# Flutter Fundamentals
# Creating Production Builds

Peter Kassenaar –
info@kassenaar.com

# Creating a Production Build

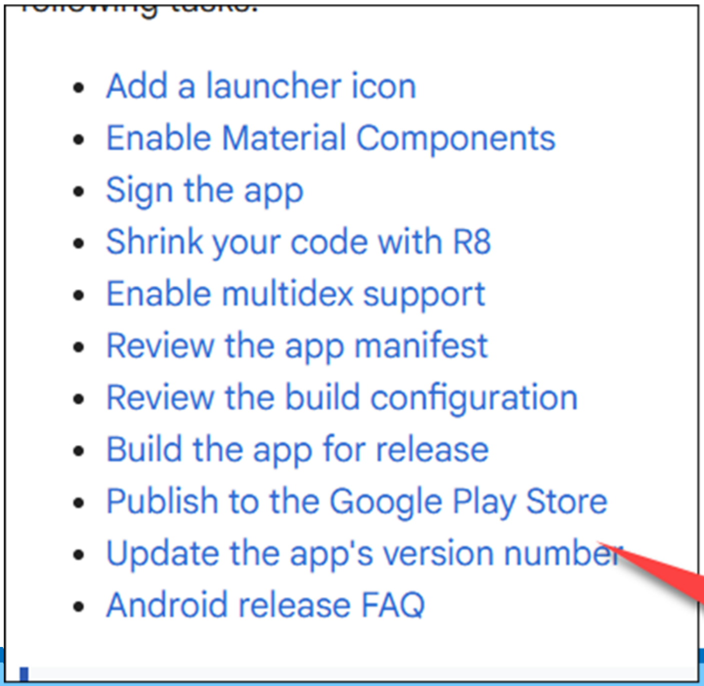
Deploying your app to various platforms

## Deployment

- Many deployments possible!
  - Android apps – publication in *Google Play Store*
  - iOS apps – publication in *Apple App Store*
  - macOS apps – *Apple App Store*
  - Linux apps – publication to *Snap Store* or other channel
  - Windows apps – publication on *internal network* (mostly)
  - Web apps – publication on *(internal) web server*
- ...

## Follow steps for YOUR platform

- Read the recipies
- Often:
  - Creating launcher icons
  - Signing your app with a specific key
  - Build for release
  - Publish to destination
  - ...



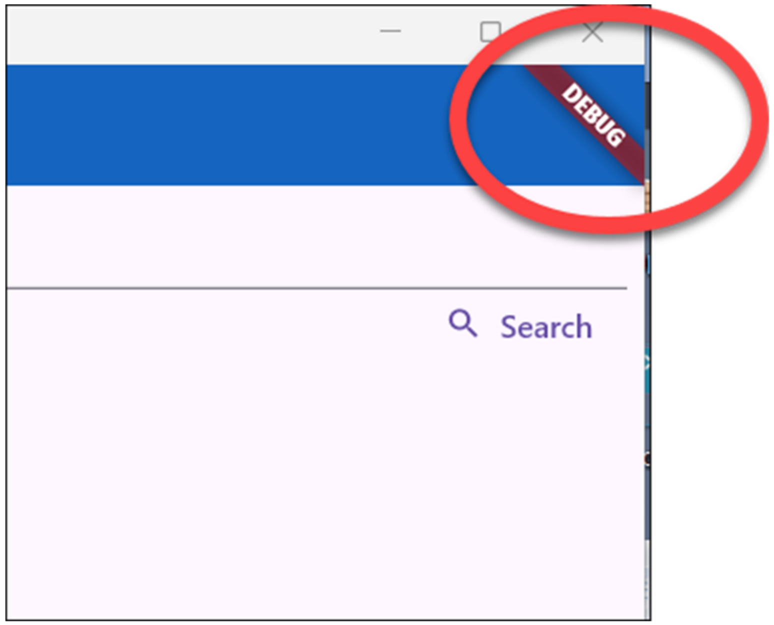
following tasks:

- Add a launcher icon
- Enable Material Components
- Sign the app
- Shrink your code with R8
- Enable multidex support
- Review the app manifest
- Review the build configuration
- Build the app for release
- Publish to the Google Play Store
- Update the app's version number
- Android release FAQ

Example

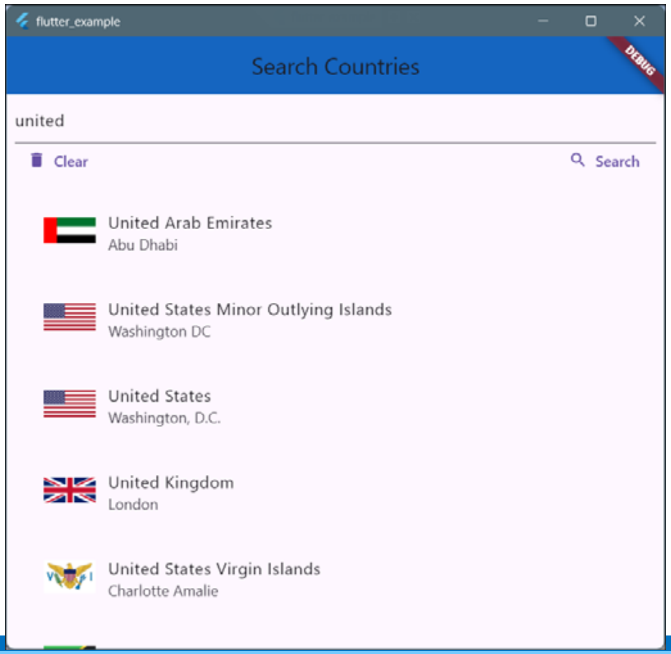
## Building for Windows

- Example – building a Windows app
- Default – when running from IDE, debug version
- Goal – create a standalone app (no debug version)



## Prerequisites

- We’re building a deployment version of `../examples/_310-routing-detail`
  - There are other apps available, this is just a choice
  - NO signing for publishing in Windows Store



united

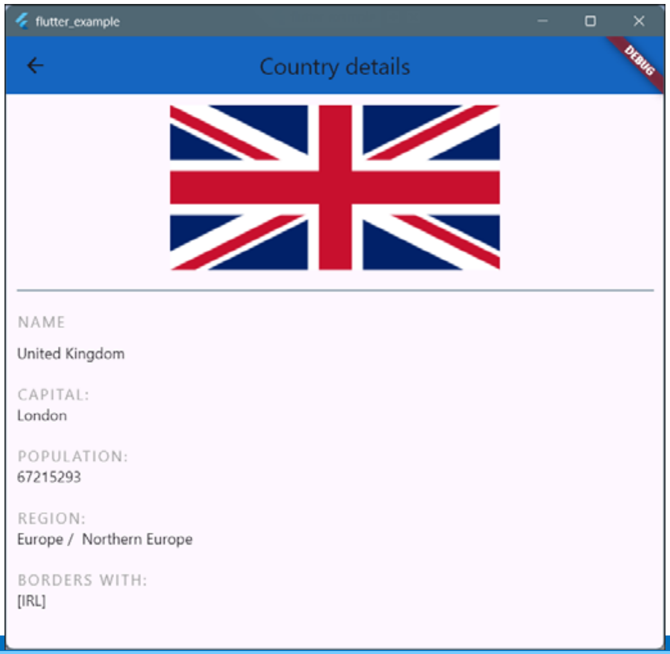
United Arab Emirates
Abu Dhabi

United States Minor Outlying Islands
Washington DC

United States
Washington, D.C.

United Kingdom
London

United States Virgin Islands
Charlotte Amalie



NAME
United Kingdom

CAPITAL:
London

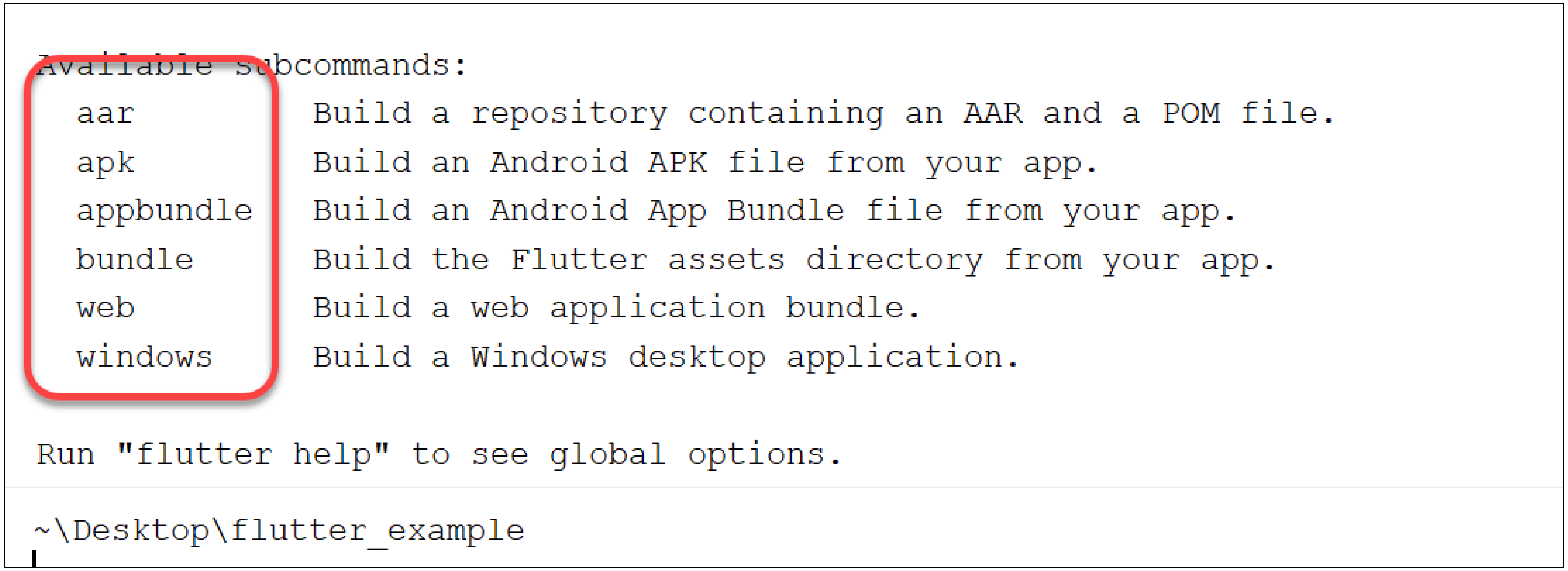
POPULATION:
67215293

REGION:
Europe / Northern Europe

BORDERS WITH:
[IRL]

## General workflow

Use the `flutter build <platform>` command

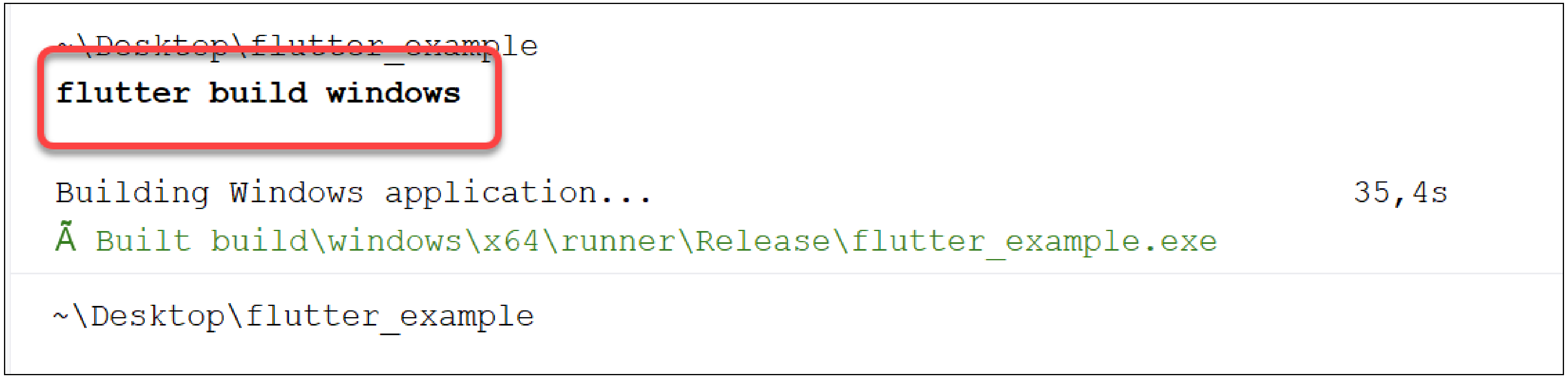


```
Available subcommands:
  aar         Build a repository containing an AAR and a POM file.
  apk         Build an Android APK file from your app.
  appbundle   Build an Android App Bundle file from your app.
  bundle      Build the Flutter assets directory from your app.
  web         Build a web application bundle.
  windows     Build a Windows desktop application.

Run "flutter help" to see global options.

~\Desktop\flutter_example
```

## We are building a Windows application

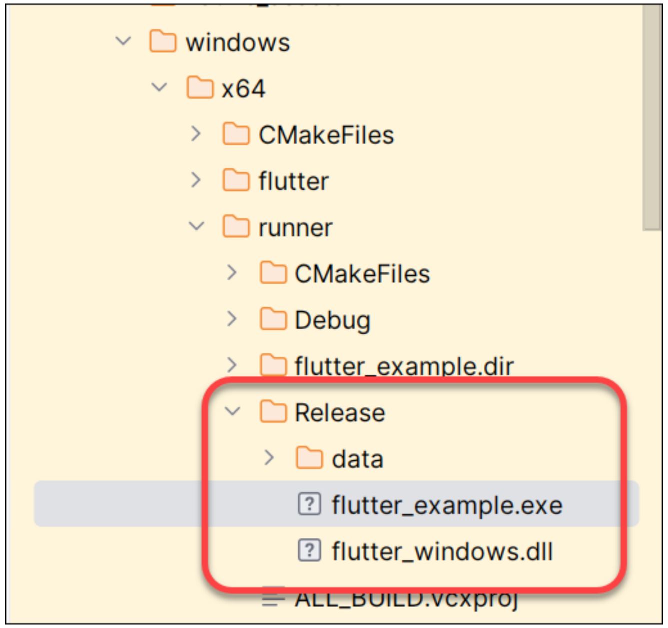
- Command `flutter build windows`
  - Use your IDE or a command line terminal
- Executable is stored in

  `build\windows\x64\runner\Release\<appName>.exe`



```
~\Desktop\flutter_example
flutter build windows

Building Windows application...                                  35,4s
Ã Built build\windows\x64\runner\Release\flutter_example.exe

~\Desktop\flutter_example
```

# Deployment of your app

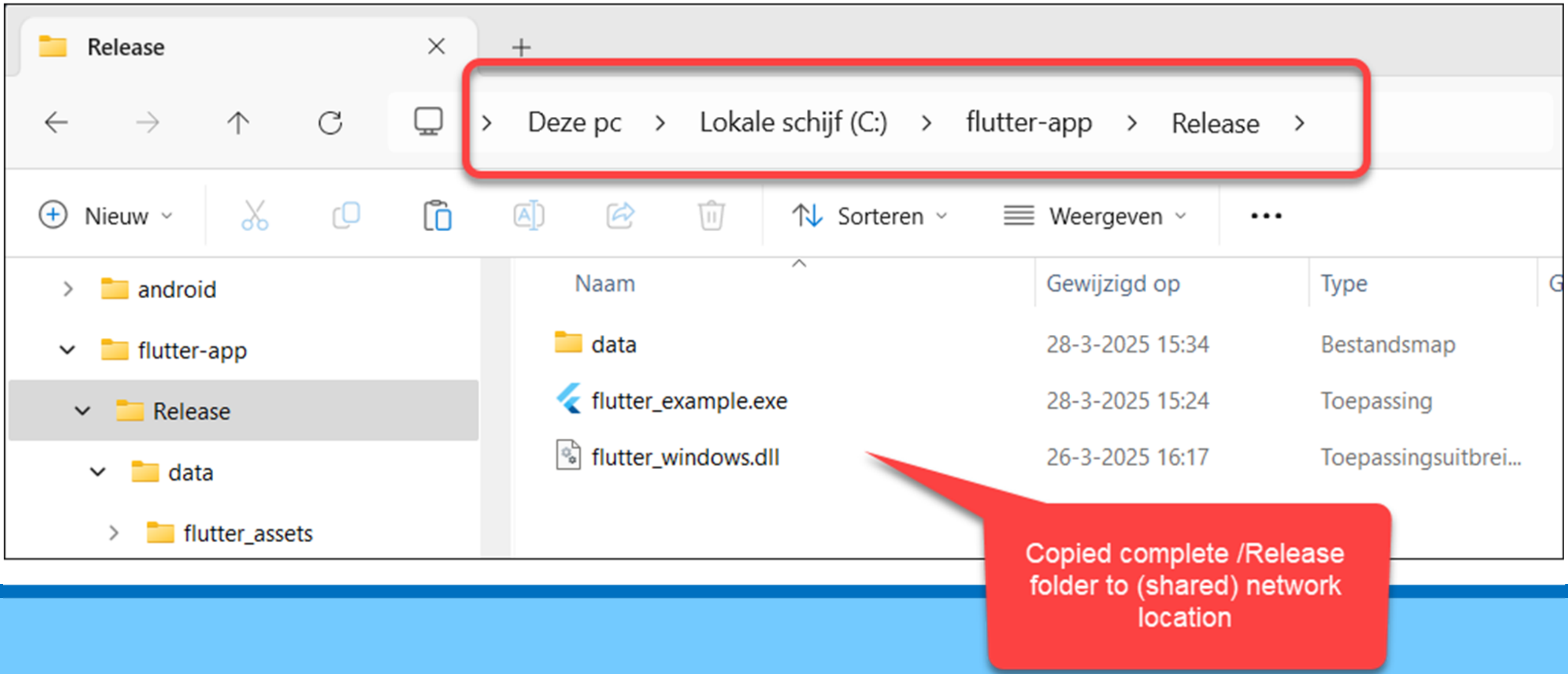


```
windows
  x64
    CMakeFiles
    flutter
    runner
      CMakeFiles
      Debug
      flutter_example.dir
      Release
        data
        flutter_example.exe
        flutter_windows.dll
      ALL_BUILD.vcxproj
```

- Distribute the complete contents of the generated `Release` directory.
  - NOT just the generated .exe file
- The .exe depends on
  - DLLs like `flutter_windows.dll`, and possibly more
  - data folder with assets, ICU data, AOT runtime, etc.
- Otherwise your app will crash or show runtime errors.

## Approach for deployment

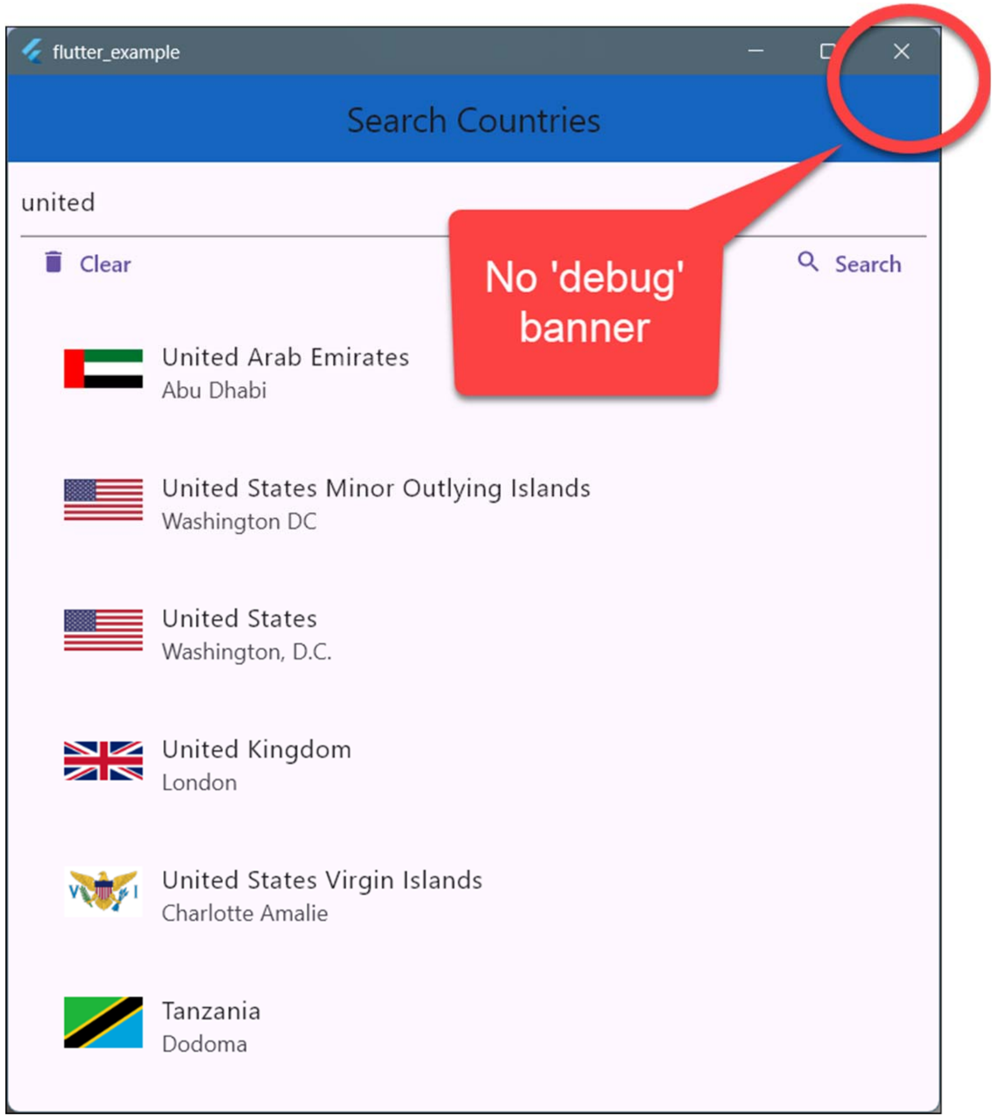
- Copy the entire Release folder to network drive or Artifactory, or:
  - Create a ZIP of it and extract it where needed
  - Write a simple script or installer that does the copy/deployment



Deze pc › Lokale schijf (C:) › flutter-app › Release ›

| Naam | Gewijzigd op | Type |
|---|---|---|
| data | 28-3-2025 15:34 | Bestandsmap |
| flutter_example.exe | 28-3-2025 15:24 | Toepassing |
| flutter_windows.dll | 26-3-2025 16:17 | Toepassingsuitbrei... |

## Results



flutter_example

Search Countries

united

Clear

Search

United Arab Emirates
Abu Dhabi

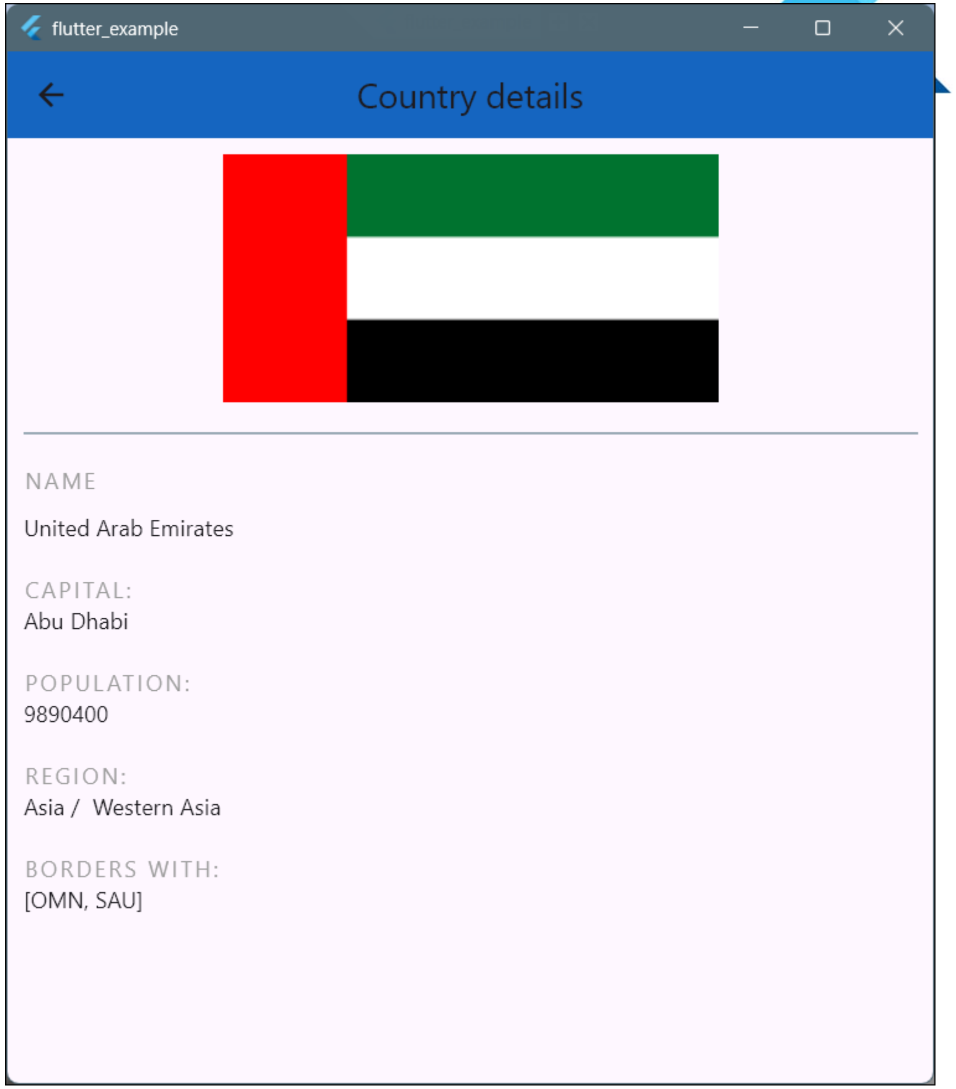
United States Minor Outlying Islands
Washington DC

United States
Washington, D.C.

United Kingdom
London

United States Virgin Islands
Charlotte Amalie

Tanzania
Dodoma



flutter_example

Country details

NAME
United Arab Emirates

CAPITAL:
Abu Dhabi

POPULATION:
9890400

REGION:
Asia /  Western Asia

BORDERS WITH:
[OMN, SAU]

## Sample PS script to automate build & deploy

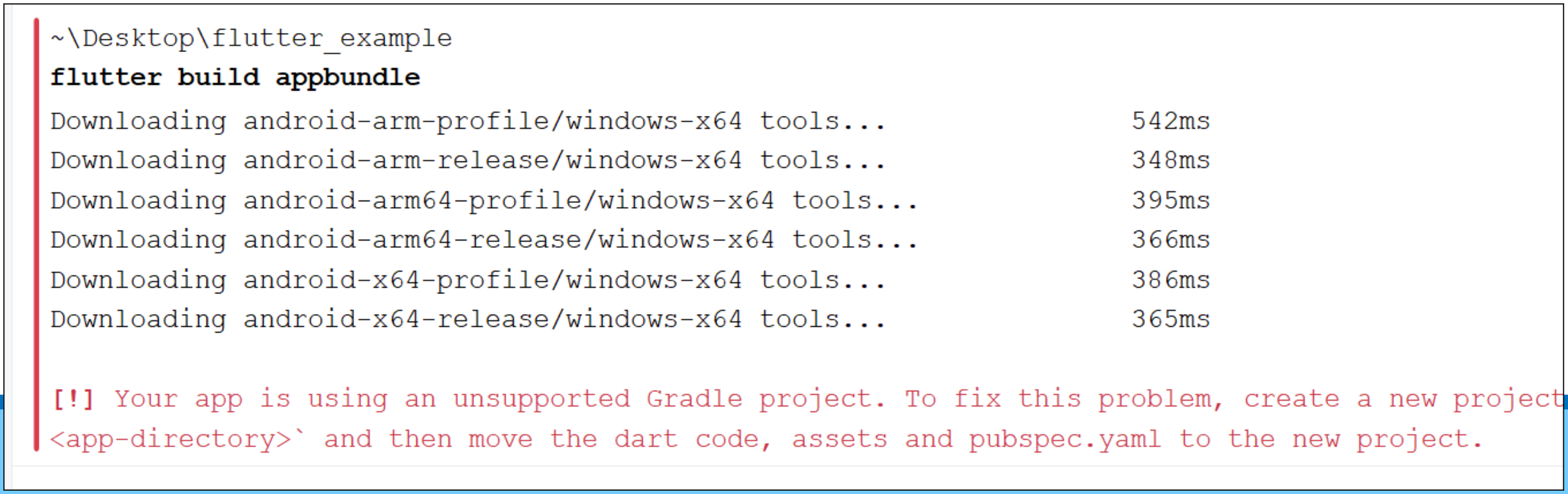
```powershell
# build_and_deploy.ps1. Easily build and deploy a Flutter-windows application
# from the command line in PowerShell.

# Set variables
$projectRoot = Split-Path -Parent $MyInvocation.MyCommand.Definition
$buildPath = Join-Path $projectRoot "build\windows\x64\runner\Release"

# Replace `your-network-drive` with actual UNC-path
$targetPath = "\\your-network-drive\path\to\app-deploy"

# Step 1: Build the app
Write-Host "Building Flutter Windows app..."
flutter build windows

# Step 2: Ensure target exists
if (!(Test-Path -Path $targetPath)) {
    Write-Host "Creating target directory at $targetPath"
    New-Item -ItemType Directory -Path $targetPath | Out-Null
}

# Step 3: Copy files
Write-Host "Copying release build to target..."
Copy-Item -Path "$buildPath\*" -Destination $targetPath -Recurse -Force

Write-Host "Done. App deployed to $targetPath"
```

## Usage

- Save the script as `build_and_deploy.ps1` in Flutter project root.
- Edit `$targetPath` to point to your actual network path.
- Run it from PowerShell
  - `./build_and_deploy.ps1`

## Building for other platforms

- Make sure to have the platform prerequisites installed
  - Android: Android SDK, correct PATH, etc
  - iOS: Xcode, command line tools, etc.
  - Web: fonts, assets, etc.
- Otherwise: errors



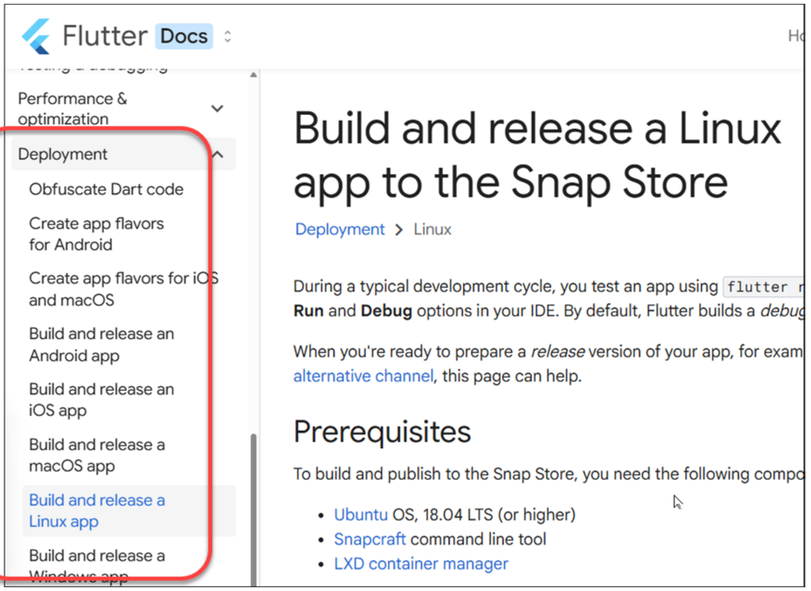
```
~\Desktop\flutter_example
flutter build appbundle
Downloading android-arm-profile/windows-x64 tools...             542ms
Downloading android-arm-release/windows-x64 tools...             348ms
Downloading android-arm64-profile/windows-x64 tools...           395ms
Downloading android-arm64-release/windows-x64 tools...           366ms
Downloading android-x64-profile/windows-x64 tools...             386ms
Downloading android-x64-release/windows-x64 tools...             365ms

[!] Your app is using an unsupported Gradle project. To fix this problem, create a new project
<app-directory>` and then move the dart code, assets and pubspec.yaml to the new project.
```

## Other platforms - more

- Android
  - Add launcher icon, sign the app, shrink app with R8, create App Manifest file, Build the app for release, Publish to Google Play Store
  - https://docs.flutter.dev/deployment/android
- iOS
  - Create app outline in App Store Connect, register Bundle ID, prepare app in Xcode, add App icon, launch image, build Archive from Xcode, create and upload app bundle
  - https://docs.flutter.dev/deployment/ios

## Workshop

- Use your own app, or use one of the example apps
- Create a distribution build for your platform
- Follow the steps described in the Flutter Docs, Deployment section



Flutter Docs
Performance & optimization
Deployment
Obfuscate Dart code
Create app flavors for Android
Create app flavors for iOS and macOS
Build and release an Android app
Build and release an iOS app
Build and release a macOS app
Build and release a Linux app
Build and release a Windows app

Build and release a Linux app to the Snap Store

Deployment > Linux

During a typical development cycle, you test an app using `flutter r`
**Run** and **Debug** options in your IDE. By default, Flutter builds a *debug*

When you're ready to prepare a *release* version of your app, for exam
alternative channel, this page can help.

Prerequisites

To build and publish to the Snap Store, you need the following compo

- Ubuntu OS, 18.04 LTS (or higher)
- Snapcraft command line tool
- LXD container manager