Programming Handheld Systems

Data Management

Today's Topics

SharedPreferences

Internal Storage

External Storage

SQLite databases

Shared preferences

Small amounts of primitive data

Internal Storage

Small to medium amounts of private data

External Storage

Larger amounts of non-private data

Databases

Store small to large amounts of private, structured data

SharedPreferences

A persistent map

Holds key-value pairs of simple data types

Automatically persisted across application sessions

SharedPreferences

Often used for long-term storage of customizable application data

Account name

Favorite WiFi networks

User customizations

Activity SharedPreferences

To get a SharedPreference object associated with a given Activity

Activity.getPreferences (int mode)

MODE_PRIVATE

Named SharedPreferences

```
Context.getSharedPreferences (
                    String name, int mode)
```

- NAME – NAME OF SharedPreferences FILE
- MODE – MODE_PRIVATE

Writing SharedPreferences

Call SharedPreferences.edit()

Returns a SharedPreferences.Editor instance

Writing SharedPreferences

Add values to SharedPreferences using SharedPreferences.Editor instance

putInt(String key, int value)

putString(String key, String value)

remove(String key)

Writing SharedPreferences

Commit edited values with
SharedPreferences.Editor.commit()

Reading SharedPreferences

Use SharedPreferences methods to read values

getAll()

getBoolean(String key, ...)

getString(String key, ...)

DataManagementSharedPreferences

When the user presses the play button, the application displays a random number

The application keeps track of the highest number seen so far


10:01
DataManagementSharedPreference
HighScore: 412
0
Play
Reset

Demonstration of the DataManagementSharedPreferences project in the IDE

PreferenceFragment

A Class that supports displaying & modifying user preferences

DataManagement PreferenceFragment

This application displays a PreferenceFragment, which allows the user to enter and change a persistent user name


10:21
DataManagementPreferenceFragment
User Name
Adam

Demonstration of the DataManagementPreferenceFragment project in the IDE

File

Class represents a file system entity identified by a pathname

File

Storage areas are classified as internal or external

Internal memory usually used for smaller, application private data sets

External memory usually used for larger, non-private data sets

File API

FileOutputStream

openFileOutput (String name, int mode)

Open private file for writing. Creates the file if it doesn't already exist

FileInputStream

openFileInput (String name)

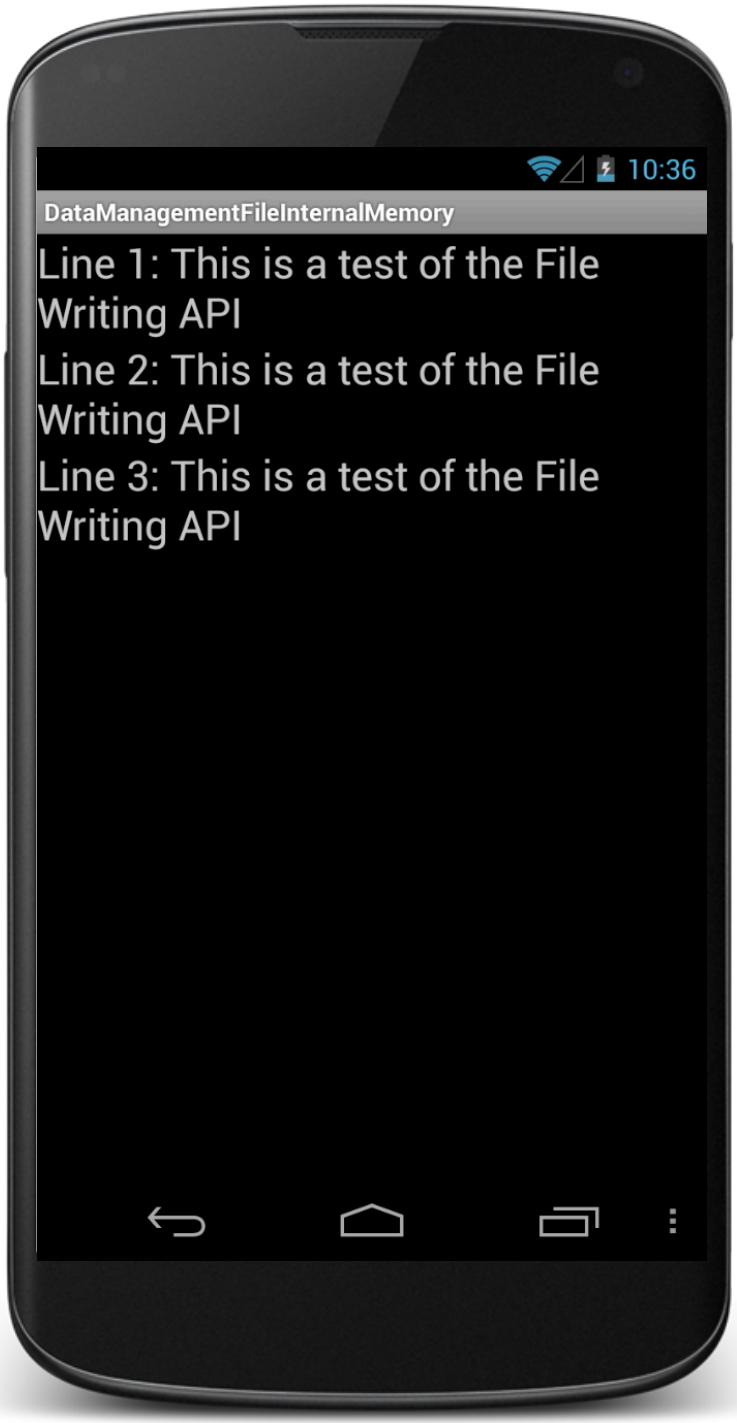
Open private file for reading

Many others. See documentation.

DataManagement
FileInternalMemory

If a text file does not already exist, application writes text to that text file

Application then reads data from the text file and displays it


10:36
DataManagementFileInternalMemory
Line 1: This is a test of the File Writing API
Line 2: This is a test of the File Writing API
Line 3: This is a test of the File Writing API

Demonstration of the DataManagementFileInternalMemory project in the IDE

Using External Memory Files

Removable media may appear/disappear without warning

Using External Memory Files

String Environment.
getExternalStorageState()

MEDIA_MOUNTED – present & mounted with read/write access

MEDIA_MOUNTED_READ_ONLY – present & mounted with read-only access

MEDIA_REMOVED – not present

Etc.

Using External Memory Files

Permission to write external files

```
<uses-permission android:name=
    “android.permission.
       WRITE_EXTERNAL_STORAGE" />
```

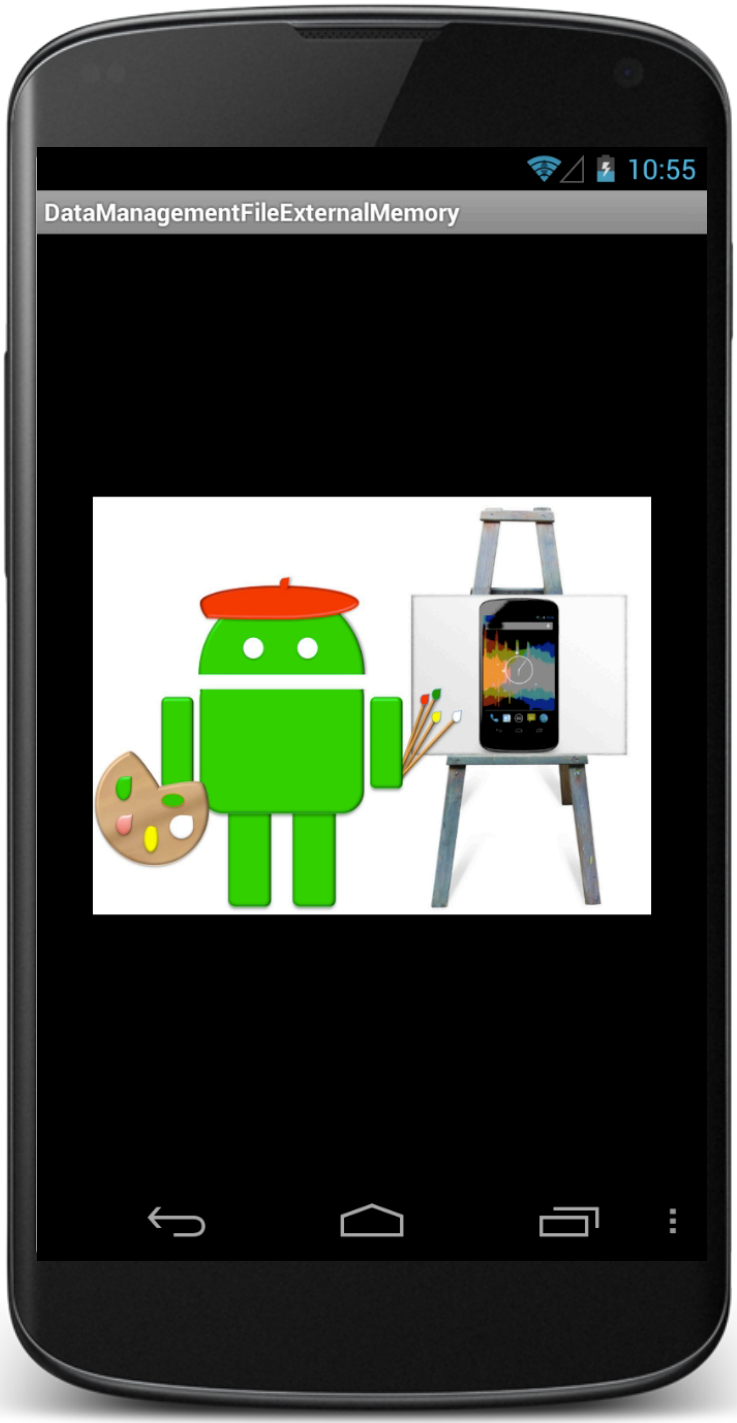
DataManagement
FileExternalMemory

Application reads an image file from the resources directory

Copies that file to external storage

Reads image data from the file in external storage

then displays the image


10:55
DataManagementFileExternalMemory

Demonstration of the DataManagementFileExternalMemory project in the IDE

Cache Files

Temporary files that may be deleted by the system when storage is low

Files removed when application uninstalled

Cache Files

File Context.getCacheDir()

Returns absolute path to an application-specific directory that can be used for temporary files

Saving cache files

Context.getExternalCacheDir()

returns a File representing external storage directory for cache files

SQLite

SQLite provides in-memory database

Designed to operate within a very small footprint (<300kB)

Implements most of SQL92

Supports ACID transactions

- Atomic, Consistent, Isolated & Durable

Using a Database

Recommended method relies on a helper class called SQLiteOpenHelper

Using a Database

Subclass SQLiteOpenHelper

Call super() from subclass constructor to initialize underlying database

Using a Database

Override onCreate()

Override onUpgrade()

Execute CREATE TABLE commands

Using a Database

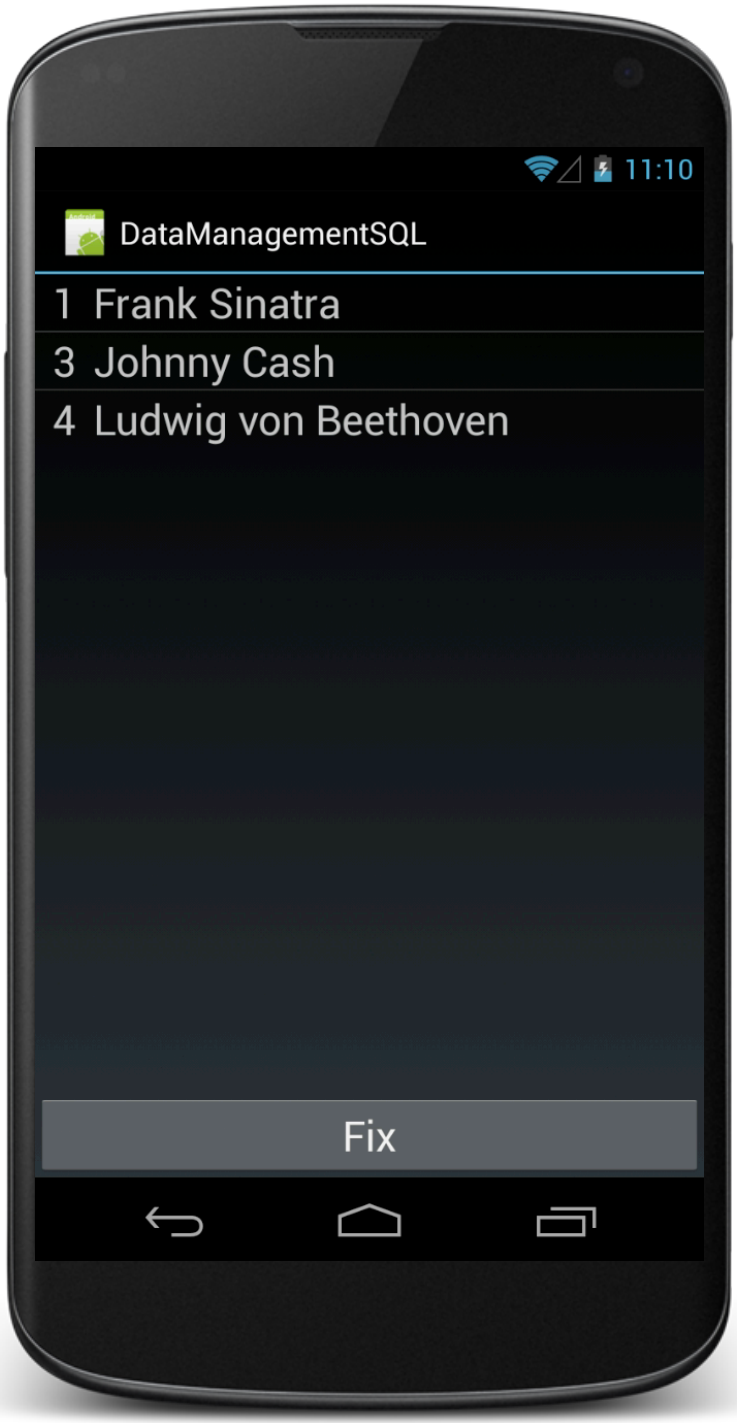
Use SQLiteOpenHelper methods to open & return underlying database

Execute operations on underlying database

DataManagementSQL

Application creates an SQLite database and inserts records, some with errors, into it

When user presses the Fix button, the application deletes, updates and redisplays the corrected database records


11:10
DataManagementSQL
1 Frank Sinatra
3 Johnny Cash
4 Ludwig von Beethoven
Fix

Demonstration of the DataManagementSQL project in the IDE

Examining the Database Remotely

Databases stored in

/data/data/<package name>/databases/

Can examine database with sqlite3

adb -s emulator-5554 shell

sqlite3 /data/data/
course.examples.DataManagement.Data
BaseExample/databases/artist_dbd

Next Time

ContentProvider