Транслятор — краткое пояснение

Содержание

Полное описание — в `TRANSLATOR.md`, на английском: там весь синтаксис, договор инструмента и границы. Здесь — что это и зачем.

Зачем он нужен

TourBox Console — единственный способ собрать пресет, и это много кликов ради раскладки, которую вы и так знаете. Пресеты неудобно сравнивать, ревьюить и хранить в системе контроля версий.

Транслятор делает раскладку текстовым файлом:

```
"side+dpadLeft": "ctrl+home"
```

Слева — какой контрол, справа — что он шлёт. Одна команда собирает из этого `.tb`, который импортируется в Console.

Как выглядит манифест

```
{
  "name": "Browser - General",
  "bind": {
    "dial":      { "a": "ctrl+shift+tab", "b": "ctrl+tab" },
    "knob":      { "a": "alt+left",       "b": "alt+right" },
    "knobPress": "ctrl+l",
    "tallx2":    "escape"
  }
}
```

Пара `{a, b}` — два направления вращения. Обычная строка — одиночное нажатие. Имена контролов английские, полный список в `FORMAT.md` §8.

Сочетание — это модификаторы (`ctrl`, `alt`, `shift`) плюс ровно одна клавиша: символ (`a`, `7`, `[`), именованная спецклавиша (`left`, `pagedown`, `f12`, `numpad7`) или управляющая словом (`space`, `enter`, `escape`).

Отдельный случай — **только модификаторы**, без клавиши: `"side": "ctrl"`. Тогда кнопка держит Ctrl, пока нажата. Так делает и сам вендор, и это то, что позволяет тащить клип мышью с зажатым Ctrl+Alt.

Чего инструмент не делает

Это не придирчивость, а суть:

- **Никогда не угадывает слот.** Незнакомое имя контрола — ошибка, а не «похожее». Опечатка, тихо попавшая на соседнюю кнопку, обнаружилась бы через месяц посреди работы.
- **Никогда не выдумывает код клавиши.** В таблице только те коды, которые Console показала под собственным именем.
- **Никогда не пишет файл, который не может прочитать обратно.** Каждая сборка перечитывает свой результат и сверяет таблицу.
- **Не трогает то, о чём не просили.** Привязки, которых нет в манифесте, остаются как в исходном файле.

Не только клавиши

В манифесте доступны ещё две вещи.

Колесо мыши. `wheelup` и `wheeldown` — такие же сочетания, как остальные, и принимают модификаторы:

```
"scroll":      { "a": "wheelup",      "b": "wheeldown" },
"side+scroll": { "a": "ctrl+wheelup", "b": "ctrl+wheeldown" }
```

Вращение с колесом крутит то, на что наведён курсор. Так поворачивается слайдер Lightroom или Lumetri, у которого своего сочетания не существует.

Скорость вращения и тактильная отдача, отдельно на каждое вращение:

```
"dial": { "a": "wheelup", "b": "wheeldown", "speed": "slower", "haptic": "off" }
```

`speed` — `normal`, `slow` или `slower`; `haptic` — `off`, `light`, `normal`. Три вращения с одним действием на трёх скоростях дают грубо и точно без переключения режимов.

Что нельзя выразить

Кнопки мыши и перетаскивание с удержанием, TourMenu, макросы, плагинные команды. Там, где пресету пришлось это обходить, в манифесте стоит поле `_deviation` — раскладка не притворяется, что делает ровно заказанное.

Рабочий цикл

```
правим manifests/40-media-potplayer.json
python tools/tbmake.py manifests/40-media-potplayer.json
        → presets/40-media-potplayer.tb
импортируем в Console как НОВЫЙ пресет
```

Дальше проверка в три шага:

1. **Прочитать список привязок в Console.** Имена должны совпасть с тем, что написано в манифесте. Сразу ловит попадание не в тот слот.

2. **Экспортировать обратно и сравнить.** `RECORDS identical` означает, что ничего не переписано молча. Заголовок отличается в одном байте — это номер слота, присвоенный Console, а не ваши данные.
3. **Поработать час.** Удобство раскладки не проверит ни один инструмент.

Где живут знания

`calibration/calibration.json` — единственное место, где хранится всё разобранное, и у каждой записи стоит статус: `confirmed` — наблюдали лично, `inferred` — вывели, `unresolved` — не знаем. Таблицы внутри инструментов берутся оттуда, а не наоборот.