

whenever any change is made to data within the object, including the creation of new data. Invalidations should not occur when changes involve only version-specific properties or other object metadata or meta information.

[0058] In an alternative embodiment, a replicator application may create and use the version-specific property. In such a case, the replicator application may want its version-specific property to be invalidated whenever any change is made to either the data or metadata of the object. These particular version-specific properties should be invalidated whenever any change is made that does not solely affect this class of version-specific properties. In this instance, by defining a specific class we prevent the presence of multiple replicators from causing an object to be replicated over and over again, where each instance of replication by one replication application appears as a need to replicate by a second replicator application.

[0059] Invalidate operation 406 invalidates the version-specific property. Invalidating a version-specific property may be accomplished, in one embodiment, by deleting or truncating that version-specific property. Alternatively, the invalidate operation 406 marks the version-specific property or otherwise provides an indication that the property is invalid. In essence, the act of invalidating the property must allow the application that relies on the version-specific property to recognize that the version-specific property has been invalidated, such as by setting its contents to an empty string. Other methods of invalidating the version-specific property may include adding more information to the invalidating access or to the version-specific property or modifying the information therein. The invalidation act 406 does not change other properties, such as the property related to the last time the object was access or written to, etc.

[0060] If the data object has more than one associated version-specific property then operations 404 and 406 are repeated for each version-specific property.

[0061] Following the invalidation act 406, perform access operation 408 performs the requested access on the object. Importantly, the act of performing the access on the object is initiated after the version-specific property has been invalidated. Otherwise, the version-specific property may be unreliable. Following perform access operation 408, flow 400 ends at end operation 410.

[0062] If determination act 404 determines that the access attempt does not relate to an invalidating access, then flow branches NO to determination operation 412. Determination operation 412 determines whether the access depends on a version-specific property, e.g., whether the access attempt is an access attempt by an application that uses the version-specific property in performing its operations. For example, a virus scan application that uses version-specific properties may be generating the access attempt, and in such a case, determination operation 412 determines that the access attempt is generated by such an application that uses the version specific properties.

[0063] If the access attempt is not performed by an application relying on version-specific properties then flow branches NO to perform access operation 408. Perform access operation 408 performs the originally requested access operation. In the case where perform access operation 408 follows determination operation 412, then the access

operation does not relate to an operation requiring invalidation, as determined by determination act 404. For example, the access attempt may relate to a read operation wherein the user is not allowed to modify any actual data. Following perform access operation act 408 the process flow ends at 410 as discussed above.

[0064] If determination operation 412 determines that the access does depend on version-specific property, then flow branches YES to determination act 414. Determination act 414 analyzes the version-specific property to determine whether it is valid. Determining whether a property is valid may involve determining whether the property exists, whether it has been marked as invalid, or whether it has been truncated so that information is not fully available, among other methods of determining whether the property is valid. If determination act 414 determines that the property is not valid, then flow branches NO to perform access related to invalid property act 416.

[0065] Perform act 416 performs a predetermined function based on the determination that the access is invalid. In an embodiment, if the access attempt is generated by a virus scan application, then perform operation 416 relates to performing such a virus scanning function on the object.

[0066] Following the performance of the operation related to invalid property 416, create version-specific property act 418 creates or validates a version-specific property with respect to that object. Validating a version-specific property may involve creating a version-specific property for that object, or adding more information to the existing version-specific property. Other embodiments may validate the property in other ways, as long as the property appears valid at a later time by providing an indication the property is valid. Following the validate version-specific property act 418, process 400 ends at end operation 410.

[0067] If determination act 414 determines that the version-specific property was valid, then flow branches YES to perform operation 420. Perform operation 420 relates to performing any act related to a determination that the version-specific property was valid. Depending upon the particular application that uses version-specific properties, perform operation 420 may comprise different types of operations. In one example, if the property is determined to be valid, then no operation is performed such as in the virus scan situation. In such a case, if the determination act 414 determines that the version-specific property is valid, then the object has been scanned previously and another scan is not necessary. Consequently, no operation is to be performed on that object and operation 420 simply passes to control end operation 410. In other embodiments, a determination that the version-specific property is valid may initiate an operation be performed at 420.

[0068] The above described system and method provides a significant advantage over prior methods of providing version information to program modules or applications since the version information is part of the object or object collection, as opposed to being resident in a separate log or database file. The accessing application is not required to access exterior-type files that may be unavailable, corrupt or otherwise unmanageable. Additionally, the property is not lost when the power is shut off and the property does not consume extraordinary amounts of operating memory, as is the case for in-memory information logs. Moreover, the