ListViews and Adapters

Sample Code:

- AccessoryViews.zip
- BasicTableAndroid.zip
- BasicTableAdapter.zip
- BuiltInViews.zip
- CustomRowView.zip
- FastScroll.zip
- SectionIndex.zip
- SimpleCursorTableAdapter.zip
- CursorTableAdapter.zip

Related Articles:

- Activity Lifecycle Tutorial
- Working with Tables and Cells (in Xamarin.iOS)

Related SDK:

- ListView Class Reference
- ListActivity Class Reference
- BaseAdapter Class Reference
- ArrayAdapter Class Reference
- CursorAdapter Class Reference

ListView is an important UI component of Android applications, used everywhere from short lists of menu options to long lists of contacts or internet favorites. It provides a simple way to present a scrolling list of rows that can either be formatted with a built-in style or customized extensively. A ListView instance requires an Adapter to feed it with data contained in row views. This document shows how to implement ListView and the different Adapter classes with Xamarin.Android. It also demonstrates how to customize the appearance of a ListView and discusses the importance of row re-use to reduce memory consumption. There is also some discussion of how the Activity Lifecycle affects ListView and Adapter use. For those working on cross-platform applications with Xamarin.iOS, the ListView control is structurally similar to the iOS UITableView (and the Android Adapter is similar to the UITableViewSource).

Overview

This article will take a comprehensive look at working with the ListView class and the different types of Adapter you can use with it. The discussion will begin with an overview of the ListView class itself before introducing progressively more complex examples of how to use it. The document structure is as follows:

- **Visual Appearance** – Parts of the ListView control and how they work.
- **Classes** – Overview of the classes used to display a ListView.
- **Displaying Data in a ListView** – How to display a simple list of data; how to implement ListView's usability features; how to use different built-in row layouts; and how Adapters save memory by

re-using row views.
- **Custom appearance** – Changing the style of the ListView with custom layouts, fonts and colors.
- **Using SQLite** – How to display data from a SQLite database with a CursorAdapter.
- **Activity Lifecycle** – Design considerations when implementing ListView Activities, including where in the lifecycle you should populate your data and when to release resources.

Each section is accompanied by a sample code download.

Sections