**Setup Wizard Library**

The Android Setup Wizard UI Library enables you to easily create Setup Wizard-like screens in other apps, including relevant parts of GMS Core, Google Play, and OEM customizations. The source code of the Setup Wizard Library is available in the Android Open Source Project (AOSP) at [/frameworks/opt/setupwizard/library](https://android.googlesource.com/platform/frameworks/opt/setupwizard/+/master/library/" \t "_blank).

**Build configurations**

The SUW UI library supports the following build configurations.

**Using Make**

If your app is in the Android Git tree and uses Make to build, first ensure the project is included in the branch manifest then include the common.mk file. You must explicitly set your LOCAL\_RESOURCE\_DIR before including [common-full-support.mk](https://android.googlesource.com/platform/frameworks/opt/setupwizard/+/master/library/common-full-support.mk" \t "_blank).

LOCAL\_RESOURCE\_DIR := $(LOCAL\_PATH)/res  
include frameworks/opt/setupwizard/library/common-full-support.mk

|  |
| --- |
|  |

**Using Gradle**

You can also build against the [Android Archive (AAR)](https://developer.android.com/studio/projects/android-library.html" \t "_blank)file. For details on referencing a library project in gradle, refer to the [Gradle Plugin User Guide](http://tools.android.com/tech-docs/new-build-system/user-guide" \l "TOC-Referencing-a-Library" \t "_blank). Example:

flatDir {  
       dirs 'libs'  
  }  
}

dependencies {  
  compile(name:'setup-wizard-lib', ext:'aar')  
}

**Available build variants**

The library is available in the following build variants:

* **fullSupport (recommended)**. This variant builds against the RecyclerView support libraries and allows provides additional classes to help with building a Setup Wizard screen with recycler views.
* **platform**. This variant does not depend on support libraries but is compatible only with the latest Android version release. Use for apps bundled with the system image or apps that otherwise do not need backwards compatibility.
* **icsCompat**. This variant provides compatibility back to Android 4.0 (Ice Cream Sandwich). Depends on the appcompat support library.
* **eclairMr1Compat**. This variant provides compatibility back to Android 2.0/2.1 (Eclair and Eclair MR1). Depends on the appcompat support library.

**TIP:** For details on Android releases, refer to [Codenames, Versions, API Levels, and NDK Releases](http://source.android.com/source/build-numbers.html" \l "platform-code-names-versions-api-levels-and-ndk-releases" \t "_blank).

**Using the layout**

To create an activity with setup wizard styles, use the GlifLayout class. It is a FrameLayout that automatically inflates the setup wizard template and adds its child views to the content area.

For example, to make the additional screens resemble the Google Setup Wizard, use:

<com.android.setupwizardlib.GlifLayout  
   xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:icon="@drawable/my\_icon"  
   app:suwHeaderText="@string/my\_header">

   <TextView  
       android:layout\_width="match\_parent"  
       android:layout\_height="wrap\_content"  
       android:text="@string/content\_text" />

</com.android.setupwizardlib.GlifLayout>

**IMPORTANT:** GlifLayout makes use of the attributes in the theme to provide different resources for dark and light themes. Any activity using GlifLayout must use a theme that inherits from SuwThemeGlif or SuwThemeGlif.Light.

**GlifLayout components**

GlifLayout provides the following components:

| **Phones** | **Tablets** |
| --- | --- |
| https://lh3.googleusercontent.com/WYqn8qNnNrIBCT6KRIQnOyKR3jhJPHYjuN7cspIRlcBj53UrSyChIDBs4SJpPzckJUIg=w313 | https://lh3.googleusercontent.com/eC4ZLhjUz5_Hlwmi_sIHobBBPOOmX0vInQS3Jh0aNkBZHvvVN0Cizs0mTB2qnyn8Y6vH=w624 |

* **pattern**. The background is different for phone and tablet layouts. On phones, the pattern background covers only the status bar area; on tablets, the pattern background covers the entire screen. The background is implemented as GlifPatternDrawable in code.
* **icon** and **header text**. These elements appear at the top of the layout and should scroll with the content. They can be customized in XML using android:icon and app:suwHeaderText, respectively.

For details on designing setup wizard screens, refer to the Setup Wizard Design Guidelines:

* Setup Wizard Design Guidelines (Android 7.1)​
* Setup Wizard Design Guidelines (Android 7.0)​
* Setup Wizard Design Guidelines (Android 6.0)

**GlifLayout family**

GlifLayout includes the following classes and subclasses.

**GlifLayout**

GlifLayout is the basic layout to implement a Setup Wizard screen.

**GlifListLayout**

GlifListLayout is a subclass of GlifLayout that provides the theme but instead of a ScrollView, it provides a ListView directly. GlifListLayout can be populated by setting the adapter calling GlifListLayout#setAdapter method, or integrating directly with the items framework, so an items XML can be specified in XML using android:entries.

<com.android.setupwizardlib.GlifListLayout  
   xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   android:id="@+id/setup\_wizard\_layout"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:entries="@xml/my\_items\_xml"  
   android:icon="@drawable/my\_icon"  
   app:suwHeaderText="@string/my\_title" />

**GlifRecyclerLayout**

Available only in full-support build variant. GlifRecyclerLayout is a subclass of GlifLayout, providing a similar interface to GlifListLayout but implemented using a RecyclerView instead. To get RecyclerView-related functionalities, you must use the full-support build variant of Setup Wizard Library.

<com.android.setupwizardlib.GlifRecyclerLayout  
   xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   android:id="@+id/setup\_wizard\_layout"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:entries="@xml/my\_items"  
   android:icon="@drawable/my\_icon"  
   app:suwHeaderText="@string/my\_title" />

**Handling clicks**

You can use GlifListLayout or GlifRecyclerLayout to handle clicks.

**Using GlifListLayout**

List views have built-in functionalities for handling clicks, so you can simply attach an OnItemClickListener onto the list view to listen for clicks. The following example is for use in conjunction with the items framework:

使用GlifListLayout

列表视图有用于处理点击的内置功能，所以你可以简单地将一个OnItemClickListener附加到列表视图中，以监听点击。下面的例子是与项目框架一起使用的：

final GlifListLayout layout = (GlifListLayout) findViewById(R.id.setup\_wizard\_layout);  
layout.getListView().setOnItemClickListener(new OnItemClickListener() {  
   @Override  
   public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
       Item item = (Item) parent.getItemAtPosition(position);  
       switch (item.getId()) {  
           case R.id.item\_1:  
               // do stuff  
               break;  
       }  
   }  
});

**Using GlifRecyclerLayout**

Recycler views do not have general support to listen to an item click. However, if you use the items framework, RecyclerItemAdapter provides setOnItemSelectedListener to listen for clicks on individual items of the recycler view. Example:

使用GlifRecyclerLayout

回收者的视图没有一般的支持来监听项目的点击。但是，如果您使用项目框架，那么RecyclerItemAdapter 提供setOnItemSelectedListener来监听回收器视图的单个项目的点击。例子:

@Override  
public void onItemSelected(IItem item) {  
   switch (((Item) item).getId()) {  
       case R.id.item\_1:  
           // do stuff  
           break;  
       case R.id.item\_2:  
           // do stuff  
           break;  
   }  
}

**Items framework**

You can create a list in XML using the android:entries attribute or ItemInflater. For example, a simple XML (my\_items.xml):

项目框架

您可以使用android:entries attribute or ItemInflater. 在XML中创建一个列表。例如，一个简单的XML（myitems.XML）：

<ItemGroup xmlns:android="http://schemas.android.com/apk/res/android">  
   
    <Item  
        android:enabled="false"  
        android:layout="@layout/item\_layout\_description\_empty" />  
   
    <Item  
        android:id="@+id/restore\_choice\_android\_device"  
        android:icon="@drawable/ic\_restore\_choice\_android\_device"  
        android:title="@string/restore\_choice\_android\_device\_label"  
        android:summary="@string/restore\_choice\_android\_device\_summary" />  
   
    <Item  
        android:id="@+id/restore\_choice\_google\_account"  
        android:icon="@drawable/ic\_restore\_choice\_google\_account"  
        android:title="@string/restore\_choice\_google\_account\_label"  
        android:summary="@string/restore\_choice\_google\_account\_summary" />  
   
    <Item  
        android:id="@+id/restore\_choice\_ios\_device"  
        android:icon="@drawable/ic\_restore\_choice\_ios\_device"  
        android:title="@string/restore\_choice\_ios\_device\_label" />  
   
</ItemGroup>

To install this XML into a GlifRecyclerLayout, use the android:entries attribute. Example:

要将此XML安装到GlifRecyclerLayout，请使用android:entries属性。例子:

<com.android.setupwizardlib.GlifRecyclerLayout  
   xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   android:id="@+id/setup\_wizard\_layout"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:entries="@xml/my\_items"  
   android:icon="@drawable/my\_icon"  
   app:suwHeaderText="@string/my\_title" />

**Using helper classes**

The Setup Wizard Library includes helper classes so clients can integrate with the setup flow more easily.

For example, if an additional screen is added using the OEM\_POST\_SETUP intent action or the wizard\_script via overlay, you will need to start the next step. The WizardManagerHelper class provides methods to help with that task:

使用辅助类

安装向导库包含助手类，这样客户端就可以更容易地与安装流程集成。

例如，如果使用OEM\_POST\_SETUP intent action or the wizard\_script via overlay,，您将需要启动下一步。向导助手类提供了帮助完成该任务的方法：

@Override  
public void onNavigateNext() {  
Intent nextIntent = WizardManagerHelper.getNextIntent(getIntent(), RESULT\_OK);  
startActivityForResult(nextIntent, REQUEST\_CODE);  
}