While existing application programs may currently support an uninstall function, current operating systems lack a mechanism to ensure that the components associated with a particular application program are removed during the uninstall process. An embodiment of the invention maintains a data store (e.g., a log) that tracks which files are associated with particular application programs. Thus, when an application is uninstalled, the operating system identifies and deletes any files left behind by the uninstall process (e.g., including those that have been virtualized in the application's namespace). This provides a more complete uninstall of the application program by uninstalling all elements of the application including those elements created by extending or modifying the behavior of the operating system or in other (e.g., lower) levels of virtualization. This method also helps to remove spy-ware, ad-ware, or other unwanted application programs that often accompany an installed application after the application program is uninstalled from the computing system.

[**0078**] In one embodiment, the particular application program to be removed is one of a plurality of application programs installed on the computing system. An embodiment of the invention receives a request to uninstall the particular application program at **602**. The request may originate from, for example, a user of the computing system. Alternatively, the request may be generated by an upgrade utility that uninstalls a previous version of an application program before installing a current version of the operating system. An embodiment of the invention determines an identifier associated with the particular application program at **604**. For example, the identifier may be part of the application program or stored separately in a memory area. An embodiment of the invention identifies, via the determined identifier, one or more files associated only with the particular application program at **606**. That is, the identified files are not associated with any of the other application programs installed on the computing system. As each file on the system has at least one application identifier associated therewith, the identification of the files associated only with the particular application program results from performing a search for the determined identifier. An embodiment of the invention deletes the identified file at **608**. In one embodiment, the invention avoids deleting any user files (e.g., word processing documents, spreadsheet documents).

[**0079**] Additionally, system settings or resource changes applied in response to installing the particular application program are identified at **610** and reverted at **612**. For example, during installation of an application program, any system settings applied to the computing system are logged and maintained by an embodiment of the invention. The changes made by an application to files and system settings are tagged for ownership tracking. The log associates each of the changes with the application identifier of the application program being installed. In one embodiment, the log is maintained to allow the rollback of one or more of the changes. For example, a user may wish to undo the most recent change made to the system. In another example, the operating system performs a complete uninstall of a particular application program by rolling back the changes associated with the particular application program. During removal or uninstallation of the application program, an embodiment of the invention uses the determined identifier to identify and revert or otherwise remove the applied settings or changes associated with the application identifier of the application program being uninstalled. For example, a change to a file type association may be logged so that uninstalling an application program does not leave a particular file type without an associated application program. That is, if a file type association is made during installation of the application program, the file type association is reverted when the application program is uninstalled.

[**0080**] Exemplary Operating Environment

[**0081**] **FIG. 7** shows one example of a general purpose computing device in the form of a computer **130**. In one embodiment of the invention, a computer or other computing system such as the computer **130** is suitable for use in the other figures illustrated and described herein. Computer **130** has one or more processors or processing units **132** and a system memory **134**. In the illustrated embodiment, a system bus **136** couples various system components including the system memory **134** to the processors **132**. The bus **136** represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[**0082**] The computer **130** typically has at least some form of computer readable media. Computer readable media, which include both volatile and nonvolatile media, removable and non-removable media, may be any available medium that may be accessed by computer **130**. By way of example and not limitation, computer readable media comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. For example, computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that may be used to store the desired information and that may be accessed by computer **130**. Communication media typically embody computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and include any information delivery media. Those skilled in the art are familiar with the modulated data signal, which has one or more of its characteristics set or changed in such a manner as to encode information in the signal. Wired media, such as a wired network or direct-wired connection, and wireless media, such as acoustic, RF, infrared, and other wireless media, are examples of communication media. Combinations of the any of the above are also included within the scope of computer readable media.

[**0083**] The system memory **134** includes computer storage media in the form of removable and/or non-removable, volatile and/or nonvolatile memory. In the illustrated embodiment, system memory **134** includes read only