

versions of the application program can be designated as higher numbers. For example, the second version can be designated as version 2.0. Version 1.0 is a previous version with respect to version 2.0. Version 2.0 is a later version with respect to version 1.0.

[0056] The present invention can be used with OPL arrays and conventional OPLs described above as the structure of the data file. However, as understood by those skilled in the art, the present invention is not limited to those file structures.

[0057] According to an exemplary embodiment of the present invention, file version watermarks can be provided in an original data file of an application program to indicate various properties of the original data file. The file version watermarks can be provided in the file header, which provides information about the file. A high version watermark can be provided to indicate the highest version of the application program used to save the file. A last version watermark can be provided to indicate the last version of the application program used to save the file. A low version watermark can be provided to indicate the lowest version of the application program used to save the file. A creation version watermark can be provided to indicate the version of the application program that first created the file. An object version watermark can be provided as a property on each object 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. When loading a file, the present invention can determine any data structures and properties that have been added or modified since the low watermark version. The added or modified items can then be corrected to be compatible with any version.

[0058] In operation, the high version watermark in the data file can be compared to the active version of the application program. By such comparison, it can be determined whether the original data file corresponds to the active version, a previous version, or a later version of the application program. Since the entire file can be saved by the application program, objects in the file include the characteristics of a file created by a version identified by the high version watermark. The application program can then determine how to load and/or save the original data file based on whether it corresponds to the active version, the previous version, or the later version. The original data file can comprise an object property list, an object property array, or other file structures.

[0059] For example, version 2.0 (the later version) of an application program can determine whether to convert information based on how version 2.0 treats the information differently from version 1.0 (the previous version). For instance, if version 1.0 had a tracking table for hyperlinks, but version 2.0 places the hyperlink data in the objects themselves, then version 2.0 can determine that it should convert that information when loading a file written by version 1.0. However, if version 2.0 was the oldest version to write the file, then version 2.0 can determine that such a conversion is not necessary.

[0060] The low and high version watermarks (and the object watermarks) can be updated if they exceed the previous mark. The last version to save the file can be

updated on save. The creation version watermark can be provided when the file is originally created and is not typically updated.

[0061] Referring now to FIG. 5, a method 500 of loading a data file according to an exemplary embodiment of the present invention will be described. FIG. 5 is a flow chart depicting method 500 for loading an original data file in an application program so that the original data file is compatible with an active version, a previous version, and a later version. Method 500 can comprise step 510 of providing a high version watermark, a low version watermark, a last version watermark, a creation version watermark, and/or an object version watermark in an original data file. In step 520, the high version watermark in the original data file is determined. The method also can determine the low version watermark, the last version watermark, the creation version watermark, and/or the object version watermark of the objects in the original data file, as shown in step 530. In step 540, the high version watermark is compared to the active version of the application program to determine if the high version is newer than the active version. Based on that comparison, the method determines how to load the original data file.

[0062] If the comparison of step 540 indicates that the high version watermark represents a version of the application program that not newer than the active version, then the original data file corresponds to the same version or a previous version of the application program with respect to the active version. In that case, the method branches to step 560 where the original data file can be loaded into the application program without changes. Because the high version watermark corresponds to the same version or a previous version of the application program, the active version can read and understand all information in the original data file.

[0063] If the comparison of step 540 indicates that the high version watermark represents a version of the application program that is newer than the active version, then the original data file corresponds to a later version of the application program with respect to the active version. In that case, the method branches to step 550 to load the original data file. Because the high version watermark corresponds to a later version of the application program, the active version may not be able to read and understand all information in the original data file. Accordingly, the original data file is loaded in the application program in step 550 by loading all known information and skipping all unknown information. When an OPL or OPL array is used as the file structure, unknown properties of an OPL associated with an object can be maintained by loading all unknown properties and storing them in a memory block until the file is saved. Alternatively, the program can track a location of the original OPL associated with an object on the disk and can read the unknown properties of an OPL back into memory as needed before overwriting the file.

[0064] After step 550 or 560, the method proceeds to step 570 where it is determined whether the last version watermark in the original computer file represents a version of the application program that is older than the active version. If the determination in step 570 is negative, then the method ends. If the determination in step 570 is affirmative, then the method branches to step 580 where the original data file can