

domibus-connector-plugin InstallationGuide

Table of contents

TABLE OF CONTENTS	2
LIST OF FIGURES	3
1. INTRODUCTION	4
1.1. SCOPE AND OBJECTIVE OF THIS DOCUMENT	4
1.2. THE DOMIBUS-CONNECTOR-PLUGIN	4
1.3. PLUGIN MANAGEMENT OF THE DOMIBUS GATEWAY	4
1.4. PRECONDITION	4
2. THE DOMIBUS-CONNECTOR-PLUGIN-DISTRIBUTION PACKAGE	5
2.1. GENERAL STRUCTURE	5
2.2. STRUCTURE OF EACH DOMIBUS-CONNECTOR-PLUGIN VERSION	6
3. THE WEB SERVICES OF THE DOMIBUS-CONNECTOR-PLUGIN	7
3.1. DOMIBUSCONNECTORGATEWAYSUBMISSIONWEBSERVICE	7
3.2. DOMIBUSCONNECTORGATEWAYDELIVERYWEBSERVICE	8
3.3. THE DOMIBUSCONNECTORAPI'S TRANSITION MODEL	8
4. SECURITY OF THE DOMIBUS-CONNECTOR-PLUGIN WEBSERVICE	9
5. INSTALLATION OF THE DOMIBUS-CONNECTOR-PLUGIN	10
5.1. INSTALLING THE DOMIBUS-CONNECTOR-PLUGIN.JAR	10
5.2. INSTALLING THE DOMIBUS-CONNECTOR-PLUGIN.XML	10
5.3. PROPERTIES FOR THE DOMIBUS-CONNECTOR-PLUGIN	10
6. CHECK AFTER INSTALLATION	12

List of Figures

Figure 1-Structure of the distribution package	5
Figure 2-Architecture of the web service bridge.....	7
Figure 3-Transition model of the domibusConnectorAPI	8
Figure 4-domibus-connector-plugin.properties.....	11
Figure 5-services list at the gateway	12

1. Introduction

1.1. Scope and Objective of this document

This document is a technical guide to install and configure the domibus-connector-plugin-4.1.0-RELEASE. It can be used as a “go-through” installation guide. Readers should be able to install and configure the domibus-connector-plugin in their DOMIBUS gateway environments without previously built know-how about the software.

The target audience of this document are technical personal or administrators of a DOMIBUS gateway that is intended to be connected to a domibusConnector-4.2.0-RELEASE.

A detailed knowledge of the own network structures and environment is a precondition.

The structure of this guide is built so that every step can be taken as listed in the document. That means all preconditions for a chapter should be given by the previous chapters.

As an InstallationGuide this document does not focus on features and functionalities on the usage of the domibus-connector-plugin.

1.2. The domibus-connector-plugin

To have a connection between the domibusConnector and the DOMIBUS Gateway, a plugin has been developed that can easily be installed on the gateway.

This plugin implements all interfaces that enable message transmission between the connector and the gateway. It is available as a distribution package of e-CODEX on the Nexus repository server:

<https://secure.e-codex.eu/nexus/content/groups/public/eu/domibus/connector/plugin/domibus-connector-plugin-Distribution/>

1.3. Plugin management of the DOMIBUS gateway

This document solely focuses on the domibus-connector-plugin to enable connection between the DOMIBUS gateway and the domibusConnector as a backend of the gateway.

The DOMIBUS gateway offers also other plugins that can be placed to connector other or additional backend solutions to the gateway.

More information on distributed default plugins and plugin management of the DOMIBUS gateway can be found at the CEF site for DOMIBUS at

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus>

1.4. Precondition

As a precondition for installing the domibus-connector-plugin a DOMIBUS gateway must be set up and configured already. The plugin is only to be installed in already existing DOMIBUS gateway environments of version 4.0 and above.

2. The domibus-connector-plugin-Distribution package

This chapter describes the contents of the domibus-connector-plugin-Distribution ZIP package as downloadable at <https://secure.e-codex.eu/nexus/content/groups/public/eu/domibus/connector/plugin/domibus-connector-plugin-Distribution/>.

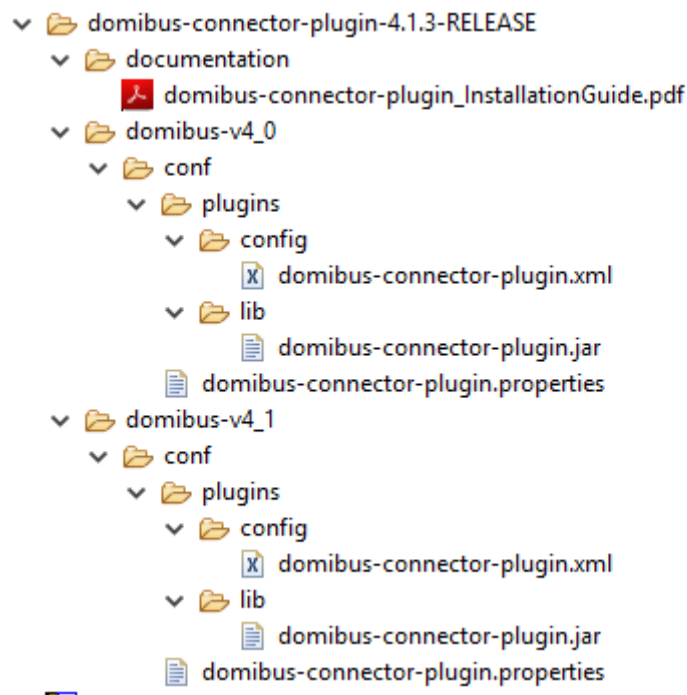


Figure 1-Structure of the distribution package

2.1. General structure

The domibus-connector-plugin-Distribution ZIP contains this document within the “documentation” folder.

There are two more folders: “domibus-v4_0” and “domibus-v4_1”.

The difference between those two versions of the domibus-connector-plugin lies in the used domibus-api integrated. Each DOMIBUS version also offers an API which needs to be used for plugin implementors. Since there may be changes within the used domibus-api, the domibus-connector-plugin was implemented with the domibus-api of the DOMIBUS 4.0 and another with the domibus-api of DOMIBUS 4.1.

So the difference on what domibus-connector-plugin to use depends on what version of the DOMIBUS gateway is should be installed in.

2.2. Structure of each domibus-connector-plugin version

Each of the two provided domibus-connector-plugin versions offers this structure:

File/directory	Description
conf/plugins	To have an easier orientation on how to install the domibus-connector-plugin it mostly follows the structure given by a DOMIBUS installation. The “conf” folder usually is given by the variable \${domibus.config.location} that needs to be set for a DOMIBUS installation.
conf/plugins/domibus-connector-plugin.properties	To configure the domibus-connector-plugin some properties have to be set in the DOMIBUS configuration. Those properties are explained in more detail in chapter Properties for the domibus-connector-plugin .
conf/plugins/config/domibus-connector-plugin.xml	This XML file holds the spring configuration of the domibus-connector-plugin. For more details see chapter Installing the domibus-connector-plugin.xml .
conf/plugins/config/domibus-connector-plugin.jar	This is the plugin library that holds the implemented JAVA classes. For more details see chapter Installing the domibus-connector-plugin.jar .

Table 1-contents of each plugin folder

3. The web services of the domibus-connector-plugin

The domibus-connector-plugin on one side, the DOMIBUS gateway side, implements the given interfaces of the “domibus-plugin-api” to communicate with the gateway’s core.

But to build a bridge to the domibusConnector web services are required that transport messages between the gateway and the connector having the correct structure.

The web service bridge between the gateway and the connector is built in PUSH-PUSH mode. This means that either of the two sides requires that the other side first needs to pull waiting messages.

This has the advantage that messages are transported immediately from the domibusConnector to the DOMIBUS gateway and vice versa.

In architectural means this bridge is provided by the domibusConnectorAPI like this

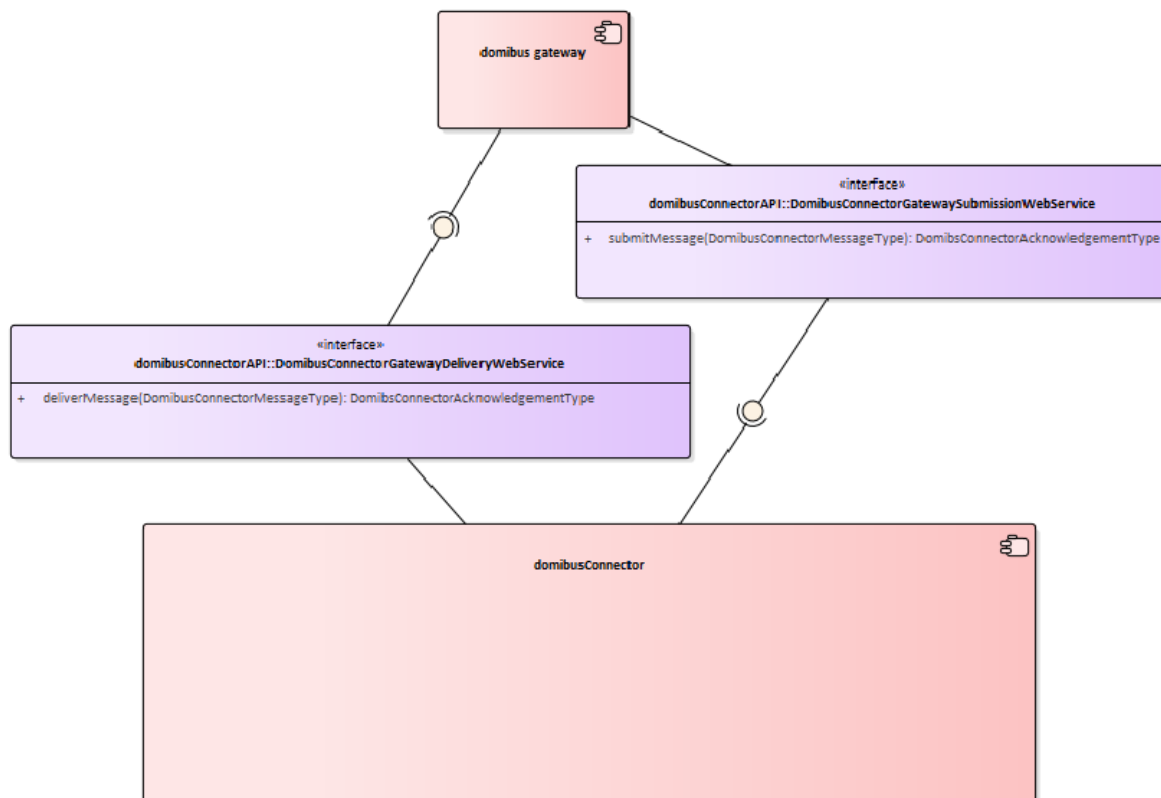


Figure 2-Architecture of the web service bridge

3.1. DomibusConnectorGatewaySubmissionWebService

This web service is placed on the domibus-connector-plugin side. The domibusConnector in this case has the client side. Via this web service the domibusConnector can SUBMIT messages directly through the domibus-connector-plugin to the gateway. Therefore the only method required in this interface is submitMessage.

3.2. DomibusConnectorGatewayDeliveryWebService

This web service is placed on the domibusConnector side. The domibus-connector-plugin in this case has the client side. Via this web service the domibus-connector-plugin can DELIVER messages directly to the domibusConnector. Therefore the only method required in this interface is deliverMessage.

3.3. The domibusConnectorAPI's transition model

To transport messages the proper way, they need to be built following the transition model of the domibusConnectorAPI.

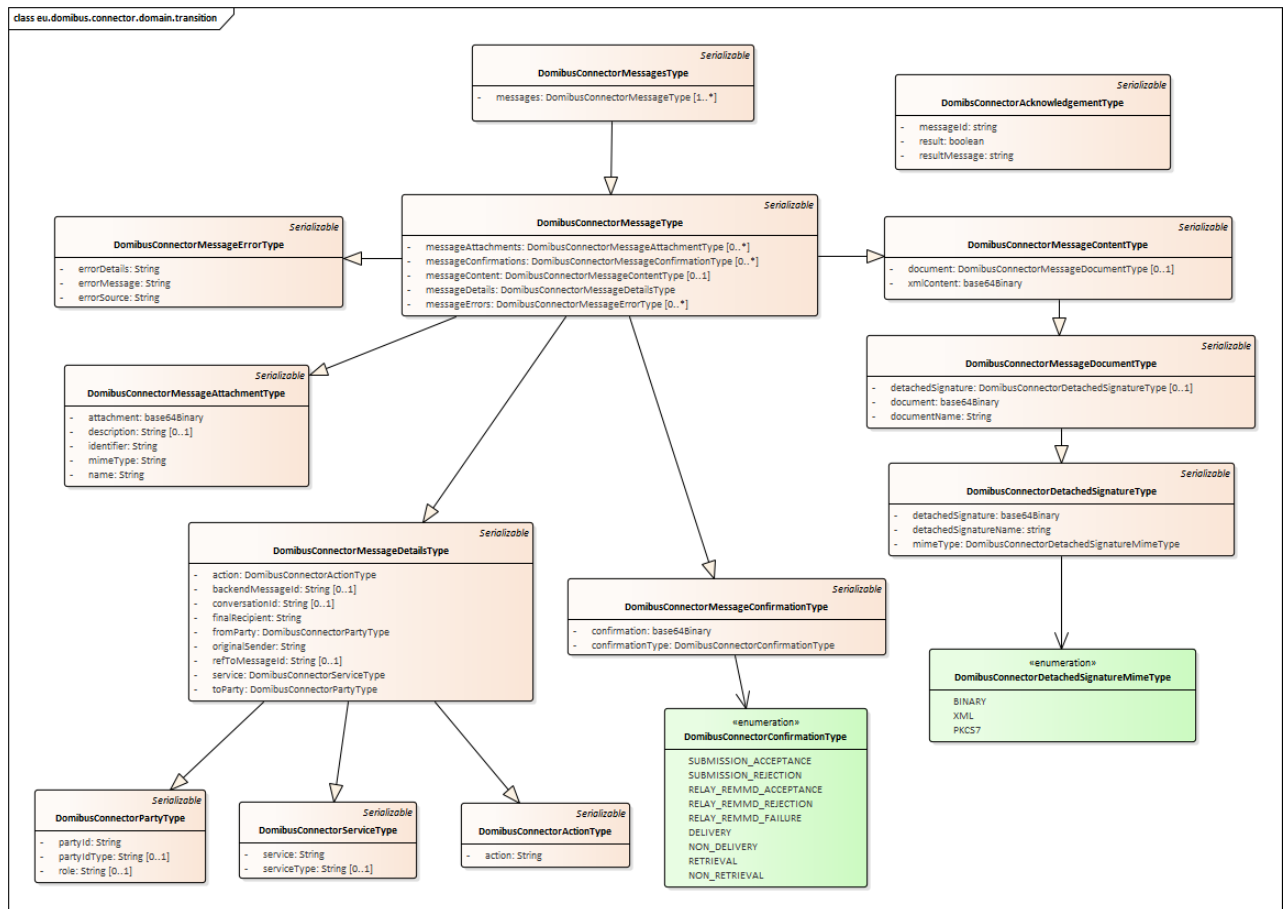


Figure 3-Transition model of the domibusConnectorAPI

4. Security of the domibus-connector-plugin webservice

The e-Codex environment has the obligation to offer the best secure way of transporting messages from one end to the other.

This also includes the transportation between the domibusConnector and the DOMIBUS gateway.

For that purpose the web service bridge between the domibusConnector and the domibus-connector-plugin transports messages following the WS-security standard by OASIS.

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

This means that messages transported between one web service and its web service client are encrypted and signed.

To be able to sign and encrypt messages on one end and validate and decrypt messages on the other certificates are required.

For details on the required certificates, key- and trust stores please refer to the document “e-Codex_key_trust_stores.pdf” distributed with the domibusConnector documentation.

Details on how to configure the required security settings are described in chapter [Properties for the domibus-connector-plugin](#).

5. Installation of the domibus-connector-plugin

The installation of the domibus-connector-plugin consists of 3 steps described in this chapter.

5.1. Installing the domibus-connector-plugin.jar

The domibus-connector-plugin.jar is a JAVA library. This library contains all implementations of the plugin following the “Plugin cookbook” distributed as a documentation for the DOMIBUS gateway. The document can be found at <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus> and is the basis for implementing a custom plugin for the DOMIBUS gateway.

Besides the implemented classes it also includes other libraries the plugin requires:

- domibus-plugin-api JAR in the version of the corresponding gateway.
- domibusConnectorAPI-4.2.0-RELEASE.jar which holds all interfaces and generated classes for the service bridge between the plugin and the domibusConnector.
- Third party libraries.

This domibus-connector-plugin.jar must be copied into the existing DOMIBUS installation at `${domibus.config.location}/plugins/lib`.

If no other backend else than the domibusConnector is used, then all other JAR files inside this folder should be deleted.

5.2. Installing the domibus-connector-plugin.xml

The domibus-connector-plugin.xml is a Spring configuration file that extends the Spring configuration of the installed DOMIBUS gateway. It configures the required beans contained in the plugin JAR and also defines the webservice bridge between the DOMIBUS gateway and the domibusConnector it should connect to.

This domibus-connector-plugin.xml must be copied into the existing DOMIBUS installation at `${domibus.config.location}/plugins/config`.

If no other backend else than the domibusConnector is used, then all other XML files inside this folder should be deleted.

The domibus-connector-plugin.xml does not require any editing and should remain unchanged.

5.3. Properties for the domibus-connector-plugin

To configure the installed environments of domibusConnector and DOMIBUS gateway for the domibus-connector-plugin a few properties are required to be set.

Those properties with explanations as comments can be found in the package in the domibus-connector-plugin.properties.

```
# The plugin is using this webservice publishAddress to deliver messages to the connector
connector.delivery.service.endpoint.address=!!CHANGE ME!! example: http://connector:8080/domibus/services/domibusConnectorDeliveryWebservice

#The File-Path to the keystore holding the certificate with which the plugin signs and decrypts messages from/to the connector.
#The path ideally should be absolute and with a "file:" prefix. Also "\" should be replaced by / or \\
connector.delivery.key-store.file=!!CHANGE ME!! Example: file:C:/<anyPath>/connector.jks
#The Password of the keystore.
connector.delivery.key-store.password=!!CHANGE ME!!

#The alias of the private key with which the plugin signs and decrypts messages from/to the connector.
connector.delivery.private-key.alias=!!CHANGE ME!!
#The Password of the private key.
connector.delivery.private-key.password=!!CHANGE ME!!

#The File-Path to the truststore holding the public certificate of the connector. When receiving a message from the connector,
#the plugin validates the signature of it. Also, when sending a message to the connector, the plugin encrypts the message with
#a public key of the plugin.
#The path ideally should be absolute and with a "file:" prefix. Also "\" should be replaced by / or \\
connector.delivery.trust-store.file=!!CHANGE ME!! Example: file:C:/<anyPath>/connector.jks
#The Password of the truststore.
connector.delivery.trust-store.password=!!CHANGE ME!!

#This is the alias of the public key with which messages sent to the connector are encrypted. The corresponding private key
#must be within the keystore of the connector.
connector.delivery.encrypt-alias=!!CHANGE ME!!
```

Figure 4-domibus-connector-plugin.properties

Those properties need to be adapted and copied into the “domibus.properties” at \${domibus.config.location}.

6. Check after installation

To check if the domibus-connector-plugin is properly installed, the services web interface of the DOMIBUS gateway when started can be called at <http://<yourGatewayAddress>/domibus/services>.

Available SOAP services:

DomibusConnectorGatewaySubmissionWebService <ul style="list-style-type: none"> • submitMessage 	Endpoint address: http://127.0.0.1:8080/domibus/services/domibusConnectorSubmissionWebservice WSDL : http://connector.domibus.eu/ws/gateway/submission/webservice DomibusConnectorGatewaySubmissionWSService Target namespace: http://connector.domibus.eu/ws/gateway/submission/webservice
MSHWebservice	Endpoint address: http://127.0.0.1:8080/domibus/services/msh WSDL : http://receiver.ebms3.domibus.eu/mshService Target namespace: http://receiver.ebms3.domibus.eu/

Figure 5-services list at the gateway

If the DomibusConnectorGatewaySubmissionWebService is listed, the plugin is successfully installed.