# [FAQ03426]当系统存在多个Launcher时，如何设置开机自动进入默认的Launcher？

[DESCRIPTION]
开机完成并解锁后，如果系统存在多个Launcher，系统会弹出一个选择框让用户选择进入某个Launcher。如果用户不想选择，而是想直接进入某一个默认的Launcher，要怎么修改？
[SOLUTION]

注意：此修改方法对Google Now Launcher(GoogleHome.apk)无效，因为此APP启动Activity的<intent-filter>中无 <category android:name="android.intent.category.HOME" />这个选项。

下面以设置Launcher2作为默认启动的Launcher示例：
其package name 为 com.android.launcher
activity name 为 com.android.launcher2.Launcher

1.请找到PackageManagerService.java的systemReady方法，在这个方法的最后增加以下示例代码：

```
if(isFirstBoot()){
//MTK ADD BEGIN
String examplePackageName = "com.android.launcher"; //请修改为需要设置的 package name
String exampleActivityName = "com.android.launcher2.Launcher"; //请修改为需要设置的 launcher activity name
Intent intent=new Intent(Intent.ACTION_MAIN);
intent.addCategory(Intent.CATEGORY_HOME);
final int callingUserId = UserHandle.getCallingUserId();

List<ResolveInfo> resolveInfoList = queryIntentActivities(intent,null, PackageManager.GET_META_DATA,callingUserId);
if(resolveInfoList != null){
int size = resolveInfoList.size();
for(int j=0;j<size;){
final ResolveInfo r = resolveInfoList.get(j);
if(!r.activityInfo.packageName.equals(examplePackageName)){
resolveInfoList.remove(j);
size -= 1;
}else
{
j++;
}
}
ComponentName[] set = new ComponentName[size];
ComponentName defaultLauncher=new ComponentName(examplePackageName, exampleActivityName);
int defaultMatch=0;
for(int i=0;i<size;i++){
```

```
final ResolveInfo resolveInfo = resolveInfoList.get(i);
Log.d(TAG, resolveInfo.toString());
set[i] = new
ComponentName(resolveInfo.activityInfo.packageName,resolveInfo.activityInfo.name);
if(defaultLauncher.getClassName().equals(resolveInfo.activityInfo.name)){
defaultMatch = resolveInfo.match;
}
}
Slog.e(TAG,"defaultMatch="+Integer.toHexString(defaultMatch));
IntentFilter filter=new IntentFilter();
filter.addAction(Intent.ACTION_MAIN);
filter.addCategory(Intent.CATEGORY_HOME);
filter.addCategory(Intent.CATEGORY_DEFAULT);

addPreferredActivity2(filter, defaultMatch, set, defaultLauncher);
}
//MTK ADD END
}
```

2. 在PackageManagerService.java中增加addPreferredActivity2

```
public void addPreferredActivity2(IntentFilter filter, int match,ComponentName[] set, ComponentName activity) {
synchronized (mPackages) {
filter.dump(new LogPrinter(Log.INFO, TAG), " ");
mSettings.editPreferredActivitiesLPw(0).addFilter(new PreferredActivity(filter, match, set, activity, true));
scheduleWriteSettingsLocked();
}
}
```

3. 请修改PackageManagerService.java的findPreferredActivity方法，将以下代码：

```
if (removeMatches) {
pir.removeFilter(pa);
if (DEBUG_PREFERRED) {
Slog.v(TAG, "Removing match " + pa.mPref.mComponent);
}
break;
}
// Okay we found a previously set preferred or last chosen app.
// If the result set is different from when this
// was created, we need to clear it and re-ask the
// user their preference, if we're looking for an "always" type entry.
if (always && !pa.mPref.sameSet(query, priority)) {
Slog.i(TAG, "Result set changed, dropping preferred activity for "
+ intent + " type " + resolvedType);
if (DEBUG_PREFERRED) {
Slog.v(TAG, "Removing preferred activity since set changed "
+ pa.mPref.mComponent);
}
pir.removeFilter(pa);
// Re-add the filter as a "last chosen" entry (!always)
PreferredActivity lastChosen = new PreferredActivity(
pa, pa.mPref.mMatch, null, pa.mPref.mComponent, false);
pir.addFilter(lastChosen);
mSettings.writePackageRestrictionsLPr(userId);
return null;
}
```

修改为：

```
if(!(intent.getAction() != null && intent.getAction().equals(intent.ACTION_MAIN) &&
intent.getCategories()!=null &&
intent.getCategories().contains(intent.CATEGORY_HOME))){ //MTK ADD
Slog.d(TAG,"launcher");
}else {
if (removeMatches) {
pir.removeFilter(pa);
if (DEBUG_PREFERRED) {
Slog.v(TAG, "Removing match " + pa.mPref.mComponent);
}
break;
}
}
// Okay we found a previously set preferred or last chosen app.
// If the result set is different from when this
// was created, we need to clear it and re-ask the
// user their preference, if we're looking for an "always" type entry.
if (always && !pa.mPref.sameSet(query, priority)) {
if(!(intent.getAction() != null && intent.getAction().equals(intent.ACTION_MAIN) &&
intent.getCategories()!=null &&
intent.getCategories().contains(intent.CATEGORY_HOME))){ //MTK ADD
Slog.i(TAG, "Result set changed, dropping preferred activity for "
+ intent + " type " + resolvedType);
if (DEBUG_PREFERRED) {
Slog.v(TAG, "Removing preferred activity since set changed "
+ pa.mPref.mComponent);
}
pir.removeFilter(pa);
// Re-add the filter as a "last chosen" entry (!always)
PreferredActivity lastChosen = new PreferredActivity(
pa, pa.mPref.mMatch, null, pa.mPref.mComponent, false);
pir.addFilter(lastChosen);
mSettings.writePackageRestrictionsLPr(userId);
return null;
}
}
```

特别说明：

JB系列findPreferredActivity代码略有不同，如下：

修改 packageManagerService.java (framework/base/services/java/com/android/server/pm/)的 findPreferredActivity方法，将此方法比较靠后的一段代码：

```
if (!pa.mPref.sameSet(query, priority)) {
Slog.i(TAG, "Result set changed, dropping preferred activity for " +
intent + " type " + resolvedType);
pir.removeFilter(pa);
return null;
}
```

修改为：

```
if (!pa.mPref.sameSet(query, priority)) {
if(!(intent.getAction() != null &&
intent.getAction().equals(intent.ACTION_MAIN) &&
intent.getCategories()!=null&&
intent.getCategories().contains(intent.CATEGORY_HOME))){//MTKADD
Slog.i(TAG, "Result set changed, dropping preferred activity for "+ intent
+ " type " + resolvedType);
pir.removeFilter(pa);
return null;
}//MTK ADD
Slog.i(TAG,"Result set not change, not drop preferred activity for special
intent: "+intent);//MTK ADD
}
```

PS：

如果用户在设置->应用程序->管理应用程序->所有应用程序列表，找到之前设置的默认Launcher，并取消了默认值，系统会重新弹出launcher选择框。