application may be dependent upon content and/or properties of the application file that the less featured (or older version) application does not support. In such circumstances, a user of the less featured application may modify the property that the less featured application supports; however, because the less featured application does not support the additional capability, content, and/or properties it may not update the application file correctly with respect to the unsupported (or unknown) content.

For ease of discussion, embodiments and examples disclosed herein will be described with respect to a diagramming application, such as VISIO® provided by the Microsoft Corporation of Redmond, Wash. In a first embodiment, a full featured version of a diagramming application may provide properties for a shape such as color, shading, and transparency. The transparency property of the full featured application may be dependent upon the value of the shading property, so the first application calculates a value for the transparency property based upon the value of the shading property. The values for these properties may then be saved into an application file. The application file may then be opened by a second, less featured application. The less featured application may support the color and shading properties for shapes, but it may not support the transparency property or it may not treat the transparency property as dependent upon the shape. The second application may then modify the shading property of the shape and write the modification to the application file. However, because the second application does not recognize or support the dependency between the transparency property and the shading property, the second application may not properly update the transparency property in response to modifying the shading property. Because the transparency property was not correctly updated by the second application, when the first application subsequently accesses the application file, the first application may not be able to correctly display the transparency property of the shape. Although a diagramming application is used in the example, one of skill in the art will appreciate that the described problem is prevalent in other types of applications such as, but not limited to, word processing applications, spreadsheet applications, presentation applications, or any other type of versioned applications (or different applications) capable of sharing files. One of skill in the art will appreciate that the embodiments disclosed herein may be practiced with other diagramming applications or other types of applications.

Embodiments of the present disclosure provide an application with the ability to provide meaningful values for both native and extended objects, properties, relationships, formulas, and/or any other component of the self-describing file. In embodiments, a native object, property, relationship, formula, and/or component is a portion of a self-describing file that the application supports. As such, an application may be capable of properly handling and/or providing a value for a native component of the file. An extended object, property, relationship, formula, and/or component may be a portion of a self-describing file that an application may not support. As such, an application may not be capable of properly handling and/or providing a value for an extended portion of the file.

Embodiments of the present disclosure solve the exemplary problems by providing a self-describing file that may be shared between different versions of applications. In embodiments, the self-describing file may be used to provide an application with the information that may be used to correctly calculate (or otherwise maintain) file data even if portions of the file data are not supported by the application manipulating the self-describing file. The self-describing file may include information that allows an application to recalculate native and extended properties when modifying the self-describing file. Native properties may be values of the self-describing file that the application supports (e.g., objects, properties, formulas, etc. that are known to the application). Extended properties may be values of the self-describing file that the application does not support. In embodiments, the self-describing file may contain an extension section, or may otherwise store data that provides information related to the treatment of one or more sections of data in the self-describing file. In one embodiment, the information related to the treatment of the data may be provided in an extension section. The extension section may describe the proper calculation of data relied upon for file content, e.g., including file content that a version of an application may not support. The extension section thereby allows the less featured version of the application to properly preserve unknown file content.

In additional embodiments, a full featured version of an application is able to detect when another application has not properly calculated values of the file content. In such embodiments, the full featured version of the application may recalculate the miscalculated values and properly update the content of the file.

FIG. **1** is an embodiment of a self-describing file **100** for preserving unknown file contents for files that may be shared across different versions of an application. In embodiments, the self-describing file may contain one or more objects, such as Object **1 106**A and Object N **106**B. Although the self-describing file **100** illustrated in FIG. **1** contains two objects, the self-describing file **100** may contain any number of objects as illustrated by the ellipsis. In embodiments, the objects may relate to text, shapes, values, data structures, or any other type of data capable of being stored a file.

In embodiments, each object may have one or more properties associated with it, such as Property **1 108**A, Property **2 108**B, and Property **3 108**C associated with Object **1 106**A and Property **1 108**D, Property **2 108**E, and Property **3 108**F associated with Object N **106**B. Although Object **1 106**A and Object N **106**B are illustrated as having a three properties associated with each, one of skill in the art will appreciate that objects stored in the self-describing file **100** may have any number of properties associated with them or may not have any properties associated with them.

The objects and properties illustrated in FIG. **1** may differ depending on the application that the self-describing file **100** is employed with. For example, if the self-describing file **100** is associated with a diagramming application, the objects may relate to different shapes and the properties may relate to different characteristics of the shape (e.g., color, shading, transparency, size, outline, etc.) In another embodiment, if the self-describing file **100** is associated with a word processing application, the object may be related to a paragraph or text and the properties may be related to characteristics such as font face, font size, style, etc. In yet another embodiment, if the self-describing file **100** is associated with a spreadsheet application an object may represent a table or cell and the properties may represent characteristics such as values, formulas, fonts, etc. In alternate embodiments, a property may be any type of data included in the application. While specific examples of object and application are described herein, one of skill in the art will appreciate that these examples do not limit the scope of the disclosure and the embodiments of the self-describing file **100** disclosed herein may be employed regardless of the application and/or application data the self-describing file **100** is associated with.

In embodiments, the self-describing file **100** may contain an extension section **102**. The extension section **102** contains