ANDROID DEVELOPMENT

DATA PERSISTENCE

Tan Cher Wah (isstcw@nus.edu.sg)

© 2019 National University of Singapore. All Rights Reserved.

Agenda

- Methods of persistence
  - Shared Preferences for application
  - Activity Instance State
  - Internal Storage
    - Device memory
  - External Storage
    - Shared external storage

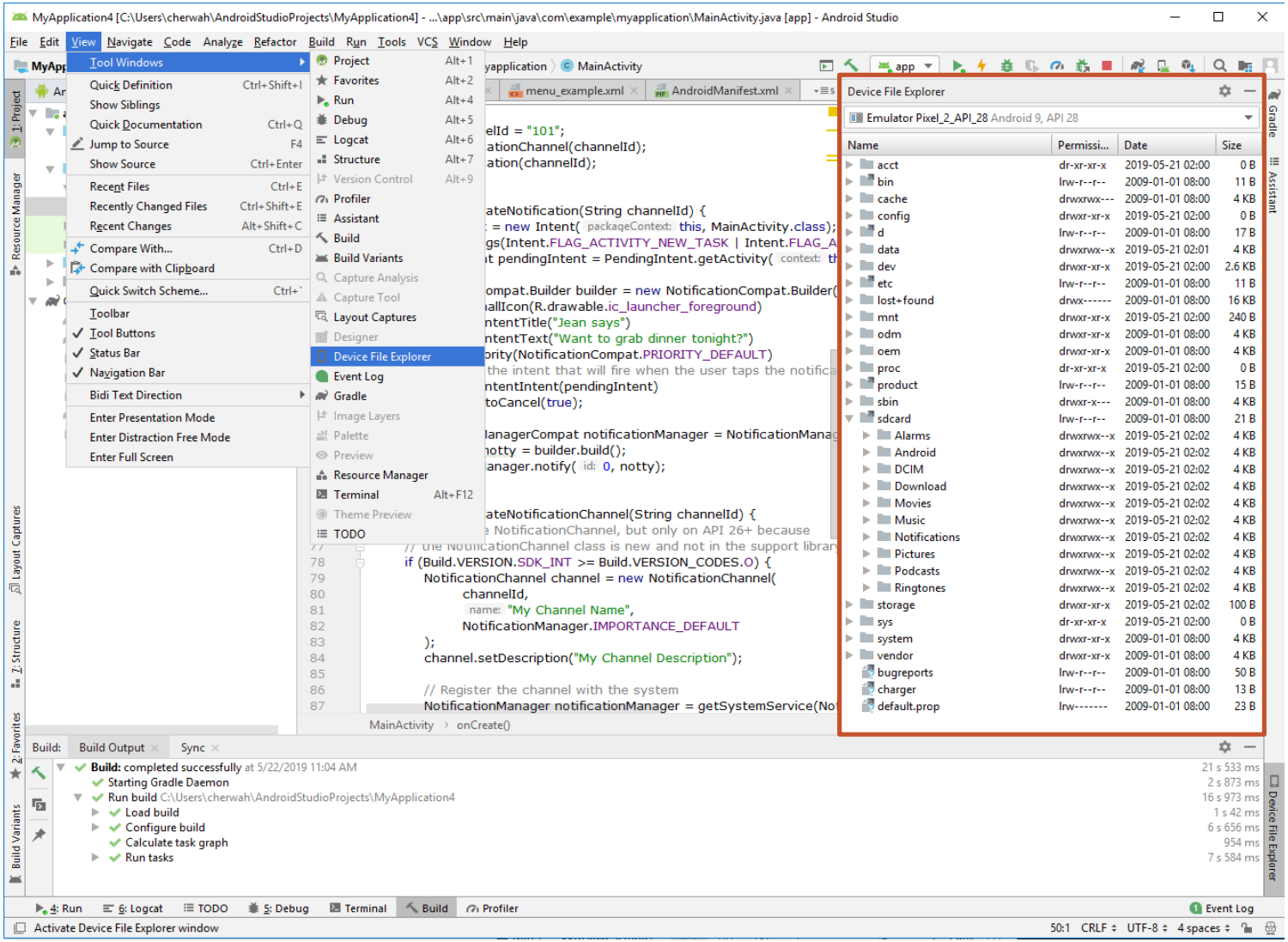
© 2019 National University of Singapore. All Rights Reserved.

Shared Preferences

- Shared Preferences allow for **key-value pairs** of primitive data types to be saved/restored
- They are persistent across **user sessions** and **system reboot**

© 2019 National University of Singapore. All Rights Reserved.

Emulator's Device File Explorer



© 2019 National University of Singapore. All Rights Reserved.

Reading SharedPreferences

- Default shared preference is read via **getDefaultSharedPreferences()**
- No need to provide a filename
- To get, use get**XXX**() where **XXX** is a **type**

```
SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());

String hostname = pref.getString("hostname", "localhost");
boolean toEcho = pref.getBoolean("echo", true);
```

© 2019 National University of Singapore. All Rights Reserved.

Editing SharedPreferences

- Shared Preference is modified with a **SharedPreferences.Editor** via a **commit() or apply()**
- commit() writes to hard-disk **synchronously**
- apply() saves to memory synchronously, then writes to hard-disk **asynchronously** (in the background)

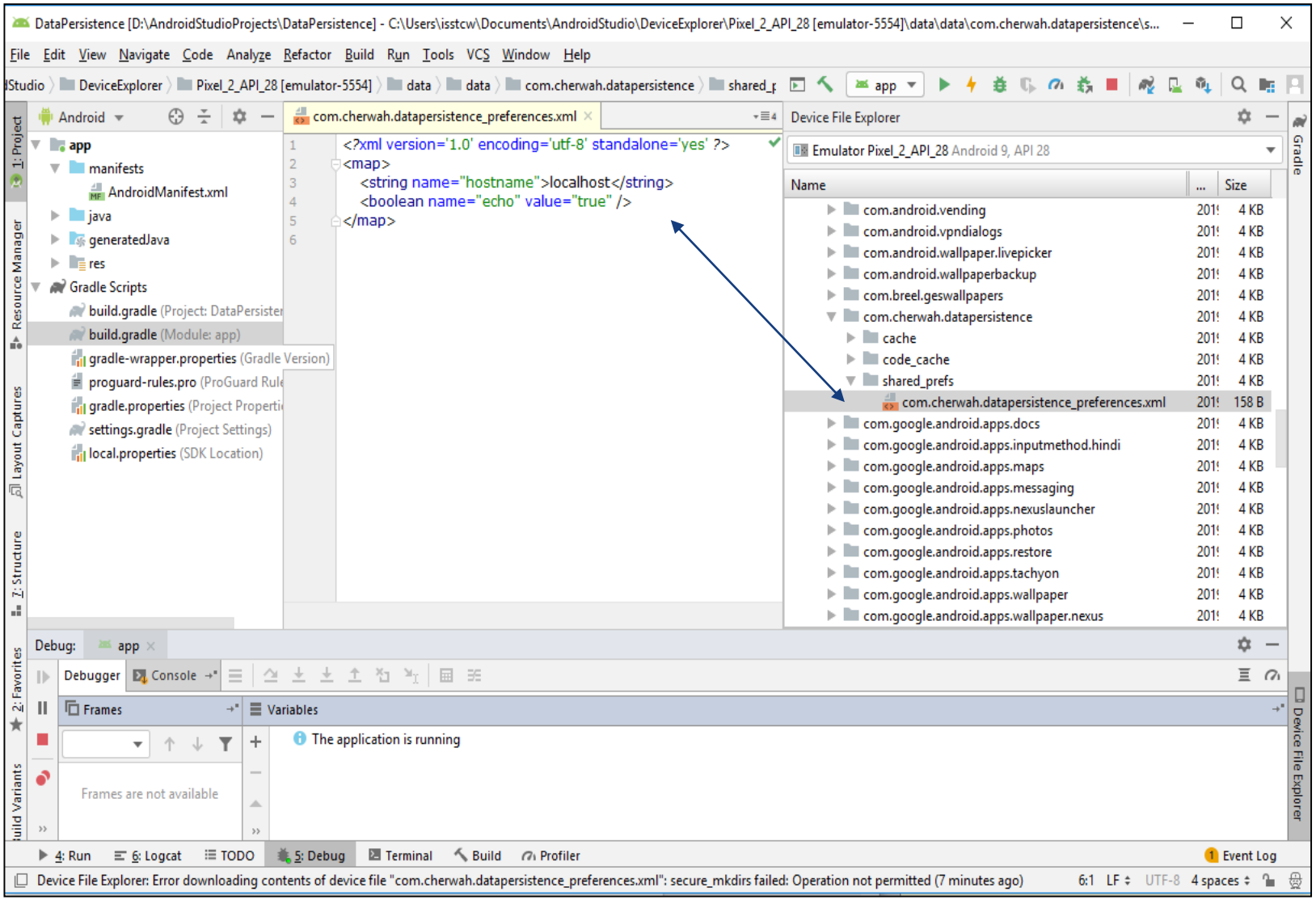
```
SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
SharedPreferences.Editor editor = pref.edit();

editor.putString("hostname", "localhost");
editor.putBoolean("echo", true);

// editor.commit();  // synchronous
editor.apply();  // asynchronous
```

© 2019 National University of Singapore. All Rights Reserved.

View saved preferences on Emulator



© 2019 National University of Singapore. All Rights Reserved.

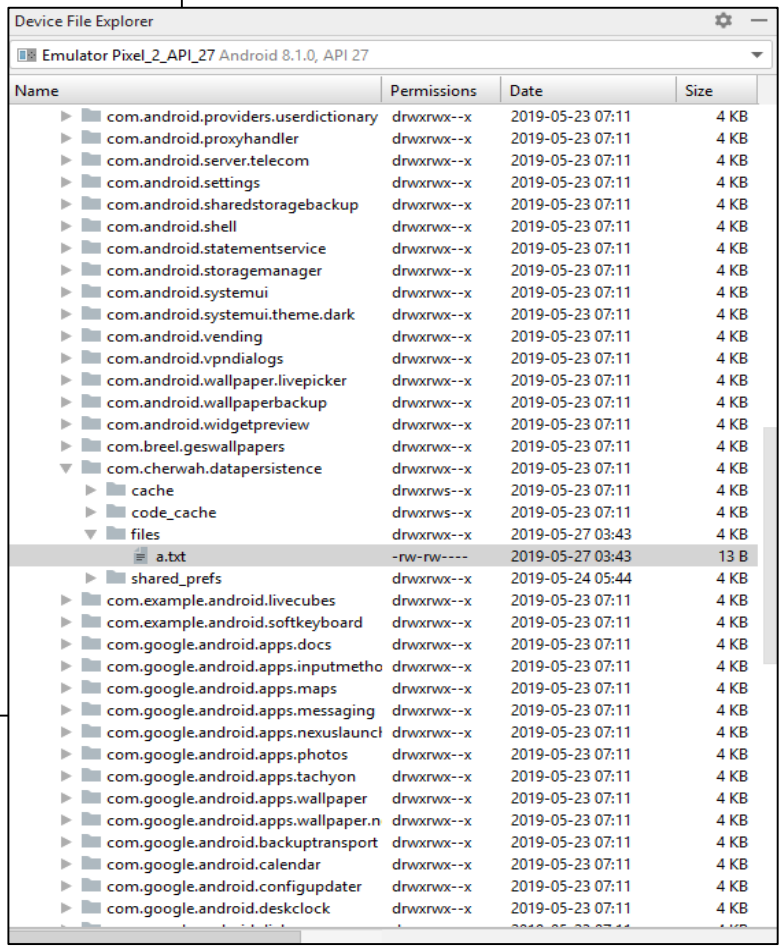
Writing Files

- MODE_PRIVATE - The created file can only be accessed by calling application

```
FileOutputStream outStream = null;
PrintWriter writer = null;

try {
    outStream = openFileOutput("a.txt", Context.MODE_PRIVATE);
    writer = new PrintWriter(outStream);
    writer.println("Android 101\n");
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (writer != null)
        writer.close();

    if (outStream != null)
    {
        try {
            outStream.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



© 2019 National University of Singapore. All Rights Reserved.

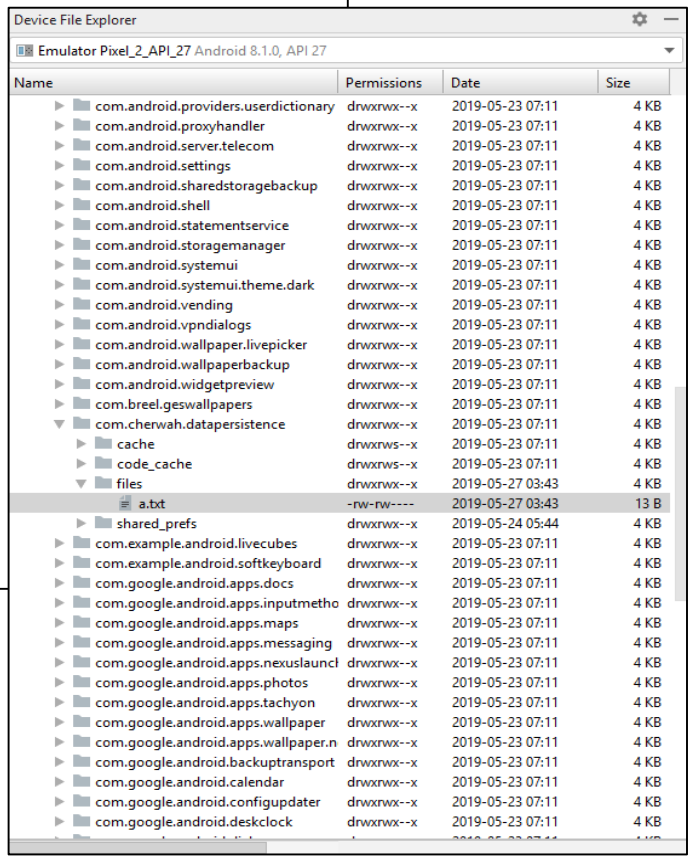
Reading Files

```
StringBuilder strBuilder = new StringBuilder();
FileInputStream inputStream = null;
BufferedReader reader =  null;

try {
    inputStream = openFileInput("a.txt");
    reader = new BufferedReader(new InputStreamReader(inputStream));

    String str;
    while ((str = reader.readLine()) != null)
        strBuilder.append(str);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (reader != null)
            reader.close();

        if (inputStream != null)
            inputStream.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```



© 2019 National University of Singapore. All Rights Reserved.

Writing / Reading external storage

- Writing/reading external storage differ only in path name **/mnt/sdcard/**
  - **getExternalStorageDirectory()**
  - **getExternalStorageState()**
- Specify **WRITE_EXTERNAL_STORAGE** permission in **AndroidManifest.xml**
- Request for **Runtime** permission as well

© 2019 National University of Singapore. All Rights Reserved.

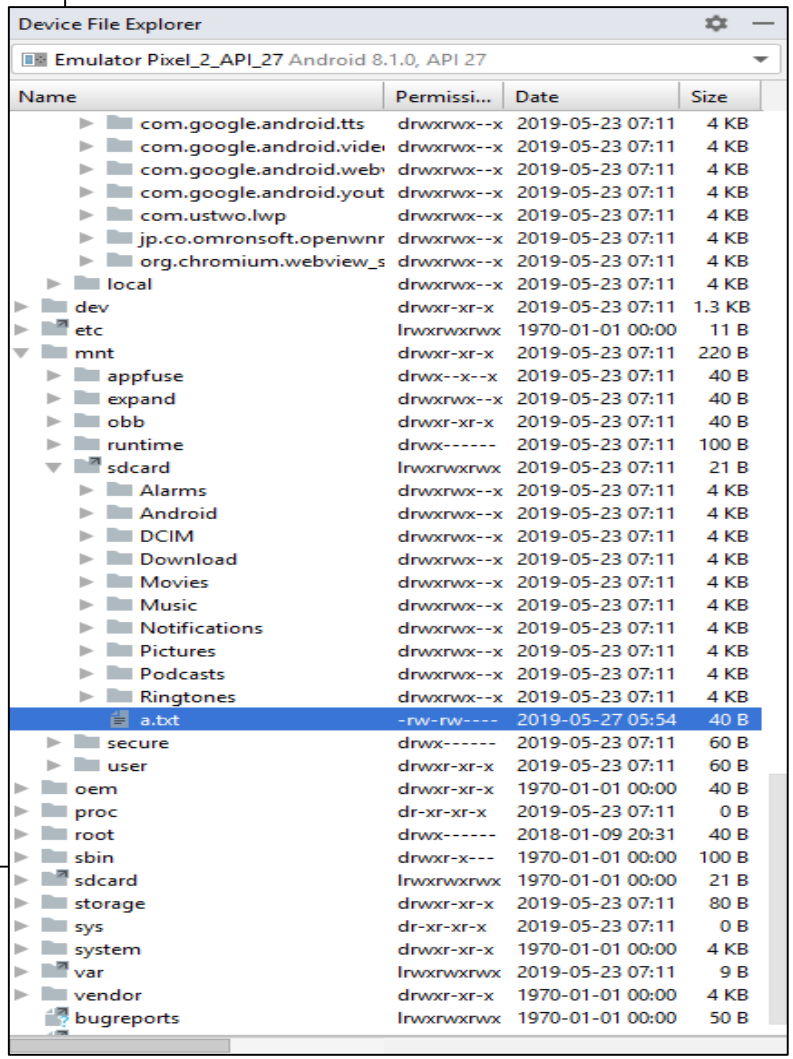
Writing / Reading external storage

```
String[] permissions = new String[] {
    Manifest.permission.WRITE_EXTERNAL_STORAGE
};

File file = Environment.getExternalStorageDirectory();
String path = file.getAbsolutePath() + "/a.txt";

int requestCode = 1;
ActivityCompat.requestPermissions(MainActivity.this,
        permissions, 1);

FileWriter writer = null;
try {
    writer = new FileWriter(path, true);
    writer.write("Android 101\n");
} catch (Exception e) {
    System.out.println(e.toString());
} finally {
    if (writer != null) {
        try {
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



© 2019 National University of Singapore. All Rights Reserved.

Network Access

- Requires **android.permission.INTERNET**
- **NetworkOnMainThreadException** if network operations are performed on the UI thread – create a new Thread instead

```
new Thread(new Runnable() {
    @Override
    public void run() {
        try {
            StringBuilder b = new StringBuilder();
            URL u = new URL("http://www.google.com/search?q=android%20httpclient");
            HttpURLConnection conn = (HttpURLConnection) u.openConnection();
            conn.setRequestMethod("GET");
            conn.connect();

            InputStream ins = conn.getInputStream();
            BufferedReader rd = new BufferedReader(new InputStreamReader(ins));
            String line;
            while ((line = rd.readLine()) != null) {
                b.append(line);
                b.append('\n');
            }
        } catch (Exception e) {
            System.out.println(e.toString());
        }
    }
}).start();
```

© 2019 National University of Singapore. All Rights Reserved.

Active Recall

© 2019 National University of Singapore. All Rights Reserved.