Statnett

Visiting adress
Nydalen allé 33, 0484 Oslo

Postal adress
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
E firmapost@statnett.no

99.9% Uptime – 100K Messages per day

How to build and maintain such an endpoint

Version: 1.01

Date: 10.06.2022

Statnett

Innhold

1 DOCUMENT HISTORY 3

2 INTRODUCTION 4

3 BUILD 5

3.1 MANUAL FAILOVER SETUP 5
3.2 WHY WE DON'T RECOMMEND AUTOMATIC FAILOVER 5
3.3 WHY WE DON'T RECOMMEND HIGH AVAILABILITY (HA) SETUP 6

4 MAINTAIN 7

4.1 PROBLEMS AND HOW TO FIX THEM 7
4.2 HOW TO CONNECT TO A DERBY-DATABASE: 10
4.3 HOW TO INSTALL HAWTIO 10
4.4 HOW TO INSTALL/USE QUEUECLEANER 10

Statnett

1 Document History

Version	Date	Changes
1.00	16/04-2021	- The first draft
1.01	10/06-2022	- Major update

Statnett

2 Introduction

An ECP/EDX-endpoint will for some require a 99.9% uptime in order to comply with the needs of the business. In this document we outline the Statnett strategy to achieve that and supply with resources on how to handle problems that eventually will occur. The document will also help the endpoint to handle a higher load (100K messages per day).

Statnett

3 Build

3.1 Manual failover setup

We recommend building a manual failover to achieve the goal. Later in the document you'll find reasoning on why we do not choose automatic failover or high-availability (HA) setup. Here is what you need to do to setup a manual failover:

1) Complete default installation of ECP/EDX-endpoint. The most robust setup will be to use the internal database, but it's possible to achieve manual failover with external database.
2) After endpoint is running, install another endpoint on another host, but do not register the endpoint.
3) Setup a job to copy data-files from the original endpoint to the failover endpoint. Locate these 3 folders for both ECP and EDX, and copy them:
    a. Database (usually named db). This only applies if you use the "internal database". If you use external database, your failover endpoint must connect to the same external database.
    b. Broker (usually named internalbroker or edx-activemq-data)
    c. Configuration (usually named conf, ecp-endpoint or edx-toolbox)
4) Other files (like JKS-files) will be created when starting the endpoint.
5) Make sure your Business Applications can retrieve send messages from both endpoints, whichever is active.[1]
6) Only run one endpoint at the time, otherwise one endpoint might "steal" acknowledgements/messages from the other or in the worst case the endpoint will stop retrieving messages from the central broker.
7) Test the arrangement

There is a downside to this arrangement apart from the obvious one; that it is manually controlled: When you stop the failover-endpoint you might be unlucky and capture a message in-flight. Such a message will never be delivered unless you start the failover-endpoint again. For this reason, you would not want to use the failover option very often, but it could be useful whenever you have a major upgrade, major mal configuration of network or hardware failure.

The downside of using external database is that you're issue with the ECP-endpoint could be related to the database engine, and then failover wont help.

3.2 Why we don't recommend automatic failover

1) We don't believe we have a fool proof way of knowing that only one endpoint will run at the time. We cannot easily detect if an endpoint is down, slow, halfway-connected (can send, but not receive – or vice versa) or has other connection problems. Thus, an automatic failover could cause two endpoints to run simultaneously – which cause problem the problems mentioned in previous chapter.
2) Automatic triggering of the failover could result in many such failovers because of some minor and frequent instability. If the instability is short lived (a few minutes), it could be better to simply wait it out. This depends upon the requirement of the traffic; the underlying

1 If your BA uses AMQP to connect to EDX, then you could perhaps use a AMQP-client-library like QPID and it supports failover-urls like this: failover:(amqp://host1:5672,amqp://host2:5672) – see Redhat doc.

assumption is that this network is not used for anything resembling real-time traffic but operates within minutes-boundaries.

Update Jun 2022: ECP (v4.8 and up) has become more stable in the past year, and the knowledge about how to detect halfway down/up has increased. It is a clear goal to for ECP development in 2022 to provide metrics that clearly state if the endpoint is fully functional. Thus, automatic failover could be the recommended solution in 2023.

3.3 Why we don't recommend High Availability (HA) setup

ECP and EDX offers HA setup, so why do we not recommend that setup? In short, the answer is that we believe it's too costly for most and we're not convinced it will increase uptime. To be blunt, we think that the complexity of HA-setup will cause some downtime in itself and it will negate the gain you get from more hardware resources. With a high investment of time/resources/knowledge into such a setup, tuned over some time, it is probable that it would be a better solution.

Unicorn (the developer of ECP) has said that the name "High Availability" is focused on performance rather than uptime. A more proper name could then be "High Performance".

Our reasoning goes as follows:

1) The benefit of HA is usually to cover breakdown in hardware. Network problems could sometimes be alleviated, but application problems cannot be expected to be helped by HA, quite the opposite (expect more complex database-setup).
2) Modern day hosting has built-in HA, so that failure in hardware should not affect the application in most cases. We expect less than 1 such incident per year. Network problems due to mal configuration (again, network hardware is expected to be redundant) is more likely, but a major incident where HA would help is expected to be rare.
3) Thus, the expected number of incidents where HA would be helpful is set to one per year as a maximum estimate. Such an incident can be handled with manual failover within 3 hours in a worst-case scenario, but in many cases much quicker. Measured across a year, this is well within 99.9% uptime.
4) Introduction of HA also introduces a more complex setup of external database (MySQL, Oracle, MSSQL) and more importantly, a less well-tested setup. It is very reasonable, and experience shows, that such setup by itself introduces some amount of downtime. Only by investing enough time/resources/knowledge into such a setup will you be able to counter the added complexity. ECP/EDX has a small user base with a wide variety of setup options – making it less likely that it will be stable in all cases.

Statnett

4 Maintain

4.1 Problems and how to fix them

Problem	Options – each option should be a complete sequence to try to solve the issue. The options are ordered from harmless to nuclear.
Messages are not being sent	This is a basic problem and will in many cases not have anything to do with ECP or EDX. The following will provide a method to pinpoint the source of the problem. 1. Go to ECP-GUI-Settings and click on "Check Connectivity" under "Component Directory". If status is OK, then you know that your endpoint is connected and synchronized with the CD (it synchronizes every minute). That means that your endpoint will know all the public certificates and the message paths of the other endpoints in the network. If status is not OK, then you should check the file ecp.log to see error messages related to traffic towards the CD. Even if you do not have connection to the CD, the message traffic may still flow well, because you have a local copy of the CD. 2. Go to ECP-GUI-Setting and click on "Check Connectivity" under "Messaging Connectivity". Choose the endpoint you want to send to and specify TEST as Message Type. Then send. If you get "OK", then you know that ECP to ECP traffic is working well. If you get "NOT CONNECTED" you can check the file ecp.log to search for reasons, but you can also try to send to other ECP-endpoints. This type of testing will not show up in ECP GUI of the receiving endpoint, so it is totally fine to use for testing. If some endpoints respond, then you know that your ECP-endpoint works fine, but some other endpoints fail. You can also check in ECP-GUI-Components -> Choose details on the receiving component. Here you can see if that component has defined Message Paths. The message path tells which broker your endpoint must connect to, to get through. Then go to ECP-GUI-Dashboard -> Click on blue tile. Check status on the various brokers you're connected to and make sure you can connect to the broker mentioned in the Message Path. 3. If the connectivity still fails, it could be related to certificates being expired or firewall problems. See advice further down in the row "ECP Cannot connect to CD/Broker due to certificate problems" 4. If ECP Connectivity is fine, then we move to EDX Connectivity. First, go to EDX-GUI-Settings and check if you have defined a ServiceCatalogue. You may have more than one, but there must be at least one. For Statnett-connected endpoints, the ServiceCatalogue must start with the code 50V. If you're missing such a catalogue, you must check edx.properties and see that you've properly defined the Service Catalogue code. If that is in place, try restarting the EDX-toolbox, because then you will see in ECP-Outbox that a request for a new EDX-NetworkConfiguration (=SC) has been sent to SC-endpoint. If your toolbox has been defined in the SC, then you'll see the respond immediately in ECP-Inbox. If not, contact ecp@statnett.no and explain the situation. **(NB! The problem might also be that you in general cannot receive any message – see next row in this table.**) If you do receive a response in ECP-Inbox, you can further track the parsing of this response in the file edx.log. If something goes wrong with the parsing of the SC, you can see it there. If everything goes

Statnett

	well, you will now have a SC in EDX-GUI-Settings. You can then create a new message in EDX-GUI-NewMessage and send it to some endpoint of your choosing. You can also send a message to your own endpoint, but that will not test the ECP-endpoint, only the EDX-toolbox. When you test, you should specify the BusinessType = TEST. 5. When you send (from both ECP and EDX) you should expect that the end-status of the message transmission is RECEIVED. You can see this in the GUI. The GUI also provides information of the various status. If you see "SUCESSFULLY SENT" in EDX, it means that your ECP-endpoint received the messages, but so far the receiving ECP-endpoint has not responded with an ACK. With the use of status you can investigate how far the message has been transmitted. However, if you see the status ACCEPTED in ECP-GUI, you do not know with full certainty whether the messages has been transmitted to the receiving ECP-endpoint or not: One reason for this status *could* be that the receiver ECP has not been able to send a DELIVERED-ACK back to your endpoint. Another reason *could* be that your local ECP-endpoint has not been able to connect to the central ECP broker.
Messages are not being received	This is a basic problem and will in many cases not have anything to do with ECP or EDX. The following will provide a method to pinpoint the source of the problem. 1. Same as point 1 in "Messages are not being sent" 2. Same as point 2 in "Messages are not being sent" 3. If the connectivity still fails, it could be related to certificates being expired or firewall problems. See advice further down in the row "ECP Cannot connect to CD/Broker due to certificate problems" 4. Check that you've defined a Message Path in ECP-GUI-Settings. This message path is clearly explained in the NEM Installation Document. 5. Check that you have two TCP-connections towards the ECP Broker. You can check this with this command "netstat -a -n -p" and look for connections towards the 5671-port (AMQPS-port on the central ECP-broker). If you're missing two connections, restart ECP-endpoint.
ECP Configuration Reload – process is stuck	1. Restart ECP-endpoint. Test message flow.
ECP Certificate renewal fails ECP Cannot connect to CD/Broker due to certificate problems	1. Restart ECP-endpoint. Test message flow. 2. Go to ECP-GUI-Settings. Push configuration. Restart ECP-endpoint. Wait 1-5 minutes and test message flow. 3. Control that your firewall is not interfering (deep packet inspection) with the SSL-traffic between ECP-endpoint and/or CD/Broker. Check by running something similar to this command (openssl s_client -connect ecp4.statnett.no:5671 -showcerts). The firewall must not touch/change anything in SSL-traffic. 4. Go to ECP-GUI-Settings. Renew Manually. Wait 1-5 minutes and test message flow. 5. Re-register: Go to ECP-GUI-Settings and press button for "Initiate Registration". You may need to ask ecp@statnett.no for a new registration keystore and you must ask ecp@statnett.no to approve a new registration after you've completed your part.

Statnett

ECP connects to a wrong broker ECP tries to send messages to the wrong broker	1. Restart ECP-endpoint 2. Manually remove some rows from some tables – the following show how to do this in Derby-database, but the same SQL applies for any database of course. a. Connect to the database (see chapter "How to connect to a Derby-database) b. Run the following SQLs: i. select * from ecp.message_path_sender; ii. select * from ecp.message_upload_route; iii. select * from ecp.message_path; c. find the set of IDs that are connected to the wrong broker (usually with type MESSAGE_PATH_TYPE of "ACKNOWLEDGMENT"). Assume you found ID 101 and 102 as common ID in all tables. If there are some variations, adjust the SQL to delete those IDs. Make sure not to delete any ID that is connected to your "local" broker. Now run the appropriate SQLs: i. delete from ecp.message_path_sender where message_path in (101,102); ii. delete from ecp.message_upload_route where id in (101,102); iii. delete from ecp.message_path where id in (101,102); 3. Re-register by following 6 above
ECP or EDX stuck/frozen; messages are not received/retrieved	It can often be hard to know where the problem is located. Even if EDX is stuck, it could be that the problem-queue is on ECP. Therefore, make sure to investigate both servers. The advice below will be to delete/purge message – it is expected that proper B2B communication have business acknowledgements, to that even loss of messages will be handled. 1. Restart ECP/EDX. It is best (but not necessary) if EDX is started 30 sec before ECP, if they're both taken down. 2. Purge a specific queue which is assumed to be full: a. Install Hawtio (see chapter "How to install Hawtio"). Wait 30 sec. b. Connect to your ECP or EDX GUI, but change the URL path to "/hawtio/". c. The top-menu should show "ActiveMQ" – click on that choice. d. All queues will be listed, find queues where queue size > 0 e. Consider purging the queue (click on queue – choose "Delete"-submenu – Purge) or delete specific messages. 3. Purge all messages from ECP/EDX-endpoint: a. For EDX: Delete the edx-activemq-data-folder and restart. b. For ECP: Delete the internalbroker-folder and restart. 4. Reset endpoint: a. For EDX: Delete both db and edx-activemq-data folders and restart EDX. Be aware that EDX will not work properly until it has received a new ServiceCatalogue-copy. This happens automatically after restart, but it requires that a message is transmitted to EDX from ECP (you can see this message in ECP inbox, but not in EDX messages). If you already have other messages coming in from ECP to EDX, the SC-copy will

Statnett

	not be processed before the other messages are processed. BUT: The EDX may not be able to process any messages without the SC-copy and will get stuck! To avoid this, you must purge messages from ECP as well. In a high-traffic-volume endpoint this may be difficult. b. For ECP: Follow option 6 in the "ECP Certificate renewal fails"-problem.

4.2 How to connect to a Derby-database:

1. Download Derby client: https://db.apache.org/derby/derby_downloads.html (we have been running 10.14.2)
2. 2. Unzip into a folder, ex /opt/derby
3. Change folder to your ECP, ex /var/lib/ecp-endpoint (your database you should now be found on "db"-folder in the folder you are located)
4. Stop ECP-endpoint - you cannot edit the DB from more than one user at the time (or make a copy of the DB-folder and work on that)
5. 5. Run /opt/derby/db-derby-10.14.2.0-bin/bin/ij
    a. ij version 10.14
    b. ij> connect 'jdbc:derby:db';
    c. The "db" marked in red above is the name of the folder, so this works if you want to copy the database to another folder and work on it while the ECP endpoint is running. While inside the ij-tool you can do all SQL and some other commands:
    d. ij> help;
    e. ij> show tables;
    f. ij> exit;
6. After exit and you've edited the database (by INSERT/UPDATE/DELETE) you should check that file permission/ownership is the same as before – and if not, change back so that the ECP/EDX process can read/change the database.

4.3 How to install Hawtio

1. Add/change the property "spring.jmx.enabled=true" to ecp.properties and edx.properties. This will make it possible for Hawtio to see the queues. Restart ECP or EDX if change was necessary.
2. Download hawtio: https://repo1.maven.org/maven2/io/hawt/hawtio-web/1.5.11/hawtio-web-1.5.11.war
3. Rename the file to "hawtio.war" and place the file in the webapps-folder of ECP and/or EDX – it will automatically install.
4. After you've done using Hawtio you should remove the war-file – Hawtio is a liability security-wise. You should also consider reversing the jmx-settings introduced in 1.

4.4 How to install/use QueueCleaner

1. Download from here: https://ediel.org/nordic-ecp-edx-group-nex/nex-statnett/
2. This tool is made by Statnett, no guarantees follow. But it seems to be working totally fine.
3. Run the tool from command line like this: java -jar QueueCleaner.jar and it will show you something like this:

Statnett

```
QueueCleaner v1.4.8 is a tool to browse or consume messages from all queues on ECP or EDX - it can be used to monitor or to
maintain the queues on brokers of ECP and EDX. It can be run locally or remotely. It can filter for certain messages, and
by so doing, be able to clean up certain messages that you want to get rid of.


Usage  : java -jar QueueCleaner.jar [OPTION] BROKER QUEUES FILTER [KEYSTORE]



The options and arguments:
 OPTION        : [-(d|m<number>)] - only one option is allowed and they may be omitted - see below for expl
                 -d                : will not consume, simply browse. Maximum 400 messages can be browsed this way.
                 -m<number>        : consume max <number> of messages. Default is to consume all message matching filters
                 -o <filename>     : will append a one-liner summary in JSON-format to the <filename>
                 -t<timeout-ms>    : specify max timeout to wait for consume on a queue. Default is 1000 (ms)
 BROKER        : <URL|PATH> - either specify a URL to a broker or a path for the ECP- or EDX-configuration
                 URL               : Specify url starting with either 'amqp' or 'amqps' + keystore-file/password args at the end
                 PATH              : Specify either the conf-path for ECP or EDX (ex: /etc/ecp-endpoint or /etc/edx-toolbox)
 QUEUES        :                   : Number of different options to specify the queues to scan:
                 (<queue-name>[,<queue-name>]*)    : will scan these queues
                 ALL                               : will scan ALL queues, but this requires PATH in previous arg
                 <edx.yml>                         : will scan ALL queues from edx.yml
                 <ecp.properties>                  : will scan ALL queues from ecp.properties (currently: only a fixed list)
 FILTER        : (<messageType>|ALL)[,TTL-<sec>][,SEN-<senderCode>][,REC-<receiverCode>][,BAU-<businessApp>] - see below for expl.
                 <messageType>|ALL  : Mandatory filter - will match both on businessType and messageType. Keyword 'ALL' matches all msg
                 TTL-<sec>          : Optional filter to specify max <sec> TTL/age for the message - if omitted it matches all msg
                 SEN-<senderCode>   : Optional filter to specify <senderCode> for the message - if omitted it matches all msg
                 REC-<receiverCode> : Optional filter to specify <receiverCode> for the message - if omitted it matches all msg
                 BAU-<businessApp>  : Optional filter to specify <BA> for the message - if omitted it matches all msg
 KEYSTORE      : <keystore-file> <keystore-password> - these two parameters are only necessary if you specify 'amqps' as URL argument.
                                                       If they are specified they will override keystore params found in PATH-config

Example 1: Consume all messages from standard ActiveMQ DLQ:
    java -jar QueueCleaner.jar amqp://localhost:5672 ActiveMQ.DLQ ALL

Example 2: Consume older than 1h messages on edx.endpoint.reply (case: typically leftover msg that BA is not consuming):
    java -jar QueueCleaner.jar amqp://localhost:5672 edx.endpoint.reply ALL,TTL-3600

Example 3: Consume older than 1h messages on ecp.endpoint.inbox and ecp.endpoint.send.event (case: msg stuck on-route from ECP to EDX):
    java -jar QueueCleaner.jar amqp://localhost:5672 ecp.endpoint.inbox,ecp.endpoint.send.event ALL,TTL-3600

Example 4: Consume max 40 message from DLQ (use SSL) (case: consume 1d old AOL-msg from DLQ sent from ECO-2 endpoint):
    java QueueCleaner.jar -m40 amqps://toolbox:password@localhost:5672 ActiveMQ.DLQ NBM-CIM-PTZ12-MTZ35,TTL-86400,SEN-50V-NBM-E2HT-ATP authKeystore.jks password

Example 5: Browse first 400 messages from standard ActiveMQ DLQ:
    java -jar QueueCleaner.jar -d amqp://localhost:5672 ActiveMQ.DLQ ALL

Example 6: Browse first 400 messages from all queues on EDX:
    java -jar QueueCleaner.jar -d /etc/edx-toolbox ALL ALL
```

This tool can be used to monitor queues on ECP/EDX or purge certain messages (specified by a filter) on specific queues. It supports AMQP and AMQPS. It can parse edx.yml and find all queues. For ECP, the QC have a fixed list of queues that it will check if you say "ALL" queues. This tool can help you make sure all queues are consumed and also clean DLQ or reply-queues if old messages are gathering there.