



DiscDex

Music Collection Manager for iPhone & iPad

Developer README

Architecture & Technical Reference

A native iOS app for cataloguing physical music collections. Scan barcodes to look up CD and vinyl releases on Discogs, track loans and condition, manage storage slots, and sync across devices via CloudKit.

Features

- **Barcode scanning** — Camera-based scanning plus NFC RFID tag support
- **Discogs integration** — Automatic metadata lookup (title, artists, labels, formats, tracklist, cover art, genres, styles, country, year) via the Discogs REST API
- **Dual collection** — Separate stores for CDs and vinyl records; unified library view with media filter
- **Inventory tracking** — Media and sleeve condition grading, physical location, storage slot assignment
- **Loan management** — Check items out to borrowers with optional due dates; overdue loan detection
- **Play tracking** — Per-release play count and last-played date
- **Smart Collections** — Dynamic filter presets: Recently Added, Wishlist, Favorites, Missing Location, Needs Cover, Overdue Loans, and more
- **Statistics dashboard** — Breakdowns by artist, label, genre, style, decade, condition, country, storage, and loans
- **Advanced search** — Query across multiple domains: title/artist, metadata, tags, tracklist, notes
- **Batch operations** — Bulk edit, bulk delete, bulk Discogs recheck for multiple selected releases
- **Barcode generation** — Create printable pseudo-barcode for items without a barcode
- **CSV import/export** — Portable catalog backup and sharing
- **CloudKit sync** — Snapshot-based iCloud backup and restore
- **Widgets** — Home screen widgets showing collection stats and last-scanned release
- **Custom attachments** — Attach photos and text notes to individual releases
- **Theming** — Customizable card colors, accent colors, and light/dark appearance

Architecture

Pure SwiftUI + UIKit (scanner/NFC) app with no third-party dependencies. Data layer uses SQLite3 via the native C API. CloudKit sync uses the native CloudKit framework.

```
CDScanner/  
■■■■ App/           CDScannerApp, ContentView, AppEnvironment, AppTab  
■■■■ Models/       CDRelease, InventoryItem, VinylCollectionItem,  
■■           Track, Artist, MusicLabel, ReleaseFormat,  
■■           ReleaseIdentifier, ReleaseImage, ReleaseAttachment  
■■■■ Managers/    LibraryViewModel, VinylCollectionViewModel,  
■■           ScannerViewModel, SettingsStore  
■■■■ Views/  
■■   ■■■■ Scanner/ ScannerView, BarcodeCameraView, ReleaseMatchCard  
■■   ■■■■ Library/  LibraryListView, ReleaseRowView  
■■   ■■■■ Release/  ReleaseDetailView  
■■   ■■■■ Vinyl/    VinylCollectionView  
■■   ■■■■ Settings/ SettingsView  
■■   ■■■■ Shared/   CoverArtworkView, StatisticsDashboardView
```

```

■■■ Sheets/      AddReleaseSheet, EditReleaseSheet,
■              CameraImagePicker, CoverCropView
■■■ Services/
■ ■■■ Database/ LibraryStore (actor), VinylCollectionStore,
■              SQLiteDatabase, DatabaseSchema
■ ■■■ Discogs/   DiscogsService, DiscogsModels, DiscogsMapper
■ ■■■ CloudKit/ CloudKitSyncService
■■■ Utilities/  CoverImageProcessor, CoverImageCleanup,
              AppNotifications, DateFormatterProvider

```

Data Storage

- **CD releases** — SQLite3 table; primary key is `id` (UUID); `discogsID` links to Discogs; `barcode` used for scan lookup
- **Vinyl releases** — Separate SQLite3 table; same structure adapted for vinyl
- **Inventory** — One-to-one table keyed by `releaseID`; stores condition, location, loan state, play count
- **Scan history** — SQLite3 table; records every scan event with result and timestamp
- **Smart Collections** — Computed dynamically from live database queries
- **Settings** — `UserDefaults` via `@AppStorage`
- **Widget data** — Shared via App Group

CDRelease Model

Each CD or release is stored as a `CDRelease` with the following key fields:

Field	Type	Description
<code>id</code>	UUID	Internal primary key
<code>discogsID</code>	Int	Discogs release identifier
<code>barcode</code>	String	Scanned or assigned barcode
<code>title</code>	String	Release title
<code>artists</code>	[Artist]	Contributing artists with roles
<code>labels</code>	[MusicLabel]	Labels with catalog numbers
<code>formats</code>	[ReleaseFormat]	Format descriptions (CD, LP, etc.)
<code>tracks</code>	[Track]	Tracklist with positions and durations
<code>genres / styles</code>	[String]	Genre and style tags
<code>country</code>	String	Release country
<code>year</code>	Int	Release year
<code>status</code>	LibraryStatus	<code>.owned</code> or <code>.wishlist</code>

Field	Type	Description
inventory	InventoryItem?	Linked collection metadata
attachments	[ReleaseAttachment]	User photos and text notes

InventoryItem Model

Each owned release optionally has one `InventoryItem` with:

Field	Description
mediaCondition	Condition grade (Mint → Poor)
sleeveCondition	Sleeve condition grade
location	Physical shelf or room label
storageSlot	Slot number in a carousel or cartridge
cartridgeID	Cartridge identifier for multi-cartridge changers
loanedTo	Borrower name when checked out
loanedAt / loanDueAt	Loan timestamps
playCount	Number of times played
lastPlayedAt	Date of most recent play
isFavorite	Starred/favourited flag

Discogs Integration

`DiscogsService` queries the Discogs REST API using a user-supplied personal access token:

Method	Purpose
<code>searchBarcode(_:)</code>	Look up a scanned barcode
<code>firstRelease(barcode:)</code>	Return the best CD match for a barcode
<code>firstVinylRelease(barcode:)</code>	Return the best vinyl match for a barcode
<code>searchReleases(title:artist:)</code>	Manual title/artist search
<code>release(id:)</code>	Fetch full release details by Discogs ID

Errors: `missingToken`, `invalidToken`, `connectionFailed`, `invalidResponse`, `notFound`.

No API key is embedded in the binary — users supply their own Discogs personal access token in Settings.

CloudKit Sync

`CloudKitSyncService` provides snapshot-level iCloud sync:

- **Upload** — Packages the SQLite database(s) as a CloudKit asset and stores it in the app's private database
- **Restore** — Downloads the latest snapshot and replaces the local database
- **Auto-sync** — Fires automatically after any library mutation when iCloud sync is enabled
- **Status** — Provides real-time status: `checking`, `available`, `noAccount`, `restricted`, `unavailable`

Smart Collections

Collection	Filter Logic
All	Every release
Recently Added	Releases added in the last 30 days
Wishlist	<code>status == .wishlist</code>
Favorites	<code>isFavorite == true</code>
Missing Location	Inventory with no location set
Needs Cover	No cover image URL
No Barcode	Empty or pseudo-generated barcode
Loaned Out	Currently loaned to someone
Overdue Loans	<code>loanDueAt</code> is in the past

Settings

Key	Type	Description
<code>discogsToken</code>	String	Personal access token for Discogs API
<code>barcodeScanTarget</code>	Picker	CD or Vinyl — controls which store receives a scanned result
<code>colorSchemePreference</code>	Picker	System / Light / Dark
<code>cardBackgroundColor</code>	Color	Background color of release cards

Key	Type	Description
<code>cardTextBrightness</code>	Double	Card text color (0 = black, 1 = white)
<code>buttonAccentColor</code>	Color	App-wide accent color for buttons and highlights
<code>libraryDisplayMode</code>	Picker	List or grid
<code>librarySortField</code>	Picker	Title / Artist / Year / Date Added
<code>librarySortDirection</code>	Picker	Ascending / Descending
<code>showCoversInListView</code>	Bool	Thumbnail covers in list mode
<code>searchField</code>	Picker	Default search domain
<code>batchScanMode</code>	Bool	Allow multiple scans without navigating away
<code>isHapticFeedbackEnabled</code>	Bool	Vibration on successful scan
<code>isAudioBeepEnabled</code>	Bool	Sound on successful scan
<code>cdStorageTrackingMode</code>	Picker	Off / Carousel / Cartridge
<code>cdStorageCapacity</code>	Int	Capacity of the storage unit
<code>isCloudSyncEnabled</code>	Bool	Enable iCloud CloudKit sync
<code>autoBackupOnLaunch</code>	Bool	Auto-backup on app launch
<code>includeCoversInCSVExport</code>	Bool	Embed cover art in CSV as Base64

Requirements

- iOS 16+
- Xcode 15+
- Discogs account + personal access token (required for metadata lookup)
- iCloud account (optional, for sync)
- NFC-capable device (optional, for RFID tag read/write)

DiscDex · GQLabs · 2026 · All rights reserved.