



(12)发明专利申请

(10)申请公布号 CN 112445513 A  
(43)申请公布日 2021.03.05

(21)申请号 201910808955.2

(22)申请日 2019.08.29

(71)申请人 天津五八到家科技有限公司  
地址 300000 天津市滨海新区梅苑路5号金座广场801

(72)发明人 马健

(74)专利代理机构 北京太合九思知识产权代理有限公司 11610  
代理人 曹威 刘戈

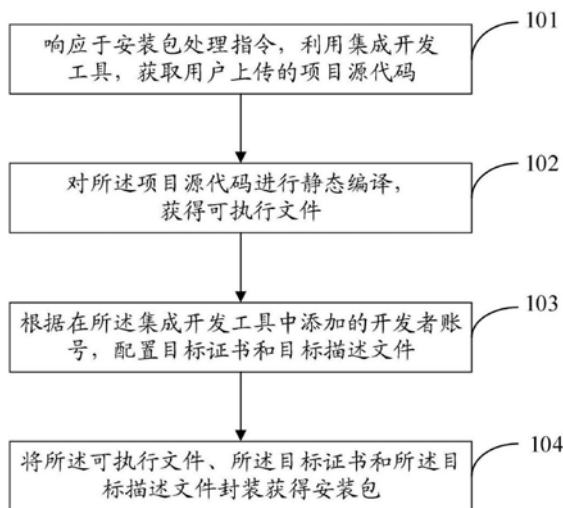
(51)Int.Cl.  
G06F 8/71(2018.01)  
G06F 8/61(2018.01)  
G06F 8/41(2018.01)

权利要求书2页 说明书9页 附图2页

(54)发明名称  
安装包处理方法及装置

(57)摘要

本申请实施例提供一种安装包处理方法及装置,涉及计算机应用技术领域。其中,响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;对所述项目源代码进行静态编译,获得可执行文件;根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。本申请实施例提供的技术方案提高了安装包生成效率。



1. 一种安装包处理方法,其特征在于,包括:  
响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;  
对所述项目源代码进行静态编译,获得可执行文件;  
根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;  
将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。
2. 根据权利要求1所述的方法,其特征在于,所述根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件包括:  
根据在所述集成开发工具中添加的开发者账号,创建应用标识符bundle ID;  
根据所述bundle ID,创建目标证书和目标描述文件。
3. 根据权利要求1所述的方法,其特征在于,还包括:  
上传所述安装包。
4. 根据权利要求3所述的方法,其特征在于,所述上传所述安装包包括:  
获取用户提供的上传路径;  
根据所述上传路径,上传所述安装包至所述上传路径对应的包管理平台,以便于从所述包管理平台下载和安装所述安装包。
5. 根据权利要求3所述的方法,其特征在于,所述上传所述安装包包括:  
利用所述开发者账号登录测试平台;  
上传所述安装包至测试平台,以供所述测试平台对所述安装包进行测试;  
若所述安装包测试通过,上传所述安装包至公共平台,以便于从所述公共平台下载和安装所述安装包。
6. 根据权利要求5所述的方法,其特征在于,所述上传所述安装包至测试平台包括:  
导出所述安装包并保存到本地;  
调用构建工具插件上传所述安装包至测试平台。
7. 根据权利要求1所述的方法,其特征在于,所述将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包包括:  
基于用户提供的目标功能配置需求,生成配置文件;  
将所述配置文件与所述可执行文件、目标证书、目标描述文件和运行需要的资源封装获得安装包;所述配置文件用于所述安装包运行时提供所述目标功能。
8. 一种安装包处理装置,其特征在于,包括:  
获取模块,用于响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;  
获得模块,用于对所述项目源代码进行静态编译,获得可执行文件;  
配置模块,用于根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;  
封装模块,用于将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。
9. 根据权利要求8所述的装置,其特征在于,所述配置模块包括:  
第一创建单元,用于根据在所述集成开发工具中添加的开发者账号,创建应用标识符bundle ID;

第二创建单元,用于根据所述bundle ID,创建目标证书和目标描述文件。

10. 根据权利要求8所述的装置,其特征在于,还包括:

上传模块,用于上传所述安装包。

11. 根据权利要求10所述的装置,其特征在于,所述上传模块包括:

获取单元,用于获取用户提供的上传路径;

第一上传单元,用于根据所述上传路径,上传所述安装包至所述上传路径对应的包管理平台,以便于从所述包管理平台下载和安装所述安装包。

12. 根据权利要求10所述的装置,其特征在于,所述上传模块包括:

登录单元,用于利用所述开发者账号登录测试平台;

第二上传单元,用于上传所述安装包至测试平台,以供所述测试平台对所述安装包进行测试;

第三上传单元,用于若所述安装包测试通过,上传所述安装包至公共平台,以便于从所述公共平台下载和安装所述安装包。

13. 根据权利要求12所述的装置,其特征在于,所述第二上传单元包括:

导出子单元,用于导出所述安装包并保存到本地;

上传子单元,用于调用构建工具插件上传所述安装包至测试平台。

14. 根据权利要求9所述的装置,其特征在于,所述封装模块包括:

生成单元,用于基于用户提供的目标功能配置需求,生成配置文件;

封装单元,用于将所述配置文件与所述可执行文件、目标证书、目标描述文件和运行需要的资源封装获得安装包;所述配置文件用于所述安装包运行时提供所述目标功能。

## 安装包处理方法及装置

### 技术领域

[0001] 本申请实施例涉及计算机应用技术领域,尤其涉及一种安装包处理方法及装置。

### 背景技术

[0002] 在APP(application,应用程序)的发布过程中,需要将包括可执行文件在内的相关文件封装打包获得安装包,再将安装包上传至公共平台以供用户下载安装。对安装包进行不同的处理,例如发布或者调试,需要配置相应功能的证书及描述文件,比如,发布安装包时,可以上传安装包至公共平台以供用户下载安装时,需要配置发布证书和发布描述文件,该发布证书即认证开发者使用的设备具有发布APP的权限;发布描述文件即标识某个设备具有安装APP的权限,记录了可以安装该APP的IMEI(International Mobile Equipment Identity,国际移动设备识别码)号。目前,iOS平台封装打包获得安装包的操作是手动完成的,即需要开发者从开发者网站上下载相应功能的证书及描述文件,与可执行文件一起封装打包获得安装包。然而,随着项目的迭代和规范化,手动封装打包获得安装包的方式繁琐复杂,效率较低。

### 发明内容

[0003] 本申请实施例提供一种安装包处理方法及装置,用以解决现有技术中安装包打包方式繁琐复杂问题。

[0004] 第一方面,本申请实施例中提供了一种安装包处理方法,包括:

[0005] 响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;

[0006] 对所述项目源代码进行静态编译,获得可执行文件;

[0007] 根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;

[0008] 将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。

[0009] 第二方面,本申请实施例提供了一种安装包处理装置,包括:

[0010] 获取模块,用于响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;

[0011] 获得模块,用于对所述项目源代码进行静态编译,获得可执行文件;

[0012] 配置模块,用于根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;

[0013] 封装模块,用于将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。

[0014] 本申请实施例中,响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;对所述项目源代码进行静态编译,获得可执行文件;根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。通过集成开发工具获取可执行文件、目标证书和目标描述文件,将其进行封装获得安装包,解决了现有方式中手动打包获得安装包的方式复杂繁

琐的问题,提升了打包获得安装包的效率。

[0015] 本申请的这些方面或其他方面在以下实施例的描述中会更加简明易懂。

### 附图说明

[0016] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0017] 图1示出了本申请提供的一种安装包处理方法一个实施例的流程图;

[0018] 图2示出了本申请提供的一种安装包处理方法另一个实施例的流程图;

[0019] 图3示出了本申请提供的一种安装包处理装置一个实施例的结构示意图;

[0020] 图4示出了本申请提供的一种安装包处理装置另一个实施例的结构示意图。

### 具体实施方式

[0021] 为了使本技术领域的人员更好地理解本申请方案,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述。

[0022] 在本申请的说明书和权利要求书及上述附图中的描述的一些流程中,包含了按照特定顺序出现的多个操作,但是应该清楚了解,这些操作可以不按照其在本文中出现的顺序来执行或并行执行,操作的序号如101、102等,仅仅是用于区分各个不同的操作,序号本身不代表任何的执行顺序。另外,这些流程可以包括更多或更少的操作,并且这些操作可以按顺序执行或并行执行。需要说明的是,本文中的“第一”、“第二”等描述,是用于区分不同的消息、设备、模块等,不代表先后顺序,也不限定“第一”和“第二”是不同的类型。

[0023] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0024] 图1为本申请实施例提供的一种安装包处理方法一个实施例的流程图,该方法是在集成开发工具中实施的,该方法可以包括以下几个步骤:

[0025] 101:响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码。

[0026] 该安装包处理指令可以为用户设置的特定时间的处理指令,如用户设置的每天某一固定时间对安装包进行处理的处理指令。响应于该安装包处理指令,在该固定时间利用集成开发工具,获取用户上传的项目源代码。

[0027] 或者,该安装包处理指令还可以为检测到用户上传的项目源代码更新时生成的处理指令。响应于该安装包处理指令,利用集成开发工具,获取用户上传的更新后的项目源代码。

[0028] 此外,该安装包处理指令还可以为基于用户对该安装包的处理需求,响应用户点击集成开发工具提供的处理控件时生成的处理指令。

[0029] 其中,项目源代码是指待开发的某个应用程序的源代码,获取用户上传的项目源代码可以是获取用户上传到项目的远程仓库中的源代码,该远程仓库可以指用于保存应用

程序的源代码的远程仓库。远程仓库中的源代码可以参与打包获得安装包过程,没有上传至该项目远程仓库的更新代码不参与上述打包过程。即,响应于安装包处理指令,利用集成开发工具,获取用户上传至远程仓库的最新的源代码进行处理。

[0030] 其中,该集成开发工具可以是指用于提供程序开发环境的应用程序,具有代码编译功能、分析功能、调试功能和打包上传功能等多项功能。对于IOS的应用程序,该集成开发工具例如可以为xcodeserver。

[0031] 102:对所述项目源代码进行静态编译,获得可执行文件。

[0032] 103:根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件。

[0033] 开发者账号即开发者开发过程中用于真机调试、证书签名,把安装包上传到公共平台或测试平台上使用的账号。

[0034] 根据开发者账号可以获得所有证书和对应的描述文件。其中,证书可以分为开发证书和发布证书。开发证书又可以称为测试证书,认证开发者使用的设备具有开发该应用程序的权限,在开发过程中需要进行真机调试时,需要使用开发证书;发布证书又可以称为生产证书,认证开发者使用的设备具有发布该应用程序的权限,在开发过程后期需要进行上线发布应用程序时,需要使用发布证书。

[0035] 描述文件也可以分为开发描述文件和发布描述文件。开发描述文件可以标识某个设备具有安装调试该应用程序的权限;发布描述文件则可以标识某个设备具有安装使用该应用程序的权限。开发描述文件对应于开发证书,发布描述文件对应于发布证书。

[0036] 安装包处理指令中可以包括处理类型,处理类型可以包括真机调试和上线发布,基于处理类型,可以直接配置对应的目标证书和目标描述文件。例如,处理类型为真机调试时,目标证书即为开发证书,目标描述文件为开发描述文件,处理类型为上线发布时,目标证书为发布证书,目标描述文件为发布描述文件。

[0037] 因此,可选地,步骤103可以具体为根据在所述集成开发工具中添加的开发者账号,响应于安装包上传指令,配置发布证书和发布描述文件。

[0038] 在实际应用中,根据在所述集成开发工具中添加的开发者账号,按照所述安装包处理指令中的处理类型,配置对应功能的目标证书和目标描述文件。

[0039] 104:将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。

[0040] 将所述可执行文件及与其对应的目标证书和目标描述文件进行封装获得安装包,以便于对该安装包进行调试或者上传等处理操作。

[0041] 本实施例中,基于集成开发工具,响应于安装包处理指令,利用集成开发工具获取用户上传的项目源代码,并对该项目源代码进行静态编译,获得可以被操作系统进行加载执行的可执行文件;再根据集成开发工具中添加的开发者账号,为该可执行文件配置对应功能的目标证书和目标描述文件。之后将该目标证书、目标描述文件与该可执行文件进行封装可以获得安装包,以便于后续对该安装包进行发布等处理操作。通过上述技术方案,基于该集成开发工具,可以实现自动配置对应的目标证书及目标描述文件,与可执行文件进行封装获得安装包的操作。实现安装包的自动打包过程,改善了现有技术中,需要手动下载获取该证书及描述文件,再经过手动封装获得安装包的方式,提高了安装包生成效率。

[0042] 在某些实施例中,根据在所述集成开发工具中添加的开发者账号,配置目标证书

和目标描述文件的方法可以包括：

[0043] 根据在所述集成开发工具中添加的开发者账号，创建bundle ID (Bundle identifier,应用标识符)；

[0044] 根据所述bundle ID,配置目标证书和目标描述文件。

[0045] Bundle ID是项目源代码对应的应用程序的标识符，也可以称为应用ID,是每一个应用程序的全球唯一标识。每开发一个新的应用程序，需要创建一个该应用程序对应的bundle ID。其中，应用程序的bundle ID可以根据开发者账号进行创建，每一个开发者账号可以创建多个bundle ID。

[0046] 证书作为对开发者使用的设备具有开发或发布应用程序权限的认证，需要根据开发者账号进行创建。每一个开发者账号创建的证书是有限的，即多个bundle ID可以使用同一个证书。而描述文件作为对某个设备具有安装某个应用程序的权限的标识，描述了应用程序和证书的对应关系，需要根据bundle ID和证书进行创建。每一个bundle ID对应一个描述文件，即每一个开发者账号可以拥有多个描述文件。其中，开发描述文件需要根据bundle ID、开发证书和测试设备进行创建，用于标识该测试设备具有测试该应用程序的权限；发布描述文件需要根据bundle ID和发布证书进行创建，由于正式发布的应用程序无需限制安装设备，因此无需选择设备。

[0047] 根据在该集成开发工具中添加的开发者账号，创建bundle ID,并根据安装包处理指令中的处理类型，结合bundle ID,配置对应功能的目标证书和目标描述文件。

[0048] 该集成开发工具集成了开发者账户的功能。基于该集成开发工具，根据开发者账号和密码，可以自动创建bundle ID,并自动创建对应的目标证书和目标描述文件。无需用户手动从开发者网站下载获取证书和描述文件，并节省了配置目标证书和目标描述文件的时间，提高了安装包处理效率。

[0049] 在实际应用中，应用程序的开发并非是某一位开发人员完成的，对于公司开发的某一个应用程序来说，通常是由开发团队协同开发的。该开发者账号可以为公司账号，该公司账号下可以添加多位开发人员的个人开发者账号。其中，每位开发人员可以使用自己的开发者账号创建开发证书，开发描述文件则由团队管理员根据各位开发人员的开发证书进行创建。而发布证书和发布描述则根据公司账号进行创建。

[0050] 因此，在基于集成开发工具获得用于真机调试的安装包时，可以根据bundle ID识别出包含该bundle ID的描述文件，再根据开发人员的开发者账号匹配对应的开发证书和对应的开发描述文件，封装获得用于真机调试的安装包。而在基于集成开发工具获得用于上传发布的安装包时，可以根据bundle ID识别出包含该bundle ID的描述文件，再根据功能名称，匹配发布证书和对应的发布描述文件，封装获得用于上传发布的安装包。

[0051] 本申请实施例中，通过该集成开发工具集合开发者账户的功能，响应于安装包不同的处理指令，自动创建并识别对应功能的目标证书和目标描述文件，实现了自动封装获得安装包的操作，改善了现有技术中需要用户手动创建并识别对应功能的目标证书和目标描述文件，继而手动封装获得安装包的方式繁琐的问题，提高了安装包处理效率。

[0052] 在某些实施例中，基于该集成开发工具封装获得安装包之后，该方法还可以包括：

[0053] 上传所述安装包。

[0054] 其中，安装包可以上传到多个平台，上传安装包也有多种方式。

[0055] 作为一种可选方式,上传所述安装包的方法可以包括:

[0056] 获取用户提供的上传路径;

[0057] 根据所述上传路径,上传所述安装包至所述上传路径对应的包管理平台,以便于从所述包管理平台下载和安装所述安装包。

[0058] 该包管理平台可以为公司开发团队使用的应用程序安装包管理平台,用于存储各种应用程序的安装包,以便于团队成员进行下载并安装安装包。

[0059] 其中,上传至包管理平台的过程中,可以记录上传路径,该上传路径中可以包括IP地址信息。不同版本的安装包上传路径不同。

[0060] 在实际应用中,基于集成开发工具,响应于上传安装包至包管理平台的指令,利用该集成开发工具封装获得安装包,并获取上传路径,根据该上传路径,可以将该安装包上传至上传路径对应的包管理平台,以便于从该包管理平台下载和安装所述安装包,并记录上传路径和IP地址等信息。通过记录上传路径等信息,实现了利用该集成开发工具自动上传安装包至包管理平台的过程,无需用户手动操作,提升了安装包上传至包管理平台的效率。

[0061] 作为另一种可选方式,上传所述安装包的方法还可以包括:

[0062] 利用所述开发者账号登录测试平台;

[0063] 上传所述安装包至测试平台,以供所述测试平台对所述安装包进行测试;

[0064] 若所述安装包测试通过,上传所述安装包至公共平台,以便于从所述公共平台下载和安装所述安装包。

[0065] 其中,安装包为IOS安装包时,该测试平台可以为testflight,即IOS官方发布测试版的应用程序。应用程序在正式上传至公共平台发布之前,需要进行一系列测试流程,测试通过及可以上传至公共平台进行发布,以便于从该公共平台下载安装包,使用应用程序的各项功能。其中,公共平台可以为应用程序商城APP store,用户可以从APP store下载安装包。

[0066] 在实际应用中,响应于上传安装包至测试平台的处理指令,利用该集成开发工具配置对应的目标证书和目标描述文件,将其与可执行文件封装获得安装包。由于该测试平台可以进行真机调试和发布至公共平台,因此该目标证书可以为开发证书,也可以为发布证书;目标描述文件也可以为开发描述文件和发布描述文件。利用开发者账号和密码登录测试平台,上传该安装包至测试平台,以供该测试平台对安装包进行测试。若该安装包通过测试,则可以上传至APP store,以便于用户从该APP store进行下载和安装,其中,该上传至APP store的安装包则需要配置发布证书和发布描述文件。

[0067] 若该安装包未通过测试,则重新对其进行修改调试,并重复以上测试流程,直至测试通过,可以将通过测试的安装包上传至APP store发布。

[0068] 在某些实施例中,上传所述安装包至测试平台的方法可以包括:

[0069] 导出所述安装包并保存到本地;

[0070] 调用构建工具插件上传所述安装包至测试平台。

[0071] 该构建工具插件可以为fastlane,fastlane是开发者构建的自动化构建工具,具有打包、上传等功能。该构建工具插件可以在本地查找待上传安装包中包含的目标证书和目标描述文件,从而上传对应的安装包至测试平台。

[0072] 在实际应用中,基于该集成开发工具,响应于上传安装包至测试平台的指令,利用



该集成开发工具封装获得安装包并将其导出保存至本地,调用构建工具插件自动识别存入本地的待上传安装包的目标证书和目标描述文件,从而将包含该目标证书和目标描述文件的安装包上传至测试平台,以便于测试平台进行测试。

[0073] 其中,该测试平台使用开发者账号进行登录,若存在包括该开发者账号在内的历史上传信息,即预配置信息,则读取该预配置信息,实现登录上传;若不存在该预配置信息,则需要用户进行手动输入,该方法适用于首次上传安装包至测试平台。

[0074] 本申请实施例中,通过利用该集成开发工具封装获得安装包,导出并存入本地,调用构建工具插件识别安装包中的证书和描述文件,上传至测试平台,实现了将安装包自动上传至测试平台的操作,改善了现有技术中需要手动从本地识别安装包并上传至测试平台的方式,简捷高效。

[0075] 图2为本申请实施例提供的一种安装包处理方法另一个实施例的流程图,该方法是基于集成开发工具实施的,该方法可以包括以下几个步骤:

[0076] 201:响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码。

[0077] 202:对所述项目源代码进行静态编译,获得可执行文件。

[0078] 203:根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件。

[0079] 204:基于用户提供的目标功能配置需求,生成配置文件。

[0080] 205:将所述配置文件与所述可执行文件、目标证书、目标描述文件和运行需要的资源封装获得安装包;所述配置文件用于所述安装包运行时提供所述目标功能。

[0081] 上述201~203可参见实施例1中的相应内容,此处不再赘述。

[0082] 上述204中,该目标功能配置可以为开发人员对应用程序添加的个性化配置,目标功能配置需求可以为该安装包运行时基于个性化配置需要实现的功能。个性化配置可以具体为应用程序的版本类型配置、应用程序发布的新特性、应用程序内的截图等内容。其中,版本类型配置具体可以包括,调试版本和发布版本。调试版本的安装包即包含了调试信息的安装包,便于开发人员在后续测试过程中进行调试;发布版本的安装包即为不包含调试信息的安装包,并进行了多种优化,便于用户很好地使用。该个性化配置包含在项目源代码内。

[0083] 由于该集成开发工具不识别项目源代码中包含的上述个性化配置内容,并且该部分内容需要及时进行调整,例如,在对该应用程序进行版本更新时,需要增加新的使用特性等信息,因此可以根据包含该个性化配置的项目源代码创建可以被集成开发工具识别的不同于可执行文件的配置文件,以便于与可执行文件一起封装获得安装包,使得安装包在运行时可以实现上述个性化配置。

[0084] 该配置文件可以为plist (Property List,属性列表文件)文件,通常用于储存用户设置,也可以用于存储捆绑的信息。在此,该plist文件具体可以包括个性化配置信息。

[0085] 上述205中,利用集成开发工具,将该plist文件与可执行文件、目标证书、目标描述文件以及运行需要的资源一并封装获得含有个性化配置的安装包,其中,该plist文件可以用于安装包运行时提供对应的功能。例如,开发人员想要获得用于安装调试的安装包,则可以先采用手动打包源代码的方式,获得包含调试版本配置信息的plist文件,再进行封装即可获得具有个性化配置的调试版本类型的安装包。

[0086] 在实际应用中,基于集成开发工具,在响应到安装包处理指令时,利用该集成开发工具获取到用户上传的项目源代码时,先对其个性化配置进行手动打包获得plist文件,之后将该plist文件与可执行文件、目标证书、目标描述文件以及应用程序运行所需要的资源文件一并封装,获得具有个性化配置的特定版本类型安装包。

[0087] 本实施例中,考虑到该集成开发工具xcodeserver不识别源代码中包含的个性化配置内容,通过将包含有个性化配置的源代码手动打包,获得可以被该集成开发工具识别的plist文件,之后利用该集成开发工具把该plist文件与可执行文件、目标证书、目标描述文件和运行所需要的资源文件进行封装,获得具有个性化配置的安装包,以较高效率实现了安装包的个性化配置。

[0088] 该具有个性化配置的安装包进行上传时,可以有多种方式。

[0089] 作为一种可实现方式,可以将该具有个性化配置的安装包上传至包管理平台,不同的版本类型对应不同的上传路径,记录该特定版本类型安装包对应的上传路径,同时,可以记录版本号和版本类型信息。

[0090] 作为另一种可实现方式,可以将该具有个性化配置的安装包上传至测试平台,利用构建工具插件可以识别本地安装包中包含的目标证书、目标描述文件、plist文件和个性化配置信息,将对应的安装包上传至测试平台。其中,上传至测试平台时,还可以上传应用程序的截图、应用程序的描述以及应用程序发布需要的所有资源信息。

[0091] 在实际应用中,基于集成开发工具封装获得安装包时,可以根据手动打包源代码获得的plist文件,判断该安装包是调试版本类型安装包还是发布版本类型安装包。上传安装包至测试平台进行测试时,可以将调试版本类型安装包上传,以便于测试员在测试过程中启动多种服务监控错误,寻找漏洞,并及时对错误进行修改。同时,也可以将发布版本类型安装包上传,考虑到发布版本类型安装包状态下测试的情况,是用户使用的真实情况。因此发布版本类型安装包的测试更有助于保证用户体验。

[0092] 若上传至测试平台的安装包通过测试,则可以上传发布版本类型安装包至公共平台APP store,以便于下载和安装该发布版本类型安装包,更好地使用应用程序的各项功能,无需调试。

[0093] 图3为本申请实施例提供的一种安装包处理装置一个实施例的流程图,该装置可以包括:

[0094] 第一获取模块301,用于响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;

[0095] 第一获得模块302,用于对所述项目源代码进行静态编译,获得可执行文件;

[0096] 第一配置模块303,用于根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;

[0097] 第一封装模块304,用于将所述可执行文件、所述目标证书和所述目标描述文件封装获得安装包。

[0098] 在某些实施例中,该配置模块可以包括:

[0099] 第一创建单元,用于根据在所述集成开发工具中添加的开发者账号,创建应用标识符bundle ID;

[0100] 第二创建单元,用于根据所述bundle ID,创建目标证书和目标描述文件。

- [0101] 在某些实施例中,该装置还可以包括:
- [0102] 上传模块,用于上传所述安装包。
- [0103] 在某些实施例中,该上传模块可以包括:
- [0104] 获取单元,用于获取用户提供的上传路径;
- [0105] 第一上传单元,用于根据所述上传路径,上传所述安装包至所述上传路径对应的包管理平台,以便于从所述包管理平台下载和安装所述安装包。
- [0106] 在某些实施例中,该上传模块还可以包括:
- [0107] 登录单元,用于利用所述开发者账号登录测试平台;
- [0108] 第二上传单元,用于上传所述安装包至测试平台,以供所述测试平台对所述安装包进行测试;
- [0109] 第三上传单元,用于若所述安装包测试通过,上传所述安装包至公共平台,以便于从所述公共平台下载和安装所述安装包。
- [0110] 在某些实施例中,该第二上传单元可以包括:
- [0111] 导出子单元,用于导出所述安装包并保存到本地;
- [0112] 上传子单元,用于调用构建工具插件上传所述安装包至测试平台。
- [0113] 在某些实施例中,该封装模块可以包括:
- [0114] 生成单元,用于基于用户提供的目标功能配置需求,生成配置文件;
- [0115] 封装单元,用于将所述配置文件与所述可执行文件、目标证书、目标描述文件和运行需要的资源封装获得安装包;所述配置文件用于所述安装包运行时提供所述目标功能。
- [0116] 图3所述的安装包处理装置可以执行图1所示实施例所述的安装包处理方法,其实现原理和技术效果不再赘述。对于上述实施例中的安装包处理装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述,此处将不做详细阐述说明。
- [0117] 图4为本申请实施例提供的一种安装包处理装置另一个实施例的流程图,该装置可以包括:
- [0118] 第二获取模块401,用于响应于安装包处理指令,利用集成开发工具,获取用户上传的项目源代码;
- [0119] 第二获得模块402,用于对所述项目源代码进行静态编译,获得可执行文件;
- [0120] 第二配置模块403,用于根据在所述集成开发工具中添加的开发者账号,配置目标证书和目标描述文件;
- [0121] 生成模块404,用于基于用户提供的目标功能配置需求,生成配置文件;
- [0122] 第二封装模块405,用于将所述配置文件与所述可执行文件、目标证书、目标描述文件和运行需要的资源封装获得安装包;所述配置文件用于所述安装包运行时提供所述目标功能。
- [0123] 图4所述的安装包处理装置可以执行图2所示实施例所述的安装包处理方法,其实现原理和技术效果不再赘述。对于上述实施例中的安装包处理装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述,此处将不做详细阐述说明。
- [0124] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,

装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0125] 以上所描述的装置实施例仅仅是示意性的,其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性的劳动的情况下,即可以理解并实施。

[0126] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到各实施方式可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件。基于这样的理解,上述技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品可以存储在计算机可读存储介质中,如ROM/RAM、磁碟、光盘等,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行各个实施例或者实施例的某些部分所述的方法。

[0127] 最后应说明的是:以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

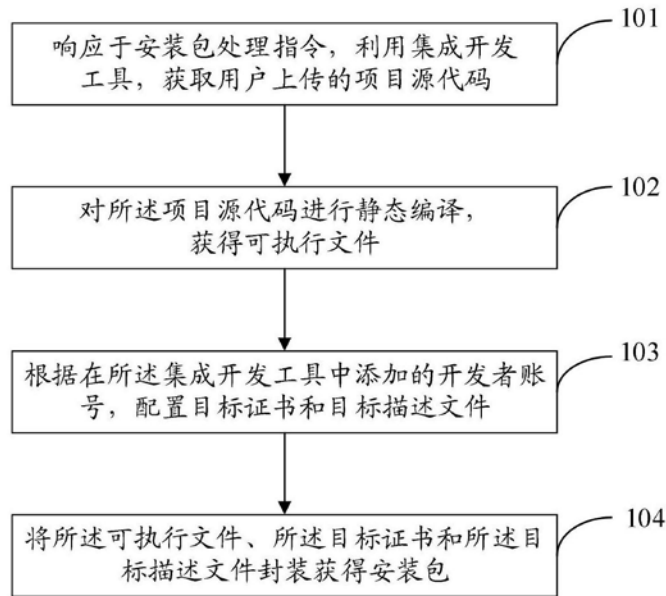


图1

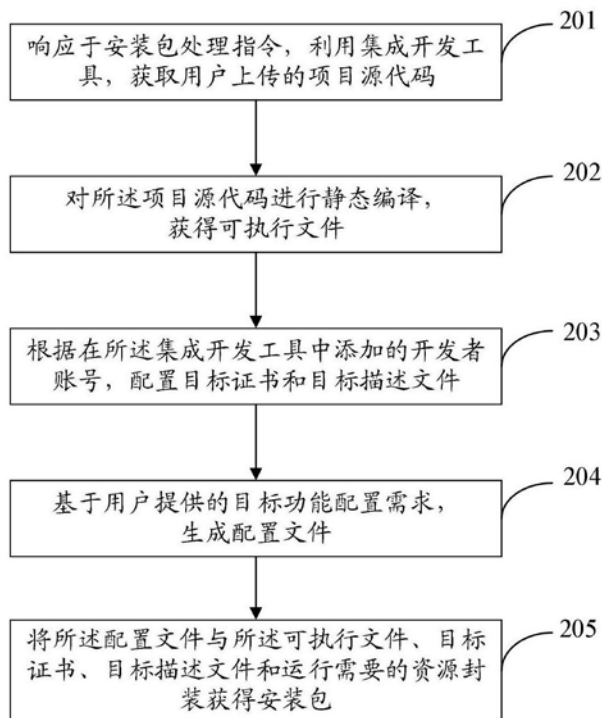


图2

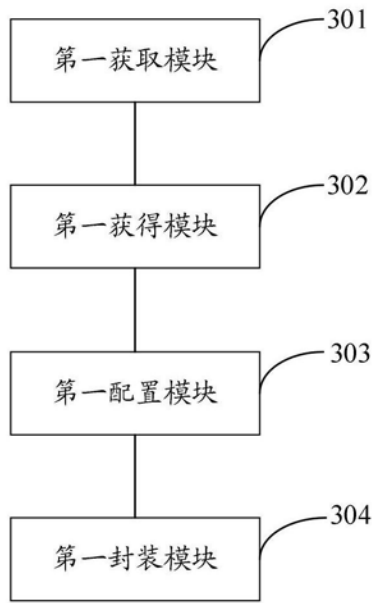


图3

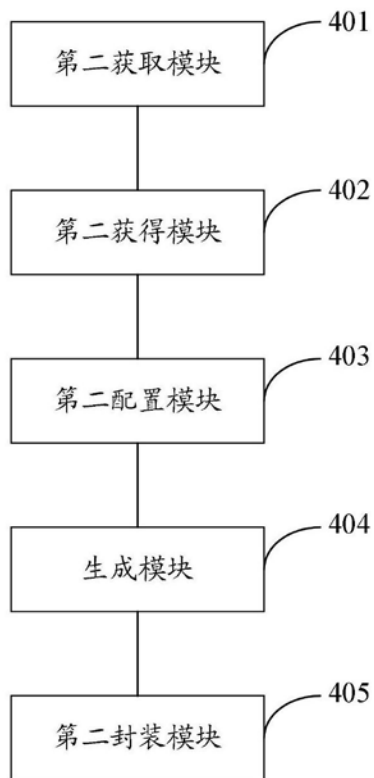


图4