Web Services Option Pack

WSOP v3.1.0 PKCS 11 Provider User Guide

10 April 2024

Table of Contents

1. Overview: Web Services PKCS #11 library

The Web Services PKCS #11 Library allows you to run PKCS #11 applications from a Web Services client.

The client machine that the PKCS #11 library is installed to does not require any existing Web Services or Security World Software to be present.

2. Install the Web Services PKCS #11 library

2.1. Linux

To install the nShield PKCS #11 library:

1. Sign in as a user with root privileges.
2. Change to the root directory.
3. Extract the PKCS #11 tar to the root directory. This installs all the files required by PKCS #11 to `/opt/nfast/webservices/pkcs11`.

   ```
   tar –xzf /path/to/nShield-WebServicesClient-Linux-1.1.0.tar.gz
   ```

4. If you are installing the PKCS #11 library for the first time, go to the `/opt/nfast/webservices/pkcs11/conf` folder and copy the `example_pkcs11webservices.cfg` file to `pkcs11webservices.cfg`.

   If you are reinstalling or updating the PKCS #11 library and do not want to overwrite your existing `pkcs11webservices.cfg` file, you can skip this step.

5. If required, change the log file destination or grant permission to the appropriate user to write to the default log folder: `/opt/nfast/webservices/pkcs11/log`. For help changing the logging path, see Log file path on Linux.

2.2. Windows

To install the nShield PKCS #11 library:

1. Ensure that `Microsoft Visual C++ 2015-2022 Redistributable (x64)` has been installed. A supported version is included in the installation media or can be downloaded from https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170.
2. Sign in as Administrator or as a user with local administrator rights.
3. Using the provided installation media, launch `setup.msi` manually.
4. Follow the onscreen instructions.
5. Accept the license terms and select **Next** to continue.

6. Specify the installation directory and select **Next** to continue.
7. If you are installing the PKCS #11 library for the first time, go to the `C:\ProgramData\nCipher\webservices\pkcs11\conf` folder and copy the `example_pkcs11webservices.cfg` file to `pkcs11webservices.cfg`.

   If you are reinstalling or updating the PKCS #11 library and do not want to overwrite your existing `pkcs11webservices.cfg` file, you can skip this step.
8. If required, change the log file destination or grant permission to the appropriate user to write to the default log folder: `C:\ProgramData\nCipher\webservices\pkcs11\log`. For help changing the logging path, see Log file path on Windows.

3. Configure the Web Services PKCS #11 library

The `pkcs11webservices.cfg` file contains an example PKCS #11 configuration. To use this file, you need to alter it to point to the correct certificate locations.

3.1. Default PKCS #11 configuration

The PKCS #11 library is installed with a default configuration. Entrust recommends reviewing and updating the initial configuration before the library is called to ensure that all configuration settings are appropriate for the deployment environment.

Pay special attention to the TLS connection and logging to ensure that the system is used securely.

Ensure that the configuration file and TLS authentication files have restrictive access control, so that only the application using the PKCS #11 client library has access to these files.

3.2. Configuration parameters for both Windows and Linux

The following configuration options are shared on both Linux and Windows.

3.2.1. Web Services server hostname

```
HOST <host name of server>
```

Set this to the Web Services server host name.

3.2.2. Web Services server port number

```
PORT 18001
```

Set this to the Web Services server port number.

3.2.3. Log level

The `LOGLEVEL` field in the configuration file controls the PKCS #11 library logging. The available log levels are:

- `0(None)`
- `1(Fatal)`
- `2(Error)`
- `3(Warning)`
- `4(Debug 1)`
- `5(Debug 3)`
- `6(Debug 9)`

By default, the PKCS #11 logging level is set at 4 (Debug 1).

Higher log levels include all logs from lower levels. If the logging level is set to `4(Debug 1)`, you only get log events with Debug 1, Warning, Error, and Fatal. If the logging level is set to `6(Debug 9)` it enables all the log levels. To turn off logging set `LOGLEVEL` to `0(None)`.

```
LOGLEVEL 4
```

Logs that are not at DEBUG9 level have the message prefixed with standard fields:

- `Timestamp with format [yyyy-mm-dd hh:mm:ss]`
- `Process ID`
- `Level (one of DEBUG3 DEBUG1 WARNING ERROR FATAL)`
- `PKCS #11 session handle (zero if not used)`
- `PKCS #11 library name`

Logs at DEBUG9 level add field for thread ID in prefix:

- `Timestamp with format [yyyy-mm-dd hh:mm:ss]`
- `Process ID`
- `Thread ID`
- `Level (DEBUG9)`
- `PKCS #11 session handle (zero if not used)`
- `PKCS #11 library name`

3.2.4. Enable logging to console

By default, PKCS #11 has console logging disabled. In order to enable this, set `LOGSTDOUT` to `1` in the configuration file.

```
LOGSTDOUT 1
```

3.2.5. Retry Web Services communication

Since the PKCS #11 library operates over a network connection, it is possible that network fluctuations could cause an internal request to the Web Services server to fail. To ensure the reliability and robustness of PKCS #11 applications, the request can be retried.

You can configure the maximum number of retries for each request and the delay (in seconds) between each retry. For example:

```
MAXRETRIES 5
RETRYDELAY 10
```

If you don't define either of these entries, the PKCS #11 library uses the defaults.

- For `MAXRETRIES`, the default is `5` and the maximum number is `256`.

  To disable retries, set `MAXRETRIES` to `0`.

- For `RETRYDELAY`, the default is `10` seconds. There is no maximum value but higher values will reduce performance on unstable systems. Only integer values are supported.

  To disable the retry delay, set `RETRYDELAY` to `0`. This will cause retries to occur consecutively without delay.

3.3. Configuration parameters for Linux only

For details on creating certificates, see Server/client mutual authentication with the Web Services PKCS #11 library.

3.3.1. TLS certificate

```
CERT <path to TLS certificate>
```

Set this to the file path of the client TLS certificate.

3.3.2. TLS private key

```
KEY <path to private key>
```

Set this to the file path of the client TLS private key.

3.3.3. TLS client authentication

```
AUTH <path to TLS CA Certificate for Mutual Authentication>
```

Set this to the server root certificate, which forms the trust anchor for authenticating the server's certificates received during TLS handshake.

3.3.4. Log file path on Linux

By default, PKCS #11 outputs the logs to `/opt/nfast/webservices/pkcs11/log/pkcs11webservices.log`. To change the default path set `LOGFILEPATH` to any valid existing path.

```
LOGFILEPATH /opt/nfast/webservices/pkcs11/log/pkcs11webservices.log
```

To enable logging for a user, change the path to a Linux user path so that the user has write permission to the folder.

```
LOGFILEPATH /home/<username>/log/pkcs11webservices.log
```

3.3.5. Enable logging to syslog

You can direct PKCS #11 logs to a syslog server. By default this configuration is disabled. In order to enable this, set `LOGSYS` to `1` in the configuration file.

```
LOGSYS 1
```

3.4. Configuration parameters for Windows only

For details on creating certificates, see Server/client mutual authentication with the Web Services PKCS #11 library.

3.4.1. TLS certificate name

```
CERTNAME <name of TLS certificate>
```

Set this to the common name of the client TLS certificate.

3.4.2. TLS certificate store

```
CERTSTORE <name of certificate store>
```

Set this to the name of the certificate store that contains the TLS certificate.

3.4.3. TLS certificate store type

```
STORETYPE <LocalMachine or CurrentUser>
```

Set this to the type of the certificate store that contains the TLS certificate.

If certificates are stored on `LocalMachine`, non-administrator users may lose access to their private key.

3.4.4. Log file path on Windows

By default, PKCS #11 outputs the logs to `C:\ProgramData\nCipher\WebServices\pkcs11\log\pkcs11webservices.log`. To change the default path set `LOGFILEPATH` to any valid existing path.

```
LOGFILEPATH C:\ProgramData\nCipher\WebServices\pkcs11\log\pkcs11webservices.log
```

To enable logging for a user, change the path to a Windows user path so that the user has write permission to the folder.

```
LOGFILEPATH C:\Users\<username>\log\pkcs11webservices.log
```

3.5. Enable logging to event viewer

You can direct PKCS #11 logs to event viewer. By default this configuration is disabled. In order to enable this, set `EVENTLOG` to `1` in the configuration file.

```
EVENTLOG 1
```

Messages in event viewer are limited to a log level of `2(Error)` and lower. Higher values will not produce additional events.

4. Server/client mutual authentication with the Web Services PKCS #11 library

4.1. Linux

For details on the TLS connection setup, see *WSOP TLS setup* in the *Web Services Option Pack User Guide*.

4.2. Windows

The Web Services PKCS #11 Library can only communicate securely with a WSOP Server if the following certificates are installed:

- The WSOP Server's CA certificate.
- An appropriate client certificate (with each PKCS #11 client using its own client certificate).
- Any intermediate CA certificates that are to be used to form a complete chain to verify the client certificate on the WSOP Server.

For information on these certificates, *WSOP TLS setup* in the *Web Services Option Pack User Guide*.

1. Install the WSOP Server's CA certificate into the `Root` store.
   Below is an example of how you can do this using `certutil.exe`.
   a. Add a CA certificate to the Root store

   ```
   certutil.exe -addstore Root <ca_certificate.pem>
   ```

   b. Check that the certificate has been installed:

   ```
   certutil.exe -store Root
   ```

2. Install any intermediate CA certificates for the client certificate.
   Below is an example of how you can do this by using `certutil.exe` to load the client certificate's intermediate certificates into the 'CA' certificate store:

   ```
   certutil.exe -addstore CA <intermediate_ca_certificate.pem>
   ```

3. Install the client certificate and its private key. This should be a PFX file that

contains a single certificate and the associated private key.

The PFX must not contain the full certificate chain.

For example, to install the PFX file in the Local Machine's certificate store

```
certutil.exe -p <password> -importPFX [certificatestorename] <client-cert.pfx>
```

or to install the PFX file into the Current User's certificate store

```
certutil.exe -p <password> -importPFX -user [certificatestorename] <client-cert.pfx>
```

4. If necessary, update the `pkcs11webservices.cfg` file so that
   - The common name of the client certificate contained within the PFX file matches the CERTNAME config file entry.
   - The `certificatestorename` used here matches the CERTSTORE config file entry.
   - The store type, `LocalMachine` or `CurrentUser` matches the STORETYPE config file entry.

For details on the Windows config file entries, see Configuration parameters for Windows only

5. Utilities in the Web Services PKCS #11 library

The following four utility programs are provided:

`ckcheckinst`	Checks basic functionality.
`ckinfo-dynamic`	Prints version information.
`cklist-dynamic`	Lists objects created on the Softcard.
`ckmechinfo`	Lists supported mechanisms.

Run these programs with the following commands:

Linux:

```
/opt/nfast/webservices/pkcs11/bin/ckcheckinst
/opt/nfast/webservices/pkcs11/bin/ckinfo-dynamic --library
/opt/nfast/webservices/pkcs11/lib/libpkcs11webservices.so
/opt/nfast/webservices/pkcs11/bin/cklist-dynamic --library
/opt/nfast/webservices/pkcs11/lib/libpkcs11webservices.so
/opt/nfast/webservices/pkcs11/bin/ckmechinfo
```

Windows:

```
C:\Program Files\nCipher\WebServices\pkcs11\bin\ckcheckinst.exe
C:\Program Files\nCipher\WebServices\pkcs11\bin\ckinfo-dynamic.exe --library "C:\Program
Files\nCipher\WebServices\pkcs11\lib\libpkcs11webservices.so"
C:\Program Files\nCipher\WebServices\pkcs11\bin\cklist-dynamic.exe --library "C:\Program
Files\nCipher\WebServices\pkcs11\lib\libpkcs11webservices.so"
C:\Program Files\nCipher\WebServices\pkcs11\bin\ckmechinfo.exe
```

5.1. Softcard generation tool

Because PKCS #11 does not directly support Softcard generation, a command line tool is provided.

The Softcard tool uses the same configuration file as the PKCS #11 library for the Web Services server secure connection. It does not support any logging. For more information, see Configure the Web Services PKCS #11 library.

To generate a new Softcard run the following command:

Linux:

```
/opt/nfast/webservices/pkcs11/bin/softcardtool -g --name=<new-softcard-name>
```

Windows:

```
C:\Program Files\nCipher\WebServices\pkcs11\bin\softcardtool.exe -g --name=<new-softcard-name>
```

When prompted, enter a new passphrase for the Softcard.

Special characters for name and passphrase are not supported.

To verify the Web Services server connection, run the tool with the verbose and list options:

Linux:

```
/opt/nfast/webservices/pkcs11/bin/softcardtool -vl
```

Windows:

```
C:\Program Files\nCipher\WebServices\pkcs11\bin\softcardtool.exe -vl
```

To delete a Softcard, remove all keys associated with the Softcard and use the following command:

Linux:

```
/opt/nfast/webservices/pkcs11/bin/softcardtool -d --ID=<deleted-softcard-ID>
```

Windows:

```
C:\Program Files\nCipher\WebServices\pkcs11\bin\softcardtool.exe -d --ID=<deleted-softcard-ID>
```

To see all the available options, run

Linux:

```
/opt/nfast/webservices/pkcs11/bin/softcardtool --help

softcardtool, 1.1.0

Usage:
       softcardtool [options]
```

Windows:

```
C:\Program Files\nCipher\WebServices\pkcs11\bin\softcardtool.exe --help

softcardtool, 1.1.0

Usage:
       softcardtool.exe [options]
```

```
Options:

Help options:
  -h, --help                Display help for `softcardtool'.
  -V, --version             Display the version number of `softcardtool'.
  -u, --usage               Display a brief usage summary for `softcardtool'.
  -v, --verbose             verbose output
  -i, --id=ID               ID of softcard to delete
  -n, --name=NAME           name of softcard to generate
  -l, --list                list softcards
  -d, --delete              delete softcard by ID
  -g, --generate            generate a new softcard

Generates softcards and deletes them.
```

6. Web Services PKCS #11 library compliance with the PKCS #11 specification

6.1. Supported functions

The following sections list the PKCS #11 functions supported by the PKCS #11 library. For a list of supported mechanisms, see Mechanisms.

6.1.1. General purpose functions

The following functions perform as described in the PKCS #11 specification:

- `C_Finalize`
- `C_GetInfo`
- `C_GetFunctionList`
- `C_Initialize`.

6.1.2. Slot and token management functions

The following functions perform as described in the PKCS #11 specification:

- `C_GetSlotInfo`
- `C_GetTokenInfo`
- `C_GetMechanismList`
- `C_GetMechanismInfo`
- `C_GetSlotList`.

`C_GetSlotList` returns a list of slot IDs. You cannot make any assumptions about the values of these handles. These handles are not equivalent to the slot numbers returned by the Web Services server.

6.1.3. Standard session management functions

The following functions perform as described in the PKCS #11 specification:

- `C_OpenSession`

- `C_CloseSession`
- `C_CloseAllSessions`
- `C_Login`
- `C_Logout`
- `C_GetSessionInfo`

6.1.4. Object management functions

The following functions perform as described in the PKCS #11 specification:

- `C_CreateObject`
- `C_DestroyObject`
- `C_GetAttributeValue`
- `C_FindObjectsInit`
- `C_FindObjects`
- `C_FindObjectsFinal`

`C_FindObjects` only returns objects that are supported by the PKCS #11 library. It does not list keys with the `CKO_PUBLIC_KEY` or `CKO_PRIVATE_KEY` object classes.

6.1.5. Encryption functions

The following functions perform as described in the PKCS #11 specification:

- `C_EncryptInit`
- `C_Encrypt`

6.1.6. Decryption functions

The following functions perform as described in the PKCS #11 specification:

- `C_DecryptInit`
- `C_Decrypt`

6.1.7. Sign functions

The following functions perform as described in the PKCS #11 specification:

- C_SignInit
- C_Sign

6.1.8. Verify functions

The following functions perform as described in the PKCS #11 specification:

- C_VerifyInit
- C_Verify

6.1.9. Key-management functions

The following function performs as described in the PKCS #11 specification:

- C_GenerateKey

C_CreateObject should not be used to create any key objects. Use C_GenerateKey to generate a secret key object.

C_GenerateKey will only generate key types supported by the PKCS #11 library.

String fields such as CKA_LABEL and CKA_APPLICATION should not contain any HTML character for example <, > or & as this can result in those values becoming corrupted when they are retrieved later.

6.2. Objects

The following table lists the objects currently supported by the PKCS #11 library.

Object	Notes
CKO_DATA	
CKO_CERTIFICATE	CKC_X_509 only
CKO_SECRET_KEY	CKK_AES CKK_SHA256_HMAC CKK_SHA384_HMAC CKK_SHA512_HMAC

6.3. Mechanisms

The following table lists the mechanisms currently supported by the PKCS #11 library and the functions available to each one.

Mechanism	Encrypt & Decrypt	Sign & Verify	SR & VR	Digest	Gen. Key/Key Pair	Wrap & Unwrap	Derive Key
CKM_AES_CBC_PAD	Y	—	—	—	—	—	—
CKM_AES_CBC	Y	—	—	—	—	—	—
CKM_AES_KEY_GEN	—	—	—	—	Y	—	—
CKM_SHA256_HMAC	—	Y	—	—	—	—	—
CKM_SHA256_HMAC_GENERAL	—	Y	—	—	—	—	—
CKM_SHA384_HMAC	—	Y	—	—	—	—	—
CKM_SHA384_HMAC_GENERAL	—	Y	—	—	—	—	—
CKM_SHA512_HMAC	—	Y	—	—	—	—	—
CKM_SHA512_HMAC_GENERAL	—	Y	—	—	—	—	—
CKM_SHA256_KEY_GEN	—	—	—	—	Y	—	—
CKM_SHA384_KEY_GEN	—	—	—	—	Y	—	—
CKM_SHA512_KEY_GEN	—	—	—	—	Y	—	—

In this table:

- Y indicates that the function is supported by the mechanism.
- — indicates that the function is not supported by the mechanism.

The AES mechanisms support three different key sizes: 16, 24, and 32 bytes. For non-padded AES mechanisms the plaintext size must be a multiple of the block size (16 bytes).

HMAC_GENERAL mechanisms support signature lengths ranging from half the output size to the full output size. For example CKM_SHA256_HMAC_GENERAL supports outputs in the range 16-32 bytes.

For more information, including minimum and maximum key sizes, run ckmechinfo in Utilities in the Web Services PKCS #11 library.

6.4. Attributes

All templates used to create objects or generate keys must contain the following attributes because there is currently no PKCS #11 library support for session or public objects:

Object Attributes	Required	Notes
CKA_TOKEN	Y	Must be CK_TRUE
CKA_PRIVATE	Y	Must be CK_TRUE

Data object creation is supported:

Data Object Attributes	Required	Notes
CKA_CLASS	Y	Must be CKO_DATA
CKA_APPLICATION	N	
CKA_OBJECT_ID	N	

The X.509 public object certificate creation is supported:

X.509 Certificate Attributes	Required	Notes
CKA_CLASS	Y	Must be CKO_CERTIFICATE
CKA_CERTIFICATE_TYPE	Y	Must be CKC_X_509
CKA_VALUE	Y	
CKA_SUBJECT	Y	
CKA_START_DATE	N	
CKA_END_DATE	N	
CKA_ISSUER	N	

The PKCS #11 library supports generation of the following key types:

- CKK_AES
- CKK_SHA256_HMAC
- CKK_SHA384_HMAC
- CKK_SHA512_HMAC

Key Attributes	Required	Notes
CKA_CLASS	Y	Must be CKO_SECRET_KEY
CKA_KEY_TYPE	Y	See list above
CKA_VALUE_LEN	Y	
CKA_LABEL	Y	
CKA_ENCRYPT	N	Should be CK_TRUE for AES
CKA_DECRYPT	N	Should be CK_TRUE for AES
CKA_SIGN	N	Should be CK_TRUE for HMAC
CKA_VERIFY	N	Should be CK_TRUE for HMAC
CKA_WRAP	N	Function not used in PKCS #11 library
CKA_UNWRAP	N	Function not used in PKCS #11 library

You should set these attributes to false while creating an object, if provided:

- CKA_DERIVE
- CKA_EXTRACTABLE
- CKA_COPYABLE
- CKA_TRUSTED
- CKA_UNWRAP
- CKA_WRAP
- CKA_WRAP_WITH_TRUSTED

You should set these attributes to true while creating an object, if provided:

- CKA_ALWAYS_SENSITIVE
- CKA_NEVER_EXTRACTABLE
- CKA_LOCAL
- CKA_SENSITIVE

The attributes not listed above or in tables are currently not supported.

The PKCS #11 library only supports token objects, not session objects.