

Создание каталогов

Каталоги нужны для подгрузки каких-либо ассетов Unity из бандлов.

В этой статье будет рассмотрен процесс создания бандлов и каталога для них.

О том, как добавить созданный каталог в ресурсы игры, написано [здесь](#).

Требования

У вас должны быть хотя бы минимальные знания по работе с редактором Unity и представление об ассет-бандлах.

Для продолжения вам понадобится редактор **Unity 2022.3.6** с пакетом **Addressables 1.21.15**, который можно установить через менеджер пакетов внутри редактора.

Работоспособность каталогов в LMR не гарантируется при использовании других версий редактора/пакета Addressables.

Платформа сборки

Каталоги собираются под текущую активную платформу в редакторе Unity. Её можно узнать, нажав на **File -> Build Settings** в меню редактора Unity или нажав **Ctrl + Shift + B**.

Если вы хотите, чтобы ассеты из каталогов работали на нескольких платформах (Windows, Android, Linux и т.п.) — вам нужно собирать их отдельно под каждую платформу.

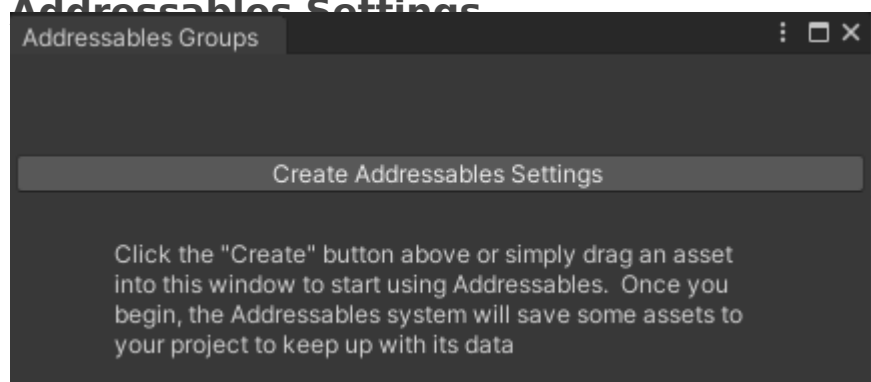
Настройка Addressables

Вы можете скачать готовый проект с необходимыми настройками.

После установки редактора и пакета, откройте окно редактирования групп Addressables:

Windows -> Asset Managment -> Addressables -> Groups

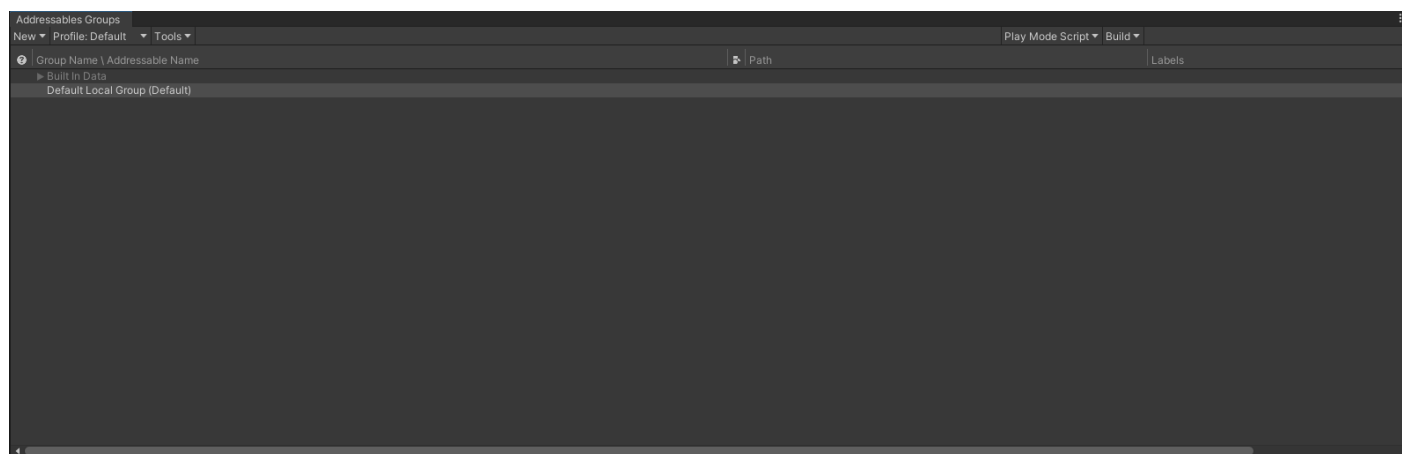
Если вы открываете это окно в первый раз, у вас появится сообщение о необходимости создать настройки Addressables. Нажмите **Create Addressables Settings**



Это сообщение появляется

только при первом запуске

После того, как завершится создание настроек, вы увидите следующее окно:



Внутри этого окна можно создавать различные группы ассетов, из которых собираются бандлы, информация о которых после сборки

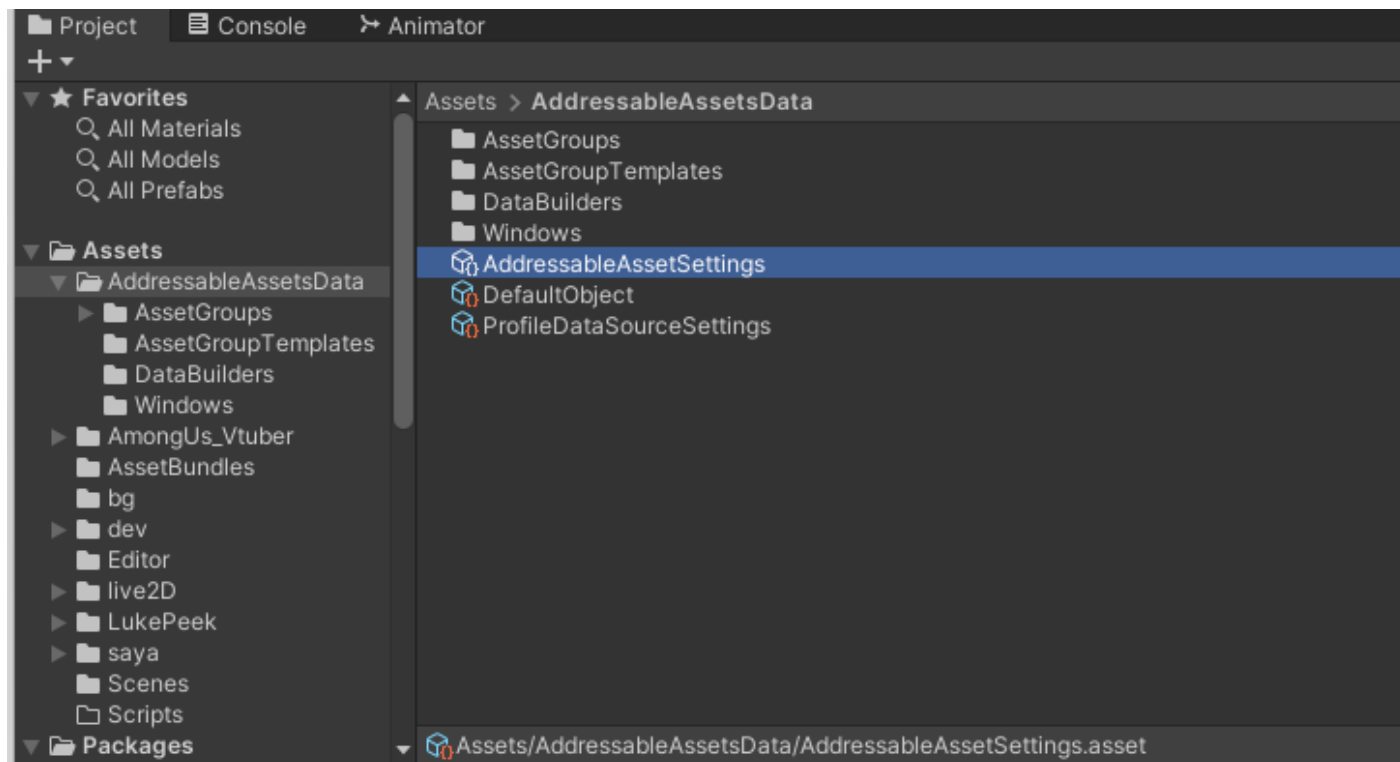
будет добавлена в **.json**-файл каталога.

Прежде, чем вы начнёте создавать группы ассетов, необходимо изменить некоторые настройки, чтобы каталог корректно подгружался в LMR.

AddressableAssetSettings

При помощи проводника проекта (по умолчанию располагается в самом низу окна редактора), найдите ассет AddressableAssetSettings, полный путь которого:

```
Assets/AddressableAssetsData/AddressableAssetSettings.asset
```



Кликните на него — в правой части окна редактора появится **Inspector** с настройками:



Manage Groups

▼ Profiles

Profile In Use Default

Manage Profiles

▼ Diagnostics

Send Profiler Events



Log Runtime Exceptions



▼ Catalog

Player Version Override [UnityEditor.PlayerSettings.bundleVersion]

Compress Local Catalog



Build Remote Catalog



Only update catalogs manually



▼ Update a Previous Build



You must enable the remote catalog in the Catalog settings to update a previous build.

Check for Update Issues List Restricted Assets (recommended)

Content State Build Path <default settings path>

▼ Path Preview

Build Path: Assets/AddressableAssetsData/Windows

▼ Downloads

Custom certificate handler <none>

Max Concurrent Web Requests 3

Catalog Download Timeout 0

▼ Build

Build Addressables on Player Build Use global Settings (stored in preferences)

Enabled

Ignore Invalid/Unsupported Files in Build



Unique Bundle IDs



Contiguous Bundles



Non-Recursive Dependency Calculation



Shader Bundle Naming Prefix Project Name Hash

MonoScript Bundle Naming Prefix Disable MonoScript Bundle Build

Strip Unity Version from AssetBundles



Disable Visible Sub Asset Representation



▼ Build and Play Mode Scripts

= Use Asset Database (fastest)

BuildScriptFastMode (Build Script Fast Mode)



= Simulate Groups (advanced)

BuildScriptVirtualMode (Build Script Virtual Mode)



= Use Existing Build (requires built groups)

BuildScriptPackedPlayMode (Build Script Packed F



= Default Build Script

BuildScriptPackedMode (Build Script Packed Mode)



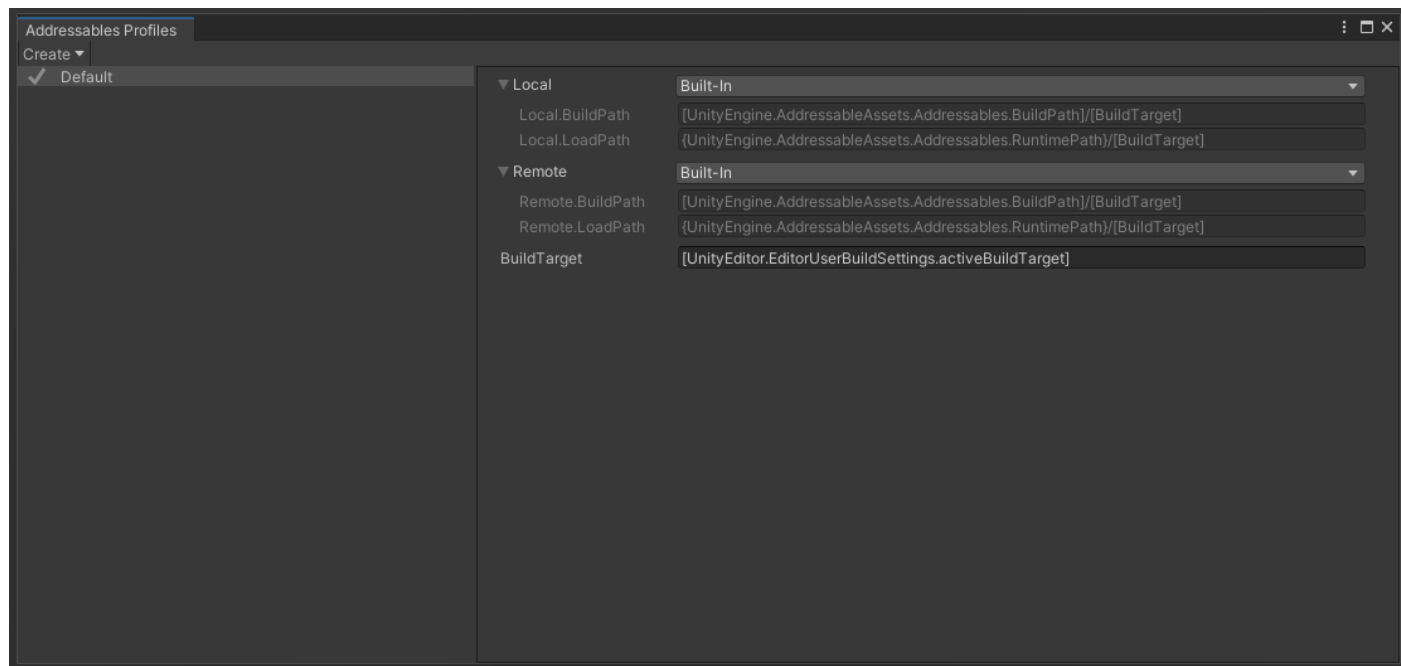
▼ Asset Group Templates

= Packed Assets

Packed Assets (Addressable Asset Group Template)



Для начала нам необходимо изменить настройки профиля по умолчанию, для этого кликните на **Manage Profiles** во вкладке **Profiles**, появится следующее окно:



Здесь нам нужно поменять настройки **Remote**. Нажмите на выпадающий список напротив и выберите в нём **Custom**.

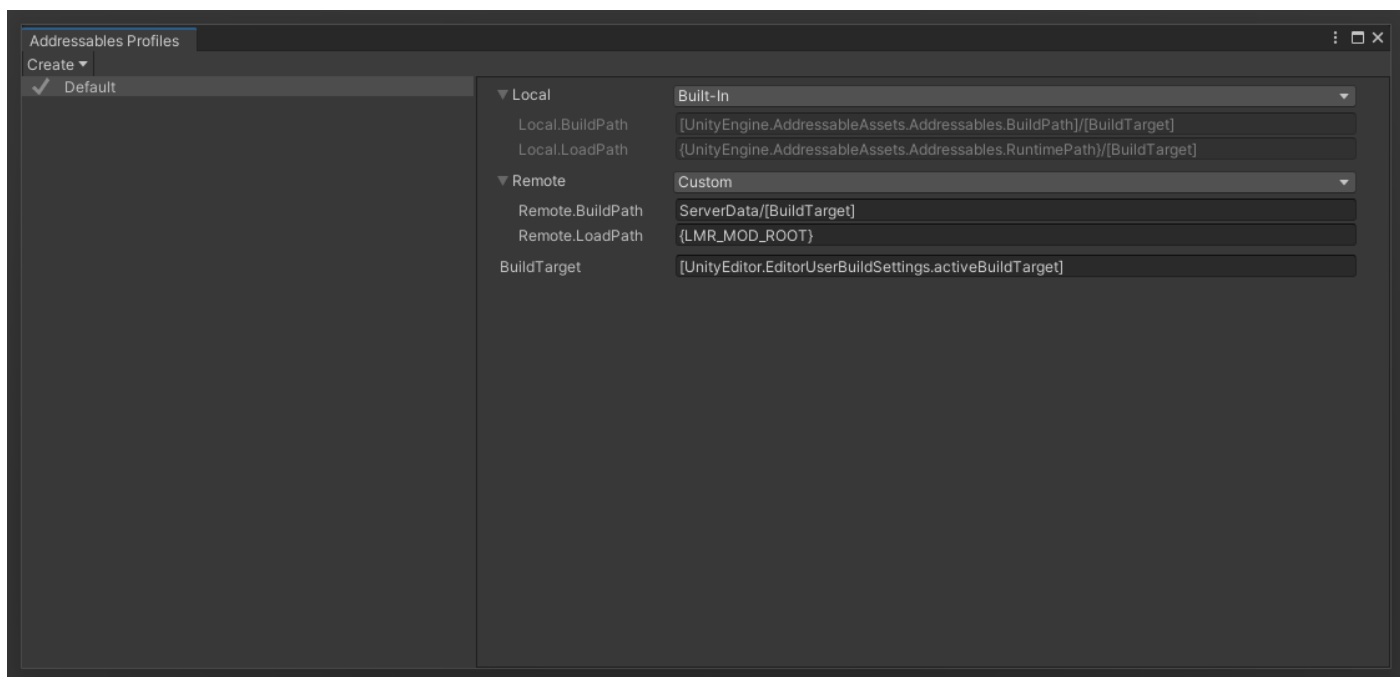
Измените значение **Remote.BuildPath** на следующую строчку:

```
ServerData/[BuildTarget]
```

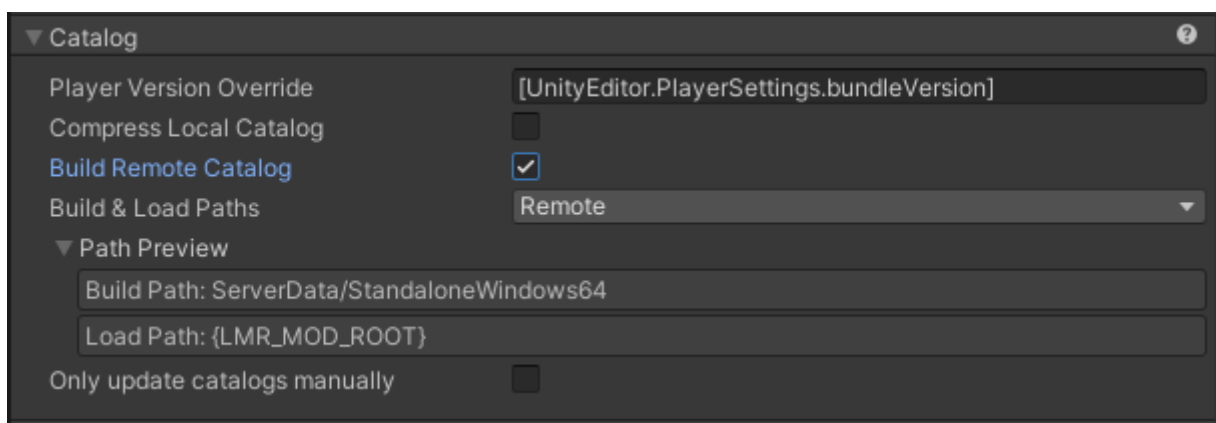
Затем измените значение **Remote.LoadPath**:

```
{LMR_MOD_ROOT}
```

Должно получиться так:

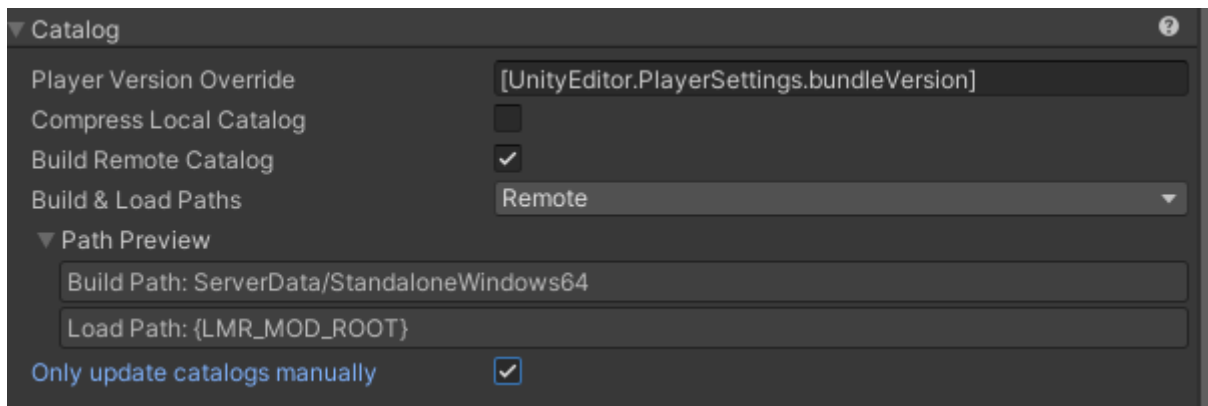


Закройте окно. Далее возвращаемся обратно к окну инспектора, и включаем чекбокс **Build Remote Catalog** во вкладке **Catalog**:

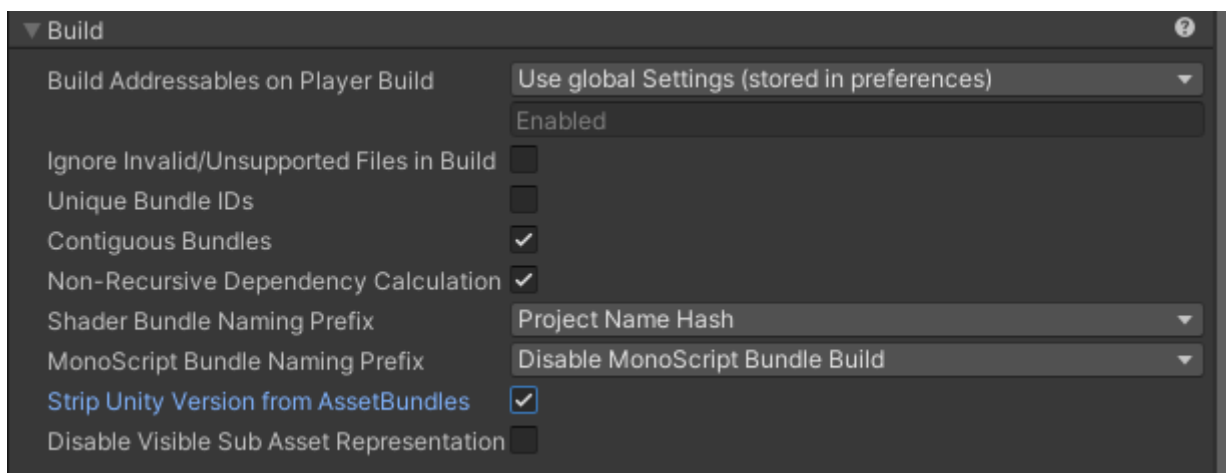


Убедитесь, что в выпадающем списке **Build & Load Paths** выбрано значение **Remote** с настройками, что мы указывали в профиле. У **Build Path** вместо `[BuildTarget]` будет указана ваша платформа — это нормально.

В этой же вкладке включаем чекбокс **Only update catalogs manually**:



Последняя настройка, которую нам нужно изменить у данного ассета — это **Strip Unity Version from AssetBundles**, которая находится на вкладке **Build** — её необходимо включить:

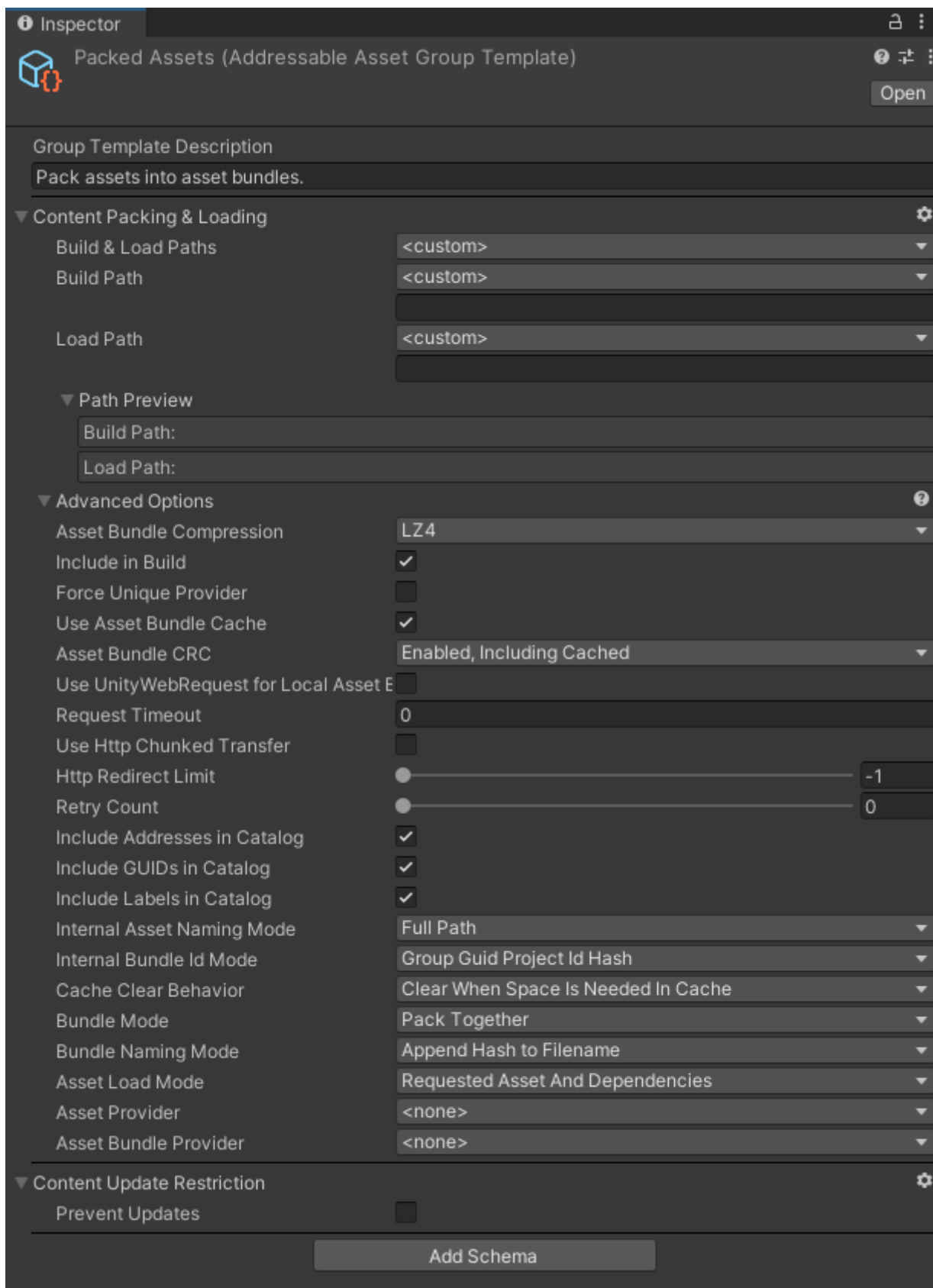


Шаблоны групп

Далее нам необходимо изменить настройки по умолчанию для создаваемых групп. Они хранятся в файле

`Assets\AddressableAssetsData\AssetGroupTemplates\Packed Assets.asset`.

Находим этот файл, кликаем по нему и видим следующее:



Здесь нас интересуют следующие настройки:

1. Build & Load Paths

Необходимо в списке выбрать **Remote**, чтобы использовались

пути, которые мы указали в настройках ассета Addressables.

2. **Use Assets Bundle Cache**

Эту настройку необходимо выключить. Значение **Build & Load Paths** может сброситься на **Local** или **<custom>** — в таком случае поменяйте обратно на **Remote**.

3. **Assets Bundle CRC**

Эту настройку также необходимо отключить. Нажмите на выпадающий список и выберите там **Disabled**.

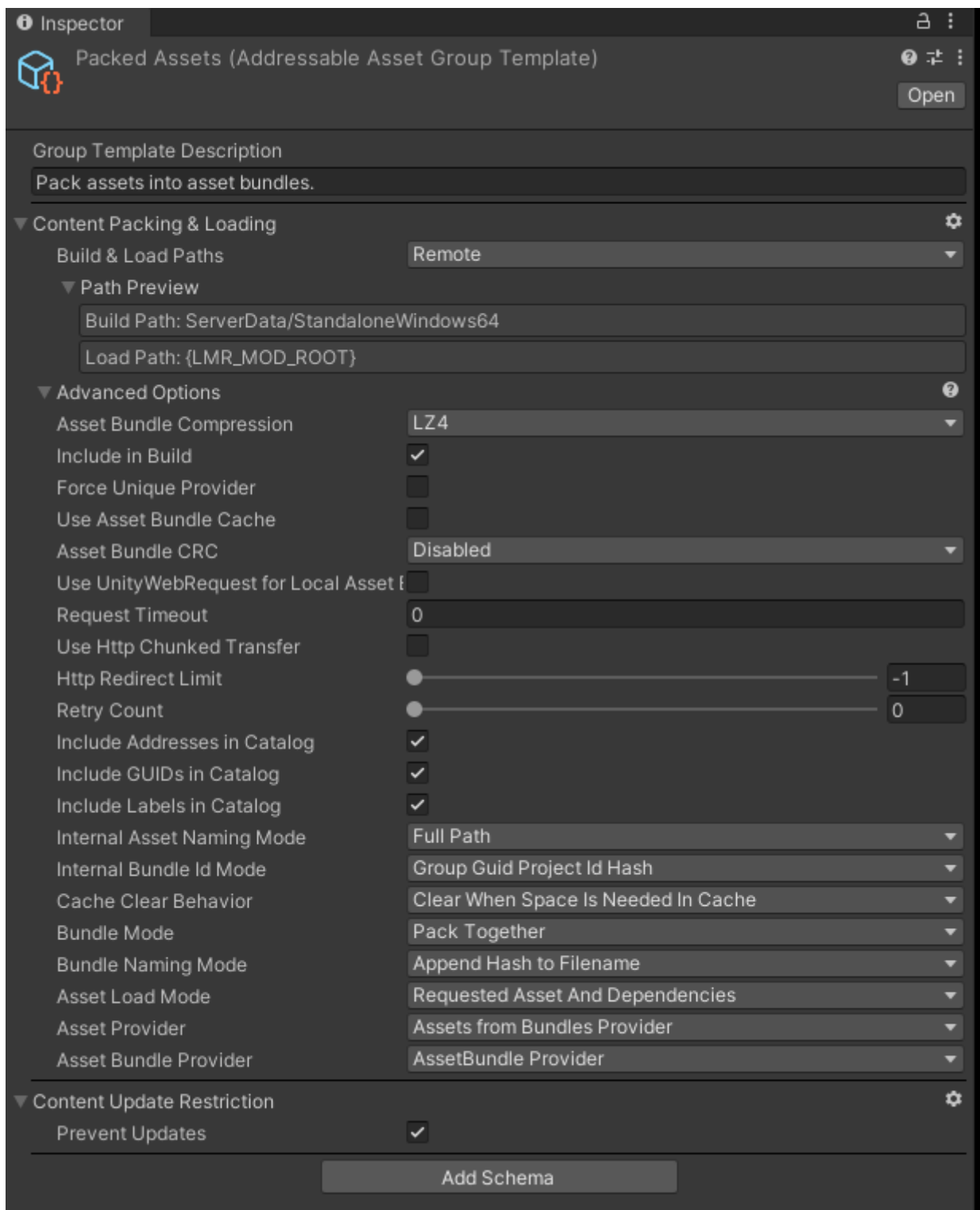
4. **Bundle Naming Mode**

Настройки нейминга бандлов. Рекомендуется выставить **Append Hash To Filename**.

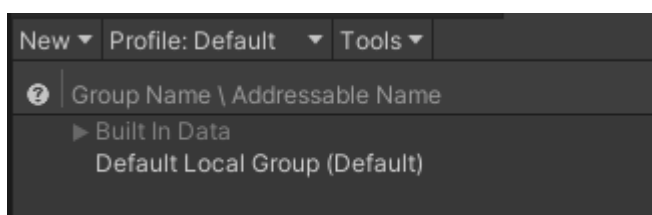
5. **Prevent Updates (вкладка Content Update Restriction)**

Включите данный чекбокс.

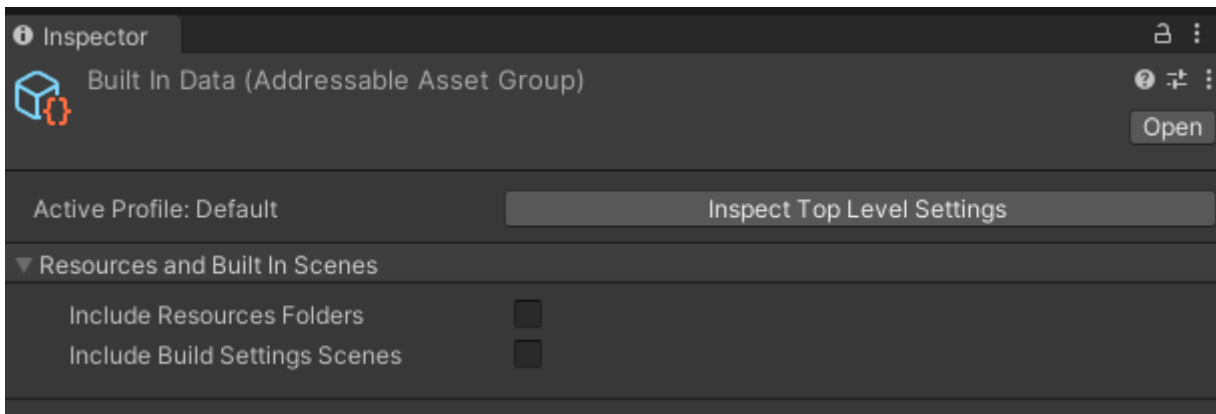
В итоге, после изменения настроек, шаблон будет выглядеть так:



Далее вернёмся к окну **Addressables Groups**. В нём сейчас всего две группы: **Built In Data** и **Default Local Group (Default)**:



Нажмите на **Built In Data** и отключите оба чекбокса в инспекторе:



Создание групп

Как было сказано ранее, внутри окна **Addressables Groups** создаются различные группы ассетов — из этих групп после сборки будут созданы бандлы. Каждая группа — отдельный бандл, в котором хранятся ассеты (зависит от режима сборки, о нём будет далее).

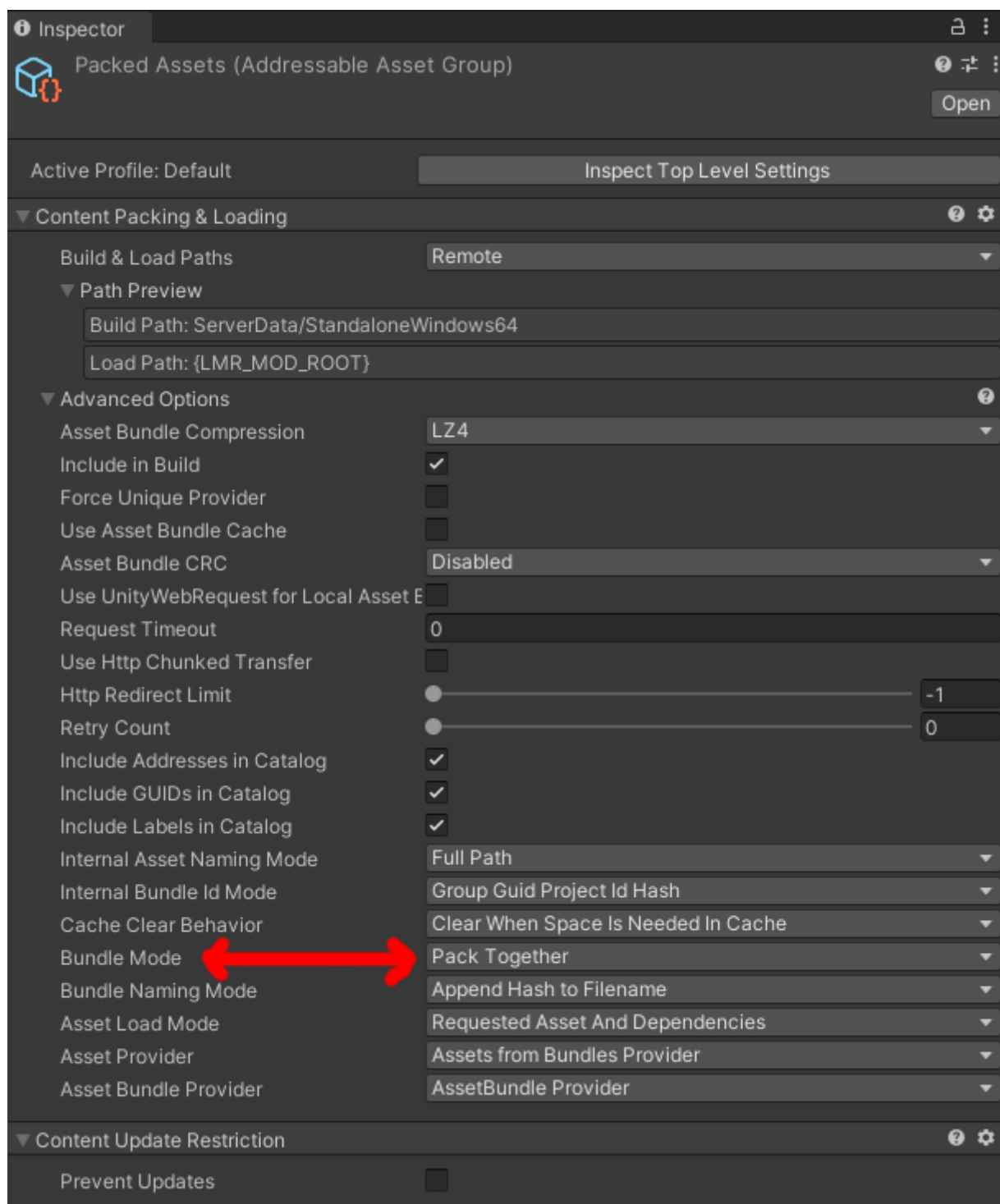
Чтобы создать группу, нажмите правой кнопкой мыши по свободному месту в окне и выберите **Create New Group -> Packed Assets**. По умолчанию название будет **Packed Assets**, но вы можете изменить его, кликнув правой кнопкой мыши по группе и выбрав соответствующий пункт в меню. Названия групп не играют важной роли и могут быть любыми.

Удаление группы Default Local Group (Default)

После создания новой группы, удалите группу **Default Local Group (Default)**, либо измените её настройки в соответствии с настройками шаблона **Packed Assets**. Если вы удалите группу, то следующая за ней группа станет группой по умолчанию.

Режим сборки ассетов

Этот параметр называется **Bundle Mode**, его можно найти в окне инспектора, кликнув по любой группе:



Он определяет, каким образом ассеты внутри группы будут собираться в бандл.

Pack Together — все ассеты из группы будут упакованы в один бандл, и, соответственно, в игре будут загружаться одновременно,

даже если используется только один ассет из группы. Этот режим рекомендуется ставить, если в группе находятся ассеты от какого-то одного объекта - вроде анимированного фона и частей, из которых он состоит.

Pack Separately — для каждого ассета в группе будет создан свой одноимённый бандл, и каждый ассет будет загружаться в игре отдельно, без загрузки остальных ассетов из группы. Этот режим подходит для наборов статичных фонов, текстур и различных звуков. В общем, для всего, что не предполагается загружать одновременно.

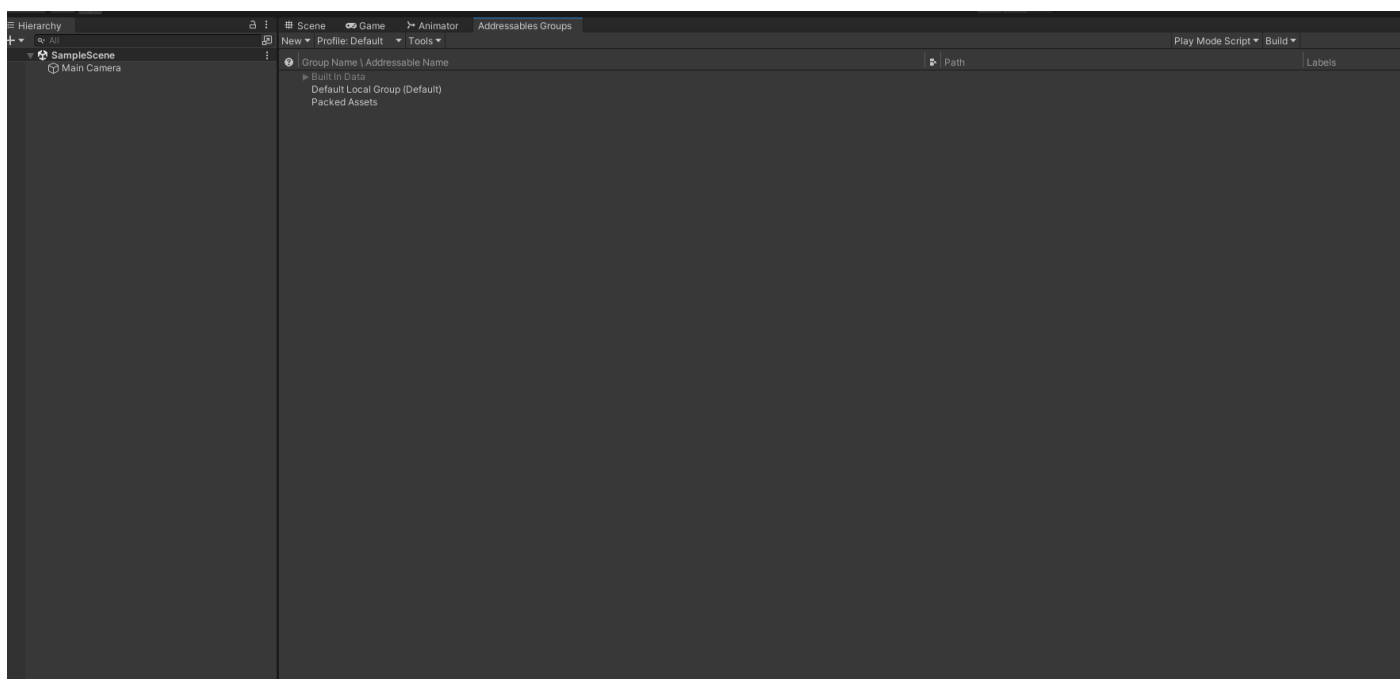
Pack Together By Label — не используется в модах.

Значения для этого параметра вы должны выбирать сами, исходя из ассетов внутри вашей группы и их назначения — это исключительно вопрос оптимизации.

Если вы добавите в группу папку — все ассеты внутри неё будут загружаться в игре одновременно, даже если у группы стоит **Pack Separately**. По сути, папка - это группа внутри группы.

Добавление ассетов в группу

Чтобы добавить какой-либо ассет в группу, нужно перетащить его в необходимую группу внутри окна.

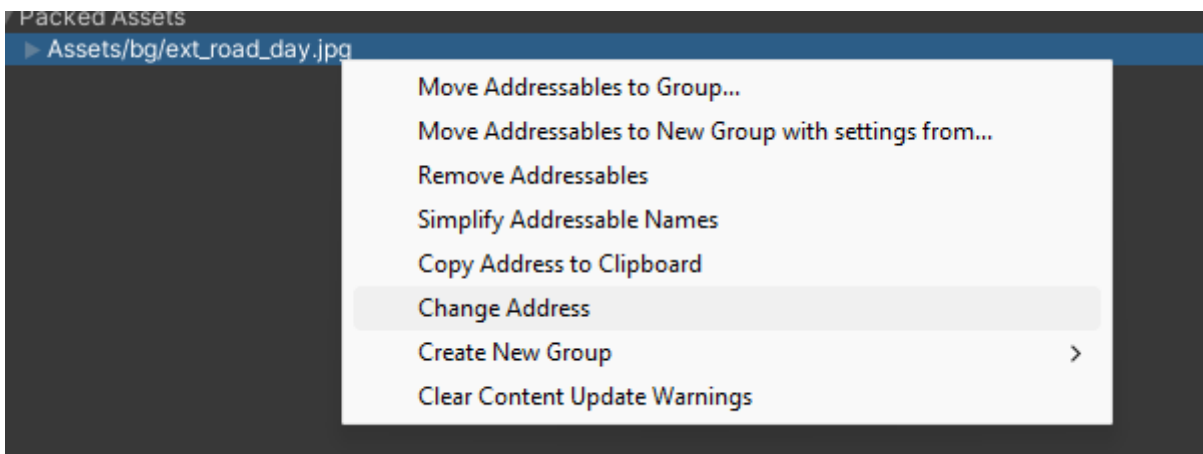


Также вы можете нажать на ассет и в окне инспектора включить чекбокс **Addressable** — в таком случае ассет будет добавлен в группу по умолчанию (вы можете перетащить его оттуда в любую группу).

Нейминг ассетов

После добавления ассета в группу, ему будет присвоено имя, соответствующее его пути внутри проекта. Такое имя называется адресом ассета (или его ключом) — именно на это имя вы будете ссылаться для загрузки ассета внутри файла ресурсов мода — `resources.yaml`.

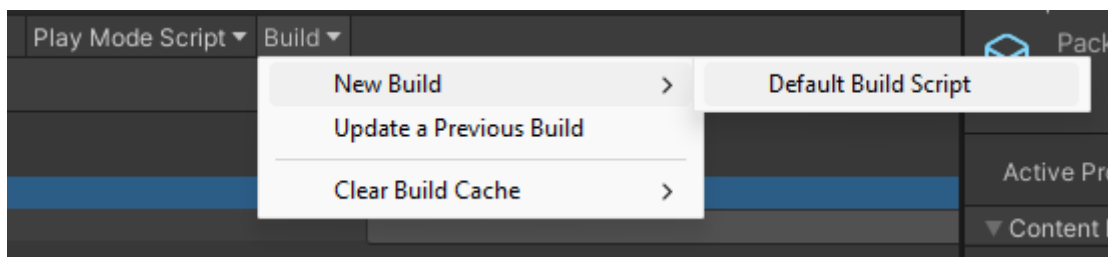
Чтобы изменить имя, кликните по ассету в группе правой кнопкой мыши и выберите **Change Address**:



Ключи чувствительны к регистру.

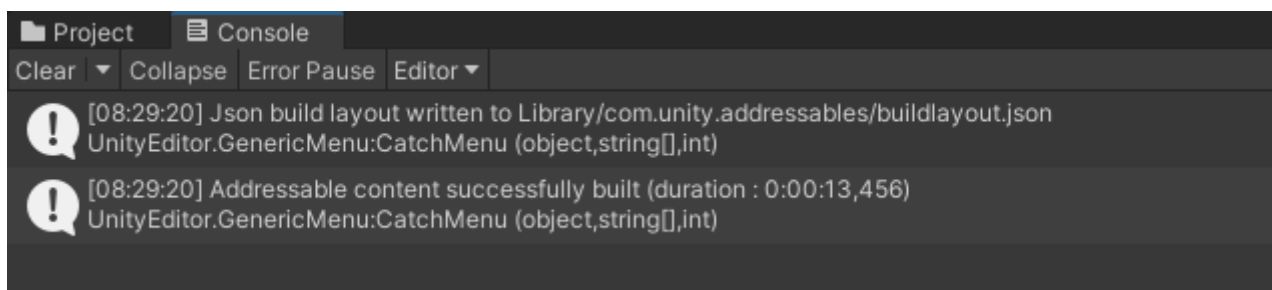
Сборка каталога

После того, как вы создали все необходимые группы и добавили в них ассеты, вам нужно собрать бандлы и каталог для них. Делается это достаточно просто, для этого нужно в окне с группами нажать **Build -> New Build -> Default Build Script**:

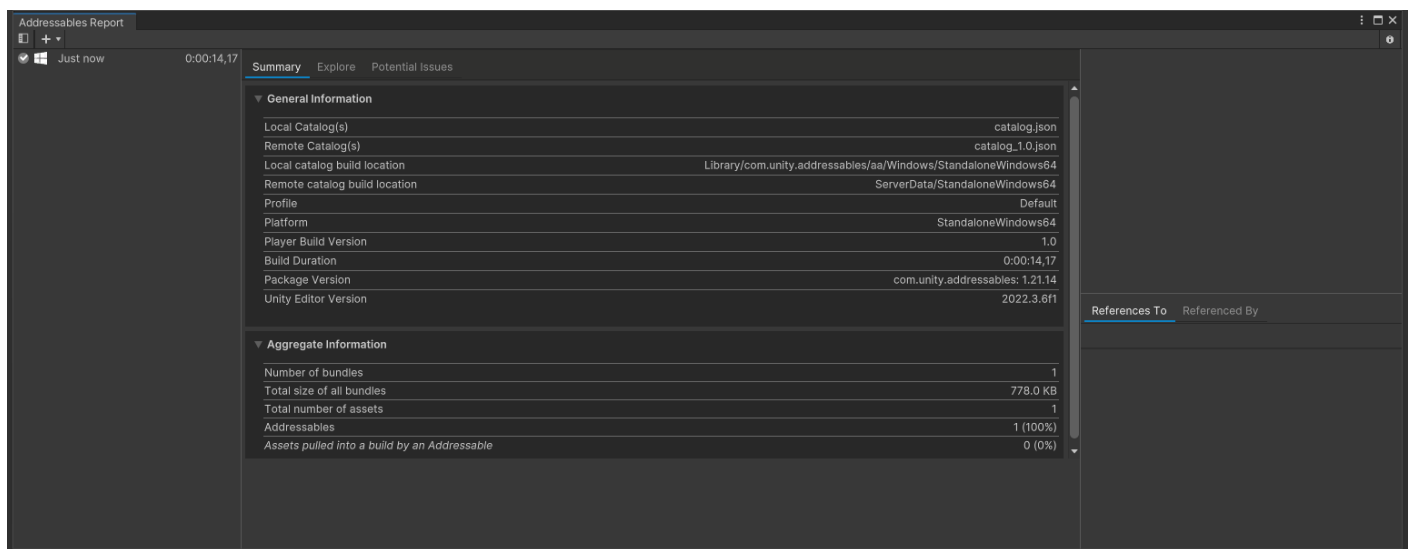


Если это первая попытка сборки, то редактор спросит вас о необходимости включить новую систему отчётов — рекомендуем согласиться.

Если всё прошло хорошо, в консоли редактора появится следующее сообщение:



А также откроется окно с отчётом, если вы его включили:



После этого вам необходимо открыть корневую папку вашего проекта в любом файловом проводнике (не в редакторе) и найти там папку **ServerData**:

Assets	File folder	08.08.2023 8:32	08.08.2023 6:33
Library	File folder	08.08.2023 8:29	08.08.2023 6:33
Logs	File folder	08.08.2023 6:41	08.08.2023 6:33
Packages	File folder	08.08.2023 6:35	08.08.2023 6:33
ProjectSettings	File folder	08.08.2023 8:32	08.08.2023 6:33
ServerData	File folder	08.08.2023 8:29	08.08.2023 8:29
Temp	File folder	08.08.2023 8:31	08.08.2023 6:33
UserSettings	File folder	08.08.2023 6:35	08.08.2023 6:33

Внутри этой папки будут находиться папки с названием платформ, под которые вы собирали каталоги. В каждой такой папке будут лежать файлы бандлов, а также каталог и его хэш:

catalog_1.0.hash	HASH File	1 KB	08.08.2023 8:37	08.08.2023 8:37
catalog_1.0.json	JSON File	4 KB	08.08.2023 8:37	08.08.2023 8:37
packedassets_assets_all_372dc61e54bbe5d4f448b5982dfeab1f.bundle	BUNDLE File	779 KB	08.08.2023 8:37	08.08.2023 8:37

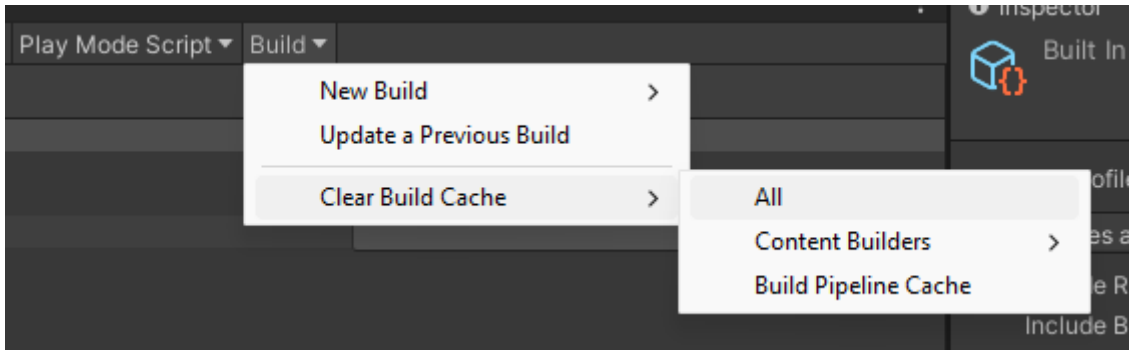
Переместите все эти файлы внутрь директории вашего мода.

После этого вы сможете указать путь к json-файлу каталога в resources.yaml и использовать ассеты из него внутри ресурсов вашего мода.

Если вы решите переименовать **.json**-файл каталога, вам нужно будет также изменить название **.hash**-файла.

Очистка кэша сборки

Перед каждой сборкой каталога рекомендуем очищать кэш предыдущих сборок. Это можно сделать из меню **Build -> Clean Build Cache -> All** внутри окна групп:



Revision #58

Created 3 August 2023 02:41:43 by Admin

Updated 27 November 2023 08:36:57 by Admin