Keystore

On this page

Certificates and key pairs are stored in one keystore per tenant, referred to also as tenant keystore.

Keystore Usage

A keystore is used to secure message exchange both at transport level and at message level.

Transport-level security (HTTPS outbound connections from the SAP Integration Suite tenant to a remote system)

- Supporting client certificate authentication

  You can protect HTTP outbound connections by specifying client certificate authentication when configuring the related receiver adapter. If you do that, the receiver system authenticates the tenant (the client) based on a client certificate.

  To make this authentication option work, the tenant keystore needs to contain a *client certificate* which is a signed key pair containing a private and a public key.

  During the TLS handshake, one of the key pairs whose certificate chain is trusted by the server is selected for the TLS communication. If the server does not have a certificate of an appropriate certification authority (CA) in its trust store, the communication fails because the server cannot authenticate the client. If the server trusts several key pairs, one key pair is chosen at random for the connection.

  If you want to avoid random selection, you can specify an alias of a key pair entry in the related receiver adapter, so that only this specific key pair can be used in the TLS communication (use the Private Key Alias parameter for this purpose). If the keystore contains only one key pair or the server only trusts one key pair, this measure is not necessary. In some cases it is necessary to adapt the chain of the key pair. For example, if the chain of the key pair contains only the public certificate and the server contains only the root CA certificate, then you need to add the intermediate certificate to the chain of the key pair.

  More information: Client Certificate Authentication (Outbound)

- Enabling the tenant to establish a trust relationship to the receiver system

  The SAP Integration Suite tenant also needs to establish a trust relationship to the receiver in such a way that the receiver can authenticate itself against SAP Integration Suite . In this case, authentication is accomplished based on a *server certificate* (as the receiver plays the role of a server). As prerequisite for this security measure, the tenant keystore needs to contain a (server) *root certificate* that is also trusted by the receiver.

Even in case you specify basic authentication when configuring the related receiver adapter, you need to make sure that the tenant keystore contains a valid root certificate that is also trusted by the receiver.

Message-level security

The keystore also contains the public and private keys used for message-level security (signing and encryption). Public keys are used in the signature verification steps (XML Signature, PKCS#7/CMS Signature Verification, WebService Security) and in the encryption steps (PKCS#7/CMS, WebService Security) of integration flows. Private keys are used in the signature creation steps (XML Signature, PKCS#7/CMS Signature, WebService Security) and decryption steps (PKCS#7/CMS, WebService Security) of integration flows. In these steps, the relevant keystore entries are referenced by their aliases. We recommend using different keys for message- and transport-level security. Keep in mind that the expiration date of the certificates is not checked in the encryption/decryption steps and in the signing steps.

Note that certain adapters (like the SOAP 1.x and the AS2 adapter) support options to sign/verify and encrypt/decrypt message content based on the Web Services Security (WS-Security) standard. To support such scenarios, the tenant keystore also needs to contain certain X.509 keys.

Keystore Content

There are the following entry types:

- Key Pair entry

  Consists of a private key and its X.509 certificate chain.

  All private keys of a keystore are encrypted with the same password. This password is also used as the keystore password (for checking the integrity of the keystore). The keystore is never stored in the same database as the encrypted/signed application data. The password is stored in a separate database.

  The certificate chain typically consists of the public key certificate and the intermediate certification authority (CA) certificate with which the signature of the public key certificate can be verified.

- Certificate entry

  In many cases this is an X.509 root certificate.

Keystore Management

A tenant keystore contains both entries owned by the tenant administrator (tenant owner) and entries owned by SAP. SAP-owned entries cannot be changed or deleted by the tenant administrator and entries owned by the tenant administrator cannot be changed or deleted by SAP.

More information: Managing Keystore Entries

Note

There is a dedicated naming convention for keystore aliases to indicate the owner of the keystore entry:

Alias names of SAP-owned entries start with `sap_` or are `hcicertificate`, `hcicertificate1`, `hcimsgcertificate`.

SAP Integration Suite does not verify the signatures of the certificates during the upload. Therefore, the user who uploads the certificates is responsible for ensuring that the signatures of the certificates are verified before the upload. Note that root certificates in particular must always be verified manually in any case.

Keystore Entries Preinstalled by SAP

When a customer starts using Cloud Integration, certain keystore entries have already been made available by SAP.

Keystore Entry	Purpose
One signed Key Pair entry with the alias sap_cloudintegrationcertificate.	• For outbound client certificate authentication • In the Cloud Foundry environment, for inbound client certificate authentication (required to enable internal communication between the involved BTP microservices)
Certain Certificate entries which are also owned by SAP.	These are root certificates that the customer can use to set up connections with other SAP cloud systems such like SAP SuccessFactors, for example.