Setting Process-Wide Security Through the Registry - Win32 apps

By stevewhims

Archived: 2026-04-06 01:13:52 UTC

If you want to set security for an entire process, one solution is to set the security levels you want in the registry. If your application cannot call **CoInitializeSecurity** or if you prefer not to use programmatic security, this might be a good option. If you decide to set process-wide security using the registry, you should be aware that if you call **CoInitializeSecurity** within your program COM will use the values in **CoInitializeSecurity** and ignore the registry values.

There are two ways to set security in the registry for your application:

- You can use Dcomcnfg.exe, which provides a simple user interface for modifying security values. All COM servers can be configured using Dcomcnfg.exe. For more information, see Setting Process-Wide Security Using DCOMCNFG. However, client applications do not normally appear in Dcomcnfg.exe unless the client creates a GUID and enters it in the registry.
- You can set security values under the AppID key for the application. The rest of this topic explains how to set security in the registry using the **AppID** key.

An AppID is a GUID that represents a server process for one or more classes. Each class is associated with exactly one AppID, and AppIDs can be assigned only to EXEs. DLLs do not get AppIDs unless they are running in a surrogate, and then it is the surrogate process that has the AppID. If multiple DLLs are loaded into a surrogate, each surrogate has only one AppID.

For some COM servers, the registration code generates an AppID and places entries in the registry that map the AppID to the name of the executable. But some COM servers do not provide this functionality. However, if the server's registration code adds an entry for HKCR\CLSID{ServerCLSID}\LocalServer32 when dcomcnfg.exe is run, it will automatically add an AppID for the CLSID.

For a COM client that is not a server, this mapping is not created because the client is never registered. Therefore, to set security using the **AppID** key, the client must create the necessary registry entries, either programmatically by using the registry functions or by using regedit.

If you decide to set process-wide security in the registry under the **AppID** key, be aware that there are two named values under the **AppID** key that you can set without having administrator permissions:

- AccessPermission
- AuthenticationLevel

The **AuthenticationLevel** and **AccessPermission** values are set independently and have separate default values. If the **AuthenticationLevel** value is not present, the LegacyAuthenticationLevel value is used as the default.

Similarly, if the **AccessPermission** value is not present, the DefaultAccessPermission value is used as the default. However, the **AuthenticationLevel** and the **AccessPermission** values are interrelated in the following ways:

- If the **AuthenticationLevel** is none, the **AccessPermission** and **DefaultAccessPermission** values are ignored for that application.
- If the **AuthenticationLevel** is not present and the **LegacyAuthenticationLevel** is none, the **AccessPermission** and **DefaultAccessPermission** values are ignored for that application.

Setting Process-Wide Security
