总体

初始化

弹出协议窗

处理跳转

WelcomeScreen

1. 设置FLAG\_INIT = true;SPLASH=Handler.attach() handler获取主线程的looper并循环获取主线程的消息，然后通过needShowAnimation()方法中的是否第一次启动或者是否更新了新版本来判断是否需要显示动画，

若不需要执行3步

若需要执行2步

1. 调用显示开机动画的方法prepareSplashAnimation(…)：

* 首先加载layout\_splash.xml配置文件以及绑定控件，
* 配置sloganIv控件的动画监听器监听动画结束事件，（实际调用的是ObjectAnimation的animationEnd()方法，利用setAnimationListener将当前的监听器与RepeatAnimImageView的AnimListener接口对象进行绑定，实现对动画结束的监听）。
* 执行完第一个动画sloganIv，会在结束监听中启动lineIv（postDelayed方法调用会延迟200ms之后在调用listern中的方法）的动画显示，当lineIv动画结束时执行传进来的监听器对象listern，调用listern里面的onAnimEnd ()方法，执行3步；

1. 利用checkRoWaitMultiDexInstall()方法：

检测Build的SDK版本，若不符合条件（sdk版本、DexFirst等，MutilDex是为适配低版本sdk的dexOpt工具时的局限性，检索类的id个数有上限，所以利用multidex工具需要将dex拆分成为多个dex）则安装mutildex，MulTI.install(BDMultiDexApplication.this);

当安装成功通过UI线程执行4步;

若上述检测条件不符合直接执行4步;

1. checkJump()方法中的文件初始化：

* bFirstLaunch：GlobalConfig中得到bFirstLaunch的值（是否为第一次登录），
* noAsk：由getFlowHintState(this)获取，该方法首先构建 Preference，然后调用Preference父类CachePreference的getBoolean方法，传入(“isask”，false)两个参数，默认返回为false，该方法中通过查询SharePreference中是否有值，有的话返回true，并且将其存入内存缓存中；没有分会默认false。
* hasFlowHint是android层流量提示判断条件，通过CstmConfigUI预装配置类的isFlowHintVisable(this)方法获取，返回return context.getResources().getBoolean(R.bool. flowHintVisable)。

1. checkJump()方法中的执行方法：

* 利用bFirstLaunch判断是否第一次登录。
* bFirstLaunch判断为是，则通过VersionUtil类的isNewVersion方法返回值判断是否为新版本（利用GlobalConfig获取上一次启动版本和利用getAppCurVersion()得到当前版本，并返回当前是否大于上一次版本）。
* 若bFirstLaunch判断为是，isNewVersion判断为是，则进入startGuidePage(false)，然后通过传入判断执行赋值NewGuideScreen给intent ，执行第9步jump()方法。
* 若bFirstLaunch判断为是，isNewVersion判断为否，进入判断(hasFlowHint && !noAsk)，判断结果为是的话，启动ServiceHintThread线程，否的话执行checkPermissions()检测权限（检测权限是否存在，不存在的话添加权限），然后执行自身Runnable 中的startMap()方法。
* 若bFirstLaunch判断为否的话，执行创建快捷方式工具方法，启动ServiceHintThread线程。

1. ServiceHintThread启动：

* 在UI主线程中执行弹出条款程序ServiceTermsHint中的showFlowHint()方法提示。
* 子线程中设置点击确认按钮执行checkPermissions检测权限，启动自身Runnable，执行方法如下：首先利用GlobalConfig获取bFirstLaunch，并做判断，若是第一次启动，配置GlobalConfig 的值，说明已经不是第一次启动，然后执行startGuidePage(true),跳转startUserGuide()方法，赋值NewUserGuide 给intent，执行第9步jump()方法，若不是第一次启动的话，跳转startMap()；结束当前界面。
* 点击取消按钮，将false的信息存入缓存，下次启动还会提示，重置“不在提示”勾选状态。

1. startMap()方法，跳转至默认界面：

* 通过BaiduMapApplication.getInstance().initEngin();方法实现地图引擎的初始化，通过实例化mapGLSurfaceView，配置一些静态变量。
* 获取当前界面的绑定的intent，赋值MapsActivity给intent以及配置启动新界面的FLAG= FLAG\_ACTIVITY\_REORDER\_TO\_FRONT。
* 判断isForeground（该值只有在onStop里面赋值成false，其他时候皆为true），若是进入显示运行闪屏方法ShowAdvert()，否则直接执行第9步jump()方法。

1. 显示运营闪屏ShowAdvert():

* 配置显示文件welcome.xml，通过URL获取到图片并转成Bitmap；
* 通过判断获取的Bitmap是否为空，若为空则直接执行第9步jump()方法，否则的话检测SDK版本，然后根据sdk是否大于16来选择给View设置背景图片的方式。
* 点击图片进入指定URL，通过http串跳转运营活动，通过赋值Uri.parse(mUrlString)给intent，执行第9步jump()方法，点击跳过或者3s之后自动执行第9步jump()方法。

1. jump()方法：

* jump()方法中执行startActivity(intent)，intent是由跳转前配置的完成的。
* 通过判断isJumped的值来避免重复的跳转，在startActivity(intent)之后将isJumped的值设置为false。
* showAnim的值是针对闪屏广告的，在闪屏中赋值true，通过overridePendingTransition()方法，为Activity之间的跳转增加一统一个扩大淡出的动画效果。

主Activity任务：MapsActivity

1. onCreate()

* 判断FLAG\_INIT是否为true，为false重启WelcomeScreen.java
* registerRootTask(…)方法将当前类MapsActivity注册成页面栈TaskManagerImpl类中的根界面；setRootRecord()设置页面栈的根记录，页面栈关联MapsActivity，设置Historyrecord对象的三个属性：task=MapsActivity，page=MapFramaPage，taskSignature=task的16进制的hashcode值（task的标识）。
* 执行父类BaseTask 的create(…)方法，传入savedInstanceState，将当前Activity的状态记录值转换成Parcelable，通过TaskManagerImpl类的restoreState(…)方法全部替换该类的mHistoryList。
* mNavigator.setContainerActivity(this);利用父类的属性，设置承载其他Fragment的Activity，后面加载MapFramePage时用到。

1. CstmUtility.getInstance().init(JNIInitializer.getCachedContext());

初始化地图，构建Preference参数配置类，获取周边的icon、厂商搜索词条key、厂商搜索词条value显示在下面

1. 判断saveInstance的值是否为空，不为空的话跳转：

* TaskManagerFactory.getTaskManager().navigateTo(TaskManagerFactory.getTaskManager().getContext(),MapFramePage.class.getName())，单例模式确保只有一个TaskManagerImpl类实例对象，然后调用TaskManagerImpl的navigateTo方法，navigateTo方法首先获取当前任务，判断是否为null，若不为空的话获取当前任务中的TaskManagerImpl对象并重新调用navigateTo方法，否则的话配置Intent属性，构造方法传入context和根task（MapsActivity），配置信息，启动。

1. onAttachedToWindow()启动addUILayersJob，处理跳转：handleIntent()

* 通过获取intent的值获取Action不等于空且等于ACTION\_MAIN，交集求出的boolean的isFromHome值，作为判断该界面是来自百度地图软件。
* 若isFromHome是true，判断getBooleanExtra取得值是否来自new user，是的话跳转UserGuideFirstPage
* 若isFromHome是false，则调用父类BaseTask的navigateTo方法，跳转PageNavigator类的页面导航navigateTo方法，该方法首先通过BaseTask传入的componentId得到类加载器，然后根据getBasePageInstance得到要加载显示的MapFramePage界面的页面基类BasePage，设置该Page的Task，通过上面传进来的（MapsActivity）,然后检测任务栈中是否存在该Page，存在的话里有FragmentManager类方法beginTransaction显示该Page（MapFramePage）。
* 其他的情况说明是由第三方应用调起，执行：

EntryController entryController = new EntryController(this);

entryController.process(getIntent()); 实现。

新用户引导

1. NewUserGuide 任务页

* 首先注册页面栈和关联当前界面和TaskManagerImpl，create同MapsActivity
* mNavigator = new PageNavigator(this, (ViewGroup) findViewById(R.id.fragment\_container));，将控件绑定在PageNavigator上。
* TaskManagerImpl的navigateTo方法跳转展示UserGuideFirstPage

1. UserGuideFirstPage 展示页

* 利用xml文件配置ViewGroup，此处若crash，则直接跳转MapsActivity，然后Finish当前界面。
* 初始化ViewPage（设置适配器）和closeButton，设置监听器。
* ViewPager适配器：UserGuideFragmentPageAdapter，利用instantiateItem方法配置，container.addView(mViews[position]);添加四个引导页，前两个是图片轮播，第三个配置家和公司地址，点击保存并退出进入MapsActivity。闪屏操作与WelcomeScreen一样。