

that each item in the list of available update components is aligned with either the identical item in the second list of installed update components, if the component is installed, or a special placeholder, if the component is not installed.

[0016] Further, the method includes having the user select one or more update components from the list of available update components, where the user may select from update components not installed as well as update components already installed. The method includes receiving one or more selected update components. The method also includes installing one or more selected update components.

[0017] The invention can be implemented to realize one or more of the following advantages. A list of available updates will be displayed in conjunction with a list of installed updates, allowing a computer user to readily see which update components have already been installed, and which update components have yet to be installed. A computer user will be able to select any available component to be installed, whether or not it has been installed previously. A list of already installed updates will be generated automatically, so that no listing of installed updates need be maintained on the user's computer.

[0018] Further, low network bandwidth is required, as little administrative data must be transferred to and from the user's computer. Privacy and security concerns are also addressed, as no information about the user's computer need be sent to anyone else, except for the identity of the software applications that the user is attempting to update for the purpose of retrieving the list of updates associated with that software application.

[0019] The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] FIG. 1 is a flow chart of a method for identifying available and installed components on a computer.

[0021] FIG. 2 is a flow chart of a method for displaying available and installed components.

[0022] FIG. 3 shows a user interface displaying available and installed components.

[0023] FIG. 4 shows a user interface displaying components ready to be installed.

[0024] Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0025] FIG. 1 shows a method 100 for identifying available and installed components on a computer. A system performing method 100 identifies installed products on a user computer (step 105). The installed products can be software applications, documentation, or other resources that require periodic updates. In one implementation, one or more installed products on the user computer are identified by means of a single file maintained on the user computer. The single file can be an Extensible Markup Language (XML) file. The single file is created or modified by the system when each installed product is originally installed, and provides the system with information necessary to identify each specific product; for instance, this information

can include the product name, the product version, the product language, and the operating system language. The single file can also include information indicating where each specific product is installed on the user computer. In one implementation, this single file contains information on all installed products from one software provider; for example, the single file contains information on all Adobe® products installed on the user computer. In an alternative implementation, the system stores information on installed products from multiple providers. Each installed product passes an application identifier to the system, to identify the specific installed product on the user computer. The application identifier can be in a file that is affiliated with each installed product that uniquely identifies each installed product to the system.

[0026] The system obtains one or more manifest files corresponding to the installed products (step 10). The manifest file contains information about each update that is available for the corresponding installed product on the user computer. In one implementation, each product configuration has a unique manifest file that corresponds with that product. For example, a manifest file would exist that corresponds to Adobe® Acrobat® 6.0 U.S. English. A different manifest file would exist for Acrobat® 6.0 German, and a third manifest file would exist for Acrobat® 5.5 U.S. English. In one implementation, the system obtains the manifest file that is required from among all of the manifest files available. The manifest files can be obtained from a variety of sources. In one implementation, the system obtains the manifest files from a server over a data communication network, e.g., the Internet. In another implementation, the manifest files are obtained from removable media, e.g., a floppy disk, CD-ROM disk, or DVD disk.

[0027] The application identifier is used to identify a specific manifest file in a specified location, e.g., on a predetermined server. The system uses the application identifier to identify the installed products for which manifest files need to be obtained. Manifest files can then be obtained for the identified installed products. The manifest file contains the information that is used to determine the available updates for the installed product on the user computer. Initially, the manifest file contains no update information. As updates are generated, a new manifest file is created that contains information on the new updates, as well as any previous updates.

[0028] In one implementation, the manifest file is an XML file. The information to be included in the manifest file is included in the XML file in a standardized form recognizable as a manifest file. The XML manifest file can be made available on specified servers for download by a user computer.

[0029] The manifest files contain information detailing the updates that are available for each installed product. The manifest files do not contain the actual update components; rather, the manifest files contain information that corresponds to each available update component. For example, the manifest file can contain the following information for each update available: the name of the update; a short description of the update; a relative path that points to a local file on the user computer showing the current version of the component being updated; and instructions for obtaining the update.

[0030] The manifest file can also contain information identifying which method to use to extract version information from the local file, and how to compare the extracted