

DoTextOK — хроника обновлений

// 20.02.2016

// двум прямым наклонным чертам и символу после них присваивается атрибут Nobreak (в библиографических списках не допускается, чтобы одна или две наклонные черты были отдельно от сопровождающего их текста)

// обработка сносок:

// - уточнена интерпретация состояния флажка «Когда рядом запятая или точка и знак сноски —»

// - в файл WordsAndUnits.jsx добавлен массив UnitsAndFootnotes -- это перечень сокращений, заканчивающихся точкой.

// Вот в чём суть изменения: если после них идет знак сноски, то для них не будет работать правило,

// согласно которому при оформлении сносок в текстах на русском языке первым идет слово, потом знак сноски и затем уже точка.

// Этот список сокращений берется в расчёт только при активном флажке «Когда рядом запятая или точка и знак сноски —» и выбранной радиокнопке «...знак сноски перед знаком пунктуации».

// Исключена обработка текстов, размещённых на линиях и контурах. При выборе такого текста будет выведено сообщение, а при групповой обработке статей такие тексты будут пропущены.

// В файле WordsAndUnits.jsx в переменной MustBeWithNextDigit убраны пробелы в начале сокращений, т.е. было:

// var MustBeWithNextDigit = «(пп\\.| ст\\.| с\\.| т\\.| п\\.| ч\\.| С\\.| Т\\.| П\\.| Ч\\.| Р\\.)»; стало:

// var MustBeWithNextDigit = «(пп\\.|ст\\.|с\\.|т\\.|п\\.|ч\\.|С\\.|Т\\.|П\\.|Ч\\.|Р\\.)»;

// Это сделано вот почему. Не обязательно перед этим сокращением должен быть пробел, или он должен начинаться с новой строки; перед ним может быть и скобка. Теперь это учтено в программе.

//

// Обнаружилась такая ошибка: установленный флажок «Только слова и буквы, абзацы не учитывать» не срабатывает, если перед полной полужирной (курсивной) строкой знак перевода строки тоже полужирный (курсивный).

// Поэтому в обрабатываемом тексте сперва всем переводам строки назначается светлое начертание.

//

// В обновлении индизайна в начале 2016 года что-то изменилось во внутренних настройках, и теперь логотип «СМΥК-бабочка» и все пиктограммы смещены в фреймах.

// В этом обновлении опробуется новое решение оформления кнопок.

// Для этого решения в WordsAndUnits.jsx добавлены две переменные, определяющие цвет фона выбранной кнопки и цвет фона, когда курсор выходит за границу кнопки

```

//      defSelColor = [.65,.65,.65, 1];
//      цвет фона кнопки, когда в её пространстве окажется курсор. Первые три числа: RGB, 0 - мин, 1 - макс.
//      defBgColor = [.9,.9,.9, 1];
//      цвет, который приобретёт фон кнопки, когда курсор покинет её пространство. [ до СС 2014 — светлый вариант ин-
//      терфейса программы ]
//      defBgColor = [.29,.29,.29, 1];
//      цвет, который приобретёт фон кнопки, когда курсор покинет её пространство. [ СС 2015 — тёмный вариант интер-
//      фейса программы ]
//      при необходимости их можно изменить.
//      04.03.2016 -- упразднена опция масштабирования шпаций. Теперь на её месте выбор шпации после тире в начале
//      диалога.
// 18.03.2016 -- в ряде случаев пропадал пробел после круглой скобки в нумерованных списках. Исправлено.
// 31.03.2016 -- не обрабатывались взятые в кавычки фамилии и инициалы. Исправлено.
// 25.04.2016 -- исправлена ошибка обработки сносок.
// 27.04.2016 -- исправлена одна редко возникавшая ошибка при обработке запросов на последней вкладке.
// 12.05.2016 -- добавляется пробел между кавычкой и открывающей скобкой / между закрывающей скобкой и кавычкой
// 12.05. 2016 -- вернулось масштабирование пробелов вокруг тире. Теперь масштабируются не любые шпации, а только пробелы,
//      и масштаб указывается не ползунком, а в переменной mySpaceScale, она в файле WordsAndUnits.jsx
// 15.05.2016 -- в скрипте grep-оператор \s заменён на \h. Этим исключена возможность случайной обработки перевода строки.
// 16.05.2016 -- если флажок «В составных сокращениях (т.д., т.е., т.к., ...):» сброшен, то эти сокращения будут без шпаций
// 17.05.2016 -- если флажок «Шпации в инициалах» сброшен, то между инициалами шпации не будет.
//      Но следующие два флажка должны быть установлены, если надо определить, какая шпация вместо пробела должна
//      быть между инициалами и фамилией.
//      Если они будут сброшены, то между блоком инициалов и фамилией будет обычный пробел. Это чревато тем, что
//      фамилия и инициалы могут оказаться в разных строках.
// 17.05.2016 -- в процедуре масштабирования пробелов вокруг тире добавлено присвоение первому из пробелов атрибута
//      Nobreak
// 24.05.2016 -- учтён случай, когда сноска идёт сразу после фамилии.
// 25.05.2016 -- символный стиль myHorizontalScaleForEmDashSpace больше автоматически не удаляется. Если он появился, зна-
//      чит, имеет место масштабирование пробела.

```

// Его можно стереть специально, но масштабирование пробелов останется без изменения.

// 25.05.2016 -- единообразие оформления сокращений д.ф.-м.н., к.т.н., д.т.н., м.н.с. и др. подобных случаев.

// 07.06.2016 -- исправлена ошибка в файле WordsAndUnits.jsx, в строке myWordsBeforeDigit. Было: [Аа]бзац\.{0,2}|[[Пп]риложени\.{1,3} стало [Аа]бзац[ывуаемых]{0,3}|[[Пп]риложени[иейям]{1,3}

// 11.06.2016 -- файл WordsAndUnits.jsx, строка 34, иной вариант записи переменной myUnits, запрещающий оставлять число в конце строки

// 12.06.2016 -- добавлена проверка, какой вид экрана, темный или светлый. Соответственно, в WordsAndUnits.jsx закомментирована информация о цвете кнопок

// 14.06.2016 -- необычное развитие обновления от 15.05.2016: оказалось, что в версии CS5.5 этот оператор не работает. Теперь в скрипте есть проверка версии индизайна и установка \s или \h в grep-запросах.

// 18.06.2016 -- доработан интерфейс работы с запросами на последней вкладке

// 11.07.2016 -- на первой вкладке новый флажок: Перед вёрсткой установить вертикальный и горизонтальный масштаб символов равным 100%

// 11.07.2016 -- при выборе обычного пробела в качестве разделителя сокращений, или между запятой и тире этому пробелу присваивается атрибут «Без переносов» (No break), чтобы исключить случайный разрыв сокращения в конце строки

// 14.07.2016 -- при добавлении очень большого числа знаков в переменную myGrecDeuSpecChars размеры окна программы могут стать шире экрана. Чтобы этого не было, информация о содержимом этой переменной помещается в справку к флажку «Создать символьные стили для специальных знаков и букв других алфавитов»

// 14.07.2016 -- переменную myGrecDeuSpecChars впору переименовывать: теперь добавлены все буквы и огласовки из иврита. Поэтому все, у кого в вёрстке два, три языка, один из которых иврит, поделитесь впечатлениями, насколько это удобно -- иметь отдельный стиль для целого языка.

// 20.07.2016 -- Изменена логика выбора пробельных элементов до и после тире, для этого снова появился на третьей вкладке ползунок выбора ширины пробела.

// Если вы продолжаете работу над публикацией, то в выборе шпаций до и после тире можете оставить выбранные ранее шпации, при этом и флажок «Заменить шпации на обычные пробелы» должен быть сброшен,

// т.к. это тот же режим работы, что в предыдущей версии: шпация-тире-шпация-дискреционный разрыв строки.

// Если этот флажок установлен, то шпации в вёрстке будут заменены на пробелы той ширины, что определено ползунком.

// Первый пробел будет с атрибутом «Без переносов» (No break). И поскольку пробел не является неразрывным символом, то и знак дискреционного разрыва строки после него не нужен.

// Изменён алгоритм выбора шпаций до и после тире -- коль скоро в большинстве случаев они одинаковые, то теперь при выборе одной, вторая изменится автоматически.

// Если такой режим не устраивает, измените в файле WordsAndUnits.jsx значение переменной sameSpaces с true на false.

// 21.07.2016 -- Добавлен флажок «Установить пользовательские значения межсловных и межбуквенных интервалов и горизонтального масштаба глифов», в выпадающем меню подробная справка.

// 25.07.2016 -- Добавлен флажок «После обработки заменить изменяемые неразрывные пробелы на обычный пробел с атрибутом ‘Без переносов’ (Nobreak)», в выпадающем меню подробная справка.

// 02.09.2016 -- Доработан функционал флажка «Установить пользовательские значения межсловных и межбуквенных интервалов и горизонтального масштаба глифов».

// 04.09.2016 -- В переменной NonbreakingSpaceTwoLetterSpace файла WordsAndUnits.jsx исключена частица ‘ст.’ Уже не вспомнить, как она там появилась, но то, что редкое сокращение ст. стало прибаваться к предыдущим словам, это непорядок. Исправлено.

// 16.09.2016 -- Изменена формулировка названия флажка «Удалить в тексте символы переносов», теперь это «Удалить в тексте символы деления на слоги».

// 16.09.2016 -- Внесены исправления в обработку флажка «Одна строка - один абзац».

// 21.09.2106 -- Исправлена ошибка обработки дефиса перед словом (исключён случай, когда он заменялся на тире).

// 28.09.2016 -- Усовершенствована работа флажка «Убедитесь, что число сносок в файле такое же, как и в переданном на вёрстку тексте» —

// теперь в случае потери сносок выводится информация не только сколько сносок есть в вёрстке, как это было в предыдущих версиях,

// но и сколько сносок пропало. При этом на вкладке текстового поиска помещается код маркера потерянной сноски, чтобы эти точки потери сносок сразу найти.

// 18.10.2016 -- На второй вкладке добавлены два флажка: «Отдельные стили для подчеркивания» и «Отдельные стили для ПРОПИСНЫХ БУКВ».

// Стандартно они сброшены, при этом программа пропускает подчеркнутые символы и символы, набранные прописными.

// Но если флажок установлен, то создаются дополнительные символьные стили для каждого случая подчёркивания или набора прописными.

// Если подчеркнутый символ светлого начертания, его стиль будет ‘Underline’, если курсивный, то

'myItalicChars+Underline', если полужирный, то 'myBoldChars+Underline', и т.п.

// Если набранный прописными символ светлого начертания, его стиль будет 'Capslock', если курсивный, то 'myItalicChars+Capslock', если полужирный, то 'myBoldChars+Capslock', и т.п.

// Если курсивный, подчёркнутый и набран прописными, то его стиль будет 'myItalicChars+Underline+Capslock', и т.п.

// 19.10.2016 -- Символьные стили курсив, полужирный, и полужирный курсив также кроме цвета теперь имеют указанное в названии начертание,

// т.е. такая привязка к начертаниям, используемым в шрифтах при наборе текста в MS Word -- таймс, калибри, ариал, вердана, и.пр.

// Но надо иметь в виду, что если в вёрстке могут быть шрифты, в которых нет таких вариантов начертаний, например Franklin Gothic.

// В файле WordsAndUnits.jsx добавлена переменная RemoveCustomColors.

// По умолчанию она равна false, и все служебные цвета после завершения работы скрипта сохраняются. Если равна true, они все удаляются.

// Но есть одна ситуация, когда цвета остаются после завершения работы, даже если переменная RemoveCustomColors равна true.

// Это будет в случае работы с формулами (вкладка «Шпации в тексте и формулах», флажок «Признаки, что в тексте формула» установлен.

// В данном случае важно убедиться, что математические выражения оформлены верно, и цвет важно оставить.

// При очередном запуске, при условии, что флажок «Признаки, что в тексте формула» сброшен, то, если RemoveCustomColors = true, все служебные цвета будут заменены на чёрный.

// 26.10.2016 -- Исправлена ошибка прекращения работы программы, если при запуске курсор стоит в сноске.

// 26.10.2016 -- Если курсор стоит в тексте (не в сноске, и не в таблице), то теперь программа после завершения обработки возвращает на экран отображение той страницы, где стоял курсор перед запуском обработки.

// Предыдущие версии в этом случае отображали страницу, где начинается обработанная статья.

// 30.10.2016 -- Пробел и дефис в конце строки заменяются на шпацию и тире.

// В файле WordsAndUnits.jsx переменная RemoveCustomColors поднята вверх, и теперь не надо её искать.

// Это сделано потому что в прошлом обновлении она осалась включённой, и все пользователи, кто контролировал авторскую разметку появлением красного цвета в вёрстке,

// были в непонимании, почему цвет исчез. Поэтому сейчас штатное значение этой опции false -- цвет остаётся, но чтобы изменить эту установку,

// не надо листать экраны WordsAndUnits.jsx, -- эта переменная доступна сразу после открытия файла установок.

// На первой вкладке добавлен флажок «Удалить символ табуляции в начале абзаца». Дело в том, что всё ещё есть наборщики,

// использующие клавишу табуляции для обозначения нового абзаца. И в предыдущих вариантах скрипта возникала коллизия, как быть, если эти табуляции в начале абзаца надо убрать,

// но табуляции внутри абзацев сохранить. Введение этого флажка снимает эту проблему.

// 01.11.2016 -- Устранена ошибка, проявлявшаяся в том, что индизайн сваливался, если запускалась проверка текста в фрейме на рабочем столе.

// Раньше этой ошибки не было, но 26.10.2016 работа программы была изменена так, чтобы она возвращалась после проверки на страницу, где был курсор в момент запуска скрипта.

// Но если фрейм на рабочем столе, то страница не определена (=null), и программа прекращала работу при попытке перейти на несуществующую страницу.

// 04.11.2016 -- При обработке параметра RemoveCustomColors алгоритм был такой -- текст, оформленный символьными стилями, сперва окрашивался цветом, а после удаления цветом становился чёрным.

// Но в рекламе тоже может быть полужирный или курсив, и если она был оформлен вывороткой, то белый цвет изменится на чёрный. В релизе 04.11.2016 эта ошибка исправлена:

// если RemoveCustomColors = true, символьным стилям цвет не назначается.

// // Выбор шпации после тире «Шпация после тире в начале диалога» предполагал, что в тексте первая буква прописная.

// Для диалогов это верно, но есть и тексты, где предложение начинается с тире, но слово начинается со строчной буквы.

// Это своего рода список. Чтобы учесть и эту ситуацию, обработка этой опции переделана, и название опции изменено.

// Теперь это «Шпация после тире в начале диалога или перед строкой списка».

// 08.11.2016 -- Мы верстаем разные книги, кто-то учебники, кто-то художественную литературу. И интересная выяснилась коллизия в обработке ситуации «дефис после перевода строки».

// До этого обновления этот дефис заменялся на тире, и это было нормально для художественных книг, т.к. там это обычно были диалоги.

// Но в технической литературе это обычно списки, и тире, если оно длинное, смотрится некрасиво.

// Чтобы разрешить эту ситуацию, на первой вкладке добавлен выбор «Дефис в начале абзаца», пусть пользователь

определит,
// как обрабатывать дефис в начале строки: не менять, заменить на минус или на тире.
// 20.11.2016 -- Учтена просьба расширить функционал скрипта новой опцией, на мой взгляд, нужной далеко не всем. И тем не менее, есть коллеги, для кого она нужна.
// Вот в чём заключается эта опция: набранное в верхнем регистре число затекстового комментария можно теперь отбивать от текста волосистой шпацией, как это делается для обычных сносок.
// Для этого надо переменную 'spaceForSupChar' (файл WordsAndUnits.jsx) установить в значение true.
// Эта установка действует только на пару «слово»&«число_в_верхнем_регистре», между числами «число»&«степень_числа» шпация помещаться не будет.
// 09.12.2016 -- Список сокращений перед числом, переменная MustBeWithNextDigit в файле WordsAndUnits.jsx, пополнилась сокращениями из Библии: Быт. Исх., Лев. и пр. Список сокращения взят тут: <https://azbyka.ru/biblia/?pJer.1>
// 08.04.2017 -- в скриптах раскомментированы строки включения пространства выполнения скрипта. Это оператор targetengine. Обновление InDesign от апреля 2017 изменило концепцию работы скриптов -- теперь если скрипт создаёт меню, он должен работать в своем пространстве.
// 07.10.2017 -- найдено лекарство для нейтрализации очередного усовершенствования среды исполнения скриптов. С какого-то времени на макинтошах пропал прогресс-бар. Именно только на маках, машин под Windows это не коснулось. Теперь прогресс-бар снова должен быть на обеих платформах.
// 08.10.2017 -- Сообщение о завершении работы скрипта держится на экране в течение 2 секунд, потом исчезает.
// 24.10.2017 -- Исправлена ошибка в работе опции «Дефис в начале абзаца».
// 20.07.2018 -- Добавлена обработка концевых сносок. С появлением двух вариантов сносок теперь в сообщениях они или 'внутритекстовые' или 'концевые'.
// Когда курсор в основном тексте, обрабатываются оба вида сносок.
// Если курсор в фрейме концевых сносок, то будут обработаны только они.
// И обработка скриптом концевых сносок выполняется только при условии установки флажка «Обрабатывать концевые сноски».
// Массив служебных цветов перенесён из скрипта в файл установок WordsAndUnits.jsx. Назначение этого массива -- в описании флажка 'RemoveCustomColors'.
// В файл установок WordsAndUnits.jsx добавлена переменная addSpaceAfterDot = false;
// стандартно false, чтобы не ломать имена сайтов. Пробел после точки в 'adobeindesign. ru' сделает эту ссылку битой. Но если в тексте нет адресов сайтов, а есть предложения между которыми нет пробела, то true решит эту проблему.

// 21.07.2018 -- В файл установок WordsAndUnits.jsx добавлена переменная spaceAfterBracketInList = true;
// В списках с номерами или буквами после номера идёт скобка. Потом пробел. И когда такой список оформлен в виде строки, возможна ситуация, что номер и скобка остались в конце строки, а текст ушёл в начало следующей строки.
// Значение true -- скрипт заменит пробел после скобки на неразрывный пробел.

// 23.07.2018 -- На второй вкладке добавлен флажок «Переносы ‘-НЕ’ и ‘НЕ-’».
// В практике книгоиздания известны два редких случая упущений в вёрстке, из-за которых при быстром чтении можно понять текст неправильно.
// Это может случиться, если: 1) частица ‘не’ оказалась в конце строки; 2) буквы ‘не’ — первая или последняя часть разделённого переносом слова.
// Первая ошибка вёрстки снимается запретом отрывать частицу НЕ от слова, см. переменную ‘SpaceTwoLetterNonbreakingSpace’ в файле WordsAndUnits.jsx.
// Установка данного флажка исключит вторую ошибку. Это правило для вёрстки книг, не для узкой колонки газеты или журнала.»

// 24.07.2018 -- Если есть потерянные сноски, то после завершения работы скрипта о них будет выведено отдельное сообщение
// «Скрипт подготовил текст к вёрстке, но к ней приступить нельзя, так как в работе есть потерянные сноски. Их число ...»

// 26.07.2108 -- Добавлен флажок «Оформление сноски определяется установками абзацного стиля».
// Его назначение: вид сноски определяется символьным стилем знака сноски, абзацным стилем текста сноски и шпацией, которая должна быть между знаком сноски и текстом сноски.
// Эти установки должны быть сделаны на вкладке «Нумерация и форматирование» в панели «Параметры сносок» (Текст > Параметры сносок документа...).

// Но чтобы это всё сразу правильно применилось, надо, во-первых, не забыть сделать эти установки, во-вторых, в тексте сносок шпации после знака сноски должны совпадать с указанными.
// Если этого не сделать, придётся какие-то из сносок приводить в порядок. Это потери времени.
// Но если один раз сделать шаблон оформления сносок, в котором в установках абзацного стиля на вкладке «Буквицы и вложенные стили» указать символьный стиль знака сноски и шпацию,
// то можно заметно ускорить время работы, если в вёрстке много сносок. От нас требуется только указать на вкладке «Нумерация и форматирование» название абзацного стиля сноски.
// А при запуске скрипта эта опция «Оформление сноски определяется установками абзацного стиля» сделает за вас четыре дела:

// 1) извлечёт из описания стиля символьный стиль и шпацию и поместит на вкладку «Нумерация и форматирование» в панели «Параметры сносок».

// 2) Поставит между знаком сноски и текстом ту шпацию, что определена в установках вкладки «Буквицы и вложенные стили».

// 3) назначит тексту сноски абзацный стиль без потери локального форматирования, которое дальше в процессе обработки будет отмечено символьными стилями.

// 4) Если вы в установках абзацного стиля определите следующий стиль для второго и последующих абзацев, то он будет приложен к этим абзацам.

// Фактически после завершения обработки текста сноски уже получают окончательное стилевое оформление.

// Опция включается при обработке отдельных статей.

// Установка для использования в варианте «Use Current Set» не сохраняется, и варианта для групповой обработки не предусмотрено.

// Объяснение этой опции в файле есть в папке Info.

// 03.08.2018 -- Исправлена ошибка замены пробела между цифрами d ddd на тонкую шпацию.

// 08.08.2018 -- В предыдущих версиях после завершения работы выводилось окно. И у многих пользователей сложилась практика гасить это окно нажатием на клавишу Escape.

// Начиная с 08.10.2017 это окно стало появляться только две секунды. Но если по привычке щелкнуть на Escape, оно оставалось до перезапуска индизайна.

// Сейчас в этом окне возвращён красный крестик закрытия окна, так что его можно убрать, не перезагружая программу.

// Но лучше отказаться от практики нажатия на Escape -- это окно исчезнет само.

// Время закрытия в миллисекундах устанавливается в переменной sleepTime в файле WordsAndUnits.jsx

// 19.08.2018 -- Исправлена ошибка в обработке флажка «Оформление сноски определяется установками абзацного стиля» -- теперь он устанавливается только в случае, если в параметрах абзацного стиля всё настроено правильно.

// 21.08.2108 -- Исправлена ошибка обработки знака табуляции. Это в той же процедуре с флажком «Оформление сноски определяется установками абзацного стиля».

// 22.08.2018 -- Доработана обработка переменной spaceAfterBracketInList = true: она активна, если перед скобкой одна буква или одна-две цифры.

// 22.08.2018 -- Изменена процедура создания симв.стиля myFootnotes. До этого если его не было, он создавался, и потом, в любом случае, был он или не был, в его свойствах появлялся Position.Superscript,

// т.е. верхний индекс. Как оказалось, это противоречит практике использования значения Позиция : Надстрочные знаки OpenType, т.е. без включения верхнего индекса.

// Поэтому, если стиль myFootnotes уже есть, то в нём ничего не меняется. Используется таким, как есть. Если нет, то в нём будет верхний индекс, он есть во всех шрифтах.

// А вариант 'Надстрочные знаки OpenType' есть в Minion Pro, Myriad Pro и других современных OTF шрифтах, но в Times New Roman его нет.

// 04.09.2018 -- Учтено, что перед сокращением т.п., т.д. и другими такими может стоять открывающая квадратная скобка. До этого бралась в расчёт только круглая.

// 04.09.2018 -- Многоточие с атрибутом «Без переносов» (Nobreak).

// 04.03.2019 -- Добавлена проверка, нет ли сносок, в которых больше одного абзаца. И если такие сноски будут найдены, будет сообщение, что для правильной работы со сносками надо сделать для второго и последующих абзацев отдельный абзацный стиль, и указать этот стиль в настройках абзацного стиля для первого абзаца.

// Эта проверка объединена с проверкой потерянных сносок, и теперь название флажка «Убедитесь, что число сносок в файле такое же, как и в переданном на вёрстку тексте»

// изменено на «Получить информацию о сносках в переданном на вёрстку тексте».

// 04.03.2019 -- При использовании словаря Hunspell часты такие переносы: кака-я-то, каки-е-то, каку-ю-то и пр. В скрипте последовательности знаков как\\| присваивается атрибут без переносов, и этой ошибки больше нет.

// 04.03.2019 -- Исправлена ошибка обработки концевых сносок, когда активна опция «Оформление сноски определяется установками абзацного стиля».

// 04.03.2019 -- Если активна опция «Оформление сноски определяется установками абзацного стиля», то в сносках знаки табуляции заменяются на пробелы независимо от того, выбрана или нет на первой вкладке установка «Заменить символы табуляции на пробелы».

// 04.03.2019 -- Подготовлен idml-файл с пятью вариантами стилей на разные варианты оформления сносок для использования их с опцией «Оформление сноски определяется установками абзацного стиля»,

// // 12.03.2019 -- Долгое время шпация между цифрами года и сокращением 'г.' или 'гг.' была неразрывным фиксированным пробелом. После нескольких просьб изменить теперь это неразрывный изменяемый пробел.

// 12.03.2019 -- В файл WordsAndUnits.jsx добавлена переменная flexNobreakOnly. Если она true, то в обрабатываемом тексте все фиксированные неразрывные пробелы будут заменены на изменяемые неразрывные пробелы.

// 12.03.2019 -- Исправлены другие глюки переносов Hunspell: во-йна, во-йсковой, ко-е-что, ко-е-как, ко-е-где, ко-е-кому и пр., . И заодно вместо ру-бля теперь будет руб-ля.

// 12.03.2019 -- В каких-то случаях между числами, набранными римскими цифрами, было тире, окружённое шпациями. Сейчас всегда тот же разделитель, что и для чисел, набранных арабскими цифрами.

// 12.03.2019 -- В комплекте программы появился скрипт DoFootnotesOK -- это для оформления сносок установками абзацного стиля, но в уже свёрстанном тексте.

// Ну может быть не до конца свёрстанном, но в таком, что DoTextOK запускать уже вредно, т.к. было много чего в плане вёрстки сделано, что DoTextOK, если запустить, то почистит.

// А вот возможность поправить только сноски -- это очень полезная опция, и она реализуется модулем FootnoteControl.jsxbin, он в папке BIN.

// Это платная опция, она продаётся вместе с появившееся 26.07.2018 дополнением «Оформление сноски определяется установками абзацного стиля» на shop.dotextok.ru.

// 14.03.2019 -- В файл WordsAndUnits.jsx добавлена переменная theDoubleNames, если она true, это значение по умолчанию, то составные собственные слова, Кирилло-Белозерский, Соколов-Микитов будут если будут переноситься, то только по дефису, разделяющему эти слова.

// 14.03.2019 -- В файл WordsAndUnits.jsx добавлена переменная splitOnHyphen. Если она в значении true, то слова с дефисами 'военно-патриотический', 'жёлто-оранжевый' и пр. если будут переноситься, то только по дефису, разделяющему эти слова.

// Минимальное число букв в каждом из пары слов определяется переменными word1N и word2N, по умолчанию word1N = 5 и word2N = 5.

// 14.03.2019 -- В файл WordsAndUnits.jsx добавлен массив слов splitExceptions = [«либо», «нибудь», «таки»]. Это вторые части слов с дефисами как указатели на двойные слова, которые не должны переноситься.

// 'кто-либо', 'когда-нибудь' всегда останутся на одной строке, т.к. перед этим двойным словом программа поставит дискреционный перенос.

// Частицу 'то' включать в этот массив не стоит, т.к. могут быть длинные слова, после которых идёт эта частица: «Председатель-то должен был подумать об этом»,

// и запрет переноса слова может стать причиной появления неплотной строки.

// 19.03.2019 -- В папке '+SpisokOfLiteraturi-AutorsAreRed' теперь другие grep-запросы.

Они называются 'nLIC=1.xml' и 'nLIC=2.xml'. Причин изменения две:

// - исправление grep-запроса обработки 'Фамилия И. О.', чтобы обрабатывались только ФИО после порядкового номера, а не все случаи ФИО в тексте

// - учёт того, что сейчас стало частым ставить запятую после фамилии, перед инициалами: 'Фамилия, И. О.'

```

// Перечисленные действия обеспечивает первый grep-запрос, nLIC=1.xml
// Исследовались такие варианты запросов для решения этой задачи:
// (?<=^d\d\.s)((\u|l+|-)*(\u|l+)+,*+s(\u|.?.-)*\u|l?\.(\s?(\u|l?\.?)*)*(, )?) + -- находит все строки списка литературы с номе-
рами в начале строки от 1. до 9, с 10 и дальше -- уже не видит.
// (?<=^d\d+\.s)((\u|l+|-)*(\u|l+)+,*+s(\u|.?.-)*\u|l?\.(\s?(\u|l?\.?)*)*(, )?) + -- не работает вообще, т.к. в используемой в
InDesign версии grep в просмотрах назад оператор + не работает.
// (^d\d\.s)\K((\u|l+|-)*(\u|l+)+,*+s(\u|.?.-)*\u|l?\.(\s?(\u|l?\.?)*)*(, )?) + -- использование \K вместо просмотра назад: ищет
через строку. Это, по-видимому, из-за того, что используется оператор ^
// (\d\d\.s)\K((\u|l+|-)*(\u|l+)+,*+s(\u|.?.-)*\u|l?\.(\s?(\u|l?\.?)*)*(, )?) + -- без оператора ^ эта запись с оператором \K ищет
фамилии перед номером и точкой в каждой строке.
// Но если отказаться от учёта начала абзаца, т.е. не использовать оператор ^ в строке, то можно сохранить положи-
тельный просмотр назад, и отказаться от команды \K, не работающей в версиях индизайна до CS6.
// Поэтому сейчас используется такой оператор:
(?<=^d\d\.s)((\u|l+|-)*(\u|l+)+,*+s(\u|.?.-)*\u|l?\.(\s?(\u|l?\.?)*)*(, )?) +
// Второй grep-запрос снимает курсив с запятой после ФИО и делает её чёрного цвета.
// Запускающая строка +SpisokOfLiteraturi-AutorsAreRed.quedtok теперь в той же папке, где все скрипты.
// Пользователь ДОЛЖЕН САМ ПОМЕСТИТЬ эти два новых запроса в пользовательскую папку с запросами исполь-
зуемой версии индизайна.
// DoTextOK не переносит эти файлы в ту папку, как это делает DoQueryList.
// 27.03.2019 -- Рекомендуются установки межсловных пробелов, межбуквенных интервалов и глифов, как оказалось,
работали только для крайних значений. Среднее не изменялось.
Исправлено.
// 27.03.2019 -- Исправлена ошибка в варианте скрипта для обработки всех статей документа.
// 16.04.2019 -- Из скриптов различных вариантов запуска убран код, выполняющий обработку текста.
Он во всех трёх программах один и тот же, и теперь он в файле DoTextOK.jsxinc.
// 20.04.2019 -- Скрипт DoFootnotesOK.jsx, входящий в комплект DoTextOK, доработан, чтобы приводить в порядок сноски в фай-
лах, в которых символьные и абзацные стили размещены в группах. Первые варианты этого скрипта работали со
стилями в предположении, что они не принадлежат никакой группе.

```