EUREX

Eurex Clearing

Eurex Clearing Messaging Interfaces Connectivity
E: AMQP Setup & Internals

Date	Rel.	Change
23 May 2022	7.1	Replaced C++ broker with Broker-J (chapter 2.2) Updated list of unsupported SSL and TLS protocol versions (chapter 3) Removed information about temporary queues and AMQP 0-10 protocol (chapter 4.1) Removed statement about flow-control on request queues (chapter 4.1.1) Replaced C++ broker with Broker-J (chapter 5.2) Replaced C++ broker with Broker-J (chapter 5.3)
9 June 2021	8.0	No changes for C7 Release 8.0
7 September 2021	8.1	No change for C7 Release 8.1

Date	Vers.	Change
4 April 2022	1.0	Update on ciphers (chapter 3)

© Eurex 2022

Deutsche Börse AG ("DBAG"), Clearstream Banking AG ("Clearstream"), Eurex Frankfurt AG ("Eurex"), Eurex Clearing AG ("Eurex Clearing"), Eurex Securities Transactions Services GmbH ("Eurex STS") and Eurex Repo GmbH ("Eurex Repo") are corporate entities and are registered under German law. Eurex Global Derivatives AG is a corporate entity and is registered under Swiss law. Clearstream Banking S.A. is a corporate entity and is registered under Luxembourg law. Eurex Frankfurt AG is the administrating and operating institution of Eurex Deutschland. Eurex Deutschland is in the following also referred to as the "Eurex Exchange".

All intellectual property, proprietary and other rights and interests in this publication and the subject matter hereof (other than certain trademarks and service marks listed below) are owned by DBAG or its affiliates and subsidiaries or used under authorization by their respective owners, including, without limitation, all patent, registered design, copyright, trademark and service mark rights. While reasonable care has been taken in the preparation of this publication to provide details that are accurate and not misleading at the time of publication DBAG, Clearstream, Eurex, Eurex Clearing, Eurex Repo as well as the Eurex Exchange and their respective subsidiaries, servants and agents (a) do not make any representations or warranties regarding the information contained herein, whether express or implied, including without limitation any implied warranty of merchantability or fitness for a particular purpose or any warranty with respect to the accuracy, correctness, quality, completeness or timeliness of such information, and (b) shall not be responsible or liable for any third party's use of any information contained herein under any circumstances, including, without limitation, in connection with actual trading or otherwise or for any errors or omissions contained in this publication.
This publication is published for information purposes only and shall not constitute investment advice respectively does not constitute an offer, solicitation or recommendation to acquire or dispose of any investment or to engage in any other transaction. This publication is not intended for solicitation purposes but only for use as general information. All descriptions, examples and calculations contained in this publication are for illustrative purposes only.
Eurex and Eurex Clearing offer services directly to members of the Eurex Exchange respectively to clearing members of Eurex Clearing. Those who desire to trade any products available on the Eurex market or who desire to offer and sell any such products to others or who desire to possess a clearing license of Eurex Clearing in order to participate in the clearing process provided by Eurex Clearing, should consider legal and regulatory requirements of those jurisdictions relevant to them, as well as the risks associated with such products, before doing so.
Only Eurex derivatives that are CFTC-approved may be traded via direct access in the United States or by United States persons. A complete, up-to-date list of Eurex derivatives that are CFTC-approved is available at: https://www.eurex.com/ex-en/rules-regs/eurex-derivatives-us/direct-market-access-from-the-us.
In addition, Eurex representatives and participants may familiarize U.S. Qualified Institutional Buyers (QIBs) and broker-dealers with certain eligible Eurex equity options and equity index options pursuant to the terms of the SEC's July 1,2013 Class No-Action Relief. A complete, up-to-date list of Eurex options that are eligible under the SEC Class No-Action Relief is available at: https://www.eurex.com/ex-en/rules-regs/eurex-derivatives-us/eurex-options-in-the-us-for-eligible-customers. Lastly, U.S. QIBs and broker-dealers trading on behalf of QIBs may trade certain
single-security futures and narrow-based security index futures subject to terms and conditions of the SEC's Exchange Act Release No. 60,194 (June 30, 2009), 74 Fed. Reg. 32,200 (July 7, 2009) and the CFTC's Division of Clearing and Intermediary Oversight Advisory Concerning the Offer and Sale of Foreign Security Futures Products to Customers Located in the United States (June 8, 2010).

Trademarks and Service Marks
Buxl®, DAX®, DivDAX®, eb.rexx®, Eurex®, Eurex Repo®, Strategy Wizard, Euro GC®, FDAX®, FTSE 100, FWB®, GC Pooling®,,GCPI®, MDAX®, ODAX®, SDAX®, TecDAX®, USD GC Pooling®, VDAX®, VDAX-NEW® and Xetra® are registered trademarks of DBAG or its affiliates and subsidiaries. All MSCI indexes are service marks and the exclusive property of MSCI Barra. ATX®, ATX® five, CECE® and RDX® are registered trademarks of Vienna Stock Exchange AG. IPD® UK Quarterly Indexes are registered trademarks of Investment Property Databank Ltd. IPD and have been licensed for the use by Eurex for derivatives. SLI®, SMI® and SMIM® are registered trademarks of SIX Swiss Exchange AG. The STOXX® indexes, the data included therein and the trademarks used in the index names are the intellectual property of STOXX Limited and/or its licensors. Eurex derivatives based on the STOXX® indexes are in no way sponsored, endorsed, sold or promoted by STOXX and its licensors and neither STOXX nor its licensors shall have any liability with respect thereto. PCS® and Property Claim Services® are registered trademarks of ISO Services, Inc. Korea Exchange, KRX, KOSPI and KOSPI 200 are registered trademarks of Korea Exchange Inc. The names of other companies and third-party products may be trademarks or service marks of their respective owners.
FTSE® is a trade mark of the London Stock Exchange Group companies and is used by FTSE International Limited ("FTSE") under license. All rights in the FTSE®100 Index (the "Index") vest in FTSE or its licensors. Neither FTSE nor any of their affiliates or licensors (a) assumes any liability, losses, damages, expenses or obligations in connection with any derivative product based on the Index; or (b) accepts any liability for any errors or omissions, fitness for a particular purpose or the results to be obtained from the use of the Index or related data. No party may rely on the Index or related data contained in this communication which Index and data is owned by FTSE or their affiliates. No use or distribution of the Index is permitted without FTSE's express written consent. FTSE does not promote, sponsor or endorse the content of this communication nor any financial or derivative product that it relates to.
PRIIPs: Eurex Deutschland qualifies as manufacturer of packaged retail and insurance-based investment products (PRIIPs) under Regulation (EU) No 1286/2014 on key information documents for packaged retail and insurance-based investment products (PRIIPs Regulation), and provides key information documents (KIDs) covering PRIIPs traded on Eurex Deutschland on its website under the following link: https://www.eurex.com/ex-en/rules-regs/priips-kids.
In addition, according to Art. 14(1) PRIIPs Regulation the person advising on, or selling, a PRIIP shall provide the KID to retail investors free of charge.

Abstract

This document provides information about how to program AMQP applications for the Eurex Clearing FIXML/FpML Interface. This document is intended to be a guide for Members interested in connecting to either the FIXML, FpML interface, Margin Calculator interface and/or Trade Entry interface.

Keywords

Eurex Clearing FIXML Interface, Eurex Clearing FpML Interface, Eurex Clearing Margin Calculator Interface, Trade Entry Interface, Eurex Clearing, EurexOTC, Advanced Message Queuing Protocol, AMQP, Clearing, FIX, FIXML, FpML

Table of Contents

1	**Introduction**	**7**
1.1	Overview	**7**
1.1.1	Eurex Clearing FIXML Interface	7
1.1.2	Eurex Clearing FpML Interface	7
1.1.3	Eurex Clearing Margin Calculator Interface	7
1.1.4	Eurex Clearing Trade Entry Interface	7
1.1.5	AMQP	7
1.1.6	FIXML	8
1.1.7	FpML	8
1.2	Intended audience	**8**
1.3	Eurex Clearing Messaging Interface Connectivity documentation	**8**
1.4	Eurex Clearing FIXML Interface documentation	**9**
1.5	Eurex Clearing FpML Interface, Margin Calculator Interface and Trade Entry Interface documentation	**9**
1.6	**Conventions used in this document**	**10**
1.7	**Organization of this document**	**10**
2	**The AMQP model**	**11**
2.1	**AMQP 1.0**	**11**
2.2	**Eurex Clearing setup**	**11**
2.3	**Existing AMQP 1.0 limitations on Eurex Clearing interfaces**	**11**
3	**Connecting to the AMQP broker**	**12**
4	**Communication with the AMQP broker**	**14**
4.1	**Requests and responses**	**14**
4.1.1	General approach	15
4.1.1.1	Step 0 - preparation	15
4.1.1.2	Step 1 – creating a receiver	16
4.1.1.3	Step 2 – creating a sender	16
4.1.1.4	Step 3 – sending a request message	16
4.1.1.5	Step 4 – receiving a response message	17
4.1.2	Interface specific information	17
4.1.2.1	Eurex Clearing FIXML Interface	17
4.2	**Broadcasts**	**18**
4.2.1	General approach	18
4.2.1.1	Step 0 – preparation	18
4.2.1.2	Step 1 – creating a broadcast receiver	18
4.2.1.3	Step 2 – receiving broadcast messages	18
4.2.2	Interface specific information	19
4.2.2.1	Eurex Clearing FIXML Interface	20
4.2.2.1.1	Product symbol	21
4.2.2.2	Eurex Clearing FpML Interface	21
4.2.2.3	Eurex Clearing Margin Calculator Interface	22

4.2.2.4	Eurex Clearing Trade Entry Interface	22
5	**Prerequisites for reliable communication**	**23**
5.1	**Acknowledgements / Settlements**	**23**
5.2	**Transactions**	**24**
5.3	**Duplicate detection**	**25**
6	**Glossary of terms and abbreviations**	**26**

1 Introduction

1.1 Overview

1.1.1 Eurex Clearing FIXML Interface

The Eurex Clearing FIXML Interface provides Eurex Clearing Members with a highly flexible, standards compliant and cost-effective way to use Eurex Clearing services. Based on this interface, Members are allowed to choose and deploy their own operating systems and access interfaces.

1.1.2 Eurex Clearing FpML Interface

The Eurex Clearing FpML Interface provides EurexOTC Members with a highly flexible, standards compliant and cost-effective way to use EurexOTC Clear services. Based on this interface, Members are allowed to choose and deploy their own operating systems and access interfaces.

1.1.3 Eurex Clearing Margin Calculator Interface

The Eurex Clearing Margin Calculator Interface provides EurexOTC Members with a highly flexible, standards compliant and cost-effective way to use EurexOTC Clear Margin Calculator service. Based on this interface, Members are allowed to choose and deploy their own operating systems and access interfaces. Messages for the Margin Calculator Interface are based upon and compliant to the widely used FpML standard.

1.1.4 Eurex Clearing Trade Entry Interface

The Eurex Clearing Trade Entry Interface provides EurexOTC Approved Trade sources with a highly flexible, standards compliant and cost-effective way to use EurexOTC Clear services. Based on this interface, Approved Trade sources are allowed to choose and deploy their own operating systems and access interfaces.

1.1.5 AMQP

The Advanced Message Queuing Protocol (AMQP) constitutes the preferred transport layer for delivering messages. AMQP is an open standard with a specific focus on the financial services industry which can be used royalty free. Members can choose the platform and programming language for their client applications. More information is available at the AMQP homepage:

- http://www.amqp.org/

1.1.6 FIXML

Application layer messages on the Eurex Clearing FIXML Interface are based upon and compliant to the widely used FIX standard. FIXML is the XML vocabulary for creating Financial Information eXchange (FIX) protocol messages based on XML.

The Futures Industry Association (FIA)/Futures and Options Association (FOA) initiative for standardized post-trade processing has chosen FIX as the standard communication protocol. More information can be found here:

- http://www.futuresindustry.org/downloads/FIMag/2007/Outlook/Outlook-Standards.pdf

The specification of FIX 5.0 SP2 is provided here:

- https://www.fixtrading.org/family-of-standards/fix-5-0-sp-2/

To learn more about supported FIX/FIXML messages, please refer to “Volume 1: Overview” and volumes 3-5 which are available for download in the public section of the Eurex Clearing website.

1.1.7 FpML

Application layer messages on the Eurex Clearing FpML Interface are based upon and compliant to the widely used FpML standard. FpML – Financial products Markup Language – is the industry standard for complex financial products which is based on XML.

The specification for FpML 5.6 is provided here:

- http://www.fpml.org

To learn more about supported XML/FpML messages, please refer to “Volume 1: Overview” and the documents of Volume 3 which are available for download in the Member Section of the Eurex Clearing website.

1.2 Intended audience

This documentation is intended for system designers and programmers who wish to develop/adapt their client application to interact with the services offered by the Eurex Clearing FIXML Interface, the Eurex Clearing FpML Interface or the Eurex Clearing Margin Calculator Interface.

1.3 Eurex Clearing Messaging Interface Connectivity documentation

The Eurex Clearing FIXML Interface, Eurex Clearing FpML Interface and Margin Calculator share common connectivity documents for AMQP and WebSphere MQ:

- A: Overview
- B: AMQP Programming Guide
- **E: AMQP Setup and Internals (this document)**

All “Eurex Clearing Interfaces – Connectivity” documents are available for download on the Eurex website under the following paths:

For Eurex Clearing’s C7:
https://www.eurex.com/ec-en/
Support > Technology > C7 > Messaging Interfaces Connectivity

Simplified (especially error & exception handling and logging) code examples to provide better overview of the functionality are available for download on GitHub.

- https://github.com/Eurex-Clearing-Messaging-Interfaces

1.4 Eurex Clearing FIXML Interface documentation

The Eurex Clearing FIXML Interface documentation is organized as follows:

- Volume 1: Overview
- Volume 3: Transaction & Position Confirmation
- Volume 4: Transaction & Position Maintenance
- Volume 5: Public Broadcasts
- Volume 6: Message Samples

All documents and the public keys of the AMQP broker are available for download in the public section of the Eurex Clearing website under the following paths:

For Eurex Clearing’s C7:
https://www.eurex.com/ec-en/
Support > Initiatives & Releases > C7 Releases > related release > System documentation > Interfaces

1.5 Eurex Clearing FpML Interface, Margin Calculator Interface and Trade Entry Interface documentation

The Eurex Clearing FpML Interface and Eurex Clearing Margin Calculator Interface documentation is organized as follows:

- Volume 1: Overview
- Volume 3: Trade Notification & Take-Up Confirmation
- Volume 3-A: Post Trade Events
- Volume 3-B: EurexOTC Eurex FpML API for Trade Entry
- Volume 3-C: EurexOTC Clear Margin Calculator Interface

All documents and the public keys of the AMQP brokers are available for download in the Member Section of the Eurex Clearing website under the following path: https://membersection.deutsche-boerse.com/

1.6 Conventions used in this document

- **Cross references** to other chapters within this document are always clickable, but not marked separately.
- **Hyperlinks to websites** are underlined.

1.7 Organization of this document

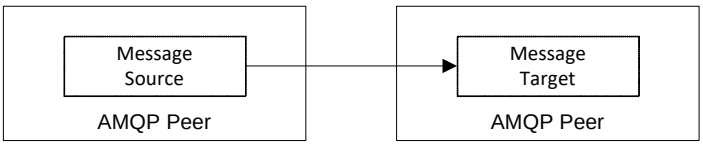
- Chapter 2 – The AMQP Model
    - Gives an overview about how AMQP 1.0 is working
- Chapter 3 – Connecting to the AMQP broker
    - Gives information about requirements needed for connecting to the AMQP brokers
- Chapter 4 - Communication with the AMQP Broker
    - Gives information about how Eurex Clearing is using the AMQP features and describes the required steps for connecting to the AMQP queues
- Chapter 5 – Prerequisites for Reliable Communication
    - Gives information about reliable communication
- Chapter 6 – Glossary of Terms and Abbreviations
    - Glossary of terms and abbreviations used through the document

2 The AMQP model

The Eurex Clearing interfaces supports AMQP 1.0 protocol (ISO 19464).

2.1 AMQP 1.0

AMQP 1.0 protocol (ISO 19464) is very different from the older 0-10 version. It defines symmetric communication between two equal peers. The messages are exchanged between the message sources and targets which exist on the peers.



AMQP 1.0 doesn't require a messaging broker and unlike the AMQP 0-10 it doesn't define the internal structure of the messaging broker (queues, bindings, exchanges).

2.2 Eurex Clearing setup

The Eurex Clearing interfaces are built using Apache Qpid Broker-J software. More information about this implementation can be found on Qpid website:

- http://qpid.apache.org

AMQP 1.0 clients can connect to the broker and use the exchanges and queues as message source.

2.3 Existing AMQP 1.0 limitations on Eurex Clearing interfaces

The AMQP 1.0 support in the current version of the software used by Eurex clearing has several limitations:

- Exactly-once-delivery is not supported
- No support for link stealing in case of connection failure (see Section 2.6.1 of AMQP 1.0 protocol)

There are currently no plans to add support for these features.

3 Connecting to the AMQP broker

The connection to the AMQP broker is established through a standard TCP/IP socket. The information needed to connect to Eurex Clearing's AMQP brokers in the respective environments (specific TCP/IP addresses and port numbers) can be found in the "N7 Network Access Guide". This document is available in the public part of the Eurex Clearing website under the following path:

https://www.eurex.com/ec-en/
Support > Technology > C7 > Network Access

The connection is protected using Transport Layer Security (TLS) protocol and SASL authentication layer. Secure Sockets Layer (SSL) protocols version 2.0 and 3.0 and TLS protocol versions 1.0 and 1.1 are not considered secure and are no longer supported. Both client and server authentication with certificates is used. Eurex Clearing uses server certificates signed by a trusted Certificate Authority.

The cipher suite, which defines the combination of encryption and authentication algorithms used to secure the connection, is negotiated between the client and the server. The following cipher suites are recommended and supported by Eurex Clearing's AMQP brokers[1]:

- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256
- TLS_CHACHA20_POLY1305_SHA256
- ~~SSL_RSA_WITH_3DES_EDE_CBC_SHA~~
- ~~TLS_RSA_WITH_AES_128_CBC_SHA~~
- ~~TLS_RSA_WITH_AES_256_CBC_SHA~~

Support for other cipher suites is not guaranteed.

Both client and server authentication with certificates is used. Eurex Clearing uses server certificates signed by a trusted Certificate Authority. The public keys of the AMQP brokers are available on the website of Eurex Clearing under the following path:

Eurex Clearing FIXML Interface: *https://www.eurex.com/ec-en/*
Support > Technology > C7 > Messaging Interfaces Connectivity

Eurex Clearing FpML Interface: https://membersection.deutsche-boerse.com

Eurex Clearing Margin Calculator Interface: https://membersection.deutsche-boerse.com

In order to make the client application accept the server certificate during the connection establishment, the public keys of the AMQP brokers or the public keys of the CA have to be installed on the client side and the client application has to be able to access it.

[1] The supported cipher suites may be subject to change in the future.

The server name in the certificate should be verified against the IP address from which the server certificate is transferred to the client during the connection establishment. The server name and the IP address have to be added to the client's host file on the machine(s) from where an application connecting to the Eurex Clearing FIXML/FpML/Margin Calculator Interface is executed. This host file is on a Linux/Unix system the file:

/etc/hosts

And on a Microsoft Windows system the file is:

c:\windows\system32\drivers\etc\hosts

When establishing a connection, each client application will verify the identity of the broker using the brokers' public key and the broker will verify the identity of the client using the public key assigned to the account. The Member is logged in if the verification succeeds.

The TLS encryption and the authentication using certificates based on the SASL EXTERNAL authentication mechanism are supported by most provided client libraries. For further details concerning the connection handling and authentication, please see the Apache Qpid and AMQP documentation.

The Member is responsible for securing the transport within his own systems, application and organization.

Every Member is allowed to have up to 10 simultaneous connections per account and 100 simultaneous connections per IP address. Any additional connections will be rejected by the broker. Additionally, only 5 new connections within 10 second timeframe and 20 new connections within 60 second timeframe are allowed. Any further connection will be possible only after the timeframe is over. The time-out parameter for the AMQP handshake is set to 5 seconds. Any connection not built up successfully within this timeframe will be disconnected.[2]

[2] The numbers may be subject to change in the future.

4 Communication with the AMQP broker

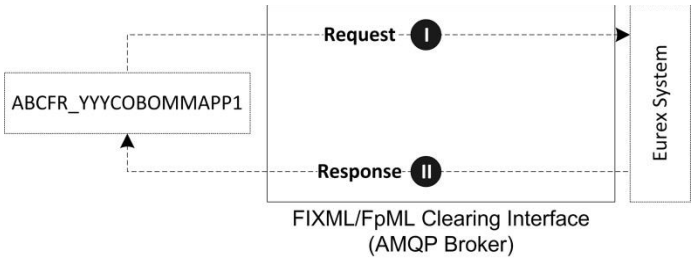
AMQP based Member applications have the possibility to configure the message delivery according to their needs. The Eurex Clearing interfaces supports two different types of communication:

- Request – Response communication.
- Broadcasts.

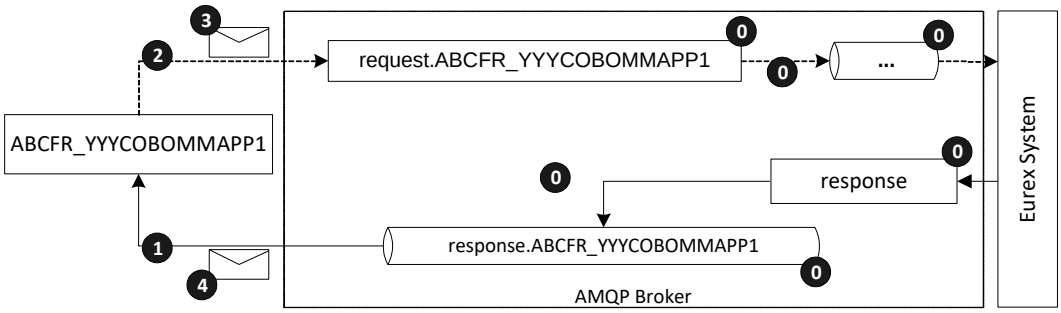
(In the subsequent sections the string "ABCFR" represents a Member ID and "ABCFR_YYYCOBOMMAPP1" represents one of the account name/ID of the Member ABCFR).

4.1 Requests and responses

The Member can trigger an action on the Eurex Clearing System by sending a request to the AMQP broker. The request message is delivered to the Eurex Clearing System, where it is processed. The Eurex Clearing System responds with a message confirming the triggered action.



The predefined response queue is also organized as a ring buffer. When the messages in it reach the queue size limit, the new messages will start overwriting the oldest messages in the queue. As a result, it cannot get full and therefore it is not mandatory for the client applications to consume the messages.



Step	Description	Executed by
0	Exchange and queues prepared by Eurex Clearing	Eurex Clearing System

1	Create receiver on response queue	Member application – initialization
2	Create sender to request exchange	Member application – initialization
3	Send request message	Member application – message processing
4	Receive response message(s)	Member application – message processing

4.1.1 General approach

4.1.1.1 Step 0 - preparation

Step 0 is the preparation which is done by Eurex Clearing during the technical maintenance (see Volume A: Connectivity) of the AMQP broker. Member applications are not involved in this step. During this step, Eurex Clearing prepares the following objects:

- The dedicated exchange for sending requests is created for each Member account (the so called "request exchange"). This exchange is used by the Member application for sending request messages. The exchange is named according to the pattern "request.<AccountID>" (e.g. the request exchange for the Member account `ABCFR_YYYCOBOMMAPP1` will be created under with name "`request.ABCFR_YYYCOBOMMAPP1`").
- The exchange for receiving responses (the so called "response exchange"). This exchange is common for all Members/Member accounts and is named "response".
- The predefined queue for receiving responses. The queue is named according to the pattern "response.<AccountID>" (e.g. for the Member account `ABCFR_YYYCOBOMMAPP1` the queue is named "response`.ABCFR_YYYCOBOMMAPP1`").
- The binding between response queue and the exchange for receiving responses. The binding key is the same as the response queue name.

The response queue is:

- Ring type (Organized as ring buffer – when the queue gets full, the newest messages start overwriting the oldest)
- **For the FIXML Interface:** Limited in size to 6,000 messages and 6,000,000 bytes.[3]
- **For the FpML Interface:** Limited in size to 1,000 messages and 1,000,000 bytes.[4]
- **For the Margin Calculator Interface:** Limited in size to 1,000 messages and 1,000,000 bytes.[5]
- **For the Trade Entry Interface:** Limited in size to 1,000 messages and 1,000,000 bytes.[6]

At the end of every business day, any response messages which were not consumed by the client applications during the business day will be deleted.

[3] The size of queues in messages and bytes may be subject to change in the future.
[4] The size of queues in messages and bytes may be subject to change in the future.
[5] The size of queues in messages and bytes may be subject to change in the future.
[6] The size of queues in messages and bytes may be subject to change in the future.

4.1.1.2 Step 1 – creating a receiver

To be able to receive response messages from the response queue it is required to create a receiver, which will pick up the messages from the response queue and deliver them to the Member's client application. Additionally, receiver can filter messages by correlation ID property which was assigned to a request message.

4.1.1.3 Step 2 – creating a sender

In order to send request message, the Member's application has to create a sender, which will deliver the message to the proper request exchange.

4.1.1.4 Step 3 – sending a request message

After the sender is created, the application can prepare and send the request message. The request message sent to the request exchange has to fulfill the following requirements in order to be processed by the Eurex Clearing System:

- The destination exchange has to be the request exchange according to the Member account.
- Additionally, a Member can also assign the correlation ID property. The correlation ID can contain Member's own identification key of the request message and will be mirrored in all response messages.

As an example, the Member `ABCFR` can send following message using his account `ABCFR_YYYCOBOMMAPP1`:

Message property	***Value***
Destination exchange	request.ABCFR_YYYCOBOMMAPP1
Persistent	False
Message body	FIXML/FpML/Margin Calculator functional message
CorrelationID	123456

There is no limit of requests sent per second. However, the request messages will be routed from the exchange to a queue with a limited size, from which they will be read and processed by the Eurex Clearing System. Due to the limited size and queue type, each Member account can have only a limited number of outstanding request messages: [7]

[7] The size of queues in messages and bytes may be subject to change in the future.

Queue size limit	*FIXML Interface*	*FpML Interface*	*MC Interface*	*TE Interface*
In messages	6,000	1,000	1,000	1,000
In bytes	6,000,000	1,000,000	1,000,000	1,000,000

In case the Member exceeds one of these limits, his next request message will be rejected. Once the Eurex Clearing System has processed some of the outstanding requests, previously rejected requests can be resubmitted.

If necessary, Members can track the number of outstanding requests by matching the correlation IDs of sent requests and received responses.

4.1.1.5 Step 4 – receiving a response message

After a request message is sent to the request exchange, it will be routed into the request queue. The Eurex Clearing System will pick up the message from the request queue and process it. As a result of the request processing, a response message will be prepared. This response message will be sent back to the Member. In case all the previous steps were executed properly, the response message should end up in the Member's response queue ("response.<AccountID>"), where it will be picked up by the receiver (created in step 3).

In case the ring type queues get full, new messages start overwriting the oldest. The overwritten messages are lost. If necessary, Member's application can use the correlation ID property assigned to the message to track received responses and match them with sent requests and to filter messages.

The response messages might in some cases contain also some other properties – for example message id or content type. Unless explicitly mentioned, these properties don't contain any important information. Client applications should be able to deal with these properties (for example by ignoring them). In case the message contains the message ID property, it should not be used for duplicate detection.

4.1.2 Interface specific information

4.1.2.1 Eurex Clearing FIXML Interface

Clearing members with large[8] volumes are advised to send their requests through different accounts and split by product. For example:

- Send all bund like products through *FIXML account 1*
- Send all index products through *FIXML account 2*
- Send all other products through *FIXML account 3*

Response messages will then be sent to response queues of requesting accounts. For example:

- Responses for all bund like products are sent to *FIXML account 1* response queue

[8] Large volume means more than 100.000 requests

- Responses for all index products are sent to *FIXML account 2* response queue
- Responses for all other products are sent to *FIXML account 3* response queue

4.2 Broadcasts

4.2.1 General approach

4.2.1.1 Step 0 – preparation

Step 0 is the preparation which is done by the Eurex Clearing System during the technical maintenance (see Volume A: Connectivity) of the AMQP broker. Member applications are not involved in this step. During this step, the Eurex Clearing System prepares the broadcast queues. For every broadcast stream, each Member account will have a specific broadcast queue prepared. All broadcast messages from the broadcast stream will be automatically delivered into this queue. This queue will be persistent and configured to be able to store all broadcast messages for the whole business day. The broadcast queues will be named according to the following pattern "`broadcast.<MemberAccount>.<BroadcastStream>`" (e.g. for FIXML "`broadcast.ABCFR_YYYCOBOMMAPP1.TradeConfirmation`").

4.2.1.2 Step 1 – creating a broadcast receiver

To read the incoming broadcasts from these queues, a receiver connected to them has to be created. The receiver picks up the messages from the queue and delivers them to the Member application. The receiver for broadcast messages works on the same principle as the receiver for responses to requests.

The receiver can operate in two modes: browsing and consuming. Browsing will read the messages from the queue without removing them after reading whereas the consuming will remove them. Although the browsing mode can be used, Eurex Clearing strongly recommends consuming the messages.

4.2.1.3 Step 2 – receiving broadcast messages

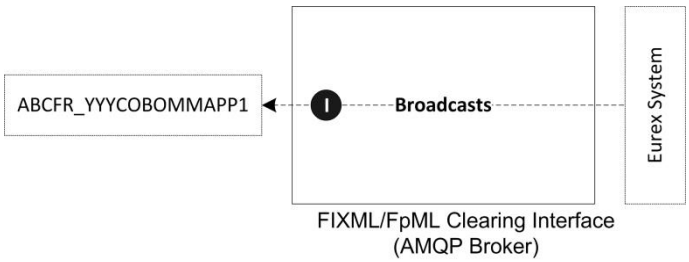
After the message is broadcasted by the Eurex Clearing System, it is delivered to the broadcast queue, where it can be picked up by the receiver created in previous step. If the receiver has been set up properly, it will receive the broadcast messages from the queue.

Broadcast messages on Eurex Clearing FIXML Interface might contain an expiration or time to live field. This field indicates the expiration time of the messages. The expiration time / time to live has no significant meaning for the member applications and can be ignored.

Broadcast messages on all Eurex Clearing interfaces might additionally contain other properties – for example message id or content type. Unless explicitly mentioned, these properties don't contain any important information. Client applications should be able to deal with these properties (for example by ignoring them). In case the message contains the message ID property, it should not be used for duplicate detection.

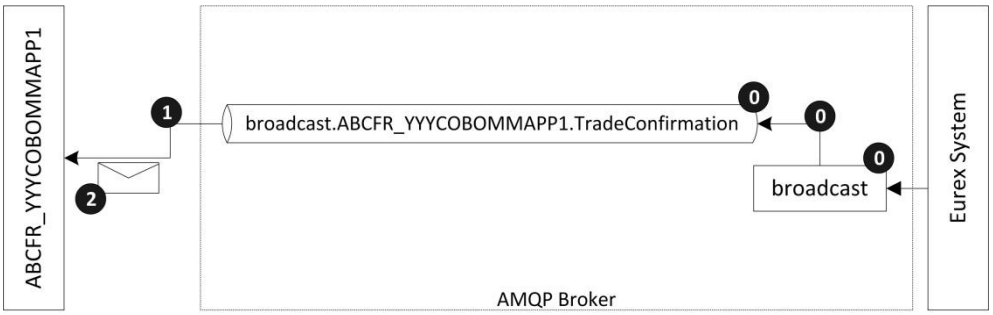
4.2.2 Interface specific information

Broadcast messages are generated and disseminated by the Eurex Clearing System. They are divided into broadcasts streams.



Messages delivered as broadcasts are described in other volumes of this documentation.

It is not guaranteed that the broadcast messages are delivered in the functional chronological order. Receiving broadcasts is similar to receiving responses to requests. It can be done in two simple steps. These steps are displayed on the following diagram using trade confirmation broadcast message and will be described in detail in subsequent chapters.



Step	Description	Executed by
0	Exchange and queues prepared by Eurex	Eurex System
1	Create broadcast receiver	Member application – initialization
2	Receive broadcast messages	Member application – message processing

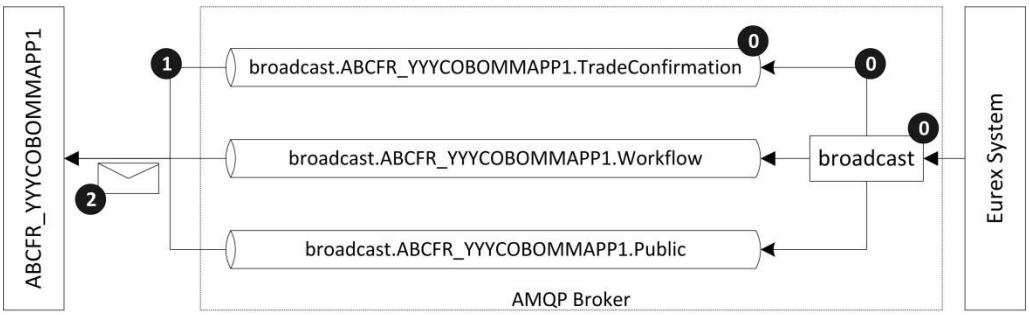
4.2.2.1 Eurex Clearing FIXML Interface

Currently, for the Eurex Clearing FIXML Interface, the following broadcast streams for all Members are provided:[9]

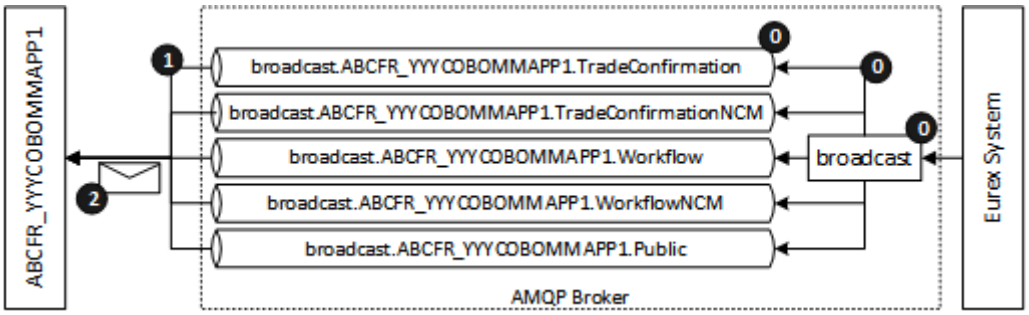
- TradeConfirmation
- Workflow
- TradeConfirmationNCM (provided only for Clearing Members)
- WorkflowNCM (provided only for Clearing Members)
- Public

The TradeConfirmation and Workflow streams contain the messages about Member's own trades and related workflow processes. The TradeConfirmationNCM and the WorkflowNCM streams contain the trade confirmation and workflow messages related to Clearer's Non-Clearing Members. The Public broadcast stream contains public information (e.g. series information or settlement prices). The public data is the same for all Members – clearers as well as Non-Clearing Members. Therefore only one stream for public data is provided for each account.

The following picture shows the streams available for Non-Clearing Members:



The following picture shows the streams available for Clearing Members:



[9] The defined broadcast streams may be subject to change in the future.

On Eurex Clearing FIXML Interface, all broadcast messages are set to expire in 24 hours after they were sent by the Eurex Clearing System. The expired messages will be automatically removed from the broadcast queues and will not be delivered to the member application. Additionally, the broadcast messages contain application property named “BusinessDate” which contains the business date to which the broadcast belongs. The date is in format YYYYMMDD, for example 20160813.

4.2.2.1.1 Product symbol

Each product-related AMQP message in its application properties contains a property named PRODUCT which value will be the *product symbol*. This property will not be present in case of not product-related public broadcast messages or response messages which could not have been validated.

With the product as a property the client application can be modified to discard messages for products that are not relevant for the respective instance.

4.2.2.2 Eurex Clearing FpML Interface

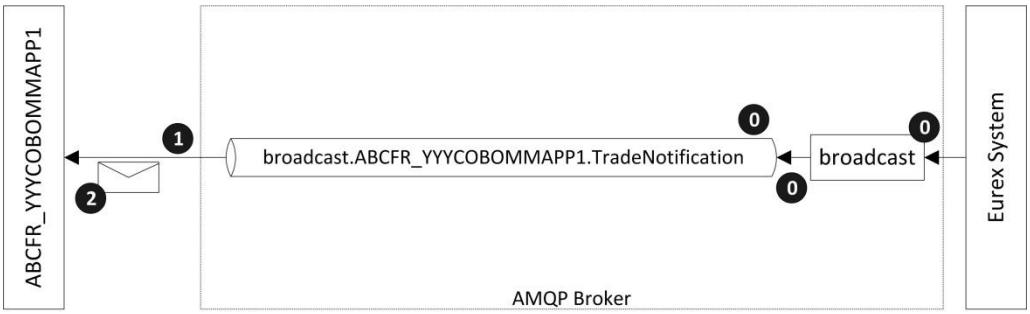
Currently, for the Eurex Clearing FpML Interface, the following broadcast streams for all Members are provided:[10]

- TradeNotification
- TradeNotificationNCM

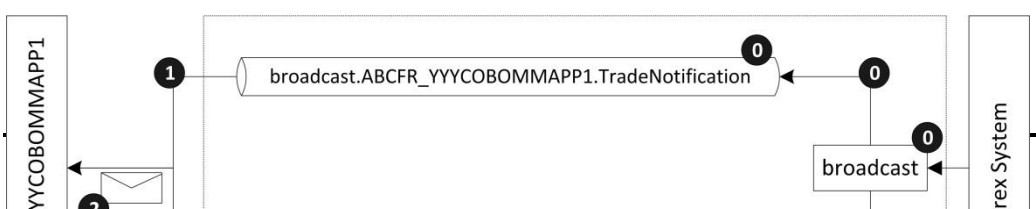
If the trade is booked into the Clearing Member’s account, then all trade notification messages will be sent to the TradeNotification queue of the Clearing Member. If the trade is booked into the Registered Customer’s account, then the trade notification will be sent twice:

- The first message will be sent to the TradeNotificationNCM queue of the Clearing Member of the Registered Customer.
- The second message will be sent to the TradeNotification queue of the Registered Customer.

The following picture shows the streams available for Non-Clearing Members:



The following picture shows the streams available for Clearing Members:



All broadcast messages which were not consumed by the member application will be automatically deleted from the queues at the end of the business day.

4.2.2.3 Eurex Clearing Margin Calculator Interface

Currently, no broadcast streams are provided for the Eurex Clearing Margin Calculator Interface. On this interface, only Request – Response communication is supported.

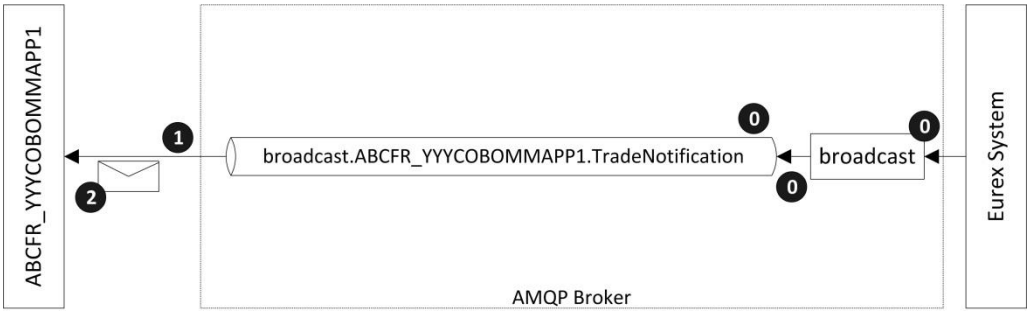
4.2.2.4 Eurex Clearing Trade Entry Interface

Currently, for the Eurex Clearing Trade Entry Interface, the following broadcast stream for all Approved Trade sources is provided:[11]

- TradeNotification

If the trade is booked into the Clearing Member's account, then all trade notification messages will be sent to the TradeNotification queue of the Approved Trade source.

The following picture shows the streams available for Approved Trade sources:



All broadcast messages which were not consumed by the member application will be automatically deleted from the queues at the end of the business day.

[11] The defined broadcast streams may be subject to change in the future.

5 Prerequisites for reliable communication

5.1 Acknowledgements / Settlements

AMQP 1.0 support reliable “at least once” communication.[12] Every message received by the client application needs to be acknowledged (settled) to the broker. Until it is acknowledged, it still remains on the broker. The message is removed only after the acknowledgement is received by the broker. During the time between the delivery to the client application and the acknowledgement, the broker does not deliver this message to any other clients connected to the same queue. In case the original receiver fails before acknowledging the message, it is made available to other clients reading the same queue.

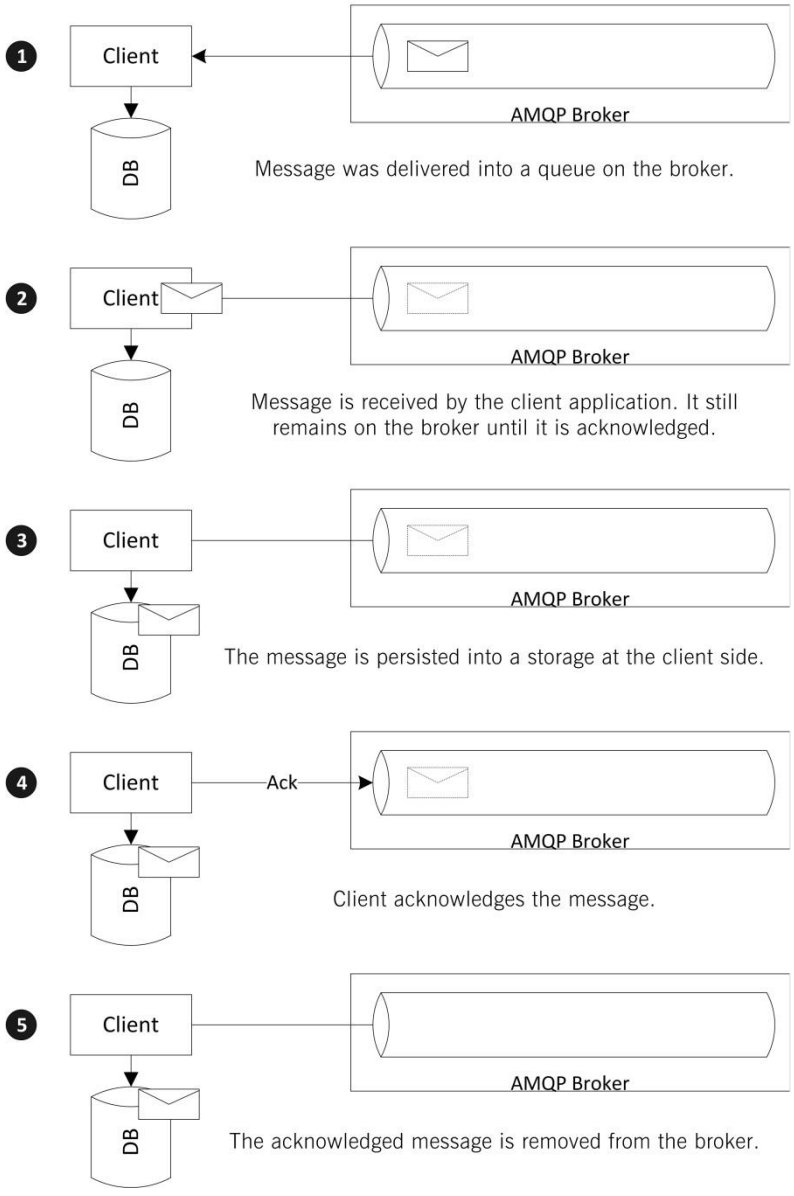
Most AMQP client libraries support two acknowledgement modes:

1. Automatic acknowledgements
2. Manual acknowledgements

When using the automatic acknowledgements outside transactions, the client library automatically acknowledges the message after it is received from the broker. It does not wait until the message is processed or stored on the client side. Therefore outside a transaction a message loss may occur.

The manual mode leaves the acknowledging of the received messages to the broker to the client application instead of doing it automatically by the library. That allows processing of received message (e.g. committing it into a database) before acknowledging and thus ensures reliability.

[12] While AMQP 1.0 protocol supports exactly-one delivery mode, it is currently not implemented in the software used by Eurex Clearing. See the chapter 2.3 for more information about the existing limitations.



5.2 Transactions

The Apache Qpid Broker-J offers support for local transactions on AMQP 1.0 protocols. XA / distributed transactions are not supported.[13] The transactions have the advantage over acknowledgements when there is a need to process multiple messages completely or not at all. For processing individual messages one by one, the transactions do not offer increased reliability compared to acknowledgements. The use of transactions has a significant impact on the performance compared to acknowledgements and in general is not recommended.

[13] XA / distributed transactions are not supported by the AMQP 1.0 protocol. Currently, there are no plans to support them in the future.

5.3 Duplicate detection

The Apache Qpid Broker-J broker provides only at least once reliability for AMQP 1.0. It is able to guarantee that all messages are delivered to the client. But duplicates can appear during the message delivery (especially in case any failovers or reconnects occurred during the communication). It is a responsibility of the client to detect the duplicates. The duplicates can be usually detected from the message payload. Even in case the message contains the message ID property, it should not be used for duplicate detection.

The exact way how to construct unique identification for each message is described in the specification of the different interfaces. Chapters 1.4 and 1.5 and contain the details about the documentation of our interfaces.

6 Glossary of terms and abbreviations

Term / Abbr.	*Definition*
AMQP	Advanced Message Queuing Protocol - standard for Messaging Middleware.
Apache Qpid	Open source implementation of AMQP protocol
Binding	A binding is a relationship between a message queue and an exchange. The binding specifies routing arguments that tell the exchange which messages the queue should get.
Broker	AMQP middleware messaging server
Browsing	AMQP queue access type; reading messages from queues, when the message stays in the queue even after it is read
Consuming	AMQP queue access type; reading messages from queues, when the message is removed from the queue after it is read
Eurex System	Eurex hosts
Exchange	An exchange accepts messages from a producer application and routes them to message queues according to prearranged criteria.
EXTERNAL authentication	AMQP authentication mechanism based on SSL certificates
FIX	The Financial Information Exchange Protocol
FIXML	FIX business messages in XML syntax
Message	A message is the atomic unit of routing and queuing. Messages have a header consisting of a defined set of properties, and a body that is an opaque block of binary data.
Queue	A message queue stores messages in memory or on disk and delivers these in sequence to one or more consumer applications. Message queues are message storage and distribution entities. Each message queue is entirely independent.
Routing key	A message property used in bindings to specify the exchange – queue relationship.
SASL	Simple Authentication and Security Layer
SSL	Secure Sockets Layer – cryptographic protocol designed to provide communication security over the Internet.
TLS	Transport Layer Security – cryptographic protocol designed to provide communication security over the Internet and successor to SSL protocol.