

Crossword+ / KROSS+ - Documentation for Buyer

Source code + crossword/scanword packs | .NET 9 Android (MvvmCross)

Not a Play Console account transfer. Not a published listing sale.

1. Build environment (what you need)

This is NOT a classic Android Studio + Gradle Java/Kotlin project. It is a .NET 9 Android app built with MSBuild / Visual Studio / the dotnet CLI. Android SDK is still required for packaging.

- OS: Windows 10/11 (recommended)
- .NET SDK: 9.0.313+ (see global.json; rollForward: latestMinor)
- IDE: Visual Studio 2022 17.12+ with .NET/Android mobile workload, or VS Code + .NET Android tooling
- Android SDK: Platform API 35 (compile/target), Min SDK 21; Build-Tools 35.0.0+ (16 KB page size)
- JDK: 17 (bundled with Visual Studio / Android tooling is fine)
- Gradle: not used as the main build system (no root app/build.gradle)
- Emulator/device: API 21+; for Play upload tests prefer API 34/35 arm64

Quick build:

- Debug: dotnet build Crossword.sln -c Debug
- Install: dotnet build App\App.csproj -c Debug -t:Install
- Release AAB: dotnet publish App\App.csproj -c Release -f net9.0-android
- Output AAB: typically under App\bin\Release\net9.0-android\ (*-Signed.aab / publish folder)

2. Project structure

- Crossword.sln - solution entry
- App\ - Android UI, Activities/Fragments, Resources, Manifest, signing (App.csproj, package id ua.lyv.blitzcrossword)
- Core\ - business logic, ViewModels, HTTP/ZIP download, puzzle parsers (PuzzlesService), billing logic
- Bindings\CrosswordLib, Bindings\LuckyWheel - Android library bindings (AAR)
- App\ThirdParty\CrossPuzzle.dll - third-party puzzle helper
- App\Keys\ - release keystore path referenced by App.csproj (replace with YOUR keystore)
- global.json - pins .NET SDK version
- Crossword/scanword CONTENT is NOT embedded as Assets JSON. Packs are password-protected ZIP files downloaded from the content server (URL base in Core\Constants\StringConstants.cs -> NewUrl60000).

3. Crossword / scanword format and how to add new packs

Each pack is a ZIP (password-protected). App downloads a URL like: https://<your-host>/.../01_kr-leg/kr-leg_001.zip (path templates in Core\API\Implementations\PuzzlesService.ConvertPuzzleTypeToName).

- ZIP password (hardcoded): see HttpService constant keyArchive - change it if you re-pack archives.
- Inside ZIP: a text file. Lines separated by CR/LF (rebuses use \$).
- Classic / German / Estonian style: first line = puzzle title; then lines H x y ANSWER question... (horizontal) or V x y ANSWER question... (vertical). Coordinates in the file are 1-based.
- Scanwords: H/V x y ARROW ANSWER row-col question... (arrow + answer cell).
- Honeycombs (sot-4/6/8): title line, then coordinate / clue lines (see ParseCrossword court branches).
- Fillwords: title; then ANSWER r-c r-c ... question style lines.
- Numbering: most packs use 3 digits (001), some types use 2 digits (01) - see GetCrosswordFromNewServer.

To add new packs:

- 1) Prepare .txt in the format above -> ZIP with the same password -> name like kr-leg_002.zip.
- 2) Upload into the matching folder on your CDN/host mirroring ConvertPuzzleTypeToName.
- 3) Point StringConstants.NewUrl60000 (and prize URL in FactService if used) to YOUR server.
- 4) Optionally ship packs as a separate archive for the buyer to host.

4. Ads (AdMob) and In-App Purchases - keys to replace

Ads: Google AdMob only (no Appodeal in this .NET port).

- App ID (Manifest): App\Properties\AndroidManifest.xml -> com.google.android.gms.ads.APPLICATION_ID
- Banner: App\Infrastructure\Implementation\AdbForBannerAdmob.cs -> _bannerKey
- Rewarded / Interstitial: App\Infrastructure\Implementation\AdbGeneralAdmob.cs -> RewardedKey, InterstitialKey

- Replace ALL ca-app-pub-... values with your AdMob App ID and Ad Unit IDs.
- Purchases: Plugin.InAppBilling -> Google Play Billing.
- Product IDs in Core\Infrastructure\PaymentService.cs:
- Disable ads (~6 months): ua.lyv.blitzcrossword.adv_disable2
- Disable ads + endless hints: ua.lyv.blitzcrossword.adv_disable_hint_endless
- Hints packs: ua.lyv.blitzcrossword.hints_{N}
- Create the same product IDs (or rename code + Play Console products) under YOUR Play app.
- Also replace ApplicationId / package name in App.csproj + Manifest if you publish as a new app.
- App Center (analytics/crashes): replace with your App Center app key if you use it.

5. Sign and release build

- Release signing is configured in App\App.csproj (AndroidKeyStore, AndroidSigningKeyStore, alias, passwords).
- IMPORTANT: do NOT reuse the seller's production keystore for a new Play listing. Generate your own .keystore / upload key and update the AndroidSigning* properties.
- Release format: AAB (AndroidPackageFormat=aab), Target SDK 35, RIDs android-arm64;android-arm, 16 KB page alignment enabled.
- Publish command: dotnet publish App\App.csproj -c Release -f net9.0-android
- Upload the signed .aab to Google Play Console -> Testing / Production.
- Bump ApplicationVersion (versionCode) and ApplicationDisplayVersion before each Play upload.

6. If moving the app to another Play Console account

This listing sells SOURCE CODE + CONTENT, not the existing Google Play application or developer account. To publish under a buyer's account:

- Create a new app in the buyer's Play Console (official Play app transfer is a separate Google process and is NOT included by default).
- Change ApplicationId (package name) unless you legally transfer the existing listing.
- New signing key (upload key) -> register SHA-1/SHA-256 in Play Console / App signing.
- New AdMob app + ad units; update Manifest and AdMob classes.
- Recreate all IAP products with matching IDs; test with license testers.
- Host crossword ZIPs on buyer infrastructure; update NewUrl60000 / prize URLs.
- Privacy Policy / Data safety / Ads declaration / Content rating - fill for the new developer.
- Remove or replace seller branding, store listing assets, and any leftover keys before public release.

Stack summary: .NET 9.0 / MvvmCross 9.3.1 / Android API 21-35 / AdMob (Play Services Ads) / Plugin.InAppBilling 8.x / Content: remote password-protected ZIP packs. Document for SellMyCode buyers.