

**Visiting address**

Nydalen allé 33, 0484 Oslo

**Postal address**

PB 4904 Nydalen, 0423 Oslo

**Enterprise No.**

NO 962 986 633 MVA

**T** +47 23 90 30 00

**F** +47 23 90 30 01

**W** [statnett.no](http://statnett.no)

**E** [firmapost@statnett.no](mailto:firmapost@statnett.no)

# Installation and Upgrade Guide for a participant's ECP/EDX-endpoint in Statnett's network

Version: 4.8.1.1

Date: 08.10.2021

# Statnett

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>DOCUMENT HISTORY</b>   | <b>3</b>  |
| <b>2</b> | <b>TERMINOLOGY</b>  | <b>6</b>  |
| <b>3</b> | <b>THE BIG PICTURE</b>  | <b>7</b>  |
| <b>4</b> | <b>CONSIDERATIONS TO MAKE</b>                                     | <b>8</b>  |
| <b>5</b> | <b>CONTRACT SIGNING, REQUIREMENTS, PREREQUISITES AND FIREWALL</b> | <b>9</b>  |
| <b>6</b> | <b>INSTALLATION OR UPGRADE OF ECP-ENDPOINT ON WINDOWS</b>         | <b>11</b> |
| <b>7</b> | <b>INSTALLATION OR UPGRADE OF EDX-TOOLBOX ON WINDOWS</b>          | <b>19</b> |
| <b>8</b> | <b>INSTALLATION OF ECP/EDX ON LINUX (RED HAT 7 OR CENTOS 7)</b>   | <b>28</b> |
| <b>9</b> | <b>APPENDIX</b>   | <b>29</b> |

## 1 Document History

| Version | Date       | Changes   |
|---------|------------|---|
| 1.00    | 20/11-2018 | <ul style="list-style-type: none"> <li>- A complete rewrite of an older version (up-to-date with ECP-endpoint 4.2.0 and EDX-toolbox 1.3.1)</li> </ul>   |
| 1.02    | 22/11-2018 | <ul style="list-style-type: none"> <li>- Improved description of service properties on ECP configuration (chapter 6.3.1)</li> <li>- Improved description of service properties on EDX configuration (chapter 7.3.1)</li> <li>- Added a chapter on how to switch network or to install in other TSO's network.</li> <li>- Many small fixes to wordings, spelling, etc.</li> </ul>  |
| 1.03    | 23/11-2018 | <ul style="list-style-type: none"> <li>- Firewall chapter (5.5) includes actual hostnames of Statnett's CD and Broker</li> </ul>  |
| 1.04    | 23/11-2018 | <ul style="list-style-type: none"> <li>- Corrected wrong client name in firewall chapter (5.5)</li> </ul>   |
| 1.05    | 23/11-2018 | <ul style="list-style-type: none"> <li>- Added troubleshooting of non-responding port 8080 on ECP-endpoint and how to change that particular port.</li> </ul>   |
| 1.06    | 23/11-2018 | <ul style="list-style-type: none"> <li>- Added chapters about monitoring (9.3) and administration (9.4)</li> <li>- Added contact email policy in chapter 6.6.1</li> <li>- Added standalone server recommendation in chapter 5.4</li> </ul>  |
| 1.07    | 26/11-2018 | <ul style="list-style-type: none"> <li>- Clarified how to read edx.yml configuration in chapter 7.3.3</li> </ul>  |
| 1.08    | 26/11-2018 | <ul style="list-style-type: none"> <li>- Updated chapter explanation about Message Path in chapter 6.7</li> <li>- Updated comment on internalBroker.URL in chapter 7.3.2</li> </ul>   |
| 1.09    | 27/11-2018 | <ul style="list-style-type: none"> <li>- Miniscule word changes</li> <li>- Terminology long explanation references in bold</li> <li>- Updated chapter 5 with software requirement, moved hardware requirement from chapter 4 to 5.</li> <li>- Updated section about hardware requirements with ENTSOE's recommendations and adjusted Statnett's recommendations.</li> <li>- Added chapter 5.2 about contract signing</li> </ul> |
| 1.10    | 28/11-2018 | <ul style="list-style-type: none"> <li>- Improved chapter 7.5 with verification tests</li> <li>- Changed what to send to Statnett in chapter 5.1</li> <li>- Moved configuration of edx.yml from appendix to chapter 7.6</li> </ul>  |
| 1.11    | 30/11-2018 | <ul style="list-style-type: none"> <li>- Minor changes to make it harder to misunderstand</li> <li>- Improved chapter 9.3.2 about monitoring .</li> </ul>   |
| 1.12    | 10/12-2018 | <ul style="list-style-type: none"> <li>- Added chapter <b>Feil! Fant ikke referanseilden.</b> about how to upgrade EDX/ECP</li> </ul>   |
| 1.13    | 12/12-2018 | <ul style="list-style-type: none"> <li>- Added a screenshot to chapter <b>Feil! Fant ikke referanseilden.</b></li> <li>- Added troubleshooting-chapter 0 for EDX</li> </ul>   |
| 1.14    | 12/12-2018 | <ul style="list-style-type: none"> <li>- Added user/pass for Hawtio-login (chapter 9.3)</li> <li>- Added two tips on how to edit the edx.yml</li> </ul>   |
| 1.15    | 07/02-2019 | <ul style="list-style-type: none"> <li>- Tiny changes in reading instructions</li> <li>- Added reference to more in-depth documentation of the edx.yml in chapter 7.6</li> </ul>  |
| 1.16    | 15/03-2019 | <ul style="list-style-type: none"> <li>- Added an important notice about the crypto.policy.</li> <li>- Switch order of the URL/CD-code for readability</li> </ul>   |
| 1.17    | 22/03-2019 | <ul style="list-style-type: none"> <li>- Rearranging some content into chapter <b>Feil! Fant ikke referanseilden.</b> and updated info about Java</li> <li>- Some minor changes to the text</li> <li>- Aligned with the new Update Guide (v1.0)</li> </ul>  |

|      |            |  |
|------|------------|--|
| 1.18 | 15/04-2019 | <ul style="list-style-type: none"> <li>- Improved settings for connectivity-check (9.3.2.1)</li> <li>- Fixed a wrong description on implication of CD being offline (9.4)</li> </ul>   |
| 1.19 | 25/04-2019 | <ul style="list-style-type: none"> <li>- Tiny change in the example of edx.yml configuration</li> </ul>  |
| 1.20 | 29/05-2019 | <ul style="list-style-type: none"> <li>- Small improvement i chapter 7.6 regarding which broker to choose</li> </ul>   |
| 2.00 | 11/02-2020 | <ul style="list-style-type: none"> <li>- MAJOR CHANGE: INSTALLATION of ECP v4.6 AND EDX v1.7.</li> <li>- New and simplified agreement to be signed to participate in the network (chapter</li> <li>- Higher HDD-requirement for EDX (chapter 5.4)</li> <li>- Updated links and description about JRE update (chapter <b>Feil! Fant ikke referansekinden.</b>)</li> <li>- Updated documentation about configuration (chapter 6.3)</li> <li>- Updated some text and screenshots in registration (chapter 6.6)</li> <li>- Added section about automatic renewal (chapter <b>Feil! Fant ikke referansekinden.</b>)</li> <li>- Updated documentation about configuration (chapter 7.3)</li> <li>- Major update of edx.yml-configuration (chapter 7.6)</li> </ul>  |
| 2.01 | 23/04-2020 | <ul style="list-style-type: none"> <li>- Changed an error for spring-profile in EDX-configuration part</li> </ul>  |
| 2.02 | 26/05-2020 | <ul style="list-style-type: none"> <li>- Small change in explanation of edx.yml to point out which ServiceCatalogue codes to use for test and production</li> </ul>  |
| 2.03 | 27/05-2020 | <ul style="list-style-type: none"> <li>- Installation procedure of JRE has been improved</li> </ul>  |
| 2.04 | 28/05-2020 | <ul style="list-style-type: none"> <li>- Small, but importat fix in edx.properties configuration ('ecp-nonha' changed to 'edx-nonha')</li> <li>- Added a warning about to have only one yml-file in EDX-config directory</li> </ul>  |
| 2.05 | 09/06-2020 | <ul style="list-style-type: none"> <li>- Adjustments to the JRE-installation (chapter <b>Feil! Fant ikke referansekinden.</b>)</li> <li>- Warning about Windows 2019 installation – it has never been supported! (chapter 5.3)</li> </ul>  |
| 2.06 | 08/09-2020 | <ul style="list-style-type: none"> <li>- Improved the automatic monitoring sub (chapter 9.3.2)</li> </ul>  |
| 2.07 | 25/09-2020 | <ul style="list-style-type: none"> <li>- Improved more of the appendix-part (chapter 9)</li> </ul>   |
| 2.08 | 01/10-2020 | <ul style="list-style-type: none"> <li>- Improved section about monitoring (chapter 9.3.2)</li> </ul>  |
| 2.09 | 16/04-2021 | <ul style="list-style-type: none"> <li>- Important changes in ecp.properties for a stable process (chapter 6.3.2)</li> <li>- Added a reference to a new document for high performance/ high uptime endpoint (chapter 9.5)</li> </ul>   |
| 2.10 | 22/04-2021 | <ul style="list-style-type: none"> <li>- Small change in the doc about JRE-version</li> </ul>  |
| 3.00 | 17/08-2021 | <ul style="list-style-type: none"> <li>- MAJOR CHANGE: INSTALLATION of ECP v4.7.x and EDX v1.8.x</li> <li>- Chapter 4.2 (considerations to make) has been rewritten to reflect the latest and most up-to-date understanding of how to run ECP/EDX.</li> <li>- Chapter 5.1 has been rewritten to reflect new location for binaries &amp; software and proper version to download</li> <li>- Chapter <b>Feil! Fant ikke referansekinden.</b> has been rewritten to highlight some information about JRE</li> <li>- Chapter 6.6 has been rewritten to highlight a change from default non-SSL ports to SSL ports</li> <li>- Chapter 6.7/7.5.1 has been rewritten to the change to SSL-ports and to make sure "valid-from timestamp" is set correctly</li> <li>- Added chapter 6.9/7.7 to inform about timezone settings</li> <li>- Briefly mentions updates to v4.8 and v1.9, points to other doc.</li> </ul> |

# Statnett

|         |            |   |
|---------|------------|---|
| 3.01    | 25/08-2021 | <ul style="list-style-type: none"><li>- AMQPS between ECP and EDX has become default, in contrast to previous installations. Chapter 6.3.2 and 7.3.2</li></ul>  |
| 4.8.1   | 04/10-2021 | <ul style="list-style-type: none"><li>- MAJOR CHANGE: INSTALLATION of ECP 4.8.1 and EDX 1.9.1</li><li>- Re-use the official documentation from Unicorn to a higher degree</li><li>- A major overhaul of the entire document</li><li>- Version numbering on this document will be kept in sync with ECP versions</li></ul> |
| 4.8.1.1 | 08/10-2021 | <ul style="list-style-type: none"><li>- Updated configuration of edx/ecp.properties and changed a screenshot and upgraded a property in the kafka configuration in edx.yml</li></ul>  |

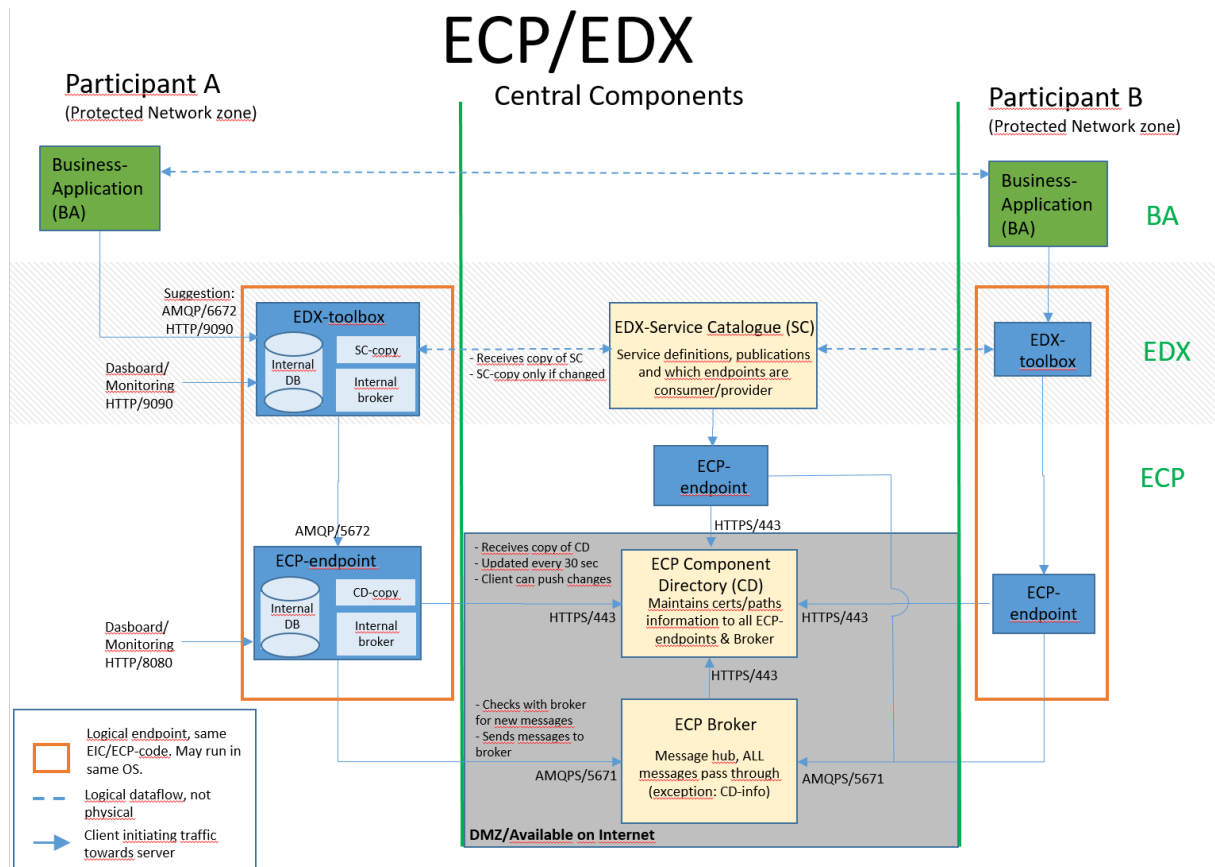
## 2 Terminology

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 <b>ECP</b> , communicating through <b>ECP</b> with other <b>BAs</b> . The <b>BA</b> must connect with an <b>EDX-toolbox</b> to send/receive messages from the network.   |
| Broker       | Central Broker                    | The central broker in an <b>ECP</b> -network is directly reachable for all <b>ECP-endpoints</b> and all messages in the network are sent to and retrieved from this broker. It supports <b>AMQP(S)</b> . In addition to the central broker, each <b>ECP-endpoint</b> and <b>EDX-toolbox</b> also have an "internal broker" (same type of broker) for message handling.                |
| CD           | Component Directory               | An <b>ECP</b> -component/server, maintaining information about all <b>ECP-endpoints</b> , <b>Broker</b> and <b>SC</b> . This is like the phone book of an <b>ECP</b> -network.  |
| EIC-code     | Energy Identification Codes       | These codes are provided by Statnett and identifies your particular endpoint (shared by <b>ECP-endpoint</b> and <b>EDX-toolbox</b> ) and is stored in the <b>CD</b> .   |
| ECP          | Energy Communication Platform     | A platform developed for <b>ENTSO-E</b> , by Unicorn, according to <b>MADES 1.1/2.x</b> specification – intended to provide secure and reliable messaging between the actors ( <b>TSOs</b> and others) in the energy sector. The platform consists of <b>EDX-toolbox</b> , <b>ECP-endpoints</b> , <b>Broker</b> , <b>Component Directory (CD)</b> and <b>Service Catalogue (SC)</b> . |
| ECP-endpoint | ECP-endpoint                      | A specific component/server in the <b>ECP</b> -network, responsible for sending/receiving messages to/from the central <b>Broker</b> .  |
| EDX-toolbox  | EDX-toolbox                       | A "front" to the <b>ECP-endpoint</b> , logically a part of the same endpoint. <b>EDX</b> offers a richer set of interfaces for a <b>BA</b> to connect to. <b>EDX</b> 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 <b>ECP-endpoint</b> and the <b>EDX-toolbox</b> .  |
| ENTSO-E      | European Network of TSOs          | An organization of <b>TSOs</b>  |
| eRoom        | eRoom                             | Statnett's filesharing solution, all necessary documents for <b>ECP</b> 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 <b>ECP</b> -endpoints, but it is possible to use MySQL or MSSQL for such a setup.  |
| Hawtio       | Hawtio                            | A monitoring software – access it on <b>ECP-endpoint</b> and <b>EDX-toolbox</b> on /hawtio on whichever port your webserver is running. You can browse you internal broker queues (see <b>Broker</b> ).   |
| MADES        | Market Data Exchange Standard     | A specification developed by <b>ENTSO-E</b> describing a communication system between actors in the energy sector.  |
| SC           | Service Catalogue                 | An <b>ECP</b> -component/server which keeps information about which endpoints consume/provides certain services. Without registration here, an <b>EDX-toolbox</b> cannot access services.   |
| TSO          | Transmission System Operator      | Responsible for the distribution of energy (electricity or natural gas), in an area/country.  |

### 3 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 big picture is shown below:



The drawing 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?

Even though a picture says more than a thousand words, a little explanation might still be in order:

- You must install ECP- and EDX-server, strongly preferred to be in the same OS-runtime.
- During the installation phase you will need to connect in the Component Directory.
- After the approval by Statnett, 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 Statnett with information about your endpoint and which services you will consume.
- When EDX-to-EDX traffic works you may test BA-to-BA traffic.

## 4 Considerations to make

The considerations often concern an "endpoint". This is a "logical endpoint", which is an ECP-endpoint and an EDX-toolbox combined.

### 4.1 Test environment is recommended

Statnett recommends setting up a test-endpoint which will be connected to Statnetts test-network.

**NB!** It is **not possible** to connect from test to production (or vice versa) in the ECP-networks set up by Statnett. Unless you are confident on going straight into production, you should install a test-endpoint.

### 4.2 Number of endpoints

Statnett did previously recommend as few endpoints as possible. However, experience over the past three years has led to a re-evaluating of this policy. It turns out that handling errors and malfunctions and upgrades is harder when you have traffic to/from many business-systems going through your endpoint. Therefore, Statnett would argue for a nuanced approach, where we no longer require "as few as possible", but rather "one endpoint pr major business system". Although this will lead into more management/administration to maintain, it will improve the robustness of the overall system in face of operational issues.

### 4.3 Failover (and uptime)

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

#### 4.3.1 Business acknowledgement (needs no action on ECP/EDX)

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".

#### 4.3.2 Failover server on standby (a modest investment to get 99.9% uptime)

Statnett 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 registry (see chapter 6.6). 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 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. The copy of the failover server must have the same updated certificates as the main server and these certificates are updated every 11<sup>th</sup> month. This has been explained in greater depth here: <https://ediel.org/wp-content/uploads/2021/05/NEX-Statnett-99.9-100K.pdf>

#### 4.3.3 High Availability (a big investment to get 99.99% uptime)

Read the documentation that follows the download-package of ECP/EDX (see next page).



## 5 Contract Signing, Requirements, Prerequisites and Firewall

### 5.1 Prerequisites

- Go to <https://ediel.org/nordic-ecp-edx-group-nex/market-actor-onboarding/> and find links to software and documentation
  - Download package for ECP v4.8.1
  - Download package for EDX v1.9.1
  - In these downloads you'll find official documentation, but the documentation is not tailored for Statnett's Market Actors and can be voluminous (if you ever thought this document was long!). You will need the following document:
    - ECP/EDX Installation Guide – will be referenced later in this doc
    - ECP/EDX Upgrade Guide – will be referenced later in this doc
    - ECP/EDX Administration Guide – a reference for advanced configuration, may not be necessary
    - EDX User Guide – a reference for how BA might connect to the EDX
- Go to <https://ediel.org/nordic-ecp-edx-group-nex/nex-statnett/> and find links to contract (terms of use/bruksvilkår), this guide + some other information.
- For each endpoint you install you must contact [ecp@statnett.no](mailto:ecp@statnett.no) with this information:
  - Company name
  - Which network you want to connect to (test or production)
  - Which services are you going to use<sup>1</sup>
- In return you'll get information you need later on:
  - Registration keystore (jks-file)
  - EIC-/ECP-/V-code (a 16-char string)

### 5.2 Contract Signing

To participate into the ECP-network of Statnett, one must sign an agreement (available on the location mentioned above). The agreement is clarification your responsibilities for using the network without any license fees.

### 5.3 Software requirements

- OS must be Windows Server 2012R2 or 2016, Linux Red Hat 7/8 or CentOS 7/8. Windows Server 2019 is not supported, although not impossible to get working<sup>2</sup>.
- We recommend running EDX/ECP on a standalone<sup>3</sup> server to avoid port-conflicts.

---

<sup>1</sup> Available services in production are "FASIT" and "MMS", while in test there are more options

<sup>2</sup> One issue might be that the port 8005 is taken by Windows. Change that port to "-1" the ECP's server.xml. Other strange things has also been reported, but the reports are generally not consistent throughout the 2019-installations.

<sup>3</sup> An OS dedicated to run ECP-endpoint and EDX-toolbox and no other servers/processes. It doesn't matter if it's a virtual or physical server.

## 5.4 Hardware requirements for an endpoint

|               | Statnett (100s msg/h) |                  | ENTSOE (1000s msg/h) |         |
|---------------|-----------------------|------------------|----------------------|---------|
|               | ECP                   | EDX              | ECP                  | EDX     |
| <b>CPU</b>    | 1C-2GHz               | 1C-2GHz          | 4C-2GHz              | 2C-2GHz |
| <b>Memory</b> | 4GB <sup>4</sup>      | 4GB <sup>5</sup> | 8GB                  | 4GB     |
| <b>Disk</b>   | 20 GB                 | 100 GB           | 40 GB                | 100 GB  |

Even more important than a specific setup is the ability to upgrade the hardware on a short notice. Statnett does not assume any responsibility should these recommendations be inadequate.

## 5.5 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): <b>eCP4prod.statnett.no</b>     | 443          | HTTPS    | Synch of ECP-network information |
|              | CD (test): <b>eCP4.statnett.no</b>         |              |          |                                  |
| ECP-endpoint | Broker (prod): <b>eCP4prod.statnett.no</b> | 5671         | AMQPS    | Message transport                |
|              | Broker (test): <b>eCP4.statnett.no</b>     |              |          |                                  |

The options you have on the port on EDX-toolbox depends on whether or not you install the EDX on the same OS as ECP-endpoint. If you want to setup an HA-database (external database) you must open the ports to that particular database (supported databases are MySQL and MSSQL).

Optional firewall configuration for those who will use Nordic Services (an example is eSETT which is hosted in the Fingrid-controlled part of the ECP-network). To make it easy we'll provide all the brokers that can be accessed within this network. The only thing you need to think of here is to make sure there is an open connection to the broker and that your firewall does not "tamper" with the SSL-certs:

| Country/TSO       | Broker  | Port/Protocol |
|-------------------|---|---------------|
| Finland/Fingrid   | (prod): <b>eCP.fingrid.fi</b><br>(test): <b>eCP-test.fingrid.fi</b>               | 5671/AMQPS    |
| Denmark/Energinet | (prod): <b>ieCP.prod.energinet.dk</b><br>(test): <b>ieCP.preprod.energinet.dk</b> | 5671/AMQPS    |
| Sweden/SvK        | (prod): <b>eCP4.svk.se</b><br>(test): <b>eCP4-test.svk.se</b>                     | 5671/AMQPS    |

<sup>4</sup> Default setup for Java memory requirement in ECP is 4GB, but one can most likely run with 2GB.

<sup>5</sup> Default setup for Java memory requirement in EDX is 1GB, but 2GB is preferable.

## 6 Installation or Upgrade of ECP-endpoint on Windows

### 6.1 Installation

If you do a first-time installation then execute ECP Installation Guide chapter 5.1 (Java installation), then move on to chapter 7.2 until 7.2.4 (should cover basic ECP Endpoint installation on Windows).

### 6.2 Upgrade

If this is an upgrade, execute the Upgrade Guide chapter 5.1 until 5.1.6 (should cover basic ECP Endpoint upgrade on Windows).

NB! Some tips to get a successful upgrade – perform this check AFTER java-installation/upgrade and BEFORE you install the ECP-package:

- Make absolutely sure that you have the proper JRE (not JDK!) installed
- JRE\_HOME is pointing to the correct location
- JAVA\_HOME is not defined (this variable shows up if you've installed JDK)
- Do not have any MMC-consoles open
- Do not have any Services-windows (services.msc) open – services might not be created correctly if you "occupy" that particular window
- Do not have any explorer-window or command-window opened within the file locations of ECP – files may not be deleted correctly if you are "in the way".

### 6.3 ECP-endpoint configuration

#### 6.3.1 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:

- `cd <install_path>\tomcat\bin`
- `tomcat9w.exe //ES//ecp-endpoint` (see footnote for syntax explanation <sup>6</sup>)

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

- Make sure "Use default" is on/true – important **if you upgraded Java/JRE and for future upgrades**
- 2048 MB is usually more than enough memory
- **The settings can have been subjected to change over the past versions of ECP, please report back to [ecp@statnett.no](mailto:ecp@statnett.no) if you have information**

#### 6.3.2 Configuration of ecp.properties (you should go over these settings every time you upgrade)

NB! Only the application and the root-administrator should have read-access to this file. The same goes for write-access.

---

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

| Property   | Description  |
|--|--|
| <pre>ece.endpoint.amqpApiEnabled = true ece.endpoint.sendHandler[0].beanName=amqpApiSendHandler ece.endpoint.sendHandler[0].typeName=* ece.networks = DefaultNetwork</pre>   | You <b>MUST ADD</b> these four properties, otherwise messages cannot be sent from ECP to EDX.  |
| <pre>spring.profiles.active= ecp-nonha</pre>   | With setting setting you require authenticated access to dashboard and webservices. For other options, see ECP Administration Guide.   |
| <pre>ece.directory.client.synchronization.directorySynchronizatio nInterval=0 * * * * * ece.directory.client.statistics.directorySynchronizationInte rval=30 * * * * * ece.directory.client.synchronization.messagePathSynchronizat ionInterval=45 * * * * *</pre> | You may have changed them according to Unicorn Guide, but this is a <b>better setup for our network. Use this one.</b>   |
| <pre>internalBroker.host=127.0.0.1 internalBroker.useAuthentication=false</pre>  | These settings defines how ECP will make itself available to EDX and assumes you install EDX on the same host as ECP. Otherwise use the settings '0.0.0.0' and 'true'.<br><br><b>NB!</b> Authentication became default 'true' in v4.7.x. |
| <pre>logging.file.name</pre>   | Changed from logging.file  |

### 6.3.3 Configuration of ecp-users.properties

To enable authentication, check the spring.profiles.active -parameter in the ecp.properties configuration above.

**NB!** Only the application and the root-administrator should have read-access to this file. The same goes for write-access. The file defines the user, role and passwords. Here is a simple example of the two types of users available:

```
ece.endpoint.users[0].login=admin
ece.endpoint.users[0].password=supersecret
ece.endpoint.users[0].role=admin

ece.endpoint.users[1].login=user
ece.endpoint.users[1].password=secret
ece.endpoint.users[1].role=user
```

To add more users, simply add new lines and increase the index.

## 6.4 Starting and stopping ECP-endpoint

### 6.4.1 GUI

- Start >services.msc

# Statnett

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

## 6.4.2 Command Line

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

## 6.5 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.

Check <install\_path>\logs\catalina.out (or .log) – this shows the startup sequence of Tomcat, and which ports are in use and which parameters were used to trigger the server. By inspecting this file you can learn a lot about the startup sequence and if parameters (typically paths) make sense.

Check to see that the dashboard is shown in the browser. NB! Starting from v4.7.2 of ECP and v1.8.2 of EDX, the default ports used is SSL-ports, thus port 8443 is the default port – 8080 no longer responds by default. If you want to open port 8080, please find <install\_path>/conf/server.xml and uncomment a Connector-tag for the 8080 port (there are usually commented examples of such Connectors in the server.xml). The standard URL to check will approximately like this:

[https://<YOUR-ECP-ENDPOINT>:8443/ECP\\_MODULE](https://<YOUR-ECP-ENDPOINT>:8443/ECP_MODULE)

In case login is required (due to settings in ecp.properties), the default login is admin/password. See previous chapter for configuration.

## 6.6 ECP-endpoint Registration – skip this one in Upgrade

### 6.6.1 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: [https://<YOUR-ECP-ENDPOINT>:8443/ECP\\_MODULE](https://<YOUR-ECP-ENDPOINT>:8443/ECP_MODULE)<sup>7</sup>, examples:
  - [https://localhost:8443/ECP\\_MODULE](https://localhost:8443/ECP_MODULE)
  - [https://127.0.0.1:8443/ECP\\_MODULE](https://127.0.0.1:8443/ECP_MODULE)
- Insert the registration key store provided by your component directory administrator (see 5.1) and wait for the file to upload. Password is "password". Click "Continue" to proceed to the next step
- Enter the appropriate URL and Code (see below), then click on the "check connectivity", then "Continue".
  - Test
    - URL: [https://ecp4.statnett.no/ECP\\_MODULE](https://ecp4.statnett.no/ECP_MODULE)
    - Code: 50V00000000111W
  - Production

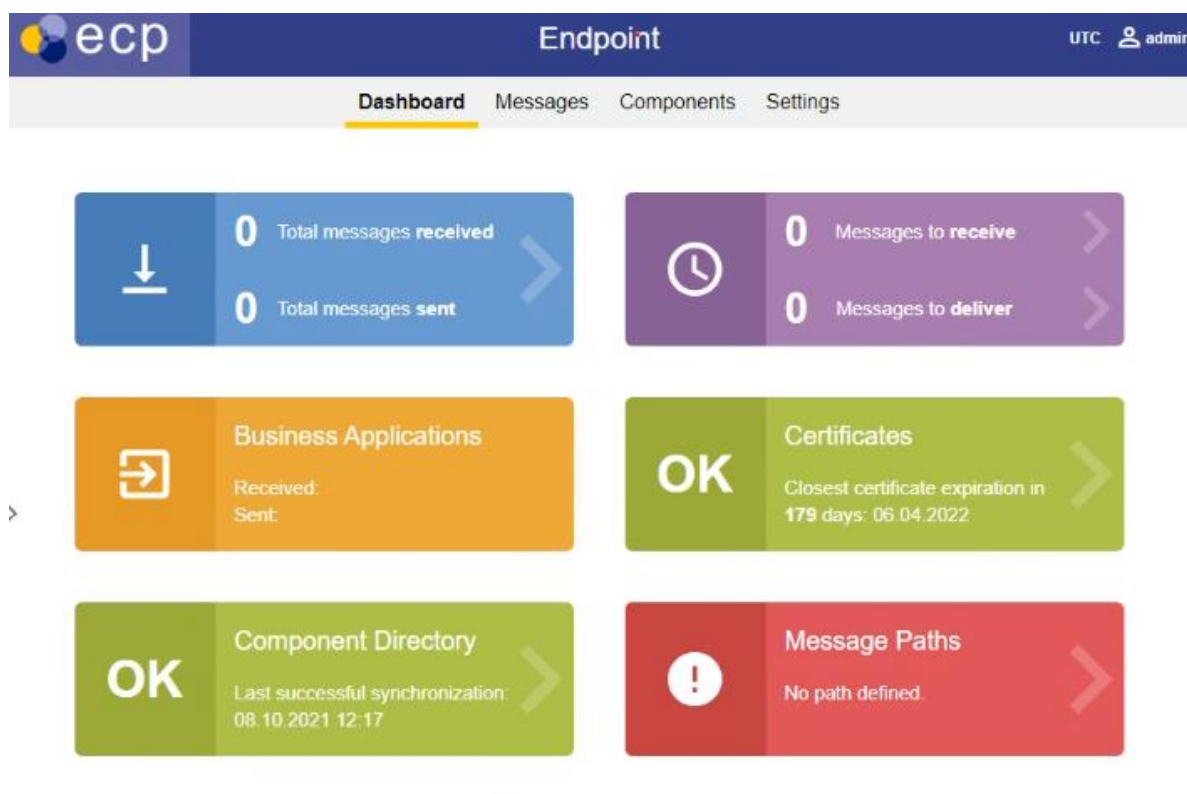
---

<sup>7</sup> If no response, see troubleshooting on 6.10.1. If presented with a login-screen, try admin/password and if that doesn't work check the contents of ecp-users.properties to find the admin-password.

- URL: [https://ecp4prod.statnett.no/ECP\\_MODULE](https://ecp4prod.statnett.no/ECP_MODULE)
- Code: 50V00000000118I
- Fill in your component code (EIC-code, see 5.1)
- For contact email in production, use a **monitored<sup>8</sup> email address**. Statnett will not approve if personal email-address, because Statnett will use this email-address to send information about upgrades and issues. In the test-network, personal email address is allowed.
- Phone number is not as important, it's a secondary option if email contact fails.

<sup>8</sup> Monitored means that some organization will be responsible for process the email at least daily.

At this point, you must wait until the component directory administrator approves your registration request. You must send an email to [ecp@statnett.no](mailto:ecp@statnett.no) to notify of the approval request (no automatic notification is built into ECP). There is a dashboard where you can monitor whether your endpoint has been approved or not – the "Component Directory" and "Certificates" tile should be green. The dashboard will look like this after a couple of minutes after the approval.



Once the request is approved your ECP-endpoint automatically becomes operational. It is now possible to send messages to other ECP-nodes in the network. However, you will not receive messages until you do the next step!

## 6.7 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 you will **not receive any messages** and the dashboard will show a red tile.

- Open a web browser and navigate to the URL of your ECP-endpoint (e.g. [https://<endpoint-ip>:8443/ECP\\_MODULE](https://<endpoint-ip>:8443/ECP_MODULE)), then choose menu "Settings"
- Choose button "+ New Path"
- Set Message Type to "\*", Path to "Indirect". In the drop-down, choose the broker which starts with 50V (Statnett-broker).
- Set "Valid from" back in time since the endpoint is running UTC-time as default (2 hours behind). There is no danger in setting yesterday as "valid from".
- Press "Save" – approximately 60 seconds later this configuration should be known to everyone in the ECP-network.

## New Path

Senders \* All Selected

Message Type \*

Path \* Direct Indirect

Valid from \*

|

44V000000000018F

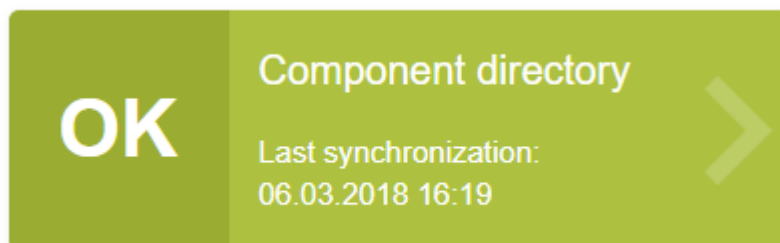
45V000000000058P

45V000000000062Y

50V000000000112U

### 6.8 Verify the installation

- Open a web browser and navigate to the URL of your endpoint (e.g. [https://<endpoint-ip>:8443/ECP\\_MODULE](https://<endpoint-ip>:8443/ECP_MODULE)).
- Check that the box indicating synchronization is green



- Make a "New Message" and send to a Statnett-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:



**Endpoint** admin

Dashboard **Messages** Components Settings

**New Message**

**Inbox**

**Outbox**

Receiver's endpoint code \* 50V000000001150

Message Type \* TEST

BA message ID

Sender application

HalloVerden.txt

Send Cancel

**Endpoint** admin

Dashboard **Messages** Components Settings

**New Message**

**Inbox**

**Outbox**

Status  Delivered  Received  Accepted  Failed

Sending Date

Receiver

Message type

Message ID

Search

| State    | Receiver        | Message Type | Sending Time     |
|----------|-----------------|--------------|------------------|
| Received | 50V000000001150 | TEST         | 28.01.2020 09:31 |

10 25 50 100

## 6.9 Change timezone of the endpoint (new)

Change the timezone of the endpoint on the Settings page, so the GUI will reflect correct timestamps. The file logs will continue to be in UTC-time.

### System Configuration

UTC

- UTC/UTC+1 Western European Time
- UTC+1/UTC+2 Central European Time
- UTC+2/UTC+3 Eastern European Time
- UTC+3 Further-Eastern European Time

Time zone

Select time zone

Save

### Registration

Initiate Registration

## 6.10 Troubleshooting

### 6.10.1 ECP-Endpoint does not respond on port 8080

- You've upgraded to ECP 4.7.2/EDX v1.8.2 or above, and default ports are now SSL-port 8443. 8080 is by default disabled.
- Another 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 restart the service.

### 6.10.2 Message status "Failed"

- Something is wrong with the configuration?

### 6.10.3 Message status "Accepted"

- Possibly, your firewall does not allow outgoing traffic to the Central Broker (see "Big Picture" in chapter 3)

### 6.10.4 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 3)

### 6.10.5 You're able to send, but do not receive any messages

- Your Message Path is not defined in "Settings" (see chapter 6.7)

## 7 Installation or Upgrade of EDX-toolbox on Windows

### 7.1 Installation

If you do a first-time installation then execute EDX Installation Guide chapter 5.1 (Java installation), then move on to chapter 6.2 until 6.4. You may skip 6.2.5 and 6.2.6 – should be covered below.

### 7.2 Upgrade

If this is an upgrade, execute the Upgrade Guide chapter 5.1 until 5.1.8 (should cover basic ECP Endpoint upgrade on Windows).

NB! Some tips to get a successful upgrade – perform this check AFTER java-installation/upgrade and BEFORE you install the ECP-package:

- Make absolutely sure that you have the proper JRE (not JDK!) installed
- JRE\_HOME is pointing to the correct location
- JAVA\_HOME is not defined (this variable shows up if you've installed JDK)
- Do not have any MMC-consoles open
- Do not have any Services-windows (services.msc) open – services might not be created correctly if you "occupy" that particular window
- Do not have any explorer-window or command-window opened within the file locations of ECP – files may not be deleted correctly if you are "in the way".

### 7.3 EDX-toolbox Configuration

Before starting EDX-toolbox, the application should be configured. The application configured with service properties (7.3.1) and three configuration files (next chapters). The table shows where to find the configuration files and the overall content of them.

| 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. |
| <install_path>\edx-users.properties | An optional configuration for User/pass for logins to EDX Web and WS  |

#### 7.3.1 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`
- `tomcat9w.exe //ES//edx-toolbox` (see footnote for syntax explanation <sup>9</sup>)

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

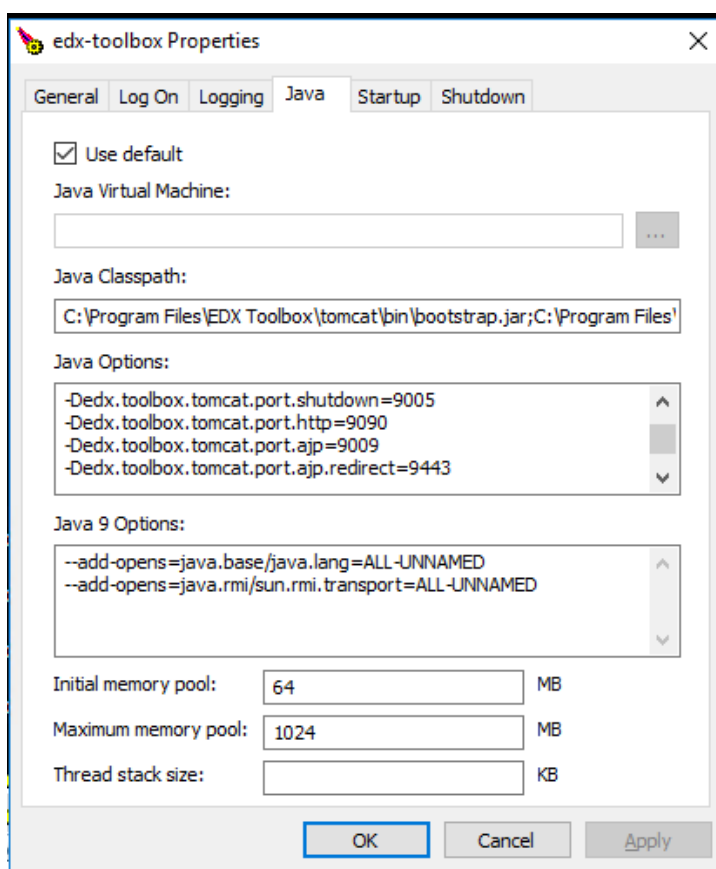
- Tick off "Use default" checkbox – important if you ever upgrade JRE on your system
- Although 1024 MB memory is probably ok, consider increasing it to 2048 MB if you have the capacity for it.
- If EDX runs in same OS as ECP, you need to change ports to avoid conflict. That should be covered in the Installation Guide (chapter 6.3.2). It is a different way of doing than what

<sup>9</sup> <https://tomcat.apache.org/tomcat-7.0-doc/windows-service-howto.html>

# Statnett

we've done in this guide before – were you could specify the various ports in the dialog shown below. Please make sure that there are no conflict/differences between the two approaches. At this point I (the author) am unsure whether these two approaches are fundamentally the same, that is just different views, or not.

- 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



Example of properties dialogue – not all Java Options are shown

- **The settings can have been subjected to change over the past versions of EDX, please report back to [ecp@statnett.no](mailto:ecp@statnett.no) if you have information**

## 7.3.2 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). Most of the configuration parameters use default values, but the following parameters must be configured for each installation:

| Parameter                         | Description  |
|-----------------------------------|--|
| edx.toolbox.code                  | ECP-endpoint code assigned to this Toolbox. This is the same EIC-code you received in chapter 5.1 and which you specified in the last step of chapter 6.6.1.   |
| edx.serviceCatalogue.code         | In the test-network: <b>50V000000000113S</b><br>In the prod-network: <b>50V000000000120V</b>   |
| ecpBroker.amqp.port               | Port number of your ECP AMQP Broker, most likely <b>5672</b> (see Big Picture in chapter 3)  |
| ecpBroker.amqp.host               | <b>127.0.0.1</b> if you install EDX in same OS as ECP. Otherwise, the <b>IP/hostname</b> of you ECP-endpoint.  |
| ecp.broker.url                    | <b>amqp://...</b> <sup>10</sup> - this was the default choice until v1.8 and still our preferred choice.<br><br>You're advised to use <b>amqps://...</b> if EDX is not running on the same host as ECP, in which case please check EDX Administration Guide for more information. This is the URL to which EDX connect to ECP (see footnote) |
| internalBroker.amqp.port          | <b>6672</b> is recommended if you install EDX in the same OS as ECP. Otherwise you may keep default setting <b>5672</b> . This is the port your BAs will connect (if they choose to use AMQP).   |
| internalBroker.amqp.host          | <b>0.0.0.0</b> – to indicate it accepts BA-requests from everywhere. 127.0.0.1 is the default configuration, which means that your BA must be running in the same OS (not recommended).  |
| internalBroker.useAuthentication  | <b>false</b> – this was the default choice until v1.8. You're advised to use <b>true</b> (=AMQPS) for higher security, in which case please check EDX Administration Guide for more information.   |
| edx.broker.url                    | <b>amqp://...</b> <sup>11</sup> - this was the default choice until v1.8.<br><br>You're advised to use <b>amqps://...</b> for improved security, in which case please check EDX Administration Guide for more information. This is the URL to which BA connect to EDX (see footnote)   |
| spring.profiles.active= edx-nonha | With the recommended setting you require authenticated access to dashboard and webservices. For other options, see EDX Administration Guide.   |
| logging.file.name                 | Changed from logging.file  |

### 7.3.3 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. The configuration is explained in chapter 7.6. NB! Make sure to have only **one** yml-file in the config-directory – since EDX will read all files with yml-suffix.

<sup>10</sup> The complete property is **amqp://\${ecpBroker.amqp.host}:\${ecpBroker.amqp.port}** and the content of the property is the end substituted by the ecpBroker.amqp.host/port-settings that we have specified.

<sup>11</sup> The complete property is **amqp://\${internalBroker.amqp.host:127.0.0.1}:\${internalBroker.amqp.port:5672}**, and the content of the property is in the end substituted by the internalBroker.amqp.host/port-settings that we have specified.

# Statnett

## 7.3.4 Configuration of edx-users.properties

To enable authentication, check the `spring.profiles.active`-parameter in the `edx.properties` configuration above.

NB! Only the application and the root-administrator should have read-access to this file. The same goes for write-access. The file defines the user, role and passwords. Here is a simple example of the two types of users available:

```
edx.toolbox.users[0].login=admin
edx.toolbox.users[0].password=supersecret
edx.toolbox.users[0].role=serviceManager

edx.toolbox.users[1].login=user
edx.toolbox.users[1].password=secret
edx.toolbox.users[1].role=user
```

To add more users, simply add new lines and increase the index.

## 7.4 Starting and stopping EDX-toolbox

### 7.4.1 GUI

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

### 7.4.2 Command Line

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

## 7.5 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 7.3.1). The status of the service, if installed, can be checked either via command line: "**SC query edx-toolbox**" or in the Windows Services tool.

### 7.5.1 Send test messages

- Open a web browser and navigate to the URL of your endpoint (e.g. <http://<endpoint-ip>:9443/> - if you specified 9443 as your port). You should see this, except you won't have any messages in the list:

The screenshot shows the 'Messages' monitoring page in the EDX interface. It features a search form with fields for EDX Message ID, Status, Message Type, Direction, Sender, Receiver, Service, and Date Range. Below the search form is a table with the following data:

| Status   | Message Type | Direction | Sender           | Receiver         | Time             |
|----------|--------------|-----------|------------------|------------------|------------------|
| Received | TEST         | Receive   | 50V000000000115O | 50V000000000115O | 28.11.2018 15:16 |

- Make a "New Message" and send to a Statnett SF-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 interface. The form includes the following fields and content:

- Receiver: 50V000000000115O
- Message Type: TEST
- BA Message ID: (empty)
- BA Correlation ID: (empty)
- Sender Application: (empty)
- File content: HalloVerden.txt
- Buttons: Send, Cancel

- 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 Statnett. Please contact [ecp@statnett.no](mailto:ecp@statnett.no) to investigate further.

## 7.6 Configuring the edx.yml file

NB! Make sure to have only **one** yml-file in the config-directory – since EDX will read all yml-files.

This is an example of a relatively complete edx.yml, carefully crafted to show a number of features. It should hopefully provide enough examples to help you configure your own edx.yml. The file will be

# Statnett

explained in detail below. Make sure not to introduce any tabs in this file, only spaces are allowed. Also, be very careful about the number of spaces used for indentation – otherwise it will not be parsed correctly. If you accidentally miss a comma, EDX might not warn you about it. Some line breaks are introduced in the example below for readability of very long lines; remove them! Make sure to read the `edx.log` and `catalina.log` carefully after startup of EDX, it should show if the file was parsed as expected.

```
integrationChannels:
  amqpEndpoints:
    - {direction: in, code: amqp-ba1-outbox, queueName: ba1.outbox, redeliveryAttempts: 1, replyQueueName: ba1.reply}
    - {direction: out, code: amqp-ba1-inbox, queueName: ba1.inbox, redeliveryAttempts: 1}
  fssfEndpoints:
    - {direction: in, code: fssf-ba2-outbox, directory: /ba2/outbox, redeliveryAttempts: 1, replyDirectory: /ba2/reply}
    - {direction: out, code: fssf-ba2-inbox, directory: /ba2/inbox, redeliveryAttempts: 1}
    - {direction: out, code: edx-errors, directory: /edx-errors, redeliveryAttempts: 1}
  ftpEndpoints:
    - {direction: in, code: sftp-ba3-outbox, directory: ba3/outbox, redeliveryAttempts: 1, replyDirectory: ba3/reply, protocol: sftp, hostname: sftp.host.org,
      port: 22, username: user, password: pass, connectionParams: {stepwise: true, separator: UNIX, knownHostsFile: /home/edx-toolbox/.ssh/known_hosts }}
    - {direction: out, code: sftp-ba3-inbox, directory: ba3/inbox, redeliveryAttempts: 1, tempPrefix: ../tmp/, protocol: sftp, hostname: sftp.host.org,
      port: 22, username: user, password: pass, connectionParams: {stepwise: true, separator: UNIX, knownHostsFile: /home/edx-toolbox/.ssh/known_hosts }}
  kafkaEndpoints:
    - {direction: out, code: kafka-publish, topicName: publish, redeliveryAttempts: 1, connectionURI: "k1.host.org:9092,k2.host.org:9092",
      partitionKeyMadesHeaders: [businessType, sender], options: {"compressionCodec=gzip&maxRequestSize=30000000"}}
    - {direction: in, code: kafka-inbox, topicName: MMS, domainCode: DEFAULT_DOMAIN, serviceCatalogueCode: 50V000000001135, start: toolbox-gateway,
      replyTopicName: inbox_reply, connectionURI: " k1.host.org:9092,k2.host.org:9092", options:
      "groupId=edxGroup&sslTruststoreLocation=/var/lib/ecp-endpoint/authKeystore.jks&sslTruststorePassword=password&sslKeystoreLocation=/var/lib/ecp-
      endpoint/authKeystore.jks&sslKeystorePassword=password&sslKeystoreType=JKS&sslTruststoreType=JKS&securityProtocol=SSL"}

components:
  validations: []
  transformations: []
  externalProcessing: []
routing:
  routes:
    - {code: R-amqp-ba1, service: {serviceCode: FASIT, domainCode: DEFAULT_DOMAIN, serviceCatalogueCode: 50V000000001135 }, start: toolbox-gateway,
      end: amqp-ba1-inbox, fail: edx-errors, steps: [] }
    - {code: R-fssf-ba2, service: {serviceCode: MMS, domainCode: DEFAULT_DOMAIN, serviceCatalogueCode: 50V0000000001135 }, start: toolbox-gateway,
      end: fssf-ba2-inbox, fail: edx-errors, steps: [] }
    - {code: R-ba1-ba2, service: {serviceCode: MNA, domainCode: DEFAULT_DOMAIN, serviceCatalogueCode: 45V000000000059N }, start: toolbox-gateway,
      end: [amqp-ba1-inbox, fssf-ba2-inbox], fail: [amqp-ba1-fail, fssf-ba2-fail], steps: [] }
    - {code: R-sftp-ba3, messageType: EXT-EI-MAGASINDATA, start: toolbox-gateway,
      end: sftp-ba3-inbox, fail: edx-errors, steps: [] }
    - {code: R-kafka-ba4, service: {serviceCode: NUCS, domainCode: DEFAULT_DOMAIN, serviceCatalogueCode: 50V0000000001135 }, start: toolbox-gateway,
      end: kafka-publish, fail: edx-errors, steps: [] }
  sendProcessDefaultRoute: {start: "*", end: toolbox-gateway, fail: ecp-endpoint, steps: [] }
  receiveProcessDefaultRoute: {start: toolbox-gateway, end: ecp-endpoint, fail: edx-errors, steps: [] }
```

There are three sections in this file: **integrationChannels**, **components** and **routing**:

## 7.6.1 Components

We are not interested in **components** – this section has no configuration, `[]` simply means an empty array. The lack of interest in **components** configuration is deliberate: We don't want to introduce validations and transformations in the EDX, even though it works quite nice. The point is that from the moment EDX takes on the responsibility of validating and transforming the messages, it becomes more than a simple messenger – it becomes part of the business logic and fault handling. It is a Statnett recommendation to deliver the message unaltered from one BA to another. This will ensure less trouble in the transport-layer and more flexibility for the BA. The cost is that each Business Application must handle validation/transformation for themselves.

## 7.6.2 IntegrationChannels

**IntegrationChannels** define a set of "endpoints" which specifies where BA and EDX can place or pick up a message. These "endpoints" are not the same kind explained in chapter 2, so please do not confuse them. There are five types of endpoints:

- AMQP (Advanced Message Queue Protocol)
- FSSF (File System Shared Folders)
- FTP (File Transfer Protocol)
- Kafka (Statnett use this for internal publish/subscribe)
- WS (Web Service) – the default endpoint, not specifically configured

Each endpoint type is placed in its own section. The endpoint must specify a **direction** and a **code**. The **direction** can be



# Statnett

- "in": Location where BA place a message and EDX picks it up and delivers it to the receiver
- "out": Location where EDX place a message coming from a sender and BA then picks it up

The **code** must be a unique identifier of this endpoint. The codes in "out"-endpoints will be used in routes, because routes define where to place messages when they're received from the network (other endpoints).

Further important notes:

- You may create as many endpoints as you wish
- If you don't want any endpoints, simply type "[]" after the colon. (ex: sftpEndpoints: [])
- We're using the naming convention inbox/outbox as seen from the BA's point of view, which may cause some confusion with the **direction** (seen from EDX' point of view)
- Default redelivery-attempts is 10 in EDX, but we override this in our example to 1. The reason is that redelivery seldom solves any issue, it just creates a lot of "noise" in the logs.
- You should specify one in-endpoint for each BA. The reason is that you will have access to a reply-endpoint for each BA. The BA should/could monitor the reply-endpoint to receive messages with information about the transmission of your original message. It can tell you if the message has safely arrived to the receiver's EDX. It can also tell you if you've used a non-existent EDX address (receiverCode). However, the only way to know if the receiving BA has picked up the message from the receiving EDX is to listen for a regular message from the other BA with "business acknowledgement" (see chapter 4.3.1).
- When EDX receives a message from the network, it may fail to deliver it to the correct out-endpoint. The reason could be that you have specified validation or that the message is too big (can happen with Kafka) or something else. In those cases, the failed message will be placed on a shared folder defined at the very last line in the config: fail: edx-errors (which in turn points to the fssfEndpoint with code "edx-errors"). It will fall to the manager of the EDX to resolve such issues.

## 7.6.2.1 AMQP-endpoint

The configuration suggested in the example shows 2 queues for a BA named "ba1". One queue is for messages from ba1 to other recipients (ba1.outbox) with the corresponding reply-queue (ba1.reply) mentioned above. Another for messages to ba1 from other BAs in the network (ba1.inbox).

Queues are automatically created by EDX.

## 7.6.2.2 FSSF-endpoint

The configuration suggested in the example show 2 folders for a BA named "ba2", following the same pattern as for AMQP. A reply folder is also specified in the same manner as for AMQP.

You must create this folder yourself and assign read/write/execute-privileges to EDX-Toolbox process for these folders.

## 7.6.2.3 FTP-endpoint

The configuration suggested in the example shows the same setup as for AMQP and FSSF, now for BA "ba3". What happens here is that EDX has an FTP/SFTP-client which connects to an FTP/SFTP-server. The connectionParams are optional, but useful. The parameters sent directly the underlying apache-component and are documented here:

<https://camel.apache.org/components/latest/file-component.html>

<https://camel.apache.org/components/latest/ftp-component.html>

# Statnett

By setting the tempPrefix-attribute to "../tmp" you ensure that the file is not moved into the correct folder until it's completely written. The "../tmp"-folder must be created (as ba3/tmp) and given proper privileges.

## 7.6.2.4 Kafka-endpoint

EDX can both consume and produce from a Kafka topic, both with and without SSL. We've provided both examples in the configuration. The example shows how we've used the ECP-certificates to authenticate to the Kafka-broker – with that "trick" we don't have to renew SSL-certs yearly.

Again, as for the FTP-endpoints, there are optional attributes which can be specified. These attributes are specified here:

<https://camel.apache.org/components/latest/kafka-component.html>

Statnett recommends "compressionCodec" and "maxRequestSize", because Kafka is usually not suited for big messages (greater than 1MB?). However, with compression, many text-messages can be compressed up till 90%. Specify the maxRequestSize so that all messages are attempted to be sent to Kafka and not rejected before EDX has tried to compress it.

## 7.6.3 Routes

The routes listed in the example show how each BA listens to a particular type of message, determined by the filter (service or messageType) and the end-attribute specification.

- Ba1 listens to messages with EDX-service code = FASIT
- Ba2 listens to messages with EDX-service code = MMS
- Ba1 and Ba2 listens to messages with EDX-service code = MNA. MNA is provided by another EDX ServiceCatalogue than for FASIT, MMS and NUCS.
- The ServiceCatalogue code provided in the example is for the test environment. For the production environment you should use SC code found in chapter 8.2.2.
- Ba3 listens to messages with MessageType = EXT-EI-MAGASINDATA
- Ba4 listens to messages with EDX-service code = NUCS

A few notes about this:

- We advise you to keep the routing as simple as possible.
- We advise you to avoid using MessageType in the routing. This is because the MessageType is specified by the BA and it's better for the routing to be independent of changes in the BA.
- EDX-Service is used here as something like a "system-to-system" channel. This is defined solely within EDX and makes it well suited to perform routing (the BA may change, but the routing stays the same).
- You may have multiple filters, both Service, MessageType and even Sender. EDX will use the routing rule which matches the message and is most specific. Statnett advise against such rules, it will be hard to maintain.

At the end of the routing section you'll find the two default send/receive-routes. If the routes above do not match anything, the message will go to the default endpoint, which is a WS-endpoint. The WS-endpoint does not have the ability to retrieve messages based on Service, only based on MessageType. Therefore, we advise against using it, since it will be more problematic if two BAs will use WS-endpoint; they would have to retrieve based on MessageType.

# Statnett

## 7.7 Change timezone of the endpoint (new)

Change the timezone of the endpoint on the Settings page, so the GUI will reflect correct timestamps. The file logs will continue to be in UTC-time.

The screenshot shows the 'Settings' page in the Statnett interface. It includes fields for 'Toolbox Code', 'Toolbox Name', 'Party Name', and 'Publish/Subscribe Version'. Below these is a table of service catalogues with columns for 'Service Catalogue', 'Network Configuration Version', and 'Last Synchronization time'. At the bottom, there is a 'Time Zone' dropdown menu, which is circled in red in the original image, and a 'Save' button.

| Service Catalogue | Network Configuration Version | Last Synchronization time | Download | Request from SC |
|-------------------|-------------------------------|---------------------------|----------|-----------------|
| 50V000000000120V  | 265                           | 12.06.2021 13:26:52       | Download | Request from SC |
| 45V000000000055V  | 40                            | 18.06.2021 13:35:03       | Download | Request from SC |

Time Zone: Select time zone [v] Save

## 7.8 Troubleshooting

### 7.8.1 You have not received Service Catalogue (AKA network configuration)

Check settings on your EDX Dashboard (see screenshot above). You should see that the network configuration from Statnett (50V000000000120V in production) 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 6.10.5. Another reason may be that Statnett has not updated the Service Catalogue with your endpoint (this is a manual process in Statnett). Please notify Statnett at [ecp@statnett.no](mailto:ecp@statnett.no) if you suspect this to be the case.

### 7.8.2 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 7.3.2), 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 Statnett). Please notify Statnett at [ecp@statnett.no](mailto:ecp@statnett.no) 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".

## 8 Installation of ECP/EDX on Linux (Red Hat 7 or CentOS 7)

The installation is not covered here, but a few tips are in order.

- The installation is made for the Red Hat 7 and CentOS 7 distribution. RPM-packages available in the download-packages (see chapter 5.1)
- Some of the document are specially tailored for Windows, but most would apply to all OS
- You can find much more documentation in the download-packages.
- Contact [ecp@statnett.no](mailto:ecp@statnett.no) if you have more questions. If many requires Linux installation, we can expand this installation guide.

# Statnett

## 9 Appendix

### 9.1 Uninstallation of ECP-endpoint

Stop the application and execute the following command:

- `java -jar <install_path>\Uninstaller\uninstaller.jar -c`

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

### 9.2 Changing network or re-register

If you want to change network or simply re-register, you can go to the Settings-page of the ECP and find the Re-register button. This will work for ECP v4.7+. For earlier version you'll have to delete the db-folder of ECP and restart. In both cases you will enter the registration dialog and must perform the steps below from chapter 6.6 and onwards.

For EDX the only change will be to change settings in `edx.properties` (see chapter 7.3.2)

### 9.3 Monitoring EDX-toolbox and ECP-endpoint

#### 9.3.1 Manual monitoring with Hawtio

In `edx.properties` and `ecp.properties` specify the following property (or change the property if it exists):

- `spring.jmx.enabled=true`

You may use Hawtio to monitor queues – it can be very useful. To install it, download Hawtio from [here](#)

<https://repo1.maven.org/maven2/io/hawt/hawtio-web/1.5.11/hawtio-web-1.5.11.war>

and copy it in the `webapps`-folder of the ECP/EDX-application with the filename **hawtio.war**. It should auto-deploy and you can access it on the following URL (admin/password):

- <http://<YOUR-ECP-ENDPOINT>:8080/hawtio/>
- <http://<YOUR-EDX-TOOLBOX>:9090/hawtio/>

You can, among other things, browse queues and see if messages are picked up, and even purge queues if something is stuck. Very useful.

#### 9.3.2 Automatic monitoring

##### 9.3.2.1 Message monitoring

What you'll do is to ask you own endpoint to send a "test-message" to a Statnett-endpoint. This is because you in all likelihood will communicate with Statnett on a Statnett-controlled network, but connectivity to other endpoints might also be necessary or useful to monitor. The ECP-codes used for Statnett is 50V000000000115O or 50V-SN-DK----ATT for Test-environment, and 50V000000000188Y or 50V000000000121T for Production-environment.

The simplest approach is to run an HTTP-request like this:

PUT [http://<Your-ECP-endpoint>:<port>/ECP\\_MODULE/settings/connectivityCheck](http://<Your-ECP-endpoint>:<port>/ECP_MODULE/settings/connectivityCheck)

Set HTTP header for "Content-Type" to "application/json"

Set HTTP body to {"receiver": "<ECP-code>", "messageType": "TEST"}

# Statnett

The more advanced approach is when your endpoint require login: Then you need a token from login to perform the connectivity-check. We've made a python3-script, which is now updated to support ECP 4.8.1, using port 8443 in this example. Some long lines are wrapped (shown with 2 space indentation):

```
import requests
import json
import time
import urllib3
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
requests.packages.urllib3.disable_warnings()

host = '{{ ansible_hostname }}'
endpoint_code_to_check = '{{ connectivity_check_code }}'
username = 'admin'
password = '{{ lookup('env','ECP_ADMIN_LOGIN') }}'

session = requests.Session();
response = session.get('https://%s:8443/ECP_MODULE/' % host, timeout=30, verify=False).content
token = (session.cookies.get_dict())
xxtoken = token['XSRF-TOKEN']
headers = {'X-XSRF-TOKEN': xxtoken, 'Content-Type': 'application/json;charset=UTF-8'}
response_conncheck = session.put('https://%s:8443/ECP_MODULE/settings/connectivityCheck' % host,
auth=(username, password), verify=False, data={"receiver": "%s", "messageType": "TEST"}' %
endpoint_code_to_check, headers=headers).text

response_status_json = json.loads(response_conncheck)
```

The response JSON can be parsed and dumped to file like this:

```
status_check = response_status_json['status']
output_json = {}
statustime = time.strftime("%Y-%m-%d %H:%M:%S")
timestamp = int(time.time())
output_json['statustime'] = statustime
output_json['timestamp'] = timestamp
output_json['connectivity_check_status'] = status_check
output_json['hostname'] = '%s' % host
output_json['endpoint_code_checked'] = '%s' % endpoint_code_to_check

print (json.dumps(output_json))
```

And it will print something like this to your output\_json file:

```
{"timestamp": 1599561602, "statustime": "2020-09-08 12:40:02", "connectivity_check_status": "OK",
"hostname": "your-own-ecp-endpoint-host", "endpoint_code_checked": "50V00000000121T"}
```

The connectivity-check will give up after 5 seconds, which is can be a bit too soon. We recommend to change this to 30 seconds, using these parameters settings in the ecp.properties file:

```
ecp.endpoint.connectivityCheckAttemptTimeout=3000
ecp.endpoint.connectivityCheckAttemptsCount=10
```

When the monitoring alerts you of a problem, most likely the problem is your own ECP-endpoint. Do not contact Statnett to alert about issues, unless you are very sure that your own endpoint is fine and that "some time" has passed.

# Statnett

## 9.3.2.2 Queue monitoring

Sometimes it can be valuable to monitor the queues in EDX. For that to work you need to install Hawtio (see 9.3.1). The you can run a simple curl-command, the queue-name is marked in red font.

```
curl 'https://<EDX-toolbox>:<port>/hawtio/jolokia/' -H 'Content-Type: text/json' -
-data-binary
' [{"type": "read", "mbean": "org.apache.activemq:type=Broker,brokerName=localhost,des
tinationType=Queue,destinationName=edx.endpoint.inbox", "config": {}} ]' --compressed
--insecure
```

If you need to login to run the command, you'll need to adapt the script in the previous chapter.

## 9.3.2.3 Log monitoring

You can of course monitor the logs, namely the ecp.log (or sometimes ecp-endpoint.log) and edx.log (or sometimes edx-toolbox.log). One particular statement is a trigger for "serious error" and that is the phrase "Failed delivery". Such messages indicate that the ECP or the EDX is not able to deliver a message in some direction.

Her is an example of such an error-message:

```
2020-09-22 15:12:32.020 ERROR 81894 --- [Camel (camel-1) thread #18 -
JmsConsumer[ecp.endpoint.upload.10V1001C--000446]]
o.a.camel.processor.DefaultErrorHandler : Failed delivery for (MessageId:
ID:5c954a19-69a5-4c72-b2f7-8fd174dc84cd:1:2:1-45099 on ExchangeId: ID-x1-a-ecp-
app01-statnett-no-1596701161654-0-5632588). On delivery attempt: 5 caught:
org.springframework.jms.UncategorizedJmsException: Uncategorized exception
occurred during JMS processing; nested exception is javax.jms.JMSException:
java.nio.channels.ClosedChannelException
org.springframework.jms.UncategorizedJmsException: Uncategorized exception
occurred during JMS processing; nested exception is javax.jms.JMSException:
java.nio.channels.ClosedChannelException
```

## 9.4 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 "\*" (chapter 6.7), otherwise you cannot receive 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 see chapter 5.5 for hostname and port number and try to telnet directly from the endpoint and from another location to determine whether the problem is on Statnett's end or your own.
- Component Directory: Connectivity check to see if the CD (over port 443) is available. Data is exchanged here every minute. If CD is offline for a long time (usually many hours or days) you will not be allowed to send messages any more.

The ECP-server offers a Dashboard which shows

# Statnett

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

## 9.5 [Create an endpoint with 99.9% uptime & 100K messages per day](#)

Read the 999-document publish in same location as this document ([ediel.org](http://ediel.org))