



ENERGINET

Energinet
Tonne Kjærsvej 65
DK-7000 Fredericia

+45 70 10 22 44
info@energinet.dk
CVR-nr. 28 98 06 71

Dato:
23. juni 2020

Forfatter:
KBC/KBC

NOTAT

ECP ENERGINET INSTALLATION GUIDE

Content

1	Terminology	4
2	The big picture.....	6
2.1	Components in the Energinet ECP network.....	7
3	Considerations to make	8
3.1	PreProd (External Test) environment is recommended	8
3.2	Number of endpoints.....	8
3.3	Failover (and uptime).....	8
3.3.1	Business acknowledgement (needs no action on ECP servers).....	8
3.3.2	Failover server on standby (a modest investment to get decent uptime)8	
4	Contract Signing, Requirements, Prerequisites and Firewall	9
4.1	Prerequisites	9
4.2	Contract Signing.....	9
4.3	Software requirements.....	9
4.4	Hardware requirements for an endpoint	9
4.5	Firewall configuration	10
5	Installation of ECP-endpoint on Windows	11
5.1	ECP-endpoint configuration.....	11
5.1.1	Specify service properties	11
5.1.2	Configuration of ecp.properties	11
5.2	Starting and stopping ECP-endpoint.....	11
5.2.1	GUI.....	11
5.2.2	Command Line.....	11
5.3	Installation verification	11
5.4	ECP-endpoint Registration.....	12
5.4.1	Registration process	12
5.5	Message Path.....	14
5.6	Verify the installation.....	15
5.7	Troubleshooting.....	16
5.7.1	ECP-Endpoint does not respond on port 8080.....	16
5.7.2	Message status "Failed"	16
5.7.3	Message status "Accepted", not "Received"	17
5.7.4	Component Directory is not synchronized	17
5.7.5	You're able to send, but does not receive any messages.....	17
6	Installation of EDX-toolbox on Windows	18
6.1	EDX-toolbox Configuration	18
6.1.1	Specify service properties	18
6.1.2	Configuration of edx.properties	19
6.1.3	Configuration of edx.yml	20
6.2	Starting and stopping EDX-toolbox.....	20
6.2.1	GUI.....	20
6.2.2	Command Line.....	20
6.3	Installation verification	20
6.3.1	Send test messages	20
6.4	Configuring the edx.yml file.....	21
6.5	Troubleshooting.....	23

6.5.1	You have not received Service Catalogue (AKA network configuration)	23
6.5.2	You cannot send/receive on a particular service	23
7	Installation of ECP/EDX on Linux (Red Hat / CentOS / Fedora).....	24
8	Appendix.....	25
8.1	Administration of ECP-endpoint	25
8.2	Upgrade procedure.....	25

1 Terminology

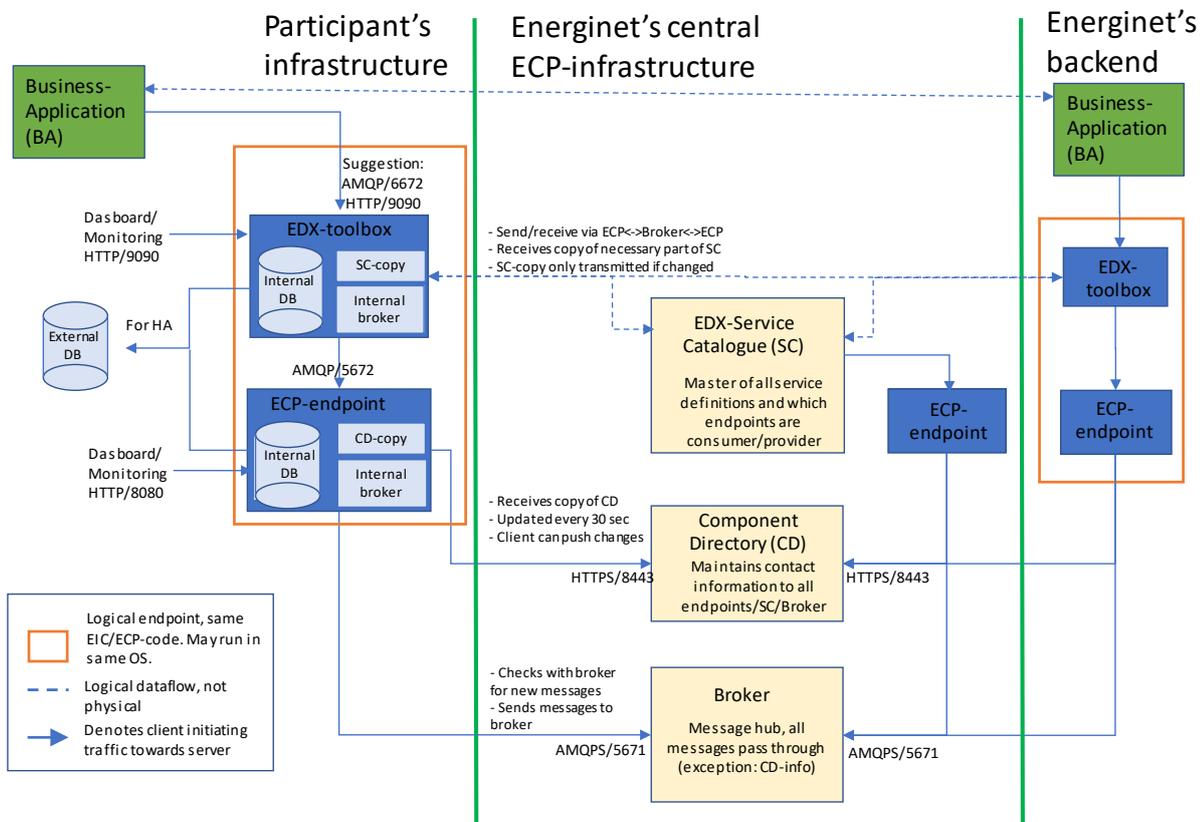
Read later - this is a reference for acronyms/names used in this installation guide. Defined words are in bold font.

Short	Expanded	Long
AMQP	Advanced Message Queuing Protocol	A protocol/standard developed in 2014 by OASIS for reliable (persisted) message communication.
BA	Business Application	A "normal" application outside ECP , communicating through ECP with other BAs . The BA must connect with an EDX-toolbox to send/receive messages from the network.
Broker	Central Broker	The central broker in an ECP -network is directly reachable for all ECP-endpoints and all messages in the network are sent to and retrieved from this broker. It supports AMQP(S) . In addition to the central broker, each ECP-endpoint and EDX-toolbox also have an "internal broker" (same type of broker) for message handling.
CD	Component Directory	An ECP -component/server, maintaining information about all ECP-endpoints , Broker and SC . This is like the phone book of an ECP -network.
EIC-code	Energy Identification Codes	These codes are provided by Energinet and identifies your particular endpoint (shared by ECP-endpoint and EDX-toolbox) and is stored in the CD .
ECP	Energy Communication Platform	A platform developed for ENTSO-E , by Unicorn, according to MADES 1.1/2.x specification – intended to provide secure and reliable messaging between the actors (TSOs and others) in the energy sector. The platform consists of EDX-toolbox , ECP-endpoints , Broker , Component Directory (CD) and Service Catalogue (SC) .
ECP-endpoint	ECP-endpoint	A specific component/server in the ECP -network, responsible for sending/receiving messages to/from the central Broker .
EDX-toolbox	EDX-toolbox	A "front" to the ECP-endpoint , logically a part of the same endpoint. EDX offers a richer set of interfaces for a BA to connect to. EDX has a Service concept which allows for more advanced routing of messages and addressing of endpoints.
Endpoint	Endpoint	An endpoint is the "logical endpoint" – a combination of both the ECP-endpoint and the EDX-toolbox .
ENTSO-E	European Network of TSOs	An organization of TSOs
Rush files	Rush Files	Energinet 's filesharing solution, all necessary documents for ECP and EDX are found there.
HA	High Availability	A term used for a database-setup, with multiple databases in a cluster. This is not part of the regular setup of ECP -endpoints, but it is possible to use MySQL or MSSQL for such a setup.
Hawtio	Hawtio	A monitoring software – access it on ECP-endpoint and EDX-toolbox on /hawtio on whatever port your webserver is running. You can browse you internal broker queues (see Broker).

MADES	Market Data Exchange Standard	A specification developed by ENTSO-E describing a communication system between actors in the energy sector.
SC	Service Catalogue	An EDX -component/server which keeps information about which endpoints consume/provides certain services. Without registration here, an EDX-toolbox cannot access services.
TSO	Transmission System Operator	Responsible for the distribution of energy (electricity or natural gas), in an area/country.
Statnett	Statnett	A norwegian TSO , the author of the first versions of this document

2 The big picture

The goal of ECP is to provide secure and reliable messaging between participants in the energy sector, from one Business Applications (BA) to another. The diagram below provides a “big picture”.



The diagram aims at answering the following questions you may have:

- What components exist in the ECP and which must a participant in the network install?
- Which ports must be opened in which direction and in which firewall?
- What is the general purpose of the various components in ECP?
- How does the messages flow in the system and how is the logical flow?

Please find below some explanations of what is seen in the diagram above:

- A participant must install an ECP-endpoint and an EDX-toolbox. If installed in the same OS, caution must be taken with respect to inbound TCP ports.
- During the installation phase you will need to register with the Component Directory (the registration requires approval by Energinet).
- After the approval, you may send messages from one ECP-endpoint to another.
- By adding the EDX "on top", you will gain the service concept which among many things will allow you to address a service provider without knowing a specific address (example of destination: "SERVICE-FASIT").
- You can therefore test ECP-to-ECP traffic (using the ECP-Dashboard) before you test EDX-to-EDX traffic (using the EDX-Dashboard).
- EDX-traffic will not work unless the Service Catalogue is updated by Energinet with information about your endpoint and which services you can consume.
- When EDX-to-EDX traffic works you may test BA-to-BA traffic.

2.1 Components in the Energinet ECP network

The following table lists the ECP /EDX components available in the ECP network hosted by Energinet. The PreProd network is used for external test.

Environment	Component	EIC code	URL
PreProd	Component Directory	45V000000000057R	https://iecp.preprod.energinet.dk:8443/ECP_MODULE
	Broker 01	45V000000000058P	amqps://iecp.preprod.energinet.dk:5671
PreProd	Endpoint Service Catalogue 01	45V000000000059N	-
PreProd	Service Catalogue 01	45V000000000059N	-
PreProd	Endpoint Energinet 01	45V000000000060I	-
PreProd	Toolbox Energinet 01	45V000000000060I	-
Prod	Component Directory	45V000000000053Z	https://iecp.prod.energinet.dk:8443/ECP_MODULE
Prod	Broker	45V000000000054X	amqps://iecp.prod.energinet.dk:5671
Prod	Endpoint Service Catalogue	45V000000000055V	-
Prod	Service Catalogue	45V000000000055V	-
Prod	Endpoint Energinet	45V000000000056T	-
Prod	Toolbox Energinet	45V000000000056T	-

3 Considerations to make

Please find below a list of considerations with respect to installing and running an endpoint.

3.1 PreProd (External Test) environment is recommended

Energinet recommends setting up a test-endpoint which will be connected to Energinet's Pre-Prod-network. **NB!** It is **not possible** to connect from PreProd to production (or vice versa) in the ECP-networks set up by Energinet. Unless you are confident on going straight into production, you should install a test-endpoint in PreProd first.

3.2 Number of endpoints

Energinet recommends as few endpoints as possible. We expect no performance issues. However, you may install as many endpoints as you think you need to, for example out of legal or management reasons.

3.3 Failover (and uptime)

All components may fail, but how to deal with it? Let's explore some options:

3.3.1 Business acknowledgement (needs no action on ECP servers)

First, we expect BAs to send and process "business acknowledgement" if necessary. Consider a scenario where BA-1 sends bid offer to BA-2. Then BA-2 responds with a "business acknowledgement" to let BA-1 know it has received the bid. Now BA-1 knows that its bid has been accepted and processed by BA-2 and can act accordingly. Failover-setups in the network cannot replace such a procedure, because it can never cover "end-to-end".

3.3.2 Failover server on standby (a modest investment to get decent uptime)

Energinet recommend a failover server on standby. Make sure to make a copy of the whole database of the master server after the master has completed registration.

This makes it possible for the failover server to startup without going through the registration process. Then make sure, with automatic or manual procedure, that only one of the servers are running at a time. If both are running at the same time, whichever of them are first to grab messages from the central broker, will take them.

4 Contract Signing, Requirements, Prerequisites and Firewall

4.1 Prerequisites

- You must have access to Energinet Rushfiles: <https://energinet.rushfiles.net/client/default.aspx>
You should find all relevant documentation under the folder, Energinet ECP Network. If you do not have access, contact plansupport@energinet.dk. Retrieve ECP installation files from Energinet Rushfiles.
- For each ECP-endpoint/EDX-toolbox installed, an **international** V-type EIC code is needed.
 - EIC codes are country specific 16-character strings, and must be requested from the body responsible for issuing EIC codes for the country where the company installing the endpoint is registered.
 - For companies registered in Denmark, please send an email to eic-administration@energinet.dk with information as specified on the web page <https://energinet.dk/El/Elmarkedet/EIC>
- With a V-type EIC code at hand, please write an email to plansupport@energinet.dk with the following information:
 - Company name and V-type EIC code
 - Which network you want to connect to (preprod or production)
 - Which services you are going to use
 - Contact information – email address and phone number
 - In return you'll get a registrationKeystore.jks file, that you will need later.

4.2 Contract Signing

To participate into the ECP-network of Energinet, one must sign an agreement. Contact Energinet settlement department: afregning@energinet.dk (the contract is available on the <https://energinet.dk/El/Ny-paa-elmarkedet/EIC> location mentioned above). The contract is clarification of each party's responsibilities.

4.3 Software requirements

Check the official Unicorn documentation for the OS requirements. The documentation is available from the Energinet Rushfiles.

4.4 Hardware requirements for an endpoint

Statnett has run this software for some time and makes some recommendations expecting a peak at a few hundred messages per hour. The official recommendation from ENTSO-E has much higher numbers based on a peak at a few thousands of messages per hour. ENTSO-E's recommendation is presented in parenthesis. The recommendation format is separate numbers for ECP and EDX separated by a plus-sign.

- CPU: 1C-2GHz + 1C-2GHz (4C-2GHz + 2C-2GHz)
- RAM: 4GB + 1GB¹ (8GB + 4GB)
- Disk: 20² GB + 20 GB (40 GB + 100 GB)

Even more important than a specific setup is the ability to upgrade the hardware on a short notice. Energinet, nor Statnett, does not assume any responsibility should these recommendations be inadequate. The software installations must be able to adapt to the load.

¹ Default setup for Java memory requirement in EDX is 1GB, but 2GB is preferable.

² Calculations indicate that 10 GB is plenty for 1M messages per 14 days (default behavior is to delete message logs older than 14 days). Then you have 10GB for other stuff. This is assuming you don't do archiving of message contents.

4.5 Firewall configuration

Look at "the big picture" (chapter 2) to identify which ports you need to open. The arrow on the traffic denotes from where the traffic is initiated. The table below summarizes the information:

Client	Server	Port	Protocol	Doing what?
Operator	EDX-toolbox	8080 or 9090	HTTP	Monitoring/Dashboard
BA	EDX-toolbox	8080 or 9090	HTTP	Message transport
BA	EDX-toolbox	5672 or 6672	AMQP	Message transport
EDX-toolbox	ECP-endpoint	5672	AMQP	Message transport
Operator	ECP-endpoint	8080	HTTP	Monitoring/Dashboard
ECP-endpoint	CD (prod): iecp.prod.energinet.dk	8443	HTTPS	Synch of ECP-network information
	CD (preprod): iecp.preprod.energinet.dk			
ECP-endpoint	Broker 01 (prod): iecp.prod.energinet.dk	5671	AMQPS	Message transport
	Broker 01 (preprod): iecp.preprod.energinet.dk			

The options you have on the port on EDX-toolbox depends on whether you install the EDX on the same OS as ECP-endpoint.

Also, if you want to secure the Operator Monitoring/Dashboard and BA Message transport ports with SSL, please consult the ECP / EDX administration guides.

If you want the software to connect to an external database, please make sure that no firewall blocks access to the port of that database. Supported database servers are MySQL, MSSQL, PostgreSQL and Oracle – for details on the specific versions supported, please check the ECP and EDX installation guides.

5 Installation of ECP-endpoint on Windows

Please consult the ECP Installation Guide on how to install the software on windows.

5.1 ECP-endpoint configuration

5.1.1 Specify service properties

When ECP-endpoint is registered as a Windows service, it is possible to change its system properties using the Tomcat application for management of Windows services (the method shown is cumbersome but apparently robust):

- `cd <install_path>\tomcat\bin`
- `copy tomcat9w.exe ecp-endpointw.exe (copy into <servicename>w.exe)`
- `ecp-endpointw.exe //ES//ecp-endpoint (see footnote for syntax explanation 3)`

When the property dialogue opens, change tab to Java. Consider changing the following settings:

- Tick off "Use default"-checkbox – important if you ever upgrade JRE on your system
- 4096 MB memory is more than enough, one could test with 2048 MB

5.1.2 Configuration of ecp.properties

Config file	Description
<code><install_path>\ecp.properties</code>	<p>This is the main ECP configuration file with all the properties such as database connection, paths to filesystem, etc. It is not necessary to change any property! However, you MUST ADD the following four properties:</p> <pre>ecp.endpoint.amqpApiEnabled = true ecp.endpoint.sendHandler[0].beanName=amqpApiSendHandler ecp.endpoint.sendHandler[0].typeName=* ecp.networks = Energinet-I-ECP-Network,DefaultNetwork</pre> <p>Later, when you're an expert, you can change other configuration settings.</p>

5.2 Starting and stopping ECP-endpoint

5.2.1 GUI

- Start `>services.msc`
- Find service with name like " ecp-endpoint " and description "ECP-endpoint"
- Start/stop service by buttons on the top left

5.2.2 Command Line

Start/stop application using the command "SC start/stop ecp-endpoint"

5.3 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`.

Check the status of the service, if installed: `>SC query ecp-endpoint` or in the Windows Services tool.

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

³ <https://tomcat.apache.org/tomcat-7.0-doc/windows-service-howto.html>

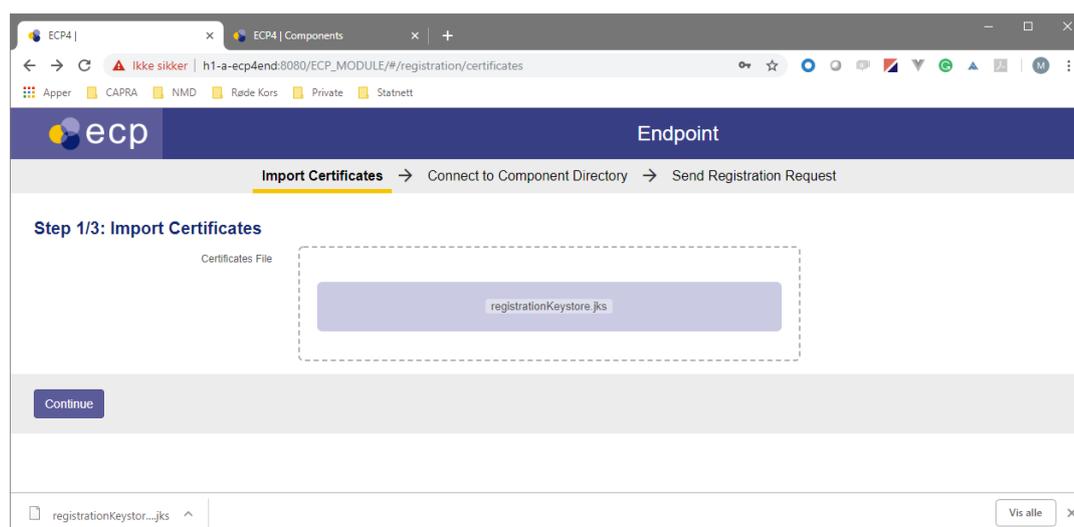
5.4 ECP-endpoint Registration

Screenshots from the registration process is shown on next page.

Please note that the screenshots shows the process of registering an endpoint the Statnett-hosted network, meaning that the URLs and EIC codes are different than those relevant for an the Energinet hosted network.

5.4.1 Registration process

- NB! Please read all the points below before going through the registration – to avoid rejection of the request!
- In a web browser, open: http://<YOUR-ECP-ENDPOINT>:8080/ECP_MODULE⁴, examples:
 - http://localhost:8080/ECP_MODULE
 - http://127.0.0.1:8080/ECP_MODULE
- Insert the registration key store provided by your component directory administrator and wait for the file to upload. Click “Continue” to proceed to the next step
- Enter the appropriate Component Directory code:
 - CD-code (preprod): 45V00000000057R (Energinet PreProd Network for use in external testing)
 - CD-code (prod): 45V00000000053Z
- Enter the appropriate URL (see below) and then click on the "check connectivity". Please note that a successful connection check is required to proceed to the next step
 - URL (preprod): https://iecp.preprod.energinet.dk:8443/ECP_MODULE
 - URL (prod): https://iecp.prod.energinet.dk:8443/ECP_MODULE
- Fill in your component code (EIC-code)
- **NB!** For contact email in production, please use a **monitored**⁵ email address. Energinet may decide to deny approval if a personal email-address is being used, because Energinet will use this email address to send information about upgrades and issues. In the test-network, personal email address may be allowed, but it should also be monitored (not abandoned or forgotten when people change positions) although less frequent than in production.
- Fill in the phone number (please note that phone number must be in international format – i.e. 00<country-code><number>) and click “Submit request”



⁴ If no response, see the Troubleshooting chapter.

⁵ Monitored means that some organization will be responsible for processing the email at least daily.

Endpoint

Import Certificates → **Connect to Component Directory** → Send Registration Request

Connectivity test was successful.

Step 2/3: Connect to Component Directory

Component Directory URL *

Component Directory Code *

Endpoint

Import Certificates → Connect to Component Directory → **Send Registration Request**

Step 3/3: Send Registration Request

Endpoint Code *

Company Information

Organization *

Contact Person *

Contact Email *

Contact Phone *

The registration request will be approved by your component directory administrator.

At this point, you must wait until the component directory administrator approves your registration request. You should send an email to plansupport@energinet.dk to notify of the request (no automatic notification is built into CD). There is a dashboard where you can monitor whether your endpoint has been approved or not – the "Component Directory" tile should be green:

Endpoint

Dashboard
Messages
Components
Settings

↓

244 Total messages received

238 Total messages sent

>

🕒

0 Messages to receive

2 Messages to deliver

>

📁

Business Applications

Received: 28.11.2018 14:26

Sent: 28.11.2018 14:26

>

OK

Certificates

Closest certificate expiration in **350** days: 14.11.2019 00:00

>

OK

Component directory

Last synchronization: 28.11.2018 14:27

>

OK

Message paths

There is 0 invalid message path.

>

Endpoint Code
50V0000000001150

Description

Version
4.3.0.750

Status
ACTIVE

Once the request is approved, your ECP-endpoint automatically becomes operational with respect to sending messages. It needs to be configured in order to be able to receive messages from other ECP-endpoints in the network.

5.5 Message Path

You must define a Message Path to tell how message are supposed to be routed **to** your endpoint. If you forget this step or enter something wrong in Message Type, you will **not receive any messages** (and you will not see any error messages or alerts!).

- Open a web browser and navigate to the URL of your ECP-endpoint (e.g. http://<endpoint-ip>:8080/ECP_MODULE), then choose menu "Settings"
- Choose "New Path"
- Set Message Type to "*", Path to "Indirect". The drop-down will show the only valid Broker to send/retrieve to/from.
- Press "Save" – approximately 30 seconds later this configuration should be known to everyone in the ECP-network. You may also use the "Push Configuration" button further down on the page if you suspect the automated push did go through.

New Path

Senders * All Selected

Message Type *

Path * Direct Indirect ▼

Valid from * × to

← Back Save

5.6 Verify the installation

- Open a web browser and navigate to the URL of your endpoint (e.g. http://<endpoint-ip>:8080/ECP_MODULE).
- Check that the box indicating synchronization is green



- Go to the "Messages" page and select the "New Message" tab.
 - Send a message to the endpoint just registered to verify the connectivity with the broker and that the endpoint is fully operational. The MESSAGE TYPE should be set to "TEST" and the file you send should be a small text file. Press the Send-button, wait a few seconds and then refresh the page. If the message (check Outbox) get the status "Received" the test is successful. See screenshots below.
 - Next, send a message to an Energinet endpoint. You can find which endpoint belong to which organization on the Components-page. The MESSAGE TYPE should be set to "TEST" and the file you send should be a small text file. Press the Send-button, wait a few seconds and then refresh the page. If the message (check Outbox) get the status "Received" the test is successful. See screenshots below.

Dashboard **Messages** Components Settings

New Message

Inbox

Outbox

Receiver's endpoint code * 50V000000000115O

Message Type * TEST

BA message ID 1

Sender application webGUI

test.txt

Send Cancel

New Message

Inbox

Outbox

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

Refresh

5.7 Troubleshooting

5.7.1 ECP-Endpoint does not respond on port 8080

- One possible reason is that you have other processes listening on the same port. Check logs/catalina.out – it should log "BindException: Address already in use". To find which process this is, run (in console as admin) "netstat -a -n -p". You can then identify the PID of the process listening on port 8080 and find that process in Task Manager. If you want to, you can change the 8080 port in tomcat\conf\server.xml and re-start the service.

5.7.2 Message status "Failed"

- Something is wrong with the configuration?

5.7.3 Message status "Accepted", not "Received"

- Your firewall does not allow outgoing traffic to the central Broker (see "Big Picture" in chapter 2)

5.7.4 Component Directory is not synchronized

- You're not approved yet – wait a little or remind the administrator of the ECP-network
- Your firewall does not allow outgoing traffic to the CD (see "Big Picture" in chapter 2)

5.7.5 You're able to send, but does not receive any messages

- Your Message Path is not defined in "Settings" (see chapter 5.5)

6 Installation of EDX-toolbox on Windows

Please consult the EDX Installation Guide on how to install the software on Windows

6.1 EDX-toolbox Configuration

Before starting EDX-toolbox, the application must be configured. The application is configured with service properties (6.1.1) and two configuration files (6.1.2 and 6.1.3). The table shows where to find the configuration files and the overall content of them.

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

6.1.1 Specify service properties

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:

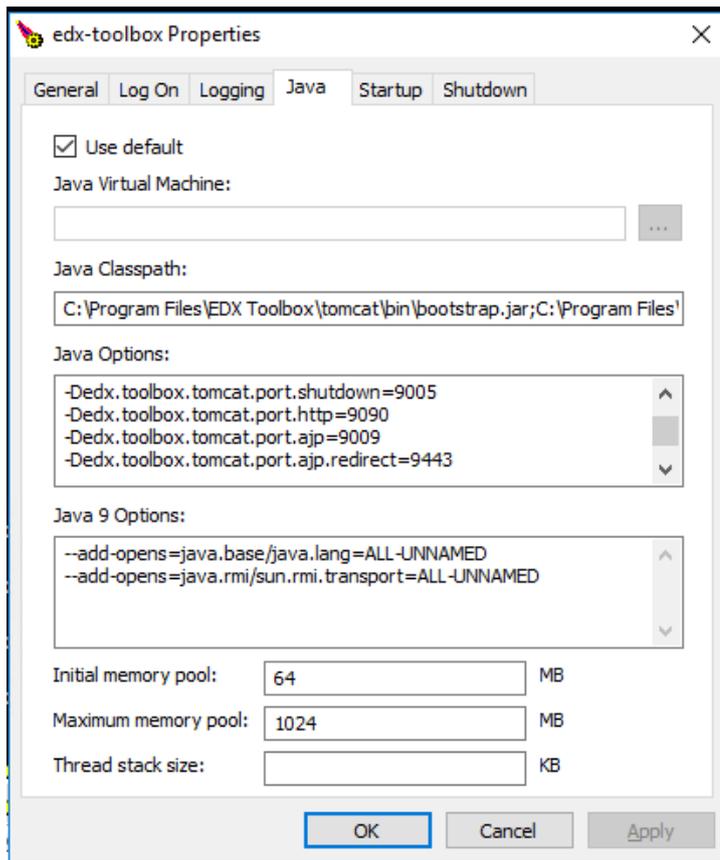
- `cd <install_path>\tomcat\bin`
- `copy tomcat9w.exe edx-toolboxw.exe (copy into <servicename>w.exe)`
- `edx-toolboxw.exe //ES//edx-toolbox (see footnote for syntax explanation 6)`

When the property dialogue opens, change tab to Java. Consider changing the following settings:

- Tick off "Use default" checkbox – important if you ever upgrade JRE on your system
- Although 1024 MB memory is probably ok, consider increasing it to 2048 MB if you have the capacity for it.
- If EDX runs in same OS as ECP, you need to change ports to avoid conflict. Here's a suggestion of port to set in Java Options:

```
-Dedx.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
```

⁶ <https://tomcat.apache.org/tomcat-7.0-doc/windows-service-howto.html>



Example of properties dialogue – not all Java Options are shown

6.1.2 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:

Parameter	Description
<code>edx.toolbox.code</code>	ECP-endpoint code assigned to this Toolbox. This is the same EIC-code you received in chapter 4.1 and which you specified in the last step of chapter 5.4.1.
<code>edx.serviceCatalogue.code</code>	In the preprod-network: 45V00000000059N In the prod-network: 45V00000000055V
<code>eCPBroker.amqp.port</code>	Port number of your ECP AMQP Broker, most likely 5672 (see Big Picture in chapter 2)
<code>eCPBroker.amqp.host</code>	127.0.0.1 if you install EDX in same OS as ECP. Otherwise, the IP/hostname of you ECP-endpoint.
<code>internalBroker.amqp.port</code>	6672 is recommended if you install EDX in the same OS as ECP. Otherwise you may keep default setting 5672 . This is the port your BAs will connect (if they choose to use AMQP).
<code>internalBroker.amqp.host</code>	0.0.0.0 – to indicate it accepts requests from anyone. 127.0.0.1 is the default configuration, which most likely ⁷ means that your BA must be running in the same OS (not recommended).
<code>edx.toolbox.pull.publicUrl.<Network></code>	If you want the BA to retrieve your messages simply as files on your server, specify this property. Replace <code><Network></code> with physical network code from which this interface is accessible.

⁷ This is not tested

Example of a correct setup could – assuming you chose 9090 as the HTTP port (see chapter 0 and 6.1.1):

```
edx.toolbox.pull.publicUrl.Internet=
http://localhost:9090/api/rest/pull/download/
```

6.1.3 Configuration of edx.yml

The configuration in the edx.yml deals with the various interfaces available to access EDX from the BA. Default settings (allow Web Service interface) is ok to begin with. You can use the default edx.yml configuration, move on to next chapter, and later change the edx.yml to suit your needs.

6.2 Starting and stopping EDX-toolbox

6.2.1 GUI

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

6.2.2 Command Line

Start/stop application using "SC start/stop edx-toolbox"

6.3 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. You could browse through edx-toolbox.log to see if any ERROR-entries occur – there should be none. One of the other logs will tell you if there is a port conflict (if so go back to chapter 6.1.1). The status of the service, if installed, can be checked either via command line: "SC query edx-toolbox" or in the Windows Services tool.

6.3.1 Send test messages

- Open a web browser and navigate to the URL of your endpoint (e.g. <http://<endpoint-ip>:9090/> - if you specified 9090 as your port). You should see this, except you won't have any messages in the list:

- Make a "New Message" and send to an Energinet endpoint. You can find which endpoint belong to which organization on the Components-page in the ECP-server. The MESSAGE TYPE should be set to "TEST" and the file you send should be a small text file. Press the Send-button, wait a few seconds and then refresh the page. If the message get the status "Received" the test is successful. See screenshot below:

The screenshot shows the 'New Message' form in the EDX / Toolbox interface. The form has the following fields and values:

- Receiver: 50V000000001150
- Message Type: TEST
- BA Message ID: (empty)
- BA Correlation ID: (empty)
- Sender Application: (empty)

A file named 'HalloVerden.txt' is attached to the message. The 'Send' button is highlighted in blue.

- The previous test was to send a message from one EDX to another, without specifying which service to use. Now, we want to test that your EDX has been added to the correct services. To test it, make a new message and set Receiver to SERVICE-
<SERVICENAME> (example: SERVICE-FASIT or SERVICE-MMS). The rest of the message is the same as in the previous test. This message will now be routed to the application providing this service. Since this message actually will be consumed by a proper BA, it's content should be just a simple text file (or even better – a real payload/message which is possible to process for the BA). If this test fails, it might be that your endpoint is still not added to the service by Energinet. Please contact plansupport@energinet.dk to investigate further.

6.4 Configuring the edx.yml file

This configuration file is about on which integration channel (Web Service (default), AMQP, FTP or file sharing) your services will be available to the BA accessing the EDX-toolbox.

Suggestion: Read from the bottom up of the edx.yml example below, it's perhaps a more logical way of reading:

- First, default routes are defined – that is, all traffic in/out of EDX will follow these rules
- Second, specific rules for specific services - they override the default rules. It was briefly mentioned in chapter 2 that you could specify address "SERVICE-FASIT" in a message. Messages with this address follow the FASIT-routes. **NB!** Please note that the default edx.yml has defined routes as an empty array "routes: []" – remember to remove the brackets before you add specific service routes.
- Third, we define what particular "integrationChannels" from which BA is supposed to connect to the EDX. **NB!** In case of AMQP, make sure that "queueName" matches the queueName used in the BA.

Confusion alert: In the configuration below, when the keyword **direction** is set to "in" it denotes traffic *from* BA *to* EDX-toolbox, in line with standard client-server paradigm. However all queue names are named in line with email-message-paradigm, so that "inbox" or "in" is going *from* EDX-toolbox *to* BA.

Example of an edx.yml – you will have to customize the file according to your needs. If you want to understand on how to configure this file, check out chapter 5/6 in "EDX User Guide" in <https://energinet.rushfiles.net/client/default.aspx>

```
##### Integration channels
#####
```

integrationChannels:

```
# 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)
# redeliveryAttempts 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
```

```
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 conventions), 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/re-
reply,
# username: user, password: password, code: my-ftp-endpoint-in, connectionParams: {stepwise: true, separa-
tor: 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 conventions), 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}
- {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 applica-
tion.
# 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
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: [] }

```

6.5 Troubleshooting

6.5.1 You have not received Service Catalogue (AKA network configuration)

Check settings on your EDX Dashboard. You should see that the network configuration has been updated at some point. If not, you cannot send/receive messages from EDX. The reason may be because you cannot receive files – see chapter 5.7.5. Another reason may be that Energinet has not updated the Service Catalogue with your endpoint (this is a manual process at Energinet). Please notify Energinet at plansupport@energinet.dk if you suspect this to be the case.

6.5.2 You cannot send/receive on a particular service

If you can send messages unrelated to a specific service (ex-address: the endpoint-code for Service Catalogue in chapter 6.1.2), but cannot send to the service you're supposed to be a part of (ex-address: SERVICE-FASIT), then the error may be that the Service Catalogue has not been updated properly (this is a manual process in Energinet). Please notify Energinet at plansupport@energinet.dk if you suspect this to be the case. Also, please check the Services->Consumed menu on the EDX Dashboard. A list of services should appear to show which services your endpoint may "consume".

7 Installation of ECP/EDX on Linux (Red Hat / CentOS / Fedora)

- Find installation guides which cover Linux installation in EDX Installation Guide:
<https://energinet.rushfiles.net/client/default.aspx>
- Contact plansupport@energinet.dk if you have more questions. If many requires Linux installation, we can expand this installation guide.

8 Appendix

8.1 Administration of ECP-endpoint

The ECP-server offers a Settings-page which has some important features:

- Message Path: Check that you have a path defined for message type "*", otherwise you cannot receive some or even any messages.
- Certificates: You can delete old certificates (Preferred = No, Valid to < Today). Old certificates may make noise in your logs.
- Message Connectivity: Connectivity check is the simplest way to test if another ECP-Endpoint is reachable. If no endpoint is reachable, the conclusion will usually be that your own endpoint has lost connection to the (central) Broker. In that case try to telnet directly from the endpoint and from another location to determine whether the problem is on Energinets end of things or your own.
- Component Directory: You should push the configuration after you upgrade to a new version, to allow the CD to know the version number of all endpoints.
- Component Directory: Connectivity check to see if the CD (over port 443) is available. Data is exchanged here every 30 seconds. If CD is offline your ECP endpoint cannot send/receive messages.

The ECP-server offers a Dashboard which shows

- Component Directory synchronization is ok (or not)
- Certificates are valid (or not)
- Messages are delivered (or not)

8.2 Upgrade procedure

In the official ECP/EDX documentation (<https://energinet.rushfiles.net/client/default.aspx>) you'll find documents for upgrading EDX and ECP. Read and follow these guides:

- EDX Upgrade Guide
- ECP Upgrade Guide

Before doing an upgrade, please pay close attention to any changes in the configuration files.

Also, reading the Release Notes may provide valuable knowledge on what to pay close attention to when upgrading or having upgraded the software.