domibusConnectorClient-4.0-RELEASE -

Administration- and User-Guide

Table of contents

[Table of contents 2](#_Toc511909880)

[1. Introduction 4](#_Toc511909881)

[1.1. Scope and Objective of this document 4](#_Toc511909882)

[1.2. The domibusConnector as a web application 4](#_Toc511909883)

[1.3. The domibusConnectorClient 4](#_Toc511909884)

[1.4. The gateway 4](#_Toc511909885)

[1.5. The domibus-connector-plugin 5](#_Toc511909886)

[2. Preconditions and technical requirements 6](#_Toc511909887)

[2.1. The domibusConnector distribution package 6](#_Toc511909888)

[2.2. Supported operating systems 7](#_Toc511909889)

[2.3. Java Runtime 7](#_Toc511909890)

[2.4. Database 7](#_Toc511909891)

[2.5. Web container 7](#_Toc511909892)

[2.6. Internet connection 8](#_Toc511909893)

[2.7. Technical specifications 8](#_Toc511909894)

[3. Database Installation 9](#_Toc511909895)

[3.1. Supported Database vendors 9](#_Toc511909896)

[3.2. New Database / Fresh Installation 9](#_Toc511909897)

[3.2.1. Using the scripts 9](#_Toc511909898)

[3.2.2. Using liquibase 9](#_Toc511909899)

[3.3. Database Upgrade 3.5 to 4.0 9](#_Toc511909900)

[3.3.1. Using the script 10](#_Toc511909901)

[3.3.2. Using liquibase 10](#_Toc511909902)

[3.4. Upgrade with Liquibase 10](#_Toc511909903)

[4. Configuration properties 12](#_Toc511909904)

[5. Certificate, Key-Stores and Truststores 13](#_Toc511909905)

[5.1. Connector Backend Key Store 14](#_Toc511909906)

[5.2. Connector Key Store 14](#_Toc511909907)

[5.3. Evidence Key Store 14](#_Toc511909908)

[5.4. Connector truststore 15](#_Toc511909909)

[5.5. TLS Key Store (System Key Store) 15](#_Toc511909910)

[6. Deployment 16](#_Toc511909911)

[7. Import of p-modes 17](#_Toc511909912)

[7.1. Import of a p-mode file 17](#_Toc511909913)

[7.2. DataTables 17](#_Toc511909914)

[8. Backend configuration 19](#_Toc511909915)

[8.1. Backend types 19](#_Toc511909916)

[8.1.1. Push/pull backend 19](#_Toc511909917)

[8.1.2. Push/push backend 19](#_Toc511909918)

[8.2. Adding the backend client keys to the Connector Backend Key Store 19](#_Toc511909919)

[8.3. Configuring the backend at the database 20](#_Toc511909920)

[8.3.1. DOMIBUS\_CONNECTOR\_BACKEND\_INFO 20](#_Toc511909921)

[8.3.2. DOMIBUS\_CONNECTOR\_BACK\_2\_S 20](#_Toc511909922)

[8.3.3. Example scripts 21](#_Toc511909923)

# Introduction

## Scope and Objective of this document

This document is a technical guide on how the domibusConnectorClient can give support in all its variants is to be integrated, installed and used to connect the domibusConnector web application and your national implementation.

The focus of this document is to give technical guidance on taking the decision on the variant of the domibusConnectorClient to be used to fit the needs of your national environment best. It also describes the functionalities of the domibusConnectorClient and gives an overview of preconditions to be met to integrate the domibusConnectorClient into your environment.

Therefore this guide addresses technical staffs that have knowledge of the national implementation in place and network infrastructure.

This guide document for the domibusConnectorClient only focuses on the client itself.

For more detailed information on the domibusConnector web application, please see the documentation shipped with the domibusConnector-4.0-RELEASE.

## The domibusConnector as a web application

Starting with version 4.0-RELEASE, the domibusConnector is on the technical basis of a web application.

This means, that the domibusConnector itself is a “ready-to-use” software component that only needs to be configured, set-up and deployed in a web container.

Once installed and configured properly, the domibusConnector should run on its own, besides some maintenance.

The domibusConnectorClient otherwise is delivered in different variants to support the connection between the domibusConnector web application and your national environment.

It is advised though, that the domibusConnector web application is already properly installed and configured in your environment before starting with the integration of the domibusConnectorClient.

## The domibusConnectorClient

The domibusConnector web application offers different interfaces that can be used to approach its functionalities. Those interfaces can be used directly, if intended. A detailed description of the interfaces offered by the domibusConnector web application can be found in the documentation shipped with the domibusConnector-4.0-RELEASE.

To close the missing link between your own implementation of the e-CODEX use cases and the services of the domibusConnector, also a domibusConnectoClient has been implemented to support the connection to the domibusConnector.

The domibusConnectorClient is distributed in the following different variants that are described in detail in this document:

* domibusConnectorClient-Standalone: This client replaces the prior versions of the domibusConnector-Standalone. It runs without having other implementations in place with a graphical user interface that interoperates with the file system to receive and send messages from and to the domibusConnector.
* domibusConnectorClientLibrary: This is an integrate able library that can be used to be embedded into an already implemented application. It can also be a basis for new developments as well.
* domibusConnectorClientScheduler: This library is an extension of the domibusConnectorClientLibrary. It enhances the functionality of the library with time triggered jobs that can be configured to run the functionalities of the library automatically triggered.
* domibusConnectorClientWebLib: If your national application is a web application that runs inside of a web service container, the domibusConnectorClientWebLib offers the opportunity, to start a web service itself for the delivery of messages from the domibusConnector. The advantage of this variant is, that no jobs need to be triggered, as the connection between the domibusConnector and the client work as a push/push web service.
* domibusConnectorClient35Library: This library only addresses implementers that had previous versions of the domibusConnector framework prior to 4.0-RELEASE in place. It offers access to the functionalities of the new domibusConnectorClient by using the old interfaces that were in place up to version 3.5.1 of the domibusConnector framework. All of those interfaces are marked as deprecated though.

All those variants, how they work and how they can be installed/integrated are described in separate chapters of this document.

# Preconditions and technical requirements

This chapter describes what requirements have to be fulfilled to use the functionalities of the domibusConnectorClient. It also lists some technical specifications of the domibusConnectorClient to give a more detailed insight.

## Supported operating systems

The domibusConnectorClient is a software product, that was completely implemented using the JAVA programming language.

As JAVA is by definition a platform independent environment, every operating system with a proper JAVA installation should fit the needs of setting up/ integrate the domibusConnectorClient.

During implementation and testing phase of the domibusConnectorClient, it was tested and installed on the following environments:

* Microsoft Windows 7
* Linux
* IBM AIX

## Java Runtime

As the domibusConnectorClient is a JAVA application, it also requires a proper installation of a Java Runtime to be able to run the software.

The recent version 4.0-RELEASE of the domibusConnectorClient was implemented and compiled with an Oracle JDK jdk-8u161. So at least this version or above should be in place to avoid incompatibilities.

## Technical specifications

For your information the main frameworks and technologies the domibusConnectorClient was implemented with is listed here.

* Java 8 (Oracle jdk-8u161)
* Spring framework 4.3.12.RELEASE
* Spring-boot 1.5.8.RELEASE
* Apache CXF 3.2.1
* Apache Maven 3

## The domibusConnectorClient distribution package

To get started, you first need to download and extract the distribution package.

The domibusConnectorClient distribution package is placed on the e-CODEX Nexus repository server at:

https://secure.e-codex.eu/nexus/content/groups/public/eu/domibus/connector/domibusConnectorDistribution/4.0-RELEASE/

* domibusConnectorDistribution-4.0-RELEASE.war
* domibusConnectorDistribution-4.0-RELEASE.zip

are the most important ones.

Whereas the “domibusConnectorDistribution-4.0-RELEASE.war” is only the deployable WAR package and intended to be used to simply upgrade already set up domibusConnector installations, the “domibusConnectorDistribution-4.0-RELEASE.zip” is of interest for setting up a new domibusConnector.

Once downloaded and extracted it has the following structure:

|  |  |
| --- | --- |
| **File/directory** | **Description** |
| Webapp (directory) | This directory contains the application itself distributed as “domibusConnector-4.0-RELEASE.war” |
| Documentation/database-scripts (directory) | This directory contains all necessary database scripts to set up the database for the domibusConnector. The scripts are prepared for the database vendors MySQL and Oracle.  For more details see chapter …. Database Setup |
| Documentation/databaseInitializer (directory) | Contains the “domibusConnectorDatabaseInitializer.jar” which is a helper application to set up the database.  For more details see chapter …. Database Setup |
| Documentation/properties (directory) | The “properties” folder contains example properties that show how to configure the domibusConnector. Also empty properties are there that only have the keys set without values. The log4j configuration is also contained as an example. |
| domibusConnector\_Monitoring\_Interfaces.pdf | A document that describes what monitoring interfaces the domibusConnector offers and how to approach them. |
| domibusConnector-Technical-documentation-and-UserGuide.pdf | This document merges the documentation for the domibusConnector for administrators and users. This document covers all distributions of the domibusConnector. |

# Certificate, Key-Stores and Truststores

To ensure the highest reasonable level of security, the domibusConnector uses several certificates for different purposes:

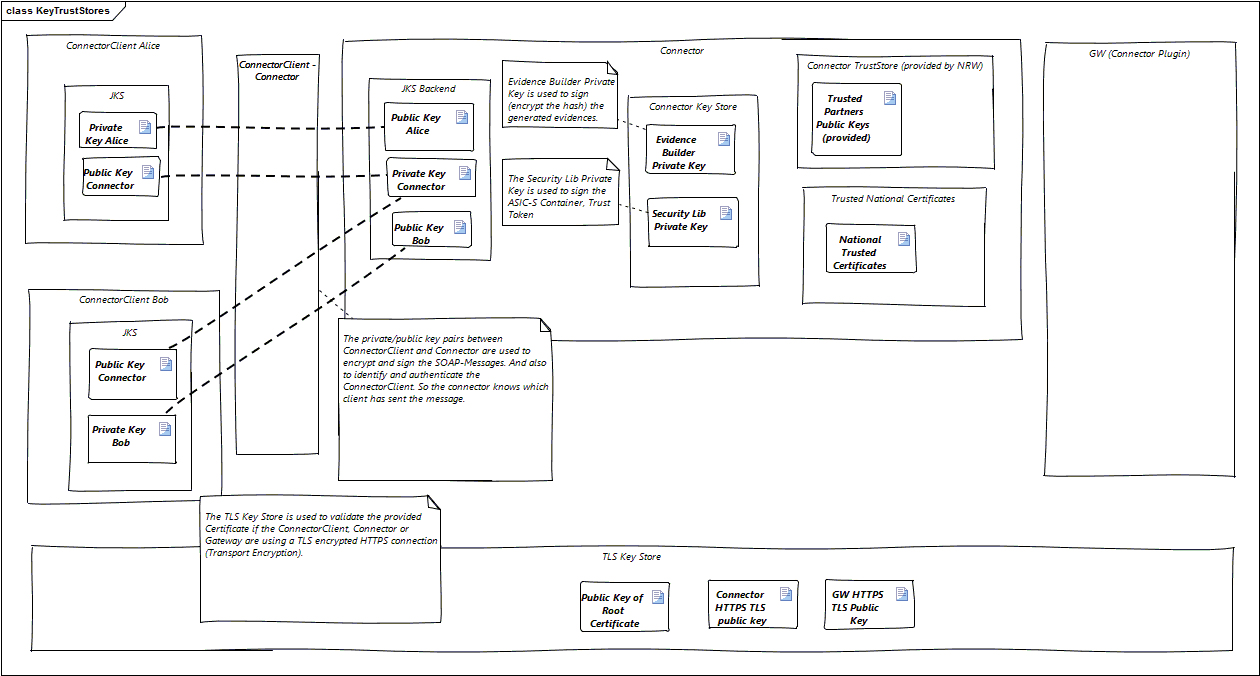
* Signing and Encrypting SOAP messages between the backend client and the Connector.
* Establishing Transport Security (TLS) between the backend client and the Connector.
* Singing and Encrypting SOAP messages between the Connector and the Gateway.
* Establishing Transport Security (TLS) between the Connector and the Gateway.
* Validating the signature of the main document (mostly a PDF) of the message (if configured).
* Validating the signature of the secure container (ASIC-S) received with incoming messages.
* Signing the secure container (ASIC-S) that is created by the Connector.
* Signing the ETSI-REM evidences.

In most of those cases the same certificate can be used, though we do not recommend that. For higher security it is more efficient to use different certificates.

In case of multiple backend clients that connect to the domibusConnector it is required to have an own certificate per backend.

Each certificate holds a private key, which should always stay inside your organization, and a public key.

The following graphic shows an example of what key is used at what point. It also shows the purpose of the different keys:



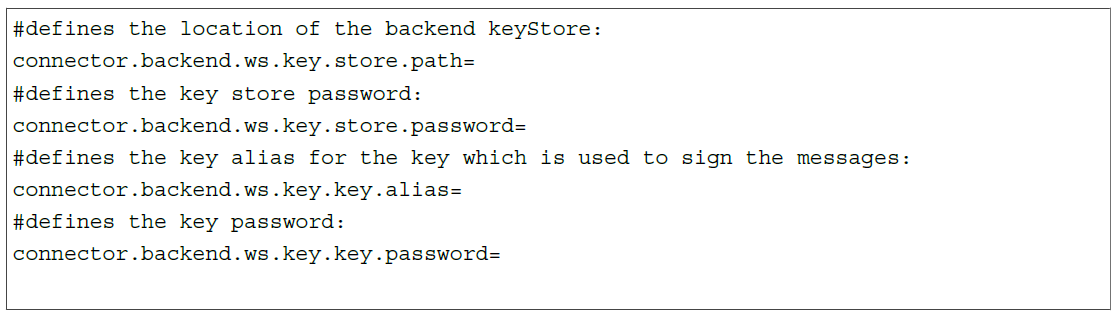
This guide focuses on the keys and stores that are required for a proper installation of the domibusConnector. Other keys and stores, for the backend client(s) for example, are explained in more detail in other documentations.

## Connector Backend Key Store

The connector backend key store holds the private key of the connector which is used to decrypt

and sign the messages which are sent to the connector clients. It also contains all public keys of all the backend clients to verify the signature of received messages. The common name (CN) of the certificate is also used to identify the backend clients name.

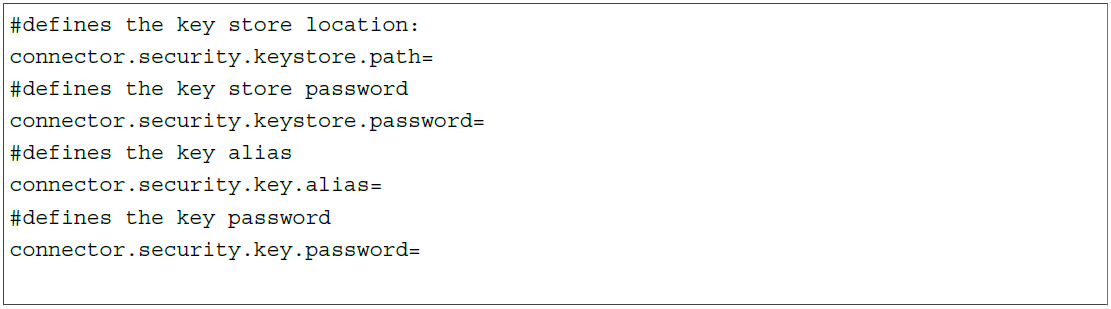
The configuration properties are:



## Connector Key Store

The connector key store holds the private key for signing the ASIC-S container.

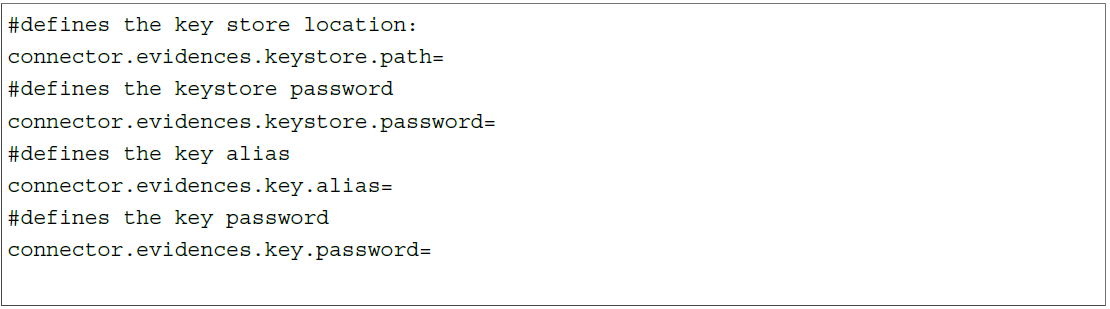
The configuration properties are:



## Evidence Key Store

The evidence key store holds the private key for signing the generated ETSI-REM evidences. This private key and key store can be the same for signing the ASIC-S container (Connector Key Store).

The configuration properties are:

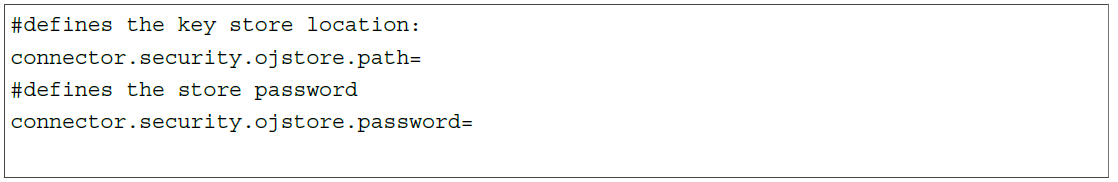


## Connector truststore

This truststore only holds public keys. The connector truststore (in configuration management called the “connectorstore”) is provided by the configuration management of the project and contains the public keys of the e-CODEX partners. They are used to verify the signature of the ASIC-S container received from an e-CODEX partner.

Additionally, if your organization uses signed documents (mostly PDF) as the main content of the message when sending a message to an e-CODEX partner, the public key of the certificate with which the document was signed with should be imported into this truststore. The security library uses this public key to verify the signature of the document then (configured as SIGNATURE\_BASED).

The configuration properties are:



# Configuration properties

To give the domibusConnector the missing links about your environment, some properties have to be set in a property file.

Usually this is called “connector.properties”.

Also, the possibility is given to adopt the logging configuration. This gives the opportunity to control where logs are written at and what to log.

Example properties and an empty property file, as well as an example for logging configuration can be found in the distribution package at “documentation/properties”.

The properties in those file are all well described on what is expected there.

The variants on how to include the properties into your web server environment is dependent on what product you have in place.

For the web server products Apache Tomcat and BEA Weblogic this is described exemplarily in the Chapter [Deployment](#_Deployment).

# domibusConnectorClientLibrary

# domibusConnectorClientScheduler

# domibusConnectorClient-Standalone

# domibusConnectorClient35Library