**Этап подготовки компьютера, где будет осуществляться выгрузка**

1. Если вы открыли этот файл до изучения инструкции на главной странице репозитория, то перейдите туда и прочтите сначала ее.
2. После выполнения всех пунктов необходимо установить библиотеку openpyxl.

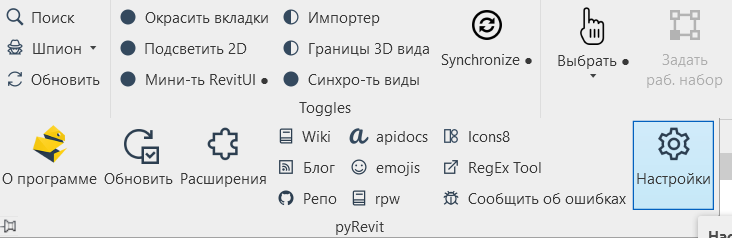
Делается это либо через командную строку (нужен доступ к интернету). В пуске наберите «Командная строка», откройте консоль и введите команду «pip install openpyxl» (без кавычек).

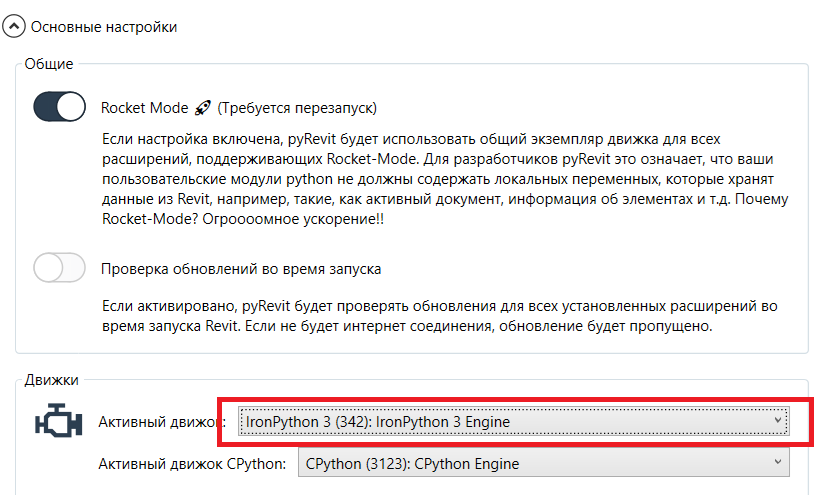
Либо скопируйте содержимое папки из репозитория «\\_for\_python\site-packages» в соответствующую папку вашего Python. Обычно это путь «C:\ Program Files\Python312\Lib\site-packages», в общем случае вам необходимо просто найти папку куда установлен Python и там пройти по пути «Lib\site-packages». Данный механизм альтернативный, предпочтительней ставить библиотеки python через pip (см. выше).

1. Смена движка для pyRevit.

Команда: «pyrevit attach master 342 2022» (без кавычек).

Первое число — это версия IronPython, чтобы посмотреть какая версия есть в плагине, зайдите в «Настройки» - «Основные настройки» и разверните список «Активный движок». Использовать только версии 3.



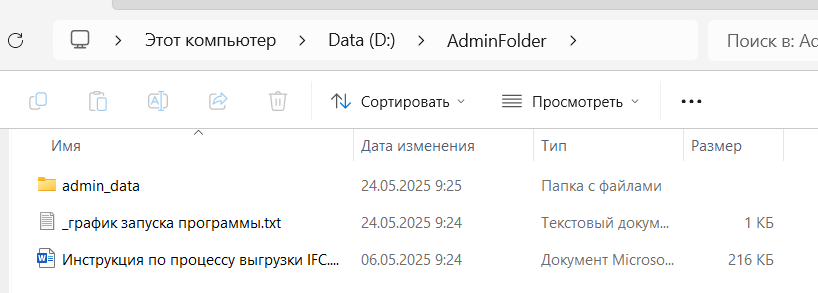


Версия указывается в скобках

Второе числе – это версия Revit для которой вы меняете активный движок. Необходимо последовательно ввести команду для всех используемых на компьютере версий Revit.

**Рекомендуемый формат размещения файлов для настройки процесса**

1. Создаем папку где-то в сетевых папках или на локальном компьютере, в зависимости от того, как вы планируете управлять процессом. Например, если добавлением путей до моделей и файлов маппирования будет заниматься только тот, кто имеет доступ к компьютеру по выгрузке, то можно все разместить на нем же. Если планируется представить функционал группе лиц, которые могут заходить и редактировать Excel табличку и добавлять свои пути, то лучше использовать «сетевой» подход.



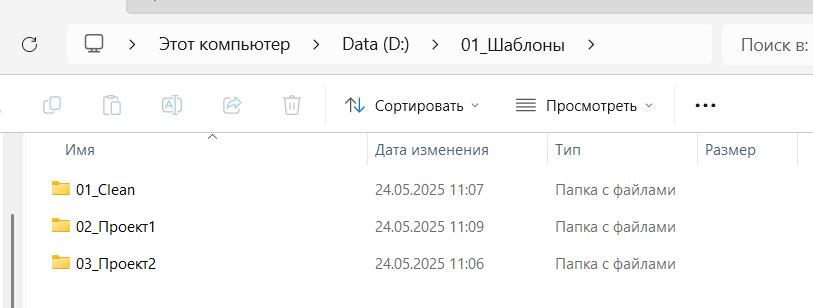
Пример создания папки «AdminFolder» и копирования в нее папок/файлов «admin\_data», «\_график запуска программы.txt» и «Инструкция по процессу выгрузки IFC.docx». Все скопированный файлы находятся в репозитории.

Из скриншота выше видно, что я рекомендую размещать папку «admin\_data» вместе с графиком запуска программы и инструкцией, как ей управлять. Инструкцию следует написать самостоятельно на основании текущей (конкретно под то, как вы развернули процесс у себя).

1. Необходимо определить, где у вас будет находится папка с шаблонами маппирования.

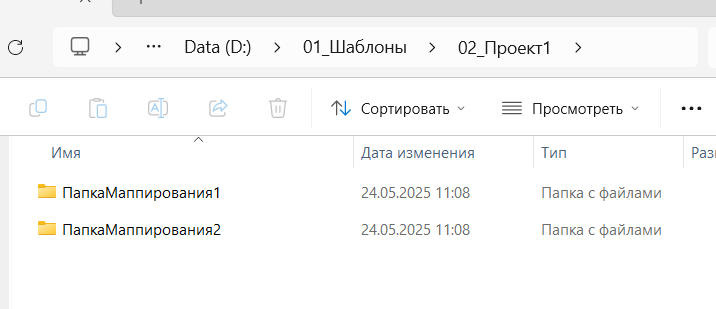
В скрипте предусмотрено использование множество файлов маппирования под разные проекты и даже внутри проектов под разные здания/дисциплины/т.д.

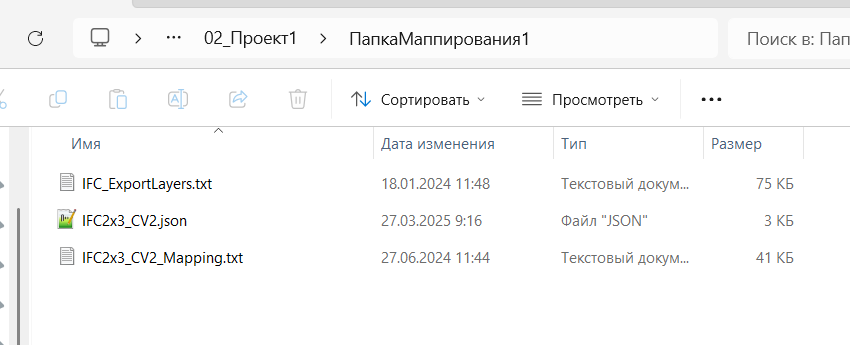
Вот рекомендуемая структура:



Первая папка используется под выгрузку «чистых» и облегченных IFC, чтобы, например, передавать их кому-то. В ней необходимо расположить только \*.json файл с настройками по выгрузке, без какого-либо маппирования и с минимальным уровнем детализации. (об это подробнее ниже)

Остальные папки как раз создаются под проекты. Внутри них названия подпапок выбираете как угодно (вы сами должны продумать, как вы будете разделять разные правила маппирования, если для вас это необходимо).





*Для справки. В папке с маппированием должны быть расположены 3 файла. Первый это сопоставление категорий Revit и классов IFC, второй с настройками выгрузки и третий само сопоставление свойств (маппирование).*

**Этап редактирования скрипта**

Это один из самых важных этапов от которого зависит функционирование всего процесса.

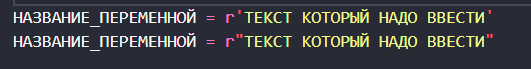
Необходимо в репозитории найти файл «…\const\constants.py»

Он выглядит следующим образом

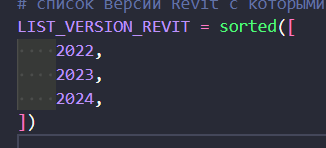
****

Описание значений переменных, которые необходимо изменить:

Общий принцип ввода текста следующий. Могут присутствовать двойные кавычки или одинарные, это одно и тоже, текст нужно перебивать внутри них.



1. IFC\_EXPORTER\_UI - если на компьютере установлена другая версия Revit, то путь необходимо переназначить. Рекомендуется использовать самую раннюю версию из всех что стоят на компьютере, если будут возникать какие-то проблемы, то необходимо поменять.
2. LIST\_VERSION\_REVIT – в этом списке необходимо перечислить все версии Revit, которые планируется использовать в процессе. Добавление осуществляется переходом на следующую строчку и обязательно в конце ставьте запятую, чтобы не ошибиться.



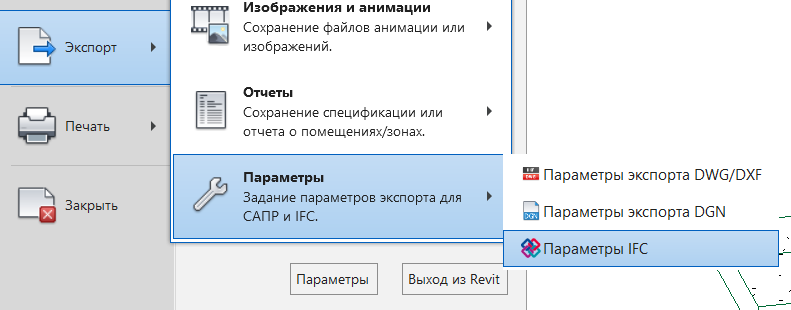
Пример добавление версии 2024

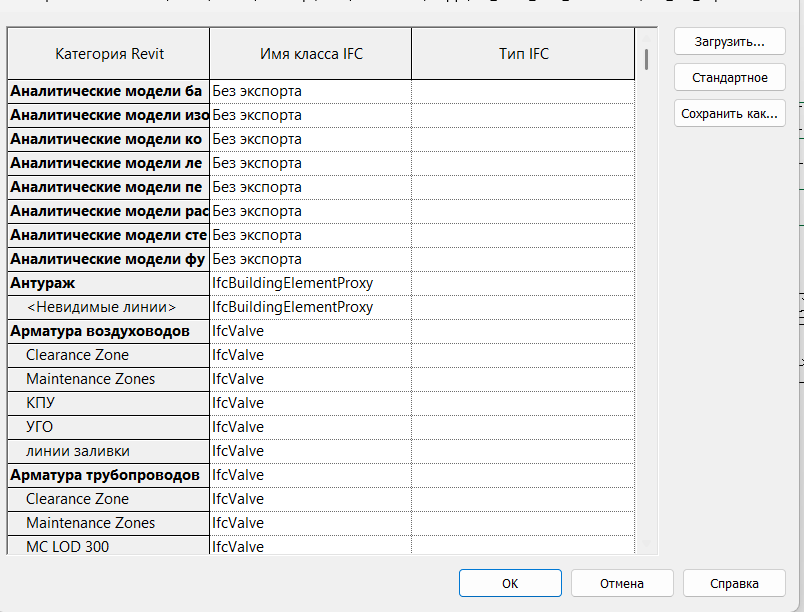
1. LAYERS, JSON\_CONFIG – имена файлов, которые вы планируете использовать для сопоставления категорий и классов IFC и настроек выгрузки.

Эти файлы создаются Revit’ом и вы можете их отредактировать под ваши нужны. Пример этих файлов находится в папке с инструкцией – «Пример настроек выгрузок».

Сопоставления категорий и классов IFC осуществляется через «Файл» - «Эскорт» - «Параметры» - «Параметры IFC». Редактировать этот файл можно прям в этом окне, подробности описаны в «Руководство по работе с IFC в Autodesk Revit» - <https://damassets.autodesk.net/content/dam/autodesk/draftr/2528/revit-ifc-handbook-ru.pdf>

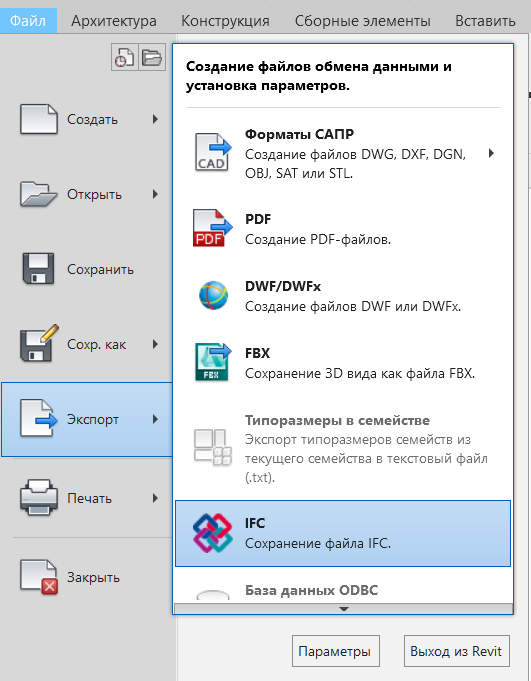
После настройки сохраните полученный файл через «Сохранить как…».

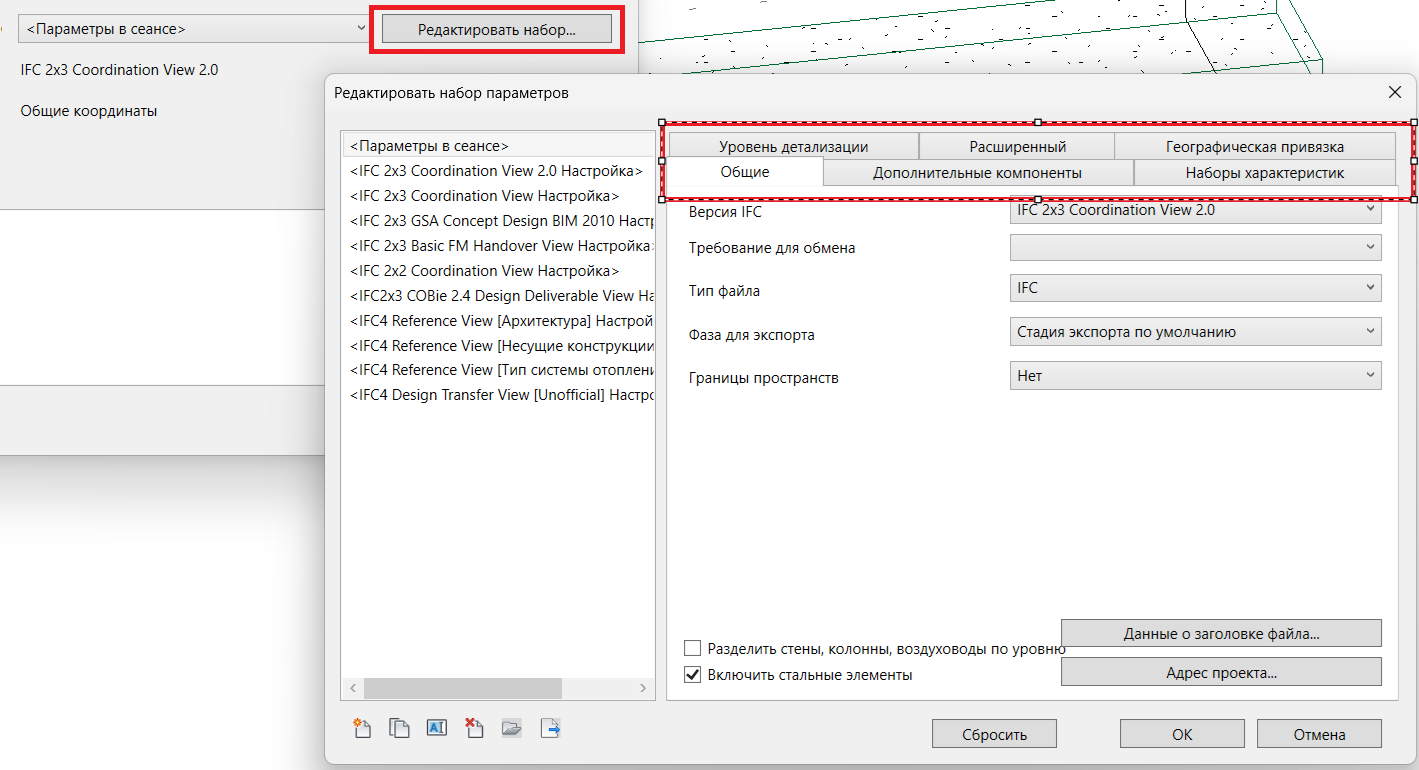




Можно его переименовать или назвать, как в скрипте – «IFC\_ExportLayers.txt».

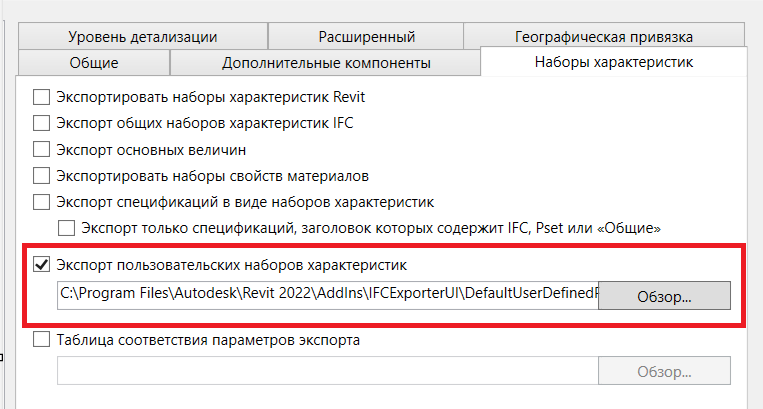
Файл с настройками по выгрузке можно получить следующим образом «Файл» - «Эскорт» - «IFC».



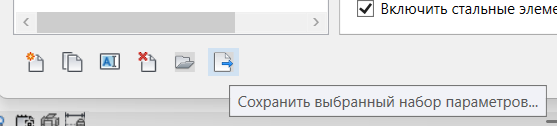


За подробностями, как настраиваются все параметры можно обратиться к Руководству выше или к тематическим статья в интернете.

Главное при настройки параметров выгрузки указать соотвествующий файл маппирования, который должен располагаться рядом с файлом \*.json

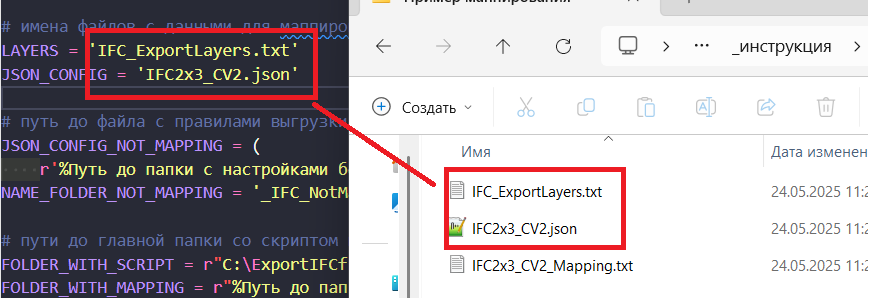


Здесь указать актуальный путь



Далее сохраняете полученный набор параметров через соответствующую иконку. Можно его переименовать или назвать, как в скрипте – «IFC2x3\_CV2.json».

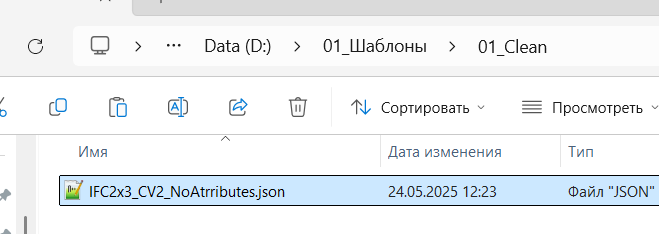
Не забудьте, если захотите переименовывать файла, то значения переменных тоже необходимо будет изменить.



Важный момент, во всех папках названия файлов должны быть такие, как вы назначили в этих переменных.

1. JSON\_CONFIG\_NOT\_MAPPING – путь до файла с настройки для «чистой» выгрузки. Указываете абсолютный путь до файла.

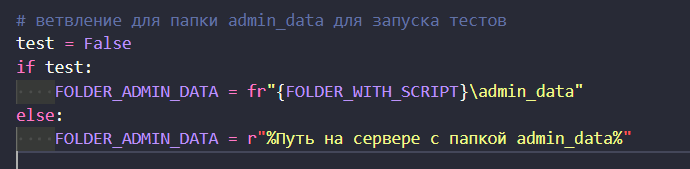
Для примера выше это был бы путь -"D:\01\_Шаблоны\01\_Clean\IFC2x3\_CV2\_NoAtrributes.json"



NAME\_FOLDER\_NOT\_MAPPING – переменная для названия папок, который будут создаваться рядом с основной выгрузкой IFC, называть можете как угодно.

1. FOLDER\_WITH\_SCRIPT – основной путь до скрипта на вашем компьютере. Путь необходимо указывать до которую вы скачали с репозитория, по умолчанию она называется «ExportIFCfromRevit».
2. FOLDER\_WITH\_MAPPING – это путь до папки, расположение которой описано выше. В примере она называется «01\_Шаблоны».
3. FOLDER\_ADMIN\_DATA – это путь до основной папки «admin\_data» с данными по управлению всем процессов. Переменную где написан текст необходимо изменить под ваш случай.

Если напишите в переменной test = True. То перенаправите работу скрипта на папку «admin\_data» которая расположена в скачанном репозитории, удобно использовать для локальных тестов. По умолчанию настроено на работу из другой папки, которую вы назначите в блоке «else».

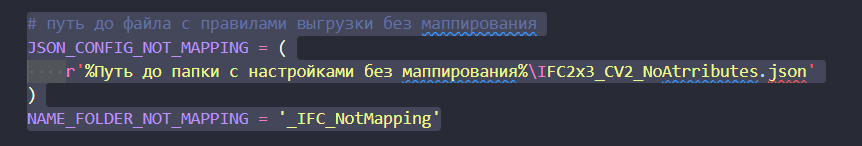


Последующие переменные трогать не следует, только если не хотите переименовать все файлы таблиц и логов.

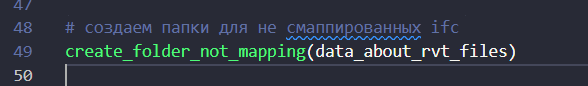
Примечание:

Если вам не требуется выгружать «чистые» IFC, то необходимо удалить следующие строчки кода

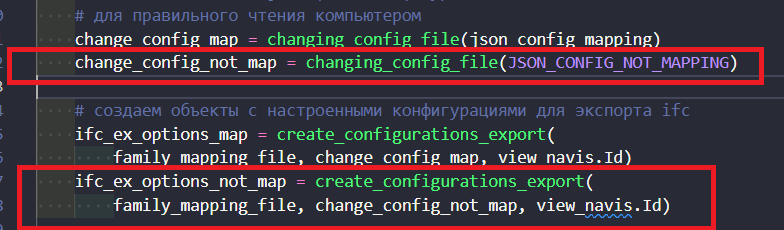
Файл constants.py

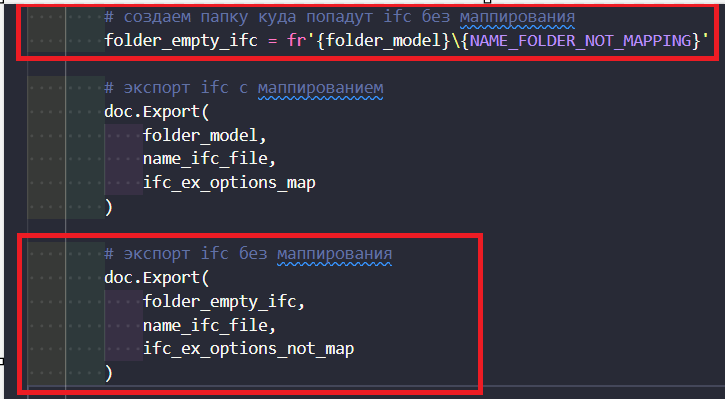


Файл main.py



Файл ExportIFC.py





Этап настройки скрипта под ваше окружения окончен.

**Описание администрирования процесса выгрузки IFC из Revit**

Важные моменты, которые необходимо понимать для правильного функционирования процесса:

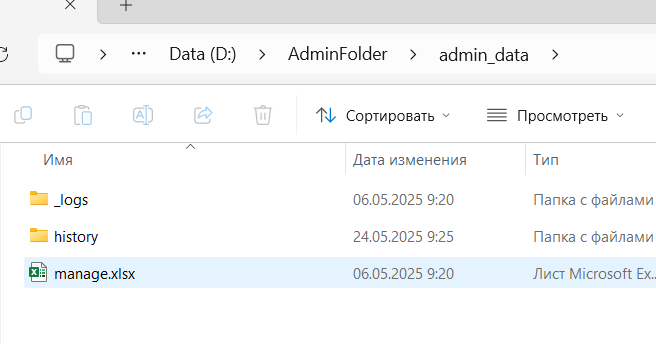
1. Выгружаться IFC будут только из файлов Revit, которые выполнены в версиях программы, указанных в файле constants.py (см. описание выше). *Также следует заметить, что скрипт работает только с \*.rvt форматом, другие форматы Revit он не будет видеть. Также в наименовании файла должна быть одна точка, которая используется для расширения. Примеры наименований, игнорируемые скриптом: Титул1\_AR.ifc.0001.RVT, Титул2\_AR.0001.rvt, Титул3\_KR.KM.rvt и т.д.*
2. В моделях обязательно должен быть настроен 3D вид с именем «Navisworks» (без кавычек). На данном виде должны быть скрыты все элементы, которые не нужно отображать в IFC, таким образом мы можем контролировать, что передается при экспорте. Если данного вида нет, то будет сознан лог с перечислением таких моделей (об этом ниже)
3. Структура папок, где хранятся модели с файлами rvt должна быть примерно следующей:

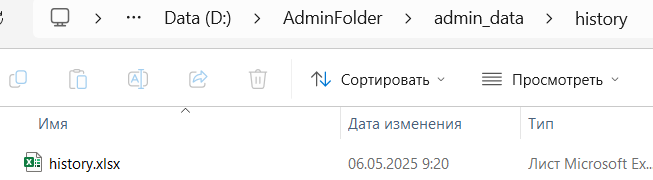


На скриншоте выше файлы IFC будут сохраняться в предыдущую папку от «\_Actual\_Rvt», то есть в «01\_Files». При заполнении файла «manage.xlsx» необходимо помнить, что файлы сохранятся в предыдущую папку от указанной в столбце «Путь до папок» (подробности ниже)

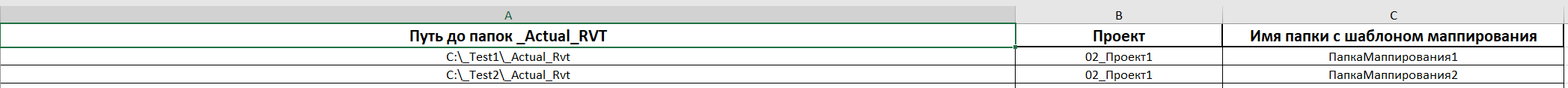
Описание настроек:

1. Если опираться на пункт выше по рекомендуемому формату размещения файлов, нами для примера была создана папка «AdminFolder» (вы могли назвать иначе). В ней находится следующие папки - «admin\_data», которая содержит файл «manage.xlsx», папки «\_logs» и «history», внутри которой хранится файл «history.xlsx». Данные файлы и папки нельзя удалять или переименовывать. Все остальные файлы, которые будут создаваться в процессе работы скрипта можно удалять (логи, файлы для заданий и т.д.), но необязательно.





1. В файле «manage.xlsx» необходимо внести данные в таблицу на листе «Path», которые содержат абсолютные пути до папок c RVT файлами, которые нужно выгружать и названия проекта/папки с данными для маппирования (пропуски по строчкам не допускаются, список должен быть цельным).



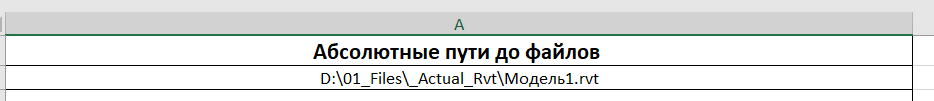
Скрипт собирает RVT только в корне той папки, которую вы прописали в таблице выше.

Название проектов/папок с данными для маппирования находятся по пути, которые вы завели на этапах выше. В примере выше это папка «01\_Шаблоны».

Также в «manage.xlsx» размещен лист «Пример таблицы Path» с примерами, можете его подправить под свой случай, чтобы людям было удобнее. В скрипте этот лист не участвует.

*Так же предусмотрена возможность добавлять отдельные модели в список игнорирования.*

Для того чтобы пропустить какую-то модель необходимо в файле «manage.xlsx» открыть лист «IgnoreList» и добавить абсолютный путь до файла модели. Такой функционал дает возможность точечно пропускать файлы с ошибками или неподдерживаемыми версиями Revit.



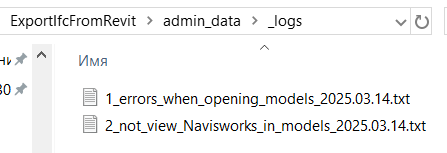
Пример добавления абсолютного пути до rvt

1. В файле «history\history.xlsx» на листе «History» расположены данные о предыдущем состоянии файла, они нужны для оптимизации работы алгоритма в том случае если файл не менялся и нет потребности его выгружать. Пока это работает только с привязкой к дате и времени сохранения файла. Если нужно перевыгрузить файл IFC, но модель не изменялась, то можно удалить самую свежую дату изменения файла rvt в этой таблице (удалять нужно всю строчку, чтобы не было пропусков) или просто удалить сам IFC. В этих двух случаях скрипт поймет, что нужно начать выгружать IFC. Когда файл обновился относительно данных из последнего запуска скрипта, то скрипт по умолчанию поставить его на выгрузку.

*Примечание: нельзя долго держать открытым файл history.xlsx так как скрипт работает на запись и если файл кто-то займет, то скрипт не сможет его сохранить. Если хочется посмотреть список с данными о файлах, то лучше скопировать «history.xlsx» к себе на компьютер и работать с ним локально. Если же требуется удалить строчки в этом файле (чтобы скрипт перевыгрузил эти модели заново), то после проведения этой процедуры не забудьте его сохранить и закрыть.*

1. Варианты файлов с логами после отработки скрипта

Общий вид создающихся txt файлов логов



Предусмотрено создание 4-х вариантов файлов:

1. Первый лог называется «1\_errors\_when\_opening\_models\_%дата создания лога%»

Это самый неблагоприятный лог, так как он отвечает за то, чтобы фиксировать модели, которые не смогли открыться алгоритмом по различным причинам. Таких причин, может быть, множество и некоторые из них могут потребовать доработки алгоритма скрипта.

Первое что необходимо сделать, это открыть модель в соответствующей версии Revit. Далее посмотреть какие окна появляются в процессе открытия, если возникают какие-то ошибки и Revit дает возможность открыть файл, игнорируя их, то лучше так и сделать, а затем пересохранить модель. Большая вероятность, что при следующем запуске скрипта модель сможет открыться.

1. Второй лог называется «2\_not\_view\_Navisworks\_in\_models\_%дата создания лога%»

Данный лог создается, когда из модели не удалось выгрузить IFC по причине отсутствия внутри нее 3D вида «Navisworks», эти данные нужно передавать разработчику модели, чтобы он настроил в своих моделях такой вид.

1. Третий лог может появляться с разной степенью вероятности, он отвечает за фиксирование файлов, которые созданы ***в версиях Revit выше, чем заявленные в скрипте***. Называется он «3\_not\_supported\_versions\_rvt\_%дата создания лога%». В скрипт можно добавлять новые версии Revit, но предварительно данные версии необходимо установить на серверный компьютер, который занимается выгрузками.

*Доп. информация:*

*Все модели проходят алгоритм определения версии Revit, в которой они были сохранены и затем только передаются в скрипт с нужными настройками.*

1. Четвертый лог, был предусмотрен, чтобы отлавливать модели, где определить версию Revit не получилось. Он называется «4\_not\_found\_versions\_in\_rvt\_%дата создания лога%». Отвечает он за то, чтобы проверять в моделях отсутствие информации о версии Revit в которой они были созданы, данная информация нужна для последующего функционирования алгоритма.

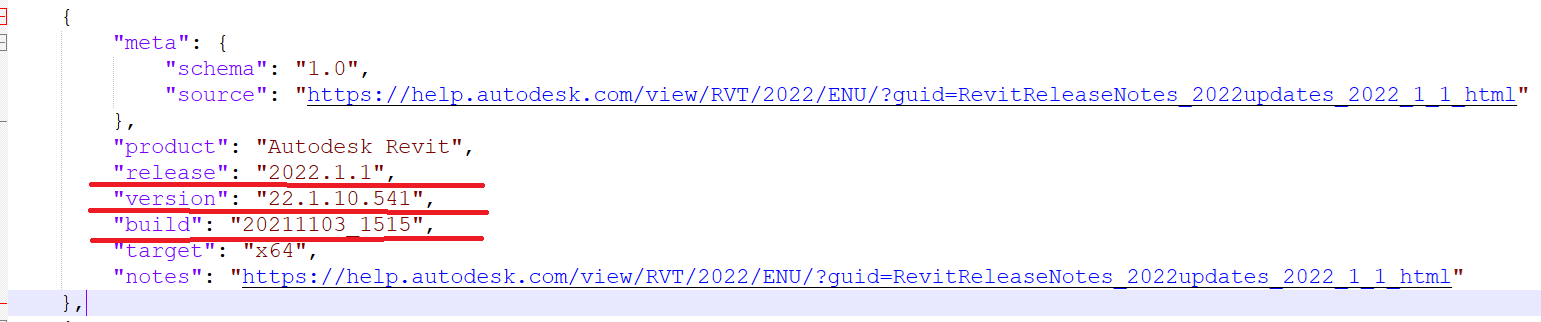
**Лайфхак для pyRevit**

В плагине используется файл «pyrevit-hosts.json», который содержит в себе перечень версий Revit (в том числе промежуточных), содержание всех версий из которых выгружаются файлы обязательно в этом файле, иначе он не поймет в какой программе открывать ту или иную версию. Так как данный файл редактируется разработчиками и какие-то новые и актуальные версии могут туда не попадать, то можно его редактировать самостоятельно.

Путь до файла «pyrevit-hosts.json»:

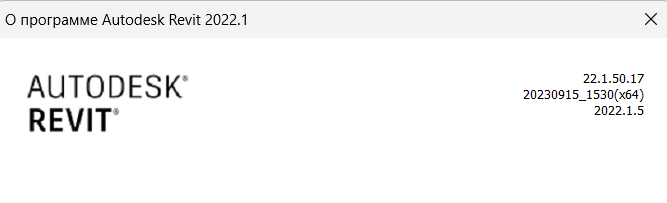
%Путь куда вы установили pyRevit%\pyRevit-Master\bin

Надо скопировать весь текст какой-то близкой к вашей добавляемой версии.



Копируется от фигурной скобки до фигурной скобки.

Далее меняется текст в кавычках со скриншота выше. Эти данные берутся либо с официального сайта Autodesk под вашу версию, либо из самого Revit в «Справка» > «О программе»



Пример (для версии в json файле не пишется (x64))