TABLE 3-continued

```
hash="ac72753e5bb20446d88a48c8f0aaae769a962338"
hashalg="SHA1"/>
    <file name="atl.dll"
hash="a7312a1f6cfb46433001e0540458de60adcd5ec5"
hashalg="SHA1">
        <comClass description="Registrar Class"
clsid="{44EC053A-400F-11D0-9DCD-00A0C90391D3}"
progid="ATL.Registrar"/>
        <interface iid=" {B6EA2051-048A-11D1-82B9-
00C04FB9942E}" name="IAxWinAmbientDispatch"
proxyStubClsId32="{00020424-0000-0000-C000-000000000046}"/>
        <typelib tlbid="{44EC0535-400F-11D0-9DCD-
00A0C90391D3}" version="1.0" helpdir=""/>
    </file>
    <file name="msvcrt.dll"
hash="ba62960ceb15073d2598379307aad84f3a73dfcb"
hashalg="SHA1"/>
    <file name="msvcirt.dll"
hash="84eb92153ff379c67c2727cc7f6931e011ff8121"
hashalg="SHA1"/>
    <file name="msvcp60.dll"
hash="96952787a1676e38107ab93c6a33b9bcda1c912e"
hashalg="SHA1"/>
</assembly>
```

[0048] Although the application manifest **204** lists its dependencies on assembly versions, it should be noted that technically, the application **200** is only dependent on the manifest-specified assemblies themselves, not necessarily the exact versions that are specified. Instead, the specified versions are only those tested with the application and known to work.

[0049] In accordance with one aspect of the present invention, the operating system (e.g., including an assembly loading mechanism) can bind the application to another version of a needed assembly in place of one specified in the application manifest **204**. To this end, configuration information that is separate from the shared assembly may be interpreted to determine which version to bind. Note that a configuration is separate from the shared assembly or assemblies to which it refers, for example, in that it is a separable entity therefrom, normally contained as a file, that can be added to or removed from a system at a later time, and so on.

[0050] In a first alternative mode, as shown in **FIG. 2**A, an assembly publisher can issue a publisher configuration **220** with data **222** therein that redirects a requested assembly version (e.g., requested via the manifest or the application configuration) to one that is specified by the assembly publisher. In general, a publisher configuration file applies to a single assembly, although a single assembly may have multiple associated publisher configuration files. Note that a publisher configuration applies to any application in the system that uses the corresponding assembly.

[0051] The publisher configuration enables an assembly publisher to effectively replace one assembly version with another version, such as when a service pack is installed. As described herein, in a first alternative mode, the assembly publisher configuration is interpreted after the manifest is interpreted, and thus can change an assembly initially specified by the application author. For example, in **FIG. 2**A, the publisher configuration **220** may include data that redirects the binding specified in the manifest (version 1.0.0.0) to version 2.0.0.0 (shared assembly$_x$ **208**$_2$).

[0052] As also represented in **FIG. 2**A, an application configuration **216** including updated assembly version dependency data **218** may be stored in the application folder **202** to similarly override some or all of the information in the application manifest **206** and/or any modification by the publisher configuration **220**. For example, as represented in **FIG. 2**A, the application configuration **218** may include data (e.g., an instruction) that redirects the current binding from version 2.0.0.0 to version 3.0.0.0 (the shared assembly$_x$ **208**$_3$), e.g., redirect version 2.0.0.0 to version 3.0.0.0. In this manner, an application author can effectively update a manifest and/or override a publisher configuration by distributing such a configuration **216**, without having to actually make changes to the application manifest **204** or have a new one reinstalled. Note that other instructions directed to other dependent assemblies may be in the same configuration, and that there may not be any relevant instruction for a given assembly.

[0053] In a second alternative mode, the order of applying configurations is modified, and also at least one other configuration may be applied. More particularly, as shown in **FIG. 2**B, the application configuration **216** including updated assembly version dependency data **218** may be stored in the application folder **202** to override some or all of the information in the application manifest **206**, and is applied before the publisher configuration is applied. For example, as represented in **FIG. 2B** by the arrow from block **218** to block **208**$_4$, the application configuration **218** may include data (e.g., an instruction) that redirects the manifest-specified binding to version 2.0.0.0 (the shared assembly$_x$ **208**$_2$), e.g., redirect version **1.0.0.0** to version **2.0.0.0**. In this second alternative mode, an application author can thus similarly effectively update a manifest without actually changing the manifest.

[0054] As also represented in **FIG. 2**B, the assembly publisher can similarly issue a publisher configuration **220** with data **222** therein that thereafter redirects the currently requested assembly version (in the present example modified by the application configuration) to one that is specified by the assembly publisher. This mode enables assembly publishers to effectively replace selected assemblies with other versions. In this second alternative mode, because the assembly publisher configuration is interpreted after the manifest and/or any application configuration is interpreted, the publisher can thus change the assembly specified by the application author. In the example shown in **FIG. 2**B, the publisher configuration **220** includes data that redirects the binding to version 4.0.0.0 (shared assembly$_x$ **208**$_4$), as shown in **FIG. 2B** via the arrow from block **222** to block **208**$_3$.

[0055] Lastly, in this second mode, an administrator configuration **224** and its version data **226** may have the final decision as to which version of an assembly an application (any application on the machine) will be bound. For example, in **FIG. 2**B, the administrator configuration **224** is stored as a file in a system folder **228** and includes data **226** that redirects the current binding to an earlier version, version 3.0.0.0 (shared assembly$_x$ **208**$_3$), as shown in **FIG. 2B** via the arrow from block **226** to block **208**$_3$. As can be understood, the administrator configuration **224** allows an administrator or the like to bind any system applications to a new assembly, or a restore a binding to an older assembly, (e.g., version 4.0.0.0 back to version 3.0.0.0 as shown in