

RemoteViews的作用是在其他进程显示并更新界面，为了更好地理解它的内部机制，我们先来看一下它的主要功能。首先最常用的构造方法 `public RemoteViews(String packageName, int layoutId)`，接受两个参数，当前应用的包名和待加载的布局文件。

## RemoteViews支持的View类型

- Layout FrameLayout、LinearLayout、RelativeLayout、GridLayout
- View AnalogClock、Button、Chronometer、ImageButton、ImageView、ProgressBar、TextView、ViewFlipper、ListView、GridView、StackView、AdapterViewFlipper、ViewStub

除了上述的View，其他所有的View类型及子View类型都不支持。

## RemoteViews的set方法

RemoteViews没有提供findViewById方法，因此无法直接访问里面的View元素，而必须通过RemoteViews所提供的一系列set方法完成。

方法名	作用
setTextViewText(int viewId, CharSequence text)	设置TextView的文本
setTextViewTextSize(int viewId, int units, float size)	设置TextView的字体大小
setTextColor(int viewId, int color)	设置TextView的字体颜色
setImageResource(int viewId, int srcId)	设置ImageView的图片
setInt(int viewId, String methodName, long value)	反射调用View对象的方法，参数类型为int
setLong(int viewId, String methodName, long value)	反射调用View对象的方法，参数类型为long
setBoolean(int viewId, String methodName, boolean value)	反射调用View对象的方法，参数类型为boolean
setOnClickListener(int viewId, PendingIntent pendingIntent)	为View添加单击事件，事件类型只能为PendingIntent

## RemoteViews的内部机制

由于RemoteViews主要用于通知栏和桌面小部件之中，这里就通过它们来分析RemoteViews的工作过程。通知栏和桌面小部件分别有NotificationManager和AppWidgetManager管理，而NotificationManager和AppWidgetManager通过Binder分别和SystemServer进程中的NotificationManagerService以及AppWidgetService进行通信。由此可见，通知栏和桌面小部件的布局文件实际上是在NotificationManagerService以及AppWidgetService中被加载的，而它们运行在系统的SystemServer进程中。

RemoteViews通过Binder传递到SystemServer进程，因为RemoteViews实现了Parcelable接口。系统会根据RemoteViews中的包名等信息去得到该应用的资源，然后通过LayoutInflater去加载RemoteViews中的布局文件。set方法对View的更新不是立刻执行的，而是等RemoteViews被加载以后才执行。

系统提供了一个Action的概念，Action代表一个View操作，同样实现了Parcelable接口。每调用一次set方法，RemoteViews会添加一个对应的Action，通过NotificationManager和AppWidgetManager来提交更新时，这些Action会被传输到远程进程。远程进程通过RemoteViews的apply方法进行View的更新操作。

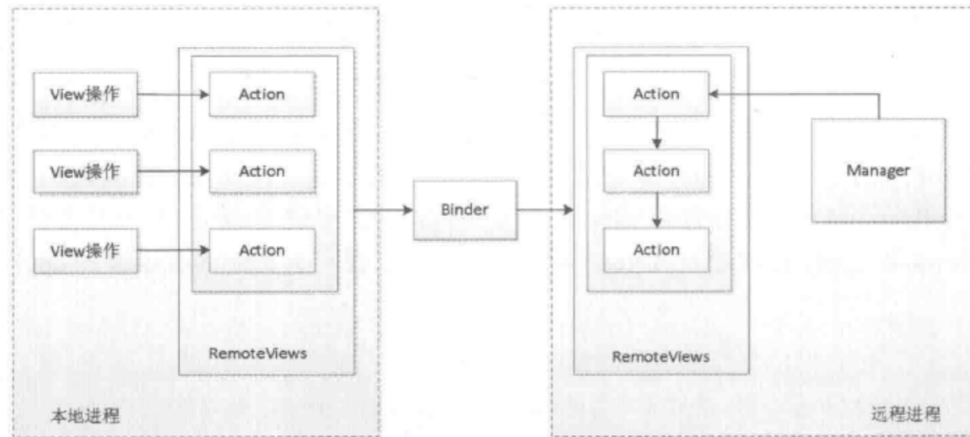


图 5-3 RemoteViews 的内部机制

这样避免了定义大量的Binder接口，其次通过在远程进行批量执行RemoteViews的修改，避免了大量的IPC操作。

## setOnClickPendingIntent、setPendingIntentTemplate和setOnClickFillInIntent

关于单击事件，RemoteViews只支持发起PendingIntent,不支持onClickListener。

setOnClickPendingIntent用于给普通View设置单击事件，但不能给集合（ListView和StackView）中的View设置单击事件，比如我们不能给ListView中的item通过setOnClickPendingIntent这种方式添加单击事件，因为开销比较大。

如果给ListView和StackView中的item添加单击事件，则必现将setPendingIntentTemplate和setOnClickFillInIntent组合使用。

## 模拟通知栏实现跨进程的UI更新