Find out the running process ID by package name

I am working on a script in which I need to supply the PID of my application. I am able to list all the processes with their PIDs by following command and could see the entry of my application.

> adb shell ps

This gives me a huge list of processes. And I need a single entry (which I can further supply to another command), so I want to filter this results with a package name. The grep command does not work on my windows machine. Also tried following command but it didn't help.

> adb shell ps name:my_app_package

android   adb

edited Nov 17 '16 at 17:25
Alex P.
15.5k 10 49 87

asked Mar 25 '13 at 6:25
Mukesh Bhojwani
809 2 9 21

3 Answers

Since Android 7.0 the easiest way to find out the process ID by package name is to use `pidof` command:

```
usage: pidof [-s] [-o omitpid[,omitpid...]] [NAME]...

Print the PIDs of all processes with the given names.

-s      single shot, only return one pid.
-o      omit PID(s)
```

Just run it like this:

```
adb shell pidof my.app.package
```

In Android before 7.0 people used `ps` command and then parsed its output using either built-in filter by `comm` value (which for android apps is the last 15 characters of the package name) or `grep` command. The `comm` filter did not work if the last 15 characters of the name started with a digit and the `grep` was not included by default until Android 4.2. But even after the proper process line was found the `PID` value still needed to be extracted.

There were multiple ways to do that. Here is how finding the process and extracting PID could be done with a single `sed` command:

```
adb shell "ps | sed -n 's/^[^ ]* *\([0-9]*\).* my\.app\.package$/\1/p'"
```

Again the problem is that `sed` was was not included by default until Android 6.0.

But if you must use an older device you can always use the following Android version independent solution. It does not use any external commands - just Android shell built-ins: