

Восстановление ложных правок регистра после запятых

Инструмент работает в этом проекте и не требует изменений в DocFlow.

Путь в GUI:

Специализированные команды -> Восстановить регистр после запятых

Входные артефакты:

- JSON-отчёт DocFlow `comma_lowercase.json`;
- исходный DOCX или папка исходных DOCX;
- исправленный DOCX или папка исправленных DOCX;
- JSON решений LLM или ручная карта восстановления.

Что куда класть

Рекомендуемая раскладка внутри `Audion Docs AI`:

Что	Куда положить	GUI-поле / CLI-параметр
JSON-отчёт DocFlow <code>comma_lowercase*.json</code>	<code>report\comma_lowercase.json</code>	JSON-отчёт DocFlow / <code>--report</code>
Исходный DOCX до правки	<code>input\</code>	Исходный DOCX / папка / <code>--source input</code>
Исправленный DOCX после DocFlow	<code>output\comma_lowercase\</code>	Исправленный DOCX / папка / <code>--fixed output\comma_lowercase</code>
Готовые решения LLM <code>restore/keep</code>	<code>report\comma_lowercase_decisions.json</code>	JSON решений LLM / <code>--decisions</code>

Что	Куда положить	GUI-поле / CLI-параметр
Ручная карта устойчивых восстановлений	<code>config\comma_restore_map.yaml</code>	Карта восстановления / <code>--restore-map</code>
Восстановленные DOCX-копии	<code>output\comma_lowercase_restored\</code>	Папка результата / <code>--output</code>

Имена исходного и исправленного DOCX могут отличаться, если JSON-отчёт DocFlow содержит исходное имя файла, а исправленный файл лежит в обычном DocFlow-формате

`<имя>_comma_lowercase.docx`. Если файлов несколько, кладите исходники в `input\`, исправленные копии в `output\comma_lowercase\`, а общий отчёт - в `report\comma_lowercase.json`.

В GUI можно не копировать файлы в managed-папки и указать абсолютные пути, но для повторяемой работы проще держать комплект в этих местах.

Поле **Источник решений** определяет, откуда брать смысловое решение:

- **Готовый JSON / карта** - не вызывать LLM, а взять **JSON решений LLM** или ручную карту;
- **OpenAI** - классифицировать hits через OpenAI;
- **Gemini** - классифицировать hits через Gemini.

Если выбираете **OpenAI** или **Gemini**, заполните тот же провайдерский блок, что в других LLM-окнах:

- **Ключ OpenAI** и **Модель OpenAI** для режима **OpenAI**;
- **Ключ Gemini** и **Модель Gemini** для режима **Gemini**;
- поле **Модель ... вручную** важнее dropdown, если оно заполнено;
- при пустой модели используется проектный fallback из настроек/избранного, как в остальных окнах.

Если выбран **Готовый JSON / карта**, ключи и модели не нужны.

После запуска:

Что появится	Где искать
Восстановленные DOCX	<code>output\comma_lowercase_restored\</code>
JSON/MD-отчёт GUI-запуска	<code>report\<data>_restore_comma_lowercase\</code>

Что появится	Где искать
JSON/MD-отчёт CLI-запуска по умолчанию	<code>report\comma_lowercase_restore.json</code> и <code>report\comma_lowercase_restore.md</code>
JSON решений, если LLM вызывалась из этого проекта	<code>report\ <data>_restore_comma_lowercase\comma_lowercase_decisions.json</code>

Почему нужен LLM или карта

Исходный и исправленный DOCX показывают только факт правки: например, `Ясенево -> ясенево`. Они не доказывают, была ли исходная заглавная буква правильной. Решение принимает LLM или человек.

Скрипт `system_core\comma_lowercase_restore.py` только применяет это решение: находит hit из отчёта, проверяет координаты в исходном и исправленном DOCX, и восстанавливает регистр в отдельной копии документа.

JSON решений LLM

Формат:

```
{
  "decisions": [
    {
      "file": "Мастер-план.docx",
      "hit": 1,
      "action": "restore",
      "reason": "топоним"
    },
    {
      "file": "Мастер-план.docx",
      "hit": 2,
      "action": "keep",
      "reason": "обычное слово в перечислении"
    }
  ]
}
```

restore возвращает регистр из исходного DOCX. **keep** оставляет исправленный DOCX как есть.

CLI

```
runtime\python.exe system_core\comma_lowercase_restore.py ^  
  --report report\comma_lowercase.json ^  
  --source input ^  
  --fixed output\comma_lowercase ^  
  --decisions report\comma_lowercase_decisions.json ^  
  --output output\comma_lowercase_restored
```

Можно попросить проект сразу классифицировать hits через LLM:

```
runtime\python.exe system_core\comma_lowercase_restore.py ^  
  --report report\comma_lowercase.json ^  
  --source input ^  
  --fixed output\comma_lowercase ^  
  --llm-provider openai ^  
  --output output\comma_lowercase_restored
```

По умолчанию восстанавливаются только срабатывания в ячейках таблиц.

Ручная карта

Если известны устойчивые слова и фразы, их можно добавить в

config\comma_restore_map.yaml. Карта не решает спорные случаи, а только ускоряет повторяемые восстановления.