

**SYSTEM, METHOD, AND PROGRAM FOR
CHECKING DEPENDENCIES OF
INSTALLED SOFTWARE COMPONENTS
DURING INSTALLATION OR
UNINSTALLATION OF SOFTWARE**

**CROSS-REFERENCE TO RELATED
APPLICATIONS**

This application is related to the following co-pending and commonly-assigned patent applications, which applications were filed on the same date herewith, and which applications are incorporated herein by reference in their entirety:

- “A Cross-Platform Program, System, and Method Having a Global Registry Object for Mapping Registry Equivalent Functions in an AIX Operating System Environment,” having U.S. patent application Ser. No. 09/280,345;
- “A Cross-Platform Program, System, and Method Having a Global Registry Object for Mapping Registry Equivalent Functions in an OS/2 Operating System Environment,” having U.S. patent application Ser. No. 09/280,350; “A Cross-Platform Program, System, and Method Having a Global Registry Object for Mapping Registry Functions in a Windows Operating System Environment,” having U.S. patent application Ser. No. 09/280,349;
- “Global Registry Object for Mapping Registry Functions and Registry Equivalent Functions Across Multiple Operating Systems in a Cross-platform Program,” having U.S. patent application Ser. No. 09/280,371;
- “A Cross-platform Program, System, and Method Having a System Independent Registry for Use on Operating Systems Irrespective of a Registry Equivalent,” having U.S. patent application Ser. No. 09/280,368;
- “A System, Method, and Program for Enabling a Use of Property Object Having a Variable for a Property Value in a Cross-Platform Program,” having U.S. patent application Ser. No. 09/280,344;
- “A System, Method, and Program for Overriding Program Properties,” having U.S. patent application Ser. No. 09/280,346;
- “A System, Method and Program for Providing an Object-Oriented Install Architecture,” having U.S. patent application Ser. No. 09/280,352;
- “A System, Method, and Program for Automatic Error Detection While Utilizing a Software State Machine for Carrying out the Process Flow of a Software Program,” having U.S. patent application Ser. No. 09/280,375;
- “A System, Method, and Program for Utilizing a Software State Machine for Carrying Out the Process Flow of a Software Program,” having U.S. patent application Ser. No. 09/280,376;
- “A System, Method, and Program for Enabling a Software Program to Automatically Select a System-dependent Function,” having U.S. patent application Ser. No. 09/280,369;
- “A System, Method, and Program for Mapping a Global Object to Desktop Elements of Different Operating Systems,” having U.S. patent application Ser. No. 09/280,374;
- “A System, Method, and Program for Modifying a Text File,” having U.S. patent application Ser. No. 09/280,348;

“A System, Method, and Program for Updating Registry Objects With a Cross-platform Installation Program,” having U.S. patent application Ser. No. 09/280,351;

“A System, Method, and Program For Preserving Background Settings During Install and Uninstall Operations,” having Ser. No. 09/280,374;

“A System, Method, And Program for Modifying a Library Object,” having Ser. No. 09/280,347;

“A System, Method, And Program For Installation on Drives Using a Drive Object,” having U.S. patent application Ser. No. 09/280,353; and

“A System, Method, and Program for Performing Program Specific Uninstall Operations,” having U.S. patent application Ser. No. 09/280,373.

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a system, method, program, and data structure for installing and uninstalling programs and in particular for checking dependencies of installed components.

2. Description of the Related Art

An installer program is a software program that enables a programmer to write specific code to install a given application program onto the drives of a computer in a way that enables the given application program to work correctly with the computer's environment, including its operating system. There are several types of installers—Java installers and operating system specific installers, e.g., Microsoft Windows installers, and International Business Machines (“IBM”) OS/2 and AIX operating system installers, etc. Many of the install programs existing today have various limitations in their functionality as discussed below.

One type of Java installer is provided by a software company known as InstallShield. Currently, the functionality and configurability of this Java installer is somewhat limited. For example, the Java installer provides default panels, but the text cannot be changed or configured. Also, this Java installer is not easy to customize. In other words, a programmer cannot easily add a function that is not provided by the installer program. In addition, the current Java installer provides only limited registry support. A registry is a central repository for all possible information for the computer such as hardware configurations and software settings, etc. The registry maintains information on installed components. A programmer or user can access the registry to determine information about installed products, based upon what the user/programmer placed into the registry or what information is useful to the user. Presently, the Java installer only works with a Microsoft Windows' registry; and the support provided with Microsoft Windows is limited. For example, the current Java installer does not enable the Windows' system registry to be updated directly. Instead, the keys to update in the registry are in a file which must be imported into the registry through a system call. It would be desirable if such a Java installer program supported multiple directories, splash screens (which are images that come up while a software product is being