

to the appropriate OPL dictionary. For example, the appropriate dictionary can be a page object dictionary for a page object, another existing dictionary, or a new dictionary. In step **910**, the default value for the new CMYK color property can be defined in the OPL dictionary. The old RGB property remains unchanged. In step **920**, the original data file can be loaded into the active version, and the new CMYK color property can be updated based on the old RGB property of the object. After the original data file has been modified, it can be saved in step **930**. In step **930**, the new CMYK color property can be saved without reference to the old RGB property from the original data file. The new CMYK property can be converted to a corresponding old RGB property and saved in the file. Accordingly, the new CMYK color property can be saved for use with the active and future versions of the application program. Previous versions will simply ignore the new CMYK color property. In step **940**, the high version watermark can be updated to the active version.

The method of the present invention can also resolve inconsistencies between two versions of the application program that are saved in the data file. In the CMYK-RGB example discussed above, previous versions can read the file because the later version will convert the new CMYK color property down to a corresponding RGB property. However, inconsistencies can develop when the last version to write the file is older than both the active version and the high version to write the file. The active version can be the same version or a later version than the high version to write the file. If the high version, which is not the last version, to write the file recognized the new CMYK color property, then it will write out the new CMYK color property value into the data file, as well as a corresponding RGB color property value. Additionally, if the last version to write the file recognized only the RGB color property, then it will write out the RGB color property value into the data file. When the data file is loaded into the active version that supports the new CMYK color property, the CMYK color property in the data file corresponds to old information. Two versions of the color property information exist in the data file because the high version wrote the CMYK color property value in the file and the last version wrote RGB color property value in the file. The RGB and CMYK values do not match because the last version to write the file updated the RGB value, but retained the value of the unknown CMYK property. When the file is loaded into the active version, the last version to write the file can be determined (see step **570** of FIG. **5**). Based on that information, it can be determined that the CMYK color property value is outdated and that the RGB color property value should be loaded and converted to an updated CMYK color property value.

An object is a collection of properties. Accordingly, changes to an object in different versions of the application program can be reflected by changes in properties that correspond to the object. When the data file containing the object is loaded or saved, the steps of methods **700**, **800**, and **900** can be reiterated for individual properties of the object.

An embodiment of the present invention can be used as part of a document publishing system known as the Microsoft® Publisher, which is available from Microsoft® Corporation of Redmond, Wash. That publishing system can allow a user to edit documents and insert various objects containing graphics data, text data, spreadsheet data, or other kinds of data. In addition, that system can allow the user to modify the properties of the objects. The system can have a number of predefined object types that have predefined properties which are set to standard values (default values).

Each object type can have a different set of properties and/or property values. Each object can be an instance of their object type. As such, the settings of an object in its original form can be readily obtainable. Although an embodiment is described with reference to a document publishing system, one skilled in the art will recognize that the techniques described herein can be applied to a virtually unlimited number of other types of systems.

The present invention can be used with computer hardware and software that performs the processing functions described above. As will be appreciated by those skilled in the art, the systems, methods, and procedures described herein can be embodied in a programmable computer, computer executable software, or digital circuitry. The software can be stored on computer readable media. For example, computer readable media can include a floppy disk, RAM, ROM, hard disk, removable media, flash memory, memory stick, optical media, magneto-optical media, CD-ROM, etc. Digital circuitry can include integrated circuits, gate arrays, building block logic, field programmable gate arrays (FPGA), etc.

Although specific embodiments of the present invention have been described above in detail, the description is merely for purposes of illustration. Various modifications of, and equivalent steps corresponding to, the disclosed aspects of the exemplary embodiments, in addition to those described above, may be made by those skilled in the art without departing from the spirit and scope of the present invention defined in the following claims, the scope of which is to be accorded the broadest interpretation so as to encompass such modifications and equivalent structures.

What is claimed is:

1. A method for reading an original data file to be compatible with an active version and a later version of an application program, said method comprising the steps of:
  - providing a high version watermark in the original data file, the high version watermark indicating a highest version of the application program used to save the file;
  - comparing the high version watermark to the active version of the application program to determine whether the original data file corresponds to the later version of the application program; and
  - loading the original data file into the active version, wherein, when said comparing step determines that the original data file corresponds to the later version, said loading step comprises the steps of:
    - ignoring unknown information in the original data file; and
    - loading known information from the original data file into the active version.
2. A method according to claim **1**, wherein said original data file comprises an object property list.
3. A method according to claim **2**, wherein said object property list contains an object property list array.
4. A method according to claim **3**, further comprising the steps of:
  - tracking a position of objects in the object property list array; and
  - updating the object property list array based on the position of the objects tracked in said tracking step.
5. A method according to claim **1**, wherein the unknown and known information each comprise a property of an object property list.
6. A method according to claim **1**, further comprising the step of saving a modified version of the original data file, said saving step comprising the steps of: