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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 August 2010 (12.08.2010)

PCT

(10) International Publication Number
WO 2010/089826 A1

(51) International Patent Classification:
G06F 3/12 (2006.01) *G06F 9/445* (2006.01)

(21) International Application Number:
PCT/JP2009/006281

(22) International Filing Date:
20 November 2009 (20.11.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2009-025293 5 February 2009 (05.02.2009) JP

(71) Applicant *(for all designated States except US)*: **RICOH COMPANY, LTD.** [JP/JP]; 3-6, Nakamagome 1-chome, Ohta-ku, Tokyo, 1438555 (JP).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **DAN, Koji** [JP/JP]; 75-7, Makigahara, Asahi-ku, Yokohama-shi, Kanagawa, 2410836 (JP).

(74) Agent: **ITOH, Tadahiko**; 32nd Floor, Yebisu Garden Place Tower, 20-3,Ebisu 4-chome, Shibuya-ku, Tokyo 1506032 (JP).

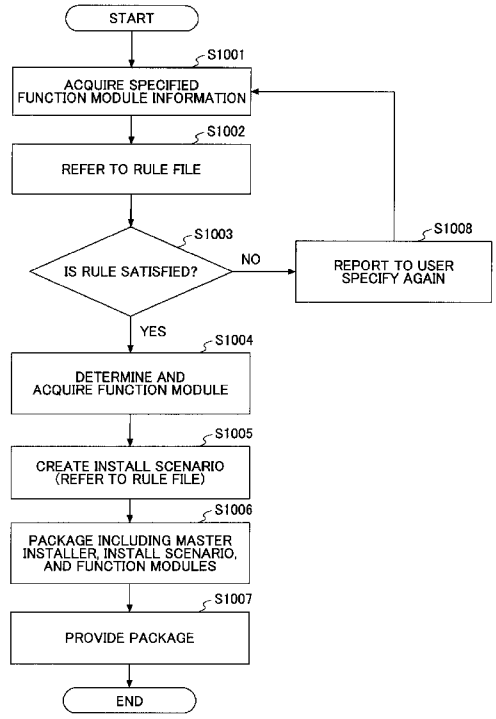
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

[Continued on next page]

(54) Title: SERVER DEVICE AND COMPUTER-READABLE RECORDING MEDIUM

[Fig. 10]



(57) Abstract: A server device, which provides a client device with a driver package including a core module of a driver, includes a storage unit for storing the core module, function modules, and a rule file defining packaging conditions for the function modules; a receiving unit for receiving, from the client device, a request specifying function modules; a determining unit for determining function modules to be included in the driver package from the stored function modules, based on the specified function modules and the rule file; a creating unit for creating an install scenario that specifies a method for installing the function modules when the driver package is installed in the client device, wherein the install scenario is created based on the determined function modules and the rule file; and a package generating unit for generating the driver package including the core module, the determined function modules, and the created install scenario.

WO 2010/089826 A1

Published:

— *with international search report (Art. 21(3))*

Description

Title of Invention: SERVER DEVICE AND COMPUTER-READABLE RECORDING MEDIUM

Technical Field

[0001] The present invention relates to a server device and a computer-readable recording medium in which a driver package generating program is stored for generating a driver package.

Background Art

[0002] A printer driver is a program that is installed in, for example, a user terminal, for giving printing instructions including various printing settings to a printer. Generally a printer driver is available on the Internet, and a user can download a printer driver corresponding to the machine type of the printer. FIG. 1 illustrates an example of a printer driver download page.

[0003] The printer driver download page may provide plural types of printer drivers for one machine type. For example, as shown in FIG. 1, different printer driver packages are provided according to the included functions, such as a standard (recommended) version, a lightweight version, and an enterprise version. Furthermore, each user can select the functions to be downloaded by selecting which function is required and which function is not required (optional function).

[0004] FIG. 2 illustrates an example of an internal data configuration of a printer driver download page. The printer driver download page and the driver packages (entities) on the server are linked to each other, and the driver packages corresponding to the download page can be downloaded from the server.

[0005] FIG. 3 schematically illustrates examples of files that are packaged in the driver packages. Different files (modules) are included according to the package. A driver installer is created for installing a driver core and various functions included in the package. A driver core must be included in every package, because the driver core is for providing essential functions of the printer driver. Functions A, B, and C are not necessarily essential functions, and may thus be selected according to purposes and requirements of the user.

[0006] As described above, a driver package includes multiple files, and each file is provided according to the function and task. A package does not need to include all of the files under regular conditions, unless the user wants to use all of the functions. Conventionally, a package is provided with files that are assumed to be typically used by users.

[0007] This is because it would be unrealistic to provide packages corresponding to all of

the combinations of functions, as costs would increase significantly. For example, in the above case, a total of 16 types of packages would need to be created by the provider (for example, the manufacturer), in order to provide all combinations of functions A, B, and C, in addition to the essential driver core. Therefore, a conventional package is provided with a reasonable number of files.

[0008] However, when all of the files (or many files) are packaged in the driver package, the package size increases significantly. Some users may not need all of the functions, in which case the package size would be needlessly large. Furthermore, complicated operations may be required to use the driver package because of the unnecessary functions for some users, thereby reducing user-friendliness. Nevertheless, as described above, it would be unrealistic to provide packages corresponding to all of the combinations of functions, because costs would increase significantly.

[0009] Patent document 1 describes a pertinent technology. Specifically, when downloading a package file of a driver, the package file is generated in accordance with the client device. Therefore, even if client devices are provided in a wide variety of environments, submodules can be appropriately added or changed, so that an appropriate package file of a device driver can be easily prepared and provided. That is, when a printer driver is downloaded, the server device selects functions that are required by the user, to generate a package file in which only necessary functions are packaged.

[0010] A printer is often provided with many additional options (functions). The basic function of a printer is a printing function. However, in recent years and continuing, technologies related to an MFP (Multi-Function Peripheral) are advancing, such as improved CPU performance, increased memory capacity, increased communications speed, and enhanced digital imaging technologies. Thus, an MFP may not only provide a printing function, but may also provide a variety of functions, thereby providing versatile usage formats in user environments. Accordingly, the corresponding printer driver would need to accommodate a large number of functions. Therefore, it would be unrealistic to provide packages corresponding to all of the combinations of functions.

[0011] Furthermore, in the invention of patent document 1, when performing the operation of selecting the functions required by the user and generating (providing) a package file packaging only the necessary functions, it is not good enough to merely package the necessary functions to generate the package file. In a printer having multiple functions, there may be option functions that are in an exclusive relationship or in a subordinate-superior relationship. The functions are to be combined in consideration of such mutual relationships between the functions according to the function properties. For example, if two functions having an exclusive relationship are included in the package, one of the packages is not actually used, and therefore a redundant function would be included in the package. In another example, when there are two functions

having a subordinate-superior relationship, but only the subordinate function is included in the package, the subordinate function cannot be actually used because the superior function is missing, and therefore a useless function would be included in the package.

[0012] One approach is to create packages corresponding to all of the combinations of functions in consideration of these mutual relationships between the functions; however, this is unrealistic because costs would increase significantly.

[0013] Patent Document 1: Japanese Laid-Open Patent Application No. 2006-244009

[0014] Accordingly, there is a need for a server device and a computer-readable recording medium in which a driver package generating program is stored for generating a driver package that can be easily provided by the provider, whereby the driver package is generated such that only function modules required by the user are packaged and unnecessary functions are not packaged.

Disclosure of Invention

[0015] Aspects of the present invention provide a server device and a computer-readable recording medium in which a driver package generating program is stored that solve or reduce one or more problems caused by the limitations and disadvantages of the related art.

[0016] An aspect of the present invention provides a server device for providing a client device with a driver package including at least a core module of a driver, the server device including a storage unit configured to store the core module, a group of function modules, and a rule file defining packaging conditions for the function modules included in the group of function modules, wherein the core module and the group of function modules are used as components of the driver package; a receiving unit configured to receive, from the client device, a driver package request specifying one or more of the function modules; a determining unit configured to determine one or more of the function modules to be included in the driver package from the group of function modules, based on the one or more of the function modules specified in the driver package request and the rule file; an install scenario creating unit configured to create an install scenario that specifies an installing method for installing the function modules included in the driver package when the driver package is installed in the client device, wherein the install scenario is created based on the one or more of the function modules determined to be included in the driver package by the determining unit and the rule file; and a package generating unit configured to generate the driver package including the core module, the one or more of the function modules determined to be included in the driver package by the determining unit, and the install scenario created by the install scenario creating unit.

[0017] An aspect of the present invention provides a computer-readable recording medium that stores therein a driver package generating program that causes a server device to execute a procedure for providing a client device with a driver package including at least a core module of a driver, the procedure including a storage step of storing the core module, a group of function modules, and a rule file defining packaging conditions for the function modules included in the group of function modules, wherein the core module and the group of function modules are used as components of the driver package; a receiving step of receiving, from the client device, a driver package request specifying one or more of the function modules; a determining step of determining one or more of the function modules to be included in the driver package from the group of function modules, based on the one or more of the function modules specified in the driver package request and the rule file; an install scenario creating step of creating an install scenario that specifies an installing method for installing the function modules included in the driver package when the driver package is installed in the client device, wherein the install scenario is created based on the one or more of the function modules determined to be included in the driver package at the determining step and the rule file; and a package generating step of generating the driver package including the core module, the one or more of the function modules determined to be included in the driver package at the determining step, and the install scenario created at the install scenario creating step.

Brief Description of Drawings

[0018] [fig.1]FIG. 1 illustrates an example of a printer driver download page;
[fig.2]FIG. 2 illustrates an example of an internal data configuration of a printer driver download page;
[fig.3]FIG. 3 schematically illustrates examples of files that are packaged in the driver packages;
[fig.4]FIG. 4 illustrates a network configuration according to an embodiment of the present invention;
[fig.5]FIG. 5 illustrates a main hardware configuration of a server according to an embodiment of the present invention;
[fig.6]FIG. 6 is a functional block diagram of the main functions of the server according to an embodiment of the present invention;
[fig.7]FIG. 7 schematically illustrates a configuration of stored data;
[fig.8]FIG. 8 illustrates an example of the contents described in a rule file;
[fig.9]FIG. 9 illustrates a concept of acquiring a driver package from the server;
[fig.10]FIG. 10 is a flowchart for describing a processing operation of the server;
[fig.11]FIG. 11 illustrates an example of an install scenario;

[fig.12]FIG. 12 illustrates an example of data included in a driver package;

[fig.13]FIG. 13 is a flowchart for describing the process of installing a driver package;

[fig.14]FIG. 14 illustrates an install scenario; and

[fig.15]FIG. 15 illustrates another concept of acquiring a driver package from the server.

Best Mode for Carrying out the Invention

[0019] Embodiments of the present invention are described below with reference to the accompanying drawings.

[0020] <Configuration>

[0021] First, before describing details of the present invention, a description is given of a system configuration for implementing an embodiment of the present invention. FIG. 4 illustrates a network configuration according to an embodiment of the present invention. A server 1 and a client PC (hereinafter, "PC") 2 according to an embodiment of the present invention are connected via a network 3 such as the Internet. The server 1 is a server device for providing a printer driver package to the PC 1. Furthermore, the server 1 has a Web function to facilitate access from the PC 1 with the use of Web technology.

[0022] The PC 1 is a user terminal, and downloads the necessary printer driver from the server 1 via a Web browser. The printer driver package is installed in the PC 1, and gives printing instructions to a printer 4. The printer 4 is a printer device such as an image forming apparatus, which receives the printing instructions from the printer driver of the PC 1 and executes the printing instruction.

[0023] The server 1 has a storage unit (e.g., a database) as described below. However, the function of the storage unit does not need to be in the same casing as the server 1. The function of the storage unit may be provided in a file server connected to the network 3.

[0024] (Server)

[0025] FIG. 5 illustrates a main hardware configuration of the server 1 according to an embodiment of the present invention. A brief description is given of the hardware configuration of the server 1. The server 1 may be a typical information processing device, including a CPU 101, a ROM (Read Only Memory) 102, a RAM (Random Access Memory) 103, a secondary storage device 104, a storage medium reading device 105, an input device 106, a display device 107, and a communications device 108.

[0026] The CPU 101 includes a microprocessor and peripheral circuits, and controls the entire PC 1. The ROM 102 is a memory for storing predetermined control programs (software components) executed by the CPU 101. The RAM 103 is used as a work area when the CPU 101 executes the predetermined control programs (software

components) stored in the ROM 102 to implement various control operations.

[0027] The secondary storage device 104 is, for example, a HDD (Hard Disk Drive) that is a nonvolatile storage device, for storing various information items including a general-purpose OS (Operating System) and a printer driver. The above-described information items may be stored not only in the secondary storage device 104 but also in a storage medium such as a CD-ROM (Compact Disk-ROM) and a DVD (Digital Versatile Disk). The various information items stored in such a storage medium may be read via a drive device such as the storage medium reading device 105.

[0028] The input device 106 is used by the server administrator to perform various input operations. The input device 106 includes a mouse and a keyboard. The display device 107 is a so-called display screen which may be a LCD (Liquid Crystal Display) or a CRT (Cathode Ray Tube). The communications device 108 is for performing communications with the PC 2 via the network 3, and supports communications according to various network formats including a wired network and a wireless network.

[0029] FIG. 6 is a functional block diagram of the main functions of the server 1 according to an embodiment of the present invention. The main function units are a storage unit 601, a receiving unit 602, a determining unit 603, an install scenario creating unit 604, and a package generating unit 605. A brief description is given herein, and further details of functional operations of the function units are described below.

[0030] The storage unit 601 includes a core module and a group of function modules that are used as components of the driver package, as well as a rule file defining packaging conditions for the function modules. The data stored in the storage unit 601 is described below.

[0031] The receiving unit 602 is for receiving, from the PC 2, a driver package request specifying a function module. The driver package request includes information indicating the function that the user wants to include in the driver package, (functions that the user wants to use with the printer 4 by means of the driver), such as a function A and a function B.

[0032] The determining unit 603 is for determining the function module to be included in the driver package among the group of function modules in the storage unit 601, based on the function module specified by the user and the rule file in the storage unit 601.

[0033] The install scenario creating unit 604 is for creating an install scenario for specifying the method of installing the modules when the driver package is installed in the PC 2. The install scenario is created based on the function module determined at the determining unit 603 and the rule file in the storage unit 601.

[0034] The package generating unit 605 is for generating a driver package including the core module, the determined function module, and the created install scenario. The core module is an essential module that must be included in every printer driver. The driver

package also includes a master installer, which is described below.

[0035] The above-described functions are actually implemented by a computer (server 1) according to a driver package generating program executed by the CPU 101. The driver package generating program may be stored in a computer-readable recording medium for causing the computer (server 1) to execute a procedure of the driver package generating program.

[0036] (Data configuration)

[0037] A description is given of a configuration of the data stored in the storage unit 601. FIG. 7 schematically illustrates a configuration of the stored data. As described above, the storage unit 601 stores a core module, a group of function modules, and a rule file.

[0038] The core module and each of the function modules have corresponding installers, as shown in FIG. 7. The module entities are data items that are independent from each other. The essential core module and the required function modules are packaged in the driver package. When the driver package is installed in the PC 2, the driver package is first opened. In accordance with the procedures specified in the install scenario, the respective modules are installed. Each module is installed by its own installer.

[0039] The rule file defines the packaging conditions of the function modules. The rule file is released at the same time as the modules (components) of the driver. For example, the rule file defines exclusive relationships between the function modules, and the determining unit 603 determines the function modules to be included in the driver package among the group of function modules, based on the user-specified function modules and the exclusive relationships defined by the rule file. That is, when mutually exclusive function modules are included among the function modules specified by the user, one of the functions included in the exclusive relationship is not included in the package. Furthermore, the rule file defines the order of installing the function modules, as well as install attributes. The install scenario creating unit 604 creates an install scenario specifying the install order and the install attributes to be applied when installing the function modules determined by the determining unit 603 in the PC 2. Specifically, an appropriate install scenario is created by referring to the rule file.

[0040] FIG. 8 illustrates an example of the contents described in the rule file. For example, in the rule file, property conditions for the functions are listed in a CSV format. For example, in FIG. 8, the "rule for function A" defines "Any_Core, Must". "Any_Core, Must" means that "function A must include one of the core modules".

[0041] Furthermore, "PCL5_Core, IF_Opt, -pcl5" means that "when function A is installed together with the core module of PCL5, the option -pcl5 is specified". Specifically, function A includes a Locked Print function for each PDL, and can thus accommodate any core module. However, when the core module is PCL5, it is specified that function

A is to be installed as a function for PCL5. Thus, an install option is specified for the installer of function A.

[0042] Next, the "rule for function B" defines "FuncD, Want", which means that "function (module) D is recommended for function B". Function D is an extended function of function B, and therefore a higher function can be provided by adding function D. Obviously, function D can only be added after function B has been added. For example, when function B is a basic tint block function and function D is an extended tint block function, the PC 2 needs to have the basic tint block function installed as the base function, in order to install and use the extended tint block function in the PC 2. This is because the extended function uses various process units of the basic function. In another example, when function B is a watermark function and function D is a (basic) tint block function, and the tint block function is installed alone, only a simplified version of the tint block function can be used. However, if the tint block function is added after the watermark function is installed, a higher version of the tint block function can be used., With such a higher version, the watermark function can also be fully implemented.

[0043] Furthermore, the "rule for function C" defines "FuncA, Not", which means that "function C excludes function (module) A". That is, function C and function A have a mutually exclusive relationship in terms of their functions and configurations, and thus cannot be simply installed together. For example, function C may be a function for a PS Driver (for example, a PS Passthrough function), which cannot be added to the core of a PCL6 Driver. Next, a description is given of the processing operations of the server 1 by specifically applying the above-described rules.

[0044] <Operation>

[0045] FIG. 9 illustrates a concept of acquiring a driver package from the server 1. When the user specifies (selects) necessary functions from a printer driver download page (Web screen page, etc.), the server 1 dynamically acquires the specified functions (modules), performs a packaging process to generate a driver package, and provides the generated driver package to the user. The "PCL6 Driver (core)" is an essential module, and therefore there may be a default setting such that the "PCL6 Driver (core)" is mandatorily specified.

[0046] FIG. 10 is a flowchart for describing the processing operation of the server 1. Specifically, FIG. 10 is for describing the driver package generating process.

[0047] In step S1001, when a driver package request specifying function modules is received from the PC 2, the receiving unit 602 acquires the specified function module information. It is assumed that the specified function modules are "PCL6 Driver (core)", "PCL6 Driver (function A)", "PCL6 Driver (function B)", and "PCL6 Driver (function C)", so as to correspond to FIG. 9.

[0048] In step S1002, the determining unit 603 refers to the rule file to determine whether the specified function modules have an exclusive relationship.

[0049] In step S1003, the determining unit 603 determines whether the rules are satisfied. When the specified function modules do not have an exclusive relationship, the process proceeds to step S1004, where the determining unit 603 determines that the specified function modules are to be included in the driver package, and acquires the specified function modules (entities) from the storage unit 601.

[0050] When the specified function modules have an exclusive relationship, in step S1008, a pertinent report is sent to the user, and a screen page prompting the user to make another setting is displayed. The screen page preferably prompts the user to make a specification (selection) that satisfies the rules, by notifying the user which specific function cannot be incorporated in the driver package.

[0051] In this example, the specified function modules are "PCL6 Driver (core)", "PCL6 Driver (function A)", "PCL6 Driver (function B)", and "PCL6 Driver (function C)", as described above. Referring to the rule file shown in FIG. 8, the function modules "PCL6 Driver (function A)" and "PCL6 Driver (function C)" have an exclusive relationship, and therefore the process proceeds to step S1008. Then, a report is sent to the user, saying that "PCL6 Driver (function C)" cannot be specified. Then, it is assumed that the user specifies "PCL6 Driver (core)", "PCL6 Driver (function A)", and "PCL6 Driver (function B)" in a new setting. Other than reporting an exclusive relationship, the report to the user may also indicate that "PCL6 Driver (function D)" is recommended for "PCL6 Driver (function B)", to prompt the user to make another setting for including "PCL6 Driver (function D)".

[0052] In step S1005, the install scenario creating unit 604 refers to the rule file once again, and creates an install scenario specifying the method for installing the determined function modules. FIG. 11 illustrates an example of an install scenario. The install scenario schematically defines the order of installing the modules. More details of the install scenario are described below. The description of the processing operation is continued.

[0053] In step S1006, the package generating unit 605 generates a driver package including the core module, the determined function modules, and the created install scenario. The package generating unit 605 includes the master installer in the driver package, and then packages the function modules (compresses the function modules to convert them into a single file). The master installer is for the whole driver package. Actually, the master installer is for activating the installers included in each of the modules, in the order defined (described) in the install scenario.

[0054] FIG. 12 illustrates an example of data included in a driver package. A driver package, which has been dynamically generated according to a user specification, includes a

master installer, an install scenario, a "PCL6 Driver core entity", a "PCL6 Driver function A entity", and a "PCL6 Driver function B entity".

[0055] In step S1007, the server 1 provides the generated driver package to the PC 2.

[0056] (Installing process)

[0057] The driver package that has been downloaded to the PC 2 is installed in the PC 2 as a printer driver before being used. A description is given of a flow of installing the driver package in the PC 2, in order to describe the master installer and the install scenario in more detail.

[0058] FIG. 13 is a flowchart for describing the process of installing a driver package. The driver package generated at the server 1 is downloaded to the PC 2. The driver package is installed in the PC 2 so that the driver package can be used.

[0059] In step S1301, the PC 2 decompresses the driver package, because the driver package had been compressed at the server 1.

[0060] In step S1302, the master installer is activated. The master installer is for the whole driver package.

[0061] In step S1303, the master installer reads (refers to) the install scenario. The master installer activates the installers included in each of the modules, in the order that is defined (described) in the install scenario.

[0062] In step S1304, the installers are activated in the order defined in the install scenario. Referring back to FIG. 11, ".\Core\Install.exe" is described as [Driver File1]. Thus, the master installer activates ".\Core\Install.exe", which is the installer program. The ".\Core\Install.exe" corresponds to the "core installer" in the "PCL6 Driver core entity", with reference to FIG. 12.

[0063] In step S1305, each module installs all of the files included in itself. That is, each module has its own installer, so when the installer is activated, all of the files included in the module can be installed. Referring back to FIG. 11, when the "core installer" in the "PCL6 Driver core entity" is activated, the "driver core" is installed. As described above, the master installer simply activates the installers of the modules in the defined order, and then the installers perform the installing process for the corresponding modules.

[0064] In step S1306, the server 1 determines whether all of the installers in the install scenario have been activated. When all of the installers have been activated, it means that all of the modules (core driver, function drivers) have been installed, and therefore the installing process ends. When all of the installers have not been activated, the same process as described above is performed until all of the installers have been activated in order. Specifically, the master installer activates the installers in the order of ".\Core\Install.exe", ".\FuncA \Install.exe", and ".\FuncB \Install.exe".

[0065] (Supplementary description)

[0066] The above describes the process of generating a driver package performed by the server 1. A supplementary description is given for cases that have not been described with reference to the rule file.

[0067] A description is given of a case where the specified modules are "PCL6 Driver (core)", "PCL6 Driver (function A)", "PCL6 Driver (function B)", and "PCL6 Driver (function D)".

[0068] Referring back to FIG. 8, the "rule for function A" specifies "Any_Core, Must" and "PCL6_Core, IF_Opt, -pcl6". This means that for function A, a core module must be included, and when the core module is a PCL6 version, "-pcl6" is to be specified as an install option.

[0069] Specifically, function A includes a Locked Print function for each PDL, and can thus accommodate any core module. The rule specifies that when the core module is PCL6, function A is to be installed as a function for PCL6. Therefore, "-pcl6" is specified as an install option for the installer of function A.

[0070] This means that function A may include only a part of the function, which corresponds to PCL6. There is obviously no problem if the entire function of function A is included. However, if the core module is PCL6, it is only necessary to install the part of the function corresponding to PCL6. Accordingly, only a part of function A is installed, and therefore unnecessary functions and data can be reduced. When there are no particular rules for the function, function A is packaged such that the entire function A is included.

[0071] As described above, with the use of a rule file, the characteristics and properties of the functions are taken into consideration, and the install options are determined based on the characteristics of the functions. Thus, a driver such as a printer driver, which includes functions having various characteristics, can be installed without including unnecessary functions.

[0072] Referring back to FIG. 8, "rule for function B" specifies that function D is recommended for function B. Thus, function D is an extended function of function B, and therefore function D can only be added after function B has been added. For example, when function B is a basic tint block function and function D is an extended tint block function, the PC 2 needs to have the basic tint block function installed as the base function, in order to install and use the extended tint block function in the PC 2. This is because the extended function uses various process units of the basic function. Thus, the created install scenario describes that the functions are to be installed in the order of "PCL6 Driver (function B)", "PCL6 Driver (function D)". The install scenario should clearly indicate that the functions are not to be installed in the order of "PCL6 Driver (function D)", "PCL6 Driver (function B)".

[0073] As described above, with the use of the rule file, the characteristics of the functions

are taken into consideration, and the order of installing the functions is determined based on the characteristics of the function. Thus, a driver such as a printer driver, which includes functions having various characteristics, can be reliably and correctly installed.

[0074] Referring back to FIG. 8, "rule for function B" defines "FuncD, IF_Opt, -Half". This means that although function D is an extended function of function B, only half or a part of function D (-Half) is required. There is obviously no problem if the entire function of function D is included (-Full). However, in consideration of the relationship with function B, function D can be sufficiently implemented with half or part of function D (-Half). Accordingly, only half or part of function D is installed, and therefore unnecessary functions and data can be reduced. When there are no particular rules for the function, function D is packaged such that the entire function D is included (-Full).

[0075] FIG. 14 illustrates an install scenario created in the above manner. The relationship between function B and function D is indicated. It is clearly indicated that ".\FuncD\Install.exe" is to be executed after ".\FuncB\Install.exe" has been executed (activated). As for function D, ".\FuncD\Install.exe -Half" is described, clearly indicating that half or part of function D (-Half) is to be installed.

[0076] <Modification>

[0077] FIG. 15 illustrates another concept of acquiring a driver package from the server 1. In the printer driver download page illustrated in FIG. 9, the user simply checked the check boxes corresponding to the required functions.

[0078] In the printer driver download page illustrated in FIG. 15, different printer driver packages are provided according to the included functions. For example, a standard (recommended) version, a lightweight version, and an enterprise version are provided. The driver packages are provided in units that are made clear to the user by the concept or catch-phrase. Accordingly, compared to the case of presenting check boxes, the user can select a driver package that substantially matches his requirements based on the indicated concept, even when the user is unaware of the specific functions included in each of the driver packages.

[0079] In FIG. 15, when "PCL6 Driver (lightweight version)" is specified (selected), the server 1 identifies the specified function. In this case, "PCL6 Driver (function A entity)" is identified as the specified function module. Therefore, "install scenario", "PCL6 Driver (driver core entity)" (essential), and "PCL6 Driver (function A entity)" (specified) are packaged in the final driver package provided to the user.

[0080] The printer driver download page may be created by embedding the information of "PCL6 Driver (function A entity)" as the specified function module, in the link of the "PCL6 Driver (lightweight version)". In another example, a predetermined flag may be

embedded in the link of the "PCL6 Driver (lightweight version)", and the function module corresponding to the predetermined flag may be determined at the server 1.

[0081] An embodiment of the present invention described above provides a server device and a computer-readable recording medium in which a driver package generating program is stored for generating a driver package that can be easily provided by the provider, whereby the driver package is generated such that only function modules required by the user are packaged and unnecessary functions are not packaged.

[0082] The present invention is not limited to the specifically disclosed embodiment, and variations and modifications may be made without departing from the scope of the present invention. For example, the present invention is not only applicable to printer driver packages, but also to peripheral devices including various functions having particular relationships.

[0083] The present application is based on Japanese Priority Application No. 2009-025293 filed on February 5, 2009 with the Japan Patent Office, the entire contents of which are hereby incorporated by reference.

Claims

[Claim 1] A server device for providing a client device with a driver package including at least a core module of a driver, the server device comprising:
a storage unit configured to store the core module, a group of function modules, and a rule file defining packaging conditions for the function modules included in the group of function modules, wherein the core module and the group of function modules are used as components of the driver package;
a receiving unit configured to receive, from the client device, a driver package request specifying one or more of the function modules;
a determining unit configured to determine one or more of the function modules to be included in the driver package from the group of function modules, based on the one or more of the function modules specified in the driver package request and the rule file;
an install scenario creating unit configured to create an install scenario that specifies an installing method for installing the function modules included in the driver package when the driver package is installed in the client device, wherein the install scenario is created based on the one or more of the function modules determined to be included in the driver package by the determining unit and the rule file; and
a package generating unit configured to generate the driver package including the core module, the one or more of the function modules determined to be included in the driver package by the determining unit, and the install scenario created by the install scenario creating unit.

[Claim 2] The server device according to claim 1, wherein
the rule file defines an exclusive relationship between the function modules included in the group of function modules; and
the determining unit determines the one or more of the function modules to be included in the driver package based on the one or more of the function modules specified in the driver package request and the exclusive relationship defined by the rule file.

[Claim 3] The server device according to claim 1, wherein
the rule file defines an installing order and installing properties of the function modules included in the group of function modules; and
the install scenario creating unit creates the install scenario that specifies the installing order and the installing properties of the one or

more of the function modules determined to be included in the driver package by the determining unit.

[Claim 4] The server device according to claim 1, further comprising:
a reporting unit configured to send a report to the client device when the one or more of the function modules specified in the driver package request and the one or more of the function modules determined to be included in the driver package by the determining unit are different.

[Claim 5] A computer-readable recording medium that stores therein a driver package generating program that causes a server device to execute a procedure for providing a client device with a driver package including at least a core module of a driver, the procedure comprising:
a storage step of storing the core module, a group of function modules, and a rule file defining packaging conditions for the function modules included in the group of function modules, wherein the core module and the group of function modules are used as components of the driver package;
a receiving step of receiving, from the client device, a driver package request specifying one or more of the function modules;
a determining step of determining one or more of the function modules to be included in the driver package from the group of function modules, based on the one or more of the function modules specified in the driver package request and the rule file;
an install scenario creating step of creating an install scenario that specifies an installing method for installing the function modules included in the driver package when the driver package is installed in the client device, wherein the install scenario is created based on the one or more of the function modules determined to be included in the driver package at the determining step and the rule file; and
a package generating step of generating the driver package including the core module, the one or more of the function modules determined to be included in the driver package at the determining step, and the install scenario created at the install scenario creating step.

[Claim 6] The computer-readable recording medium according to claim 5, wherein
the storage step includes storing the rule file defining an exclusive relationship between the function modules included in the group of function modules; and
the determining step includes determining the one or more of the

function modules to be included in the driver package based on the one or more of the function modules specified in the driver package request and the exclusive relationship defined by the rule file.

[Claim 7] The computer-readable recording medium according to claim 5, wherein

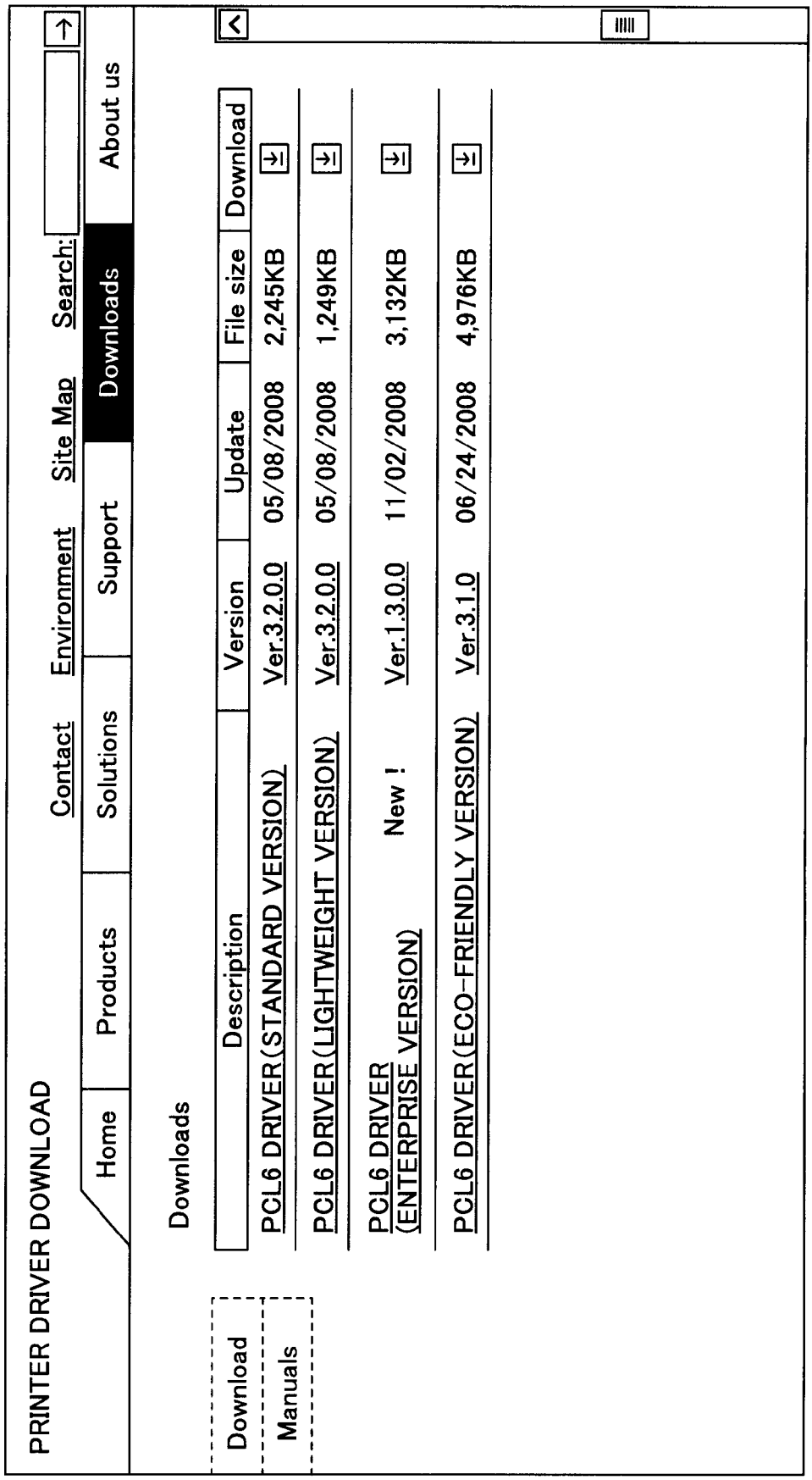
the storage step includes storing the rule file defining an installing order and installing properties of the function modules included in the group of function modules; and

the install scenario creating step includes creating the install scenario that specifies the installing order and the installing properties of the one or more of the function modules determined to be included in the driver package at the determining step.

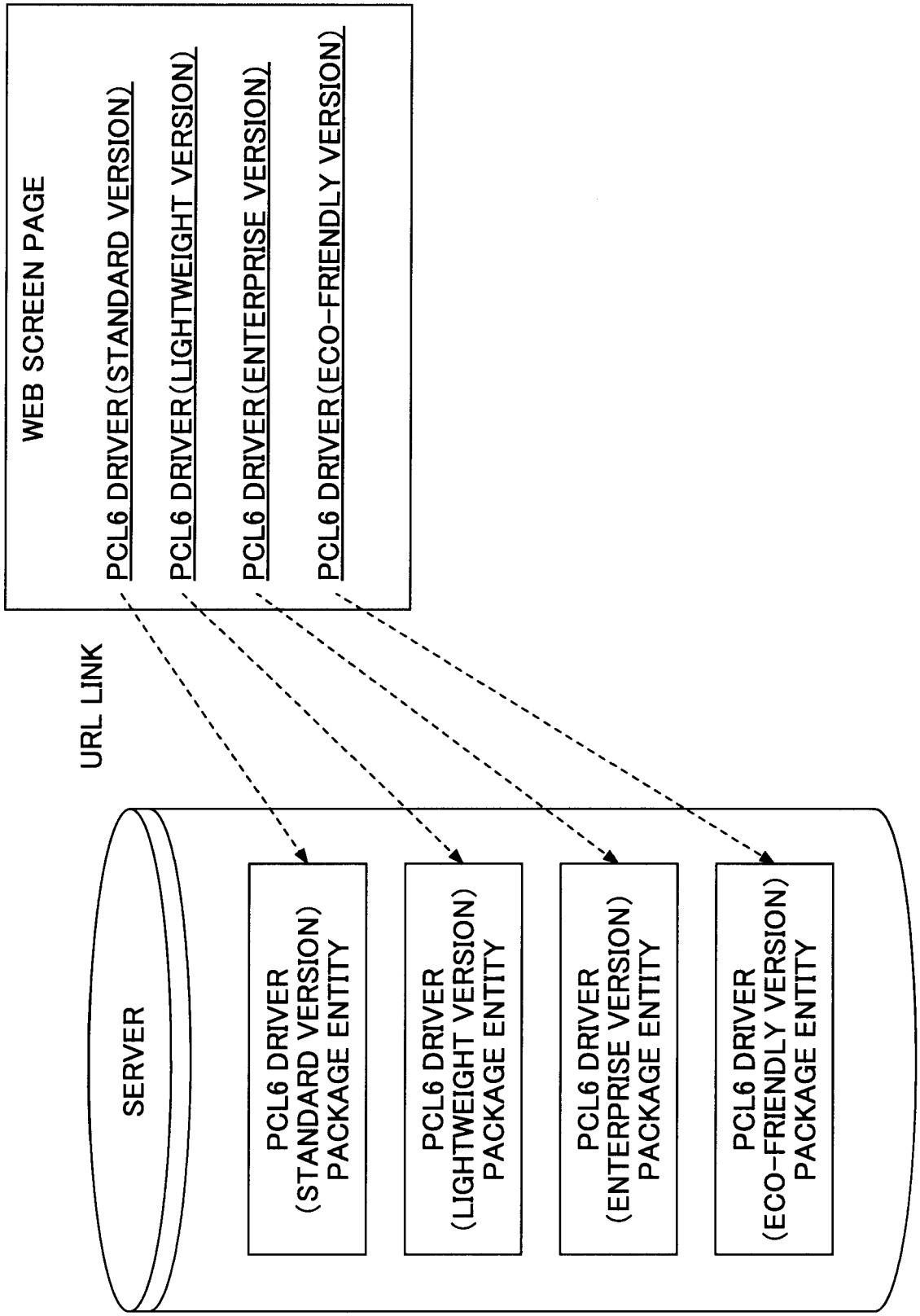
[Claim 8] The computer-readable recording medium according to claim 5, wherein the procedure further comprises:

a reporting step of sending a report to the client device when the one or more of the function modules specified in the driver package request and the one or more of the function modules determined to be included in the driver package at the determining step are different.

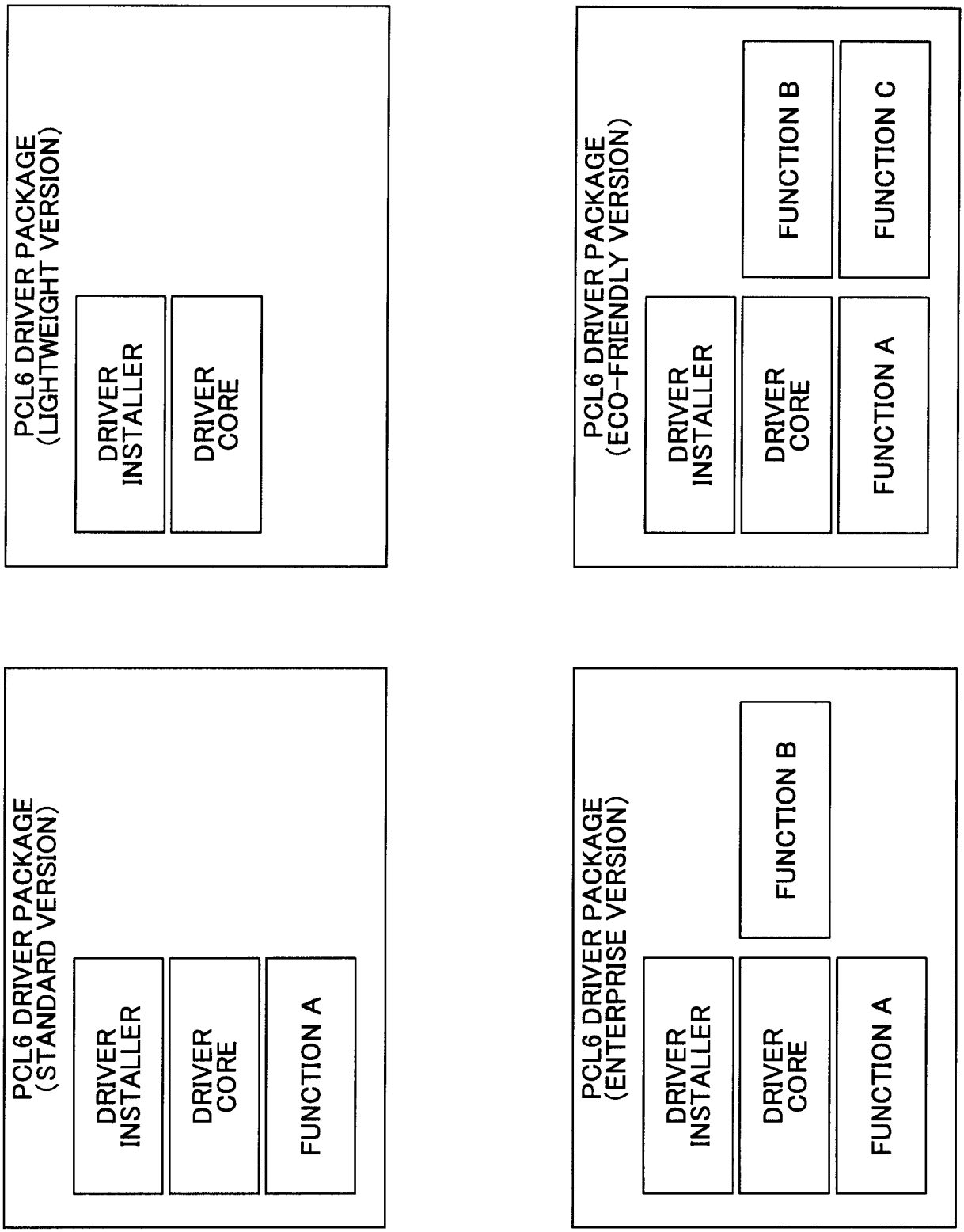
[Fig. 1]


PRINTER DRIVER DOWNLOAD
Contact
Environment
Site Map
Search:
Home
Products
Solutions
Support
Downloads
About us
Downloads
Download
Manuals
Description
Version
Update
File size
Download
PCL6 DRIVER(STANDARD VERSION)
Ver.3.2.0.0
05/08/2008
2,245KB
PCL6 DRIVER(LIGHTWEIGHT VERSION)
Ver.3.2.0.0
05/08/2008
1,249KB
PCL6 DRIVER (ENTERPRISE VERSION)
New !
Ver.1.3.0.0
11/02/2008
3,132KB
PCL6 DRIVER(ECO-FRIENDLY VERSION)
Ver.3.1.0
06/24/2008
4,976KB

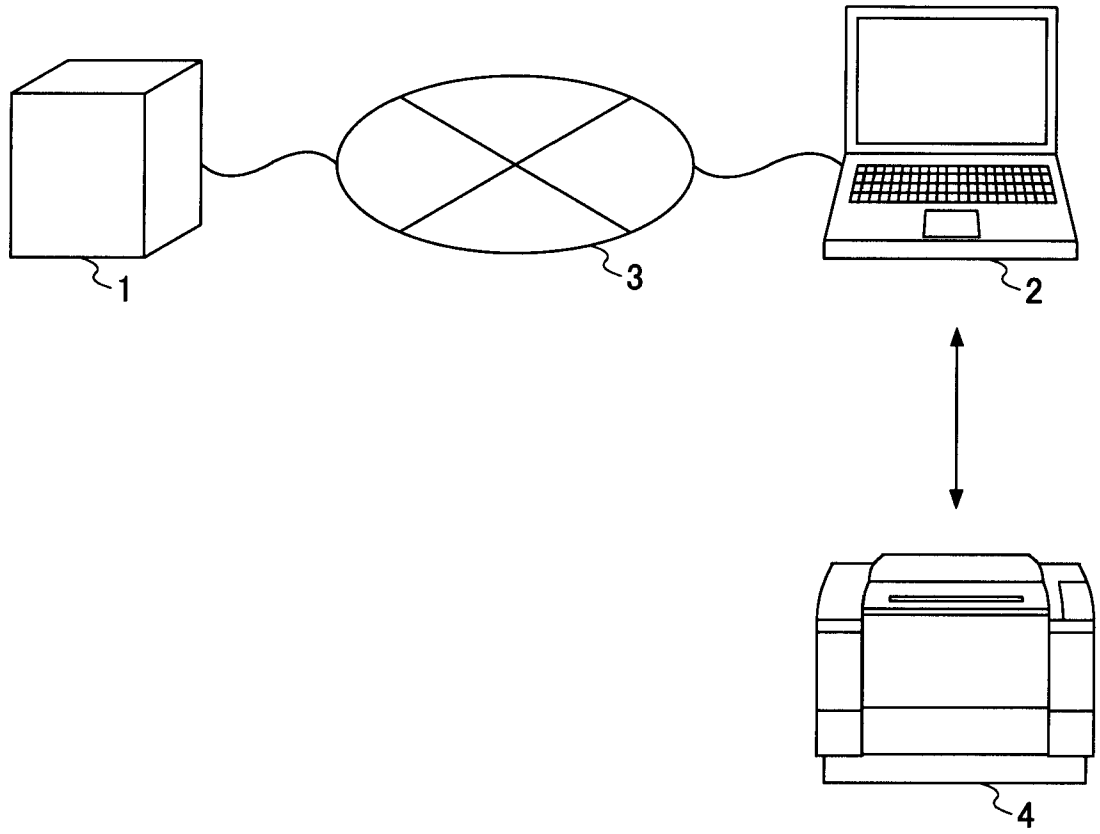
[Fig. 2]


WEB SCREEN PAGE
PCL6 DRIVER(STANDARD VERSION)
PCL6 DRIVER(LIGHTWEIGHT VERSION)
PCL6 DRIVER(ENTERPRISE VERSION)
PCL6 DRIVER(ECO-FRIENDLY VERSION)
URL LINK
SERVER
PCL6 DRIVER (STANDARD VERSION) PACKAGE ENTITY
PCL6 DRIVER (LIGHTWEIGHT VERSION) PACKAGE ENTITY
PCL6 DRIVER (ENTERPRISE VERSION) PACKAGE ENTITY
PCL6 DRIVER (ECO-FRIENDLY VERSION) PACKAGE ENTITY

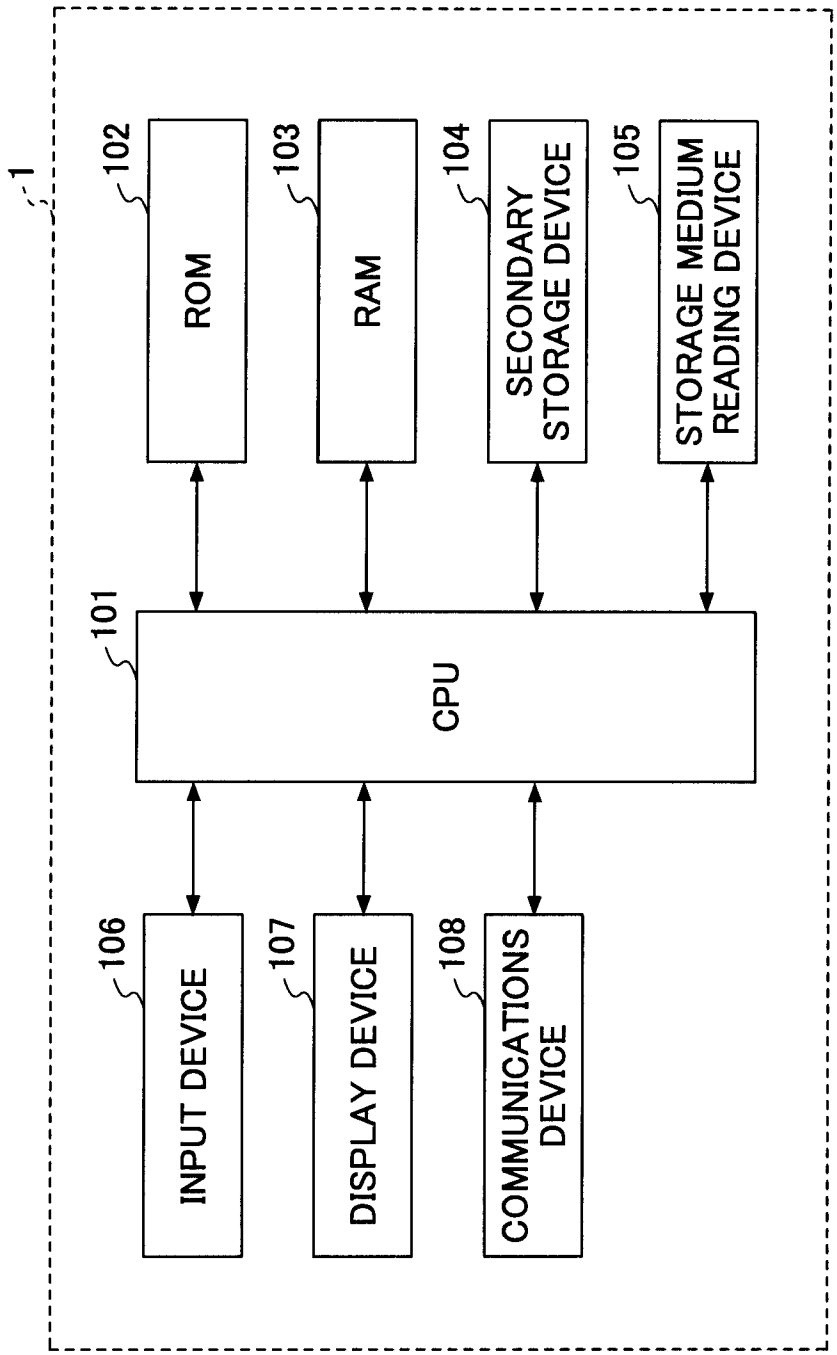
[Fig. 3]


PCL6 DRIVER PACKAGE
(LIGHTWEIGHT VERSION)
DRIVER INSTALLER
DRIVER CORE
PCL6 DRIVER PACKAGE
(ECO-FRIENDLY VERSION)
DRIVER INSTALLER
DRIVER CORE
FUNCTION A
FUNCTION B
FUNCTION C
PCL6 DRIVER PACKAGE
(STANDARD VERSION)
DRIVER INSTALLER
DRIVER CORE
FUNCTION A
PCL6 DRIVER PACKAGE
(ENTERPRISE VERSION)
DRIVER INSTALLER
DRIVER CORE
FUNCTION A
FUNCTION B

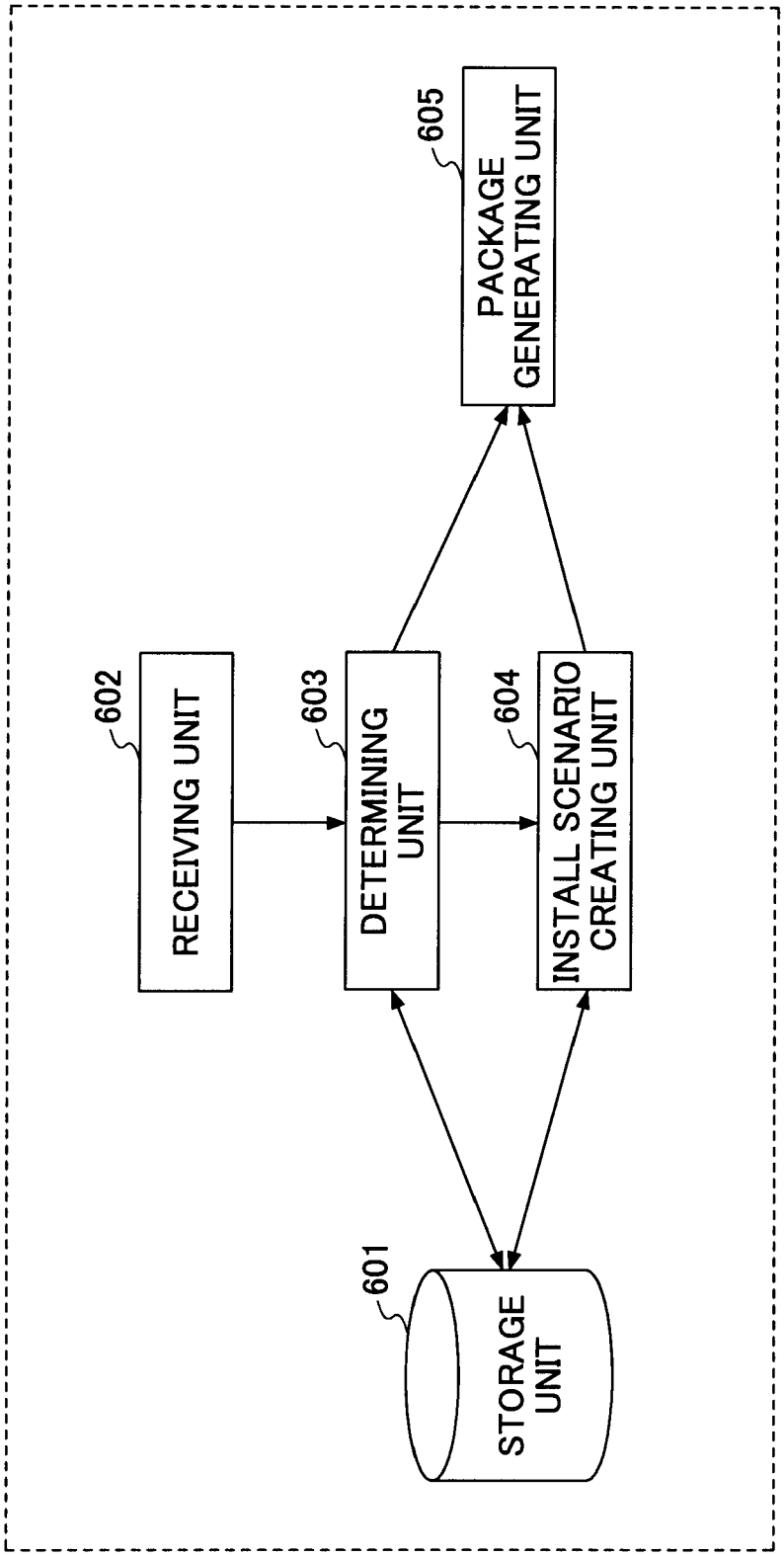
[Fig. 4]



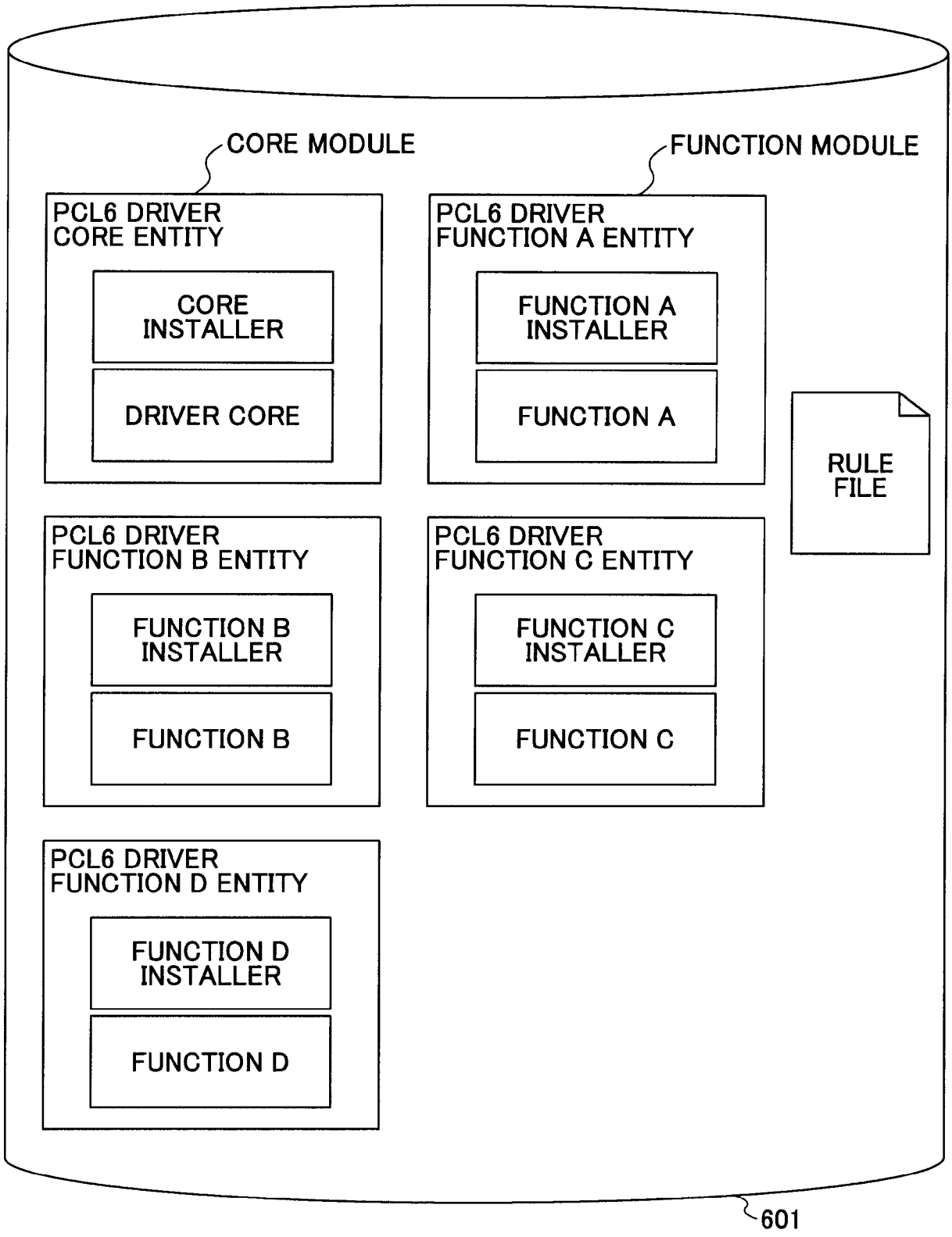
[Fig. 5]


1
101
CPU
102
ROM
103
RAM
104
SECONDARY STORAGE DEVICE
105
STORAGE MEDIUM READING DEVICE
106
INPUT DEVICE
107
DISPLAY DEVICE
108
COMMUNICATIONS DEVICE

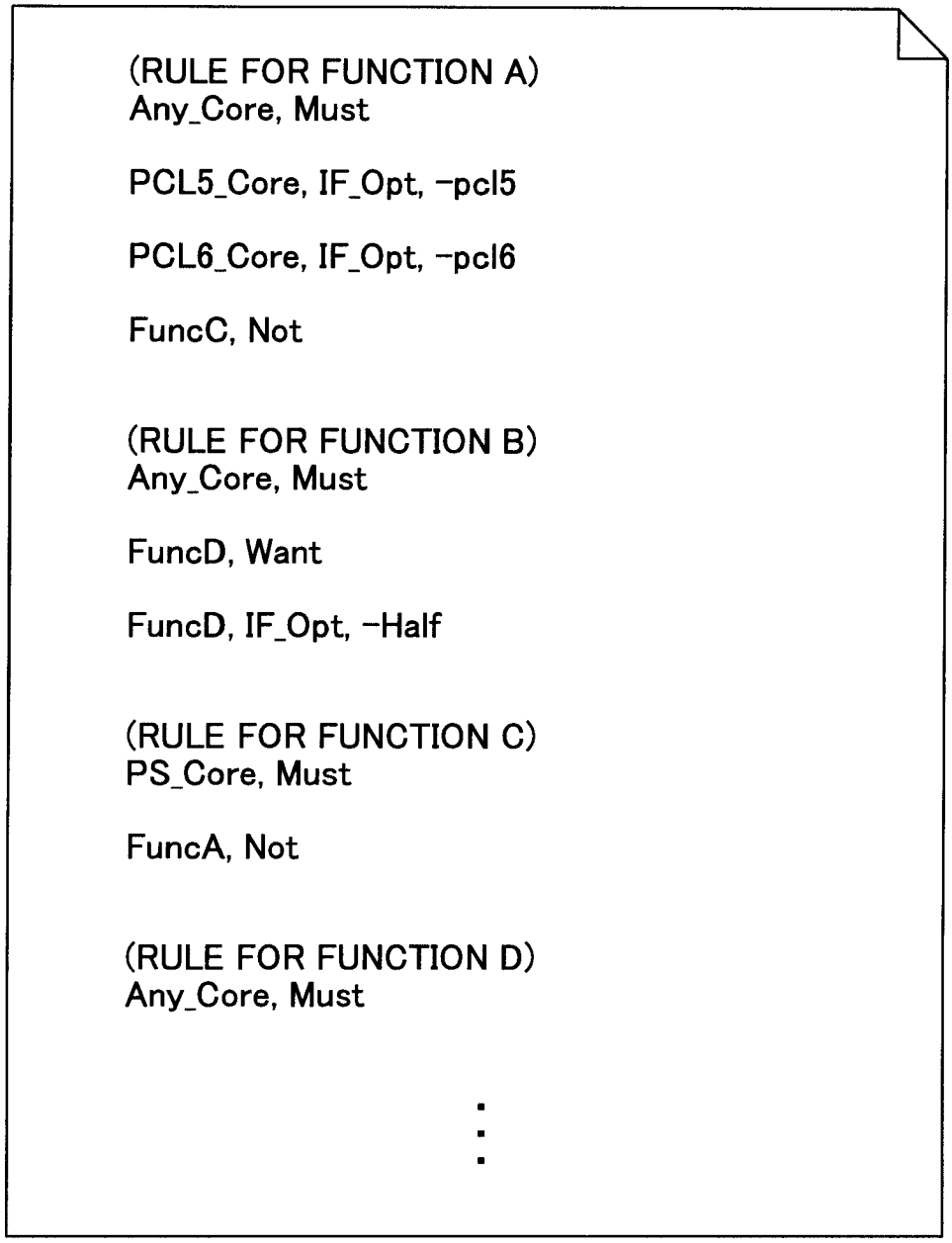
[Fig. 6]


605
PACKAGE GENERATING UNIT
602
RECEIVING UNIT
603
DETERMINING UNIT
604
INSTALL SCENARIO CREATING UNIT
601
STORAGE UNIT

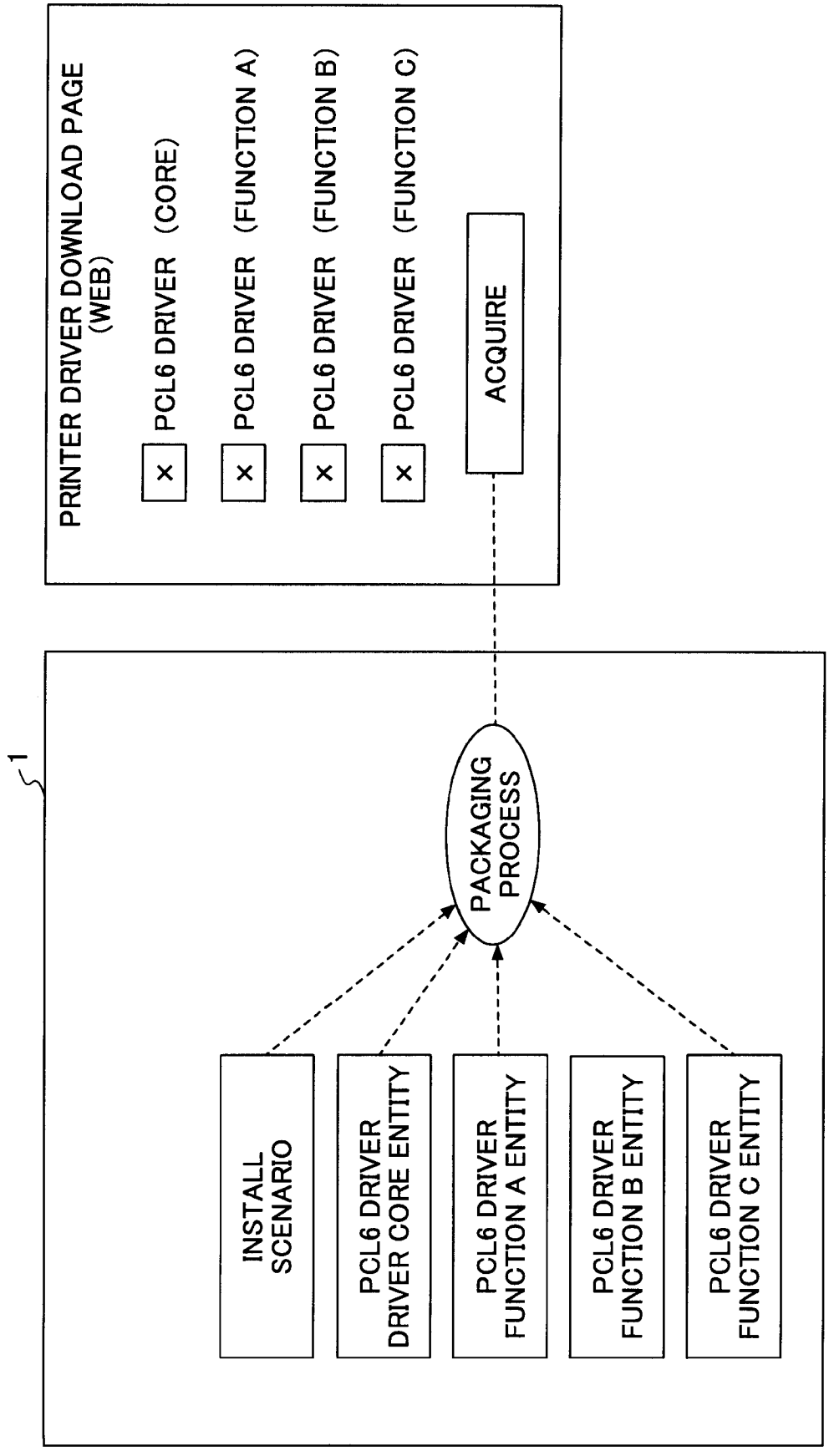
[Fig. 7]


CORE MODULE
FUNCTION MODULE
PCL6 DRIVER
CORE ENTITY
CORE
INSTALLER
DRIVER CORE
PCL6 DRIVER
FUNCTION A ENTITY
FUNCTION A
INSTALLER
FUNCTION A
RULE
FILE
PCL6 DRIVER
FUNCTION B ENTITY
FUNCTION B
INSTALLER
FUNCTION B
PCL6 DRIVER
FUNCTION C ENTITY
FUNCTION C
INSTALLER
FUNCTION C
PCL6 DRIVER
FUNCTION D ENTITY
FUNCTION D
INSTALLER
FUNCTION D
601

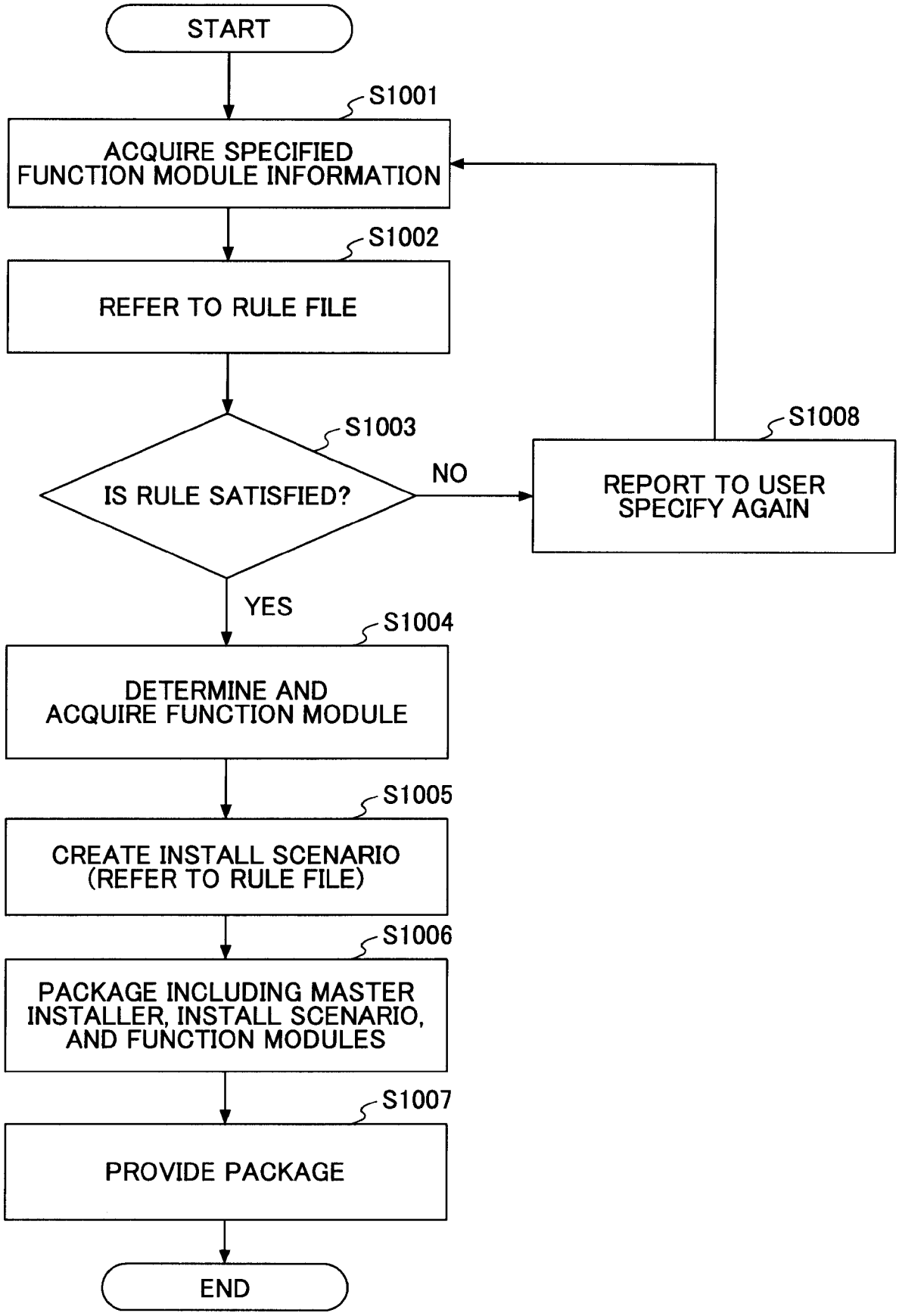
[Fig. 8]


(RULE FOR FUNCTION A)
Any_Core, Must
PCL5_Core, IF_Opt, -pcl5
PCL6_Core, IF_Opt, -pcl6
FuncC, Not
(RULE FOR FUNCTION B)
Any_Core, Must
FuncD, Want
FuncD, IF_Opt, -Half
(RULE FOR FUNCTION C)
PS_Core, Must
FuncA, Not
(RULE FOR FUNCTION D)
Any_Core, Must

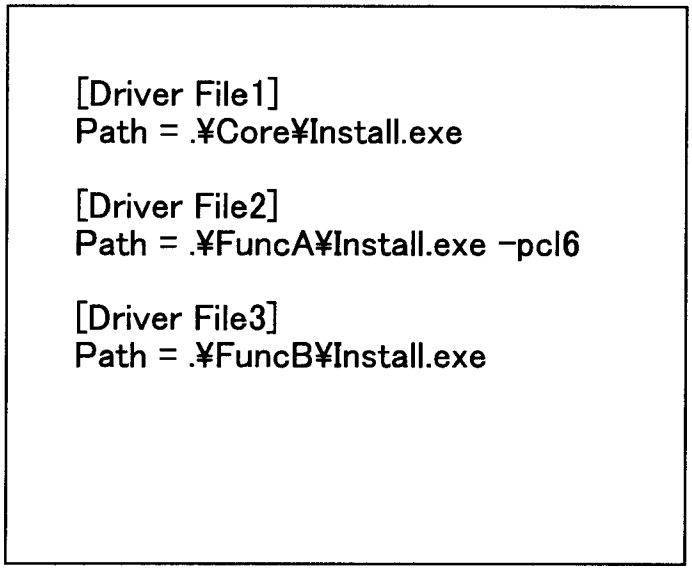
[Fig. 9]


PRINTER DRIVER DOWNLOAD PAGE
(WEB)
PCL6 DRIVER (CORE)
PCL6 DRIVER (FUNCTION A)
PCL6 DRIVER (FUNCTION B)
PCL6 DRIVER (FUNCTION C)
ACQUIRE
1
PACKAGING PROCESS
INSTALL SCENARIO
PCL6 DRIVER DRIVER CORE ENTITY
PCL6 DRIVER FUNCTION A ENTITY
PCL6 DRIVER FUNCTION B ENTITY
PCL6 DRIVER FUNCTION C ENTITY

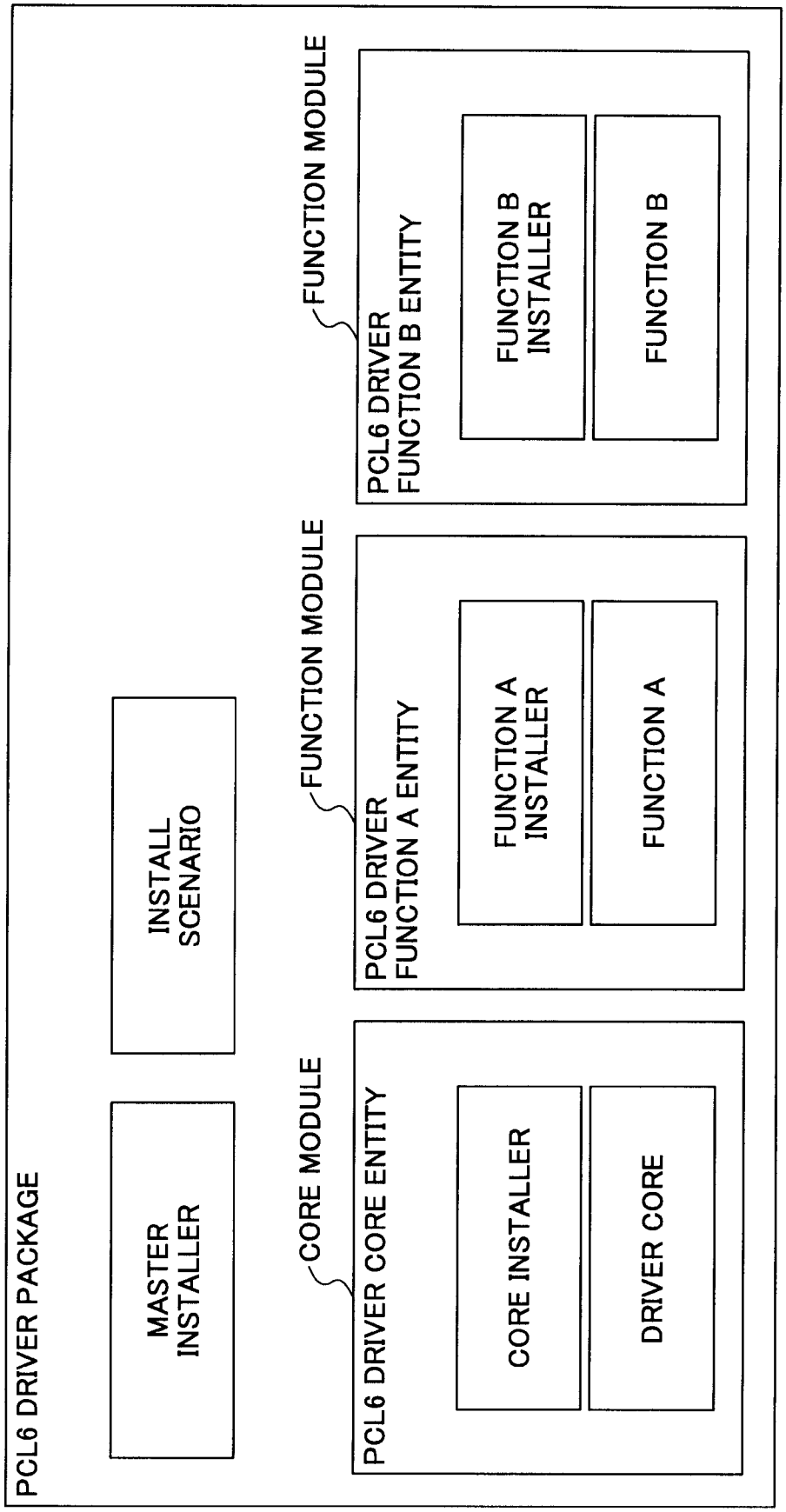
[Fig. 10]


START
S1001
ACQUIRE SPECIFIED FUNCTION MODULE INFORMATION
S1002
REFER TO RULE FILE
S1003
IS RULE SATISFIED?
NO
S1008
REPORT TO USER SPECIFY AGAIN
YES
S1004
DETERMINE AND ACQUIRE FUNCTION MODULE
S1005
CREATE INSTALL SCENARIO (REFER TO RULE FILE)
S1006
PACKAGE INCLUDING MASTER INSTALLER, INSTALL SCENARIO, AND FUNCTION MODULES
S1007
PROVIDE PACKAGE
END

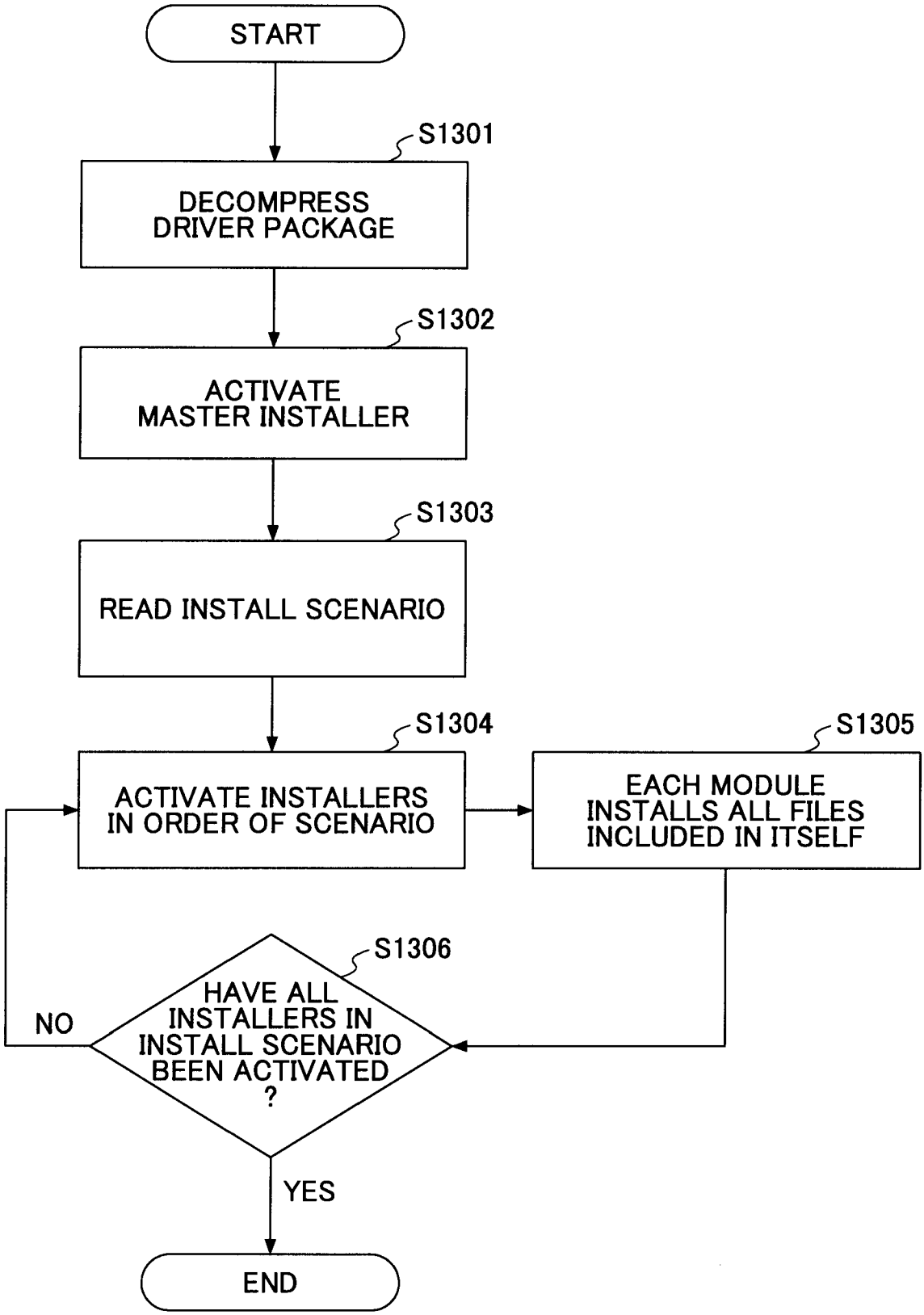
[Fig. 11]


[Driver File1]
Path = .¥Core¥Install.exe
[Driver File2]
Path = .¥FuncA¥Install.exe -pcl6
[Driver File3]
Path = .¥FuncB¥Install.exe

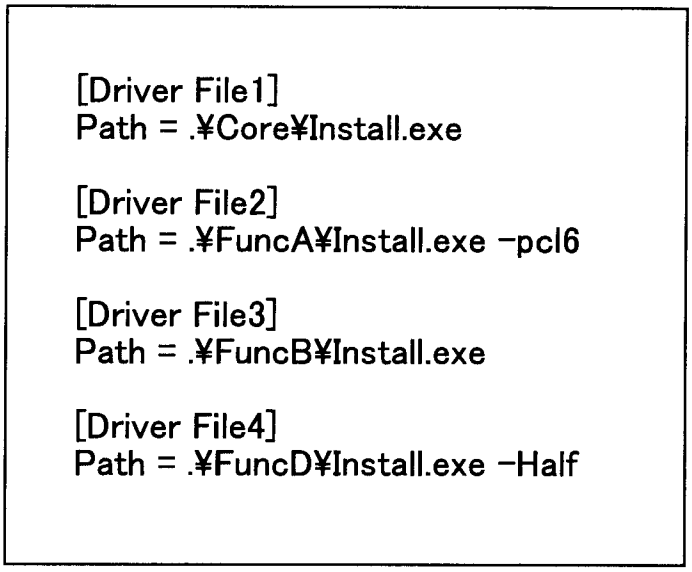
[Fig. 12]


PCL6 DRIVER PACKAGE
MASTER INSTALLER
INSTALL SCENARIO
CORE MODULE
PCL6 DRIVER CORE ENTITY
CORE INSTALLER
DRIVER CORE
FUNCTION MODULE
PCL6 DRIVER FUNCTION A ENTITY
FUNCTION A INSTALLER
FUNCTION A
FUNCTION MODULE
PCL6 DRIVER FUNCTION B ENTITY
FUNCTION B INSTALLER
FUNCTION B

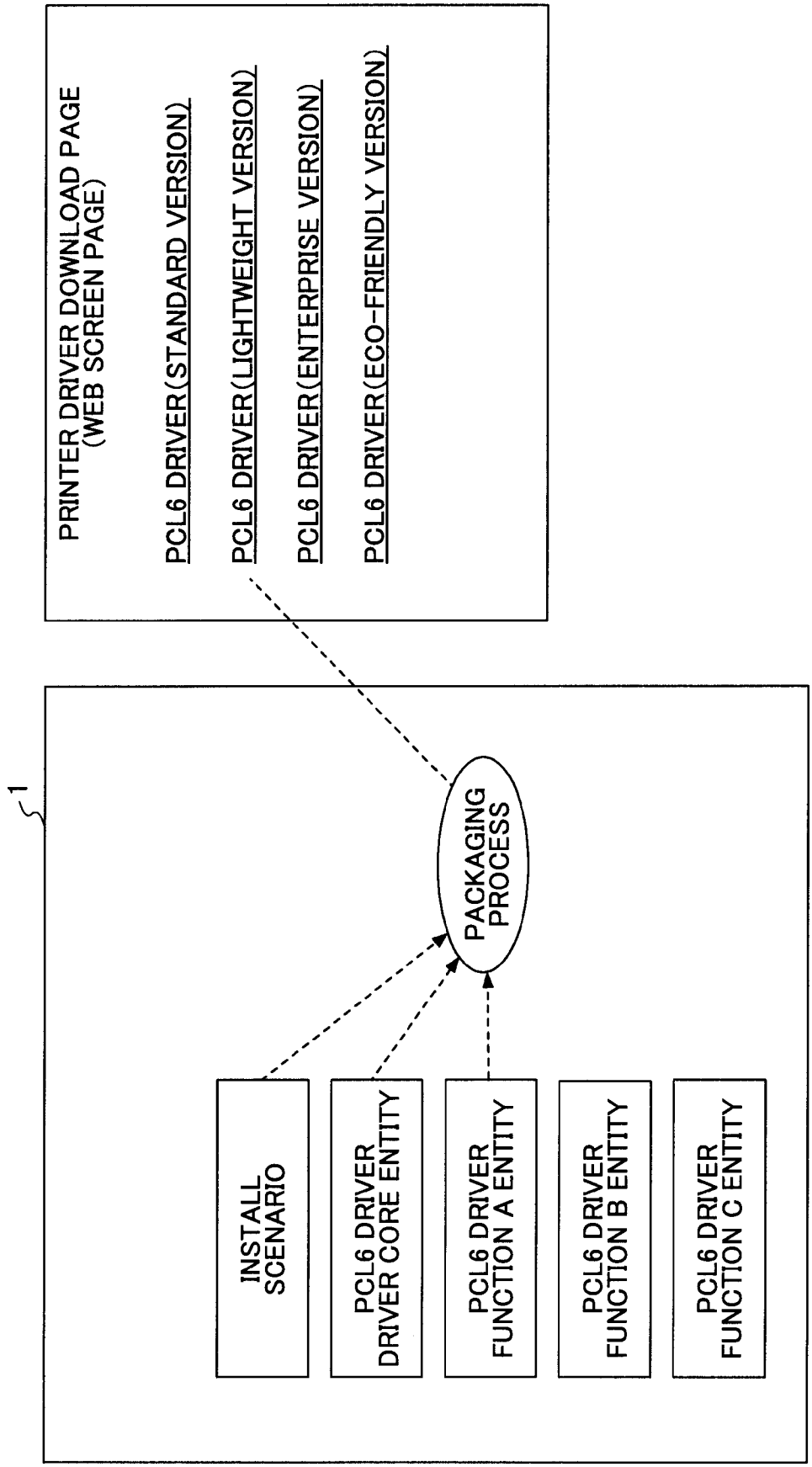
[Fig. 13]


START
S1301
DECOMPRESS DRIVER PACKAGE
S1302
ACTIVATE MASTER INSTALLER
S1303
READ INSTALL SCENARIO
S1304
ACTIVATE INSTALLERS IN ORDER OF SCENARIO
S1305
EACH MODULE INSTALLS ALL FILES INCLUDED IN ITSELF
S1306
HAVE ALL INSTALLERS IN INSTALL SCENARIO BEEN ACTIVATED ?
NO
YES
END

[Fig. 14]


[Driver File1]
Path = .¥Core¥Install.exe
[Driver File2]
Path = .¥FuncA¥Install.exe -pcl6
[Driver File3]
Path = .¥FuncB¥Install.exe
[Driver File4]
Path = .¥FuncD¥Install.exe -Half

[Fig. 15]


PRINTER DRIVER DOWNLOAD PAGE
(WEB SCREEN PAGE)
PCL6 DRIVER(STANDARD VERSION)
PCL6 DRIVER(LIGHTWEIGHT VERSION)
PCL6 DRIVER(ENTERPRISE VERSION)
PCL6 DRIVER(ECO-FRIENDLY VERSION)
1
PACKAGING PROCESS
INSTALL SCENARIO
PCL6 DRIVER DRIVER CORE ENTITY
PCL6 DRIVER FUNCTION A ENTITY
PCL6 DRIVER FUNCTION B ENTITY
PCL6 DRIVER FUNCTION C ENTITY

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2009/006281

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F3/12(2006.01)i, G06F9/445(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. G06F3/12, G06F9/445

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

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2010
Registered utility model specifications of Japan 1996-2010
Published registered utility model applications of Japan 1994-2010

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2004-265061 A (CANON KABUSHIKI KAISHA) 2004.09.24, column【0061】, 【0069】-【0093】, Fig. 8, 10, 13, 14 & US 2004/0223182 A1	1-8
Y	JP 11-282653 A (Matsushita Electric Works Co., Ltd.) 1999.10.15, abstract, column【0024】 (No Family)	1-8
Y	JP 2007-115257 A (International Business Machines Corporation) 2007.05.10, Claim 1-3 & US 2007/0094400 A1	1-8
Y	JP 2003-288210 A (NEC System Technologies, Ltd.) 2003.10.10, abstract (No Family)	3,7

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26.01.2010	09.02.2010

Name and mailing address of the ISA/JP	Authorized officer
Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	UCHIDA Masakazu 5E 9065 Telephone No. +81-3-3581-1101 Ext. 3521