MapReduce Phase 2 - Build and Run Instructions

This project extends Phase 1 by adding support for dynamically loaded Map and Reduce DLLs (plugins). The program can run either using built-in Mapper/Reducer classes (Phase 1 mode) or external DLLs (Phase 2 mode).

Folder Structure

include/ — header files (e.g., mr/*.hpp) dlls/ — contains MapDLL.cpp and ReduceDLL.cpp for plugin

build output/ — program writes final results here temp/ — intermediate MapReduce output

sample_input/ — input text files to process out/build/x64-Debug/bin/ — executables and DLLs are generated here

Building the Project

1. Open the project in Visual Studio (Community Edition 2022 or later). 2. Select **x64-Debug** configuration. 3. Build All (CMake will create mapreduce_gui.exe, mapreduce_cli.exe, Map.dll, Reduce.dll). 4. Output binaries will appear in out/build/x64-Debug/bin.

Running the Program

Phase 2 (DLL Mode)

From a Developer Command Prompt or PowerShell, navigate to the binary folder:

cd "C:\Users\power\Documents\syracuse_university\Object Oriented Design\Project#2\CSE-687-\out\build\x64-Debug\bin"

Then run:

mapreduce_cli.exe <inputDir> <dllDir> [outputDir] [tempDir]

Example:

mapreduce_cli.exe sample_input . output temp

This will process all .txt files in sample_input, use Map.dll and Reduce.dll from the current directory, and output results to output/word_counts.txt.

Phase 1 (Built-in Mode)

To run using the built-in Mapper/Reducer (without DLLs), launch the GUI executable:

mapreduce_gui.exe

Then click the **Run MapReduce** button to process the default sample_input directory.

Expected Output

After a successful run, the following files will appear:

temp/intermediate.txt — intermediate word-count pairs output/word_counts.txt — final aggregated

results output/_SUCCESS.txt — success marker

Troubleshooting

Always wrap paths with spaces in double quotes ("..."). If DLLs are not found, confirm Map.dll and

Reduce.dll exist in the DLL directory you pass as argument. If CMake complains about mismatched

cache, select “Delete Cache and Reconfigure” in Visual Studio. Ensure you use the x64 toolchain for

both build and run.