

Porting Asynchronous Background Tasks from Android Honeycomb to Windows Phone Mango - Fundamental Differences

This article explains the high level differences of asynchronous background tasks in Windows Phone 7.5 (Mango) and Android Honeycomb.

Introduction



There has been quite a bit of discussion on several blogs and discussion forums on how Android and Windows Phone deal with background tasks. The platforms have APIs for managing background jobs. These APIs are not that different.

Different APIs for different purposes

The two APIs are designed for different purpose - the Android API serves both app developers and platform developers while the Windows Phone API is for application developers only:

- Android developers are not only creating new applications but are also implementing new features for the platform, either using the Android flavor of the Java programming tools or a combination of Java and native code.
- As the Windows Phone audience is not expected to implement new features for the platform itself the API can be made easier to use, and the cost of being less flexible.

As the use cases for the two APIs are different, it is reasonable that the APIs look different. Android provides a more abstract solution to the problem; the APIs are more generic than the Microsoft ones. Microsoft provides more off-the-shelf oriented solution to manage the key use cases for background processing.

API usability

Windows Phone Mango APIs are designed to be as easy to use as possible. The platform makes every effort that you would not write bad code that would hog all system resources. Mango background tasks run once in every 30 minutes. So called resource intensive background jobs only run if some predefined conditions, such as the availability of a high-speed network, are met. Application developers can only do what is allowed by the framework. This is the big picture.

The Microsoft approach has some advantages. Restricting applications to only run their background tasks once in every 30 minutes leaves the rest of the system with more execution time. Also, application developers are not given an option to make the services accidentally run at a too high priority with no particular reason. By designing the different APIs for different purposes, such as playing audio, makes it easy for developers to use them; generic APIs are typically more complicated to use as there is more configuration to be done – different use cases, different parameters, several overloads and so on. On the other hand there is already a great deal of easy-to-use APIs for Android that do the same things as their Mango counterparts.

Application developer's perspective - more or less resources in use depending on the platform

The APIs to create background services for Android and Windows Phone Mango are functionally very similar. It is often a question of playing audio, notifying the user, setting an alarm and so forth. The key difference in implementing background tasks for the two platforms is the degree of freedom in defining how much execution time and other resources you can possible use.

In case of Android, the developer can create a background task with so called foreground priority. This means that the system will not kill the task or at least tries to keep the task alive as long as possible.

More information on the fundamental differences

You can make such an Android service run in the background, but keep in mind that your threads are scheduled at a relatively high priority and the resources you are using are considered high priority ones, too. Your task is not sacrificed first if the system is running out of resources. Your task should be a jolly important one. With great power comes great responsibility. For Android you use the same APIs to write platform code and application code, though there are some APIs which are for internal use only. Dianne Hackborn from Google has been constantly answering somewhat tough background task related questions. A very good overview document of Android multitasking can be found at <http://android-developers.blogspot.com/2010/04/multitasking-android-way.html>. The document leaves some questions, such as thread priority issues unanswered, but is definitely a good starting

point.

For Windows Phone Mango you are restricted to use the dedicated background APIs to manage certain use cases such as audio playback with heavy restrictions on resource usage. The Mango approach is somewhat safer when it comes to the thousands of applications that users can download from the different marketplaces. Microsoft is making a great effort to make it impossible for an application developer to write an application that eats up all resources. You can find more information on implementing background tasks for Windows Phone at <http://www.microsoft.com/download/en/details.aspx?id=27411>.

Summary

It will be interesting to see where we will be about a year from now - will the APIs still be the same for both platforms and what kind of porting issues we are likely to bump into when we port code from Android to Windows Phone that is supposed to run in the "background".