

How K9mail works

index

How K9mail works.....	1
When it starts with only one account	3
app/k9mail/src/main/AndroidManifest.xml	3
com/fsck/k9/activity/Accounts.java.....	4
com/fsck/k9/activity/MessageList.java.....	5
com/fsck/k9/activity/Accounts.java.....	6
com/fsck/k9/activity/MessageList.java.....	7
title	8
search.....	9
com/fsck/k9/activity/Accounts.java.....	9
com/fsck/k9/activity/MessageList.java.....	10
xml/searchable.xml	12
k9mail\k-9\app\k9mail\src\main\AndroidManifest.xml	13
When search is clicked in MessageList.....	14
com/fsck/k9/activity/MessageList.java.....	14
Start search when enter is pressed in the system search UI.....	15
com/fsck/k9/activity/MessageList.java.....	15
onCreate -> onStart.....	16
When one account is clicked:.....	17
Return to AccountList from FolderList:	19

Attachment	19
app\ui\src\main\java\com\fsck\k9\fragment\MessageListAdapter.java	19
app\ui\src\main\java\com\fsck\k9\ui\messageview\MessageContainerView.java	20
com/fsck/k9/ui/messageview/AttachmentViewCallback.java	21
com/fsck/k9/ui/messageview/AttachmentView.java	21
com/fsck/k9/activity/MessageLoaderHelper.java	24
app\ui\src\main\java\com\fsck\k9\ui\messageview\MessageViewFragment.java	24
app\ui\src\main\java\com\fsck\k9\ui\messageview\MessageViewFragment.java	27
layout/message_view_attachment.xml	28
app\ui\src\main\res\layout\message.xml	28
app\ui\src\main\res\values\strings.xml	28
loadMoreMessages	30
app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java	30
com/fsck/k9/controller/MessagingController.java	32
app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java	33
displayCount	35
com/fsck/k9/Account.java	35
com/fsck/k9/K9.java	36
Message Item Clicked	36
app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java	36
com/fsck/k9/fragment/MessageListHandler.java	37
com/fsck/k9/activity/MessageList.java	38
com/fsck/k9/ui/messageview/MessageViewFragment.java	38
com/fsck/k9/activity/MessageLoaderHelper.java	39
com/fsck/k9/ui/message/LocalMessageExtractorLoader.java	40
PullToRefresh	41

app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java	41
download_remainder	43
app\ui\src\main\java\com\fsck\k9\ui\messageview\MessageTopView.java	43

When it starts with only one account

app/k9mail/src/main/AndroidManifest.xml

```

<activity
    android:name=".activity.Accounts"
    android:configChanges="locale"
    android:label="@string/app_name"
    android:launchMode="singleTop"
    android:uiOptions="splitActionBarWhenNarrow">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>

        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.LAUNCHER"/>

        <!-- TODO: Remove once minSdkVersion has been changed to 24+ -->
        <category android:name="android.intent.category.MULTIWINDOW_LAUNCHER"/>
    </intent-filter>
</activity>

```

```
        <category android:name="android.intent.category.PENWINDOW_LAUNCHER"/>
    </intent-filter>
</activity>
```

com/fsck/k9/activity/Accounts.java

```
public void onCreate(Bundle icle) {
    ...
    List<Account> accounts = Preferences.getPreferences(this).getAccounts();
    ...
} else if (startup && accounts.size() == 1 && onOpenAccount(accounts.get(0))) {

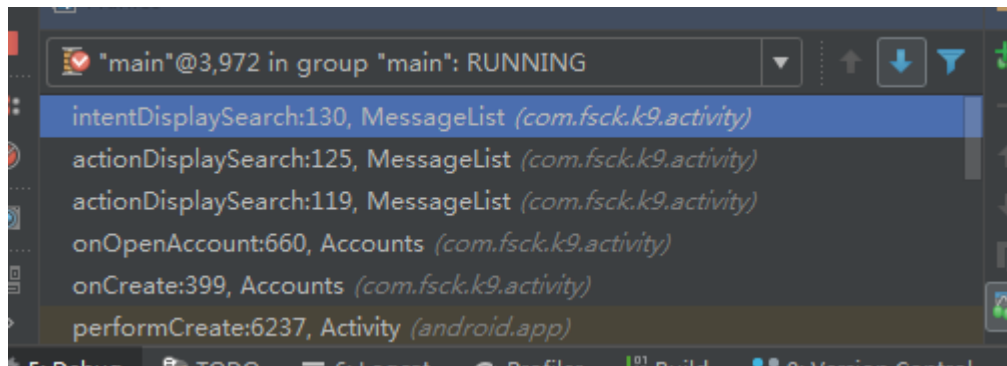
/**
 * Show that account's inbox or folder-list
 * or return false if the account is not available.
 * @param account the account to open ({@link SearchAccount} or {@link Account})
 * @return false if unsuccessful
 */
private boolean onOpenAccount(BaseAccount account) {
    ...
    LocalSearch search = new LocalSearch(realAccount.getAutoExpandFolder());
    search.addAllowedFolder(realAccount.getAutoExpandFolder());
```

```
search.addAccountUuid(realAccount.getUuid());  
MessageList.actionDisplaySearch(this, search, false, true);}
```

com/fsck/k9/activity/MessageList.java

```
public static void actionDisplaySearch(Context context, SearchSpecification search,  
    boolean noThreading, boolean newTask, boolean clearTop) {  
    context.startActivity(  
        intentDisplaySearch(context, search, noThreading, newTask, clearTop));  
}  
  
public static Intent intentDisplaySearch(Context context, SearchSpecification search,  
    boolean noThreading, boolean newTask, boolean clearTop) {  
    Intent intent = new Intent(context, MessageList.class);  
    intent.putExtra(EXTRA_SEARCH, ParcelableUtil.marshall(search));  
    intent.putExtra(EXTRA_NO_THREADING, noThreading);  
  
    if (clearTop) {  
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);  
    }  
  
    if (newTask) {
```

```
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
    }  
  
    return intent;  
}
```

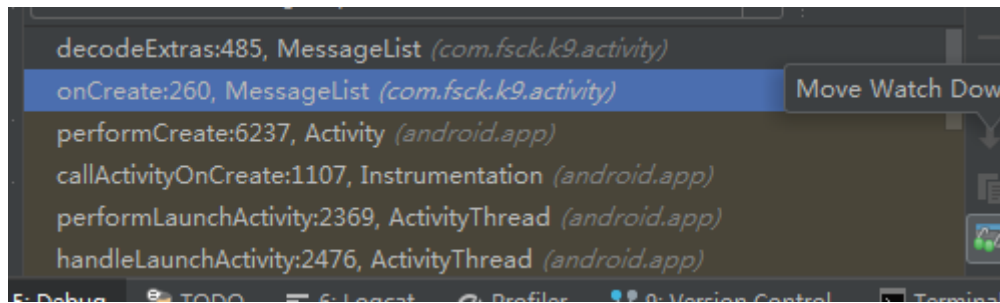


com/fsck/k9/activity/Accounts.java

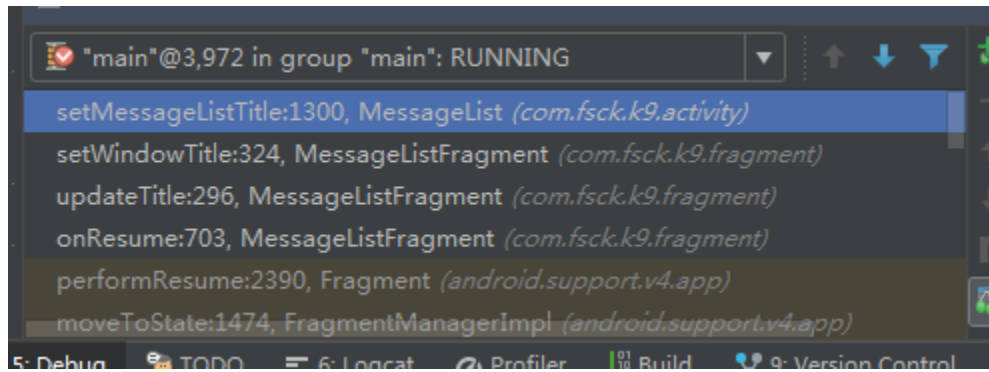
```
public void onCreate(Bundle savedInstanceState) {  
    ...  
    if (!decodeExtras(getIntent())) {  
        return;  
    }  
}
```

com/fsck/k9/activity/MessageList.java

```
} else {  
    // regular LocalSearch object was passed  
    search = intent.hasExtra(EXTRA_SEARCH) ?  
        ParcelableUtil.unmarshall(intent.getByteArrayExtra(EXTRA_SEARCH), LocalSearch.CREATOR) : null;  
    noThreading = intent.getBooleanExtra(EXTRA_NO_THREADING, false);  
}
```



title



search

com/fsck/k9/activity/Accounts.java

```
/**
 * Show that account's inbox or folder-list
 * or return false if the account is not available.
 * @param account the account to open ({@link SearchAccount} or {@link Account})
 * @return false if unsuccessful
 */
private boolean onOpenAccount(BaseAccount account) {
    ...
    LocalSearch search = new LocalSearch(realAccount.getAutoExpandFolder());
    search.addAllowedFolder(realAccount.getAutoExpandFolder());
    search.addAccountUuid(realAccount.getUuid());
    MessageList.actionDisplaySearch(this, search, false, true);}
}
```

com/fsck/k9/activity/MessageList.java

```
public static void actionDisplaySearch(Context context, SearchSpecification search,
    boolean noThreading, boolean newTask) {
    actionDisplaySearch(context, search, noThreading, newTask, true);
}

public static void actionDisplaySearch(Context context, SearchSpecification search,
    boolean noThreading, boolean newTask, boolean clearTop) {
    context.startActivity(
        intentDisplaySearch(context, search, noThreading, newTask, clearTop));
}

public static Intent intentDisplaySearch(Context context, SearchSpecification search,
    boolean noThreading, boolean newTask, boolean clearTop) {
    Intent intent = new Intent(context, MessageList.class);
    intent.putExtra(EXTRA_SEARCH, ParcelableUtil.marshall(search));
    intent.putExtra(EXTRA_NO_THREADING, noThreading);

    if (clearTop) {
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    }
    if (newTask) {
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    }
}
```

```
    return intent;
}
```

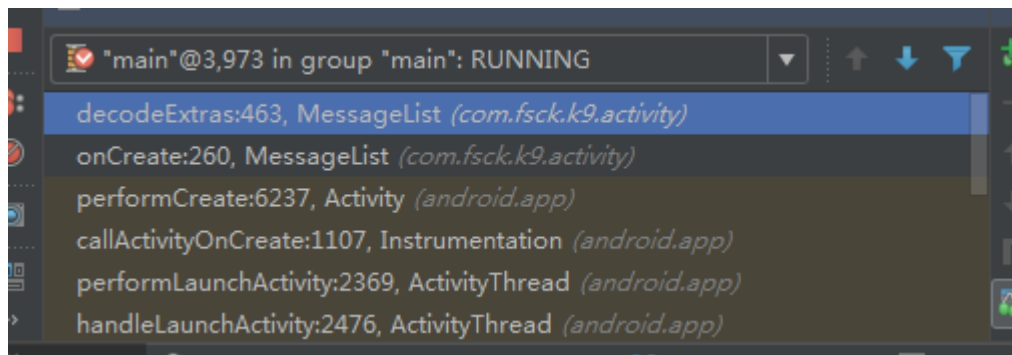
```
private boolean decodeExtras(Intent intent) {
    ...
} else if (intent.getStringExtra(SearchManager.QUERY) != null) {
    // check if this intent comes from the system search ( remote )
    if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
        //Query was received from Search Dialog
        String query = intent.getStringExtra(SearchManager.QUERY).trim();

        search = new LocalSearch(getString(R.string.search_results));
        search.setManualSearch(true);
        noThreading = true;

        search.or(new SearchCondition(SearchField.SENDER, Attribute.CONTAINS, query));
        search.or(new SearchCondition(SearchField.SUBJECT, Attribute.CONTAINS, query));
        search.or(new SearchCondition(SearchField.MESSAGE_CONTENTS, Attribute.CONTAINS, query));

        Bundle appData = intent.getBundleExtra(SearchManager.APP_DATA);
        if (appData != null) {
            search.addAccountUuid(appData.getString(EXTRA_SEARCH_ACCOUNT));
            // searches started from a folder list activity will provide an account, but no folder
            if (appData.getString(EXTRA_SEARCH_FOLDER) != null) {
```

```
        search.addAllowedFolder(appData.getString(EXTRA_SEARCH_FOLDER));
    }
} else {
    search.addAccountUuid(LocalSearch.ALL_ACCOUNTS);
}
}
```



xml/searchable.xml

```
<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:label="@string/search_action"
    android:hint="@string/search_action"
    android:voiceSearchMode="showVoiceSearchButton|launchRecognizer"
    android:voiceLanguageModel="free-form"
```

```
>  
</searchable>
```

k9mail\k-9\app\k9mail\src\main\AndroidManifest.xml

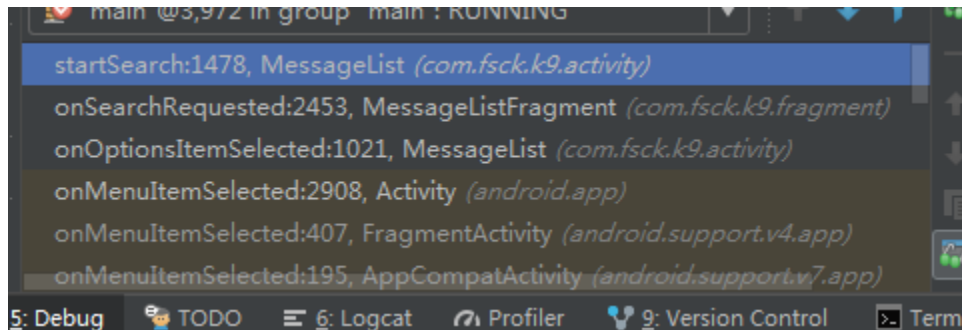
...

```
<!-- Search Activity - searchable -->  
<activity  
    android:name=".activity.Search"  
    android:configChanges="locale"  
    android:label="@string/search_action"  
    android:uiOptions="splitActionBarWhenNarrow">  
    <intent-filter>  
        <action android:name="android.intent.action.SEARCH"/>  
        <category android:name="android.intent.category.DEFAULT"/>  
    </intent-filter>  
    <meta-data  
        android:name="android.app.searchable"  
        android:resource="@xml/searchable"/>  
</activity>
```

When search is clicked in MessageList

com/fsck/k9/activity/MessageList.java

```
public boolean startSearch(Account account, String folderServerId:"INBOX") {  
    // If this search was started from a MessageList of a single folder, pass along that folder info  
    // so that we can enable remote search.  
    if (account != null && folderServerId != null) {  
        final Bundle appData = new Bundle();  
        appData.putString(EXTRA_SEARCH_ACCOUNT, account.getUuid());  
        appData.putString(EXTRA_SEARCH_FOLDER, folderServerId);  
        startSearch(null, false, appData, false);  
    } else {  
        // TODO Handle the case where we're searching from within a search result.  
        startSearch(null, false, null, false);  
    }  
}
```



Start search when enter is pressed in the system search UI

com/fsck/k9/activity/MessageList.java

```
private boolean decodeExtras(Intent intent) {  
    ...  
} else if (intent.getStringExtra(SearchManager.QUERY) != null) {  
    // check if this intent comes from the system search ( remote )  
    if (Intent.ACTION_SEARCH.equals(intent.getAction())) {  
        //Query was received from Search Dialog  
        String query = intent.getStringExtra(SearchManager.QUERY).trim();  
  
        search = new LocalSearch(getString(R.string.search_results));  
        search.setManualSearch(true);  
    }  
}
```

```

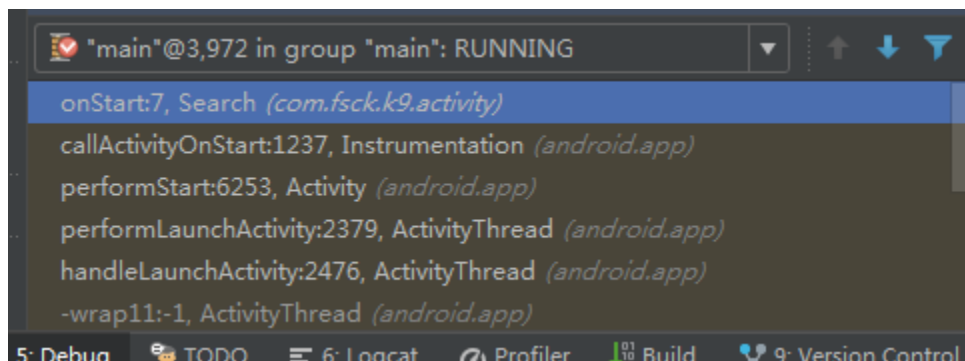
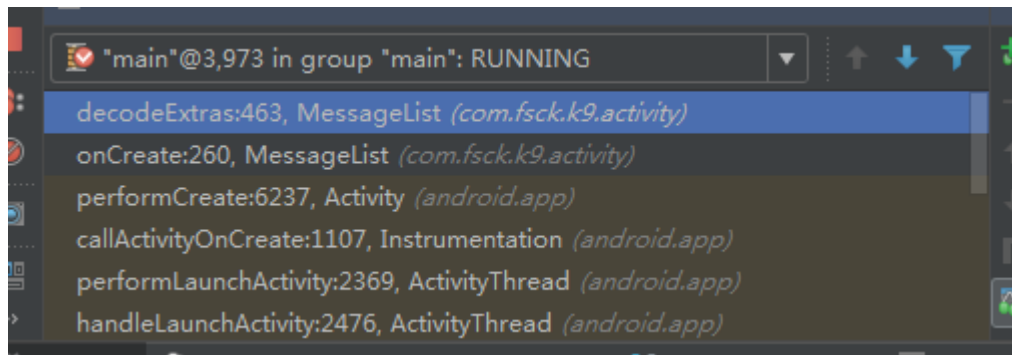
noThreading = true;

search.or(new SearchCondition(SearchField.SENDER, Attribute.CONTAINS, query));
search.or(new SearchCondition(SearchField.SUBJECT, Attribute.CONTAINS, query));
search.or(new SearchCondition(SearchField.MESSAGE_CONTENTS, Attribute.CONTAINS, query));

Bundle appData = intent.getBundleExtra(SearchManager.APP_DATA);
if (appData != null) {
    search.addAccountUuid(appData.getString(EXTRA_SEARCH_ACCOUNT));
    // searches started from a folder list activity will provide an account, but no folder
    if (appData.getString(EXTRA_SEARCH_FOLDER) != null) {
        search.addAllowedFolder(appData.getString(EXTRA_SEARCH_FOLDER));
    }
} else {
    search.addAccountUuid(LocalSearch.ALL_ACCOUNTS);
}
}

```

onCreate -> onStart



When one account is clicked:

main@3,972 in group "main": RUNNING

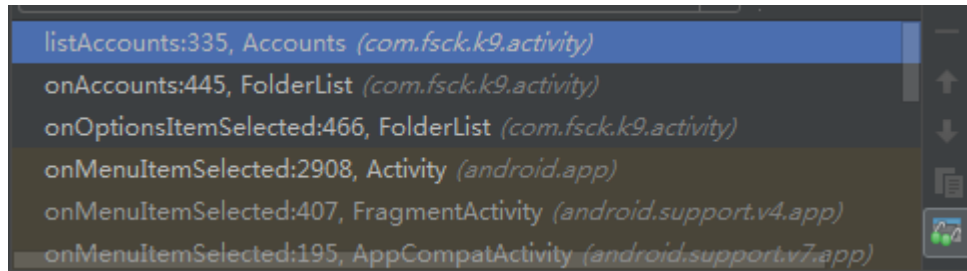
```
intentDisplaySearch:130, MessageList (com.fsck.k9.activity)
actionDisplaySearch:125, MessageList (com.fsck.k9.activity)
actionDisplaySearch:119, MessageList (com.fsck.k9.activity)
onOpenAccount:660, Accounts (com.fsck.k9.activity)
onItemClick:1211, Accounts (com.fsck.k9.activity)
performItemClick:310, AdapterView (android.widget)
```

5: Debug 6: Logcat 7: Profiler 8: Version Control

```
onOpenAccount:662, Accounts (com.fsck.k9.activity)
onItemClick:1211, Accounts (com.fsck.k9.activity)
performItemClick:310, AdapterView (android.widget)
performItemClick:1145, AbsListView (android.widget)
run:3042, AbsListView$PerformClick (android.widget)
run:3879, AbsListView$3 (android.widget)
```

5: Debug 6: Logcat 7: Profiler 8: Version Control

Return to AccountList from FolderList:



Attachment

app\ui\src\main\java\com\fsck\k9\fragment\MessageListAdapter.java

```
MessageListAdapter(MessageListFragment fragment) {  
...  
    mAttachmentIcon = fragment.getResources().getDrawable(R.drawable.ic_email_attachment_small);  
...  
}
```

```

public void bindView(View view, Context context, Cursor cursor) {
...
    boolean hasAttachments = (cursor.getInt(ATTACHMENT_COUNT_COLUMN) > 0);
...
        holder.from.setCompoundDrawablesWithIntrinsicBounds(
            statusHolder, // left
            null, // top
            hasAttachments ? mAttachmentIcon : null, // right
            null); // bottom
...
        holder.subject.setCompoundDrawablesWithIntrinsicBounds(
            statusHolder, // left
            null, // top
            hasAttachments ? mAttachmentIcon : null, // right
            null); // bottom
    }
}

```

app\ui\src\main\java\com\fsck\k9\ui\messageview\MessageContainerView.java

```

public void renderAttachments(MessageViewInfo messageViewInfo) {
    AttachmentView view =
        (AttachmentView) mInflater.inflate(R.layout.message_view_attachment, mAttachments, false);
}

```

...

```
OnMenuItemClickListener listener = new OnMenuItemClickListener() {  
    @Override  
    public boolean onMenuItemClick(MenuItem item) {  
        switch (item.getItemId()) {  
            case MENU_ITEM_IMAGE_VIEW: {  
                if (inlineImage) {  
                    attachmentCallback.onViewAttachment(attachmentViewInfo);  
                }  
            }  
        }  
    }  
};
```

...

com/fsck/k9/ui/messageview/AttachmentViewCallback.java

```
interface AttachmentViewCallback {  
    void onViewAttachment(AttachmentViewInfo attachment);  
}
```

com/fsck/k9/ui/messageview/AttachmentView.java

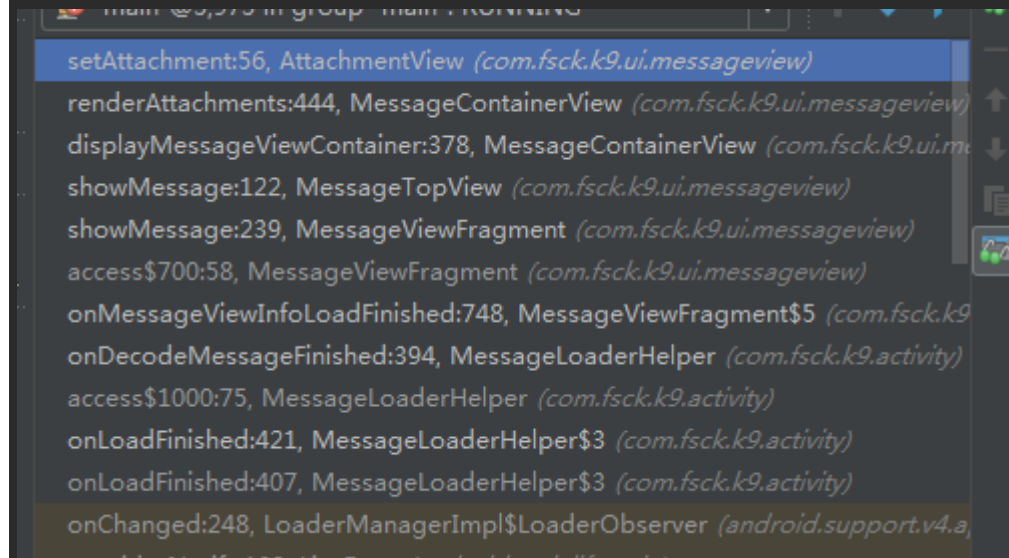
```
public void onClick(View view) {  
    int id = view.getId();  
    if (id == R.id.view) {  
        onViewButtonClick();  
    }  
}
```

```

...
public void setAttachment(AttachmentViewInfo attachment) {
    this.attachment = attachment;

    displayAttachmentInformation();
}
...

```



```

"main@3973" prio=5 runnable
  java.lang.Thread.State: RUNNABLE
    at com.fsck.k9.ui.messageview.AttachmentView.setAttachment(AttachmentView.java:56)

```

```
at com.fsck.k9.ui.messageview.MessageContainerView.renderAttachments(MessageContainerView.java:444)
at com.fsck.k9.ui.messageview.MessageContainerView.displayMessageViewContainer(MessageContainerView.java:378)
at com.fsck.k9.ui.messageview.MessageTopView.showMessage(MessageTopView.java:122)
at com.fsck.k9.ui.messageview.MessageViewFragment.showMessage(MessageViewFragment.java:239)
at com.fsck.k9.ui.messageview.MessageViewFragment.access$700(MessageViewFragment.java:58)
at com.fsck.k9.ui.messageview.MessageViewFragment$5.onMessageViewInfoLoadFinished(MessageViewFragment.java:748)
at com.fsck.k9.activity.MessageLoaderHelper.onDecodeMessageFinished(MessageLoaderHelper.java:394)
at com.fsck.k9.activity.MessageLoaderHelper.access$1000(MessageLoaderHelper.java:75)
at com.fsck.k9.activity.MessageLoaderHelper$3.onLoadFinished(MessageLoaderHelper.java:421)
at com.fsck.k9.activity.MessageLoaderHelper$3.onLoadFinished(MessageLoaderHelper.java:407)
at android.support.v4.app.LoaderManagerImpl$LoaderObserver.onChange(LoaderManagerImpl.java:248)
at android.arch.lifecycle.LiveData.considerNotify(LiveData.java:109)
at android.arch.lifecycle.LiveData.dispatchingValue(LiveData.java:126)
at android.arch.lifecycle.LiveData.setValue(LiveData.java:282)
at android.arch.lifecycle.MutableLiveData.setValue(MutableLiveData.java:33)
at android.support.v4.app.LoaderManagerImpl$LoaderInfo.setValue(LoaderManagerImpl.java:188)
at android.support.v4.app.LoaderManagerImpl$LoaderInfo.onLoadComplete(LoaderManagerImpl.java:173)
at android.support.v4.content.Loader.deliverResult(Loader.java:130)
at com.fsck.k9.ui.message.LocalMessageExtractorLoader.deliverResult(LocalMessageExtractorLoader.java:48)
at com.fsck.k9.ui.message.LocalMessageExtractorLoader.deliverResult(LocalMessageExtractorLoader.java:18)
at android.support.v4.content.AsyncTaskLoader.dispatchOnLoadComplete(AsyncTaskLoader.java:257)
at android.support.v4.content.AsyncTaskLoader$LoadTask.onPostExecute(AsyncTaskLoader.java:82)
at android.support.v4.content.ModernAsyncTask.finish(ModernAsyncTask.java:487)
at android.support.v4.content.ModernAsyncTask$InternalHandler.handleMessage(ModernAsyncTask.java:504)
at android.os.Handler.dispatchMessage(Handler.java:102)
at android.os.Looper.loop(Looper.java:148)
```

```
at android.app.ActivityThread.main(ActivityThread.java:5417)
at java.lang.reflect.Method.invoke(Method.java:-1)
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
```

com/fsck/k9/activity/MessageLoaderHelper.java

```
private LoaderCallbacks<MessageViewInfo> decodeMessageLoaderCallback = new LoaderCallbacks<MessageViewInfo>() {
...

public void onLoadFinished(Loader<MessageViewInfo> loader, MessageViewInfo messageViewInfo) {
    if (loader.getId() != DECODE_MESSAGE_LOADER_ID) {
        throw new IllegalStateException("loader id must be message decoder id");
    }
    onDecodeMessageFinished(messageViewInfo);
}
```

app/ui/src/main/java/com/fsck/k9/ui/messageview/MessageViewFragment.java

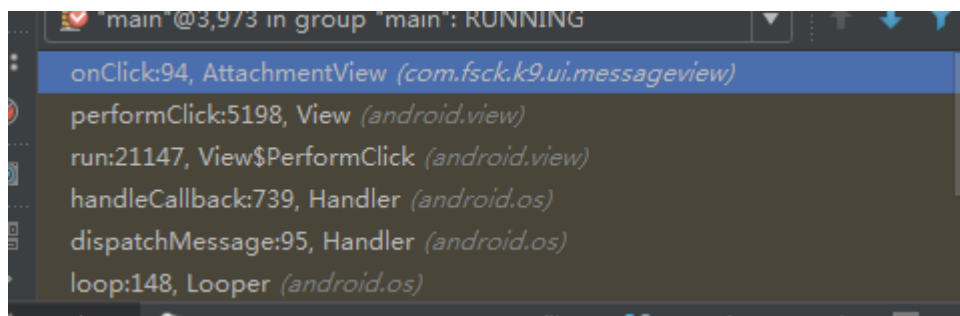
```
import com.fsck.k9.activity.MessageLoaderHelper;
import com.fsck.k9.activity.MessageLoaderHelper.MessageLoaderCallbacks;
    private MessageLoaderHelper messageLoaderHelper;
```



```

private MessageLoaderHelper messageLoaderHelper;
    messageLoaderHelper = new MessageLoaderHelper(
    messageLoaderHelper = new MessageLoaderHelper(
        messageLoaderHelper.onDestroyChangingConfigurations();
    messageLoaderHelper.onDestroy();
        messageLoaderHelper.downloadCompleteMessage();
    messageLoaderHelper.asyncStartOrResumeLoadingMessage(messageReference, null);
        messageLoaderHelper.onActivityResult(requestCode, resultCode, data);
    messageLoaderHelper.asyncReloadMessage();
    messageLoaderHelper.asyncRestartMessageCryptoProcessing();
    public void startIntentSenderForMessageLoaderHelper(IntentSender si, int requestCode, Intent fillIntent,

```



"main@3973" prio=5 runnable

java.lang.Thread.State: RUNNABLE

at com.fsck.k9.ui.messageview.AttachmentView.setAttachment(AttachmentView.java:56)

at com.fsck.k9.ui.messageview.MessageContainerView.renderAttachments(MessageContainerView.java:444)

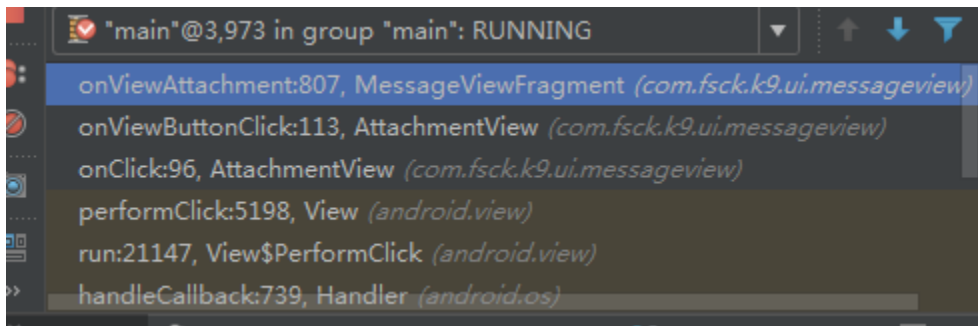
at com.fsck.k9.ui.messageview.MessageContainerView.displayMessageViewContainer(MessageContainerView.java:378)
at com.fsck.k9.ui.messageview.MessageTopView.showMessage(MessageTopView.java:122)
at com.fsck.k9.ui.messageview.MessageViewFragment.showMessage(MessageViewFragment.java:239)
at com.fsck.k9.ui.messageview.MessageViewFragment.access\$700(MessageViewFragment.java:58)
at com.fsck.k9.ui.messageview.MessageViewFragment\$5.onMessageViewInfoLoadFinished(MessageViewFragment.java:748)
at com.fsck.k9.activity.MessageLoaderHelper.onDecodeMessageFinished(MessageLoaderHelper.java:394)
at com.fsck.k9.activity.MessageLoaderHelper.access\$1000(MessageLoaderHelper.java:75)
at com.fsck.k9.activity.MessageLoaderHelper\$3.onLoadFinished(MessageLoaderHelper.java:421)
at com.fsck.k9.activity.MessageLoaderHelper\$3.onLoadFinished(MessageLoaderHelper.java:407)
at android.support.v4.app.LoaderManagerImpl\$LoaderObserver.onChanged(LoaderManagerImpl.java:248)
at android.arch.lifecycle.LiveData.considerNotify(LiveData.java:109)
at android.arch.lifecycle.LiveData.dispatchingValue(LiveData.java:126)
at android.arch.lifecycle.LiveData.setValue(LiveData.java:282)
at android.arch.lifecycle.MutableLiveData.setValue(MutableLiveData.java:33)
at android.support.v4.app.LoaderManagerImpl\$LoaderInfo.setValue(LoaderManagerImpl.java:188)
at android.support.v4.app.LoaderManagerImpl\$LoaderInfo.onLoadComplete(LoaderManagerImpl.java:173)
at android.support.v4.content.Loader.deliverResult(Loader.java:130)
at com.fsck.k9.ui.message.LocalMessageExtractorLoader.deliverResult(LocalMessageExtractorLoader.java:48)
at com.fsck.k9.ui.message.LocalMessageExtractorLoader.deliverResult(LocalMessageExtractorLoader.java:18)
at android.support.v4.content.AsyncTaskLoader.dispatchOnLoadComplete(AsyncTaskLoader.java:257)
at android.support.v4.content.AsyncTaskLoader\$LoadTask.onPostExecute(AsyncTaskLoader.java:82)
at android.support.v4.content.ModernAsyncTask.finish(ModernAsyncTask.java:487)
at android.support.v4.content.ModernAsyncTask\$InternalHandler.handleMessage(ModernAsyncTask.java:504)
at android.os.Handler.dispatchMessage(Handler.java:102)
at android.os.Looper.loop(Looper.java:148)
at android.app.ActivityThread.main(ActivityThread.java:5417)

```
at java.lang.reflect.Method.invoke(Method.java:-1)
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
```

app\ui\src\main\java\com\fsck\k9\ui\messageview\MessageViewFragment.java

```
public void onCreate(Bundle savedInstanceState) {
    ....
    messageLoaderHelper = new MessageLoaderHelper(
        context, getLoaderManager(), getFragmentManager(), messageLoaderCallbacks);

    public void onViewAttachment(AttachmentViewInfo attachment) {
        currentAttachmentViewInfo = attachment;
        getAttachmentController(attachment).viewAttachment();
    }
}
```



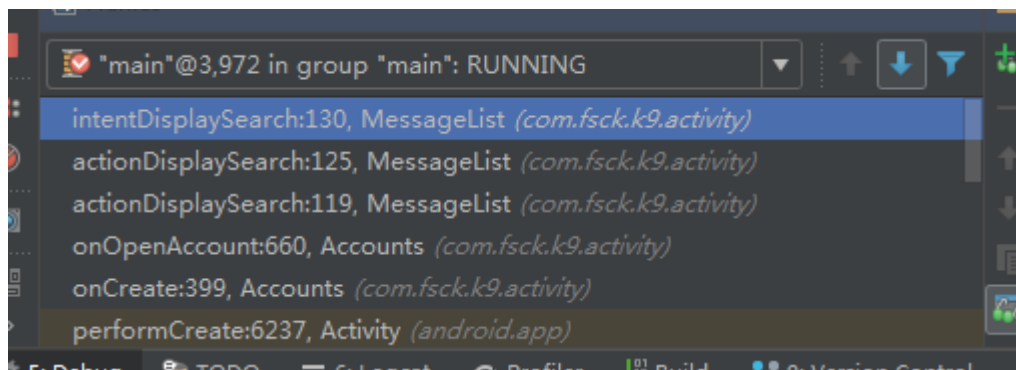
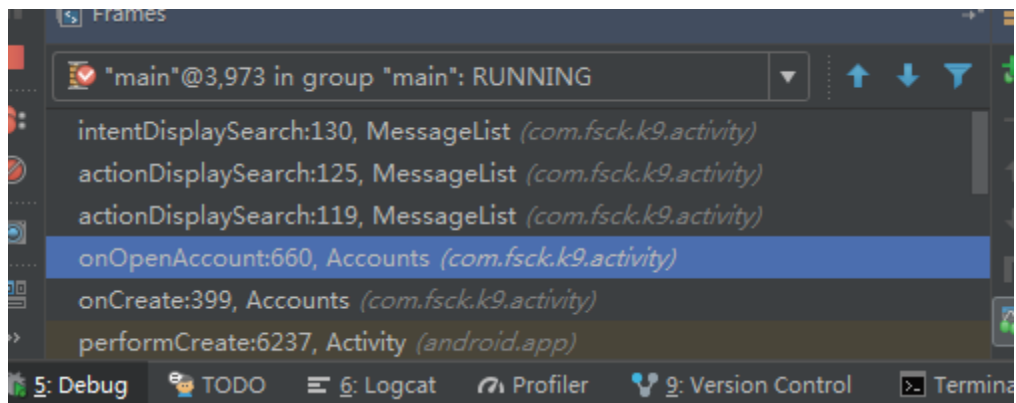
layout/message_view_attachment.xml

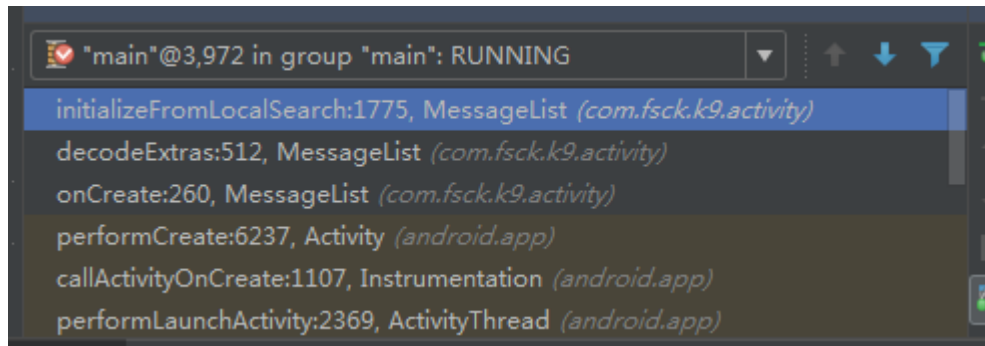
app\ui\src\main\res\layout\message.xml

```
<Button android:id="@+id/download_remainder"
        android:text="@string/message_view_download_remainder"
        android:layout_height="wrap_content"
        android:visibility="gone"
        android:layout_width="fill_parent"/>
```

app\ui\src\main\res\values\strings.xml

```
<string name="message_view_download_remainder">Download complete message</string>
```





loadMoreMessages

app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java

```
updateMoreMessagesOfCurrentFolder();
    if (currentFolder != null && !search.isManualSearch() && currentFolder.moreMessages) {
        messagingController.loadMoreMessages(account, folderServerId, null);
        activeMessages = null; // don't need it any more
        fragmentListener.showMoreFromSameSender(senderAddress);
        handler.updateFooter(String.format(context.getString(R.string.load_more_messages_fmt), maxResults));
        updateFooter(context.getString(R.string.status_loading_more));
    } else if (!currentFolder.moreMessages) {
```

```
        message = context.getString(R.string.message_list_load_more_messages_action);
        message = String.format(context.getString(R.string.load_more_messages_fmt),
        message = context.getString(R.string.status_loading_more_failed);
    void showMoreFromSameSender(String senderAddress);
    private void updateMoreMessagesOfCurrentFolder() {
        currentFolder.setMoreMessagesFromFolder(folder);
```

```
private void updateFooterView() {
    if (!search.isManualSearch() && currentFolder != null && account != null) {
        if (currentFolder.loading) {
            updateFooter(context.getString(R.string.status_loading_more));
        } else if (!currentFolder.moreMessages) {
            updateFooter(null);
        } else {
            String message;
            if (!currentFolder.lastCheckFailed) {
                if (account.getDisplayCount() == 0) {
                    message = context.getString(R.string.message_list_load_more_messages_action);
                } else {
                    message = String.format(context.getString(R.string.load_more_messages_fmt),
                    account.getDisplayCount());
                }
            }
            message = context.getString(R.string.status_loading_more_failed);
        }
        updateFooter(message);
    }
```

```
    }  
    } else {  
        updateFooter(null);  
    }  
}
```

com/fsck/k9/controller/MessagingController.java

```
public void loadMoreMessages(Account account, String folder, MessagingListener listener) {  
    try {  
        LocalStore localStore = account.getLocalStore();  
        LocalFolder localFolder = localStore.getFolder(folder);  
        if (localFolder.getVisibleLimit() > 0) {  
            localFolder.setVisibleLimit(localFolder.getVisibleLimit() + account.getDisplayCount());  
        }  
        synchronizeMailbox(account, folder, listener, null);  
    } catch (MessagingException me) {  
        throw new RuntimeException("Unable to set visible limit on folder", me);  
    }  
}  
  
/**  
 * Start background synchronization of the specified folder.  
 */
```



```
public void synchronizeMailbox(final Account account, final String folder, final MessagingListener listener,
    final Folder providedRemoteFolder) {
    putBackground("synchronizeMailbox", listener, new Runnable() {
        @Override
        public void run() {
            synchronizeMailboxSynchronous(account, folder, listener, providedRemoteFolder);
        }
    });
}
```

app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java

```
messagingController = MessagingController.getInstance(getActivity().getApplication());
```

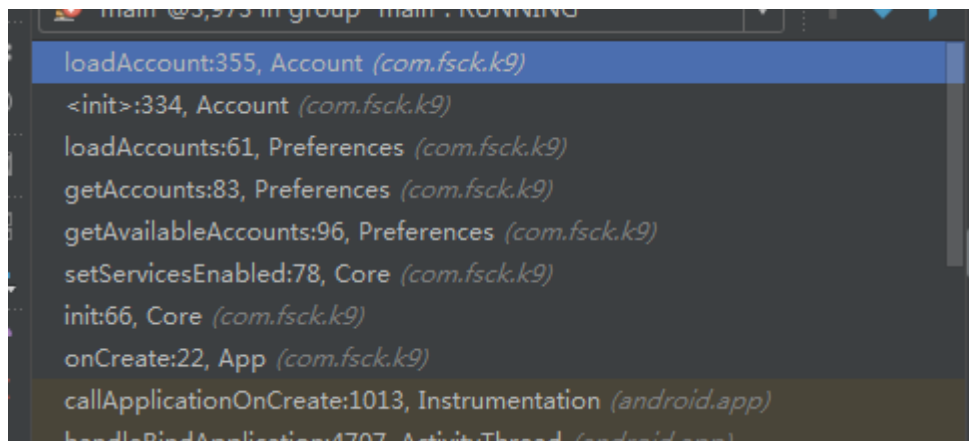
```
intentDisplaySearch:132, MessageList (com.fsck.k9.activity)
buildMessageListBackStack:280, K9NotificationActionCreator (com.fsck.k9.notification)
buildMessageViewBackStack:290, K9NotificationActionCreator (com.fsck.k9.notification)
createViewMessagePendingIntent:46, K9NotificationActionCreator (com.fsck.k9.notification)
createBigTextStyleNotification:48, BaseNotifications (com.fsck.k9.notification)
```

buildStackedNotification:28, WearNotifications (com.fsck.k9.notification)
createStackedNotification:154, NewMailNotifications (com.fsck.k9.notification)
addNewMailNotification:54, NewMailNotifications (com.fsck.k9.notification)
addNewMailNotification:84, NotificationController (com.fsck.k9.notification)
syncNewMessage:3120, MessagingController\$ControllerSyncListener (com.fsck.k9.controller)
messageFinished:571, ImapSync\$3 (com.fsck.k9.backend.imap)
fetch:753, ImapFolder (com.fsck.k9.mail.store.imap)
downloadSmallMessages:541, ImapSync (com.fsck.k9.backend.imap)
downloadMessages:376, ImapSync (com.fsck.k9.backend.imap)
synchronizeMailboxSynchronous:210, ImapSync (com.fsck.k9.backend.imap)
sync:52, ImapSync (com.fsck.k9.backend.imap)
sync:108, ImapBackend (com.fsck.k9.backend.imap)
syncFolder:715, MessagingController (com.fsck.k9.controller)
synchronizeMailboxSynchronous:693, MessagingController (com.fsck.k9.controller)
run:678, MessagingController\$10 (com.fsck.k9.controller)
runInBackground:210, MessagingController (com.fsck.k9.controller)
access\$000:108, MessagingController (com.fsck.k9.controller)
run:153, MessagingController\$1 (com.fsck.k9.controller)
run:818, Thread (java.lang)

displayCount

com/fsck/k9/Account.java

```
/**
 * Load stored settings for this account.
 */
private synchronized void loadAccount(Preferences preferences) {
    ...
    displayCount = storage.getInt(accountUuid + ".displayCount", K9.DEFAULT_VISIBLE_LIMIT);
    if (displayCount < 0) {
        displayCount = K9.DEFAULT_VISIBLE_LIMIT;
    }
}
```



com/fsck/k9/K9.java

```
/**
 * Specifies how many messages will be shown in a folder by default. This number is set
 * on each new folder and can be incremented with "Load more messages..." by the
 * VISIBLE_LIMIT_INCREMENT
 */
public static final int DEFAULT_VISIBLE_LIMIT = 25;
```

Message Item Clicked

app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
```

```
...
```

```
    // This item represents a message; just display the message.
```

```
    openMessageAtPosition(listViewToAdapterPosition(position));
```

```
...
```

```
private void openMessageAtPosition(int position) {
```

```
...
```

```
MessageReference ref = getReferenceForPosition(position);
```

```
// For some reason the listView.setSelection() above won't do anything when we call
// onOpenMessage() (and consequently adapter.notifyDataSetChanged()) right away. So we
// defer the call using MessageListHandler.
handler.openMessage(ref);
```

com/fsck/k9/fragment/MessageListHandler.java

```
public void openMessage(MessageReference messageReference) {
    android.os.Message msg = android.os.Message.obtain(this, ACTION_OPEN_MESSAGE,
        messageReference);
    sendMessage(msg);
}
...
public void handleMessage(android.os.Message msg) {
    ...
    case ACTION_OPEN_MESSAGE: {
        MessageReference messageReference = (MessageReference) msg.obj;
        fragment.fragmentListener.openMessage(messageReference);
        break;
    }
}
```

com/fsck/k9/activity/MessageList.java

```
public void openMessage(MessageReference messageReference) {
    Account account = preferences.getAccount(messageReference.getAccountUuid());
    String folderServerId = messageReference.getFolderServerId();
    ...
    MessageViewFragment fragment = MessageViewFragment.newInstance(messageReference);
    ...
    if (displayMode != DisplayMode.SPLIT_VIEW) {
        showMessageView();
    }
}
```

com/fsck/k9/ui/messageview/MessageViewFragment.java

```
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);

    Bundle arguments = getArguments();
    String messageReferenceString = arguments.getString(ARG_REFERENCE);
    MessageReference messageReference = MessageReference.parse(messageReferenceString);
}
```

```

        displayMessage(messageReference);
    }

    private void displayMessage(MessageReference messageReference) {
        ...

        mAccount = Preferences.getPreferences(getApplicationContext()).getAccount(mMessageReference.getAccountUuid());
        messageLoaderHelper.asyncStartOrResumeLoadingMessage(messageReference, null);
        ...
    }

```

com/fsck/k9/activity/MessageLoaderHelper.java

```

    public void downloadCompleteMessage() {
        startDownloadingMessageBody(true);
    }

    ...

    public void onLoadFinished(Loader<MessageViewInfo> loader, MessageViewInfo messageViewInfo) {
        if (loader.getId() != DECODE_MESSAGE_LOADER_ID) {
            throw new IllegalStateException("loader id must be message decoder id");
        }
        onDecodeMessageFinished(messageViewInfo);
    }

```

com/fsck/k9/ui/message/LocalMessageExtractorLoader.java

```
public void deliverResult(MessageViewInfo messageViewInfo) {  
    this.messageViewInfo = messageViewInfo;  
    super.deliverResult(messageViewInfo);  
}
```

```
main @5,375 in group main: K9MailView  
setAttachment:56, AttachmentView (com.fsck.k9.ui.messageview)  
renderAttachments:444, MessageContainerView (com.fsck.k9.ui.messageview)  
displayMessageViewContainer:378, MessageContainerView (com.fsck.k9.ui.messageview)  
showMessage:122, MessageTopView (com.fsck.k9.ui.messageview)  
showMessage:239, MessageViewFragment (com.fsck.k9.ui.messageview)  
access$700:58, MessageViewFragment (com.fsck.k9.ui.messageview)  
onMessageViewInfoLoadFinished:748, MessageViewFragment$5 (com.fsck.k9.ui.messageview)  
onDecodeMessageFinished:394, MessageLoaderHelper (com.fsck.k9.activity)  
access$1000:75, MessageLoaderHelper (com.fsck.k9.activity)  
onLoadFinished:421, MessageLoaderHelper$3 (com.fsck.k9.activity)  
onLoadFinished:407, MessageLoaderHelper$3 (com.fsck.k9.activity)  
onChanged:248, LoaderManagerImpl$LoaderObserver (android.support.v4.a...
```


PullToRefresh

app\ui\src\main\java\com\fsck\k9\fragment\MessageListFragment.java

```
swipeRefreshLayout.setOnRefreshListener(  
    new SwipeRefreshLayout.OnRefreshListener() {  
        public void onRefresh() {  
swipeRefreshLayout.setOnRefreshListener(  
    new SwipeRefreshLayout.OnRefreshListener() {  
        public void onRefresh() {  
  
private void initializePullToRefresh(View layout) {  
    swipeRefreshLayout = (SwipeRefreshLayout) layout.findViewById(R.id.swiperefresh);  
    listView = (ListView) layout.findViewById(R.id.message_list);  
  
    } else if (isCheckMailSupported()) {  
        swipeRefreshLayout.setOnRefreshListener(  
            new SwipeRefreshLayout.OnRefreshListener() {  
                @Override  
                public void onRefresh() {  
                    checkMail();  
                }  
            }  
        );  
    }  
};
```

```
}

// Disable pull-to-refresh until the message list has been loaded
swipeRefreshLayout.setEnabled(false);
}
```

```
public void checkMail() {
    if (isSingleAccountMode() && isSingleFolderMode()) {
        messagingController.synchronizeMailbox(account, folderServerId, activityListener, null);
        messagingController.sendPendingMessages(account, activityListener);
    } else if (allAccounts) {
        messagingController.checkMail(context, null, true, true, activityListener);
    } else {
        for (String accountId : accountUids) {
            Account account = preferences.getAccount(accountId);
            messagingController.checkMail(context, account, true, true, activityListener);
        }
    }
}
```

download_remainder

app\ui\src\main\java\com\fsck\k9\ui\messageview\MessageTopView.java

```
public void onFinishInflate() {  
    ....  
    mDownloadRemainder = (Button) findViewById(R.id.download_remainder);  
    mDownloadRemainder.setVisibility(View.GONE);  
}
```