

# **ANEXO B**

## **IMPLEMENTACION DE LOS ENSAMBLADOS**

La implantación de los assemblies puede realizarse utilizando cuatro mecanismos:

- a.- Instalación en el directorio privado de la aplicación: un simple XCOPY.
- b.- Utilización del mecanismo de probing.
- c.- Utilización de codebase.
- d.- Instalación en la Global Assembly Cache (GAC).

### **a.- Instalación en el directorio privado de la aplicación: XCOPY**

Es un mecanismo muy simple, y acorde con la idea de .NET de que las dependencias de una determinada aplicación se almacenen en el propio directorio (o subdirectorios) de la aplicación.

<b>Ventajas</b>	<b>Inconvenientes</b>
<ul style="list-style-type: none"><li>- Simplicidad.</li><li>- El código es prácticamente privado, y ninguna otra aplicación lo podrá utilizar.</li></ul>	<ul style="list-style-type: none"><li>- La localización del assembly se realiza sólo por el nombre. Por tanto, no es posible conseguir el efecto lado a lado.</li><li>- Las actualizaciones son más complejas, dado que se necesita saber con exactitud dónde está localizado el assembly.</li><li>- El strong name (si lo tuviese) se verifica en cada carga del assembly.</li><li>- Posible working set adicional. Es decir, si diez aplicaciones tienen que cargar el mismo assembly, habría diez assemblies cargados en memoria.</li></ul>

### **b.- Probing**

Especifica los subdirectorios base de la aplicación en los que busca Common Language Runtime cuando se cargan los ensamblados.

```
<probing privatePath="paths"/>
```

Donde:

**privatePath:** Especifica los subdirectorios del directorio base de la aplicación que puede contener ensamblados. Delimita cada subdirectorio con un punto y coma.

A continuación se muestra cómo especificar los subdirectorios base de la aplicación en los que el motor de ejecución debe buscar ensamblados.

```
<configuration>
  <runtime>
    <assemblyBindingxmlns="urn:schemas-microsoft-com:asm.v1">
      <probing privatePath="bin;bin2\subbin;bin3"/>
    </assemblyBinding>
  </runtime>
</configuration>
```

<b>Ventajas</b>	<b>Inconvenientes</b>
<ul style="list-style-type: none"><li>- Simplicidad.</li><li>- El código es prácticamente privado, y ninguna otra aplicación lo podrá utilizar.</li><li>- Posibilita el lado a lado</li></ul>	<ul style="list-style-type: none"><li>- Se necesita archivo de configuración para la aplicación.</li><li>- Las actualizaciones son más complejas (se necesita saber con exactitud dónde está localizado el assembly).</li><li>- El strong name (si lo tuviese) se verifica en cada carga del assembly.</li><li>- Posible working set adicional.</li></ul>

### **c.- Codebase**

La sección codebase en un archivo de configuración de una aplicación, especifica la URL en la que se puede localizar un assembly concreto.

```
<codeBase
  version="Assembly version"
  href="URL of assembly"/>
```

**href :** Especifica la dirección URL en la que el motor de ejecución puede localizar la versión especificada del ensamblado.

**version:** Especifica la versión de ensamblado que aplica el código base. El formato del número de versión de un ensamblado es *principal.secundaria.compilación.revisión*. Los valores válidos para cada parte del número de versión van de 0 a 65535.

```

<configuration>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="myAssembly"
          publicKeyToken="32ab4ba45e0a69a1"
          culture="neutral" />
        <codeBase version="2.0.0.0"
          href="http://www.litwareinc.com/myAssembly.dll"/>
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>

```

<b>Ventajas</b>	<b>Inconvenientes</b>
<ul style="list-style-type: none"> <li>- Soporte paralelo</li> <li>- Al utilizar una URL para localizar y cargar el assembly, el proceso de actualización (desde un servidor, por ejemplo) es más sencillo.</li> </ul>	<ul style="list-style-type: none"> <li>- Se necesita archivo de configuración para la aplicación.</li> <li>- El assembly requiere strong name.</li> <li>- El strong name se verifica en cada carga del assembly.</li> </ul>

**d.- Global Assembly Cache (GAC)**

La Global Assembly Cache (GAC) es una caché utilizada para almacenar assemblies que puedan ser utilizados por cualquier aplicación dentro del equipo local. La GAC está presente en cualquier equipo en el que esté instalada la CLR de .NET. Se recomienda utilizar la GAC para compartir assemblies sólo cuando sea realmente necesario.

Hay varias formas de instalar un assembly en la GAC la mas recomendada es utilizando Microsoft Windows Installer, por proporcionar un mecanismo automatizado de instalación en el cliente. Además proporciona contador de referencias, de cara a la posterior desinstalación.

<b>Ventajas</b>	<b>Inconvenientes</b>
<p>Ubicación compartida. Los ensamblados que van a usar las aplicaciones se pueden poner en la caché de ensamblados global. Si las aplicaciones usan un ensamblado de la caché de ensamblados global, se puede agregar una instrucción de directiva de versión al archivo Machine.config que redirige las referencias al ensamblado.</p> <p>Seguridad de archivos. Los administradores protegen el directorio WINNT con una Lista de control de acceso (ACL) para controlar el acceso de escritura y ejecución. La caché de ensamblados global al estar instalada en el directorio WINNT, hereda la lista (ACL) de dicho directorio.</p> <p>Control de versiones simultáneas. Se pueden guardar muchas copias de ensamblados con el mismo nombre pero con distinta información de versión.</p> <p>Ubicación de búsqueda adicional</p>	<ul style="list-style-type: none"> <li>- El assembly requiere strong name.</li> <li>- Requiere instalación (gacutil o típicamente Windows Installer). Además, dicha instalación requiere permisos de Administrador.</li> <li>- No es posible un Xcopy.</li> <li>- Cualquier aplicación podrá acceder al assembly.</li> </ul>