

# **Sales Dashboard - Windows EXE Build Instructions**

## **Prerequisites**

- Docker Desktop installed
- Ubuntu WSL (Windows Subsystem for Linux) with Docker support, OR Docker on Windows
- Approximately 5-10 GB free disk space
- All shapefile files (.shp, .shx, .dbf, .prj, .sbn, .sbx, .xml, .cpg, .json)

## **Project Structure**

Create this folder structure on your machine:

```
SalesDashboard/  
├── app.py          # Main application file  
├── Dockerfile      # Docker build configuration  
├── requirements.txt # Python dependencies  
├── app_icon.ico    # Application icon (optional but recommended)  
├── build_windows_exe.sh # Build script  
└── shapefiles/  
    ├── adm01.shp  
    ├── adm01.shx  
    ├── adm01.dbf  
    ├── adm01.prj  
    ├── adm01.sbn  
    ├── adm01.sbx  
    ├── adm01.xml  
    ├── adm01.cpg  
    └── adm01.json
```

## Step-by-Step Build Instructions

### 1. Prepare Your Files

1. Copy `app.py` (from the provided code) to your project folder
2. Copy `requirements.txt` to your project folder
3. Copy `Dockerfile` to your project folder
4. Create a `shapefiles/` subfolder
5. Place all your shapefile files (.shp, .shx, .dbf, .prj, etc.) in the `shapefiles/` folder
6. (Optional) Create or add an `app_icon.ico` file to the root folder

## 2. Create Build Script (Windows)

Save this as `build_windows_exe.bat`:

```
batch

@echo off
REM Build Sales Dashboard Windows EXE using Docker

echo Building Sales Dashboard EXE...
docker build -t sales-dashboard-builder .

echo Creating dist folder...
if not exist dist mkdir dist

echo Running PyInstaller inside Docker...
docker run --rm -v %cd%\dist:/app/dist sales-dashboard-builder

echo Build complete!
echo EXE file should be in: dist\SalesDashboard.exe
pause
```

## 3. Create Build Script (Linux/Mac)

Save this as `build_windows_exe.sh`:

```
bash
```

```
#!/bin/bash
```

```
# Build Sales Dashboard Windows EXE using Docker
```

```
echo "Building Sales Dashboard EXE..."
```

```
docker build -t sales-dashboard-builder .
```

```
echo "Creating dist folder..."
```

```
mkdir -p dist
```

```
echo "Running PyInstaller inside Docker..."
```

```
docker run --rm -v $(pwd)/dist:/app/dist sales-dashboard-builder
```

```
echo "Build complete!"
```

```
echo "EXE file should be in: dist/SalesDashboard.exe"
```

Make it executable: `chmod +x build_windows_exe.sh`

## 4. Build the EXE

### On Windows:

1. Open Command Prompt or PowerShell in your project folder
2. Run: `build_windows_exe.bat`

### On Linux/Mac:

1. Open Terminal in your project folder
2. Run: `./build_windows_exe.sh`

## 5. Wait for Build to Complete

The build process will:

- Take 15-30 minutes depending on your system
- Download and package all Python dependencies
- Create the standalone executable
- Generate the `dist/SalesDashboard.exe` file

## 6. Distribute the EXE

The `SalesDashboard.exe` file is now standalone and requires NO additional installation. Simply distribute this file to end users.

## Running the Application

1. Double-click `SalesDashboard.exe`
2. Streamlit app opens in your default browser (usually <http://localhost:8501>)
3. Upload your CSV file
4. The shapefile data is already embedded - no additional files needed

## CSV File Format

Your CSV file should include these columns:

### Required Columns:

- `Amount_(BDT)` - Sales amount in Bangladeshi Taka
- `Qty_(Pcs)` - Quantity in pieces

- Qty\_(KG) - Quantity in kilograms

**Time Columns:**

- Month - Numeric (1=January, 2=February, ..., 12=December)
- Year - Numeric year (2022, 2023, etc.)
- Quarter - Numeric quarter (1, 2, 3, 4) [Optional]
- FY - Fiscal year [Optional]

**Categorical Columns (Optional but recommended):**

- Brand - Product brand
- DivName - Division name
- SKU - Stock keeping unit
- TeaType - Type of tea
- PackType - Package type
- Segment - Market segment
- AdmDiv - Administrative division (for geographic mapping)
- Flavor - Tea flavor
- FlavorType - Type of flavor
- Misc - Miscellaneous category

**Example CSV Row:**

Amount\_(BDT),Qty\_(Pcs),Qty\_(KG),Month,Year,Quarter,FY,Brand,DivName,SKU,TeaType,AdmDiv  
150000,1000,50,1,2023,1,FY2023,BrandA,Division1,SKU001,BlackTea,Dhaka

## Troubleshooting

### EXE not starting

- Check that Windows Defender/Antivirus isn't blocking it
- Run Command Prompt and execute: `SalesDashboard.exe`
- Look for error messages in the command prompt

### Port already in use error

- The app uses port 8501. If occupied, close other Streamlit apps
- Or Streamlit will automatically use the next available port

### CSV upload fails

- Verify CSV is comma-separated (not semicolon)
- Check column names match exactly (case-sensitive)
- Ensure numeric columns contain only numbers (no text)

### Map not displaying

- Verify `AdmDiv` column exists in your CSV
- Check shapefile files are all present in the dist folder
- Ensure division names in CSV match shapefile division names

### Build fails with Docker error

- Ensure Docker is running
- Try: `docker system prune` to free space
- Increase Docker's memory allocation to 4GB+

## Technical Details

The executable:

- Uses Python 3.10 runtime (embedded)
- Bundles all Python packages
- Includes complete geopandas/shapely for GIS operations
- Shapefile data is embedded within the executable
- Requires 500MB-1GB disk space when extracted during first run
- Creates temporary files in Windows %TEMP% folder

## Support for Month Numbering

The application automatically:

- Converts numeric month values (1-12) to month names
- Displays month names in charts and tables
- Uses month numbers internally for calculations
- Example: Input `Month=7` displays as "July" in visualizations

## Updating the Application

To update the application with new features:



1. Modify `app.py` as needed
2. Update `requirements.txt` if adding new Python packages
3. Run the build script again
4. Distribute the new `SalesDashboard.exe`

## File Size Reference

- Final EXE: ~400-600 MB
- Installation size (unpacked): ~1-1.5 GB
- Runtime memory: 500MB-1GB depending on data size

## Version Information

Built with:

- Python 3.10
- Streamlit 1.28.1
- Geopandas 0.13.2
- Plotly 5.17.0
- Pandas 2.0.3