Mobile Programming

App Deployment & Publishing

Lecturer: Ali Najimi
Faculty of Computer Engineering
Sharif University of Technology
Fall 2027

Google Play | App Store | Signing | Release



Table of Contents

1. Build Configurations (Debug / Profile / Release)
2. Android Signing & APK/AAB
3. iOS Signing & Provisioning
4. Google Play Store Publishing
5. Apple App Store Publishing
6. Flutter Build Commands
7. App Icons & Splash Screens
8. Versioning & Release Management
9. Fastlane — Automated Deployment
10. Post-Release: Monitoring & Updates
11. Summary & Key Takeaways

Build Configurations

Flutter has three build modes, each with a different purpose.

Mode	Use Case	Assertions	Performance	Size
Debug	Development	Enabled	Slow	Large
Profile	Performance testing	Disabled	Near-release	Medium
Release	Production	Disabled	Optimized	Small

```
flutter run                    # Debug
flutter run --profile          # Profile
flutter run --release          # Release
flutter build apk --release    # Build release APK
flutter build ios --release    # Build release iOS
```

Android Signing

Every Android app must be **digitally signed** before installation.

Two keystores:

- **Debug keystore:** Auto-generated by Flutter/Android SDK. For development only.
- **Release keystore:** You create this. For production.

Android Signing

```
# Generate a release keystore
keytool -genkey -v -keystore upload-keystore.jks \
  -keyalg RSA -keysize 2048 -validity 10000 \
  -alias upload

# Create key.properties
cat > android/key.properties << 'EOF'
storePassword=your_password
keyPassword=your_key_password
keyAlias=upload
storeFile=/path/to/upload-keystore.jks
EOF
```

> **Critical:** Back up your keystore! If you lose it, you **cannot update** your app on Google Play.

Android Build — APK vs AAB

APK (Android Package)

- Complete app ready to install.
- Larger size (includes all densities).
- Good for sideloading and testing.

AAB (Android App Bundle)

- Google Play's required format.
- Smaller download (optimized per device).
- Google generates split APKs from AAB.

```
# Build APK (for testing/distribution)
flutter build apk --release \
  --obfuscate \
  --split-debug-info=./symbols

# Build AAB (for Google Play)
flutter build appbundle --release \
  --obfuscate \
  --split-debug-info=./symbols

# Check size
ls -lh build/app/outputs/
  flutter-apk/app-release.apk
  bundle/release/app-release.aab
```

iOS Signing & Provisioning

iOS has a more complex signing process than Android.

Key Components:

- **Apple Developer Account:** $99/year (required for App Store).
- **Certificates:** Developer (dev) + Distribution (release).
- **Provisioning Profiles:** Link cert + device + app ID.
- **App ID:** Unique identifier (e.g., `edu.sharif.mp`).

iOS Signing & Provisioning

Steps:

1. Create an **App ID** in Apple Developer Portal.
2. Create a **Distribution Certificate**.
3. Create a **Provisioning Profile** (App Store).
4. Download and install in Xcode.
5. Configure signing in Xcode → Signing & Capabilities.

```
# Build iOS release
flutter build ios --release --obfuscate \
  --split-debug-info=./symbols

# Archive and upload via Xcode
# Xcode → Product → Archive → Distribute App
```

Google Play Store Publishing

Steps to publish on Google Play:

1. **Create a Developer Account:** $25 one-time fee.
2. **Create an App:** Fill in store listing (title, description, screenshots).
3. **Upload AAB:** `flutter build appbundle --release`.
4. **Content Rating:** Complete the questionnaire.
5. **Pricing & Distribution:** Free/paid, target countries.
6. **Review:** Google reviews (usually 1–3 days for new apps).
7. **Publish:** App goes live after approval.

Google Play Store Publishing

Key Requirements:

- Target API level 33+ (Android 13) for new apps.
- Privacy policy URL (if app collects data).
- Data safety section (declare what data you collect).
- App content rating.

Apple App Store Publishing

Steps to publish on the App Store:

1. **Apple Developer Account:** $99/year.
2. **App Store Connect:** Create app record.
3. **Configure:** Bundle ID, version, screenshots, description.
4. **Build & Archive:** `flutter build ipa --release`.
5. **Upload:** Via Xcode or Transporter app.
6. **TestFlight:** Beta testing (optional but recommended).
7. **Submit for Review:** Apple reviews (1–7 days).
8. **Release:** Manual or automatic after approval.

Apple App Store Publishing

Key Requirements:

- Must follow Apple's **Human Interface Guidelines**.
- Privacy policy required.
- App must be **useful** and **not crash**.
- No private API usage.
- Explain why you need each permission.

Flutter Build Commands

```
# Android
flutter build apk --release                    # APK
flutter build apk --split-per-abi              # Smaller APKs per CPU
flutter build appbundle --release              # AAB (Google Play)

# iOS
flutter build ios --release                    # Build for iOS
flutter build ipa --release                    # Build IPA for upload

# Web
flutter build web --release                    # Web app

# All with obfuscation
flutter build apk --release --obfuscate \
  --split-debug-info=./symbols

# Build with flavors
flutter build apk --release --flavor production \
  --dart-define=ENV=prod \
  --dart-define=API_KEY=xxx
```

App Icon

```yaml
# Use flutter_launcher_icons package
# pubspec.yaml
dev_dependencies:
  flutter_launcher_icons: ^0.13.0

flutter_launcher_icons:
  android: true
  ios: true
  image_path: "assets/icon/app_icon.png"
  adaptive_icon_background: "#1966AB"
  adaptive_icon_foreground: "assets/icon/app_icon_fg.png"

# Generate
dart run flutter_launcher_icons
```

Splash Screens

```
# Use flutter_native_splash package
# pubspec.yaml
dev_dependencies:
  flutter_native_splash: ^2.3.0

flutter_native_splash:
  color: "#1966AB"
  image: assets/icon/splash.png
  android_12:
    image: assets/icon/splash_android12.png

# Generate
dart run flutter_native_splash:create
```

Versioning & Release Management

Semantic Versioning: `MAJOR.MINOR.PATCH+BUILD`

```
# pubspec.yaml
version: 1.2.3+45
#        | | |  └── Build number (must increment for each upload)
#        | | └──── Patch (bug fixes)
#        | └────── Minor (new features, backward compatible)
#        └──────── Major (breaking changes)
```

Android: `versionCode` (+45) and `versionName` (1.2.3) in `build.gradle`.
iOS: `CFBundleVersion` (45) and `CFBundleShortVersionString` (1.2.3) in `Info.plist`.

> **Rule:** Every upload to Google Play or App Store must have a **higher build number** than the previous one.

Fastlane: Automated Deployment

Fastlane automates screenshots, beta deployment, and release.

```
# Install
gem install fastlane

# Initialize (per platform)
cd android && fastlane init
cd ios && fastlane init
```

Fastlane: Automated Deployment

```
# android/fastlane/Fastfile
default_platform(:android)
platform :android do
  desc "Deploy to Play Store"
  lane :release do
    sh("flutter build appbundle --release --obfuscate")
    upload_to_play_store(
      track: 'production',
      aab: '../build/app/outputs/bundle/release/app-release.aab',
    )
  end
end

# ios/fastlane/Fastfile
platform :ios do
  desc "Deploy to App Store"
  lane :release do
    sh("flutter build ipa --release --obfuscate")
    upload_to_app_store(
      ipa: "../build/ios/ipa/myapp.ipa",
    )
  end
end
```

Post-Release: Monitoring

Crash Reporting:

- **Firebase Crashlytics:** Real-time crash reports, stack traces.
- **Sentry:** Cross-platform error tracking.

Analytics:

- **Firebase Analytics:** User behavior, events, funnels.
- **Mixpanel / Amplitude:** Advanced product analytics.

Performance Monitoring:

- **Firebase Performance:** Network traces, screen rendering.
- **Flutter DevTools:** CPU profiler, memory, network inspector.

Post-Release: Updates

Update Strategy:

- Monitor crash-free rate (target: **99.5%+**).
- Respond to user reviews.
- Release hotfixes for critical bugs.
- Regular updates for OS compatibility.

Summary: Key Takeaways

Build & Sign

- Three modes: Debug, Profile, Release.
- Android: Keystore + APK/AAB.
- iOS: Certificates + Provisioning.
- Always obfuscate release builds.

Publishing

- Google Play: $25 one-time, AAB format.
- App Store: $99/year, IPA format.
- Both require privacy policy.
- Plan for review time (1–7 days).

Automation

- Fastlane for deployment.
- GitHub Actions for CI/CD.
- `flutter_launcher_icons` for icons.
- `flutter_native_splash` for splash.

Post-Release

- Crashlytics for crash reporting.
- Analytics for user behavior.
- Monitor crash-free rate.
- Regular updates for OS changes.

Thank You! 🙌

Questions?

> *"It works on my machine.*
> *— Every developer before their first App Store rejection."*