

tputil.dll, independent of which languages are installed on the phone (via different LANGPHONE\_XXXX packages).

[0102] Additionally, the flexibility of the system is such that language-specific executable code is allowed where necessary. For example, different languages and locales have different Input Method Editors (IMEs) which are used for capturing text in that language. These IMEs require special code for each language, and thus are placed inside one of the LANG regions.

[0103] Lastly, note that cabinet verification may be performed. The cabinet verification module extracts the device manifest file 260 from the final package file 232 (a cabinet file) and verifying the cabinet file against the contents of the device manifest 260.

[0104] As can be seen from the foregoing detailed description, there is provided various mechanisms that facilitate the updating of some subset of an operating system image. A self-describing package file is provided, including dependency, shadow and other features that make updating of an image straightforward and correct.

[0105] While the invention is susceptible to various modifications and alternative constructions, certain illustrated embodiments thereof are shown in the drawings and have been described above in detail. It should be understood, however, that there is no intention to limit the invention to the specific forms disclosed, but on the contrary, the intention is to cover all modifications, alternative constructions, and equivalents falling within the spirit and scope of the invention.

What is claimed is:

1. In a computing environment, a method comprising:
  - building a package, the package corresponding to part of an operating system image and containing files and settings; and
  - associating information with the package that describes the contents of the package such that an installation mechanism can determine how to install the package to a device.
2. The method of claim 1 wherein the information associated with the package comprises a device manifest file, and further comprising, adding the device manifest file to the package.
3. The method of claim 2 further comprising writing data into the device manifest file comprising dependency information that describes a dependency of at least some contents of the package on another entity.
4. The method of claim 3 wherein the other entity comprises another package, and further comprising extracting the device manifest file from the package and extracting another device manifest file from the other package.
5. The method of claim 2 further comprising writing data into the device manifest file comprising shadow information that describes precedence of settings data relative to other settings data.
6. The method of claim 1 wherein building a package comprises, determining which files are executable files and performing a relocation process on the executable files.
7. The method of claim 1 wherein associating information with the package that describes the contents of the package

comprises associating dependency information that describes a dependency of at least some contents of the package on another entity.

8. The method of claim 1 wherein associating information with the package that describes the contents of the package comprises associating shadow information that describes precedence of settings data relative to other settings data.

9. The method of claim 1 further comprising adding settings information to the package.

10. The method of claim 1 wherein building a package comprises mapping a package to a definition for that package that defines existing files, reading a build manifest file that specifies the file contents for that package, and generating the package based on the package definition and the build manifest file.

11. The method of claim 10 further comprising creating the build manifest file from a binary image builder file for the operating system image and a component to package mapping file.

12. The method of claim 11 further comprising processing executable code identified in the build manifest file, to enable executable code relocation at install time to a device.

13. The method of claim 1 wherein building the package comprises reading a binary image builder file that contains a list of files to be included in the package when built.

14. The method of claim 1 wherein building the package comprises creating a build manifest file corresponding to the binary image builder file and reading the build manifest file.

15. The method of claim 1 wherein building the package comprises reading a settings file that contains a list of setting information to be included in the package when built.

16. The method of claim 1 wherein building the package comprises reading a component mapping file that maps module-related information and files to the package.

17. The method of claim 1 wherein building the package comprises reading a component relationships file that defines shadow relationships.

18. The method of claim 17 wherein the information associated with the package comprises a device manifest file, and further comprising, writing information to the device manifest file based on the shadow relationships, and adding the device manifest file to the package.

19. The method of claim 1 wherein building the package comprises reading a component relationships file that defines dependency relationships.

20. The method of claim 17 wherein the information associated with the package comprises a device manifest file, and further comprising, writing information to the device manifest file based on the dependency relationships, and adding the device manifest file to the package.

21. The method of claim 1 wherein building the package comprises reading a package definition file that defines what packages exist.

22. The method of claim 2 wherein building a package comprises having at least some executable code files that are separate from language-dependent data files such that at least some of the executable code is independent of any language.

23. At least one computer-readable media having computer-executable instruction which when executed perform the method of claim 1.