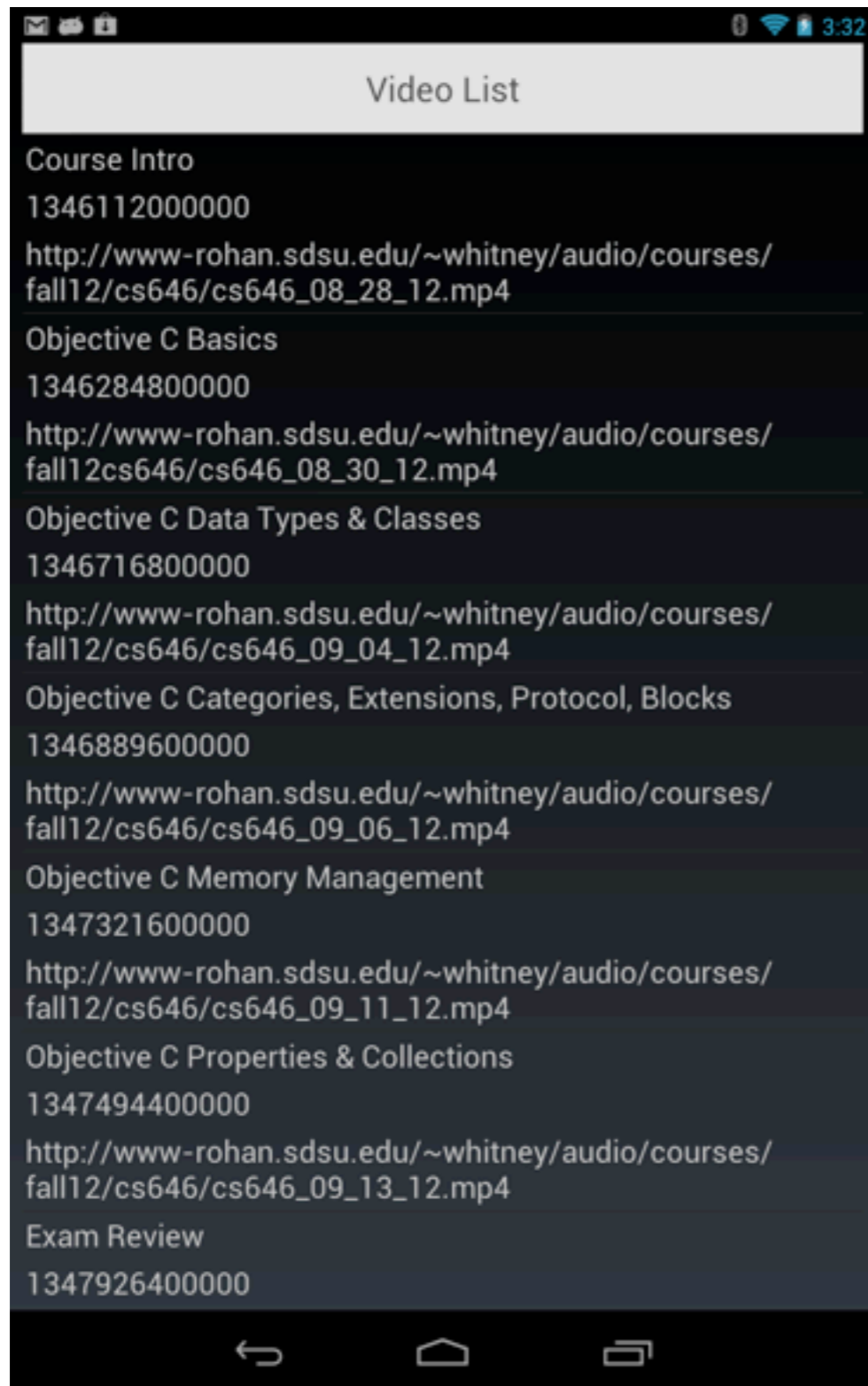


CS 580 Client-Server Programming
Fall Semester, 2012
Doc 16 Milestone 1 Comments & Passing Data to Activities
Oct 25, 2012

Copyright ©, All rights reserved. 2012 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

Users will not put up with this



Really?



Again

```
int read;  
int sread;  
StringBuffer instr = new StringBuffer();  
StringBuffer instr2 = new StringBuffer();  
StringBuffer instr3 = new StringBuffer();
```

Make sure your files are included

| Description | Resource | Path | LocationType |
|---|-------------------------------------|------------|--------------------|
| Project 'com.example.myfirstapp.MainActivity' is missing required library: 'C:/Users/ServiceNow/Desktop/mars.jar' | com.example.myfirstapp.MainActivity | Build path | Build Path Problem |

Static

```
public class Client {  
    static Socket connection;  
    static String nonce;  
  
    public synchronized static void nonce() {blah }  
  
    public synchronized static void convertNonce() { blah }  
  
    public synchronized static void login(String hash) { blah }  
  
    public synchronized static void videolist() { blah }  
  
    public static String loginMessage(String userID, String userPW) { blah }  
  
    public static String videoMessage(int courseID) { blah }
```

What is wrong with using Static?

Only one instance can exist
Rarely is that what you want

Did you

Analyze the situation

Determine that we never will need more than one instance in a program

Compare the alternatives

Determine that using all static methods is the better approach

Or did you just

Start using all static methods & static fields?

If this is the case you are at least 99.99% sure it is the wrong thing to do

android.os.NetworkOnMainThreadException

10-22 19:19:54.971: E/AndroidRuntime(12328): FATAL EXCEPTION: main

10-22 19:19:54.971: E/AndroidRuntime(12328):

android.os.NetworkOnMainThreadException

10-22 19:19:54.971: E/AndroidRuntime(12328): at android.os.StrictMode

NetworkOnMainThreadException

Thrown when an application attempts to perform a networking operation on its main thread

Thrown for applications targeting the Honeycomb SDK or higher

Applications targeting earlier SDK versions are allowed to do networking on their main event loop threads, but it's heavily discouraged

Recall

Don't block the UI thread

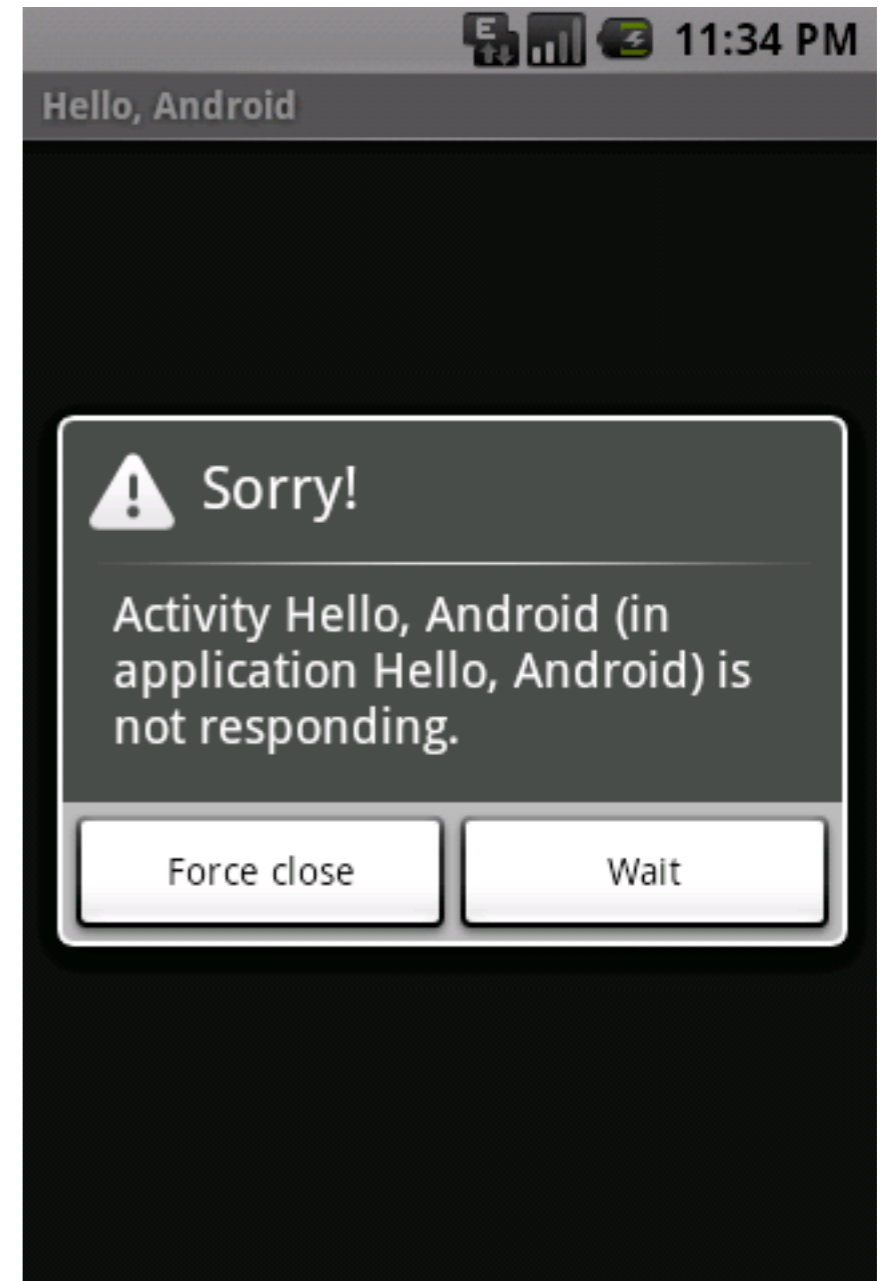
Activity code runs on the UI thread
Create threads to perform long operations

Do not access the Android UI toolkit from outside the UI thread

Use the following to access UI thread
`Activity.runOnUiThread(Runnable)`
`View.post(Runnable)`
`View.postDelayed(Runnable, long)`

ANR - Application Not Responding

No response to an input event (e.g. key press, screen touch) within 5 seconds



Strick Mode

```
public void onCreate() {
    StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()
        .detectDiskReads()
        .detectDiskWrites()
        .detectNetwork() // or .detectAll() for all detectable problems
        .penaltyLog()
        .build());
    StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()
        .detectLeakedSqlLiteObjects()
        .detectLeakedClosableObjects()
        .penaltyLog()
        .penaltyDeath()
        .build());
    super.onCreate();
}
```

So

**Don't perform
network activity on
the main thread**

But

it works in Android 2.x

Don't perform
network activity
on the main
thread

But

I already did it that way

Don't perform
network activity
on the main
thread

But

I promise not to do it after assignment 4

Don't perform
network activity
on the main
thread

But

what is the big deal?

Well

App will not run on Android 3 or greater

On slow network user gets ANR

App rejected from app store

If you can not do it correctly when slides are screaming at you
when will you do it correctly

So

Don't perform
network
activity on the
main thread

Passing Data to an Activity

Sending the data

```
public void onClick(View v) {  
    Intent go;  
    go = new Intent();  
    go.setAction("android.intent.action.EDIT");  
    go.addCategory("person_editor");  
    String newAge = numberText.getText().toString();  
    go.putExtra("age", newAge);  
    go.putExtra("name", "Roger");  
    startActivityForResult(go, INTENT_EXAMPLE_REQUEST);  
}
```


Receiving the Data

```
public class PersonEditor extends Activity implements View.OnClickListener {
    private EditText ageText;
    private EditText nameText;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.person_editor);
        Button done = (Button) findViewById(R.id.edit_done);
        done.setOnClickListener(this);
        ageText = (EditText) this.findViewById(R.id.edit_age);
        nameText = (EditText) this.findViewById(R.id.edit_name);
        Bundle personData = getIntent().getExtras();
        String age = personData.getString("age");
        String name = personData.getString("name");
        if ((age != null) && (name != null)) {
            ageText.setText(age);
            nameText.setText(name);
        }
    }
}
```

But

Can only send Serializable data to other Activity

Serializable

Able to convert object to format that can be stored in a file or sent on a network

Sockets, SocketChannels are not serializable

Objects that contain sockets or socket channels can not be serialized with the socket

So - How to pass Socket/Client to Activity

Singleton

android.app.Application

Singleton

Ensure a class only has one instance

Provide global point of access to single instance

```
public class VideoClient {  
    private Socket serverConnection;  
    private static VideoClient instance;  
    private VideoClient() { blah}  
  
    public static Video instance() {  
        if (instance == null)  
            instance = new VideoClient();  
        return instance;  
    }  
    etc.
```

When you need a client

```
client = VideoClient.getInstance();
```

Issues

Arguments to constructor

Exceptions thrown in constructor

Garbage collection

Don't care about restricting to one instance

Misuse of singleton

Arguments & Exceptions

```
public class VideoClient {
    private Socket serverConnection;
    private static VideoClient instance;
    public VideoClient(String host,int port) throws SomeException
        { blah}

    public static Video instance(String host,int port) throws SomeException{
        if (instance == null)
            instance = new VideoClient(host, port);
        return instance;
    }
    etc.
```

Now don't need arguments each time

```
public class VideoClient {
    private Socket serverConnection;
    private static VideoClient instance;
    public VideoClient(String host,int port) throws SomeException
        { blah}

    public static Video instance() {
        return instance;
    }

    public static void setInstance(VideoClient aClient ) {
        instance = aClient;
    }
    etc.
```

Multiple Instances

What if we want multiple instance?

Store them in hash table

Need way of requesting particular instance

```
public class VideoClient {
    private static HashMap<String,VideoClient> instances =
        new HashMap<String,VideoClient>();

    public static Video instance(String name) {
        if (!instances.containsKey(name))
            instances.put(name, new VideoClient() );
        return instances.get(name);
    }
}
```


Garbage Collection

Static variables can be garbage collected

So may get new instance after already created an instance

Not so common in regular Java programs

Android apps

Normally go to background

Activities and other objects reclaimed as memory is needed

Expect activities, objects & statics to be reclaimed

android.app.Application

Each app has single instance of android.app.Application

Can subclass android.app.Application and store data in subclass

New Application Class

```
public class VideoApplication extends Application {  
    private VideoClient client;  
  
    public VideoClient getClient(){  
        return client;  
    }  
  
    public void setClient(VideoClient client){  
        this.client = client;  
    }  
}
```

Setting the client

```
public class MainActivity extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
  
        VideoApplication appInstance = ((VideoApplication)getApplicationContext());  
        appInstance.setClient(new VideoClient);  
    }  
}
```

In other Activities

```
VideoApplication appInstance = ((VideoApplication)getApplicationContext());  
VideoClient = appInstance.getClient();
```

In Manifest File

```
package="edu.sdsu.cs.mars"  
android:versionCode="1"  
android:versionName="1.0" >
```

```
<uses-sdk  
    android:minSdkVersion="8"  
    android:targetSdkVersion="15" />  
    <uses-permission android:name="android.permission.INTERNET"/>
```

```
<application  
    android:icon="@drawable/ic_launcher"  
    android:label="@string/app_name"  
    android:name=".VideoApplication"  
    android:theme="@style/AppTheme" >
```