

.NET Envelope Runtime API

The .NET Envelope Runtime API helps you implement secure license management using the Sentinel LDK .NET Envelope. Methods are encrypted using the AES key of a specific Feature ID.

If the license for the specific Feature is not available or has expired, decryption through HASP is also not available. If the end user attempts to use the Feature and a protected method is invoked, the Envelope Runtime cannot decrypt the method in order to continue the execution. The end user is shown a notification (message box), asking the user for the license. At this point, the only option that can be offered to the user is “Retry” (check for the license again in order to decrypt the method). If the user does not have the required license, the program is in a “deadlock” state.

To avoid this state, and to provide more useful options to the end user, the developer can use **hasp_get_info()** functionality to check which Features are available. If a Feature is not available, the application can be prevented from calling a method that is protected using the specific Feature.

However, it can be difficult to implement the logic of checking beforehand and blocking access to all functionality that requires a license which is not available.

The .NET Envelope Runtime API simplifies this task without compromising security. The API enables the developer to register a **NotificationDelegate** handler. This handler is invoked whenever the .NET Envelope Runtime cannot decrypt a protected method.

The event is thrown in case a method is protected, but the license is missing.

The **NotificationDelegate** handler implemented by the vendor can decide how to react to this event. Available responses are:

- **EnvelopeRuntimeStatus.StatusThrowException**
Discontinue execution path by throwing an exception.
- **EnvelopeRuntimeStatus.StatusReturnNothing**
Discontinue execution path by returning a null reference (Nothing).
- **EnvelopeRuntimeStatus.StatusAlertAndRetry**
(Default behavior) Display a message box that asks the user for the license, and then retry.
- **EnvelopeRuntimeStatus.StatusRetry**
Silently retry.

The **NotificationDelegate** handler also receives the reason for the failure (HASP status code).

Using this information, the developer can display a custom “license required” dialog and allow the user to abort or retry the action.

To use this API, include the **Aladdin.HASP.EnvelopeRuntime** namespace in your source code.

For unprotected assemblies, the **Aladdin.HASP.EnvelopeRuntime.dll** assembly is required only at compile-time and protection-time. The .dll is *not* required for a protected assembly.

An example demonstrating the use of the .NET Envelope Runtime API can be found in the following locations:

- On the Sentinel LDK installation DVD:

\Windows\Installed\Samples\Envelope\EnvelopeRuntime.NET

- In a Sentinel LDK Installation:

\Program Files\SafeNet Sentinel\Sentinel LDK\Samples\Envelope\EnvelopeRuntime.NET

The file **Form1.cs** contains a C Sharp project script employing the .NET Envelope Runtime API. The compiled program **Sample.exe** can be found under the subdirectory **\bin\Release**.