



- (51) **International Patent Classification:**
G06F 9/44 (2006.01) *G06F 15/16* (2006.01)
G06F 21/62 (2013.01)
- (21) **International Application Number:**
PCT/US2013/020519
- (22) **International Filing Date:**
7 January 2013 (07.01.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/350,810 15 January 2012 (15.01.2012) US
- (71) **Applicant:** MICROSOFT CORPORATION [US/US];
One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) **Inventors:** HAUBOLD, Jeremy; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). PACCIARINI, Claudio; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,

DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- with international search report (Art. 21(3))

(54) **Title:** INSTALLATION ENGINE AND PACKAGE FORMAT FOR PARALLELIZABLE, RELIABLE INSTALLATIONS

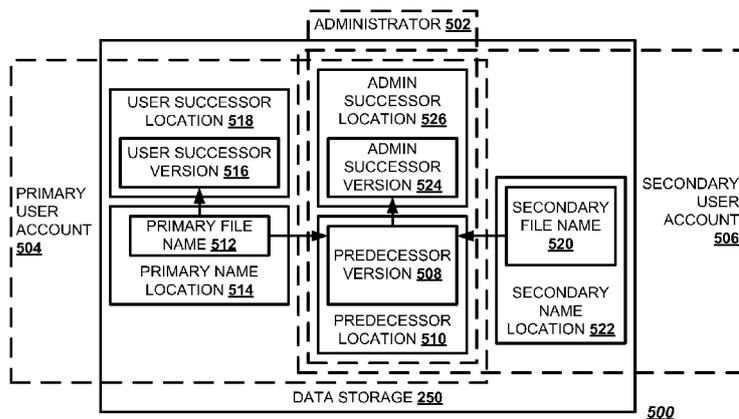


Figure 5

(57) **Abstract:** In one embodiment, an application updater may cleanly update a computer application without causing a fault or a re-boot of either the system or the process. A processor may execute a predecessor version of a computer application. The processor may install a successor version of the computer application on a user account level. The processor may execute the successor version of the computer application.

WO 2013/106276 A1

INSTALLATION ENGINE AND PACKAGE FORMAT FOR PARALLELIZABLE, RELIABLE INSTALLATIONS

BACKGROUND

[0001] A software developer may continually upgrade a computer program on a user
5 device. The user device may contact a server administered by the software developer to
see if upgrades are available. If an upgrade is available, the user device may download the
new sections of software and add those sections to the existing program currently installed
on the user device.

SUMMARY

10 [0002] This Summary is provided to introduce a selection of concepts in a simplified form
that is further described below in the Detailed Description. This Summary is not intended
to identify key features or essential features of the claimed subject matter, nor is it
intended to be used to limit the scope of the claimed subject matter.

[0003] Embodiments discussed below relate to cleanly updating a computer application
15 without causing a fault or a reboot of either the system or the process. A processor may
execute a predecessor version of a computer application. The processor may install a
successor version of the computer application on a user account level. The processor may
execute the successor version of the computer application.

DRAWINGS

20 [0004] In order to describe the manner in which the above-recited and other advantages
and features can be obtained, a more particular description is set forth and will be rendered
by reference to specific embodiments thereof which are illustrated in the appended
drawings. Understanding that these drawings depict only typical embodiments and are not
therefore to be considered to be limiting of its scope, implementations will be described
25 and explained with additional specificity and detail through the use of the accompanying
drawings.

[0005] FIG. 1 illustrates, in a block diagram, one embodiment of a computer network.

[0006] FIG. 2 illustrates, in a block diagram, one embodiment of a computing device.

[0007] FIG. 3 illustrates, in a data flow diagram, one embodiment of an update service.

30 [0008] FIG. 4 illustrates, in a block diagram, one embodiment of an update for a computer
application.

[0009] FIG. 5 illustrates, in a block diagram, one embodiment of a data storage layout.

[0010] FIG. 6 illustrates, in a flowchart, one embodiment of a method for upgrading a
computer application.

DETAILED DESCRIPTION

[0011] Embodiments are discussed in detail below. While specific implementations are discussed, it should be understood that this is done for illustration purposes only. A person skilled in the relevant art will recognize that other components and configurations may be used without parting from the spirit and scope of the subject matter of this disclosure. The implementations may be a machine-implemented method, a tangible machine-readable medium having a set of instructions detailing a method stored thereon for at least one processor, or an application updater for a computing device.

[0012] A computer device may maintain a user account for each user of a computer device. The user account may protect certain data files from access by other users of the computer device. An application updater may update a computer application on a per-user basis in addition to a traditional per-machine basis. The per-user basis may update the computer application for a user account, rather than the other user accounts on the machine. A user account may elevate a user privilege of the user account to install on a machine level.

[0013] Each time the application updater runs, the application updater may extract the update in a new directory and data storage location, thus eliminating any file-in-use problems causing a reboot of the machine. Further, different versions of the computer application may be installed side-by-side without interfering with other versions of the computer application previously installed on the machine, allowing the computer application to update silently, without causing any issues to the smooth performance of the computer application. The application updater may install the same version or a newer version of the computer application as part of an install, reinstall, or repair of the computer application without interfering with the current version. If the available version is an older version, the application updater may execute a null operation to prevent downgrading the computer application. The application updater may install multiple computer applications in parallel.

[0014] Thus, in one embodiment, an application updater may cleanly update a computer application without causing a fault or a reboot of either the system or the process. A processor may execute a predecessor version of a computer application. The processor may install a successor version of the computer application on a user account level. The processor may execute the successor version of the computer application.

[0015] FIG. 1 illustrates, in a block diagram, one embodiment of a computer network 100. A user device 110 may be a desktop computer, a laptop computer, a tablet computer, a

handheld computer, or other computing device. The user device 110 may run a computer application 112. The computer application 112 may be any software program or firmware program executed by the user device 110. The computer application 112 may have a version indicating the iteration of the computer application 112. The computer application 112 may have an updater module 114 that connects to an application server 120 via a data network connection 130. The data network connection 130 may be a local area network connection, an internet connection, a mobile network connection, or other network connection. The updater module 114 may query the application server 120 to see if a new version of the computer application 112 is available. A predecessor version is the version of the computer application 112 currently being executed on the user device. A successor version is the new version of the computer application 112. The updater module 114 may download the successor version to replace the predecessor version.

[0016] FIG. 2 illustrates a block diagram of an exemplary computing device 200 which may act as an application updater. The computing device 200 may combine one or more of hardware, software, firmware, and system-on-a-chip technology to implement an application updater. The computing device 200 may include a bus 210, a processor 220, a memory 230, a read only memory (ROM) 240, a storage device 250, an input device 260, an output device 270, and a communication interface 280. The bus 210 may permit communication among the components of the computing device 200.

[0017] The processor 220 may include at least one conventional processor or microprocessor that interprets and executes a set of instructions. The memory 230 may be a random access memory (RAM) or another type of dynamic storage device that stores information and instructions for execution by the processor 220. The memory 230 may also store temporary variables or other intermediate information used during execution of instructions by the processor 220. The ROM 240 may include a conventional ROM device or another type of static storage device that stores static information and instructions for the processor 220. The storage device 250 may include any type of tangible machine-readable medium, such as, for example, magnetic or optical recording media and its corresponding drive. A tangible machine-readable medium is a physical medium storing machine-readable code or instructions, as opposed to a transitory medium or signal. The storage device 250 may store a set of instructions detailing a method that when executed by one or more processors cause the one or more processors to perform the method.

[0018] The input device 260 may include one or more conventional mechanisms that permit a user to input information to the computing device 200, such as a keyboard, a

mouse, a voice recognition device, a microphone, a headset, etc. The output device 270 may include one or more conventional mechanisms that output information to the user, including a display, a printer, one or more speakers, a headset, or a medium, such as a memory, or a magnetic or optical disk and a corresponding disk drive. The

5 communication interface 280 may include any transceiver-like mechanism that enables computing device 200 to communicate with other devices or networks. The communication interface 280 may include a network interface or a transceiver interface. The communication interface 280 may be a wireless, wired, or optical interface.

[0019] The computing device 200 may perform such functions in response to processor
10 220 executing sequences of instructions contained in a computer-readable medium, such as, for example, the memory 230, a magnetic disk, or an optical disk. Such instructions may be read into the memory 230 from another computer-readable medium, such as the storage device 250, or from a separate device via the communication interface 280.

[0020] FIG. 3 illustrates, in a data flow diagram, one embodiment of an update service
15 300 for a computer application 112. Upon initiation, the current version 302 of the computer application 112 may direct an updater module 114 to request any updates from an application server 120. The application server 120 may send an update package to the updater module 114. The update package may have an update manifest 304 and a current
20 binary 306 of the computer application 112. The update manifest 304 is a list of the files that get installed. The update manifest 304 may describe a file path, a file size, any registry keys, and any error checking mechanisms for each file. The update manifest 304 may be in an extensible markup language (XML) format. The current binary 306 is an executable file or associated file, such as a digital link library, for the computer application
25 112. The updater module 114 may use the update package to create the installer module 308 for installation.

[0021] FIG. 4 illustrates, in a block diagram, one embodiment of an update 400 for a
computer application. An update package 402 sent from an application server 120 may have an installer code 404, a set of neutral resources 406, a set of localizable resources
30 408, and a payload 410. The installer code 404 instantiates the installer module 308 on the user device 110. The set of neutral resources 406 is a set of general resources used during installation, such as icons, non-localizable strings, and non-localizable images to be shown during installation. The set of localizable resources 408 is a set of language specific resources used during installation, such as messages, prompts, and sounds. The payload 410 may be the update manifest 304 and the current binary 306.

[0022] FIG. 5 illustrates, in a block diagram, one embodiment of a data storage layout 500. A user device 100 may support multiple user accounts for multiple users. A user account may be password protected from other users. The data storage device 250 may store multiple application profiles for multiple users. A first user may update a computer application 112 while a second user may forgo the update. An administrator 502 may update the computer application 112 for each user account. A primary user account 504 is the user account that is executing the update of the computer application. A secondary user account 506 is a user account that is not actively executing an update of the computer application. The secondary user account 506 may be active or dormant.

[0023] The data storage 250 may store a predecessor version 508 of a computer application 112 in a predecessor location 510. A primary user account 504 may store a primary file name 512 in a primary name data storage location 514 referencing the predecessor version 508. A file name is a hard link to a data file. The primary file name 512 may be a hard link to the predecessor version 508. The predecessor version 508 may be immutable unless changed by the administrator account 502. The primary user account 504 may install a user successor version 516 of the computer application 112 on a user account level. A user account level installation installs a user successor version 516 for the primary user account 504 but forgoes any update to a secondary user account 506. The primary user account 504 may place the user successor version 516 in a clean user successor data storage location 518. A clean data storage location does not currently have a version of the computer application 112 stored at that location. A data storage location may be considered clean even if a version of the computer application 112 was present at the data storage location and then conventionally erased, such as through deregistering. The user successor version 516 may be a complete successor version of the computer application 112. A complete successor version is a successor version that has not been compared with the predecessor version and not had any overlapping sections removed. The primary file name 512 may be overwritten to be a hard link to the user successor version 516.

[0024] A secondary user account 506 may store a secondary file name 520 of the computer application 112 at a secondary name data storage location 522. The secondary file name 520 may be a hard link to the predecessor version 508. The secondary file name 520 may maintain the hard link to the predecessor version 508 even as the primary file name 512 becomes a hard link to the user successor version 516.

[0025] An administrator account 502 may replace the predecessor version 508 with an administrator successor version 524 of the computer application 112 on a machine level. The administrator account 502 may place the administrator successor version 524 in a clean administrator successor data storage location 524. The operating system may
5 elevate a user status for a user account to an administrative privilege to make updates at a machine level. Updates on a machine level install a successor version to each user account on the user device 110. The administrator account 502 may reset the primary file name 512 and the secondary file name 520 to be hard links to the administrator successor version 524.

10 [0026] FIG. 6 illustrates, in a flowchart, one embodiment of a method 600 for upgrading a computer application 112. The application updater may execute a predecessor version 508 of a computer application 112 (Block 602). The application updater may execute an updater module 114 invisibly to a user (Block 604). The application updater may download a complete successor version 516 using the updater module 114 disregarding
15 the predecessor version 508 (Block 606). If the installation is on a per machine basis (Block 608), the application updater may elevate the user status to an administrative privilege (Block 610). The application updater may install the complete successor version 524 of the computer application 112 on a machine level (Block 612). If the installation is on a per user basis (Block 608), the application updater may install a complete successor
20 version 516 of the computer application on a user account level (Block 614). The application updater may maintain execution of the predecessor version 508 while installation of the complete successor version 516 occurs (Block 616). The application updater may place the complete successor version 516 in a clean successor data storage location 514 (Block 618). The application updater may execute the predecessor version
25 508 and the complete successor version 516 of the computer application 112 simultaneously (Block 620). The application updater may execute a switch from the predecessor version 508 to the complete successor version 516 (Block 622). The application updater may maintain a presentation of a user interface of the computer application 112 during the switch (Block 624). The application updater may schedule the
30 predecessor version 508 for an uninstall operation (Block 626). If the uninstall operation is scheduled during a system reboot (Block 628), the application updater may uninstall predecessor version 508 during the system reboot (Block 630). Otherwise, the application updater may uninstall the predecessor version 508 while executing the complete successor version 516 (Block 632).

[0027] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms for
5 implementing the claims.

[0028] Embodiments within the scope of the present invention may also include non-transitory computer-readable storage media for carrying or having computer-executable instructions or data structures stored thereon. Such non-transitory computer-readable storage media may be any available media that can be accessed by a general purpose or
10 special purpose computer. By way of example, and not limitation, such non-transitory computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to carry or store desired program code means in the form of computer-executable instructions or data structures. Combinations of the above should
15 also be included within the scope of the non-transitory computer-readable storage media.

[0029] Embodiments may also be practiced in distributed computing environments where tasks are performed by local and remote processing devices that are linked (either by hardwired links, wireless links, or by a combination thereof) through a communications network.

[0030] Computer-executable instructions include, for example, instructions and data
20 which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. Computer-executable instructions also include program modules that are executed by computers in stand-alone or network environments. Generally, program modules include routines,
25 programs, objects, components, and data structures, etc. that perform particular tasks or implement particular abstract data types. Computer-executable instructions, associated data structures, and program modules represent examples of the program code means for executing steps of the methods disclosed herein. The particular sequence of such
executable instructions or associated data structures represents examples of corresponding
30 acts for implementing the functions described in such steps.

[0031] Although the above description may contain specific details, they should not be construed as limiting the claims in any way. Other configurations of the described embodiments are part of the scope of the disclosure. For example, the principles of the disclosure may be applied to each individual user where each user may individually

deploy such a system. This enables each user to utilize the benefits of the disclosure even if any one of a large number of possible applications do not use the functionality described herein. Multiple instances of electronic devices each may process the content in various possible ways. Implementations are not necessarily in one system used by all end users.

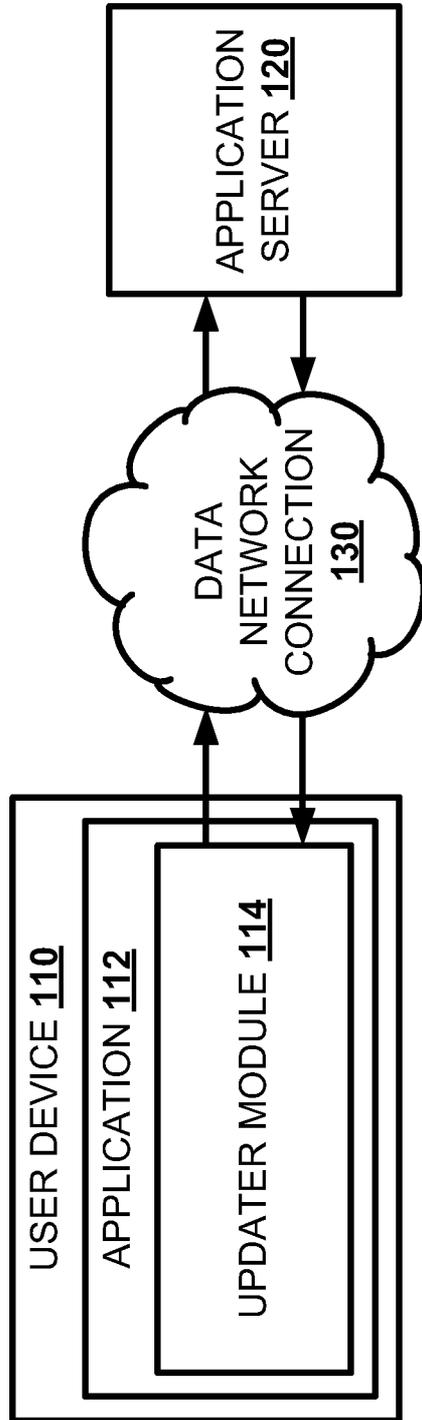
5 Accordingly, the appended claims and their legal equivalents should only define the invention, rather than any specific examples given.

CLAIMS

We claim:

1. 1. A machine-implemented method, comprising:
executing a predecessor version of a computer application;
5 installing a successor version of the computer application on a user account level;
and
executing the successor version of the computer application.
2. The method of claim 1, further comprising:
downloading a complete successor version using an updater module; and
10 executing the updater module invisibly to a user.
3. The method of claim 1, further comprising:
maintaining execution of the predecessor version while installation of the
successor version occurs.
4. The method of claim 1, further comprising:
15 executing a switch from the predecessor version to the successor version;
and
maintaining a presentation of a user interface of the computer application during
the switch.
5. The method of claim 1, further comprising:
20 executing the predecessor version and the successor version simultaneously.
6. The method of claim 1, further comprising:
scheduling the predecessor version for an uninstall operation during a system
reboot.
7. The method of claim 1, further comprising:
25 uninstalling the predecessor version while executing the successor version.
8. A tangible machine-readable medium having a set of instructions detailing a
method stored thereon that when executed by one or more processors cause the one or
more processors to perform the method, the method comprising:
executing a predecessor version of a computer application;
30 downloading a complete successor version using an updater module disregarding
the predecessor version;
placing the complete successor version of the computer application in a clean
successor data storage location.

9. An application updater, comprising:
- a communications interface that downloads a complete successor version of a computer application;
 - a data storage that stores a predecessor version of the computer application in a predecessor data storage location and the complete successor version in a clean successor data storage location; and
 - a processor that executes the predecessor version and installs the complete successor version of the computer application on a user account level.
10. The application updater of claim 9, wherein the processor maintains execution of the predecessor version while installation of the complete successor version occurs.



100

Figure 1

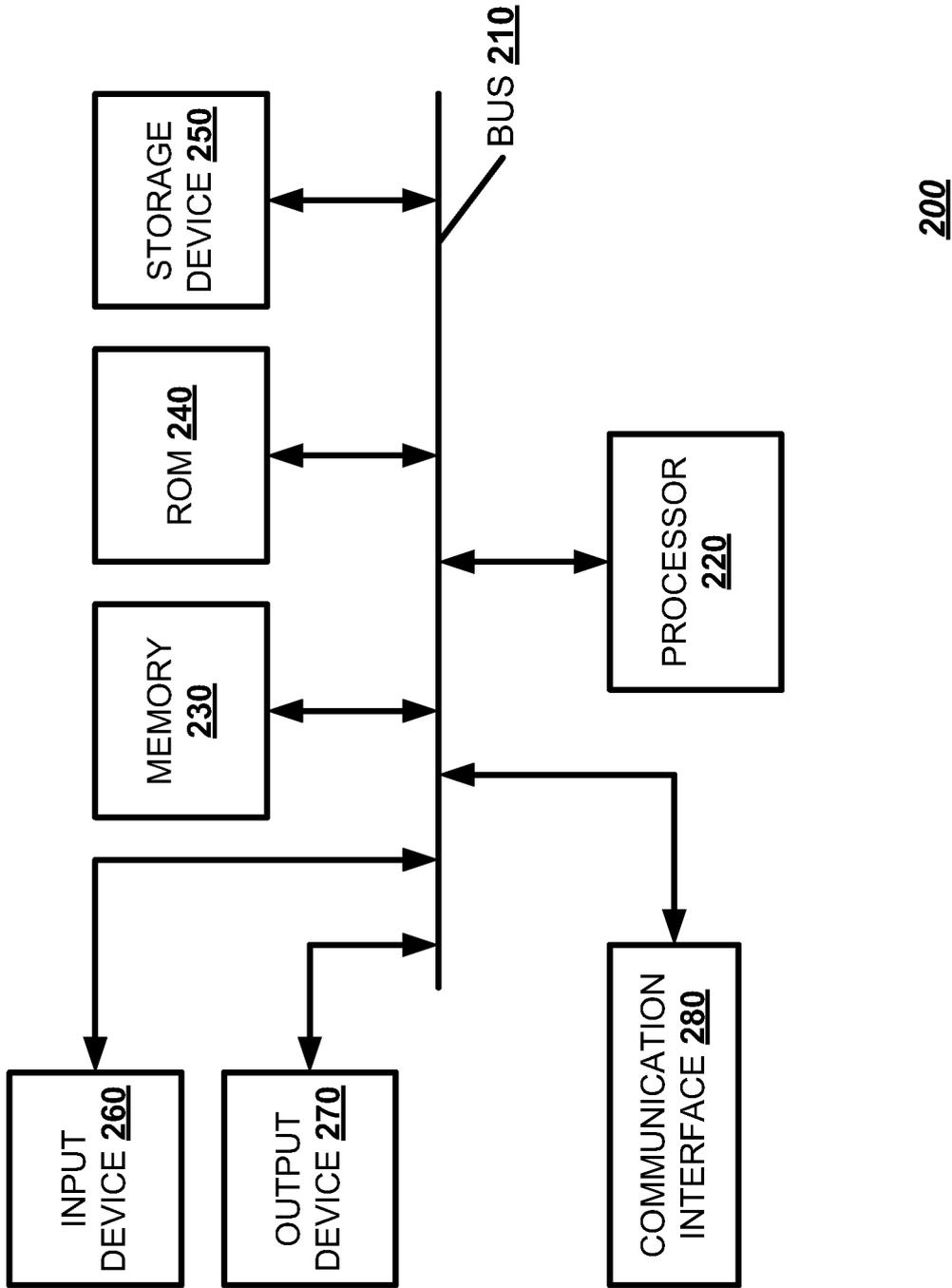
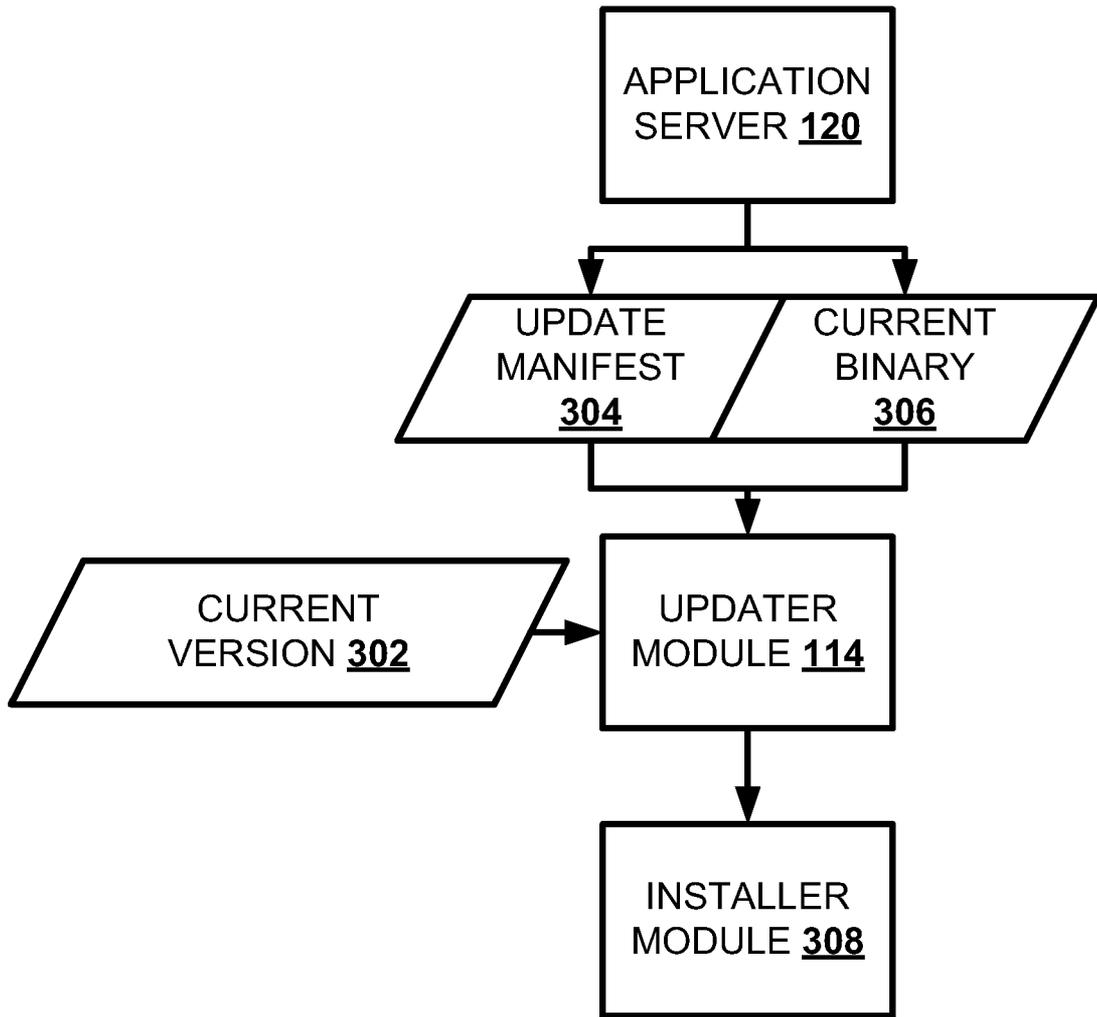
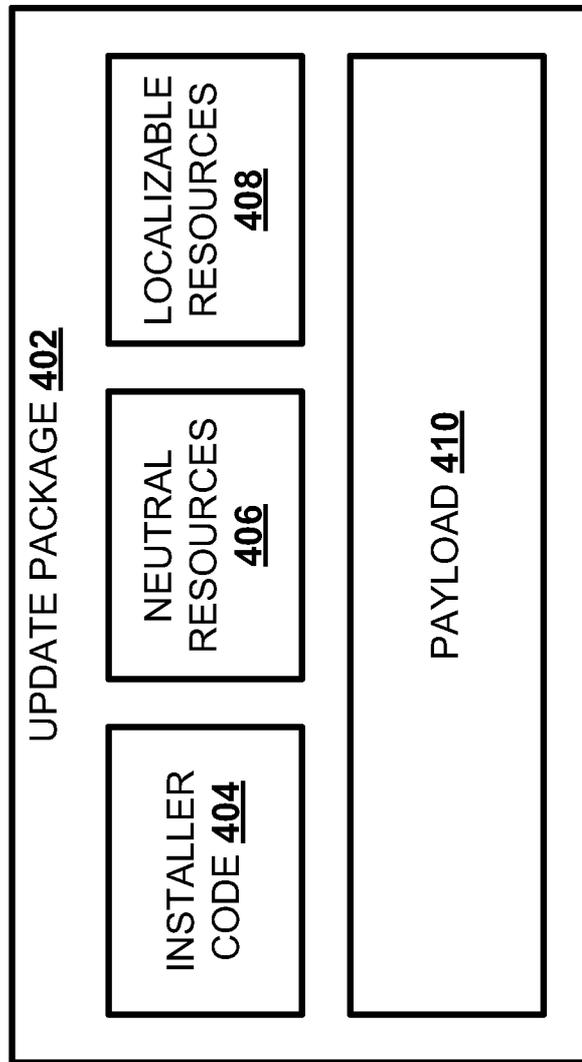


Figure 2



300

Figure 3



400

Figure 4

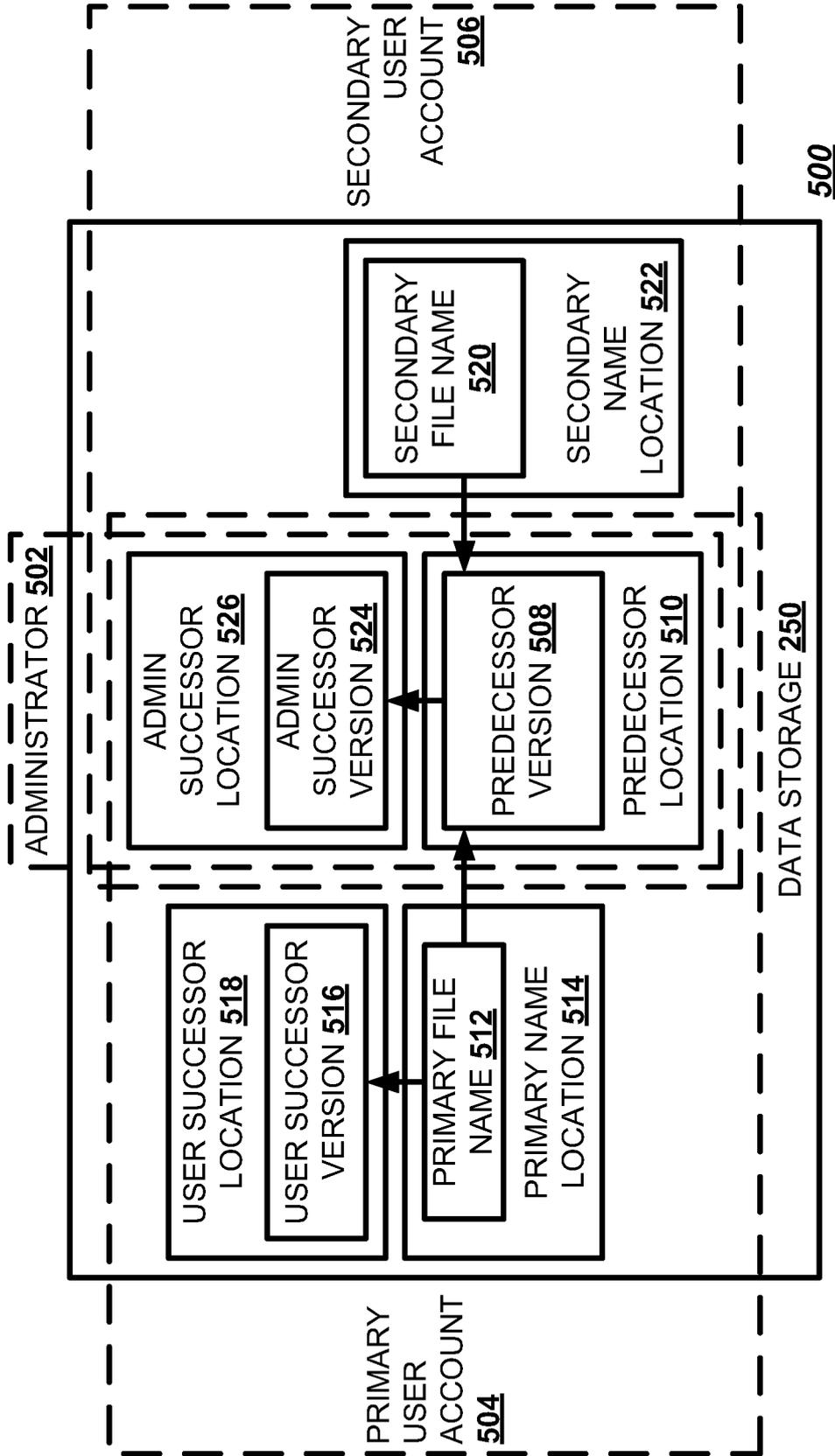


Figure 5

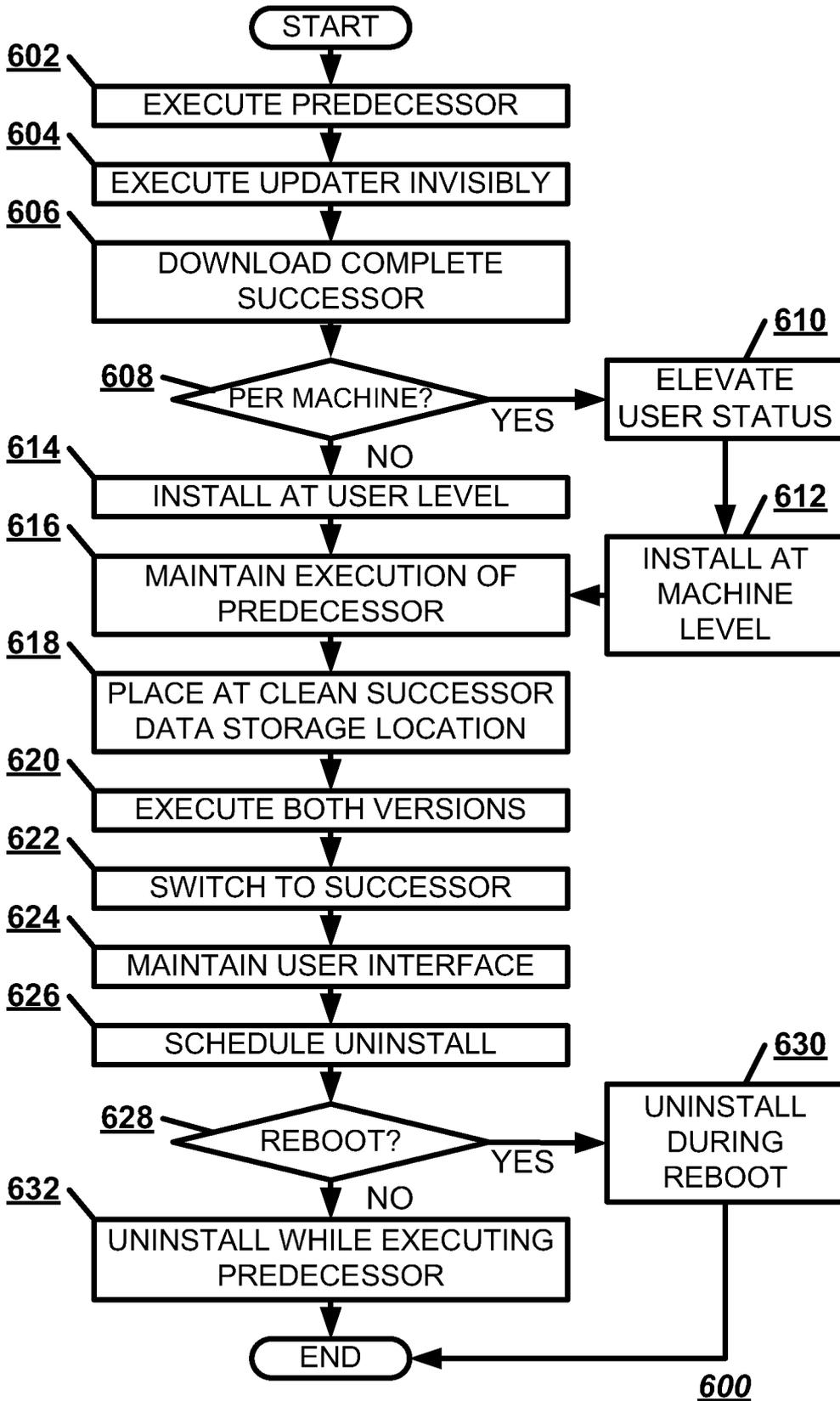


Figure 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/020519**A. CLASSIFICATION OF SUBJECT MATTER****G06F 9/44(2006.01)i, G06F 21/62(2013.01)i, G06F 15/16(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/44; G06K 5/00; G06F 11/30; G06F 9/445; G06F 7/06

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: update, software, program, application, version, user account

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2010-0242037 A1 (XIE, JIANHUI et al.) 23 September 2010 See paragraphs [0042]-[0043], [0057]-[0066], [0094]-[0110] and figures 1, 3.	1,2,7
A		3-6,8-10
A	US 2006-0086785 A1 (SAKATA, YASUJI) 27 April 2006 See paragraphs [0009], [0117]-[0134] and figures 10, 11.	1-10
A	US 2004-0003266 A1 (MOSHIR, SEAN et al.) 01 January 2004 See paragraphs [0020]-[0032], [0061]-[0086].	1-10
A	US 5999740 A (ROWLEY, DAVID JOHN) 07 December 1999 See column 5, line 23 - column 6, line 67 and figures 3A, 3B.	1-10

 Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

11 April 2013 (11.04.2013)

Date of mailing of the international search report

12 April 2013 (12.04.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan
City, 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

JANG, Ho Keun

Telephone No. 82-42-481-8187



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/020519

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010-0242037 A1	23.09.2010	None	
US 2006-0086785 A1	27.04.2006	CN 1763716 A CN 1763716 C0 EP 1650717 A1 JP 2006-119901 A KR 10-2006-0051023 A SG 121941 A1	26.04.2006 26.04.2006 26.04.2006 11.05.2006 19.05.2006 26.05.2006
US 2004-0003266 A1	01.01.2004	AU 2001-94677 A1 AU 2004-222883 A1 CA 2423175 A1 CA 2517223 A1 CN 1894661 A CN 1894661 C0 EP 1327191 A1 EP 1611509 A2 JP 2006-520975 A KR 10-2005-0120643 A US 2002-0100036 A1 US 2005-0257214 A1 US 2011-0029966 A1 US 6990660 B2 US 7823147 B2 US 8407687 B2 WO 02-25438 A1 WO 2004-086168 A2 WO 2004-086168 A3	02.04.2002 07.10.2004 28.03.2002 07.10.2004 10.01.2007 10.01.2007 16.07.2003 04.01.2006 14.09.2006 22.12.2005 25.07.2002 17.11.2005 03.02.2011 24.01.2006 26.10.2010 26.03.2013 28.03.2002 07.10.2004 07.10.2004
US 5999740 A	07.12.1999	AU 1997-44368 B2 EP 0841615 A2 EP 0841615 A3	12.10.2000 13.05.1998 14.05.2003