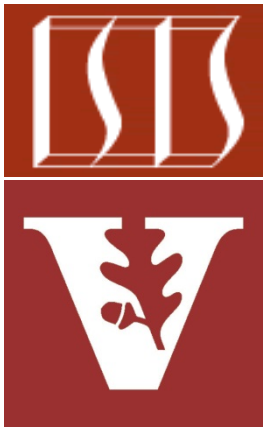
Android Concurrency: Programming with Android Concurrency Frameworks (Part 2)

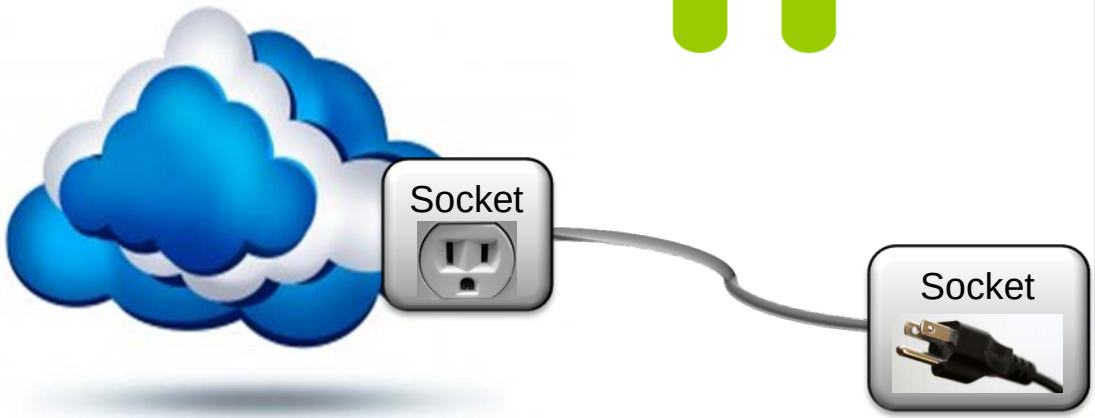


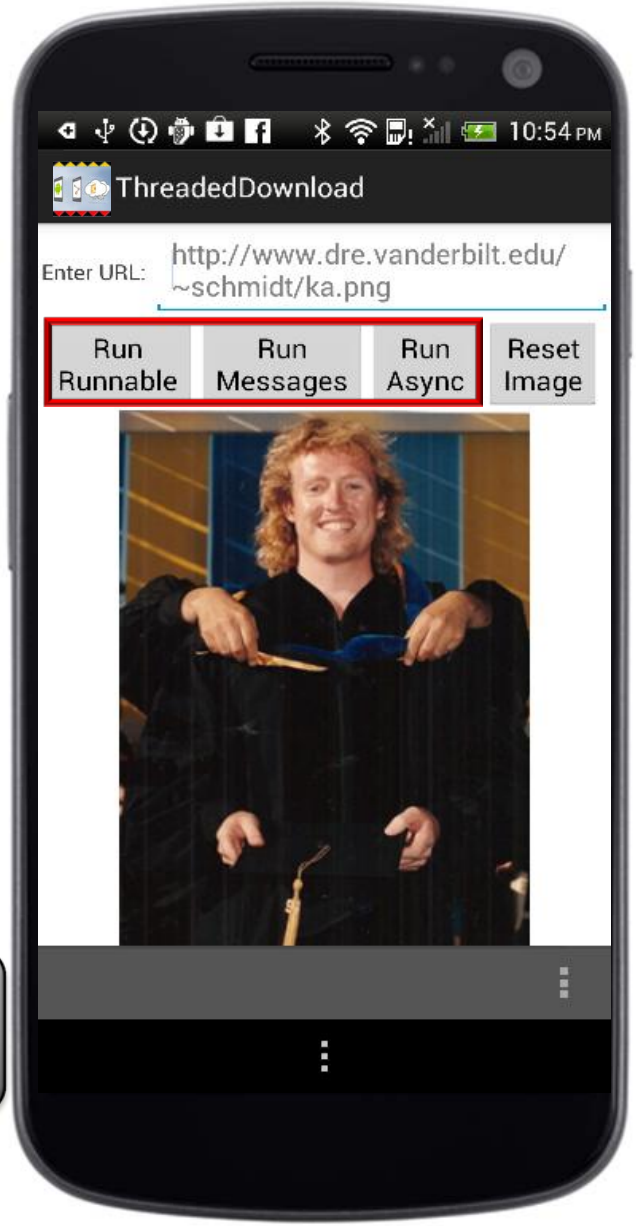
Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

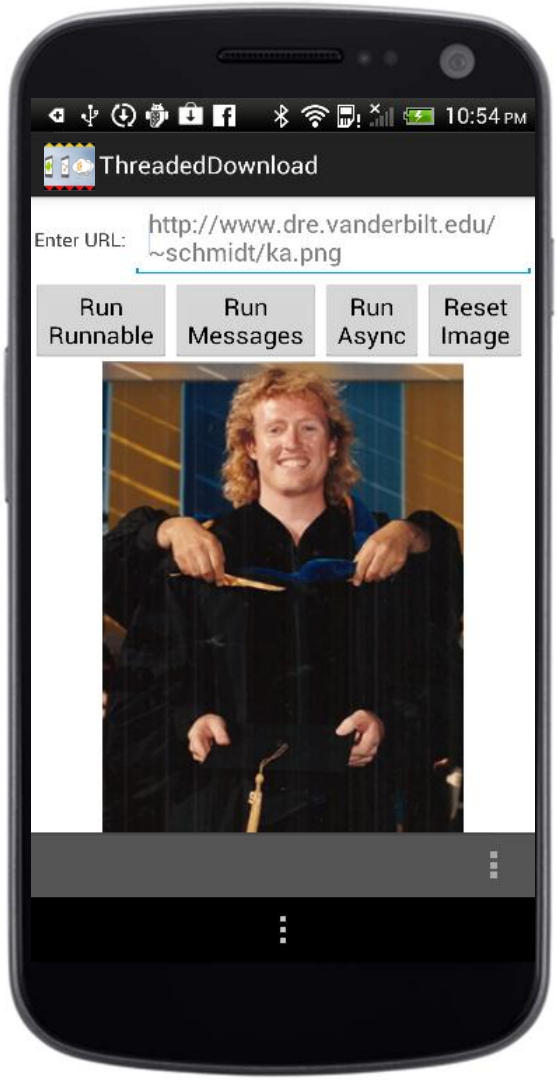
Learning Objectives in this Part of the Module

- Understand how to program the Threaded Downloads application using Android's HaMeR & AsyncTask frameworks

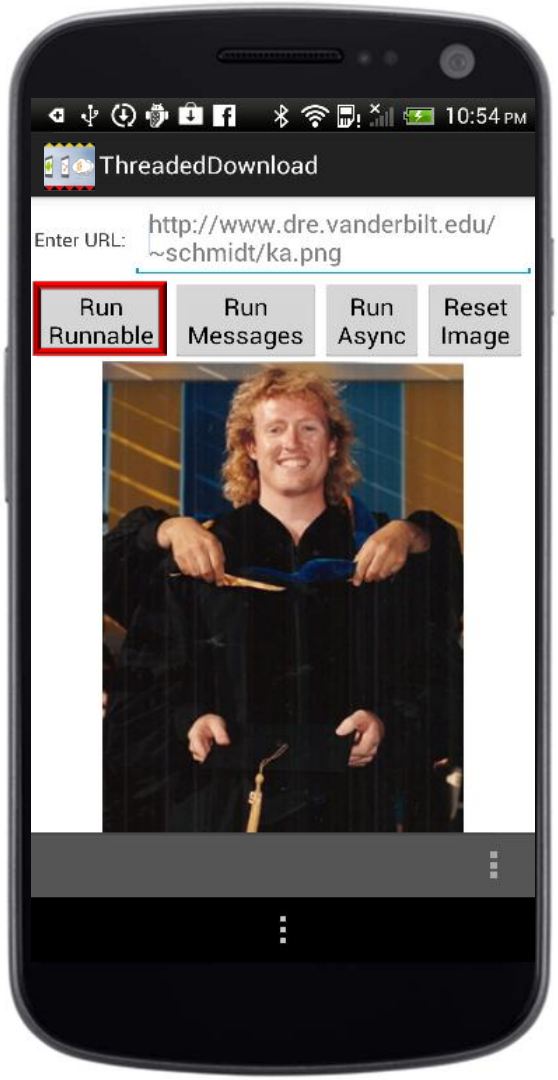




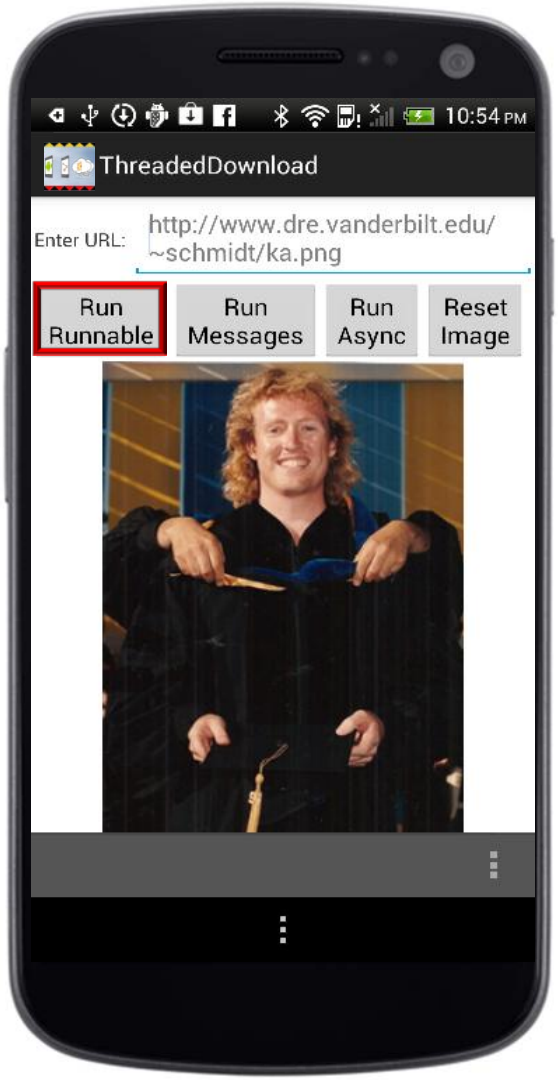
"Run Runnable" Behavior for ThreadedDownloads



"Run Runnable" Behavior for ThreadedDownloads

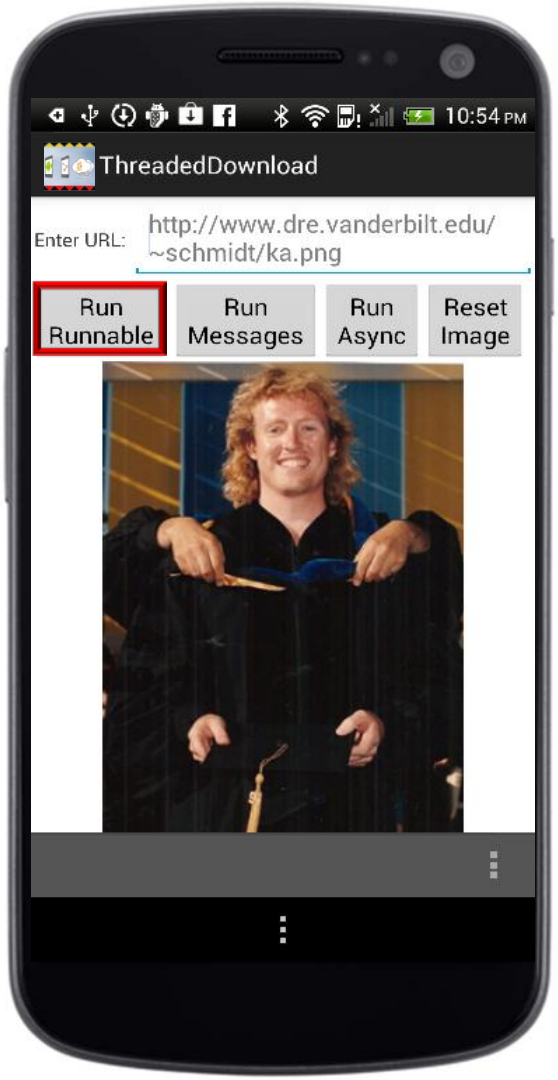


"Run Runnable" Behavior for ThreadedDownloads



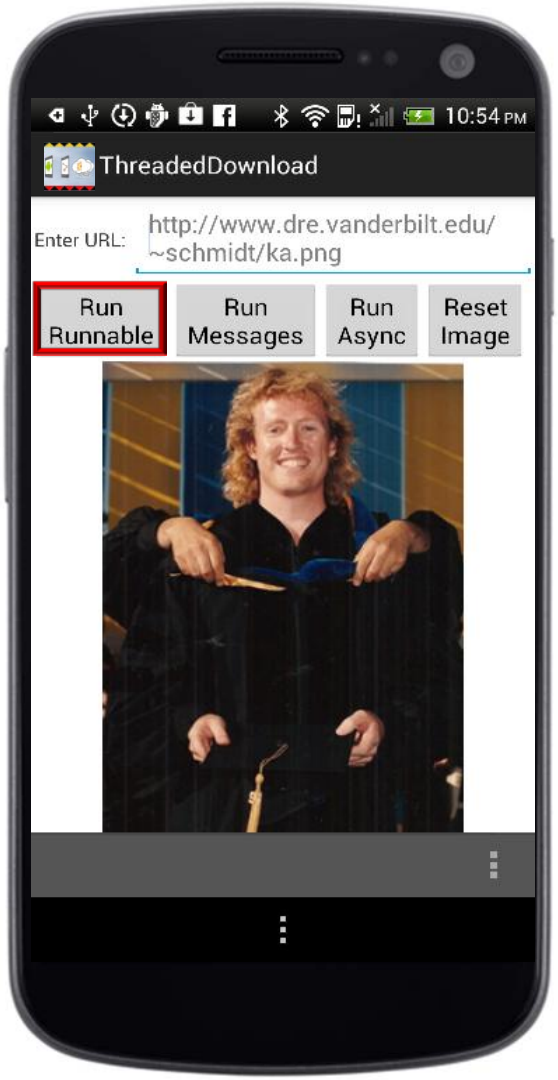

```
<Button
   ...
   android:onClick="runRunnable"
   android:text="@string/runRunnable" />
```

"Run Runnable" Behavior for ThreadedDownloads




```
<Button
   ...
   android:onClick="runRunnable"
   android:text="@string/runRunnable" />


public class ThreadedDownloads
             extends Activity {
   ...
   public void runRunnable
     (View view) {...}
   public void runMessages
     (View view) {...}
   public void runAsyncTask
     (View view) {...}
```

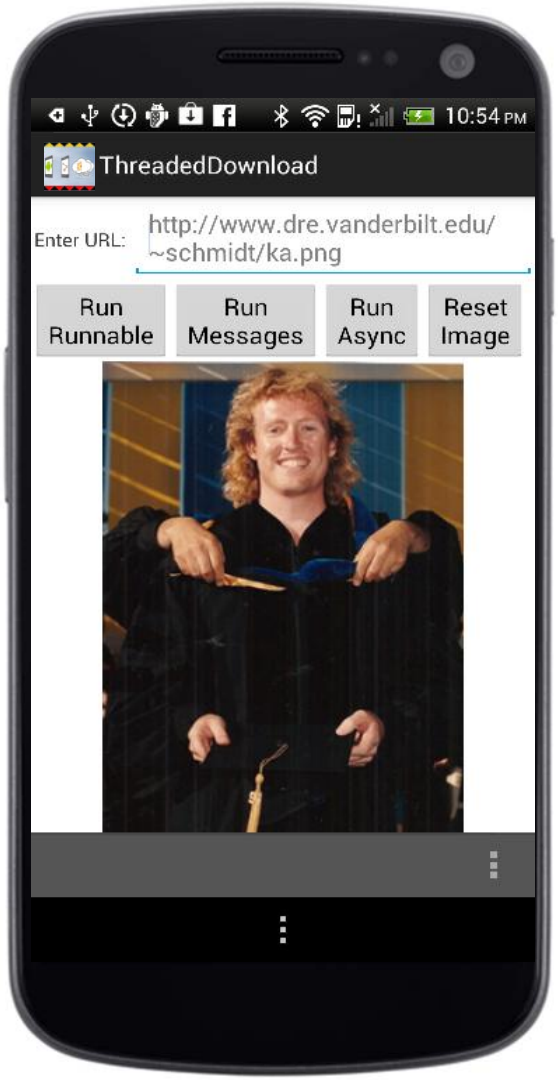
"Run Runnable" Behavior for ThreadedDownloads

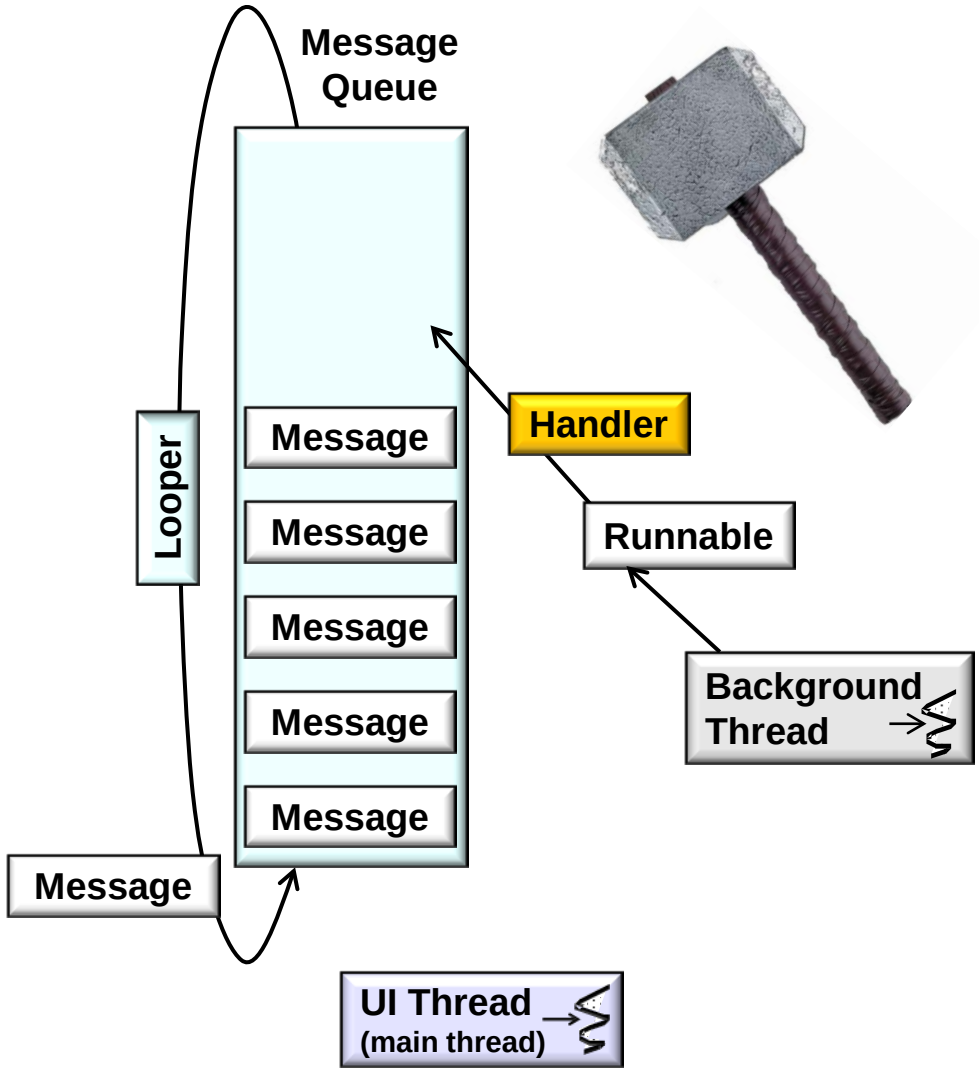



```
<Button
   ...
   android:onClick="runRunnable"
   android:text="@string/runRunnable" />


public class ThreadedDownloads
             extends Activity {
   ...
   public void runRunnable
     (View view) {...}
   public void runMessages
     (View view) {...}
   public void runAsyncTask
     (View view) {...}
```

ThreadedDownloads/src/edu/vuum/mocca/ThreadedDownloads.java has code

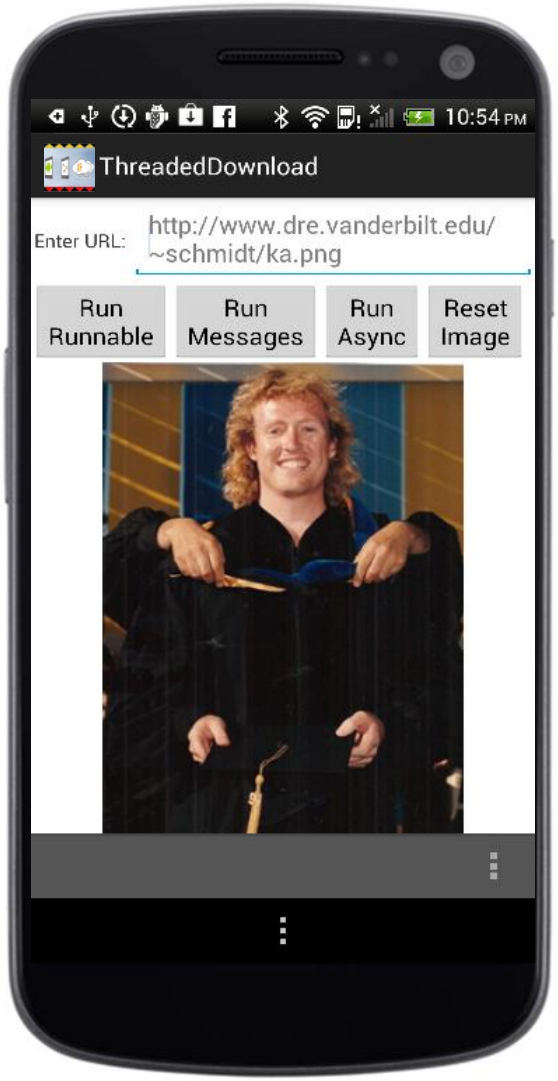
"Run Runnable" Behavior for ThreadedDownloads

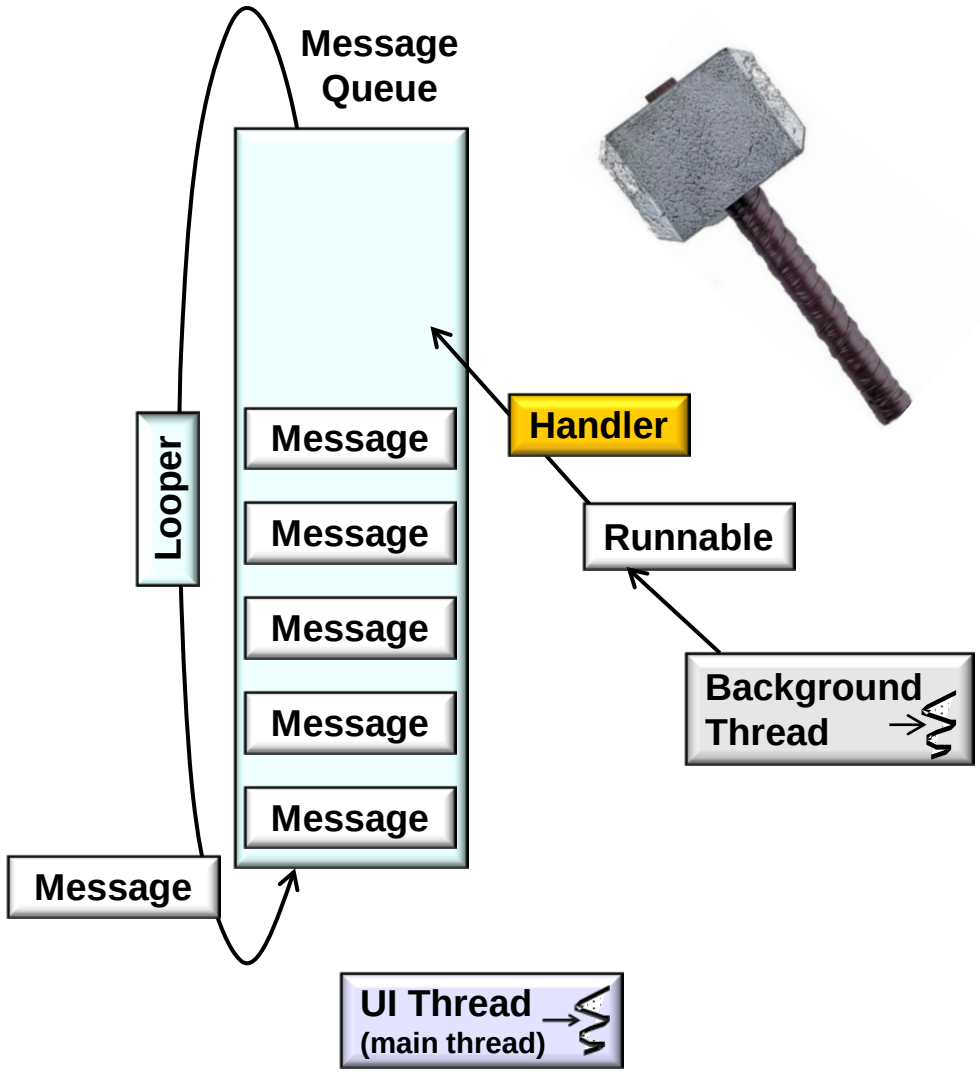




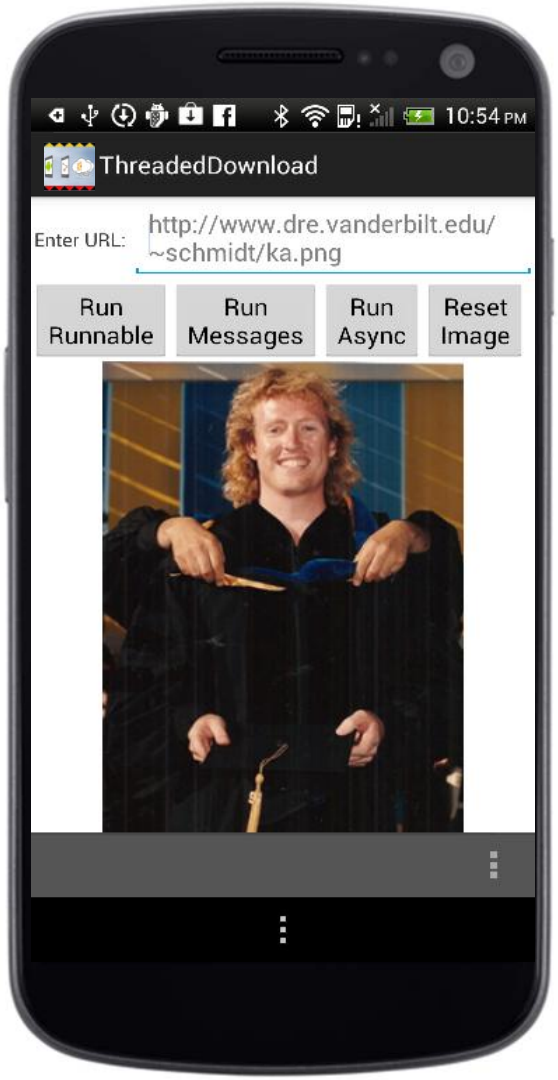
See earlier part on "Posting & Processing Runnables in Android Handler"

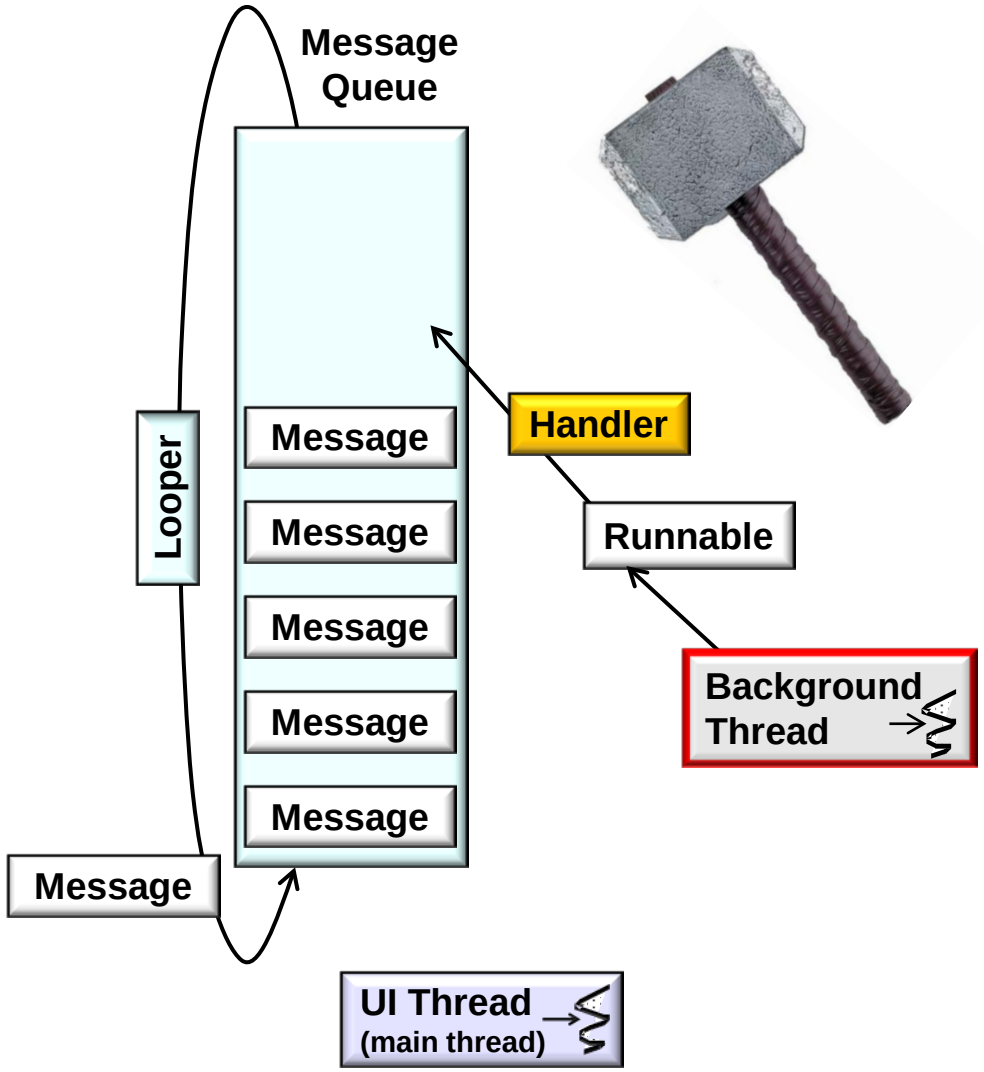
"Run Runnable" Behavior for ThreadedDownloads



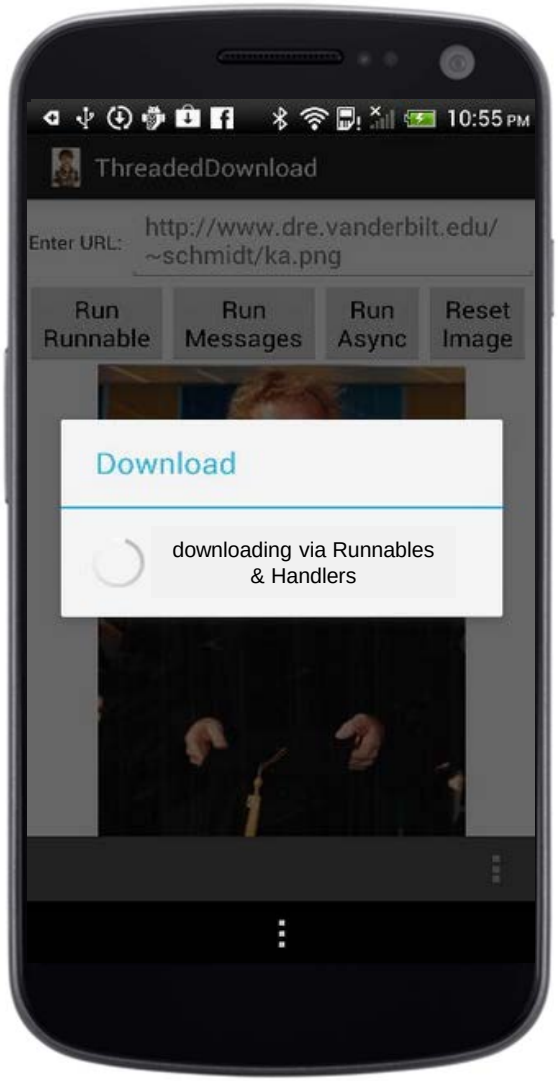


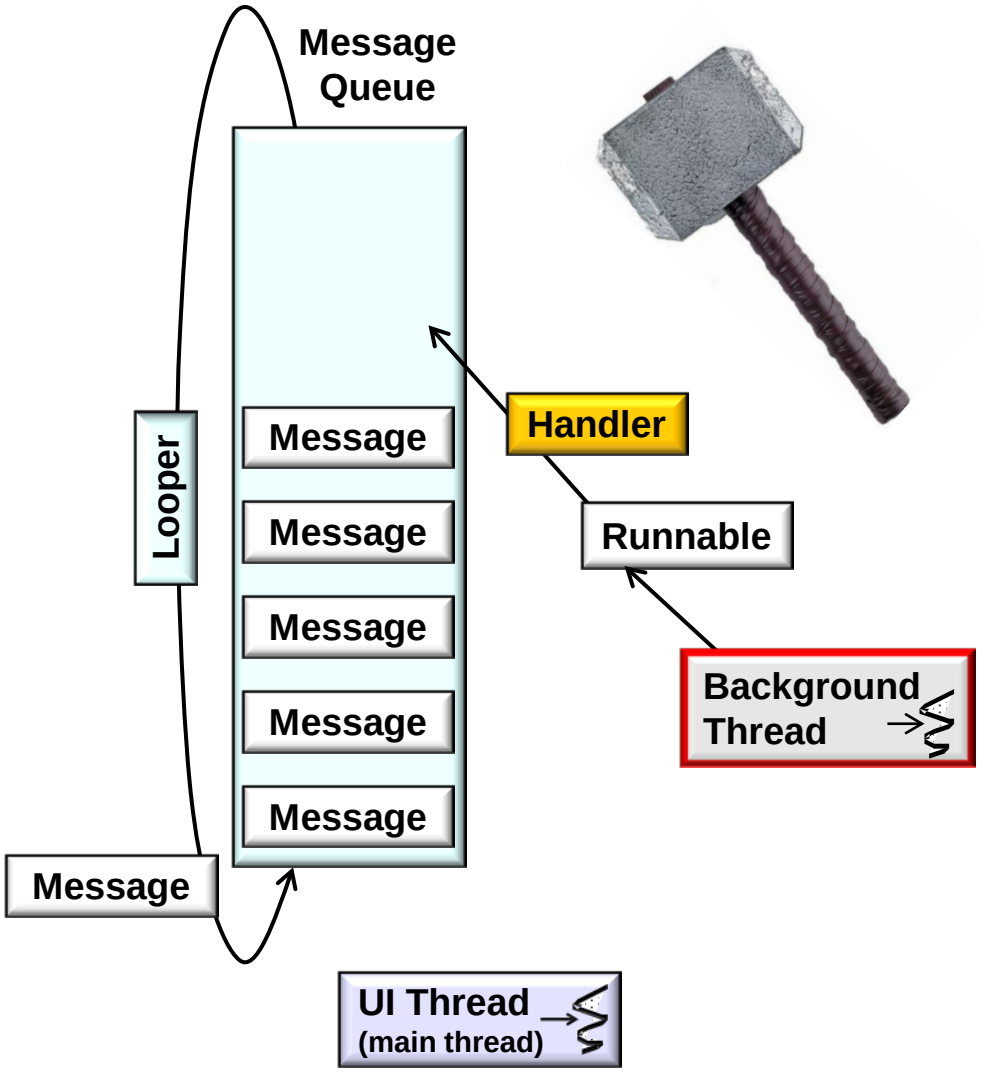
"Run Runnable" Behavior for ThreadedDownloads



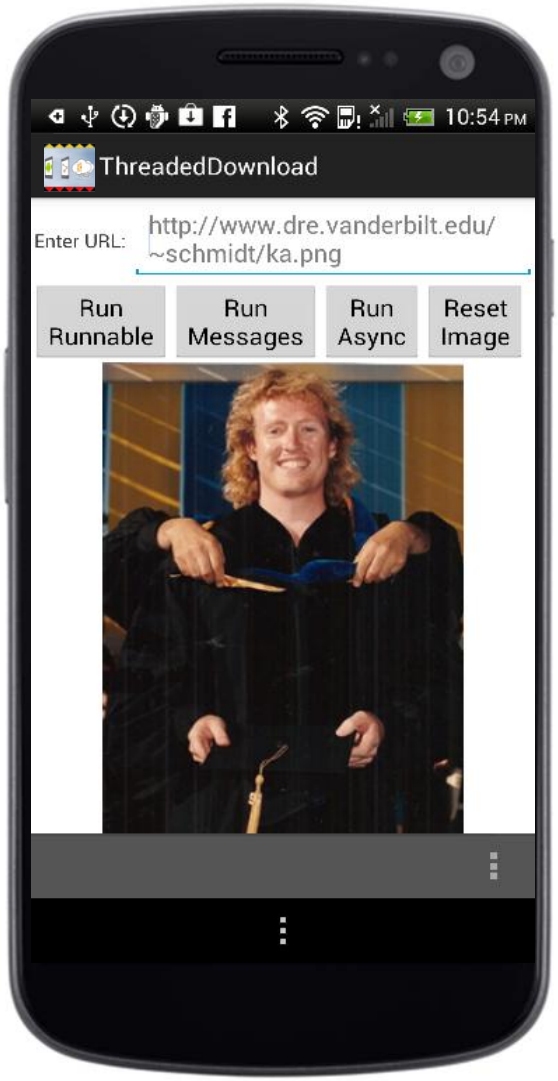


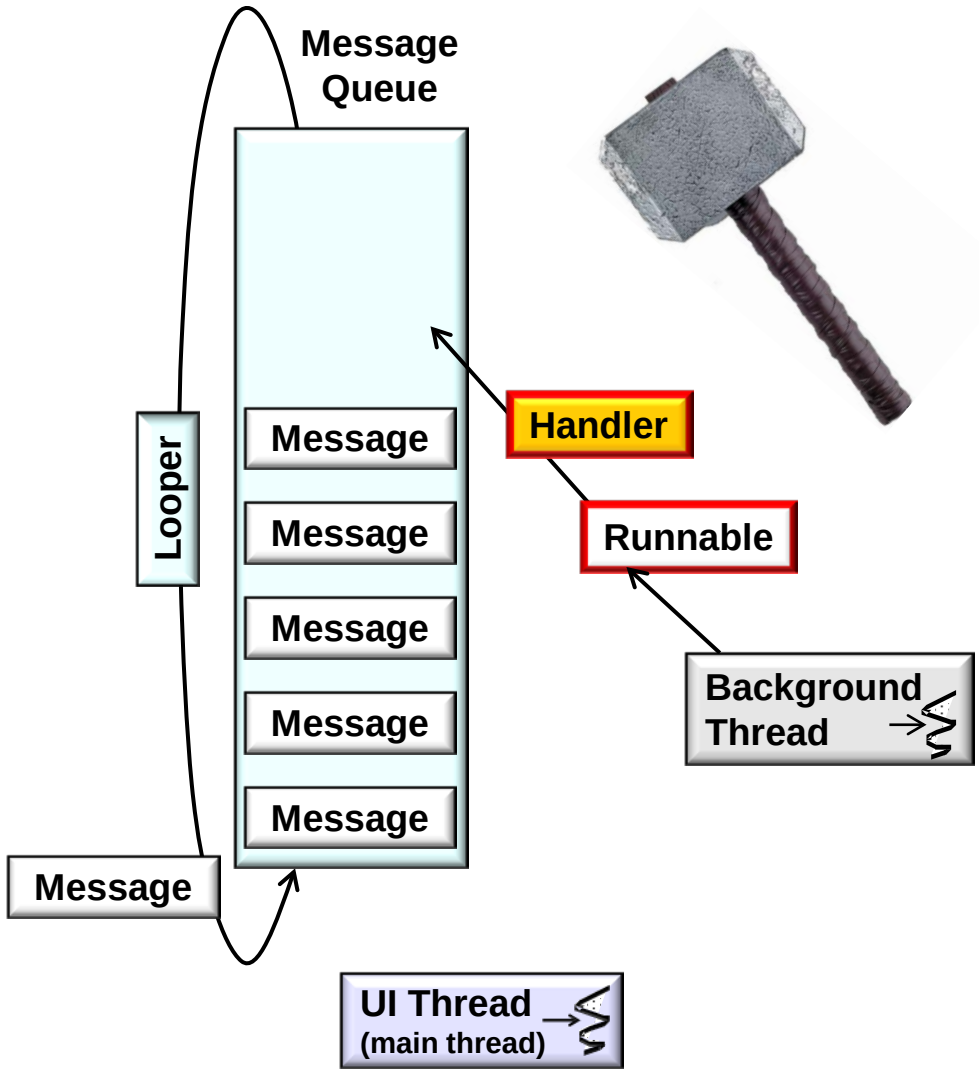
"Run Runnable" Behavior for ThreadedDownloads



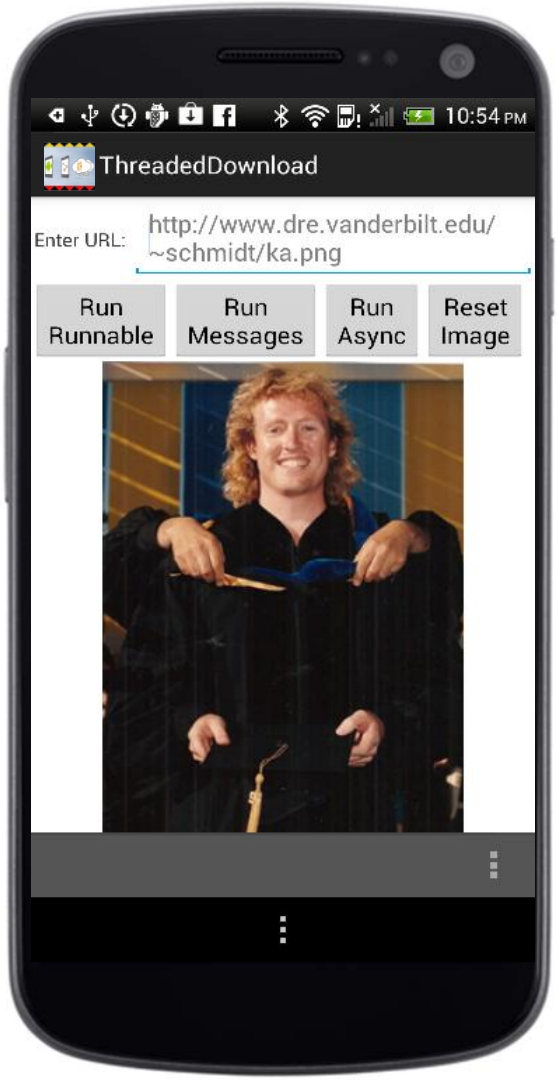


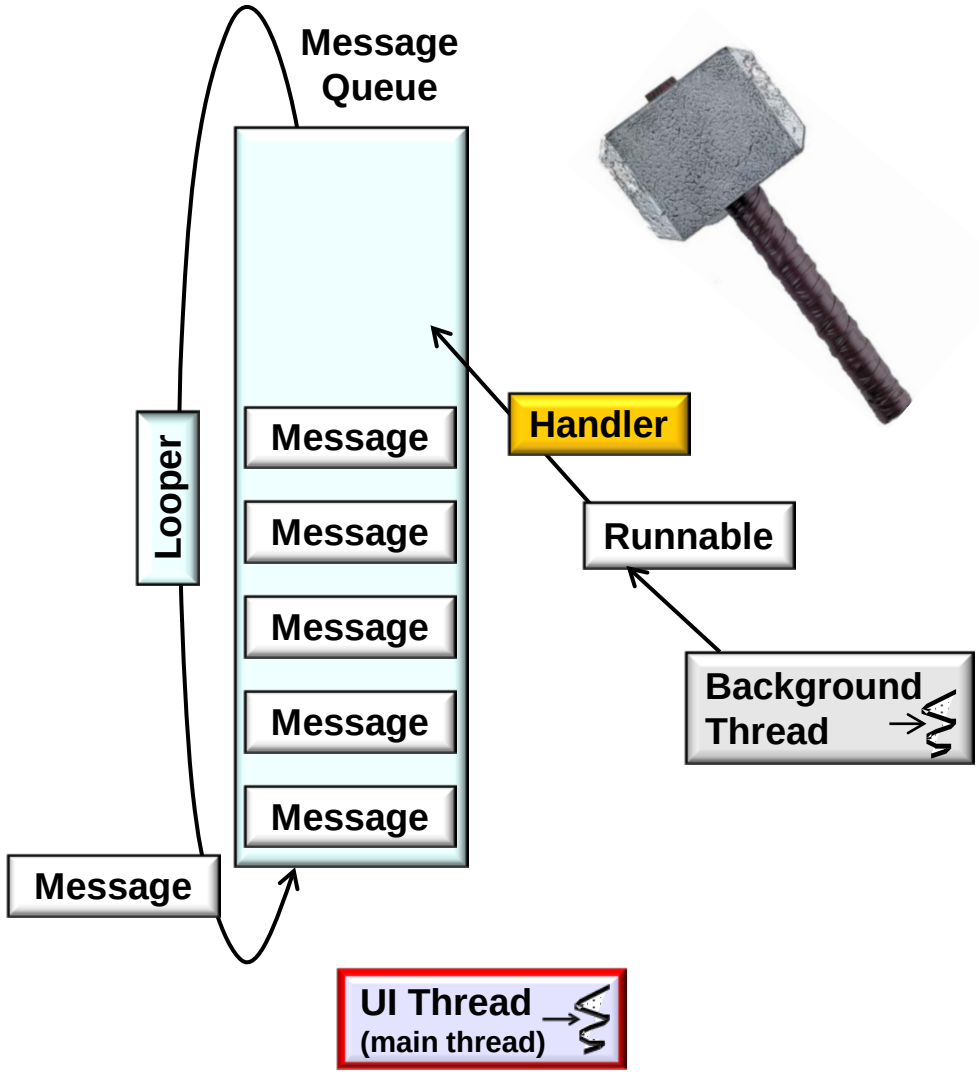
"Run Runnable" Behavior for ThreadedDownloads



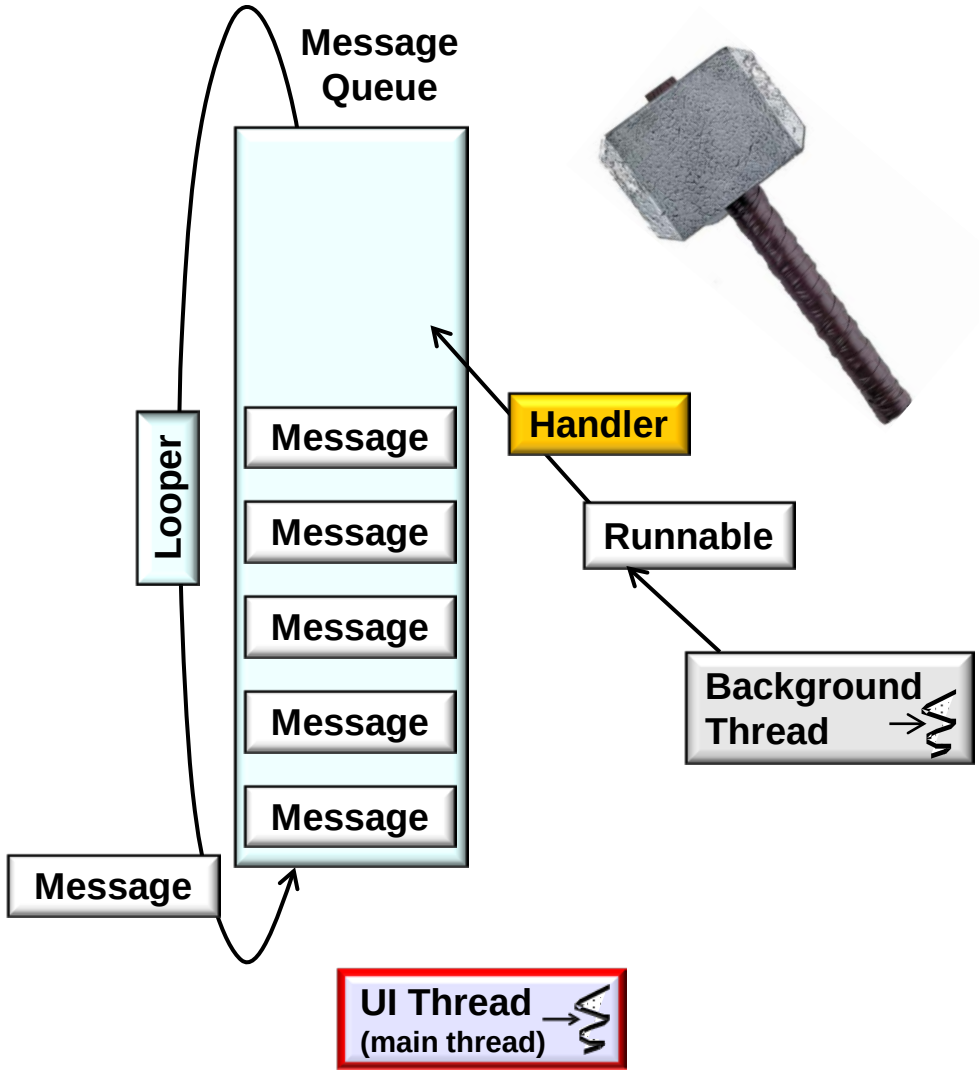


"Run Runnable" Behavior for ThreadedDownloads





"Run Runnable" Behavior for ThreadedDownloads



ThreadedDownloads RunRunnable() Method

- Called when a user clicks the “Run Runnables” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runRunnables(View view) {
    String url = getUrlString();
    ...


    showDialog("downloading via Runnables and Handlers");


    new Thread(new RunnableWithHandlers(url)).start();

  }
  ...
```

ThreadedDownloads RunRunnable() Method

- Called when a user clicks the “Run Runnables” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runRunnables(View view) {
    String url = getUrlString();
    ...

    showDialog("downloading via Runnables and Handlers");


    new Thread(new RunnableWithHandlers(url)).start();

  }
  ...
```

Obtain requested URL from user input

ThreadedDownloads RunRunnable() Method

- Called when a user clicks the “Run Runnables” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runRunnables(View view) {
    String url = getUrlString();
    ...

    showDialog("downloading via Runnables and Handlers");


    new Thread(new RunnableWithHandlers(url)).start();

  }
  ...
```

Inform user the download is starting

ThreadedDownloads RunRunnable() Method

- Called when a user clicks the “Run Runnables” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runRunnables(View view) {
    String url = getUrlString();
    ...

    showDialog("downloading via Runnables and Handlers");

    new Thread(new RunnableWithHandlers(url)).start();
  }
  ...
```

Create & start a new Thread to download an image in the background via a Runnable CMD

ThreadedDownloads RunRunnable() Method

- Called when a user clicks the “Run Runnables” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runRunnables(View view) {
    String url = getUrlString();
    ...

    showDialog("downloading via Runnables and Handlers");

    new Thread(new RunnableWithHandlers(url)).start();
  }
  ...
```

Create & start a new Thread to download an image in the background via a Runnable CMD

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    String mUrl;



    RunnableWithHandlers(String url) {
      mUrl = url;
    }
```

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    String mUrl;



    RunnableWithHandlers(String url) {
      mUrl = url;
    }
```

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    String mUrl;
```

URL to download

```
    RunnableWithHandlers(String url) {
      mUrl = url;
    }
```

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    public void run() {
```

Executes in a background Thread

```
      final Bitmap image = downloadImage(mUrl);
      ...
```

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    public void run() {

      Retrieve the image from the server

      final Bitmap image = downloadImage(mUrl);
      ...
```

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    public void run() {
      ...

      ThreadedDownloads.this.runOnUiThread(new Runnable() {
        public void run() {


          mProgressDialog.dismiss();

          displayImage(image);
        }
      });
```

Post a Runnable command to display downloaded image in the UI Thread

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    public void run() {
      ...

      ThreadedDownloads.this.runOnUiThread(new Runnable() {
        public void run() {


          mProgressDialog.dismiss();

          displayImage(image);
        }
      });
```

Post a Runnable command to display downloaded image in the UI Thread

See earlier part on “Posting & Processing Runnables in Android Handler”

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    public void run() {
      ...

      ThreadedDownloads.this.runOnUiThread(new Runnable() {
        public void run() {


          mProgressDialog.dismiss();

          displayImage(image);
        }
      });
```

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    public void run() {
      ...

      ThreadedDownloads.this.runOnUiThread(new Runnable() {
        public void run() {

          mProgressDialog.dismiss();

          displayImage(image);
        }
      });
```

Dismiss the progress dialog

ThreadedDownloads RunnableWithHandlers

- Downloads image in a background Thread using a Runnable & a Handler

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithHandlers implements Runnable {

    public void run() {
      ...

      ThreadedDownloads.this.runOnUiThread(new Runnable() {
        public void run() {


          mProgressDialog.dismiss();

          displayImage(image);
        }
      });
```

Display downloaded image to user

ThreadedDownloads displayImage() Method

- Displays a successfully downloaded & converted image or reports an error

```
public class ThreadedDownloads extends Activity {
  ...
  private void displayImage(Bitmap image) {
    ...
    else if (image != null)
      mImageView.setImageBitmap(image);



    else
      showErrorToast("image is corrupted,"
                     + " please check the requested URL.");
  }
```

displayImage() is shared by all three concurrency models

ThreadedDownloads displayImage() Method

- Displays a successfully downloaded & converted image or reports an error

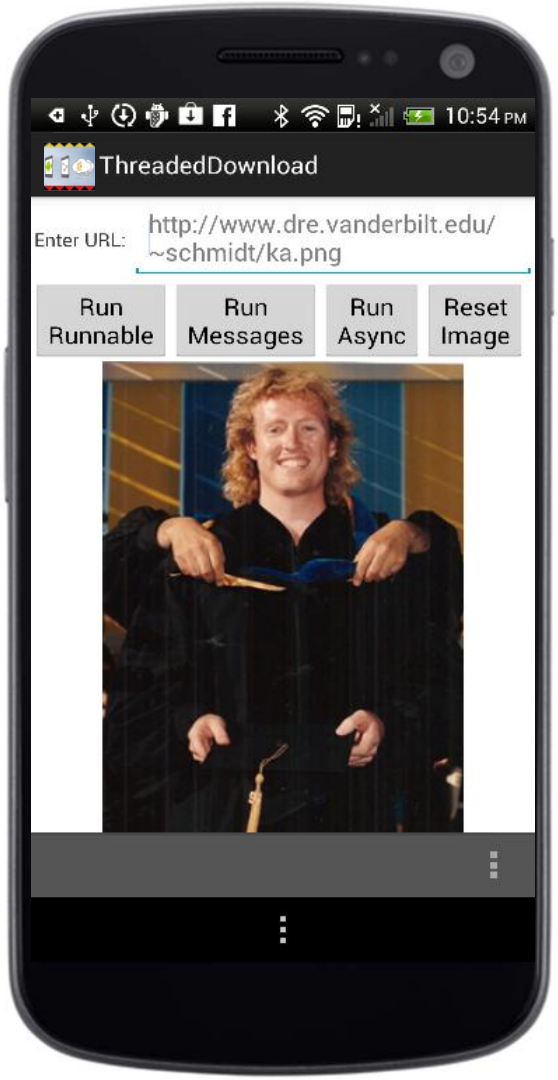
```
public class ThreadedDownloads extends Activity {
  ...
  private void displayImage(Bitmap image) {
    ...
    else if (image != null)
      mImageView.setImageBitmap(image);

    else
      showErrorToast("image is corrupted,"
                     + " please check the requested URL.");
  }
```

Display downloaded image to user if download & conversion process succeeded

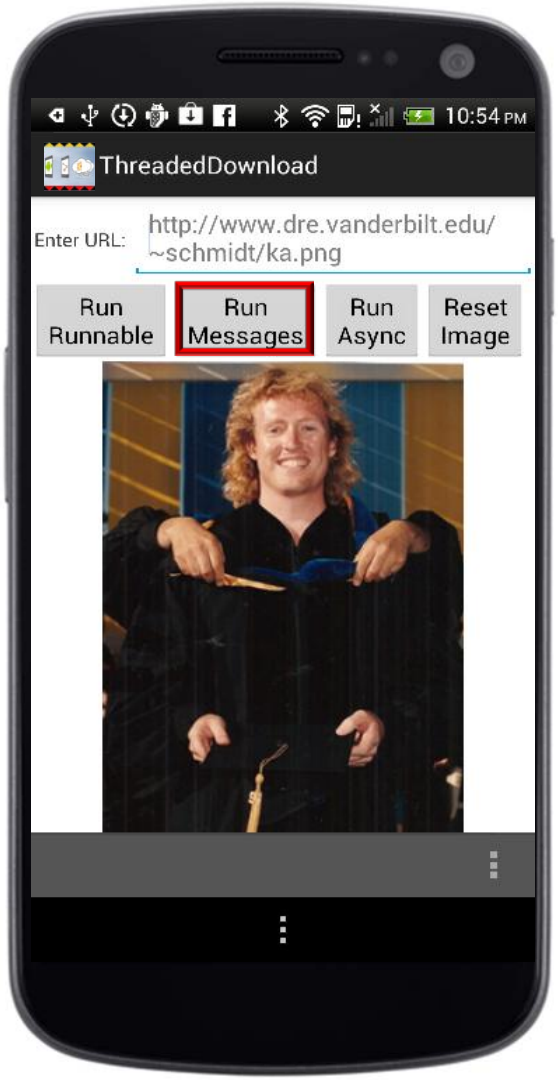
displayImage() is shared by all three concurrency models

Programming with Handlers & Messages

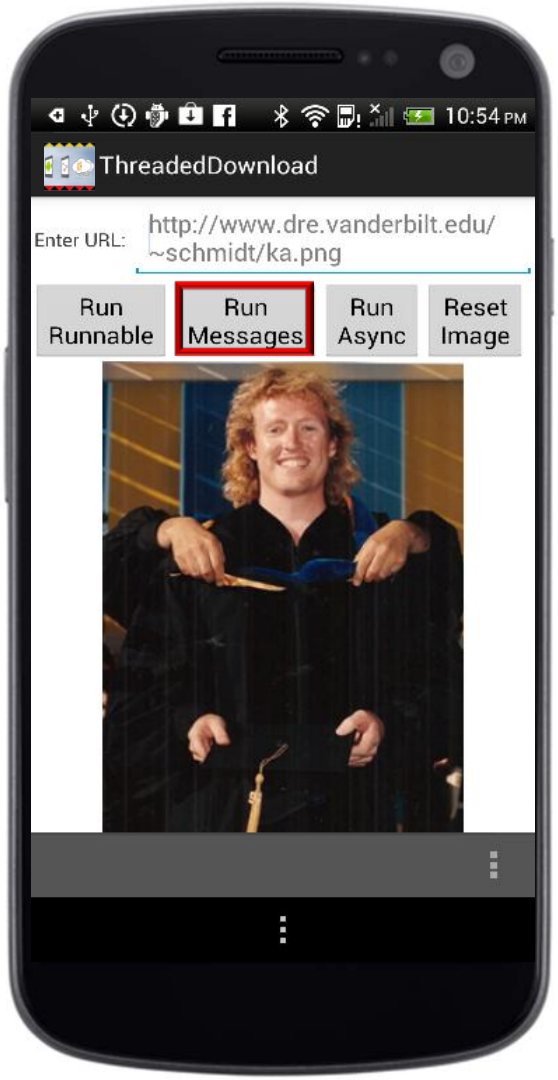
"Run Messages" Behavior for ThreadedDownloads



"Run Messages" Behavior for ThreadedDownloads

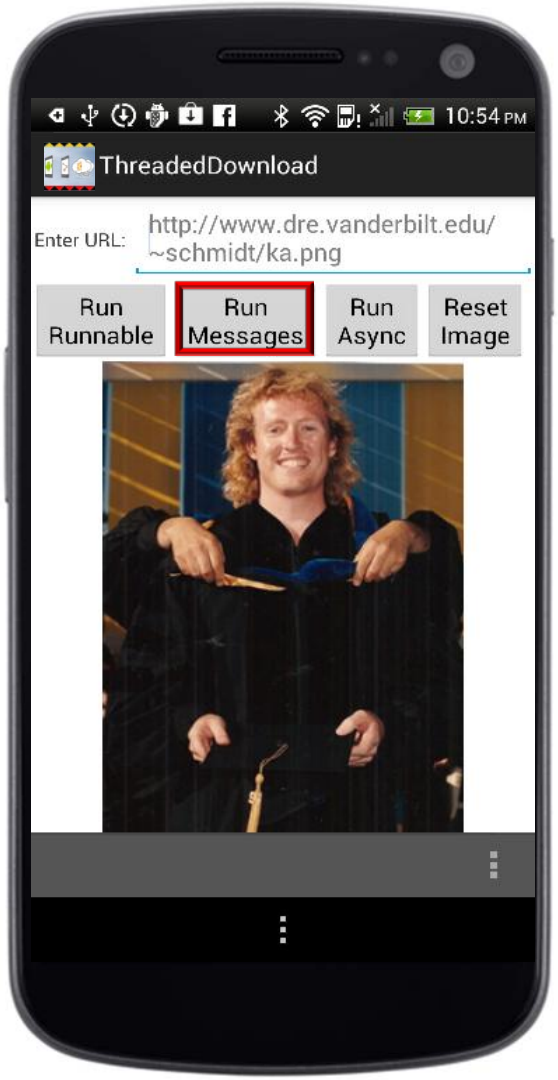


"Run Messages" Behavior for ThreadedDownloads



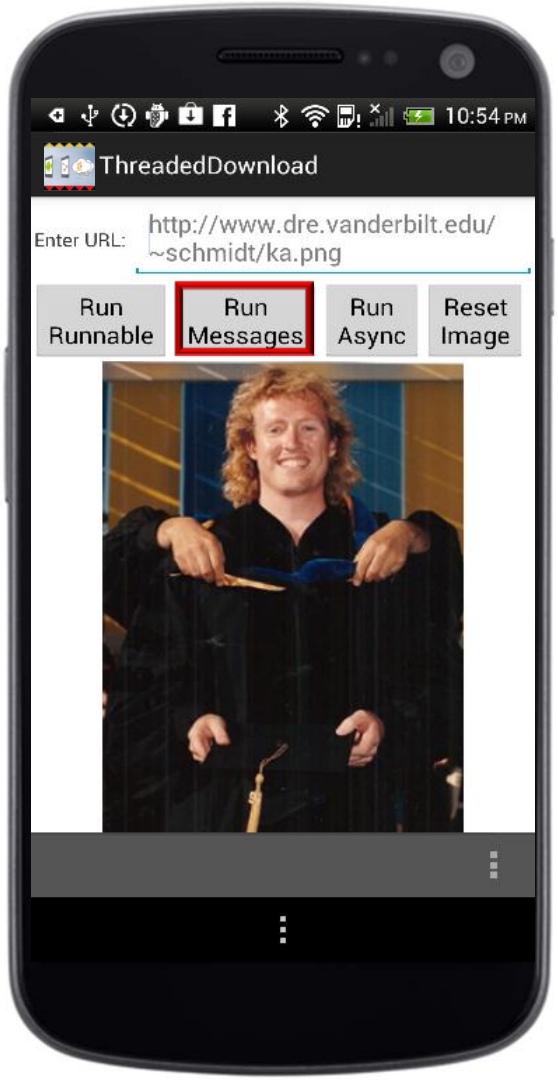

```
<Button
   ...
   android:onClick="runMessages"
   android:text="@string/runMessages" />
```

"Run Messages" Behavior for ThreadedDownloads




```
<Button
  ...
  android:onClick="runMessages"
  android:text="@string/runMessages" />


public class ThreadedDownloads
             extends Activity {
  ...
  public void runRunnable
    (View view) {...}
  public void runMessages
    (View view) {...}
  public void runAsyncTask
    (View view) {...}
```

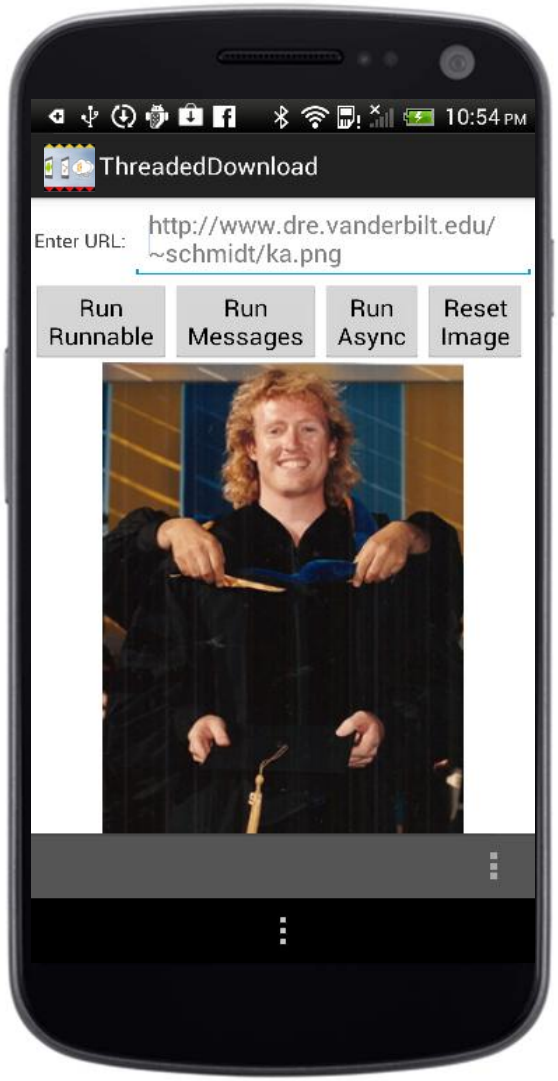
"Run Messages" Behavior for ThreadedDownloads

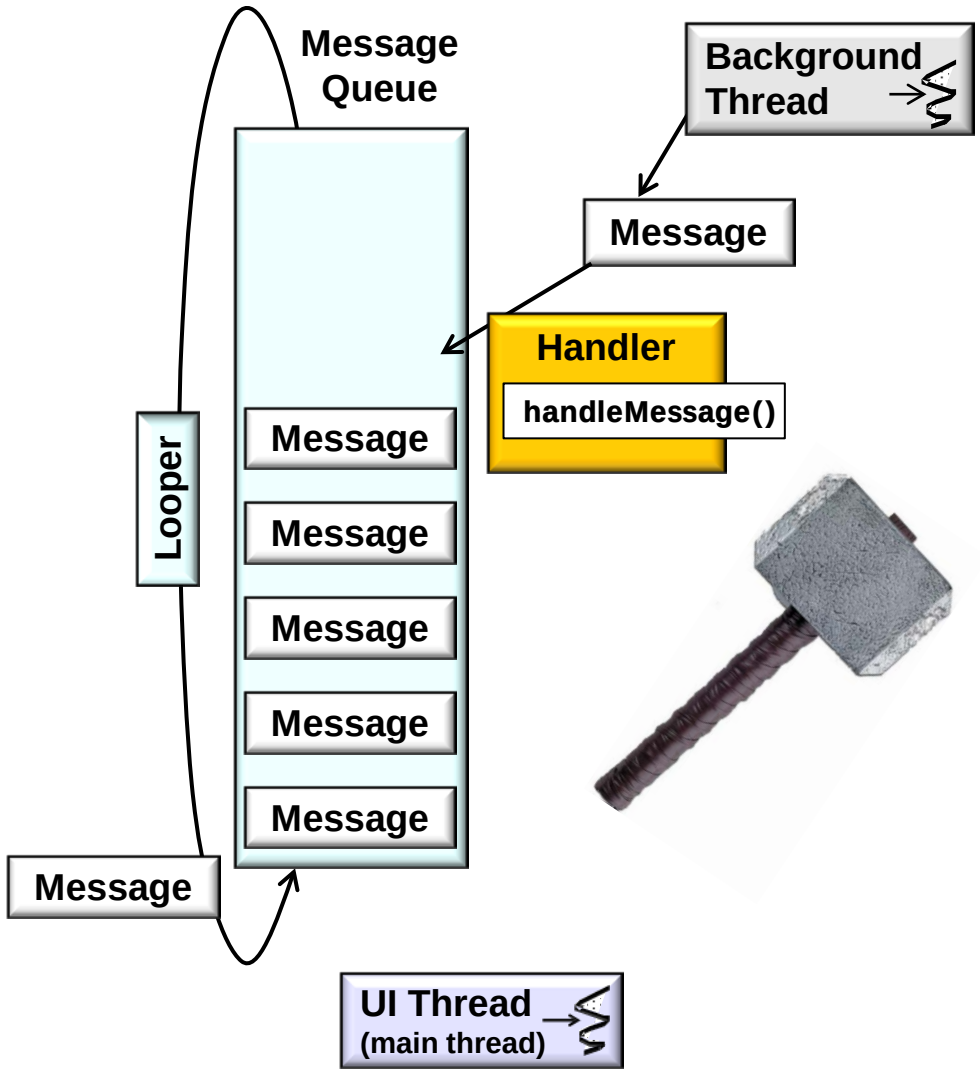



```
<Button
  ...
  android:onClick="runMessages"
  android:text="@string/runMessages" />


public class ThreadedDownloads
             extends Activity {
  ...
  public void runRunnable
    (View view) {...}
  public void runMessages
    (View view) {...}
  public void runAsyncTask
    (View view) {...}
```

ThreadedDownloads/src/edu/vuum/mocca/ThreadedDownloads.java has code

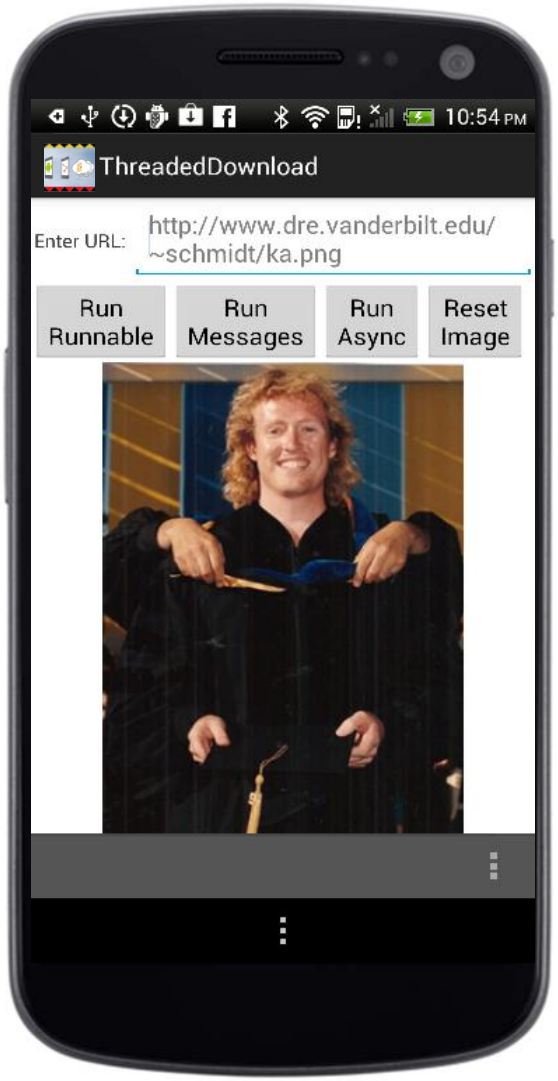
"Run Messages" Behavior for ThreadedDownloads

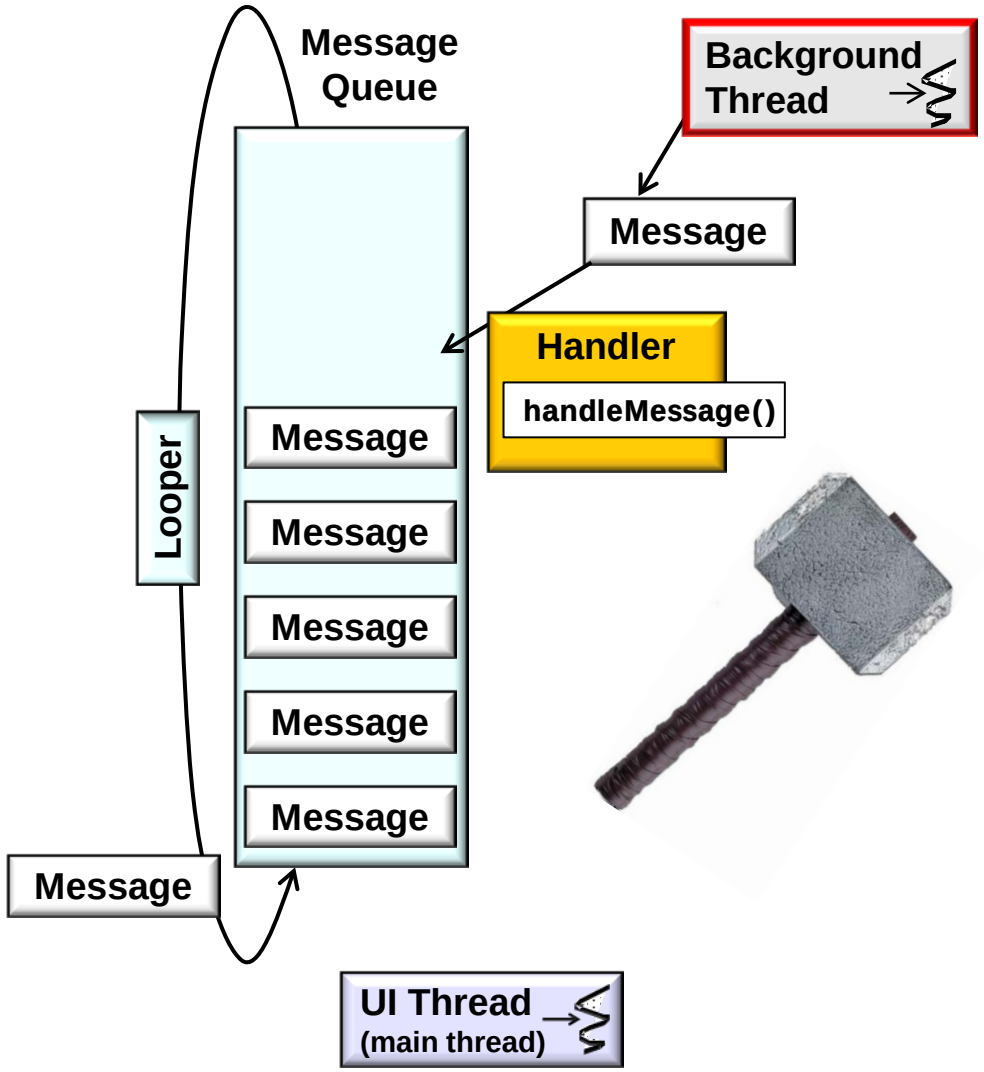




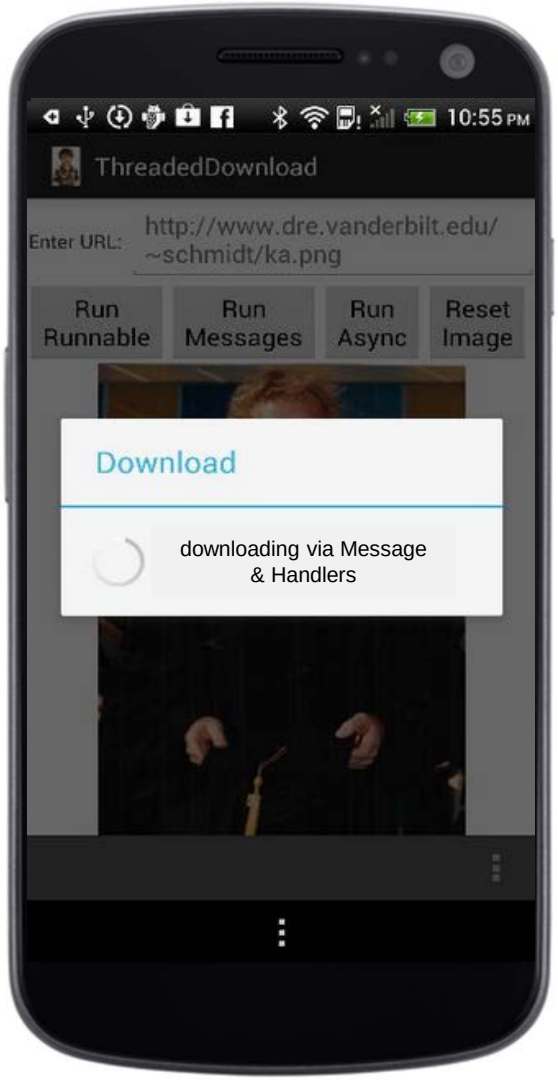
See earlier part on "Sending & Handling Messages in Android Handler"

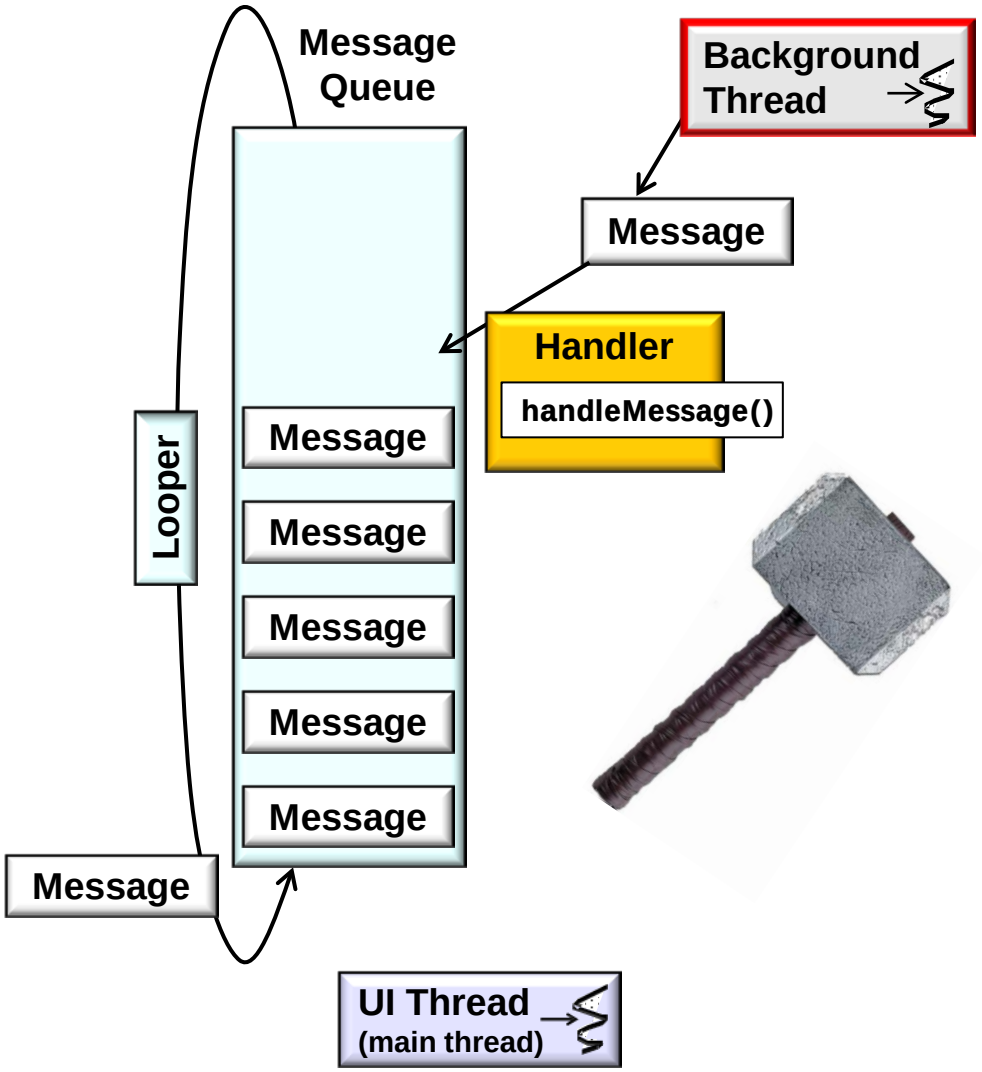
"Run Messages" Behavior for ThreadedDownloads



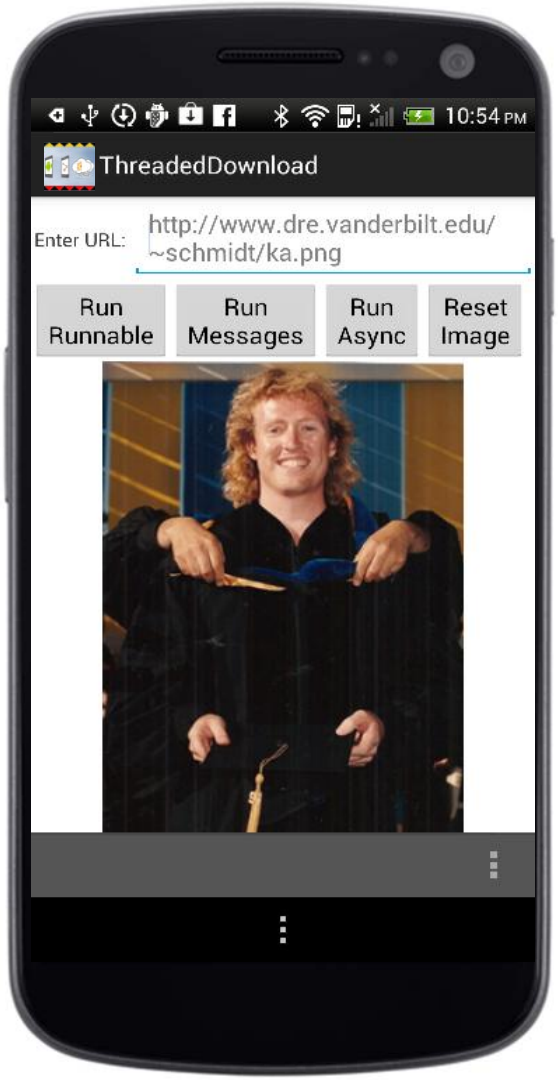


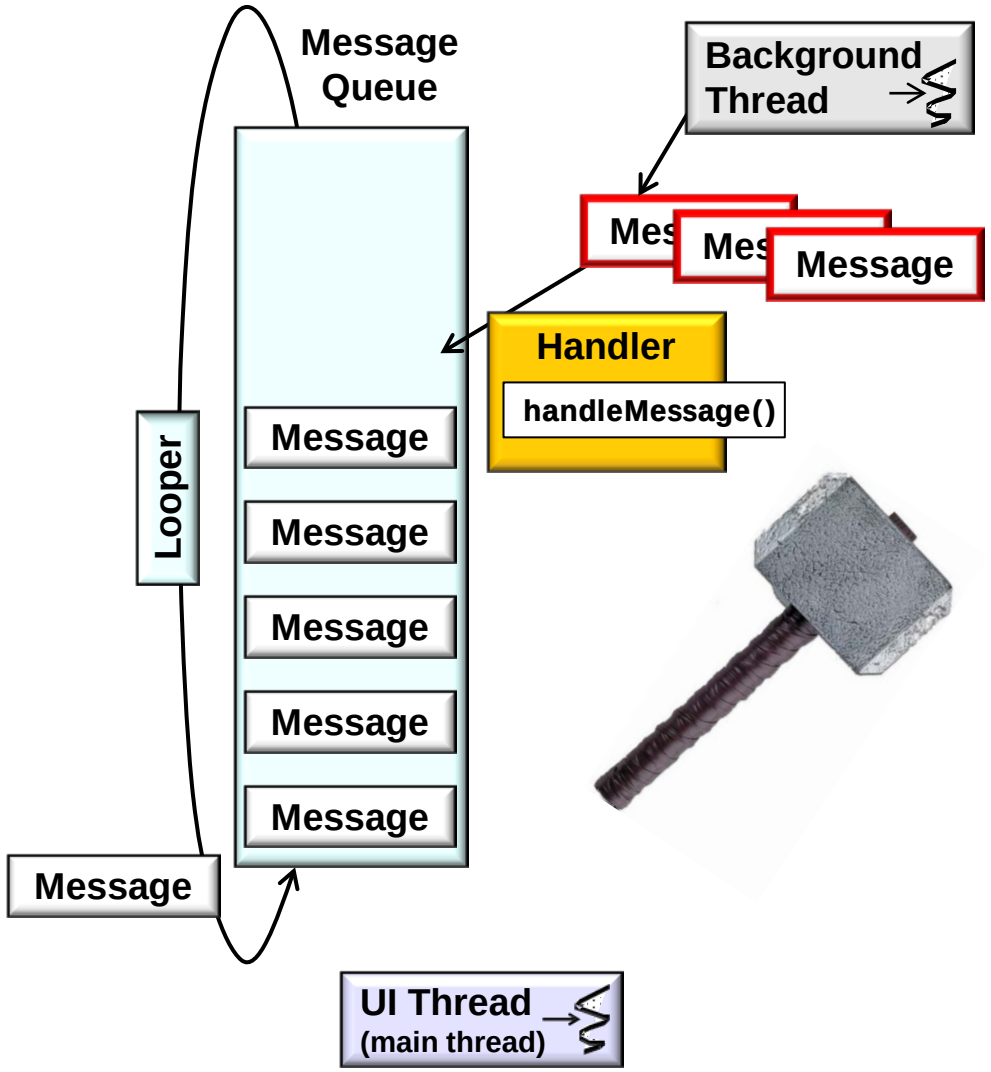
"Run Messages" Behavior for ThreadedDownloads



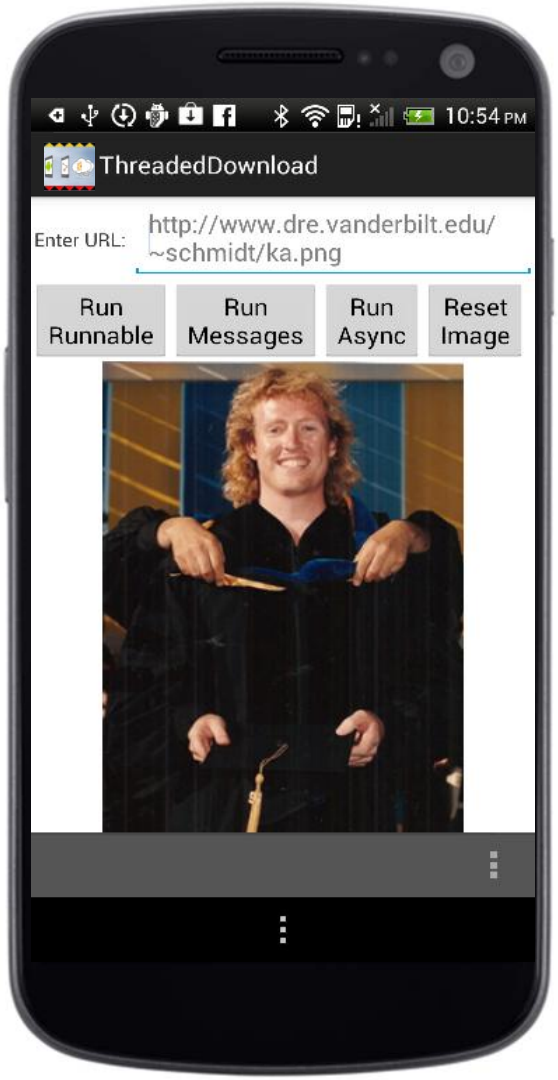


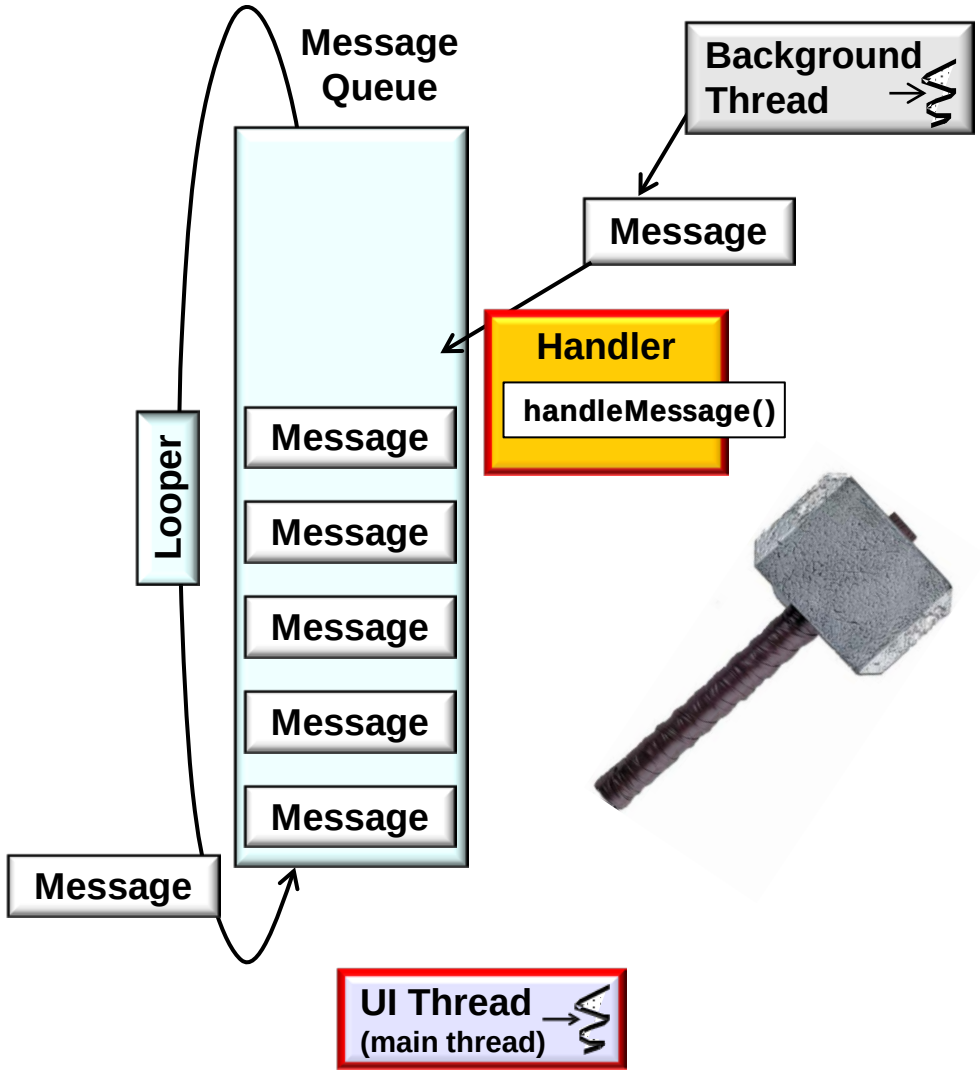
"Run Messages" Behavior for ThreadedDownloads



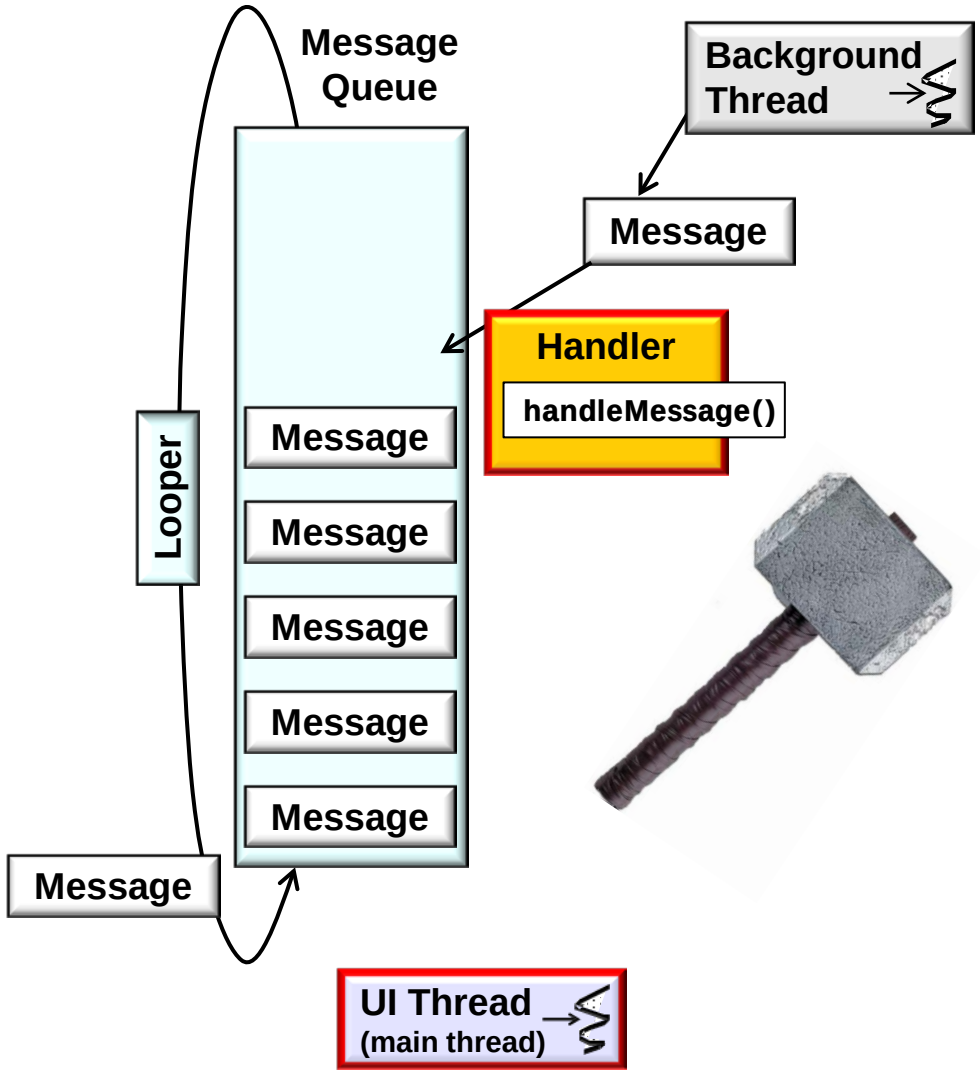


"Run Messages" Behavior for ThreadedDownloads

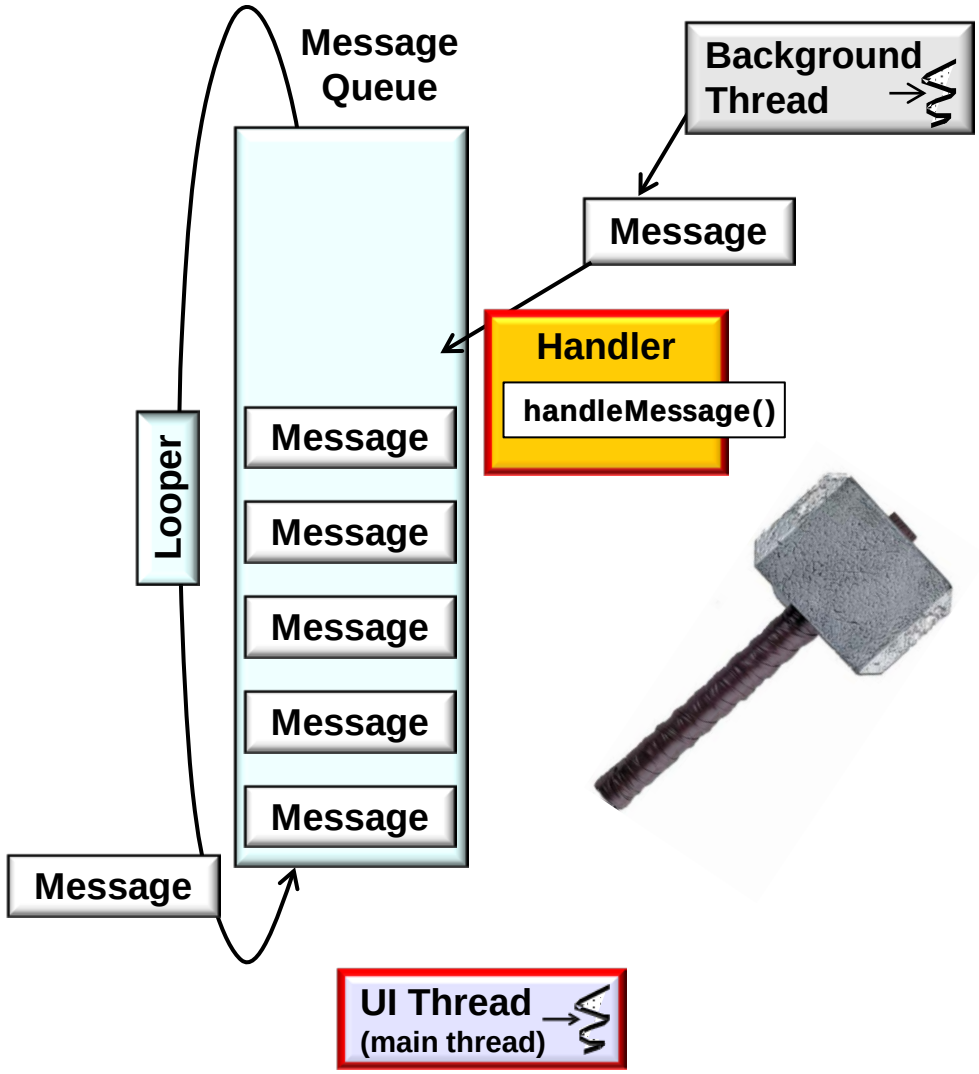




"Run Messages" Behavior for ThreadedDownloads



"Run Messages" Behavior for ThreadedDownloads



ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    static final int SHOW_DIALOG = 1;
    static final int DISMISS_DIALOG = 2;
    static final int DISPLAY_IMAGE = 3;
    ...
```

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    static final int SHOW_DIALOG = 1;
    static final int DISMISS_DIALOG = 2;
    static final int DISPLAY_IMAGE = 3;
    ...
```

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    static final int SHOW_DIALOG = 1;
    static final int DISMISS_DIALOG = 2;
    static final int DISPLAY_IMAGE = 3;
    ...
```

Types of Messages passed from a background Thread to UI Thread to specify which processing to perform

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...

    WeakReference<ThreadedDownloads> mActivity;


    MessageHandler(ThreadedDownloads activity) {
      mActivity = new WeakReference<ThreadedDownloads>
                                              (activity);
    }
```

Allows Activity to be garbage collected properly

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...

    WeakReference<ThreadedDownloads> mActivity;

    MessageHandler(ThreadedDownloads activity) {
      mActivity = new WeakReference<ThreadedDownloads>
                                              (activity);
    }
```

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...

    WeakReference<ThreadedDownloads> mActivity;

    MessageHandler(ThreadedDownloads activity) {
      mActivity = new WeakReference<ThreadedDownloads>
                                              (activity);
    }
```

Constructs mActivity as weak reference to the activity

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...
    public void handleMessage(Message msg) {
      ...
```

Process the Messages passed to the MessageHandler

This hook method runs in the context of the UI Thread

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...
    public void handleMessage(Message msg) {
      ...
      switch (msg.what) {
        ...
```

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...
    public void handleMessage(Message msg) {
      ...
      switch (msg.what) {
        case SHOW_DIALOG:
           mActivity.get().showDialog
             ("downloading via Handlers and Messages");
          break;
          ...
```

Start showing the process dialog

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...
    public void handleMessage(Message msg) {
      ...
      switch (msg.what) {
        case SHOW_DIALOG:
        ...
        case DISMISS_DIALOG:
          mActivity.get().dismissDialog();
          break;
        ...
```

Dismiss the process dialog

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
    ...
    public void handleMessage(Message msg) {
      ...
      switch (msg.what) {
        case SHOW_DIALOG:
        ...
        case DISMISS_DIALOG:
        ...
        case DISPLAY_IMAGE:
          mActivity.get().displayImage((Bitmap) msg.obj);
          break;
        ...
```

Display downloaded image to user

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
  }

  MessageHandler messageHandler = new MessageHandler(this);
  ...
```

Instance of MessageHandler holds a reference to the ThreadedDownloads Activity

ThreadedDownloads MessageHandler Class

- Handles Messages received from a background Thread

```
public class ThreadedDownloads extends Activity {
  ...
  private static class MessageHandler extends Handler {
  }

  MessageHandler messageHandler = new MessageHandler(this);
  ...
```

Instance of MessageHandler holds a reference to the ThreadedDownloads Activity

ThreadedDownloads runMessages() Method

- Called when a user clicks the “Run Messages” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runMessages(View view) {


    String url = getUrlString();


    new Thread(new RunnableWithMessages(url)).start();
  }
```

ThreadedDownloads runMessages() Method

- Called when a user clicks the “Run Messages” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runMessages(View view) {

    String url = getUrlString();

    new Thread(new RunnableWithMessages(url)).start();
  }
```

Obtain requested URL from user input

ThreadedDownloads runMessages() Method

- Called when a user clicks the “Run Messages” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runMessages(View view) {


    String url = getUrlString();


    new Thread(new RunnableWithMessages(url)).start();
  }
```

Create & start a new Thread to download an image in background & then use Messages & MessageHandler to cause it to be displayed in UI Thread

ThreadedDownloads runMessages() Method

- Called when a user clicks the “Run Messages” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runMessages(View view) {


    String url = getUrlString();


    new Thread(new RunnableWithMessages(url)).start();
  }
```

Create & start a new Thread to download an image in background & then use Messages & MessageHandler to cause it to be displayed in UI Thread

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    String mUrl;



    RunnableWithMessages(String url) {
      mUrl = url;
    }
    ...
```

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    String mUrl;



    RunnableWithMessages(String url) {
      mUrl = url;
    }
    ...
```

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    String mUrl;

    RunnableWithMessages(String url) {
      mUrl = url;
    }
    ...
```

URL to download

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      ...
```

Executes in a background Thread & sends Messages to the MessageHandler, which processes them in the UI Thread

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;
        ...
```

Store a copy of the reference to the MessageHandler

See mail.openjdk.java.net/pipermail/core-libs-dev/2010-May/004165.html

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;

      Message msg = mHandler.obtainMessage
                                    (MessageHandler.SHOW_DIALOG,
                                     mProgressDialog);
```

Create & send a Message to initiate the ProgressDialog

```
      mHandler.sendMessage(msg);
      ...
```

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;

      Message msg = mHandler.obtainMessage
                                    (MessageHandler.SHOW_DIALOG,
                                     mProgressDialog);
```

Create & send a Message to initiate the ProgressDialog

```
      mHandler.sendMessage(msg);
      ...
```

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;

      ...
      final Bitmap image = downloadImage(mUrl);
      ...
```

Retrieve the image from the remote server

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;

      ...
      msg = mHandler.obtainMessage
                                  (MessageHandler.DISMISS_DIALOG,
                                   mProgressDialog);
```

Create & send a Message to dismiss the ProgressDialog

```
      mHandler.sendMessage(msg);
      ...
```

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;

      ...
      msg = mHandler.obtainMessage
                                (MessageHandler.DISMISS_DIALOG,
                                 mProgressDialog);
```

Create & send a Message to dismiss the ProgressDialog

```
      mHandler.sendMessage(msg);
      ...
```

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;

      ...
      msg = mHandler.obtainMessage
                                  (MessageHandler.DISPLAY_IMAGE,
                                   image);
```

Create & send a Message to display the image

```
      mHandler.sendMessage(msg);
      ...
```

ThreadedDownloads RunnableWithMessages

- Downloads an image in a background Thread using a Handler & Messages

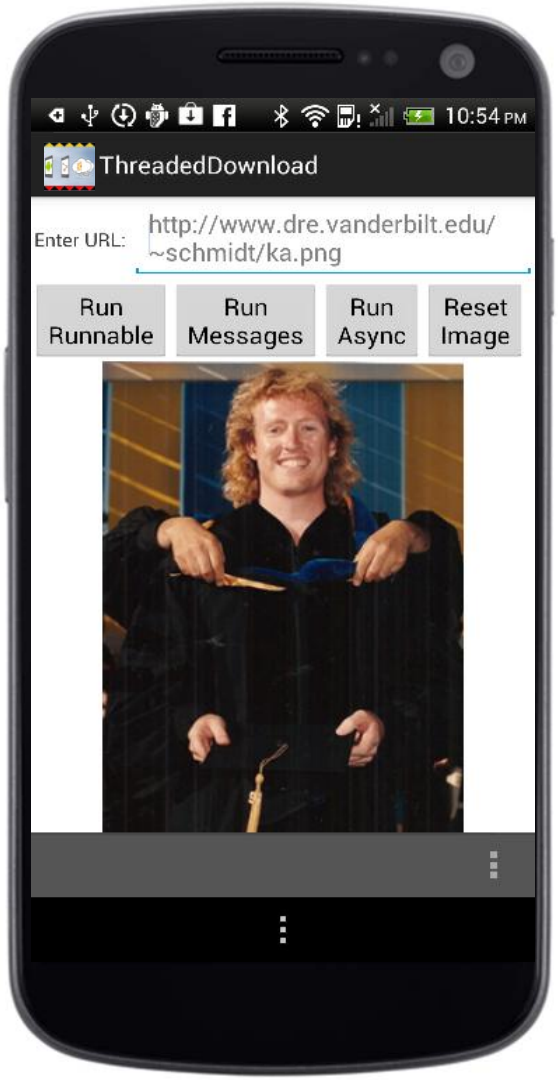
```
public class ThreadedDownloads extends Activity {
  ...
  private class RunnableWithMessages implements Runnable {
    public void run() {
      final MessageHandler mHandler =
        ThreadedDownloads.this.messageHandler;

      ...
      msg = mHandler.obtainMessage
                                (MessageHandler.DISPLAY_IMAGE,
                                 image);
```

Create & send a Message to display the image

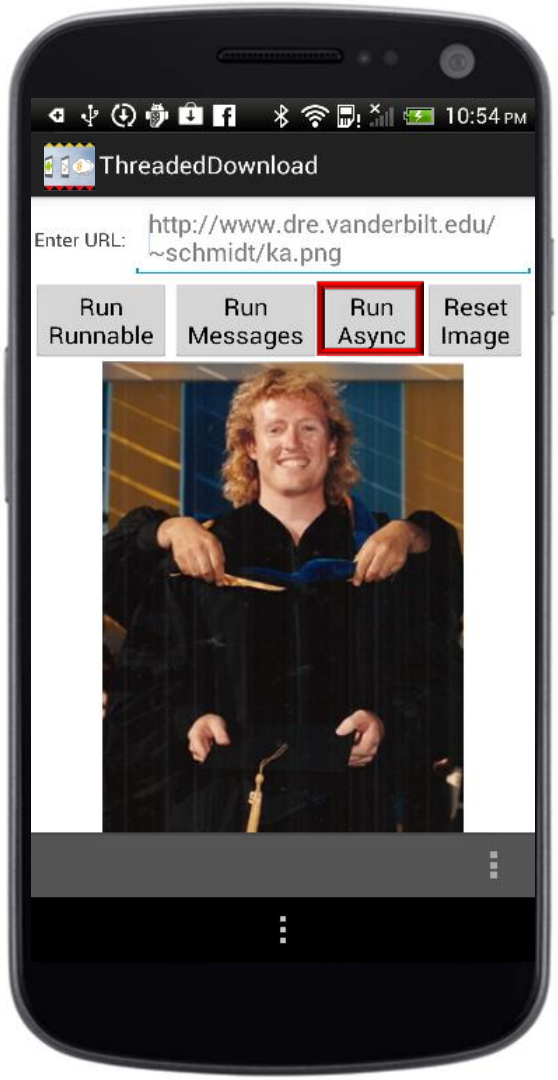
```
      mHandler.sendMessage(msg);
      ...
```

Programming with AsyncTask

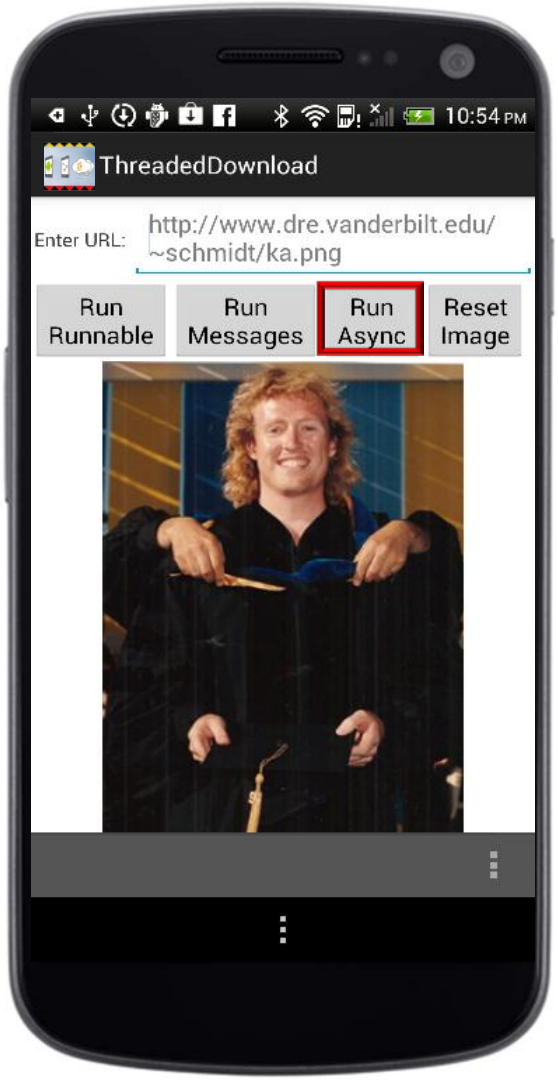
"Run Async" Behavior for ThreadedDownloads



"Run Async" Behavior for ThreadedDownloads

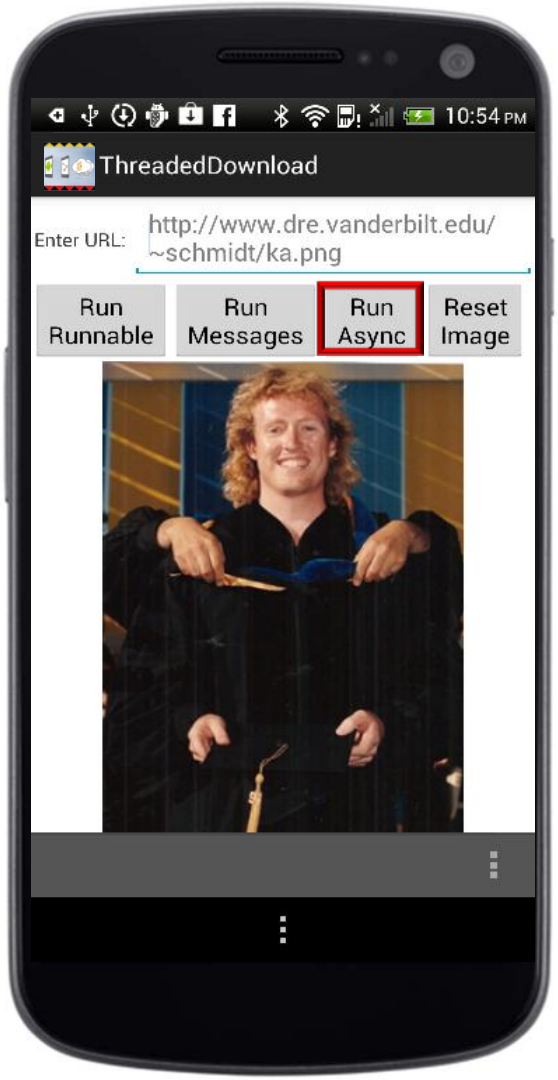


"Run Async" Behavior for ThreadedDownloads



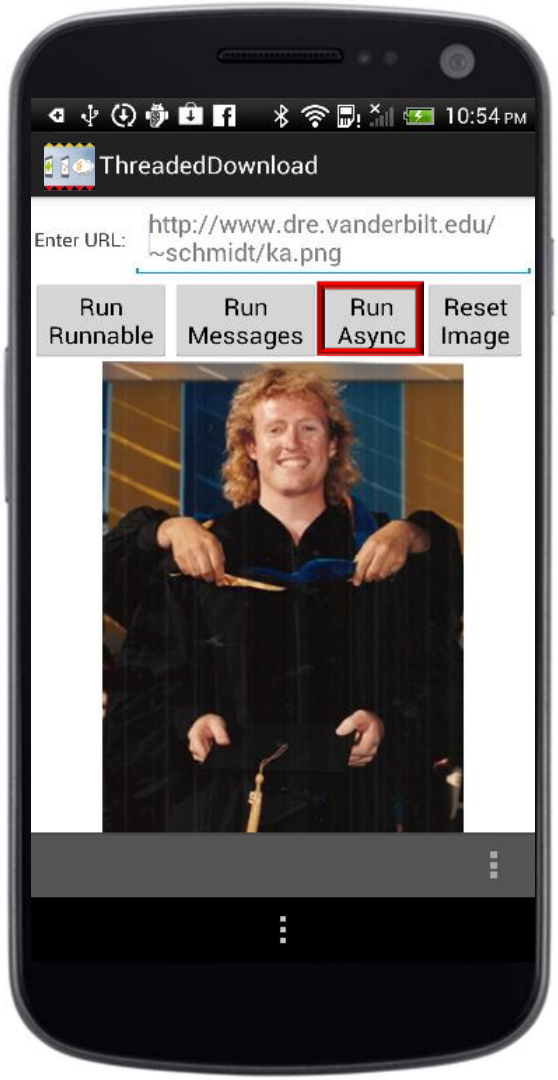

```
<Button
   ...
   android:onClick="runAsync"
   android:text="@string/runAsync" />
```

"Run Async" Behavior for ThreadedDownloads




```
<Button
   ...
   android:onClick="runAsync"
   android:text="@string/runAsync" />


public class ThreadedDownloads
             extends Activity {
   ...
   public void runRunnable
     (View view) {...}
   public void runMessages
     (View view) {...}
   public void runAsyncTask
     (View view) {...}
```

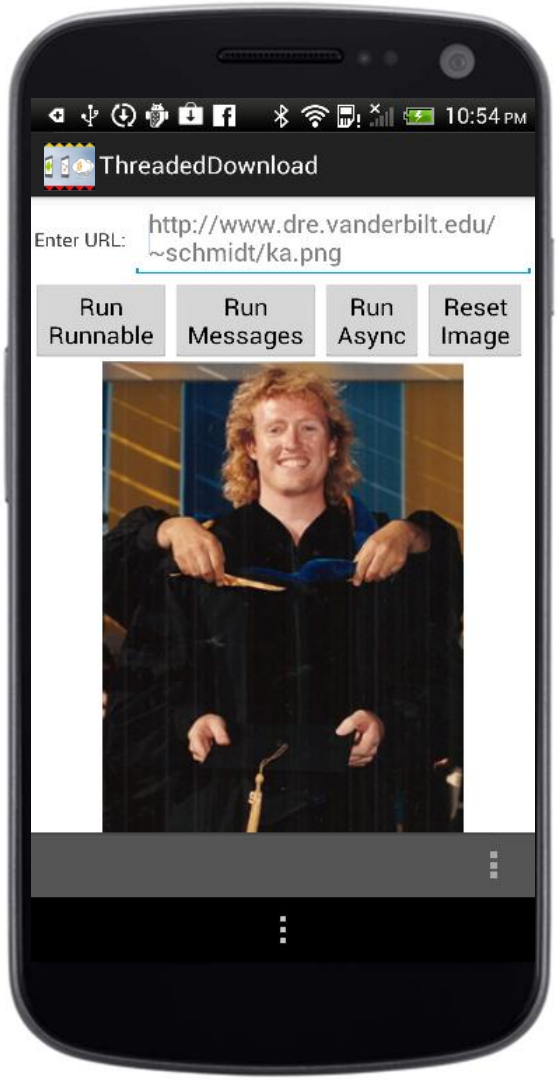
"Run Async" Behavior for ThreadedDownloads

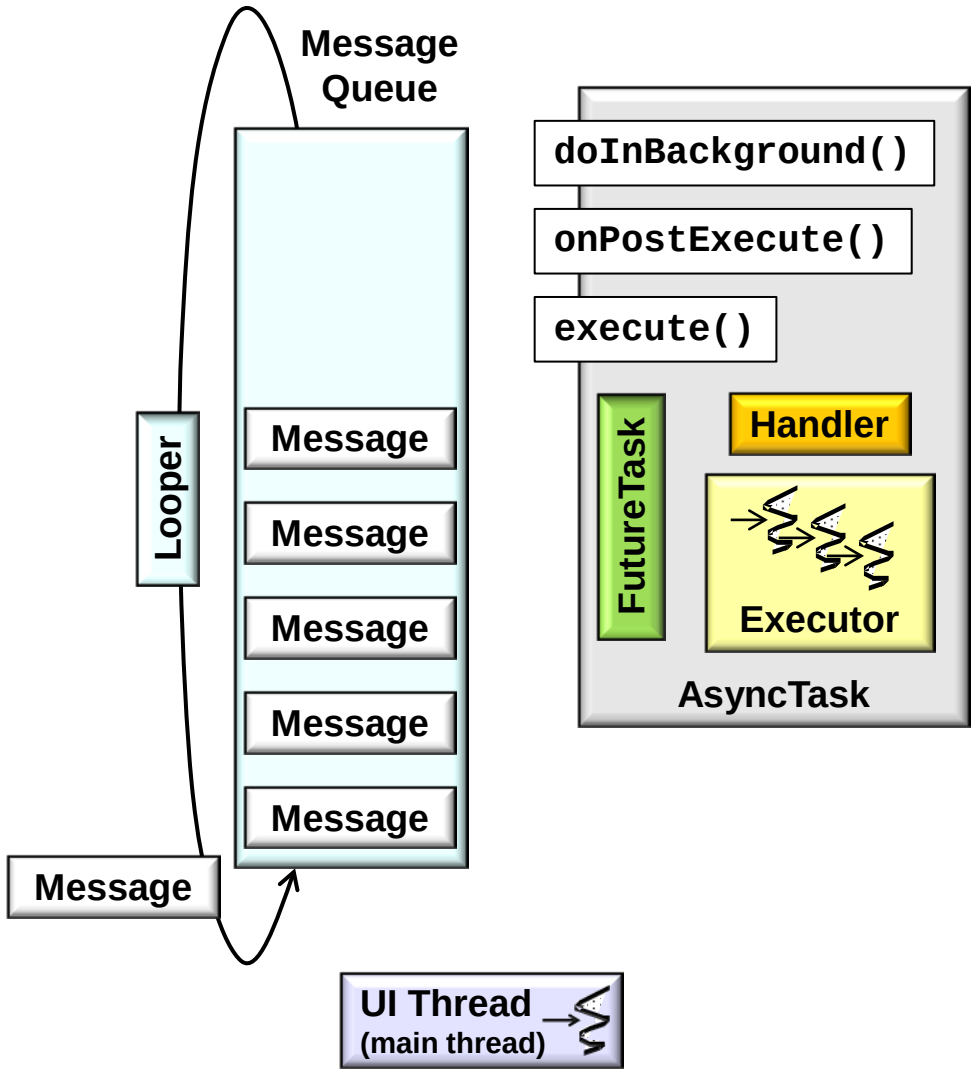



```
<Button
  ...
  android:onClick="runAsync"
  android:text="@string/runAsync" />


public class ThreadedDownloads
             extends Activity {
  ...
  public void runRunnable
    (View view) {...}
  public void runMessages
    (View view) {...}
  public void runAsyncTask
    (View view) {...}
```

ThreadedDownloads/src/edu/vuum/mocca/ThreadedDownloads.java has code

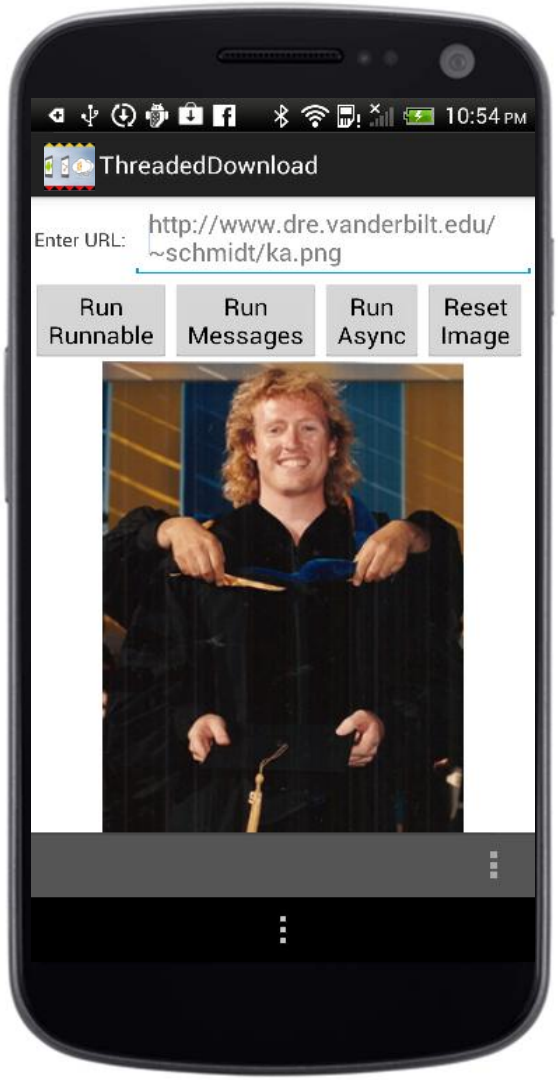
"Run Async" Behavior for ThreadedDownloads

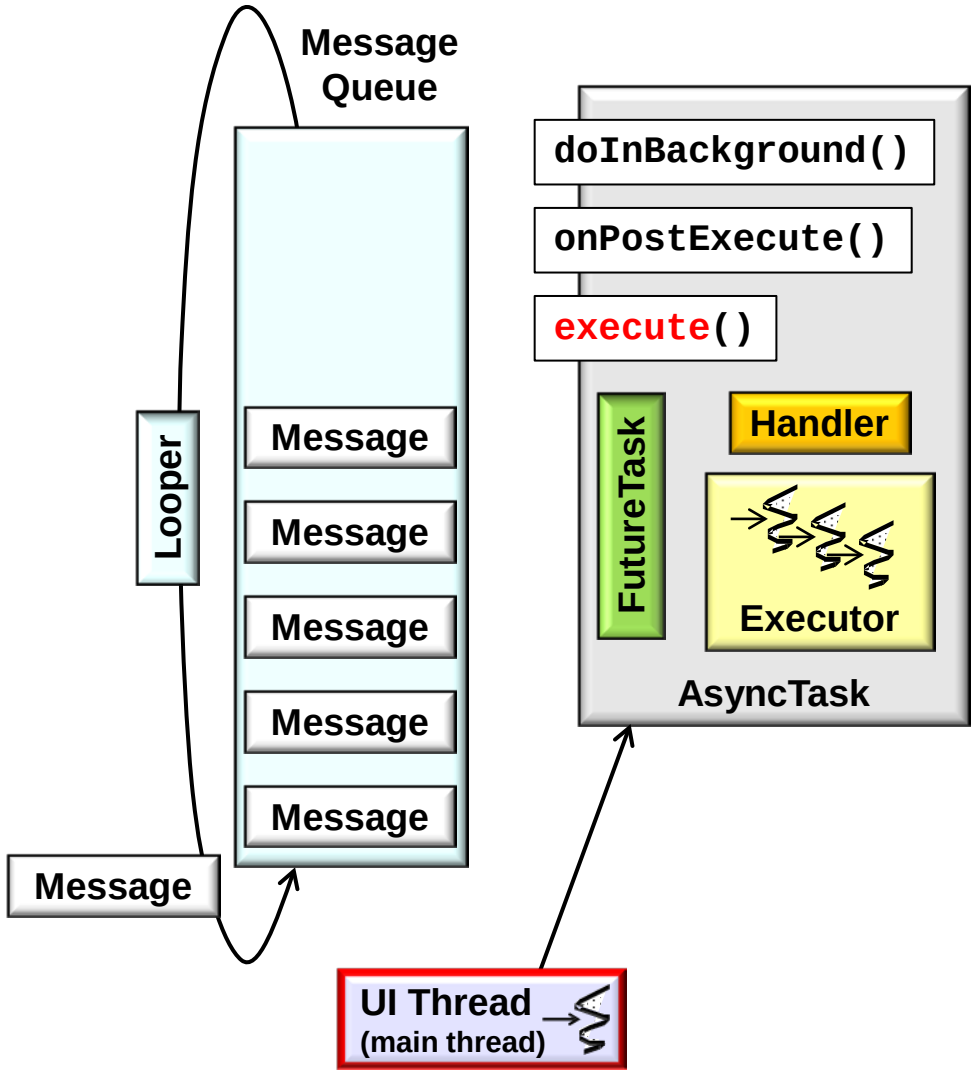




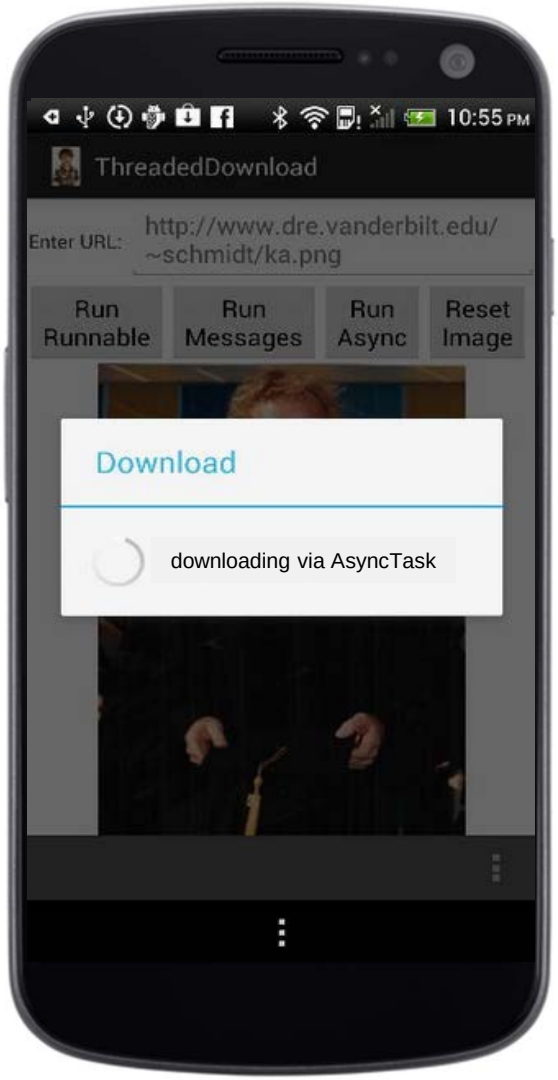
See earlier parts on "the AsyncTask Framework"

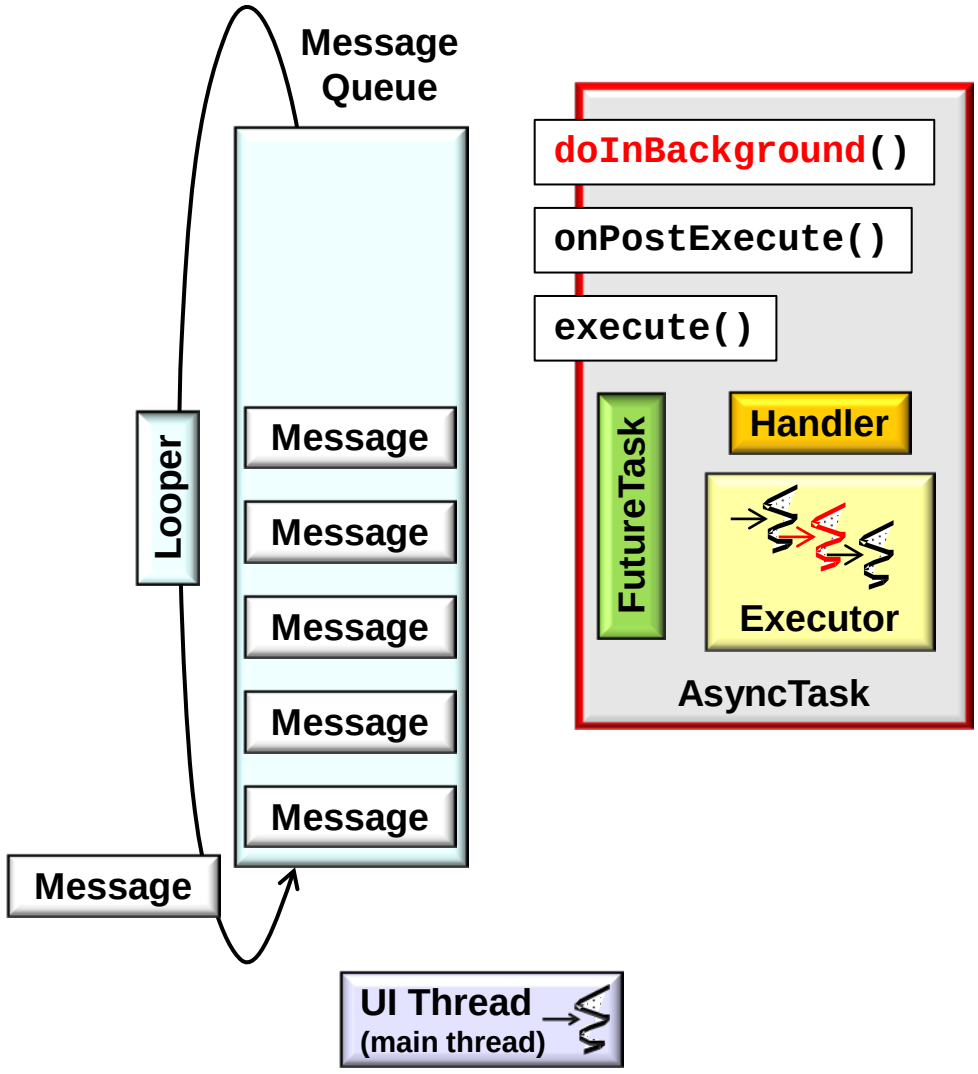
"Run Async" Behavior for ThreadedDownloads



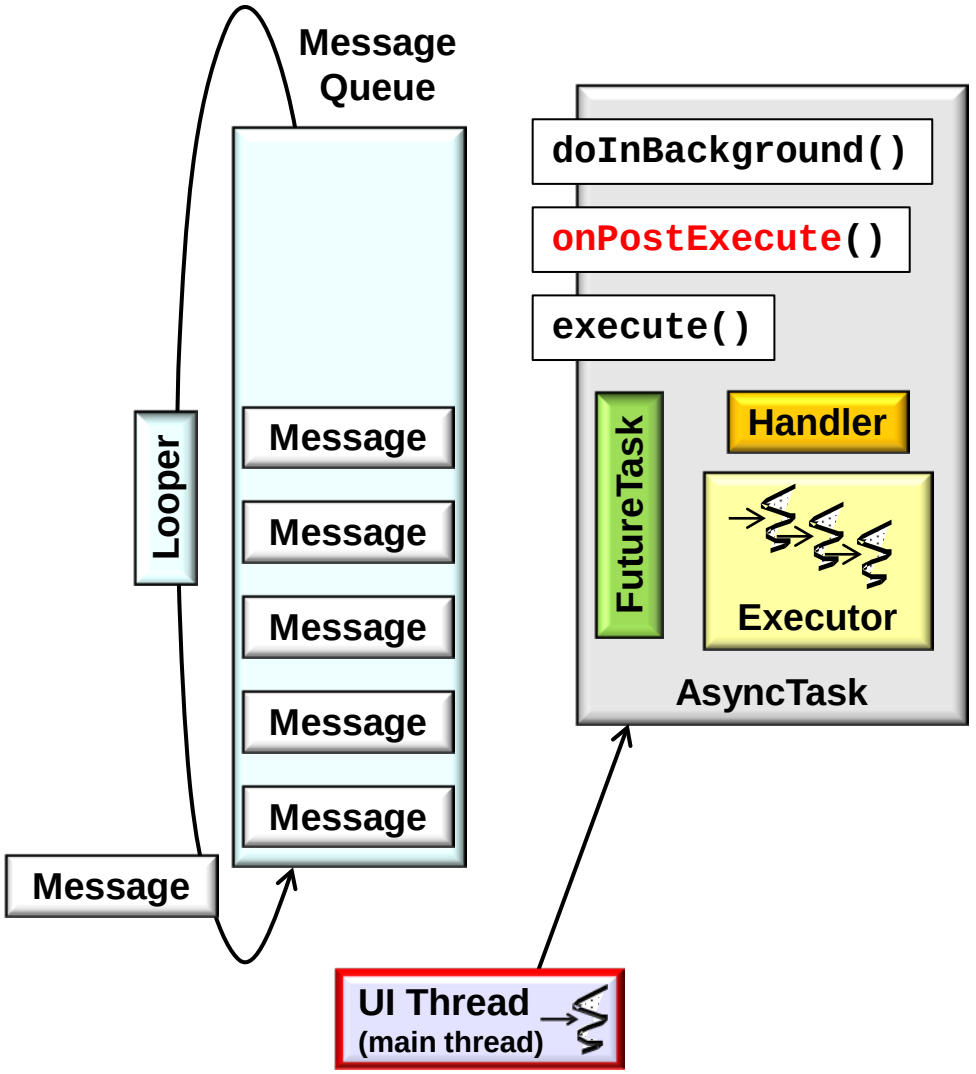


"Run Async" Behavior for ThreadedDownloads

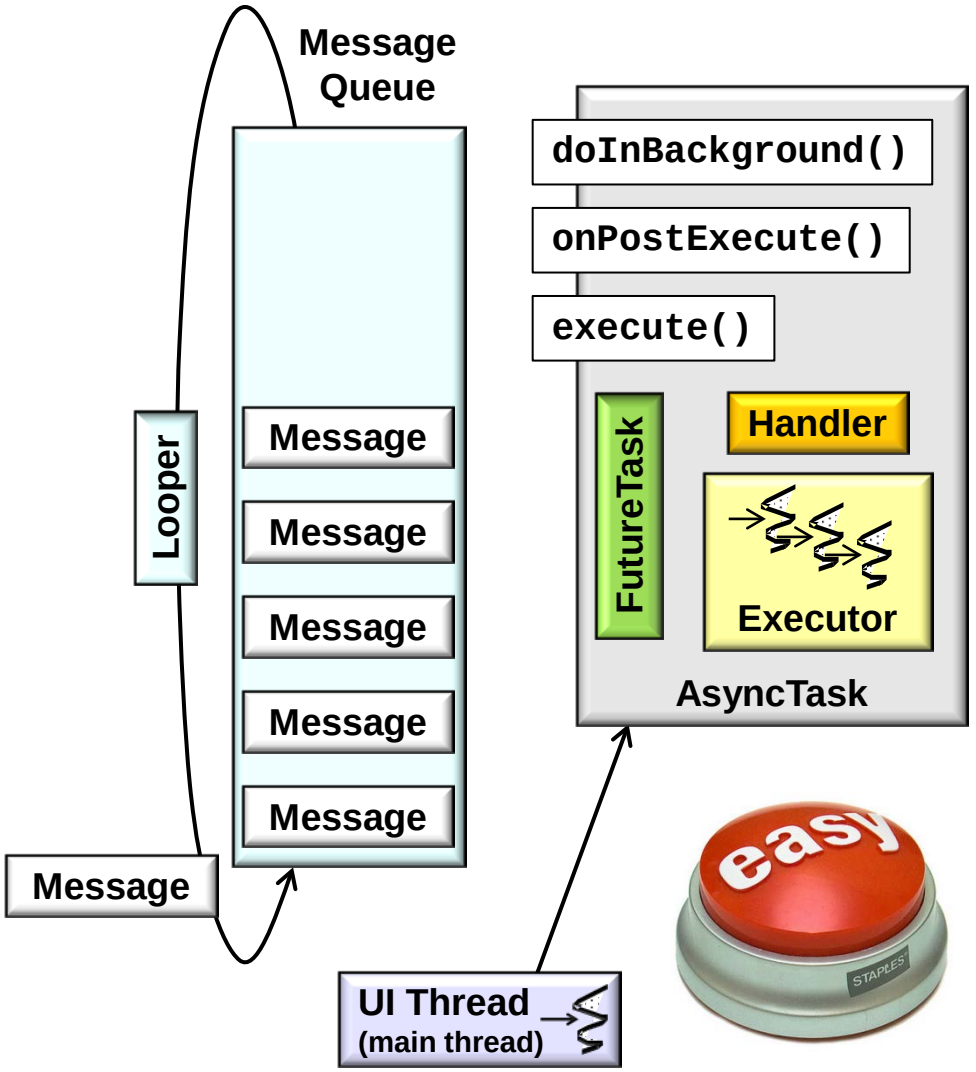




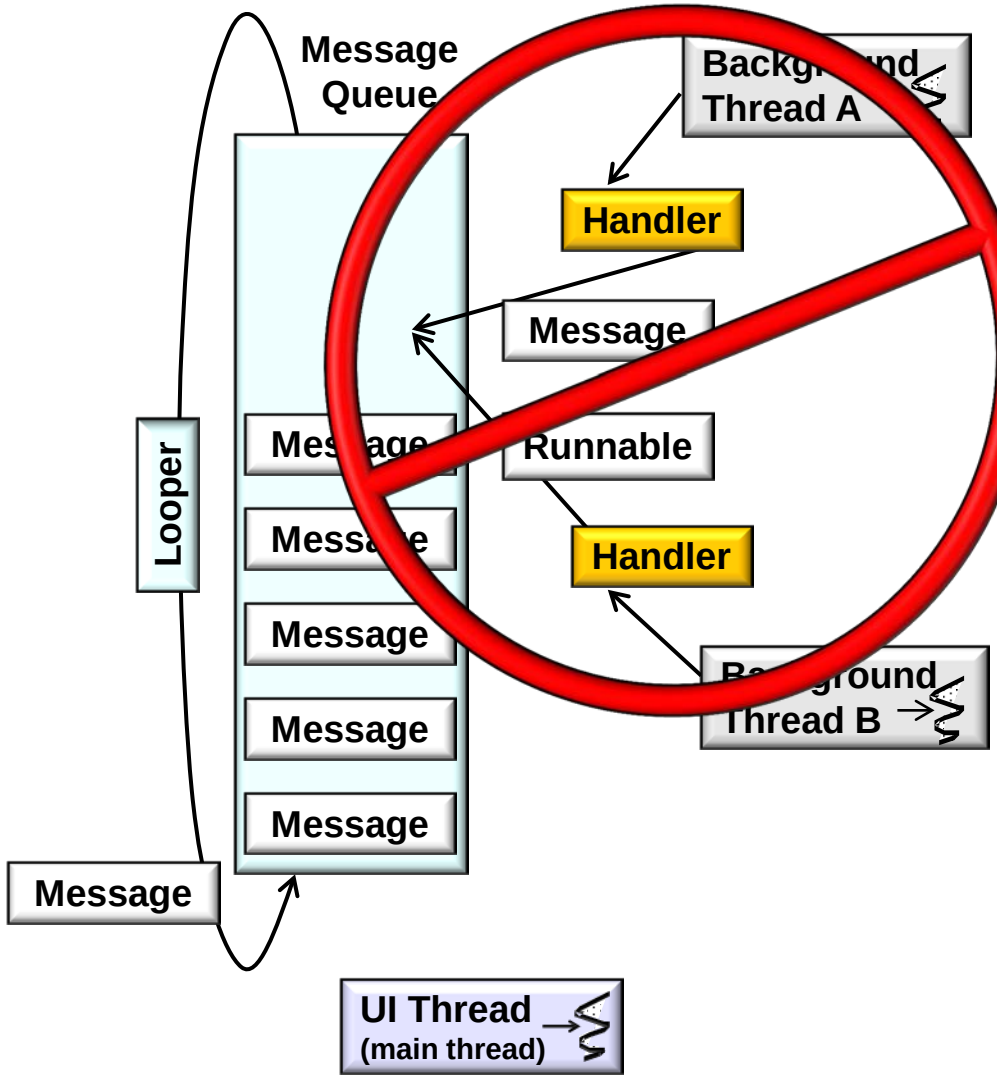
"Run Async" Behavior for ThreadedDownloads



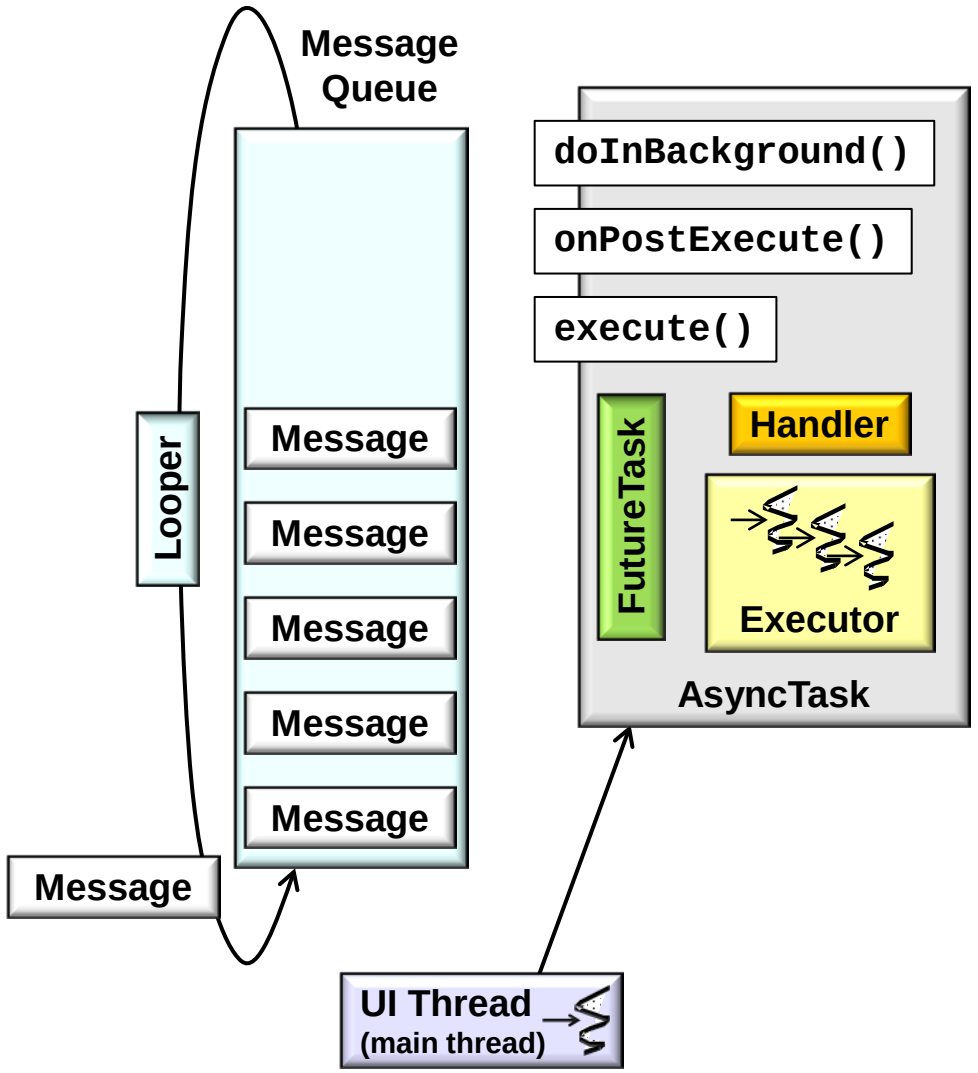
"Run Async" Behavior for ThreadedDownloads



"Run Async" Behavior for ThreadedDownloads

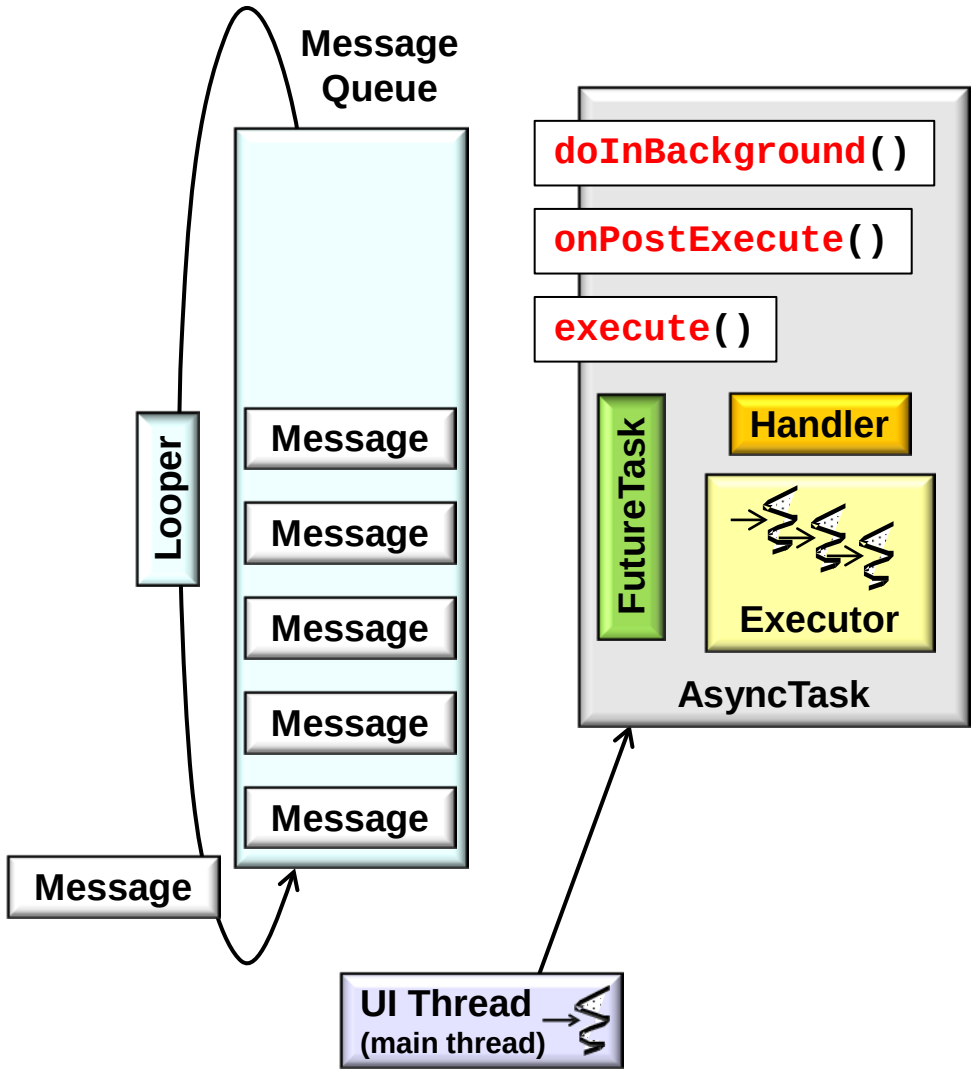


"Run Async" Behavior for ThreadedDownloads



en.wikipedia.org/wiki/Template_method_pattern has more info

"Run Async" Behavior for ThreadedDownloads



en.wikipedia.org/wiki/Template_method_pattern has more info

ThreadedDownloads runAsync() Method

- Called when a user clicks the “Run Async” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runAsync(View view) {


    String url = getUrlString();


    new DownloadTask().execute(url);
  }
```

ThreadedDownloads runAsync() Method

- Called when a user clicks the “Run Async” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runAsync(View view) {

    String url = getUrlString();

    new DownloadTask().execute(url);
  }
```

Obtain requested URL from user input

ThreadedDownloads runAsync() Method

- Called when a user clicks the “Run Async” button on the user interface

```
public class ThreadedDownloads extends Activity {
  ...
  public void runAsync(View view) {


    String url = getUrlString();


    new DownloadTask().execute(url);
  }
```

Create & executes the DownloadTask object to retrieve & display the requested image

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected Bitmap doInBackground(String... urls) { ... }


    protected void onPostExecute(Bitmap image) { ... }
    ...
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected Bitmap doInBackground(String... urls) { ... }

    protected void onPostExecute(Bitmap image) { ... }
    ...
```

Bitmap is used by hook methods

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected Bitmap doInBackground(String... urls) { ... }

    protected void onPostExecute(Bitmap image) { ... }
    ...
```

Bitmap is used by hook methods

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected void onPreExecute(Bitmap image) {
      showDialog("downloading via AsyncTask");
      ...
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected void onPreExecute(Bitmap image) {
      showDialog("downloading via AsyncTask");
      ...
```

Show the progress dialog before starting the download in a background Thread

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected Bitmap doInBackground(String... urls) {
      return downloadImage(urls[0]);
    }
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected Bitmap doInBackground(String... urls) {
      return downloadImage(urls[0]);
    }
```

Download the requested image & return it to the AsyncTask framework

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected void onPostExecute(Bitmap image) {
      dismissDialog();



      displayImage(image);
   }
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected void onPostExecute(Bitmap image) {
      dismissDialog();



      displayImage(image);
   }
```

ThreadedDownloads DownloadTask

- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected void onPostExecute(Bitmap image) {
      dismissDialog();



      displayImage(image);
   }
```

ThreadedDownloads DownloadTask

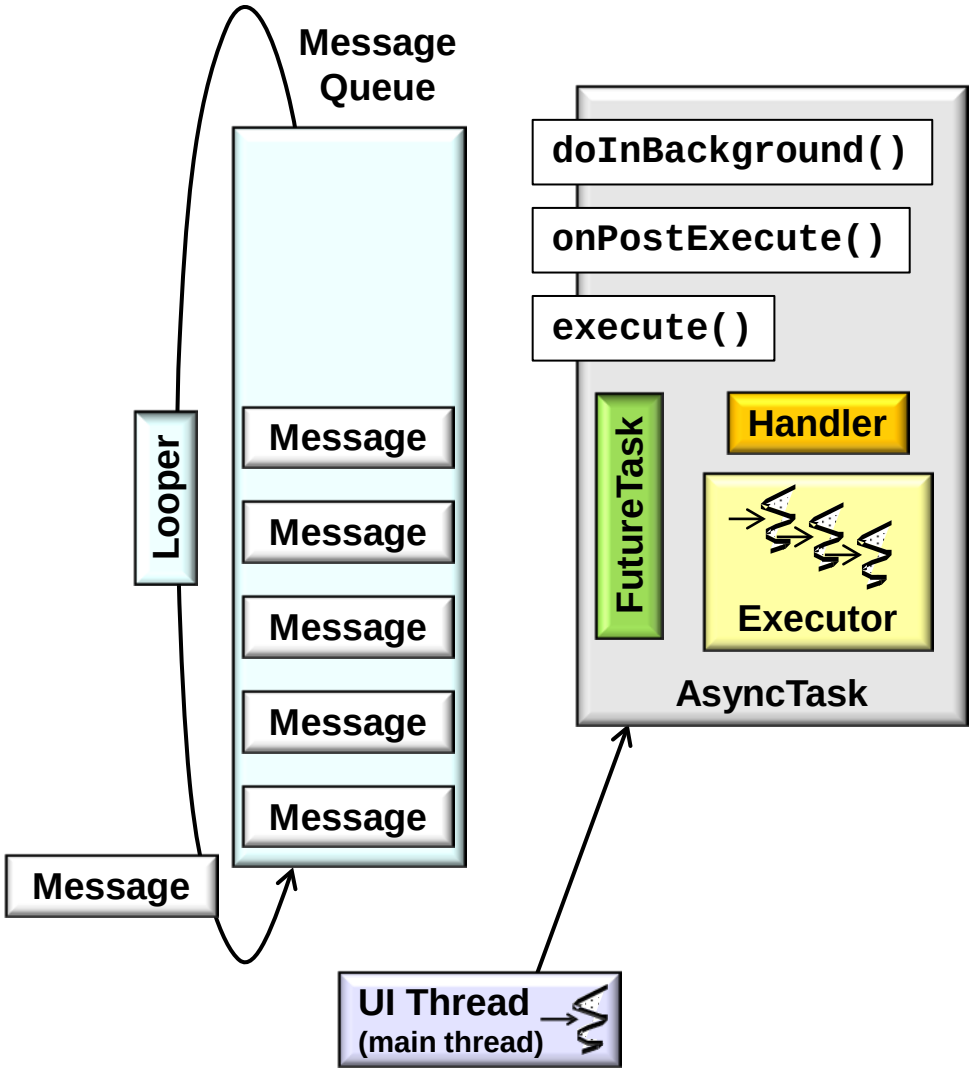
- Downloads an image in a background Thread using AsyncTask

```
public class ThreadedDownloads extends Activity {
  ...
  private class DownloadTask
                extends AsyncTask<String, Integer, Bitmap> {
    ...
    protected void onPostExecute(Bitmap image) {
      dismissDialog();



      displayImage(image);
   }
```

Display the image to the user

ThreadedDownloads DownloadTask

- Switching of contexts between the UI Thread & background Thread is handled seamlessly & transparently by the AsyncTask framework



Comparing & Contrasting the Three Concurrency Model Solutions

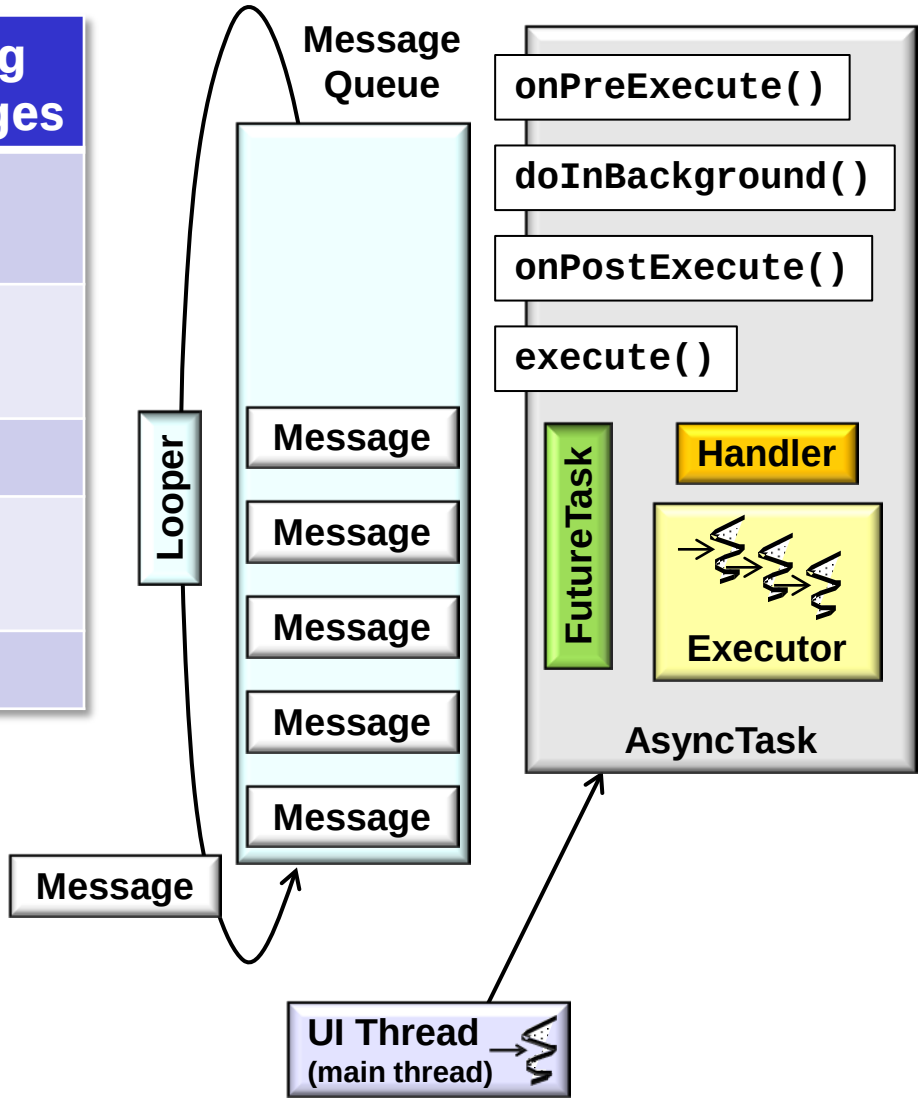
Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

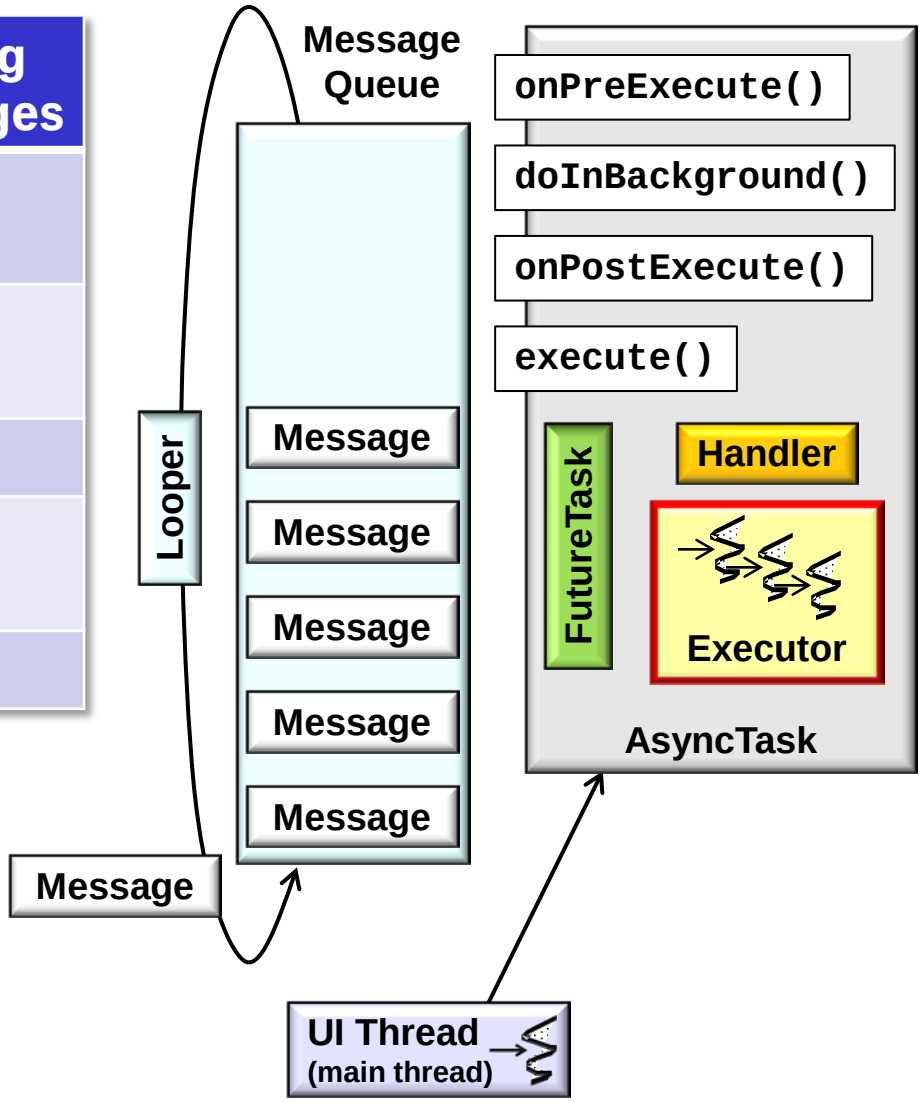
- Simplifies development of both simply & more complicated concurrent applications



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

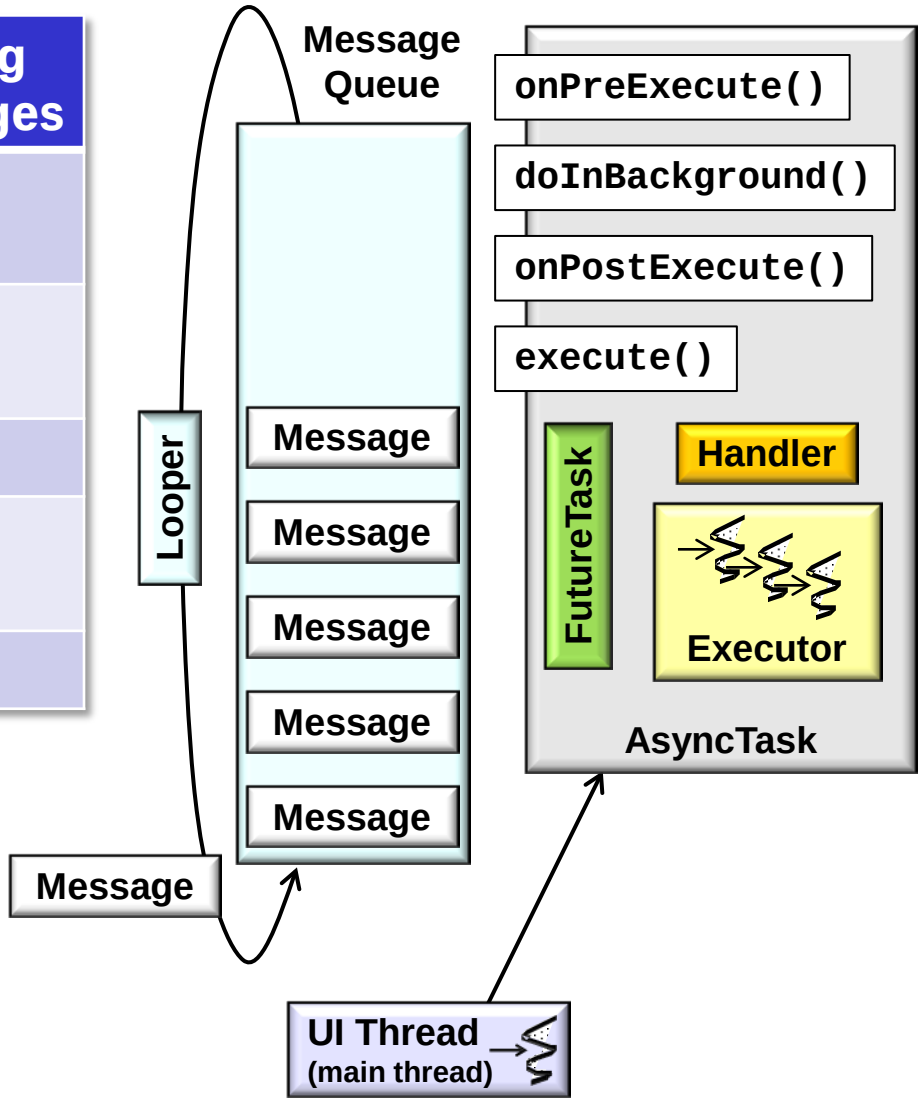
- Enables relatively transparent scalability via its Thread Pool Executor



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

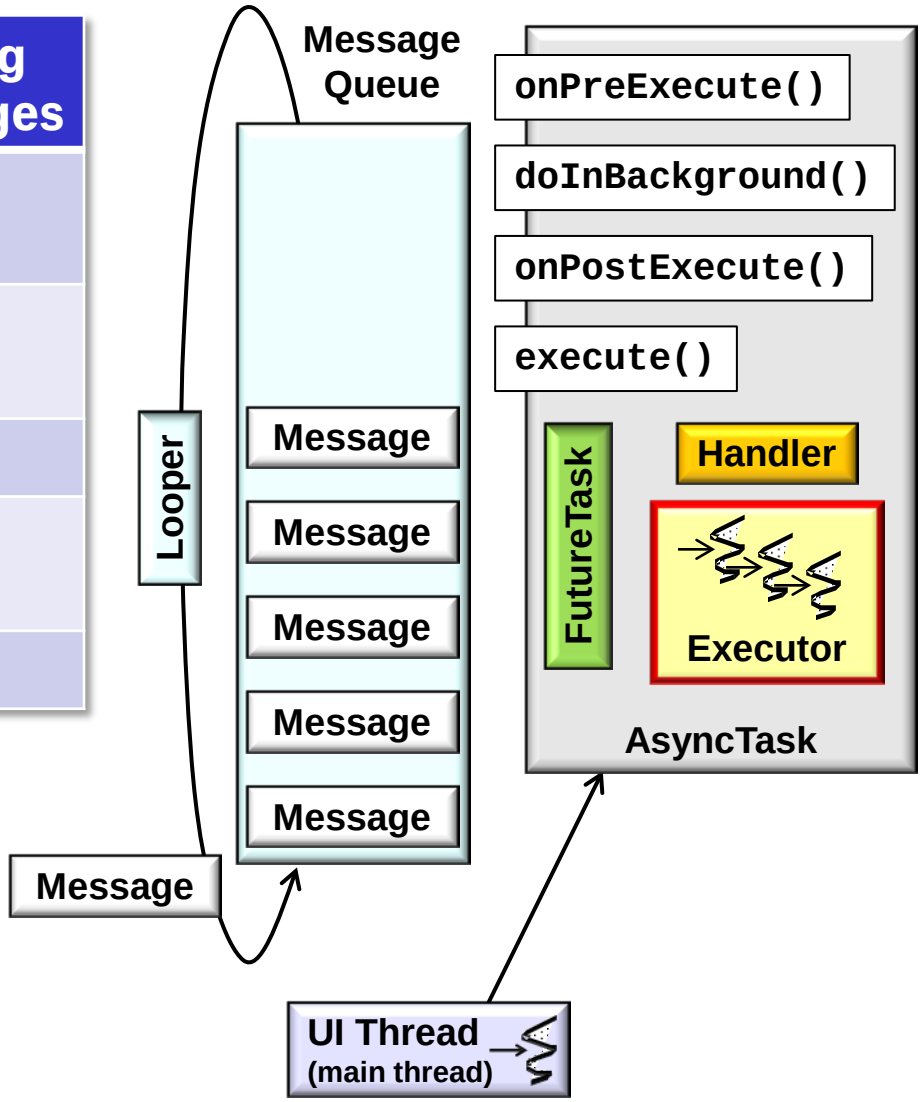
- Only intended for interactions between UI Thread & background Threads
  - But not interactions between background Threads



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

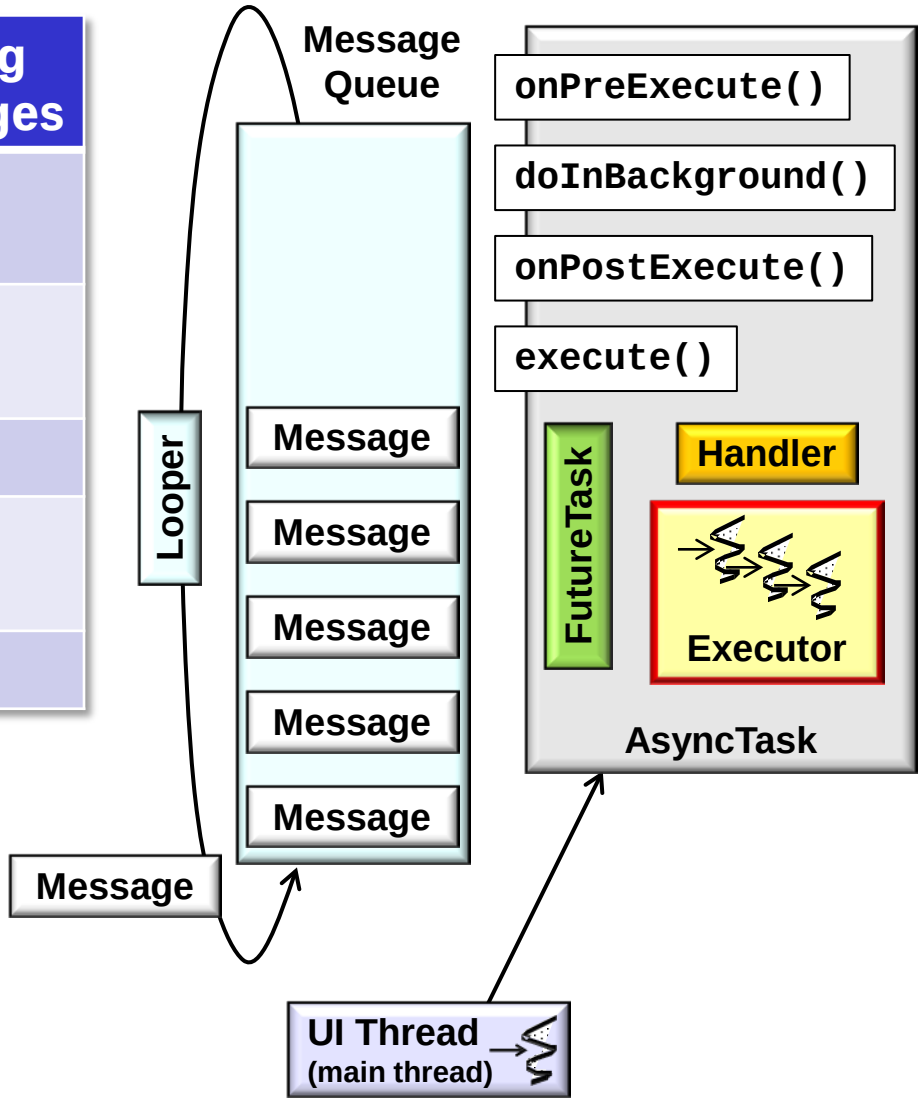
- Incurs higher overhead due to



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

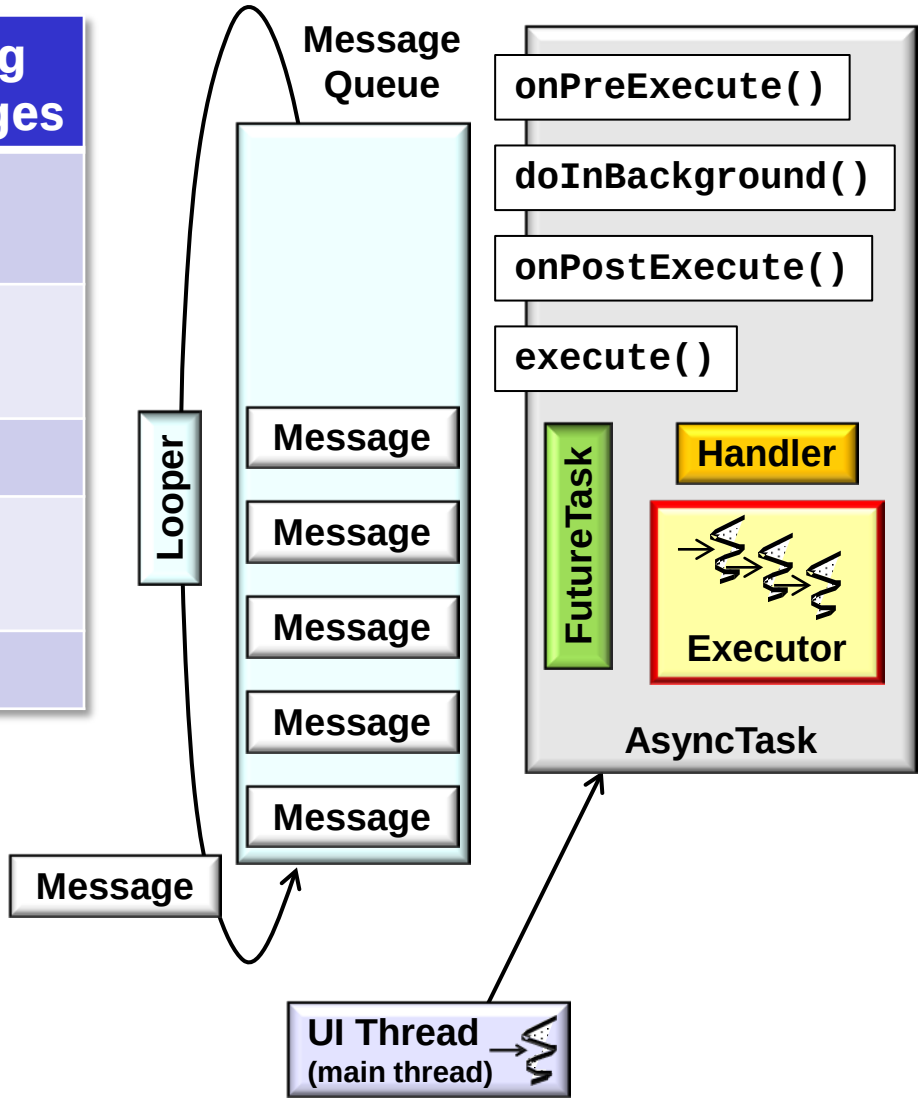
- Incurs higher overhead due to
  - Extra levels of indirection &



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

- Incurs higher overhead due to
  - Extra levels of indirection &
  - Inter-Thread communication costs
    - e.g., synchronization context switching, & data movement

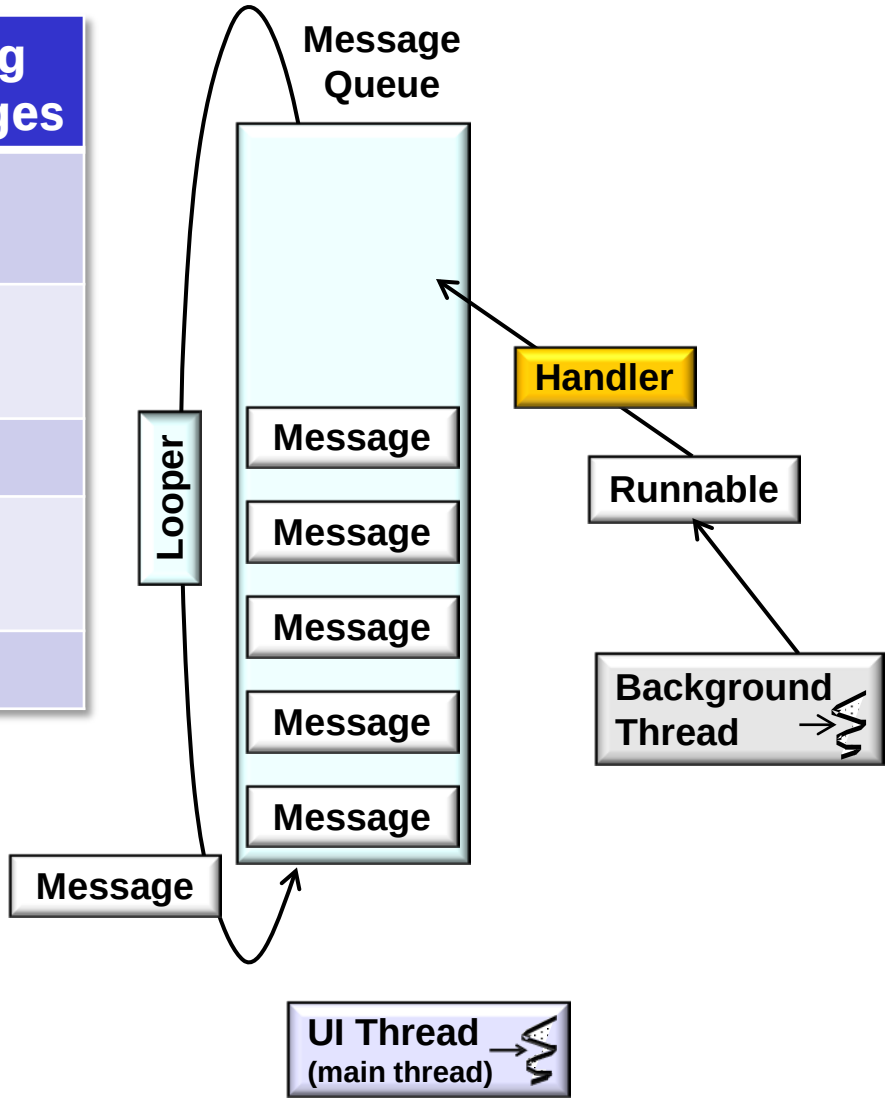


www.dre.vanderbilt.edu/~schmidt/PDF/INFOCOM-94.pdf has more info

Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	□□□	□□□	□□
Usability (Complex)	□□□	□	□□
Scalability	□□□	□	□
Flexibilty	□□	□	□□□
Efficiency	□□	□□□	□□□

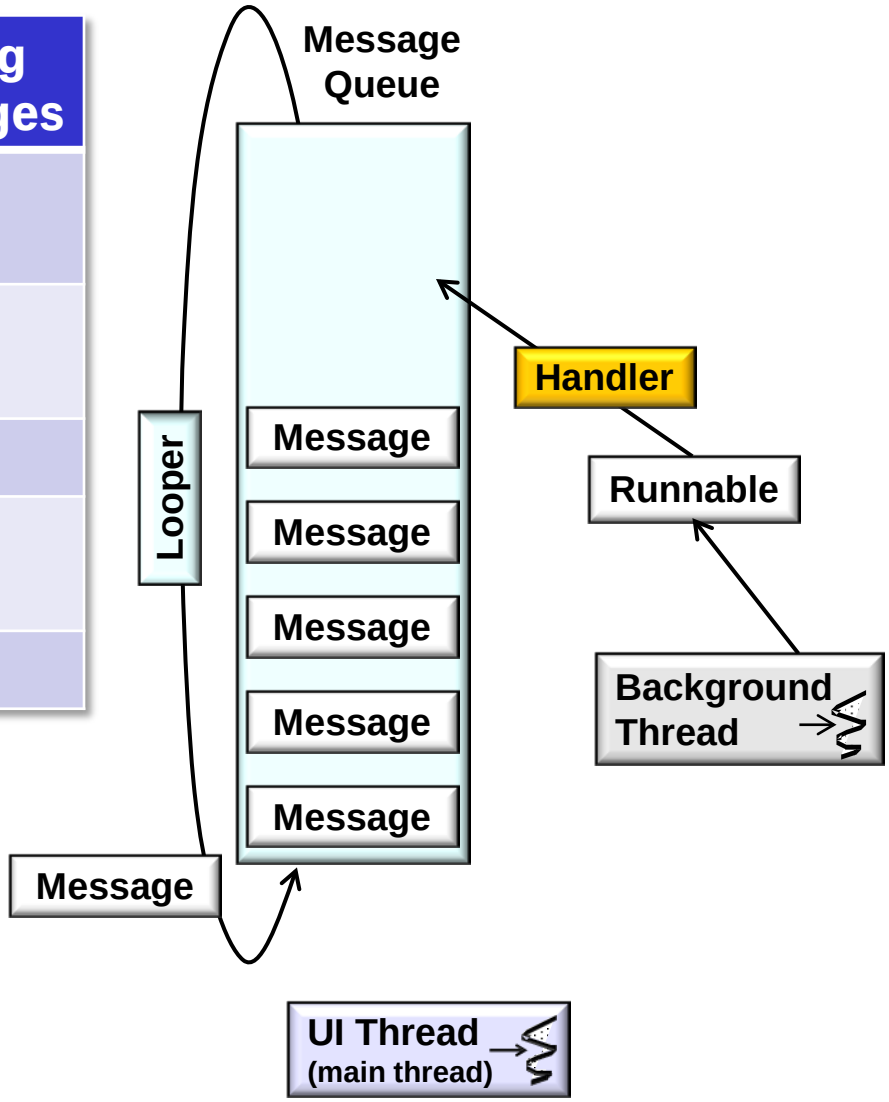
- Efficient & easy to use for simple use cases



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

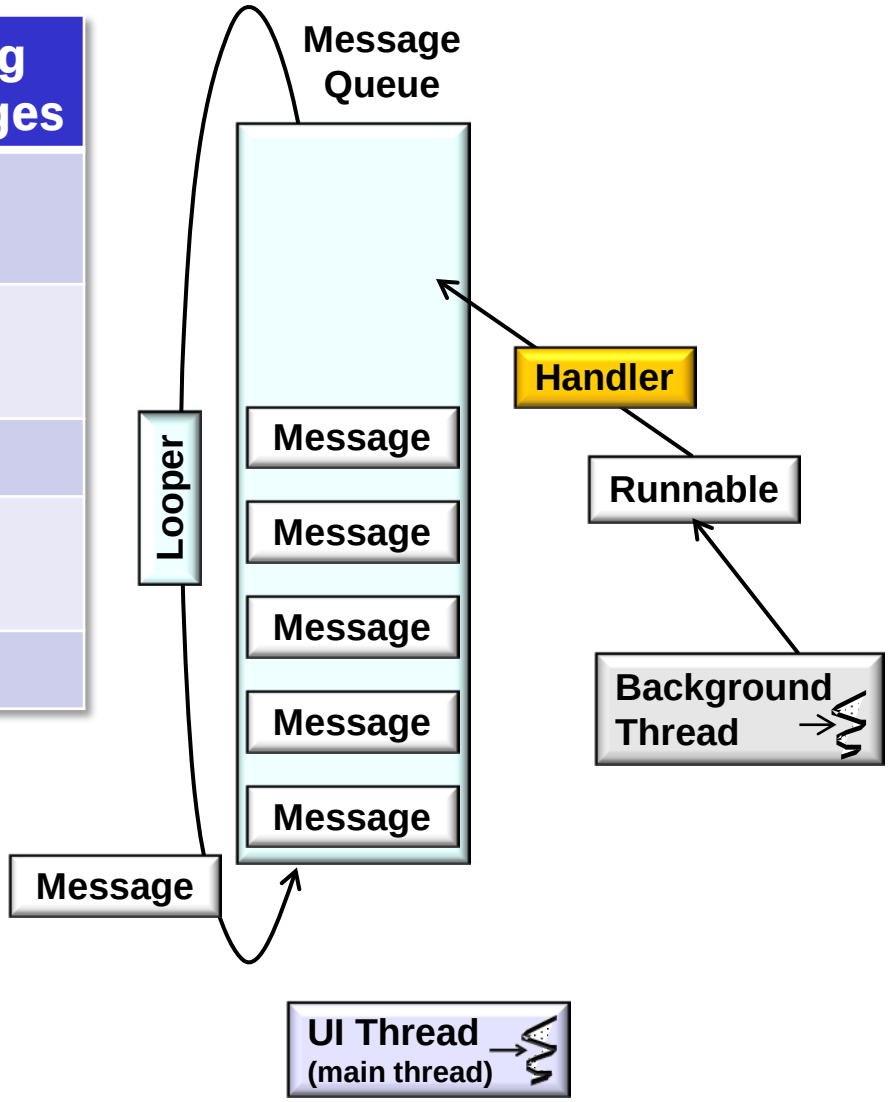
- Less flexibly & easy to use for more complex use cases



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

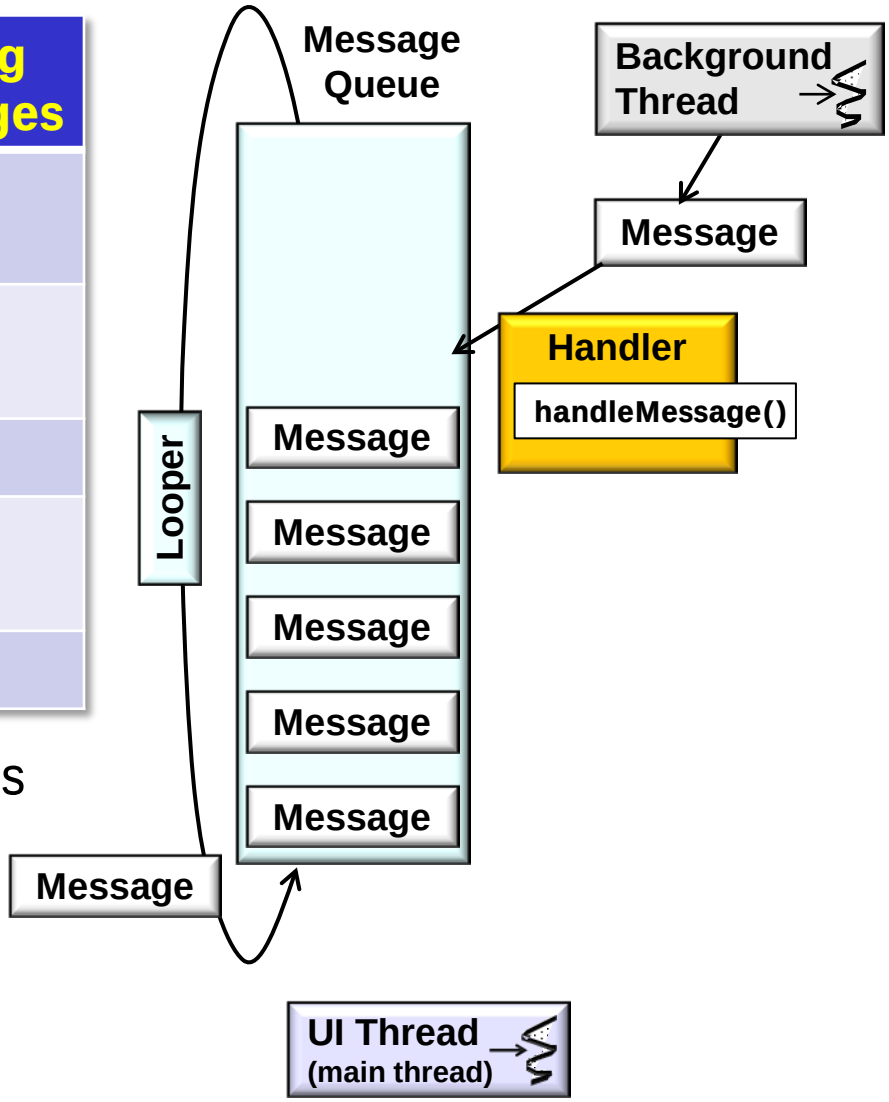
- Thread pools must be managed manually, which is hard to scale



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

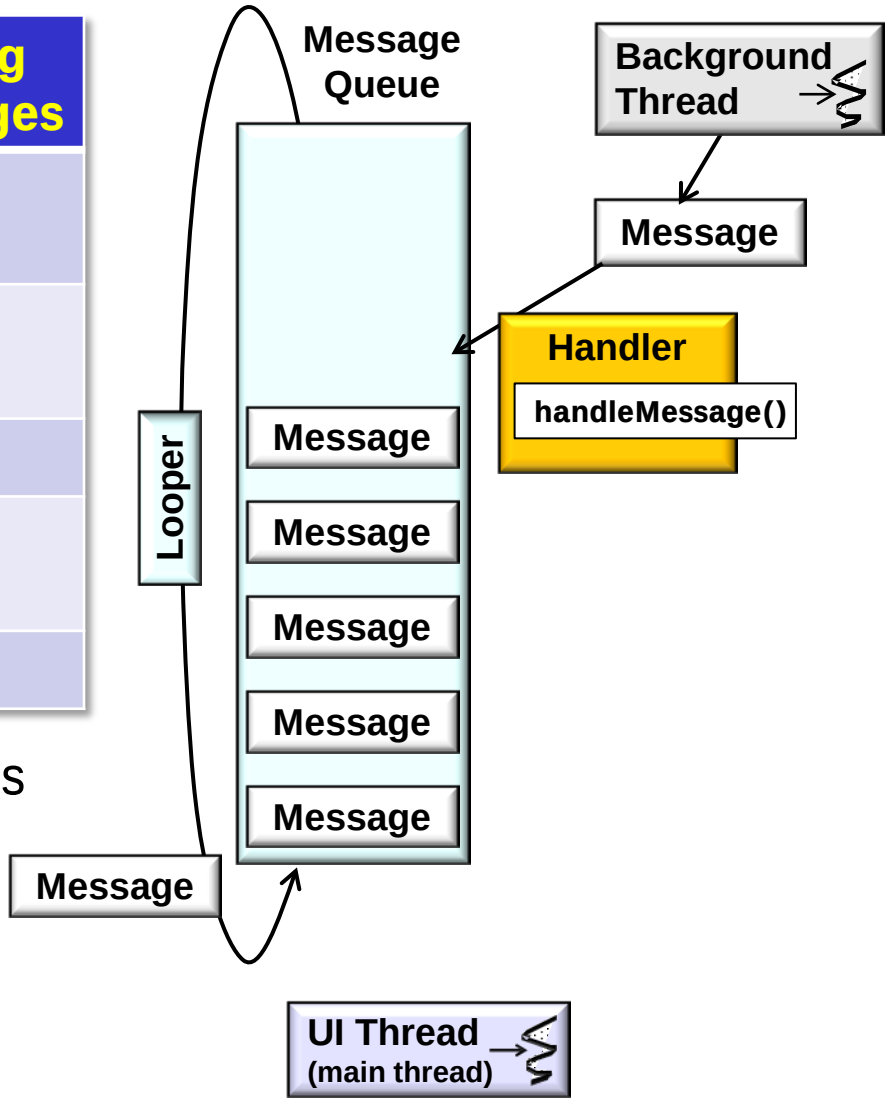
- Flexible & efficient for passing Messages between various types of Threads



Comparing & Contrasting the Three Solutions

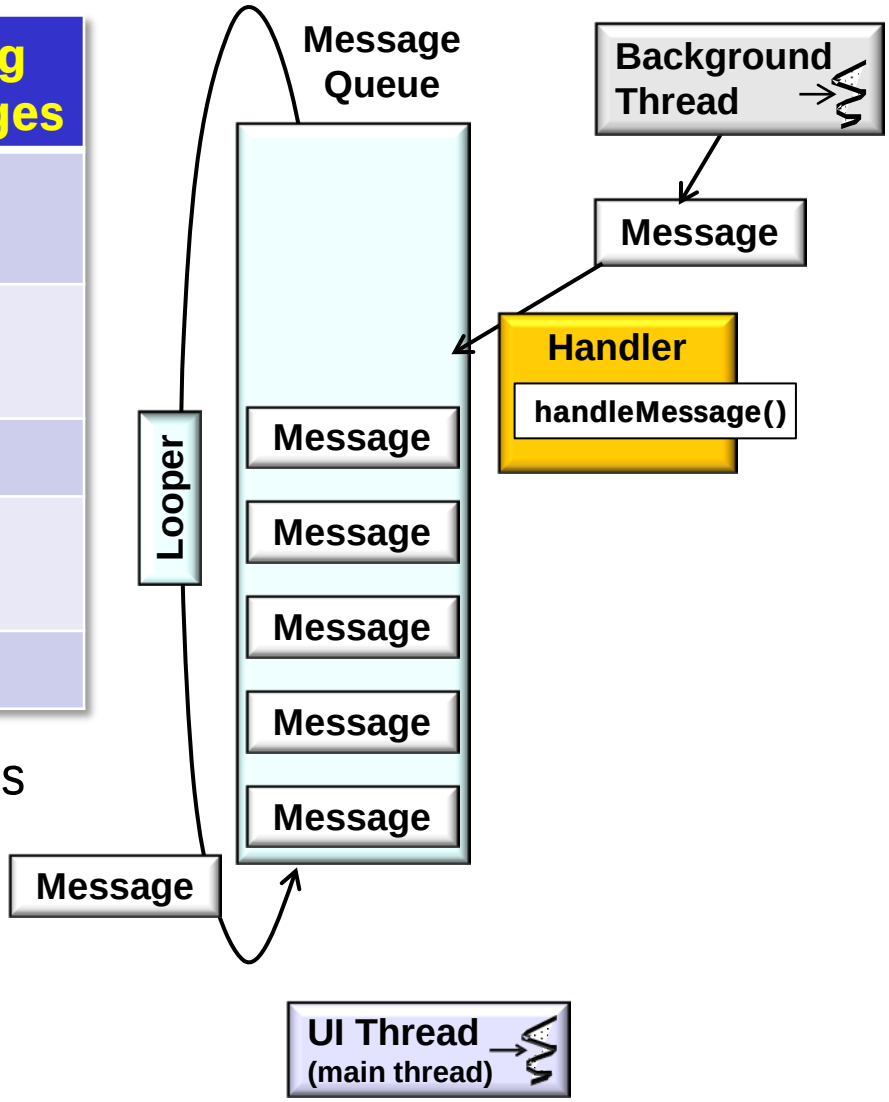
	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

- Flexible & efficient for passing Messages between various types of Threads
  - Messages contain arbitrary data



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

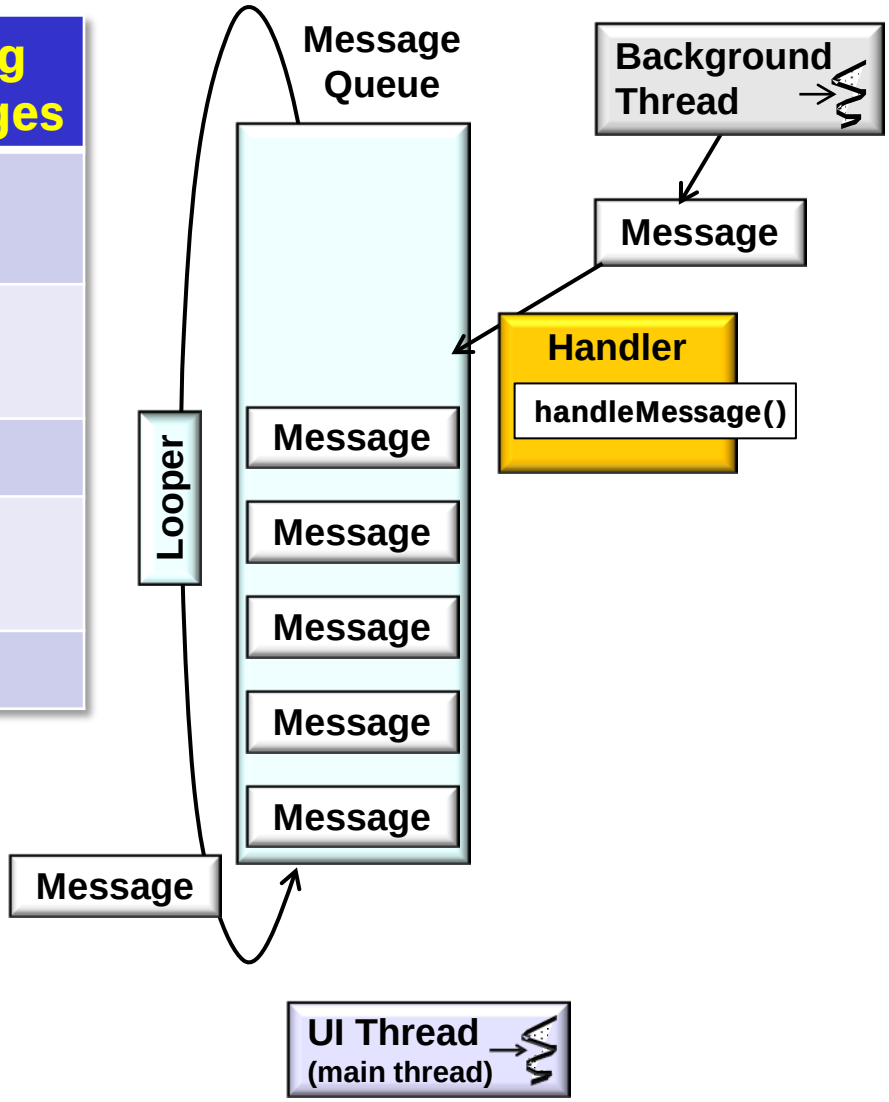


- Flexible & efficient for passing Messages between various types of Threads
  - Messages contain arbitrary data
  - Peer-to-peer conversations between various types of Threads

Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

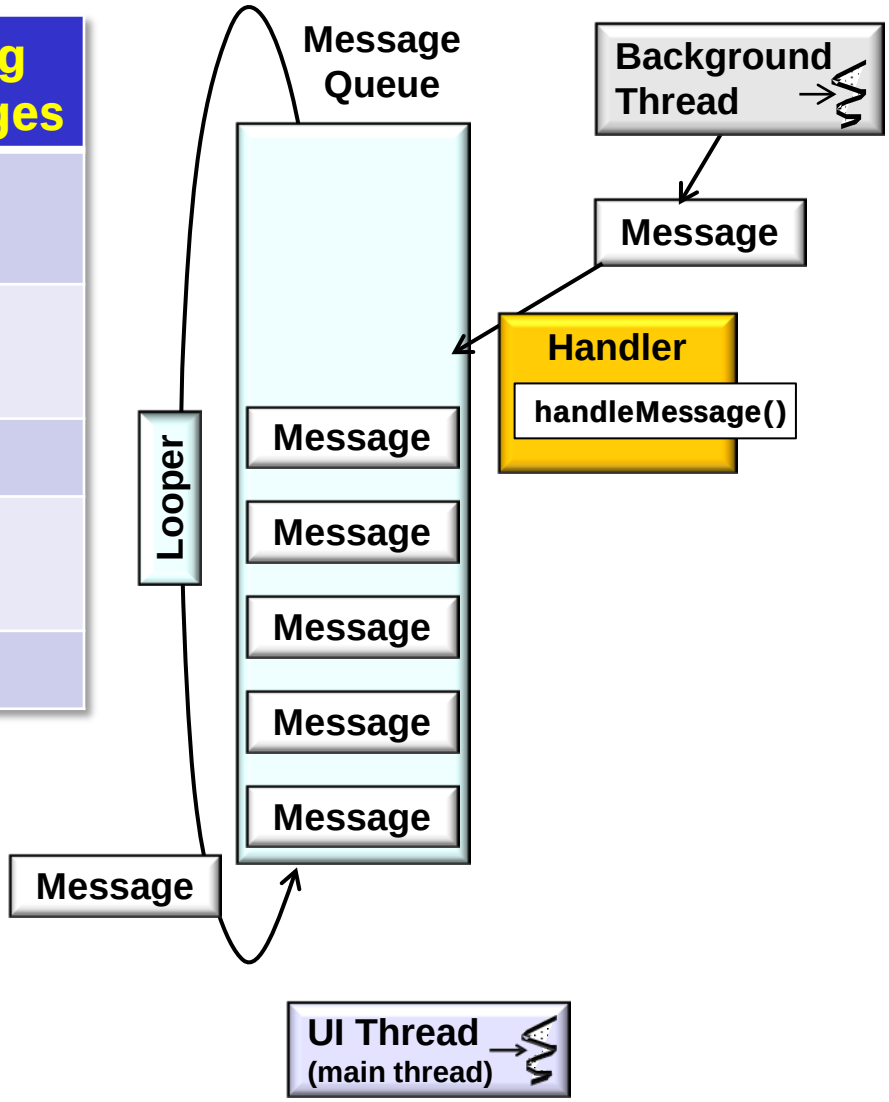
- Sending Messages is more complicated than posting Runnable commands



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

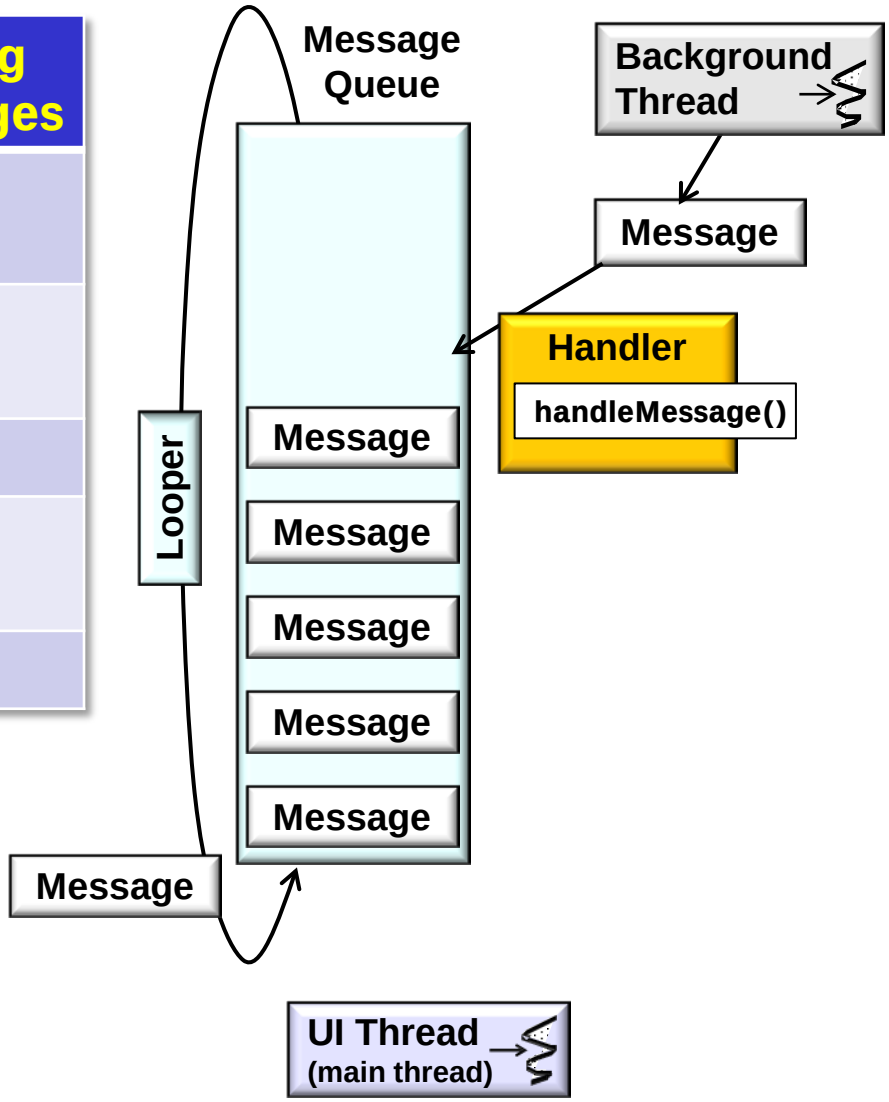
- Sending Messages is more complicated than posting Runnable commands
  - Developers must extend Handler & override its handleMessage() hook method



Comparing & Contrasting the Three Solutions

	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

- Thread pools must be managed manually, which is hard to scale



Comparing & Contrasting the Three Solutions

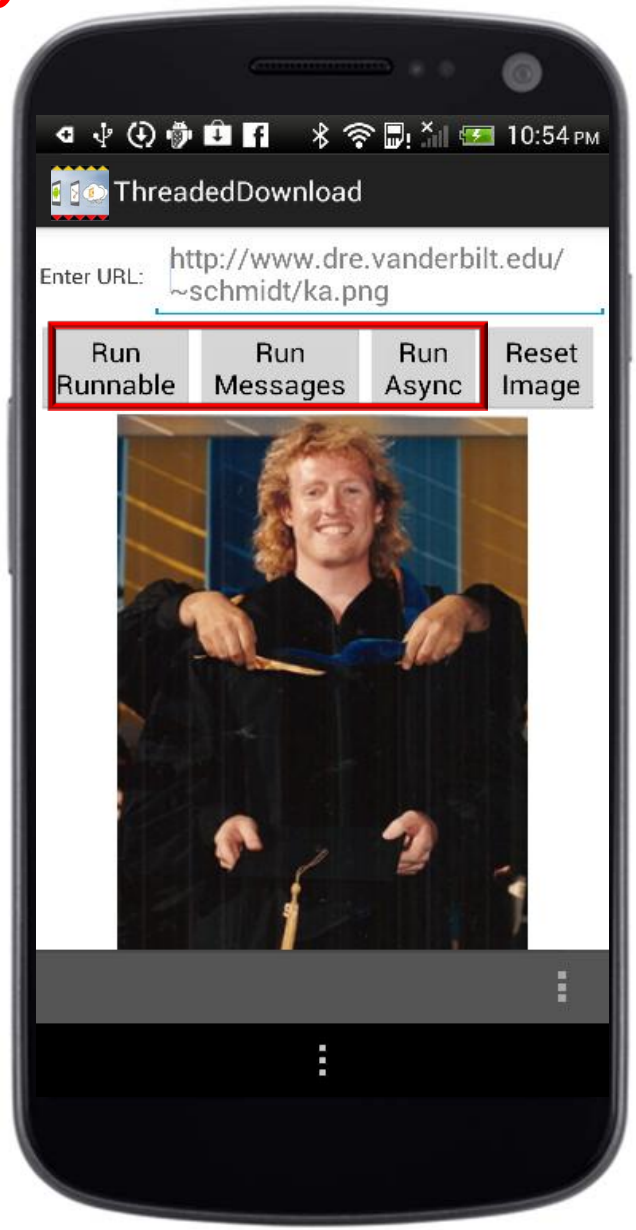
	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	☐☐☐	☐☐☐	☐☐
Usability (Complex)	☐☐☐	☐	☐☐
Scalability	☐☐☐	☐	☐
Flexibilty	☐☐	☐	☐☐☐
Efficiency	☐☐	☐☐☐	☐☐☐

Choosing the right concurrency model depends on requirements & expertise

Summary

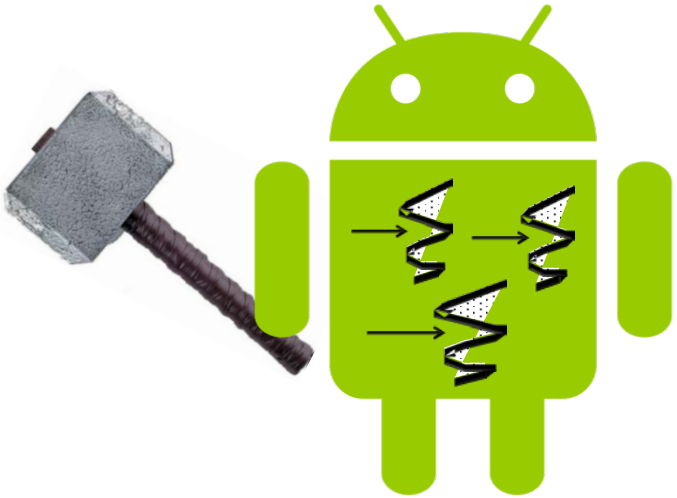
Summary

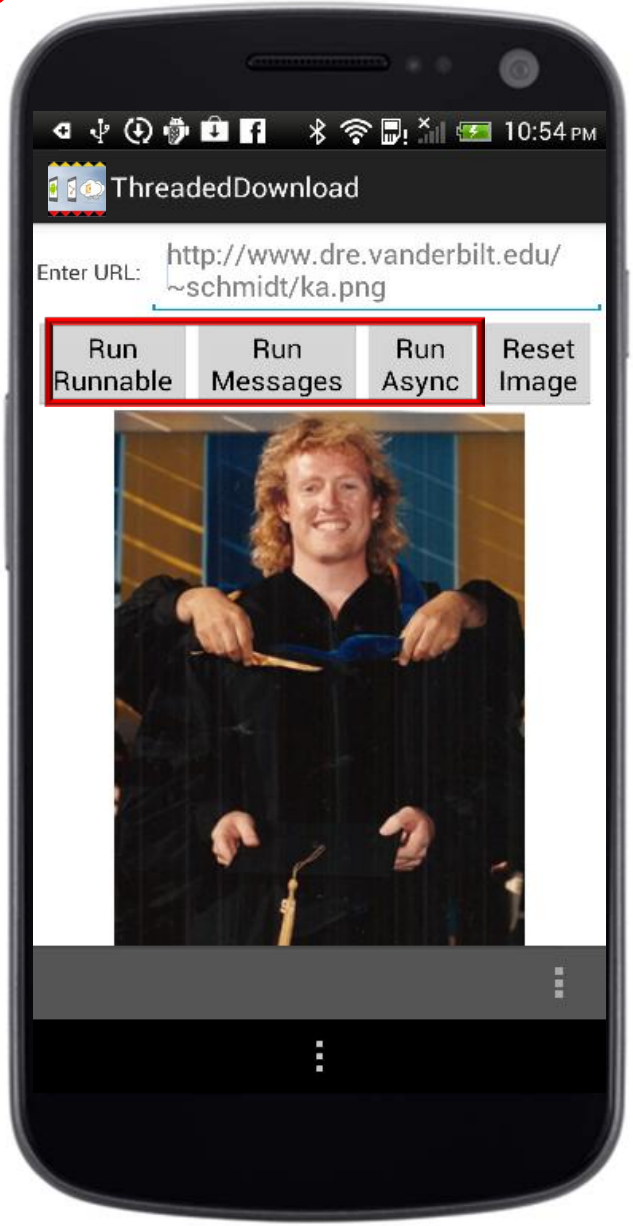
- Threaded Downloads implements three different concurrency models



Summary

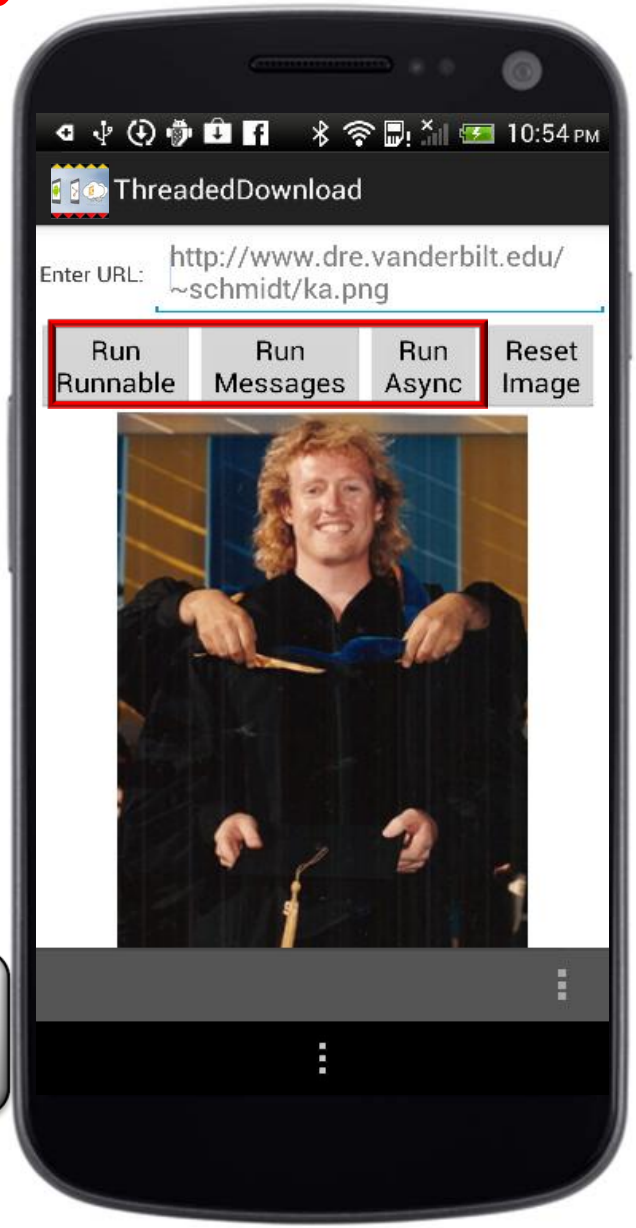
- Threaded Downloads implements three different concurrency models
  - Uses the Android HaMeR & AsyncTask frameworks

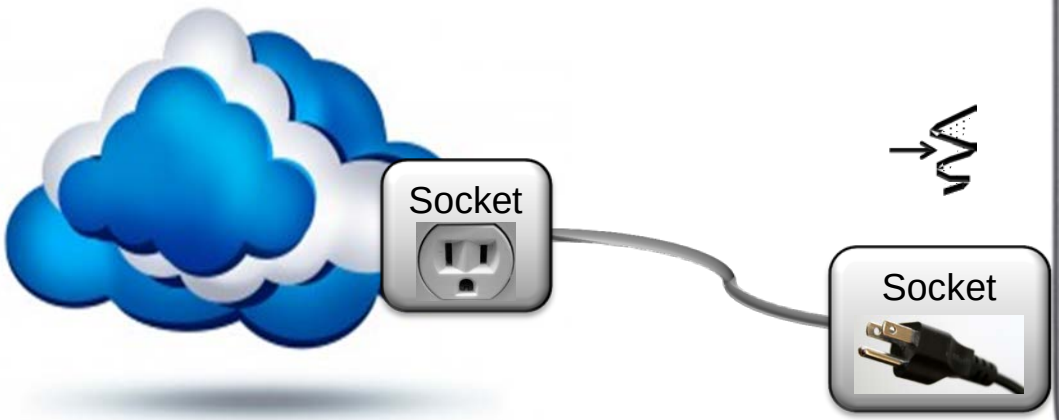




Summary

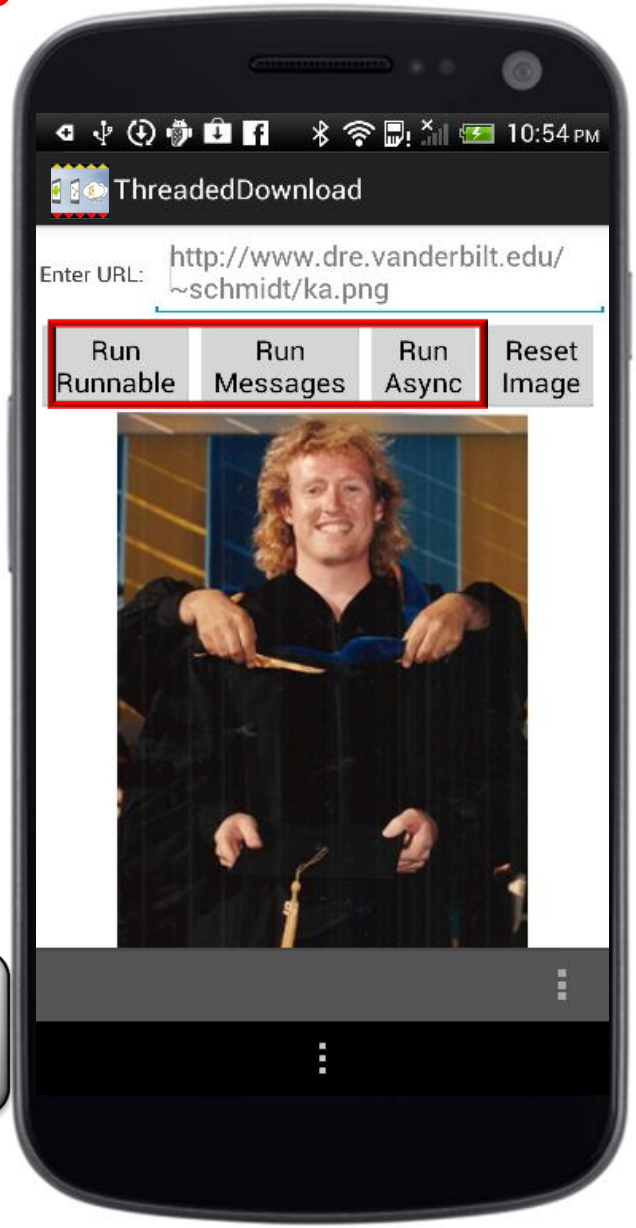
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common

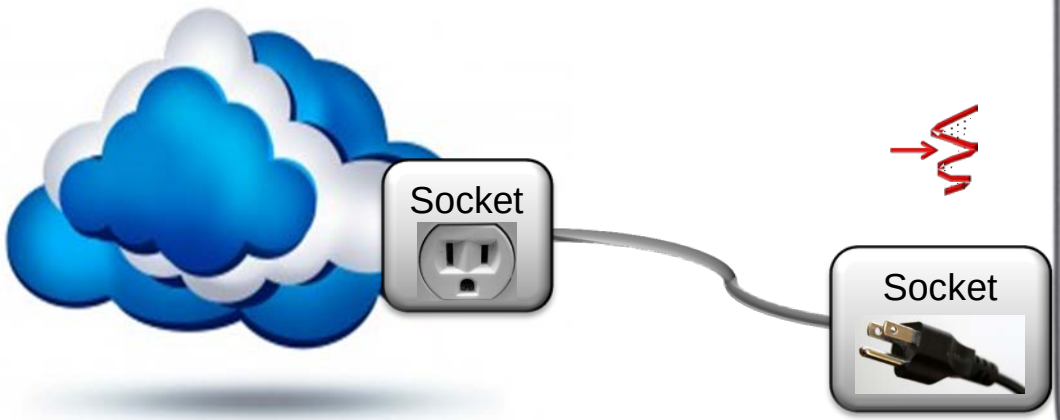




Summary

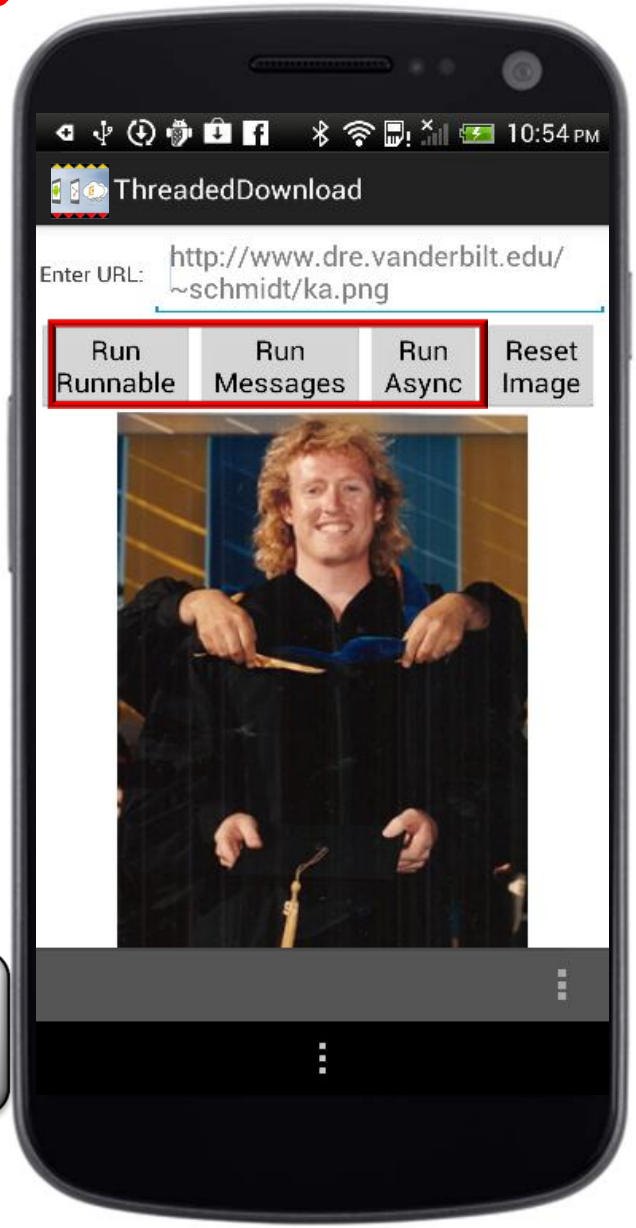
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
  - Long duration operations run in a background Thread

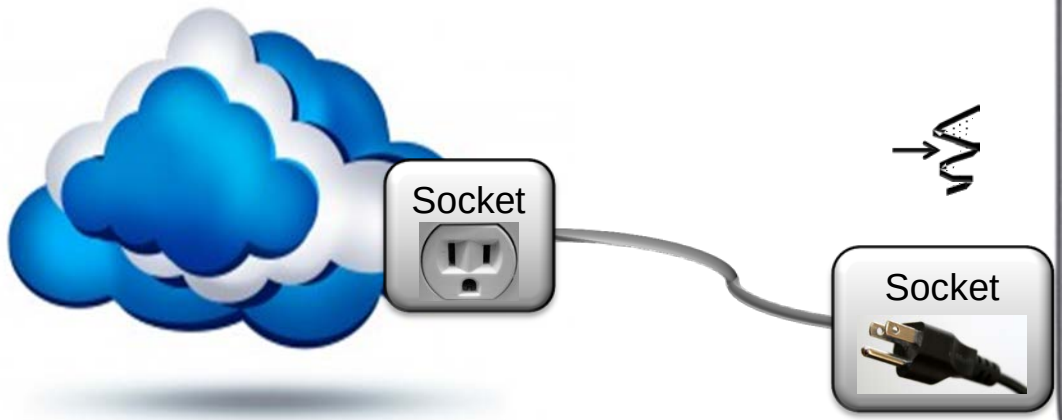




Summary

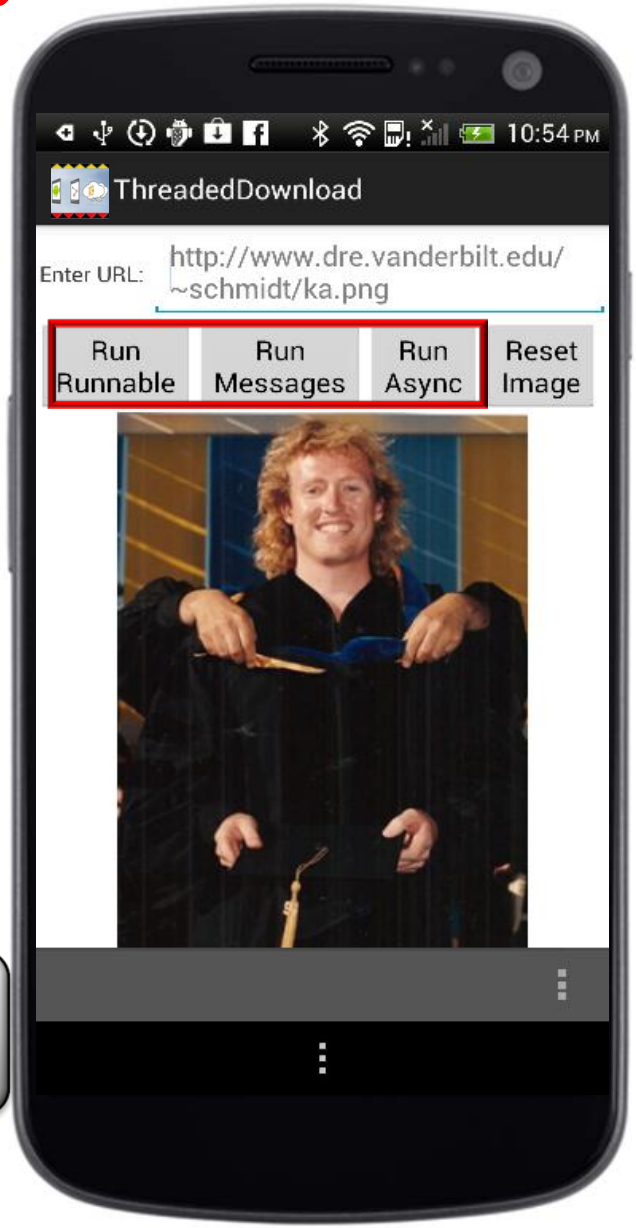
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
  - Long duration operations run in a background Thread
  - Short duration operations run in the UI Thread

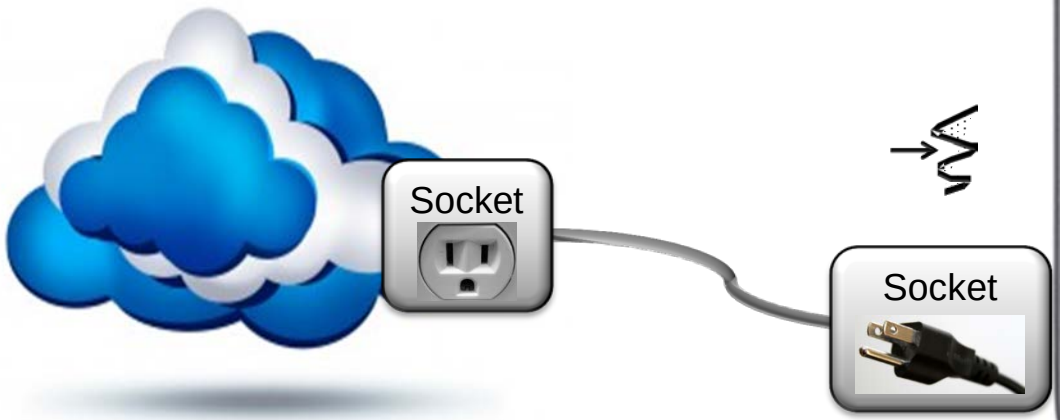




Summary

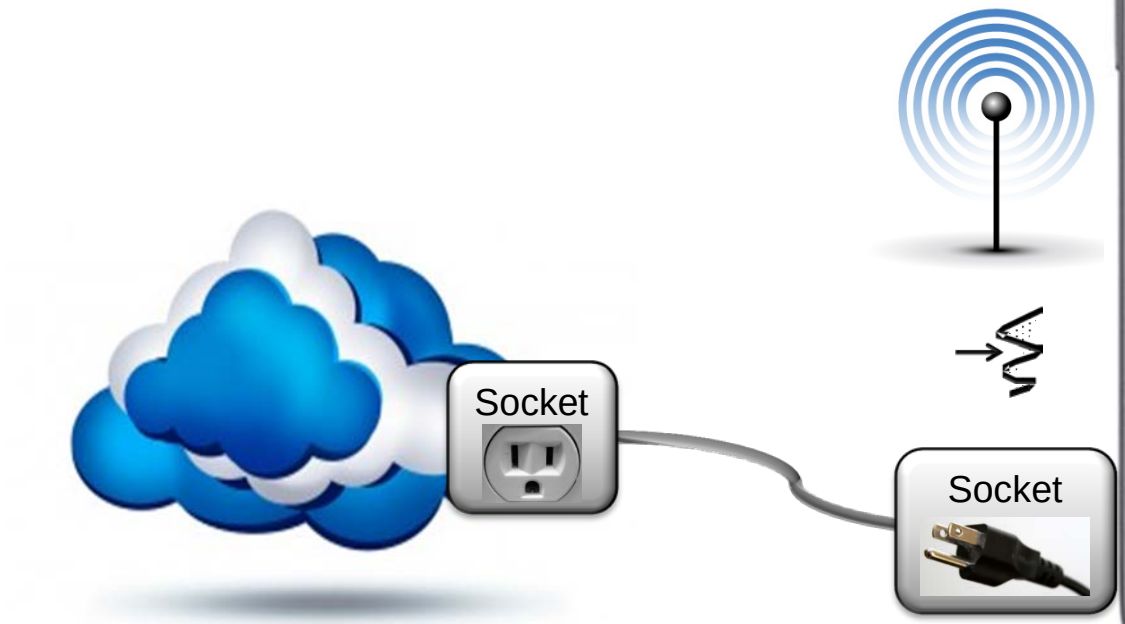
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences

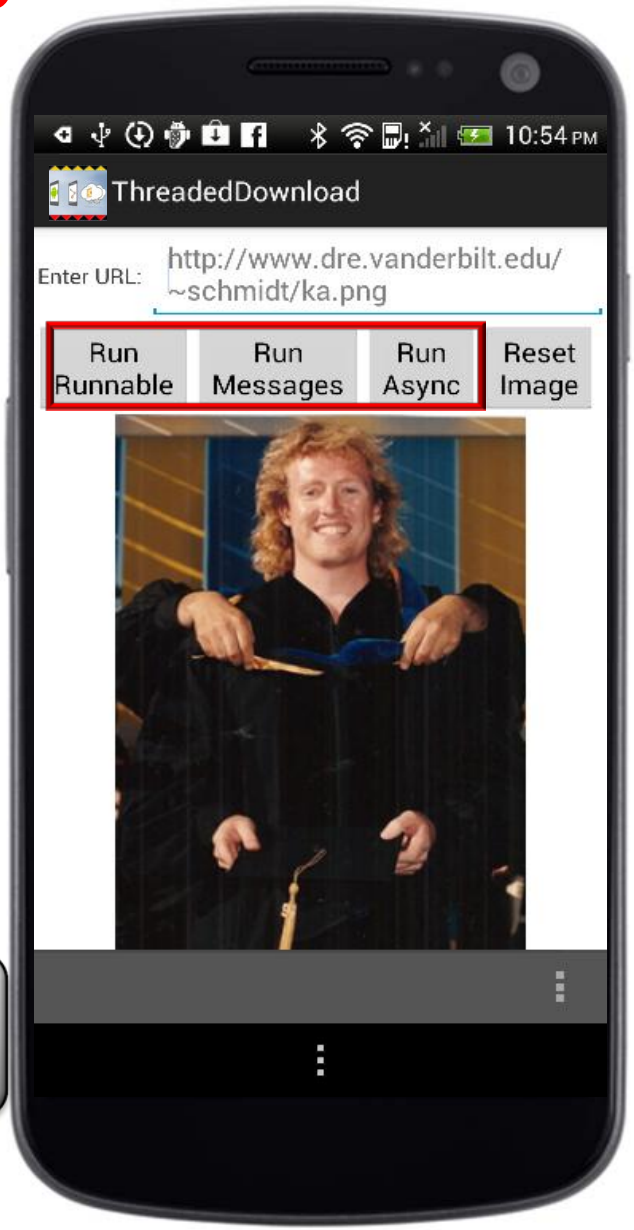




Summary

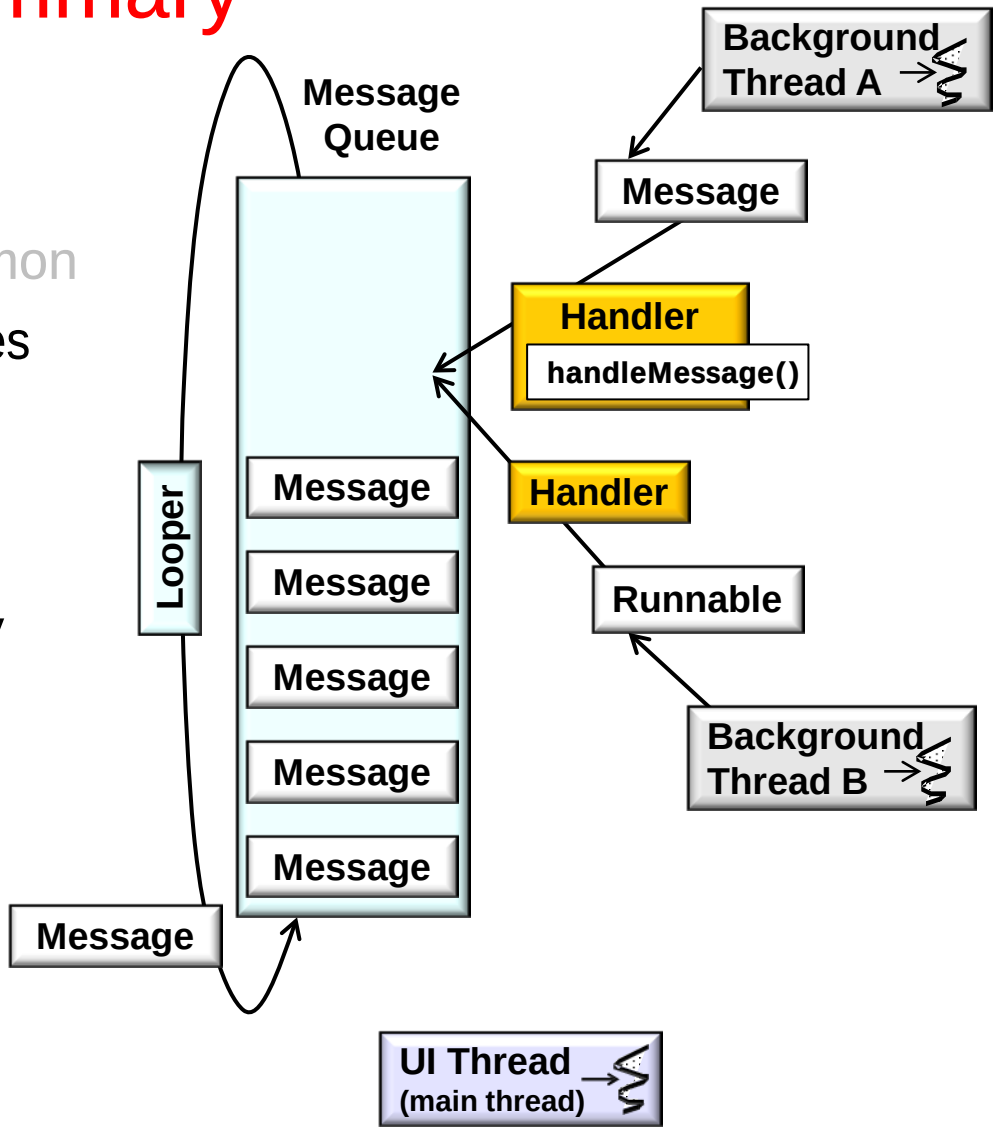
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways





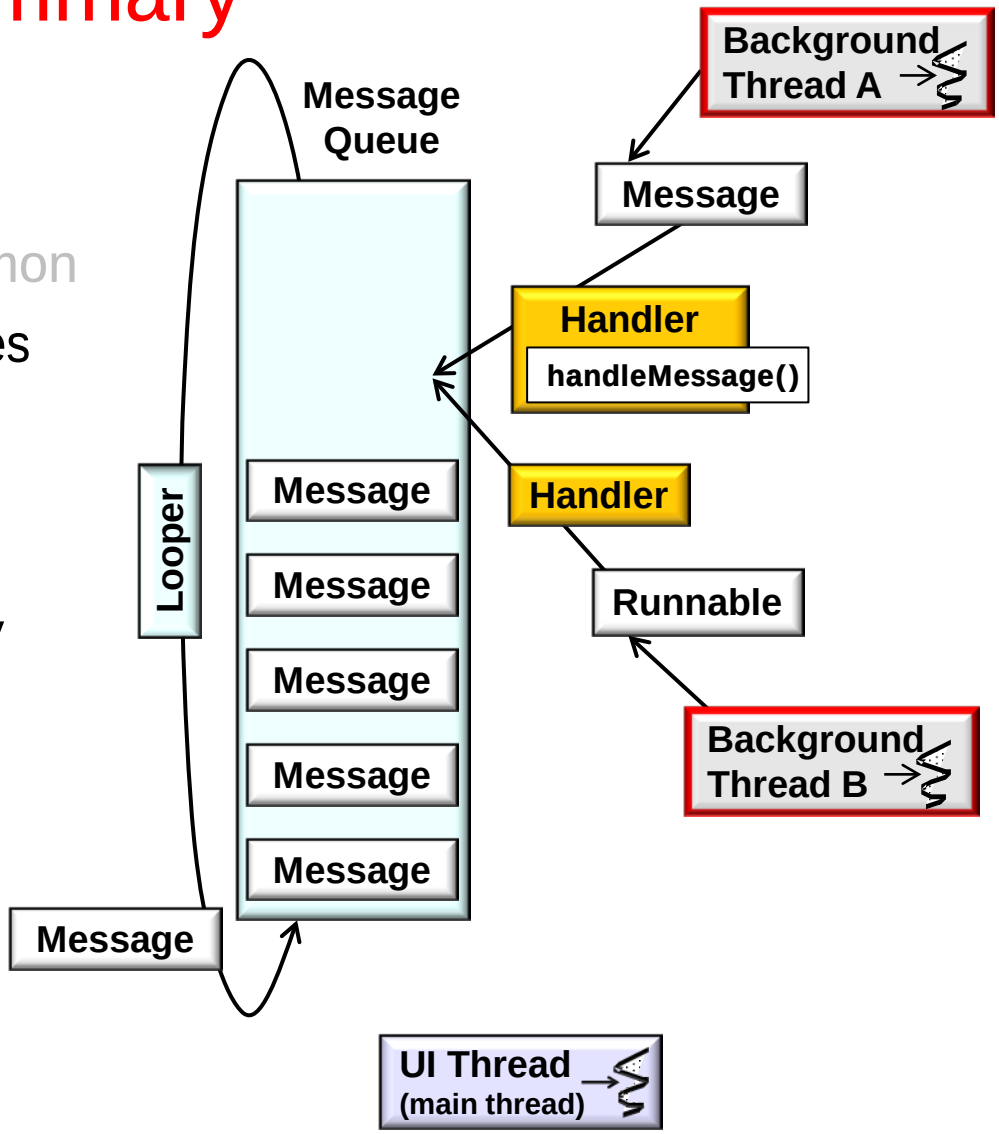
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread



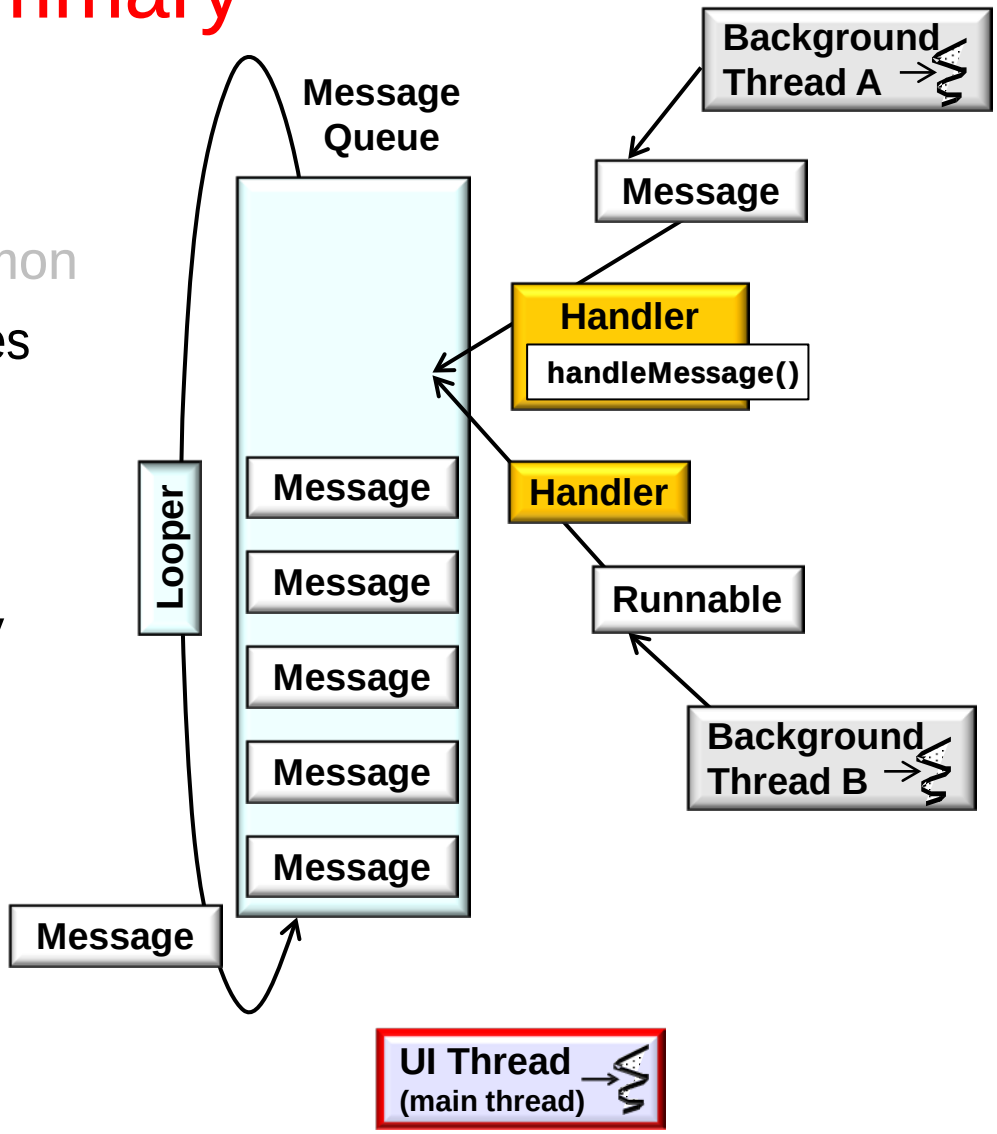
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread



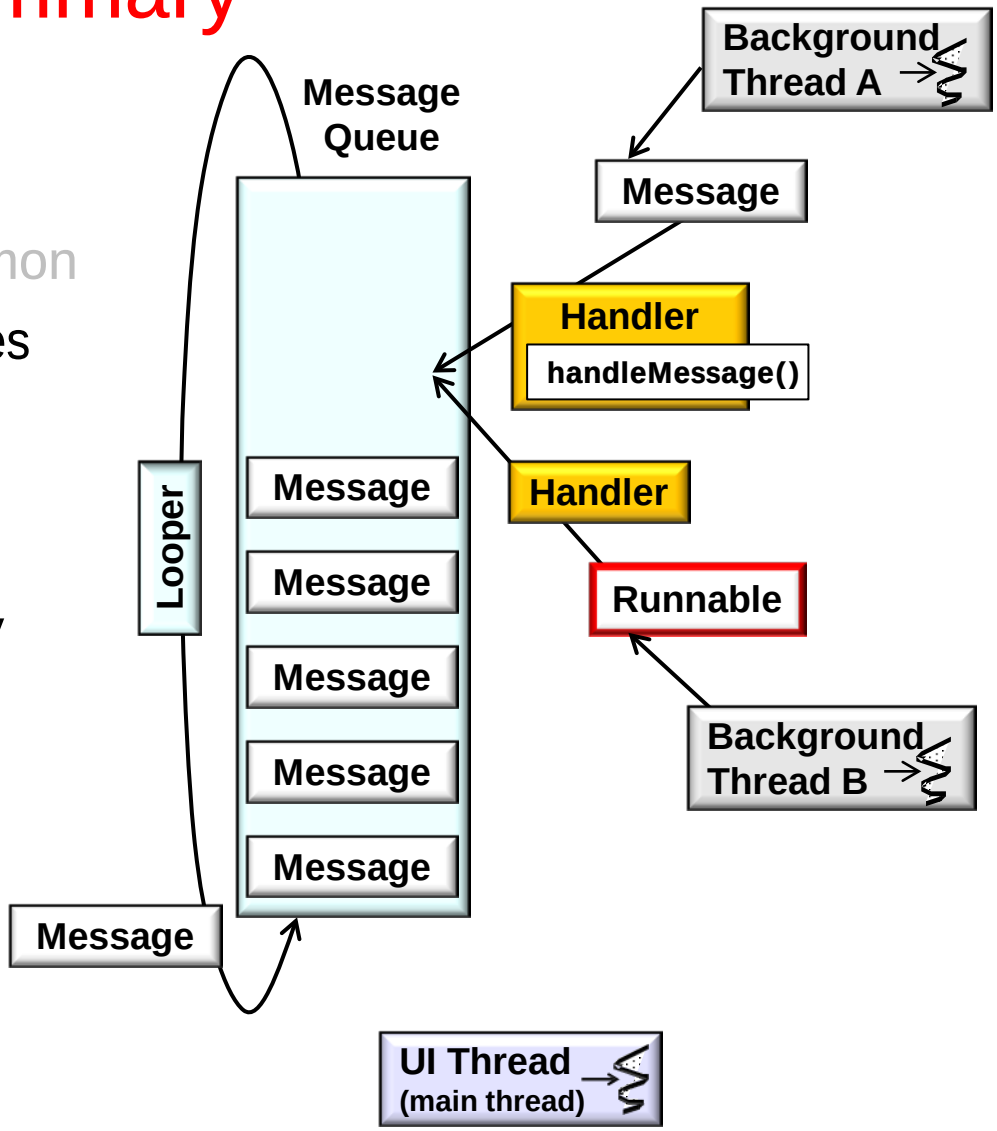
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread



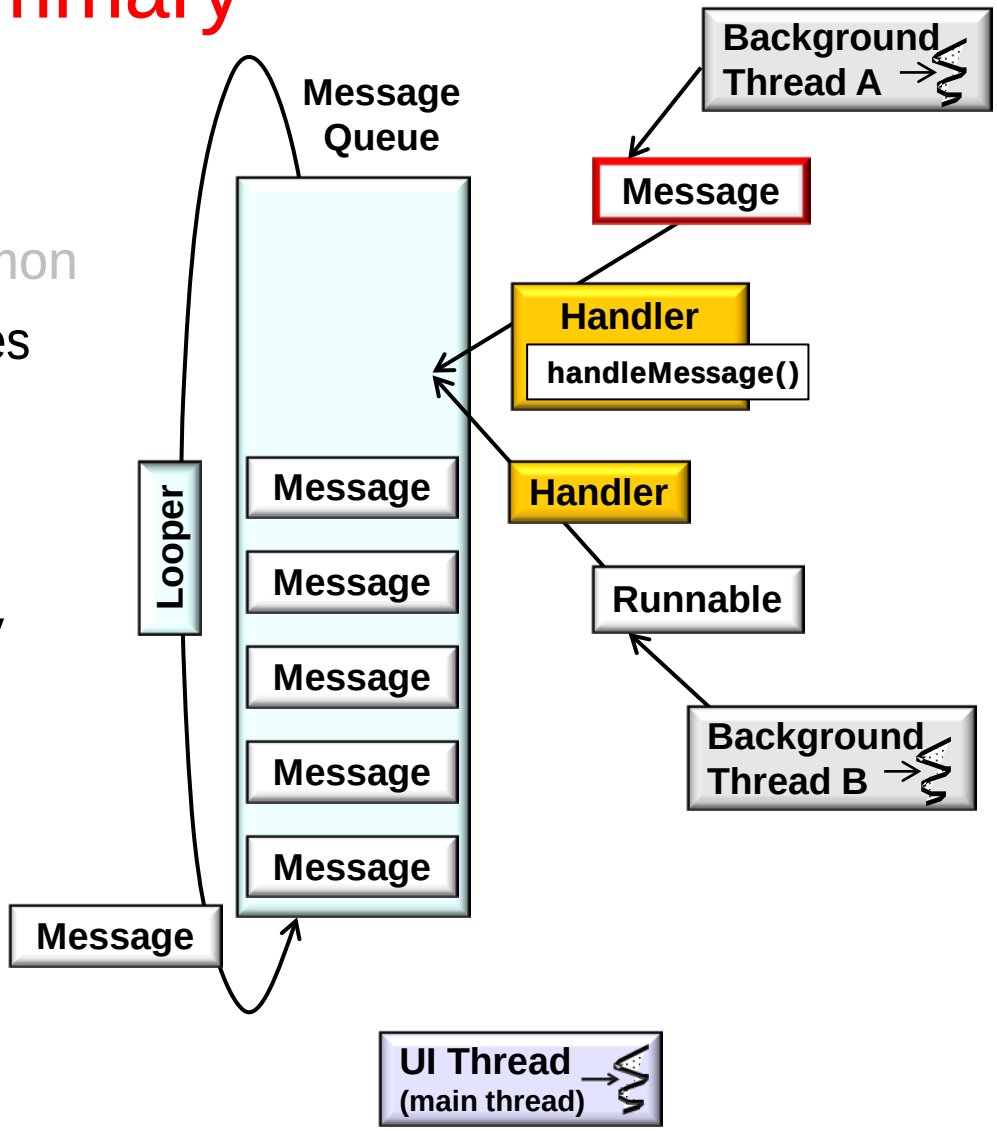
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread



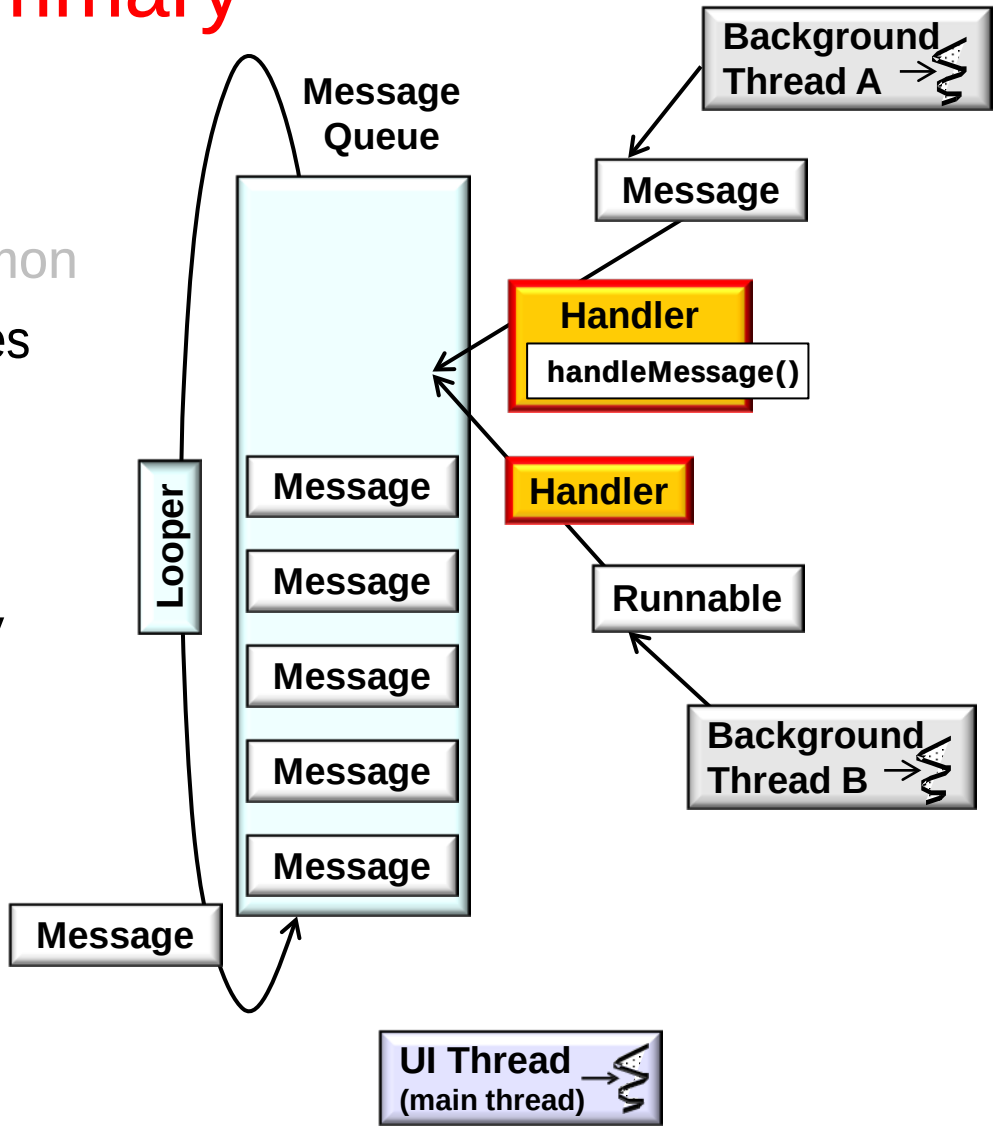
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread



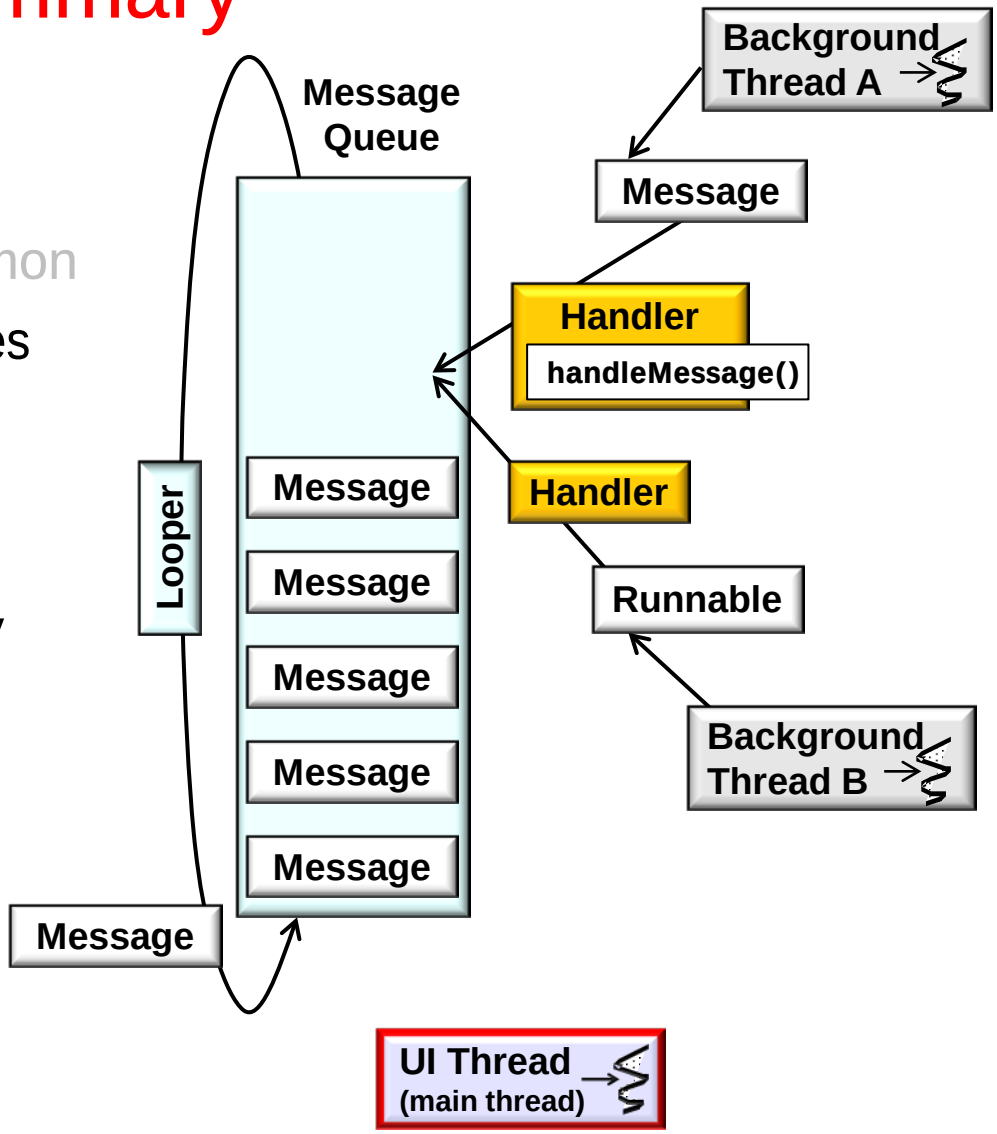
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread



Summary

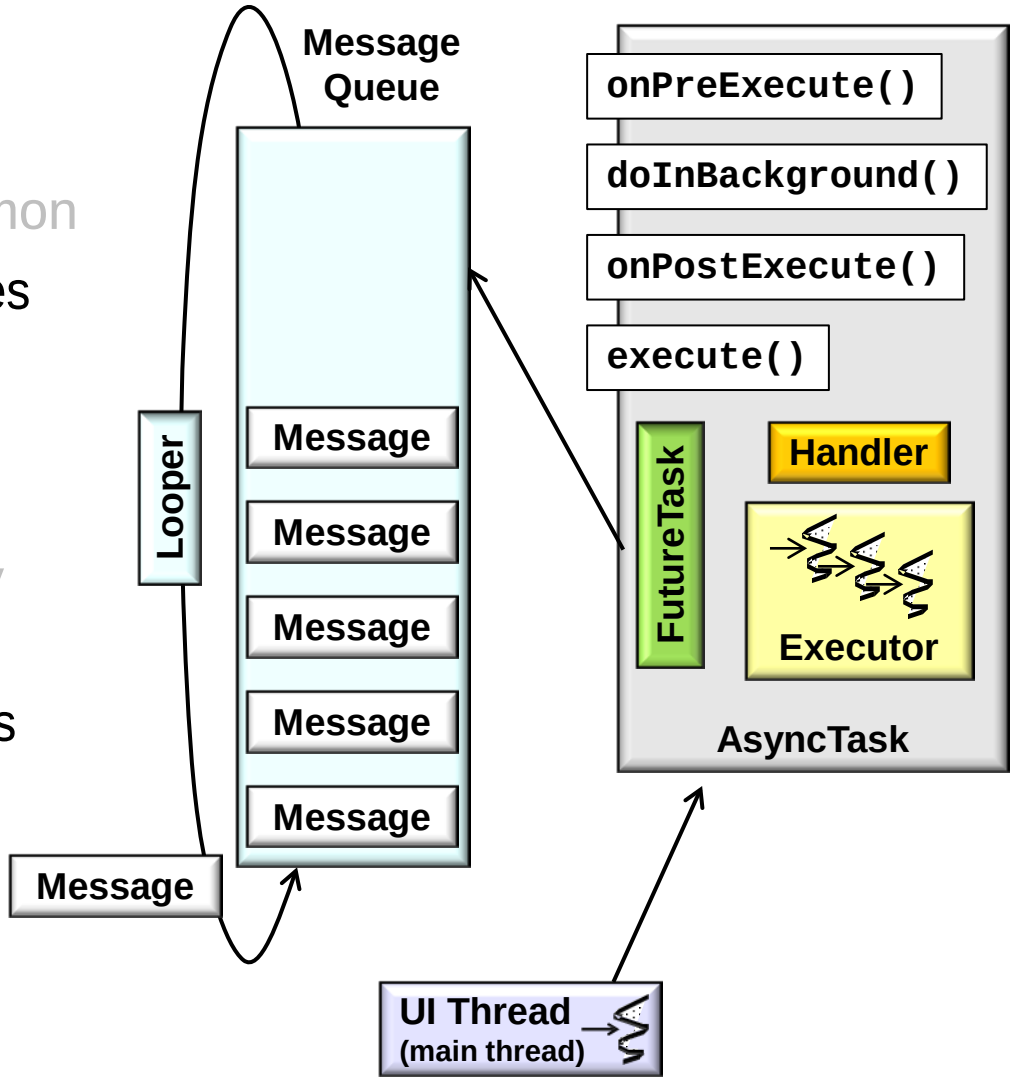
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread



See earlier parts on using the Android Handler class

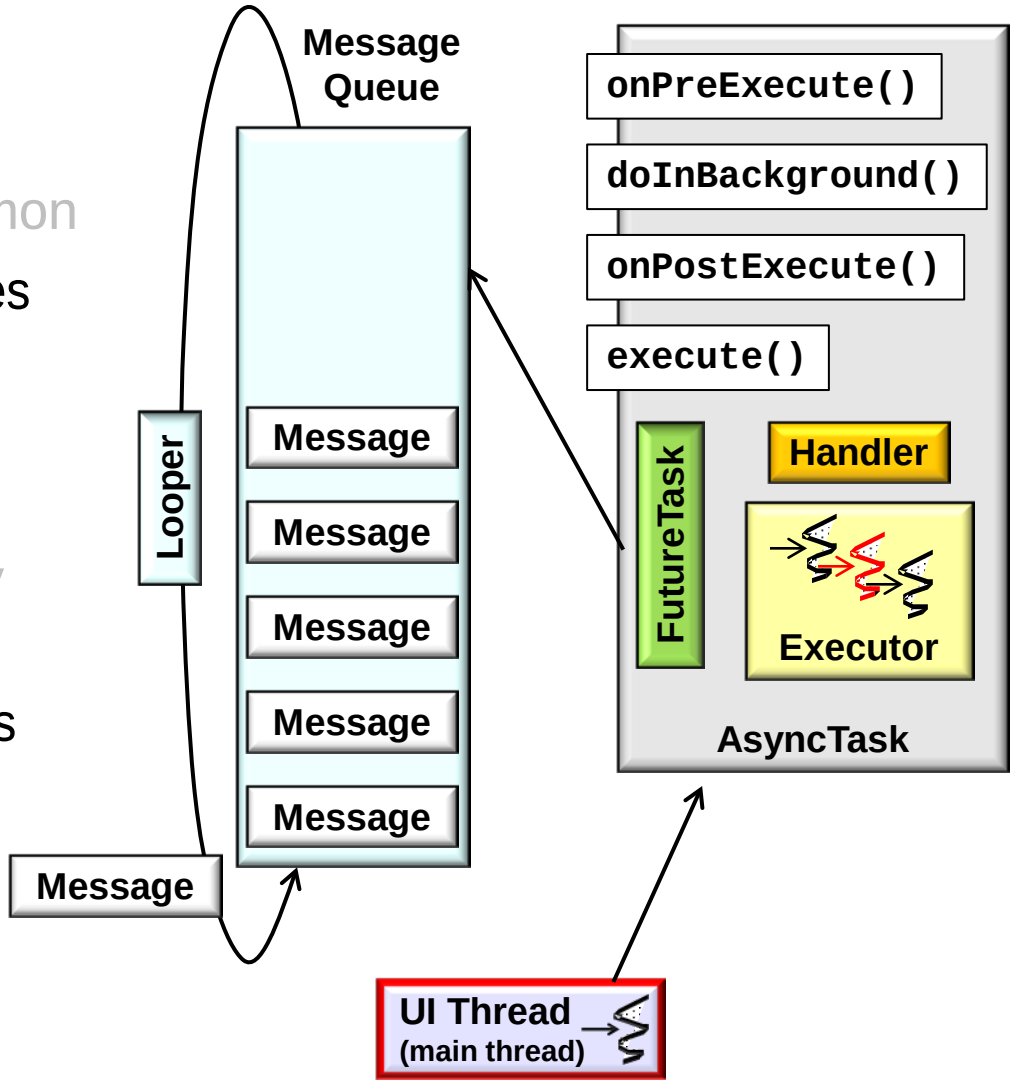
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread
  - AsyncTask framework solution has background Thread implicitly communicate to the UI Thread



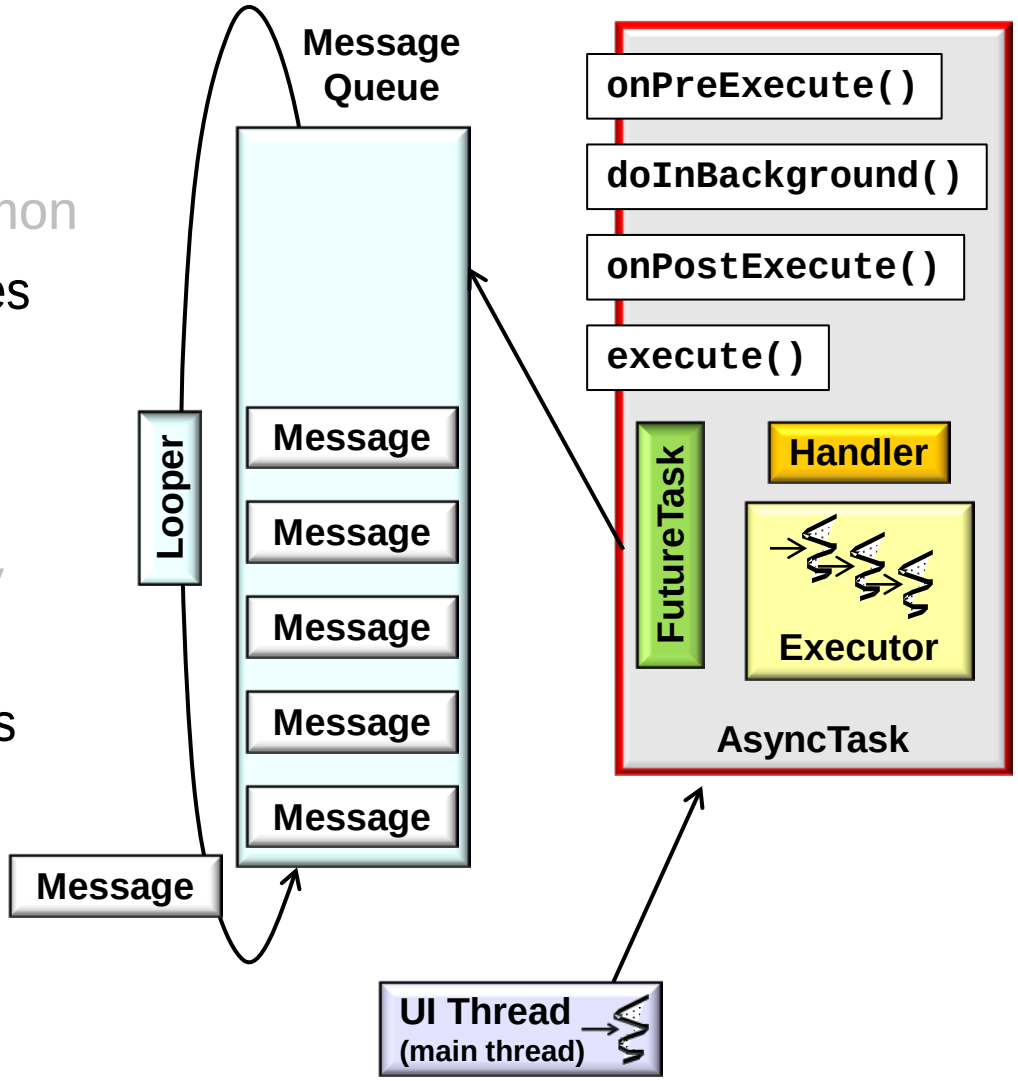
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread
  - AsyncTask framework solution has background Thread implicitly communicate to the UI Thread



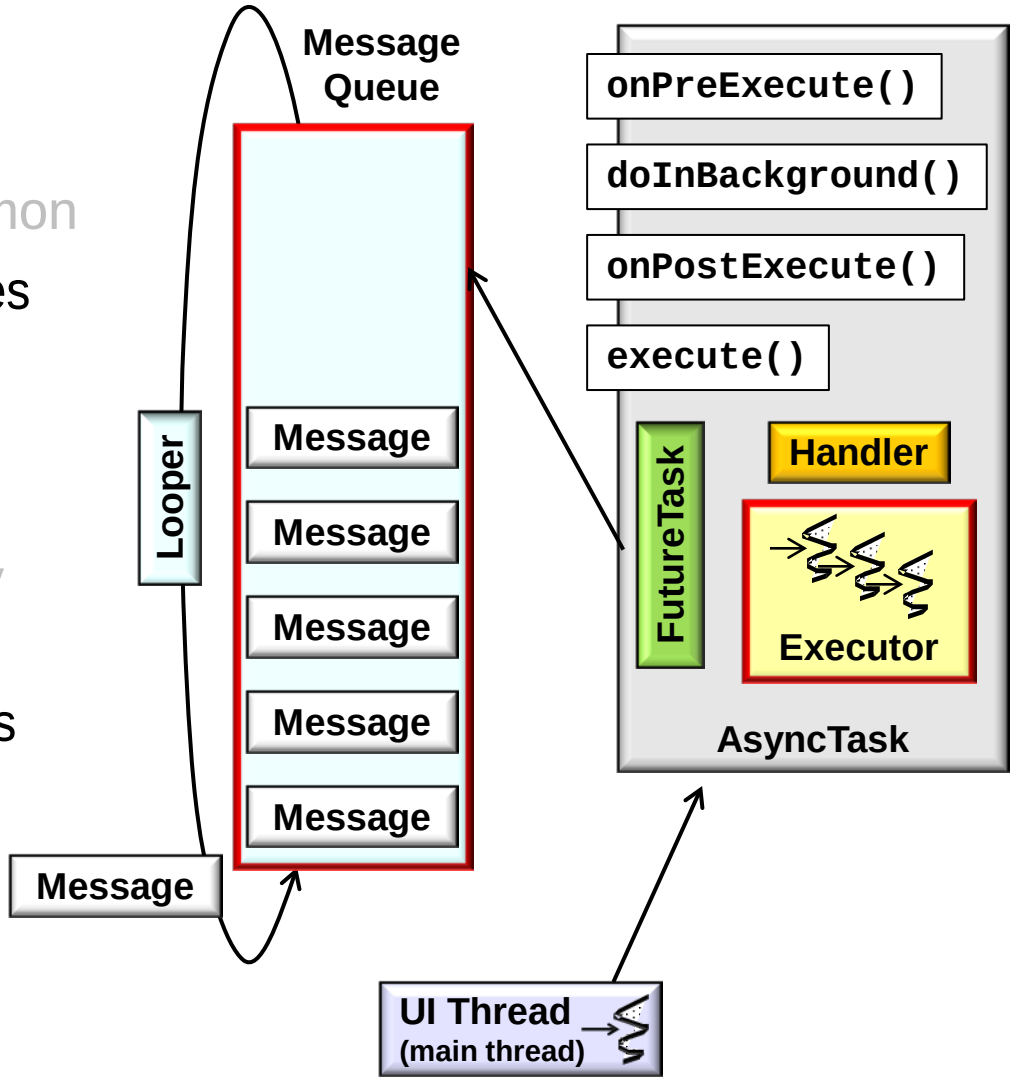
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread
  - AsyncTask framework solution has background Thread implicitly communicate to the UI Thread



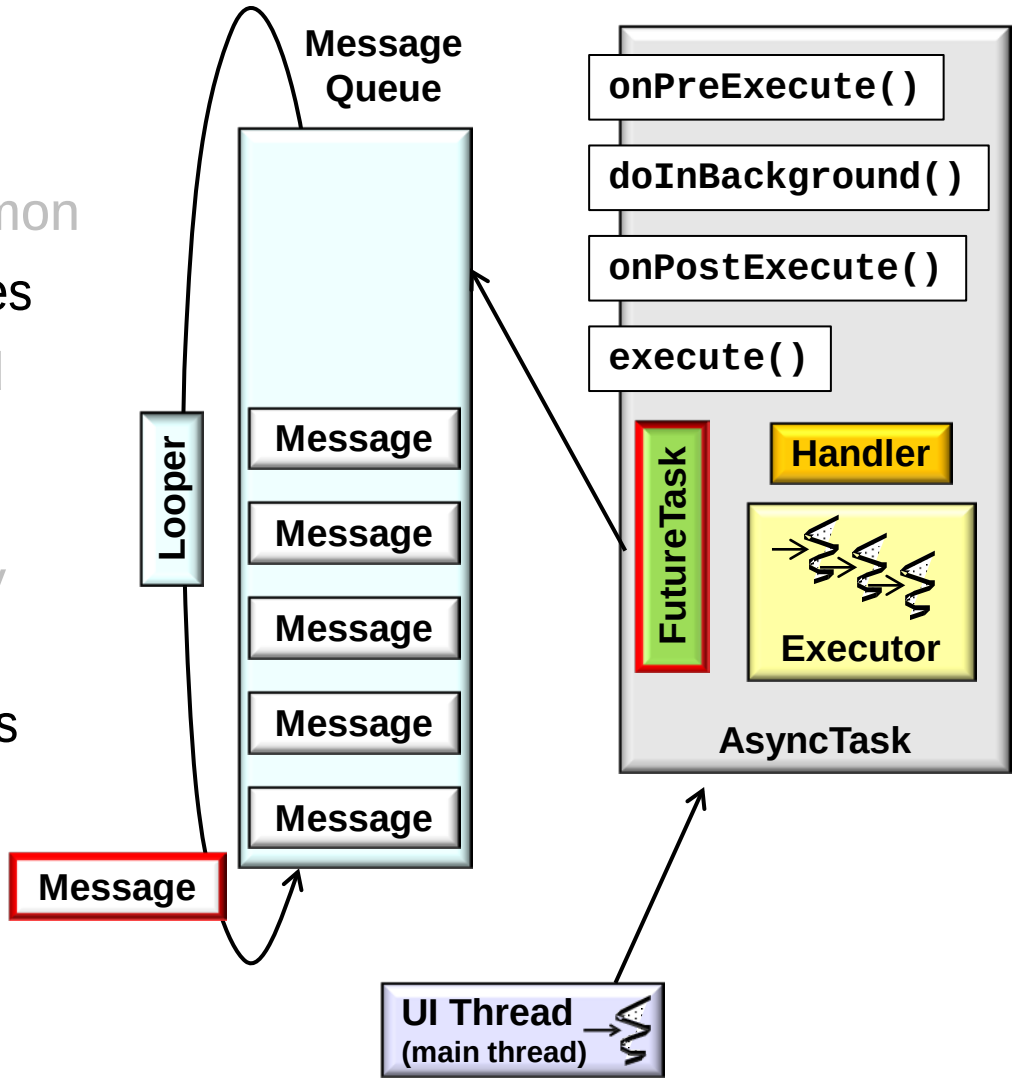
Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread
  - AsyncTask framework solution has background Thread implicitly communicate to the UI Thread



Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
  - Background Threads & UI Thread communicate in different ways
  - In HaMeR framework solutions the background Threads explicitly communicate to the UI Thread
  - AsyncTask framework solution has background Thread implicitly communicate to the UI Thread



See earlier parts on “The AsyncTask Framework”

Summary

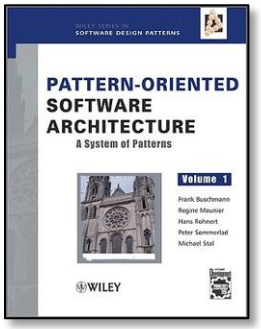
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
- Each solution has pros & cons

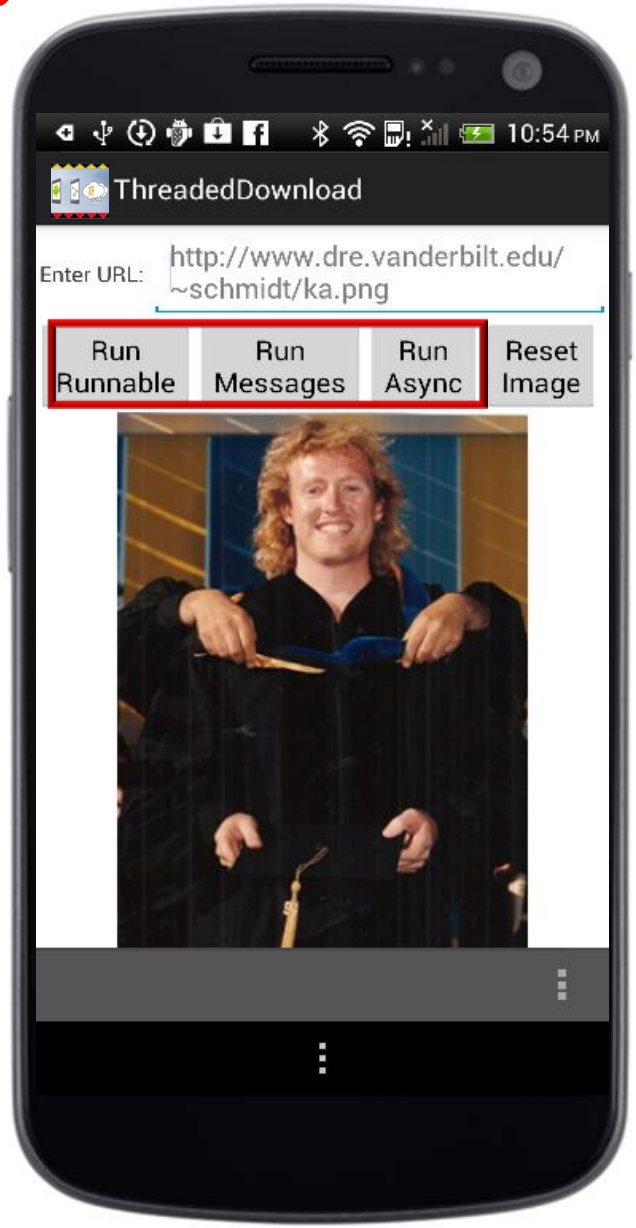
	Async Task	Posting Runnables	Sending Messages
Usability (Simple)	□□□	□□□	□□
Usability (Complex)	□□□	□	□□
Scalability	□□□	□	□
Flexibilty	□□	□	□□□
Efficiency	□□	□□□	□□□

It's important to understand application requirements to make the right choice

Summary

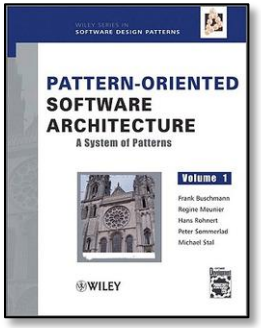
- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
- Each solution has pros & cons
- These solutions are based on GoF & POSA patterns

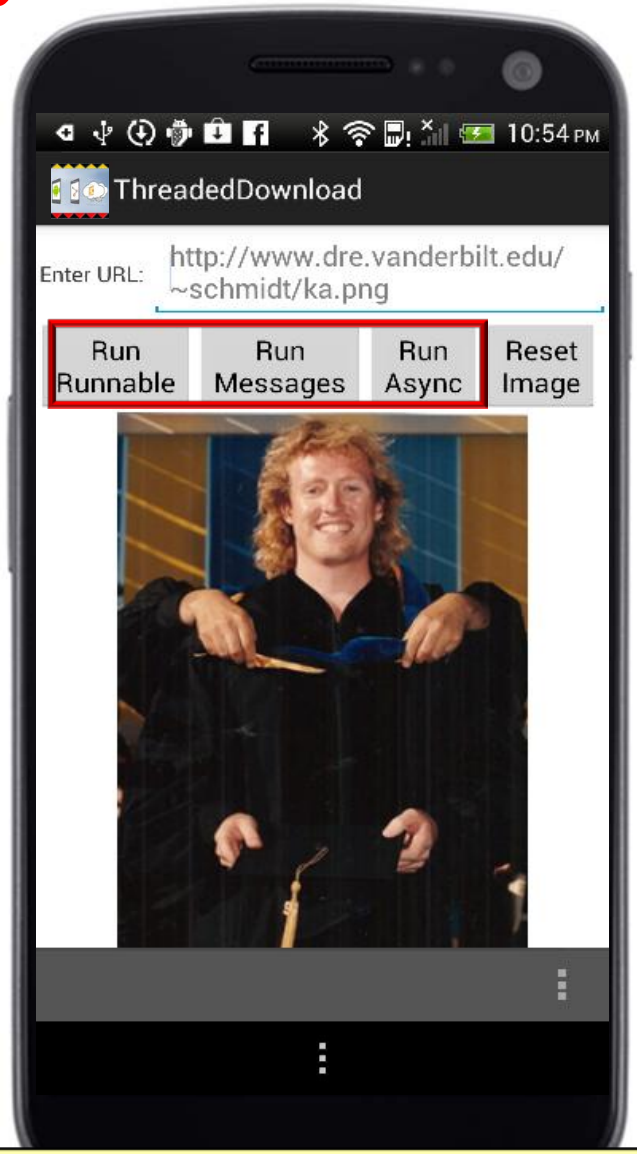




Summary

- Threaded Downloads implements three different concurrency models
- Solutions share some thing in common
- Solutions also have some differences
- Each solution has pros & cons
- These solutions are based on GoF & POSA patterns





Next module discusses these patterns & framework implementations in detail