

target computer-readable media **108** can be accessed by the computer **102** as a network share or as a local medium.

[0031] Those skilled in the art will note that the illustration of FIG. 1 including the network **106** is merely exemplary. In another embodiment, the target computer-readable media **108** is locally accessible by the computer **102**. That is, network **106** and server **104** are not included in that the target software **110** is not physically on another computer attached via a network or other connection. The target software **110** can be another volume, a logical set of folders, or an image on the computer **102**. In this embodiment, an operating system associated with the target software **110** is not currently active or running in the context of the operating system associated with the computer **102**. For example, the computer **102** may be the same computer used to install the operating system of the target software **110** on the target computer-readable media **108**. In this example, the computer **102** halts execution of the installed operating system of the target software **110** and boots another operating system (e.g., via a removable computer-readable medium). The computer **102** boots the other operating system or some environment, separate from the target software **110**, which executes with the invention software to manipulate the offline target software **110**. For example, the computer **102** may boot off a floppy or a CD-ROM. In the context of this booted operating system, a user could modify the target software **110** stored on local computer-readable medium such as the C:\drive. Such modification occurs offline in that the computer **102** is booted, but not into an operating system associated with the target software **110**. For example, the invention software allows complex modifications to occur in this offline fashion by re-directing all tasks associated with the installation of an application program to the offline target software **110**. After re-directing performance of the tasks and after a reboot into the operating system associated with the target software **110**, the application program has been installed.

[0032] Generally, the invention provides software (such as driver software **304** in FIG. 3) to access the data on the target computer-readable medium **108**. In one scenario of the invention as described above, the user locally boots from a bootable removable media to modify local, offline software. In another scenario of the invention, the user boots a local system to modify remote, offline software. In one embodiment, the invention software mounts the target computer-readable medium **108** using mounting methods known in the art to provide basic file input/output access. For example, an image foo.img can be mounted and appear to the online system as X:\ (i.e., X:\system, X:\config, etc.). The invention also provides software routines to modify system settings associated with the target computer **110**. In one embodiment, the invention provides an application programming interface to allow a user to modify such system settings (e.g., registry values). For example, an API of the invention allows a user to load a registry hive associated with the target software **110** in the computer **102**. Using the API, the user can modify the loaded registry hive to effect changes in the target software **110**. For example, the following registry key illustrates a registry hive loaded from offline software such as target software **110** into the computer **102**.

[0033] HKLM\offline_system\setup\Setupinprogress=1

[0034] After loading the registry hive, the loaded registry hive points to the mounted, offline software image. As in the

previous example, the offline registry hive points to the image foo.img at X:\ (i.e., X:\system, X:\config, etc.). All registry key updates are directed to the loaded, offline registry hive. In an alternative embodiment, all the registry key updates that occur during modification of the offline file system are queued and then applied after the modifications to the offline file system are complete.

[0035] Referring next to FIG. 2, an exemplary block diagram illustrates the target computer-readable medium **108** that stores target software **110**. The target software **110** includes an operating system **206** and/or an application program **208**. The operation system **206** includes one or more operating system files **210** such as file #1 through file #N. The application program **208** includes one or more application program files **212** such as file #1 through file #M.

[0036] Referring next to FIG. 3, an exemplary block diagram illustrates the driver software components of the computer **102** that communicate with the target software **110** stored on the target computer-readable media **108**. In this example, the computer **102** includes an online operating system **302** and driver software **304**. While the computer **102** is shown connected to the target computer-readable media **108** via network **106**, such a connection is optional as described above.

[0037] The driver software **304** includes components that can be executed when the computer **102** receives a command from the user. In one embodiment, driver software **304** includes a registry application programming interface (API) component **306**, a setup API component **308**, a service install API component **310**, a file input and output (I/O) component **312**, and a mass storage device component **314**. The registry API component **306** updates one or more system settings on the for the target software **110** by modifying one or more files (e.g., a registry) on the target computer-readable media **108**. The setup API component **308** configures the next boot state of the target software **110** (e.g., out of box, first boot, or audit boot) and configures setup configuration options such as the install directory. The service install API component **310** adds or removes a service to or from the target software **110** by installing one or more files associated with the service to the target computer-readable media **108**. The file I/O component **312** performs file input and file output operations on the target computer-readable media **108**. The mass storage device component **314** adds at least one device driver for a mass storage device controller to the target software **110** by installing one or more files associated with the device driver to the target computer-readable media **108**. In addition, the components of the driver software **304** may configure system settings (e.g., registry keys) for the target software **110** via the registry API component **306**. The components of the driver software **304** operate on the offline target software **110**.

[0038] Those skilled in the art will appreciate that the driver software **304** may include other components not specifically described herein that allow the computer **102** to manipulate the target software **110** while the target software **110** is offline. It is contemplated by the inventors that such alternate embodiments of the driver software **304** are within the scope of the invention. For example, the mass storage device component **314** is one example of a device component installing a device driver, but other device components