# domibusConnector-4.1.0-RELEASE - Administration Guide

# Table of contents

# 1. Introduction

## 1.1. Scope and Objective of this document

This document should give an introduction into administrating the connector and also give a short overview of the internals of the connector.

The document starts with the architectural overview which provides a brief overview of the connector internals. It also covers where the connector should be placed in an e-Codex environment.

Further on the administration guide is split into two parts. In the first part the possibilities of the web user interface are described. The next part covers tasks that need interaction with the connector database.

This guide does not cover things like configuring the database connection, the web application server or the gateway or connector client connections. For this information we refer to the document ”domibusConnector_InstallationGuide” distributed together with this document.

# 2. Architectural Overview

## 2.1. Connector in the e-codex environment

In the e-Codex world the messages are transported as AS4 messages. The payload of this AS4 message are a business document (xml) and an ASIC-S container which contains all the other related documents (pdf, xml, …). Also a trust token is generated which hides the national trust system for the other participants in the e-Codex environment. During the message transport ETSI-REM confirmations are generated. This confirmations or evidences are the proof that a message has been processed by a specific system.

The transportation and handling of AS4 messages are done by an AS4 gateway. In the e-Codex world usually the DOMIBUS gateway is used for that purpose.

The ASIC-S container handling, Trust-Token generation and the confirmation evidence handling are done by the connector. So the domibusConnector is a part of the toolchain to connect the national system with the international e-Codex world.

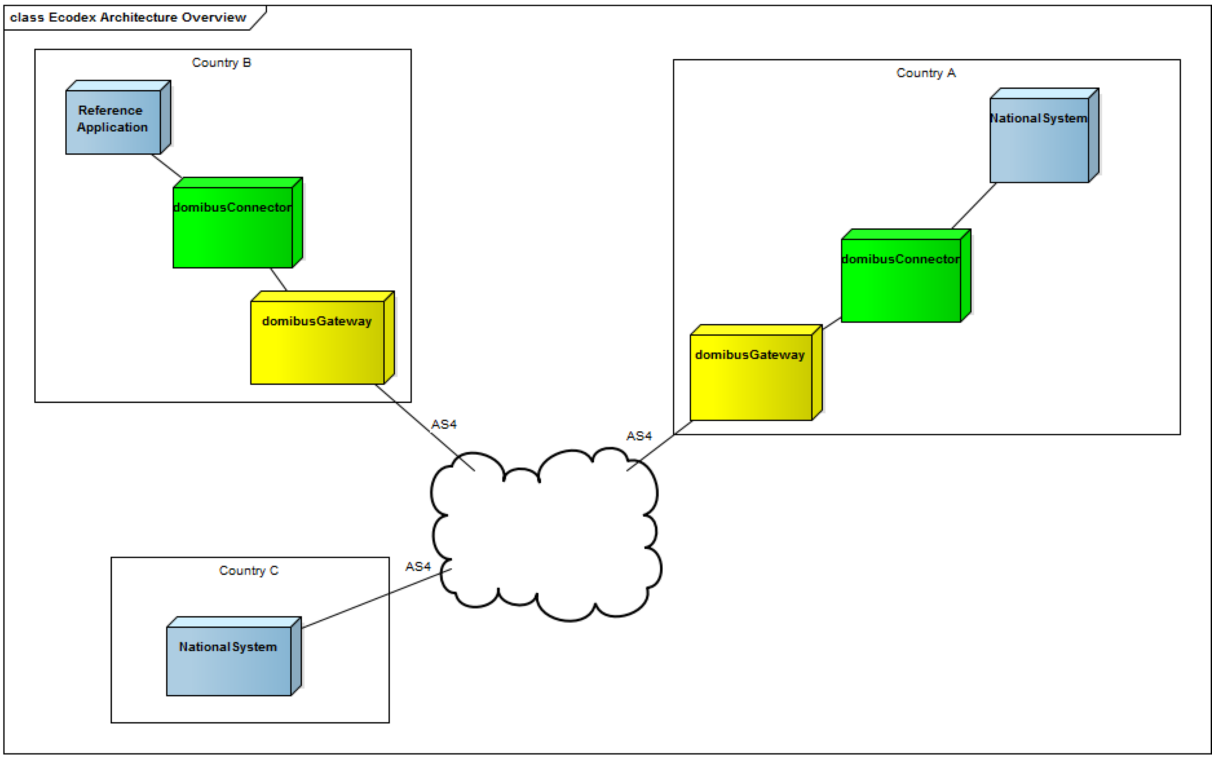


*Figure 1: e-CODEX environment overview*

## 2.2. Connector in the national environment

The connector is a web application. The connector has two sides. A backend side which is a national solution or a domibusConnectorClient, and the gateway side. Messages are always delivered from the gateway to the connector's backend. In the other direction the messages are submitted.

As Figure 2 shows the connector clients may use the domibusConnectorClient[1] to establish a connection to the connector or they can directly use the provided web service interfaces of the domibusConnector. The domibusConnector supports multiple clients.[2]

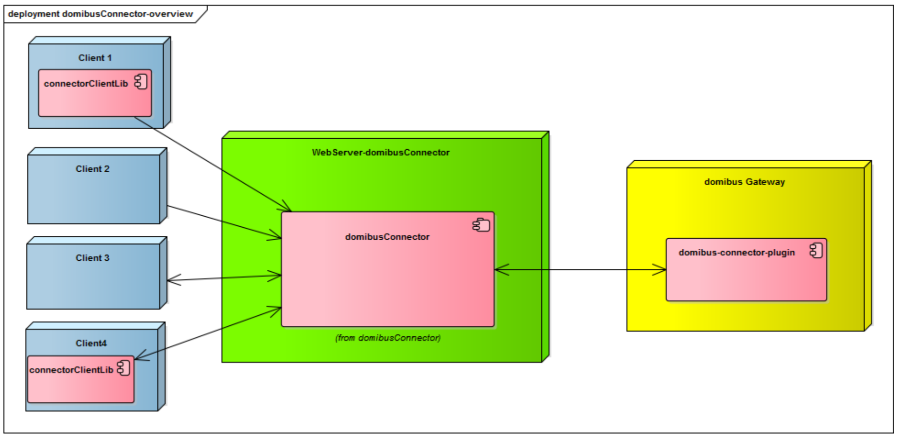


***Figure 2: domibusConnector in the national environment***

The Client 1 and Client 4 in this example use the domibusConnectorClient-library to establish the connection to the domibusConnector. The domibusConnectorClient-library is included as a jar dependency inside the client's implementation. For further information refer to the distributed documentation of the domibusConnectorClient.

The Client 2 and Client 3 of this example address the web service interfaces of the domibusConnector directly, so there is no need for including any domibusConnectorClient jar dependency.

A domibus-connector-plugin installed at the DOMIBUS gateway is responsible for the communication between the domibusConnector and the gateway. This plugin is also available as a distribution on the Nexus and also contains documentation on usage and installation.

### 2.2.1. domibusConnector interfaces

The domibusConnector provides web service interfaces on both sides. These interfaces are specified using the WSDL-language. The specification can be downloaded over the e-Codex Nexus repository server[3]. Figure 3 gives an overview of these public interfaces. On the client side there are two variants possible, the push/push or the push/pull interface. The push/push interface also requires the client to provide a web service so the connector can push messages to the client.

[1] The domibusConnectorClient is also available over nexus [7] and contains a detailed documentation

[2] Figure 5 shows how the message routing is done

[3] https://secure.e-codex.eu/nexus/content/repositories/releases/eu/domibus/connector/domibusConnectorAPI/4.1.0-RELEASE/domibusConnectorAPI-4.1.0-RELEASE-wsdl.zip

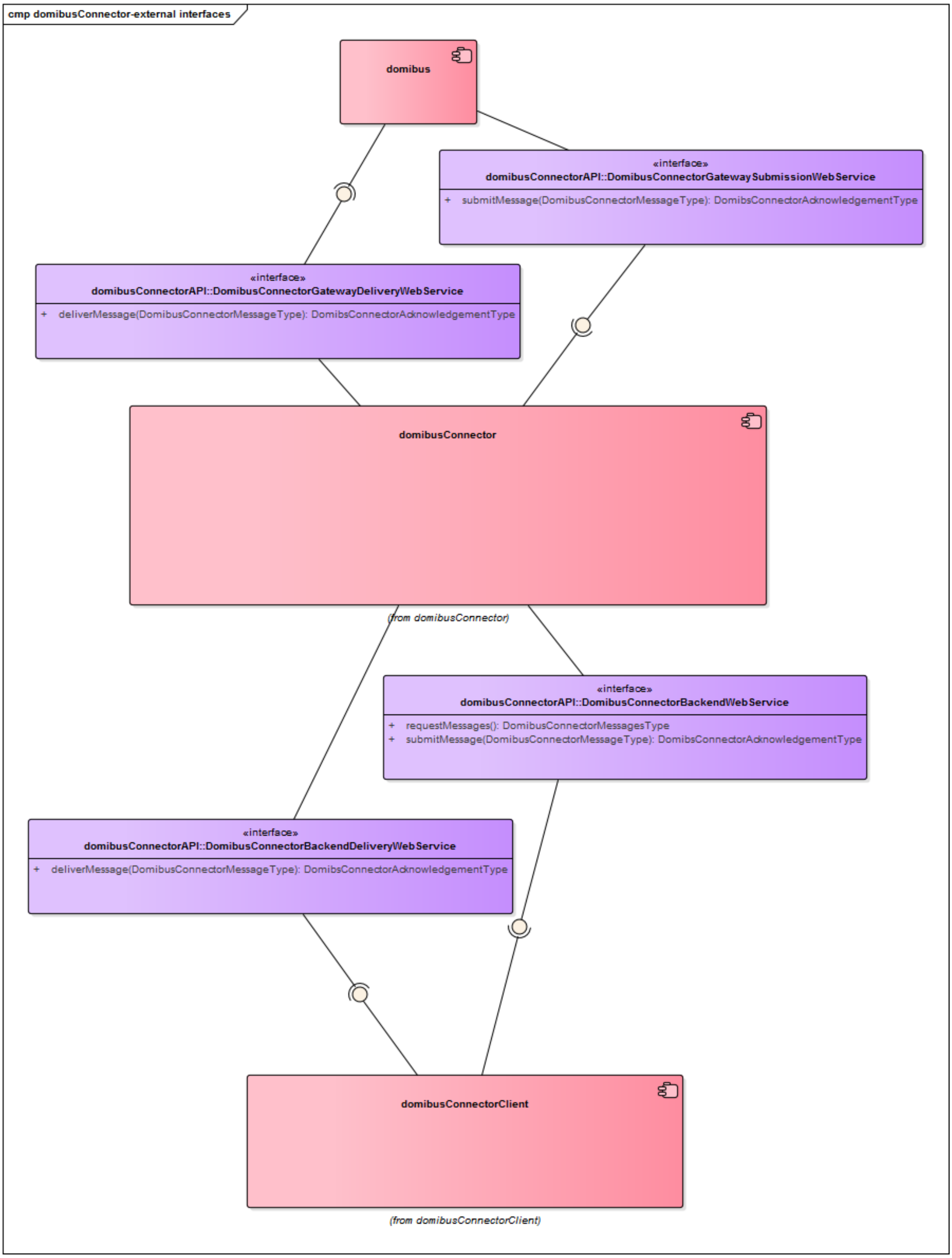

cmp domibusConnector-external interfaces
domibus
«interface»
domibusConnectorAPI::DomibusConnectorGatewaySubmissionWebService
+ submitMessage(DomibusConnectorMessageType): DomibsConnectorAcknowledgementType
«interface»
domibusConnectorAPI::DomibusConnectorGatewayDeliveryWebService
+ deliverMessage(DomibusConnectorMessageType): DomibsConnectorAcknowledgementType
domibusConnector
(from domibusConnector)
«interface»
domibusConnectorAPI::DomibusConnectorBackendWebService
+ requestMessages(): DomibusConnectorMessagesType
+ submitMessage(DomibusConnectorMessageType): DomibsConnectorAcknowledgementType
«interface»
domibusConnectorAPI::DomibusConnectorBackendDeliveryWebService
+ deliverMessage(DomibusConnectorMessageType): DomibsConnectorAcknowledgementType
domibusConnectorClient
(from domibusConnectorClient)

*Figure 3: Connector Web Service Interfaces*

#### 2.2.1.1. DomibusConnectorGatewayDeliveryWebService

The domibusConnector implements this interface for message delivery from the gateway to the connector. This interface is implemented by the domibusConnectorGatewayLink (see chapter 2.3.10) module.

#### 2.2.1.2. DomibusConnectorGatewaySubmissionWebService

The DomibusConnectorGatewaySubmissionWebService interface is the counterpart. It is implemented by the domibus-connector-plugin installed at the DOMIBUS gateway and allows the connector to push messages to the gateway.

#### 2.2.1.3. DomibusConnectorBackendWebService

The DomibusConnectorBackendWebService interface is implemented by the domibusConnectorBackendLink module (see chapter 2.3.9) and provides two methods:

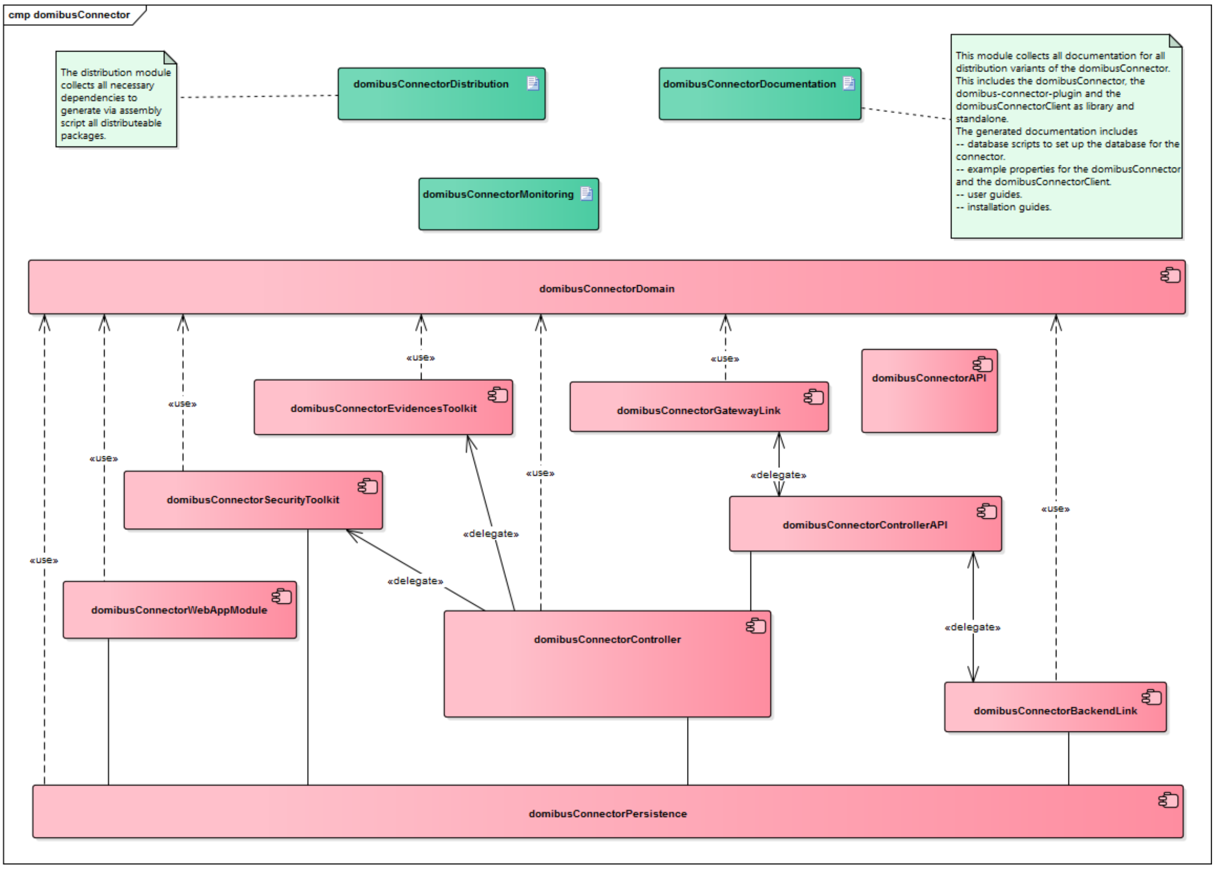
- requestMessages: allows the client to pull pending messages
- submitMessage: is called to submit a message from the client to the connector

#### 2.2.1.4. DomibusConnectorBackendDeliveryWebService

For pushing messages to the client, the client has to implement the DomibusConnectorBackendDeliveryWebService which allows the connector to push messages to the client.

## 2.3. domibusConnector Components

The domibusConnector itself is a java web application. The application is developed as a multi module maven project. The project is separated into multiple modules (Figure 4: domibusConnector 4.0 component overview). Each module has a specific purpose.


cmp domibusConnector
domibusConnectorDistribution
domibusConnectorDocumentation
domibusConnectorMonitoring
domibusConnectorDomain
«use»
domibusConnectorEvidencesToolkit
domibusConnectorGatewayLink
domibusConnectorAPI
domibusConnectorSecurityToolkit
«delegate»
domibusConnectorControllerAPI
domibusConnectorWebAppModule
domibusConnectorController
domibusConnectorBackendLink
domibusConnectorPersistence

The distribution module collects all necessary dependencies to generate via assembly script all distributeable packages.

This module collects all documentation for all distribution variants of the domibusConnector.
This includes the domibusConnector, the domibus-connector-plugin and the domibusConnectorClient as library and standalone.
The generated documentation includes
-- database scripts to set up the database for the connector.
-- example properties for the domibusConnector and the domibusConnectorClient.
-- user guides.
-- installation guides.

***Figure 4: domibusConnector 4.0 component overview***

### 2.3.1. domibuConnectorAPI

Holds the public interface descriptions, the wsdl files and the web service policy.

### 2.3.2. domibusConnectorDomain

Contains the domain model, only used by the connector internally.

### 2.3.3. domibusConnectorControllerAPI

Provides an internal API for connectorController and the link modules.

### 2.3.4. domibusConnectorController

This module contains all the business logic. It uses other modules for resolving the asic-s container, sending messages to the gateway or persisting messages.

### 2.3.5. domibusConnectorEvidencesToolkit

This module is responsible for creating the evidences. It is called by the connectorController.

### 2.3.6. domibusConnectorWebApp

This module includes the other modules and also contains a web application for managing the connector. Also the deployable ware is generated from this module.

### 2.3.7. domibusConnectorSecurityToolkit

This module is responsible for validation and creation of the ASIC-S container. It does all the work related to the ASIC-S container. For this purpose it makes use of the dss security library.

### 2.3.8. domibusConnectorPersistence

This module is responsible for persisting messages, confirmations and message states. Message content is only persisted between the message has been received by the gateway/connectorClient and will be removed after the message has been delivered to the client/submitted to the gateway.

### 2.3.9. domibusConnectorBackendLink

This module is responsible for the communication with the connectorClients (national system, backend, connectorClient). For this purpose a web service is created. For delivering messages a push client the web service of the client is called. For pull clients the messages are stored until the client asks for them.

This module also determines the correct backend where the message is delivered to. Figure 5: domibusConnectorBackendLink message routing shows an activity diagram how this is done.

### 2.3.10. domibusConnectorGatewayLink

This module is responsible for providing the web service for the gateway communication. It also submits the messages to the gateway for the connectorController. The current release only supports web services for transport, later releases may also support JMS.

### 2.3.11. Other modules

The maven project contains some more modules. Those modules are used for creating the distribution, holding the documentation and providing test data and executing tests. Those modules are not part of this description because they are only used during development.

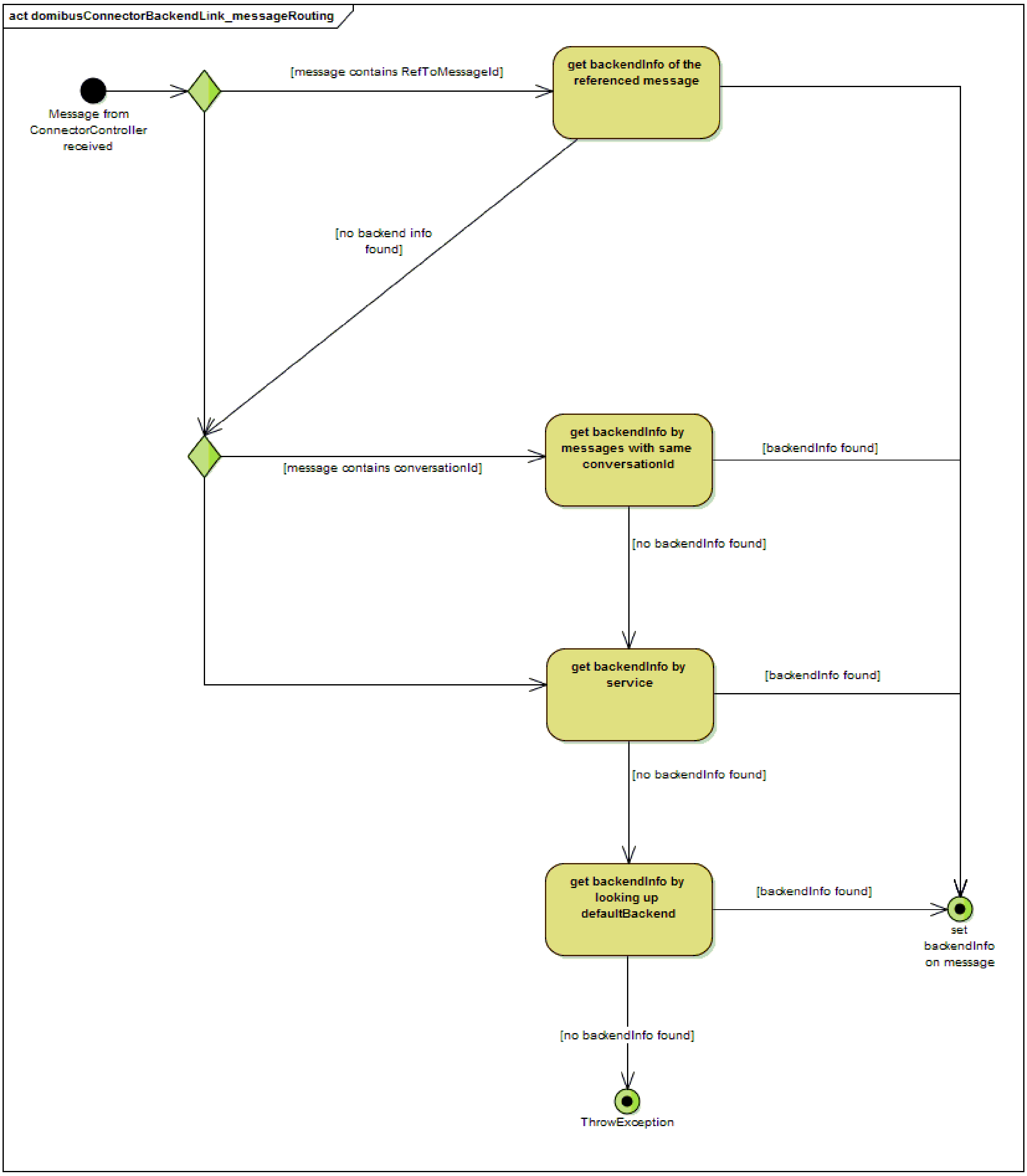

act domibusConnectorBackendLink_messageRouting
Message from ConnectorController received
[message contains RefToMessageId]
get backendInfo of the referenced message
[no backend info found]
[message contains conversationId]
get backendInfo by messages with same conversationId
[backendInfo found]
[no backendInfo found]
get backendInfo by service
[backendInfo found]
[no backendInfo found]
get backendInfo by looking up defaultBackend
[backendInfo found]
set backendInfo on message
[no backendInfo found]
ThrowException

***Figure 5: domibusConnectorBackendLink message routing***

## 2.4. Internal Queues

The domibusConnector uses an internal queueing system for decoupling its components. This queueing system should also allow a vertical scalability and to improve failover in future releases. These queues are for internal connector use only and must not be used by anything else.

### 2.4.1. Configuring the internal message broker

The domibusConnector uses spring boot [4]to start its own ActiveMQ[5] message broker for these queues. So all properties used in spring boot can be used to configure this internal message broker. It is also possibly to deactivate the internal message broker and configure an external broker.

| | |
|---|---|
| spring.activemq.broker-url | URL of the ActiveMQ broker. Auto-generated by default.<br>For a remote broker connection consider this example[6]:<br>*tcp://remotehost:61616*<br>For a more complex configuration (failover,tls,..) consult the activemq documentation. [1] |
| spring.activemq.password | Password for the broker connection |
| spring.activemq.user | Username for the connection |

***Table 1: internal message broker configuration properties example***

A more detailed description is provided by the spring project [2].

### 2.4.2. Queue Description

The current version 4.1.0-RELEASE uses 3 queues (see Figure 6):

- gatewayToControllerQueue
- backendToControllerQueue
- backendWaitQueue

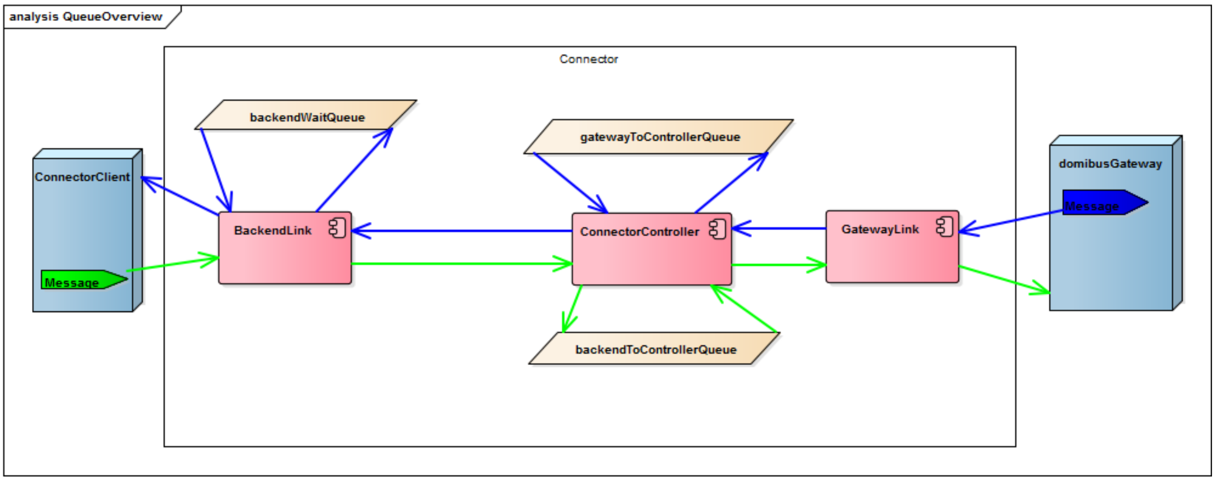


***Figure 6: Connector internal queues overview and message flow***

[4] http://spring.io/projects/spring-boot

[5] http://activemq.apache.org/

[6] http://activemq.apache.org/uri-protocols.html

### 2.4.2.1. GatewayToControllerQueue

Messages are put on this queue immediately after they have been successfully received from the gateway and stored to the connector database. The controller is fetching messages from this queue and processes (ASICS-S container handling, message confirmation handling) them.

### 2.4.2.2. backendToControllerQueue

Messages are put on this queue immediately after they have been successfully received from the backend/connectorClient and stored to the connector database. The controller is fetching messages from this queue and processes (creating ASIC-S container, creating confirmation/evidence message) them.

### 2.4.2.3. backendWaitQueue

Messages processed by the backendLink are put on the backendWaitQueue. On this queue the messages are waiting for being fetched by a pull client or fetched by the backendLink push for pushing them to the push client. For all messages on this queue the correct backend is already determined.

# 3. Configuration

The domibusConnector has multiple configuration options. This chapter contains a list of configuration properties. Some of those properties are for more experienced users. Those properties are highlighted with blue colour.

The properties are grouped by the domibusConnector modules:

## 3.1. domibusConnectorController

| Property name | Description |
|---|---|
| connector.controller.evidence.timeout-active | This property configures if timeouts for message confirmations should be checked! |
| connector.controller.evidence.check-timeout | This property defines how often the timeouts for the evidences should be checked the default value is 5 minutes. |
| connector.controller.evidence.delivery-timeout | This property defines the timeout how long the connector should wait for a DELIVERY confirmation message after a message has been successfully submitted to the gateway. The default value is 24 hours. If the timeout limit is reached the timeoutProcessor is started and a NON_DELIVERY confirmation is created for the according business message. |
| connector.controller.evidence.relayREMMD-timeout | This property defines how long the connector should wait for a relayREMMD evidence message after a message has been successfully submitted to the gateway. The default value is 24 hours. If the timeout exceeds the according timeoutProcessor is started and a NON_DELIVERY confirmation is created. |
| connector.controller.evidence.retrieval-timeout | This property defines how long the connector should wait for a retrieval evidence message after a message has been successfully submitted to the gateway. The default value is 24 hours. If the timeout exceeds a NON_DELIVERY confirmation is created for the according message. |
| domibus.connector.internal.gateway.to.controller.queue | This property defines the name of the internal gateway to controller queue. For details see chapter 2.4.2.1 page 13.<br>The default name for the queue is " gatewayToControllerQueue” |
| domibus.connector.internal.backend.to.controller.queue | This property defines the name of the internal backend to controller queue. For details see chapter 2.4.2.2 page 13.<br>The default value for this property is “backendToControllerQueue”. |

*Table 2: domibusConnectorController configuration properties*

## 3.2. domibusConnectorGatewayLink

| Property name | Description |
|---|---|
| connector.gatewaylink.ws.address | Configures the address part for the deliver message service. The service url is defined relative to the url of the CXF-Servlet (usually configured under /services), the defined url is appended. So the default url for pushing messages from domibus gateway to connector will be<br>"${SERVER_CONTEXT}/services/ domibusConnectorDeliveryWebservice"<br>Default value is "/domibusConnectorDeliveryWebservice". |
| connector.gatewaylink.ws.submission-endpoint-address | Configures the URL for the submitting messages service to the Gateway.<br>Must be configured so the connector can submit messages to the gateway. |
| connector.gatewaylink.ws.tls-key-store.password | This key/trust-store related properties are described in the Key/TrustStore Guide. [3] |
| connector.gatewaylink.ws.tls-key-store.path | |
| connector.gatewaylink.ws.tls-key.alias | |
| connector.gatewaylink.ws.tls-key.password | |
| connector.gatewaylink.ws.tls-trust-store.password | |
| connector.gatewaylink.ws.tls-trust-store.path | |

*Table 3: domibusConnectorGatewayLink configuration properties*

## 3.3. domibusConnectorBackendLink

| Property name | Description |
|---|---|
| connector.backend.internal.wait-queue.name | See chapter 2.4.2.3 on page 13 |
| connector.backend.internal.wait-queue.receive-timeout | Configures how long the polling command should wait for messages. The queue is polled if a pull client asks for messages. |
| connector.backend.ws.backend-publish-address | Specifies the address where the Backend WebService should be published the path specefied here is added to the path of the CXF-Servlet (which is per default |

| | |
|---|---|
| | configured as /service - this leads to the default URL of "/services/backend"<br>The default value is "/backend" |
| connector.backend.ws.ws-policy | Contains the path to the wsdl policy which is used communicating between connectorBackendClient and connectorBackend<br>Default value is "classpath:wsdl/backend.policy.xml" |
| connector.backend.ws.key.store.path | This key/trust-store related properties are described in the Key/TrustStore Guide. [3] |
| connector.backend.ws.key.store.password | |
| connector.backend.ws.key.key.alias | |
| connector.backend.ws.key.key.password | |
| connector.backend.ws.trust.loadCaCerts | |
| connector.backend.ws.trust.store.path | |
| connector.backend.ws.trust.store.password | |

***Table 4: domibusConnectorBackendLink configuration properties***

## 3.4. domibusConnectorSecurityToolkit

| Property name | Description |
|---|---|
| connector.security.ojstore.path | Those key/trust-store related properties are described in the Key/TrustStore Guide. [3] |
| connector.security.ojstore.password | |
| connector.security.trustStore.path | |
| connector.security.trustStore.password | |

***Table 5: domibusConnectorSecurityToolkit configuration properties***

## 3.5. domibusConnectorEvidencesToolkit

| Property name | Description |
|---|---|
| connector.evidences.keystore.path | Those key/trust-store related properties are described in the Key/TrustStore Guide. [3] |
| connector.evidences.keystore.password | |
| connector.evidences.key.alias | |
| connector.evidences.key.password | |
| connector.evidences.hashAlgorithm | The hashAlgorithm used for signing the evidences. Possibly are:<br>MD5, SHA1, SHA256, SHA512<br>The default value is SHA256. |

***Table 6: domibusConnectorEvidencesToolkit configuration properties***

# 4. Logging

The connector uses log4j2 logging framework for logging. The logging system supports different logging levels and different logging for different components of the connector. For further information consult the log4j2 [4] documentation.

## 4.1. Logging Level

The used logging levels are listed by increasing criticality:

- TRACE – logs almost everything (do not use that in production!)
- DEBUG – for DEBUG purposes, logs almost everything
- INFO – for Information, perfect log level to put this log messages into an audit system
- WARN – for Warnings, something might have been gone wrong, the administrator should take a look on it, but the system was able to move on
- ERROR – for Errors, where something happened and the system was not able to move on, maybe a message ended up in error state

## 4.2. Logger Names

The connector uses different logger names, which can be configured with different logging levels. For the description of the logger names consult the example log4j.properties which are provided by the distribution package of the domibusConnector.

# 5. domibusConnector Administration UI

The domibusConnector-4.1.0-RELEASE is a web application which already contains the domibusConnector Administration UI (User Interface). This domibusConnector Administration UI gives information over processed messages and makes it possible to do configuration over a graphical user interface.

This chapter contains information about the domibusConnector Administration UI features. It also describes how to use them.

## 5.1. Login

When you deployed the domibusConnector on your application server it is placed at the relative path http://[your servers ip or name and port]/[web application context]/admin. When entering the site it first wants you to login with a valid user and password.

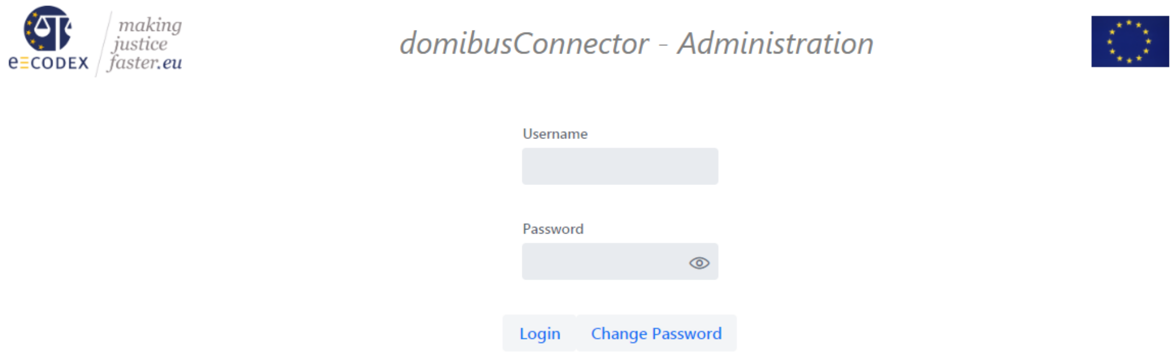

e≡CODEX making justice faster.eu
domibusConnector - Administration
Username
Password
Login
Change Password

*Figure 7: Screenshot login screen*

### 5.1.1. Preconfigured users

When the provided database scripts of the domibusConnector have been used to install and prepare or migrate the database of the domibusConnector then two default users were created.

| Username | Initial Password | Role | Description |
|---|---|---|---|
| **admin** | admin | ADMIN | Default administration user |
| **user** | user | USER | Default user read-only user. |

*Table 7 Preconfigured users from the Administration UI*

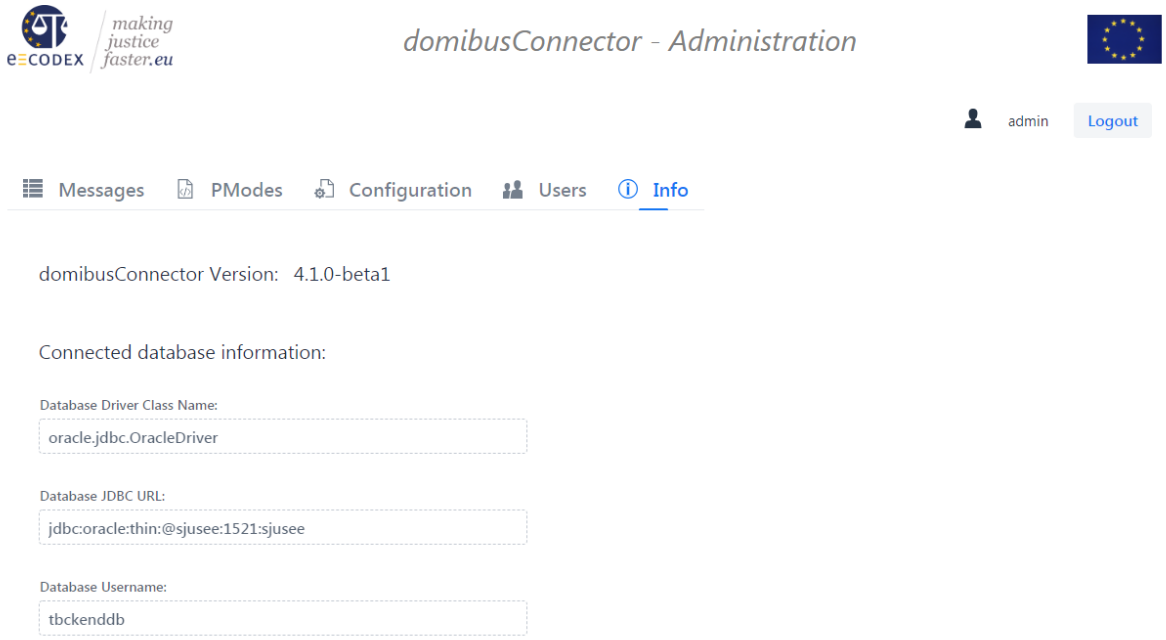
The role concept of the domibusConnector Administration UI foresees 2 Roles:

- USER: This role contains privileges to search messages, create reports and read configuration.
- ADMIN: In addition to the USER role, users of this role can edit configuration, upload P-Modes and create or edit other users.

**!!! When first logging in with one of the predefined users, the domibusConnector Administration UI immediately forces a password change. !!!**

## 5.2. Static information and main menu

After login, you should find yourself on the "Info" page which shows some useful static information about the domibusConnector instance like version and connected database.



domibusConnector Version: 4.1.0-beta1

Connected database information:

Database Driver Class Name:
oracle.jdbc.OracleDriver

Database JDBC URL:
jdbc:oracle:thin:@sjusee:1521:sjusee

Database Username:
tbckenddb

***Figure 8: Screenshot Info page***

### 5.2.1. User Information

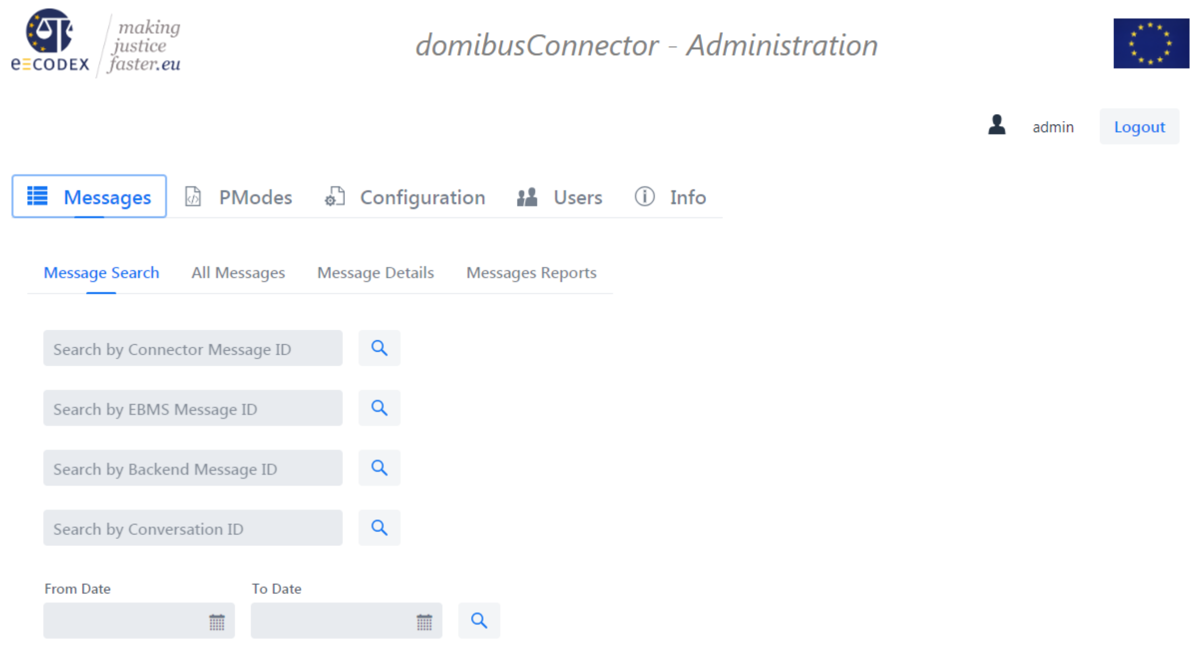
Between the header of the Administration UI and the main menu, the logged in user can be seen. There is also a "Logout" button that should be used when work has been done.

### 5.2.2. Main menu

The main menu holds the main sections of the Administration UI. The section chosen can be seen as it is highlighted in blue and underlined.

## 5.3. Messages section

The "Messages" section shows all information from the database of the domibusConnector to processed messages:



domibusConnector - Administration

admin Logout

Messages PModes Configuration Users Info

Message Search All Messages Message Details Messages Reports

Search by Connector Message ID

Search by EBMS Message ID

Search by Backend Message ID

Search by Conversation ID

From Date To Date

*Figure 9: Screenshot Messages section*

### 5.3.1. Message Search

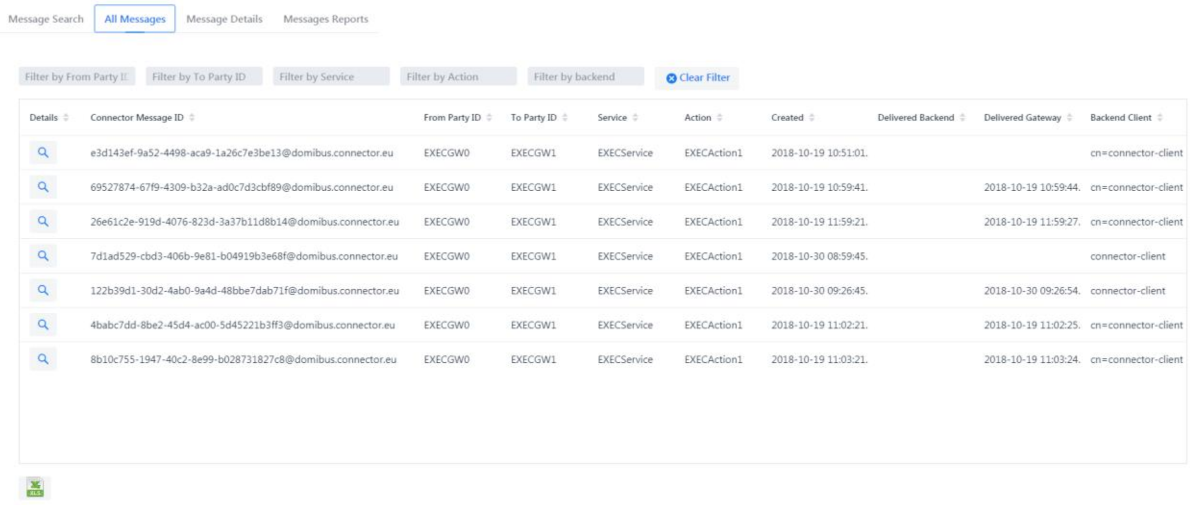
If a certain message is searched, where particular message information is known, this message can be searched here providing the particular information.

Also, if this particular information finds multiple results, like with the ConversationID, or with a certain time period, a list of results is displayed.

Once a single result selection is chosen, or, if the search result is a single result, the “Message Details” described in Message Details are shown.

### 5.3.2. All Messages

This sub-section provides a complete list of all messages processed by this instance of the domibusConnector.



| Details | Connector Message ID | From Party ID | To Party ID | Service | Action | Created | Delivered Backend | Delivered Gateway | Backend Client |
|---|---|---|---|---|---|---|---|---|---|
| 🔍 | e3d143ef-9a52-4498-aca9-1a26c7e3be13@domibus.connector.eu | EXECGW0 | EXECGW1 | EXECService | EXECAction1 | 2018-10-19 10:51:01. | | | cn=connector-client |
| 🔍 | 69527874-67f9-4309-b32a-ad0c7d3cbf89@domibus.connector.eu | EXECGW0 | EXECGW1 | EXECService | EXECAction1 | 2018-10-19 10:59:41. | | 2018-10-19 10:59:44. | cn=connector-client |
| 🔍 | 26e61c2e-919d-4076-823d-3a37b11d8b14@domibus.connector.eu | EXECGW0 | EXECGW1 | EXECService | EXECAction1 | 2018-10-19 11:59:21. | | 2018-10-19 11:59:27. | cn=connector-client |
| 🔍 | 7d1ad529-cbd3-406b-9e81-b04919b3e68f@domibus.connector.eu | EXECGW0 | EXECGW1 | EXECService | EXECAction1 | 2018-10-30 08:59:45. | | | connector-client |
| 🔍 | 122b39d1-30d2-4ab0-9a4d-48bbe7dab71f@domibus.connector.eu | EXECGW0 | EXECGW1 | EXECService | EXECAction1 | 2018-10-30 09:26:45. | | 2018-10-30 09:26:54. | connector-client |
| 🔍 | 4babc7dd-8be2-45d4-ac00-5d45221b3ff3@domibus.connector.eu | EXECGW0 | EXECGW1 | EXECService | EXECAction1 | 2018-10-19 11:02:21. | | 2018-10-19 11:02:25. | cn=connector-client |
| 🔍 | 8b10c755-1947-40c2-8e99-b028731827c8@domibus.connector.eu | EXECGW0 | EXECGW1 | EXECService | EXECAction1 | 2018-10-19 11:03:21. | | 2018-10-19 11:03:24. | cn=connector-client |

***Figure 10 Screenshot All Messages sub-section***

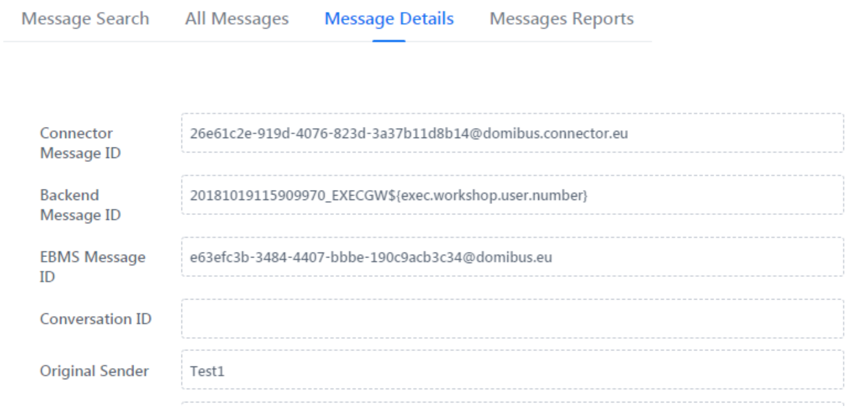
This list can be filtered using provided filter criteria. Those filter criteria can be combined and take effect as soon as edited. The "Clear Filter" button clears all criteria and the full list will be shown again.

This table, as well as any table within the Administration UI, can be sorted by clicking the column header in any direction. Also combined sorting of multiple columns is possible.

The currently shown message list (with or without filters) can be exported and downloaded as a Microsoft Excel file for further usage.

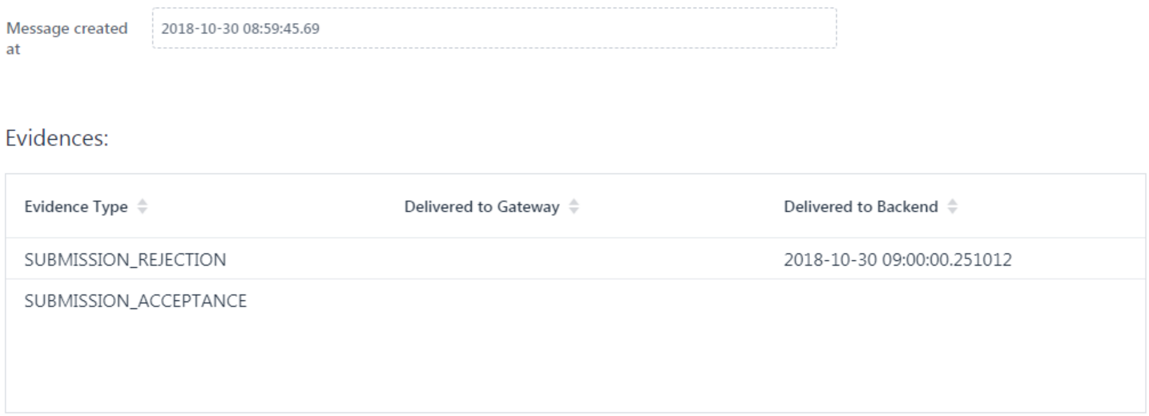
### 5.3.3. Message Details

Once a single message is chosen, whether this is done by selecting a message from a list or searching for single resulting criteria, this message is then shown with the "Message Details".



| | |
|---|---|
| Connector Message ID | 26e61c2e-919d-4076-823d-3a37b11d8b14@domibus.connector.eu |
| Backend Message ID | 20181019115909970_EXECGW${exec.workshop.user.number} |
| EBMS Message ID | e63efc3b-3484-4407-bbbe-190c9acb3c34@domibus.eu |
| Conversation ID | |
| Original Sender | Test1 |

….



Message created at: 2018-10-30 08:59:45.69

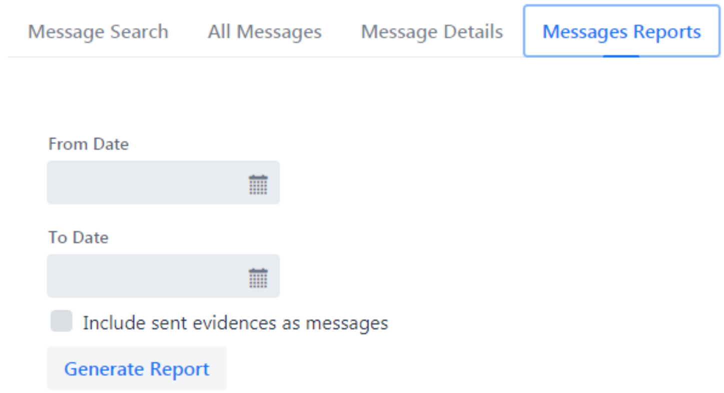
Evidences:

| Evidence Type | Delivered to Gateway | Delivered to Backend |
|---|---|---|
| SUBMISSION_REJECTION | | 2018-10-30 09:00:00.251012 |
| SUBMISSION_ACCEPTANCE | | |

Those message details contain ALL available information to this particular message, including all ETSI-REM evidences created or received for this message.

### 5.3.4. Messages Reports

Other than the lists of messages shown before, this sub-section only focuses on numbers.



From Date:

To Date:

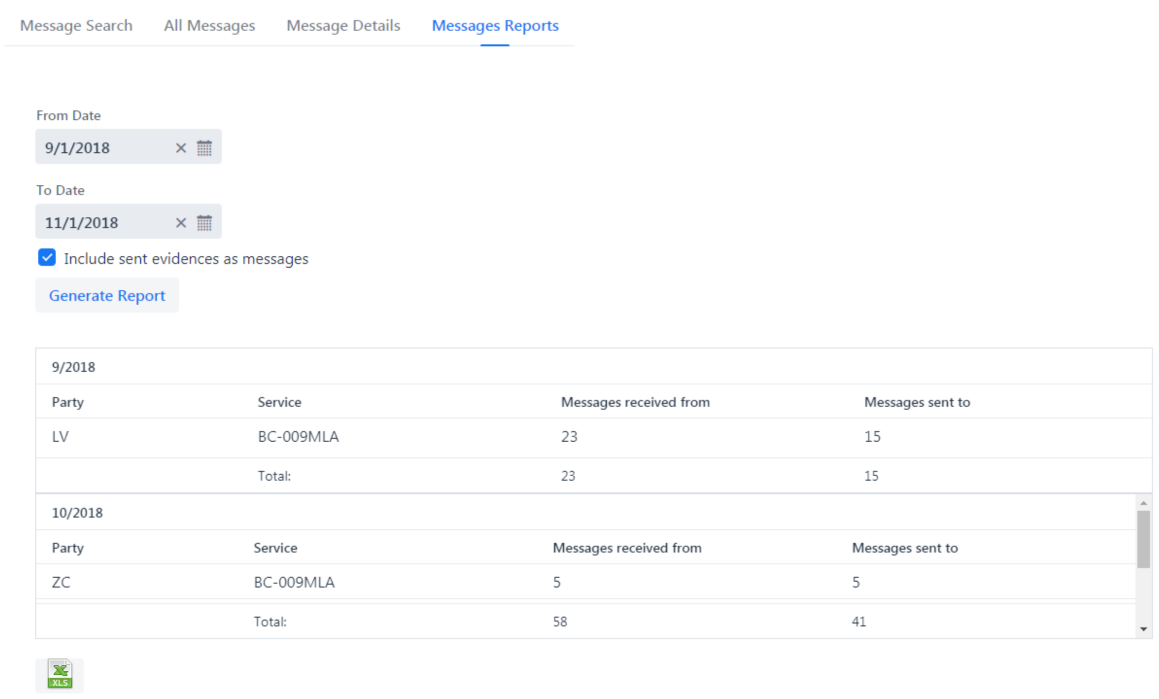
□ Include sent evidences as messages

***Figure 11: Screenshot Messages Reports – search criteria***

For first a period has to be selected that define the time range of the reports that have to be generated. By using the "calendar" symbols and picking the date there is the most comfortable way to not run into date format issues.

The checkbox "Include sent evidences as messages" indicates, if ETSI-REM evidences that are sent or received for messages as own messages should be counted as such.

Once criteria are selected, the "Generate Report" Button must be clicked to generate and show the reports.



From Date: 9/1/2018

To Date: 11/1/2018

☑ Include sent evidences as messages

9/2018

| Party | Service | Messages received from | Messages sent to |
|---|---|---|---|
| LV | BC-009MLA | 23 | 15 |
|  | Total: | 23 | 15 |

10/2018

| Party | Service | Messages received from | Messages sent to |
|---|---|---|---|
| ZC | BC-009MLA | 5 | 5 |
|  | Total: | 58 | 41 |

*Figure 12 Screenshot Message Reports results*

As seen in Figure 12 the results are separated by month. The generated reports can be downloaded as an XLS file for further usage.

## 5.4. PModes section

The “p-modes” that are distributed by the configuration management for the DOMIBUS gateway can also be used for the domibusConnector database to have necessary data to support business use cases in your domibusConnector database.

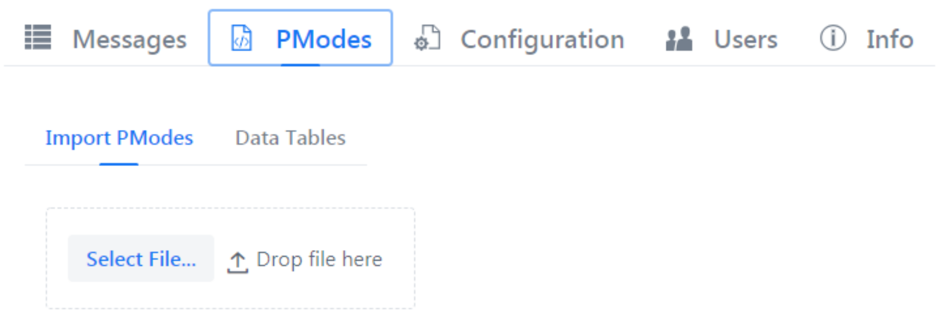
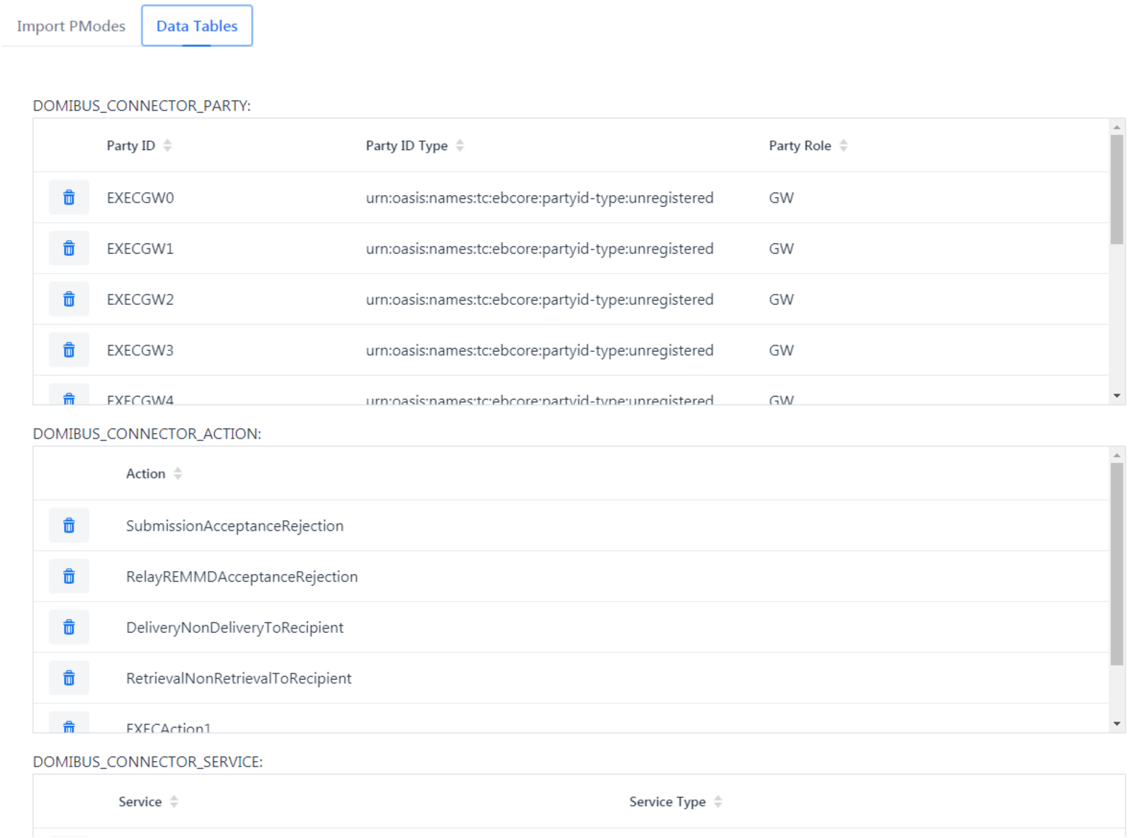


*Figure 13: Screenshot PModes section*

The p-mode file from the configuration management can be selected here and will be uploaded into the database of the domibusConnector.

Every ACTION, PARTY and SERVICE that is found in the p-modes and which do not exist in the database already, will be created. The domibusConnector will neither change any existing database items (service, action or party), nor will it delete any of those. If the p-mode-file cannot be processed an error message will appear.

After importing PModes, or if there are already imported PModes you can see and edit the results from this import in sub-section "Data Tables"



DOMIBUS_CONNECTOR_PARTY:

| | Party ID | Party ID Type | Party Role |
|---|---|---|---|
| 🗑 | EXECGW0 | urn:oasis:names:tc:ebcore:partyid-type:unregistered | GW |
| 🗑 | EXECGW1 | urn:oasis:names:tc:ebcore:partyid-type:unregistered | GW |
| 🗑 | EXECGW2 | urn:oasis:names:tc:ebcore:partyid-type:unregistered | GW |
| 🗑 | EXECGW3 | urn:oasis:names:tc:ebcore:partyid-type:unregistered | GW |
| 🗑 | EXECGW4 | urn:oasis:names:tc:ebcore:partyid-type:unregistered | GW |

DOMIBUS_CONNECTOR_ACTION:

| | Action |
|---|---|
| 🗑 | SubmissionAcceptanceRejection |
| 🗑 | RelayREMMDAcceptanceRejection |
| 🗑 | DeliveryNonDeliveryToRecipient |
| 🗑 | RetrievalNonRetrievalToRecipient |
| 🗑 | EXECAction1 |

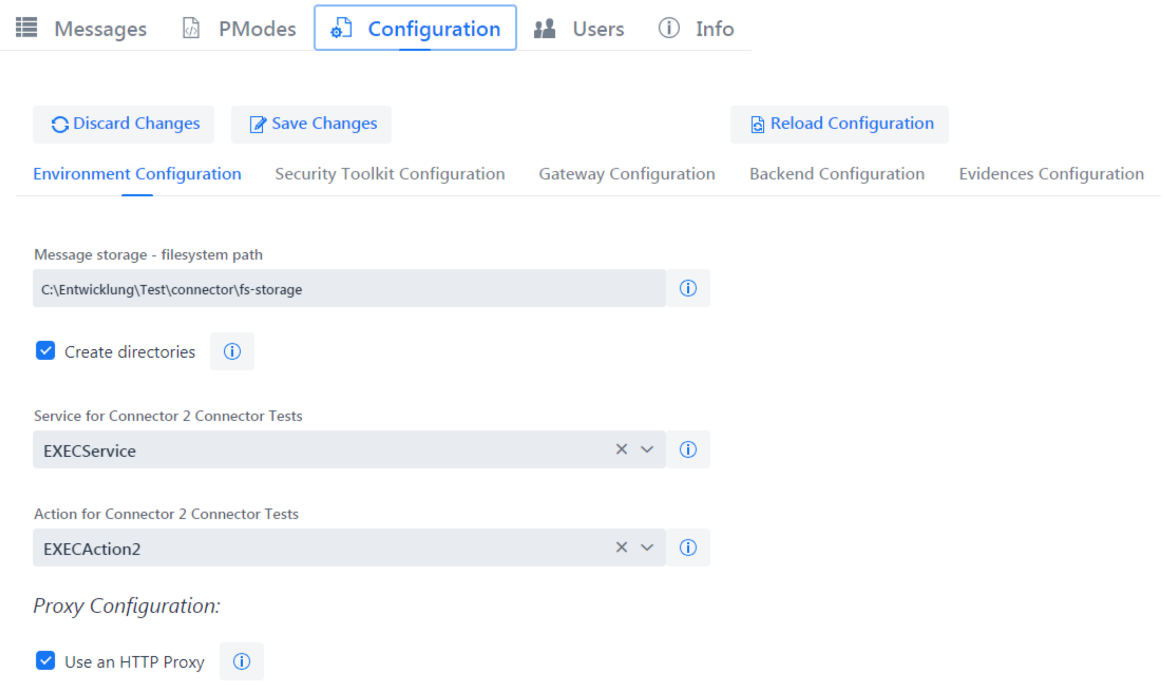
DOMIBUS_CONNECTOR_SERVICE:

| | Service | Service Type |
|---|---|---|

*Figure 14 Screenshot Data Tables sub-section*

## 5.5. Configuration section

All properties described in chapter Configuration can also be viewed and edited within the domibusConnector Administration UI.



Message storage - filesystem path: C:\Entwicklung\Test\connector\fs-storage

☑ Create directories

Service for Connector 2 Connector Tests: EXECService

Action for Connector 2 Connector Tests: EXECAction2

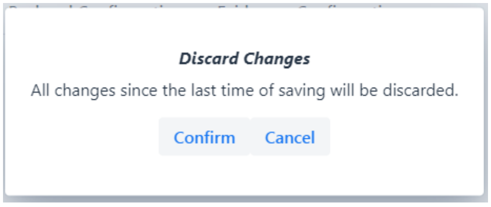
*Proxy Configuration:*

☑ Use an HTTP Proxy

***Figure 15 Screenshot Configuration section***

Be aware that edited and saved configuration properties will NOT be written into the properties file they initially need to be set in, but will be stored into the domibusConnector database.

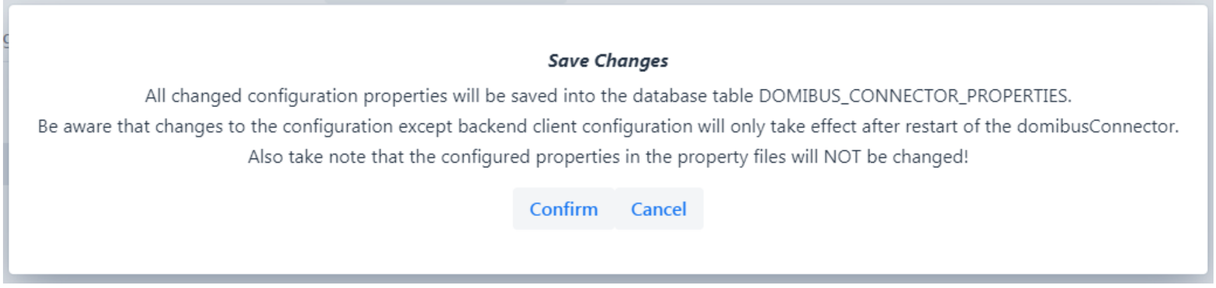
The three possibilities of handling the configuration edited within the UI are given by the buttons in the button bar. Each of them will open an explanatory dialog before taking action.



**Discard Changes**

All changes since the last time of saving will be discarded.

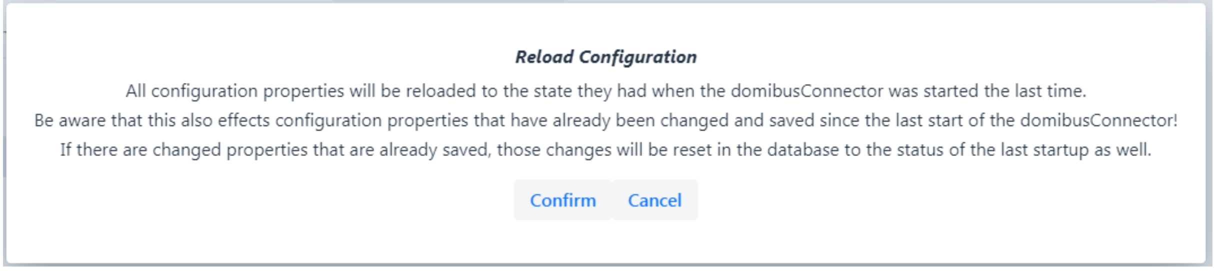
***Figure 16 Screenshot Discard Changes Dialog***



**Save Changes**

All changed configuration properties will be saved into the database table DOMIBUS_CONNECTOR_PROPERTIES.
Be aware that changes to the configuration except backend client configuration will only take effect after restart of the domibusConnector.
Also take note that the configured properties in the property files will NOT be changed!

***Figure 17 Screenshot Save Changes Dialog***



**Reload Configuration**

All configuration properties will be reloaded to the state they had when the domibusConnector was started the last time.
Be aware that this also effects configuration properties that have already been changed and saved since the last start of the domibusConnector!
If there are changed properties that are already saved, those changes will be reset in the database to the status of the last startup as well.

***Figure 18 Screenshot Reload Configuration Dialog***

This whole section is readable only for users of role USER.

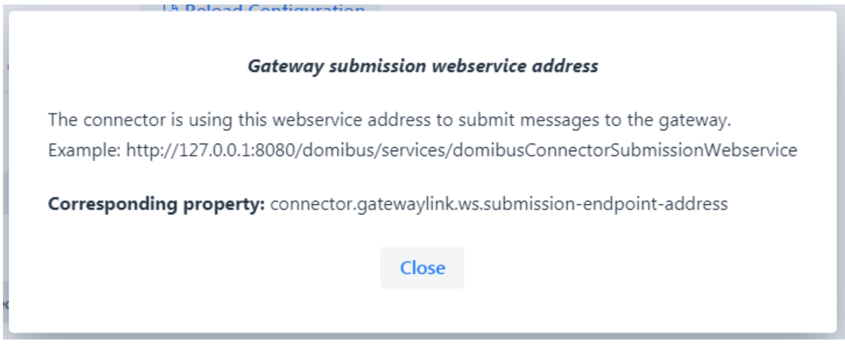
The different properties are grouped thematically into sub-sections.

- Environment Configuration
- Security Toolkit Configuration
- Gateway Configuration
- Backend Configuration
- Evidences Configuration

Each configuration item has the same structure:

- Name of the configuration item.
- Value of the configuration item. This can be text, select box or check box.
- Information on the configuration item.

When opening the information on a configuration item a dialog appears



**Gateway submission webservice address**

The connector is using this webservice address to submit messages to the gateway.
Example: http://127.0.0.1:8080/domibus/services/domibusConnectorSubmissionWebservice

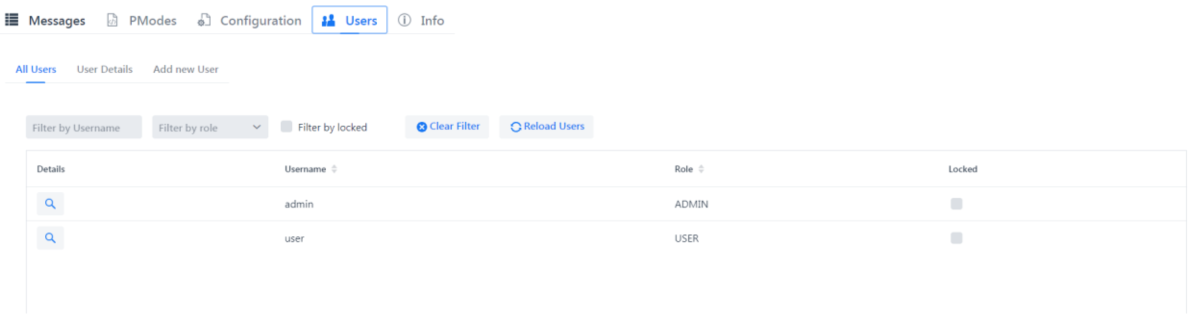
**Corresponding property:** connector.gatewaylink.ws.submission-endpoint-address

***Figure 19 Screenshot Information on a configuration item***

This dialog displays the name of the configuration item, an explanation what it means, mostly an example and the corresponding property name. That way, every configuration item can be found either in the database or in the property file.

## 5.6. Users section

The “Users” section gives users of the role ADMIN the possibility to create new users, edit existing users or deactivate/delete users.



| Details | Username | Role | Locked |
|---|---|---|---|
| 🔍 | admin | ADMIN | □ |
| 🔍 | user | USER | □ |

*Figure 20 Screenshot Users section*

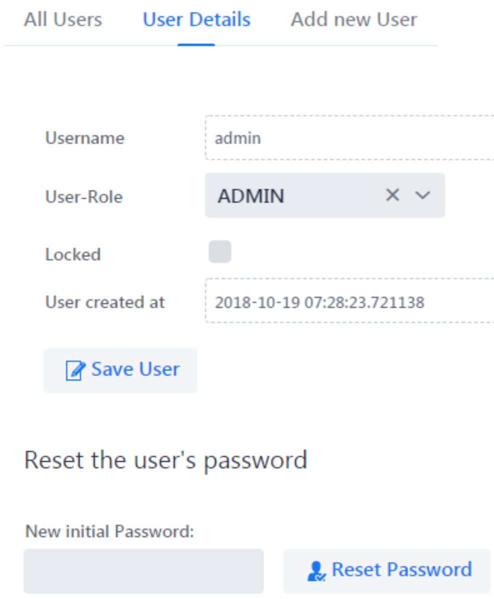
## 5.6.1. All Users

The first displayed sub-section is “All Users”. It gives an overview of existing users, their roles and the status of a user.

This list can also be filtered.

## 5.6.2. User Details

When selecting a user, the “User Details” will show.



| | |
|---|---|
| Username | admin |
| User-Role | ADMIN |
| Locked | □ |
| User created at | 2018-10-19 07:28:23.721138 |

Reset the user's password

New initial Password:

*Figure 21 Screenshot User Details*

Some of the user’s details must not be edited. Like the Username. This is static and cannot be changed.

What can be changed is the role of that user. Also the user can be locked. If a user is locked already because of failed login trials, it can be unlocked here. This should always be combined with reset of the user’s password. The new password that can be given here is NOT a persistent one. This is a new initial one for the user. The next time the user will login with this given initial password, it is forced to change the password immediately.

### 5.6.3. Add new User

Also new users can be created.

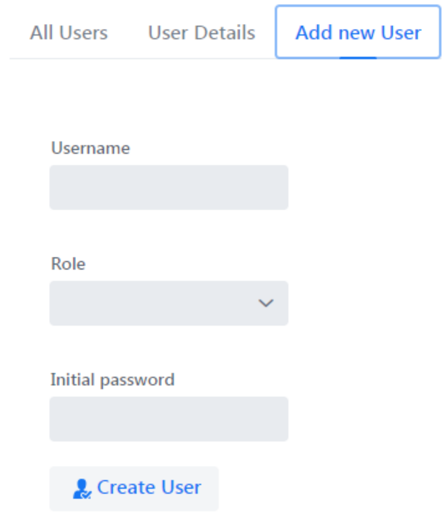


*Figure 22 Screenshot Add new User*

All that needs to be given is a Username (static once the user is created), the role the user attends to and an initial password.

# 6. Tasks involving database access

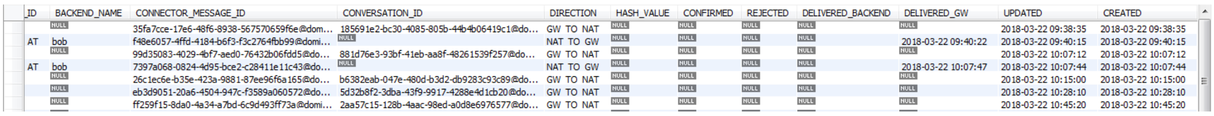
This chapter describes typical problems or tasks which are not supported by the user interface yet. So the main objective is to give an overview of important tables and columns in the database and their meaning.

## 6.1. Finding a message

Sometimes the question appears what happened to a specific message. It is also possible to use the user interface for finding a message (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**, page **Fehler! Textmarke nicht definiert.**). First of all you need to know something about the message you are looking for. Good identifiers for the message are

- EBMS Message ID (assigned by the Gateway)
- Backend Message ID (assigned by the national system, or backend application)
- Connector Message ID (assigned by the connector, is not communicated to any other system)

After you gathered the information open the connector database. The connector contains a table named DOMIBUS_CONNECTOR_MESSAGE (Figure 23 shows an example of the table). This table contains all business messages received by the connector. You can search the message by the EBMS_MESSAGE_ID, BACKEND_MESSAGE_ID or also by the CONVERSATION_ID. The column DIRECTION tells use if the message was submitted to the gateway or was received by the gateway.



| _ID | BACKEND_NAME | CONNECTOR_MESSAGE_ID | CONVERSATION_ID | DIRECTION | HASH_VALUE | CONFIRMED | REJECTED | DELIVERED_BACKEND | DELIVERED_GW | UPDATED | CREATED |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | NULL | 35fa7cce-17e6-48f6-8938-567570659f6e@dom... | 185691e2-bc30-4085-805b-44b4b06419c1@do... | GW TO NAT | NULL | NULL | NULL | NULL | NULL | 2018-03-22 09:38:35 | 2018-03-22 09:38:35 |
| AT | bob | f48e6057-4ffd-4184-b6f3-f3c2764fbb99@domi... | NULL | NAT TO GW | NULL | NULL | NULL | NULL | 2018-03-22 09:40:22 | 2018-03-22 09:40:15 | 2018-03-22 09:40:15 |
| | NULL | 99d35083-4029-4bf7-aed0-76432b06fdd5@do... | 881d76e3-93bf-41eb-aa8f-48261539f257@do... | GW TO NAT | NULL | NULL | NULL | NULL | NULL | 2018-03-22 10:07:12 | 2018-03-22 10:07:12 |
| AT | bob | 7397a068-0824-4d95-bce2-c28411e11c43@do... | NULL | NAT TO GW | NULL | NULL | NULL | NULL | 2018-03-22 10:07:47 | 2018-03-22 10:07:44 | 2018-03-22 10:07:44 |
| | NULL | 26c1ec6e-b35e-423a-9881-87ee96f6a165@do... | b6382eab-047e-480d-b3d2-db9283c93c89@do... | GW TO NAT | NULL | NULL | NULL | NULL | NULL | 2018-03-22 10:15:00 | 2018-03-22 10:15:00 |
| | NULL | eb3d9051-20a6-4504-947c-f3589a060572@do... | 5d32b8f2-3dba-43f9-9917-4288e4d1cb20@do... | GW TO NAT | NULL | NULL | NULL | NULL | NULL | 2018-03-22 10:28:10 | 2018-03-22 10:28:10 |
| | NULL | ff259f15-8da0-4a34-a7bd-6c9d493ff73a@domi... | 2aa57c15-128b-4aac-98ed-a0d8e6976577@do... | GW TO NAT | NULL | NULL | NULL | NULL | NULL | 2018-03-22 10:45:20 | 2018-03-22 10:45:20 |

***Figure 23: Connector 4.0 – example of DOMIBUS_CONNECTOR_MESSAGE table***

## 6.2. Message State

After you have found your message in the DOMIBUS_CONNECTOR_MESSAGE table you are interested to determine the state of your message. For this purpose there are 4 interesting columns

- CONFIRMED – Is null by default, will be set to the timestamp of the confirmation
- REJECTED – Is null by default, will be set to the timestamp of the rejection
- DELIVERED_BACKEND – is null by default, will be set to the delivery time
- DELIVERED_GW – is null by default, will be set to the delivery time

### 6.2.1. Column CONFIRMED description

Is null by default and will be set to the timestamp of the confirmation. The message is confirmed if and only if the message has not been rejected yet and the according DELIVERY confirmation is received by the connector.

### 6.2.2. Column REJECTED description

Is null by default and will be set to the timestamp of the rejection. The message is rejected if there is a NON_DELIVERY received for this message. Or the evidence timeout is reached, then a NON_DELIVERY is created and the message is marked as rejected.

### 6.2.3. Column DELIVERED_BACKEND description

Is null by default and will be set to the timestamp of the delivery to the backend/national system. Depending on the backend type the behaviour when this timestamp is set differs slightly.

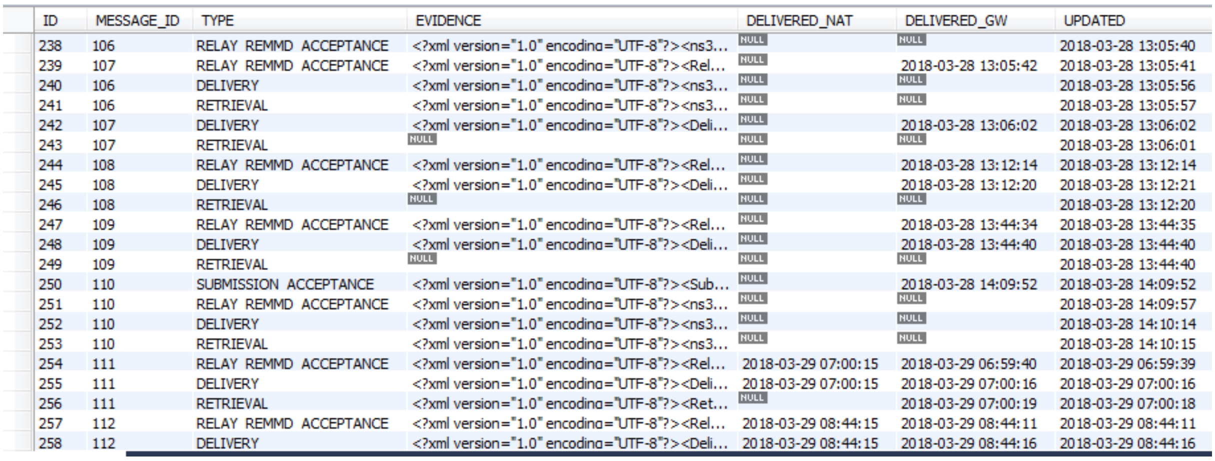
### 6.2.4. Column DELIVERED_GW description

Is null by default and will be set to the timestamp of the delivery to the gateway. This timestamp is set after the gateway has accepted the message.

## 6.3. Looking for Evidences / Confirmations

The message confirmations or ETSI-REM [5] Evidences are stored in a different table. This confirmation messages are providing proof that a message has been accepted, relayed, delivered, rejected. The evidences are created by the connector.

The evidences are stored in the DOMIBUS_CONNECTOR_EVIDENCE table and the MESSAGE_ID is referencing the message.



| ID | MESSAGE_ID | TYPE | EVIDENCE | DELIVERED_NAT | DELIVERED_GW | UPDATED |
|---|---|---|---|---|---|---|
| 238 | 106 | RELAY REMMD ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><ns3... | NULL | NULL | 2018-03-28 13:05:40 |
| 239 | 107 | RELAY REMMD ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><Rel... | NULL | 2018-03-28 13:05:42 | 2018-03-28 13:05:41 |
| 240 | 106 | DELIVERY | <?xml version="1.0" encoding="UTF-8"?><ns3... | NULL | NULL | 2018-03-28 13:05:56 |
| 241 | 106 | RETRIEVAL | <?xml version="1.0" encoding="UTF-8"?><ns3... | NULL | NULL | 2018-03-28 13:05:57 |
| 242 | 107 | DELIVERY | <?xml version="1.0" encoding="UTF-8"?><Deli... | NULL | 2018-03-28 13:06:02 | 2018-03-28 13:06:02 |
| 243 | 107 | RETRIEVAL | NULL | NULL | NULL | 2018-03-28 13:06:01 |
| 244 | 108 | RELAY REMMD ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><Rel... | NULL | 2018-03-28 13:12:14 | 2018-03-28 13:12:14 |
| 245 | 108 | DELIVERY | <?xml version="1.0" encoding="UTF-8"?><Deli... | NULL | 2018-03-28 13:12:20 | 2018-03-28 13:12:21 |
| 246 | 108 | RETRIEVAL | NULL | NULL | NULL | 2018-03-28 13:12:20 |
| 247 | 109 | RELAY REMMD ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><Rel... | NULL | 2018-03-28 13:44:34 | 2018-03-28 13:44:35 |
| 248 | 109 | DELIVERY | <?xml version="1.0" encoding="UTF-8"?><Deli... | NULL | 2018-03-28 13:44:40 | 2018-03-28 13:44:40 |
| 249 | 109 | RETRIEVAL | NULL | NULL | NULL | 2018-03-28 13:44:40 |
| 250 | 110 | SUBMISSION ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><Sub... | NULL | 2018-03-28 14:09:52 | 2018-03-28 14:09:52 |
| 251 | 110 | RELAY REMMD ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><ns3... | NULL | NULL | 2018-03-28 14:09:57 |
| 252 | 110 | DELIVERY | <?xml version="1.0" encoding="UTF-8"?><ns3... | NULL | NULL | 2018-03-28 14:10:14 |
| 253 | 110 | RETRIEVAL | <?xml version="1.0" encoding="UTF-8"?><ns3... | NULL | NULL | 2018-03-28 14:10:15 |
| 254 | 111 | RELAY REMMD ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><Rel... | 2018-03-29 07:00:15 | 2018-03-29 06:59:40 | 2018-03-29 06:59:39 |
| 255 | 111 | DELIVERY | <?xml version="1.0" encoding="UTF-8"?><Deli... | 2018-03-29 07:00:15 | 2018-03-29 07:00:16 | 2018-03-29 07:00:16 |
| 256 | 111 | RETRIEVAL | <?xml version="1.0" encoding="UTF-8"?><Ret... | NULL | 2018-03-29 07:00:19 | 2018-03-29 07:00:18 |
| 257 | 112 | RELAY REMMD ACCEPTANCE | <?xml version="1.0" encoding="UTF-8"?><Rel... | 2018-03-29 08:44:15 | 2018-03-29 08:44:11 | 2018-03-29 08:44:11 |
| 258 | 112 | DELIVERY | <?xml version="1.0" encoding="UTF-8"?><Deli... | 2018-03-29 08:44:15 | 2018-03-29 08:44:16 | 2018-03-29 08:44:16 |

***Figure 24: Connector 4.0 – example of DOMIBUS_CONNECTOR_EVIDENCE table***

Two columns are helping us to inspect the transmission state of the confirmation message:

- DELIVERED_NAT
- DELIVERED_GW

### 6.3.1. Column DELIVERED_NAT description

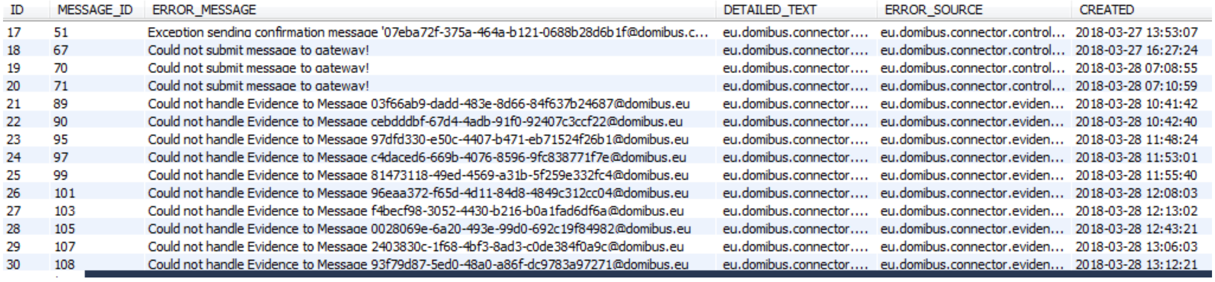
The DELIVERED_NAT column is null by default and will be set to the timestamp when the message is delivered to the national system/connector client. The behaviour of setting this timestamp can be slightly different between the different backends (push/push, push/pull).

### 6.3.2. Column DELIVERED_GW description

The DELIVERED_GW column is null by default and will be set with the current time when the confirmation message is successfully delivered to the gateway.

## 6.4. Search for errors

Every error which can be related to a message is appended to the message and stored into the database. The table DOMIBUS_CONNECTOR_MSG_ERROR (Figure 25) holds these errors.



| ID | MESSAGE_ID | ERROR_MESSAGE | DETAILED_TEXT | ERROR_SOURCE | CREATED |
|---|---|---|---|---|---|
| 17 | 51 | Exception sending confirmation message '07eba72f-375a-464a-b121-0688b28d6b1f@domibus.c... | eu.domibus.connector.... | eu.domibus.connector.control... | 2018-03-27 13:53:07 |
| 18 | 67 | Could not submit message to gateway! | eu.domibus.connector.... | eu.domibus.connector.control... | 2018-03-27 16:27:24 |
| 19 | 70 | Could not submit message to gateway! | eu.domibus.connector.... | eu.domibus.connector.control... | 2018-03-28 07:08:55 |
| 20 | 71 | Could not submit message to gateway! | eu.domibus.connector.... | eu.domibus.connector.control... | 2018-03-28 07:10:59 |
| 21 | 89 | Could not handle Evidence to Message 03f66ab9-dadd-483e-8d66-84f637b24687@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 10:41:42 |
| 22 | 90 | Could not handle Evidence to Message cebdddbf-67d4-4adb-91f0-92407c3ccf22@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 10:42:40 |
| 23 | 95 | Could not handle Evidence to Message 97dfd330-e50c-4407-b471-eb71524f26b1@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 11:48:24 |
| 24 | 97 | Could not handle Evidence to Message c4daced6-669b-4076-8596-9fc838771f7e@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 11:53:01 |
| 25 | 99 | Could not handle Evidence to Message 81473118-49ed-4569-a31b-5f259e332fc4@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 11:55:40 |
| 26 | 101 | Could not handle Evidence to Message 96eaa372-f65d-4d11-84d8-4849c312cc04@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 12:08:03 |
| 27 | 103 | Could not handle Evidence to Message f4becf98-3052-4430-b216-b0a1fad6df6a@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 12:13:02 |
| 28 | 105 | Could not handle Evidence to Message 0028069e-6a20-493e-99d0-692c19f84982@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 12:43:21 |
| 29 | 107 | Could not handle Evidence to Message 2403830c-1f68-4bf3-8ad3-c0de384f0a9c@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 13:06:03 |
| 30 | 108 | Could not handle Evidence to Message 93f79d87-5ed0-48a0-a86f-dc9783a97271@domibus.eu | eu.domibus.connector.... | eu.domibus.connector.eviden... | 2018-03-28 13:12:21 |

***Figure 25: Connector 4.0 – example of DOMIBUS_CONNECTOR_MSG_ERROR table***

The column ERROR_MESSAGE contains a description of the occurred error. The whole stacktrace is stored in the DETAILED_TEXT column. The connector is also printing the connector message id to the logging output, with this information you can follow the message process (see 6.5).

## 6.5. Following the message processing

When the connector is handling a message it assigns a unique connector message id. This message id is also printed out during logging. So this id can be used to follow the message process.

As an example:

The following command on a unix system will show us all log entries related to the message with the connector message id: 03f66ab9-dadd-483e-8d66-84f637b24687@domibus.eu

```
$> cat connector.log | grep msgid=03f66ab9-dadd-483e-8d66-84f637b24687@domibus.eu
```

# I. LIST of FIGURES

# II. LIST of TABLES

# III. LIST of REFERENCES

[1] „ActiveMQ,“ [Online]. Available: http://activemq.apache.org/.

[2] „Spring Commona Application Properties,“ [Online]. Available: https://docs.spring.io/spring-boot/docs/1.5.8.RELEASE/reference/html/common-application-properties.html.

[3] *Required certificates, key- and truststores in the e-Codex environment,* 2018/05.

[4] “Apache Log4j 2,” [Online]. Available: https://logging.apache.org/log4j/2.x/.

[5] [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102600_102699/10264002/02.01.01_60/ts_10264002v020101p.pdf.

[6] „domibusConnectorAPI 4.0.0 RELEASE WSDL,“ [Online]. Available: https://secure.e-codex.eu/nexus/content/repositories/releases/eu/domibus/connector/domibusConnectorAPI/4.0.0-RELEASE/domibusConnectorAPI-4.0.0-RELEASE-wsdl.zip.

[7] „nexus repositroy,“ [Online]. Available: https://secure.e-codex.eu/nexus/content/repositories/releases/eu/domibus.