

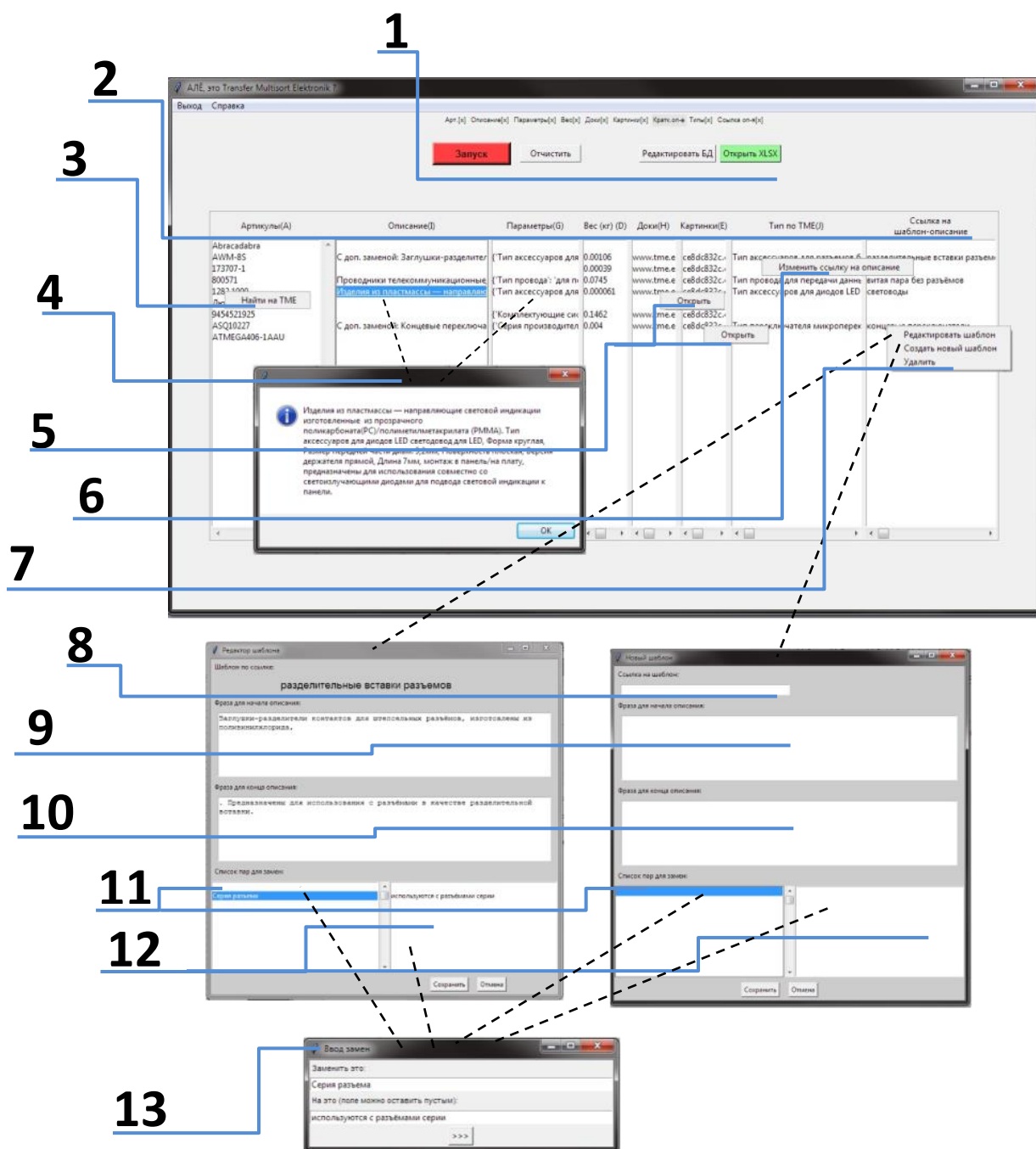
АЛЁ, это Transfer Multisort Elektronik ?

Программа для получения информации по артикулам товаров с сайта TME (Transfer Multisort Elektronik - <https://www.tme.eu/ru/>), а также по составлению краткого описания в контексте товарной номенклатуры внешнеэкономической деятельности Евразийского экономического союза (ТН ВЭД), для использования этого описания при заполнении деклараций таможенного оформления.

Для получения информации используются команды программного интерфейса (API) сайта TME.

Описания составляются на основе базы текстовых шаблонов составляемых пользователем.

Интерфейс



1. **Кнопки.** В самом верху окна находятся 9 кнопок, отвечающих за скрытие/отображение колонок. При скрытии колонки, кнопка подсвечивается темным цветом. При включении отображения, колонка отобразится справа.

!!! Программа запоминает какие колонки были скрыты, но не в состоянии запомнить новый порядок колонок, поэтому при следующем включении отобразит в порядке, установленном по умолчанию.

Кнопка **ЗАПУСК**, включает процесс обработки артикулов из колонки *Артикулы(A)*.

ПРОЦЕСС ОБРАБОТКИ заключается в запросе данных с сайта ТМЕ с последующим размещением в файле **productdata.xlsx**, находящемся в директории с файлами программы.

После сбора данных программа отделяет от основных параметров продукта (колонка *Параметры(G)*) основную характеристику-классификатор, называемую **«ТИП ПРОДУКТА»**. Классификаторы используют для привязки *шаблона-описания*, с помощью которого происходит сборка описания (колонка *Описание(I)*).

Классификаторы **«ТИП ПРОДУКТА»** автоматически собираются в коллекцию в файле базы данных **database.json**, находящемся в директории с файлами программы. Пользователь привязывает к классификатору **ССЫЛКУ НА ШАБЛОН-ОПИСАНИЕ** (колонка *Ссылка на шаблон-описание*). Далее происходит составление описания для всех привязанных к шаблону-описанию позиций.

ШАБЛОНЫ-ОПИСАНИЯ состоят из **ФРАЗЫ ДЛЯ НАЧАЛА ОПИСАНИЯ**, **ФРАЗЫ ДЛЯ КОНЦА ОПИСАНИЯ** и списка **вторичных текстовых замен**. Шаблоны-описания хранятся в том же файле базы данных **database.json**. Основной алгоритм составления описания заключается в помещении *параметров* продукта между **ФРАЗой ДЛЯ НАЧАЛА ОПИСАНИЯ** и **ФРАЗой ДЛЯ КОНЦА ОПИСАНИЯ**, затем производятся **первичные замены текста** (*первичные замены производятся для всех описаний что бы избежать ненужных символов*, список первичных замен хранится в файле **config.py**, в формате `replace_list = [{"заменяемое 1", "заменитель 1"}, {"заменяемое 2", "заменитель 2"}, ..., {"заменяемое n", "заменить n"}]`), затем производятся **вторичные текстовые замены** прописанные в **шаблоне-описании** (вторичные текстовые замены устанавливаются в редакторе *шаблонов-описаний*, они индивидуальны для каждого *шаблона-описания*).

Полученные данные и собранные описания записываются в файл **productdata.xlsx**, а затем отображаются в колонках графического интерфейса.

!!! После редактирования списка первичных замен (в файле **config.py**) требуется перезапустить программу так, что бы все переменные обновились.

Кнопка **ОТЧИСТИТЬ** полностью стирает данные из файла **productdata.xlsx**.

Кнопка **Редактировать БД** открывает файл **database.json**, кнопка **Открыть XLSX** открывает файл **productdata.xlsx**. *Файлы database.json и productdata.xlsx нельзя открыть пока идет процесс обработки.*

!!! Процесс обработки нельзя запускать если открыт файл **database.json** или **productdata.xlsx**.

2. **Колонки.** Колонки содержат информацию из столбцов файла **productdata.xlsx**. В названии колонки, в скобках, указывается буква идентичного столбца в экселе. В колонке **«Ссылка на шаблон-описание»** отображаются названия *шаблонов-описаний*, которые привязаны к классификатору продуктов **«ТИП ПРОДУКТА»**. Классификаторы продуктов помещены в

колонку «Тип по ТМЕ(J)».

Основная, рабочая колонка «**Артикулы(A)**» используется для помещения в неё наименований артикулов. Артикулы вставляются из буфера обмена, для вставки более одной позиции данные копируются из столбца стороннего эксель-файла или из столбца табличных данных подобного формата.

!!! Для вставки из буфера обмена используется комбинация Ctrl+v, следует учитывать, что все клавишные комбинации работают только в латинской раскладке клавиатуры и не работают при русской.

!!! При вырезании(Ctrl+x), вставке(Ctrl+v) и удалении(Del) данных из колонки «Артикулы(A)» все полученные данные удалятся и потребуется снова запускать цикл программы.

!!! Для более достоверного поиска допускается вставлять наименования артикулов с названием фирмы-производителя через пробел.

!!! Для одновременной прокрутки колонок используется ползунок справа от колонки «Артикулы(A)». При прокрутке с помощью колеса мыши данные в колонке прокручиваются независимо от других колонок, это может привести к путанице. Для выравнивания данных всех колонок нужно подвигать ползунок.

!!! Все дополнительные меню колонок доступны при выделении пункта-строки в колонке левой кнопкой мыши.

3. **Найти на ТМЕ.** Функция открывает поиск на сайте ТМЕ с данной позицией. Меню вызывается правой кнопкой мыши.
 4. **Информационные панели.** В колонках «Описание(I)», «Параметры(G)» и «Краткое описание(C)» по двойному щелчку левой кнопки мыши вызываются информационные панели для удобства чтения данных.
 5. **Открыть.** В колонках «Доки(H)» и «Картинки(E)» содержатся ссылки на документацию и картинки. Для быстрого открытия в браузере используйте меню *открыть*. Меню вызывается правой кнопкой мыши.
 6. **Изменить ссылку на описание.** В колонке «Тип по ТМЕ(J)» содержатся выделенные для классификации заголовки «**ТИП ПРОДУКТА**», к которым привязывается ссылка на шаблон-описание. Меню для открытия списка ссылок вызывается правой кнопкой мыши. Ссылки, заканчивающиеся на *_func* указывают на функцию включения альтернативного сценария сборки описаний, это надо учитывать при создании нового шаблона-описания, не допуская в его названии сочетания символов *_func*.
 7. **Меню редактирования шаблона.** В колонке «Ссылка на шаблон-описание» содержатся названия привязанных шаблонов. Меню редактирования шаблона вызывается правой кнопкой мыши.
 8. **Редактор шаблона-описания: ссылка на шаблон.** Название шаблона используется для привязки его к типу продукта.
!!! После создания нового шаблона-описания его будет невозможно переименовать в редакторе, только вручную, в базе данных.
 9. **Поле ввода для начала описания.** Поле для ввода фразы для начала описания.
 10. **Поле ввода для конца описания.** Поле для ввода фразы для конца описания.
 11. **Поле для создания текстовой замены.** Внизу редактора располагаются окошки, в которых содержатся пары текстовых замен. Для создания новой замены нужно кликнуть двойным щелчком левой кнопкой мыши по верхнему пункту-строке в любом из окон. Откроется окно ввода текстовых замен.
 12. **Поле редактирования текстовых замен.** Для редактирования кликнуть двойным щелчком левой кнопкой мыши по пункту-строке (в любом из окошек), которую нужно отредактировать. Откроется окно ввода текстовых замен. Для удаления ненужной пары замен выделить пункт-строку и нажать DEL.
-

13. Окно ввода текстовых замен.

!!! Все действия программы, приводящие к изменению базы данных автоматически создают её резервную копию с именем **database.json_backup_copy.json**

Структура БД database.json

БД **database.json** содержит два массива данных типа *список из двух словарей*, с такой структурой:

```
[  
1      { Тип продукта : ссылка-название шаблона },  
2  { ссылка-название шаблона : [начало, конец описания, [[это заменить, заменить на это],...[это заменить, заменить на  
      это]]] }  
]
```

{первый словарь} - это множество типов-товаров полученное отделением параметра "тип" от списка параметров (столбец экселя G) , к которым пользователь привязывает ссылку-название шаблона через двоеточие, по которым получаем шаблон-описание из второго словаря.

{второй словарь} - по ссылке-названию отдает для сборки шаблон-описание в виде списка [0-Начало описания, 1-конец описания, [[заменяемое,заменитель],[..., ...]...[..., ...]]].

Описание составляется конкатенацией: нулевой элемент словаря + параметры из экселя + первый элемент словаря, при этом происходят общие замены (указанные в файле config.py), затем происходят частные замены, которые указываются в шаблоне.

Структура стандартного шаблона во-втором словаре:

ссылка-название шаблона : [0-Начало описания, 1-конец описания, [[заменяемое 1,заменитель 1],[заменяемое 2, заменитель 2]...[
заменяемое n, заменитель n]]],

Или в случае, если замены делать не надо:

ссылка-название шаблона : [0-Начало описания, 1-конец описания],

Если при обращении по ссылке-названию шаблона, в качестве *фразы начала описания* прописан текст: "MY_FUNCTION", то включается *альтернативный сценарий*, запускающий функцию название которой указывается на месте *фразы для конца описания*, в этом же шаблоне. Для таких сценариев название ссылки-шаблона описания пишется латиницей, с обязательным окончанием названия *_func*.

Использование альтернативного сценария textTemplate, для вставки параметров в готовый текстовый шаблон, ручное редактирование БД.

В программе предусмотрен альтернативный сценарий для вставки значений параметров в готовый шаблон. Для создания таких шаблонов не предусмотрен редактор, но их можно создавать и редактировать вручную. Для этого в файле *database.json*, во-втором массиве с шаблонами описаний нужно описать новый шаблон таким образом:

Название нового шаблона пишется латиницей, а в конце приписывается *_func*, например:

```
"mysupertext_func": [
```

Вместо *фразы для начала описания* пишется "MY_FUNCTION". Это указывает программе на запуск альтернативного сценария.

Вместо *фразы для конца описания* пишется название включаемой функции, это:

```
"textTemplate(i,link_key)"
```

Таким образом должно получиться такое начало:

```
"diode_func": ["MY_FUNCTION", "textTemplate(i,link_key)",
```

Далее через запятую вместо *замен* пишется такая структура:

```
"Текст {0[текст который будет вставляться1]} шаблона. И вот ты дочитываешь этот {0[текст который будет вставляться2]} текст.",
```

```
{ "текст который будет вставляться1 ":"какого-то там",
```

```
"текст который будет вставляться2 ":"выразительный" },
```

```
[[заменяемое 1,заменитель 1],[ заменяемое 2, заменитель 2]...[ заменяемое n, заменитель n]]],
```

Если все сделано верно, программа увидит шаблон *mysupertext_func*. Если не учитывать замены программа соберет вот такой текст:

Текст *какого-то там* шаблона. И вот ты дочитываешь этот *выразительный* текст.

В данном примере в структуре { "текст который будет вставляться1 ":"какого-то там", "текст который будет вставляться2 ":"выразительный" }, указаны *теги-описания* «текст который будет вставляться1» и «текст который будет вставляться2» и *параметры* на которые ссылают эти теги: *«какого-то там»* и *«выразительный»*. Подобную структуру, только с другими параметрами ожидается получить от сайта ТМЕ (см. столбец экселя G).

В текстовом шаблоне, в места, где задумывается вставить *параметры* устанавливается вставка в фигурных скобках вида: {0[текст который будет вставляться1]}, в которой указывается *тег-описание* параметра, которое ожидается получить от сайта ТМЕ (столбец экселя G), ноль указывается всегда, из-за специфики синтаксиса. Если *тег-описание* будет найден в структуре ответа от сайта ТМЕ, то *параметр* на которые он указывает будет вставлен в шаблон. В случае если на сайте ТМЕ не будет подходящих тегов-описаний, то будут взяты *параметры* из структуры, описанной в шаблоне, для нашего примера это *«какого-то там»* и *«выразительный»*.

Пример рабочего шаблона для разъемов "connectors_func":

```
"connectors_func": [  
  "MY_FUNCTION",  
  "textTemplate(i,link_key)",  
  "Штепсельные разъемы типа {0[Разъем]} {0[Тип разъема]}, серии {0[Серия разъема]}, для монтажа {0[Электрический  
монтаж]}, {0[Кол-во выводов]} контактов расположены с шагом {0[Шаг контактов]}, с конфигурацией выводов  
{0[Конфигурация выводов разъема]}. Номинальное напряжение {0[Номинальное напряжение]}, номинальный ток  
{0[Номинальный ток]}, контакты изготовлены из медного сплава с гальваническим покрытием, материал корпуса полиамид,  
рабочие температуры от -40 до 105°C, предназначены для радиоэлектронного оборудования общепромышленного  
назначения.",  
  {  
    "Разъем": "",  
    "Тип разъема": "",  
    "Серия разъема": "...",  
    "Электрический монтаж": "на кабель",  
    "Кол-во выводов": "",  
    "Шаг контактов": "2.54 мм",  
    "Конфигурация выводов разъема": "в один ряд",  
    "Номинальное напряжение": "48 В",  
    "Номинальный ток": "3 А"  
  },  
  [  
    [  
      "плоский",  
      "одноконтakтный плоский разъем"  
    ],  
    [  
      "обжим",  
      "на проводник с помощью обжима"  
    ],  
    [  
      "зажим под винт",  
      "с помощью винтов на корпус или панель"  
    ]  
  ]  
],
```

Текстовый шаблон

Структура тег : параметр

Замены

!!! База данных выполнена в формате данных JSON, при несоответствии символов или ошибки в структуре даст сбой. Если после ручного редактирования базы возникли ошибки, следует проверить текст базы на наличие ошибок, для этого можно использовать онлайн ресурс <https://jsonlint.com/> или подобные ресурсы. В случае невозможности исправления базы воспользоваться бэкап-файлом database.json_backup_copy.json