



(12) 发明专利

(10) 授权公告号 CN 110858143 B

(45) 授权公告日 2023.07.28

(21) 申请号 201810960846.8

G06F 8/71 (2018.01)

(22) 申请日 2018.08.22

(56) 对比文件

(65) 同一申请的已公布的文献号  
申请公布号 CN 110858143 A

CN 103645900 A, 2014.03.19

CN 107357558 A, 2017.11.17

CN 105991797 A, 2016.10.05

(43) 申请公布日 2020.03.03

CN 108319460 A, 2018.07.24

US 2012266135 A1, 2012.10.18

(73) 专利权人 广州虎牙信息科技有限公司  
地址 511400 广东省广州市番禺区南村镇  
万博二路79号万博商务区万达商业广  
场北区B-1栋28层10单元

审查员 李奕福

(72) 发明人 李武军

(74) 专利代理机构 北京品源专利代理有限公司  
11332

专利代理师 孟金喆

(51) Int. Cl.

G06F 8/61 (2018.01)

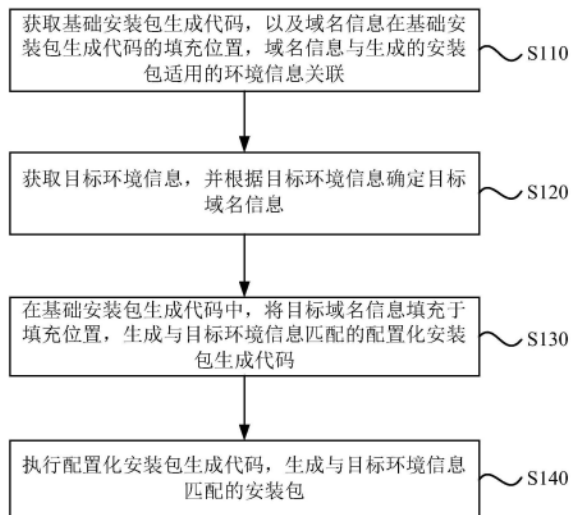
权利要求书2页 说明书8页 附图5页

(54) 发明名称

一种安装包生成方法、装置、设备及存储介  
质

(57) 摘要

本发明实施例公开了一种安装包生成方法、装置、设备及存储介质。该方法包括：获取基础安装包生成代码，以及域名信息在基础安装包生成代码的填充位置，域名信息与生成的安装包适用的环境信息关联；获取目标环境信息，并根据目标环境信息确定目标域名信息；在基础安装包生成代码中，将目标域名信息填充于填充位置，生成与目标环境信息匹配的配置化安装包生成代码；执行配置化安装包生成代码，生成与目标环境信息匹配的安装包，通过本发明的技术方案，能够生成适用于多种环境的安装包。



1. 一种安装包生成方法,其特征在于,包括:

获取基础安装包生成代码,以及域名信息在所述基础安装包生成代码的填充位置,所述域名信息与生成的安装包适用的环境信息关联;

获取目标环境信息,并根据所述目标环境信息确定目标域名信息,其中,所述目标环境信息包括至少两种环境信息;

在所述基础安装包生成代码中,将所述目标域名信息填充于所述填充位置,生成与所述目标环境信息匹配的配置化安装包生成代码;

执行所述配置化安装包生成代码,生成与所述目标环境信息匹配的安装包。

2. 根据权利要求1所述的方法,其特征在于,所述环境信息包括正式环境、测试环境以及开发环境中的至少一个。

3. 根据权利要求1所述的方法,其特征在于,获取目标环境信息,并根据所述目标环境信息确定目标域名信息之前,还包括:

通过编译工具建立包括环境信息和域名的对应关系的应用程序文件;

相应的,根据所述目标环境信息确定目标域名信息包括:

根据所述目标环境信息查找所述应用程序文件,得到与所述目标环境信息对应的目标域名信息。

4. 根据权利要求1所述的方法,其特征在于,获取目标环境信息,并根据所述目标环境信息确定目标域名信息之前,还包括:

通过编译工具建立环境文件,其中,所述环境文件中包括目标环境信息。

5. 根据权利要求3或4所述的方法,其特征在于,所述编译工具为gradle。

6. 一种安装包生成装置,其特征在于,包括:

获取模块,用于获取基础安装包生成代码,以及域名信息在所述基础安装包生成代码的填充位置,所述域名信息与生成的安装包适用的环境信息关联;

确定模块,用于获取目标环境信息,并根据所述目标环境信息确定目标域名信息,其中,所述目标环境信息包括至少两种环境信息;

填充模块,用于在所述基础安装包生成代码中,将所述目标域名信息填充于所述填充位置,生成与所述目标环境信息匹配的配置化安装包生成代码;

生成模块,用于执行所述配置化安装包生成代码,生成与所述目标环境信息匹配的安装包。

7. 根据权利要求6所述的装置,其特征在于,所述环境信息包括正式环境、测试环境以及开发环境中的至少一个。

8. 根据权利要求6所述的装置,其特征在于,还包括:

建立模块,用于通过编译工具建立包括环境信息和域名的对应关系的应用程序文件;

相应的,确定模块具体用于:

根据所述目标环境信息查找所述应用程序文件,得到与所述目标环境信息对应的目标域名信息。

9. 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1-5中任一所述的方法。

10. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1-5中任一所述的方法。

## 一种安装包生成方法、装置、设备及存储介质

### 技术领域

[0001] 本发明实施例涉及计算机技术,尤其涉及一种安装包生成方法、装置、设备及存储介质。

### 背景技术

[0002] 随着科技的发展,计算机技术已经成为衡量科技水平的一个关键要素,计算机程序针对不同的使用者生成不同的安装包,例如,若使用者为使用应用程序的用户,则生成正式环境对应的安装包,若使用者为测试人员,则生成测试环境对应的安装包,若使用者为开发人员,则生成开发环境对应的安装包。

[0003] 域名,是由一串用点分隔的名字组成的Internet上某一台计算机或计算机组的名称,用于在数据传输时标识计算机的电子方位,上述每一种环境对应一个域名,例如,正式环境对应正式环境域名、测试环境对应测试环境域名以及开发环境对应开发环境域名。

[0004] 现有技术中,生成软件安装包的方法为手动在基础安装包生成代码中写入域名,进而生成相应的软件安装包,通过上述方式生成软件安装包需要用户具备一定的软件编程水平,且只能生成某一环境下的软件安装包,不能够一次性生成适用于多种环境的安装包。

### 发明内容

[0005] 本发明实施例提供一种安装包生成方法、装置、设备及存储介质,以实现生成适用于多种环境的安装包。

[0006] 第一方面,本发明实施例提供了一种安装包生成,包括:

[0007] 获取基础安装包生成代码,以及域名信息在所述基础安装包生成代码的填充位置,所述域名信息与生成的安装包适用的环境信息关联;

[0008] 获取目标环境信息,并根据所述目标环境信息确定目标域名信息;

[0009] 在所述基础安装包生成代码中,将所述目标域名信息填充于所述填充位置,生成与所述目标环境信息匹配的配置化安装包生成代码;

[0010] 执行所述配置化安装包生成代码,生成与所述目标环境信息匹配的安装包。

[0011] 进一步的,所述环境信息包括正式环境、测试环境以及开发环境中的至少一个。

[0012] 进一步的,获取目标环境信息,并根据所述目标环境信息确定目标域名信息之前,还包括:

[0013] 通过编译工具建立包括环境信息和域名的对应关系的应用程序文件;

[0014] 相应的,根据所述目标环境信息确定目标域名信息包括:

[0015] 根据所述目标环境信息查找所述应用程序文件,得到与所述目标环境信息对应的目标域名信息。

[0016] 进一步的,获取目标环境信息,并根据所述目标环境信息确定目标域名信息之前,还包括:

[0017] 通过编译工具建立环境文件,其中,所述环境文件中包括目标环境信息。

- [0018] 进一步的,所述编译工具为gradle。
- [0019] 第二方面,本发明实施例还提供了一种安装包生成装置,该装置包括:
- [0020] 获取模块,用于获取基础安装包生成代码,以及域名信息在所述基础安装包生成代码的填充位置,所述域名信息与生成的安装包适用的环境信息关联;
- [0021] 确定模块,用于获取目标环境信息,并根据所述目标环境信息确定目标域名信息;
- [0022] 填充模块,用于在所述基础安装包生成代码中,将所述目标域名信息填充于所述填充位置,生成与所述目标环境信息匹配的配置化安装包生成代码;
- [0023] 生成模块,用于执行所述配置化安装包生成代码,生成与所述目标环境信息匹配的安装包。
- [0024] 进一步的,所述环境信息包括正式环境、测试环境以及开发环境中的至少一个。
- [0025] 进一步的,还包括:
- [0026] 建立模块,用于通过编译工具建立包括环境信息和域名的对应关系的应用程序文件;
- [0027] 相应的,确定模块具体用于:
- [0028] 根据所述目标环境信息查找所述应用程序文件,得到与所述目标环境信息对应的目标域名信息。
- [0029] 第三方面,本发明实施例还提供了一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述程序时实现如本发明实施例中任一所述的安装包生成方法。
- [0030] 第四方面,本发明实施例还提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如本发明实施例中任一所述的安装包生成方法。
- [0031] 本发明实施例通过获取基础安装包生成代码,以及域名信息在基础安装包生成代码的填充位置,域名信息与生成的安装包适用的环境信息关联;获取目标环境信息,并根据目标环境信息确定目标域名信息;在基础安装包生成代码中,将目标域名信息填充于填充位置,生成与目标环境信息匹配的配置化安装包生成代码;执行配置化安装包生成代码,生成与目标环境信息匹配的安装包,以实现生成适用于多种环境的安装包。

## 附图说明

- [0032] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本发明的某些实施例,因此不应被看作是对范围的限定,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相关的附图。
- [0033] 图1是本发明实施例一中的一种安装包生成方法的流程图;
- [0034] 图2A是本发明实施例二中的一种安装包生成方法的流程图;
- [0035] 图2B是本发明实施例二中的一种域名信息确定方法的流程图;
- [0036] 图3是本发明实施例三中的一种安装包生成装置的结构示意图;
- [0037] 图4是本发明实施例四中的一种计算机设备的结构示意图。

## 具体实施方式

[0038] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明,而非对本发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明相关的部分而非全部结构。

[0039] 应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。同时,在本发明的描述中,术语“第一”、“第二”等仅用于区分描述,而不能理解为指示或暗示相对重要性。

[0040] 实施例一

[0041] 图1为本发明实施例一提供的一种安装包生成方法的流程图,本实施例可适用于安装包生成的情况,该方法可以由本发明实施例中的安装包生成装置来执行,该装置可采用软件和/或硬件的方式实现,如图1所示,该方法具体包括如下步骤:

[0042] S110,获取基础安装包生成代码,以及域名信息在基础安装包生成代码的填充位置,域名信息与生成的安装包适用的环境信息关联。

[0043] 其中,所述基础安装包生成代码为用于生成安装包的基础代码。

[0044] 其中,域名信息在基础安装包生成代码的填充位置为基础安装包生成代码中用于填写域名信息的位置。

[0045] 其中,域名信息与生成的安装包适用的环境信息关联指的是一个域名信息对应一种生成的安装包适用的环境信息,例如可以是,若域名信息为A,则将域名信息A填写至基础安装包生成代码中的填充位置后,生成的安装包适用的环境信息为Q;若域名信息为B,则将域名信息B填写至基础安装包生成代码中的填充位置后,生成的安装包适用的环境信息为W。

[0046] 具体的,获取基础安装包生成代码,以及域名信息在基础安装包生成代码中的填充位置,域名信息与生成的安装包使用的环境信息关联。

[0047] 可选的,所述环境信息包括正式环境、测试环境以及开发环境中的至少一个。

[0048] 具体的,若环境信息为正式环境,则对应正式环境的域名信息;若环境信息为测试环境,则对应测试环境的域名信息,若环境信息为开发环境,则对应开发环境的域名信息。例如可以是,正式环境official,开发环境dev,测试环境qa,“officialUrl”对应“http://udb3.huhu.tv”;“devUrl”对应“http://udb3-a.huhu.tv”。

[0049] S120,获取目标环境信息,并根据目标环境信息确定目标域名信息。

[0050] 其中,所述目标环境信息的获取方式为预先建立一个文件用于存储生成的安装包需要适用的环境信息,所述文件中存储的所有环境信息则为目标环境信息,例如可以是,预先通过编译工具建立一个文件,文件中存储正式环境official,开发环境dev和测试环境qa,则目标环境信息为:正式环境official,开发环境dev和测试环境qa。

[0051] 其中,根据目标环境信息确定目标域名信息的方式为预先存储关于环境信息和域名信息的对应关系,然后根据目标环境信息查询环境信息和域名信息的对应关系,进而确定目标域名信息,例如可以是,环境信息Q对应域名信息A、环境信息W对应域名信息B、环境信息E对应域名信息C,若目标环境信息为环境信息Q和环境信息W,则目标域名信息则为域名信息A和域名信息B。

[0052] 具体的,获取目标环境信息,根据目标环境信息查找环境信息与域名信息的对应

关系,进而确定目标域名信息。

[0053] S130,在基础安装包生成代码中,将目标域名信息填充于填充位置,生成与目标环境信息匹配的配置化安装包生成代码。

[0054] 其中,若目标环境信息中包括至少两种环境信息,则分别将目标环境信息中的环境信息对应的域名信息填充于填充位置,得到匹配的配置化安装包生成代码。

[0055] 具体的,在基础安装包生成代码中,将根据目标环境信息确定的目标域名信息填写于填充位置,生成与目标环境信息匹配的配置化安装包生成代码,例如可以是,若目标环境信息为环境信息Q和环境信息W,根据环境信息Q和环境信息W确定目标域名信息为域名信息A和域名信息B,将域名信息A填写于填充位置,生成与环境信息Q匹配的配置化安装包生成代码;将域名信息B填写于填充位置,生成与环境信息W匹配的配置化安装包生成代码。S140,执行配置化安装包生成代码,生成与目标环境信息匹配的安装包。

[0056] 其中,与目标环境信息匹配的安装包的个数与目标环境信息中包含的环境信息的数量有关,例如可以是,若目标环境信息中包含3个环境信息,则生成与目标环境信息匹配的安装包则包含3个安装包,每一个安装包分别与环境信息匹配。

[0057] 具体的,执行配置化安装包生成代码,生成与目标环境信息匹配的安装包,例如可以是,若目标环境信息为环境信息Q和环境信息W,根据环境信息Q和环境信息W确定目标域名信息为域名信息A和域名信息B,将域名信息A填写于填充位置,生成与环境信息Q匹配的配置化安装包生成代码X;将域名信息B填写于填充位置,生成与环境信息W匹配的配置化安装包生成代码Y;执行配置化安装包生成代码X,生成与环境信息Q匹配的安装包,执行配置化安装包生成代码Y,生成与环境信息W匹配的安装包。

[0058] 本实施例的技术方案,通过获取基础安装包生成代码,以及域名信息在基础安装包生成代码的填充位置,域名信息与生成的安装包适用的环境信息关联;获取目标环境信息,并根据目标环境信息确定目标域名信息;在基础安装包生成代码中,将目标域名信息填充于填充位置,生成与目标环境信息匹配的配置化安装包生成代码;执行配置化安装包生成代码,生成与目标环境信息匹配的安装包,能够生成适用于多种环境的安装包。

[0059] 实施例二

[0060] 图2A为本发明实施例二中的一种安装包生成方法的流程图,本实施例以上述实施例为基础进行优化,在本实施例中,获取目标环境信息,并根据所述目标环境信息确定目标域名信息之前,还包括:通过编译工具建立包括环境信息和域名的对应关系的应用程序文件;相应的,根据所述目标环境信息确定目标域名信息包括:根据所述目标环境信息查找所述应用程序文件,得到与所述目标环境信息对应的目标域名信息

[0061] 如图2A所示,本实施例的方法具体包括如下步骤:

[0062] S210,获取基础安装包生成代码,以及域名信息在基础安装包生成代码的填充位置,域名信息与生成的安装包适用的环境信息关联。

[0063] 可选的,获取目标环境信息,并根据所述目标环境信息确定目标域名信息之前,还包括:

[0064] 通过编译工具建立环境文件,其中,所述环境文件中包括目标环境信息。

[0065] 其中,所述编译工具是一个基于Apache Ant和Apache Maven概念的项目自动化构建开源工具。

- [0066] 可选的,所述编译工具为gradle。
- [0067] 其中,所述环境文件用于存储环境信息,例如可以是,正式环境official,开发环境dev,测试环境qa。
- [0068] 其中,所述环境文件中的环境信息均为目标环境信息。
- [0069] 具体的,通过编译工具建立环境文件,环境文件中存储环境信息。
- [0070] S220,通过编译工具建立包括环境信息和域名的对应关系的应用程序文件。
- [0071] 其中,所述应用程序文件为通过编译工具建立的用于存储环境信息和域名的对应关系的文件。
- [0072] 具体的,通过编译工具建立应用程序文件,在应用程序文件中存储环境信息和域名的对应关系,例如可以是,“officialUrl”对应“http://udb3.huhu.tv”;“devUrl”对应“http://udb3-a.huhu.tv”;“qaUrl”对应“http://udb3-b.huhu.tv”。
- [0073] S230,获取目标环境信息。
- [0074] 具体的,获取环境文件中的环境信息作为目标环境信息。
- [0075] S240,根据目标环境信息查找应用程序文件,得到与目标环境信息对应的目标域名信息。
- [0076] 具体的,根据目标环境信息查找应用程序文件,得到于目标环境信息对应的目标域名信息,例如可以是,若目标环境信息为正式环境official,开发环境dev和测试环境qa,则根据目标环境信息为正式环境official,开发环境dev和测试环境qa查找应用程序文件,若应用程序文件为:“officialUrl”对应http://udb3.huhu.tv;“devUrl”对应http://udb3-a.huhu.tv;“qaUrl”对应http://udb3-b.huhu.tv,则得到与正式环境official的目标域名信息http://udb3.huhu.tv,与开发环境dev对应http://udb3-a.huhu.tv;与测试环境qa对应的http://udb3-b.huhu.tv。
- [0077] S250,在基础安装包生成代码中,将目标域名信息填充于填充位置,生成与目标环境信息匹配的配置化安装包生成代码。
- [0078] S260,执行配置化安装包生成代码,生成与目标环境信息匹配的安装包。
- [0079] 在一个具体的例子中,如图2B所示,使用gradle脚本在基础安装包生成代码中建立BuildConfig类,建立appDomain.json文件,在appDomain.json文件中存储一个登录接口所对应相应环境的域名例如:登录接口:{"officialUrl"对应http://udb3.huhu.tv;“devUrl”对应http://udb3-a.huhu.tv;“qaUrl”对应http://udb3-b.huhu.tv}。建立Environment.txt文件,在Environment.txt文件中存储主要包含哪些环境,例如:正式环境official,开发环境dev,测试环境qa,读取Environment.txt文件中的目标环境信息,根据读取出的目标环境信息查找appDomain.json文件,得到与目标环境信息对应的目标域名信息,将目标域名信息保存至BuildConfig类中,apk中通过登录接口就可以访问对应目标环境信息的域名了。
- [0080] 本实施例的技术方案,通过获取基础安装包生成代码,以及域名信息在基础安装包生成代码的填充位置,域名信息与生成的安装包适用的环境信息关联,通过编译工具建立包括环境信息和域名的对应关系的应用程序文件,获取目标环境信息,根据目标环境信息查找应用程序文件,得到与目标环境信息对应的目标域名信息,在基础安装包生成代码中,将目标域名信息填充于填充位置,生成与目标环境信息匹配的配置化安装包生成代码,



执行配置化安装包生成代码,生成与目标环境信息匹配的安装包,能够生成适用于多种环境的安装包。

[0081] 实施例三

[0082] 图3为本发明实施例三提供一种安装包生成装置的结构示意图。本实施例可适用于安装包生成的情况,该装置可采用软件和/或硬件的方式实现,该装置可集成在任何提供安装包生成功能的设备中,如图3所示,所述安装包生成装置具体包括:获取模块310、确定模块320、填充模块330和生成模块340。

[0083] 其中,获取模块310,用于获取基础安装包生成代码,以及域名信息在所述基础安装包生成代码的填充位置,所述域名信息与生成的安装包适用的环境信息关联;

[0084] 确定模块320,用于获取目标环境信息,并根据所述目标环境信息确定目标域名信息;

[0085] 填充模块330,用于在所述基础安装包生成代码中,将所述目标域名信息填充于所述填充位置,生成与所述目标环境信息匹配的配置化安装包生成代码;

[0086] 生成模块340,用于执行所述配置化安装包生成代码,生成与所述目标环境信息匹配的安装包。

[0087] 可选的,所述环境信息包括正式环境、测试环境以及开发环境中的至少一个。

[0088] 可选的,还包括:

[0089] 建立模块,用于通过编译工具建立包括环境信息和域名的对应关系的应用程序文件;

[0090] 相应的,确定模块具体用于:

[0091] 根据所述目标环境信息查找所述应用程序文件,得到与所述目标环境信息对应的目标域名信息。

[0092] 可选的,还包括:

[0093] 环境文件建立模块,用于在获取目标环境信息,并根据所述目标环境信息确定目标域名信息之前,通过编译工具建立环境文件,其中,所述环境文件中包括目标环境信息。

[0094] 可选的,所述编译工具为gradle。

[0095] 上述产品可执行本发明任意实施例所提供的方法,具备执行方法相应的功能模块和有益效果。

[0096] 本实施例的技术方案,通过获取基础安装包生成代码,以及域名信息在基础安装包生成代码的填充位置,域名信息与生成的安装包适用的环境信息关联;获取目标环境信息,并根据目标环境信息确定目标域名信息;在基础安装包生成代码中,将目标域名信息填充于填充位置,生成与目标环境信息匹配的配置化安装包生成代码;执行配置化安装包生成代码,生成与目标环境信息匹配的安装包,能够生成适用于多种环境的安装包。

[0097] 实施例四

[0098] 图4为本发明实施例四中的一种计算机设备的结构示意图。图4示出了适于用来实现本发明实施方式的示例性计算机设备12的框图。图4显示的计算机设备12仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0099] 如图4所示,计算机设备12以通用计算设备的形式表现。计算机设备12的组件可以包括但不限于:一个或者多个处理器或者处理单元16,系统存储器28,连接不同系统组件

(包括系统存储器28和处理单元16)的总线18。

[0100] 总线18表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构(ISA)总线,微通道体系结构(MAC)总线,增强型ISA总线、视频电子标准协会(VESA)局域总线以及外围组件互连(PCI)总线。

[0101] 计算机设备12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被计算机设备12访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0102] 系统存储器28可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)30和/或高速缓存存储器32。计算机设备12可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例,存储系统34可以用于读写不可移动的、非易失性磁介质(图4未显示,通常称为“硬盘驱动器”)。尽管图4中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如CD-ROM,DVD-ROM或其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线18相连。存储器28可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0103] 具有一组(至少一个)程序模块42的程序/实用工具40,可以存储在例如存储器28中,这样的程序模块42包括——但不限于——操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块42通常执行本发明所描述的实施例中的功能和/或方法。

[0104] 计算机设备12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信,还可与一个或者多个使得用户能与该计算机设备12交互的设备通信,和/或与使得该计算机设备12能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口22进行。另外,本实施例中的计算机设备12,显示器24不是作为独立个体存在,而是嵌入镜面中,在显示器24的显示面不予显示时,显示器24的显示面与镜面从视觉上融为一体。并且,计算机设备12还可以通过网络适配器20与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。如图所示,网络适配器20通过总线18与计算机设备12的其它模块通信。应当明白,尽管图中未示出,可以结合计算机设备12使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0105] 处理单元16通过运行存储在系统存储器28中的程序,从而执行各种功能应用以及数据处理,例如实现本发明实施例所提供的安装包生成方法:获取基础安装包生成代码,以及域名信息在所述基础安装包生成代码的填充位置,所述域名信息与生成的安装包适用的环境信息关联;获取目标环境信息,并根据所述目标环境信息确定目标域名信息;在所述基础安装包生成代码中,将所述目标域名信息填充于所述填充位置,生成与所述目标环境信息匹配的配置化安装包生成代码;执行所述配置化安装包生成代码,生成与所述目标环境信息匹配的安装包。

[0106] 实施例五

[0107] 本发明实施例五提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如本申请所有发明实施例提供的安装包生成方法:获取基础安装包生成代码,以及域名信息在所述基础安装包生成代码的填充位置,所述域名信息与生成的安装包适用的环境信息关联;获取目标环境信息,并根据所述目标环境信息确定目标域名信息;在所述基础安装包生成代码中,将所述目标域名信息填充于所述填充位置,生成与所述目标环境信息匹配的配置化安装包生成代码;执行所述配置化安装包生成代码,生成与所述目标环境信息匹配的安装包。

[0108] 可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0109] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括——但不限于——电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0110] 计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括——但不限于——无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0111] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码,所述程序设计语言包括面向对象的程序设计语言——诸如Java、Smalltalk、C++,还包括常规的过程式程序设计语言——诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0112] 注意,上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解,本发明不限于这里所述的特定实施例,对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此,虽然通过以上实施例对本发明进行了较为详细的说明,但是本发明不仅仅限于以上实施例,在不脱离本发明构思的情况下,还可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。

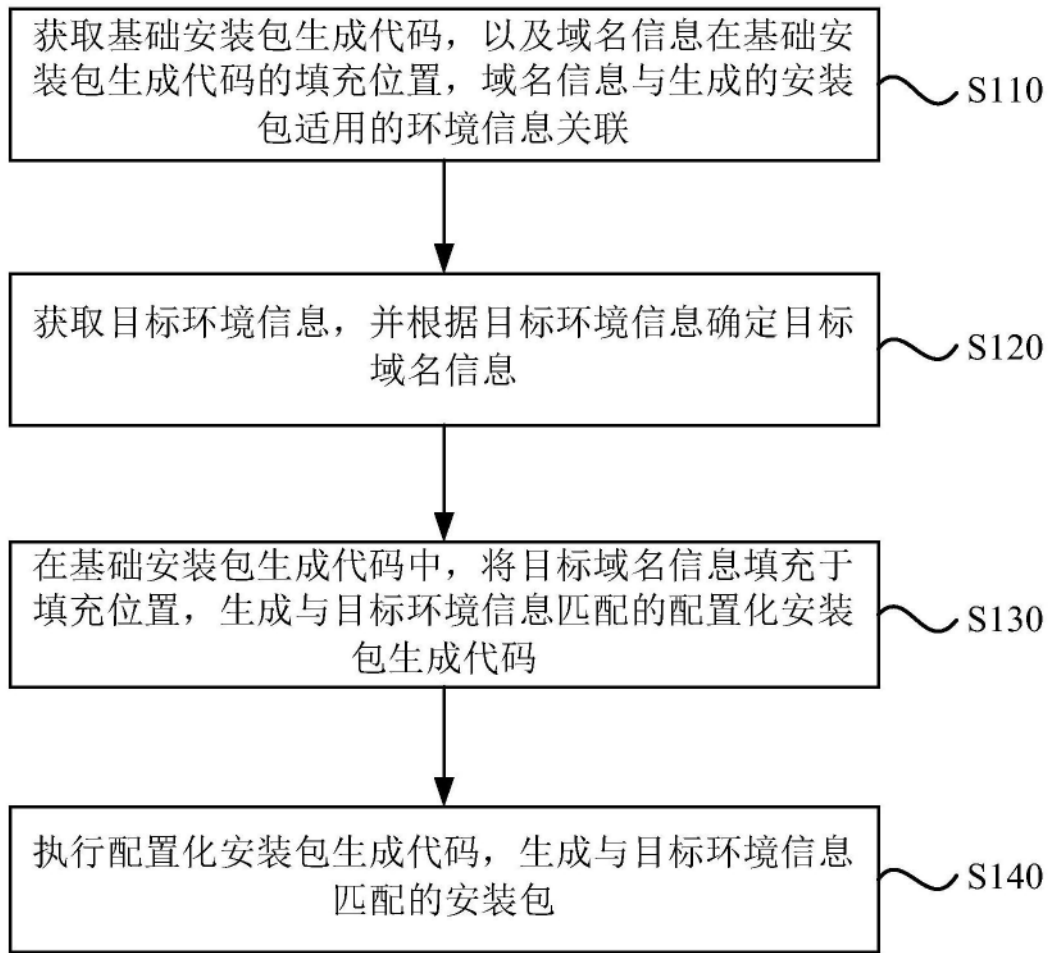


图1

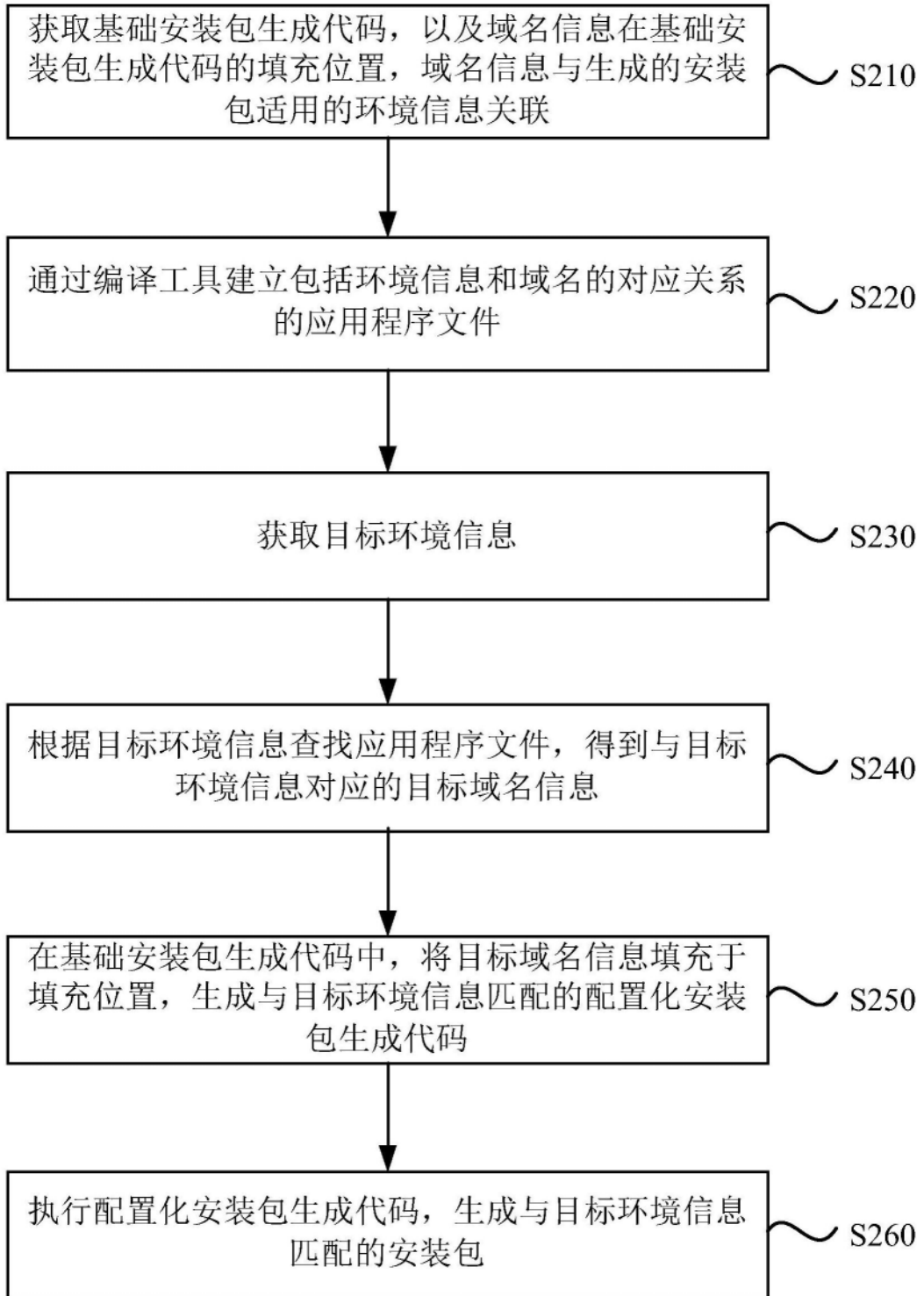


图2A

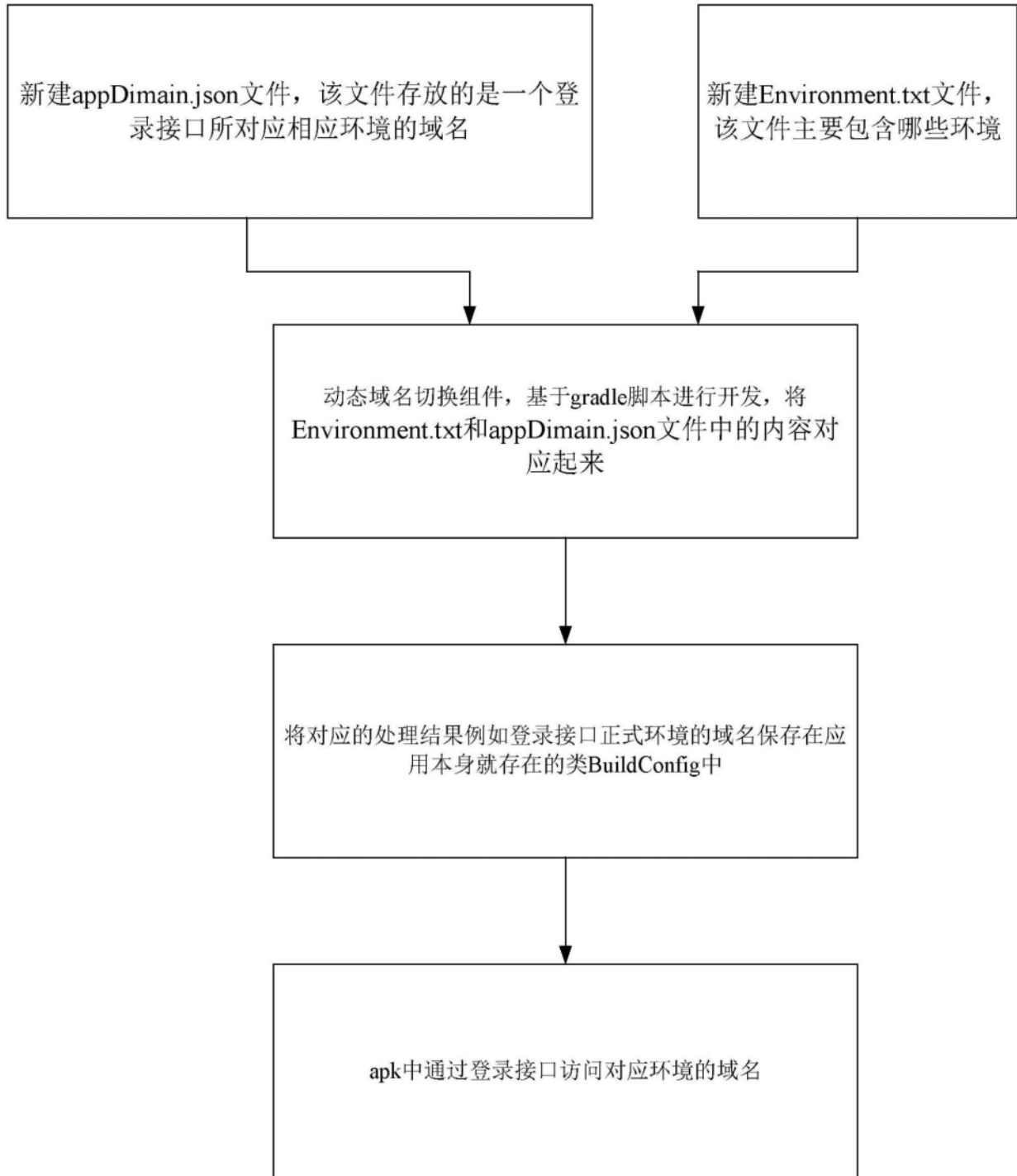


图2B

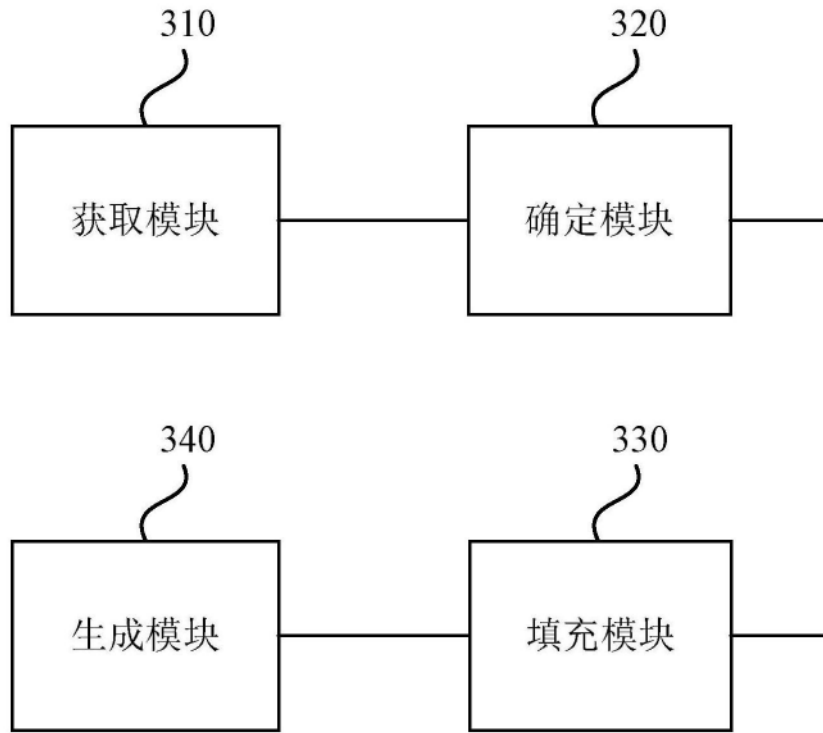


图3

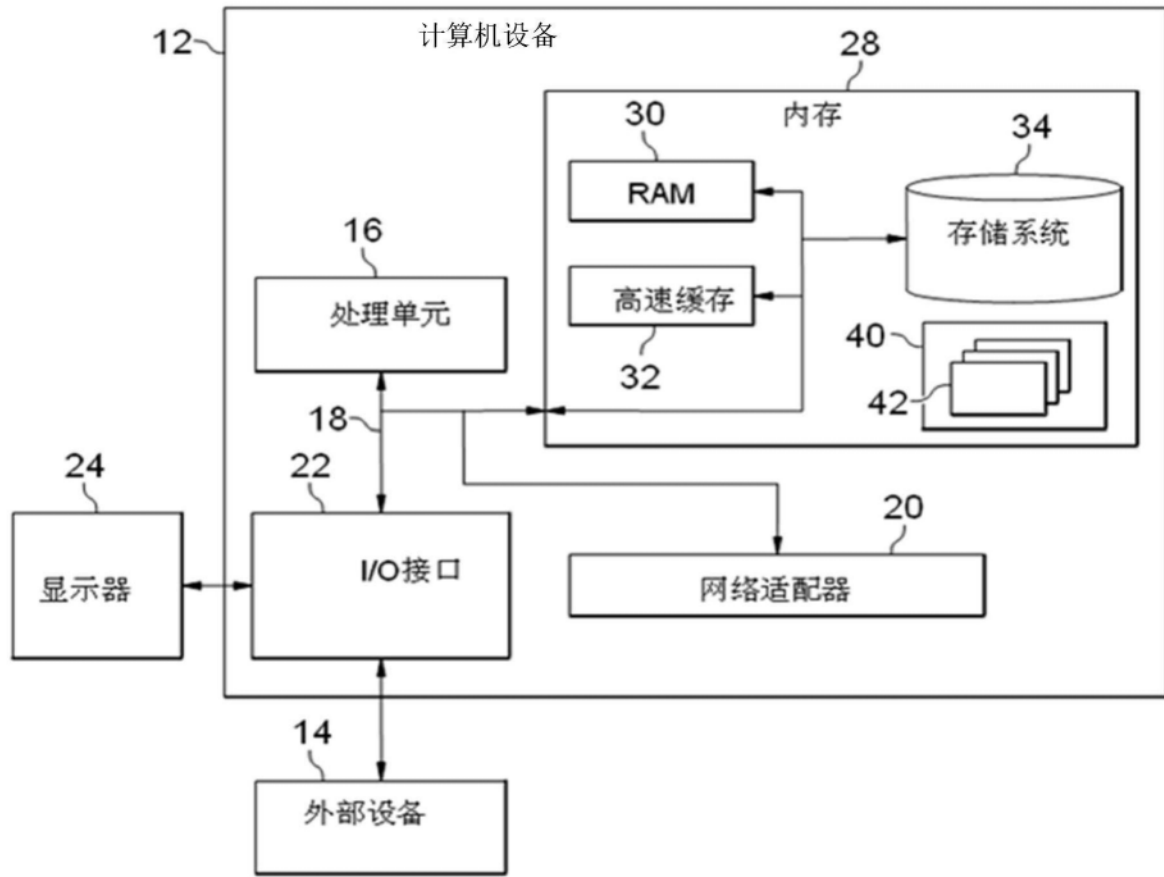


图4