

, Interacting with database via internet/intranet., Graphics and animations- custom views, Canvas, Animation APIs, location awareness, , Native hardware access.

Working with mobile database (SQLite) :

History



- SQLite is an open source embedded database. The original implementation was designed by D. Richard Hipp.
- Hipp was designing software used on board guided missile systems and thus had limited resources to work with.
- The resulting design goals of SQLite were to allow the program to be operated without a database installation or administration.



Why Sqlite Only????



- In android we will use Sqlite database only . Because it is in built DB in Android SDK more over it is lite weighted relation DB suitable for Mobile Devices.
- We need not to load any drivers and we need not to install the Sqlite separately.
- The queries also simple to understand and easy to implement.

Feature of SQLite



- **Application file format** – Transactions guarantee ACID [Atomicity, Consistency , Isolation, Durability] even after system crashes and power failures.
- **Temporary data analysis** – Command line client, import CSV files and use sql to analyze & generate reports .
- **Embedded devices** – Applicable to small, reliable and portable like mobiles.
- **Portable** - uses only ANSI-standard C and VFS, file format is cross platform (little vs. big endian, 32 vs. 64 bit)

Feature of SQLite



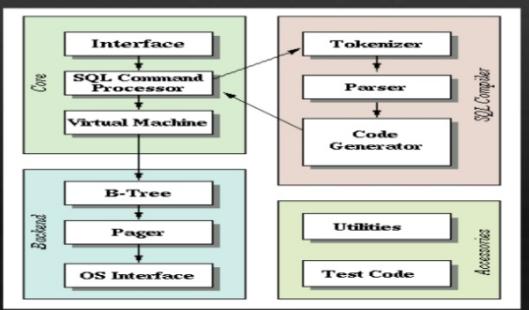
- **Reliable** – has 100% test coverage, open source code and bug database, transactions are ACID even if power fails.
- **Small** – 300 kb library, runs in 16kb stack and 100kb heap.
- **Single Database File** – An SQLite database is a single ordinary disk file that can be located anywhere in the directory hierarchy.
- **Readable source code** – The source code to SQLite is designed to be readable and accessible to the average programmer.

Disadvantages

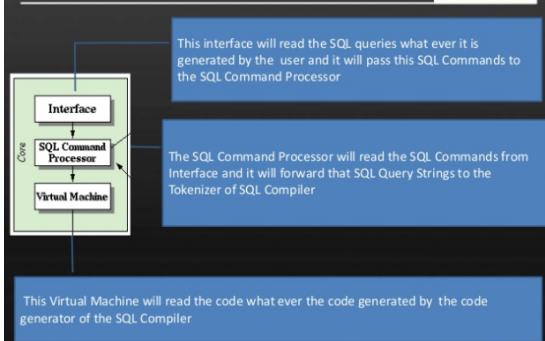


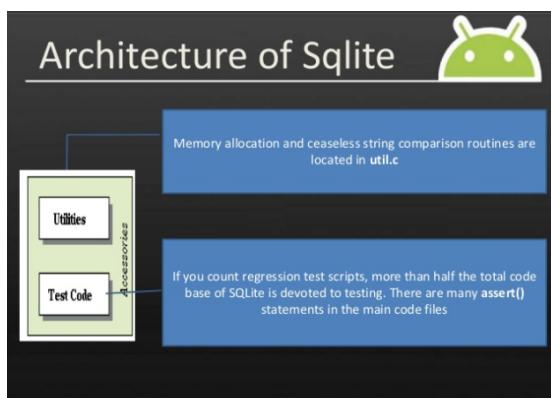
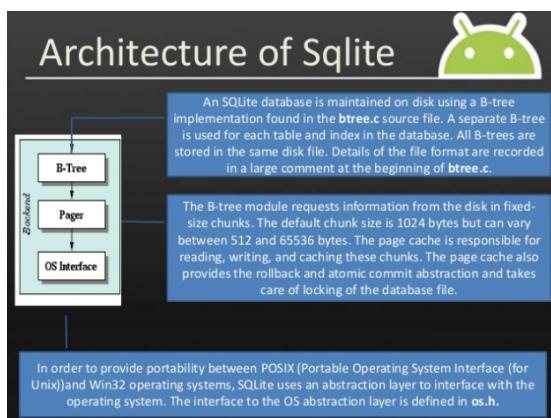
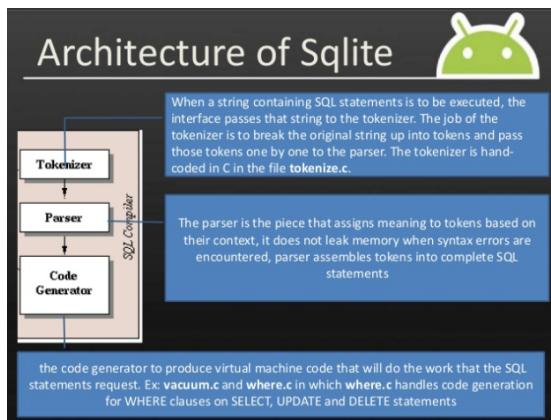
- **High concurrency** – reader/writer locks on the entire file.
- **Huge datasets** – DB file can't exceed file system limit or 2TB.
- **Access control** – we don't have any user interface to operate Sqlite database objects as in MYSQL / SQL Server /Oracle. All the objects are virtual. However there are few third party UI are available in the market.
[\[http://www.sqlabs.net/sqlitemanager.php\]](http://www.sqlabs.net/sqlitemanager.php)

Architecture of Sqlite



Architecture of Sqlite





Examples To Do



Let us do some coding

Creating Sqlite Object

Step:1 Importing package
"android.database.sqlite.SQLiteDatabase".
Step:2 Creating object
SQLiteDatabase object name=null;

Examples To Do



Let us do some coding

Creating Database Name

EX:mydb=openOrCreateDatabase("DatabaseName5",
 MODE_PRIVATE,null);
//mydb is sqlite object name .
//DatabaseName5 is nothing but database name
//MODE_PRIVATE is permissions of a table accessing

Examples To Do



Let us do some coding

Creating Table

mydb.execSQL("CREATE TABLE IF NOT EXISTS "
 +TableName+" (ColumnName DataType);");

Examples To Do



Let us do some coding

DDL(Create ,Alter ,Drop)

Create:
mydb.execSQL("CREATE TABLE IF NOT EXISTS "
 +TableName+" (ColumnName DataType);");

Alter:
ALTER TABLE TableName RENAME TO new-table-name

Drop:
DROP TABLE TableName

(View Source)

Examples To Do



Let us do some coding

DML(SELECT , INSERT ,
DELETE)

Select:

```
Cursor c=mydb.rawQuery("SELECT * FROM "+TableName+
where Name='"+city+"',null);
```

Insert:

```
mydb.execSQL("INSERT INTO "+TableName+" (Name, Area)" +
"VALUES ('RedFort','40.8 acres');");
```

Delete:

```
mydb.execSQL("Delete"+TableName);
```

[\(View Source\)](#)

SQLiteOpenHelper



- SQLiteOpenHelper is a class to manage database creation and version management.
- This class take care of opening the database if it exists, creating it if it does not, and upgrading it as necessary.
- This is for creating db “**onCreate(SQLiteDataBase)**”.
- when the database needs to be upgraded “**onUpgrade (SQLiteDatabase db, int oldVersion, int newVersion)**”.
- when the database has been opened “**onOpen (SQLiteDatabase db)**”.

Multimedia- audio/video playback and record:

Audio and Video

- The Android platform offers built-in encoding/decoding for a variety of common media types. You can easily integrate audio, video, and images into your applications.
- You can play audio or video from
 - media files stored in the application's (raw) resources
 - standalone files in the file system
 - a data stream arriving over a network connection.
- To play audio or video, use the `MediaPlayer` class.
 - Class `VideoView` simplifies embedding videos in an application.
- To record audio or video, use the `MediaRecorder` class.
(not available in emulators)

Supported Media Formats

Media Type	Format	File Type(s)
Audio	AAC/AAC+	.m4a, .3gp, .aac
	MPEG Audio	.mp3, .mp4,
	PCM/WAVE	.wav
	Ogg Vorbis	.ogg
	MIDI	.mid, .xmf, .mxmf
	AMR-NB/WB/AMR-WB	.3gp
Image	JPEG	.jpg
	GIF	.gif
	PNG	.png
	BMP	.bmp
Video	H.263	.3gp and .mp4
	H.264 AVC (Android 3.0+)	.3gp and .mp4
	MPEG-4 SP	.3gp
	VP8	WebM

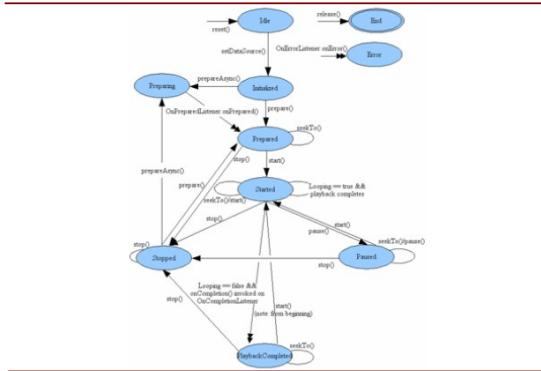
Selected Methods from Class `MediaPlayer`

- `MediaPlayer()`
 - Default constructor.
- `static MediaPlayer create(Context context, int resid)`
 - Creates a `MediaPlayer` for a given resource id.
- `static MediaPlayer create(Context context, Uri uri)`
 - Creates a `MediaPlayer` for a given Uri.
- `void pause()`
 - Pauses playback.
- `void release()`
 - Releases resources associated with this `MediaPlayer` object.

Selected Methods from Class MediaPlayer (continued)

- `void setDataSource(String path)`
 - Sets the data source (file-path or http/rtsp URL) to use.
- `void setDataSource(FileDescriptor fd)`
 - Sets the data source (FileDescriptor) to use.
- `void setDataSource(Context context, Uri uri)`
 - Sets the data source as a content Uri.
- `void setLooping(boolean looping)`
 - Sets the player to be looping or non-looping.
- `void start()`
 - Starts or resumes playback.
- `void stop()`
 - Stops playback after playback has been started or paused.

State Diagram for MediaPlayer



Example: Playing an Audio Resource File

- Create the directory (folder) `res/raw` if it does not already exist.
- Copy the audio file (e.g., `echotaps.m4a`) to the `res/raw` folder and refresh the folder.
- Create activity and declare a `MediaPlayer` object.

```
public class MainActivity extends ActionBarActivity
{
    private MediaPlayer mediaPlayer;

    ... // continued on next page
}
```

Example: Playing an Audio Resource File (in the `onCreate()` method)

```
setVolumeControlStream(AudioManager.STREAM_MUSIC);

Button playButton = (Button) findViewById(R.id.playButton);
playButton.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        if (mediaPlayer != null)
            mediaPlayer.release();

        mediaPlayer = MediaPlayer.create(MainActivity.this,
                                         R.raw.echotaps);
        mediaPlayer.start();
    }
});
```

↑
Note: No file extension.

Example: Playing an Audio Resource File (continued)

```
Button stopButton = (Button) findViewById(R.id.stopButton);
stopButton.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        if (mediaPlayer != null)
        {
            mediaPlayer.stop();
            mediaPlayer.release();
            mediaPlayer = null;
        }
    }
});
```

Example: Playing an Audio Resource File (continued)

```
@Override
public void onPause()
{
    super.onPause();

    if (mediaPlayer != null)
        mediaPlayer.pause();
}

@Override
public void onResume()
{
    super.onResume();

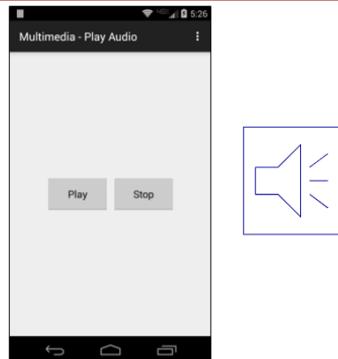
    if (mediaPlayer != null)
        mediaPlayer.start();
}
```

Example: Playing an Audio Resource File (continued)

```
@Override
public void onStop()
{
    super.onStop();

    if (mediaPlayer != null)
    {
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
    }
}
```

Example: Playing an Audio Resource File



Playing Audio from a File or Stream

- Create an instance of `MediaPlayer` using `new`.
- Call `setDataSource()` with a `String` containing the path (file system or URL) to the file you want to play.
- First `prepare()` then `start()` on the instance.

- Example

```
MediaPlayer mp = new MediaPlayer();
mp.setDataSource(PATH_TO_FILE);
mp.prepare();
mp.start();
```

Playing Audio from a File or Stream

(continued)

- Either `IllegalArgumentException` or `IOException` can be thrown by `setDataSource()` since the file may not exist.
- If the URL references an online media file, the file must be capable of progressive download.

Recording Audio Files: Permissions

- An application that records audio must set the appropriate permissions in the `AndroidManifest.xml` file.
`<uses-permission android:name="android.permission.RECORD_AUDIO" />`
- If the audio is recorded on external storage, an additional permission is required.
`<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />`

Recording Audio Files

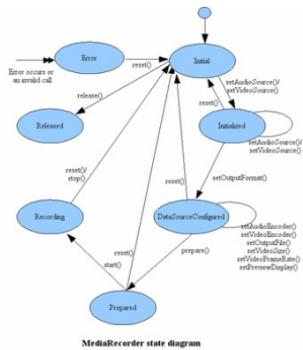
- Create an instance of `MediaRecorder`.
- Set the audio source using `MediaRecorder.setAudioSource()`.
 - usually `MediaRecorder.AudioSource.MIC`
 - other possible sources include `DEFAULT` (equivalent to `MIC`), `VOICE_CALL`, and `CAMCORDER`.
- Set output file format using `MediaRecorder.setOutputFormat()`.
 - e.g., to `MediaRecorder.OutputFormat.THREE_GPP`
 - other possible formats include `DEFAULT` (equivalent to `THREE_GPP`), `AMR_NB`, and `MPEG_4`.

Recording Audio Files

(continued)

- Set output file name using `MediaRecorder.setOutputFile()`.
- Set the audio encoder using `MediaRecorder.setAudioEncoder()`.
 - e.g., to `MediaRecorder.AudioEncoder.AMR_NB`
- Call `MediaRecorder.prepare()` on the `MediaRecorder` instance.
- To start audio capture, call `MediaRecorder.start()`.
- To stop audio capture, call `MediaRecorder.stop()`.
- Call `MediaRecorder.release()` when recording has been completed to free the resource immediately.

State Diagram for MediaRecorder



Example: Recording an Audio File

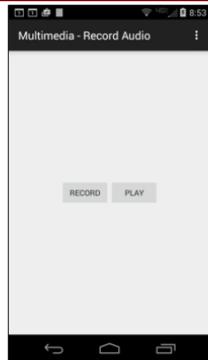
```
private MediaRecorder recorder = null;  
...  
// in onCreate()  
fileName =  
    Environment.getExternalStorageDirectory().getAbsolutePath();  
fileName += "/audiorecord.3gp";  
...  
private void stopRecording()  
{  
    recorder.stop();  
    recorder.release();  
    recorder = null;  
}
```

Example: Recording an Audio File

(continued)

```
private void startRecording()  
{  
    recorder = new MediaRecorder();  
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);  
    recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);  
    recorder.setOutputFile(fileName);  
    recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);  
  
    try  
    {  
        recorder.prepare();  
    }  
    catch (IOException e)  
    {  
        Log.e(LOG_TAG, "prepare() failed");  
    }  
  
    recorder.start();  
}
```

Example: Recording an Audio File (continued)



Example: Playing a Video File

- Copy video file to the res/raw directory or to external storage on the device.

- If copied to external storage, add permission to `AndroidManifest.xml`.

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

- Create `VideoView` in layout file.

```
<VideoView  
    android:id="@+id/video"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_gravity="center" />
```

Example: Playing a Video File (using res/raw directory)

```
// in onCreate() method  
VideoView videoView = (VideoView) findViewById(R.id.video);  
videoView.setMediaController(new MediaController(this));  
Uri uri = Uri.parse("android.resource://" + getPackageName()  
    + "/" + R.raw.winter_luck);  
videoView.setVideoURI(uri);  
videoView.start();
```

Example: Playing a Video File (using external storage)

```
// in constructor  
fileName =  
    Environment.getExternalStorageDirectory().getAbsolutePath();  
fileName += "/winter_luck.mp4";  
  
// in onCreate() method  
VideoView videoView = (VideoView) findViewById(R.id.video);  
videoView.setVideoPath(fileName);  
videoView.setMediaController(new MediaController(this));  
videoView.start();
```