Adobe Experience Manager Forms

nShield® HSM Integration Guide

29 Apr 2022

Contents

1. Introduction

Adobe Experience Manager Forms is an end-to-end digital document solution that makes it possible to create responsive forms that customers can complete and securely e-sign. Digital signatures in AEM Forms can use credentials stored in an Entrust nShield HSM to apply server-side digital signatures.

1.1. nShield configurations

We have successfully tested the integration of an nShield HSM with Adobe Experience Manager Forms in the following configurations:

nShield HSM	nShield Image	nShield Firmware	nShield Security World Software
Connect XC Connect +	12.80.4	12.50.11 12.50.8	12.80.4

1.2. Software configurations

We have successfully tested the integration of an nShield HSM with Adobe Experience Manager Forms using the AEM Forms on JEE deployment using the following versions:

Base OS	Java	AEM Forms	JBoss	MSSQL Server
Windows Server 2019	JDK 1.8.0_321	6.5.0	Red Hat JBoss EAP 7.4.0.GA	2019

1.3. Supported nShield HSM functionality

Feature	Support
Module-only key	Yes
OCS cards	Yes
Softcards	Yes
nSaaS	Yes
FIPS 140-2 Level 3	Yes

1.4. Requirements

Before starting the integration process, familiarize yourself with the Adobe Documentation and Software Requirements along with nShield Documentation. The following include links to documentation for Adobe Experience Manager Forms used in this integration:

- Supported Platforms for AEM Forms on JEE
- Installing and Deploying Adobe Experience Manager Forms on JEE for JBoss
- Preparing to install AEM Forms (Single Server)
- Managing HSM credentials

2. Procedures

2.1. Prerequisites

Before you can use Adobe Experience Manager Forms with the nShield HSM, complete the following steps:

1. Install the Java Development Kit.
2. Set up the HSM client software on the machine where Adobe Experience Manager Forms will be installed. See the *Installation Guide* for your HSM.
3. Configure the HSM(s) to have the IP address of your host machine as a client.
4. Load an existing Security World or create a new one on the HSM.
5. Create or edit the `cknfastrc` file in `nfast` directory, and add one of the following two config settings:

   Module protection:

   ```
   CKNFAST_FAKE_ACCELERATOR_LOGIN=1
   ```

   OCS or Softcard protection:

   ```
   CKNFAST_LOADSHARING=1
   CKNFAST_NO_ACCELERATOR_SLOTS=1
   ```

   Optional lines to enable debug:

   ```
   CKNFAST_DEBUG=5
   CKNFAST_DEBUGFILE=C:\pkcs11.log
   ```

6. Install Adobe Experience Manager Forms.

   For instructions, see the Adobe Documentation.

   Follow the Adobe documentation and set up AEM forms on a JEE deployment.

   Preparing to install AEM Forms (Single Server)

   When setting up, ensure that the user account has login permissions to the database server.

For more information on configuring and managing nShield HSMs, Security Worlds, and Remote File Systems, see the *User Guide* for your HSM(s).

2.2. Configure Java

You have to configure Java for the nShield HSM before you can use the HSM with Adobe Experience Manager Forms Credentials.

1. Add lines to `C:\ProgramData\nCipher\Key Management Data\config\config` about privileged and non-privileged ports:

   ```
   [server_startup]
   ...
   priv_port=9001
   nonpriv_port=9000
   ```

2. Set the path variables.

   Open a command prompt as Administrator and run:

   ```
   % setx JAVA_HOME "C:\Program Files\Java\jdk1.8.0_291"
   % setx PATH "%PATH%;%JAVA_HOME%\bin";
   ```

3. Copy the `nCipherKM.jar` file to the extensions folder of your local Java Virtual Machine installation from the following directory:

   ```
   %NFAST_HOME%\java\classes
   ```

4. Paste the file in the following directory:

   ```
   %JAVA_HOME%\jre\lib\ext
   ```

5. Download the JCE Unlimited Strength Jurisdiction Policy Files from your Java VM vendor's Web site. The downloaded Java 8 file used in this interop was `jce_policy-8`.
6. Extract and copy the extracted files `local_policy.jar` and `US_export_policy.jar` into the `security` directory:

   ```
   %JAVA_HOME%\jre\lib\security
   ```

7. Edit `%JAVA_HOME%\jre\lib\security`.
8. Add `security.provider.1=com.ncipher.provider.km.nCipherKM` to the top of the list of providers and shift the rest of the numbers down to keep them in ascending order.
9. Open a command prompt as Administrator and run:

   ```
   % java com.ncipher.provider.InstallationTest
   ```

   The output includes a list of providers and nShield JCE services.

Also check for the following phrases within the output:

```
Unlimited strength jurisdiction files are installed.
The nCipher provider is correctly installed.
```

2.3. Generate a signed certificate on the HSM

An nShield HSM will be used to generate a Certificate Signing Request to then be signed and imported. This certificate will be later used by AEM Forms Credentials.

If you are using FIPS 140-2 Level 3, PKCS #11 requires HSM OCS cards for FIPS authentication when you are importing the signed certificate. When you are running the `ckcerttool` command at a later step, you will have to insert the OCS card(s).

1. The following command can be used to generate an OCS or Softcard for the HSM:

   ```
   % createocs -m1 -Q 1/1 -N <cardset_name>
   % ppmk --new <cardset_name>
   ```

2. Open command prompt as administrator and run

   Module protection:

   ```
   % generatekey pkcs11 protect=module certreq=yes type=rsa size=2048 pubexp=65537 plainname=<key_name> nvram=no
   ```

   OCS protection:

   ```
   % generatekey pkcs11 cardset=<cardset_name> protect=token certreq=yes type=rsa size=2048 pubexp=65537
   plainname=<key_name> nvram=no
   ```

   Softcard protection:

   ```
   % generatekey pkcs11 softcard=<cardset_name> protect=softcard certreq=yes type=rsa size=2048 pubexp=65537
   plainname=<key_name> nvram=no
   ```

3. Take note of the path to the key and the CSR.
4. Take the CSR file to a Certificate Authority and have it signed.
5. Take the generated signed certificate file and place it in the same directory where the CSR file was originally generated.
6. Open command prompt as administrator and run one of the following to import the signed certificate:

   Module protection:

```
% ckcerttool -c <cardset name> -f <signed_cert_filename> -k <identof the key, the part after pkcs11_> -L
<label_for_the_key>
```

OCS and Softcard protection:

```
% ckcerttool -n -f <signed_cert_filename> -k <identof the key, the part after pkcs11_> -L <label_for_the_key>
```

OCS protection example:

```
% ckcerttool -c aemocs -f aemcertocs.cer -k ucdf5b8ad614c4790788582016043d54d23282013b-
fcc2027b509bf11dfff2d5e91c83229eb389b2c1 -L AEMocsprivateKey
```

2.4. Configure the HSM credential alias

If you completed the previous steps while the Application Server was running, you might need to restart the Application Server before you configure the HSM credential alias because AEM Forms might not recognize the HSM certificate yet.

1. Open the administrative console of AEM Forms in a web browser at `http://localhost:8080/adminui`.
2. Select **Settings**.
3. Select **Trust Store Management**.
4. Select **HSM Credentials**.
5. Enter a **Profile Name** for the HSM.
6. Enter the path of the `pkcs11` library:

   ```
   C:\Program Files\nCipher\nfast\toolkits\pkcs11\cknfast.dll
   ```

7. Select **Test HSM Connectivity**.

   A success message **HSM is available** appear.

8. For the **Token Name**, select accelerator for module protection or the card set name for OCS/Softcard protection.
9. The corresponding **Slot ID** and **Slot List Index** values will be selected automatically.
10. For the **Token Pin**, enter the administrator card passphrase if you are using module protection. If you are using OCS cards or Softcard protection, enter their passphrase.
11. Select **Next**.
12. Select the HSM's **Credentials**.
13. Select **Save**.

14. Test this credential by selecting the check box next to it and selecting **Check Status**.

    A green check mark appears.