

Sentinel Hardware Keys & API Implementation

Sentinel Keys provide hardware token-based licensing to your software application(s). In order to protect your application, you first need to design the protection strategy in the Sentinel Keys Toolkit (referred as Toolkit hereafter). The Toolkit is the main application using, which you will do everything from preparing a protection strategy to programming hardware keys.

The Toolkit provides two basic methods to protect your applications:

- 1) Shell Protection:** The method in which protective wrappers are put around the application quickly and easily. For more details, see Chapter 4, “Protecting Applications Using Shell,” on page 57.
- 2) API Protection:** The method in which you insert the Business Layer API functions into your application's source code. For more details, see Chapter 5, “Protecting Applications Using API,” on page 71.

Steps to Protect Applications Using API

1. Prepare a Conceptual Plan.
2. Add the API features to License Template.
3. Evaluate the Business layer API calls for Familiarity
4. Re-Build the license Template if required.
5. Add the business layer API's to your source code, Compile and Link.
6. Test your application

Prepare a Conceptual Plan:

In the initial stage you need to decide which software locks to use for protecting your application. The purpose of a software lock is to verify the presence of the correct Sentinel Key. You will begin by contacting the Sentinel Key for a license (SFNTGetLicense API call). Subsequently, you can craft variety of software locks to check the presence of the Sentinel Key, such as encrypting the data using the AES algorithm present in the key.

Add the API features to License Template.

A license template is a container of features that define your application protection strategy. The Toolkit assigns a unique license ID to every license template created/duplicated, so that multiple licenses can be programmed in a Sentinel Key.

A feature is the most-basic unit of an application protection strategy. The Toolkit assigns a feature ID to every feature created in a license template. When you

use API features to protect your applications, you need to add the Business Layer API functions into your application code.

You can create the following API features in the Toolkit:

(a) AES - A 128-bit AES algorithm-based feature that allows you to:

- Encrypt data
- Decrypt data
- Use the query-response protection 1
- Specify licensing controls (like, expiration date, expiration time, and an execution count).

(b) ECC - An ECC algorithm-based feature that allows you to:

- Digitally sign content
- Verify signed content
- Specify licensing controls (like, expiration date, expiration time, and an execution count).

(c) String - A data feature that can contain up to 256 ASCII printable characters.

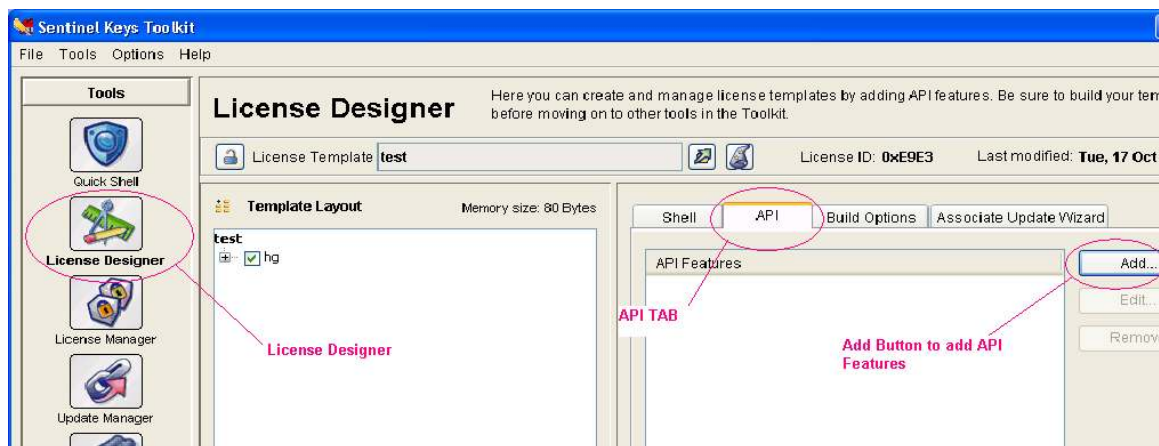
(d) Raw Data - A data feature that can contain 256-bytes of any developer-defined data type, including printable/non-printable characters and hexadecimal numbers. For example `_4ÖJëç:"A"g-Æµþ_n°_Ç&'_Â`.

(e) Integers - A data feature that can contain any of the following integers: 8-bit (0 to 255), 16-bit (0 to 65,535), or 32-bit (0 to 4,294,967,295).

(f) Boolean - A data feature that can contain a true or false value.

(g) Counter - A data feature that can contain a count-down value between 0 to 4,294,967,295.

To add API features you need to Open Toolkit -> Goto License Designer -> Select API's Tab and Click Add Feature. Here is a Sample Screen Shot showing the same:



Once we Click on Add button the Select API features Box opens where we can define and select various API features that we wish to include. Here is a sample screen:

Add Features

API Features

-  String
-  Raw Data
-  Integer
-  Boolean
-  Counter
-  AES
-  ECC

String Feature

Attributes

- Write-Random Read-Only
- Write-Once

Default instance

String value:
Password (hex): Write-Password

Override default instance

Add instances later
Maximum Size: (Maximum length can be 256)

A data feature that can contain up to 256 ASCII characters.

Feature Name: Feature ID: Constant name:

Description:

Evaluate the Business layer API calls for Familiarity

In the API Explorer screen, you can experiment with the Business Layer API prior to adding them into your source code. Corresponding to each function, it also generates the usage code for various languages.

Sentinel Keys Toolkit
File Tools Options Help

Tools

- Quick Shell
- License Designer
- License Manager
- Update Manager
- API Explorer**

Licenses

License Name: No License
License ID: 0x0000
Select License

API Explorer

Here you can evaluate the Business Layer API for the license template you have built in License Design. You may also obtain syntax for the API function in various compiler languages.

API arrangement
 List logically List alphabetically

API Functions

- Acquire License
 - SFNTSetContactServer**
 - SFNTGetLicense
- Maintain License
 - SFNTSetHeartbeat
- Release License
 - SFNTReleaseLicense
- ECC-Specific
 - SFNTSign
 - SFNTVerify
- AFS-Specific

Sets the Sentinel Keys Server to be contacted for obtaining the license.

Input value(s)
These are the business layer API's that can be executed here itself prior to modifying your source code.
Contact server: _____

License structure

Item	Value
------	-------

Information

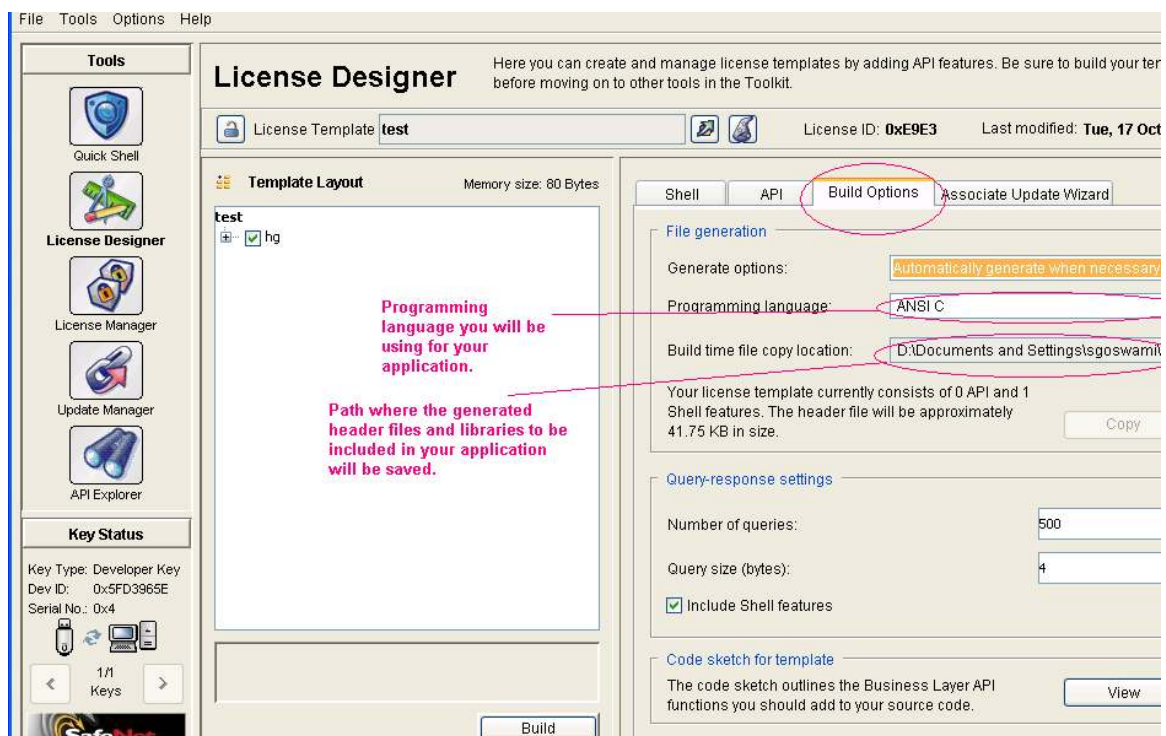
Select language: **C#.Net**

API function syntax:

```
unsigned long int SFNTSetContactServer(char *ContactServer);  
status = SFNTSetContactServer(ContactServer);  
if (status != SP_SUCCESS) {  
    // Check the status for reason of failure  
}
```

Re-Build the license Template if required.

In case you modified your API features or template properties, after evaluating the API functions, you need to re-build the license template to generate the "final set" of header file, libraries, and code sketch. The header file is generated at the time of building a license template. It contains important information for your (license) strategy, including the developer ID, license ID, feature ID, software key, query-response table (if you have included an AES feature in your template), and a public key (if you included a ECC feature in your template). The code sketch consists of an outline of the Business Layer API functions that you should incorporate in your source code. It is a good reference when you are not sure which API functions are relevant for your particular strategy.



Add the business layer API's to your source code, Compile and Link.

Add the Business Layer API Functions into Source Code, Compile, and Link
You now need to insert the Business Layer API calls into your application source code. The code sketch for your license template can guide you on the relevant API functions that can be called. Finally, compile and link your application after including the Sentinel Keys header files and libraries.

Test the Protected Application

You can now test your protected application. We recommend testing your application to verify that it executes correctly with the appropriate Sentinel Key both attached and missing.

Tip: If you are testing your protected application in network environment, make sure to restart the Sentinel Keys Server.