[How to change the WindowManager.LayoutParams' type programmatically?](https://stackoverflow.com/questions/43432591/how-to-change-the-windowmanager-layoutparams-type-programmatically)

[Ask Question](https://stackoverflow.com/questions/ask)

3

I got a view that I am able to display on the lockscreen. But I want it to make it optional. Currently I use these params to achieve this:

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.TYPE\_SYSTEM\_ERROR,

WindowManager.LayoutParams.FLAG\_SHOW\_WHEN\_LOCKED |

WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE |

WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH |

WindowManager.LayoutParams.FLAG\_LAYOUT\_NO\_LIMITS, PixelFormat.TRANSPARENT);

And for not showing it on the lockscreen I use

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.TYPE\_PHONE,

WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE |

WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH |

WindowManager.LayoutParams.FLAG\_LAYOUT\_NO\_LIMITS, PixelFormat.TRANSPARENT);

Separately they work fine, but I want to change the type and flags programmatically. So I tried changing the WindowManager.LayoutParams with

params.flags &= ~WindowManager.LayoutParams.TYPE\_SYSTEM\_ERROR;

params.flags &= ~WindowManager.LayoutParams.FLAG\_SHOW\_WHEN\_LOCKED;

params.flags |= WindowManager.LayoutParams.TYPE\_PHONE;

windowManager.updateViewLayout(myView,params);

to remove the type, remove the flag and set a new type, but that doesn't seem to work. Does anyone know how to do this correctly?

[[https://i.stack.imgur.com/bk9VA.png](https://stackoverflow.com/questions/tagged/android)android](https://stackoverflow.com/questions/tagged/android) [lockscreen](https://stackoverflow.com/questions/tagged/lockscreen) [android-windowmanager](https://stackoverflow.com/questions/tagged/android-windowmanager)

[share](https://stackoverflow.com/q/43432591)[improve this question](https://stackoverflow.com/posts/43432591/edit)

asked Apr 16 '17 at 0:38

[[https://i.stack.imgur.com/OZqtc.jpg?s=32&g=1](https://stackoverflow.com/users/2890156/denny)](https://stackoverflow.com/users/2890156/denny)

[Denny](https://stackoverflow.com/users/2890156/denny)

**1,286**21126

add a comment

1 Answer

[active](https://stackoverflow.com/questions/43432591/how-to-change-the-windowmanager-layoutparams-type-programmatically?answertab=active#tab-top)[oldest](https://stackoverflow.com/questions/43432591/how-to-change-the-windowmanager-layoutparams-type-programmatically?answertab=oldest#tab-top)[votes](https://stackoverflow.com/questions/43432591/how-to-change-the-windowmanager-layoutparams-type-programmatically?answertab=votes#tab-top)

2

You are setting type values into flag. Try this instead:

params.flags &= ~WindowManager.LayoutParams.FLAG\_SHOW\_WHEN\_LOCKED;

// type is no bit flag, so this should do it

params.type = WindowManager.LayoutParams.TYPE\_PHONE;

# [Creating a system overlay window (always on top)](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top)

[Ask Question](https://stackoverflow.com/questions/ask)

255

247

I'm trying to create an always-op-top button/clickable-image which stays on top of all the windows all the time.

The proof of concept is

* here - [Smart Taskbar (on AppBrain)](http://www.appbrain.com/app/smart-taskbar-%28sidebar%29/com.smart.taskbar)
* and here [V1.4.0 Sidebar style SWKey - Button savior (on xda-developers)](http://forum.xda-developers.com/showthread.php?t=865525)

I have been successful and have a running service now. The service displays some text on top left corner of screen all the time while user can freely interact with rest of apps in normal manner.

What I'm doing is subclass ViewGroup and add it to root window manager with flag TYPE\_SYSTEM\_OVERLAY. Now I want to add a button/clickable-image in place of this text which can receive touch events on itself. I tried overriding "onTouchEvent" for the whole ViewGroup but it does not receive any event.

How can I receive events only on certain parts of my always-on-top view group? Kindly suggest.

public class HUD extends Service {

HUDView mView;

@Override

public IBinder onBind(Intent intent) {

return null;

}

@Override

public void onCreate() {

super.onCreate();

Toast.makeText(getBaseContext(),"onCreate", Toast.LENGTH\_LONG).show();

mView = new HUDView(this);

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.TYPE\_SYSTEM\_OVERLAY,

0,

// WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE

// | WindowManager.LayoutParams.FLAG\_NOT\_TOUCHABLE,

PixelFormat.TRANSLUCENT);

params.gravity = Gravity.RIGHT | Gravity.TOP;

params.setTitle("Load Average");

WindowManager wm = (WindowManager) getSystemService(WINDOW\_SERVICE);

wm.addView(mView, params);

}

@Override

public void onDestroy() {

super.onDestroy();

Toast.makeText(getBaseContext(),"onDestroy", Toast.LENGTH\_LONG).show();

if(mView != null)

{

((WindowManager) getSystemService(WINDOW\_SERVICE)).removeView(mView);

mView = null;

}

}

}

class HUDView extends ViewGroup {

private Paint mLoadPaint;

public HUDView(Context context) {

super(context);

Toast.makeText(getContext(),"HUDView", Toast.LENGTH\_LONG).show();

mLoadPaint = new Paint();

mLoadPaint.setAntiAlias(true);

mLoadPaint.setTextSize(10);

mLoadPaint.setARGB(255, 255, 0, 0);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

canvas.drawText("Hello World", 5, 15, mLoadPaint);

}

@Override

protected void onLayout(boolean arg0, int arg1, int arg2, int arg3, int arg4) {

}

@Override

public boolean onTouchEvent(MotionEvent event) {

//return super.onTouchEvent(event);

Toast.makeText(getContext(),"onTouchEvent", Toast.LENGTH\_LONG).show();

return true;

}

}

[[https://i.stack.imgur.com/bk9VA.png](https://stackoverflow.com/questions/tagged/android)android](https://stackoverflow.com/questions/tagged/android) [overlay](https://stackoverflow.com/questions/tagged/overlay)

[share](https://stackoverflow.com/q/4481226)[improve this question](https://stackoverflow.com/posts/4481226/edit)

[edited Mar 30 '17 at 5:45](https://stackoverflow.com/posts/4481226/revisions)

[[https://i.stack.imgur.com/4jGJ2.jpg?s=32&g=1](https://stackoverflow.com/users/6243553/harshit-agrawal)](https://stackoverflow.com/users/6243553/harshit-agrawal)

[Harshit Agrawal](https://stackoverflow.com/users/6243553/harshit-agrawal)

**1,309**41526

asked Dec 19 '10 at 1:57

[[https://www.gravatar.com/avatar/e49bbc0e01f35c64a49a955736ff9b5d?s=32&d=identicon&r=PG](https://stackoverflow.com/users/453906/coresolo)](https://stackoverflow.com/users/453906/coresolo)

[coreSOLO](https://stackoverflow.com/users/453906/coresolo)

**1,501**41420

* 1

@coreSOLO, I hope you get you're answer, my oddly similar questions didnt get any response. :) +1 – [st0le](https://stackoverflow.com/users/216517/st0le) [Dec 19 '10 at 6:44](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4901076_4481226)

* 2

@st0le I promise I'll spend my night and day until I find the solution and share it with you :) – [coreSOLO](https://stackoverflow.com/users/453906/coresolo) [Dec 19 '10 at 8:02](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4901287_4481226)

* @coreSOLO, hehe ;) – [st0le](https://stackoverflow.com/users/216517/st0le) [Dec 19 '10 at 8:15](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4901322_4481226)
* @st0le Ok maybe you and me can dig it out together, give me a link of your question and let me know how much progress you were able to make... – [coreSOLO](https://stackoverflow.com/users/453906/coresolo) [Dec 19 '10 at 9:01](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4901467_4481226)
* 2

It's surprising that your onDraw() is being called. Shouldn't dispatchDraw() be overridden instead? See[groups.google.com/forum/?fromgroups#!topic/android-developers/…](https://groups.google.com/forum/?fromgroups#!topic/android-developers/oLccWfszuUo) – [faizal](https://stackoverflow.com/users/2105986/faizal) [Jul 7 '14 at 17:39](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment38146744_4481226)

[show **2** more comments](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top)

## 15 Answers

[active](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top?answertab=active#tab-top)[oldest](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top?answertab=oldest#tab-top)[votes](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top?answertab=votes#tab-top)

148

This might be a stupid solution. But it works. If you can improve it, please let me know.

OnCreate of your Service: I have used WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH flag. This is the only change in service.

@Override

public void onCreate() {

super.onCreate();

Toast.makeText(getBaseContext(),"onCreate", Toast.LENGTH\_LONG).show();

mView = new HUDView(this);

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

WindowManager.LayoutParams.TYPE\_SYSTEM\_OVERLAY,

WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH,

PixelFormat.TRANSLUCENT);

params.gravity = Gravity.RIGHT | Gravity.TOP;

params.setTitle("Load Average");

WindowManager wm = (WindowManager) getSystemService(WINDOW\_SERVICE);

wm.addView(mView, params);

}

Now, you will start getting each and every click event. So, you need to rectify in your event handler.

In your ViewGroup touch event

@Override

public boolean onTouchEvent(MotionEvent event) {

// ATTENTION: GET THE X,Y OF EVENT FROM THE PARAMETER

// THEN CHECK IF THAT IS INSIDE YOUR DESIRED AREA

Toast.makeText(getContext(),"onTouchEvent", Toast.LENGTH\_LONG).show();

return true;

}

Also you may need to add this permission to your manifest:

<uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" />

[share](https://stackoverflow.com/a/4482868)[improve this answer](https://stackoverflow.com/posts/4482868/edit)

[edited May 6 '15 at 18:15](https://stackoverflow.com/posts/4482868/revisions)

[[https://www.gravatar.com/avatar/08666e6d230bc177a3a13b3a2b8e4b27?s=32&d=identicon&r=PG](https://stackoverflow.com/users/2105657/teo-inke)](https://stackoverflow.com/users/2105657/teo-inke)

[Teo Inke](https://stackoverflow.com/users/2105657/teo-inke)

**4,043**13034

answered Dec 19 '10 at 12:12

[[https://i.stack.imgur.com/6NEKo.jpg?s=32&g=1](https://stackoverflow.com/users/408863/sarwar-erfan)](https://stackoverflow.com/users/408863/sarwar-erfan)

[Sarwar Erfan](https://stackoverflow.com/users/408863/sarwar-erfan)

**16.7k**43755

* 1

@pengwang: what errors are you getting? This code is not independent, you need to place them in your project with appropriate file names under appropriate project. – [Sarwar Erfan](https://stackoverflow.com/users/408863/sarwar-erfan) [Dec 28 '10 at 3:34](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4979629_4482868)

* 3

@pengwang: Find simplest implementation here: [docs.google.com/…](https://docs.google.com/leaf?id=0B9OlwWEcpDrCNTc1N2Y3NTAtMWJkZi00NzhiLTg4ODgtMDNjZTAxZWIwOTM0&hl=en) – [Sarwar Erfan](https://stackoverflow.com/users/408863/sarwar-erfan) [Dec 28 '10 at 12:07](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4982307_4482868)

* 6

the whole code is correct,i find my mistake,i forget the link <uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" /> Sarwar Erfan i think you edit your answer,it can become more wonderful – [pengwang](https://stackoverflow.com/users/233618/pengwang) [Dec 28 '10 at 13:23](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4982900_4482868)

* 1

This answer is very helpful, except for one thing. As used in the code, WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH, according to the documentation, "will not receive the full up/move/down gesture". In other words, it not possible to detect swipes or anything else other than a ACTION\_DOWN. Well I know that it is possible to detect swipes because Wave Launcher uses this similar concept and they are able to detect swipes. Can anyone explain to me how this works? – [Brian](https://stackoverflow.com/users/394933/brian) [Jul 17 '11 at 7:07](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment7961744_4482868)

* 6

@SarwarErfan - I used this its corking perfectly on 2.3 but onTouch event is not geeting invocked on Android 4.0, 1nd 4.1. Please help – [Sanjay Singh](https://stackoverflow.com/users/1773788/sanjay-singh) [Mar 2 '14 at 13:57](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment33574009_4482868)

[show **13** more comments](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top)

62

Following @Sam Lu's answer, Indeed Android 4.x blocks certain types from listening to outside touch events, but some types, such as TYPE\_SYSTEM\_ALERT, still do the job.

# Example

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

WindowManager.LayoutParams.MATCH\_PARENT,

WindowManager.LayoutParams.MATCH\_PARENT,

WindowManager.LayoutParams.TYPE\_SYSTEM\_ALERT,

WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE

| WindowManager.LayoutParams.FLAG\_NOT\_TOUCH\_MODAL

| WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH,

PixelFormat.TRANSLUCENT);

WindowManager wm = (WindowManager) getSystemService(WINDOW\_SERVICE);

LayoutInflater inflater = (LayoutInflater) getSystemService(LAYOUT\_INFLATER\_SERVICE);

View myView = inflater.inflate(R.layout.my\_view, null);

myView.setOnTouchListener(new OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

Log.d(TAG, "touch me");

return true;

}

});

// Add layout to window manager

wm.addView(myView, params);

# Permissions

<uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW"/>

[share](https://stackoverflow.com/a/10674462)[improve this answer](https://stackoverflow.com/posts/10674462/edit)

[edited May 6 '15 at 18:35](https://stackoverflow.com/posts/10674462/revisions)

[[https://www.gravatar.com/avatar/08666e6d230bc177a3a13b3a2b8e4b27?s=32&d=identicon&r=PG](https://stackoverflow.com/users/2105657/teo-inke)](https://stackoverflow.com/users/2105657/teo-inke)

[Teo Inke](https://stackoverflow.com/users/2105657/teo-inke)

**4,043**13034

answered May 20 '12 at 14:53

[[https://www.gravatar.com/avatar/f1f1094c46f0fe93cb96d0d88e2a14a4?s=32&d=identicon&r=PG](https://stackoverflow.com/users/738702/amirlazarovich)](https://stackoverflow.com/users/738702/amirlazarovich)

[amirlazarovich](https://stackoverflow.com/users/738702/amirlazarovich)

**846**76

* 5

Its making my screen hanged .. Any idea – [Viswanath Lekshmanan](https://stackoverflow.com/users/1870173/viswanath-lekshmanan) [Apr 5 '13 at 16:38](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment22536622_10674462)

* Great post. Hopefully this isn't disabled in future releases either. – [Cookster](https://stackoverflow.com/users/933237/cookster) [Apr 23 '13 at 16:47](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment23119467_10674462)
* Animations won't work on it... :( And it's unable to catch clicks on any view in it on 4.2.X. Any suggestions? – [Kamran Ahmed](https://stackoverflow.com/users/1468093/kamran-ahmed) [Jun 1 '13 at 5:26](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment24334221_10674462)
* It works but when I'm using that shortcut (ie System Level Alert), Other dialogs are not being shown. Why ? – [Chimu](https://stackoverflow.com/users/709355/chimu) [Aug 19 '13 at 9:58](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment26869404_10674462)
* @Arju what do you mean? are you trying to remove it? – [amirlazarovich](https://stackoverflow.com/users/738702/amirlazarovich) [Oct 1 '13 at 4:41](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment28253051_10674462)

[show **6** more comments](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top)

16

Starting with Android 4.x, Android team fixed a potential security problem by adding a new function adjustWindowParamsLw() in which it will add FLAG\_NOT\_FOCUSABLE, FLAG\_NOT\_TOUCHABLE and remove FLAG\_WATCH\_OUTSIDE\_TOUCH flags for TYPE\_SYSTEM\_OVERLAY windows.

That is, a TYPE\_SYSTEM\_OVERLAY window won't receive any touch event on ICS platform and, of course, to use TYPE\_SYSTEM\_OVERLAY is not a workable solution for ICS and future devices.

[share](https://stackoverflow.com/a/9462190)[improve this answer](https://stackoverflow.com/posts/9462190/edit)

[edited Sep 7 '14 at 4:26](https://stackoverflow.com/posts/9462190/revisions)

[[https://www.gravatar.com/avatar/f00013ceab8fb1928885c5c172fbfd4a?s=32&d=identicon&r=PG](https://stackoverflow.com/users/65863/remy-lebeau)](https://stackoverflow.com/users/65863/remy-lebeau)

[Remy Lebeau](https://stackoverflow.com/users/65863/remy-lebeau)

**339k**19262456

answered Feb 27 '12 at 8:47

[[https://i.stack.imgur.com/9mksC.jpg?s=32&g=1](https://stackoverflow.com/users/1078971/sam-lu)](https://stackoverflow.com/users/1078971/sam-lu)

[Sam Lu](https://stackoverflow.com/users/1078971/sam-lu)

**2,512**1828

* 4

so what are the alternatives? – [radhoo](https://stackoverflow.com/users/486907/radhoo) [Apr 4 '12 at 8:56](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment12794084_9462190)

* 1

See this question about ICS : [stackoverflow.com/questions/9656185/type-system-overlay-in-ics](http://stackoverflow.com/questions/9656185/type-system-overlay-in-ics) – [clemp6r](https://stackoverflow.com/users/307735/clemp6r) [Jun 28 '12 at 7:46](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment14768563_9462190)

add a comment

12

I'm one of the developers of the [Tooleap SDK](http://www.tooleap.com/). We also provide a way for developers to display always on top windows and buttons, and and we have dealt with a similar situation.

One problem the answers here haven't addressed is that of the Android "Secured Buttons".

Secured buttons have the filterTouchesWhenObscured property which means they can't be interacted with, if placed under a window, even if that window does not receive any touches. Quoting the Android documentation:

Specifies whether to filter touches when the view's window is obscured by another visible window. When set to true, the view will not receive touches whenever a toast, dialog or other window appears above the view's window. Refer to the {@link android.view.View} security documentation for more details.

An example of such a button is the **install** button when you try to install third party apks. Any app can display such a button if adding to the view layout the following line:

android:filterTouchesWhenObscured="true"

If you display an always-on-top window over a "Secured Button", so all the secured button parts that are covered by an overlay will not handle any touches, even if that overlay is not clickable. So if you are planing to display such a window, you should provide a way for the user to move it or dismiss it. And if a part of your overlay is transparent, take into account that your user might be confused why is a certain button in the underlying app is not working for him suddenly.

[share](https://stackoverflow.com/a/23591052)[improve this answer](https://stackoverflow.com/posts/23591052/edit)

[edited Dec 27 '15 at 15:23](https://stackoverflow.com/posts/23591052/revisions)

[[https://www.gravatar.com/avatar/3decb53828ce2c77a7fb28f7fd987234?s=32&d=identicon&r=PG](https://stackoverflow.com/users/1748450/chin)](https://stackoverflow.com/users/1748450/chin)

[Chin](https://stackoverflow.com/users/1748450/chin)

**7,059**2371128

answered May 11 '14 at 9:38

[[https://www.gravatar.com/avatar/4c0b52975db48576e02104ced9c37357?s=32&d=identicon&r=PG&f=1](https://stackoverflow.com/users/3615522/danny)](https://stackoverflow.com/users/3615522/danny)

[Danny](https://stackoverflow.com/users/3615522/danny)

**436**512

* Can you please answer my question: [stackoverflow.com/questions/28964771/…](http://stackoverflow.com/questions/28964771/application-level-floating-views-with-navigation-in-android) – [Nauman Afzaal](https://stackoverflow.com/users/3446836/nauman-afzaal) [Mar 11 '15 at 6:41](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment46210322_23591052)

add a comment

11

WORKING ALWAYS ON TOP IMAGE BUTTON

first of all sorry for my english

i edit your codes and make working image button that listens his touch event do not give touch control to his background elements.

also it gives touch listeners to out of other elements

button alingments are bottom and left

you can chage alingments but you need to chages cordinats in touch event in the if element

import android.annotation.SuppressLint;

import android.app.Service;

import android.content.Context;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.PixelFormat;

import android.os.IBinder;

import android.util.Log;

import android.view.Gravity;

import android.view.MotionEvent;

import android.view.View;

import android.view.View.OnTouchListener;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.Toast;

public class HepUstte extends Service {

HUDView mView;

@Override

public void onCreate() {

super.onCreate();

Toast.makeText(getBaseContext(),"onCreate", Toast.LENGTH\_LONG).show();

final Bitmap kangoo = BitmapFactory.decodeResource(getResources(),

R.drawable.logo\_l);

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

kangoo.getWidth(),

kangoo.getHeight(),

WindowManager.LayoutParams.TYPE\_SYSTEM\_ALERT,

WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE

|WindowManager.LayoutParams.FLAG\_NOT\_TOUCH\_MODAL

|WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH,

PixelFormat.TRANSLUCENT);

params.gravity = Gravity.LEFT | Gravity.BOTTOM;

params.setTitle("Load Average");

WindowManager wm = (WindowManager) getSystemService(WINDOW\_SERVICE);

mView = new HUDView(this,kangoo);

mView.setOnTouchListener(new OnTouchListener() {

@Override

public boolean onTouch(View arg0, MotionEvent arg1) {

// TODO Auto-generated method stub

//Log.e("kordinatlar", arg1.getX()+":"+arg1.getY()+":"+display.getHeight()+":"+kangoo.getHeight());

if(arg1.getX()<kangoo.getWidth() & arg1.getY()>0)

{

Log.d("tıklandı", "touch me");

}

return false;

}

});

wm.addView(mView, params);

}

@Override

public IBinder onBind(Intent arg0) {

// TODO Auto-generated method stub

return null;

}

}

@SuppressLint("DrawAllocation")

class HUDView extends ViewGroup {

Bitmap kangoo;

public HUDView(Context context,Bitmap kangoo) {

super(context);

this.kangoo=kangoo;

}

protected void onDraw(Canvas canvas) {

//super.onDraw(canvas);

// delete below line if you want transparent back color, but to understand the sizes use back color

canvas.drawColor(Color.BLACK);

canvas.drawBitmap(kangoo,0 , 0, null);

//canvas.drawText("Hello World", 5, 15, mLoadPaint);

}

protected void onLayout(boolean arg0, int arg1, int arg2, int arg3, int arg4) {

}

public boolean onTouchEvent(MotionEvent event) {

//return super.onTouchEvent(event);

// Toast.makeText(getContext(),"onTouchEvent", Toast.LENGTH\_LONG).show();

return true;

}

}

[share](https://stackoverflow.com/a/16473524)[improve this answer](https://stackoverflow.com/posts/16473524/edit)

answered May 10 '13 at 0:52

[[https://i.stack.imgur.com/nRhvl.jpg?s=32&g=1](https://stackoverflow.com/users/2002345/onur-%c3%96zt%c3%bcrk)](https://stackoverflow.com/users/2002345/onur-%c3%96zt%c3%bcrk)

[Onur Öztürk](https://stackoverflow.com/users/2002345/onur-%c3%96zt%c3%bcrk)

**152**211

* This works, but how can I make sure my image was clicked and not something around the image? – [Liam W](https://stackoverflow.com/users/1124470/liam-w) [Jul 29 '13 at 18:24](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment26199897_16473524)
* @LiamW, the coordinates of the touch are compared to kangaroo.getWidth() in this example. You can deduce what to compare it to in your own code based on where you're drawing your image. – [Genia S.](https://stackoverflow.com/users/59947/genia-s) [Oct 27 '13 at 21:51](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment29133074_16473524)
* Very nice! Works perfectly. – [Genia S.](https://stackoverflow.com/users/59947/genia-s) [Oct 27 '13 at 21:52](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment29133082_16473524)
* What is the minimum android version this works on? Im working on writing this up but im new to android so am having beginners time. – [Noitidart](https://stackoverflow.com/users/1828637/noitidart) [Aug 10 '15 at 0:06](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment51733849_16473524)

add a comment

6

Actually, you can try WindowManager.LayoutParams.TYPE\_SYSTEM\_ERROR instead of TYPE\_SYSTEM\_OVERLAY. It may sound like a hack, but it let you display view on top of everything and still get touch events.

[share](https://stackoverflow.com/a/33647326)[improve this answer](https://stackoverflow.com/posts/33647326/edit)

answered Nov 11 '15 at 9:12

[[https://i.stack.imgur.com/c90tj.jpg?s=32&g=1](https://stackoverflow.com/users/1835438/ov3r1oad)](https://stackoverflow.com/users/1835438/ov3r1oad)

[Ov3r1oad](https://stackoverflow.com/users/1835438/ov3r1oad)

**523**1421

* 1

I tried every thing to make it working, but no success....but adding TYPE\_SYSTEM\_ERROR done the job....Actually TYPE\_SYSTEM\_OVERLAY was blocking the touch.....+1. – [Robi Kumar Tomar](https://stackoverflow.com/users/2531150/robi-kumar-tomar) [Mar 24 '18 at 13:35](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment85933373_33647326)

add a comment

5

Try this. Works fine in ICS. If You want to stop service simply click the notification generated in statusbar.

public class HUD extends Service

{

protected boolean foreground = false;

protected boolean cancelNotification = false;

private Notification notification;

private View myView;

protected int id = 0;

private WindowManager wm;

private WindowManager.LayoutParams params;

@Override

public IBinder onBind(Intent intent) {

return null;

}

@Override

public void onCreate() {

super.onCreate();

// System.exit(0);

Toast.makeText(getBaseContext(),"onCreate", Toast.LENGTH\_SHORT).show();

params = new WindowManager.LayoutParams(WindowManager.LayoutParams.WRAP\_CONTENT, WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.TYPE\_PHONE, WindowManager.LayoutParams.FLAG\_NOT\_TOUCH\_MODAL

| WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH | WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE, PixelFormat.TRANSLUCENT);

params.gravity=Gravity.TOP|Gravity.LEFT;

wm = (WindowManager) getSystemService(WINDOW\_SERVICE);

inflateview();

foregroundNotification(1);

//moveToForeground(1,n,true);

}

@Override

public void onDestroy() {

super.onDestroy();

((NotificationManager) getSystemService(NOTIFICATION\_SERVICE)).cancel(0);

Toast.makeText(getBaseContext(),"onDestroy", Toast.LENGTH\_SHORT).show();

if(myView != null)

{

((WindowManager) getSystemService(WINDOW\_SERVICE)).removeView(myView);

myView = null;

}

}

protected Notification foregroundNotification(int notificationId)

{

notification = new Notification(R.drawable.ic\_launcher, "my Notification", System.currentTimeMillis());

notification.flags = notification.flags | Notification.FLAG\_ONGOING\_EVENT | Notification.FLAG\_ONLY\_ALERT\_ONCE;

notification.setLatestEventInfo(this, "my Notification", "my Notification", notificationIntent());

((NotificationManager) getSystemService(NOTIFICATION\_SERVICE)).notify(id, notification);

return notification;

}

private PendingIntent notificationIntent() {

Intent intent = new Intent(this, stopservice.class);

PendingIntent pending = PendingIntent.getActivity(this, 0, intent, PendingIntent.FLAG\_UPDATE\_CURRENT);

return pending;

}

public void inflateview()

{

LayoutInflater inflater = (LayoutInflater) getSystemService(LAYOUT\_INFLATER\_SERVICE);

myView = inflater.inflate(R.layout.activity\_button, null);

myView.setOnTouchListener(new OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

Toast.makeText(getBaseContext(),"onToasttt", Toast.LENGTH\_SHORT).show();

return false;

}

});

// Add layout to window manager

wm.addView(myView, params);

}

}

**UPDATE**

Sample [here](http://www.jawsware.mobi/code_OverlayView/)

To create an overlay view, when setting up the LayoutParams DON'T set the type to TYPE\_SYSTEM\_OVERLAY.

Instead set it to TYPE\_PHONE.

Use the following flags:

FLAG\_NOT\_TOUCH\_MODAL

FLAG\_WATCH\_OUTSIDE\_TOUCH

[share](https://stackoverflow.com/a/15841373)[improve this answer](https://stackoverflow.com/posts/15841373/edit)

[edited Dec 13 '13 at 5:13](https://stackoverflow.com/posts/15841373/revisions)

answered Apr 5 '13 at 18:52

[[https://i.stack.imgur.com/hcpwS.jpg?s=32&g=1](https://stackoverflow.com/users/1870173/viswanath-lekshmanan)](https://stackoverflow.com/users/1870173/viswanath-lekshmanan)

[Viswanath Lekshmanan](https://stackoverflow.com/users/1870173/viswanath-lekshmanan)

**7,384**12953

* This does not work for me, My screen does not interact with the background. Any other method you know... – [Vinuthan](https://stackoverflow.com/users/1767451/vinuthan) [Jul 16 '13 at 13:51](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment25753149_15841373)
* @Vinuthan Whats your output , does it respond – [Viswanath Lekshmanan](https://stackoverflow.com/users/1870173/viswanath-lekshmanan) [Jul 18 '13 at 16:35](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment25843753_15841373)
* No, it does not respond. I have padded my activity above to half the screen so that i can interact with the background, but i cannot interact with the background. – [Vinuthan](https://stackoverflow.com/users/1767451/vinuthan) [Jul 19 '13 at 11:18](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment25870859_15841373)
* FLAG\_WATCH\_OUTSIDE\_TOUCH prevents you from interacting with the background – [Richard Le Mesurier](https://stackoverflow.com/users/383414/richard-le-mesurier)[Jan 15 '14 at 10:37](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment31805508_15841373)
* 1

TYPE\_PHONE worked for me - now touches are delivered - at least I'm able to tap a button ;) thanks! – [vir us](https://stackoverflow.com/users/1713920/vir-us)[Dec 23 '14 at 17:50](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment43671414_15841373)

[show **1** more comment](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top)

2

It uses permission "android.permission.SYSTEM\_ALERT\_WINDOW" full tutorial on this link : <http://androidsrc.net/facebook-chat-like-floating-chat-heads/>

[share](https://stackoverflow.com/a/28335967)[improve this answer](https://stackoverflow.com/posts/28335967/edit)

answered Feb 5 '15 at 3:58

[[https://i.stack.imgur.com/PIuLZ.jpg?s=32&g=1](https://stackoverflow.com/users/976362/aman-yadav)](https://stackoverflow.com/users/976362/aman-yadav)

[AmAn yAdAv](https://stackoverflow.com/users/976362/aman-yadav)

**21**3

add a comment

2

Here is some simple solution, All you need is to inflate XML layout just like you do on list adapters, just make XML layout to inflate it. Here is code that all you need.

public class HUD extends Service {

View mView;

LayoutInflater inflate;

TextView t;

Button b;

@Override

public void onCreate() {

super.onCreate();

Toast.makeText(getBaseContext(),"onCreate", Toast.LENGTH\_LONG).show();

WindowManager wm = (WindowManager) getSystemService(WINDOW\_SERVICE);

Display display = wm.getDefaultDisplay(); get phone display size

int width = display.getWidth(); // deprecated - get phone display width

int height = display.getHeight(); // deprecated - get phone display height

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

width,

height,

WindowManager.LayoutParams.TYPE\_SYSTEM\_ALERT,

WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE

|WindowManager.LayoutParams.FLAG\_NOT\_TOUCH\_MODAL

|WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH,

PixelFormat.TRANSLUCENT);

params.gravity = Gravity.LEFT | Gravity.CENTER;

params.setTitle("Load Average");

inflate = (LayoutInflater) getBaseContext()

.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

mView = inflate.inflate(R.layout.canvas, null);

b = (Button) mView.findViewById(R.id.button1);

t = (TextView) mView.findViewById(R.id.textView1);

b.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

// TODO Auto-generated method stub

t.setText("yes you click me ");

}

});

wm.addView(mView, params);

}

@Override

public IBinder onBind(Intent arg0) {

// TODO Auto-generated method stub

return null;

}

}

[share](https://stackoverflow.com/a/20025996)[improve this answer](https://stackoverflow.com/posts/20025996/edit)

[edited Dec 1 '16 at 11:36](https://stackoverflow.com/posts/20025996/revisions)

[[https://i.stack.imgur.com/EvGAD.jpg?s=32&g=1](https://stackoverflow.com/users/4399126/kinjal)](https://stackoverflow.com/users/4399126/kinjal)

[Kinjal](https://stackoverflow.com/users/4399126/kinjal)

**750**3923

answered Nov 17 '13 at 1:11

[[https://www.gravatar.com/avatar/578e93edc32ff18e0630bdbf6207ca58?s=32&d=identicon&r=PG](https://stackoverflow.com/users/1668168/user1668168)](https://stackoverflow.com/users/1668168/user1668168)

[user1668168](https://stackoverflow.com/users/1668168/user1668168)

**41**16

add a comment

2

TYPE\_SYSTEM\_OVERLAY This constant was deprecated in since API level 26. Use TYPE\_APPLICATION\_OVERLAY instead.

[share](https://stackoverflow.com/a/54545761)[improve this answer](https://stackoverflow.com/posts/54545761/edit)

answered Feb 6 at 2:17

[[https://www.gravatar.com/avatar/59385acef3f9f6c6ef78556e9159531d?s=32&d=identicon&r=PG&f=1](https://stackoverflow.com/users/9146636/adrian-williamson)](https://stackoverflow.com/users/9146636/adrian-williamson)

[Adrian Williamson](https://stackoverflow.com/users/9146636/adrian-williamson)

**21**1

add a comment

1

Well try my code, atleast it gives you a string as overlay, you can very well replace it with a button or an image. You wont believe this is my first ever android app LOL. Anyways if you are more experienced with android apps than me, please try

* changing parameters 2 and 3 in "new WindowManager.LayoutParams"
* try some different event approach

[share](https://stackoverflow.com/a/4482414)[improve this answer](https://stackoverflow.com/posts/4482414/edit)

* i eventually didnt find time to pursue this, hence i'm still scarce on its details... – [st0le](https://stackoverflow.com/users/216517/st0le) [Dec 19 '10 at 11:06](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment4901971_4482414)

add a comment

1

If anyone still reading this thread and not able to get this working, I'm very sorry to tell you this way to intercept motion event is considered as bug and fix in android >=4.2.

The motion event you intercepted, although has action as ACTION\_OUTSIDE, return 0 in getX and getY. This means you can not see all the motion position on screen, nor can you do anything. I know the doc said It will get x and y, but the truth is it WILL NOT. It seems that this is to block key logger.

If anyone do have a workaround, please leave your comment.

ref: [Why does ACTION\_OUTSIDE return 0 everytime on KitKat 4.4.2?](https://stackoverflow.com/questions/22041604/why-does-action-outside-return-0-everytime-on-kitkat-4-4-2/24502062#24502062)

<https://code.google.com/p/android/issues/detail?id=72746>

[share](https://stackoverflow.com/a/26271430)[improve this answer](https://stackoverflow.com/posts/26271430/edit)

[edited May 23 '17 at 12:26](https://stackoverflow.com/posts/26271430/revisions)

add a comment

1

by using service you can achieve this :

public class PopupService extends Service{

private static final String TAG = PopupService.class.getSimpleName();

WindowManager mWindowManager;

View mView;

String type ;

@Override

public IBinder onBind(Intent intent) {

return null;

}

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

// registerOverlayReceiver();

type = intent.getStringExtra("type");

Utils.printLog("type = "+type);

showDialog(intent.getStringExtra("msg"));

return super.onStartCommand(intent, flags, startId);

}

private void showDialog(String aTitle)

{

if(type.equals("when screen is off") | type.equals("always"))

{

Utils.printLog("type = "+type);

PowerManager pm = (PowerManager) getApplicationContext().getSystemService(Context.POWER\_SERVICE);

WakeLock mWakeLock = pm.newWakeLock((PowerManager.SCREEN\_DIM\_WAKE\_LOCK | PowerManager.ACQUIRE\_CAUSES\_WAKEUP), "YourServie");

mWakeLock.acquire();

mWakeLock.release();

}

mWindowManager = (WindowManager) getSystemService(WINDOW\_SERVICE);

mView = View.inflate(getApplicationContext(), R.layout.dialog\_popup\_notification\_received, null);

mView.setTag(TAG);

int top = getApplicationContext().getResources().getDisplayMetrics().heightPixels / 2;

LinearLayout dialog = (LinearLayout) mView.findViewById(R.id.pop\_exit);

// android.widget.LinearLayout.LayoutParams lp = (android.widget.LinearLayout.LayoutParams) dialog.getLayoutParams();

// lp.topMargin = top;

// lp.bottomMargin = top;

// mView.setLayoutParams(lp);

final EditText etMassage = (EditText) mView.findViewById(R.id.editTextInPopupMessageReceived);

ImageButton imageButtonSend = (ImageButton) mView.findViewById(R.id.imageButtonSendInPopupMessageReceived);

// lp = (LayoutParams) imageButton.getLayoutParams();

// lp.topMargin = top - 58;

// imageButton.setLayoutParams(lp);

imageButtonSend.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

Utils.printLog("clicked");

// mView.setVisibility(View.INVISIBLE);

if(!etMassage.getText().toString().equals(""))

{

Utils.printLog("sent");

etMassage.setText("");

}

}

});

TextView close = (TextView) mView.findViewById(R.id.TextViewCloseInPopupMessageReceived);

close.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

hideDialog();

}

});

TextView view = (TextView) mView.findViewById(R.id.textviewViewInPopupMessageReceived);

view.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

hideDialog();

}

});

TextView message = (TextView) mView.findViewById(R.id.TextViewMessageInPopupMessageReceived);

message.setText(aTitle);

final WindowManager.LayoutParams mLayoutParams = new WindowManager.LayoutParams(

ViewGroup.LayoutParams.MATCH\_PARENT,

ViewGroup.LayoutParams.MATCH\_PARENT, 0, 0,

WindowManager.LayoutParams.TYPE\_SYSTEM\_ERROR,

WindowManager.LayoutParams.FLAG\_SHOW\_WHEN\_LOCKED

| WindowManager.LayoutParams.FLAG\_DISMISS\_KEYGUARD

// | WindowManager.LayoutParams.FLAG\_TURN\_SCREEN\_ON

| WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON ,

PixelFormat.RGBA\_8888);

mView.setVisibility(View.VISIBLE);

mWindowManager.addView(mView, mLayoutParams);

mWindowManager.updateViewLayout(mView, mLayoutParams);

}

private void hideDialog(){

if(mView != null && mWindowManager != null){

mWindowManager.removeView(mView);

mView = null;

}

}

}

[share](https://stackoverflow.com/a/28761131)[improve this answer](https://stackoverflow.com/posts/28761131/edit)

@Sarwar Erfan's answer does not work any longer as Android does not allow adding view with WindowManager.LayoutParams.TYPE\_SYSTEM\_OVERLAY to window to be touchable anymore not with even WindowManager.LayoutParams.FLAG\_WATCH\_OUTSIDE\_TOUCH.

I found solution to this problem. You can check it out in following question

[When adding view to window with WindowManager.LayoutParams.TYPE\_SYSTEM\_OVERLAY, it is not getting touch event](https://stackoverflow.com/questions/37138546/when-adding-view-to-window-with-windowmanager-layoutparams-type-system-overlay/37348311#37348311)

**249**211

* Your referred solution also doesn't work in Marshmallow – [Ahmad Shahwaiz](https://stackoverflow.com/users/1696950/ahmad-shahwaiz) [Nov 3 '16 at 14:55](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment68059516_37348503)
* @AhmadShahwaiz Please check link again I have added comment for problem you mentioned there – [patilmandar2007](https://stackoverflow.com/users/5814139/patilmandar2007) [Nov 4 '16 at 14:04](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment68097471_37348503)
* Yes, now its working. But not getting the callback response after calling startActivityForResult(intent, OVERLAY\_REQ\_CODE); in protected void onActivityResult(int requestCode, int resultCode, Intent data) method. – [Ahmad Shahwaiz](https://stackoverflow.com/users/1696950/ahmad-shahwaiz) [Nov 7 '16 at 12:55](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment68175657_37348503)
* @AhmadShahwaiz Changes mentioned in link are to be called using activity where we get callback in onActivityResult method. If you are checking above permission in Service then you might need to add a dummy transparent activity to be started from service, which can ask user for Permission and then in onActivityResult dummy activity you will get result from user. After you will require to notify your service to add view to window or not depending on user response – [patilmandar2007](https://stackoverflow.com/users/5814139/patilmandar2007) [Nov 8 '16 at 11:20](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment68214697_37348503)
* Doesn't seem to work for me, Android 5.0 – [Denny](https://stackoverflow.com/users/2890156/denny) [Jan 29 '17 at 18:47](https://stackoverflow.com/questions/4481226/creating-a-system-overlay-window-always-on-top#comment71029653_37348503)

add a comment

0

Found a library that does just that: <https://github.com/recruit-lifestyle/FloatingView>

There's a sample project in the root folder. I ran it and it works as required. The background is clickable - even if it's another app.

# [When adding view to window with WindowManager.LayoutParams.TYPE\_SYSTEM\_OVERLAY, it is not getting touch event](https://stackoverflow.com/questions/37138546/when-adding-view-to-window-with-windowmanager-layoutparams-type-system-overlay)

3

2

I need to display my view on top of dialer application, so I was using TYPE\_PHONE for this purpose which is touchable, but still on some devices like Nexus 5 dialer application of device is hiding it.

Tried using TYPE\_SYSTEM\_OVERLAY makes view visible on top of dialer application, but touch is not there. Any help regarding this will be great.

10

After lot of searching for above problem, I found solution my self. Here it is how I made view to be on top of everything inside device and also making it touchable which was not possible with TYPE\_SYSTEM\_OVERLAY.

WindowManager.LayoutParams params = new WindowManager.LayoutParams(

WindowManager.LayoutParams.MATCH\_PARENT,

WindowManager.LayoutParams.WRAP\_CONTENT,

WindowManager.LayoutParams.TYPE\_SYSTEM\_ERROR,

WindowManager.LayoutParams.FLAG\_NOT\_FOCUSABLE |

WindowManager.LayoutParams.FLAG\_LAYOUT\_IN\_SCREEN,

PixelFormat.TRANSPARENT);

params.gravity = Gravity.TOP | Gravity.CENTER\_HORIZONTAL;

WindowManager wm = (WindowManager) getSystemService(WINDOW\_SERVICE);

wm.addView(view, params);

# Android窗口管理分析（3）：窗口分组及Z-order的确定总结

[看书的小蜗牛](https://cloud.tencent.com/developer/user/2173239)发表于[Android源码框架分析](https://cloud.tencent.com/developer/column/3759)订阅

203

在Android系统中，窗口是有分组概念的，例如，Activity中弹出的所有PopupWindow会随着Activity的隐藏而隐藏，可以说这些都附属于Actvity的子窗口分组，对于Dialog也同样如此，只不过Dialog与Activity属于同一个分组。之间已经简单介绍了窗口类型划分：应用窗口、子窗口、系统窗口，Activity与Dialog都属于应用窗口，而PopupWindow属于子窗口，Toast、输入法等属于系统窗口。只有应用窗口与系统窗口可以作为父窗口，子窗口不能作为子窗口的父窗口，也就说Activity与Dialog或者系统窗口中可以弹出PopupWindow，但是PopupWindow不能在自己内部弹出PopupWindow子窗口。日常开发中，一些常见的问题都同窗口的分组有关系，比如为什么新建Dialog的时候必须要用Activity的Context，而不能用Application的；为什么不能以PopupWindow的View为锚点弹出子PopupWindow？其实这里面就牵扯都Android的窗口组织管理形式，本文主要包含以下几点内容：

* 窗口的分组管理 ：应用窗口组、子窗口组、系统窗口组
* Activity、Dialg应用窗口及PopWindow子窗口的添加原理跟注意事项
* 窗口的Z次序管理：窗口的分配序号、次序调整等
* WMS中窗口次序分配如何影响SurfaceFlinger服务

在[WMS窗口添加一文](https://www.jianshu.com/p/e4b19fc36a0e)中分析过,窗口的添加是通过WindowManagerGlobal.addView()来完成 函数原型如下

public void addView(View view, ViewGroup.LayoutParams params,

Display display, Window parentWindow)

前三个参数是必不可少的，view、params、display，其中display表示要输出的显示设备，先不考虑。view 就是APP要添加到WindowManagerGlobal管理的View，而 params是WindowManager.LayoutParams，主要用来描述窗口属性，WindowManager.LayoutParams有两个很重要的参数type与token，

public static class LayoutParams extends ViewGroup.LayoutParams

implements Parcelable {

...

public int type;

...

public IBinder token = null;

}

type用来描述窗口的类型，而token其实是标志窗口的分组，token相同的窗口属于同一分组，后面会知道这个token其实是WMS在APP端对应的一个WindowToken的键值。这里先看一下type参数，之前曾添加过Toast窗口，它的type值是TYPE\_TOAST，标识是一个系统提示窗口，下面先简单看下三种窗口类型的Type对应的值，首先看一下应用窗口

| 窗口TYPE值 | 窗口类型 |
| --- | --- |
| FIRST\_APPLICATION\_WINDOW = 1 | 开始应用程序窗口 |
| TYPE\_BASE\_APPLICATION=1 | 所有程序窗口的base窗口，其他应用程序窗口都显示在它上面 |
| TYPE\_APPLICATION =2 | 普通应用程序窗口，token必须设置为Activity的token |
| TYPE\_APPLICATION\_STARTING =3 | 应用程序启动时所显示的窗口 |
| LAST\_APPLICATION\_WINDOW = 99 | 结束应用程序窗口 |

一般Activity都是TYPE\_BASE\_APPLICATION类型的，而TYPE\_APPLICATION主要是用于Dialog，再看下子窗口类型

| 窗口TYPE值 | 窗口类型 |
| --- | --- |
| FIRST\_SUB\_WINDOW = 1000 | SubWindows子窗口，子窗口的Z序和坐标空间都依赖于他们的宿主窗口 |
| TYPE\_APPLICATION\_PANEL =1000 | 面板窗口，显示于宿主窗口的上层 |
| TYPE\_APPLICATION\_MEDIA =1001 | 媒体窗口（例如视频），显示于宿主窗口下层 |
| TYPE\_APPLICATION\_SUB\_PANEL =1002 | 应用程序窗口的子面板，显示于所有面板窗口的上层 |
| TYPE\_APPLICATION\_ATTACHED\_DIALOG = 1003 | 对话框，类似于面板窗口，绘制类似于顶层窗口，而不是宿主的子窗口 |
| TYPE\_APPLICATION\_MEDIA\_OVERLAY =1004 | 媒体信息，显示在媒体层和程序窗口之间，需要实现半透明效果 |
| LAST\_SUB\_WINDOW=1999 | 结束子窗口 |

最后看几个系统窗口类型，

| 窗口TYPE值 | 窗口类型 |
| --- | --- |
| FIRST\_SYSTEM\_WINDOW = 2000 | 系统窗口 |
| TYPE\_STATUS\_BAR = FIRST\_SYSTEM\_WINDOW | 状态栏 |
| TYPE\_SYSTEM\_ALERT = FIRST\_SYSTEM\_WINDOW+3 | 系统提示，出现在应用程序窗口之上 |
| TYPE\_TOAST = FIRST\_SYSTEM\_WINDOW+5 | 显示Toast |

了解窗口类型后，我们需要面对的首要问题是：**窗口如何根据类型进行分组归类的？Dialog是如何确定附属Activity，PopupWindow如何确定附属父窗口？**。

## 窗口的分组原理

如果用一句话概括窗口分组的话：**Android窗口是以token来进行分组的**，同一组窗口握着相同的token，什么是token呢？在 Android WMS管理框架中，token一个IBinder对象，IBinder在实体端与代理端会相互转换，这里只看实体端，它的取值只有两种:ViewRootImpl中ViewRootImpl.W，或者是ActivityRecord中的IApplicationToken.Stub对象，其中ViewRootImpl.W的实体对象在ViewRootImpl中实例化，而IApplicationToken.Stub在ActivityManagerService端实例化，之后被AMS添加到WMS服务中去，作为Activity应用窗口的键值标识。之前说过Activity跟Dialog属于同一分组，现在就来看一下Activity跟Dialog的token是如何复用的，这里的复用分为APP端及WMS服务端，关于窗口的添加流程之前已经分析过，这里只跟随窗口token来分析窗口的分组，我们知道在WMS端，WindowState与窗口的一一对应，而WindowToken与窗口分组，这可以从两者的定义看出如下：

class WindowToken {

final WindowManagerService service;

final IBinder token;

final int windowType;

final boolean explicit;

<!--当前窗口对应appWindowToken，是不是同Activity存在依附关系-->

AppWindowToken appWindowToken;

<!--关键点1 当前WindowToken对应的窗口列表-->

final WindowList windows = new WindowList();

...

}

final class WindowState implements WindowManagerPolicy.WindowState {

static final String TAG = "WindowState";

final WindowManagerService mService;

final WindowManagerPolicy mPolicy;

final Context mContext;

final Session mSession;

<!--当前WindowState对应IWindow窗口代理-->

final IWindow mClient;

<!--当前WindowState对应的父窗口-->

final WindowState mAttachedWindow;

...

<!--当前WindowState隶属的token-->

WindowToken mToken;

WindowToken mRootToken;

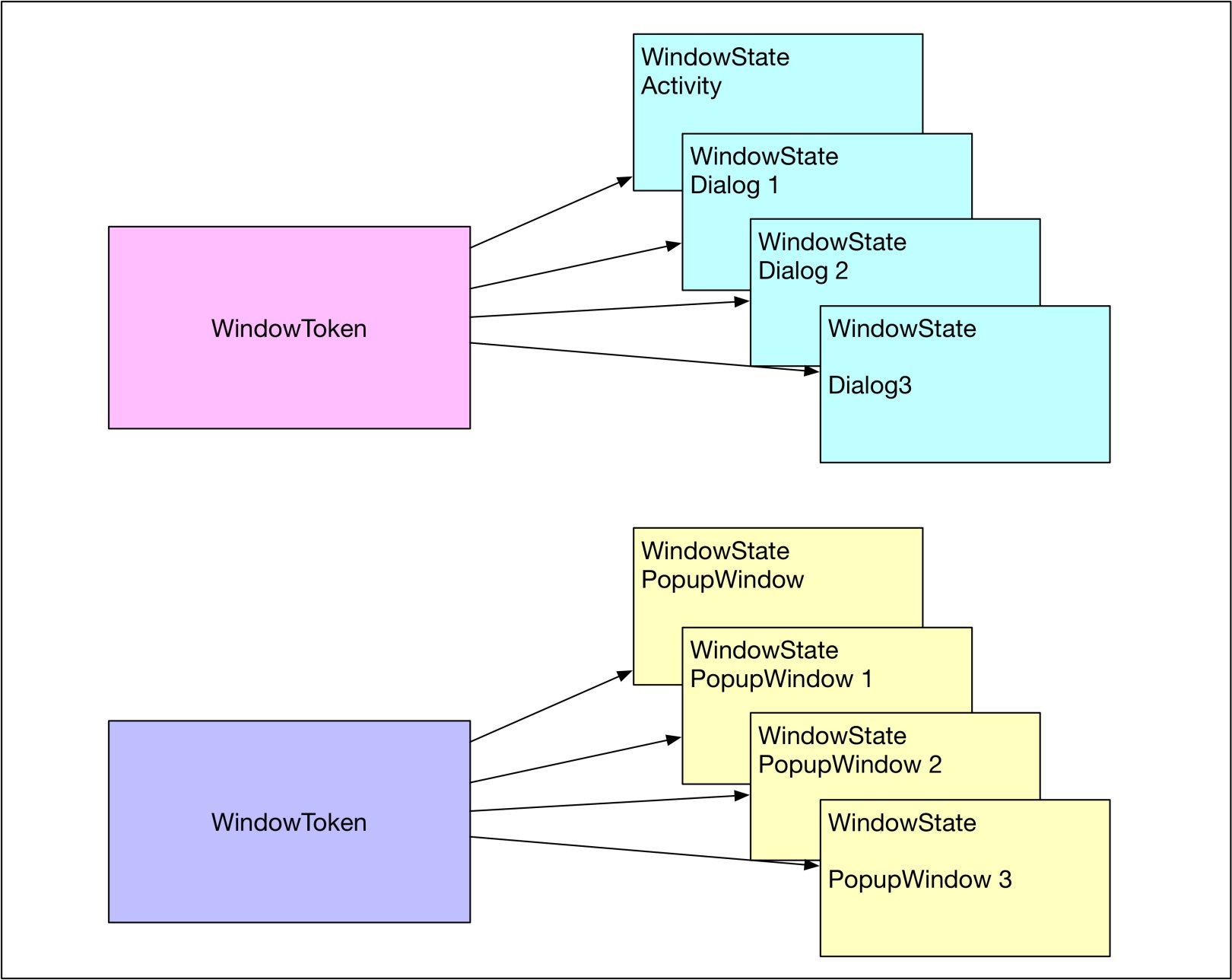
AppWindowToken mAppToken;

AppWindowToken mTargetAppToken;

...

}

可以看到WindowToken包含一个 WindowList windows = new WindowList()，其实就是WindowState列表；而WindowState有一个WindowToken mToken，也就是WindowToken包含一个WindowState列表，而每个WindowState附属一个WindowToken窗口组，示意图如下：



WindowToken与WindowState关系.jpg

### Activity对应token及WindowToken（AppWindowToken）的添加

AMS在为Activity创建ActivityRecord的时候，会新建IApplicationToken.Stub appToken对象，在startActivity之前会首先向WMS服务登记当前Activity的Token，随后，通过Binder通信将IApplicationToken传递给APP端，在通知ActivityThread新建Activity对象之后，利用Activity的attach方法添加到Activity中，先看第一步AMS将Activity的token加入到WMS中，并且为Activity创建APPWindowToken。

<!--AMS ActivityStack.java中代码 -->

final void startActivityLocked(ActivityRecord r, boolean newTask,

boolean doResume, boolean keepCurTransition, Bundle options) {

...<!--关键点1 添加Activity token到WMS-->

mWindowManager.addAppToken(task.mActivities.indexOf(r), r.appToken,XXX);

}

@Override

public void addAppToken(int addPos, IApplicationToken token, int taskId, int stackId,

int requestedOrientation, boolean fullscreen, boolean showForAllUsers, int userId,

int configChanges, boolean voiceInteraction, boolean launchTaskBehind) {

synchronized(mWindowMap) {

<!--新建AppWindowToken-->

AppWindowToken atoken = findAppWindowToken(token.asBinder());

atoken = new AppWindowToken(this, token, voiceInteraction);

...

<!--将AppWindowToken以IApplicationToken.Stub为键值放如WMS的mTokenMap中-->

mTokenMap.put(token.asBinder(), atoken);

<!--开始肯定是隐藏状态，因为还没有resume-->

atoken.hidden = true;

atoken.hiddenRequested = true;

}

}

也就是说Activity分组的Token其实是早在Activity显示之前就被AMS添加到WMS中去的，之后AMS才会通知App端去新建Activity，并将Activity的Window添加到WMS中去，接着看下APP端的流程：

private Activity performLaunchActivity(ActivityClientRecord r, Intent customIntent) {

<!--关键点1 新建Activity-->

Activity activity = null;

try {

java.lang.ClassLoader cl = r.packageInfo.getClassLoader();

activity = mInstrumentation.newActivity(

cl, component.getClassName(), r.intent);

...

try {

Application app = r.packageInfo.makeApplication(false, mInstrumentation);

if (activity != null) {

<!--关键点2 新建appContext-->

Context appContext = createBaseContextForActivity(r, activity);

CharSequence title = r.activityInfo.loadLabel(appContext.getPackageManager());

Configuration config = new Configuration(mCompatConfiguration);

<!--关键点3 attach到WMS-->

activity.attach(appContext, this, getInstrumentation(), r.token，XXX);

...

}

关键点1，新建一个Activity，之后会为Activiyt创建一个appContext，这个Context主要是为了activity.attach使用的，其实就是单纯new一个ContextImpl，之后Activity会利用attach函数将ContextImpl绑定到自己身上。

static ContextImpl createActivityContext(ActivityThread mainThread,

LoadedApk packageInfo, int displayId, Configuration overrideConfiguration) {

return new ContextImpl(null, mainThread, packageInfo, null, null, false,

null, overrideConfiguration, displayId);

}

final void attach(Context context, ActivityThread aThread,

Instrumentation instr, IBinder token, int ident,

Application application, Intent intent, ActivityInfo info,

CharSequence title, Activity parent, String id,

NonConfigurationInstances lastNonConfigurationInstances,

Configuration config, String referrer, IVoiceInteractor voiceInteractor) {

<!--关键点1 为Activity绑定ContextImpl 因为Activity只是一个ContextWraper-->

attachBaseContext(context);

mFragments.attachHost(null /\*parent\*/);

<!--关键点2 new一个PhoneWindow 并设置回调-->

mWindow = new PhoneWindow(this);

mWindow.setCallback(this);

mWindow.setOnWindowDismissedCallback(this);

mWindow.getLayoutInflater().setPrivateFactory(this);

...

<!--关键点3 Token的传递-->

mToken = token;

mIdent = ident;

mApplication = application;

...

mWindow.setWindowManager(

(WindowManager)context.getSystemService(Context.WINDOW\_SERVICE),

mToken, mComponent.flattenToString(),

(info.flags & ActivityInfo.FLAG\_HARDWARE\_ACCELERATED) != 0);

<!--将Window的WindowManager赋值给Activity-->

mWindowManager = mWindow.getWindowManager();

mCurrentConfig = config;

}

mWindow.setWindowManager并不是直接为Window设置WindowManagerImpl，而是利用当前的WindowManagerImpl重新为Window创建了一个WindowManagerImpl，并将自己设置此WindowManagerImpl的parentWindow：

public void setWindowManager(WindowManager wm, IBinder appToken, String appName,

boolean hardwareAccelerated) {

mAppToken = appToken;

mAppName = appName;

mHardwareAccelerated = hardwareAccelerated

|| SystemProperties.getBoolean(PROPERTY\_HARDWARE\_UI, false);

if (wm == null) {

wm = (WindowManager)mContext.getSystemService(Context.WINDOW\_SERVICE);

}

mWindowManager = ((WindowManagerImpl)wm).createLocalWindowManager(this);

}

public WindowManagerImpl createLocalWindowManager(Window parentWindow) {

return new WindowManagerImpl(mDisplay, parentWindow);

}

之后将Window的WindowManagerImpl传递给Activity，作为Activity的WindowManager将来Activity通过getSystemService获取WindowManager服务的时候，其实是直接返回了Window的WindowManagerImpl，

@Override

public Object getSystemService(String name) {

if (WINDOW\_SERVICE.equals(name)) {

return mWindowManager;

} else if (SEARCH\_SERVICE.equals(name)) {

ensureSearchManager();

return mSearchManager;

}

return super.getSystemService(name);

}

之后看一下关键点3，这里传递的token其实就是AMS端传递过来的IApplicationToken代理，一个IBinder对象。之后利用ContextImpl的getSystemService()函数得到一个一个WindowManagerImpl对象，再通过setWindowManager为Activity创建自己的WindowManagerImpl。到这一步，Activity已经准备完毕，剩下的就是在resume中通过addView将窗口添加到到WMS，具体实现在ActivityThread的handleResumeActivity函数中：

final void handleResumeActivity(IBinder token,

boolean clearHide, boolean isForward, boolean reallyResume) {

ActivityClientRecord r = performResumeActivity(token, clearHide);

if (r != null) {

final Activity a = r.activity;

...

if (r.window == null && !a.mFinished && willBeVisible) {

<!--关键点1-->

r.window = r.activity.getWindow();

View decor = r.window.getDecorView();

decor.setVisibility(View.INVISIBLE);

<!--关键点2 获取WindowManager-->

ViewManager wm = a.getWindowManager();

WindowManager.LayoutParams l = r.window.getAttributes();

a.mDecor = decor;

l.type = WindowManager.LayoutParams.TYPE\_BASE\_APPLICATION;

l.softInputMode |= forwardBit;

if (a.mVisibleFromClient) {

a.mWindowAdded = true;

<!--关键点3 添加到WMS管理-->

wm.addView(decor, l);

}

...

}

关键点1是为了获取Activit的Window及DecorView对象，如果用户没有通过setContentView方式新建DecorView，这里会利用PhoneWindow的getDecorView()新建DecorView，

@Override

public final View getDecorView() {

if (mDecor == null) {

installDecor();

}

return mDecor;

}

之后通过Activity的getWindowManager()获取WindowManagerImpl对象，这里获取的WindowManagerImpl其实是Activity自己的WindowManagerImpl，

private WindowManagerImpl(Display display, Window parentWindow) {

mDisplay = display;

mParentWindow = parentWindow;

}

它的mParentWindow 是非空的，获取WindowManagerImpl之后，便利用 addView(decor, l)将DecorView对应的窗口添加到WMS中去，最后调用的是

@Override

public void addView(@NonNull View view, @NonNull ViewGroup.LayoutParams params) {

applyDefaultToken(params);

mGlobal.addView(view, params, mDisplay, mParentWindow);

}

可以看到这里会传递mParentWindow给WindowManagerGlobal对象，作为调整WindowMangaer.LayoutParams 中token的依据：

public void addView(View view, ViewGroup.LayoutParams params,

Display display, Window parentWindow) {

final WindowManager.LayoutParams wparams = (WindowManager.LayoutParams) params;

<!--调整wparams的token参数-->

if (parentWindow != null) {

parentWindow.adjustLayoutParamsForSubWindow(wparams);

}

ViewRootImpl root;

View panelParentView = null;

..

<!--新建ViewRootImpl ,并利用wparams参数添加窗口-->

root = new ViewRootImpl(view.getContext(), display);

view.setLayoutParams(wparams);

..

<!--新建ViewRootImpl -->

root.setView(view, wparams, panelParentView);

}

parentWindow.adjustLayoutParamsForSubWindow是一个很关键的函数，从名字就能看出，这是为了他调整子窗口的参数：

void adjustLayoutParamsForSubWindow(WindowManager.LayoutParams wp) {

CharSequence curTitle = wp.getTitle();

<!--如果是子窗口如何处理-->

if (wp.type >= WindowManager.LayoutParams.FIRST\_SUB\_WINDOW &&

wp.type <= WindowManager.LayoutParams.LAST\_SUB\_WINDOW) {

<!--后面会看到，其实PopupWindow类的子窗口的wp.token是在上层显示赋值的-->

if (wp.token == null) {

View decor = peekDecorView();

if (decor != null) {

// 这里其实是父窗口的IWindow对象 Window只有Dialog跟Activity才有

wp.token = decor.getWindowToken();

}

}

} else {

<!--这里其实只对应用窗口有用 Activity与Dialog都一样-->

if (wp.token == null) {

wp.token = mContainer == null ? mAppToken : mContainer.mAppToken;

}

}

}

对于Activity来说，wp.token = mContainer == null ? mAppToken : mContainer.mAppToken，其实就是AMS端传过来的IApplicationToken，之后在ViewRootImpl中setView的时候，会利用IWindowSession代理与WMS端的Session通信，将窗口以及token信息传递到WMS端，其中IApplicationToken就是该Activity所处于的分组，在WMS端，会根据IApplicationToken IBinder键值，从全局的mTokenMap中找到对应的AppWindowToken。既然说分组，就应该有其他的子元素，下面看一下Activity上弹出Dialog的流程，进一步了解为什么Activity与它弹出的Dialog是统一分组（复用同一套token）。

## Dialg分组及显示原理：为什么Activity与Dialog算同一组？

在添加到WMS的时候，Dialog的窗口属性是WindowManager.LayoutParams.TYPE\_APPLICATION，同样属于应用窗口，因此，必须使用Activity的AppToken才行，换句话说，必须使用Activity内部的WindowManagerImpl进行addView才可以。Dialog和Activity共享同一个WindowManager（也就是WindowManagerImpl），而WindowManagerImpl里面有个Window类型的mParentWindow变量，这个变量在Activity的attach中创建WindowManagerImpl时传入的为当前Activity的Window，而Activity的Window里面的mAppToken值又为当前Activity的token，所以Activity与Dialog共享了同一个mAppToken值，只是Dialog和Activity的Window对象不同,下面用代码确认一下：

Dialog(@NonNull Context context, @StyleRes int themeResId, boolean createContextThemeWrapper) {

<!--关键点 1 根据theme封装context-->

if (createContextThemeWrapper) {

...

mContext = new ContextThemeWrapper(context, themeResId);

} else {

mContext = context;

}

<!--获取mWindowManager-->

mWindowManager = (WindowManager) context.getSystemService(Context.WINDOW\_SERVICE);

<!--创建PhoneWindow-->

final Window w = new PhoneWindow(mContext);

mWindow = w;

w.setCallback(this);

w.setOnWindowDismissedCallback(this);

w.setWindowManager(mWindowManager, null, null);

w.setGravity(Gravity.CENTER);

mListenersHandler = new ListenersHandler(this);

}

以上代码先根据Theme调整context，之后利用context.getSystemService(Context.WINDOW\_SERVICE)，这里Dialog是从Activity弹出来的，所以context是Activity，如果你设置Application，会有如下error，至于为什么，后面分析会看到。

android.view.WindowManager$BadTokenException: Unable to add window -- token null is not for an application

at android.view.ViewRootImpl.setView(ViewRootImpl.java:563)

at android.view.WindowManagerGlobal.addView(WindowManagerGlobal.java:269)

at android.view.WindowManagerImpl.addView(WindowManagerImpl.java:69)

接着看Activity的getSystemService，上文分析过这种方法获取的其实是Activity中PhoneWindow的WindowManagerImpl，所以后面利用WindowManagerImpl addView的时候，走的流程与Activity一样。看一下show的代码：

public void show() {

...

onStart();

mDecor = mWindow.getDecorView();

...

<!--关键点 WindowManager.LayoutParams的获取-->

WindowManager.LayoutParams l = mWindow.getAttributes();

...

try {

mWindowManager.addView(mDecor, l);

mShowing = true;

sendShowMessage();

} finally {

}

}

Window在创建的时候，默认新建WindowManager.LayoutParams mWindowAttributes

private final WindowManager.LayoutParams mWindowAttributes =

new WindowManager.LayoutParams();

采用的是无参构造方法，

public LayoutParams() {

super(LayoutParams.MATCH\_PARENT, LayoutParams.MATCH\_PARENT);

type = TYPE\_APPLICATION;

format = PixelFormat.OPAQUE;

}

因此这里的type = TYPE\_APPLICATION，也就是说Dialog的窗口类型其实是应用窗口。因此在addView走到上文的adjustLayoutParamsForSubWindow的时候，仍然按照Activity的WindowManagerImpl addView的方式处理，并利用Activity的PhoneWindow的 adjustLayoutParamsForSubWindow调整参数，赋值给WindowManager.LayoutParams token的值仍然是Activity的IApplicationToken，同样在WMS端，对应就是APPWindowToken，也就是Activity与Dialog属于同一分组。

void adjustLayoutParamsForSubWindow(WindowManager.LayoutParams wp) {

CharSequence curTitle = wp.getTitle();

<!--这里其实只对应用窗口有用 Activity与Dialog都一样-->

if (wp.token == null) {

wp.token = mContainer == null ? mAppToken : mContainer.mAppToken;

}

}

回到之前遗留的一个问题，为什么Dialog用Application作为context不行呢？Dialog的窗口类型属于应用窗口，如果采用Application作为context，那么通过context.getSystemService(Context.WINDOW\_SERVICE)获取的WindowManagerImpl就不是Activity的WindowManagerImpl，而是Application，它同Activity的WindowManagerImpl的区别是没有parentWindow，所以adjustLayoutParamsForSubWindow函数不会被调用，WindowManager.LayoutParams的token就不会被赋值，因此ViewRootImpl在通过setView向WMS在添加窗口的时候会失败：

public int addWindow(Session session, IWindow client, XXX )

...

<!--对于应用窗口 token不可以为null-->

WindowToken token = mTokenMap.get(attrs.token);

if (token == null) {

if (type >= FIRST\_APPLICATION\_WINDOW && type <= LAST\_APPLICATION\_WINDOW) {

Slog.w(TAG, "Attempted to add application window with unknown token "

+ attrs.token + ". Aborting.");

return WindowManagerGlobal.ADD\_BAD\_APP\_TOKEN;

}

WMS会返回WindowManagerGlobal.ADD\_BAD\_APP\_TOKEN的错误给APP端，APP端ViewRootImpl端收到后会抛出如下异常

public void setView(View view, WindowManager.LayoutParams attrs, View panelParentView) {

synchronized (this) {

....

case WindowManagerGlobal.ADD\_NOT\_APP\_TOKEN:

throw new WindowManager.BadTokenException(

"Unable to add window -- token " + attrs.token

+ " is not for an application");

以上就为什么不能用Application作为Dialog的context的理由（**不能为Dialog提供正确的token**），接下来看一下PopupWindow是如何处理分组的。

## PopupWindow类子窗口的添加流程及WindowToken分组

PopupWindow是最典型的子窗口，必须依附父窗口才能存在，先看下PopupWindow一般用法：

View root = LayoutInflater.from(AppProfile.getAppContext()).inflate(R.layout.pop\_window, null);

PopupWindow popupWindow = new PopupWindow(root, ViewGroup.LayoutParams.MATCH\_PARENT, ViewGroup.LayoutParams.WRAP\_CONTENT, true);

popupWindow.setBackgroundDrawable(new BitmapDrawable());

popupWindow.showAsDropDown(archorView);

PopupWindow的构造函数很普通，主要是一些默认入场、出厂动画的设置，如果在新建PopupWindow的时候已经将根View传递到构造函数中去，PopupWindow的构造函数会调用setContentView，如果在show之前，没有调用setContentView，则抛出异常。

public PopupWindow(View contentView, int width, int height, boolean focusable) {

if (contentView != null) {

mContext = contentView.getContext();

mWindowManager = (WindowManager) mContext.getSystemService(Context.WINDOW\_SERVICE);

}

setContentView(contentView);

setWidth(width);

setHeight(height);

setFocusable(focusable);

}

下面主要看PopupWindow的showAsDropDown函数

public void showAsDropDown(View anchor, int xoff, int yoff, int gravity) {

<!--关键点1 利用通过View锚点所在窗口显性构建PopupWindow的token-->

final WindowManager.LayoutParams p = createPopupLayoutParams(anchor.getWindowToken());

<!--关键点2-->

preparePopup(p);

...

<!--关键点3-->

invokePopup(p);

}

showAsDropDown有3个关键点，关键点1是生成WindowManager.LayoutParams参数，WindowManager.LayoutParams参数里面的type、token是非常重要参数，**PopupWindow的type是TYPE\_APPLICATION\_PANEL = FIRST\_SUB\_WINDOW**，是一个子窗口。关键点2是PopupDecorView的生成，这个View是PopupWindow的根ViewGroup，类似于Activity的DecorView，关键3利用WindowManagerService的代理，将View添加到WMS窗口管理中去显示，先看关键点1：

private WindowManager.LayoutParams createPopupLayoutParams(IBinder token) {

final WindowManager.LayoutParams p = new WindowManager.LayoutParams();

p.gravity = computeGravity();

p.flags = computeFlags(p.flags);

p.type = mWindowLayoutType;

<!--显性赋值token-->

p.token = token;

p.softInputMode = mSoftInputMode;

p.windowAnimations = computeAnimationResource();

if (mBackground != null) {

p.format = mBackground.getOpacity();

} else {

p.format = PixelFormat.TRANSLUCENT;

}

..

p.privateFlags = PRIVATE\_FLAG\_WILL\_NOT\_REPLACE\_ON\_RELAUNCH

| PRIVATE\_FLAG\_LAYOUT\_CHILD\_WINDOW\_IN\_PARENT\_FRAME;

return p;

}

上面的Token其实用的是anchor.getWindowToken()，**如果是Activity中的View，其实用的Token就是Activity的ViewRootImpl中的IWindow对象**，如果这个View是一个系统窗口中的View，比如是Toast窗口中弹出来的，用的就是Toast ViewRootImpl的IWindow对象，归根到底，PopupWindow自窗口中的Token是ViewRootImpl的IWindow对象，同Activity跟Dialog的token（IApplicationToken）不同，该Token标识着PopupWindow在WMS所处的分组，最后来看一下PopupWindow的显示:

private void invokePopup(WindowManager.LayoutParams p) {

if (mContext != null) {

p.packageName = mContext.getPackageName();

}

final PopupDecorView decorView = mDecorView;

decorView.setFitsSystemWindows(mLayoutInsetDecor);

setLayoutDirectionFromAnchor();

<!--关键点1-->

mWindowManager.addView(decorView, p);

if (mEnterTransition != null) {

decorView.requestEnterTransition(mEnterTransition);

}

}

主要是调用了WindowManager的addView添加视图并显示，这里首先需要关心一下mWindowManager，

mWindowManager = (WindowManager) context.getSystemService(Context.WINDOW\_SERVICE);

这的context 可以是Activity，也可以是Application，因此WindowManagerImpl也可能不同，不过这里并没有多大关系，因为PopupWindow的token是显性赋值的，就是是就算用Application，也不会有什么问题，对于PopupWindow子窗口，关键点是View锚点决定其token，而不是WindowManagerImpl对象：

@Override

public void addView(@NonNull View view, @NonNull ViewGroup.LayoutParams params) {

applyDefaultToken(params);

mGlobal.addView(view, params, mContext.getDisplay(), mParentWindow);

}

之后利用ViewRootImpl的setView函数的时候，WindowManager.LayoutParams里的token其实就是view锚点获取的IWindow对象，WindowManagerService在处理该请求的时候，

public int addWindow(Session session, IWindow client, XXX ) {

<!--关键点1，必须找到子窗口的父窗口，否则添加失败-->

WindowState attachedWindow = null;

if (type >= FIRST\_SUB\_WINDOW && type <= LAST\_SUB\_WINDOW) {

attachedWindow = windowForClientLocked(null, attrs.token, false);

if (attachedWindow == null) {

return WindowManagerGlobal.ADD\_BAD\_SUBWINDOW\_TOKEN;

}

}

<!--关键点2 如果Activity第一次添加子窗口 ，子窗口分组对应的WindowToken一定是null-->

boolean addToken = false;

WindowToken token = mTokenMap.get(attrs.token);

AppWindowToken atoken = null;

if (token == null) {

...

token = new WindowToken(this, attrs.token, -1, false);

addToken = true;

}

<!--关键点2 新建窗口WindowState对象 注意这里的attachedWindow非空-->

WindowState win = new WindowState(this, session, client, token,

attachedWindow, appOp[0], seq, attrs, viewVisibility, displayContent);

...

<!--关键点4 添加更新全部map，-->

if (addToken) {

mTokenMap.put(attrs.token, token);

}

mWindowMap.put(client.asBinder(), win);

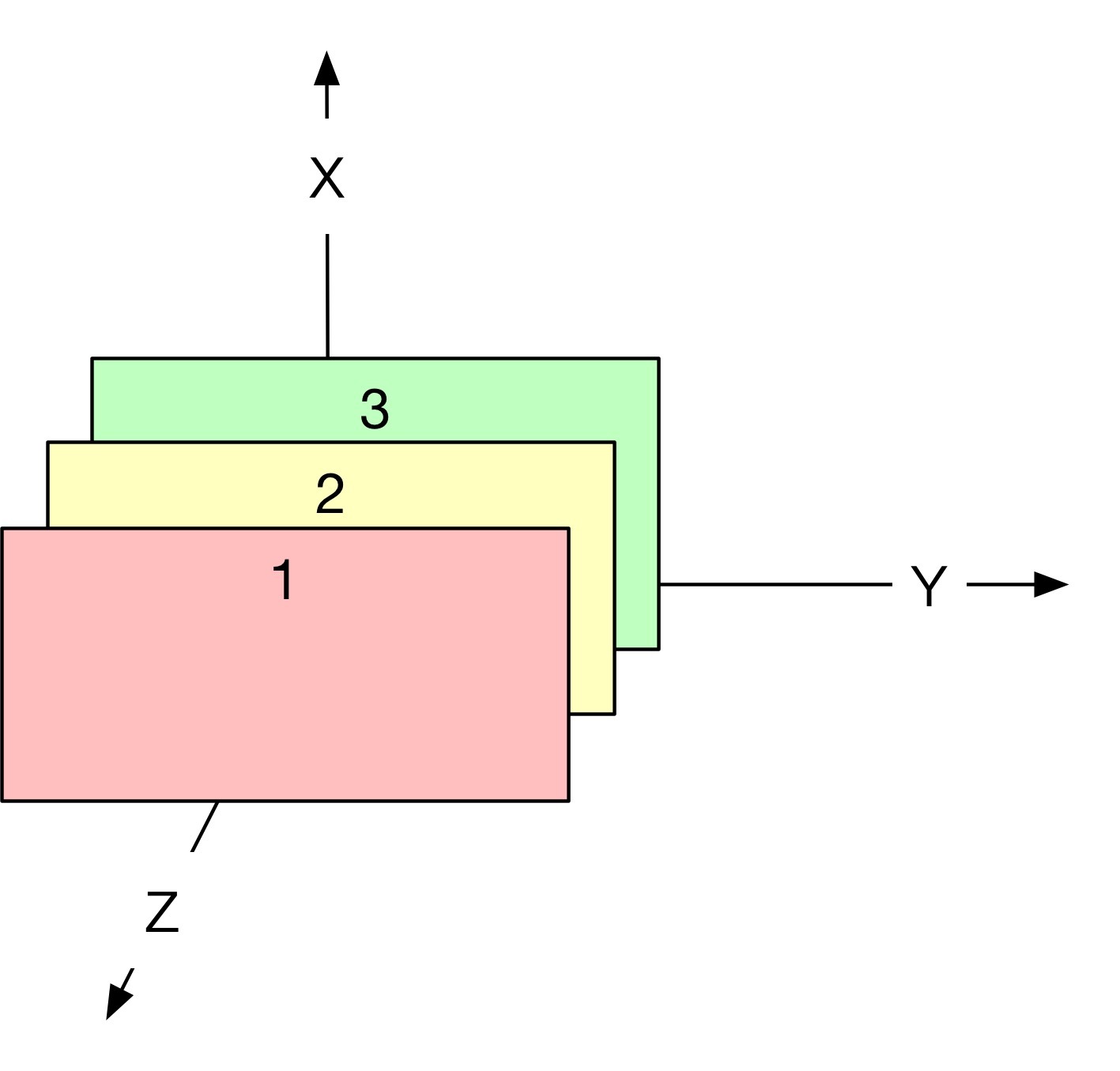
}

从上面的分析可以看出，WMS会为PopupWindow窗口创建一个子窗口分组WindowToken，每个子窗口都会有一个指向父窗口的引用，因为是利用父窗口的IWindow作为键值，父窗口可以很方便的利用自己的IWindow获取WindowToken，进而得到全部的子窗口，

关于系统窗口，前文层分析过Toast系统窗口，Toast类系统窗口在WMS端只有一个WindowToken，键值是null，这个比较奇葩，不过还没验证过。

## 窗口的Z次序管理：窗口的分配序号、次序调整等

虽然我们看到的手机屏幕只是一个二维平面X\*Y，但其实Android系统是有隐形的Z坐标轴的，其方向与手机屏幕垂直，与我们的实现平行，所以并不能感知到。



Z order.jpg

前面分析了窗口分组的时候涉及了两个对象WindowState与Windtoken，但仅限分组，分组无法决定窗口的显示的Z-order，那么再WMS是怎么管理所有窗口的Z-order的？ 在WMS中窗口被抽象成WindowState，因此WindowState内部一定有属性来标志这个窗口的Z-order，实现也确实如此，WindowState采用三个个int值mBaseLayer+ mSubLayer + mLayer 来标志窗口所处的位置，前两个主要是根据窗口类型确定窗口位置，mLayer才是真正的值，定义如下：

final class WindowState implements WindowManagerPolicy.WindowState {

final WindowList mChildWindows = new WindowList();

final int mBaseLayer;

final int mSubLayer;

<!--最终Z次序的赋值-->

int mLayer;

}

从名字很容知道mBaseLayer是标志窗口的主次序，面向的是一个窗口组，而mSubLayer主要面向单独窗口，要来标志一个窗口在一组窗口中的位置，对两者来说值越大，窗口越靠前，从此final属性知道，两者的值是不能修改的，而mLayer可以修改，对于系统窗口，一般不会同时显示两个，因此，可以用主序决定，比较特殊的就是Activity与子窗口，首先子窗口的主序肯定是父窗口决定的，子窗口只关心次序就行。而父窗口的主序却相对麻烦，比如对于应用窗口来说，他们的主序都是一样的，因此还要有一个其他的维度来作为参考，比如**对于Activity，主序都是一样的，怎么定他们真正的Z-order呢？其实Activity的顺序是由AMS保证的，这个顺序定了，WMS端Activity窗口的顺序也是定了，这样下来次序也方便定了**。

WindowState(WindowManagerService service, Session s, IWindow c, WindowToken token,

WindowState attachedWindow, int appOp, int seq, WindowManager.LayoutParams a,

int viewVisibility, final DisplayContent displayContent) {

...

<!--关键点1 子窗口类型的Z order-->

if ((mAttrs.type >= FIRST\_SUB\_WINDOW &&

mAttrs.type <= LAST\_SUB\_WINDOW)) {

mBaseLayer = mPolicy.windowTypeToLayerLw(

attachedWindow.mAttrs.type) \* WindowManagerService.TYPE\_LAYER\_MULTIPLIER

+ WindowManagerService.TYPE\_LAYER\_OFFSET;

mSubLayer = mPolicy.subWindowTypeToLayerLw(a.type);

mAttachedWindow = attachedWindow; final WindowList childWindows = mAttachedWindow.mChildWindows;

final int numChildWindows = childWindows.size();

if (numChildWindows == 0) {

childWindows.add(this);

} else {

...

} else {

<!--关键点2 普通窗口类型的Z order-->

mBaseLayer = mPolicy.windowTypeToLayerLw(a.type)

\* WindowManagerService.TYPE\_LAYER\_MULTIPLIER

+ WindowManagerService.TYPE\_LAYER\_OFFSET;

mSubLayer = 0;

mAttachedWindow = null;

mLayoutAttached = false;

}

...

}

由于窗口所能选择的类型是确定的，因此mBaseLayer与mSubLayer所能选择的值只有固定几个，很明显这两个参数不能精确的确定Z-order，还会有其他微调的手段，也仅限微调，**在系统层面，决定了不同类型窗口所处的位置，比如系统Toast类型的窗口一定处于所有应用窗口之上**，不过我们最关心的是Activity类的窗口如何确定Z-order的，在new WindowState之后，只是粗略的确定了Activity窗口的次序，看一下添加窗口的示意代码：

addWindow(){

<!--1-->

new WindowState

<!--2-->

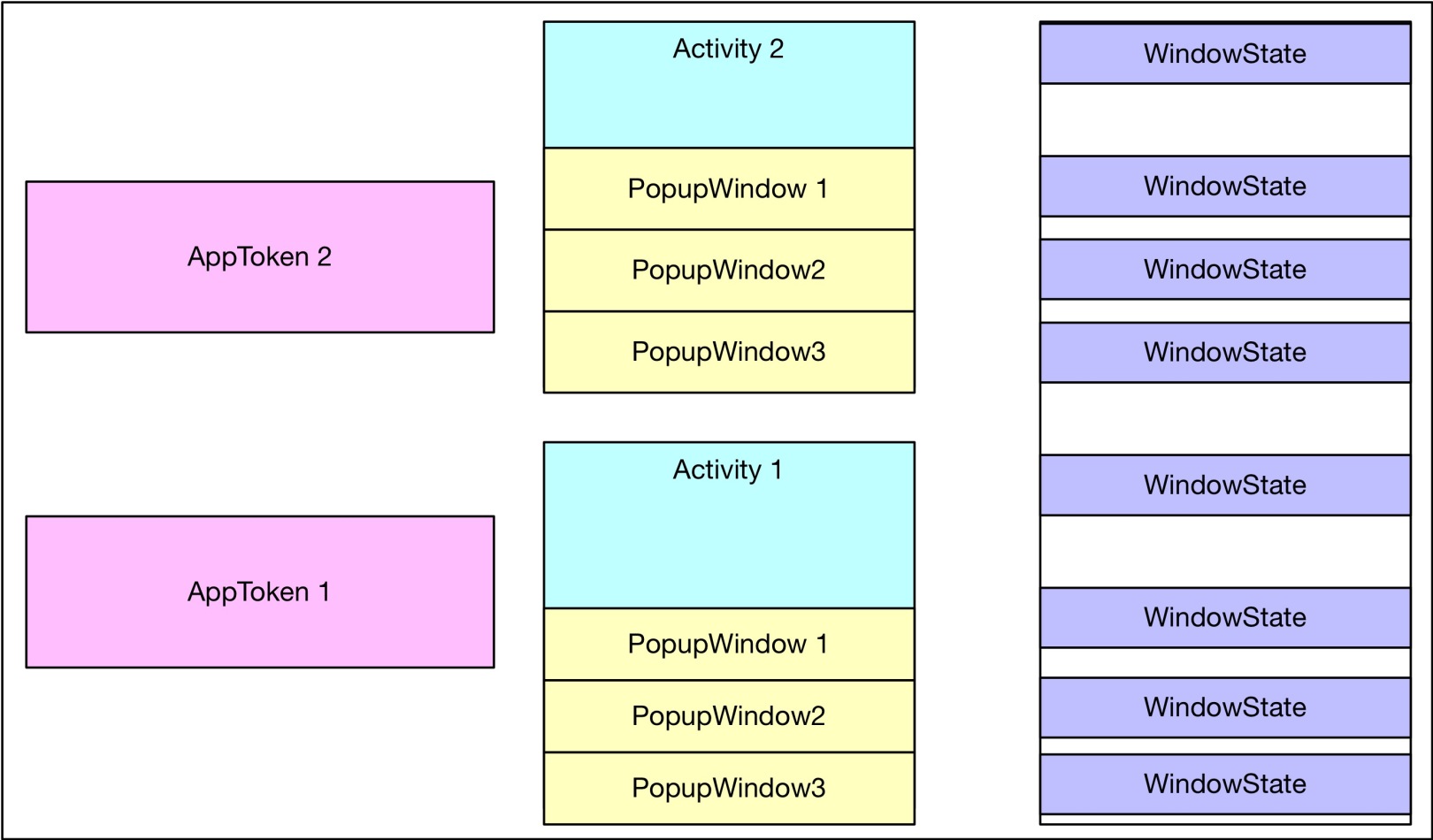
addWindowToListInOrderLocked(win, true);

<!--3-->

assignLayersLocked(displayContent.getWindowList());

}

新建state对象之后，Z-order还要通过addWindowToListInOrderLocked及assignLayersLocked才能确定，addWindowToListInOrderLocked主要是根据窗口的Token找到归属，插入到对应Token的WindowState列表，如果是子窗口还要插入到父窗口的对应位置中：



次序确定.jpg

插入到特定位置后其实Z-order就确定了，接下来就是通过assignLayersLocked为WindowState分配真正的Z-order mLayer,

private final void assignLayersLocked(WindowList windows) {

int N = windows.size();

int curBaseLayer = 0;

int curLayer = 0;

int i;

boolean anyLayerChanged = false;

for (i=0; i<N; i++) {

final WindowState w = windows.get(i);

final WindowStateAnimator winAnimator = w.mWinAnimator;

boolean layerChanged = false;

int oldLayer = w.mLayer;

if (w.mBaseLayer == curBaseLayer || w.mIsImWindow

|| (i > 0 && w.mIsWallpaper)) {

<!--通过偏移量-->

curLayer += WINDOW\_LAYER\_MULTIPLIER;

w.mLayer = curLayer;

} else {

curBaseLayer = curLayer = w.mBaseLayer;

w.mLayer = curLayer;

}

if (w.mLayer != oldLayer) {

layerChanged = true;

anyLayerChanged = true;

}

...

}

mLayer最终确定后，窗口的次序也就确定了，这个顺序要最终通过后续的relayout更新到SurfaceFlinger服务，之后，SurfaceFlinger在图层混排的时候才知道如何处理。

## WMS中窗口次序分配如何影响SurfaceFlinger服务

SurfaceFlinger在图层混排的时候应该不会混排所有的窗口，只会混排可见的窗口，比如有多个全屏Activity的时候，SurfaceFlinger只会处理最上面的，那么SurfaceFlinger如何知道哪些窗口可见哪些不可见呢？前文分析了WMS分配Z-order之后，要通过setLayer更新到SurfaceFlinger，接下来看具体流程，创建SurfaceControl之后，会创建一次事务，确定Surface的次序：

SurfaceControl.openTransaction();

try {

mSurfaceX = left;

mSurfaceY = top;

try {

mSurfaceControl.setPosition(left, top);

mSurfaceLayer = mAnimLayer;

final DisplayContent displayContent = w.getDisplayContent();

if (displayContent != null) {

mSurfaceControl.setLayerStack(displayContent.getDisplay().getLayerStack());

}

<!--设置次序-->

mSurfaceControl.setLayer(mAnimLayer);

mSurfaceControl.setAlpha(0);

mSurfaceShown = false;

} catch (RuntimeException e) {

mService.reclaimSomeSurfaceMemoryLocked(this, "create-init", true);

}

mLastHidden = true;

} finally {

SurfaceControl.closeTransaction();

}

}

这里通过openTransaction与closeTransaction保证一次事务的完整性，中间就Surface次序的调整，closeTransaction会与SurfaceFlinger通信，通知SurfaceFlinger更新Surface信息，这其中就包括Z-order。

# 总结

本文简要分析了Android窗口的分组，以及WMS窗口次序的确定，最后简单提及了一下窗口次序如何更新到SurfaceFlinger服务的，也方便将来理解图层合成。

作者：看书的小蜗牛 原文链接:[Android窗口管理分析（3）：窗口分组及Z-order的确定](https://www.jianshu.com/p/90ede7b2a64a) **仅供参考，欢迎指正**

本文参与[腾讯云自媒体分享计划](https://cloud.tencent.com/developer/support-plan)，欢迎正在阅读的你也加入，一起分享。

发表于 2018-06-29

# Android窗口类型

2014年07月22日 09:22:33 [leopard21](https://me.csdn.net/leopard21) 阅读数：2054

WindowManager会根据窗口的类型来决定Z-Order，通过Z-Order也就决定了窗口从上到下的显示顺序。下表个表明了窗口类型与Z-Order的关系。表中layer值越大表示窗口越靠上。

System

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | WINDOW TYPE | 描述 | LAYER | 是否HIDE |
| 1 | TYPE\_UNIVERSE\_BACKGROUND | Behind the universe of the real windows. In multiuser systems shows on all users' windows. | 1 | hide |
| 2 | TYPE\_PRIVATE\_PRESENTATION | Window for Presentation on top of private virtual display. | 2 |  |
| 3 | TYPE\_WALLPAPER | wallpaper window, placed behind any window that wants  to sit on top of the wallpaper. In multiuser systems shows only on the owning user's window.wallpaper is at the bottom, though the window manager may move it. | 2 |  |
| 4 | TYPE\_PHONE | These are non-application windows providing user interaction with the phone (in particular incoming calls).These windows are normally placed above all applications, but behind the status bar.In multiuser systems shows on all users' windows. | 3 |  |
| 5 | TYPE\_SEARCH\_BAR | the search bar.  There can be only one search bar window; it is placed at the top of the screen. In multiuser systems shows on all users' windows. | 4 |  |
| 6 | TYPE\_RECENTS\_OVERLAY | Same layer as {@link #TYPE\_SYSTEM\_DIALOG} but only appears on one user's screen.In multiuser systems shows on all users' windows. | 5 | hide |
| 7 | TYPE\_SYSTEM\_DIALOG | panel that slides out from the status bar In multiuser systems shows on all users' windows. | 5 |  |
| 8 | TYPE\_TOAST | transient notifications. In multiuser systems shows only on the owning user's window.toasts and the plugged-in battery thing | 6 |  |
| 9 | TYPE\_PRIORITY\_PHONE | priority phone UI, which needs to be displayed even if the keyguard is active.  These windows must not take input focus, or they will interfere with the keyguard. In multiuser systems shows on all users' windows.SIM errors and unlock.  Not sure if this really should be in a high layer. | 7 |  |
| 10 | TYPE\_DREAM | Dreams (screen saver) window, just above keyguard. In multiuser systems shows only on the owning user's window.used for Dreams (screensavers with TYPE\_DREAM windows) | 8 | hide |
| 11 | TYPE\_SYSTEM\_ALERT | system window, such as low power alert. These windows are always on top of application windows. In multiuser systems shows only on the owning user's window.like the ANR / app crashed dialogs | 9 |  |
| 12 | TYPE\_INPUT\_METHOD | internal input methods windows, which appear above the normal UI.  Application windows may be resized or panned to keep the input focus visible while this window is displayed. In multiuser systems shows only on the owning user's window. on-screen keyboards and other such input method user interfaces go here. | 10 |  |
| 13 | TYPE\_INPUT\_METHOD\_DIALOG | internal input methods dialog windows, which appear above the current input method window. In multiuser systems shows only on the owning user's window.on-screen keyboards and other such input method user interfaces go here. | 11 |  |
| 14 | TYPE\_KEYGUARD\_SCRIM | keyguard scrim window. Shows if keyguard needs to be restarted.In multiuser systems shows on all users' windows. the safety window that shows behind keyguard while keyguard is starting | 12 | hide |
| 15 | TYPE\_KEYGUARD | keyguard window.In multiuser systems shows on all users' windows. the keyguard; nothing on top of these can take focus, since they are responsible for power management when displayed. | 13 |  |
| 16 | TYPE\_KEYGUARD\_DIALOG | dialogs that the keyguard shows In multiuser systems shows on all users' windows. | 14 |  |
| 17 | TYPE\_STATUS\_BAR\_SUB\_PANEL | panel that slides out from under the status bar In multiuser systems shows on all users' windows. | 15 | hide |
| 18 | TYPE\_STATUS\_BAR | the status bar.  There can be only one status bar window; it is placed at the top of the screen, and all other windows are shifted down so they are below it. In multiuser systems shows on all users' windows. | 16 |  |
| 19 | TYPE\_STATUS\_BAR\_PANEL | panel that slides out from over the status bar In multiuser systems shows on all users' windows. | 17 |  |
| 20 | TYPE\_VOLUME\_OVERLAY | The volume level overlay/dialog shown when the user changes the system volume. In multiuser systems shows on all users' windows.the on-screen volume indicator and controller shown when the user changes the device volume | 18 | hide |
| 21 | TYPE\_SYSTEM\_OVERLAY | system overlay windows, which need to be displayed on top of everything else.  These windows must not take input focus, or they will interfere with the keyguard. In multiuser systems shows only on the owning user's window.the on-screen volume indicator and controller shown when the user changes the device volume | 19 |  |
| 22 | TYPE\_NAVIGATION\_BAR | Navigation bar (when distinct from status bar) In multiuser systems shows on all users' windows.the navigation bar, if available, shows atop most things | 20 | hide |
| 23 | TYPE\_NAVIGATION\_BAR\_PANEL | Navigation bar panel (when navigation bar is distinct from status bar) In multiuser systems shows on all users' windows. some panels (e.g. search) need to show on top of the navigation bar | 21 | hide |
| 24 | TYPE\_SYSTEM\_ERROR | internal system error windows, appear on top of everything they can.In multiuser systems shows only on the owning user's window.system-level error dialogs | 22 |  |
| 25 | TYPE\_MAGNIFICATION\_OVERLAY | Magnification overlay window. Used to highlight the magnified portion of a display when accessibility magnification is enabled.In multiuser systems shows on all users' windows.used to highlight the magnified portion of a display | 23 | hide |
| 26 | TYPE\_DISPLAY\_OVERLAY | Display overlay window.  Used to simulate secondary display devices.In multiuser systems shows on all users' windows.used to simulate secondary display devices | 24 | hide |
| 27 | TYPE\_DRAG | the drag-and-drop pseudowindow.  There is only one drag layer (at most), and it is placed on top of all other windows.In multiuser systems shows only on the owning user's window.the drag layer: input for drag-and-drop is associated with this window,which sits above all other focusable windows | 25 | hide |
| 28 | TYPE\_SECURE\_SYSTEM\_OVERLAY | secure system overlay windows, which need to be displayed on top of everything else.  These windows must not take input focus, or they will interfere with the keyguard.This is exactly like {@link #TYPE\_SYSTEM\_OVERLAY} except that only the system itself is allowed to create these overlays.  Applications cannotobtain permission to create secure system overlays.In multiuser systems shows only on the owning user's window. | 26 | hide |
| 29 | TYPE\_BOOT\_PROGRESS | The boot progress dialog, goes on top of everything in the world.In multiuser systems shows on all users' windows. | 27 | hide |
| 30 | TYPE\_POINTER | (mouse) pointer In multiuser systems shows on all users' windows.the (mouse) pointer layer | 28 | hide |
| 31 | TYPE\_HIDDEN\_NAV\_CONSUMER | Fake window to consume touch events when the navigation  bar is hidden. In multiuser systems shows on all users' windows. | 29 | hide |

Application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | WINDOW TYPE | 描述 | LAYER | 是否HIDE |
| 1 | FIRST\_APPLICATION\_WINDOW | Start of window types that represent normal application windows. | 2 |  |
| 2 | TYPE\_BASE\_APPLICATION | an application window that serves as the "base" window  of the overall application; all other application windows will appear on top of it.In multiuser systems shows only on the owning user's window. | 2 |  |
| 3 | TYPE\_APPLICATION | a normal application window.  The {@link #token} must be an Activity token identifying who the window belongs to. In multiuser systems shows only on the owning user's window. | 2 |  |
| 4 | TYPE\_APPLICATION\_STARTING | special application window that is displayed while the application is starting.  Not for use by applications themselves;this is used by the system to display something until the application can show its own windows.In multiuser systems shows on all users' windows. | 2 |  |
| 5 | LAST\_APPLICATION\_WINDOW | End of types of application windows. | 2 |  |

Sub Window

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | WINDOW TYPE | 描述 | LAYER | 是否HIDE |
| 1 | FIRST\_SUB\_WINDOW | Start of types of sub-windows.  The {@link #token} of these windows must be set to the window they are attached to.  These types of windows are kept next to their attached window in Z-order, and their coordinate space is relative to their attached window. | 1 |  |
| 2 | TYPE\_APPLICATION\_PANEL | a panel on top of an application window.  These windows  appear on top of their attached window. | 1 |  |
| 3 | TYPE\_APPLICATION\_ATTACHED\_DIALOG | like {@link #TYPE\_APPLICATION\_PANEL}, but layout of the window happens as that of a top-level window, as a child of its container. | 1 |  |
| 4 | TYPE\_APPLICATION\_MEDIA | window for showing media (such as video).  These windows are displayed behind their attached window. | 1 |  |
| 5 | TYPE\_APPLICATION\_SUB\_PANEL | a sub-panel on top of an application window.  These windows are displayed on top their attached window and any {@link #TYPE\_APPLICATION\_PANEL} panels. | 2 |  |
| 6 | TYPE\_APPLICATION\_MEDIA\_OVERLAY | window for showing overlays on top of media windows. These windows are displayed between TYPE\_APPLICATION\_MEDIA and the application window.  They should be translucent to be useful.  This is a big ugly hack so: | -1 | hide |

Sub Window对应的主窗口为0

Media Sublayer(-2) < Media Overlay sublayer (-1) < Main Layer(0) < Attached Dialog (1) < Sub panel Sublayer (2)

子窗口与父窗口共用一个token。的父窗口一般是所在的Activity，比如surfaceView是一个子窗口，他的父窗口是所在activity所在的window。

窗口类型与Layer转换的函数为windowTypeToLayerLw

/frameworks/base/policy/src/com/android/internal/policy/impl/PhoneWindowManager.java

1. public int windowTypeToLayerLw(int type) {
2. if (type >= FIRST\_APPLICATION\_WINDOW && type <= LAST\_APPLICATION\_WINDOW) {
3. return 2;
4. }
5. switch (type) {
6. case TYPE\_UNIVERSE\_BACKGROUND:
7. return 1;
8. case TYPE\_WALLPAPER:
9. // wallpaper is at the bottom, though the window manager may move it.
10. return 2;
11. case TYPE\_PHONE:
12. return 3;
13. case TYPE\_SEARCH\_BAR:
14. return 4;
15. case TYPE\_RECENTS\_OVERLAY:
16. case TYPE\_SYSTEM\_DIALOG:
17. return 5;
18. case TYPE\_TOAST:
19. // toasts and the plugged-in battery thing
20. return 6;
21. case TYPE\_PRIORITY\_PHONE:
22. // SIM errors and unlock. Not sure if this really should be in a high layer.
23. return 7;
24. case TYPE\_DREAM:
25. // used for Dreams (screensavers with TYPE\_DREAM windows)
26. return 8;
27. case TYPE\_SYSTEM\_ALERT:
28. // like the ANR / app crashed dialogs
29. return 9;
30. case TYPE\_INPUT\_METHOD:
31. // on-screen keyboards and other such input method user interfaces go here.
32. return 10;
33. case TYPE\_INPUT\_METHOD\_DIALOG:
34. // on-screen keyboards and other such input method user interfaces go here.
35. return 11;
36. case TYPE\_KEYGUARD:
37. // the keyguard; nothing on top of these can take focus, since they are
38. // responsible for power management when displayed.
39. return 12;
40. case TYPE\_KEYGUARD\_DIALOG:
41. return 13;
42. case TYPE\_STATUS\_BAR\_SUB\_PANEL:
43. return 14;
44. case TYPE\_STATUS\_BAR:
45. return 15;
46. case TYPE\_STATUS\_BAR\_PANEL:
47. return 16;
48. case TYPE\_VOLUME\_OVERLAY:
49. // the on-screen volume indicator and controller shown when the user
50. // changes the device volume
51. return 17;
52. case TYPE\_SYSTEM\_OVERLAY:
53. // the on-screen volume indicator and controller shown when the user
54. // changes the device volume
55. return 18;
56. case TYPE\_NAVIGATION\_BAR:
57. // the navigation bar, if available, shows atop most things
58. return 19;
59. case TYPE\_NAVIGATION\_BAR\_PANEL:
60. // some panels (e.g. search) need to show on top of the navigation bar
61. return 20;
62. case TYPE\_SYSTEM\_ERROR:
63. // system-level error dialogs
64. return 21;
65. case TYPE\_MAGNIFICATION\_OVERLAY:
66. // used to highlight the magnified portion of a display
67. return 22;
68. case TYPE\_DISPLAY\_OVERLAY:
69. // used to simulate secondary display devices
70. return 23;
71. case TYPE\_DRAG:
72. // the drag layer: input for drag-and-drop is associated with this window,
73. // which sits above all other focusable windows
74. return 24;
75. case TYPE\_SECURE\_SYSTEM\_OVERLAY:
76. return 25;
77. case TYPE\_BOOT\_PROGRESS:
78. return 26;
79. case TYPE\_POINTER:
80. // the (mouse) pointer layer
81. return 27;
82. case TYPE\_HIDDEN\_NAV\_CONSUMER:
83. return 28;
84. }
85. Log.e(TAG, "Unknown window type: " + type);
86. return 2;
87. }
88. public int subWindowTypeToLayerLw(int type) {
89. switch (type) {
90. case TYPE\_APPLICATION\_PANEL:
91. case TYPE\_APPLICATION\_ATTACHED\_DIALOG:
92. return APPLICATION\_PANEL\_SUBLAYER;
93. case TYPE\_APPLICATION\_MEDIA:
94. return APPLICATION\_MEDIA\_SUBLAYER;
95. case TYPE\_APPLICATION\_MEDIA\_OVERLAY:
96. return APPLICATION\_MEDIA\_OVERLAY\_SUBLAYER;
97. case TYPE\_APPLICATION\_SUB\_PANEL:
98. return APPLICATION\_SUB\_PANEL\_SUBLAYER;
99. }
100. Log.e(TAG, "Unknown sub-window type: " + type);
101. return 0;
102. }

1).  Window类的基本属性如下：  
  
          FEATURE\_OPTIONS\_PANEL = 0;                      可选的Panel标志（默认使能）   
          FEATURE\_NO\_TITLE = 1;                                 无标题栏   
          FEATURE\_PROGRESS = 2;                                 标题栏上显示加载进度，如webview加载网页时（条状进度条）  
          FEATURE\_LEFT\_ICON = 3;                                标题栏左侧显示一个图标   
          FEATURE\_RIGHT\_ICON = 4;                              标题栏右侧显示一个图标   
          FEATURE\_INDETERMINATE\_PROGRESS = 5;   不确定的进度（圆圈状等待图标）   
          FEATURE\_CONTEXT\_MENU = 6;                        Menu菜单（默认使能）   
          FEATURE\_CUSTOM\_TITLE = 7;                          自定义标题栏，该属性不能与其他标题栏属性合用  
          FEATURE\_ACTION\_BAR = 8;                               ActionBar属性，ActionBar显示的，TitleBar就没有啦                                                               
  
          FEATURE\_ACTION\_BAR\_OVERLAY = 9;            覆盖在内容之上的ActionBar  
  
          FEATURE\_ACTION\_MODE\_OVERLAY = 10;        AcitionBar的覆盖在内容的模式  
  
          FEATURE\_OPENGL = 8;                                       若开启OpenGL，那么2D将由OpenGL处理（OpenGL中2D是3D的子集）  
          PROGRESS\_VISIBILITY\_ON = -1;                         进度条可见   
          PROGRESS\_VISIBILITY\_OFF = -2;                       进度条不可见   
          PROGRESS\_INDETERMINATE\_ON = -3;              开启不确定模式   
          PROGRESS\_INDETERMINATE\_OFF = -4;             关闭不确定模式   
          PROGRESS\_START = 0;                                        第一进度条的最小值   
          PROGRESS\_END = 10000;                                     第一进度条的最大值   
          PROGRESS\_SECONDARY\_START = 20000;          第二进度条的最小值   
          PROGRESS\_SECONDARY\_END = 30000;              第二进度条的最大值   
  
       经常使用到的：   
  
    1、隐藏标题栏   
  
        requestWindowFeature(Window.FEATURE\_NO\_TITLE);   
  
    2、在标题栏显示进度条   
  
        requestWindowFeature(Window.FEATURE\_PROGRESS);   
  
        setContentView(R.layout.progressbar\_1);   
  
        setProgressBarVisibility(true);   
  
        final ProgressBar progressHorizontal = (ProgressBar)findViewById(R.id.progress\_horizontal);  
  
        setProgress(progressHorizontal.getProgress() \* 100);   
  
        setSecondaryProgress(progressHorizontal.getSecondaryProgress()\* 100);   
  
    3、使用自定义标题栏   
  
        requestWindowFeature(Window.FEATURE\_CUSTOM\_TITLE);   
  
        setContentView(R.layout.xxx);   
  
        getWindow().setFeatureInt(Window.FEATURE\_CUSTOM\_TITLE,R.layout.my\_title\_bar);  
  
    4、清除标题栏内容，而区域保留   
  
        ((ViewGroup)getWindow().findViewById(com.android.internal.R.id.title\_container)).removeAllViews();  
  
    5、隐藏标题栏   
  
        ((ViewGroup)getWindow(). findViewById(com.android.internal.R.id.title\_container)).setVisibility(View.GONE);  
  
    6、显示标题栏   
  
        ((ViewGroup)getWindow(). findViewById(com.android.internal.R.id.title\_container)).setVisibility(View.VISIBLE);  
  
    注意事项   
  
    (1) requestWindowFeature()要在setContentView()之前调用；   
  
    (2) 设置各种Feature，是具有排它性的，一旦设置，后续不可更改为别的类型；   
  
    (3)当使用TabHost(由ActivityGroup派生)时，各个Tab里的Activity，要么都是NO\_TITLE，要么都是CUSTOM\_TITLE，无法分别进行设置  
  
    ID\_ANDROID\_CONTENT = com.android.internal.R.id.content;  就是显示内容的id。  
  
  
    2). Window类的内部类接口Callback包含一系列dispatchXxxx函数和一系列onXxxx函数，用于处理UI事件  
  
        public boolean dispatchKeyEvent(KeyEvent event);  
  
        public boolean dispatchKeyShortcutEvent(KeyEvent event);  
  
        public boolean dispatchTouchEvent(MotionEvent event);  
  
        public boolean dispatchTrackballEvent(MotionEvent event);  
  
        public boolean dispatchGenericMotionEvent(MotionEvent event);                            移动事件  
  
        public boolean dispatchPopulateAccessibilityEvent(AccessibilityEvent event);  
  
        public View onCreatePanelView(int featureId);  
  
        public boolean onCreatePanelMenu(int featureId, Menu menu);  
  
        public boolean onPreparePanel(int featureId, View view, Menu menu);  
  
        public boolean onMenuOpened(int featureId, Menu menu);                                      用户选择菜单项。  
  
        public boolean onMenuItemSelected(int featureId, MenuItem item);                          用户选择菜单项。  
  
        public void onWindowAttributesChanged(WindowManager.LayoutParams attrs);        当前窗口属性改变。  
  
        public void onContentChanged();                                  窗口的view发生改变。  
  
        public void onWindowFocusChanged(boolean hasFocus);窗口的焦点发生变换  
  
        public void onAttachedToWindow();                              窗口附加到窗口管理器上。  
  
        public void onDetachedFromWindow();                          窗口从窗口管理器上摘除。  
  
        public void onPanelClosed(int featureId, Menu menu);  
  
        public boolean onSearchRequested();                       当用户按下了search按钮。如果启动了搜索[界面](https://www.baidu.com/s?wd=%E7%95%8C%E9%9D%A2&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)则返回true，如果不能执行则返回false。  
  
        public ActionMode onWindowStartingActionMode(ActionMode.Callback callback);  
  
        public void onActionModeStarted(ActionMode mode);  
  
        public void onActionModeFinished(ActionMode mode);