

# Image GPS to IFC Marker Exporter

---

This Python project processes images with GPS coordinates, transforms coordinates from WGS84 to any target EPSG code, and exports 3D markers to IFC format with support for both IFC2x3 and IFC4.3 schemas.

## Features

---

- Extract GPS coordinates, elevation, and date from image EXIF data
- Transform coordinates from WGS84 (EPSG:4326) to any target EPSG code (default: EPSG:5110)
- Create 3D cube markers at image locations
- Export markers to IFC format with image references and URLs
- Support for both IFC2x3 and IFC4.3 schemas
- Proper coordinate system representation in IFC4.3 using EPSG codes
- Support for various image formats (JPEG, PNG, TIFF)

## Installation

---

1. Install Python dependencies:

```
pip install -r requirements.txt
```

## Dependencies

The application relies on the following main libraries:

- `ifcopenshell` - For creating and manipulating IFC files
- `Pillow` (PIL) - For image processing and EXIF data extraction
- `pyproj` - For coordinate system transformations
- `tkinter` - For the graphical user interface

All dependencies are listed in the `requirements.txt` file and will be installed automatically with the pip command above.

## Usage

---

Run the EXE file or `main.py` to open the GUI.

Step 1: Create json file from images (1) Select folder where image(s) is located (2) Set EPSG code and test to see if it is correct coordinate system. (3) Generate json file (4) Edit json file if the image URL is different than the file path to the image

Step 2: Create ifc file from json file (5) Select or create a project template settings file to set name of the creator and other name than the default values for the ifc hierarchy. (6) Select folder for ifc output (7) Set name of ifc file (8) Specify schema for ifc file, 3x3 or 4.3 (9) Generate ifc

Image GPS to IFC Converter

Help

# Image GPS to IFC Converter

1

Extract Images to JSON

Image folder path: 

1 Select folder

Output coordinate system (EPSG Code): 

2 Check

3 Generate json

4

Edit json file manually in text editor

2

Generate IFC from JSON

Json image processed images: 

Select file

 — Use json from step 1

Project settings json file: 

5 Select file

Generate template file

Output folder for ifc file: 

6 Select folder

IFC filename: 

7

IFC Schema version: ☒ IFC2x3 ☐ IFC4.3 

8

9 Generate IFC

Ready

## Configuration

Edit `config.py` to customize:

- Input/output paths
- Target EPSG code for coordinate transformation
- Cube marker radius and materials
- URL patterns for hosted images
- IFC schema version (IFC2x3 or IFC4X3)

## Output

The script generates:

- `output.ifc` - Default IFC file with 3D markers
- `markers.ifc` - IFC file with 4x4x4 meter cube markers
- `extracted_images.json` - Summary of processed images and coordinates

## Using the GUI

The application includes a graphical user interface:

```
python -m src.gui.main_window
```

The GUI allows you to:

1. Select input image folders
2. Specify the target EPSG code for coordinate transformation
3. Choose between IFC2x3 and IFC4.3 schemas
4. Process images and generate IFC files in a step-by-step workflow

## Technical Details

---

### Coordinate System Handling

- Source coordinates are always in WGS84 (EPSG:4326) as provided by GPS
- The application can transform to any target EPSG code (default is EPSG:5110, Norwegian coordinate system)
- For IFC4.3 exports, the EPSG code is properly embedded in the IFC file using:
  - `IfcProjectedCRS` - Represents the target coordinate system
  - `IfcMapConversion` - Connects the project coordinate system to the EPSG reference

### GPS Extraction Process

The application handles various formats of GPS data in EXIF:

1. Extracts GPS coordinates in DMS (Degrees, Minutes, Seconds) format
2. Converts to decimal degrees
3. Transforms to the target coordinate system
4. Applies elevation data if available

## IFC File Specifications

---

- **Schemas:**
  - IFC2x3 (compatible with most viewers like Solibri and BIMvision)
  - IFC4.3 (with proper coordinate system representation)
- **Units:** Meters
- **Markers:** 4x4x4 meter cubes (`IfcBuildingElementProxy`)
- **Coordinates:**
  - IFC2x3: Direct placement using transformed coordinates
  - IFC4.3: Proper coordinate system representation using EPSG codes
- **Hierarchy:** Project → Site → Building → Storey → Markers
- **Metadata:** Complete image metadata as IFC properties
- **Document Links:** Image file paths accessible through document references