

Специализированный парсер CIFP для DCS World

Этот инструмент предназначен для обработки данных Navigraph AIRAC в формате CIFP и создания файлов с навигационными данными для карт DCS World и их аэропортов в формате, совместимом с MOOSE Framework.

Возможности

- Обработка файлов CIFP (*.dat) из Navigraph AIRAC
- Автоматическая распаковка CIFP.zip при необходимости
- Парсинг процедур SID, STAR, APPROACH и информации о ВПП
- Извлечение координат путевых точек из файлов earth_fix.dat и earth_nav.dat
- Поддержка ручного ввода координат путевых точек через CSV или JSON файлы
- Фильтрация данных по координатам карт DCS World (Caucasus, Persian Gulf, Syria, Nevada, Marianas)
- Генерация файлов в форматах JSON и Lua для MOOSE Framework
- Подробная статистика и отладочная информация

Требования

- Python 3.6 или выше
- Стандартные библиотеки Python (не требуются дополнительные установки)

Использование

Базовое использование

```
python cifr_parser.py -i /путь/к/директории/с/данными/Navigraph -o /путь/к/директории/для/результатов
```

Расширенное использование

```
# С режимом отладки  
python cifr_parser.py -i /путь/к/директории/с/данными/Navigraph -o /путь/к/
```

директории/для/результатов -d

Без фильтрации по координатам карт DCS World

```
python cifr_parser.py -i /путь/к/директории/с/данными/Navigraph -o /путь/к/директории/для/результатов -n
```

С ручными данными о путевых точках

```
python cifr_parser.py -i /путь/к/директории/с/данными/Navigraph -o /путь/к/директории/для/результатов -m путь/к/файлу/с/ручными/данными.csv
```

Создание шаблона для ручного ввода координат

```
python cifr_parser.py -t
```

Параметры командной строки

- `-i, --input` - Путь к директории с файлами Navigraph AIRAC (обязательный)
- `-o, --output` - Путь к директории для сохранения результатов (обязательный)
- `-d, --debug` - Включить режим отладки с подробным выводом
- `-n, --no-filter` - Отключить фильтрацию по координатам карт DCS World
- `-m, --manual` - Путь к файлу с ручными данными о путевых точках (CSV или JSON)
- `-t, --template` - Создать шаблон CSV файла для ручного ввода координат путевых точек
- `-v, --version` - Показать версию парсера

Структура входных данных

Парсер ожидает следующую структуру директории с данными Navigraph AIRAC:

```
/путь/к/директории/с/данными/Navigraph/  
├── earth_fix.dat (опционально)  
├── earth_nav.dat (опционально)  
├── CIPF/  
│   ├── OMSJ.dat  
│   ├── OMDB.dat  
│   └── ... (другие файлы аэропортов)
```

Или:

```
/путь/к/директории/с/данными/Navigraph/  
├── earth_fix.dat (опционально)  
├── earth_nav.dat (опционально)  
├── CIPF.zip (будет автоматически распакован)
```

Формат ручных данных о путевых точках

CSV формат

```
name,lat,lon,type,frequency,elevation
ANVIX,25.123456,55.654321,FIX,,
GONVI,25.234567,55.765432,FIX,,
RAV,25.345678,55.876543,VOR,114.9,100
ISRE,25.456789,55.987654,ILS,110.3,50
```

JSON формат

```
{
  "ANVIX": {
    "lat": 25.123456,
    "lon": 55.654321,
    "type": "FIX"
  },
  "GONVI": {
    "lat": 25.234567,
    "lon": 55.765432,
    "type": "FIX"
  },
  "RAV": {
    "lat": 25.345678,
    "lon": 55.876543,
    "type": "VOR",
    "frequency": 114.9,
    "elevation": 100
  },
  "ISRE": {
    "lat": 25.456789,
    "lon": 55.987654,
    "type": "ILS",
    "frequency": 110.3,
    "elevation": 50
  }
}
```

Структура выходных данных

Парсер создает следующую структуру директорий и файлов:

```
/путь/к/директории/для/результатов/
├── Caucasus/
```


Неизвестные координаты путевых точек

Если в логах появляются предупреждения о неизвестных координатах путевых точек, это означает, что парсер не смог найти координаты для некоторых путевых точек, используемых в процедурах. Решения:

1. Убедитесь, что файл `earth_fix.dat` присутствует и содержит нужные данные
2. Создайте файл с ручными данными о путевых точках и используйте его с опцией `-m`

Интеграция с MOOSE Framework

Файлы `*_MOOSE.lua` содержат данные в формате, совместимом с MOOSE Framework. Для использования этих файлов в вашем проекте:

1. Скопируйте нужные файлы в директорию вашего проекта
2. Подключите их в вашем коде: ````lua local OMSJ = require("OMSJ_MOOSE")`

```
-- Использование данных local waypoints = OMSJ.Waypoints local sid_procedures =  
OMSJ.SID local star_procedures = OMSJ.STAR local approach_procedures =  
OMSJ.APPROACH ```
```

Автор

Manus AI, Апрель 2025 Версия: 1.1.0