**Eurex Clearing**

# Eurex Clearing Messaging Interfaces Connectivity
# A: Connectivity

**Document version C7 Release 8.0**

| Date | Vers. | Change |
|---|---|---|
| 6 September 2021 | 7.1 | Updated list of unsupported SSL and TLS protocol versions (chapter 4.2) |
| 9 June 2021 | 8.0 | No changes for C7 Release 8.0 |

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 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.

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®, Eurex Strategy WizardSM, Euro GC Pooling®, FDAX®, 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.

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 the connectivity for the Eurex Clearing FIXML/FpML/Margin Calculator Interface. This document is intended to be a guide for Members interested in connecting to either the FIXML, FpML and/or Margin Calculator interface.

**Keywords**

Eurex Clearing FIXML Interface, Eurex Clearing FpML Interface, Eurex Clearing Margin Calculator Interface, Eurex Clearing, EurexOTC, Advanced Message Queuing Protocol, AMQP, WebSphere MQ, Clearing, FIX, FIXML, FpML

## Table of Contents

# 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 may 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 WebSphere MQ

WebSphere MQ is a message oriented middleware provided by IBM and can be used as an alternative to AMQP for the delivery of messages. More information about WebSphere MQ can be found on the webpages of IBM:

- http://www-03.ibm.com/software/products/en/websphere-mq

### 1.1.7 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.

### 1.1.8 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 document is intended for IT professionals and decision takers who have to know how to connect their in-house infrastructure and applications to 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 (this document)**
- B: AMQP Programming Guide
- E: AMQP Setup and Internals

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 > Initiatives & Releases > C7 Releases > C7 Release 8.0 > System documentation > Interfaces*

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 > C7 Release 8.0 > 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 broker are available for download in the Member Section of the Eurex Clearing website under the following path:

*https://www.eurex.com/ec-en/ -> Tech -> EurexOTC Clear -> System documentation -> Interfaces*

## 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.

Changes applied to this document after the last version has been published (other than grammar/spelling corrections) are marked with a change bar in the left margin as demonstrated in this paragraph. Old change bars will be removed from version to version.

## 1.7 Organization of this document

- Chapter 2 – Infrastructure Requirements
    - Information about the technical connection needed for the new service
- Chapter 3 – Transport Layer
    - Overview of the transport layers available
- Chapter 4 - AMQP
    - Describes how AMQP is working as a transport layer
- Chapter 5 – WebSphere MQ
    - Describes how Eurex Clearing is using WebSphere MQ
- Chapter 6 – Setup for outsourced Back Offices
    - Outlines the proposed setup solution for insourcing firms
- Chapter 7 – Glossary of Terms and Abbreviations
    - Glossary of terms and abbreviations used through the document

## 2 Infrastructure requirements

To access the Eurex Clearing FIXML Interface or the Eurex Clearing FpML Interface, one of following connection types is required:

- AMQP
    - Eurex Multi-Interface Channel connection on a leased line or via VPN
    - Eurex FIX Channel on a Eurex Consolidated Connection
- WebSphere MQ
    - z/OS connection to Deutsche Börse Group. Existing z/OS connections may be re-used.

To access the Eurex Clearing Margin Calculator Interface one of following connection types is required:

- AMQP
    - Risk Data Channel connection on a leased line or via VPN.
- WebSphere MQ
    - z/OS connection to Deutsche Börse Group. Existing z/OS connections may be re-used.

z/OS connections usually have a lower bandwidth than the Multi-Interface channel. The z/OS connection currently supports 64kBit/s, 128kBit/s, 256kBit/s, 512kBit/s or 1Mbit/s. The minimum bandwidth for the Multi Interface Channel or the Risk Data Channel is 1Mbit/s. The pricing for the connections is based on the bandwidth ordered.

Network related information for the AMQP connection of the Eurex Clearing FIXML Interface, Eurex Clearing FpML Interface and the Eurex Clearing Margin Calculator can be found in the “Eurex Exchange and Eurex Clearing Network Access Manual”. This document is available in the public section of the Eurex Clearing website under the following path:

*https://www.eurex.com/ec-en/*
*Support > Initiatives & Releases > C7 Releases > C7 Release 8.0 > Network Access*

Please note that the server certificates and the network addresses for the AMQP broker for the Eurex Clearing FIXML Interface, the Eurex Clearing FpML Interface and the Eurex Clearing Margin Calculator Interface differ.

Network related information for the WebSphere MQ connection is made available during the setup process for WebSphere MQ connections.

### 2.1 Bandwidth requirement estimation

Eurex Clearing highly recommends to ensure that the bandwidth of the used lines for connections to Eurex Clearing have enough spare resources in order to be able to process all messages in a reasonable time-frame on very high volume days. It is recommended to order increased line capacity in time.

### 2.1.1 Eurex Clearing FIXML Interface

#### 2.1.1.1 Trade confirmations, workflow broadcasts and requests & responses

The bandwidth requirements for using the Eurex Clearing FIXML Interface can be calculated based on the number of messages sent/delivered per business day and the size of the messages. The size of the functional messages can be calculated using their FIXML layouts, which can be found in the other volumes of this documentation.

The following estimations are based on the following assumptions:

- All figures for the required bandwidth calculation assume that 100,000 messages are sent or received.
- The average size of a message is 1,100 Bytes long. This applies to Trade Confirmations, which are assumed to have the highest share of the messages, as well as for other messages which are currently of approximately the same size.
- The header for each AMQP message is assumed to be 128 Bytes.
- The header for each WebSphere MQ message is assumed to be 500 Bytes.
- The protocol overhead for AMQP and WebSphere MQ are considered to be negligible. The protocol overhead, such as heartbeats, are mainly interchanged during times when no messages are interchanged and this causes no additional workload in times of message sending. Additionally, these messages are rather small.
- The protocol overhead for TCP/IP (e.g. confirmations, re-sending of packages, additional network packages due to package fragmentation, ...) is assumed to be 10%.

Then for AMQP as a transport layer, the following calculation can be made:

| | |
|---|---|
| Average message size | 1,100 Bytes |
| Number of messages | 100,000 |
| Total data amount (Bytes) | (1,100 Bytes + 128 Bytes) * 100,000 = 122,800,000 Bytes |
| Total data amount (Bits) | 982,400,000 Bits |
| Total data amount with IP overhead | 1,080,640,000 Bits |
| Receive all messages within 12 hours | 1,080,640,000 Bits / (12 * 3,600 s) ~= **25 kbit/s** |
| Receive all messages within 8 hours | 1,080,640,000 Bits / (8 * 3,600 s) ~= **38 kbit/s** |
| Receive all messages within 4 hours | 1,080,640,000 Bits / (4 * 3,600 s) ~= **76 kbit/s** |
| Receive all messages within 1 hour | 1,080,640,000 Bits / 3,600 s ~= **300 kbit/s** |
| Receive all messages within 30 minutes | 1,080,640,000 Bits / 1,800 s ~= **600 kbit/s** |
| Messages **per second** in 64kBit/s line | 64,000 / (1.1 * 1,228 * 8) ~= **6** |
| Messages **per second** in 128kBit/s line | 128,000 / (1.1 * 1,228 * 8) ~= **12** |
| Messages **per second** in 256kBit/s line | 256,000 / (1.1 * 1,228 * 8) ~= **24** |

| | |
|---|---|
| Messages **per second** in 512kBit/s line | 512,000 / (1.1 * 1,228 * 8) ~= **47** |
| Messages **per second** in 1Mbit/s line | 1,024,000 / (1.1 * 1,228 * 8) ~= **95** |

The values for 64kBits/s, 128kBit/s, 256kBit/s and 512kBit/s are listed for comparison with the values of the WebSphere MQ connectivity.

For WebSphere MQ, the following calculation applies:

| | |
|---|---|
| Average message size | 1,100 Bytes |
| Number of messages | 100,000 |
| Total data amount (Bytes) | (1,100 Bytes + 500 Bytes) * 100,000 = 160,000,000 Bytes |
| Total data amount (Bits) | 1,280,000,000 Bits |
| Total data amount with IP overhead | 1,408,000,000 Bits |
| Receive all messages within 12 hours | 1,408,000,000 Bits / (12 * 3,600 s) ~= **33 kbit/s** |
| Receive all messages within 8 hours | 1,408,000,000 Bits / (8 * 3,600 s) ~= **49 kbit/s** |
| Receive all messages within 4 hours | 1,408,000,000 Bits / (4 * 3,600 s) ~= **98 kbit/s** |
| Receive all messages within 1 hour | 1,408,000,000 Bits / 3,600 s ~= **391 kbit/s** |
| Receive all messages within 30 minutes | 1,408,000,000 Bits / 1,800 s ~= **782 kbit/s** |
| Messages **per second** in 64kBit/s line | 64,000 / (1.1 * 1,600 * 8) ~= **5** |
| Messages **per second** in 128kBit/s line | 128,000 / (1.1 * 1,600 * 8) ~= **9** |
| Messages **per second** in 256kBit/s line | 256,000 / (1.1 * 1,600 * 8) ~= **18** |
| Messages **per second** in 512kBit/s line | 512,000 / (1.1 * 1,600 * 8) ~= **36** |
| Messages **per second** in 1Mbit/s line | 1,024,000 / (1.1 * 1,600 * 8) ~= **72** |

Based on similar calculations, each Member is able to calculate his own rough bandwidth requirements.

#### 2.1.1.2 Public broadcasts

The Eurex Clearing System sends public information as broadcast streams, such as End-of-Assignment, Capital Adjustments, Contract Changes and Settlement Prices. The data is sent once for each account.[1] This information is sent throughout the whole day. However, there are certain times during the day at which usually load peaks for the disseminated public information occur:

- Between 18:30 and 19:00. During this time, the main peak for the public broadcast data occurs.[2]

[1] Clearing Members will receive the public information also once for each account. The number of Non-Clearing Members served by the Clearing Member is not affecting the amount of data sent.

[2] All times in this document are in CET/CEST.

- Between 23:30 and 24:00.

The amount of data sent is dependent on several factors:

- The number of products set up.
- The number of contracts available in the Eurex Clearing System.
- Number of expiring contracts for the current business day.

Other factors may apply as well.

In order to give estimation for the bandwidth requirement for the public broadcasts, the data for Friday, November 16, 2012 is used. On that day, 3,668 messages with a total of 10,670,652 Bytes have been sent within a 2 minute period between 18:30 and 19:00 as public broadcasts from the Eurex Clearing System. In order to estimate bandwidth requirements, the following assumptions are made:

- The header for each AMQP message is assumed to be 128 Bytes.
- The header for each WebSphere MQ message is assumed to be 500 Bytes.
- The protocol overhead for AMQP and WebSphere MQ are considered to be negligible. The protocol overhead, such as heartbeats, are mainly interchanged during times when no messages are interchanged and this causes no additional workload in times of message sending. Additionally, these messages are rather small.
- The protocol overhead for TCP/IP (e.g. confirmations, re-sending of packages, additional network packages due to package fragmentation, ...) is assumed to be 10%.

Then the following calculation can be made for AMQP as transport layer:

| | |
|---|---|
| Total FIXML Message Size | 10,670,652 Bytes |
| Number of messages | 3,668 |
| Total data amount (Bytes) | 10,670,652 Bytes + 3,668 * 128 = 11,140,156 Bytes |
| Total data amount (Bits) | 89,121,248 Bits |
| Total data amount with IP overhead | 98,033,373 Bits |
| Download time on 64kBit/s line | 98,033,373 / 64,000 = **1,532 s** ~= **26 min** |
| Download time on 128kBit/s line | 98,033,373 / 128,000 = **766 s** ~= **13 min** |
| Download time on 256kBit/s line | 98,033,373 / 256,000 = **383 s** ~= **7 min** |
| Download time on 512kBit/s line | 98,033,373 / 512,000 = **192 s** ~= **3 min** |
| Download time on 1Mbit/s line | 98,033,373 / 1,024,000 = **96 s** ~= **1.5 min** |

The values for 64kBits/s, 128kBit/s, 256kBit/s and 512kBit/s are listed for comparison with the values of the WebSphere MQ connectivity.

For WebSphere MQ, the following calculation applies:

| | |
|---|---|
| Total FIXML Message Size | 10,670,652 Bytes |
| Number of messages | 3,668 |

| | |
|---|---|
| Total data amount (Bytes) | 10,670,652 Bytes + 3,668 * 500 = 12,504,652 Bytes |
| Total data amount (Bits) | 100,037,216 Bits |
| Total data amount with IP overhead | 110,040,938 Bits |
| Download time on 64kBit/s line | 110,040,938 / 64,000 = **1,720 s** ~= **29 min** |
| Download time on 128kBit/s line | 110,040,938 / 128,000 = **860 s** ~= **15 min** |
| Download time on 256kBit/s line | 110,040,938 / 256,000 = **430 s** ~= **7 min** |
| Download time on 512kBit/s line | 110,040,938 / 512,000 = **215 s** ~= **4 min** |
| Download time on 1Mbit/s line | 110,040,938 / 1,024,000 = **108 s** ~= **2 min** |

Please note, that these transfer times are only valid on an exclusive usage of the connectivity just for receiving these broadcasts. In reality, the transfer times will be increased due to the fact that other FIXML/FpML messages and data for other services are sent in parallel on the same technical connection. During these times, unexpected delays may be expected for other messages or services if the used bandwidth is too small.

### 2.1.2 Eurex Clearing FpML Interface

The bandwidth requirements for using the Eurex Clearing FpML Interface can be calculated based on the number of messages sent/delivered per business day and the size of the messages. The size of the functional messages can be calculated using their FpML layouts, which can be found in other volumes of this documentation.

The following estimations are based on the following assumptions:

- All figures for the required bandwidth calculation assume that 100 messages are sent or received.
- The average size of a message is 10,000 Bytes long.[3]
- The header for each AMQP message is assumed to be 128 Bytes.
- The header for each WebSphere MQ message is assumed to be 500 Bytes.
- The protocol overhead for AMQP and WebSphere MQ are considered to be negligible. The protocol overhead, such as heartbeats, are mainly interchanged during times when no messages are interchanged and this causes no additional workload in times of message sending. Additionally, these messages are rather small.
- The protocol overhead for TCP/IP (e.g. confirmations, re-sending of packages, additional network packages due to package fragmentation) is assumed to be 10%.

Then for AMQP as a transport layer, the following calculation can be made:

| | |
|---|---|
| Average message size | 10,000 Bytes |
| Number of messages | 100 |

[3] The value references to the standard Swap. e.g., the XML document containing Variable Swap with tenor 50 years and payment frequency 1 month will have the size of ca. 206 Bytes

| | |
|---|---|
| Total data amount (Bytes) | (10,000 Bytes + 128 Bytes) * 100 = 1,012,800 Bytes |
| Total data amount (Bits) | 8,102,400 Bits |
| Total data amount with IP overhead | 8,912,640 Bits |
| Receive all messages within 12 hours | 8,912,640 Bits / (12 * 3,600 s) ~= **0.2 kbit/s** |
| Receive all messages within 8 hours | 8,912,640 Bits / (8 * 3,600 s) ~= **0.3 kbit/s** |
| Receive all messages within 4 hours | 8,912,640 Bits / (4 * 3,600 s) ~= **0.7 kbit/s** |
| Receive all messages within 1 hour | 8,912,640 Bits / 3,600 s ~= **2.5 kbit/s** |
| Receive all messages within 30 minutes | 8,912,640 Bits / 1,800 s ~= **5.0 kbit/s** |
| Messages **per minute** in 64kBit/s line | 60 * 64,000 / (1.1 * 10,128 * 8) ~= **43** |
| Messages **per minute** in 128kBit/s line | 60 * 128,000 / (1.1 * 10,128 * 8) ~= **86** |
| Messages **per minute** in 256kBit/s line | 60 * 256,000 / (1.1 * 10,128 * 8) ~= **172** |
| Messages **per minute** in 512kBit/s line | 60* 512,000 / (1.1 * 10,128 * 8) ~= **344** |
| Messages **per minute** in 1Mbit/s line | 60* 1,024,000 / (1.1 * 10,128 * 8) ~= **689** |

The values for 64kBits/s, 128kBit/s, 256kBit/s and 512kBit/s are listed for comparison with the values of the WebSphere MQ connectivity.

For WebSphere MQ, the following calculation applies:

| | |
|---|---|
| Average message size | 10,000 Bytes |
| Number of messages | 100 |
| Total data amount (Bytes) | (10,000 Bytes + 500 Bytes) * 100 = 1,050,000 Bytes |
| Total data amount (Bits) | 8,400,000 Bits |
| Total data amount with IP overhead | 9,240,040 Bits |
| Receive all messages within 12 hours | 9,240,040 Bits / (12 * 3,600 s) ~= **0.2 kbit/s** |
| Receive all messages within 8 hours | 9,240,040 Bits / (8 * 3,600 s) ~= **0.3 kbit/s** |
| Receive all messages within 4 hours | 9,240,040 Bits / (4 * 3,600 s) ~= **0.7 kbit/s** |
| Receive all messages within 1 hour | 9,240,040 Bits / 3,600 s ~= **2.6 kbit/s** |
| Receive all messages within 30 minutes | 9,240,040 Bits / 1,800 s ~= **5.1 kbit/s** |
| Messages **per minute** in 64kBit/s line | 60 * 64,000 / (1.1 * 10.500 * 8) ~= **42** |
| Messages **per minute** in 128kBit/s line | 60 * 128,000 / (1.1 * 10.500 * 8) ~= **83** |
| Messages **per minute** in 256kBit/s line | 60 * 256,000 / (1.1 * 10.500 * 8) ~= **166** |
| Messages **per minute** in 512kBit/s line | 60* 512,000 / (1.1 * 10.500 * 8) ~= **332** |

| | |
|---|---|
| Messages **per minute** in 1Mbit/s line | 60* 1,024,000 / (1.1 * 10.500 * 8) ~= **665** |

Based on similar calculations, each Member is able to calculate his own rough bandwidth requirements.

**Note:** z/OS connections usually have simulation and production on the same line. It is not possible to "reserve" a certain bandwidth for production. Therefore, it has to be kept in mind that excessive usage of simulation, e.g. for performance tests, can have a negative impact on the available bandwidth for production.

### 2.1.3 Eurex Clearing Margin Calculator Interface

The message sizes and bandwidth to be expected are rather difficult to estimate because the message sizes are very dependent on the content of the messages sent:

- Interest Rate Swap (IRS) messages depend mainly, but not only, on the number of trades
- Variable Notable Swaps (VNS) messages depend mainly, but not only, on the scheduling of the VNS.

The bandwidth estimations are based on the following assumptions:

- All figures for the bandwidth estimation assume a portfolio of 100 swaps containing 90 Interest Rate swaps and 10 Variable Note swaps.
- It is assumed that such a portfolio is sent 100 times a day.
- The estimated average size of an Interest Rate swap request message is 5,750 Bytes long.
- The estimated average size of an Variable Note swap request message is 10,000 Bytes long.
- The estimated average size of a Interest Rate swap response message is 300 Bytes long.
- The estimated average size of a Variable Note swap response message is 300 Bytes long.
- The header for each AMQP message is assumed to be 128 Bytes.
- The header for each WebSphere MQ message is assumed to be 500 Bytes.
- The protocol overhead for AMQP and WebSphere MQ are considered to be negligible. The protocol overhead, such as heartbeats, are mainly interchanged during times when no messages are interchanged and this causes no additional workload in times of message sending. Additionally, these messages are rather small.
- The protocol overhead for TCP/IP (e.g. confirmations, re-sending of packages, additional network packages due to package fragmentation) is assumed to be 10%.

Then for AMQP as a transport layer, the following calculation can be made:

| | |
|---|---|
| Average message size | 5,750 Bytes * 90 + 10,000 Bytes * 10 + 128 Bytes = 617,628 Bytes |
| Number of messages | 100 |
| Total data amount (Bytes) | 617,628 Bytes * 100 = 61,762,800 Bytes |
| Total data amount (Bits) | 494,102,400 Bits |
| Total data amount with IP overhead | 543,512,640 Bits |
| Data mount of 1 message | 617,628 Bytes * 8 * 1.1 = 5,435,127 Bits |

| | |
|---|---|
| Perform all calculations within 12 hours | 543,512,640 Bits / (12 * 3,600 s) ~= **13 kbit/s** |
| Perform all calculations within 8 hours | 543,512,640 Bits / (8 * 3,600 s) ~= **19 kbit/s** |
| Perform all calculations within 4 hours | 543,512,640 Bits / (4 * 3,600 s) ~= **38 kbit/s** |
| Perform all calculations within 1 hour | 543,512,640 Bits / 3,600 s ~= **151 kbit/s** |
| Perform all calculations within 30 minutes | 543,512,640 Bits / 1,800 s ~= **302 kbit/s** |
| Upload of 1 message on 64 kbit/s line | 5,435,127 Bits / 64,000 = **85 seconds** |
| Upload of 1 message on 128 kbit/s line | 5,435,127 Bits / 128,000 = **43 seconds** |
| Upload of 1 message on 256 kbit/s line | 5,435,127 Bits / 256,000 = **22 seconds** |
| Upload of 1 message on 512 kbit/s line | 5,435,127 Bits / 512,000 = **11 seconds** |
| Upload of 1 message on 1 Mbit/s line | 5,435,127 Bits / 1,024,000 = **6 seconds** |
| Upload of 1 message on 2 Mbit/s line | 5,435,127 Bits / 2,048,000 = **3 seconds** |
| Upload of 1 message on 4 Mbit/s line | 5,435,127 Bits / 4,096,000 = **2 seconds** |
| Upload of 1 message on 5 Mbit/s line | 5,435,127 Bits / 5,120,000 = **1 second** |

### 2.1.4 Eurex Clearing Trade Entry Interface

The bandwidth requirements for using the Eurex Clearing Trade Entry Interface can be calculated based on the number of messages sent/delivered per business day and the size of the messages. The size of the functional messages can be calculated using their FpML layouts, which can be found in other volumes of this documentation.

The following estimations are based on the following assumptions:

- All figures for the required bandwidth calculation assume that 100 messages are sent or received.
- The average size of a message is 10,000 Bytes long.[4]
- The header for each AMQP message is assumed to be 128 Bytes.
- The header for each WebSphere MQ message is assumed to be 500 Bytes.
- The protocol overhead for AMQP and WebSphere MQ are considered to be negligible. The protocol overhead, such as heartbeats, are mainly interchanged during times when no messages are interchanged and this causes no additional workload in times of message sending. Additionally, these messages are rather small.
- The protocol overhead for TCP/IP (e.g. confirmations, re-sending of packages, additional network packages due to package fragmentation) is assumed to be 10%.

Then for AMQP as a transport layer, the following calculation can be made:

| | |
|---|---|
| Average message size | 10,000 Bytes |

[4] The value references to the standard Swap. e.g., the XML document containing Variable Swap with tenor 50 years and payment frequency 1 month will have the size of ca. 206 Bytes

| | |
|---|---|
| Number of messages | 100 |
| Total data amount (Bytes) | (10,000 Bytes + 128 Bytes) * 100 = 1,012,800 Bytes |
| Total data amount (Bits) | 8,102,400 Bits |
| Total data amount with IP overhead | 8,912,640 Bits |
| Receive all messages within 12 hours | 8,912,640 Bits / (12 * 3,600 s) ~= **0.2 kbit/s** |
| Receive all messages within 8 hours | 8,912,640 Bits / (8 * 3,600 s) ~= **0.3 kbit/s** |
| Receive all messages within 4 hours | 8,912,640 Bits / (4 * 3,600 s) ~= **0.7 kbit/s** |
| Receive all messages within 1 hour | 8,912,640 Bits / 3,600 s ~= **2.5 kbit/s** |
| Receive all messages within 30 minutes | 8,912,640 Bits / 1,800 s ~= **5.0 kbit/s** |
| Messages **per minute** in 64kBit/s line | 60 * 64,000 / (1.1 * 10,128 * 8) ~= **43** |
| Messages **per minute** in 128kBit/s line | 60 * 128,000 / (1.1 * 10,128 * 8) ~= **86** |
| Messages **per minute** in 256kBit/s line | 60 * 256,000 / (1.1 * 10,128 * 8) ~= **172** |
| Messages **per minute** in 512kBit/s line | 60* 512,000 / (1.1 * 10,128 * 8) ~= **344** |
| Messages **per minute** in 1Mbit/s line | 60* 1,024,000 / (1.1 * 10,128 * 8) ~= **689** |

The values for 64kBits/s, 128kBit/s, 256kBit/s and 512kBit/s are listed for comparison with the values of the WebSphere MQ connectivity.

For WebSphere MQ, the following calculation applies:

| | |
|---|---|
| Average message size | 10,000 Bytes |
| Number of messages | 100 |
| Total data amount (Bytes) | (10,000 Bytes + 500 Bytes) * 100 = 1,050,000 Bytes |
| Total data amount (Bits) | 8,400,000 Bits |
| Total data amount with IP overhead | 9,240,040 Bits |
| Receive all messages within 12 hours | 9,240,040 Bits / (12 * 3,600 s) ~= **0.2 kbit/s** |
| Receive all messages within 8 hours | 9,240,040 Bits / (8 * 3,600 s) ~= **0.3 kbit/s** |
| Receive all messages within 4 hours | 9,240,040 Bits / (4 * 3,600 s) ~= **0.7 kbit/s** |
| Receive all messages within 1 hour | 9,240,040 Bits / 3,600 s ~= **2.6 kbit/s** |
| Receive all messages within 30 minutes | 9,240,040 Bits / 1,800 s ~= **5.1 kbit/s** |
| Messages **per minute** in 64kBit/s line | 60 * 64,000 / (1.1 * 10.500 * 8) ~= **42** |
| Messages **per minute** in 128kBit/s line | 60 * 128,000 / (1.1 * 10.500 * 8) ~= **83** |
| Messages **per minute** in 256kBit/s line | 60 * 256,000 / (1.1 * 10.500 * 8) ~= **166** |

| | |
|---|---|
| Messages **per minute** in 512kBit/s line | 60* 512,000 / (1.1 * 10.500 * 8) ~= **332** |
| Messages **per minute** in 1Mbit/s line | 60* 1,024,000 / (1.1 * 10.500 * 8) ~= **665** |

Based on similar calculations, each Member is able to calculate his own rough bandwidth requirements.

**Note:** z/OS connections usually have simulation and production on the same line. It is not possible to “reserve” a certain bandwidth for production. Therefore, it has to be kept in mind that excessive usage of simulation, e.g. for performance tests, can have a negative impact on the available bandwidth for production.

# 3 Transport layer

## 3.1 General information

Depending on individual requirements, Members have the choice to use either AMQP or WebSphere MQ as transport layer. The preferred transport layer for Eurex Clearing is AMQP. This applies to the Eurex Clearing FIXML Interface, to the Eurex Clearing FpML Interface and the Margin Calculator Interface.

## 3.2 AMQP

This section contains details about the connection to the Advanced Message Queuing Protocol (AMQP) transport service. The Eurex Clearing interfaces support only AMQP protocol 1.0 (ISO 19464). The link to the protocol specification can be found on the AMQP website:

- http://www.amqp.org

There are multiple implementations of AMQP protocol. For example Apache Software Foundation maintains Open source implementations compatible with AMQP 1.0 in the Qpid and ActiveMQ projects. For the client side they provides libraries written in multiple programming languages running on multiple operating systems. Documentation, client APIs and the source codes can be found on the Qpid website:

- http://qpid.apache.org

And on the ActiveMQ website:

- http://activemq.apache.org

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

- http://qpid.apache.org/components/broker-j/index.html

More details about the available client libraries are available in the Volume B: AMQP Programming Guide.

Connection to the Eurex Clearing AMQP brokers is established through a standard TCP/IP socket whereby any operating system and programming language supporting TCP/IP can be used. The broker is listening on a specific port for incoming connections. Transport Layer Security (TLS) protocol is used to encrypt connections and both client and server authentication with certificates is mandatory.

## 3.3 WebSphere MQ

On the Member’s side, the following requirements must – along with others – be fulfilled:

- TCP/IP via a z/OS connection.
- IBM WebSphere MQ server which is supported by IBM.
- Hardware and operating system that must be capable to support the IBM WebSphere MQ Server operation.

Neither Eurex Clearing, nor the Deutsche Börse Group, is capable to deliver any support for WebSphere MQ or capable to deliver any software to the Member. The acquisition and licensing procedure necessary for the components needed, such as IBM WebSphere MQ, is within each individual Member's responsibility.

## 3.4 Accounts

In order to configure the Eurex Clearing system correctly, an account needs to be created. This account is needed to ensure that the messages are sent from the Eurex Clearing backend to the correct Member queues. Further on, these accounts enable Eurex Clearing to verify that only messages from the "correct" Member are received from the respective queues.[5] The name of the account is reflected in the whole setup to the Member. Every Member can request multiple accounts for the Eurex Clearing FIXML Interface, the Eurex Clearing FpML Interface and the Margin Calculator Interface.

**For AMQP as connection layer:** These accounts can be requested and maintained using the Member section of the Eurex Clearing website.[6]

**For WebSphere MQ as connection layer:** Eurex Clearing will ask for the name of the account at the beginning of the setup process.

The account name has to be created according to following rules:

1. Characters 1 through 5 are the Member ID of the Eurex Member. Only uppercase letters are allowed.

2. Character 6 is an underscore sign separating the Member ID from the rest of the account name.

3. Characters 7 through 11 identify the vendor, Approved Trade source or the Member who developed the application. Only uppercase letters are allowed.

4. Character 12 and 13 identify, whether the application is a front, middle or back office. Valid keys are FO (Front Office), MO (Middle Office), BO (Back Office), FM (Front/Middle Office), FB (Front/Back Office), MB (Middle/Back Office), AL (Front-, Middle- and Back Office).

5. **For the Eurex Clearing FIXML Interface:** Character 14 identifies the trade adjustment processing. Valid keys are A (Automated), M (Manual), B (Automated / Manual) and N (None of the above).

6. **For the Eurex Clearing FIXML Interface:** Character 15 identifies the give-up and take-up processing. Valid keys are A (Automated), M (Manual), B (Automated / Manual) and N (None of the above).

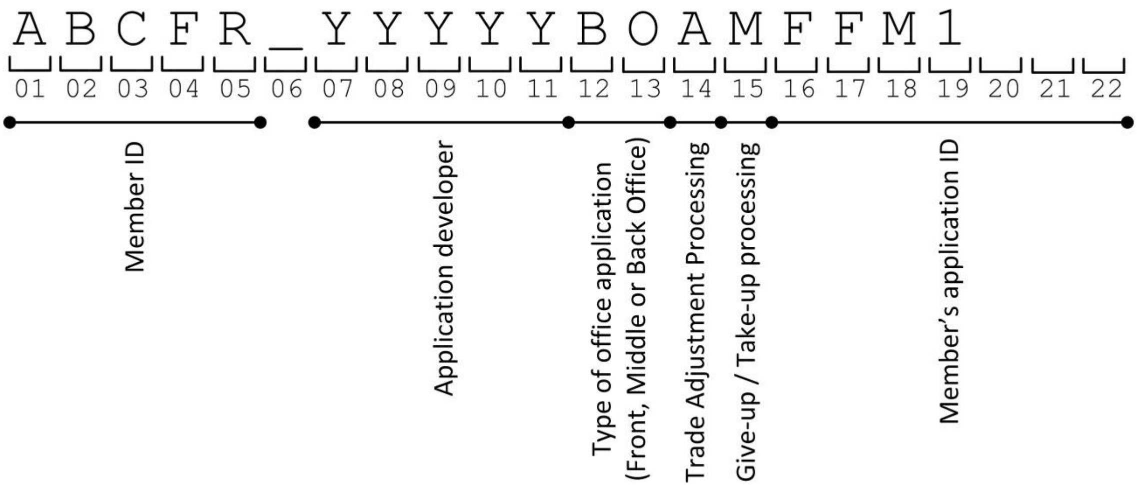
7. **For the Eurex Clearing FIXML Interface:** Characters 16 through 22 identify Member's application name or a combination of application name and location. The application name

[5] Note that for outsourcing purposes, the "correct" member may be an insourcing firm. Please refer to "Setup for out-sourced back offices" on page 43 for further detail about the available outsourcing alternatives.

[6] For the upload and maintenance of the certificates, the right "Technical User Administration" is needed in the Member Section.

must contain uppercase characters and numbers only. The application name is optional and may contain up to 7 characters.

8. **For the Eurex Clearing FpML Interface/Margin Calculator Interface:** Characters 14 through 22 identify the Member's application name or a combination of application name and location. The application name must contain uppercase characters and numbers only. The application name is optional may contain up to 9 characters.


A B C F R _ Y Y Y Y Y B O A M F F M 1
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
Member ID
Application developer
Type of office application (Front, Middle or Back Office)
Trade Adjustment Processing
Give-up / Take-up processing
Member's application ID

The account name has to be unique for the combination of the interface (FIXML, FpML or Margin Calculator), the given Member ID and the environment (simulation or production). This means, that the same account name can be used for simulation and production, but it cannot be used for an AMQP connection and a WebSphere connection towards the same system. If an AMQP connection is configured with exactly the same account name as it is used for an already existing WebSphere MQ connection, then the AMQP account configuration will be ignored. The following table lists examples of valid and invalid account names for the Eurex Clearing FIXML Interface:

| ***Account name*** | ***Validity*** |
|---|---|
| ABCFR_XXXFRBOAALONDON1 | Valid |
| ABCFR_YYYCOALNNFFM | Valid |
| ABCFR_YYYCOMBMM | Valid |
| ABCFR_**XXX**BOAACLEARNG | Invalid (application developer identification is too short) |
| ABCFR_YY@COBOAAFFM | Invalid (application developer identification contains @ sign) |
| ABCFR_**YYYYCO**BOAALONDON1 | Invalid (application developer identification is too long) |
| ABCFR_YYYYYALONCLEARNG | Invalid (incorrect trade adjustment processing key) |
| ABCFR_YYYYYALA**Z**CLEARNG | Invalid (incorrect give-up/take-up processing key) |

| | |
|---|---|
| ABCFR_YYYYY**TO**AACLEARNG | Invalid (incorrect “office” key) |
| ABCFR_YYYCOBOAA**AbcDef** | Invalid (lowercase characters in application name) |
| ABCFR_YYYCOBOAA**ABC DEF** | Invalid (application name contains space) |
| ABCFR_YYYCOBOAA**ABC@DEF** | Invalid (application name contains @ sign) |
| ABCFR_YYYCOBOAA**0123456789** | Invalid (application name is too long) |

The connecting application is always authenticated only on the Member (and his Eurex Clearing FIXML/FpML/Margin Calculator Interface account) level. It is up to the Member application, to provide the identification of the acting user (e.g. BOM005) as a part of the functional FIXML/FpML/Margin Calculator message.

# 4 AMQP

## 4.1 Certificates

The account authentication during the connection will be done using certificates. Therefore, the Member has to deliver the public key of his certificate to Eurex Clearing and assign this key to the account. The delivery of the key to Eurex Clearing will be done using the Eurex Clearing Member Portal (https://member.eurexclearing.com/). The key has to be delivered in the format as defined in the Internet Engineering Task Force (IETF) document RFC 1421 (see http://datatracker.ietf.org/doc/rfc1421/?include_text=1).

The certificate has to fulfill the following criteria:

- The certificate has to be compliant with the X.509v3 standard.
- The certificate is to be base64 encoded.
- It is sufficient that the certificate is self-certified.[7]
- The used key needs to have a minimum key length of 1976 bits.
- The used key needs to have a maximum key length of 4096 bits.
- The subject attribute of the certificate has to contain the account name as common name (e.g. "`CN=ABCFR_YYYCOBOMMAPP1`" in string representation of distinguished names according to IETF document RFC 2253). Exactly one common name must be specified. The subject attribute may not contain any domain components ("`DC=…`") or emails ("`E=…`"). Other additional data can be specified but has just informational purpose (e.g. "`O=<organization/company>`" or "`C=<country code>`").
- The validity can be chosen by the Member, but must not exceed a validity of 3 years (1095 days). Prior to the certificate's expiration the Member has to deliver a new certificate with exactly the same subject; otherwise no more connection to the Eurex Clearing FIXML Interface servers will be possible.
- The following key algorithms are supported: RSA.
- The following signature algorithms are supported: SHA-2 algorithm family (e.g. SHA224, SHA256, SHA384 or SHA-512).

After uploading the public key of the certificate, Eurex Clearing needs one batch in order to load the new or modified data into the AMQP brokers.[8] There is no possibility for intra-day modifications.

### 4.1.1 Generation of self-signed certificates

There are three recommended tools for generating correct certificates. The following chapters are describing these tools and the commands needed to generate the certificates. Eurex Clearing recommends to use the NSS certutil utility for applications using C++ as a programming language and Linux/UNIX as an operating system and to use JAVA for applications using JAVA as a programming language.

[7] The certificate can be signed by any (official) CA. However, the Eurex Clearing AMQP brokers will not verify if the certificate has been revoked by the CA. It is up the Member's responsibility to replace any expired or revoked certificates and to replace the public key in the Member section for the corresponding Member account(s).

[8] The upload needs to finished latest 30 minutes prior to the batch start in order to ensure that the certificate is loaded with the batch (this does not apply to short-term batch start changes). Otherwise, the certificate will be loaded with the next batch.

#### 4.1.1.1 Using the NSS certutil utility

The following example shows the creation of a self-signed client certificate with the open source utility `certutil` in a Linux environment. The `certutil` utility is a part of Network Security Services libraries. Further detail and additional documentation about certutil and Network Security Services is available on https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/tools/NSS_Tools_certutil.

`certutil` stores its certificate data in a certificate database and its key data in a key database. Assuming that these databases should be stored in the subdirectory `cert_dir` of the current directory then the following command creates the databases. The `cert_dir` directory has to exist before creating the database. A password has to be entered twice which is used to encrypt the keys later on.

```
$ mkdir cert_dir
$ certutil –N –d cert_dir
```

The next step is to create the self-signed client certificate. This includes the creation of a private-public key pair. The key pair will be stored in the key database; therefore it is required to enter the password chosen during the database creation in the previous step. For the generation of the key pair a random seed is required. For this purpose the utility uses the timing of keystrokes on the keyboard, i.e. the user has to enter a certain number of arbitrary keystrokes in arbitrary speed.

```
$ certutil –S –d cert_dir –s “CN=ABCFR_ABCFRALMMACC1” –n cert_name –x
–t “P,,” -v 12 -g 2048 -Z SHA512
```

- with the `–s` option the subject of the certificate is set; as stated in the requirements the subject has to contain exactly one common name (`CN`) with the Members’ account name.
- with the `–n` option the certificate is given a name which is used in other operations to identify a certificate, i.e. this name must be unique in the certificate database.
- the `-x` option instructs the utility to create a self-signed certificate.
- with the `–t` option it is specified for what purpose and on which level the certificate should be trusted; on the Member side this information is not important for a self-signed certificate (simply use the value shown above).
- the `–v` option allows to specify how long the certificate should be valid, the given value is a number of months; the certificate is valid immediately; the maximal validity is 36 months.
- the `–g` option specifies the size of the key in bits; the minimal size is 1976 bits.
- the `–Z` option specifies the signature algorithm used to sign the key; allowed are only signature algorithms from the SHA2 family (e.g. SHA-512).

The utility should automatically generate the certificate according to the X.509 version 3 standard. To ensure that the certificate has been created as desired it is possible to display all data in ASCII format:

```
$ certutil –L –d cert_dir –n cert_name
```

This produces an output like the following:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            00:95:cc:e2:e9
        Signature Algorithm: PKCS #1 SHA-512 With RSA Encryption
        Issuer: "CN=ABCFR_ABCFRALMMACC1"
        Validity:
            Not Before: Mon Feb 01 16:01:42 2013
            Not After : Wed Feb 01 16:01:42 2105
        Subject: "CN=ABCFR_ABCFRALMMACC1"
        Subject Public Key Info:
            Public Key Algorithm: PKCS #1 RSA Encryption
            RSA Public Key:
                Modulus:
                    be:60:64:f6:3a:62:ad:58:bc:92:44:24:ec:ed:c9:23:
                    6e:6d:e9:01:fd:8e:d4:6f:1a:23:3e:6c:7a:35:f1:8c:
                    71:2a:1d:9c:5b:06:61:e3:0f:49:bb:d7:da:a0:87:e5:
                    eb:c9:1a:9e:ff:8f:51:e2:a9:e8:00:90:42:f8:6a:e2:
                    33:6b:66:9d:20:2d:0b:ba:64:e3:5f:06:14:a0:25:47:
                    a3:9a:32:fd:9b:3b:da:3c:f3:9c:9d:1d:0d:19:9b:6a:
                    94:e1:5b:47:bf:a8:8e:3d:5a:a0:d9:a5:20:10:dd:66:
                    33:97:c7:0d:76:26:7e:bb:fc:50:35:dd:52:86:f0:65:
                    4a:39:5f:36:61:41:be:0e:6b:21:d1:52:ae:53:22:59:
                    17:1d:da:12:03:12:f3:60:a4:b1:c9:21:83:4f:20:9d:
                    72:2c:d3:a1:1f:ed:f3:12:62:bb:a7:cb:ab:52:c8:be:
                    36:ec:dc:5e:c4:61:1b:af:ed:6b:31:af:31:8a:dc:30:
                    1e:8a:f9:b4:25:03:63:39:74:8a:78:c8:e1:b7:4c:28:
                    a0:fd:d1:a1:e3:77:a9:d9:1c:b2:74:c3:96:2d:6e:59:
                    09:4e:5c:e7:9b:a0:5a:4e:d1:0d:3f:18:cc:ea:cc:46:
                    67:30:0a:9e:8c:08:b3:63:40:ef:b2:68:ac:92:56:bb
                Exponent: 65537 (0x10001)
    Signature Algorithm: PKCS #1 SHA-512 With RSA Encryption
    Signature:
        59:6a:20:12:2c:dc:58:99:76:ee:ce:ec:00:dc:50:11:
        de:cf:74:e5:38:38:41:ec:74:c0:6a:8f:1e:7b:07:08:
        5e:72:e4:90:50:f9:60:79:6b:ee:d8:d0:c7:ba:38:6b:
        73:98:c3:5a:a0:47:e8:ad:55:d7:dd:45:59:96:b7:77:
        19:97:92:ff:b9:d9:fe:da:c6:52:c8:9d:6e:5b:2c:dd:
        b4:ed:72:6b:a5:8e:cc:5b:08:83:0d:e3:2c:36:15:f7:
        b8:4f:d0:f3:63:a5:ec:c2:53:bd:69:ed:e0:e3:c3:68:
        67:42:2a:58:40:62:e8:65:5a:ba:bd:c4:20:5e:53:99:
        f1:8e:e6:34:d4:6e:05:31:46:43:a2:96:55:47:7d:04:
        59:38:7f:1a:e1:45:7d:e8:3e:29:02:ba:f4:91:99:f5:
        5d:3e:a5:1a:0c:e4:bf:b9:59:52:5a:71:89:2f:79:b6:
        d4:68:96:16:47:7b:9d:0e:d3:0c:4e:33:d5:b0:62:99:
        42:43:7d:11:ec:b3:1d:95:0a:e4:59:f0:4e:8d:81:f0:
        b5:22:0d:30:ef:b3:e2:58:b2:5c:eb:3b:f8:f0:b3:8c:
        80:e4:e1:50:7d:9a:40:af:2c:8d:9c:53:60:1b:9e:27:
        3d:a3:e1:cf:57:92:2a:e9:4d:4e:d1:ed:08:b1:b9:33
    Fingerprint (MD5):
        A7:BC:0B:DE:45:70:5F:4D:7D:49:38:BF:10:66:4B:AF
    Fingerprint (SHA1):
        95:9E:32:1F:1E:1B:C2:A8:FE:25:52:48:9D:3F:E9:41:6D:E2:B6:D1

    Certificate Trust Flags:
        SSL Flags:
            Valid Peer
            Trusted
            User
        Email Flags:
            User
        Object Signing Flags:
            User
```

Now the public part of the certificate has to be exported to allow a transfer to Eurex Clearing.

```
$ certutil -L -d cert_dir -n cert_name -a
```

An output like the following is generated:

```
-----BEGIN CERTIFICATE---
MIICoTCCAYmgAwIBAgIFAJXM4ukwDQYJKoZIhvcNAQENBQAwETEPMA0GA1UEAxMG
Y2JnYzAxMCAXDTExMTAwMzE2MDE0MloYDzIxMDMwMTAzMTYwMTQyWjARMQ8wDQYD
VQQDEwZjYmdjMDEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC+YGT2
OmKtWLySRCTs7ckjbm3pAf2O1G8aIz5sejXxjHEqHZxbBmHjD0m719qgh+XryRqe
/49R4qnoAJBC+GriM2tmnSAtC7pk418GFKAlR6OaMv2bO9o885ydHQ0Zm2qU4VtH
v6iOPVqg2aUgEN1mM5fHDXYmfrv8UDXdUobwZUo5XzZhQb4OayHRUq5TIlkXHdoS
AxLzYKSxySGDTyCdcizToR/t8xJiu6fLq1LIvjbs3F7EYRuv7WsxrzGK3DAeivm0
JQNjOXSKeMjht0wooP3RoeN3qdkcsnTDli1uWQlOXOeboFpO0Q0/GMzqzEZnMAqe
jAizY0Dvsmiskla7AgMBAAEwDQYJKoZIhvcNAQENBQADggEBAFlqIBIs3FiZdu7O
7ADcUBHez3TlODhB7HTAao8eewcIXnLkkFD5YHlr7tjQx7o4a3OYw1qgR+itVdfd
RVmWt3cZl5L/udn+2sZSyJ1uWyzdtO1ya6WOzFsIgw3jLDYV97hP0PNjpezCU71p
7eDjw2hnQipYQGLoZVq6vcQgXlOZ8Y7mNNRuBTFGQ6KWVUd9BFk4fxrhRX3oPikC
uvSRmfVdPqUaDOS/uVlSWnGJL3m21GiWFkd7nQ7TDE4z1bBimUJDfRHssx2VCuRZ
8E6NgfC1Ig0w77PiWLJc6zv48LOMgOThUH2aQK8sjZxTYBueJz2j4c9XkirpTU7R
7QixuTM=
-----END CERTIFICATE-----
```

This is the complete public certificate data, base64 encoded as it is specified as part of the IETF document RFC 1421. This ASCII data has to be delivered to Eurex Clearing using the Member Section of the Eurex Clearing website.

The private key must be kept secret and must never be provided to anyone. Members might consider securing it using a hardware token to increase the protection level. The `certutil` tool supports this with the "`-h`" option.
If necessary, the private key can be exported from the NSS database to the PKCS12 format using the `pk12util` utility:

```
$ pk12util -d cert_dir -n cert_name -o abcfr_abcfralmmacc1.p12
```

#### 4.1.1.2 Using the keytool utility

The following example shows the creation of a self signed client certificate with the keytool utility, which is part of the Java Runtime Edition and Java Development Kit distributions from Sun/Oracle. More details and additional documentation about keytool can be found on http://docs.oracle.com/javase/8/docs/technotes/tools/windows/keytool.html. As a Java utility, it is available for most platforms, including Windows and Linux. In order to generate the certificate according to the X.509 version 3 standard, Java 1.6 or higher needs to be used.

The keytool stores the data in keystores. The keystore is a file, which can contain multiple private and public keys. The keystore file is protected by a password. The use of keytool is especially practical when the keystore is used in Java based applications, however, it can also be easily exported/converted to other formats like PKCS12, etc.

The self-signed certificate can be generated using the following command. When generating the new certificate in an existing keystore, the keytool utility will ask for the password of this keystore. When generating the new certificate without an existing keystore, a password for a new keystore has to be entered twice for verification. The key password should be set to the same as the keystore password.

```
$ keytool -genkeypair -alias abcfr_abcfralmmacc1 -dname
"CN=ABCFR_ABCFRALMMACC1" -validity 365 -keysize 2048 -keyalg RSA -
sigalg SHA512withRSA -keystore keystore_abcfr_abcfralmmacc1
```

- The `–genkeypair` option instructs the `keytool` to generate a self-signed certificate.
- With the `-alias` option the certificate is given a name, which has to be unique in the keystore. This name can be used to reference the certificate from the application[9].
- The `–dname` option defines the subject of the certificate. As stated in the requirements the subject has to contain exactly one common name (CN) with the Members' account name (see chapter 3.4 for more details about the account names).
- The `–validity` option specifies for how long the certificate should stay valid. The value is the number of days of validity.
- The `–keysize` option specifies the size of the private key. According to the interface specification, the size should be at least 1976 bits. In the example above, the size of the generated key will be 2048 bits.
- The `–keyalg` option specifies the used key algorithm. In order to use the SHA2 family for signatures, RSA algorithm must be used.
- The `–sigalg` option specifies the used signature algorithm. An algorithm from the SHA2 family must be used. The example above is using the SHA-512 signature algorithm. Other possibilities are for example "SHA256withRSA" for SHA-256 etc.
- The `–keystore` option is used to specify the filename of the keystore. If the keystore doesn't exist, a new one will be created.

The content of the keystore file can be verified using the –list option:

```
$ keytool -list -keystore keystore_abcfr_abcfralmmacc1
keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

abcfr_abcfralmmacc1, May 19, 2011, PrivateKeyEntry,
Certificate fingerprint (MD5):
37:08:54:4C:8B:6C:42:E4:E2:C0:03:92:9C:68:45:2B
```

The output should be similar to the output above. The public part of the certificate has to be exported and uploaded into the Member Section of Eurex Clearing. The export can be done using following command:

```
$ keytool -export -alias abcfr_abcfralmmacc1 -keystore
keystore_abcfr_abcfralmmacc1 -rfc –file abcfr_abcfralmmacc1.crt
```

- The `–export` instructs the `keytool` to read the key and save it to a file.

[9] The keytool utility will convert the alias to lowercase in the generated keystore. That should be taken into account when using the keystore and the certificate should be referred to with the alias in lowercase.

- With the `-alias` option specifies the certificate name.
- The `–keystore` option is used to specify the filename of the keystore containing the certificate.
- The `–rfc` option specifies that the public key will be saved in printable encoding format according to RFC 1421 standard.
- The `–file` option specifies the output file, where the public key should be stored.

The keytool utility will request entering the keystore password before exporting the public key. The content of the `abcfr_abcfralmmacc1.crt` file should be similar to this:

```
-----BEGIN CERTIFICATE-----
MIICVDCCAhICBE2xdyIwCwYHKoZIzjgEAwUAMBAxDjAMBgNVBAMTBWpha3ViMB4XDTExMDQyMjEy
NDAwMloXDTEyMDQyMTEyNDAwMlowEDEOMAwGA1UEAxMFamFrdWIwggG3MIIBLAYHKoZIzjgEATCC
AR8CgYEA/X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9sub
VWzXgTuAHTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7V+fGqKYVDwT7g/bTxR7DAjVUE1oW
kTL2dfOuK2HXKu/yIgMZndFIAccCFQCXYFCPFSMLzLKSuYKi64QL8Fgc9QKBgQD34aCF1ps93su8
q1w2uFe5eZSvu/o66oL5V0wLPQeCZ1FZV4661FlP5nEHEIGAtEkWcSPoTCgWE7fPCTKMyKbhPBZ6
i1R8jSjgo64eK7OmdZFuo38L+iE1YvH7YnoBJDvMpPG+qFGQiaiD3+Fa5Z8GkotmXoB7VSVkAUw7
/s9JKgOBhAACgYB2Oe1XnnFC5TDGe1CBkRroIWDVHwuWDU2vaUFfq5lVlVOaVa1+aM3UQFK9TZc1
bXlLVWoL0a4WZu/a5djKxEEBSiEP95zXmFbZZmdAPkzCpEO6XbBht9mggFESzjovWHshI8SZXBeK
oB2hy4JH4Mw4eHR60jZWcvwVSk7PrS7V9DALBgcqhkjOOAQDBQADLwAwLAIUPBzxB1Cpc49ttwcN
iDw6yBiV6rQCFBlMzRSHrX1KIcUUHr+JIVzloWRw
-----END CERTIFICATE-----
```

This data/file should be delivered to Eurex Clearing using the Member Section of the Eurex Clearing website.

The private key must be kept secret and must never be provided to anyone.

If necessary, the keystore key can be converted to the PKCS12 format using the following command:

```
$ keytool -importkeystore -srckeystore keystore_abcfr_abcfralmmacc1
-destkeystore ./abcfr_abcfralmmacc1.p12 -deststoretype PKCS12
```

Similarly, if the key already exists in PKCS12 format, it can be converted into the keystore format:

```
$ keytool -importkeystore -srckeystore ./abcfr_abcfralmmacc1.p12 -
srcstoretype PKCS12 -destkeystore keystore_abcfr_abcfralmmacc1
```

#### 4.1.1.3 Using the openssl utility

The following example shows the creation of a self signed client certificate using the openssl utility, which is a part of an open-source SSL / TLS toolkit named OpenSSL (http://www.openssl.org/). The toolkit comes either pre-installed or it is available via a package manager in most Linux distributions. The project’s website offers source code downloads. Informal list of third party products offering binaries can be found on the OpenSSL Wiki (https://wiki.openssl.org/index.php/Binaries). A detailed description of all openssl utility command-line parameters is available in the openssl man-page at http://www.openssl.org/docs/apps/openssl.html.

The certificate creation process has multiple steps. First, a private key needs to be generated. The openssl tool is capable of generating private keys of varying length, either unprotected or password-protected using a selection of ciphers, e.g.

```
$ openssl genrsa -des3 -out ABCFR_ABCFRALMMACC1.key 2048
```

The utility interactively asks for a password protecting the new private key being created:

```
Generating RSA private key, 2048 bit long modulus
....................................................+++
.....................+++
e is 65537 (0x10001)
Enter pass phrase for ABCFR_ABCFRALMMACC1.key:
Verifying - Enter pass phrase for ABCFR_ABCFRALMMACC1.key:
```

The above command generates a 2048-bit RSA private key, password-protected using the triple-DES cipher, into a file named `ABCFR_ABCFRALMMACC1.key`, e.g.:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,C078A362F0A50C98

hkDfkYV1lH1IJ2MgRoVfRvcLX3BgnspejOSt1DH88jEHPYRe3TQf/3bu/kRrlYyh
/BOPg6TMTkeD9xUg006lytamPNyUTmhWttbKJd53g+5rhGky/4mvIprW4IkvGMsO
C5SUp0+tCHCmGLzeMBh5YZSbcCjWNA1wO/UgkWveU72Fr4v0BOjRQGrfKx3nFALD
EumpcQ/JQAC6vUNzv+BMPnq47D4QGyGmHAKtDJOshU/ubjVQ80FsnTo629pvW3Xh
lQ4dr6t+WSYq//zKhguCARGNQ3RnCrDkfydmVU++NBJ8Q42fnOnV9iw8EcPTAWlw
uX1oAAWuPDGnU+DgT5t2sHyzAKGjk1C/5auXWm/Gpvw6TxJTcaaJD/LmEc0I3144
SF6e4v7RE1cGZK58qhLwfNPRVwUMQofXqus8hW25qPK9h8prKDVog+OzwRQrnUi4
UN/nk5mQJNyXi5efi26c4cOE0aWq4odsxqVs1kG7/9LNeSPj6HRj1usGBB3KQRHQ
PxeRZXW882yk42yIjlMGEnCtGEU8LMz8mFKgBOmbENqBkJs7dUDBnrC5c2L0tis3
6cc6EqmcdTuO3N0wmuf4vwXrjwcIG0mS11vMMO6L5IY58P7abjUhCd/oyHAzOd4W
lQMXPx5JVZDnN83YVpxToJIdXHFJLDbxpCPKRy42q1hiFOUygFj58r32n4P7ji/H
tEK1zZrBgzGvjBSu/Ce3daj2pfROy+28EARKyvcTsv3PZIdgneCQUCjrV4//JWSx
//gb/7PM5OIipb2G2PdG/Q+KQ3zFjRmL5RUejhOKj2jI2N9b1tzMUOydRs/RxW6f
PWQcQ6cm7VhfPeKO3ObJjNAOju6jYx3onI31+xWBz/do9p55DU23igSHVRbt1jGd
6oTXkBkiLdArHvM2MAPC8p8DNgy8Nxb7yQvyexwuIdFLAr6bHMLToCP0ad86aj9m
TidjTlWZYm5oW+PKx9vU7kaqrvaEKrhWjKr5wqjNovLNI7Abihins8sLwxrBUN39
BSB+QzXeK1TbGqGYET9E+HVdVoT4jVGrjSE/TIG+mFdqtZS0upxkEGbcj6uFJb75
YMjHB+BGwCNmNtgTxstyq4dyKrrIU1bXNLo5vzUKH5y4UcV+9I7dIJ9FxdawK3hn
badH6AhnL6SBGeoiopR1b0wsdglUw/7K+wUajJVStqqtX008XwewolBdCc+T+TeX
joBs9Xl8/q+VJo6iD8vSSeLm4kJ9FVKN5XYogzFJUXLC58PApKKhReAv5Trzt8i5
Le4Dp7GSW+Ov2v4NcKIb8zJQIFJBBn0m9N3m9m6yomWE0TkDkfTAd6t1AuPvnM9P
```

```
w88myfGEQdwnreh8zIJ1NJxu1vxRVClConwqJNCCZjOO1eQqZ2K4iIfK81F1ICGG
ol05dPf9N+w05zxZogWnWBHBGEzvSrsgdZiJ1L9Pzqmt+3gjESZKuoOXj0qHoR78
ZWFpkYyYv1f7uDUffEdeXl0bxp40rSv0Kiyqis/HjyMG75IzGbA//ktV8XwiBq7/
3Hfh5OfBwcNtiGnnojBR6VTpPJ6456fj0TBP6MM5DFZFDEci6txg4+HkjBmK2Enu
-----END RSA PRIVATE KEY-----
```

The next step is the creation of a self-signed certificate corresponding to the above private key, e.g.:

```
$ openssl req -new -x509 -sha512 -days 365 -key
ABCFR_ABCFRALMMACC1.key -out ABCFR_ABCFRALMMACC1.crt
```

The command expects some interactive input. First, the private key password (from the previous step) has to be provided. The account name needs to be entered when prompted for the Common Name (CN) (ABCFR_ABCFRALMMACC1 in this example). The Email Address has to be empty (just press return) as certificates having a non-empty e-mail field are not supported. Other fields – for example country name, city or organization are optional:

```
Enter pass phrase for ABCFR_ABCFRALMMACC1.key:
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:DE
State or Province Name (full name) []:Hessen
Locality Name (eg, city) [Default City]:Frankfurt
Organization Name (eg, company) [Default Company Ltd]:ABC AG
Organizational Unit Name (eg, section) []:Clearing
Common Name (eg, your name or your server's hostname)
[]:ABCFR_ABCFRALMMACC1
Email Address []:
```

The above command generates an X.509v3-standard self-signed certificate with Common Name (CN) equal to the desired member account name, using the SHA-512 hashing algorithm, expiring after one year. It is stored in a file named ABCFR_ABCFRALMMACC1.crt, e.g.:

```
-----BEGIN CERTIFICATE-----
MIIDuzCCAqOgAwIBAgIJAOQDoY1RdnB1MA0GCSqGSIb3DQEBDQUAMHQxCzAJBgNV
BAYTAkRFMQ8wDQYDVQQIDAZIZXNzZW4xEjAQBgNVBAcMCUZyYW5rZnVydDEPMA0G
```

```
A1UECgwGQUJDIEFHMREwDwYDVQQLDAhDbGVhcmluZzEcMBoGA1UEAwwTQUJDRlJf
QUJDRlJBTE1NQUNDMTAeFw0xMzA0MTYxNTUzMzJaFw0xNDA0MTYxNTUzMzJaMHQx
CzAJBgNVBAYTAkRFMQ8wDQYDVQQIDAZIZXNzZW4xEjAQBgNVBAcMCUZyYW5rZnVy
dDEPMA0GA1UECgwGQUJDIEFHMREwDwYDVQQLDAhDbGVhcmluZzEcMBoGA1UEAwwT
QUJDRlJfQUJDRlJBTE1NQUNDMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAOS06Bk+35sRFo+LIrhWWaVJsFI+12o7ruG2NJIICms1KBWjoy7ictIH0Dea
zI7cheZUAMZgYKT04O0/7kiz+DgfSwpXRWzc/QFIlP/x3SnFDNE5L1xPJf6s2gXr
z6S44tntK/b3MnIarTok0EFJ1pkgdmUHUGoU41k2xIa7IpewrHbDHPLmrvqY0yLK
XiEvhW45pDyCAz29B3OYdFLwy0NGXVKWvVmch+xdCXEPd4YzfWUbYShc15ZOHL1f
5n7wu1RP+ju7xgruX3sSguc58NnSP1n0MfmfqpyJPIjhOi1BcWBWAnGTvVmwqLC8
mnKcGnlAENiasfSZInz78HWtGhkCAwEAAaNQME4wHQYDVR0OBBYEFNRXD/CVV8uG
N92Xi4QQo1xLyfb+MB8GA1UdIwQYMBaAFNRXD/CVV8uGN92Xi4QQo1xLyfb+MAwG
A1UdEwQFMAMBAf8wDQYJKoZIhvcNAQENBQADggEBAB8hNpkIxkK4NJ2T8KHAARnE
wBh+fJM7gqUUdQ3d7GjNwSpXu+69FuSZCNn5QgxMy9pyLRJzgXEoGpbhFM/nlONT
5U/yXdQSNDC07H5d+T/CBkVGzTtVbreMPIBydbRelO65D3cyhPE+0psTbtKt1HDt
3EuN0j1f8RhWvGqBSTdedBBycCatlJbVZbdeccBx9xfNyiOi3X/dB3Zck+suQvXv
mAGjEHxlncl7MBKd4LZpP37nySkfgj9/LXWFB7lKFrcCZ/2UdzEKzf6Q2iiTmD5A
AN/+2534a/9Rt7Fd4mdNMiKMM+15ipicr+JhPclS52Iojo17q9BF0x6mIuCEoqE=
-----END CERTIFICATE-----
```

This certificate should be delivered to Eurex Clearing using the Member Section of the Eurex Clearing website.

The corresponding private key should be used by the Member's applications to authenticate with Eurex Clearing system interfaces. The private key must be kept secret and must never be disclosed to anyone.

A human-readable certificate information may be obtained via the following command, e.g.:

```
$ openssl x509 -text -noout -in ABCFR_ABCFRALMMACC1.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            e4:03:a1:8d:51:76:70:75
        Signature Algorithm: sha512WithRSAEncryption
        Issuer: C=DE, ST=Hessen, L=Frankfurt, O=ABC AG, OU=Clearing, 
CN=ABCFR_ABCFRALMMACC1
        Validity
            Not Before: Apr 16 15:53:32 2013 GMT
            Not After : Apr 16 15:53:32 2014 GMT
        Subject: C=DE, ST=Hessen, L=Frankfurt, O=ABC AG, OU=Clearing,
CN=ABCFR_ABCFRALMMACC1
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:e4:b4:e8:19:3e:df:9b:11:16:8f:8b:22:b8:56:
                    59:a5:49:b0:52:3e:d7:6a:3b:ae:e1:b6:34:92:08:
                    0a:6b:35:28:15:a3:a3:2e:e2:72:d2:07:d0:37:9a:
                    cc:8e:dc:85:e6:54:00:c6:60:60:a4:f4:e0:ed:3f:
                    ee:48:b3:f8:38:1f:4b:0a:57:45:6c:dc:fd:01:48:
                    94:ff:f1:dd:29:c5:0c:d1:39:2f:5c:4f:25:fe:ac:
                    da:05:eb:cf:a4:b8:e2:d9:ed:2b:f6:f7:32:72:1a:
                    ad:3a:24:d0:41:49:d6:99:20:76:65:07:50:6a:14:
                    e3:59:36:c4:86:bb:22:97:b0:ac:76:c3:1c:f2:e6:
                    ae:fa:98:d3:22:ca:5e:21:2f:85:6e:39:a4:3c:82:
                    03:3d:bd:07:73:98:74:52:f0:cb:43:46:5d:52:96:
                    bd:59:9c:87:ec:5d:09:71:0f:77:86:33:7d:65:1b:
                    61:28:5c:d7:96:4e:1c:bd:5f:e6:7e:f0:bb:54:4f:
                    fa:3b:bb:c6:0a:ee:5f:7b:12:82:e7:39:f0:d9:d2:
                    3f:59:f4:31:f9:9f:aa:9c:89:3c:88:e1:3a:2d:41:
                    71:60:56:02:71:93:bd:59:b0:a8:b0:bc:9a:72:9c:
                    1a:79:40:10:d8:9a:b1:f4:99:22:7c:fb:f0:75:ad:
                    1a:19
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:

D4:57:0F:F0:95:57:CB:86:37:DD:97:8B:84:10:A3:5C:4B:C9:F6:FE
            X509v3 Authority Key Identifier:

keyid:D4:57:0F:F0:95:57:CB:86:37:DD:97:8B:84:10:A3:5C:4B:C9:F6:FE


            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha512WithRSAEncryption
        1f:21:36:99:08:c6:42:b8:34:9d:93:f0:a1:c0:01:19:c4:c0:
        18:7e:7c:93:3b:82:a5:14:75:0d:dd:ec:68:cd:c1:2a:57:bb:
        ee:bd:16:e4:99:08:d9:f9:42:0c:4c:cb:da:72:2d:12:73:81:
```

```
71:28:1a:96:e1:14:cf:e7:94:e3:53:e5:4f:f2:5d:d4:12:34:
30:b4:ec:7e:5d:f9:3f:c2:06:45:46:cd:3b:55:6e:b7:8c:3c:
80:72:75:b4:5e:94:ee:b9:0f:77:32:84:f1:3e:d2:9b:13:6e:
d2:ad:d4:70:ed:dc:4b:8d:d2:3d:5f:f1:18:56:bc:6a:81:49:
37:5e:74:10:72:70:26:ad:94:96:d5:65:b7:5e:71:c0:71:f7:
17:cd:ca:23:a2:dd:7f:dd:07:76:5c:93:eb:2e:42:f5:ef:98:
01:a3:10:7c:65:9d:c9:7b:30:12:9d:e0:b6:69:3f:7e:e7:c9:
29:1f:82:3f:7f:2d:75:85:07:b9:4a:16:b7:02:67:fd:94:77:
31:0a:cd:fe:90:da:28:93:98:3e:40:00:df:fe:db:9d:f8:6b:
ff:51:b7:b1:5d:e2:67:4d:32:22:8c:33:ed:79:8a:98:9c:af:
e2:61:3d:c9:52:e7:62:28:8e:8d:7b:ab:d0:45:d3:1e:a6:22:
e0:84:a2:a1
```

## 4.2 Connecting to the Eurex Clearing AMQP broker

The connection to the Eurex Clearing 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 "Eurex Exchange and Eurex Clearing Network Access Manual". This document is available in the public section of the Eurex Clearing website under the following path:

https://www.eurex.com/ec-en/ → *Tech* → *C7* → *System documentation* → *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.

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. Following cipher suites are recommended and supported by Eurex Clearing's AMQP brokers[10]:

- 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. In order to make the client application accept the server certificate during the connection establishment, the public keys of the corresponding Eurex Clearing 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.

[10] 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/Trade Entry 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 established successfully within this timeframe will be disconnected.[11]

## 4.3 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.

### 4.3.1 Requests & 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.

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

### 4.3.2 Broadcasts

Broadcast messages are generated and disseminated by the Eurex Clearing System. They are divided into broadcasts streams. Each of these streams is delivering messages into one queue. The number of used broadcast queues and the functional split of these broadcasts queues is different for each Eurex Clearing interface.

It is not guaranteed that the broadcast messages are delivered in the functional chronological order. The messages delivered as broadcasts are described in the other volumes of this documentation.

#### 4.3.2.1 Eurex Clearing FIXML Interface

Currently, for the Eurex Clearing FIXML Interface, the following broadcast streams for all Members are provided:[12]

- 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 a Clearer's Non-Clearing Members. The Public broadcast stream contains public information (e.g. series information or settlement prices). The public data is identical for all Members – Clearers as well as Non-Clearing Members. Therefore only one stream for the public data is provided.

#### 4.3.2.2 Eurex Clearing FpML Interface

Currently, for the Eurex Clearing FpML Interface, the following broadcast streams for all Members are provided[13]:

- 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.

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

#### 4.3.2.3 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.

#### 4.3.2.4 Eurex Clearing Trade Entry Interface

Currently, for the Eurex Clearing Trade Entry Interface, the following broadcast stream for all Approved Trade sources is provided[14]:

- TradeNotification

If the a trade received from the Approved Trade Sources is booked in the system, then all trade notification messages will be sent to the TradeNotification queue of the Approved Trade source.

## 4.4 Communication phases

The Eurex Clearing interfaces consist of two parts:

- Eurex Clearing System
- AMQP broker

The AMQP broker delivers the messages between the Eurex Clearing System and the Member. The Eurex Clearing System is processing request messages and generating broadcasts. The Eurex Clearing System and the AMQP broker may be in a different state.

The state of the Eurex Clearing System is determined by the Eurex Clearing System phases. The Eurex Clearing System processes requests[15] and generates broadcasts only during the ONLINE system state. The AMQP interface is designed to be available through-out the whole day. Since the AMQP broker will be running even when the Eurex Clearing System is not online, the following situation may appear:

> The AMQP broker is online, but the Eurex Clearing System is not online (e.g. already in EOD batch processing). Members are able to connect to the AMQP broker and send requests. However, they will not receive any responses, because the Eurex Clearing System is not processing requests anymore.

During the night the AMQP broker undergoes so called “technical maintenance” procedure. During this phase, the broker is being reconfigured for the next business day. Member accounts and certificates are being added or removed as requested by members through the Member Section of the Eurex Clearing website. During this procedure, it might happen that the broker will be temporarily unavailable. In such case the member application should reconnect once the broker is available again.

On the Eurex Clearing FpML Interface, all messages which were unconsumed in the broadcast queues when the technical maintenance starts will be deleted. On Eurex Clearing FIXML Interface, all unconsumed messages will stay in the queues until they expire.

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

[15] The fact that the Eurex Clearing System is processing the requests does not necessarily mean that all actions which can be requested over the Eurex Clearing FIXML/FpML/Margin Calculator Interface will be executed. Depending on the system state or instrument state, some specific operations may not be allowed. In case that the requested action is not allowed during the system state, the request may be responded with a reject message.

## 4.5 Reliability and Duplicate detection

The messages will be delivered with at least once reliability. Duplicate messages may appear – especially in case any failovers or reconnects occurred during the communication. More details about the reliability and duplicate detection on the AMQP connectivity are available in Volume E, chapter 5.

# 5 WebSphere MQ

For each interface (FIXML/FpML/Trade Entry Interface) and for each environment (simulation/production), a separate WebSphere MQ configuration will be set-up on the Eurex Clearing side.

The Member has the possibility to "order" more than one connectivity for any interface and environment combination. In this case, the Member has the option to connect different locations to Eurex Clearing's WebSphere MQ infrastructure.

SSL / TLS connections are possible, if the Member requests these. In this case, a certificate from an official CA is best operational procedure. Certificates signed by a Member-owned CA are the next choice. Self-signed certificates are not supported.

## 5.1 Setup process

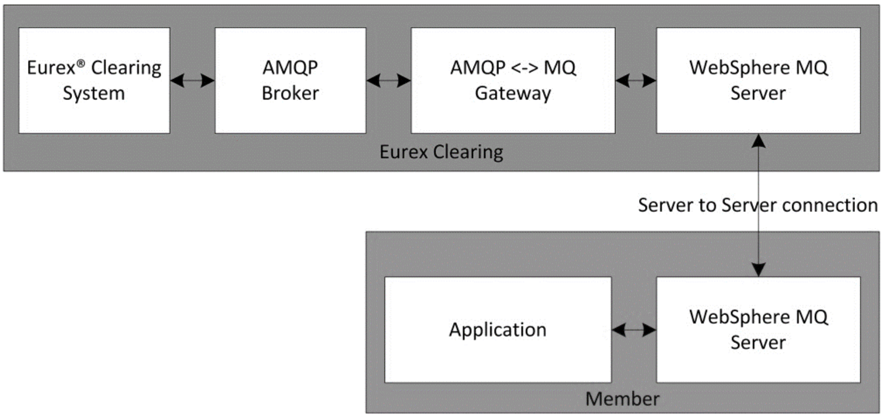
The Member should contact his personal Technical Key Account Manager in order to establish a WebSphere MQ connection to Eurex Clearing. The responsible Technical Key Account Manager will then coordinate all tasks needed for the setup of the connection. The administrator team responsible for the WebSphere MQ infrastructure used by Eurex Clearing will then contact the Member's WebSphere MQ administrator(s) in the context of the setup process of the WebSphere MQ objects. In this context, the Member will receive a configuration file from Eurex Clearing's WebSphere MQ administrators which contains the configuration of the WebSphere MQ objects as used by Eurex Clearing. Eurex Clearing is using alias queues and queue-manager alias definitions. Certain objects of this configuration can be modified by the Member in order to fulfill his own standards and requirements (e.g. storage classes, queue depth, xmitq triggering, …). Specific objects cannot be modified by the Member:

- Channel names.
- Queue names the Eurex Clearing is referring to (all receiving `QALIAS` and queue-manager alias queue names).
- All `RNAME` and `RQMNAME` names that are referring to Eurex Clearing.

Eurex Clearing is using an "IVP" queue (installation verification procedure) which is a loopback queue for the connection verification. It is used only selectively for the connection setup and after major changes in order to verify connectivity and channel sequence numbers. The Member should make sure that any message received from Eurex Clearing in this queue is sent right back to Eurex Clearing without being processed by any application.

## 5.2 Eurex Clearing setup

Eurex Clearing is using a gateway process to "translate" and forward messages from AMQP to WebSphere MQ and vice versa. The following illustration shows a high level overview of the WebSphere MQ integration:



The communication between the Member and Eurex Clearing is established through the WebSphere MQ server-to-server connection. The Eurex Gateway process is responsible for delivering the messages between the AMQP broker and the WebSphere MQ server.

## 5.3 Channel

Eurex Clearing will keep the sender channel to the Member “running” through service hours. The Member is free to choose if he wishes to keep the sender channel to Eurex Clearing running during these hours as well or if the Member wants to use disconnect interval/channel triggering.

The receiving channel on Eurex Clearing’s WebSphere MQ server is a RQSTR channel. It may be started on Eurex Clearing’s side once a day or week (at channel/application startup time) or in case of problems (to re-establish the MQ connection fast). Except for these reasons, Eurex Clearing will not start this channel. It is up to the Member to make sure that the channel to Eurex Clearing is "running" in order to send messages to Eurex Clearing.

## 5.4 Queues

Eurex Clearing will provide the messages in different queues. This separation of messages is done according to functional considerations, e.g. message type or message content. It is up to the responsibility of the Member to decide if the receiving queues on the Member side are bound to one or several of the queues as provided by Eurex Clearing. The exact names of the queues and the name of the queue manager are provided during the setup process of the WebSphere MQ connection.

Eurex Clearing will not empty the Member’s channel during Eurex Clearing’s overnight processing or during channel restarts. It is up to the Member’s responsibility to read new messages in time and to ensure that his applications are able to process outdated message correctly from a functional point of view.

Eurex Clearing will always setup all theoretically available queues – independent of Member status. This will ensure that in case of a change of the Member status, no further reconfigurations on the WebSphere MQ connection configuration are needed.

### 5.4.1 Requests & responses

Eurex Clearing will use one queue for responses and one queue for requests. The request queue is filled by the Member with requests from the Member's applications. Depending on the request, an answer is given by the Eurex Clearing System into either the response queue or into the response queue and into the broadcasts queues. If necessary, Members can track the requests and responses by matching the correlation IDs of sent requests and received responses.

### 5.4.2 Broadcasts

In addition to the request and response queues, Eurex Clearing will use queues for broadcasts. The number of used broadcast queues and the functional split of these broadcast queues is different for each Eurex Clearing interface.

It is not guaranteed that the broadcast messages are delivered in the functional chronological order. The messages delivered as broadcasts are described in the other volumes of this documentation.

#### 5.4.2.1 Eurex Clearing FIXML Interface

Currently, for the Eurex Clearing FIXML Interface, the following broadcast streams for all Members are provided:[16]

1. TradeConfirmation
2. Workflow
3. TradeConfirmationNCM (will contain messages only for Clearing Members)
4. WorkflowNCM (will contain messages only for Clearing Members)
5. 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.

#### 5.4.2.2 Eurex Clearing FpML Interface

Currently, for the Eurex Clearing FpML Interface, the following broadcast streams for all Members are provided:[17]

1. TradeNotification
2. 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:

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

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

#### 5.4.2.3 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.

#### 5.4.2.4 Eurex Clearing Trade Entry Interface

Currently, for the Eurex Clearing Trade Entry Interface, the following broadcast stream for all Approved Trade Sources is provided[18]:

- TradeNotification

If the a trade received from the Approved Trade Sources is booked in the system, then all trade notification messages will be sent to the TradeNotification queue of the Approved Trade Source.

## 5.5 Communication phases

The Eurex Clearing interfaces consist of two parts:

1. Eurex Clearing System
2. WebSphere MQ server

The WebSphere MQ server delivers the messages between the Eurex Clearing System and the Member. The Eurex Clearing System is processing request messages and generating broadcasts. The Eurex Clearing System and the WebSphere MQ server may be in a different state.

The state of the Eurex Clearing System is determined by the Eurex Clearing phases. The Eurex Clearing System processes requests[19] and generates broadcasts only during the ONLINE system state. The WebSphere MQ interface is designed to be available through-out the whole day. This includes batch processing phases as well. Since the WebSphere MQ server will be running even when the Eurex Clearing System is already in batch processing, the following situation may appear:

The WebSphere MQ server is online, but the Eurex Clearing System is not online (e.g. already in EOD batch processing). Members are able to connect, and send requests. However, they will not receive any responses, because the Eurex Clearing System is not processing requests anymore.

## 5.6 Data encoding

Because Eurex Clearing FIXML Interface is using message properties the message format is MQHRF2, and the message data is extended with RFH2 header in which named properties are stored along with other JMS header field or properties for which there is no MQMD equivalent.

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

[19] The fact that the Eurex Clearing System is processing the requests does not necessarily mean that all actions which can be requested over the Eurex Clearing FIXML/FpML/Margin Calculator Interface will be executed. Depending on the system state or instrument state, some specific operations may not be allowed. In case that the requested action is not allowed during the system state, the request may be responded with a reject message.

The message data itself is in ASCII format using MQFMT_STRING and proper CCSID values (set in the RFH2 header).

The Member's application must be aware of the message format and message properties. E.g. receive them, or ignore them (PROPCTL of channels and queues).

For the channels, the CONVERT(NO) option is used. Any conversion of received messages has to be done by the Member's application programs along with the read out of the WebSphere MQ queue (MQGET-CONVERT option used with MQGET). When sending messages, the Member must use MQFMT_STRING and proper CCSID values in the MQMD header depending on the encoding of the message data to enable Eurex Clearing to convert incoming messages.

## 5.7 Reliability and Duplicate detection

The messages will be delivered with at least once reliability. It is guaranteed that they will be delivered to the WebSphere MQ server on the member side at least once. However, duplicate messages may appear – especially in case any failovers or reconnects occurred during the communication. 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 contain the details about the documentation of our interfaces.

# 6 Setup for outsourced back offices

This chapter applies only to the Eurex Clearing FIXML Interface.

## 6.1 Prerequisites

Outsourcing comprises an arrangement between a Eurex Member (outsourcer) and an Approved Trade source (insourcing firm) by which that Approved Trade source performs functions and activities legally related to the Members business as an admitted Eurex Clearing Member. The outsourcing firm must have a user setup in the Eurex Clearing System for the insourcing firm and provide the user ID to the insourcing firm. The outsourcer is able to set the entitlement for the acting user according to the contractual outsourcing agreement and fulfills his legal obligations.

Two technical setup alternatives are available. Note that for both options the actual FIXML messages are to be sent as if they had been submitted by the outsourcing member.

## 6.2 Outsourcing Setup

If AMQP as a transport layer is to be used, then the outsourcer has to create a client certificate to authenticate the insourcing firm application to the AMQP servers. The public part of the certificate has to be delivered to Eurex Clearing using the Member Portal. The outsourcer has to provide the private key to the insourcing firm.

The outsourcers are advised to use a secure transport mechanism when providing the private key to the insourcing firm. If WebSphere MQ as a transport layer is to be used, then the outsourcer needs to be configured on Eurex Clearing’s WebSphere MQ servers. The insourcing firm needs then access to the WebSphere MQ queues of the outsourcer.

As soon as the insourcing firm is able to connect to the AMQP broker or the WebSphere MQ server, the same technical setup can be performed as described in the previous chapters.

## 6.3 Simplified Outsourcing Setup

In order to reduce operational and technical effort, members may use the simplified outsourcing setup. This alternative allows the insourcing member to send requests via their own request queue. The Eurex Clearing System will check if the insourcing member is allowed to submit messages for the outsourcer and process them accordingly. The outsourcer will still receive all FIXML trade confirmation and position update messages via its own queues.

Eurex Clearing maintains a table of allowed member/user ID combinations for the simplified outsourcing setup. Consequently, the outsourcing partners are required to inform Eurex Clearing about the user ID applicable to their agreement. The respective forms are available at *https://www.eurex.com/ec-en/*
→ Services → Transaction management → Listed derivatives → Simplified Outsourcing
Simplified outsourcing can be used via both AMQP and WebSphere MQ.

## 6.4 FIXML Message Formatting

The insourcing firm has to send the identification of the acting user (e.g. OUT001) as part of the standard header for each FIXML message. In the standard header the field SID and SSub must be filled (please refer also to Volumes 3/4 for more details about the standard header).

The SID, the sender company ID (FIX tag 49), must be filled with the outsourcer Member ID and the SSub, the sender sub ID (FIX tag 50), must be filled with the provided user ID for the acting user.

# 7 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. |
| Broker | AMQP middleware messaging server. |
| 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/TLS certificates. |
| FIX | The Financial Information Exchange Protocol. |
| FIXML | FIX business messages in XML syntax. |
| FpML | Financial products Markup Language is the industry-standard protocol for complex financial products. It is based on XML. |
| 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. |
| NCM | Non-Clearing Member. |
| 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. |
| 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. |
| WebSphere MQ | Message oriented middleware from IBM |
| XML | Extensible Markup Language |
| | |