

Коллега, здравствуйте.

Сегодня тема — использование инструмента Книга (Book) при вёрстке файлов с концевыми сносками. Такой опции в индизайне нет, и чтобы и книгу сверстать, и концевые сноски в конце книги иметь, в стандартном варианте надо ворочать весь файл. Концевые сноски появились в версии 2018, и к сожалению, чем новее версия, тем она медленнее работает. Эта тема обсуждается на разных форумах, но лучше решения — делить файл на главы, верстать каждую отдельно и потом собирать всё в инструменте Книга — пока не предложено.

Но если в главах есть концевые сноски, то Книга не поможет.

Вот для простоты объяснения три файла с концевыми сносками, в первом одна обычная и одна концевая сноски, во втором их по две и в третьем по три.

Вот как выглядят индизайновские файлы после импорта этих doc-файлов.

И вот они собраны в книгу. Надо бы иметь концевые сноски в конце книги, а они в конце каждой главы. И эти страницы с концевыми сносками влияют на нумерацию страниц. Вторая глава должна идти сразу после первой, а тут между ними страница с концевыми сносками.

Но можно скриптом на месте каждой концевой сноски поместить примечание (note) и спрятать в нём текст сноски. И страницу с этими сносками вообще убрать. В результате мы получим обычный

файл, его можно оформлять стилями, и прочее, и никаких лишних страниц.

Эту задачу решает скрипт **#1.KeepTextOfEndnoteInNote.jsx**, его надо запустить в каждом файле Книги.

В книгах с концевыми сносками всегда есть указание названия главы, а потом идут сами такие сноски. Поэтому этот скрипт должен запомнить название главы, точнее, пользователь должен его ввести. Для этого есть отдельное поле.

Кроме того, на месте каждой сохранённой концевой сноски мы поставим её заместитель, по умолчанию это ноль в верхнем регистре. Данный заместитель смоделирует наличие знака в вёрстке, без него есть вероятность, что абзац со сноской и без сноски могут быть скомпонованы по-разному.

Эти заместители оформляются служебным символьным стилем **endnotePlaceholder**, размещённым в группе **charStyleGroupForEndnote**.

Если концевая сноска в таблице, то скрипт закончит работу с сообщением об ошибке. Эти особые случаи обрабатываются так:

- 1) Клавишами Ctrl+Z откатить вёрстку к состоянию до запуска скрипта.
- 2) Концевая сноска берётся в буфер, на её месте ставится какой-нибудь знак, например, # в верхнем регистре.
- 3) Сама сноска вставляется в текст за пределами таблицы, например, в конце заголовка таблицы.

На третьем шаге мы вернём её на место этого знака #.

И после того как скрипт завершит работу, все концевые сноски будут в примечаниях, на месте этих сносок будут нули в верхнем регистре, и исчезнут страницы, на которых были концевые сноски. Если в обработанном этим скриптом файле надо добавить еще концевую сноску, то её надо добавить и запустить этот скрипт ещё раз. В файле появятся ещё одно примечание, ещё один ноль — заместитель концевой сноски, и будет удалена страница, появившаяся при добавлении новой концевой сноски. Теперь эти файлы можно верстать.

~~~~~

После того, как всё оформлено так, как надо, мы должны превратить эти отдельные файлы Книги в один файл, чтобы потом извлечь из примечаний концевые сноски в конец этого объединённого файла.

Задача сбора всех файлов в один решается скриптом

**#2.MergeBookDocsToOneFile.jsx**. Сводный файл называется **It'sAllBook.indd**. При разработке этого скрипта, ему уже несколько лет, была замечена странная, трудно воспроизводимая ошибка — иногда картинки смещаются влево. Чтобы этого не было, скрипт при копировании объектов из файла Книги в сводный файл запоминает их координаты, а потом проходит по этим объектам и перезаписывает текущие координаты на сохранённые. Это гарантирует, что всё будет на своем месте.

Вот короткое видео, показывающее, что происходит, когда есть только перенос, и что, когда координаты каждого объекта обновляются.

=====

Вторая задача — связать текстовые фреймы в единую цепь — проста для скриптинга, т.к. у фреймов текстового потока есть информация, из какого фрейма пришел текст, и куда уйдёт. Но основании этих данных просто восстановить все текстовые потоки.

Третья задача — извлечение концевых сносок из примечаний, решается скриптом **#3.TextFromNoteToEndnote.jsx**. Скрипт просматривает всё текстовые фреймы документа в поисках нужных примечаний. Из примечаний, созданных скриптом **#1.KeepTextOfEndnoteInNote.jsx**, извлекаются концевые сноски, и название главы перед ними, а сами примечания и ноли-заместители удаляются.

Останется только оформить абзацными стилями этот собранный из содержимого примечаний блок концевых сносок.

Когда-то я делал книгу по этой технологии, и думал, что после завершения работы третьего скрипта работа фактически закончена. Но пришло сообщение, что надо «перезалить примечания»:



**Михаил** 19:04  
Взял в работу.

11 июня



**Андрей** 11:24  
Михаил, здравствуйте! Отправляю на перезаливку примечания.



Примечания\_ЮМ.doc

12 июня



**Михаил** 2:26  
А это засада, Андрей.  
Дело не в названии, по факту это не примечания, а концевые сноски.  
Книга свёрстана, и сейчас щелкаешь по тексту концевой сноски — попадаешь внутрь книги, где она есть, точнее, её номер.  
Если сейчас просто залить этот текст ("отправляю на перезаливку"), то все ссылки на концевые сноски внутри книги станут выглядеть так: <?>

Там больше сотни концевых сносок, оформленных так: номер, точка, пробел и дальше абзац текста новой концевой сноски. Думаю, вы понимаете, что тягомотина аккуратно выделять текст от первого пробела новой сноски до конца абзаца, брать выделенное в буфер, в обновляемой сноске выделять текст от табуляции до перевода строки и заменять выделенное на содержимое буфера, и так больше ста раз, мне совсем не нравилась. Вёрстка должна быть в кайф, нам должен нравиться этот процесс. А перспектива такого тупого копирования — совсем не айс.

И был сделан ещё один скрипт **#4.UpdateEndnotes.jsx**. Перед его запуском надо поместить новые сноски в отдельный файл и

запустить скрипт DoTextOK, чтобы сохранить начертание курсив, полужирный, полужирный курсив.

*Так выглядит docx-файл с исправлениями. Есть варианты начертания. Закрываем его, затем в новый файл импортируем этот docx, и что видим?*

*Курсив и полужирный потерялись. Это ужасный дефект этого формата, docx. Используйте doc или rtf, с ними потеря начертания не будет.*

Теперь надо сделать на экране два окна — одно с новыми сносками, а на втором текст обновляемых сносок. Теперь надо запустить скрипт. Его интерфейс проще некуда, две кнопки '+' и '>'. Ставишь курсор в текст новой сноски и нажимаешь '+'. Программа находит текст после первого пробела до конца абзаца и выделяет его. Становится доступной кнопка '>'. Ставишь курсор в обновляемую сноску, программа выделяет текст между знаком табуляции и переводом строки и заменяет его на содержимое буфера. Кнопка '>' снова недоступна, т.к. в буфере нет текста новой концевой сноски.

И так прощелкать эти сноски гораздо быстрее и проще, нежели тупо копировать.

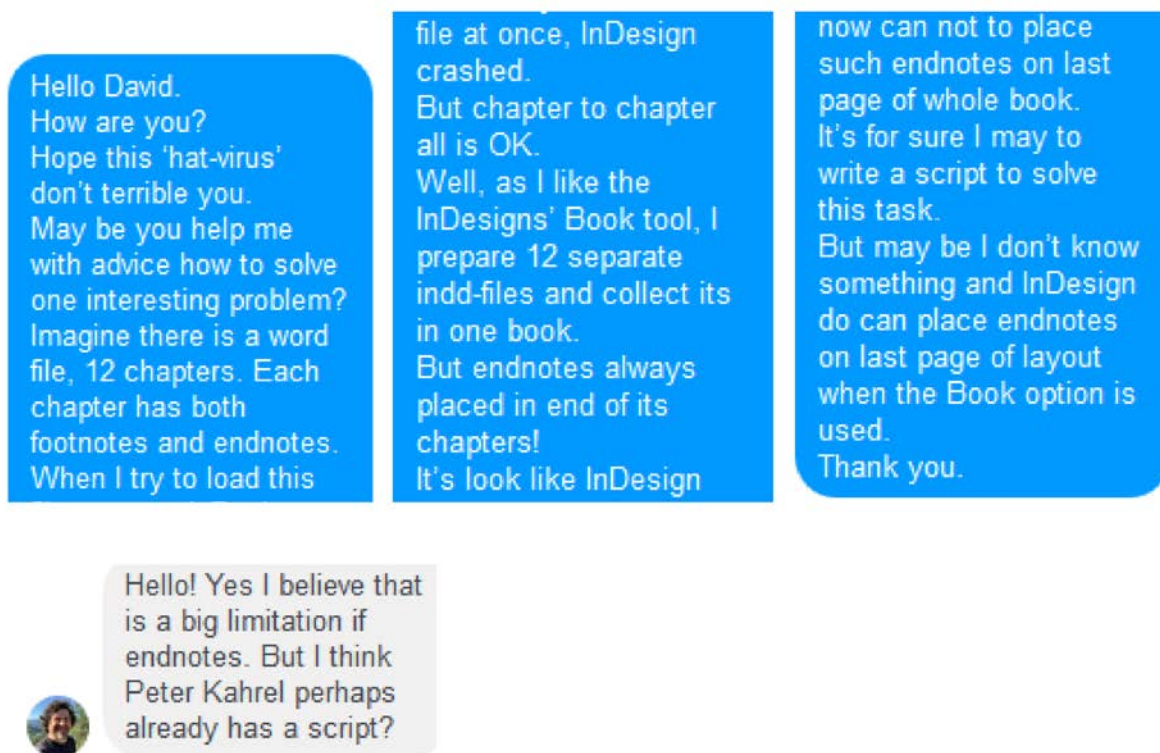
В принципе это, пожалуй, всё, что есть сказать об этом решении.

Итог — есть комплект программ, позволяющий использовать инструмент Книга при работе с концевыми сносками. Программа объединения файлов книги в один файл, пожалуй, тоже может быть интересна, поэтому её можно купить отдельно.

\*\*\*

Когда я это сделал, был червячок сомнения, вдруг я что-то не знаю, вдруг в Книге где-то есть опция работы с концевыми сносками.

Написал Дэвиду Блатнеру.



Он ответил, что это большое ограничение концевых сносок, что они не могут использоваться в инструменте Книга. И добавил, что, может быть, Питер Карел уже сделал нужный скрипт.

Я не сомневаюсь, что Карел может сделать такой скрипт. И если сделает, интерфейс программы будет на языке Шекспира и Черчилля. Но тут программа уже есть. И не одна, а комплект. И на том языке, на котором мы думаем и говорим.

На этом всё. Голос за кадром — Михаил Иванюшин.

Заходите на сайт **DoTextOK.ru**, там есть много чего, чтобы вёрстка была в кайф, а не тягомотиной.