2018-07-03 Version 0.9

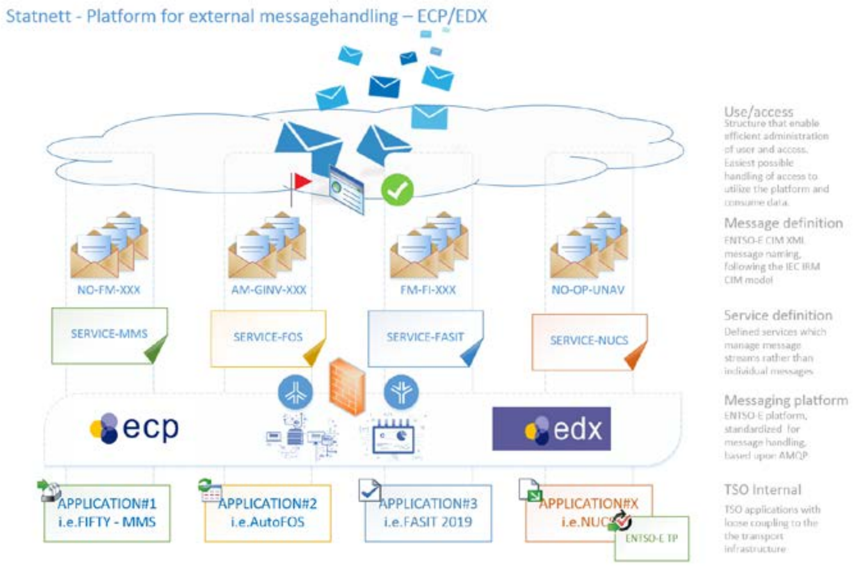
ECP – Platform for external message handling

Installation guide for ECP/EDX-components in Svenska kraftnät's service network

Index

Introduction

Below is an example from Statnett's environment.


Statnett - Platform for external messagehandling – ECP/EDX
NO-FM-XXX
AM-GINV-XXX
FM-FI-XXX
NO-OP-UNAV
SERVICE-MMS
SERVICE-FOS
SERVICE-FASIT
SERVICE-NUCS
ecp
edx
APPLICATION#1 i.e.FIFTY - MMS
APPLICATION#2 i.e.AutoFOS
APPLICATION#3 i.e.FASIT 2019
APPLICATION#X i.e.NUCS
ENTSO-E TP
Use/access
Structure that enable efficient administration of user and access. Easiest possible handling of access to utilize the platform and consume data.
Message definition
ENTSO-E CIM XML message naming, following the IEC IRM CIM model
Service definition
Defined services which manage message streams rather than individual messages
Messaging platform
ENTSO-E platform, standardized for message handling, based upon AMQP
TSO Internal
TSO applications with loose coupling to the the transport infrastructure

ECP is defined as the current standard for message handling between Svenska kraftnät and the market participants, and additional services will be added along the same standardized process.

Svenska kraftnät offers instructions and implementation support for market participants, their system vendors and others involved in the operation. We will reach out in relevant forums to inform and engage, as well as share necessary information and help for securing the needed momentum.

This installation guide is constructed to support the process for setting up the local installed components needed to take part in the communication network.

Support

Will not be in operation until after the summer of 2018.

Description	Purpose	Contact
Implementation support	Supporting the implementation between ECP and local Business application	ecp@svk.se
Installation support	Support for the local installation of needed technical components to participate in the ECP network	ecp@svk.se
<TBD> Operation support	<TBD> In ordinary operation Svenska kraftnät will offer technical support for participants in the network, based upon the User Agreement	<TBD>

1. Common prerequisites

1.1. ECP Endpoint Installation

1.1.1.General prerequisites before installing an ECP endpoint:

- **ECP endpoint installer** (jar or rpm package, depending on the deployment and OS type)
- **Registration keystore** (this keystore will be provided by your home component directory administrator/Statnett)
- **EIC-code** (issued by your Local Issuing Officer (LIO), Svenska kraftnät)

1.1.2. Firewall configuration

Ensure that your firewall allows communication as described in the table.

ECP Network communication protocol

ECP uses two main protocols for messaging – AMQPS and HTTPS. Please note that the port numbers specified below are default, however different ports may be explicitly configured for ECP depending on the installation environment.

Following is the list of all ports <TBD> that must be open to allow ECP communication:

Component	Function	Port	Protocol	Description
ECP endpoint	WAN			
	Outgoing	5671	Amqp + ssl	- to ECP brokers
		443	https	- to ECP component directories

Component	Function	Port	Protocol	Description
	Internal Network			NB! Not to be opened for external communication, only informative for the local network administration.
	Incoming from internal	5672	amqp	- AMQP API (only when the AMQP channel is enabled)
		8080	http	- Web service API, GUI and monitoring

1.2. Installation on Windows

The application may be installed in either command line or GUI mode. Both modes are described in this chapter.

1.2.1. Variables used in the guide

Several variables are used throughout this installation guide using the <variable_name> format. These variables are used only within this guide and are not to be confused with variables of your OS.

Name	Description
<jre_install_path>	Location of the JRE 8 used by the application
<install_path>	Path pointing to the target installation directory
<version>	Arbitrary version name
<ecp-jar>	Path to the ECP endpoint installation package, typically ecp-endpoint-<version>.jar

1.2.2. Prerequisites

Before processing the installation, check that the following prerequisites are met:

- ECP endpoint Windows installer (jar file).
- **Java Runtime Environment 8 64-bit with Java Cryptography Extension (JCE) is required.**
  - Check current Java version using the command:
    java version
  - Expected output:
    java version "1.8.0_60-ea"
- If the version does not start with “1.8” or the output is anything else:

  Download and install Java Runtime Environment 8 for your OS from the Oracle website: www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
- After installation, it is required to set JRE_HOME system variable. Open This PC, select Computer ->Properties -> Advanced system settings -> Environment Variables -> System Variables -> New -> Name: JRE_HOME, value: path to jre (e.g. C:\Program Files\Java\jre1.8.0_111).
- Download JCE from: www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

- Unzip the file into <jre_install_path>\jre\lib\security. All files should be overwritten.

The installer verifies that JCE is available at the beginning of the installation process. If this check fails, a warning message is displayed without aborting the installation.

Please make sure that MMC (Microsoft Management Console) or Services snap-in is not running during the installation as it prevents successful registration of the ECP service.

2. Installation – high level process for participation in Svenska kraftnät ECP Network - Platform for external message handling

- Communication is enabled by 2 local installed components; ECP and EDX
    - Downloadable from <TBD>
- Ensure that your network is configured to open for outgoing communication
    - Ref.: Chapter 1.1.2 Firewall configuration
    - Ports 5671 and 443 <TBD>
- Install and verify the ECP component on your local server
    - Installation procedure according to Chapter 3
    - This step involves communication with Svenska kraftnät
        - All general communication: ecp@svk.se
        - Retrieve ECP specific details (EIC Code, Keystore ++)
        - Retrieve project/service specific details
    - Finalized by the Endpoint Registration process in Chapter 3.7
- Install and verify EDX component on your local server
    - Installation procedure according to Chapter 4
    - Finalized by the EDX Service/Messaging process in Chapter 4.1.9
- Application specific communication with Svenska kraftnät
    - I.e. MMS applications, configuring the correct channel is needed, ref appendix A.
    - Normally this is managed by the system vendor/provider
- **Ready to go!**

3. Installation of ECP

3.1. Installation using GUI

1 Double-click the <ecp-jar> file. If Java was registered to execute jar files, an installer window should open. Skip to step 5. If an installer window does not open follow steps 2-4.

2 Open command line with administrator rights:

- Start > type “cmd” > right click > run as administrator

3 Change directory to location of ecp-endpoint.jar, e.g.

- cd c:\ecp-installer

4 Run the ecp-endpoint.jar file using

- java -jar <ecp-jar>

5 ECP endpoint installer window should open.

- Select target installation path
- Next
- Choose whether the ECP endpoint should be installed as a Windows service
- Next > Next
- If desired, an automatic installation script can be generated. It can be used in the future to install the application using the same options.
- Done

6 Configure the ECP endpoint **(see 3.6 ECP endpoint configuration)**

3.2. Installation using the command line

1 Open command line with administrator rights:

- Start > type “cmd” > right click > run as administrator

2 Change directory to the location of <ecp-jar>, e.g.

- cd c:\ecp-installer

3 Run the ecp-endpoint.jar file using

- java -jar <ecp-jar> console

4 Follow the console installation wizard

5 Configure the ECP endpoint **(see 3.6 ECP endpoint configuration)**

3.3. Starting and stopping ECP Endpoint

3.3.1. Using GUI

- If installed as a service:
  - Start > Run > “services.msc”
  - Find service with name like " ecp-endpoint " and description "ECP endpoint"
  - Start / stop service by buttons on the top left
- If not installed as a service:
  - Start / stop application by using
    - <install_path>\tomcat\bin\startup.bat
    - <install_path>\tomcat\bin\shutdown.bat

3.3.2. Using the command line

- If installed as a service:
  - Start service using
    - SC start ecp-endpoint
  - Stop service using
    - SC stop ecp-endpoint
- If not installed as a service:
  - Start application using
    - <install_path>\tomcat\bin\startup.bat
  - Stop application using
    - <install_path>\tomcat\bin\shutdown.bat

3.4. Installation verification

The application should be installed in the <install_path> folder. This installation folder should contain configuration files, Tomcat folder, and uninstaller. After the application is successfully started, it creates additional folders for data and log files. Default names and locations of these folders are <install_path>\data and <install_path>\logs.

3.4.1. Service status

The status of the service, if installed, can be checked either via command line:

- SC query ecp-endpoint

or in the Windows Services tool (see section 2.3.1).

3.4.2. Log files

Check the application log files for more information about its status. The default location is <install_path>\logs.

3.5. Uninstallation on Windows

Stop the application and then navigate to the installation directory. Change the directory to Uninstaller and execute the following command:

> java -jar uninstaller.jar –c

Uninstaller will remove all application components and deregister the service if present.

3.6. ECP Endpoint configuration

IMPORTANT: Before starting ECP endpoint, the application must be configured. The application is supplied with two configuration files.

Config file	Description
/etc/ecp-endpoint/ecp.properties	This is the main ECP configuration file with all the properties such as database connection, paths to filesystem, etc.
Svenska kraftnät definitions to be updated	ecp.endpoint.amqpApiEnabled = true
	ecp.endpoint.sendHandler[0].beanName=amqpApiSendHandler
	ecp.endpoint.sendHandler[0].typeName=*
	ecp.networks = DefaultNetwork
/etc/default/ecp-endpoint/ ecpendpoint.defaults	Configuration of JVM options (memory settings)

3.7. ECP Endpoint registration

3.7.1. Prerequisites

Before starting the endpoint registration please ensure that the following criteria are met:

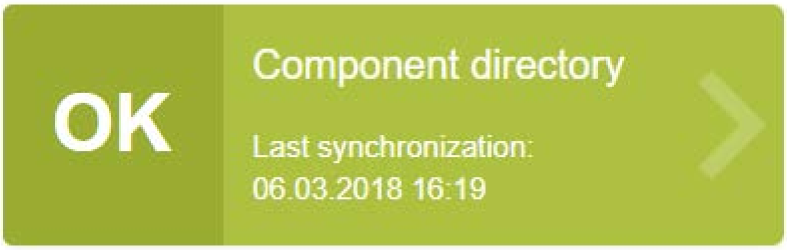
- The component directory administrator has provided you with the component directory URL, component directory code, EIC-code and registration certificate.
- Your home component directory is running.
- Communication with Svenska kraftnät is established, for registration and message validation.

3.7.2. Registration process

- Open a web browser and navigate to the URL of your endpoint (e.g. http://<endpoint-ip>:8080/ECP_MODULE).
- Insert the registration key store provided by your component directory administrator and wait for the file to upload. Click “Next” to proceed to the next step.
- Enter the component directory code <CD Directory Code> and URL <TBD> then click on the check connectivity. Please note that a successful connection check is required to proceed to the next step.
- Fill in your component code (EIC-code) and your company information (please note that the phone number must have an international country code - e.g.00<countrycode><number>) and click “Submit request”.
- At this point you must wait until the component directory administrator approves your registration request. Once the request is approved, your endpoint automatically downloads the results and becomes operational.

3.7.3. Registration validation

- Open a web browser and navigate to the URL of your ECP Endpoint (e.g. http://<endpoint-ip>:8080/ECP_MODULE).
- Check that the box indicating synchronization is green.
  - This confirms that the ECP Endpoint has reached the central components and is updated with appropriate settings.



OK

Component directory

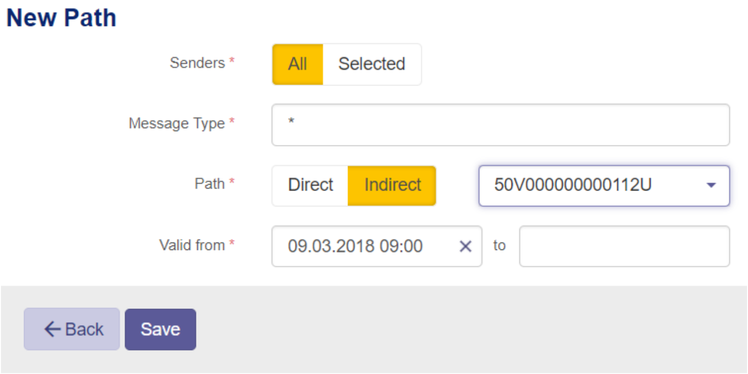
Last synchronization:
06.03.2018 16:19

3.7.4. Messaging validation

<TBD>

- Establish a message path in your ECP Endpoint.
  - Go to (e.g. http://<endpoint-ip>:8080/ECP_MODULE), choose tab "Settings" and click on the button named **"+ New Path"**

- Fill in according to the example below. Note that x112U is the broker in Svenska kraftnät’s test environment. When connected to the production environment a different EIC code will be visible in the drop down box.



Senders *: All

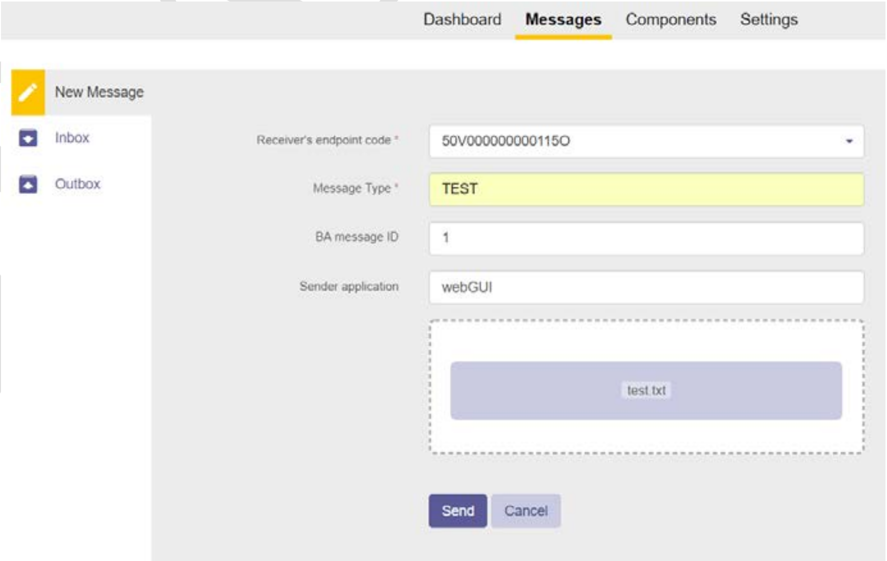
Message Type *: *

Path *: Indirect 50V000000000112U

Valid from * 09.03.2018 09:00 to

Click save, and choose the tab **"Messages"**

- Send a test message to Svenska kraftnät via the **"New Message"** link, fill in according to the example below (note: file for sending cannot be empty, size has to > 0kb)



Receiver's endpoint code * 50V000000000115O

Message Type * TEST

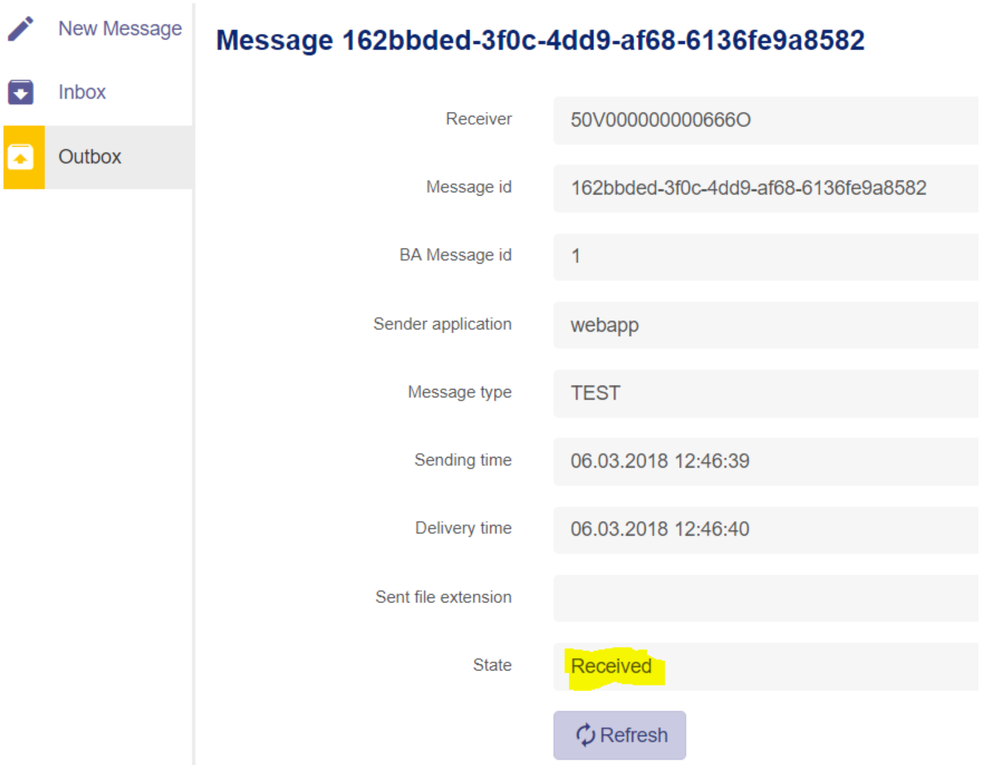
BA message ID 1

Sender application webGUI

test.txt

Make sure that the message is in state **"Received"** in the outbox view.

- This confirms that an appropriate communication between your local ECP Endpoint and Svenska kraftnät’s central components has been established.



Message 162bbded-3f0c-4dd9-af68-6136fe9a8582

Receiver	50V000000000666O
Message id	162bbded-3f0c-4dd9-af68-6136fe9a8582
BA Message id	1
Sender application	webapp
Message type	TEST
Sending time	06.03.2018 12:46:39
Delivery time	06.03.2018 12:46:40
Sent file extension	
State	Received

4. Installation of EDX

4.1. Installation on Windows

The application may be installed in either command line or GUI mode. Both modes are described in the Command line installation and Configure EDX Toolbox (see chapter 6 - Toolbox Configuration) GUI installation sections.

4.1.1. Variables

Several variables are used throughout this installation guide using the <variable_name> format.

Name	Description
<java_home>	Location of the JRE 8 used by the application
<install_path>	Path pointing to the target installation directory
<version>	Arbitrary version name
<edx-jar>	Path to the EDX Toolbox installation package, typically dx-toolbox-<version>.jar

4.1.2. Prerequisites

EDX Toolbox Windows installer (jar file).

Values for all configuration parameters are described in chapter 6 EDX Toolbox configuration, these are necessary for successful deployment of EDX Toolbox.

4.1.3. Command line installation

> Open command line with the administrator rights:

- Start > type “cmd” > right click > run as administrator

> Change directory to location of <edx-jar>, e.g.:

- cd c:\edx-installer

> Run the edx-toolbox.jar file using:

- java -jar <edx-jar> -console

> Follow the console installation wizard

> Configure EDX Toolbox **(see chapter 6.1 – Configuration of edx.properties)**

4.1.4. GUI Installation

1 Double-click the <edx-jar> file. If Java was registered to execute jar files, an installer window should open. Skip to step 5. If an installer window does not open follow steps 2-4.

2 Open command line with the administrator rights:

- Start > type “cmd” > right click > run as administrator

3 Change directory to location of edx-toolbox.jar, e.g.:

- cd c:\edx-installer

4 Run the edx-toolbox.jar file using:

- java -jar <edx-jar>

5 EDX Toolbox installer window should open.

- Select target installation path
- Next
- Choose whether EDX Toolbox server should be installed as a Windows service. The preferred option is to install EDX Toolbox as a Windows service
- Next > Next
- If desired, an automatic installation script can be generated. It can be used in the future to install the application using the same options
- Done

6 Configure EDX Toolbox (see chapter 6.1– EDX Toolbox Configuration)

4.1.5. Starting and stopping EDX Toolbox

4.1.5.1. Command line

If installed as a service:

- Start service using
  - SC start edx-toolbox
- Stop service using
  - SC stop edx-toolbox

If not installed as a service:

- Start application using
  - <install_path>\tomcat\bin\startup.bat
- Stop application using
  - <install_path>\tomcat\bin\shutdown.bat

Note: when starting the server using startup.bat, environment variable JAVA_HOME may need to be set to <java_home> path. Apache Tomcat displays a notification in this case.

4.1.5.2. GUI

If installed as a service:

- Start > Run > “services.msc”
- Find service with name like "edx-toolbox and description "edx-toolbox"
- Start/stop service by buttons on the top left

If not installed as a service:

- Start/stop application by using
  - <install_path>\tomcat\bin\startup.bat
  - <install_path>\tomcat\bin\shutdown.bat

4.1.6. Installation verification

The application should be installed in the <install_path> folder. This installation folder should contain configuration files, tomcat folder, and uninstaller. After the application is successfully started, it creates additional folders for data and log files. Default names and locations of these folders are <install_path>\data and <install_path>\tomcat\logs.

4.1.7. Service status

The status of the service, if installed, can be checked either via command line

- SC query edx-toolbox

or in the Windows Services tool (see chapter 3.1.5.2).

4.1.8. Log files

Check the application log files for more information about its status. The default location is <install_path>\tomcat\logs.

4.1.9. EDX Messaging / Service Validation

Ensure that ECP is configured according to ECP Endpoint configuration Chapter 2.6. Remember that ECP needs to restart after reconfiguration.

Select **"New message"** in the EDX Toolbox WebGUI.
3 different addressing policies are available, and have specified functions related.

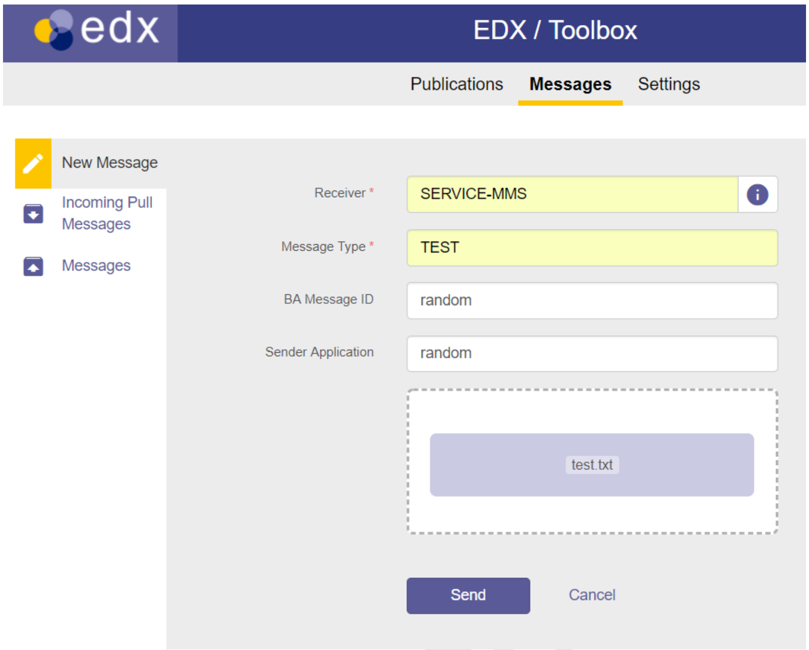
1 Receiver: <TBD> type: TEST, other values use "random"

2 Receiver: <TBD>type: TEST, other values as above

3 Receiver: <TBD> type: TEST, other values as above

Svenska kraftnät will for most use cases utilize the addressing policy named as 1 SERVICE-xxxx, since Svenska kraftnät is the primary party in the services published. This makes the addressing easy to understand and apply at the user side of ECP/EDX services.

NOTE: this example is for MMS.



Receiver * SERVICE-MMS

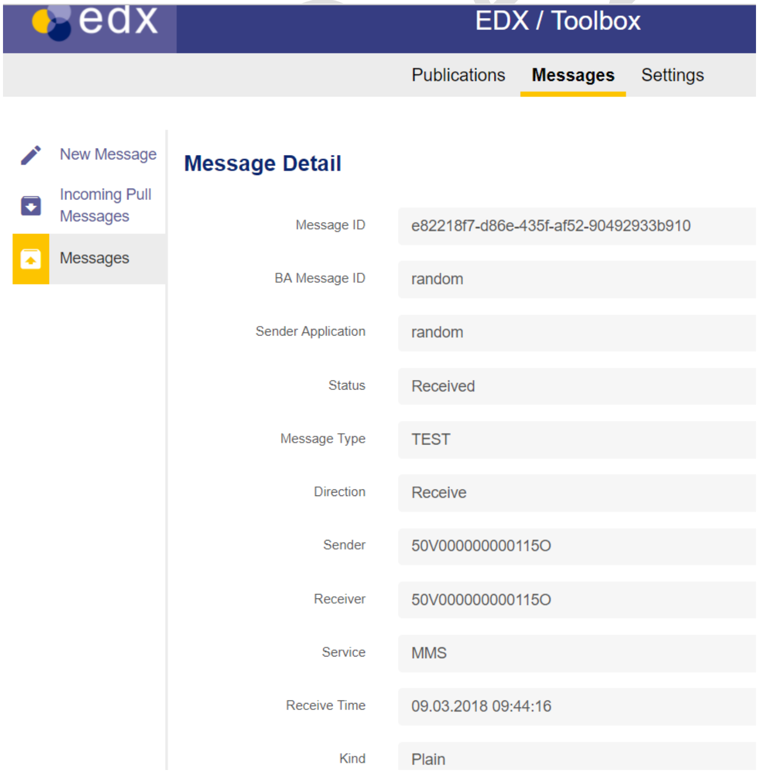
Message Type * TEST

BA Message ID random

Sender Application random

test.txt

In the Message outbox view the messages will be updated to **"Received"** state. This confirms that your local EDX Toolbox communicates over the ECP Network to Svenska kraftnät's central components.



Message Detail

Message ID	e82218f7-d86e-435f-af52-90492933b910
BA Message ID	random
Sender Application	random
Status	Received
Message Type	TEST
Direction	Receive
Sender	50V000000000115O
Receiver	50V000000000115O
Service	MMS
Receive Time	09.03.2018 09:44:16
Kind	Plain

5. Installation on Windows on same machine with ECP Endpoint

For installations of ECP Endpoint and EDX Toolbox on the same machine please set up ECP first, then follow standard installation procedure described in chapter 3 Installation of ECP. In addition to this procedure, it is necessary to change values of all used ports (http, internal tomcat ports). Configuration of used ports and JVM settings is different for installations where EDX Toolbox is registered as a Windows service versus the alternative, where it is not registered as a Windows service. Please also note that ECP Endpoint should be running when installing EDX Toolbox.

5.1. Configuration of internal broker

First it is necessary to **change the port of the internal broker** used by EDX Toolbox. Please change value of property internalBroker.amqp.port in configuration file <install_path>/edx.properties to a value which is not used by ECP Endpoint (by default 5672). It is possible to use commands like netstat –abn | more to find out which process uses which port.

5.2. Change ports – EDX Tollbox not registered as Windows service

When EDX Toolbox is not registered as a Windows service, it is possible to change values of ports in configuration file <install_path>/tomcat/bin/setenv.bat.
Note - this configuration file is not considered when EDX Toolbox is used as a Windows service.

Example of <install_path>/tomcat/bin/setenv.bat for deployment of ECP Endpoint and EDX Toolbox on same machine:

```
set "JAVA_OPTS=%JAVA_OPTS% -Xms64m -Xmx1024m -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=C:\Program
Files\EDX Toolbox\edx-dump.hprof edx.toolbox.tomcat.port.shutdown=9005 -
Dedx.toolbox.tomcat.port.http=9090 -Dedx.toolbox.tomcat.port.ajp=9009 -
Dedx.toolbox.tomcat.port.ajp.redirect=9443-
Dedx.toolbox.tomcat.port.http.redirect=9443 -
Dcom.sun.management.jmxremote.port=2099 -
Dcom.sun.management.jmxremote.rmi.port=2098 -
Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.ssl=false
Dcom.sun.management.jmxremote.authenticate=false" set CATALINA_OPTS="-
Dspring.config.location=file:C:\Program Files\EDX
Toolbox\edx.properties,file:C:\Program Files\EDXToolbox\edx.yml"
```

5.3. Change ports - EDX Toolbox registered as Windows service

When EDX Toolbox is registered as a Windows service, it is possible to change values of ports via configuration properties. These can be changed via Tomcat application for management of Windows services. Please also see chapter 6 EDX Toolbox installed as Windows service.

```
cd <install_path>/tomcat/bin
tomcat8w.exe //ES//edx-toolbox
```

GUI screen should appear, change tab to Java. In textbox Java Options it is possible to change system properties.

It is necessary to change at least the following port values:

Dedx.toolbox.tomcat.port.shutdown=8005

Dedx.toolbox.tomcat.port.http=8080

Dedx.toolbox.tomcat.port.ajp=8009

Dedx.toolbox.tomcat.port.ajp.redirect=8443

Dedx.toolbox.tomcat.port.http.redirect=8443

Dcom.sun.management.jmxremote.port=1099

Dcom.sun.management.jmxremote.rmi.port=1098

6. EDX Toolbox Configuration

Before starting EDX Toolbox, the application must be configured. The application is supplied with three configuration files on Linux:

Config file	Description
/etc/edx-toolbox/edx.properties	This is the main EDX configuration file with all the properties such as database connection, paths to filesystem, etc. Service Catalogue Code to be provided by the ECP administrator.
/etc/edx-toolbox/edx.yml	In this file it is possible to configure integration channels and EDX features like validations, routing, external processing etc.
/etc/default/edx-toolbox/ edx-toolbox.defaults	Configuration of used ports, memory settings.

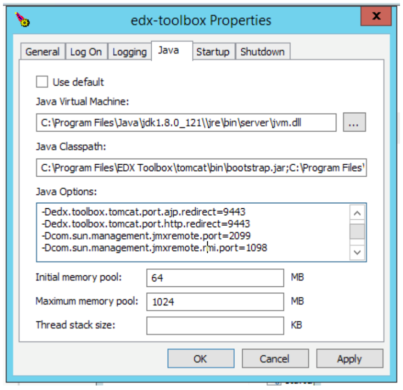
Two configuration files are supplied for Windows installations:

Config file	Description
<install_path>/edx.properties	This is the main EDX configuration file with all the properties such as database connection, paths to filesystem etc.
<install_path>/edx.yml>	In this file it is possible to configure integration channels and EDX features like validations, routing, external processing etc.

For configuration of system properties like used ports and memory settings:

System configuration like http ports, JVM memory setting can be set in Windows in two different locations, depending on whether the EDX toolbox is installed as a Windows service or not.

EDX Toolbox installed as Windows service



☐ Use default

Java Virtual Machine:
C:\Program Files\Java\jdk1.8.0_121\\jre\bin\server\jvm.dll

Java Classpath:
C:\Program Files\EDX Toolbox\tomcat\bin\bootstrap.jar;C:\Program Files'

Java Options:

```
-Dedx.toolbox.tomcat.port.ajp.redirect=9443
-Dedx.toolbox.tomcat.port.http.redirect=9443
-Dcom.sun.management.jmxremote.port=2099
-Dcom.sun.management.jmxremote.rmi.port=1098
```

Initial memory pool: 64 MB

Maximum memory pool: 1024 MB

Thread stack size: KB

When EDX Toolbox is registered as a Windows service, it is possible to change its system properties using the Tomcat application for management of Windows services. Please note that in this case, setenv.bat file is not considered by the application.

```
cd <install_path>/tomcat/bin
tomcat8w.exe //ES//edx-toolbox
```

GUI screen should appear, change tab to Java. In textbox Java Options it is possible to change system properties.

EDX Toolbox not installed as Windows service

When EDX Toolbox is not registered as a Windows service, it is possible to change values of system properties in configuration file

<install_path>/tomcat/bin/setenv.bat. Note - this configuration file is not considered when EDX Toolbox is installed as a Windows service.

6.1. Configuration of edx.properties

The edx.properties configuration file comes with many configuration parameters (all of them are described briefly in the edx.properties file and also in Appendix A). Most of the configuration parameters use default values, but the following parameters must be configured for each installation:

Parameters that require change for non-HA installation:

Parameter	Description
edx.toolbox.code	ECP endpoint code assigned to this Toolbox.
edx.serviceCatalogue.code	ECP endpoint code assigned to Service Catalogue.
edx.toolbox.pull.publicUrl.CODE	Public address of external pull interface for specific network (address of the server where Toolbox is installed). Replace CODE with physical network code from which this interface is accessible. If the toolbox is accessible from multiple physical networks, this property should be defined for each of them. It is also possible to have multiple interfaces for the same network, in this case value of this property is a comma separated list of URLs.
ecpBroker.amqp.port	Port number of the ECP AMQP Broker.
ecpBroker.amqp.host	Address of the ECP AMQP Broker

Appendix A

Configuring the edx.yml file

EDX Routing Configuration Example

Please note that the highlighted text exemplifies the use of amqp and shared file folder as integration channels from your business application and EDX. As application vendor/system supplier you will have to make a choice of integration channel.

```
#################Integration channels#######################
# Integration channels - general settings
# code - each endpoint shall have its unique code
# direction is "in" for input channels (from business applications to the toolbox) or out for output channels (from the toolbox to business applications)
# redeliveryAttemps is number of message delivery attempts, after these attempts message fails, default is 10, use -1 for unlimited redelivery attempts
# redeliveryDelay is delay between redelivery attempts, default is 5 seconds

### AMQP channels settings
# queueName is the name of input or output queue
# replyQueueName is mandatory for input AMQP endpoints, it is used for acknowledgements when message reaches its final state
# connectionURI is URI of the AMQP broker, when not specified EDX internal broker is used

<TBD>
amqpEndpoints:
  - {direction: in,  queueName: edx.endpoint.outbox, replyQueueName: edx.endpoint.reply, code: edx-endpoint-outbox}
  - {direction: out,  queueName: edx.endpoint.inbox, code: default-in-queue, redeliveryAttempts: 100}
  - {direction: out,  queueName: edx.endpoint.inbox.fasit, code: fasit-in-queue, redeliveryAttempts: 100}
  - {direction: out,  queueName: edx.endpoint.inbox.mms, code: mms-in-queue, redeliveryAttempts: 100}
  - {direction: out,  queueName: edx.endpoint.inbox.fos, code: fos-in-queue, redeliveryAttempts: 100}

### FTP channels settings
# protocol may be ftp, ftps or sftp
# hostname - hostname of the FTP server
# port - port number that should be used for FTP communication
# directory - directory under the FTP root that should be used as input or output message directory
# replyDirectory - used for input channels to deliver acknowledgements for sent messages, default is ".reply"
# failDirectory - used to store failed messages (e.g. messages that does not match naming convetions), default is ".failed"
# username - username used for FTP connection
# password - password used for FTP connection
# connectionParams - additional Camel connection parameters for special cases, see http://camel.apache.org/ftp.html for details

#  ftpEndpoints:
#    - {direction: in, protocol: ftp, hostname: 192.168.80.208, port: 21, directory: in-folder, replyDirectory: in/reply, username: user, password: password, code: my-ftp-endpoint-in, connectionParams: {stepwise: true, separator: UNIX}}
#    - {direction: out, hostname: 192.168.80.208, port: 21, directory: out-folder, code: my-ftp-endpoint}

### FSSF channels settings
# directory - local directory under that should be used as input or output message directory
# replyDirectory - used for input channels to deliver acknowledgements for sent messages, default is ".reply"
# failDirectory - used to store failed messages (e.g. messages that does not match naming convetions), default is ".failed"

  fssfEndpoints:
#     Example with absolute path on unix system
#    - {direction: in, directory: "/opt/data/fssf/in", code: fssf-endpoint-1}
#    - {direction: in, directory: "/opt/MadesEkstern/out", code: fssf-endpoint-1}
#     Example with absolute path on windows system
#    - {direction: out, directory: "c:/Program Files/EDX Toolbox/data/fssf/out/MT2", code: fssf-endpoint-3}
<TBD>
   - {direction: in, directory: "/opt/MadesEkstern/out", code: fssf-out-endpoint-1}
   - {direction: out, directory: "/opt/MadesEkstern/in/fasit", code: fssf-in-FASIT}
   - {direction: out, directory: "/opt/MadesEkstern/in/mms", code: fssf-in-MMS}

### ECP webservice endpoint is started automatically and can be also used in routing rules
# ECP webservice endpoint code for use as route start or end is "ecp-endpoint"

##################### Routing rules########################
# Routing rules for specific part of EDX processing
# Components to be used in defined routes, one or more codes of these components may be used as routing steps components:
# validate against xsd
  validations: []
#  - {code: my-validation-1, xsd: "file:c:/Program Files/EDX Toolbox/custom/xsd/person.xsd"}
#  - {code: my-validation-2, xsd: "file:/opt/edx/custom/xsd/person.xsd"}

# transform via xslt
  transformations: []
#  - {code: my-transformation-1, xslt: "file:c:/Program Files/EDX Toolbox/custom/xslt/person.xslt"}
#  - {code: my-transformation-2, xslt: "file:/opt/edx/custom/xslt/person.xslt"}

# let message be processed by external system
  externalProcessing: []
#  - {code: ext-processing-by-qap, in: edx.queue.qap.in, out: edx.queue.qap.out}
#  - {code: ext-processing-by-ba1, in: edx.queue.ba1.in, out: edx.queue.ba1.out}

# For all routing rules, it is possible to specify: messageType, service, sender, receiver or any combination of these attributes.
# Start of the route is receiving endpoint code, "*" for all input endpoints or "toolbox-gateway" for messages from another toolboxes.
# End of the route is one or more output endpoint's code(s) (including "ecp-endpoint" for ECP WS interface) or "toolbox-gateway" for messages that should be delivered to another toolbox/toolboxes.
# Fail endpoint is output endpoint used to deliver failure notification message.
# Steps are custom processing steps composed of components defined above. After the last step, message is passed to delivery.

# When route starts by "toolbox-gateway", it is incoming message from another toolbox for business application.
# When route starts by an endpoint code (defined in integrationChannels section), it is route for messages from an endpoint that can be routed to another toolbox using "toolbox-gateway" as end.
# Route that starts with an input endpoint code can also end with another output endpoint (message is not send to another toolbox) - receiver should be local toolbox.
# NOTE: Default route is delivering message to WS interface channel
<TBD>
routing:
  routes:
# Using AMQP
    - {code: route-FASIT, service: FASIT, start: toolbox-gateway, end: fasit-in-queue, fail: fasit-in-queue, steps: [] }
```

```
   - {code: route-MMS, service: MMS, start: toolbox-gateway, end: mms-in-queue, fail: mms-in-queue, steps: [] }
   - {code: route-FOS, service: FOS, start: toolbox-gateway, end: fos-in-queue, fail: mms-in-queue, steps: [] }
# Using file system
#    - {code: route-FASIT, service: FASIT, start: toolbox-gateway, end: fssf-in-FASIT, fail: fssf-in-FASIT, steps: [] }
#    - {code: route-MMS, service: MMS, start: toolbox-gateway, end: fssf-in-MMS, fail: fssf-in-MMS, steps: [] }

# default routes for outgoing messages (from business applications to anoter toolboxes) and incoming messages (from another toolboxes for business applications)
  sendProcessDefaultRoute: {start: "*", end: toolbox-gateway, fail: ecp-endpoint, steps: [] }
  receiveProcessDefaultRoute: {start: toolbox-gateway, end: ecp-endpoint, fail: ecp-endpoint, steps: [] }
```