

## Batch Processing in Python - AEM Forms to JSON Conversion

### Overview

This document provides a guide to implementing batch processing for converting AEM Forms (XFA/XML) to Digital Forms JSON using Python. The solution is designed to be cross-platform (Windows, Mac, and Linux) and supports parallel processing for improved efficiency.

---

### Steps to Ensure Cross-Platform Batch Processing

1. **Read all XDP files** from a folder (`data/input/`).
  2. **Use multiprocessing** for parallel processing to handle multiple files simultaneously.
  3. **Convert XDP to JSON** using the defined conversion logic.
  4. **Store the JSON output** in the `data/output/` directory.
  5. **Log successes, errors, and summary statistics** in the `data/report/` directory.
- 

### Cross-Platform Compatibility

#### 1. OS-Independent File Handling

- Different OS use different path formats:
  - **Windows:** Uses backslashes (`C:\Users\...`).
  - **Mac/Linux:** Uses forward slashes (`/home/user/...`).
- **Solution:** Use `os.path.join()` or `pathlib.Path()` instead of hardcoded paths.

#### 2. Path Separators

- Windows uses `\` (backslash) in file paths.
- Mac/Linux use `/` (forward slash).
- **Solution:** Use `os.sep` to handle different path separators dynamically.

#### 3. Script Execution Differences

- **Mac/Linux:** Use shebang (`#!/usr/bin/env python3`) at the start of the script.
- **Windows:** Ensure execution with `python` instead of `python3`.

#### 4. Multiprocessing Compatibility

- Python's `multiprocessing` module has different behaviors on Windows:

- **Windows requires:** `if __name__ == "__main__":` before using multiprocessing.
  - **Linux/Mac** don't have this restriction.
- 

## How Error Handling & Reporting Works

### 1. Try-Except Blocks

- If an error occurs in `convert_xdp_to_json()`, it is caught and logged instead of stopping the script.
- Example: If a file is corrupted, it will be logged, and processing will continue for other files.

### 2. Logging & Reporting

- **INFO:** Logs successful conversions.
- **WARNING:** Logs when no XDP files are found.
- **ERROR:** Logs errors encountered during conversion.
- **Summary Statistics:** Captures the total number of fields mapped, unmapped fields, and any required manual interventions.

### 3. Report File Location

- All reports are stored in the `data/report/` directory with filenames following this format:
    - `{xml_filename}_report_<YYYYMMDD_HHMMSS>.json`
- 

## Mapping JSON: How It Works & Future Enhancements

### How Mapping JSON Works

- The mapping JSON file (`xml_mapping.json`) defines how XML elements are converted into JSON.
- Each XML field is mapped to a corresponding JSON field using predefined rules.
- If an XML field is missing a mapping, it is logged under **manual intervention required** in the report.

### How to Modify Mapping JSON for Future Enhancements

- To **add new fields**, update `xml_mapping.json` with the required mappings.

- To **change existing mappings**, modify the JSON keys and values to match the new structure.
  - **For new XML formats**, ensure all required elements are mapped before running the script.
  - **Ensure consistency** by following existing naming conventions and structure in the mapping file.
- 

## Conclusion

This document outlines the batch processing approach for AEM Forms to JSON conversion. It ensures cross-platform compatibility, efficient error handling, and logging for debugging and tracking. The mapping JSON structure allows flexibility for future enhancements, ensuring adaptability as requirements evolve.

---