**SyscAdpter**

The sync adapter component in your app encapsulates the code for the tasks that transfer data between the device and a server. The sync adapter component does not automatically do data transfer. Instead, it encapsulates your data transfer code, so that the sync adapter framework can run the data transfer in the background, without involvement from your app.

**To add a sync adapter component to your app, you need to add the following pieces:**

1. **Sync adapter class:** A class that wraps your data transfer code in an interface compatible with the sync adapter framework.
2. **Bound Service:** A component that allows the sync adapter framework to run the code in your sync adapter class.
3. **Sync adapter XML metadata file:** A file containing information about your sync adapter. The framework reads this file to find out how to load and schedule your data transfer.
4. **Declarations in the app manifest:** XML that declares the bound service and points to sync adapter-specific metadata.
5. Create a SyncAdapter we have to extend the **AbstractThreadedSyncAdapter**
6. Use the **constructors** to run setup tasks each time your sync adapter component is created from scratch, just as you use **Activity.onCreate()** to set up an activity.
7. When the framework is ready to sync your application's data, it invokes your implementation of the method **onPerformSync().**   
     
   **Account**: An Account object associated with the event that triggered the sync adapter. If your server doesn't use accounts, you don't need to use the information in this object.  
     
   **Extras:** A Bundle containing flags sent by the event that triggered the sync adapter.  
     
   **Authority:** The authority of a content provider in the system. Your app has to have access to this provider. Usually, the authority corresponds to a content provider in your own app.  
     
   **Content provider client**: A ContentProviderClient for the content provider pointed to by the authority argument. A ContentProviderClient is a lightweight public interface to a content provider. It has the same basic functionality as a ContentResolver. If you're using a content provider to store data for your app, you can connect to the provider with this object. Otherwise, you can ignore it.  
     
   **Sync result:** A SyncResult object that you use to send information to the sync adapter framework.