

page. The browser invokes the package manager to execute the software package if it is already installed or to install it if not. If not already installed, the package manager instructs the browser to download the distribution file unit and proceeds with the installation as described in the previous section. The "CODEBASE" element in <OBJECT> and the "useslibrarycodebase" tag in <APPLET> can point to the manifest file or to the distribution unit file.

In a third exemplary embodiment of the invention for purposes of this section, a distribution unit file is used to automatically distribute software from Fred's Software Company's server to the user's computer. This automatic distribution across a network employs "channels" to which the user subscribes to automatically "push" software components through a client agent such as a browser. The channel is described using a Channel Definition Format (CDF) which is also based on XML. A CDF file uses the OSD elements to inform a CDF-aware client agent as to what software components should be downloaded and installed.

```

<CHANNEL HREF="http://www.fsc.com.intropage.htm">
<SELF="http://www.fsc.com/software.cdf"/>
<TITLE>A Software Distribution Channel</TITLE>
<SOFTPKG
  HREF="http://www.fsc.com/aboutsoftware.htm"
  AUTOINSTALL="yes"
  NAME="{D27CDB6E-AE6D-11CF-96B8-44455340000}"
  VERSION="1.0,0,0">
<IMPLEMENTATION>
<OS VALUE="WinNT"><OSVERSION
VALUE="4,0,0,0"/></OS>
<OS VALUE="Win95"/>
<PROCESSOR VALUE="x86"/>
<CODEBASE HREF="http://www.fsc.com/
coolestapp.cab"/>
</IMPLEMENTATION>
</SOFTPKG>
</CHANNEL>

```

This section has described a particular implementation of the package manager which is directed to install software by OSD elements embedded in an XML document. The processing of a manifest file described in previous section when written as XML document is described. In addition, alternate embodiments in which a separate XML document resides on a Web page to direct a browser to invoke the package manager to install a software package is also described in this section.

CONCLUSION

A software package manager has been described which manages the installation, execution and uninstallation of software packages acquired through various media. The software manager uses a manifest file, a distribution unit, and a code store data structure to accomplish its functions. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention.

For example, those of ordinary skill within the art will appreciate that the file and data structures described herein can be easily adapted to future distribution media. Furthermore, those of ordinary skill within the art will appreciate that future extensible languages which are platform and

operating system independent can be used to direct the software package managers actions.

The terminology used in this application is meant to include all hardware and software platforms. Therefore, it is manifestly intended that this invention be limited only by the following claims and equivalents thereof.

We claim:

1. In a computer, a method of processing one or more software dependencies, the method comprising:
  - for one or more of the software dependencies, determining whether software associated with the software dependency is present on the computer;
  - responsive to determining the software associated with the software dependency is not present on the computer, acquiring, by a software package manager running on the computer, the software associated with the software dependency;
  - after acquiring the software associated with the software dependency, updating a database at the computer indicating the software associated with the software dependency is installed on the computer;
 wherein at least one of the software dependencies specifies a plurality of software items forming a software package, wherein the software package comprises a mixture of native code components and platform-independent code components;
  - wherein at least one of the software dependencies refers to a list comprising one or more other software dependencies; and
  - wherein the acquiring the software associated with the software dependency comprises recursively processing the one or more other software dependencies.
2. The method of claim 1 wherein acquiring the software associated with the software dependency comprises acquiring a file comprising the list comprising one or more other software dependencies.
3. The method of claim 1 wherein acquiring the software associated with the software dependency comprises acquiring a list of one or more files from a remote location and acquiring the files in the list.
4. The method of claim 1 wherein one or more of the software dependencies is associated with a location whereat the list of other software dependencies can be found.
5. The method of claim 1 wherein the database is operable to indicate whether a plurality of software components are installed via a single name associated with the plurality of software components.
6. The method of claim 1 wherein the database is operable to indicate whether a plurality of software dependencies are installed via a single name associated with the plurality of software dependencies.
7. The method of claim 1 wherein:
  - one or more dependencies in the list of software dependencies is associated with a version number; and
  - determining the dependency is not present on the computer comprises determining software satisfying the version number is not present on the computer.
8. The method of claim 7 wherein at least two software dependencies are associated with different version numbers.
9. The method of claim 1 wherein acquiring dependencies is deferred until execution of software associated with the dependencies is requested.
10. The method of claim 1 further comprising:
  - after acquiring the software associated with the software dependency, installing the software.