

Полный обзор системы построек – tutorial

Содержание

- Терминология
- Структура таблиц
- Подробный разбор таблиц
- Tutorial – Создание новой структуры в существующей субцепи
- Tutorial – Создание новой субцепи
- Tutorial – создание новой цепи
- Tutorial – создание новой суперцепи

Терминология

- **Структура** – описание группы параметров, которые включает в себя одна постройка (например, Загоны для скота)
- **Субцепь** – одиночная линия развития в цепи (например, линия Загоны для скота – Усадьба с выпасом – Скотный двор)
- **Инстанция** – группа из одной или более линий развития в цепи
- **Цепь** – группа субцепей, начинающаяся с первоначальной структуры для фракции/культуры (например, Земледелие)
- **Суперцепь** – группировка цепей, имеющих общие параметры для любой группы фракций/культур

Структура таблиц

- **Building_superchains**
 - **Building_chains**
 - **Building_levels**
 - **Building_effects_junction**
 - **Building_level_required_technology**
 - **Building_units_allowed**
 - **Building_level_armed_citizenry_junctions**
 - **Building_upgrades_junction**
 - **Building_cultural_variants**
 - **Building_culture_variants.loc**
 - **Building_short_description_texts.loc**

- **Building_flavour_texts.loc**
- **Building_instances**
- **Campaign_settlement_display_buildings**
 - **Campaign_settlement_display_buildings_ids**
 - **Campaign_settlement_display_buildings_construction**
- **Cai_base_building_context_values**
 - **Building_chain_availabilities**
 - **Building_set_to_building_junctions**
 - **Building_sets**
 - **Cai_construction_system_buildings_values**

Подробный разбор таблиц

Building_superchains

- Описывает группу цепей, имеющих одинаковые параметры в разных культурах/фракциях. Также описывает, в какие структуры можно превратить что-либо при заимствовании зданий из других субкультур
- Колонку **maxinstances_per_region** можно не использовать

Building_chains

- Описывает первоначальную структуру, от которой ответвляются все остальные структуры, в группе структур для культуры/фракции
- Задаёт категорию цепи и привязывает её к суперцепи
- **building_chain_availabilities_tables**
- Здесь, посредством ключей **building_chain**, постройки привязываются к компании, к культуре, к субкультуре. Поэтому когда вы перекидываете фракцию из Шарлеманя, она начинает строить те постройки, ключ **building_chain** которых, привязан в этой таблице к компании Велизария и к культуре данной фракции.

Building_levels

- Описывает каждую отдельную структуру, которая может быть построена
- Ключевые параметры, такие как цепь, уровень постройки, цена постройки, необходимое для строительства количество ходов,

уникальная фракция, необходимые ресурсы и инстанция описываются здесь

- Возможные будущие характеристики, в данный момент не используемые (стоимость содержания, стоимость разрушения, рост благополучия и т.п.)

Building_chain_availabilities

- Описывает, какие культуры используют данную цепь
- Также дополнительные поля для субкультур или фракций

Building_set_to_building_junctions

- Описывает, к какой категории (set) относится каждая цепь или структура (сельское хозяйство, город, промышленность, порт, религия, водопровод и канализация и т.д.)
- Если конкретная структура относится (одновременно) к другой категории, это описывается дополнительно

Cai_construction_system_buildings_values

- Используется ИИ для того, чтобы он знал, как строить здания

Building_effects_junction

- Описывает, какие эффекты будет иметь построенное здание
- Также могут быть заданы значения для каждого эффекта при поврежденном или разрушенном здании

Building_level_required_technology

- Описывает, какие технологии необходимы для постройки здания
- Должны быть описаны абсолютно все необходимые технологии, не только та, что разблокирует структуру

Building_units_allowed

- Описывает, какие юниты могут быть наняты при построенном здании
- Колонку **Experience** можно не использовать

Building_level_armed_citizenry_junctions

- Описывает гарнизон, появляющийся после постройки здания в регионе
- Упомянутые группы юнитов описаны в **armed_citizenry_groups**

Building_upgrades_junction

- Описывает связи между структурами в цепи
-
- Позволяет разветвлять цепи (у каждой структуры, не только у первоначальной, может быть два или более апгрейда на каждом уровне)

Building_cultural_variants

- Описывает, какие культуры могут использовать эту структуру (постройку)
- Устанавливает иконку для постройки (ссылка **ui/buildings/*.png**)
- Дополнительная колонка для ограничения использования структуры в пределах субкультуры или фракции

Building_instances

- Описывает группировки субцепей по типу
- Членство в группе описывается в **building_levels**
- Может быть использована для создания взаимной исключительности субцепей (?) или для создания множества структур из одной субцепи (ни то, ни другое не было использовано в ванилле)

Campaign_settlement_display_buildings

- Описывает модель, которая будет показываться, как только постройка будет возведена
- Ключевая ссылка (? – references key) описывается в **campaign_display_building_ids**, не в id постройки (хотя это может быть один и тот же id)

Cai_base_building_context_values

- Заявляет структуры для ИИ

Building_sets

- Описывает широкие категории (broad categories) построек
- Используется для определения, какие здания подвержены влиянию (негативному) конкретных эффектов (эффекты описаны в `effect_bonus_value_building_set_junctions`)

Building_culture_variants.loc

- Имя структуры
- Ключевой формат: `building_culture_variants_name_<id постройки><id культуры><id субкультуры>`
- Оставьте поля культуры/субкультуры пустыми, если они не прописаны в db таблице `building_culture_variants`

Building_short_description_texts.loc

- Короткое предложение, описывающее структуру (постройку)
- Ключ задан в `building_culture_variants` в формате: `building_short_descriptions_texts_short_description_<ключ>`

Building_flavour_texts.loc

- Длинное описание, обычно включающее исторические данные. Появляется в сообщении о событии (event message), когда структура (постройка) закончена.
- Ключ задан в `building_cultural_variants` в формате: `building_flavour_texts_flavour_<ключ>`

Campaign_settlement_display_building_ids

- Описывает связь между ключами структуры и ключами отображаемой модели для построек на стратегической карте
- Ключ структуры может быть таким же, как и ключ модели

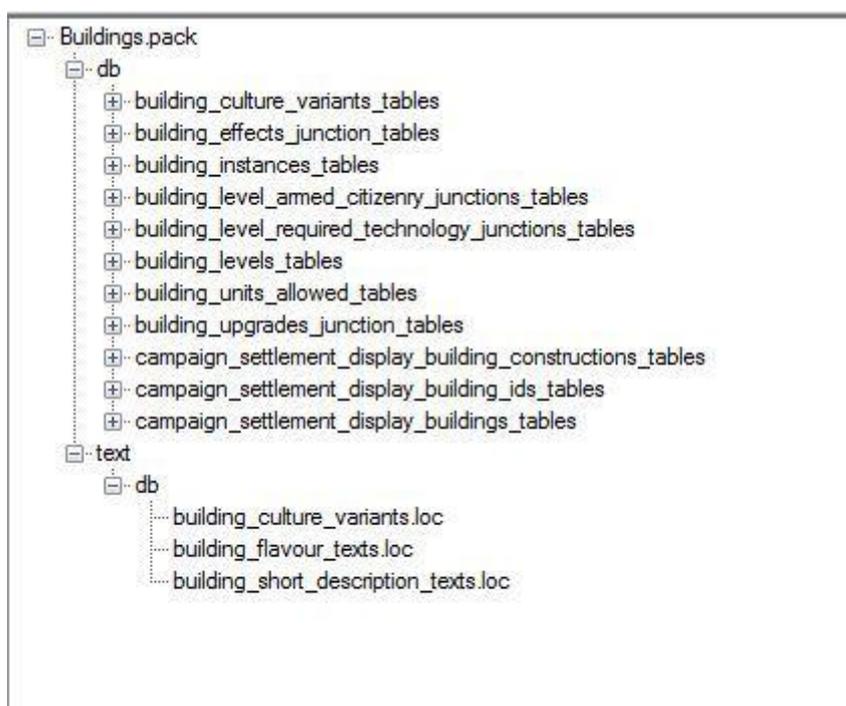
Campaign_settlement_display_building_construction

- Записи, необходимые для создания анимации строительства и разрушения здания (на стратегической карте)
- Нужна ссылка на ключ отображаемой модели, не на ключ структуры

Создание новой структуры в существующей субцепи

Для создания совершенно новой структуры в существующей цепи вам нужна таблица **Building_levels** и большинство ее подуровней (все до Campaign_settlement_display_building_construction) в вашем pack файле. Эффекты, технологии, юниты и гарнизон добавляются в зависимости от назначения структуры, но если вы делаете больше одной постройки, то в любом случае лучше сразу их охватить.

Таким образом, ваш Building_mod.pack должен начинаться со следующих таблиц:



Building_levels:

Первый шаг в описании новой структуры – описание ее базовых параметров и присвоение ей имени, на которое можно будет ссылаться (ключа). Я добавил 5 уровень к главной линии развития эллинистического города, используя ключ: **rom_HELLENIC_city_major_5_Crzy**

Как и в большинстве случаев, описанных в данном tutorialе, быстрее способ сделать то, что нужно – скопировать строку структуры, предшествующей вашей создающейся структуре. В данном случае это

rom_HELLENIC_city_major_4. В колонке *levelname* нужно написать новый ключ, в колонку *level* добавить 1 и изменить содержимое колонки *uniqueindex* на уникальное значение. Также можно задать значения колонок *createtime* (время создания) и *createcost* (цена создания), но это необязательно.

Building_upgrades_junction

Следующий шаг – связывание новой структуры с предшествующей постройкой (той, из которой делается апгрейд на новую постройку). Нужно просто поместить предшествующую постройку в колонку *from* (в данном случае это **rom_HELLENIC_city_major_4**), а новую постройку в колонку *to* (в данном случае **rom_HELLENIC_city_major_5_Crzy**).

Building_culture_variants

Это последняя таблица, необходимая для того, чтобы ваша новая структура показывалась в игре. Простейший путь – опять же копирование строки предыдущей постройки в цепи. Важные колонки здесь – *icon*, *shortdescription* и *flavour*. В колонке *icon* содержится имя изображения в формате *.png с установленными параметрами, которое будет показываться в UI как только здание будет построено, остальные две колонки содержат ссылки на текстовые описания из loc файлов, которые мы добавим позже. После копирования предыдущей строки все, что нужно – сменить ключ постройки на новый ключ нашей структуры.

Campaign_settlement_display_buildings

Если эта таблица не заполнена, при постройке вашего нового здания на стратегической карте на его месте окажется лишь большая круглая дыра.

Чтобы этого не произошло, нужно обратиться к таблице **campaign_settlement_display_building_ids** и создать связь между ключом новой структуры и ключом модели на стратегической карте. Проще всего сделать оба ключа одинаковыми, так, как это показано в первых двух колонках нижеследующей таблицы (остальные колонки следует заполнить опять же копированием информации из предыдущей структуры).

key	buildinglevel_key	culture	unknown	spreadpiece_level	spreadpiece_type	unknown7
-----	-------------------	---------	---------	-------------------	------------------	----------

rom_HELLENIC_city_major_5_Crzy	rom_HELLENIC_city_major_5_Crzy	<blank>	0
--------------------------------	--------------------------------	---------	---

Затем нужно вернуться в **Campaign_display_buildings** и еще раз скопировать строку предыдущей постройки. Изменить значение колонки *key* на любой придуманный уникальный ключ, а в колонку *displaybuilding* поместить только что описанный ключ отображаемой модели.

key	buildingpath	displaybuilding_key	<unknown X>
50000	RigidModels\...\greek_city_major_4.rigid_model_v2	rom_HELLENIC_city_major_5_Crzy	<leave blank>

Наконец, чтобы создать анимацию строительных кранов в течение строительства/разрушения постройки, нужно обратиться к таблице **campaign_settlement_display_building_construction**. Скопируйте любые 2 строки, в колонке *id* присвойте им два любых уникальных значения, убедитесь, что в колонке *constructiontype* в одной строке стоит "default", а в другой "dismantle" и измените значение колонки *displaybuildingkey* на ключ вашей постройки.

id	buildingpath	constructiontype	displaybuilding_key	phase	phase6
70001	RigidModels\...\building_construction_anim.rigid_model_v2	default	rom_HELLENIC_city_major_5_Crzy	0	0
70002	RigidModels\...\building_construction_anim.rigid_model_v2	dismantle	rom_HELLENIC_city_major_5_Crzy	0	0

Теперь у вашей новой структуры есть отображаемая модель на стратегической карте (она будет такой же, как и у постройки предыдущего уровня, если только у вас нет подходящей модели для добавления в **Campaign_display_buildings**).

Buildings_level_required_technology

Эта таблица устанавливает необходимый для разблокировки новой постройки уровень технологического развития. Она очень проста – в первую колонку следует поместить ключ новой постройки, во вторую – необходимые для создания новой постройки технологии. Желательно описать все

необходимые технологии, а не только ту, что непосредственно разблокирует постройку.

Building_effects_junctions

Здесь описываются эффекты от постройки, такие как установленный доход, бонус к доходу в процентах, пища, загрязнение и т.д. В данной таблице не описываются гарнизон и найм юнитов.

В целом таблица не требует особых разъяснений. Первая колонка – ключ вашей структуры, во второй колонке описывается эффект, который вы хотите ей придать (полный список эффектов можно найти в **effects_tables** в data_rome2.pack). Третья колонка описывает, как широко будет применяться эффект, а последние три колонки задают значение эффекта при нормальном, поврежденном и полностью разрушенном здании соответственно.

Я установил для своего здания одиночный эффект – повышение дохода города на 3000 следующей записью:

building	effect	effectscope	value	valuedamaged	valueruin
rom_hellenic_city_major_5_Crzy	rom_building_gdp_subsi stence	this_buildi ng	3000	500	0

Building_units_allowed

Эта таблица устанавливает, какие юниты могут быть наняты при построенном здании.

В настоящий момент известно значение только 4 колонок этой таблицы – это колонки: *key* (уникальный номер id), *building* (ключ постройки), *unit* (id нанимаемого юнита) и *XP* (устанавливается количество опыта, который будет иметь нанятый юнит).

Для моей структуры я добавил только один юнит, *Царские пельтасты*, и установил для них значение опыта 3:

key	building	unit	XP	enabled	unknown6	enabled7	enabled8
70020	rom_HELLENIC_city_major_5_Crzy	Gre_Royal_Pelta sts	3	FALSE	0	FALSE	FALSE

Building_level_armed_citizenry_junctions

Эта таблица отвечает за группы гарнизонных юнитов, которые появятся, если поселение с возведенной постройкой будет атаковано. Многие гарнизонные группы уже прописаны в таблице **armed_citizenry_unit_groups**. Вы также можете создать свою гарнизонную группу с помощью таблицы **armed_citizenry_units_to_unit_groups_junctions**.

Эта таблица состоит из 4 колонок: *id* (уникальный номер), *buildinglevel* (ключ новой постройки), колонка с неизвестным значением и колонка *unitgroup* (появляющаяся гарнизонная группа).

Следующей записью я добавляю для моей постройки нескольких юнитов ближнего боя, используя гарнизонную группу **rom_hellenic_strong_melee**:

key	buildinglevel	unknown3	unitgroup
70030	romHELLENIC_city_major_5_Crzy	0	rom_hellenic_melee_strong

Теперь для завершения создания новой структуры осталось только добавить UI элементы.

building_culture_variants.loc

Здесь описывается имя новой постройки, которое будет отображаться в игре. Формат ключа постройки (в колонке *Tag*) в этой таблице таков:

building_culture_variants_name_<id постройки><id культуры><id субкультуры><id фракции>

Субкультуру и фракцию нужно прописывать только если они были прописаны в таблице *culture_variants*.

Tag	Localised String
building_culture_variants_name_romHELLENIC_city_major_5_Crzyrom_Hellenistic	Test City

building_short_description_texts

Отвечает за короткое описание под именем постройки. Формат ключа в этой таблице: **building_short_description_texts_short_description_<ключ, написанный в колонке short description таблицы culture_variants >**

Tag	localised string
building_short_description_texts_short_description_twr2_rome_greek_Test_short	Test town description

building_flavour_texts

Я не вполне уверен, где именно в игре отображается информация из этой таблицы, но в ней обычно пишется несколько предложений, описывающих назначение постройки.

Заполняется аналогично предыдущей таблице. Формат ключа:

building_flavour_texts_flavour_<ключ, написанный в колонке flavour таблицы culture_variants>

Tag	Localised String
-----	------------------

Иконка

Для завершения новой структуры нужна иконка для UI. Она должна быть изображением в формате PNG разрешением 66x66 пикселей с точно таким же названием, как прописано в колонке *icon* таблицы **culture_variants**.

Добавьте иконку в свой pack файл с путем: **ui/buildings/icons/<имя файла>.png**

На этом создание новой структуры закончено.

ПРИМЕЧАНИЕ

Для того, чтобы ИИ мог использовать новую структуру, нужны следующие таблицы.

cai_base_building_context_values

Скопируйте строку предыдущей постройки и вставьте туда (с заменой) ключ своей новой структуры.

cai_construction_system_building_values

Если вы добавляете новую структуру в конец существующей субцепи, вам нужно найти ключ существующей структуры и заменить значение колонки **building_range_end_inclusive** данными своей структуры. Если вы создали новое ответвление субцепи, вам нужно скопировать строку и заменить ту же запись на данные вашей новой постройки.

Создание новой субцепи

Создать новую субцепь (линию развития) очень просто. Вам понадобится всего 2 элементарных шага:

- 1) Создать новый ключ субцепи (также известен как building instance). Можно использовать и существующий ключ, но в большинстве случаев вам понадобится новый (позже я объясню, почему).
- 2) Соединение вместе 1 или более структур для формирования линии развития субцепи, подобно тому, что вы делали для добавления новой структуры в существующую субцепь.

building_instances

Очень простая таблица, состоит из ключа и целого числа.

Формат ключа неизвестен (предположительно <цепь>_<инстанция>), поэтому для описания продовольственных инстанций цепи порта будет использован ключ **rom_Port_Food**, а для экономических инстанций – ключ **rom_Port_Trade**. Для примера я собираюсь создать новую субцепь порта, для чего добавляю ключ **rom_Port_Crzy**.

Целое число – это количество субцепей этой группы, которые могут быть построены в 1 регионе (1 городе). В ванилле этот параметр установлен на 1

(поэтому нельзя построить 3 одинаковых постройки в 1 регионе). Как правило, именно так и следует создавать новые субцепи (число 1 подходит для большинства случаев). Интересно то, что несколько субцепей могут иметь одну инстанцию, то есть если вы создаете 2 субцепи с одной инстанцией (у которой прописано 1 в колонке лимита), в каждом регионе можно будет построить только 1 из этих субцепей). Таким образом, вы можете создавать взаимоисключающие субцепи (в этом случае вы будете использовать существующий ключ субцепи, не создавать новый).

building_upgrades

Так же, как и раньше, свяжите первоначальную структуру с первой из ваших новых структур, первую со второй и т.д.

Теперь новая субцепь создана (добавлена к главному римскому порту).



Создание новой цепи построек

Для создания новой цепи понадобится 2 шага:

- 1) Заявить новую цепь, установить ее категорию и привязать ее к суперцепи
- 2) Установить, какие культуры/субкультуры/фракции могут строить здания из этой цепи

building_chains

В этой таблице важны 3 колонки. В колонке *key* содержится внутреннее имя цепи, которое должно быть уникальным (его следует придумать). Колонка *chain_category*, по-видимому, используется в основном ИИ и содержит в себе название одной из 3 категорий цепи (*military* (военная), *happiness* (счастье/благополучие) и *money* (деньги)). Колонка *building_superchain* привязывает цепь к определенной суперцепи.

Я создал следующую запись для моей новой цепи амуниции:

key	tech_category_tab	tech_category_tab3	chain_category	in_encyclopedia	building_superchain
Crzy_Equipment_Chain	<empty>	<empty>	military	False	rom_MilitaryEquipment

building_chain_availabilities

Сокращенная версия таблицы **building_culture_variants**. Здания, входящие в вашу цепочку, должны соответствовать записям в обеих этих таблицах.

Примечание: Для первой постройки в цепи должен быть установлен уровень 0 в таблице **building_levels**.

На этом создание новой цепи построек завершено.

Создание новой суперцепи

Для создания новой суперцепи нам понадобятся всего две простые таблицы.

№1 building_superchains

пример строки

att_sch_eos_marib_dam

№2 building_chain

пример строки

att_bch_eos_marib_dam

№3 building_set

пример строки

att_building_set_legendary

№4 building set to building junctions

пример строки

att_building_set_legendary

В этой таблице просто заявляется ключ суперцепи (в колонке *key*). Колонка *max_instances_per_region*, по всей видимости, не используется (лучше всего оставить в ней стоящее там по умолчанию значение 1).

key	max_instances_per_region
rom_Equipment_Crzy	1

slot_template_to_building_superchain_junctions

Эта таблица устанавливает, строятся ли структуры, входящие в данную суперцепь, в первом (главном) слоте, дополнительном портовом (dock) слоте или во вторичных основных слотах.

Также она определяет, будут ли постройки доступны в крупных (major) или мелких (minor) поселениях.

id	building_superchain	slot_template
72000	rom_Equipment_Crzy	rom_minor_secondary

Создание суперцепи завершено.