

HTTPS Configuration for Table Reader Mappings in Streaming Mode for PowerExchange for SAP NetWeaver

Abstract

This article explains how to configure HTTPS streaming with Informatica Developer when you read data from SAP tables.

Supported Versions

- PowerExchange for SAP NetWeaver 10.4.0

Table of Contents

Overview	2
Prerequisites	3
Downloading and Installing OpenSSL	3
Downloading and Installing the SAPGENPSE Cryptography Tool	3
Configuring HTTPS on the Data Integration Service Machine.	3
Creating a Certificate.	3
Converting an OpenSSL Certificate to PSE Format Using the SAPGENPSE Tool.	5
Configuring HTTPS for SAP Connector.	5
Downloading and Installing the SAP Java Connector 3.0 Library Files.	6
Enabling HTTPS in an SAP connection and Specifying Keystore Details.	6
Install the HTTPS Streaming Transport in the SAP System.	7
Troubleshooting.	7
Configuring the SAP Parameters.	7
Setting the Trace Level.	8

Overview

You can configure PowerExchange for SAP NetWeaver to use HTTPS and securely run table reader mappings in streaming mode when you use the Developer tool. When you run a table reader mapping in streaming mode, PowerExchange for SAP NetWeaver starts a secure HTTP session and requests data from SAP. SAP streams data over the HTTPS protocol to the Data Integration Service.

To configure HTTPS streaming for SAP table reader mappings, perform the following tasks:

1. Complete the following prerequisite tasks:
 - a. Configure the Informatica services.
 - b. Install the Developer Tool.
 - c. Download and Install OpenSSL.
 - d. Download and Install the SAPGENPSE Cryptography Tool.
 - e. Configure HTTPS on the Data Integration Service Machine.
2. Create an SAP connection and specify the keystore file path, keystore password, and private key password.
3. Install the HTTPS streaming transport in the SAP system.

Prerequisites

Before you configure HTTPS, make sure that you perform the PowerExchange for SAP NetWeaver configuration tasks.

1. Download and install OpenSSL on a local directory on the Data Integration Service machine.
2. Download and install the SAPGENPSE tool on a local directory on the Data Integration Service machine.

Downloading and Installing OpenSSL

Download and install OpenSSL to a local directory on the Data Integration Service machine.

1. Download the OpenSSL binaries from the following URL:
<https://www.openssl.org/community/binaries.html>
2. Click the following link for a Windows 32-bit operating system:
<https://slproweb.com/products/Win32OpenSSL.html>
Click the following link for a Windows 64-bit operating system:
https://slproweb.com/download/Win64OpenSSL_Light-1_0_2d.exe
3. Run the .exe file and follow the steps in the Install wizard.

After the installation is complete, the `openssl.exe`, `ssleay32.dll`, `libeay32.dll`, and `openssl.cfg` files get copied in the OpenSSL installation directory on the machine.

Downloading and Installing the SAPGENPSE Cryptography Tool

Download the SAPGENPSE Cryptography tool that is part of the SAP Cryptographic Library in the SAP Service Marketplace and install it.

1. On the Data Integration Service machine, download the latest available patch for the SAPGENPSE tool based on the operating system.
2. At the command prompt, navigate to the directory that contains the `SAPCAR.EXE` file and the `SAPCRYPTOLIB_*.SAR` file.
3. Extract the SAR file. For example, enter the following command at the command prompt:
`sapcar.exe -xvf SAPCRYPTOLIB_39-10010895.SAR`

The SAPGENPSE files are extracted to the `nt-x86_64` directory within the current directory.

Configuring HTTPS on the Data Integration Service Machine

1. Create a certificate using OpenSSL and JAVA KeyTool.
2. Convert the OpenSSL PKCS#12 certificate to the SAP-specific PSE format using the SAPGENPSE cryptographic tool.

Creating a Certificate

Create a certificate using OpenSSL and JAVA KeyTool.

1. Set the `OPENSSL_CONF` variable to the absolute path to the `openssl.cfg` file. For example, at the command prompt, enter the following command:
`set OPENSSL_CONF= C:\OpenSSL-Win64\bin\openssl.cfg`
2. Navigate to the following directory:
`<openssl installation directory>\bin.`

Verify that the `openssl.exe` file is available in this directory.

3. To generate a 2048-bit RSA private key, enter the following command:
`openssl.exe req -new -newkey rsa:2048 -sha1 -keyout <rsa_key_name>.key -out <rsa_key_name>.csr`
4. When prompted, enter the following values:
 - Private key password. Private key password is also known as PEM pass phrase. Enter a phrase that you want to use to encrypt the secret key. Re-enter the password for verification.
Note: Make a note of the PEM password. You need to specify this value in the following steps.
 - Two letter code for country name.
 - State or province name.
 - Locality name. For example, you can enter the name of your city.
 - Organization name.
 - Organization unit name. For example, the business unit in your organization.
 - Common name (CN). Mandatory. Enter the fully qualified host name of the Data Integration Service machine.
 - Email address.
5. Optionally, enter the following attributes you want to send along with the certificate request:
 - Challenge password. Enter a string, which is embedded in the CSR and is shared between you and the SSL issuer. If you ever need to re-install your certificate for any reason, you will be required to enter that password for authentication.
 - Optional company name.**Note:** A new RSA private key of 2048-bit size is created. The `<rsa_key_name>.key` and `<rsa_key_name>.csr` files are generated in the current location.
6. To generate a self-signed key using the RSA private key, enter the following commands:
`openssl x509 -req -days 11499 -in <rsa_key_name>.csr -signkey
<rsa_key_name>.key -out <certificate_name>.crt`
7. When prompted, enter the PEM pass phrase for the RSA private key. This is the same password that you entered in step 4.
The `<certificate_name>.crt` file is generated in the current location.
8. Concatenate the contents of the `<certificate_name>.crt` file and the `<rsa_key_name>.key` file to a `.pem` file.
 - a. Open the `<certificate_name>.crt` file and the `<rsa_key_name>.key` files in a Text editor.
 - b. Create a new file and save it as `<PEM file name>.pem`.
 - c. Copy the contents of the `<certificate_name>.crt` file and paste it in the `.pem` file. Copy text beginning from `-----BEGIN CERTIFICATE-----` to `-----END CERTIFICATE-----`.
 - d. Copy the contents of the `<rsa_key_name>.key` file and append it to the existing contents of the `.pem` file. Copy text beginning from `-----BEGIN RSA PRIVATE KEY-----` to `-----END RSA PRIVATE KEY-----`.
 - e. Save the `<PEM file name>.pem` file.
9. To create a PKCS#12 certificate, enter the following command at the command prompt:
`openssl pkcs12 -export -in <PEM file name>.pem -out <p12 file name>.p12 - name <domain name>`
10. When prompted, enter the following details:
 - PEM pass phrase for the `.pem` file. This is the same password that you entered in step 4.

- Export password that will be used to protect the P12 file. Re-enter the password for verification.
 - Note:** Make a note of this export password for the P12 file. You need to specify this value in some of the following steps and while creating the SAP Table connection in Developer tool. The <p12 file name>.p12 file is generated in the current location.
11. To create a Java keystore file, enter the following command:


```
keytool -v -importkeystore -srckeystore <p12 file name>.p12 -srcstoretype PKCS12 -destkeystore <JKS file name>.jks -deststoretype JKS -srcalias <unique alias associated with the source keystore> -destalias <destination alias>
```
 12. When prompted, enter the following details:
 - Password for the destination keystore, the JKS file.
 - Note:** Make a note of this password. You need to specify this password while creating the SAP Table connection in Developer tool.
 - Password for the source keystore, the P12 file. Enter the Export password you specified for the P12 file in step 10.
 - The <JKS file name>.jks file is generated in the current location.
 - Note:** While enabling HTTPS in an SAP Table connection, you must specify the name and location of this keystore file. You must also specify the destination keystore password as the Keystore Password and the source keystore password as the Private Key Password.

Converting an OpenSSL Certificate to PSE Format Using the SAPGENPSE Tool

Convert the OpenSSL certificate to SAP specific format (PSE) using the SAPGENPSE cryptographic tool.

1. At the command prompt, navigate to the <sapgenpse extraction directory> directory. Verify that the sapgenpse.exe file is available in this location. For example, navigate to the c:\sapgenpse\nt-x86_64 directory.
2. To generate a PSE file, enter the following command:


```
sapgenpse import_p12 -p <directory where you want to generate the PSE file>\<PSE file name>.pse <path to the P12 certificate file>\<P12 file name>.p12
```
3. When prompted, enter the following details:
 - Password for the P12 file. Enter the Export password that you specified for the P12 file.
 - Personal identification number (PIN) to protect the PSE. Re-enter the PIN for verification. The <PSE file name>.pse file is generated in the specified directory.
4. To generate the certificate based on the PSE format, enter the following command:


```
sapgenpse export_own_cert -p <path to the PSE file>\<PSE file name>.pse -o <certificate name>.crt
```
5. When prompted, enter the PSE PIN number you specified in step 3.

The <certificate name>.crt file is generated in the current location. You must import this certificate file to the SAP system.

Configuring HTTPS for SAP Connector

1. Download and install the SAP Java Connector 3.0 library files.
2. Enable HTTPS in the SAP connection and specify the keystore details.

Downloading and Installing the SAP Java Connector 3.0 Library Files

The Developer tool requires the SAP Java Connector 3.0 (SAP JCo 3.0) library files to work with SAP connections and data objects.

Download the SAP JCo 3.0 files from the SAP Service Marketplace:

<http://service.sap.com/connectors>

If you have problems downloading the SAP JCo 3.0 files from the SAP web site, contact SAP Customer Support.

Download the SAP JCo 3.0 files to the machine that hosts the Developer tool. Extract the SAP JCo 3.0 files and copy the files to the following directories:

File	Directory
sapjco3.jar	<Informatica installation directory>\clients\DeveloperClient\plugins \com.infa.adapter.sap.jco_<version>\lib
sapjco3.dll	<Informatica installation directory>\clients\DeveloperClient\bin

Note: Because the Developer tool is installed on a 64-bit machine, you must use the 64-bit JCo libraries.

If you do not download and install the SAP JCo 3.0 files, the Developer tool displays the following error message when you create SAP connections and data objects:

SAPJCo library files might not be installed. Install the SAPJCo library files and try again.

To successfully test an SAP connection in Informatica Administrator, download the SAP JCo 3.0 files to the machine that hosts the master gateway node. Extract the SAP JCo 3.0 files and copy the files to the following directories on the machine that hosts the master gateway node:

Operating System	File	Directory
AIX 64-bit, Linux64-X86, Linux Itanium 64-bit, Linux-X86	sapjco3.jar	<Informatica installation directory>/services/shared/jars/ thirdparty
AIX 64-bit, Linux64-X86, Linux Itanium 64-bit, Linux-X86	libsapjco3.so	<Informatica installation directory>/services/shared/bin
Windows EM64T, Windows 32-bit	sapjco3.dll	<Informatica installation directory>/services/shared/bin
Windows EM64T, Windows 32-bit	sapjco3.jar	<Informatica installation directory>/services/shared/jars/ thirdparty

Enabling HTTPS in an SAP connection and Specifying Keystore Details

To enable the Data Integration Service to connect to SAP through HTTPS, you must enable HTTPS and specify the keystore details when you configure an SAP Table connection.

1. Click **Manage > Connections**.
2. Click **Actions > New > Connection**.
The **New Connection** page appears.
3. Scroll down and click **SAP** connection type.

4. Click **OK**.
5. Enter a name for the SAP Table Connection.
6. Enter the ID for the connection.
7. Enter an SAP user name with the appropriate user authorization.
8. Enter the password for the user.
9. Select **Application** or **Load Balancing** connection type.
10. Enter the host name of the environment where you want to run the tasks.
11. Enter the system number of the system.
12. Enter the client number.
13. Click **Next**.
14. Select the **Use HTTPS** check box.
15. Enter the Key store file path.
Note: Key store file path is the path to the keystore file that contains the private or public key pairs and associated certificates.
16. Enter the Key store password.
Note: Key store password is the password for the keystore file.
17. Enter the Private key password.
Note: Private key password is the password to decrypt the private key file.
18. Click **Finish**.

Install the HTTPS Streaming Transport in the SAP System

To use HTTPS streaming for SAP table reader mappings, you must install the applicable transports in the SAP system.

Install the following transports in the SAP system:

- TBL_READ_V2_R900086.DU5 (Data file)
- TBL_READ_V2_K900086.DU5 (Cofile)
- DU5K900086(Transport request)

Troubleshooting

You can configure SAP parameters and set the trace level to troubleshoot errors in PowerExchange for SAP NetWeaver Connector.

Configuring the SAP Parameters

On the SAP system, you can set the SAP parameters.

Verify that you set the following SAP parameters:

- icm/server_port
- ssl/ssl_lib
- sec/libsapsecu
- ssf/ssfapi_lib
- ssf/name

- icm/HTTPS/verify_client
- ssl/client_pse
- wdisp/ssl_encrypt

Setting the Trace Level

1. Log on to SAP and go to the SMICM transaction.
2. Select Go to > Trace Level > Set.
3. Enter **3** and press Enter to view detailed logs.

Author