Distributing to Specific Screens

Quickview

- If necessary, you can control distribution of your application based on the device screen configuration

In this document

See also

Supporting Multiple Screens
Optimizing Apps for Android 3.0

Although we recommend that you design your application to function properly on multiple configurations of screen size and density, you can instead choose to limit the distribution of your application to certain types of screens, such as only tablets and other large devices or only handsets and similar-sized devices. To do so, you can enable filtering by external services such as Android Market by adding elements to your manifest file that specify the screen configurations your application supports.

However, before you decide to restrict your application to certain screen configurations, you should understand the techniques for supporting multiple screens and implement them to the best of your ability. By supporting multiple screens, your application can be made available to the greatest number of users with different devices, using a single `.apk`.

Filtering a Handset Application from Tablets

Because the system generally scales applications to fit larger screens well, you shouldn't need to filter your application from larger screens. As long as you follow the Best Practices for Screen Independence, your application should work well on larger screens such as tablets. However, you might discover that your application can't scale up well or perhaps you've decided to publish two versions of your application for different screen configurations. In such a case, you can use the `<compatible-screens>` element to manage the distribution of your application based on combinations of screen size and density. External services such as Android Market use this information to apply filtering to your application, so that only devices that have a screen configuration with which you declare compatibility can download your application.

The `<compatible-screens>` element must contain one or more `<screen>` elements. Each `<screen>` element specifies a screen configuration with which your application is compatible, using both the `android:screenSize` and `android:screenDensity` attributes. Each `<screen>` element **must include both attributes** to specify an individual screen configuration—if either attribute is missing, then the element is invalid (external services such as Android Market will ignore it).

For example, if your application is compatible with only small and normal size screens, regardless of screen density, you must specify eight different `<screen>` elements, because each screen size has four density configurations. You must declare each one of these; any combination of size and density that you do *not* specify is considered a screen configuration with which your application is *not* compatible. Here's what the manifest entry looks like if your application is compatible with only small and normal screen sizes:

```
<manifest ... >
    ...
    <compatible-screens>
        <!-- all small size screens -->
        <screen android:screenSize="small" android:screenDensity="ldpi" />
        <screen android:screenSize="small" android:screenDensity="mdpi" />
        <screen android:screenSize="small" android:screenDensity="hdpi" />
        <screen android:screenSize="small" android:screenDensity="xhdpi" />
        <!-- all normal size screens -->
        <screen android:screenSize="normal" android:screenDensity="ldpi" />
        <screen android:screenSize="normal" android:screenDensity="mdpi" />
        <screen android:screenSize="normal" android:screenDensity="hdpi" />
        <screen android:screenSize="normal" android:screenDensity="xhdpi" />
    </compatible-screens>
    <application ... >
        ...
    <application>
</manifest>
```

> **Note:** Although you can also use the `<compatible-screens>` element for the reverse scenario (when your application is not compatible with smaller screens), it's easier if you instead use the `<supports-screens>` as discussed in the next section, because it doesn't require you to specify each screen density your application supports.

Filtering a Tablet Application from Handsets

If your application's UI is adversely affected when the system scales your application down to smaller screens, you should add alternative layouts for smaller screens to adjust the layout for those screens. However, sometimes your layout still might not fit a smaller screen or you've explicitly designed your application only for tablets and other large devices. In this case, you can manage the availability of your application to smaller screens by using the `<supports-screens>` manifest element.

For example, if you want your application to be available only to large and extra large screens, you can declare the element in your manifest like this:

```
<manifest ... >
    ...
    <supports-screens android:smallScreens="false"
                      android:normalScreens="false"
                      android:largeScreens="true"
                      android:xlargeScreens="true" />
    <application ... >
        ...
    <application>
</manifest>
```

External services such as Android Market read this manifest element and use it to ensure that your application is available only to devices with either a large or an extra large screen.

> **Caution:** If you use the `<supports-screens>` element for the reverse scenario (when your application is not compatible with *larger* screens) and set the larger screen size attributes to `"false"`, then external services such as Android Market **do not** apply filtering. Your application will still be available to larger screens, but when it runs, it will not resize to fit the screen. Instead, the system will draw it in a "postage stamp" window that's the same relative size as the screen size that your application does support (see compatibility mode for more information). If you want to prevent your application from being downloaded on larger screens, use `<compatible-screens>`, as discussed in the previous section about Filtering a Handset Application from Tablets.

Remember, you should strive to make your application available to as many devices as possible by applying all necessary techniques for supporting multiple screens. You should use `<compatible-screens>` or `<supports-screens>` only when you cannot provide compatibility on all screen configurations or you have decided to provide different versions of your application for different sets of screen configurations.

← Back to Supporting Multiple Screens

↑ Go to top

Android 3.1 r1 - 17 Jun 2011 10:58