

Part 1 - Introduction

When to use a Database

While the storage and processing capabilities of mobile devices are increasing, phones and tablets still lag behind their desktop & laptop counterparts. For this reason it is worth taking some time to plan the data storage architecture for your app rather than just assuming a database is the right answer all the time. There are a number of different options that suit different requirements, such as:

- **Preferences** – Both iOS and Android offer a built-in mechanism for storing simple key-value pairs of data. If you are storing simple user settings or small pieces of data (such as personalization information) then use the platform's native features for storing this type of information. For iOS you can also take advantage of iCloud synchronization for this data, both for backup and synchronization for users with multiple devices.
- **Text Files** – User input or caches of downloaded content (eg. HTML) can be stored directly on the file-system. Use an appropriate file-naming convention to help you organize the files and find data.
- **Serialized Data Files** – Objects can be persisted as XML or JSON on the file-system. The .NET framework includes libraries that make serializing and de-serializing objects easy. Use appropriate names to organize data files.
- **Database** – The SQLite database engine is available on both iOS and Android platforms, and is useful for storing structured data that you need to query, sort or otherwise manipulate. Database storage is suited to lists of data with many properties.
- **Image files** – Although it's possible to store binary data in the database on a mobile device, it is recommended that you store them directly in the file-system. If necessary you can store the filenames in a database to associate the image with other data. When dealing with large images, or lots of images, it is good practice to plan a caching strategy that deletes files you no longer need to avoid consuming all the user's storage space.

If a database is the right storage mechanism for your app, the remainder of this document discusses how to use SQLite in a cross-platform application on the Xamarin platform.

Advantages of using a Database

There are a number of advantages to using an SQL database in your mobile app:

- SQL databases allow efficient storage of structured data.
- Specific data can be extracted with complex queries.
- Query results can be sorted.
- Query results can be aggregated.
- Developers with existing database skills can utilize their knowledge to design the database and data access code.
- The data model from the server component of a connected application may be re-used (in whole or in part) in the mobile application.

SQLite Database Engine

SQLite is an open-source database engine that has been adopted by both Apple and Google for their mobile platforms. The SQLite database engine is built-in to both operating systems so there is no additional work for developers to take advantage of it. SQLite is well suited to cross-platform mobile development because:

- The database engine is small, fast and easily portable.
- A database is stored in a single file, which is easy to manage on mobile devices.
- The file format is easy to use across platforms: whether 32- or 64-bit, and big- or little-endian systems.
- It implements most of the SQL92 standard.

Because SQLite is designed to be small and fast, there are some caveats on its use:

- Some OUTER join syntax is not supported.
- Only table RENAME and ADDCOLUMN are supported. You cannot perform other modifications to your schema.
- Views are read-only.

You can learn more about SQLite on the website - SQLite.org - however all the information you need to use SQLite with Xamarin is contained in this document and associated samples. The SQLite database engine is built-in to all versions of iOS and was added to Android 2. Although not covered in this chapter, SQLite is also available for use on Windows Phone and Windows Store (8 & RT) applications. There is more information on the [Xamarin documentation](http://Xamarin.com/documentation) website.

Windows and Windows Phone

SQLite can also be used on Windows platforms, although those platforms are not covered in this document. The following Xamarin samples include Windows or Windows Phone applications that use SQLite – [Tasky Pro](#) and [MWC 2012](#) case studies and the [Field Service](#) pre-built app.

Read more in the [MWC 2012](#) and [Tasky Pro](#) case studies, and review [Tim Heuer's blog](#).