

Content Providers

Content Providers expose an application's data across all other applications, we can use content providers to store or retrieve data of one application from any other application.

Content Providers are a generic interface mechanism that lets you share data between applications. By abstracting away the underlying data source, Content Providers let you decouple your application layer from the data layer, making your applications data-source agnostic.

Content Providers feature full permission control and are accessed using a simple URI model. Shared content can be queried for results as well as supporting write access. As a result, any application with the appropriate permissions can add, remove, and update data from any other applications — including some native Android databases.

Many of the native databases have been made available as Content Providers, accessible by third-party applications. This means that your applications can have access to the phone's Contact Manager, media player, and other native database once they've been granted permission.

By publishing your own data sources as Content Providers, you make it possible for you (and other developers) to incorporate and extend your data in new applications.

Android default content providers:

There are many built-in content providers supplied by OS. They are defined in the **android.provider** package, they include:

- Browser.
- Calllog.
- Live Folders.
- Contacts Contract.
- Media Store.
- Settings.

content providers basics: The concept of content providers is to provide data encapsulation through exposing data by **URIs**. Any content provider is invoked by a URI in the form of **content://provider_name** .

For example the URi of the Contacts content provider that retrieves all contacts is in the following form **content://contacts/people**. If you want to retrieve a particular contact (by its ID) then it would be in this form: **content://contacts/people/5**.

Note: the contact's content provider we are considering for this example is Android's default Contact List which is developed using Content Providers.

The URIs of the content providers are not written manually as they are stored as constant values in their respective content provider classes.

Following are some of the examples of Content Providers from Android's default applications:

The Uri of the Contacts phones content provider is defined in:

- ContactsContract.CommonDataKinds.Phone.CONTENT_URI
(<content://com.android.contacts/data/phones>)
- The Uri of the browser Bookmarks content provider is defined in Browser.BOOKMARKS_URI
(<content://browser/bookmarks>)
- The Media store (Video) stored in external device (SD Card) is defined in MediaStore.Video.Media.EXTERNAL_CONTENT_URI
(<content://media/external/video/media>) and so on.

Content Providers and Content Resolver:

Access to Content Providers is handled by the ContentResolver class.

Introducing Content Resolvers

Each application Context has a single ContentResolver, accessible using the getContentResolver method, as shown in the following code snippet:

```
ContentResolver cr = getContentResolver();
```

Content Resolver includes several methods to transact and query Content Providers. You specify the provider to interact using a URI.

A Content Provider's URI is defined by its *authority* as defined in its application manifest node. An authority URI is an arbitrary string, so most providers expose a CONTENT_URI property that includes its authority.

Content Providers usually expose two forms of URI, one for requests against all the data and another that specifies only a single row. The form for the latter appends /<rowID> to the standard CONTENT_URI.

Content Provider queries take a very similar form to database queries. Using the query method on the ContentResolver object, pass in:

- ☐ The URI of the content provider data you want to query
- ☐ A projection that represents the columns you want to include in the result set
- ☐ A where clause that defines the rows to be returned. You can include ? wild cards that will be replaced by the values stored in the selection argument parameter.
- ☐ An array of selection argument strings that will replace the ?'s in the where clause
- ☐ A string that describes the order of the returned rows

Content Providers and CRUD:

Content providers allow you to perform basic **CRUD** operations: Create, Read, Update and Delete on data.

Working of Content Providers:

To retrieve data from a content provider we run a sql-like query using query function. It returns a cursor holding the result set.

Steps to use Content Providers [Following steps are considered with reference to Contacts App]:

A) We first define the projection of our query, we define the columns we want to retrieve in the result set.

B) We specify the Uri of the content provider

C) We retrieve the data by a cursor.

The cursor is retrieved by executing a **query** which has the following parameters:

- The Uri of the content provider.
- A String Array of the columns to be retrieved (projection)
- Where clause.
- String array containing selection arguments values.

D) Create a list adapter using the cursor and bind the ListView to it.

Note: It is must to define the permissions in manifest file for respective Content Providers we are using.

For more details refer:

<http://developer.android.com/guide/topics/providers/content-providers.html>