(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 January 2002 (17.01.2002)

PCT

(10) International Publication Number
WO 02/05184 A2

(51) International Patent Classification⁷: G06F 17/60

(21) International Application Number: PCT/US01/21502

(22) International Filing Date: 6 July 2001 (06.07.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/217,662	10 July 2000 (10.07.2000)	US
09/640,043	15 August 2000 (15.08.2000)	US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:

US	09/640,043 (CON)
Filed on	15 August 2000 (15.08.2000)

(71) Applicant *(for all designated States except US)*: **CRITICAL DEVICES, INC.** [US/US]; 13155 Noel Road, Suite 600, Dallas, TX 75240 (US).

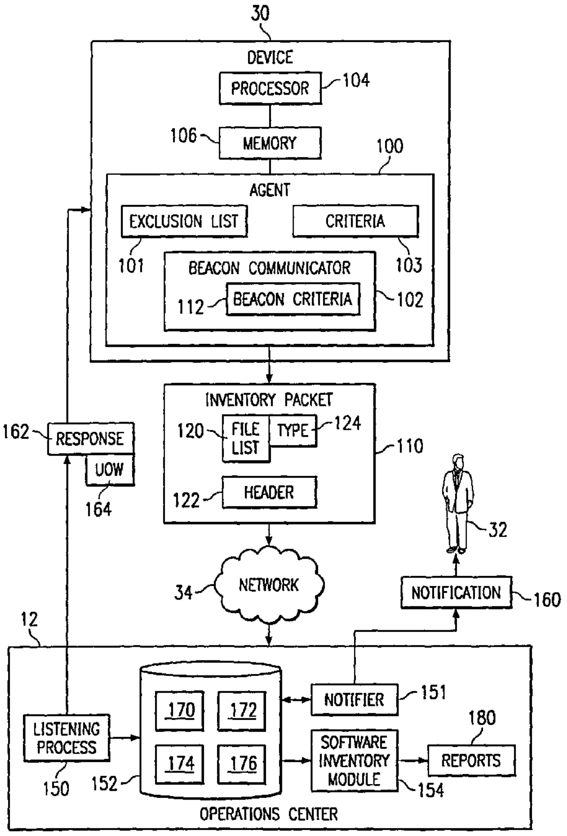
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **LEVI, Andrew, E.** [US/US]; 6717 Shadowcrest Drive, Plano, TX 75093 (US). **ODOM, Daniel, W.** [US/US]; 1100 Meredith Lane, #724, Plano, TX 75093 (US). **BAUER, Bradley, W.** [US/US]; 3511 Excalibur Court, Richardson, TX 75082 (US).

(74) Agent: **FISH, Charles, S.**; Baker Botts L.L.P., Suite 600, 2001 Ross Avenue, Dallas, TX 75201-2980 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EC, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR SOFTWARE INVENTORY MANAGEMENT USING A GLOBAL CENTRAL REPOSITORY



(57) Abstract: A device (30) at a site (14) includes and agent (100) with a software inventory service (220). The software inventory service scans the device for file names (410). The file names and information associated with the file names, such as a size (422), a date (420) and a platfrom (424) are communicated to an operations center (12) for processing. At the operations center, the file names are assocaited with descriptions (430), manufacturers (432), languages (422) and application/suite names (440) and stored in a repository (172). Once the file names are stored in the repository (172). Once the file names are stored in the repository, the devices may be linked to the applcations and file names in the repository which are installed on the device. A software inventory module (154) at the operations center supports the association of software on devices to the central repository at the operatons center. The software inventory module also supports an assocation of sofware on devices to licenses maintained at the operations center for license compliance analysis and may disable unlicensed software on the devices.

GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND SYSTEM FOR SOFTWARE INVENTORY MANAGEMENT USING A GLOBAL CENTRAL REPOSITORY

TECHNICAL FIELD OF THE INVENTION

This invention relates in general to networked computers and, more particularly, to a method and system for software inventory management using a global central repository.

BACKGROUND OF THE INVENTION

As computer systems and networks have grown in complexity and usefulness, businesses have become increasingly reliant on the proper functioning of their computers and the networks which connect the computers. In particular, as the number of computers connected to networks, such as corporate internets, has grown, so has the administrative burden associated with managing software on the computers. Traditional software inventory systems have concentrated on corporate networks and the inventorying of software on the corporate intranet.

SUMMARY OF THE INVENTION

From the foregoing, it may be appreciated that a need has arisen for a method and apparatus for software inventory management using a global central repository.

One aspect of the present invention involves a method for software inventory management comprising installing an agent at a device, the agent having an associated exclusion list and associated criteria and scanning, with the agent, a file system associated with the device. The method further comprises generating a list associated with the device based on the scan and the criteria, the list having at least one associated file name, storing at least one of the file names in an inventory packet, and communicating the inventory packet to an operations center.

Another aspect of the present invention involves a method for software inventory management comprising receiving, at an operations center, at least one first file name from a first device. At least one of the first file names has an associated size and an associated date. The method further comprises storing at least one of the first file names in a first table and examining at least one of the first file names in the first table. The method also comprises selecting a second table to associate with a portion of the first file names in the first table based on the examination, storing the selected ones of the first file names on the second table, and associating the first file names in the second table with the first device.

A further aspect of the present invention involves a method for software inventory management comprising receiving, at an operations center, a plurality of file names from a plurality of devices, wherein ones of the file names have an associated one of the devices and associating respective portions of the file names with respective applications. The method further comprises storing the associations between the file names and the applications in a central repository and associating respective ones of the devices with at least one of the applications such that applications installed on the device are associated with the device. The method also comprises storing the associations between the applications and the devices in the central repository, accessing the associations between the devices and the applications by an administrator associated with the device, and generating a report for at least one of the devices based on the associations in the central repository, the report indicating the applications respectively associated with the respective devices.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention will be realized from the detailed description that follows, taken in conjunction with the accompanying drawings, in which:

FIGURE 1 is a block diagram showing an exemplary configuration of a remote device software inventory management system utilizing the present invention;

FIGURE 2 is a block diagram illustrating further details of a device and an operations center according to one embodiment of the present invention;

FIGURE 3 is a block diagram illustrating details of an agent according to one embodiment of the present invention;

FIGURE 4 is a block diagram illustrating further details of a device inventory table according to one embodiment of the present invention;

FIGURE 5 is a block diagram illustrating details of an inventory repository according to one embodiment of the present invention;

FIGURE 6 is a block diagram illustrating details of a management table according to one embodiment of the present invention;

FIGURE 7 illustrates an exemplary embodiment of an exclusion table according to one embodiment of the present invention;

FIGURE 8 is a flowchart illustrating a method for performing software inventorying at the devices according to one embodiment of the present invention; and

FIGURE 9 is a flowchart illustrating a method for providing software inventory management according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

FIGURE 1 is a block diagram illustrating an exemplary configuration of a remote device software inventory management system 10 utilizing the present invention. System 10 provides the capability for centralized software inventory management. A central repository may be shared by multiple subscribers, such as corporations. The central repository allows for the organization of file names associated with applications and storage of information about the file names and applications in a shared repository. The shared repository decreases the typical administrative burden associated with software inventory management by avoiding duplicative effort with respect to organizing information about applications and associating file names with applications.

Remote device software inventory management system 10 comprises an operations center 12, one or more sites 14 and 16, one or more devices 30 and one or more contacts 32. A communications network 34 such as the Internet may be used to couple operations center 12, sites 14 and 16, and contact 32. Contact 32 may further be connected to operations center 12 by a communications link 36.

Operations center 12 comprises a central administration and operations center for providing software inventory management services to devices 30 and sites 14 and 16. Operations center 12 is described in more detail in FIGURE 2.

Each site 14 and 16 comprises one or more devices 30. Hereinafter sites will be referred to generally as "site or sites 14" with the other reference number (16) being used to refer to that particular site. Sites 14 may represent physical and logical entities that have contracted with operations center 12 for software inventory management services. Site 14 may be a company, a department within a company, a building, a geographic area, a logical entity occupying multiple geographic locations, or other suitable logical or physical entities capable of communicating over network 34 with operations center 12. The software inventory management services provided by operations center 12 may comprise determining the software installed on devices 30, remotely removing and/or installing software on devices 30, supporting software license enforcement and administration on devices 30 and sites 14, and supporting the management of other information, such as contracts and service agreements, associated with software installed on devices 30.

Each site 14 and 16 has an associated site identifier 20. Site identifiers 20 respectively comprise text, alphanumeric value or other suitable value for identifying the associated site. Site identifiers 20 are unique with respect to each other so as to distinguish sites 14. Identifier 20 may be determined and associated with sites 14 using any suitable method. For example, operations center 12 may generate identifier 20 in response to site 14 subscribing for service from operations center 12.

Device 30 comprises any of a plurality of electronic devices having simple or advanced data processing capabilities that may communicate with a remote location, such as operations center 12. For example, device 30 may be a server, a workstation, a personal computer, a laptop, a soft drink dispensing machine, a network postage machine, a printer, a heating/ventilation/air conditioning (HVAC) system, a box with an electronically readable tag, a personal digital assistant (PDA), a cellular phone, a general purpose or specialized computer system, or another suitable device. Device 30 may also include location information, such as Global Positioning System (GPS) information, that may be communicated to a remote location, such as operation center 12. Each device 30 is associated with at least one site 14.

Each device 30 has a respective associated device identifier 40. Device identifier 40 comprises any suitable text, alphanumeric value or other suitable value for identifying device 30. In one embodiment, device identifiers 40 are unique with respect to each other so as to distinguish devices 30. Identifier 40 may be determined and associated with device 30 using any suitable method. For example, operations center 12 may generate identifier 40 in response to device 30 subscribing for service from operations center 12.

Contact 32 represents one or more personnel who may be contacted in order to handle situations associated with devices 30 handled by system 10. In the disclosed embodiment, contact 32 may be notified by any one or more means of communication such as electronic mail, a pager, a phone, or a fax. Contact 32 may respond to operations center 12 that the notification of the situation has been received.

Network 34 may be the Internet or any other suitable combination of local area networks (LAN), medium and metropolitan area networks (MAN), wide area networks (WAN), intranets, and other suitable wireless or wire-based communication links.

Communications link 36 comprises a wireless or wireline communications system operable to communicate data from operations center 12 to contact 32. For example, link 36 may comprise a pager, a plain old telephone service (POTS) phone, a Voice over Packet (VoP) phone, a fax machine, electronic mail or any other suitable communications device.

FIGURE 2 is a block diagram illustrating further details of devices 30 and operations center 12. Device 30 further comprises an agent 100, a beacon communicator 102, a processor 104 and a computer readable memory 106. Processor 104 comprises any suitable electronic data processor operable to read data from memory 106 and execute applications. Memory 106 comprises any suitable computer readable storage device, such as magnetic storage, optical storage, persistent and transient memory such as Dynamic Random Access Memory (DRAM) and Static Ram (SRAM), and other suitable transient and/or persistent storage devices and technologies in any combination.

Agent 100 comprises a software application or applet executing on device 30 and operable to collect software information for inventory packet 110. Agent 100 is configurable by an administrator associated with device 30 to collect information at various times. For example, agent 100 may be configured to scan for software once a day, twice a day, every 5 hours or other suitable intervals. Agent 100 is described in more detail in association with FIGURE 3.

Agent 100 further comprises an agent exclusion table 101 and criteria 103. Agent exclusion table 101 comprises a list of file names not to be sent to operations center 12 as part of software inventory management system 10. More specifically, a size and a date may be associated with each file in table 101. The size and the date allow for the handling of different versions of the same file name. For example, table 101 may include "file1.exe", indicating that all files with a name of "file1.exe" are to be excluded by agent 100. For another example, table 101 may include a file name of "file2.exe" and an associated size of "32,767" indicating that all files with both a file name of "file2.exe" and a size of "32,767". In one embodiment, exclusion table 101 may be customized based on device 30. For example, exclusion table 101 for a personal digital assistant (PDA) may be different from exclusion table 101 for a Windows 95/98 based machine since different files may be ignored for different types of devices 30.

Criteria 103 indicates which types of files agent 100 should scan for. For example, criteria 103 may indicate all files ending in ".exe" are to be found. In one embodiment, criteria 103 indicates that all files ending in ".exe", ".dll" and ".com" are to be scanned for on device 30. Criteria 103 also indicates when to perform scans and what type of scan to perform.

Beacon communicator 102 comprises a software application executing on device 30 and operable to communicate with operations center 12 and to generate an inventory packet 110. Beacon communicator 102 further comprises beacon criteria 112. In one embodiment, beacon communicator 102 comprises an operating system extension. Alternatively, beacon communicator 102 may comprise any suitable combination of hardware and software, either alone or in combination, for supporting communication of device 30 with operations center 12 and may be created using any suitable programming or scripting language, such C, C++ and Java.

Beacon criteria 112 comprises a plurality of configurable parameters which indicate when beacon communicator 102 is to communicate software inventory data to operations center 12. In one embodiment, beacon criteria 112 indicates that inventory packet 110 is to be generated when device 30 is first activated. Alternatively, beacon criteria 112 may indicate that beacon packet 110 be generated at shutdown or at periodic intervals, such as every minute, every hour and/or once a day, while device 30 is activated. In addition, beacon criteria 112 may interact with power management and software power control features associated with device 30, such as when device 30 is a Personal Digital Assistant (PDA), for example, a PalmPilot by Palm, Inc., to activate device 30 without user intervention to generate beacon packet 110. For example, a Palm VII wireless PDA by Palm, Inc. could be instructed to automatically activate every day at midnight and transmit software inventory information so that software on the Palm VII can be tracked by system 10.

Inventory packet 110 comprises a data packet generated by agent 100, a file list 120 and a header 122. File list 120 comprises a plurality of file names found on device 30. More specifically, file list 120 includes file names found on device 30 by agent 100 and not excluded based on exclusion table 101. File list 120 further comprises a type 124. Type 124 indicates the type of list file list 120 contains. In one embodiment, type 124 indicates either a "brief" scan or a "verbose" scan. When type 124 indicates a "brief" scan, a file size and a date are associated with each file name in file list 120. When type 124 indicates a "verbose" scan, agent 100 has attempted to determine the manufacturer associated with the file name, title associated with the file name and the version associated with the file name. For example, the file name may be "word.exe" and the title "MS Word 95". File list 120 may optionally be encrypted for increased security as specified by beacon criteria 112.

Header 122 comprises routing and control information for use with internet 34 for communicating packet 110 to operations center 12. Header 122 may also include any suitable additional routing and control information.

Operations center 12 comprises a listening process 150, a notifier 151, a database 152 and a software inventory module 154. Listening process 150 comprises a software application operable to receive inventory packets 110 from devices 30 over network 34. Listening process 150 may alternatively be located remote from both operations center 12 and device 30. Listening process 150 is operable to communicate a notification 160 generated by notifier 151 and communicate notification 160 to contact 32. Notifier 151 comprises a software application operable to generate notification 160. Notifier 151 monitors database 152 for changes and communicates notifications 160 as appropriate to contacts 32. Notification 160 comprises a fax, a page, a phone call, an electronic mail or other suitable technique for communicating information to contact 32. Listening process 150 is further operable to communicate a response 162 generated by module 154 to device 30. Typically, response 162 is communicated over network 34. Response 162 comprises information to be communicated to agent 100 and one or more Units of Work (UOW) 164 for agent 100. Each UOW 164 comprises an indication to agent 100 of an action to perform. An action may comprise adding software to device 30, removing software from device 30, deactivating software on device 30 and other suitable actions.

Database 152 comprises a device inventory table 170, an inventory repository 172, a license management portion 174 and a new files table 176. Device inventory table 170 comprises information associated with software installed on devices 30 at sites 14. Table 170 is described in more detail in association with FIGURE 4. Repository 172 comprises one or more tables with entries for tracking software and associated information for system 10 and is described in more detail in FIGURE 5. License management portion 174 comprises software management related information, such as support contracts, licenses and other information. Portion 174 is described in more detail in association with FIGURE 6. New files table 176 comprises a table listing file names which have been received in a file list 120, but not yet added to repository 172. Table 176 may store a size, a date, a title, a manufacturer and other information in association with the file names.

Software inventory module 154 comprises a software and/or hardware application operable to receive inventory packets from listening process 150 and process information in file list 120. More specifically, module 154 processes information in file list 120 and associates devices 30 in device inventory table 170 with software applications in repository 172 and new files table 176. Module 154 may also accept input from operations center personnel associated with operations center 12 for moving file names from new files table 176 to repository 172.

Module 154 is further operable to generate one or more reports 180. Reports 180 may include any portion of the files, software applications, contracts, licenses and other information stored in device inventory table 170, repository 172, license management portion 174 and new files table 176. For example, a user of system 10 may generate a report 180 listing all software installed on a particular device 30. For another example, a user of system 10 may generate a report 180 listing all software installed on a particular device 30 and license information associated with the software.

In operation, beacon communicator 102 generates inventory packet 110 with the information collected by agent 100 and stored in file list portion 120 of agent 100. File list portion 120 is generated by agent 100 by searching file systems associated with device 30. More specifically, agent 100 searches for files as specified in criteria 103. Agent 100 generates file list 120 with all files found during the scan for inclusion in inventory packet 110 and agent 100 then removes files based on exclusion table 101 from list 120.

In one embodiment, agent 100 performs either a brief scan or a verbose scan. When agent 100 performs a brief scan, a file size and a date are associated with each file name in file list 120 and type 124 indicates a brief scan. When agent 100 performs a verbose scan, a file size, a date, a manufacturer, a title and a version may be associated with each file name in list 120 and type 124 indicates a verbose scan. Agent 100 typically performs a brief scan by scanning the file system associated with device 30. Agent 100 may perform a verbose scan by both scanning the file system and by scanning, for example, a registry associated with a Windows 95 computer. In general, a verbose scan is performed by scanning both the file system and any other information available regarding software on device 30. For example, information regarding the software associated with icons, shortcuts, configuration files and other sources may be scanned.

Inventory packet 110 is then transmitted over network 34 to listening process 150 at operations center 12. Listening process 150 receives packet 110 and forwards packet 110 to module 154 for processing. Module 154 examines list 120 and determines if each file in list 120 is listed in repository 172. If so, module links device id 40 associated with device 30 which generated packet 110 in device inventory table 170 to the entry for the file in repository 172. When a file in list 120 is not found in repository 172, the file is added to new files table 176 with the information, such as size and date, associated with the file name in file list 120. Module 154 may also generate one or more reports 180 in response to requests from users of system 10.

Module 154 determines whether any UOWs 164 need to be generated for device 30. Response 162 is then generated by module 154 for communication to device 30 and any UOWs 164 are attached to response 162. More specifically, response 162 may provide agent 100 and communicator 102 with software changes. For example, a new version of agent 100 may be available for device 30 and response 162 instructs agent 100 to update itself and provides the necessary information for agent 100 to perform the update in an appropriate UOW 164. For another example, license management portion 174 may indicate that device 30 is running unlicensed software and a UOW 164 may be generated to disable, remove and/or deactivate the unlicensed software. In general, response 162 and UOWs 164 may be used to simply acknowledge the receipt of inventory packet 110 by operations center 12, and may be used to also communicate actions to device 30 using UOWs 164.

FIGURE 3 is a block diagram illustrating details of agent 100. Agent 100 comprises foundation services 200 and software inventory services 202.

Foundation services 200 comprise beacon communicator 102, a class library 210, a communication system 212 and an encryption system 214. Class library 210 comprises a portion of agent 100 operable to receive information from services 202 and format the information into inventory packet 110. Communication system 212 comprises a portion of agent 100 operable to support communication over network 34 and other data communication systems by class library 210. Communication system 212 allows the use of multiple and different networking protocols and technology with agent 100 without requiring a distinct agent 100 for each networking protocol or technology. System 212 is typically configured to operate with network 34 to which device 30 having agent 100 will be coupled. Encryption system 214 comprises a portion of agent 100 operable to support encryption of information in beacon packet 110 using class library 210.

Software inventory services 202 comprises exclusion list 101, software inventory service 220, and criteria 103. Service 220 comprises a portion of agent 100 operable to traverse a file system associated with device 30 and collect information regarding files stored on device 30.

In operation, software inventory service 220 scans the file system associated with device 30 and stores, in list 120, files found during the scan. Service 220 also stores additional information, such as the size, date, manufacturer and title, associated with file names on device 30 depending on whether service 220 is performing a brief scan or a verbose scan. In one embodiment, criteria 103 specifies that a verbose scan is performed by service 220 when agent 100 is first installed on device 30 and a brief scan otherwise. Module 154 may also generate a UOW 164 requesting that service 220 perform a verbose scan or a brief scan. For example, when device 30 is a Windows 95/98 based computer, a verbose scan may include searching both the file system and the registry for information about files in the directory structure.

Agent 100 then uses communications system 212 to properly format the information from service 220 for transmission over network 34. In particular, class library 210 packetizes the information into file list 120 of inventory packet 110 and applies the appropriate network header and routing information based on system 212 to header 122 for communication over network 34 or other suitable network to which device 30 is coupled. Class library 210 then encrypts file list 120 based on encryption system 214.

In the disclosed embodiment, a public key-private key system is used wherein encryption system 214 provides a public key associated with operations center 12 for use by class library 210 in encrypting file list 120. Class library 210 then communicates inventory packet over network 34 to operations center 12.

FIGURE 4 is a block diagram illustrating further details of device inventory table 170. Device inventory table 170 comprises one or more site identifiers 20, one or more device identifiers 40, hardware data 300, an update date 302, and a list of installed applications 304. Hardware data 300 comprises information regarding hardware installed on device 30. Update date 302 comprises an indication of the date and time when device table 170 was last updated in response to processing of file list 120 by module 154. For example, module 154 may check update date 302 to check that information about device 30 is current and generate a UOW 164 requesting an updated scan if the information is not current. Installed applications 304 comprises a list of file names listed in information repository 172 that are installed on device 30 indicated by device identifier 40. In one embodiment, installed applications 304 comprises links or references to file names in repository 172 indicating applications and file names installed on particular devices 30.

In operation, device inventory table 170 associates device identifier 40 with data regarding hardware installed on device 30 associated with device identifier 40 in hardware data 300 and a list of installed applications on device 30 associated with identifier 40 in installed applications 304. As module 154 processes file names in file list 120, module 154 updates installed applications 304 for device identifier 40 associated with device 30 which generated file list 120. More specifically, module 154 adds a link to the entry in table 176 or inventory repository 172 representing the file name in file list 120 to installed applications 304.

FIGURE 5 is a block diagram illustrating details of inventory repository 172. Inventory repository 172 comprises an exclusion table 400, an unresolved table 402, an untitled table 404, and a titled table 406.

Exclusion table 400 comprises one or more file names 410 which are considered to be ignorable or unimportant by module 154. Typically, operations center personnel will configure exclusion table 400, but table 400 may be configured by others. For example, standard operating system configuration files may be excluded as providing no useful information regarding device 30. Each file name 410 comprises an alphanumeric string, or other suitable identifier, for identifying files. For example, file name 410 may comprise the file names and extensions used by UNIX file systems, Microsoft Windows and Microsoft Disk Operating System (DOS) file names and associated extensions, Macintosh file names and/or Palm, Inc. PalmPilot application names. In general, file name 410 comprises the name or identifier used by device 30 to identify files in a file system associated with device 30, and/or, when device 30 lacks a distinct file system, such as Palm, Inc. PalmPilots, the technique used to identify applications installed on device 30.

Each file name may have an associated file size 420, file date 422 and platform 424. File size 420 comprises an indication of the size of the associated file name 410. Typically, size 420 expresses the size of the file in bytes, but any suitable representation may be used, such as number of memory records used. File date 422 comprises a date associated with file name 410. File date 422 may represent a creation date, a last modified date, a last accessed date or any other suitable date associated with file name 410. It should be noted that while date 422 is discussed in the singular, multiple dates 420 may be utilized within the scope of the invention. For example, UNIX based file systems may associate a creation date and a last accessed date with file names 410 and table 400 may store both dates in date 422. Platform 424 comprises an indication of the operating system and/or computer platform associated with file name 410. For example, platform 424 may comprise Microsoft Windows 95/98, Microsoft Windows NT, Microsoft Windows 2000, Microsoft Windows CE and associated version, Palm, Inc. Palm Operating System and associated version, the Mac OS and associated version, IBM's AIX, a UNIX version and associated vendor, Sun Microsystem's Solaris and associated version, a Linux version and associated vendor, and other suitable platform indicators. In various embodiments, platform 424 may be optionally used in association with exclusion table 400.

Unresolved table 402 comprises one or more file names 410 with associated date 420, size 422 and platform 424. Unresolved table 402 indicates files that have been received from devices 30 and moved from new files table 176, but about which no further information has been determined by administrators associated with operations center 12. Stated another way, files in unresolved table 402 have been examined by personnel associated with operations center 12 and removed from new files table 176, but sufficient information to transfer file names 410 to untitled table 404 or titled table 406 is not available.

Untitled table 404 comprises one or more files names 410. Each file name 410 in untitled table 404 has an associated date 420, size 422, platform 424, a file description 430 and a manufacturer 432. File description 430 comprises further details regarding a file name 410 beyond the file name itself. For example, description 430 for a file name 410 of "word.exe" may be "word processing software". Manufacturer 432 comprises an indication of the company which created the file name. For example, manufacturer 432 may comprise "Microsoft Corp." File names 410 in untitled table 404 have additional information associated with the file names 410 in comparison to unresolved table 402. More specifically, file description 430 and/or manufacturer 432 have been associated with file name 410. In one embodiment, personnel associated with operations center 12 have associated description 430 and/or manufacturer 432 with file names 410 in unresolved table 402 or new files table 176 and moved such file names 410 to untitled table 404.

Titled table 406 comprises an application/suite name 440, manufacturer 432, language 442 and one or more file names 410. Titled table 406 comprises file names 410 which have been organized by the application or application suite which the file names 410 are associated with. Application/suite name 440 comprises an indication of the name of the application or applications file names 410 form. For example, name 440 may be "Microsoft Office 97."

Language 442 comprises an indication of a language associated with application/suite name 440, such as English, Japanese, Spanish or other languages. File names 410 associated with name 440 indicate the files used to support name 440. Continuing the above example, name 440 of "Microsoft Office 97" may have associated file names 410 of "word.exe" and "word.dll". Manufacturer 432 associated with name 440 indicates the manufacturer of the application or suite in name 440. In one embodiment, titled table 406 tracks applications as applications, in contrast to tables 400, 402 and 404, which track file names individually. Stated another way, titled table 406 groups file names 410 by the application the file names 410 support instead of tracking file names 410 individually. It should be noted that while tables 400, 402, 404 and 406 are shown and discussed as separate tables, tables 400, 402, 404 and 406 may be combined in any suitable manner or their functionality realized using other techniques within the scope of the invention.

In operation, personnel associated with operations center 12 examine file names 410 in new files table 176 and determine which table 400, 402, 404 or 406, file names 410 in table 176 should be moved to. In one embodiment, personnel associated with operations center 12 may determine which table 400, 402, 404 or 406 to use based on information associated with the file names 410 in table 176 and information available to the operations center personnel. For example, operations center personnel may use publicly available information to determine the manufacturer 432, application/suite name 440, language 442 and description 430 associated with file names 410. For another example, operations center personnel may examine documentation files associated with file names 410 to determine the manufacturer 432, application/suite name 440 and description 430. If the operations center personnel lack additional information about file names 410 on table 176 then the file name may be moved to unresolved table 402. If manufacturer 432 and/or file description 430 are available then file name 410 may be moved to untitled table 404. If manufacturer 432 and/or file description 430, and application/suite name 440 are available then file name 410 may be moved to titled table 406 and associated with name 440.

Repository 172 provides a central location for associating file names 410 and application/suite names 440. Repository 172 also provides a central location for associating file names 410 with descriptions 430 and manufacturers 432. As repository 172 may be shared between multiple sites 14, each site 14 is not required to maintain individualized information regarding software installed on devices 30 at sites 14. In contrast to traditional software inventory management systems, a corporation does not need to maintain an individually maintained software inventory. Instead, a central repository may be maintained and the corporation may access the central repository using the Internet. Since repository 172 is centrally maintained, once file names 410 have been resolved to application names 440 once, the resolution does not have to be performed again. For example, the Microsoft Office 97 suite of products is commonly used in corporate environments and once the file names 410 associated with Microsoft Office 97 are resolved in repository 172, a link may simply be made between device ids 40 with Microsoft Office 97 installed and application/suite name 440 for "Microsoft Office 97" in table 170 to indicate that Microsoft Office 97 is installed on a particular device 30.

FIGURE 6 is a block diagram illustrating details of license management portion 174. License management portion 174 comprises one or more site identifiers 20, one or more device identifiers 40, inventory repository 172, one or more licenses 500 and one or more agreements 502. In one embodiment, portion 174 maintains licenses 500 for software for sites 14 identified by site identifiers 20. Alternatively, licenses 500 may be maintained for both devices 30 and sites 14 using identifiers 20 and 40. Licenses 500 comprise data related to licenses associated with files on devices 30. More specifically, each license 500 is associated with one or more device ids 40 and one or more site ids 20 indicating the software on the devices 30 associated with site id 20 and device id 40 are licensed to use. Agreements 502 comprise support agreements and other software related agreements associated with software on devices 30 associated with device ids 40 and site ids 20. License management portion 174 allows for the association of agreements 502 with particular device ids 40 and site ids 20 for increased ease of management by administrative personnel associated with sites 14 and devices 30.

In general, license management portion 174 associates site ids 20 and device ids 40. Portion 174 also associates device ids 40 and site ids with agreements 502, licenses 500 and application names 440 and file names 410 in inventory repository 172. Portion 174 supports the use of system 10 to track file names 410 installed on devices 30 and manage licenses and agreements associated with the software installed on devices 30.

FIGURE 7 illustrates an exemplary embodiment of exclusion table 400. Exclusion table 400 may use wildcards, such as "*", in association with file name 410, date 420 and size 422. The first exemplary entry in FIGURE 7 may match any file name 410 of "fish.dat" regardless of the associated date 420 and size 422. The first exemplary entry may be used when the particular version of the file name 410 is unimportant. The second exemplary entry may match file name 410 of "fish.cfg" and date 420 of "1/1/99" with any size 422, such as when the particular version of "fish.cfg" with a date of "1/1/99" is important. The third exemplary entry may match file name 410 of "fish.doc" and size 422 of "41,000" regardless of date 420. The fourth exemplary entry matches only the particular version of "fish.zip" with the specified date 420 and size 422. The fifth and sixth exemplary entries illustrate that wildcards may be used within file name 410 itself. In particular, the fifth exemplary entry may match any number of any characters ending in ".sys" with any date 420 and size 422. The sixth exemplary entry may match any number of any characters beginning with "bowl.*" with date 420 of "3/1/99" and any size 422. In general, any suitable wildcards and combination of wildcards may be used in any of file name 410, date 420 and size 422 within the scope of the invention. Wildcards may also be used with tables 402, 404 and 406. For example, unresolved table 402 may use wildcards with file names 410, date 420 and size 422. For another example, untitled table 404 may use wild cards with file names 410, date 420, size 422, manufacturer 432 and file description 430. For yet another example, titled table 406 may use wild cards with file names 410, date 420, size 422, manufacturer 432, file description 430, language 442 and name 440. In one embodiment, name 440 may use wild cards to form a portion of name 440. Wild cards may be used with tables 402, 404 and 406 to provide increased flexibility for handling multiple versions of file names 410 associated with different versions of applications.

FIGURE 8 is a flowchart illustrating a method for performing software inventorying at a device 30. The method begins at step 700 where software inventory service 220 is deployed to a device 30 as part of an agent 100. In one embodiment, deploying service 220 comprises installing agent 100 with service 220 on a device 30. Service 220 may also be installed to a device 30 which already has an agent 100 installed on it. Then, at step 702, service 220 performs a verbose scan for files at device 30 based on criteria 103. Criteria 103 may alternatively specify that a brief scan be performed when service 220 is installed. As part of the scan, file list 120 may be generated. Next, at step 704, file names are removed from file list 120 based on exclusion list 101. Proceeding to step 706, agent 100 generates inventory packet 110 and includes file list 120 therein. Then, at step 708, inventory packet 110 is compressed and encrypted and, at step 710, sent to operations center 12 over network 34.

Proceeding to step 712, agent 100 and device 30 perform normal operations. At decisional step 714, agent 100 determines whether another scan is to be performed based on criteria 103 or in response to a UOW 164 received from operations center 12 specifying that a scan be performed.

Criteria 103 may specify that a scan be performed when new software is installed at device 30, after a predetermined period of time, such a day, and/or based on other suitable events or occurrences at device 30. If no new scan is to be performed then the NO branch of decisional step 714 returns to step 712. If another scan is to be performed then the YES branch of decisional step 714 proceeds to decisional step 716.

At decisional step 716, the type of scan to be performed is determined based on criteria 103 or UOW 164. If a brief scan is to be performed, then the BRIEF branch of decisional step 716 leads to step 718 where a brief scan is performed by service 220. If a verbose scan is to be performed, then the VERBOSE branch of decisional step 716 leads to step 720 where a verbose scan is performed by service 220. Then, at step 722, old or duplicate files found in previous scans may be removed from the new list 120 generated by the new scan. The method then continues to step 704.

Returning to step 712, a processing path may also lead to decisional step 724. At decisional step 724 exclusion list 101 and/or criteria 103 may be updated in response to a UOW 164 received from operations center 12 or criteria 103. If no UOW 164 has been received or criteria 103 has not indicated that an update be requested, then the NO branch of decisional step 724 returns to step 712. If a UOW 164 has been received or criteria 103 specifies that an update is to be requested from operations center 12, then the YES branch of step 724 proceeds to step 726. At step 726, an update to criteria 103 and/or exclusion list 101 is received from operations center 12. Then, at step 728, exclusion list 101 and/or criteria 103 is updated.

FIGURE 9 is a flowchart illustrating a method for providing software inventory management. The method beings at step 800 where inventory packet 110 is received at operations center 12 from device 30. Then, at step 802, packet 110 is decrypted at operations center 12 and, at step 804, decompressed at operations center 12. Next, at step 806, packet 110 is queued for processing by module 154. Proceeding to step 808, module 154 analyzes list 120 in packet 110. More specifically, module 154 compares each file in list 120 to tables 176, 400, 402, 404 and 406.

At decisional step 810, if module 154 matches a file name 410 in exclusion table 400 to a file in list 120, then the YES branch of decisional step 810 leads to step 812 where the matched file is ignored because the file is matched on exclusion table 400. More specifically, to match a file comprises determining if the file name in list 120 matches file name 410, and date 420 and/or size 422 if date 420 and/or size 422 is specified. Then, at step 814, the next file in list 120 is examined and the method returns to step 810. If the file name in list 120 is not matched in exclusion table 400, then the NO branch of step 810 leads to step 816.

At decisional step 816, module 154 determines if the file in list 120 matches a file name 410 in titled table 406. If module 154 matches the file in list 120 to a file name 410 in titled table 406, then the YES branch of decisional step 816 leads to step 818. At step 818, device id 40 is associated with application name 440 associated with file name 410 matched in step 816. More specifically, application name 440 is added to the list of installed applications 304 associated with device id 40 in device inventory table 170. The method then proceeds to step 814. If the file name in list 120 is not matched in titled table 406, then the NO branch of decisional step 816 leads to step 820.

At decisional step 820, module 154 determines if the file in list 120 matches a file name 410 in unresolved table 402. If module 154 matches the file in list 120 to a file name 410 in unresolved table 402, then the YES branch of decisional step 820 leads to step 822. At step 822, device id 40 is associated with file name 410 matched in the preceding decisional step. More specifically, file name 410 is added to the list of installed applications 304 associated with device id 40 in device inventory table 170. The method then proceeds to step 814. If the file name in list 120 is not matched in unresolved table 402, then the NO branch of decisional step 820 leads to step 824.

At decisional step 824, module 154 determines if the file in list 120 matches a file name 410 in untitled table 404. If module 154 matches the file in list 120 to a file name 410 in untitled table 404, then the YES branch of decisional step 824 leads to step 822. If the file name in list 120 is not matched in untitled table 404, then the NO branch of decisional step 824 leads to step 826.

At step 826, if the file in file list 120 failed to match any of the tables 400, 402, 404 or 406, the file is added to new files table 176 if the file does not already exist in new files table 176. Files with the same name 410, but differing sizes 422 and/or dates 420, may be added to new files table 176 (and tables 400, 402, 404 and 406) as indicating multiple versions of the same file name 410. Proceeding to decisional step 828, module 154 determines if more files in list 120 remain to be processed. If more files remain to be processed in list 120, then the YES branch of decisional step 828 leads to step 814. If no more files remain to be processed then the NO branch of decisional step 828 leads to step 830.

At step 830, file names 410 on new files table 176 are analyzed by personnel associated with operations center 12. More specifically, operations center personnel determine whether additional information is available for file names 410 on new files table 176 and which table 400, 402, 404 or 406, file names 410 on table 176 should be moved to. Next, at step 832, once the operations center personnel have determined which table 400, 402, 404 or 406, to move file names 410 from table 176 to, the file names 410 are moved to the appropriate table 400, 402, 404 or 406.

The present invention provides a number of technical advantages. One such technical advantage is the capability to provide software inventory management over the Internet. The inventory repository supported at the operations center provides a central location for associating file names with other information. Thus, multiple corporations may now out-source software inventory management services to a single global repository and avoid duplicative effort in managing separate software inventory systems. By providing software inventory management over the Internet, once file names have been resolved to particular applications, a link may simply be made between a device with the particular application and the entry representing that application in the inventory repository.

Although one embodiment has been illustrated and described in detail, it should be understood that various substitutions and alterations can be made therein without departing from the scope of the present invention. For example, although the operations center is described as a single location, it could encompass several locations, for example, to serve different geographic areas.

It should also be recognized that other changes, substitutions and alterations are also possible without departing from the spirit and scope of the present invention, as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method for software inventory management comprising:

installing an agent at a device, the agent having an associated exclusion list and associated criteria;

scanning, with the agent, a file system associated with the device;

generating a list associated with the device based on the scan and the criteria, the list having at least one associated file name;

storing at least one of the file names in an inventory packet; and

communicating the inventory packet to an operations center.

2. The method for software inventory management according to Claim 1, wherein the file name further comprises an associated size, an associated date, an associated language and an associated platform.

3. The method for software inventory management according to Claim 1 and further comprising:

compressing the inventory packet; and

encrypting the inventory packet.

4. The method for software inventory management according to Claim 1, wherein the criteria comprises at least one indication of file names to scan for.

5. The method for software inventory management according to Claim 1, wherein the exclusion list comprises at least one file name to discard and removing at least one of the file names from the list based on the exclusion list.

6. The method for software inventory management according to Claim 1, wherein the file name comprises an application identifier.

7. A method for software inventory management comprising:

receiving, at an operations center, at least one first file name from a first device, at least one of the first file names having an associated size and an associated date;

storing at least one of the first file names in a first table;

examining at least one of the first file names in the first table;

selecting a second table to associate with a portion of the first file names in the first table based on the examination;

storing the selected ones of the first file names on the second table; and

associating the first file names in the second table with the first device.

8. The method for software inventory management according to Claim 7, wherein examining at least one of the file names comprises determining a version associated with the file name based on the file name and the associated size and date.

9. The method for software inventory management according to Claim 7, wherein examining at least one of the file names comprises analyzing, by a human, a relationship between at least one first file name on the first table and at least one first file name on the second table.

10. The method for software inventory management according to Claim 7, wherein receiving the file names from the first device comprises receiving a list of the file names from the first device.

11. The method for software inventory management according to Claim 7, wherein the second table comprises a titled table and selecting the second table comprises determining an application associated with the first file names.

12. The method for software inventory management according to Claim 7 and further comprising determining a file description and a manufacturer associated with the first file names.

13. The method for software inventory management according to Claim 7, wherein the second table comprises an untitled table and selecting the second table comprises determining a file description, a manufacturer, a language and a platform associated with the file name.

14. The method for software inventory management according to Claim 7, wherein the second table comprises an unresolved table.

15. The method for software inventory management according to Claim 7 and further comprising:

receiving at least one second file name associated with a second device distinct from the first device;

determining a relationship between at least one of the second file names and the first file names; and

associating the second device with the first file name related to the second file name.

16. A method for software inventory management comprising:

receiving, at an operations center, a plurality of file names from a plurality of devices, wherein ones of the file names have an associated one of the devices;

associating respective portions of the file names with respective applications;

storing the associations between the file names and the applications in a central repository;

associating respective ones of the devices with at least one of the applications such that applications installed on the device are associated with the device;

storing the associations between the applications and the devices in the central repository;

accessing the associations between the devices and the applications by an administrator associated with the device; and

generating a report for at least one of the devices based on the associations in the central repository, the report indicating the applications respectively associated with the respective devices.

17. The method for software inventory management according to Claim 16, wherein the device comprises a personal computer.

18. The method for software inventory management according to Claim 16, wherein each file name has an associated one of the devices.

19. The method for software inventory management according to Claim 16 and further comprising determining the applications installed at the device based on the file names on the device and the associations between file names and applications.

20. The method for software inventory management according to Claim 16, wherein the file names each have an associated size, date and description.

21. The method for software inventory management according to Claim 16, wherein the applications each have an associated manufacturer.

22. The method for software inventory management according to Claim 16 and further comprising:

associating a license with a first one of the applications and the device associated with the one of the applications; and

storing the association between the license, the first one of the applications and the device in a license management database.

23. The method for software inventory management according to Claim 22 and further comprising:

determining whether a second one of the applications has an associated license based on the license management database; and

deactivating the second one of the applications when the second one of the applications lacks an associated license.

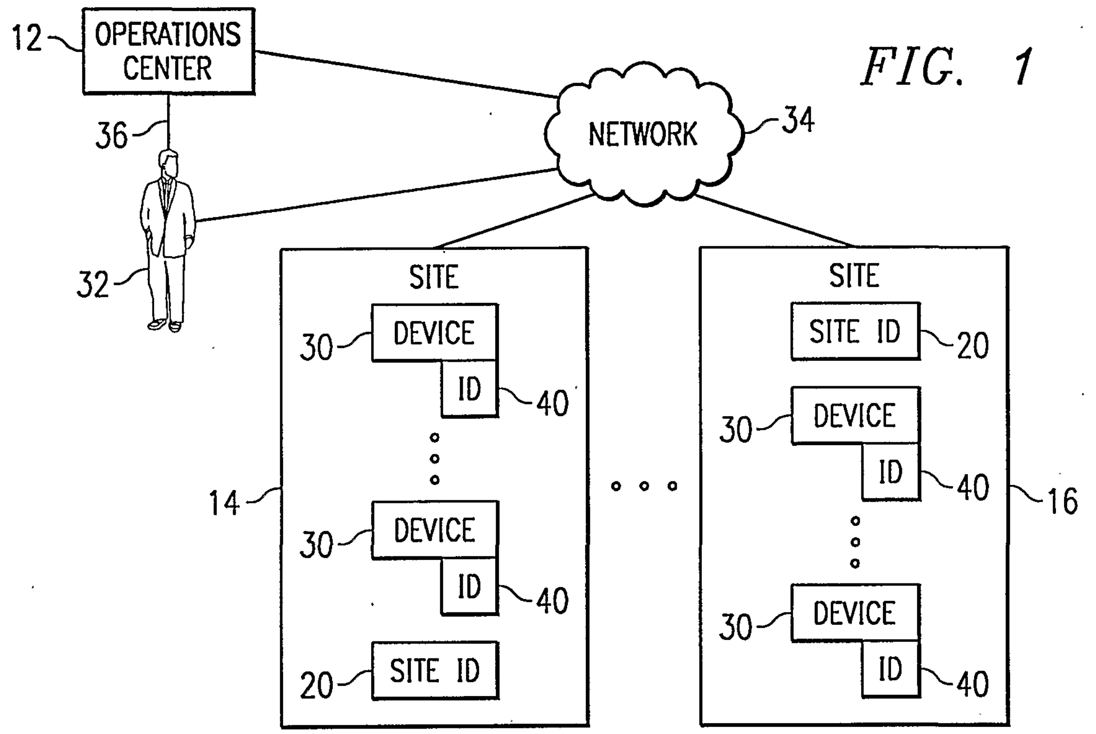

12
OPERATIONS CENTER
36
32
NETWORK
34
SITE
DEVICE
ID
30
40
14
DEVICE
ID
30
40
SITE ID
20
SITE
SITE ID
20
DEVICE
ID
30
40
16
DEVICE
ID
30
40

FIG. 1

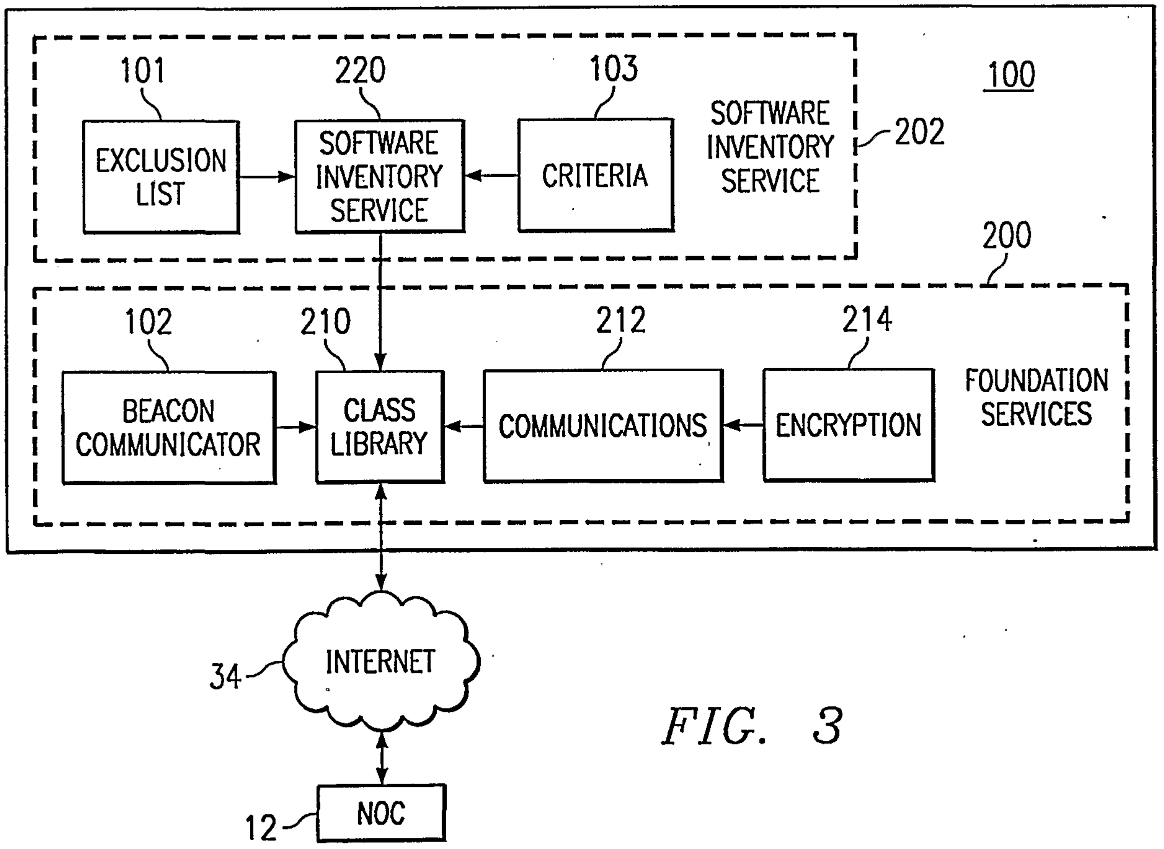

101
220
103
EXCLUSION LIST
SOFTWARE INVENTORY SERVICE
CRITERIA
SOFTWARE INVENTORY SERVICE
202
100
200
102
210
212
214
BEACON COMMUNICATOR
CLASS LIBRARY
COMMUNICATIONS
ENCRYPTION
FOUNDATION SERVICES
34
INTERNET
12
NOC

FIG. 3

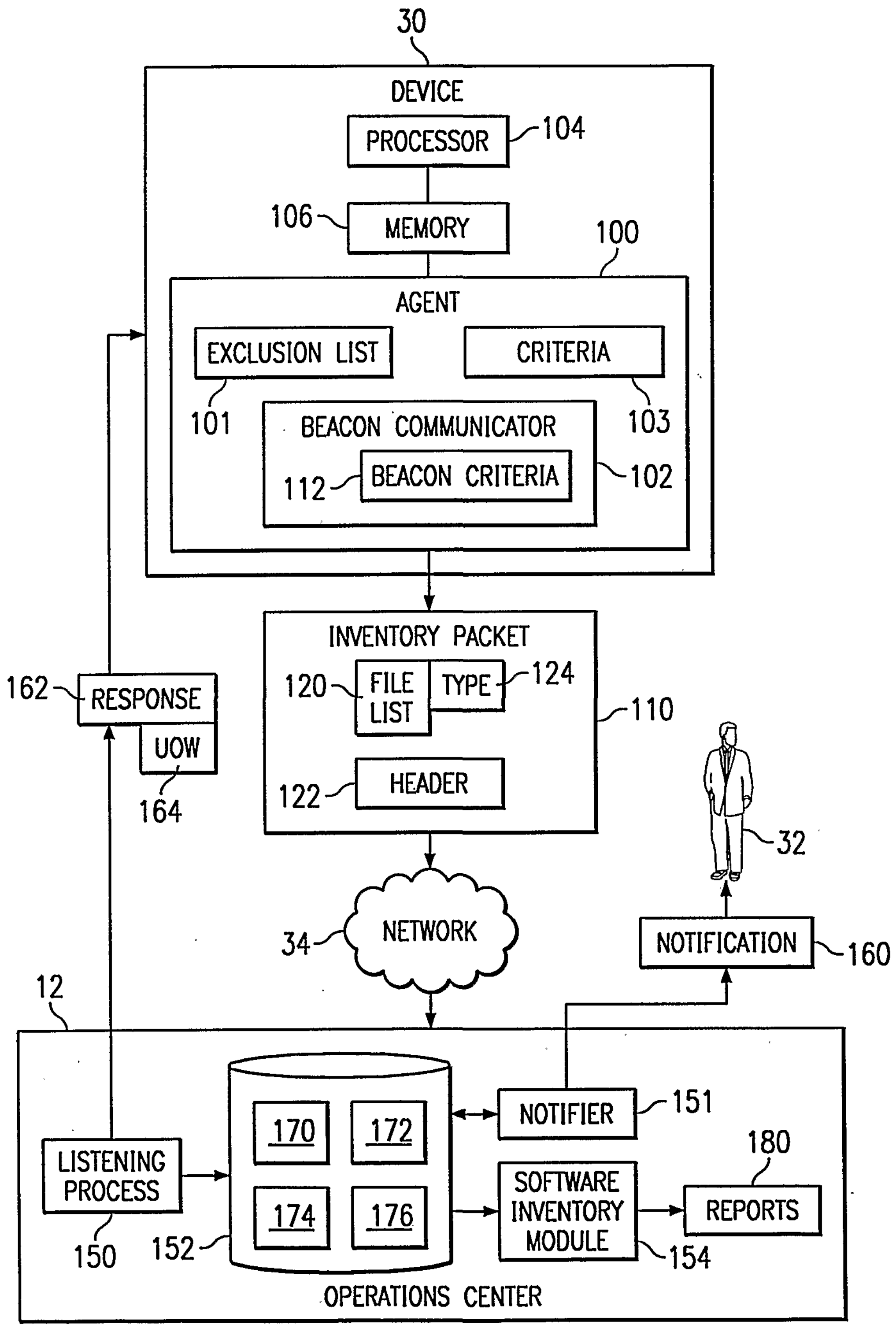

30
DEVICE
PROCESSOR
104
106
MEMORY
100
AGENT
EXCLUSION LIST
CRITERIA
101
103
BEACON COMMUNICATOR
112
BEACON CRITERIA
102
INVENTORY PACKET
120
FILE LIST
TYPE
124
110
162
RESPONSE
UOW
164
122
HEADER
32
34
NETWORK
NOTIFICATION
160
12
NOTIFIER
151
170
172
174
176
180
LISTENING PROCESS
SOFTWARE INVENTORY MODULE
REPORTS
150
152
154
OPERATIONS CENTER

FIG. 2

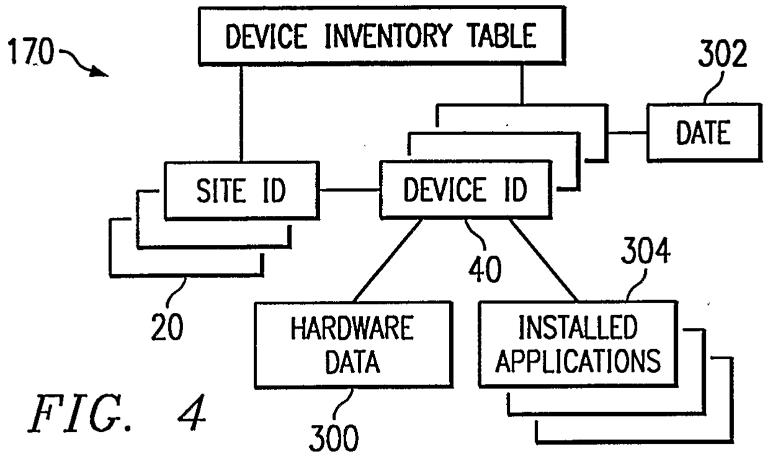


FIG. 4

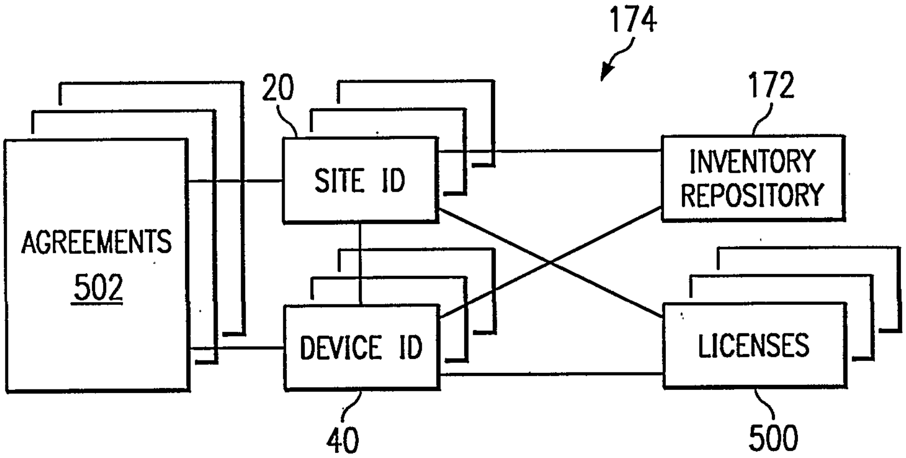
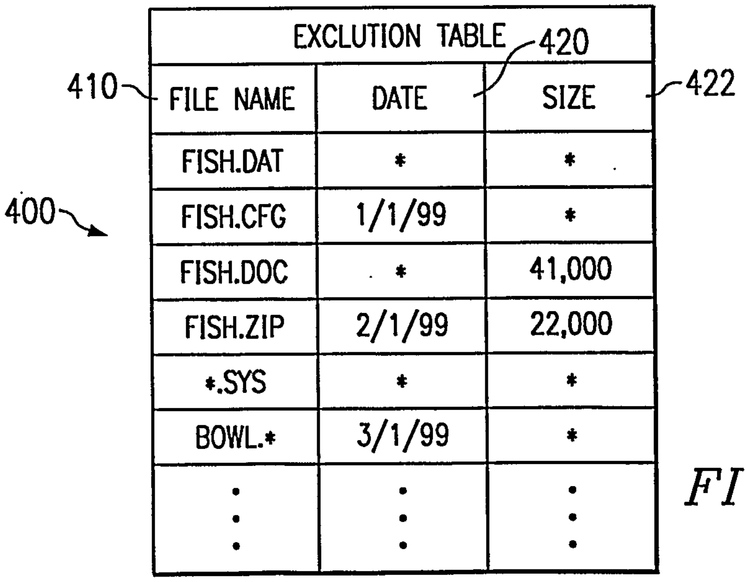


FIG. 6


400
410
420
422

EXCLUTION TABLE		
FILE NAME	DATE	SIZE
FISH.DAT	*	*
FISH.CFG	1/1/99	*
FISH.DOC	*	41,000
FISH.ZIP	2/1/99	22,000
*.SYS	*	*
BOWL.*	3/1/99	*
⋮	⋮	⋮

FIG. 7

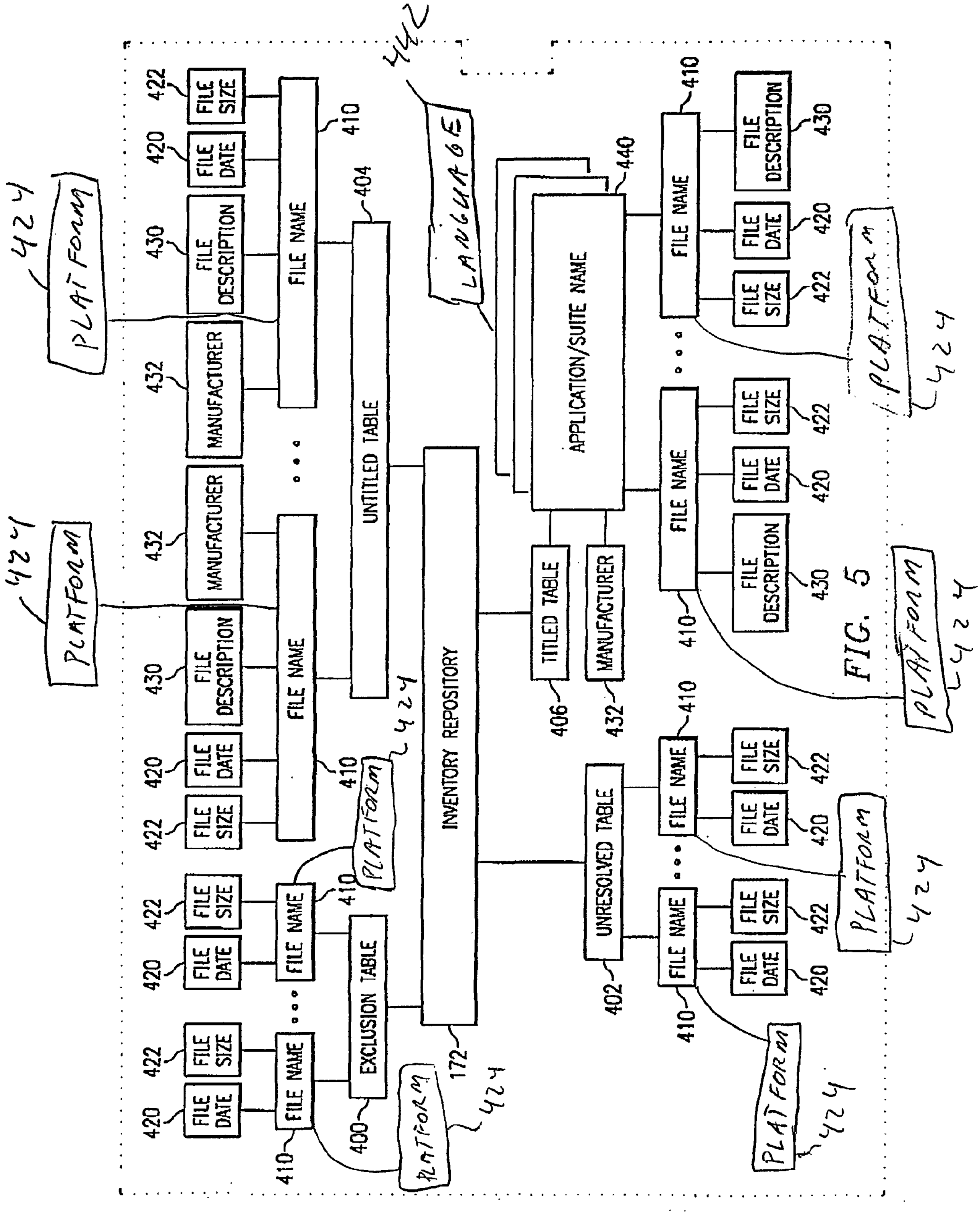

INVENTORY REPOSITORY
172
EXCLUSION TABLE
400
UNTITLED TABLE
404
TITLED TABLE
406
MANUFACTURER
432
UNRESOLVED TABLE
402
APPLICATION/SUITE NAME
440
FILE NAME
410
FILE DATE
420
FILE SIZE
422
FILE DESCRIPTION
430
MANUFACTURER
432
PLATFORM
424
LANGUAGE
442

FIG. 5

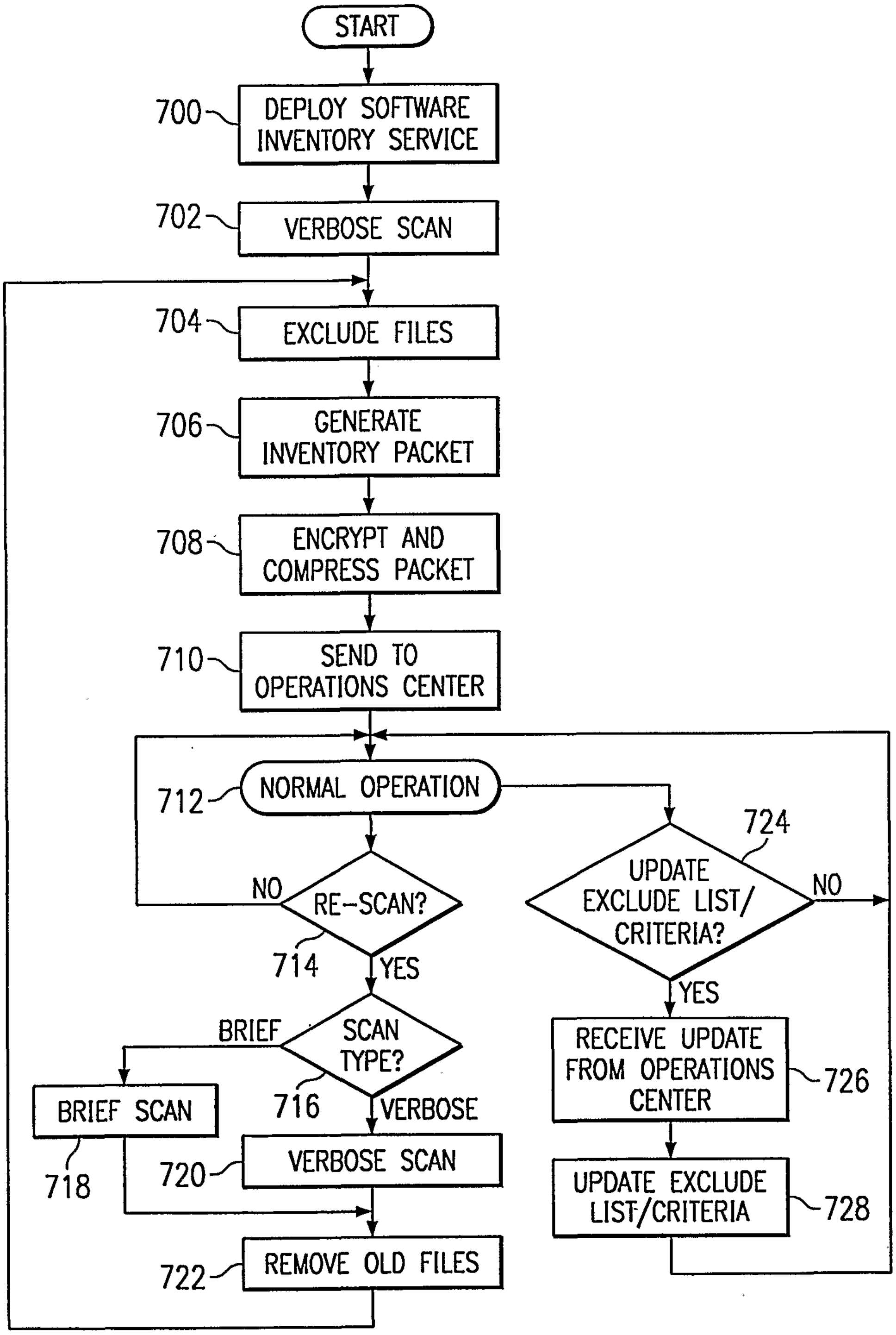

START
700
DEPLOY SOFTWARE INVENTORY SERVICE
702
VERBOSE SCAN
704
EXCLUDE FILES
706
GENERATE INVENTORY PACKET
708
ENCRYPT AND COMPRESS PACKET
710
SEND TO OPERATIONS CENTER
712
NORMAL OPERATION
NO
RE-SCAN?
714
YES
BRIEF
SCAN TYPE?
716
VERBOSE
BRIEF SCAN
718
720
VERBOSE SCAN
722
REMOVE OLD FILES
724
UPDATE EXCLUDE LIST/ CRITERIA?
NO
YES
RECEIVE UPDATE FROM OPERATIONS CENTER
726
UPDATE EXCLUDE LIST/CRITERIA
728

FIG. 8

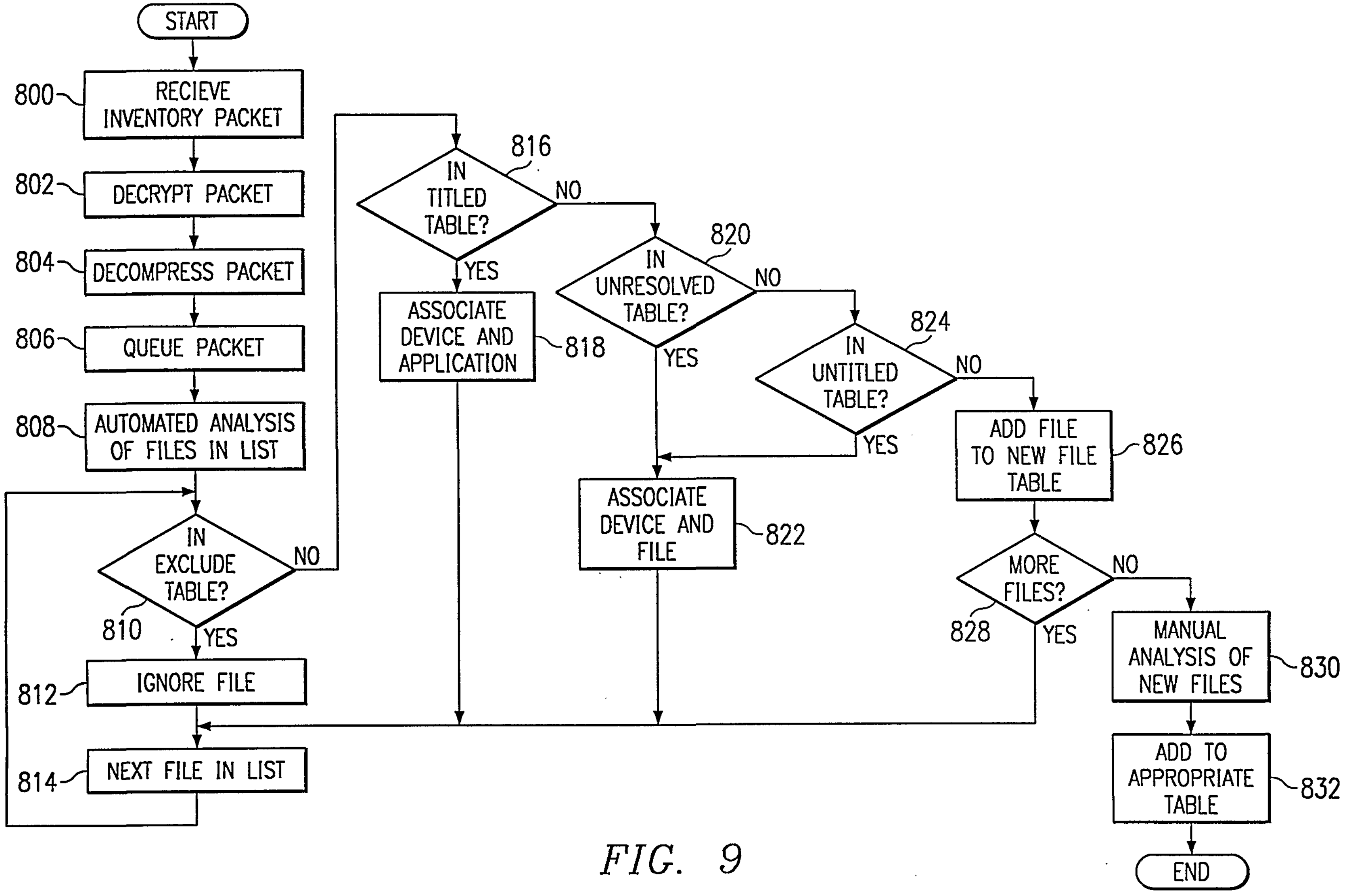

START
800
RECIEVE INVENTORY PACKET
802
DECRYPT PACKET
804
DECOMPRESS PACKET
806
QUEUE PACKET
808
AUTOMATED ANALYSIS OF FILES IN LIST
810
IN EXCLUDE TABLE?
YES
NO
812
IGNORE FILE
814
NEXT FILE IN LIST
816
IN TITLED TABLE?
YES
NO
818
ASSOCIATE DEVICE AND APPLICATION
820
IN UNRESOLVED TABLE?
YES
NO
822
ASSOCIATE DEVICE AND FILE
824
IN UNTITLED TABLE?
YES
NO
826
ADD FILE TO NEW FILE TABLE
828
MORE FILES?
YES
NO
830
MANUAL ANALYSIS OF NEW FILES
832
ADD TO APPROPRIATE TABLE
END

FIG. 9