



概览屏幕

本文内容

- [将任务添加到概览屏幕](#)
 - [使用 Intent 标志添加任务](#)
 - [使用 Activity 属性添加任务](#)
- [移除任务](#)
 - [使用 AppTask 类移除任务](#)
 - [保留已完成的任务](#)

关键类

- [ActivityManager.AppTask](#)
- [Intent](#)

示例代码

- [以文档为中心的应用](#)

概览屏幕（也称为最新动态屏幕、最近任务列表或最近使用的应用）是一个系统级别 UI，其中列出了最近访问过的 [Activity](#) 和 [任务](#)。用户可以浏览该列表并选择要恢复的任务，也可以通过滑动清除任务将其从列表中移除。对于 Android 5.0 版本（API 级别 21），包含不同文档的同一 Activity 的多个实例可能会以任务的形式显示在概览屏幕中。例如，Google Drive 可能对多个 Google 文档中的每个文档均执行一个任务。每个文档均以任务的形式显示在概览屏幕中。

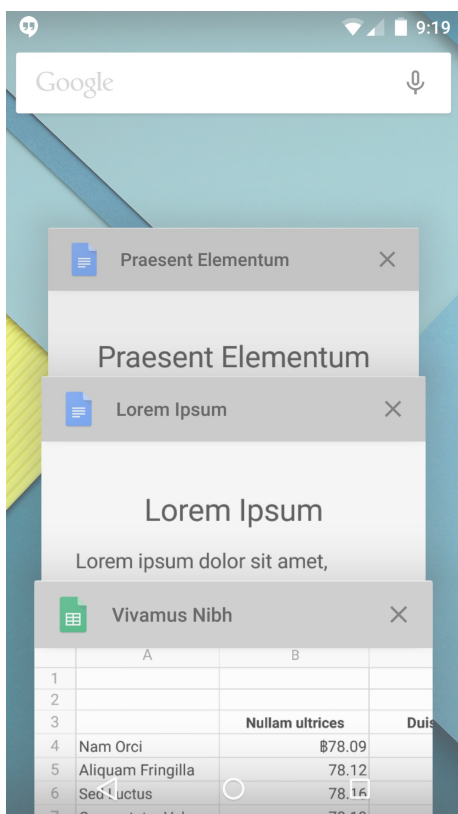


图 1. 显示了三个 Google Drive 文档的概览屏幕，每个文档分别以一个单独的任务表示。

通常，您应该允许系统定义任务和 Activity 在概览屏幕中的显示方法，并且无需修改此行为。不过，应用可以确定 Activity 在概览屏幕中的显示方式和时间。您可以使用 [ActivityManager.AppTask](#) 类来管理任务，使用 [Intent](#) 类的 Activity 标志来指定某 Activity 添加到概览屏幕或从

中移除的时间。此外，您也可以使用 `<activity>` 属性在清单文件中设置该行为。

将任务添加到概览屏幕

通过使用 `Intent` 类的标志添加任务，您可以更好地控制某文档在概览屏幕中打开或重新打开的时间和方式。使用 `<activity>` 属性时，您可以选择始终在新任务中打开文档，或选择对文档重复使用现有任务。

使用 Intent 标志添加任务

为 Activity 创建新文档时，可调用 `ActivityManager.AppTask` 类的 `startActivity()` 方法，以向其传递启动 Activity 的 `Intent`。要插入逻辑换行符以便系统将 Activity 视为新任务显示在概览屏幕中，可在启动 Activity 的 `Intent` 的 `addFlags()` 方法中传递 `FLAG_ACTIVITY_NEW_DOCUMENT` 标志。

注：`FLAG_ACTIVITY_NEW_DOCUMENT` 标志取代了 `FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET` 标志，后者自 Android 5.0（API 级别 21）起已弃用。

如果在创建新文档时设置 `FLAG_ACTIVITY_MULTIPLE_TASK` 标志，则系统始终会以目标 Activity 作为根创建新任务。此设置允许同一文档在多个任务中打开。以下代码演示了主 Activity 如何执行此操作：

`DocumentCentricActivity.java`

```
public void createNewDocument(View view) {
    final Intent newDocumentIntent = newDocumentIntent();
    if (useMultipleTasks) {
        newDocumentIntent.addFlags(Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
    }
    startActivity(newDocumentIntent);
}

private Intent newDocumentIntent() {
    boolean useMultipleTasks = mCheckbox.isChecked();
    final Intent newDocumentIntent = new Intent(this, NewDocumentActivity.class);
    newDocumentIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_DOCUMENT);
    newDocumentIntent.putExtra(KEY_EXTRA_NEW_DOCUMENT_COUNTER, incrementAndGet());
    return newDocumentIntent;
}

private static int incrementAndGet() {
    Log.d(TAG, "incrementAndGet(): " + mDocumentCounter);
    return mDocumentCounter++;
}
}
```

注：使用 `FLAG_ACTIVITY_NEW_DOCUMENT` 标志启动的 Activity 必须具有在清单文件中设置的 `android:launchMode="standard"` 属性值（默认）。

当主 Activity 启动新 Activity 时，系统会搜遍现有任务，看看是否有任务的 `Intent` 与 Activity 的 `Intent` 组件名称和 `Intent` 数据相匹配。如果未找到任务或者 `Intent` 包含 `FLAG_ACTIVITY_MULTIPLE_TASK` 标志，则会以该 Activity 作为其根创建新任务。如果找到的话，则会将该任务转到前台并将新 `Intent` 传递给 `onNewIntent()`。新 Activity 将获得 `Intent` 并在概览屏幕中创建新文档，如下例所示：

`NewDocumentActivity.java`

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_new_document);
    mDocumentCount = getIntent()
        .getIntExtra(DocumentCentricActivity.KEY_EXTRA_NEW_DOCUMENT_COUNTER, 0);
    mDocumentCounterTextView = (TextView) findViewById(
        R.id.hello_new_document_text_view);
    setDocumentCounterText(R.string.hello_new_document_counter);
}

@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    /* If FLAG_ACTIVITY_MULTIPLE_TASK has not been used, this activity
    is reused to create a new document.
    */
    setDocumentCounterText(R.string.reusing_document_counter);
}

```

使用 Activity 属性添加任务

此外，Activity 还可以在其清单文件中指定始终通过使用 `<activity>` 属性 `android:documentLaunchMode` 进入新任务。此属性有四个值，会在用户使用该应用打开文档时产生以下效果：

"intoExisting"

该 Activity 会对文档重复使用现有任务。这与不设置 `FLAG_ACTIVITY_MULTIPLE_TASK` 标志、但设置 `FLAG_ACTIVITY_NEW_DOCUMENT` 标志所产生的效果相同，如上文的[使用 Intent 标志添加任务](#)中所述。

"always"

该 Activity 为文档创建新任务，即便文档已打开也是如此。使用此值与同时设置 `FLAG_ACTIVITY_NEW_DOCUMENT` 和 `FLAG_ACTIVITY_MULTIPLE_TASK` 标志所产生的效果相同。

"none"

该 Activity 不会为文档创建新任务。概览屏幕将按其默认方式对待此 Activity：为应用显示单个任务，该任务将从用户上次调用的任意 Activity 开始继续执行。

"never"

该 Activity 不会为文档创建新任务。设置此值会替代 `FLAG_ACTIVITY_NEW_DOCUMENT` 和 `FLAG_ACTIVITY_MULTIPLE_TASK` 标志的行为（如果在 Intent 中设置了其中一个标志），并且概览屏幕将为应用显示单个任务，该任务将从用户上次调用的任意 Activity 开始继续执行。

注：对于除 `none` 和 `never` 以外的值，必须使用 `launchMode="standard"` 定义 Activity。如果未指定此属性，则使用 `documentLaunchMode="none"`。

移除任务

默认情况下，在 Activity 结束后，文档任务会从概览屏幕中自动移除。您可以使用 `ActivityManager.AppTask` 类、`Intent` 标志或 `<activity>` 属性替代此行为。

通过将 `<activity>` 属性 `android:excludeFromRecents` 设置为 `true`，您可以始终将任务从概览屏幕中完全排除。

您可以通过将 `<activity>` 属性 `android:maxRecents` 设置为整型值，设置应用能够包括在概览屏幕中的最大任务数。默认值为 16。达到最大任务数后，最近最少使用的任务将从概览屏幕中移除。`android:maxRecents` 的最大值为 50（内存不足的设备上为 25）；小于 1 的值无效。

使用 AppTask 类移除任务

在于概览屏幕创建新任务的 Activity 中，您可以通过调用 `finishAndRemoveTask()` 方法指定何时移除该任务以及结束所有与之相关的 Activity。

`NewDocumentActivity.java`

```
public void onRemoveFromRecents(View view) {
    // The document is no longer needed; remove its task.
    finishAndRemoveTask();
}
```

注：如下所述，使用 `finishAndRemoveTask()` 方法代替使用 `FLAG_ACTIVITY_RETAIN_IN_RECENTS` 标记。

保留已完成的任务

若要将任务保留在概览屏幕中（即使其 Activity 已完成），可在启动 Activity 的 Intent 的 `addFlags()` 方法中传递 `FLAG_ACTIVITY_RETAIN_IN_RECENTS` 标志。

`DocumentCentricActivity.java`

```
private Intent newDocumentIntent() {
    final Intent newDocumentIntent = new Intent(this, NewDocumentActivity.class);
    newDocumentIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_DOCUMENT |
        android.content.Intent.FLAG_ACTIVITY_RETAIN_IN_RECENTS);
    newDocumentIntent.putExtra(KEY_EXTRA_NEW_DOCUMENT_COUNTER, incrementAndGet());
    return newDocumentIntent;
}
```

要达到同样的效果，请将 `<activity>` 属性 `android:autoRemoveFromRecents` 设置为 `false`。文档 Activity 的默认值为 `true`，常规 Activity 的默认值为 `false`。如前所述，使用此属性替代 `FLAG_ACTIVITY_RETAIN_IN_RECENTS` 标志。