Lifestance Scope Conversion Documentation

# Overview

This workflow automates the conversion of raw provider and location data into a structured Excel output. Using a predefined template and a suite of Python scripts, it extracts, cleans, validates, and formats data, and applies dropdowns and formulas for consistent downstream use.

# Main Workflow

1. Input File: 'Excel Files/***Input.xlsx***' (raw provider/location data)

2. Template File: 'Excel Files/***New Business Scope Sheet - Practice Locations and Providers.xlsx'***

* Note: Modify this file if structure, columns, or validations need updates.

3. Main Script: '***\_main\_1.py***' orchestrates the process:

* Extracts and transforms data via helper scripts
* Copies template structure and populates output
* Calls ’***Location.py***' for the Location sheet
* Adds dropdowns and validations
* Outputs to 'Excel Files/***Output.xlsx***'

# Python Scripts Summary

|  |  |
| --- | --- |
| Script Name | Purpose |
| \_main\_1.py | Main orchestrator of the workflow. |
| Location.py | Generates the Location sheet, standardizes and cleans addresses. |
| Name.py | Extracts provider names and gender. |
| Npi.py | Extracts NPI numbers. |
| Headshot.py | Extracts headshot URLs. |
| professional\_suffix.py | Extracts and applies professional suffixes (e.g., MD, PhD). |
| Specialty.py | Extracts specialties and applies specialty dropdowns. |
| PatientsAccepted.py | Handles patient types accepted dropdown. |
| Education.py | Extracts education and school details. |
| Professional\_statement.py | Extracts provider bios/statements. |
| Board\_certification.py | Extracts board certifications and subspecialties. |
| Boardcertification.py | Duplicate of the above; legacy script. |
| optoutrating.py | Adds 'Opt Out of Ratings' dropdown. |
| ESF.py | Adds 'Enterprise Scheduling Flag' dropdown. |
| Langauge.py | Extracts spoken languages. |
| provider\_dropdowns.py | Applies dropdowns and formulas to Provider sheet. |
| specialtydropdown.py | Applies specialty dropdowns. |
| \_status \_check.py | Validates output file completeness and structure. |
| Searchname.py | Unused placeholder. |

# Excel Files Summary

* ***Input.xlsx:*** Raw input data from Lifestance.
* ***New Business Scope Sheet -*** Practice Locations and Providers.xlsx: Template for output structure and validation.
* ***C1 Street Suffix Abbreviations.xlsx:*** Reference for standardizing street suffixes.
* ***Output.xlsx***: Final output file generated by the workflow.

# How the Scripts Interact

***\_main\_1.py*** is the main script that imports and calls all other scripts. Each helper script focuses on specific data fields. Location.py is called as a subprocess for generating the Location sheet. Dropdowns and validations are applied by dedicated scripts. Output completeness is checked by ***\_status \_check.py***.

# How to Generate the Output File

## Prerequisites

1. Ensure Python 3.x is installed.

2. Install required packages with: ***pip install -r requirements.txt***

3. Ensure the following files are in the 'Excel Files/' directory:

* ***Input.xlsx***
* ***New Business Scope Sheet - Practice Locations and Providers.xlsx***
* ***C1 Street Suffix Abbreviations.xlsx***

## Steps to Run

1. Open terminal/command prompt.

2. Navigate to the project directory: ***cd <path-to-project-directory>***

3. Run the main script: ***python \_main\_1.py***

4. Output will be generated as: 'Excel Files/***Output.xlsx***'

## Post-Generation Actions

1. ***Output.xlsx*** automatically to verify results.

2. Check for required data, dropdowns, and validations.

3. If needed, manually edit in Excel using dropdowns or rerun the process with updated input/template.