_breakthrough_points = {
    specialization = <sp_specialization_id>
    value = 3
}
```

```
add_breakthrough_points = {
    specialization = all
    value = -1
}
```

raid_reduce_project_progress_ratio

作用域为地区（state）

```
Reduce progress to the special project in state. Root scope is raid instance scope.
The input value is a ratio of the total needed progress to complete the special project, i.e. a decimal number between 0 and 1.
ex:
# Root scope is raid
state = {
    raid_reduce_project_progress_ratio = 0.1 # Reduces the project progress by 10%
}
```

random_scientist

与其他 random_效果相同

every_scientist

与其他 every_效果相同

create_colonial_division_template

```
Create a colonial division template for overlord/owner. Available parameters are subject and division_template, where the subject parameter is the country tag for an overlords subject. And the division_template is the regular effect to create a division template.
Example.
In country scope of overlord, E.g. ROOT = ENG
create_colonial_division_template = {
    subject = RAJ # Country tag
    division_template = {
        name = "Infantry Division"
        division_names_group = RAJ_INF_01
        ...
        regiments = {
```

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

add_scientist_level

```
Add levels to a special project specialization for a scientist character in
scope.
The `level` parameter is a scoped variable

#### Example
```
my_character = {
    add_scientist_level = {
        level = 2 # accepts variables
        specialization = specialization_nuclear
    }
}
```
```

add_scientist_xp

```
Add experience to a special project specialization for a scientist character in
scope.
The `experience` parameter is a scoped variable.

#### Example
```
ex: my_character = {
    add_scientist_xp = {
        experience = 2 # accepts variables
        specialization = specialization_nuclear
    }
}
```
```

add_scientist_trait

```
Add a trait to a scientist character in scope.
```

```
ex:
my_character = {
    add_scientist_trait = my_trait_token
}
```

set_project_flag

与其他设置 flag 标签效果相同，但作用域为工程（project）

modify_project_flag

与其他设置 flag 标签效果相同，但作用域为工程（project）

clr_project_flag

与其他 random_效果相同，但仅用于可用科学家（active scientists）

every_active_scientist

与其他 every_效果相同，但仅用于可用科学家（active scientists）

random_active_scientist

与其他 random_效果相同，但仅用于可用科学家（active scientists）

injure_scientist_for_days

```
Injure a scientist for x amount of days to a scientist character in scope.
ex:
my_character = {
    injure_scientist_for_days = 12
}
```

add_breakthrough_progress

```
Add breakthrough progress to one specialization or all for a country scope.
The value can either be an absolute value or a script constant.

#### Example
Adding 3 breakthrough points to land specialization:
```
add_breakthrough_progress = {
    specialization = specialization_land
    value = 3
}
```
```

```
Adding -1 breakthrough points to all specializations:
```
add_breakthrough_progress = {
    specialization = all
    value = -1
}
```
Adding the value of the script constant `sp_breakthrough_progress.medium` to all specializations:
```
add_breakthrough_progress = {
    specialization = all
    value = sp_breakthrough_progress.medium
}
```
```

mark_technology_tree_layout_dirty

```
Forces the refresh of the hidden technologies for the scoped country
mark_technology_tree_layout_dirty = yes
```

add_design_template_bonus

```
Add free bonus design discount to given types with a set of uses.
The value for uses and cost_factor can either be an absolute value or a script constant.
Can use several equipment types, where 1 is mandatory

#### Example
Adding 40% discount to an equipment type:
```
add_design_template_bonus = {
    uses = 1
    cost_factor = 0.4
    equipment = light_tank_flame_chassis_0
    name = light_flame_chassis_loc
}
```
Adding 40% discount to an equipment type and archetype with scripted constant:
```
```
```

```
add_design_template_bonus = {
    uses = 2
    cost_factor = cost.high
    equipment = light_tank_flame_chassis_0
    equipment = light_tank_chassis
}
```

add_raid_history_entry

作用域为突袭行动

```
Add history entry to a raid.
Example:
add_raid_history_entry = yes/no
```

add_equipment_bonus

作用域为国家

```
Adds the specified equipment bonuses to the country. As description the given loc key or the name of given special project will be used. Same usage as in Ideas/National spirits.
Example:
add_equipment_bonus = {
    project = FROM # Optional special project scope for using special project name. If not set, the name will be used.
    bonus = {
        armor = { # Type of equipment
                    armor_value = 3 # Bonus to apply to the stats of the equipment type
                    soft_attack = 3
                    instant = yes # Optional. Default no. If true, the bonus will be applied immediately. Otherwise it will be applied only on new equipment variant creation.
        }
        small_plane_naval_bomber_airframe = {
                    air_range = 0.1 naval_strike_attack = 0.1
        }
    }
}

add_equipment_bonus = {
```

```
    name = SUPER_BONUS_NAME # Optional loc key to use as name.
    bonus = {
        small_plane_naval_bomber_airframe = {
                    air_range = 0.1 naval_strike_attack = 0.1
        }
    }
}
```

add_contested_owner

```
Adds a contested owner to a state.
The effect can be used either from a country or a state scope and accepts the
other as parameter.
The effect is localized with a localization environment containing `Country` and
`State`.

### Example
The following example has the same end result and localization.
```
42 = {
    add_contested_owner = GER
}
GER = {
    add_contested_owner = 42
}
```
Standard scope accessors can also be used:
```
Assuming current scope is a state and FROM is a country scope
add_contested_owner = FROM
```
```

activate_shine_on_focus

```
Activates the shine effect on the focus with the given id. Focuses that are
completed cannot have an activated shine effect.

Note that tooltips are only shown in debug mode.

### Example:
```
```
```

```
activate_shine_on_focus = GER_prioritize_economic_growth
```

deactivate_shine_on_focus

```
Deactivate the shine effect on the focus with the given id. The current focus
cannot have it's shine effect removed.

Note that tooltips are only shown in debug mode.

### Example:
```
deactivate_shine_on_focus = GER_prioritize_economic_growth
```
```

complete_special_project

```
Complete a special project for the country in scope.
This effect will not take into account the current state of the project tree and
will allow you to unlock a project even if the one before is not unlocked.
Since the project is not completed within a facility, the facility state and
scientist effects are NOT applied.
ex:
SOV = { complete_special_project = sp:my_project }
SOV = { complete_special_project = var:my_project_var }
SOV = { complete_special_project = PREV } # accepts variables and keywords
SOV = {
    complete_special_project = {
        # project, scientist, state accepts variables and keywords.
        project = sp:my_project
        scientist = my_scientist # Optional if no iteration_output, default to
current scientists on the project if active otherwise to none
        state = my_state # Optional if no iteration_output, default to current
state of the project if active otherwise to none
        iteration_output = { # Can be a single reward or reward = option, if it
contains a multiple option choice but no option specified the default will be
used. The reward must be available to the project
            my_reward
            my_other_reward # multiple choice, chose the default
            my_third_reward = my_option_1 # Specified option to use
        } # Optional amount of iteration rewards
```

```
        show_modifiers = no # Optional, default = yes
    }
}
```

generate_scientist_character

```
Generate a new character with a scientist role and recruit it in the country in scope.
Examples:
SOV = {
    generate_scientist_character = {
        portrait = GFX_portrait # optional - random portrait by default
        portrait_tag_override = CHI # optional - accepts variable and keyword - only relevant if using random portrait - by default use country in scope
        gender = male / female # optional - by default random gender
        skills = {
            # optional array
            # same format as in scientist role in character DB
            # by default all skills are at 1
            specialization_token = 2
        }
        traits = { trait_token } # optional array
    }
}
```

add_scientist_role

```
Add scientist role to a character. The character can come from the scope or from an input parameter.
The scientist role format is the same as in the character DB.
Except the visible trigger - a scientist role created via effect cannot have triggers.
Examples:
# From character scope
my_character = {
    add_scientist_role = {
        scientist = {
            desc = desc_loc_key # Optional
            traits = { scientist_trait_token ... } # Optional
            skills = { specialization_token = 2 ... }
            # cf. game/common/characters/_documentation/md for full explanation
```

```
        }
    }
}

# From country scope
SOV = {
    add_scientist_role = {
        character = my_character / var:my_char_var / PREV # accepts variables and
keywords
        scientist = { ... }
    }
}
```

remove_scientist_role

```
Remove the scientist role from a character.The character can come from the scope
or from an input parameter.
The scientist role format is the same as in the character DB.
Except the visible trigger - a scientist role created via effect cannot have
triggers.
Examples:
# From character scope
my_character = {
    remove_scientist_role = yes
}

# From country scope
SOV = {
    remove_scientist_role = {
        character = my_character / var:my_char_var / PREV # accepts variables and
keywords
    }
}
```

random_allied_country

与其他 random_效果相同

every_allied_country

与其他 every_效果相同

raid_damage_units

```
Damage the units performing the raid in scope (the attackers inflict losses).

Damage is applied to ground units while damage to plane is defined as the amount
of planes lost.
If 'ratio = yes', then all damage / losses are applied as a fraction of the
current amount.
For units, damage can be defined through one value 'damage' or separately through
'org_damage' and 'str_damage'

ex:

# Apply 50% damage to units
raid_damage_units = {
    damage = 0.5
    ratio = yes
}

# Apply 10 strength loss and 20 organization loss to units
raid_damage_units = {
    org_damage = 20
    str_damage = 10
}

# Lose 40% of all planes
raid_damage_units = {
    plane_loss = 0.4
    ratio = yes
}

# Lose 5 planes
raid_damage_units = {
    plane_loss = 5
}
```

raid_add_unit_experience

```
Give experience to the units performing the raid (raid instance scope).
```

```
Will give experience to any type of unit assigned to the raid, e.g. divisions or air wings.
The value defines the progress towards the max level, e.g. 0.2 = gain 20% of the experience needed to reach max level.

Can use either an explicit value or a variable

ex.
raid_add_unit_experience = 0.2
```

promote_officer_to_general

```
Promote the officer of the division to a general.
Example:
promote_officer_to_general = yes # yes/no is ignored
```

store_ref

可能用于 debug 过程，无作用范围

remove_contested_owner

````
Removes a contested owner to a state.
The effect can be used either from a country or a state scope and accepts the other as parameter.
The effect is localized with a localization environment containing `Country` and `State`.

### Example
The following example has the same end result and localization.
```
42 = {
    remove_contested_owner = GER
}
GER = {
    remove_contested_owner = 42
}
```
Standard scope accessors can also be used:
```
### Assuming current scope is a state and FROM is a country scope
````
````

```
remove_contested_owner = FROM
```

add_project_progress_ratio

```
Add progress to the project's prototype phase.
The input value is a ratio of the total needed progress to complete the special
project, i.e. a decimal number between -1 and 1.
ex:
sp:my_project = {
  add_project_progress_ratio = 0.1
  add_project_progress_ratio = var:my_var
}
```

custom_override_tooltip

参考【绑定本地化案例】以查看绑定本地化如何生效

```
Executes the provided effects but with a custom tooltip surpressing all tooltips
from all other effects inside this block.
The custom tooltip support bindable localization

### Examples
```
custom_override_tooltip = {
    tooltip = MY_TOOLTIP # Simple loc key tooltip
    <other effects>
}
```
```

complete_prototype_reward_option

作用域为特殊工程

```
Complete a prototype reward option for the project in scope
The effect will respect the fire only once and allowed property of prototype
rewards.
ex:
complete_prototype_reward_option = {
    prototype_reward = my_reward
    prototyp_reward_option = my_option # Optional, if multiple choice use default
one if not set
    show_modifiers = yes # Yes if the effects of the prototype reward should be
shown (default no)
```

```
}
```

add_unit_bonus

```
Adds permanent subunit and subunit category bonuses for country.

Example:
add_unit_bonus = {
    category_light_infantry = { # Subunit category bonuses
        soft_attack = 0.05
    }

    cavalry = { # Subunit bonuses
        soft_attack = 0.05
        hard_attack = 0.05
    }
}
```

construct_building_in_random_province

```
Set facility level in a random province of state and country scope.
ex:
GER = {
    65 = {
        construct_building_in_random_province = {
            land_facility = 1
        }
    }
}
```

新增修正:

```
river_crossing_factor_against = army
scientist_breakthrough_bonus_factor = character
scientist_research_bonus_factor = character
scientist_xp_gain_factor = character
female_random_scientist_chance = country
production_speed_facility_factor = country
```

```
resource_trade_cost_bonus_per_factory = country
special_project_facility_supply_consumption_factor = country
special_project_speed_factor = country
thermonuclear_production = country
thermonuclear_production_factor = country
underway_replenishment_convoy_cost = country
underway_replenishment_range = country
naval_commando_raid_distance = naval
female_random_scientist_chance = scientist
state_production_speed_facility_factor = state
```

动态修正:

<Building>_max_level_terrain_limit = country

<SpecialProject>_speed_factor = country
支持范围：特殊工程，科研专精，特殊工程标签

<Technology>_cost_factor = country
支持范围：科技类别

state_<Building>_max_level_terrain_limit = state

新增条件:

any_scientist

与其他 any_条件用法相同

all_scientists

与其他 all_条件用法相同

has_scientist_level

```
Checks if the scientist of the character in scope matches the skill level
condition for a specialization. Supports < > = operators.
level = <int>
specialization = <specialization_token>
ex:
my_character = {
```

```
    has_scientist_level = {
        level > 2
        specialization = specialization_nuclear
    }
}
```

`is_active_scientist`

作用域为角色

```
Checks if the scientist of the character in scope is assigned to a project
is_scientist_active = <bool>
ex:
my_character = {
    is_scientist_active = yes
    is_scientist_active = no
}
```

`has_project_flag`

作用域为工程

与其他标记检测（flag check）条件用法相同

`any_active_scientist`

与其他 any_条件用法相同

`all_active_scientist`

与其他 all_条件用法相同

`is_scientist_injured`

作用域为角色

```
Checks if the scientist of the character in scope is injured
is_scientist_injured = <bool>
ex:
my_character = {
    is_scientist_active = yes
}
```

`has_breakthrough_points`

```
Checks if the country in scope has enough breakthroughs within a given
specialization.
specialization = <specialization_token>
```

```
value = <point>
ex:
GER = {
    has_breakthrough_points = {
        specialization = specialization_nuclear
        value = 1
    }
}
```

has_naval_invasion_against_state

```
Check if the scoped country has a naval invasion against the specified state.
Example 1:
has_naval_invasion_against_state = <STATE_ID>
Example 2:
has_naval_invasion_against_state = {
    state = <STATE_ID>
    preparation > 0.0  # (optional: preparation percentage, with a default value
of 0.0)
    activated = no   # (optional: if set, also check if invasion is activated or
not)
}
```

any_state_in

```
check if any state in the given category meets the trigger.
tooltip=key can be defined to override title.
The trigger takes one of the followings:
    array: an array of states.
    continent: A continent.
    ai_area: The id of an area.
    strategic_region: The id of an region.

* Note that no default tooltip is available for array and ai_area.
Example:
Check if the trigger is valid in any state in a continent:
any_state_in = {    conteinent = europe
    FOO_TRIGGER = BAR
}
```

scope_exists

```
Check if the current scope exist.
This differ from for example exists that checks if the country of the scope
exists.
This checks if the scope for the country exists and the other if the country
itself exists in the game.
Note that variable scopes are always valid scopes.
Example:
DEN = { exists = yes } # Should always be true since DEN is always a valid scope
sp:sp_land_flamethrower_tank = {
    character = {
        scope_exists = yes
    }
} # True if the project has an assigned scientist.
var:my_var = {
    scope_exists = yes # Always true since variables are always valid scopes
}
```

equipment_cost

```
check cost of equipment production line
```

is_special_project_completed

```
Checks if the country in scope has completed the special project in input.
ex:
SOV = {
    is_special_project_completed = sp:my_project
    is_special_project_completed = var:my_project_var
    is_special_project_completed = PREV # accepts variables and keywords
}
```

is_special_project_being_researched

```
Checks if the country in scope is currently researching the special project in
input.
ex:
SOV = {
    is_special_project_being_researched = sp:my_project
    is_special_project_being_researched = var:my_project_var
    is_special_project_being_researched = PREV # accepts variables and keywords
}
```

has_market_access_with

```
Check if the country has market access with another country. Example:
has_market_access_with = GER
```

has_officer_name

推断作用域为部队单位

```
checks if division has an officer with the provided name key.
Examples:
has_officer_token = FIN_nikke_parmi
```

has_artillery_ratio

作用域为参战者

```
Check that ratio of atrillery battalions in the composition of a side of
combating troops are over a certain level.
For example:
has_artillery_ratio > 0.1
```

has_unit_type

作用域为参战者

```
Check if the combatant has at least one of the provided unit types.
For example:
has_unit_type = amphibious_mechanized
```

province_vp

作用域为参战者

```
Check if the victory points of the combatants province is larger or less than the
provinded amount.
For example:
province_vp > 2
province_vp < 3
```

has_shine_effect_on_focus

```
Check if country has shine effect on focus (either manually achieved or by being
worked on).

Note that tooltips are only shown in debug mode.

### Example
```
```
```

```
has_shine_effect_on_focus = GER_prioritize_economic_growth
```

custom_override_tooltip

参考【绑定本地化案例】

```
An `AND` trigger that has an overriden custom tooltip.
A positive tooltip can be set with `tooltip` and the tooltip to be used inside a
NOT can be set with `not_tooltip`.
If no positive tooltip is provided and the root key is a localization key (not a
formatter, see [formatted
localization](script_concept_documentation.md#formatted_localization)),
then a negative tooltip will be generated by appending `_NOT` to the root
localization for the positive tooltip.
Both tooltip and `not_tooltip` are [bindable
localizations](script_concept_documentation.md#bindable_localization).

### Examples
```
custom_override_tooltip = {
    tooltip = MY_TOOLTIP # Simple loc key tooltip
    not_tooltip = MY_TOOLTIP_NOT
    <other triggers>
}
```
```
custom_override_tooltip = {
    tooltip = MY_TOOLTIP
    # Implicit:
    #not_tooltip = MY_TOOLTIP_NOT
    <other triggers>
}
```
```

has_contested_owner

```
Checks if a state has the specified country as a contested owner.
The trigger can be used either from a country or a state scope and accepts the
other as parameter.
The trigger is localized with a localization environment containing `Country` and
`State`.
```

```
### Example
The following example has the same end result and localization.
```
42 = {
    has_contested_owner = GER
}
GER = {
    has_contested_owner = 42
}
```
Standard scope accessors can also be used:
```
Assuming current scope is a state and FROM is a country scope
has_contested_owner = FROM
```
```

fighting_army_strength_ratio

```
Compares the total army fighting strength between the scope country and the one set with 'tag'

Example 1:
fighting_army_strength_ratio = {
    tag = TAG
    ratio > 0.7 # can be '<','>' or '='
}

Example 2:
fighting_army_strength_ratio = {
    tag = TAG
    ratio > VARIABLE # can be '<','>' or '='
}
```

has_scientist_specialization

```
Checks if the country in scope has a scientist with a skill level of at least 1 in specialization.
ex:
SOV = {
    has_scientist_specialization = specialization_nuclear
```

```
}
```

has_facility_specialization

```
Checks if the country in scope has a facility with specialization.
ex:
SOV = {
    has_facility_specialization = specialization_nuclear
}
```

can_construct_building

```
Checks if the country (as ROOT) and state in scope can build a building in the
state.
ex:
GER = {
    65 = {
        can_construct_building = land_facility
    }
}
```

num_nukes_being_dropped

```
total number of nukes that are currently ready to be dropped
```

num_nukes_left_to_drop

```
number of nukes left to drop during this game tick (only useful in-between nuke
drops, like in on_nuke_drop on-action, for example)
```

新增变量:

作用域为国家的变量:

total_equipment_produced_crossing_support_vehicle

- 经制造的支援车辆 typecrossing_support_vehicle 总装备数

total_equipment_produced_armored_recovery_vehicle

- 经制造的装甲救护车 typearmored_recovery_vehicle 总装备数

num_of_civilian_factories_in_cores

- 统计当前国家所控制的核心领土内的民用工厂数量，形如@Tag <Tag | ROOT | my_var>，例如 num_of_civilian_factories_in_cores@GER

num_of_military_factories_in_cores

- 统计当前国家所控制的核心领土内的军用工厂数量，形如@Tag <Tag | ROOT | my_var>，例如 num_of_military_factories_in_cores@GER

num_of_naval_factories_in_cores

- 统计当前国家所控制的核心领土内的海军船坞数量，形如@Tag <Tag | ROOT | my_var>，例如 num_of_naval_factories_in_cores@GER

total_equipment_produced_sam_missile

- 经制造的地空导弹 typesam_missile 总装备数

total_constructed_gun_emplacement

- 经建造的火炮阵地 gun_emplacement 总建造数

total_equipment_produced_clearance_vehicle

- 经制造的工程车 typeclearance_vehicle 总装备数

total_equipment_produced_ballistic_missile

- 经制造的弹道导弹 typeballistic_missile 总装备数

total_equipment_produced_land_cruiser

- 经制造的陆地巡洋舰 typeland_cruiser 总装备数

total_equipment_produced_emplacement_gun_ammo

- 经制造的重炮弹药 typeemplacement_gun_ammo 总装备数

total_equipment_produced_missile_launcher

- 经制造的导弹发射器 typemissile_launcher 总装备数

num_nukes_being_dropped

- 准备好能够发射的核武器件数

num_nukes_left_to_drop

- 该游戏刻剩余可以发射的核武器数量（仅在两次核爆之间起效，如用在 on_action 语句 on_nuke_drop 中）

作用域为特殊工程的变量:

全部以特殊工程为作用域的变量前都需要添加“var:”。

facility_state

- 特殊工程研究场所存在的地区

facility_province_id

- 特殊工程研究场所存在的省份

scientist

- 正在研究特殊工程的科学家

作用域为突袭行动的变量

全部以突袭行动为作用域的变量前都需要添加“var:”。此外，如果需要返回 ROOT 或作用域外的上一级，都需要添加 ROOT。

actor_country
- 发起突袭行动的国家

victim_country
- 突袭行动的目标国家

target_state
- 突袭行动发生的地区

target_province
- 突袭行动发生的省份

定义:

被移除的定义:

```
NUKE_MIN_DAMAGE_PERCENT = 0.1,                  -- Minimum damage from nukes as a
percentage of current strength/organization
NUKE_MAX_DAMAGE_PERCENT = 0.9,                  -- Minimum damage from nukes as a
percentage of current strength/organization
STRATEGIC_BOMBER_NUKE_AIR_SUPERIORITY       -- How much air superiority is needed
for a tactical bomber to be able to nuke a province
ANNEX_RATIO = 0.5,
-- How many railway guns will be transferred on annexation
HOURS_BETWEEN_REDISTRIBUTION = 24,
-- Number of hours between redistribution of attached railway guns, tracked per
army
PLAN_FRONT_SECTION_MAX_LENGTH = 18,
-- When a front is longer than this it will be split in two sections for the AI
```

```
PLAN_FRONT_SECTION_MIN_LENGTH = 10,
-- When two front sections together are this short they will be merged for the AI
DIVISION_DESIGN_WEIGHTS
DIVISION_DESIGN_MANPOWER_WEIGHT = 0.005,
DIVISION_DESIGN_STOCKPILE_WEIGHT = 0.01,
DIVISION_DESIGN_COMBAT_WIDTH_WEIGHT = -1.0,
-- This score is reduced the higher width is when comparing pure changes with no
target
DIVISION_DESIGN_COMBAT_WIDTH_TARGET_WEIGHT = -200.0,
-- This score is reduced the farther the width is from the target width (if set)
COMBINED_ARMS_LEVEL = 1,
-- 0 = Never, 1 = Infantry/Artillery, 2 = Go wild
INVASION_DISTANCE_RANDOMNESS = 300,
-- This higher the value, the more unpredictable the invasions. Compares to
actual map distance in pixels.
ARMY_LEADER_ASSIGN_KEEP_LEADER = 500,
-- Score for keeping the leader if already assigned
```

Changed Defines:

```
FEMALE_UNIT_LEADER_BASE_CHANCE
 - added scientist chance
MISSION_COMMAND_POWER_COSTS
added barrage, sam
MISSION_FUEL_COSTS
added barrage, nuclear, sam

* `RAILWAY_GUN_POSSIBLE_RANGES = { 30, 15, 45 }`
    -- Possible values for railway gun range in pixel.
    -- For optimization reasons, they are listed here and equipment DB must use
one of those.
    -- when writing railway gun in equipment, use the index in this array
    -- the first value in array is the default value
—----------------------------------

HOUR_BAD_COMBAT_REEVALUATE = 48,
-- if we are in combat for this amount and it goes shitty then try skipping it
```

```
–Changed into–
CANCEL_COMBAT_DISADVANTAGE_RATIO = 1.5,
-- If the enemy's advantage ratio over us during (normal) combat is more than
<value>, allow canceling the attack
CANCEL_COMBAT_MIN_DURATION_HOURS = 48,
-- Only allow canceling (normal) combat if at least <value> hours have passed
CANCEL_INVASION_COMBAT_DISADVANTAGE_RATIO = 3.5,
-- If the enemy's advantage ratio over us during invasion combat is more than
<value>, allow canceling the attack
CANCEL_INVASION_COMBAT_MIN_DURATION_HOURS = 120,
    -- Only allow canceling invasion combat if at least <value> hours have passed
–--------------------------------



FAILED_INVASION_AVOID_DURATION = 60, -- after a failed invasion, AI will down-
prioritize invading the same area again for this number of days
–Changed into–
INVASION_TARGET_DISTANCE_DENOMINATOR = 1000,
-- When selecting invasion target, divide this with (pixel) distance to get
distance score factor. (Doesn't really affect the relative scoring, but it
affects the linearity of the score function.)
INVASION_TARGET_NO_PORT_FACTOR = 0.3,
-- When selecting invasion target, multiply score with this if the target has no
port
INVASION_TARGET_TRUNCATION_SELECT_THRESHOLD = 0.6,
-- When selecting invasion target, use this threshold for truncation selection.
(1.0 means select highest scored target, 0.0 means select randomly from all
possible target, 0.5 means select randomly from all targets with more than 50 %
of highest score)
INVASION_TARGET_PRIO_NOT_ENEMY_FACTOR = 0.17,
-- When calculating priority for an invasion, factor the score with this if the
target is not an actual enemy.

–--------------------------------
PLAN_PROVINCE_LOW_VP_IMPORTANCE_AREA = 2.0,
-- Used when calculating the value of defense area in the battle plan system
PLAN_PROVINCE_MEDIUM_VP_IMPORTANCE_AREA = 5.0,
 -- Used when calculating the value of defense area in the battle plan system
```

```
PLAN_PROVINCE_HIGH_VP_IMPORTANCE_AREA = 10.0,
 -- Used when calculating the value of defense area in the battle plan system
PLAN_PROVINCE_CAPITAL_IMPORTANCE_AREA = 50.0,
-- Used when calculating the balance of defense area in the battle plan system

—Changed into—

PLAN_PROVINCE_LOW_VP_DEFENSE_THRESHOLD = 2.0,
-- For area defense VP orders, what are the thresholds for "low", "medium" and
"high" VP values
PLAN_PROVINCE_MEDIUM_VP_DEFENSE_THRESHOLD = 8.0,
 -- see above
PLAN_PROVINCE_HIGH_VP_DEFENSE_THRESHOLD = 25.0,
-- see above
PLAN_PROVINCE_LOW_VP_DEFENSE_IMPORTANCE = 2.0,
-- For area defense VP orders, use this value for relative importance
PLAN_PROVINCE_MEDIUM_VP_DEFENSE_IMPORTANCE = 5.0,
-- see above
PLAN_PROVINCE_HIGH_VP_DEFENSE_IMPORTANCE = 10.0,
 -- see above
PLAN_PROVINCE_CAPITAL_DEFENSE_IMPORTANCE = 50.0,
 -- For area defense VP orders, boost importance value with this if it's the
capital
```

新增定义:

```
New defines:
* `THERMONUCLEAR_BOMB_DROP_WAR_SUPPORT_EFFECT_MAX_INFRA`    -- Reduce enemy
national war support on nuking a province, the value scales with infrastructure
up to this number
* THERMONUCLEAR_BOMB_DROP_WAR_SUPPORT_EFFECT_MAX_VP -- War support will be scaled
down if there's less VP than this in the province
* `MAX_MIL_FACTORIES_VISIBLE_FOR_MIL_EQUIPMENT_LINE`
* `SAM_MISSION_SUPERIORITY` -- How much air superiority each SAM mission gives
per rocket wing performing SAM missions.
* `PLAN_AREA_DEFENSE_FACILITY`-- Used when calculating the value of defense area
provinces for the battle plan system
```

```
* `MISSILE_LAUNCHER_CAPACITY`-- The number of missiles per slot
* `MISSILE_LAUNCHER_SLOTS`-- The number of missile slots a missile launcher unit can have
* `UNDERWAY_REPLENISHMENT_PRIORITY`-- Default convoy priority for underway replenishment
* `UNDERWAY_REPLENISHMENT_RANGE_FACTOR`-- bonus factor applied to task force's range when underway replenishment is activated (e.g. 0.2 means +20%)
* `UNDERWAY_REPLENISHMENT_CONVOY_COST_PER_FUEL`-- Cost in convoys for underway replenishment multiplied by max daily fuel consumption (rounded up)
* `MIN_SHIPS_FOR_HIGHER_SHIP_RATIO_PENALTY`-- the minimum fleet size in ships that a fleet must be before having the large fleet penalty applied to them
* `GUN_EMPLACEMENT_MIN_ASSIGN_SCORE`-- Minimum total score for region to be considered for gun emplacement air missions
* `GUN_EMPLACEMENT_MIN_PRIO_ASSIGN_SCORE`-- Minimum total score for region to be considered for critical gun emplacement air missions
* `GUN_EMPLACEMENT_ASSIGN_SCORE_REDUCTION_PER_ASSIGNMENT`-- each assigned gun emplacement reduces the score of a region by this amount
* `REMOVE_OBSOLETE_TEMPLATE_DAYS`-- Remove obsolete and unused templates if they have been marked as obsolete for x days. Non-positive value means "never remove".
LAND_DEFENSE_RAID_IMPORTANCE = 500,
-- Strategic importance of detected raids targetting us
LAND_DEFENSE_FIGHERS_PER_RAID = 100,
-- Amount of air superiority planes requested per detected raid targetting us
LAND_DEFENSE_INTERCEPTORS_PER_RAID = 100,
-- Amount of interceptor planes requested per detected raid targetting us
LAND_DEFENSE_SAM_MISSILE_IMPORTANCE_FACTOR = 0.2,
    -- Importance factor of using sam missiles for regions strategic importance. Higher value will increase the usage
LAND_COMBAT_MISSILE_IMPORTANCE_FACTOR = 1.5,
-- Importance factor of using missiles for regions strategic importance. Higher value will increase the usage
* `CONSTRUCTION_PRIO_SUPPLY_BUILDING`-- base prio for supply buildings (supply hubs, ports) in the construction queue
* `MIN_INVASION_ORG_FACTOR_TO_EXECUTE`-- ai will only activate invasions if average org factor is above this
* `ARMY_LEADER_ASSIGN_KEEP_CURRENT_LEADER_FACTOR` -- Boosts the score for keeping the current leader. Value > 1.0 favors the current leader.
* `ARMY_LEADER_ASSIGN_DONT_STEAL_OTHER_FACTOR` -- Reduces the score for leaders assigned elsewhere. Value < 1.0 discourages reassigning these leaders.
```

```
* `AREA_DEFENSE_SETTING_BORDERS` bools
* `AREA_DEFENSE_SETTING_FACILITY` bools

RAID_MIN_INTEL_FOR_WARNING_ON_LAUNCH = 0.1,
-- how much intel (of the relevant type) is needed to show a warning when raid is launched
RAID_MIN_INTEL_FOR_WARNING_HALFWAY_TO_LAUNCH = 0.5,
-- how much intel (of the relevant type) is needed to show a warning halfway through preparation
 --     (this limit is a dummy value only used for communicating the role of intel in the intel ledger )
--  (in reality, detection scales linearly with intel. 70% intel = detection at 30% preparation, 50% intel = detection at 50% preparation, etc.
RAID_MIN_INTEL_FOR_WARNING_EARLY_PREPARATION = 0.8,
 -- how much intel (of the relevant type) is needed to show a warning early in the preparation
  --     (this limit is a dummy value only used for communicating the role of intel in the intel ledger )
--  (in reality, detection scales linearly with intel. 70% intel = detection at 30% preparation, 50% intel = detection at 50% preparation, etc.
NUCLEAR_RAID_CATEGORY_NAME = "nuclear_raids",
-- The raid category to activate when clicking on the "nuclear" mission button for a rocket
```

- 全部 aifc(AI 集中突破)相关的定义均为新添加的
- 全部特殊工程与突袭相关定义均为新添加的

AI 变更:

改进的 AI 决策行为:

《钢铁雄心 IV》的 AI 现在将根据其 AI 政治权力中的可选选项正确保存和花费政治点数。您可以在游戏中输入 imgui show ai_pp_spend 来查看此内容。

所有政治权力决策的 AI 优先级现在直接来自 ai_pp_spend 队列中的队列。AI 将不再随机选择决策，现在它将有一个为特定决策或其他决策保存的概念。

更改的理由是它将使 AI 以某种适当的确定性水平做出决策。设计师现在可以根据 ai_will_do 或其他游戏状态因素编写具有适当优先级的决策脚本。

具体说明:

- AI 不再会任意选择决议来完成，除非决议拥有 AI 权重>0 并花费 0 政治点数。
- 全部花费政治点数的机制如今都会互相比较权重（如雇佣顾问，执行决议，修改法案等）。其中部分权重通过游戏状态参数定义，而另一部分可以通过脚本进行修改（ai_will_do 语句）。
- 花费较低的决议不会被选中，除非他们已经处于决议执行队列的最上方。
- 花费多种资源的决议仍然会造成问题，因为 AI 仍旧只能识别政治点数花费。你需要通过 ai_will_do 语句以确保 AI 不会执行错误的决议从而导致游戏失败。

你仍旧可以使用传统的政治点数决定倾向机制，不过这一机制并不会长期存在，而会逐渐被移除。

改进 AI 突破行为:

AI 现在会在进行突破行动时更有逻辑性，如在特定区域集中部队以突破敌军防线。AI 现在会将部队按照特定数据进行分配。如果某单位突破数值更高，AI 会操纵他们在特定区域进行突破以合围敌方口袋内的部队，或在敌方防线上取得突破。你能够通过 imgui 的某条指令观察 AI 的表现效果：imgui show ai_force_concetration。

另外，相当一部分新 AI 策略如今已经对模组制作开放，你可以使用它们以修改 AI 逻辑、操纵 AI 行为并改善 AI 集中部队的机制。

v1.15.x & "Götterdämmerung" HOI Migration/DLC Update Guide 附件原文

Author(s): AngriestBird of Millennium Dawn, BiscuitCookie of Equestria at War, SpicyAlfredo of Road to 56, and et al.

This document details migrating your mod from HOI v1.14.x to v1.15.x. It will also include information regarding new mechanics, changes, features, and other important information regarding the new DLC and the patch for Hearts of Iron IV.

Initial Migration Issues:

Immediate CTD Issues:

- State files that have accented characters in the file name need to be changed to English characters
- There is a new right_vertical_slider container that is required in the core.gui

Potential CTD Issues:

If your mod still has CTD issues after fixing the Immediate CTD problems or is experiencing new CTD's after modding please check for the possible CTD issues below:

- If the '_economic.txt' file is overwritten in common/ideas, you may now need to include these new ELSE statements inside of the available block of ideas if an IF statement is inside of an OR statement like the example below:

```
OR = {
    has_idea = war_economy
    has_idea = tot_economic_mobilisation
    if = {
        limit = {
            original_tag = GER
        }
        has_idea = totaler_krieg_economy
    }
    else = {
        hidden_trigger = { always = no } #NEED
    }
}
```

- If air.tx file in common\units is overwritten. You may need to add a reference to "mega_carrier_air_wing_size = x" in the unit attributes.
- If air.tx file in common\units is overwritten. You may need to add references to the new Missile and Special project techs in this file.
- Currently when modding raids, the game needs *at least* one raid that targets a facility. If you do not wish to have any raids that target a facility simply make this blank raid in one of your raid files:

```
types = {

    stop_crashing_my_game = {

        target_type = {
            building = {
                tags = facility
            }
        }
    }

}
```

- Overridden Defines may need to be updated, see defines below for changes.

Map/Building Changes:

- The map files rocketsites.txt/airports.txt are now deprecated. You will need to create entities for all buildings and then place their placement in the state. If you wish to move these buildings you will need to move them in nudger.
- There are several new buildings and a new building definition known as **Spawn Points.**
- **Spawn Points** have to be placed in conjunctions for air bases and rocket sites to work appropriately.

- Buildings now dynamically populate depending on whether they're enabled or not in the construction UI.
- 'landmark_spawn' and 'dam_spawn' locations may not auto-populate as desired in the buildings.txt file if 'Random All States' is used. They may have to be placed manually in the desired state using the nudger by selecting those building types individually.

Other GUI Changes:

Duplicative container names are now reported in the error.log. It is important to check these and make sure there is no issue. This is only relevant when they are in the same file/screen. Nested similar name containers do not error.

Files you may need to resolve issues for in your mod's UI depending on changes:

- airselectionview.gui
    - rocket_selection_view is now included as a container for the support of the new missile system. Needed otherwise you'll hit crashes.
- Countrytechtreeview.gui
    - "research_groups_grid" in "technology_unit_statlist_item_equipment" needs to be removed as it will cause crashes if it persisted in your GUI before.
    - New containers and other updates are also present here in the container
- mapicons.gui
    - Required the addition of the raid filters element to ensure functionality for the raid dynamic filters
- naviesview.gui
    - Underway replenishment is now a feature. This will need a new button in the UI so be sure to check here.
- stateview.gui
    - Adds a new state building modifiers section
    - Adds a new dynamic modifier section to help fix the weird push-off
- unitview.gui
    - divisions_summary_item was moved from unitview.gui to its own file
- alerts.gfx
    - If you have any custom alerts in the standard alert strip this was expanded by 8 more alert types

---

Script/Code Changes:

Scripted Localization Changes

- Inline localization is no longer valid in GUIs and in some other screens. It must have a localization string attached to the scripted loc localization_key to work properly.

  Old Example from v1.14.x:
  ```
     text = { trigger = { check_variable = { pos_array^1 = 12 } }
          localization_key = "[?party_pop_array^12|%1]"
      }
  ```
  New Working Version with v1.15.x:
  ```
     text = { trigger = { check_variable = { pos_array^0 = 14 } }
          localization_key = num_partypos_1_14
      }
  ```

Effect Changes

- **custom_effect_toolip** now allows bindable loc
- **add_province_modifier** can add a modifier temporarily by setting `days =` field, accepts ints and variables
- **remove_building** now accepts the `tags = <buildingtag>` or `tags = { <buildingtag> }` field tags are set in the building file for buildings
- **create_unit** adds `divisional_commander_xp = <int?>` field gives division commander xp on creation
- **damage_building** also adds `tags =` field and the `repair_speed_modifier = <float>` which allows one to set how fast or slow it can repair until it is fully repaired
- **force_update_dynamic_modifier**, **remove_dynamic_modifier** support the special_project scope?
- **launch_nuke** allows to choose **nuke_type =** nuke type used (e.g. nuclear_bomb, thermonuclear_bomb etc.)

- Bindable Localization Example:

  Code for the Localization:

```
add_pink_fluffy_unicorn = 1


custom_effect_tooltip = {
```

```
    localization_key = ADD_UNICORN # Root look key
    UNICORN_AMOUNT = "1"
}

Loc Entry:
ADD_UNICORN:0 "Adds $UNICORN_AMOUNT$ unicorn to your rainbow parade"

Resulting in the following:
"Adds 1 unicorn to your rainbow parade"
```

```
custom_override_tooltip = {
    tooltip = {
        localization_key = MY_TOOLTIP # Root look key
        IMPORTANT_QUESTION = { # ID IMPORTANT_QUESTION in MY_TOOLTIP will get
value:
            localization_key = MEANING_OF_LIFE # Root loc key in
IMPORTANT_QUESTION
            ANSWER = "42" # ID ANSWER in IMPORTANT_QUESTION will get value 42
        }
        JUST_AS_IMPORTANT = OR_NOT # ID JUST_AS_IMPORTANT in MY_TOOLTIP will get
value OR_NOT
    }
    <other effects>
}
```

Trigger Changes:

- seems `custom_trigger_tooltip` has been replaced with `custom_override_tooltip` and is kept as an alias so it now also supports bindable loc

Script Constants:

- The new patch for v1.15.x introduces a new concept called “script constants” which can be found under common/script_constants.

- Functionally they are a way to define simple and complex script values that are reusable in different files in scripts. Only entries that are documented to support script constants can utilize it in their script.

- We can use this to set a value in one area and then manipulate that value from one source of truth in the script_constants.

Example Constant Definition:

```
sp_military_xp_gain = {
    schema = {
            any_key = yes
            data = int
    }
    low = 15
    medium = 30
    high = 50
}
```

Example Usage in Script:

```
country_effects = {
    army_experience = constant:sp_military_xp_gain.medium
    FROM = { set_project_flag = sp_land_generic_reward_army_xp_flag }
}
```

Any place that supports scoped variables will support script constants by prefixing it with `constant:<yourconstantformathere>`
`add_design_template_bonus` and `add_breakthrough_progress` support constants directly without the prefix
in special projects, the `prototype_time` and `complexity` fields also support constants directly.

DLC Locked Features:

DLC Locked features primarily surround the custom content of Raids and Special Projects with some remaining generic.

DLC Locked Content:

- Landmarks
    - This feature can still be used but the vanilla versions of them are DLC locked.
- Custom Raids & Special Projects
    - This includes raids on Big Ben, the Eiffel Tower, etc
    - Re. special projects this will include most of the custom new units such as Land Cruisers, Cruiser Submarines, etc
- Reworked Nuclear System
    - This is adjacent to the missile system with some
- New Missile System
    - Guided_missile_equipment is the old-school version and this is still free. The new missions and missile types like ICBMs, SAMs, IRBMs, and others are all DLC locked and their missions should be assumed locked as well.
- New trees for the nations that were reworked (Alt-History Germany, Austria, Hungary, Belgium, and the Congo)

Free Patch Content:

This section outlines all of the non-DLC content and features that can be used in your modding.

- Special Projects (code and generic ones)
    - The special projects are free to mod and include in your mod without locking content behind DLC unless it uses DLC assets such as the cruiser submarine, ice carrier, new missiles, or otherwise.
- Raids (code and generic ones)
    - The raids are free to mod and include in your mod without locking content behind DLC unless it uses DLC assets such as the new missiles or otherwise.
- Landmark System (no vanilla content)
    - The vanilla landmark system from DLC can be used, however, you cannot use any of the landmarks from vanilla unless you own the DLC or have implemented allow (models are the main point of concern here)
- Improved AI for spearheads
    - AI improvements are just generally available to all.
- Improved Decision System AI
    - This is the AI being better at saving political power for important decisions. There is nothing required to change here and there is no required migration change here.

New Effects:

add_breakthrough_point

Country scope

```
Add breakthrough points to one specialization or all for a country scope.
ex:
    add_breakthrough_points = {
    specialization = <sp_specialization_id>
    value = 3
}
add_breakthrough_points = {
    specialization = all
    value = -1
}
```

raid_reduce_project_progress_ratio

State scope

```
Reduce progress to the special project in state. Root scope is raid instance
scope.
The input value is a ratio of the total needed progress to complete the special
project, i.e. a decimal number between 0 and 1.
ex:
# Root scope is raid
state = {
    raid_reduce_project_progress_ratio = 0.1 # Reduces the project progress by
10%
}
```

random_scientist

Same as other random_ effects

every_scientist

Same as other every_ effects

`create_colonial_division_template`

```
Create a colonial division template for overlord/owner. Available parameters are subject and division_template, where the subject parameter is the country tag for an overlords subject. And the division_template is the regular effect to create a division template.
Example.
In country scope of overlord, E.g. ROOT = ENG
create_colonial_division_template = {
    subject = RAJ # Country tag
    division_template = {
        name = "Infantry Division"
        division_names_group = RAJ_INF_01
        ...
        regiments = {
            infantry = { x = 0 y = 0 }
            infantry = { x = 0 y = 1 }
        }
    }
}
```

`add_scientist_level`

```
Add levels to a special project specialization for a scientist character in scope.
The `level` parameter is a scoped variable

#### Example
```
my_character = {
    add_scientist_level = {
        level = 2 # accepts variables
        specialization = specialization_nuclear
    }
}
```
```

`add_scientist_xp`

```
Add experience to a special project specialization for a scientist character in scope.
```

```
The `experience` parameter is a scoped variable.

#### Example
```
ex: my_character = {
    add_scientist_xp = {
        experience = 2 # accepts variables
        specialization = specialization_nuclear
    }
}
```
```

add_scientist_trait

```
Add a trait to a scientist character in scope.
ex:
my_character = {
    add_scientist_trait = my_trait_token
}
```

`set_project_flag`

same as other set flags only in a project scope

`modify_project_flag`

same as other set flags only in a project scope

`clr_project_flag`

same as other set flags only in a project scope

`every_active_scientist`

same as other every_ effects but for only active scientists

`random_active_scientist`

same as other random_ effects but for only active scientists

`injure_scientist_for_days`

```
Injure a scientist for x amount of days to a scientist character in scope.
ex:
my_character = {
```

```
    injure_scientist_for_days = 12
}
```

add_breakthrough_progress

```
Add breakthrough progress to one specialization or all for a country scope.
The value can either be an absolute value or a script constant.

#### Example
Adding 3 breakthrough points to land specialization:
```
add_breakthrough_progress = {
    specialization = specialization_land
    value = 3
}
```
Adding -1 breakthrough points to all specializations:
```
add_breakthrough_progress = {
    specialization = all
    value = -1
}
```
Adding the value of the script constant `sp_breakthrough_progress.medium` to all specializations:
```
add_breakthrough_progress = {
    specialization = all
    value = sp_breakthrough_progress.medium
}
```
```

mark_technology_tree_layout_dirty

```
Forces the refresh of the hidden technologies for the scoped country
mark_technology_tree_layout_dirty = yes
```

add_design_template_bonus

```
Add free bonus design discount to given types with a set of uses.
The value for uses and cost_factor can either be an absolute value or a script constant.
```

```
Can use several equipment types, where 1 is mandatory

#### Example
Adding 40% discount to an equipment type:
```
add_design_template_bonus = {
    uses = 1
    cost_factor = 0.4
    equipment = light_tank_flame_chassis_0
    name = light_flame_chassis_loc
}
```
Adding 40% discount to an equipment type and archetype with scripted constant:
```
add_design_template_bonus = {
    uses = 2
    cost_factor = cost.high
    equipment = light_tank_flame_chassis_0
    equipment = light_tank_chassis
}
```
```

`add_raid_history_entry`

Raid scope

```
Add history entry to a raid.
Example:
add_raid_history_entry = yes/no
```

`add_equipment_bonus`

country scope

```
Adds the specified equipment bonuses to the country. As description the given loc key or the name of given special project will be used. Same usage as in Ideas/National spirits.
Example:
add_equipment_bonus = {
    project = FROM # Optional special project scope for using special project name. If not set, the name will be used.
    bonus = {
        armor = { # Type of equipment
```

```
                    armor_value = 3 # Bonus to apply to the stats of the
equipment type
                    soft_attack = 3
                    instant = yes # Optional. Default no. If true, the bonus will
be applied immediately. Otherwise it will be applied only on new equipment
variant creation.
        }
        small_plane_naval_bomber_airframe = {
                    air_range = 0.1 naval_strike_attack = 0.1
        }
    }
}

add_equipment_bonus = {
    name = SUPER_BONUS_NAME # Optional loc key to use as name.
    bonus = {
        small_plane_naval_bomber_airframe = {
                    air_range = 0.1 naval_strike_attack = 0.1
        }
    }
}
```

add_contested_owner

```
Adds a contested owner to a state.
The effect can be used either from a country or a state scope and accepts the
other as parameter.
The effect is localized with a localization environment containing `Country` and
`State`.

### Example
The following example has the same end result and localization.
```
42 = {
    add_contested_owner = GER
}
GER = {
    add_contested_owner = 42
}
```
```

```
Standard scope accessors can also be used:
```
Assuming current scope is a state and FROM is a country scope
add_contested_owner = FROM
```
```

activate_shine_on_focus

```
Activates the shine effect on the focus with the given id. Focuses that are completed cannot have an activated shine effect.

Note that tooltips are only shown in debug mode.

### Example:
```
activate_shine_on_focus = GER_prioritize_economic_growth
```
```

deactivate_shine_on_focus

```
Deactivate the shine effect on the focus with the given id. The current focus cannot have it's shine effect removed.

Note that tooltips are only shown in debug mode.

### Example:
```
deactivate_shine_on_focus = GER_prioritize_economic_growth
```
```

complete_special_project

```
Complete a special project for the country in scope.
This effect will not take into account the current state of the project tree and will allow you to unlock a project even if the one before is not unlocked.
Since the project is not completed within a facility, the facility state and scientist effects are NOT applied.
ex:
SOV = { complete_special_project = sp:my_project }
SOV = { complete_special_project = var:my_project_var }
SOV = { complete_special_project = PREV } # accepts variables and keywords
SOV = {
```

```
    complete_special_project = {
        # project, scientist, state accepts variables and keywords.
        project = sp:my_project
        scientist = my_scientist # Optional if no iteration_output, default to
current scientists on the project if active otherwise to none
        state = my_state # Optional if no iteration_output, default to current
state of the project if active otherwise to none
        iteration_output = { # Can be a single reward or reward = option, if it
contains a multiple option choice but no option specified the default will be
used. The reward must be available to the project
            my_reward
            my_other_reward # multiple choice, chose the default
            my_third_reward = my_option_1 # Specified option to use
        } # Optional amount of iteration rewards
        show_modifiers = no # Optional, default = yes
    }
}
```

generate_scientist_character

```
Generate a new character with a scientist role and recruit it in the country in
scope.
Examples:
SOV = {
    generate_scientist_character = {
        portrait = GFX_portrait # optional - random portrait by default
        portrait_tag_override = CHI # optional - accepts variable and keyword -
only relevant if using random portrait - by default use country in scope
        gender = male / female # optional - by default random gender
        skills = {
            # optional array
            # same format as in scientist role in character DB
            # by default all skills are at 1
            specialization_token = 2
        }
        traits = { trait_token } # optional array
    }
}
```

add_scientist_role

```
Add scientist role to a character. The character can come from the scope or from an input parameter.
The scientist role format is the same as in the character DB.
Except the visible trigger - a scientist role created via effect cannot have triggers.
Examples:
# From character scope
my_character = {
    add_scientist_role = {
        scientist = {
            desc = desc_loc_key # Optional
            traits = { scientist_trait_token ... } # Optional
            skills = { specialization_token = 2 ... }
            # cf. game/common/characters/_documentation/md for full explanation
        }
    }
}

# From country scope
SOV = {
    add_scientist_role = {
        character = my_character / var:my_char_var / PREV # accepts variables and keywords
        scientist = { ... }
    }
}
```

remove_scientist_role

```
Remove the scientist role from a character.The character can come from the scope or from an input parameter.
The scientist role format is the same as in the character DB.
Except the visible trigger - a scientist role created via effect cannot have triggers.
Examples:
# From character scope
my_character = {
    remove_scientist_role = yes
}
```

```
# From country scope
SOV = {
    remove_scientist_role = {
        character = my_character / var:my_char_var / PREV # accepts variables and
keywords
    }
}
```

`random_allied_country`

same as random_ effects

`every_allied_country`

same as other every_ effects

`raid_damage_units`

```
Damage the units performing the raid in scope (the attackers inflict losses).

Damage is applied to ground units while damage to plane is defined as the amount
of planes lost.
If 'ratio = yes', then all damage / losses are applied as a fraction of the
current amount.
For units, damage can be defined through one value 'damage' or separately through
'org_damage' and 'str_damage'

ex:

# Apply 50% damage to units
raid_damage_units = {
    damage = 0.5
    ratio = yes
}

# Apply 10 strength loss and 20 organization loss to units
raid_damage_units = {
    org_damage = 20
    str_damage = 10
}
```

```
# Lose 40% of all planes
raid_damage_units = {
    plane_loss = 0.4
    ratio = yes
}

# Lose 5 planes
raid_damage_units = {
    plane_loss = 5
}
```

raid_add_unit_experience

```
Give experience to the units performing the raid (raid instance scope).

Will give experience to any type of unit assigned to the raid, e.g. divisions or
air wings.
The value defines the progress towards the max level, e.g. 0.2 = gain 20% of the
experience needed to reach max level.

Can use either an explicit value or a variable

ex.
raid_add_unit_experience = 0.2
```

promote_officer_to_general

```
Promote the officer of the division to a general.
Example:
promote_officer_to_general = yes # yes/no is ignored
```

store_ref

possibly debugging/unused effect

remove_contested_owner

```
Removes a contested owner to a state.
The effect can be used either from a country or a state scope and accepts the
other as parameter.
The effect is localized with a localization environment containing `Country` and
`State`.
```

```
### Example
The following example has the same end result and localization.
```
42 = {
    remove_contested_owner = GER
}
GER = {
    remove_contested_owner = 42
}
```
Standard scope accessors can also be used:
```
Assuming current scope is a state and FROM is a country scope
remove_contested_owner = FROM
```
```

`add_project_progress_ratio`

```
Add progress to the project's prototype phase.
The input value is a ratio of the total needed progress to complete the special
project, i.e. a decimal number between -1 and 1.
ex:
sp:my_project = {
  add_project_progress_ratio = 0.1
  add_project_progress_ratio = var:my_var
}
```

`custom_override_tooltip`

See bindable localization example for how bindable localization works

```
Executes the provided effects but with a custom tooltip surpressing all tooltips
from all other effects inside this block.
The custom tooltip support bindable localization

### Examples
```
custom_override_tooltip = {
    tooltip = MY_TOOLTIP # Simple loc key tooltip
    <other effects>
}
```
```

```
```

complete_prototype_reward_option

special project scope

```
Complete a prototype reward option for the project in scope
The effect will respect the fire only once and allowed property of prototype rewards.
ex:
complete_prototype_reward_option = {
    prototype_reward = my_reward
    prototyp_reward_option = my_option # Optional, if multiple choice use default one if not set
    show_modifiers = yes # Yes if the effects of the prototype reward should be shown (default no)
}
```

add_unit_bonus

```
Adds permanent subunit and subunit category bonuses for country.

Example:
add_unit_bonus = {
    category_light_infantry = { # Subunit category bonuses
        soft_attack = 0.05
    }

    cavalry = { # Subunit bonuses
        soft_attack = 0.05
        hard_attack = 0.05
    }
}
```

construct_building_in_random_province

```
Set facility level in a random province of state and country scope.
ex:
GER = {
    65 = {
        construct_building_in_random_province = {
            land_facility = 1
        }
```

```
    }
}
```

New Modifiers:

```
river_crossing_factor_against = army
scientist_breakthrough_bonus_factor = character
scientist_research_bonus_factor = character
scientist_xp_gain_factor = character
female_random_scientist_chance = country
production_speed_facility_factor = country
resource_trade_cost_bonus_per_factory = country
special_project_facility_supply_consumption_factor = country
special_project_speed_factor = country
thermonuclear_production = country
thermonuclear_production_factor = country
underway_replenishment_convoy_cost = country
underway_replenishment_range = country
naval_commando_raid_distance = naval
female_random_scientist_chance = scientist
state_production_speed_facility_factor = state
```

Dynamic modifiers:

```
<Building>_max_level_terrain_limit = country

<SpecialProject>_speed_factor = country
supports: special projects, specializations, special project tags

<Technology>_cost_factor = country
supports: technology categories

state_<Building>_max_level_terrain_limit = state
```

New Triggers:

`any_scientist`

same as other any_ triggers

`all_scientists`

same as other all_ triggers

`has_scientist_level`

```
Checks if the scientist of the character in scope matches the skill level
condition for a specialization. Supports < > = operators.
level = <int>
specialization = <specialization_token>
ex:
my_character = {
    has_scientist_level = {
        level > 2
        specialization = specialization_nuclear
    }
}
```

`is_active_scientist`

character scope

```
Checks if the scientist of the character in scope is assigned to a project
is_scientist_active = <bool>
ex:
my_character = {
    is_scientist_active = yes
    is_scientist_active = no
}
```

`has_project_flag`

project scope
same as other flag check triggers

`any_active_scientist`

same as other any_ triggers

all_active_scientist

same as other all_ triggers

is_scientist_injured

character scope

```
Checks if the scientist of the character in scope is injured
is_scientist_injured = <bool>
ex:
my_character = {
    is_scientist_active = yes
}
```

has_breakthrough_points

```
Checks if the country in scope has enough breakthroughs within a given
specialization.
specialization = <specialization_token>
value = <point>
ex:
GER = {
    has_breakthrough_points = {
        specialization = specialization_nuclear
        value = 1
    }
}
```

has_naval_invasion_against_state

```
Check if the scoped country has a naval invasion against the specified state.
Example 1:
has_naval_invasion_against_state = <STATE_ID>
Example 2:
has_naval_invasion_against_state = {
    state = <STATE_ID>
    preparation > 0.0  # (optional: preparation percentage, with a default value
of 0.0)
    activated = no   # (optional: if set, also check if invasion is activated or
not)
}
```

any_state_in

```
check if any state in the given category meets the trigger.
tooltip=key can be defined to override title.
The trigger takes one of the followings:
    array: an array of states.
    continent: A continent.
    ai_area: The id of an area.
    strategic_region: The id of an region.

* Note that no default tooltip is available for array and ai_area.
Example:
Check if the trigger is valid in any state in a continent:
any_state_in = {    conteinent = europe
    FOO_TRIGGER = BAR
}
```

scope_exists

```
Check if the current scope exist.
This differ from for example exists that checks if the country of the scope
exists.
This checks if the scope for the country exists and the other if the country
itself exists in the game.
Note that variable scopes are always valid scopes.
Example:
DEN = { exists = yes } # Should always be true since DEN is always a valid scope
sp:sp_land_flamethrower_tank = {
    character = {
        scope_exists = yes
    }
} # True if the project has an assigned scientist.
var:my_var = {
    scope_exists = yes # Always true since variables are always valid scopes
}
```

equipment_cost

```
check cost of equipment production line
```

is_special_project_completed

```
Checks if the country in scope has completed the special project in input.
```

```
ex:
SOV = {
    is_special_project_completed = sp:my_project
    is_special_project_completed = var:my_project_var
    is_special_project_completed = PREV # accepts variables and keywords
}
```

`is_special_project_being_researched`

```
Checks if the country in scope is currently researching the special project in
input.
ex:
SOV = {
    is_special_project_being_researched = sp:my_project
    is_special_project_being_researched = var:my_project_var
    is_special_project_being_researched = PREV # accepts variables and keywords
}
```

`has_market_access_with`

```
Check if the country has market access with another country. Example:
has_market_access_with = GER
```

`has_officer_name`

I assume in the unit scope

```
checks if division has an officer with the provided name key.
Examples:
has_officer_token = FIN_nikke_parmi
```

`has_artillery_ratio`

combatant scope

```
Check that ratio of atrillery battalions in the composition of a side of
combating troops are over a certain level.
For example:
has_artillery_ratio > 0.1
```

`has_unit_type`

combatant scope

```
Check if the combatant has at least one of the provided unit types.
For example:
has_unit_type = amphibious_mechanized
```

province_vp

combatant scope

```
Check if the victory points of the combatants province is larger or less than the provinded amount.
For example:
province_vp > 2
province_vp < 3
```

has_shine_effect_on_focus

```
Check if country has shine effect on focus (either manually achieved or by being worked on).

Note that tooltips are only shown in debug mode.

### Example
```
has_shine_effect_on_focus = GER_prioritize_economic_growth
```

custom_override_tooltip

See example of bindable localization

```
An `AND` trigger that has an overriden custom tooltip.
A positive tooltip can be set with `tooltip` and the tooltip to be used inside a NOT can be set with `not_tooltip`.
If no positive tooltip is provided and the root key is a localization key (not a formatter, see [formatted localization](script_concept_documentation.md#formatted_localization)),
then a negative tooltip will be generated by appending `_NOT` to the root localization for the positive tooltip.
Both tooltip and `not_tooltip` are [bindable localizations](script_concept_documentation.md#bindable_localization).

Examples
```
custom_override_tooltip = {
    tooltip = MY_TOOLTIP # Simple loc key tooltip
    not_tooltip = MY_TOOLTIP_NOT
    <other triggers>
}
```
```
```

```
custom_override_tooltip = {
    tooltip = MY_TOOLTIP
    # Implicit:
    #not_tooltip = MY_TOOLTIP_NOT
    <other triggers>
}
```

has_contested_owner

```
Checks if a state has the specified country as a contested owner.
The trigger can be used either from a country or a state scope and accepts the
other as parameter.
The trigger is localized with a localization environment containing `Country` and
`State`.

### Example
The following example has the same end result and localization.
```
42 = {
    has_contested_owner = GER
}
GER = {
    has_contested_owner = 42
}
```
Standard scope accessors can also be used:
```
Assuming current scope is a state and FROM is a country scope
has_contested_owner = FROM
```
```

fighting_army_strength_ratio

```
Compares the total army fighting strength between the scope country and the one
set with 'tag'

Example 1:
fighting_army_strength_ratio = {
```

```
    tag = TAG
    ratio > 0.7 # can be '<','>' or '='
}

Example 2:
fighting_army_strength_ratio = {
    tag = TAG
    ratio > VARIABLE # can be '<','>' or '='
}
```

has_scientist_specialization

```
Checks if the country in scope has a scientist with a skill level of at least 1
in specialization.
ex:
SOV = {
    has_scientist_specialization = specialization_nuclear
}
```

has_facility_specialization

```
Checks if the country in scope has a facility with specialization.
ex:
SOV = {
    has_facility_specialization = specialization_nuclear
}
```

can_construct_building

```
Checks if the country (as ROOT) and state in scope can build a building in the
state.
ex:
GER = {
    65 = {
        can_construct_building = land_facility
    }
}
```

num_nukes_being_dropped

```
total number of nukes that are currently ready to be dropped
```

num_nukes_left_to_drop

```
number of nukes left to drop during this game tick (only useful in-between nuke drops, like in on_nuke_drop on-action, for example)
```

New variables:

Country Scoped Variables:

total_equipment_produced_crossing_support_vehicle
- total produced equipment of typecrossing_support_vehicle

total_equipment_produced_armored_recovery_vehicle
- Total produced equipment of typearmored_recovery_vehicle

num_of_civilian_factories_in_cores
- calculates the number of civilian factories on core states of current country scope, on those states that are under control of @Tag <Tag | ROOT | my_var> example num_of_civilian_factories_in_cores@GER

num_of_military_factories_in_cores
- calculates the number of civilian factories on core states of current country scope, on those states that are under control of @Tag <Tag | ROOT | my_var> example num_of_military_factories_in_cores@GER

num_of_naval_factories_in_cores
- calculates the number of civilian factories on core states of current country scope, on those states that are under control of @Tag <Tag | ROOT | my_var> example num_of_naval_factories_in_cores@my_var

total_equipment_produced_sam_missile
- Total produced equipment of typesam_missile

total_constructed_gun_emplacement
- Total constructions of gun_emplacement

total_equipment_produced_clearance_vehicle
- Total produced equipment of typeclearance_vehicle

total_equipment_produced_ballistic_missile
- Total produced equipment of typeballistic_missile

total_equipment_produced_land_cruiser
- Total produced equipment of typeland_cruiser
total_equipment_produced_emplacement_gun_ammo
- Total produced equipment of typeemplacement_gun_ammo
total_equipment_produced_missile_launcher
- Total produced equipment of typemissile_launcher
num_nukes_being_dropped
- total number of nukes that are currently ready to be dropped"
num_nukes_left_to_drop
- number of nukes left to drop during this game tick (only useful in-between nuke drops, like in on_nuke_drop on-action, for example)

Special Project Scoped Variables:

You must prepend with “var:” for all special project scoped variables.
facility_state
- State that the project is researched in
facility_province_id
- The province that the project is researched in
scientist
- The scientist who researches the project

Raid Scoped Variables

You must prepend with “var:” for all raid scoped variables. Furthermore, if you need to go back to the ROOT or previous scope outside of the raid scoped variable you will need to prepend with ROOT.

actor_country
- country that launched the raid
victim_country
- country that is target of the raid
target_state
- state that the raid happens in
target_province
- province that the raid happens in

Defines:

Removed Defines:

```
NUKE_MIN_DAMAGE_PERCENT = 0.1,                  -- Minimum damage from nukes as a
percentage of current strength/organization
NUKE_MAX_DAMAGE_PERCENT = 0.9,                  -- Minimum damage from nukes as a
percentage of current strength/organization
STRATEGIC_BOMBER_NUKE_AIR_SUPERIORITY       -- How much air superiority is needed
for a tactical bomber to be able to nuke a province
ANNEX_RATIO = 0.5,
-- How many railway guns will be transferred on annexation
HOURS_BETWEEN_REDISTRIBUTION = 24,
-- Number of hours between redistribution of attached railway guns, tracked per
army
PLAN_FRONT_SECTION_MAX_LENGTH = 18,
-- When a front is longer than this it will be split in two sections for the AI
PLAN_FRONT_SECTION_MIN_LENGTH = 10,
-- When two front sections together are this short they will be merged for the AI
DIVISION_DESIGN_WEIGHTS
DIVISION_DESIGN_MANPOWER_WEIGHT = 0.005,
DIVISION_DESIGN_STOCKPILE_WEIGHT = 0.01,
DIVISION_DESIGN_COMBAT_WIDTH_WEIGHT = -1.0,
-- This score is reduced the higher width is when comparing pure changes with no
target
DIVISION_DESIGN_COMBAT_WIDTH_TARGET_WEIGHT = -200.0,
-- This score is reduced the farther the width is from the target width (if set)
COMBINED_ARMS_LEVEL = 1,
-- 0 = Never, 1 = Infantry/Artillery, 2 = Go wild
INVASION_DISTANCE_RANDOMNESS = 300,
-- This higher the value, the more unpredictable the invasions. Compares to
actual map distance in pixels.
ARMY_LEADER_ASSIGN_KEEP_LEADER = 500,
-- Score for keeping the leader if already assigned
```

Changed Defines:

```
FEMALE_UNIT_LEADER_BASE_CHANCE
 - added scientist chance
MISSION_COMMAND_POWER_COSTS
added barrage, sam
MISSION_FUEL_COSTS
added barrage, nuclear, sam

* `RAILWAY_GUN_POSSIBLE_RANGES = { 30, 15, 45 }`
    -- Possible values for railway gun range in pixel.
    -- For optimization reasons, they are listed here and equipment DB must use
one of those.
    -- when writing railway gun in equipment, use the index in this array
    -- the first value in array is the default value
—----------------------------------

HOUR_BAD_COMBAT_REEVALUATE = 48,
-- if we are in combat for this amount and it goes shitty then try skipping it
—Changed into—
CANCEL_COMBAT_DISADVANTAGE_RATIO = 1.5,
-- If the enemy's advantage ratio over us during (normal) combat is more than
<value>, allow canceling the attack
CANCEL_COMBAT_MIN_DURATION_HOURS = 48,
-- Only allow canceling (normal) combat if at least <value> hours have passed
CANCEL_INVASION_COMBAT_DISADVANTAGE_RATIO = 3.5,
-- If the enemy's advantage ratio over us during invasion combat is more than
<value>, allow canceling the attack
CANCEL_INVASION_COMBAT_MIN_DURATION_HOURS = 120,
    -- Only allow canceling invasion combat if at least <value> hours have passed
—---------------------------------



FAILED_INVASION_AVOID_DURATION = 60, -- after a failed invasion, AI will down-
prioritize invading the same area again for this number of days
—Changed into—
INVASION_TARGET_DISTANCE_DENOMINATOR = 1000,
```

```
-- When selecting invasion target, divide this with (pixel) distance to get
distance score factor. (Doesn't really affect the relative scoring, but it
affects the linearity of the score function.)
INVASION_TARGET_NO_PORT_FACTOR = 0.3,
-- When selecting invasion target, multiply score with this if the target has no
port
INVASION_TARGET_TRUNCATION_SELECT_THRESHOLD = 0.6,
-- When selecting invasion target, use this threshold for truncation selection.
(1.0 means select highest scored target, 0.0 means select randomly from all
possible target, 0.5 means select randomly from all targets with more than 50 %
of highest score)
INVASION_TARGET_PRIO_NOT_ENEMY_FACTOR = 0.17,
-- When calculating priority for an invasion, factor the score with this if the
target is not an actual enemy.

—--------------------------------
PLAN_PROVINCE_LOW_VP_IMPORTANCE_AREA = 2.0,
-- Used when calculating the value of defense area in the battle plan system
PLAN_PROVINCE_MEDIUM_VP_IMPORTANCE_AREA = 5.0,
 -- Used when calculating the value of defense area in the battle plan system
PLAN_PROVINCE_HIGH_VP_IMPORTANCE_AREA = 10.0,
 -- Used when calculating the value of defense area in the battle plan system
PLAN_PROVINCE_CAPITAL_IMPORTANCE_AREA = 50.0,
-- Used when calculating the balance of defense area in the battle plan system

—Changed into—

PLAN_PROVINCE_LOW_VP_DEFENSE_THRESHOLD = 2.0,
-- For area defense VP orders, what are the thresholds for "low", "medium" and
"high" VP values
PLAN_PROVINCE_MEDIUM_VP_DEFENSE_THRESHOLD = 8.0,
 -- see above
PLAN_PROVINCE_HIGH_VP_DEFENSE_THRESHOLD = 25.0,
-- see above
PLAN_PROVINCE_LOW_VP_DEFENSE_IMPORTANCE = 2.0,
-- For area defense VP orders, use this value for relative importance
PLAN_PROVINCE_MEDIUM_VP_DEFENSE_IMPORTANCE = 5.0,
-- see above
PLAN_PROVINCE_HIGH_VP_DEFENSE_IMPORTANCE = 10.0,
```

```
 -- see above
PLAN_PROVINCE_CAPITAL_DEFENSE_IMPORTANCE = 50.0,
 -- For area defense VP orders, boost importance value with this if it's the
capital
```

New Defines:

```
New defines:
* `THERMONUCLEAR_BOMB_DROP_WAR_SUPPORT_EFFECT_MAX_INFRA`    -- Reduce enemy
national war support on nuking a province, the value scales with infrastructure
up to this number
* THERMONUCLEAR_BOMB_DROP_WAR_SUPPORT_EFFECT_MAX_VP -- War support will be scaled
down if there's less VP than this in the province
* `MAX_MIL_FACTORIES_VISIBLE_FOR_MIL_EQUIPMENT_LINE`
* `SAM_MISSION_SUPERIORITY` -- How much air superiority each SAM mission gives
per rocket wing performing SAM missions.
* `PLAN_AREA_DEFENSE_FACILITY`-- Used when calculating the value of defense area
provinces for the battle plan system
* `MISSILE_LAUNCHER_CAPACITY`-- The number of missiles per slot
* `MISSILE_LAUNCHER_SLOTS`-- The number of missile slots a missile launcher unit
can have
* `UNDERWAY_REPLENISHMENT_PRIORITY`-- Default convoy priority for underway
replenishment
* `UNDERWAY_REPLENISHMENT_RANGE_FACTOR`-- bonus factor applied to task force's
range when underway replenishment is activated (e.g. 0.2 means +20%)
* `UNDERWAY_REPLENISHMENT_CONVOY_COST_PER_FUEL`-- Cost in convoys for underway
replenishment multiplied by max daily fuel consumption (rounded up)
* `MIN_SHIPS_FOR_HIGHER_SHIP_RATIO_PENALTY`-- the minimum fleet size in ships
that a fleet must be before having the large fleet penalty applied to them
* `GUN_EMPLACEMENT_MIN_ASSIGN_SCORE`-- Minimum total score for region to be
considered for gun emplacement air missions
* `GUN_EMPLACEMENT_MIN_PRIO_ASSIGN_SCORE`-- Minimum total score for region to be
considered for critical gun emplacement air missions
* `GUN_EMPLACEMENT_ASSIGN_SCORE_REDUCTION_PER_ASSIGNMENT`-- each assigned gun
emplacement reduces the score of a region by this amount
* `REMOVE_OBSOLETE_TEMPLATE_DAYS`-- Remove obsolete and unused templates if they
have been marked as obsolete for x days. Non-positive value means "never remove".
```

```
LAND_DEFENSE_RAID_IMPORTANCE = 500,
-- Strategic importance of detected raids targetting us
LAND_DEFENSE_FIGHERS_PER_RAID = 100,
-- Amount of air superiority planes requested per detected raid targetting us
LAND_DEFENSE_INTERCEPTORS_PER_RAID = 100,
-- Amount of interceptor planes requested per detected raid targetting us
LAND_DEFENSE_SAM_MISSILE_IMPORTANCE_FACTOR = 0.2,
    -- Importance factor of using sam missiles for regions strategic importance. Higher value will increase the usage
LAND_COMBAT_MISSILE_IMPORTANCE_FACTOR = 1.5,
-- Importance factor of using missiles for regions strategic importance. Higher value will increase the usage
* `CONSTRUCTION_PRIO_SUPPLY_BUILDING`-- base prio for supply buildings (supply hubs, ports) in the construction queue
* `MIN_INVASION_ORG_FACTOR_TO_EXECUTE`-- ai will only activate invasions if average org factor is above this
* `ARMY_LEADER_ASSIGN_KEEP_CURRENT_LEADER_FACTOR` -- Boosts the score for keeping the current leader. Value > 1.0 favors the current leader.
* `ARMY_LEADER_ASSIGN_DONT_STEAL_OTHER_FACTOR` -- Reduces the score for leaders assigned elsewhere. Value < 1.0 discourages reassigning these leaders.
* `AREA_DEFENSE_SETTING_BORDERS` bools
* `AREA_DEFENSE_SETTING_FACILITY` bools

RAID_MIN_INTEL_FOR_WARNING_ON_LAUNCH = 0.1,
-- how much intel (of the relevant type) is needed to show a warning when raid is launched
RAID_MIN_INTEL_FOR_WARNING_HALFWAY_TO_LAUNCH = 0.5,
-- how much intel (of the relevant type) is needed to show a warning halfway through preparation
 --     (this limit is a dummy value only used for communicating the role of intel in the intel ledger )
--  (in reality, detection scales linearly with intel. 70% intel = detection at 30% preparation, 50% intel = detection at 50% preparation, etc.
RAID_MIN_INTEL_FOR_WARNING_EARLY_PREPARATION = 0.8,
 -- how much intel (of the relevant type) is needed to show a warning early in the preparation
  --    (this limit is a dummy value only used for communicating the role of intel in the intel ledger )
```

```
--  (in reality, detection scales linearly with intel. 70% intel = detection at
30% preparation, 50% intel = detection at 50% preparation, etc.
NUCLEAR_RAID_CATEGORY_NAME = "nuclear_raids",
-- The raid category to activate when clicking on the "nuclear" mission button
for a rocket
```

- Every aifc define is new.
- All defines in the special projects and raids sections are new.

AI Changes:

Improved Decision AI:

The Hearts of Iron IV AI will now properly save and spend political power depending on the queue of options in their AI political power spending queue. You can view this by going to imgui show ai_pp_spend once you are in the game.

AI priorities for all political power decisions are now directly derived from the queue in ai_pp_spend queue. The AI will no longer randomly choose decisions and it will now have a concept of saving for particular decisions or otherwise.

The rationale for the change is that it would make the AI make decisions with some appropriate level of determinism. Designers can now script out decisions with appropriate priority depending on the ai_will_do or other game state factors.

Caveats:

- The AI will no longer randomly choose decisions unless the AI has a priority of greater than > 0 and costs 0 pp.
- All spend options are now weighed against one another (advisors, decisions, laws, etc) and weighted against one another. Some of these have these set through game state factors and some have nudges in script (ai_will_do).
- Low-cost decisions will not be taken unless they find themselves in the queue and at the top of the queue.
- Dual cost decisions will still be a massive issue due to the fact the AI does not conceptualize costs other than political power. You will need to script ai_will_do chances here to ensure that is taking it without killing itself.

You can use the legacy political power system for some time but this is not going to be a long-kept system and will eventually be phased out.

Improved Spearhead AI:

The AI will now be more logical with their units and form them up to spearhead areas and concentrate force in regions to fight and push through enemy lines. The AI will now assign divisions based on the stats. If the unit has a high breakthrough they will concentrate the AI and push in specific areas hoping to secure pockets and breakthroughs against the enemy AI. You can view the performance of your AI by viewing the imgui using the following command: imgui show ai_force_concetration.

Furthermore, there are a number of new AI strategies you can use to manipulate and improve the AI force concentration metrics.

霜泽图书馆 ParaWikis 汉化组呈上

最后，向大家推荐一下一个代码制作交流群和美工交流群，欢迎各位对代码（美工）感兴趣的萌新和代码（美工）功成的大佬加入：

霜泽图书馆 群号：378525932 霜泽美术馆 群号：976242794

白雪老师的完美教室：768450264

并且向各位附上代码图书馆，内包含诸多代码文字教程和方便制作的软件和工具：

秋起图书馆 GitHub 地址 ：

https://github.com/jingzhouzhidi/QIUQI-LIBRARY.git

并且，为了各位的制作上的方便，我们急需翻译和审核进行钢铁雄心 4wiki 的汉化工作，因而在此发布招新的地址，感谢支持：

霜泽图书馆 PARAWIKIS 汉化组

群号：996535362

Paratranz 组链接：https://paratranz.cn/projects/11411/

如本篇翻译出现任何问题：

请及时联系我们：378525932（霜泽图书馆 群主或管理）