

GPX Parts Extractor

Some downloadable GPX files, like the TET files contain many single domains. This tool can extract them to single files.

This allows much cleaner planning in our route planners

[GPX Parts Extractor](#)

- [ReadMe](#)

ReadMe

GPX Parts Extractor

A Python tool for extracting different components from a GPX file (waypoints, tracks, routes) into separate files for easier management and analysis.

Overview

This tool takes a single GPX file containing multiple components and splits them into individual files:

- **Waypoints** → Single file with all waypoints
- **Tracks** → Individual files for each track
- **Routes** → Individual files for each route (if present)

Features

- **Namespace-aware parsing** - Handles GPX files with XML namespaces correctly
- **Metadata preservation** - Maintains original metadata in extracted files
- **Smart file naming** - Uses track/route names for descriptive filenames
- **Progress reporting** - Shows detailed extraction progress
- **Summary generation** - Creates a summary of all extracted files
- **Error handling** - Graceful failure handling with clear error messages

Requirements

- Python 3.6+
- Standard library only (no external dependencies)

Installation

Simply download or clone the `gpx_extractor.py` file to your local machine.

Usage

Basic Usage

```
```bash
python3 gpx_extractor.py input.gpx
```
```

This extracts all components to a directory named `{filename}_extracted` in the same location as the input file.

Custom Output Directory

```
```bash
python3 gpx_extractor.py input.gpx /path/to/output/directory
```
```

Examples

```
```bash
Extract from your GPS device data
python3 gpx_extractor.py my_vacation_trip.gpx

Extract to a specific location
python3 gpx_extractor.py route_collection.gpx ./separated_routes/

Extract the example file
python3 gpx_extractor.py S.gpx
```
```

Output Structure

After extraction, you'll get:

```
...
output_directory/
├─ filename_waypoints.gpx # All waypoints in one file
├─ filename_track_01_trackname.gpx # Individual track files
├─ filename_track_02_trackname.gpx
├─ ...
├─ filename_route_01_routename.gpx # Individual route files (if any)
├─ ...
└─ filename_extraction_summary.txt # Summary of extraction
...

```

Example Output

When you run the tool on a GPX file like `S.gpx`, you'll see output like this:

```
...
✓ Loaded GPX file: S.gpx
✓ Root namespace: http://www.topografix.com/GPX/1/1

Extracting components from S.gpx...
Output directory: S_extracted
-----
✓ Extracted 909 waypoints to: S_extracted/S_waypoints.gpx
✓ Extracted track 1: 'TET_S-01_20260102' (4557 points) to: S_extracted/S_track_01_TET_S-01_20260102.gpx
```

✓ Extracted track 2: 'TET_S-02_20260102' (4501 points) to: S_extracted/S_track_02_TET_S-02_20260102.gpx

...

✓ Extracted track 28: 'TET_S-28_20260102' (2123 points) to: S_extracted/S_track_28_TET_S-28_20260102.gpx

i No routes found

✓ Created extraction summary: S_extracted/S_extraction_summary.txt

✓ Extraction complete! Created 29 files.

☐ All done! Check the output directory for extracted files.

```

## ## Use Cases

### ### 1. **GPS Device Data Management**

When your GPS device exports a single large GPX file with multiple tracks from different trips, extract them into individual files for better organization.

### ### 2. **Route Planning**

Separate waypoints (POIs) from tracks (actual routes) to use them independently in different mapping applications.

### ### 3. **Data Analysis**

Extract individual tracks for statistical analysis, elevation profiling, or performance comparison.

### ### 4. **Sharing Specific Routes**

Share only specific tracks from a collection without exposing all your GPS data.

### ### 5. **Application Compatibility**

Some GPS applications work better with individual track files rather than combined collections.

## ## File Naming Convention

The tool creates descriptive filenames based on the content:

- **Waypoints**: `{original\_filename}\_waypoints.gpx`
- **Tracks**: `{original\_filename}\_track\_{number:02d}\_{track\_name}.gpx`
- **Routes**: `{original\_filename}\_route\_{number:02d}\_{route\_name}.gpx`

Track and route names are sanitized to be filesystem-safe (spaces become underscores, special characters removed).

## ## Metadata Handling

Each extracted file maintains:

- Original GPX version and namespace declarations

- Preserved metadata from the source file
- Updated timestamp reflecting the extraction time
- Description indicating the extraction source

## ## Error Handling

The tool provides clear error messages for common issues:

- File not found
- Invalid GPX format
- Permission errors
- XML parsing errors

## ## Testing

The tool has been tested with:

- ☐ Large GPX files (164,000+ lines)
- ☐ Multiple namespaced XML elements
- ☐ Complex track structures
- ☐ Files with 900+ waypoints
- ☐ Files with 28+ individual tracks
- ☐ Garmin device exports
- ☐ Various GPS application formats

## ## Example: S.gpx Results

The included `S.gpx` test file demonstrates the tool's capabilities:

**\*\*Input\*\*:** `S.gpx` (164,775 lines, 20+ MB)

**\*\*Results\*\*:**

- 909 waypoints extracted
- 28 individual tracks extracted (ranging from 703 to 9,934 track points each)
- Total of 136,000+ track points processed
- Complete extraction in under 30 seconds

## ## Command Line Options

```

```
usage: gpx_extractor.py [-h] [--verbose] gpx_file [output_dir]
```

Extract waypoints, tracks, and routes from a GPX file into separate files

positional arguments:

gpx_file Input GPX file to extract from

output_dir Output directory (default: same as input file)

optional arguments:

-h, --help show this help message and exit

--verbose, -v Enable verbose output

``

Troubleshooting

Common Issues

1. **"No waypoints/tracks found"**

- Check if your GPX file uses XML namespaces
- Verify the file is a valid GPX format

2. **Permission denied**

- Ensure you have write permissions to the output directory
- Try running with appropriate permissions

3. **Large file processing**

- The tool handles large files efficiently
- For very large files (>100MB), expect longer processing times

Validation

To verify successful extraction:

1. Check the extraction summary file
2. Verify file counts match expectations
3. Open extracted files in your preferred GPS application
4. Compare total point counts with the original file

License

This tool is provided as-is for educational and practical use. Feel free to modify and distribute according to your needs.

Contributing

Suggestions and improvements are welcome! Common enhancement areas:

- Route support improvements
- Additional output formats
- GUI interface
- Batch processing capabilities

****Created for the extract-gpx-parts project****

Making GPX file management easier, one extraction at a time.