Processes and Application Lifecycle

It is important that application developers understand how different application components (in particular Activity, Service, and BroadcastReceiver) impact the lifetime of the application's process. Not using these components correctly can result in the system killing the application's process while it is doing important work.

Processes and Application Lifecycle

To determine which processes should be killed when low on memory, Android places each process into an "importance hierarchy" based on the components running in them and the state of those components. These process types are (in order of importance):

- A foreground process
- A visible process
- A service process
- A cached process

Processes and Application Lifecycle

A foreground process

- It is running an Activity at the top of the screen that the user is interacting with (its onResume() method has been called).
- It has a BroadcastReceiver that is currently running (its BroadcastReceiver.onReceive() method is executing).
- It has a Service that is currently executing code in one of its callbacks (Service.onCreate(), Service.onStart(), or Service.onDestroy()).

Processes and Application Lifecycle

A visible process

- It is running an Activity that is visible to the user on-screen but not in the foreground (its onPause() method has been called). This may occur, for example, if the foreground Activity is displayed as a dialog that allows the previous Activity to be seen behind it.
- It has a Service that is running as a foreground service, through Service.startForeground() (which is asking the system to treat the service as something the user is aware of, or essentially visible to them).
- It is hosting a service that the system is using for a particular feature that the user is aware, such as a live wallpaper, input method service, etc.

Processes and Application Lifecycle

A service process:

A service process is one holding a Service that has been started with the startService() method. Though these processes are not directly visible to the user, they are generally doing things that the user cares about (such as background network data upload or download), so the system will always keep such processes running unless there is not enough memory to retain all foreground and visible processes.

Processes and Application Lifecycle

A cached process:

A cached process is one that is not currently needed, so the system is free to kill it as desired when memory is needed elsewhere. In a normally behaving system, these are the only processes involved in memory management: a well running system will have multiple cached processes always available (for more efficient switching between applications) and regularly kill the oldest ones as needed. Only in very critical (and undesireable) situations will the system get to a point where all cached processes are killed and it must start killing service processes.