

version compatibility with the same files without limiting the features that can be added to future versions.

[0011] In one exemplary aspect, the present invention can comprise including file version watermarks in an original data file of an application program. The file version watermarks can be an element in the original data file that can indicate various properties of the original data file. For example, a high version watermark can be provided in the original data file to indicate the highest version of the application program used to save the file. A last version watermark can be provided in the original data file to indicate the last version of the application program used to save the file. A low version watermark can be provided in the original data file to indicate the lowest version of the application program used to save the file. A creation version watermark can be provided in the original data file to indicate the version of the application program that first created the file. An object version watermark can be provided in the original data file to indicate the highest version of the application program to write a particular object. Each file version watermark can indicate changes, additions, or deletions made to the file.

[0012] In another exemplary aspect, the present invention can comprise determining how to load or save the original data file based on the information provided in one or more file version watermarks. In one exemplary aspect, the present invention can determine whether the high version watermark in the file represents a newer version than the active version of the application program that is loading or saving the file. The high version watermark represents a newer version if the original data file has been saved by a newer version of the application program with respect to the active version. In that case, the present invention can determine whether the original data file contains information that is unknown to the active version. The unknown information can then be ignored by the active version of the application program and retained for use by newer (later) versions.

[0013] In another exemplary aspect, the present invention can determine whether the high version watermark in the file represents an older version than the active version of the application program that is loading or saving the file. The high version watermark represents an older version than the active version if the original data file has been saved by an older version of the application program with respect to the active version. In that case, the present invention can determine whether the active version contains new information that is based on old information in the original data file. The new information can then be updated based on the old information. Additionally, the new information can be saved for use by older (previous), active, and newer (later) versions by storing data in the format of the older version, as well as in the format of the active and newer versions.

[0014] According to another exemplary aspect of the present invention, the in-memory structure of the application program can be separated from the data file format. Accordingly, future versions of the application program can include many new features without changing the data file format. For example, structures can be moved around in memory to be more efficient for a certain processor type without being incompatible with a different version because of differing data file formats.

[0015] The present invention also can minimize the amount of memory and processing needed for loading and

saving data files corresponding to different versions of an application program. The file version watermarks can identify particular information in a data file that corresponds to a different version. Accordingly, any processing or conversion can be performed on only the identified information.

[0016] Another exemplary aspect of the present invention incorporates an object property list ("OPL") or an object property list array ("OPL array") as the data file structure. An OPL or OPL array can allow "round-tripping" of unknown property data from future versions by propagating or returning the unknown data back to the saved data file. The OPL or OPL array can be loaded into memory. All of its properties that are known by a particular version of the application program can be overwritten as needed. The remaining properties (i.e., properties that are unknown to the particular version) were created by a newer version and can remain in the file untouched. For each memory structure saved in a data file, there can be an associated OPL or OPL array for saving that structure.

[0017] These and other aspects, objects, and features of the present invention will become apparent from the following detailed description of the preferred embodiments, read in conjunction with, and reference to, the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] FIG. 1 is a block diagram illustrating an object used by an application program;

[0019] FIG. 2 is a block diagram illustrating an exemplary computer suitable for practicing an exemplary embodiment of the present invention;

[0020] FIG. 3 is a block diagram illustrating characteristics of a conventional OPL;

[0021] FIG. 4 is a block diagram illustrating characteristics of an OPL array according to an exemplary embodiment of the present invention;

[0022] FIG. 5 is a flow chart depicting a method for loading an original data file according to an exemplary embodiment of the present invention;

[0023] FIG. 6 is a flow chart depicting a method for saving an original data file according to an exemplary embodiment of the present invention;

[0024] FIG. 7 is a flow chart depicting a method for loading and saving an original data file according to the present invention, where the active version has an additional object property that is not included in a previous version;

[0025] FIG. 8 is a flow chart depicting a method for loading and saving an original data file according to the present invention, where the active version has deleted an object property that was used in a previous version; and

[0026] FIG. 9 is a flow chart depicting a method for loading and saving an original data file according to the present invention, where the active version has modified an old object property of a previous version.

DETAILED DESCRIPTION OF THE INVENTION

[0027] The present invention will be described in detail with reference to the accompanying drawings, in which like reference numerals represent like elements.