Eurex Clearing

Eurex Clearing Messaging Interfaces – Connectivity

D: Simplified AMQP broker setup for test installations

Version 1.0.0

Change History

Date	Ver.	Reason
15.02.2013	140.01	First public version
28.02.2014	1.0.0	– Former Volume(s) 2 are now stand-alone “Eurex Clearing Messaging Interfaces – Connectivity” documents – New versioning – Updated breadcrumb paths

Table of Contents

1 Introduction . . . 5
1.1 Overview . . . 5
1.1.1 Eurex Clearing FIXML Interface . . . 5
1.1.2 Eurex Clearing FpML Interface . . . 5
1.1.3 AMQP . . . 5
1.1.4 FIXML . . . 5
1.1.5 FpML . . . 5
1.2 Intended audience . . . 5
1.3 Eurex Clearing Messaging Interface Connectivity documentation . . . 6
1.4 Eurex Clearing FIXML Interface documentation . . . 6
1.5 Eurex Clearing FpML Interface and Margin Calculator Interface documentation . . . 7
1.6 Conventions used in this document . . . 7
1.7 Examples used in this document . . . 7
1.8 Organization of this document . . . 7
2 Simplified broker setup . . . 8
2.1 Installation . . . 8
2.1.1 Requirements . . . 8
2.1.1.1 Software requirements . . . 8
2.1.1.2 Network requirements . . . 9
2.1.2 Installing using rpm . . . 9
2.1.3 Installing manually . . . 9
2.2 Broker setup . . . 9
2.2.1 Preparing the directory structure . . . 10
2.2.2 Generate the broker certificate . . . 10
2.2.3 Generate client certificates . . . 10
2.2.4 Generating of PLAIN user accounts . . . 10
2.2.5 Generating broker configuration . . . 11
2.2.6 Generating the ACL file . . . 11
2.2.7 Starting the broker and creating AMQP objects . . . 11
2.3 Broker operations . . . 11
2.3.1 Stopping the broker . . . 11
2.3.2 Starting the broker . . . 11
2.3.3 Deleting the broker . . . 11
3 Glossary of terms and abbreviations . . . 12

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 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/

1.1.4 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:

→ http://www.fixprotocol.org/FIXimate3.0/

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.

1.1.5 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 5.0 is provided here:

→ http://www.fpml.org

To learn more about supported XML/FpML messages, please refer to “Volume 1: Overview”, “Volume 3: Trade Notification & Take-up Confirmation” and “Volume 3-A: Post Trade Events” which are available for download in the Member Section of the Eurex Clearing website.

1.2 Intended audience

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

1.3 Eurex Clearing Messaging Interface Connectivity documentation

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

- A: Overview
- B: AMQP Programming Guide
- C: Compiling Apache Qpid on Solaris
- D: Simplified Broker Setup for Test Installations (this document)
- E: AMQP Setup & Internals

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

For the Eurex Clearing Classic System:

http://www.eurexclearing.com → Technology → Eurex Clearing classic system → System documentation

For Eurex Clearing's C7:

http://www.eurexclearing.com → Technology → Eurex Clearing's C7 → System documentation

1.4 Eurex Clearing FIXML Interface documentation

The Eurex Clearing FIXML Interface documentation is organized as follows:

- Volume 1: Overview
- ~~Volume 2: Connectivity~~
  Has been replaced by Eurex Clearing Messaging Interfaces – Connectivity documentation, see above.
- Volume 3: Trade Confirmation & Capture
- Volume 4: Trade & 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 the Eurex Clearing Classic System:

http://www.eurexclearing.com → Technology → Eurex Clearing classic system → System documentation

For Eurex Clearing's C7:

http://www.eurexclearing.com → Technology → Eurex Clearing's C7 → System documentation

1.5 Eurex Clearing FpML Interface and Margin Calculator Interface documentation

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

- Volume 1: Overview
- ~~Volume 2: Connectivity~~
  Has been replaced by Eurex Clearing Messaging Interfaces – Connectivity documentation, see above.
- 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://member.eurexclearing.com → Clearing Resources → Releases → EurexOTC Clear → Interest Rate Swaps → IRS Release 3.0 → Interfaces → Eurex Clearing FpML Interface

1.6 Conventions used in this document

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

1.7 Examples used in this document

The Member **ABCFR** and the Eurex Clearing FIXML/FpML Interface account **ABCFR_ABCFRALMMACC1** are used in the examples in all chapters of this document.

1.8 Organization of this document

- Chapter 2 - Simplified broker setup
  - Gives information about the setup of the broker for test installations
- Chapter 3 - Glossary of terms and abbreviations
  - Glossary of terms and abbreviations used through the document

2 Simplified broker setup

When developing applications for the Eurex Clearing FIXML Interface or the Eurex Clearing FpML Interface, it may be helpful to have a testing environment with the configuration of the AMQP brokers which will be reasonably similar to the configuration of the Eurex Clearing FIXML/ FpML brokers in simulation and production. The AMQP broker it self is available as open source software and can be easily installed. But the configuration of the queues, exchanges or security is not available out of the box.

Therefore, a script which creates a simplified broker setup has been prepared by Eurex Clearing. The simplified broker includes following features:

- Queue and exchanges configuration for request, response and broadcasts
- ACL (access rights) configuration
- SSL encryption and authentication
- Persistent store

The simplified broker doesn't include:

- Clustering / High availability setup
- Emulation of the Eurex Clearing System (responding to requests, broadcasting functionality)

The script which prepares the simplified broker is written using Bash Unix shell and is named `amq_simple_setup.sh`. It has been written and tested on Red Hat Enterprise Linux 6.1 and CentOS Linux 6.2. Functionality under other Linux distributions or other Unix based operating systems is not guaranteed and may require some modifications to the script. Microsoft Windows operating systems are not supported.

The simplified broker is based on Red Hat MRG broker or on Apache Qpid broker. The development and tests have been done with Red Hat MRG 2.0 (Apache Qpid 0.10), Apache Qpid 0.12 and Apache Qpid 0.14.

The simplified broker doesn't require a dedicated HW. A virtual machine should be fully sufficient.

2.1 Installation

The `amq_simple_setup.sh` script can be downloaded from the Member Section of Eurex Clearing website in a form of TGZ archive or as an RPM package.

2.1.1 Requirements

2.1.1.1 Software requirements

The Apache Qpid software can be downloaded from the project website http://qpid.apache.org - available are pre-built RPM packages for Fedora distributions and source codes for other distributions. The Eurex Clearing FIXML Interface and the Eurex Clearing FpML Interface are using persistency layer, which is not part of the standard Apache Qpid source codes and has to be downloaded separately from http://www.qpidcomponents.org/ and build independently. For CentOS and Red Hat enterprise Linux, the RPM packages should be available in the package repositories.

Apart from the AMQP broker it self, also other software is necessary for successfully running the broker and using the `amq_simple_setup.sh` script:

- Qpid broker (package `qpid-cpp-server`)
- SSL support for the broker (package `qpid-cpp-server-ssl`)
- Command line tools for managing qpid broker (package `qpid-tools`)
- NSS Util (package `nss-util`)
- NSS Tools (package `nss-tool`)
- Java JRE 1.6.X or higher (package `java-1.6.0`)
- Cyrus sasl library with the plain plugin (package named `cyrus-sasl-plain`)

Package names listed above apply for the RHEL and CentOS Linux distributions. Software packages in different distributions may vary but will probably be similar.
When using the RPM package, the dependencies should be installed automatically. When using the TGZ archive on CentOS Linux or Red Hat Enterprise Linux, the software dependencies can be installed using following command:

```
# yum install qpid-cpp-server qpid-cpp-server-ssl python-qpid-qmf nss-tools java-
1.6.0 cyrus-sasl-plain qpid-tools
```

On other Linux distributions, the software dependencies have to be installed manually.

2.1.1.2 Network requirements

The AMQP connections are based on TCP/IP. The broker is always listening on two different ports: one for SSL communication and one for unencrypted communication. The connecting applications need to have open access to these ports.

2.1.2 Installing using rpm

The rpm contains information about its dependencies and after downloading it can be completely installed with one command:

```
# yum install amq_simple_setup-1.0-1.el6.noarch.rpm
```

After this, the script is installed in the `/usr/bin` directory and can be run with the simple command:

```
$ amq_simple_setup.sh --help
```

2.1.3 Installing manually

The archive downloaded from the Member Section of the Eurex Clearing website has to be unpacked and started with the command:

```
$ ./amq_simple_setup.sh --help
```

2.2 Broker setup

The broker prepared with our scripts always operates based on an “environment ID”. The environment ID is used to clearly distinguish the instances of the brokers, the directories used by the brokers as well as the used TCP/IP ports. The environment ID has to consist of a 4 digit number. The environment ID is specified using the `--env-id` option of the `amq_simple_setup.sh` script. Additionally, since the script is able to do not only setup the

broker, but to execute also other actions, the requested action has to be specified by the `--action` option.

At the beginning of the `amq_simple_setup.sh` script, a list of Member accounts is specified. By default it contains following Member accounts:

- ABCFR_ABCFRALMMACC1
- ABCFR_ABCFRALMMACC2
- DEFFR_DEFFRALMMACC1
- DEFFR_DEFFRALMMACC2

These example accounts can be used for testing. If required, the Member accounts can be changed directly in the shell script.

To start the broker setup/preparation the following command should be executed:

```
./amq_simple_setup.sh --env-id=<endID> --action=setup
```

The `<envID>` placeholder should be replaced with the ID of the environment which should be created. For example:

```
./amq_simple_setup.sh --env-id=1234 --action=setup
```

With this command, the script will start with the setup and preparation of the broker, which consists from several tasks.

2.2.1 Preparing the directory structure

The broker environment is installed into a directory named according to the environment ID. This directory is by default created in the current directory. That can be changed using the `--prefix` option. The broker environment has several directories. The temp directory contains the private keys of the generated certificates. The cert directory contains the public keys. The data directory contains the broker files, all configurations, certificate database and username database.

2.2.2 Generate the broker certificate

The broker certificate is used for the SSL encryption and authentication. It is stored in NSS certificate database. The public key is exported and stored in `cert/<node>.crt` and in `temp/truststore` (in Java Keystore format suitable for Java clients).

2.2.3 Generate client certificates

The self-signed certificates for the Member accounts are generated and their public keys are loaded into the brokers certificate database. The private keys are available in the temp directory in several formats. Since some of them require password protection, the password `123456` is used by default (can be changed in the `amq_simple_setup.sh` script). The files with private keys will be necessary for the Member applications to successfully connect and authenticate on the broker.

2.2.4 Generating of PLAIN user accounts

The broker has one user account for connecting without SSL, just by username and password. This account is named `administrator`. The password is the same as the username. The administrator account is an account without limited privileges - as such, it can be used for administration (adding/removing AMQP objects) and monitoring (number of messages in queues) of the broker. It can also be used to connect scripts/applications which will be sending the broadcasts or responding to requests.[1]

1. The `administrator` account doesn't exists on the real Eurex Clearing AMQP brokers - it exists only on the simplified broker.

2.2.5 Generating broker configuration

The broker configuration file is generated by the script. It configures the broker to listen on ports 1<envID> for SSL connections and 2<envID> for unencrypted connections (e.g. 11234 and 21234 based on the setup command above).

2.2.6 Generating the ACL file

The ACL file specifies the access rights for each account. The ACL file is generated by `amq_simple_setup.sh` to closely resemble the ACL setup used by real Eurex Clearing FIXML/FpML Interface brokers.

2.2.7 Starting the broker and creating AMQP objects

At the end, the broker is started and the appropriate AMQP objects are created. The queues, exchanges and bindings created by `amq_simple_setup.sh` closely resemble the real production setup. Their configuration is described in the interface specification, Volume 2.

2.3 Broker operations

Apart from broker setup, the `amq_simple_setup.sh` script can also be used to start, stop or delete the broker.

2.3.1 Stopping the broker

The broker can be stopped by using the action `stop`:

```
./amq_simple_setup.sh --env-id=1234 --action=stop
```

2.3.2 Starting the broker

The broker can be started by using the action `start`:

```
./amq_simple_setup.sh --env-id=1234 --action=start
```

2.3.3 Deleting the broker

The broker can be deleted by using the action `clean`. It will stop the broker and delete all its files:

```
./amq_simple_setup.sh --env-id=1234 --action=clean
```

3 Glossary of terms and abbreviations

Term/Abbr.	Definition
AMQP	Advanced Message Queuing Protocol - standard for messaging middleware.
Apache Qpid	Open source implementation of AMQP 0-10.
Binding	A binding is a relationship between a message queue and an exchange. The binding specifies routing arguments that tell the exchange which messages the queue should get.
Broker	AMQP middleware messaging server.
Eurex System	Eurex hosts.
Exchange	An exchange accepts messages from a producer application and routes them to message queues according to prearranged criteria.
EXTERNAL authentication	AMQP authentication mechanism based on SSL certificates.
FIX	The Financial Information Exchange Protocol.
FIXML	FIX business messages in XML syntax.
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.
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.
Red Hat MRG	Implementation of AMQP provided by Red Hat.
Routing key	A message property used in bindings to specify the exchange - queue relationship.
RPM	Software package format used on Red Hat based Linux operating systems.
SSL	Secure Sockets Layer - encryption of connections between Members and Eurex Clearing FIXML Interface.
TGZ	File archive created using the GZIP program.