

```

import sys

from pipelines.registry_pipelines import all_registries_pipeline
from utils.fms import init_registry_dirs
from utils.helpers import handle_cli_verbosity

def update_all_registries(argv):
    """
    Runs steps to extract aircraft registry data from all remote sources, transform the extracted data into a
    standardized format, and load the transformed data into tabular stores used to seed or update components of a data
    warehouse

    :param (list) argv: command line arguments - pass 'verbose' to print progress messages
    :return: effect - creates [DATA_DIR]/world[REGISTRY.name] file;
             effect - creates [DATA_DIR]/aircrafts[TABLE.name] file;
             effect - creates [DATA_DIR]/registrants[TABLE.name] file;
             effect - creates [DATA_DIR]/registrations[TABLE.name] file;
             effect - creates [DATA_DIR]/[REGISTRY_INPUT_DIR]/[registry_code]/[file or directory] asset(s) for each remote source;
             effect - creates [DATA_DIR]/[REGISTRY_OUTPUT_DIR]/[registry_code]/[registry_code][REGISTRY.name] file for each remote source
    """

    # initialize registry data directories
    init_registry_dirs()

    # assign whether progress messages should be printed
    verbose = handle_cli_verbosity(argv, 1)

    # extract, transform, and load each remote registry data source and compile world registry data
    all_registries_pipeline(verbose=verbose)

if __name__ == "__main__":
    update_all_registries(sys.argv)

```