

# Developer Guide

## Overview

This project is a C++14 console application for scanning directories and reporting file statistics, designed for easy extension and maintenance.

## Project Structure

### DirectoryScannerTool/

```
|
|  └─ include/
|      └─ DirectoryScanner.h    # DirectoryScanner class declaration
|  └─ source/
|      └─ DirectoryScanner.cpp  # DirectoryScanner class implementation
|  └─ main.cpp                 # Application entry point
```

## Building the Application

### Requirements

- Visual Studio 2022 or later (C++14 support)
- Windows OS (using Windows API for file operations)

### Steps

- Open the solution in Visual Studio.
- Build the solution (Build > Build Solution).

## Code Overview

### main.cpp

- Parses command-line arguments and calls the main menu function.

### DirectoryScanner

- Handles user interaction, directory scanning, and output.
- Supports both interactive and command-line-driven operation.
- Uses Windows API for directory and file operations.
- Class is separated by Core functions (MainMenu, ScanInputDirectory, etc) and Helper functions (IsValidExtensions, ScanDirectoryRecursive, etc).

## Adding/Modifying File Extensions

- Valid file extensions are defined in DirectoryScanner.h as a **std::vector<std::string>** **m\_ValidExtensions**.

- To add or remove extensions, modify this vector.

## **Error Handling**

- The application validates user input and file paths.
- Exits gracefully on user request or unrecoverable errors.

## **Maintenance Tips**

- Keep logic for user interaction and file scanning separated for clarity.
- Use the provided setter methods for changing internal state.
- Follow the existing commenting and code style for consistency.