**Android PackageManagerService**

**Android Permission Model**

date:2018/06/11

Context

ContextImpl

ContextWrapper

Service

ContextThemeWrapper

Activity

frameworks/base/core/java/android/app/ContextImpl.java

|  |
| --- |
| public PackageManager getPackageManager() { |
|  | if (mPackageManager != null) { |
|  | return mPackageManager; |
|  | } |
|  |  |
|  | IPackageManager pm = ActivityThread.getPackageManager(); |
|  | if (pm != null) { |
|  | // Doesn't matter if we make more than one instance. |
|  | return (mPackageManager = new ApplicationPackageManager(this, pm)); |
|  | } |
|  |  |
|  | return null; |
|  | } |

public abstract class PackageManager

{

}

class ContextImpl extends Context{

private PackageManager mPackageManager;

}

final class ApplicationPackageManager extends PackageManager{

private final IPackageManager mPM;

public List<PackageInfo> getInstalledPackages(int flags){

mPM.getInstalledPackages(flags,lastKey)

}

}

ActivityThread.getPackageManager();

public final class ActivityThread{

static volatile IPackageManager sPackageManager;

public static IPackageManager getPackageManager(){

if (sPackageManager != null) return sPackageManager;

IBinder b = ServiceManager.getService(“package”)

sPackageManager = IPackageManager.Stub.asInterface(b);

}

}

public interface IPackageManager extends android.os.IInterface{

public static abstract class Stub extends android.os.Binder implements IPackageManager{

public static IPackageManager asInterface(android.os.IBinder){

}

}

}

PackageManagerService.java

frameworks/base/services/java

package com.android.server.pm;

public class PackageManagerService extends IPackageManager.Stub{

}

<http://www.520monkey.com/archives/856>

# Android Permission Model

1 mContext.getPackageManager().getPackageGids(app.info.packageName);

2

3

权限记录在什么地方？

安装权限

/data/system/packages.xml

/data/system/appops.xml

/data/system/users/0/runtime-permissions.xml

源码分析

权限定义

frameworks/base/core/res/AndroidManifest.xml

frameworks/base/core/res/AndroidManifest.xml

Android 6.0.1

frameworks/base/core/java/android/app/AppOpsManager.java

public class AppOpsManager{

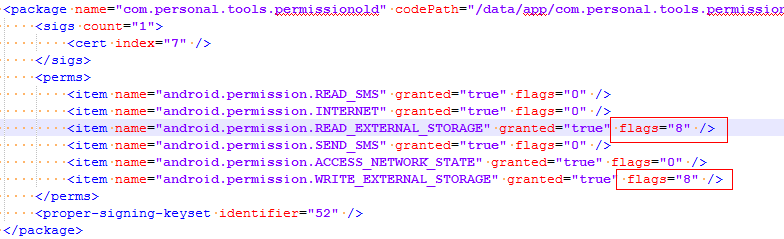
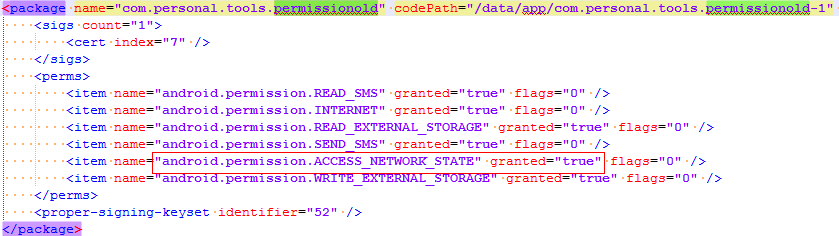
public static final int OP\_READ\_EXTERNAL\_STORAGE = 59;

public static final int OP\_WRITE\_EXTERNAL\_STOREAGE = 60;

}

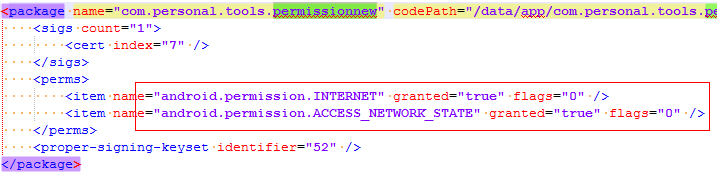
targetSdkVersion < 23

所有权限都放在packages.xml里

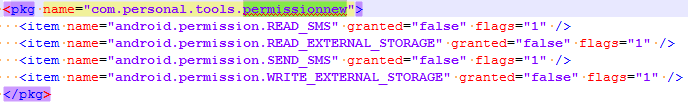


targetSdkVersion >= 23

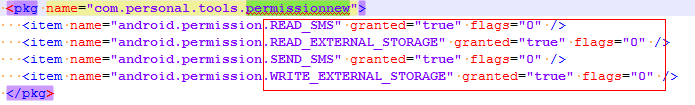
安装权限



运行时权限



运行时权限



## Application API

checkSelfPermission

shouldShowRequestPermissionRationale

requestPermissions

onRequestPermissionsResult

TargetSdkVersion >= 23

If your application is targeting an API level before 23 (Android M) then both:ContextCompat#checkSelfPermission and Context#checkSelfPermission doesn't work and always returns 0 (PERMISSION\_GRANTED). Even if you run the application on Android 6.0 (API 23).

It's not fully true that if you targeting an API level before 23 then you don't have to take care of permissions. If you targeting an API level before 23 then:

Android < 6.0: Everything will be ok.

Android 6.0: Application's run-time permissions will be granted by default (compatibility mode applies), but the user can change run-time permissions in Android Settings, then you may have a problem.

As I said in the 1st point, if you targeting an API level before 23 on Android 6.0 then ContextCompat#checkSelfPermission and Context#checkSelfPermission doesn't work. Fortunately you can use PermissionChecker#checkSelfPermission to check run-time permissions.

AppOps

/data/system/appops.xml

AppOpsManager

AppOpsSummary

AppOpsCategory

AppOpsService

mAppOpsService = new AppOpsService(new File(systemDir,”appops.xml”),mHandler)

Runtime Permission

app安装时权限默认是disable的

# PMS

PMS Source Code

<http://androidxref.com/6.0.1_r10/xref/frameworks/base/services/core/java/com/android/server/pm/PackageManagerService.java>

getPackageManager().getPackageGids(app.info.packageName)

PackageManager.java

<http://androidxref.com/6.0.1_r10/xref/frameworks/base/core/java/android/content/pm/PackageManager.java>

public abstract class PackageManager{

public static final int FLAG\_PERMISSION\_USER\_SET = 1 << 0;

public static final int FLAG\_PERMISSION\_USER\_FIXED = 1 << 1;

public static final int FLAG\_PERMISSION\_POLICY\_FIXED = 1 << 2;

public static final int FLAG\_PERMISSION\_REVOKE\_ON\_UPGRADE = 1 << 3;

public static final int FLAG\_PERMISSION\_SYSTEM\_FIXED = 1 << 4;

public static final int FLAG\_PERMISSION\_GRANTED\_BY\_DEFAULT = 1 << 5;

public static final int MASK\_PERMISSION\_FLAGS = 0xFF;

}

# PackageInstaller

<http://androidxref.com/6.0.1_r10/xref/packages/apps/PackageInstaller>

Permission.java

<http://androidxref.com/6.0.1_r10/xref/packages/apps/PackageInstaller/src/com/android/packageinstaller/permission/model/Permission.java>

public final class Permission{

private final String mName;

private final int mAppOp;

private Boolean mGranted;

private Boolean mAppOpAllowed;

private int mFlags

}

Public final class AppPermissionGroup implements Comparable<AppPermissionGroup>{

Private final PackageInfo mPackageInfo;

Private final String mName;

Private final ArrayMap<String,Permission> mPermissions = new ArrayMap<>();

}

// y

Public final class AppPermissions{

Private final ArrayList<AppPermissionGroup> mGroups = new ArrayList<>();

Private final LinkedHashMap<String,AppPermissionGroup> mNameToGroupMap =

New LinkedHashMap<>();

}