

METHOD, SYSTEM AND SERVER FOR COLLECTING SOFTWARE VERSIONS

Technical Field

5 The present invention relates to the computer technology, and particularly to a method, a system and a server for collecting software versions.

Background

With the development of the computer technology, electronic devices, such personal computers, personal digital assistances (PDA), intelligent handsets, Internet televisions and so on, are installing more and more software applications. The software applications
10 are used in more and more fields, such as instant messaging, audio/video playing, resources downloading, web browsing, texting, system assistance and so on. As the Internet becomes more and more popular, ordinary users now generally turn to the Internet for obtaining software applications which were distributed via compact disks ten years ago, which is more convenient for users to obtain software. Meanwhile, the interval
15 between two updates of a software application is shrinking from a year of half a year in ten years ago to one month or a couple of days now. A new version of a software application may provide new functions, and patching vulnerabilities and defects in older versions. Using a new version of a software application, a user can experience new functions of the software and can avoid security risks brought by known vulnerabilities of
20 the software. Therefore, currently there is management software for managing software applications installed by a user, helping the user obtain the version of the installed software applications, and detecting whether an installed software applications has a new version.

Conventional management software detects whether a software application needs an
25 update by storing information of latest versions and corresponding description information of various software applications at a server device of the management software. Operating personnel obtains the latest versions of the various software applications from respective official websites or from websites providing software downloading. When a software application is updated, the operating personnel inputs the
30 latest version and corresponding description information of the software application into a

software update configuration system at the server device of the management software. Thus, a client of the management software is able to obtain knowledge of the latest versions of various software applications from the server of the management software.

5 The above manner of detecting software updates generates high costs, because there are tens of thousands of software applications needed to be maintained at the server of the management software and a large number of operating persons are needed for manually collecting the updated versions of software applications.

Summary

10 Therefore, embodiments of the present invention provide a method of collecting software versions which can obtain information of new software versions automatically.

Embodiments of the present invention also provide a server which is able to obtain information of new software versions automatically.

Embodiments of the present invention still provide a system of collecting software versions which can obtain information of new software versions automatically.

15 A method of collecting software versions may include:

receiving, by a server, information of a first version of a software installed in a client device sent by a client;

judging whether the first version is newer than a second version based on the information of the first version and information of the second version of the software
20 recorded in the server;

updating the information of the second version recorded in the server in response to a determination that the first version is newer than the second version.

A server may include:

a first module, for receiving information of a first version of a software installed in a
25 client device sent by a client, and provides the information for a second module;

a second module, for judging whether the first version is newer than a second version based on the information of the first version received by the first module and information of the second version of the software recorded in a third module; updating the

information of the second version recorded in the third module in response to a determination that the first version is newer than the second version.

A system for collecting software versions may include a server and a client; and the server is for:

5 receiving information of a first version of a software installed in a client device sent by a client;

judging whether the first version is newer than a second version based on the information of the first version and information of the second version of the software recorded in the server;

10 updating the information of the second version recorded in the server in response to a determination that the first version is newer than the second version.

Brief Description of the Drawings

Fig. 1 is a flowchart illustrating a method of a client collecting software versions.

Fig. 2 is a flowchart illustrating a method of detecting software updates.

15 Fig. 3 is a flowchart illustrating a working process of a client.

Fig. 4 is a block diagram illustrating a structure of a software version management system and a software updates detecting system.

Embodiments of the Invention

In order to solve the problem of high cost and time consuming resulted from
20 collecting software versions manually, embodiments of the present invention provide a method for collecting software versions. The method mainly includes:

receiving, by a server, version information of a software installed in a client device sent by a client;

judging whether a first software version in the client device is newer than a second
25 software version recorded in the server based on the received version information and information of the second software version recorded in the server;

recording the received version information in response to a determination that the software version in the client device is newer than the software version recorded in the server.

Information of a version refers to information which can be used for determining the version is newer or older than another version. The version information may generally be
5 a version number, or the release time of the version, or the name of the version and so on, and is not limited in the present invention.

The above method can automatically get knowledge of a new version of a software. Because there are always some users who download a new version of a software and
10 update the software shortly after the new version is released by the software publisher or set their software at an automatic update mode, there are always some clients in the network that use the latest version of a software. By collecting version numbers of the software from clients, the server device of the management software can identify a new version of a software, and inform clients of the management software of the new version.
15 Then the clients of the management software may prompt information suggesting the users to update the software. The whole process is implemented by computers without involving any manual work, thus greatly improves the efficiency of obtaining updated software versions.

The above method may be used for obtaining version information of a software
20 specified by a server, or for obtaining version information of all software used in daily life and installed in clients, including web browsing software, file downloading software, office software, anti-virus software, media playing software, game software and so on.

The above is the work process of the server device. The above server may be a stand alone device, or a module in another device; may reside in one device or may be
25 distributed in several devices, thus it is also referred to as a server side device, server end device or server end of the management software, which is from the perspective of the client. The present invention does not limit the implementation manner of the server side device or the number of the server devices.

An embodiment of the present invention provides a method for detecting software
30 updates. Detection of software updates refers to detecting whether a software has an updated version, and is an example of the above process of collecting software versions.

The method may include the following procedures: receiving a client side version number of a software in a software list of the client; judging whether the client side version number represents a version newer than a version represented by a server side version number of the software; updating the server side version number if the client side version number represents a version newer than the version represented by the server side version number.

The client side version number refers to the version number of the software installed in the client device; the server side version number refers to the version number of the software recorded in the server.

Specifically, the procedure of judging whether the client side version number represents a version newer than a version number represented by the server side version number may include: collecting statistics of the client side version numbers, taking a version number which represents a version newer than a version represented by the server side version and the number of which exceeds a threshold as the client side version number which represents a version newer than a version represented by the server side version number.

An embodiment of the present invention also provides a procedure of pushing the server side version number to a client. Specifically, it is judged whether the server side version number is of a version newer than a version corresponding to the client side version number received from the client, and the server side version number is sent to the client if the server side version number is of a version newer than a version corresponding to the client side version number.

Accordingly, an embodiment of the present invention provides a system for detecting software updates. The system may include:

a server side sending/receiving unit, for receiving a client side version number of a software in a list of software in the client;

a storage unit, for storing a server side version number;

a processing unit, connected with the server side sending/receiving unit and the storage unit, for judging whether the client side version number is of a version newer than a version corresponding to the server side version number of the software, updating the

server side version number in the storage unit if the client side version number is of a version newer than the version corresponding to the server side version number.

The processing unit may collect statistics of client side version numbers of clients, and take a client side version number whose amount exceeds a threshold and which is of a version newer than a version corresponding to the server side version number as the client
5 side version number of a version newer than the version corresponding to the server side version number. By collecting the statistics, only the client side version number whose amount is larger than a threshold is regarded as a new version to avoid an incorrect version or an unstable beta version being regarded the new version, which can thus
10 increase the stability of the software provided for the user.

The processing unit may further judge whether the server side version number is of a version newer than a version corresponding to a client side version number received from a client, and sending the server side version number to the client if the server side version number is of a version newer than the version represented by the client side version
15 number.

Below is an exemplary processing flow of a client according to an embodiment of the present invention. The client, which is also referred to as the client of the management software, may be implemented by hardware modules or by software. A physical device where the client resides and/or a physical device accessible by the client directly or
20 indirectly via wired or wireless connections can be regarded as the client device of the present invention. Throughout the description, the so called client may sometimes refer to the client for collecting software versions, and sometimes refer to a client device. But from the context it is clear that the entity performing the software version collection is the client for collecting software versions, while information such as the list of software, the
25 software version number and so on is obtained from the device client.

A method for collecting software versions may include: obtaining a client side version number of a software in a list of software in the client, sending the client side version number to the server.

The client side version number refers to the version number of the software installed
30 in the client device. The list of software in the client refers to a list formed by the client with information of one or multiple software installed in the client. The list of software

may include one or multiple software, i.e., the client may select one or multiple software from all software installed in the client device to form the list of software.

5 The client side version number may be obtained from a DisplayVersion key value of a software uninstall node in a registry, or from executable resources of the main program of the software, or from a configuration file in an installation directory of the software. It is very simple and convenient to obtain the client side version number from the DisplayVersion key value of software uninstall node in a registry, or from executable resources of the main program of the software, or from a configuration file in an installation directory of the software.

10 A list of software installed in the client may also be obtained.

The software list of the client may be obtained from a software uninstall list of the registry, or from desktop shortcuts, or from the application list in Windows start menu, or from shortcuts in Windows quick launch toolbar.

15 An embodiment of the present invention provides a system for managing software versions which can obtain an updated version number of a software. The system may include:

a version number obtaining unit, for obtaining a client side version number of a software in a list of software installed in a client;

20 a client side sending/receiving unit, connected with the version number obtaining unit, for sending the client side version number to a server side device.

The client side version number may be obtained from a DisplayVersion key value of a software uninstall node in a registry, or from executable resources of the main program of the software, or from a configuration file in an installation directory of the software.

25 The system may further include a software list obtaining unit which is connected with the version number obtaining unit and is for obtaining a list of software installed in the client.

The software list obtaining unit may obtain the software list of the client from a software uninstall list of the registry, or from desktop shortcuts, or from the application list in Windows start menu, or from shortcuts in Windows quick launch toolbar.

The system may further include a judging unit and an updating unit. The client side sending/receiving unit may further be for receiving a server side version number; the judging unit is connected with the client side sending/receiving unit and is for judging whether the server side version number is of a version newer than a version represented by the client side version number. The updating unit is connected with the judging unit, for updating the software automatically or providing prompt information for the user and updating the software according to a user instruction when the judging unit determines the server side version number is of a version newer than the version represented by the client side version number. By judging whether the server side version number is of a version newer than the version represented by the client side version number, prompting the user when the server side version number represents a version newer than the version represented by the client side version number, and updating the software automatically or providing prompt information for the user and updating the software according to a user instruction, the updating process is enabled to be performed either at the background or according to the needs of the user, which is more convenient for the user.

The above system illustrates a structure of the client side.

The above method and management for collecting software versions, and the method and the management system for detecting software updates determine whether a software has an updated version by making the client reporting the client side version number of the software or by receiving the client side version number sent by the client, and do not need manual work for collecting and inputting, thus speed up the process of obtaining an updated version of a software.

Embodiments of the present invention will be described in detail with reference to the following examples.

Fig. 1 is a flowchart illustrating a method of a client collecting software versions. As shown in Fig.1, the process may include the following procedures.

In block S110, a list of software installed in a client device is obtained.

The software list of the client device may be obtained from a software uninstall list of the registry, or from desktop shortcuts, or from the application list in Windows start menu, or from shortcuts in Windows quick launch toolbar, and so on. For example, the location of the registry may be:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall.
1.

The client may obtain information of all software installed in the client device to form the software list, or may obtain information of only some of the software for generating the software list. The manner adopted may be decided by pre-set configuration parameters or configuration information obtained from the server side device at fixed intervals or each time when the client starts up. The configuration parameters may be: forming the software list based on one or multiple pre-designated software, or forming the software list based on all software installed in the client device, or forming the software list based on software most frequently used. The time and the number of times a software is visited can be obtained from statistics collected by certain statistical software in the client device, or may be collected by the client. The configuration information obtained from the server side device may be a list in the server which lists all software whose information needs to be collected by the server. After obtaining the list, the client may obtain information of the software listed in the list from the client device to form the software list.

If the client of the management software has already had a list of software whose information needs to be collected configured in advance, the procedure in this block can be omitted. Software in the software list may include IM software, audio/video playing software, resources downloading software, web browsing software, texting software and so on. The time to obtain the software list of the client may be determined according to pre-set parameters, e.g., at pre-set intervals, or every time when the client device is started up, or every time when the client of the management software is started up, and so on.

In block S120, a client side version number of a software in the software list of the client is obtained. The client side version number may be obtained from a DisplayVersion key value of a software uninstall node in a registry, or from exe resources (executable resources) of the main program of the software, or from a configuration file in an installation directory of the software. In the registry of a Windows system, the DisplayVersion key value of a software uninstall node records the version number of the software. A version tag in attributes of exe resources of the main program also records the version number of the software. The configuration file in an installation directory of the software may generally records the version number of the software.

In block S130, the client side version number is sent to a server side device. After the client side version number is collected, the client side version number of each software may be sent to the server side device together with the name of the software and/or codes of the software. The client may send the client side version number immediately after the client side version number is obtained, or at pre-set time, or according to network conditions (e.g. when the network is not busy).

The client side version number is reported by the client without manual collecting and inputting work, thus the process of obtaining an updated version of a software is less time-consuming.

Fig. 2 is a flowchart illustrating a method of detecting software updates. As shown in Figure 2, the method may include the following procedures.

In block S210, a client side version number of a software in a software list is received from a client. The client side version number and a name and/or codes of the software are received via a network. The server may store the received information, or may analyze the received information and update statistics stored in the server. For example, the server may count the number of times for which information of a certain version of a software is received, the time when the information is received, an identification of the client sending the information, an identification of the client device, and so on.

In block S220, it is judged whether the client side version number is of a version newer than a version corresponding to a server side version number of the software. Specifically, the server may compare the version number of the software received from the client with the version number of the software stored in the server to determine whether the received version is more recent.

The server may compare the client side version number with the server side version number on a one-to-one basis, and determine the client side version is newer than the server side version if the client side version number is larger than the server side version number. According to another embodiment, the processing unit may collect statistics of client side version numbers of the software in all clients, and take a client side version number which is received for a number of times which exceeds a threshold and which is of a version newer than the version corresponding to the server side version number as

the client side version number of a version newer than the version corresponding to the server side version number. By collecting the statistics, only the client side version whose amount is larger than a threshold is regarded as a new version to prevent an incorrect version number or an unstable beta version from being regarded as the new version, which can thus increase the stability of the software provided for the user. For example, the version number of the latest version of a software is 5.0, and a total of 1 million data items about version numbers of the software are received. Among the received data items, a client side version number of 6.0 is received for 10 times (this version may be a beta version in test of the software publisher). Since the 10 data items forms a small proportion of the total number of data items, the 10 data items can be filtered out by using a threshold. For example, the threshold may be set as 128, and the version number 6.0 is only regarded as a new version when the number of received data items of version number 6.0 exceeds the threshold of 128. The threshold may also be set to reflect a specified proportion of the number of a received version number to the total number of all received data items, e.g., may be 5%, 1/10 and so on. The calculation may be performed at pre-defined intervals, e.g., once every day, once every three days, or once in a week and so on, and may also be performed when the total number of received data items reaches a specified value. For example, the calculation may be performed when every 10000 items of software version information sent by clients are received. When calculating, the server may perform repetition removal for received version information from clients, i.e. to delete version information received from a client from which the same version information has been received previously to only save information received from different clients. The data collected may be cleared after the calculation is performed according to pre-set configurations in the server, or be accumulated according to a pre-set policy. The manner of collecting the statistics may be designed according to practical needs, and is not limited in the present invention.

The above examples takes version numbers in the conventional form (i.e. Arabic numerals with a radix point) as an example. In other examples, the version numbers may be represented in other forms such as including letters or even symbols.

In block S230, the server side version number is updated in response to a determination that the client side version number represents a version newer than that represented by the server side version number. Through the updating procedure, the

version number recorded in the server side device of the management software is modified to be the new version number to keep the status of the software recorded at the server end stay up-to-date.

5 In block S240, the server side version number is sent to a client side device. The server side version number of the software may be sent to a client via the network together with the name or codes of the software. Since the version number recorded at the server side stays up-to-date after the above described procedures, the server side version number sent to the client may be used for informing the user and prompting the user to update and synchronize the software installed in the client.

10 In other examples, procedures in this block may be omitted. Instead, the client may send an inquiry request, and the server responds to the request with the server side version number. In addition, in other examples, the procedures in this block may alternatively include: judging whether the server side version number is of a version newer than the version represented by the client side version number received in block S210, and sending
15 the server side version number to the client in response to a determination that the server side version is newer than the client side version. According to this example, when there are mass amounts of software and clients, the clients can be informed of which software has updated versions even if the client does not inquire the server, and at the same time there is no need to transmit a large quantity of data. The above procedure of
20 the server judging whether the software version stored in the server is newer than the software version received from the client may be performed next time when the software version information is received from the client, or be performed after the server recorded software versions sent by clients and updated the version information. Alternatively, the server may record a relation which associates a version of a software with an identification of a client which reported information of the version. For example, a
25 version 4.0 is correlated with clients 1, 2, 3, and a version 5.0 is correlated with clients 4, 5 and 6; after the server updated the version information, the server may identify identifications of clients which is associated with an old version, and sends information of the new version to the clients. In order to reduce the workload of the server, the server
30 may send information of the new version to all clients, then the client judges whether the version in the client device is the most recent version and prompts the user that there is a new version of the software.

After a large number of reports of client side version numbers are received, it can be judged whether there is a new version based on the large amount of reports of client side version numbers. Because there are always some users who would download and update to a new version of a software shortly after the new version is released by the software publisher or who set their software at an automatic update mode, there are always some clients in the network that use the latest version of a software. By collecting version numbers of the software from clients, the server of the management software can identify a new version of a software.

Fig. 3 illustrates a work flow of a client when the server sends a server side version number to the client on the server's own initiative or when the client sends an inquiry request and the server side device responds to the request with a server side version number. Fig. 3 is a flowchart illustrating a work process of a client. The process may include the following procedures.

In block S310, a server side version number is received.

In block S320, it is judged whether there is a new version. The client judges whether the server side version number is of a version newer than a version represented by a client side version number, and the procedure in block 330 is performed in response to a determination that the server side version is newer than a client side version, or the process is ended in response to a determination that the server side version is not newer than the client side version.

In block S330, the user is prompted to update or automatic update is performed. For example, a prompt box is popped up to inform the user there is a new version available and ask the user whether to update, and the software is updated when the user chooses to update. Or, the prompt information may not be displayed for the user but stored in the local device, and provided for the user when the user sends an inquiry request.

By judging whether the server side version is newer than the client side version, prompting the user when the server side version is newer than the client side version, and updating the software automatically or providing prompt information for the user and updating the software according to a user instruction, the updating process is performed either at the background or according to the needs of the user, which is more convenient for the user.

In addition, a system for managing software versions at the client side and a system for detecting software updates at the server side are provided, and are as shown in Fig. 4.

5 The system for managing software versions may include a software list obtaining unit, a version number obtaining unit, a client side sending/receiving unit, a judging unit and an updating unit.

The software list obtaining unit is for obtaining a list of software installed in a client. The software list obtaining unit may obtain the software list from a software uninstall list of the registry, or from desktop shortcuts, or from an application list in a Windows start
10 menu, or from shortcuts in Windows quick launch toolbar.

The version number obtaining unit is connected with the software list obtaining unit, for obtaining a client side version number of a software in the software list of the client. The client side version number may be obtained from a DisplayVersion key value of a software uninstall node in a registry, or from executable resources of the main program of
15 the software, or from a configuration file in an installation directory of the software.

a client side sending/receiving unit, connected with the version number obtaining unit, for sending the client side version number to a server side device.

The client side sending/receiving unit may be further for receiving a server side version number. The judging unit is connected with the client side sending/receiving unit,
20 for judging whether the server side version number is of a version newer than a version corresponding to the client side version number. The updating unit is connected with the judging unit, for prompting a user when the judging unit determines that the server side version is newer than the client side version and performing update according to an instruction of the user or performing automatic update.

25

The system for detecting software updates may include a server side sending/receiving unit, a storage unit and a processing unit.

The server side sending/receiving unit is for receiving a client side version number of a software in a list of software installed in the client.

30 The storage unit is for storing a server side version number.

The processing unit is connected with the server side sending/receiving unit and the storage unit, for judging whether the client side version number is of a version newer than a version corresponding to the server side version number of the software, updating the server side version number in the storage unit if the client side version is newer than the server side version. According to another embodiment, the processing unit may collect statistics of client side version numbers of the software in all clients, and take a client side version number whose information has been received for a number of times which exceeds a threshold and which is of a version newer than the version corresponding to the server side version number as the client side version number of a version newer than the version corresponding to the server side version number. The processing unit may further judge whether the server side version number is of a version newer than a version corresponding to a client side version number received from a client, and sending the server side version number to the client if the server side version is newer than the client side version.

The above embodiments are merely examples of implementation of the present invention, and may include unnecessary details, thus should not be used for limiting the protection scope of the present invention. It should clear for those skilled in the art that any modification and improvement within the principle of the invention should be covered in the protection scope of the invention. Therefore, the protection scope of the present invention should be determined by the extent defined by the indispensable technical features stated in the claims.

Claims

1. A method for collecting software versions, comprising:

receiving, by a server, information of a first version of a software installed in a client device sent by a client;

5 judging whether the first version is newer than a second version based on the information of the first version and information of the second version of the software recorded in the server;

updating the information of the second version recorded in the server in response to a determination that the first version is newer than the second version.

10 2. The method of claim 1, further comprising:

judging whether the information of the first version has been received for a number of times which exceeds a first threshold when the first version is newer than the second version, performing the step of updating the information of the second version recorded in the server when the information of the first version has been received for a number of
15 times which exceeds the first threshold; or

judging whether a proportion of the number of times the information of the first version has been received to the total number of times information of versions of the software has been received exceeds a second threshold when the first version is newer than the second version, performing the step of updating the information of the second
20 version recorded in the server when the proportion of the number of reception of the information of the first version to the total number of recorded reception of the information of versions of the software exceeds the second threshold.

3. The method of claim 2, wherein:

the number of times the first version has been received is the number of times the
25 first version was received within a pre-defined period of time.

4. The method of claim 2, further comprising:

judging whether the information of the first version has been received from the client after receiving the information of the first version from the client, performing no modification to the number of times the information of the first version has been received
30 when the information of the first version has been received from the client.

5. The method of claim 1, further comprising:

providing recorded information of the second version for the client.

6. The method of claim 1, further comprising: obtaining, by the client, information of the first version of the software installed in a client device;

5 sending the information of the first version to the server.

7. The method of claim 6, wherein the information of the first version is obtained from a DisplayVersion key value in a software uninstall node in a registry, or from executable resources of a main program or from a configuration file in an installation directory of the software.

10 8. The method of claim 6, further comprising: obtaining a list of software installed in a client device, performing the step of obtaining information of the first version of the software installed in the client device for each software in the list of software.

9. The method of claim 8, wherein the list of software is obtained from a software uninstall list in a registry, or from desktop shortcuts, or from an application list in a
15 Windows start menu, or from shortcuts in a Windows quick launch bar.

10. A server, comprising:

a first module, for receiving information of a first version of a software installed in a client device sent by a client, and provides the information for a second module;

a second module, for judging whether the first version is newer than a second
20 version based on the information of the first version received by the first module and information of the second version of the software recorded in a third module; updating the information of the second version recorded in the third module in response to a determination that the first version is newer than the second version.

11. The server of claim 10, wherein:

25 the second module is for judging whether the number of times the information of the first version has been received exceeds a first threshold in response to a determination that the first version is newer than the second version, performing the step of updating the information of the second version recorded in the third module when the number of times the information of the first version has been received exceeds the first threshold; or
30 judging whether a proportion of the number of times the information of the first version

has been received to the total number of times information of versions of the software has been received exceeds a second threshold in response to a determination that the first version is newer than the second version, performing the step of updating the information of the second version recorded in the third module when the proportion of the number of reception of the information of the first version to the total number of recorded reception of the information of versions of the software exceeds the second threshold.

12. The server of claim 11, wherein:

the second module is for comparing the number of times the information of the first version has been received within a pre-defined period of time with the first threshold.

10 13. The server of claim 11, wherein:

the first module is further for: judging whether the information of the first version has been received from the client after receiving the information of the first version from the client, skipping the step of informing the second module in response to a determination that the information of the first version has been received.

15 14. The server of claim 10, further comprising:

a fourth module, for providing the information of the second version recorded in the third module for the client.

15. A system for collecting software versions, comprising: a server and a client; wherein

20 the server is for:

receiving information of a first version of a software installed in a client device sent by a client;

judging whether the first version is newer than a second version based on the information of the first version and information of the second version of the software recorded in the server;

25 updating the information of the second version recorded in the server in response to a determination that the first version is newer than the second version.

16. The system of claim 15, wherein

the server is further for:

judging whether the number of times the information of the first version has been received exceeds a first threshold in response to a determination that the first version is newer than the second version, performing the step of updating the information of the second version recorded in the server when the number of times the information of the first version has been received exceeds the first threshold; or

judging whether a proportion of the number of times the information of the first version has been received to the total number of times information of versions of the software has been received exceeds a second threshold in response to a determination that the first version is newer than the second version, performing the step of updating the information of the second version recorded in the server when the proportion of the number of times the information of the first version has been received to the total number of times the information of versions of the software has been received exceeds the second threshold.

17. The system of claim 16, wherein
the server is further for:

the second module is for comparing the number of times the information of the first version has been received within a pre-defined period of time with the first threshold.

18. The system of claim 16, wherein
the server is further for:
judging whether the information of the first version has been received from the client after receiving the information of the first version from the client, performing no modification to the number of times the information of the first version has been received when the information of the first version has been received from the client.

19. The system of claim 15, wherein
the server is further for:
providing the recorded information of the second version for the client.

20. The system of claim 15, wherein
the client is for:

obtaining the information of the first version of the software installed in the client device;

sending the information of the first version to the server.

21. The system of claim 20, wherein

5 the client is for obtaining the information of the first version from a DisplayVersion key value of a software uninstall node in a registry, or from executable resources of a main program of the software, or from a configuration file in an installation directory of the software.

22. The system of claim 20, wherein

10 the client is further for: obtaining a list of software installed in the client device, performing the obtaining step for each software in the list of software.

23. The system of claim 22, wherein

15 the client is for obtaining the list of software from a software uninstall list of the registry, or from desktop shortcuts, or from the application list in Windows start menu, or from shortcuts in Windows quick launch toolbar in the client device.