



SalientSNKFile Quick Start Guide

<http://www.SalientPoint.co.uk/>

Features

- *KeyFileInfo* and *KeyFileInfoPrivate* read and write SNK files
 - Access both public and private key SNK file formats
 - Use keys from SNK files with other RSA Cryptography functions in the .NET framework
 - Obtain the "Public Key Token" for any key

Installation

SalientSNKFile can be downloaded as either a ZIP or MSI (Windows Installer) file. In either case, you will be able to choose where the files are installed on your system. To use classes from SalientSNKFile in your project, you must add a DLL reference to SalientWatcher.dll (from bin folder under the installation directory) to your project.

Licensing

The default trial license for SalientSNKFile will cause a licensing exception to be thrown from time to time when using the library. When a license is obtained these exceptions will stop appearing. Licensing uses SalientProtect and works by adding an attribute to your application assembly or assemblies, enabling SalientSNKFile for assemblies signed with your strong name key. Licenses can be obtained through our website: <http://www.SalientPoint.co.uk/>.

More Information

Details of the classes in SalientSNKFile can be found in the supplied help file. Further information on all SalientPoint products including SalientWatcher can be found on our website: <http://www.SalientPoint.co.uk/>

Reading SNK Files

- Use the static *ReadSNKFile* function to obtain a *KeyFileInfo* object
- Extract the contents in RSA format using *GetRSAParameters*
- The key length can be obtained using the *KeyLength* property

The following example shows how to read a key file:

```
KeyFileInfo kfi = KeyFileInfo.ReadSNKFile( fileName );  
  
RSAParameters rsa = kfi.GetRSAParameters();  
  
int keyLength = kfi.KeyLength;
```

Writing SNK Files

- Create a new *RSACryptoServiceProvider* containing a new key
- Extract the key as *RSAParameters*
- Create a *KeyFileInfo* object wrapping the key
- Call the *WriteSNKFile* function on the *KeyFileInfo* object

The following example shows how to write a key file with a new key:

```
RSA crypto = new RSACryptoServiceProvider();  
  
RSAParameters privateKey = crypto.ExportParameters(true);  
  
KeyFileInfoPrivate kfi = new KeyFileInfoPrivate( privateKey );  
  
kfi.WritePrivateKey( "KeyFile_PrivateKey.snk" );  
  
kfi.WritePublicKey( "KeyFile_PublicKey.snk" );
```

Obtaining the Public Key Token

- Create a *KeyFileInfo* containing the key
- Read the token using the *PublicKeyToken* property

The following example shows how to obtain the Public Key Token:

```
KeyFileInfo kfi = KeyFileInfo.ReadSNKFile( fileName );  
  
string token = kfi.PublicKeyToken.ToString("X16");
```