UNLIMITED PRODUCTION

eXpress Communication System

# Administrator's Guide

## Updating

Build 3.36
03/17/2025

| | |
|---|---|
| Mailing address: | 127030, Moscow,<br>24/1 Novoslobodskaya Street |
| Phone: | +7 (499) 288-01-22 |
| E-mail: | sales@express.ms |
| Web: | https://express.ms/ |

# TABLE OF CONTENTS


**TABLE OF CONTENTS** ..... **3**

**INTRODUCTION** ..... **4**

**UPDATING OPERATION SYSTEM** ..... **5**

**UPDATING THE SERVER MANUALLY** ..... **6**

**Single CTS** ..... **6**

**Front CTS and Back CTS** ..... **7**

**Updating eXpress to Version 3.25** ..... **8**

Preparation before updating the servers ..... 8

Updating Media Server ..... 9

Updating Media on Front or Single Servers ..... 12

Updating Single, Front and Back Servers ..... 13

**Fault-Tolerant Configuration** ..... **13**

**UPDATING POSTGRESQL** ..... **17**

**Update Procedure** ..... **17**

**Backup** ..... **18**

**Version Rollback** ..... **18**

**Updating without Access to apt.postgresql.org** ..... **18**

**EMERGENCY SITUATIONS WHEN UPDATING FROM THE LOCAL REGISTRY REPOSITORY** ..... **20**

**CERTIFICATE UPDATE PROCEDURE** ..... **21**

**UPDATING THE CALLS VOEX SERVER SETTINGS** ..... **22**

**MIGRATION OF MESSAGING FROM VERSION 3.10 AND LOWER TO VERSION 3.10 AND HIGHER** ..... **24**

**Determining the size of the tables** ..... **24**

**Migrating Large Databases** ..... **25**

Preparing for Migration ..... 26

Migration ..... 28

**Updating Statistics** ..... **31**

**CHANGE HISTORY** ..... **32**

# INTRODUCTION

This manual is intended for administrators of the product eXpress Communication System (hereinafter referred to as eXpress CS, eXpress, system). It contains information necessary for updating and configuring the system.

**Product Support Service** You can contact the product support service by e-mail support@express.ms. The page of the product support service on the Unlimited Production website is available at https://express.ms/faq/.

**Website.** Information on the product by Unlimited Production can be found on the website https://express.ms/.

# UPDATING OPERATION SYSTEM

### To update the operating system of eXpress server components:

1. Stop the eXpress software containers. To do so, run the following command:

   ```
   cd /opt/express && DPL_PULL_POLICY=never dpl --dc stop
   ```

   Then, run the following command:

   ```
   cd /opt/express-voice && DPL_PULL_POLICY=never dpl --dc stop
   ```

2. Make sure all containers are stopped (are in "exited" status). To do this, run the following command on each server:

   ```
   docker ps -a
   ```

3. Shut down the Docker service using the following command:

   ```
   systemctl stop docker.service docker.socket
   ```

4. Back up the /var/lib/docker directory or make a backup image of the virtual machine.
5. Update operating system components.

   **Note.** The update procedure depends on the type of OS. When updating, follow the manufacturer's instructions for updating the operating system.

   Example of command for Debian/Ubuntu/Astra Linux:

   ```
   apt-get update && apt-get upgrade
   ```

   Example of command for Red Hat/Centos/AlmaLinux/Rocky Linux:

   ```
   yum update
   ```

6. After updating the operating system, reboot the system and make sure that the operating system has booted.
7. Check if all containers are running using the following command:

   ```
   docker ps -a
   ```

   If the containers are not running, run the following command to view the event log:

   ```
   dpl --dc logs --tail=200  <container_that_was_not_launched>
   ```

   If the server update procedure is completed correctly, the administrator console (web interface) https://ccs_host/admin will be available within five minutes.

   Check to see if users appear in the "Users" section of the administrator console.

8. Check whether the following works in eXpress CS:
   - messages in chats;
   - individual calls;
   - group calls.

# UPDATING THE SERVER MANUALLY

**Attention!** Make a backup before performing the update procedure! The following update process is relevant for eXpress 3.x and higher. If the eXpress version is lower than 3.0, go to the "Upgrading eXpress to Version 3.0" section.

## SINGLE CTS

**Attention! Starting with version 3.26, the architecture of the eXpress app has changed!**

- **to update the eXpress app to version 3.26 and above**, Move the eXpress media components to a separate Media Server as described in the document "Administrator's Guide. Media и Transcoding Servers".
- **if the eXpress media components remain deployed on the CTS Front (or Single) server**, update servers according to the section "Updating Media on Front or Single Servers".

### Updating the Media server:

1. Go to the eXpress directory: cd /opt/express-voice/.
2. Apply the new configuration:

```
dpl -p
```

3. Run the update:

```
dpl -d
```

4. Check the logs for errors with the following command:

```
dpl --dc logs --tail=200 -f
```

### To update the CTS server:

1. Go to the eXpress directory: cd /opt/express/.
2. Stop the services from the eXpress directory:

```
DPL_PULL_POLICY=never dpl --dc stop
```

3. Back up /var/lib/docker/volumes (or wherever they are located on this system).

   **If the previous version of NGINX is lower than 1.20.1 , and Let's Encrypt certificates are used:**

   - Clear the letsencrypt storage (one time):

```
rm -rf cts/letsencrypt
dpl cadvinstall && dpl nxinstall
```

4. Update node exporter and container advisor:

```
dpl cadvinstall && dpl nxinstall
```

5. Run the update:

```
dpl -d
```

   After starting the update, some time is required to carry out internal procedures (approximate duration is 10-15 minutes).

6. Check the logs for errors with the following command:

```
dpl --dc logs --tail=200 -f
```

**To roll back the update**, edit the **settings** file by specifying the appropriate parameter, for example:

```
images:
     trusts: trusts:1.28.0
```

## FRONT CTS AND BACK CTS

**Attention! Starting with version 3.26, the architecture of the eXpress app has changed!**

- **to update the eXpress app to version 3.26 and above**, Move the eXpress media components to a separate Media Server as described in the document "Administrator's Guide. Media и Transcoding Servers".
- **if the eXpress media components remain deployed on the CTS Front (or Single) server**, update servers according to the section "Updating Media on Front or Single Servers".

### Updating the Media server:

1. Go to the eXpress directory: cd /opt/express-voice/.
2. Stop the services from the eXpress directory:

   ```
   DPL_PULL_POLICY=never dpl --dc stop
   ```

3. Run the update:

   ```
   dpl -d
   ```

4. Check the logs for errors with the following command:

   ```
   dpl --dc logs --tail=200 -f
   ```

---

Attention! Before starting the update procedure, check the changes in the network interaction table!

---

The Front CTS server is updated first, then the Back CTS server.

### To update the Front CTS server:

1. Go to the eXpress directory: cd /opt/express/.
2. Stop the app with the following command:

   ```
   DPL_PULL_POLICY=never dpl --dc stop
   ```

3. Back up the /var/lib/docker/volumes files (after a few days of operation, the copies of the files can be deleted).

   If the previous version of NGINX is lower than 1.20.1 , and Let's Encrypt certificates are used:

   - Clear the letsencrypt storage (one time):

     ```
     rm -rf cts/letsencrypt
     ```

4. Run the update:

   ```
   dpl -d
   ```

   After starting the update, some time is required to carry out internal procedures (approximate duration is 10-15 minutes).

5. Check the logs for errors with the following command:

   ```
   dpl --dc logs --tail=200 -f
   ```

### To update the Back CTS server:

1. Go to the eXpress directory: cd /opt/express/.
2. Stop the app with the following command:

```
DPL_PULL_POLICY=never dpl --dc stop
```

3. Back up the /var/lib/docker/volumes files (after a few days of use, the copied files can be deleted).
5. Apply the new nginx configuration:

```
dpl -p
```

6. Start the utility services:

```
dpl -d nginx postgres kafka etcd redis
```

7. Run the update:

```
dpl -d
```

After starting the update, some time is required to carry out internal procedures (approximate duration is 10-15 minutes).

4. Check the logs for errors with the following command:

```
dpl --dc logs --tail=200 -f
```

## UPDATING EXPRESS TO VERSION 3.25

**Attention! Starting with version 3.26, the architecture of the eXpress app has changed!**

- **to update the eXpress app to version 3.26 and above**, Move the eXpress media components to a separate Media Server as described in the document "Administrator's Guide. Media и Transcoding Servers".
- **if the eXpress media components remain deployed on the CTS Front (or Single) server**, update servers according to the section "Updating Media on Front or Single Servers".

## PREPARATION BEFORE UPDATING THE SERVERS

**Attention!** Before performing these steps, please make sure that on all eXpress servers, FQDN-media resolves to an IP address from the Media server's local network.

1. Go to the Administrator Console (web interface).
2. Go to the "VoEX" section.
3. In the "Janus WS URL" field, enter the Media server URL in the following format: ws://fqdn-media:8188
4. In the "External Janus host" field, enter the IP address specified in the "janus_nat_1_1_mapping" parameter.

**Note.** In case you make calls between different servers in the local network, in the "External Janus host" field, specify the FQDN entered in the Split DNS settings (the name must resolve to an external IP address on the Internet, and to an internal IP address in the local network).

5. Click the following button:  .
6. Make sure that the new server appears in the list on the screen and the message "Janus URL added" is displayed in the window header.

7. Check the connection of the CTS server to the Media server using the "Load" parameter (see Figure 1).

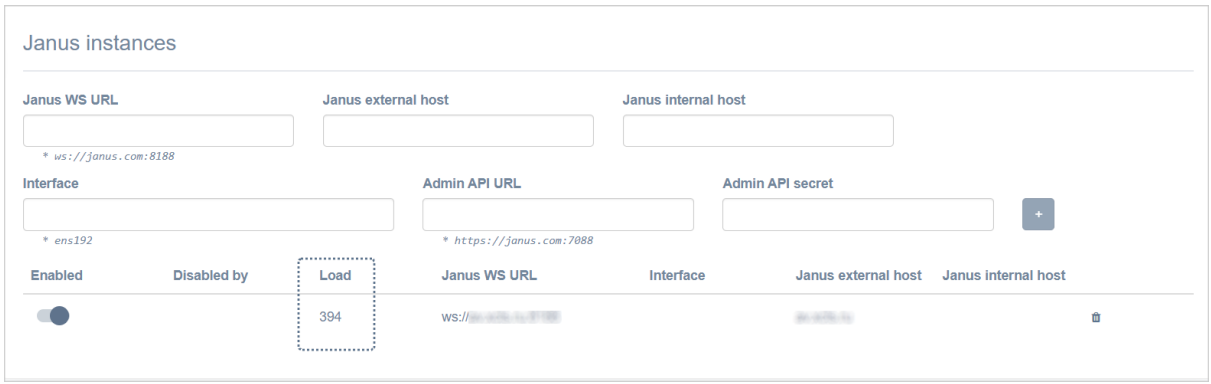



*Figure 1*

8. If a Janus server was previously added at the address, in the "Janus instances" tab, remove the Janus server by clicking the 🗑 icon (see Figure 2).

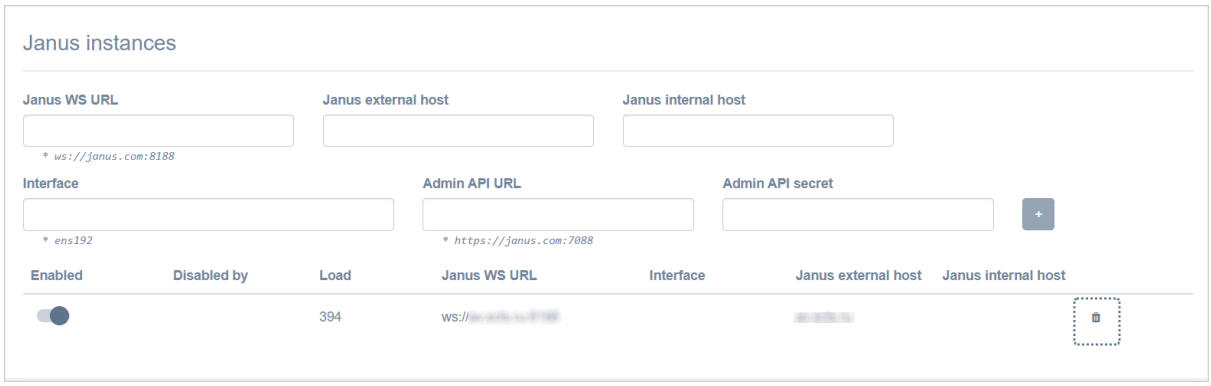



*Figure 2*

9. Go to the eXpress directory: cd /opt/express/.
10. Update the eXpress services:

```
DPL_PULL_POLICY=never dpl --dc stop && dpl -d
```

## UPDATING MEDIA SERVER

### If recording is not used:

1. Go to the eXpress directory: cd /opt/express-voice
2. Edit the settings.yaml configuration file by adding the following at the end of the file:

```
transcoding_storage_enabled: false
```

3. Update the eXpress services:

```
DPL_PULL_POLICY=never dpl --dc stop && dpl -d
```

### If recording is used:

1. Go to the eXpress directory: cd /opt/express-voice
2. Add certificates from FQDN-media (automatic issuance of certificates from Let's Encrypt is not supported):

```
certs/express.crt
certs/express.key
```

3. Edit the settings.yaml configuration file, the parameters for editing in the example are highlighted in red. The list of parameters is listed in Table 1:

```
transcoding_hosts:
  cts:
    ccs_host: fqdn-cts
    api_internal_token: **********
    storages_tokens_mapping:
      fqdn-media:
        token: **********
        ssl_envs_prefix: "TSS"

api_internal_token: **********
ccs_host: fqdn-media
transcoding_storage_enabled: true

# optional parameters
# if the certificates are not public, then disable certificate
verification:
tc-cts_env_override:
  TSS_SSL_ENABLED: true
  TSS_SSL_VERIFY: verify_none
```

*Table 1*

| Setting name | Value |
|---|---|
| transcoding_hosts | The list of hosts objects (CTS) consists of the following parameters:<br>• ccs_host — FQDN of the CTS server;<br>• api_internal_token — token for API queries (copy from the /opt/express/settings.yaml files located on the corresponding ccs_hosts servers). |
| storages_tokens_mapping | The list of hosts objects (FQDN name of the Media server) consists of the following parameters:<br>• token — api_internal_token of the Media server;<br>• ssl_envs_prefix — certificate prefix. |
| api_internal_token | Token for API requests, you can generate any 16 characters (CTS has a similar parameter, the value of which may not match the current one) |
| transcoding_storage_enabled | Enable temporary storage of records service (disabled by default) |
| ccs_host | FQDN name of the Media server |
| tc-ct_env_override | Additional parameters for transcoding |
| TSS_SSL_ENABLE | Enable/disable advanced transcoding settings |
| TSS_SSL_VERIFY | Verification of certificate for transcoding |

**Attention!** The update command must be executed on the Media server only.

4. Run the update with the following command:

```
dpl --dc down && docker volume prune -a && dpl -d
```

5. Check the availability of the Media server using the following command:

```
curl https://fqdn-media/testtest
```

6. To get the nginx container logs on the Media server, run the following command:

```
docker logs voex-nginx-1 | grep testtest
```

The log should show the curl request(s), for example:

```
fqdn-media X.X.X.X - - [02/Oct/2024:08:50:34 +0000] "GET
/testtest HTTP/1.1" 204 0 "-" "curl/8.5.0" "X.X.X.X"
fqdn-media X.X.X.X - - [02/Oct/2024:08:50:42 +0000] "GET
/testtest HTTP/1.1" 204 0 "-" "curl/8.5.0" "X.X.X.X"
```

If there are no curl requests in the logs, configure split DNS to the local IP address of the Media server.

## UPDATING MEDIA ON FRONT OR SINGLE SERVERS

1. Go to the eXpress directory: cd /opt/express-voice
2. Add certificates from FQDN-media (automatic issuance of certificates from Let's Encrypt is not supported):

```
transcoding_storage_enabled: true
traefik_bind_port_https: 8443
nginx_listen_http: false
ccs_host: fqdn_media
phoenix_secret_key_base: phoenix_secret_key
api_internal_token: api_internal_token_cts
coturn_options:
  cpuset: 10-11
  mem_limit: 2048m
janus_options:
  cpuset: 12-13
tc-cts_options:
  cpuset: 14-15
```

*Table 2*

| Setting name | Value |
|---|---|
| api_internal_token | Token for API requests, you can generate any 16 characters (CTS has a similar parameter, the value of which may not match the current one) |
| transcoding_storage_enabled | Enable temporary storage of records service (disabled by default) |
| ccs_host | FQDN name of the Media server |
| phoenix_secret_key | api token of the CTS server specified on the Back server in the /opt/express/settings.yaml file in the phoenix_secret_key_base parameter |
| cpuset | limitation of the use of CPU cores by container by core numbers, core numbering starts from 0 |
| mem_limit | limitation of container memory usage |
| nginx_listen_http | listen on port 80 (true by default)<br>turn it off because it is busy |
| traefik_bind_port_https | port for listening https, busy by default |

3. Change the block responsible for setting Transcoding according to the example below:

```
transcoding_hosts:
  cts:
    ccs_host: fqdn-cts
    api_internal_token: **********
    storages_tokens_mapping:
      fqdn-media:
        token: **********
        ssl_envs_prefix: "TSS"
        override_endpoint: https://fqdn_media:8443
      # This block is optional (if janus_url over ip)
      ip-media:
        token: **********
        ssl_envs_prefix: "TSS"
        override_endpoint: https://fqdn_media:8443


tc-cts_env_override:
  TSS_SSL_ENABLED: true
  TSS_SSL_VERIFY: verify_none
```

*Table 3*

| Setting name | Value |
|---|---|
| transcoding_hosts | The list of hosts objects (CTS) consists of the following parameters:<br>• ccs_host — FQDN of the CTS server;<br>• api_internal_token — token for API queries (copy from the /opt/express/settings.yaml files located on the corresponding ccs_hosts servers). |
| storages_tokens_mapping | The list of hosts objects (FQDN name or IP of the Media server) consists of the following parameters:<br>• token — api_internal_token of the Media server, which was made in item 3;<br>• ssl_envs_prefix — certificate prefix.<br>• override_endpoint - url transcoding_storage |
| api_internal_token | Token for API requests, you can generate any 16 characters (CTS has a similar parameter, the value of which may not match the current one) |
| transcoding_storage_enabled | Enable temporary storage of records service (disabled by default) |
| ccs_host | FQDN name of the Media server |
| tc-ct_env_override | Additional parameters for transcoding |
| TSS_SSL_ENABLE | Enable/disable advanced transcoding settings |
| TSS_SSL_VERIFY | Verification of certificate for transcoding |

## UPDATING SINGLE, FRONT AND BACK SERVERS

1. Go to the eXpress directory: cd /opt/express/.
2. Stop the services from the eXpress directory:

```
DPL_PULL_POLICY=never dpl --dc stop
```

3. Back up /var/lib/docker/volumes (or wherever they are located on this system).
4. Update node exporter and container advisor:

```
dpl cadvinstall && dpl nxinstall
```

5. Run the update using the following command:
   - for Front and Single server:

```
dpl -d
```

   - for Back server:

```
dpl -d traefik nginx postgres redis kafka etcd && dpl -d
```

## FAULT-TOLERANT CONFIGURATION

If it is impossible to use automatic update scripts, perform the update manually.

### To update a fault-tolerant configuration:

1. Copy the new versions of the Docker images and the image loading script to the /tmp/images directory and load the images using the load.sh script as follows:

```
cp *.tar /tmp/images/
cp /opt/deploy/script/load.sh /tmp/images/
cd /tmp/images/
./load.sh
```

2. Connect to the console of the Back and Front server of the cluster with index 01 and 02.
3. Run the following command to stop the antivirus:

```
systemctl stop kesl klnagent64
```

**Note.** If the server freezes when you stop the antivirus, reboot both cluster nodes through the virtualization system.

4. Run the following command:

```
pcs status
```

5. Make sure that the cluster resources are running according to the list below:
   - dlm-clone [dlm] (back cluster) – running on both cluster nodes;
   - clvmd-clone [clvmd] (back cluster) – running on both cluster nodes;
   - clusterfs-clone [clusterfs] (back cluster) – running on both cluster nodes;
   - cluster_ip – running on one cluster node;
   - dockerd – running on one cluster node;
   - node_exporter (back cluster) – running on one cluster node;
   - cadvisor (back cluster) – running on one cluster node;
   - vmfence (back cluster) – running on one cluster node.

   If the status of the cluster resources does not correspond to those listed above, run the following command, replacing *resource_name* with the name of the problematic resource:

```
pcs resource cleanup resource_name
```

6. On the Back cluster nodes with index 01 and 02, run the following command:

```
ls -la /opt/ex_data/files
```

**Note.** If the directory list display freezes, it is necessary to reboot both cluster nodes through the virtualization system.

7. Connect to the console of the Back server of the cluster with index 01 or 02 and run the following command:

```
pcs status | grep dockerd
```

**Note.** The command is run to determine the current primary node running cluster resources.

8. Connect to the console of the current primary node of the Back cluster and sequentially run the following commands:

```
cd /opt/express
dpl -g
```

9. Connect to the console of the Front server of the cluster with index 01 or 02 and run the following command:

```
pcs status | grep dockerd
```

**Note.** The command is run to determine the current primary node running cluster resources.

10. Connect to the console of the current primary node of the Front cluster and sequentially run the following commands:

```
cd /opt/express
```

```
dpl -g
cd /opt/express-voice
dpl -g
```

11. Connect to the console of the Back cluster secondary node and run the following commands sequentially, replacing full_fqdn_slave_server with the fully qualified domain name of the secondary cluster node:

```
pcs resource move cluster_ip full_fqdn_slave_server
pcs resource move dockerd full_fqdn_slave_server
```

12. Wait for the dockerd and cluster_ip cluster resources to switch over to the secondary node of the Back cluster. To monitor the status of resources, periodically run the following command:

```
pcs status
```

13. After the resources are switched to the secondary node of the Back cluster, run the following commands sequentially:

```
cd /opt/express
dpl -g
```

14. Connect to the console of the Front cluster secondary node and run the following commands sequentially, replacing full_fqdn_slave_server with the fully qualified domain name of the secondary cluster node:

```
pcs resource move cluster_ip full_fqdn_slave_server
pcs resource move dockerd full_fqdn_slave_server
```

15. Wait for the dockerd and cluster_ip cluster resources to switch over to the secondary node of the Front cluster. To monitor the status of resources, periodically run the following command:

```
pcs status
```

16. After the resources are switched to the secondary node of the Front cluster, run the following commands sequentially:

```
cd /opt/express
dpl -g
cd /opt/express-voice
dpl -g
```

17. Connect to the console of the current primary node of the Back cluster and sequentially run the following commands:

```
cd /opt/express
dpl --dc stop
dpl nxinstall && dpl cadvinstall
dpl -d
```

18. After the server update is complete, open the container operation logs:

```
dpl --dc logs --tail=100 -f
```

19. Wait until the output of container logs stops except for the nginx container.
20. Connect to the console of the Front server of the cluster with index 01 or 02 and run the following command:

```
pcs status | grep dockerd
```

**Note.** The command is run to determine the current primary node running cluster resources.

21. Connect to the console of the current primary node of the Front cluster and sequentially run the following commands:

```
cd /opt/express
dpl --dc stop
dpl -d
cd /opt/express-voice
dpl --dc stop
dpl -d
```

22. After updating the primary nodes of the Front and Back clusters, check the functioning of the system, check the logs for errors by running the following command:

```
dpl --dc logs --tail=200 -f
```

23. Connect to the console of the Back cluster secondary node and run the following commands sequentially, replacing full_fqdn_slave_server with the fully qualified domain name of the secondary cluster node:

```
pcs resource move cluster_ip full_fqdn_slave_server
pcs resource move dockerd full_fqdn_slave_server
```

24. Wait for the dockerd and cluster_ip cluster resources to switch over to the secondary node of the Back cluster. To monitor the status of resources, periodically run the following command:

```
pcs status
```

25. After the resources are switched to the secondary node of the Back cluster, run the following commands sequentially:

```
cd /opt/express
dpl --dc stop
dpl nxinstall && dpl cadvinstall
dpl -d
```

26. Once the server update is complete, open the output of the container operation log and wait until the output of container logs stops except for the nginx container:

```
dpl --dc logs --tail=100 -f
```

27. Connect to the console of the Front cluster secondary node and run the following commands sequentially, replacing full_fqdn_slave_server with the fully qualified domain name of the secondary cluster node:

```
pcs resource move cluster_ip full_fqdn_slave_server
pcs resource move dockerd full_fqdn_slave_server
```

28. Wait for the dockerd and cluster_ip cluster resources to switch over to the secondary node of the Front cluster. To monitor the status of resources, periodically run the following command:

```
pcs status
```

29. After the resources are switched to the secondary node of the Front cluster, run the following commands sequentially:

```
cd /opt/express
dpl --dc stop
dpl -d
cd /opt/express-voice
dpl --dc stop
dpl -d
```

30. Connect to the console of the Back and Front server of the cluster with index 01 and 02, run the following command to start the antivirus:

```
systemctl start kesl klnagent64
```

# UPDATING POSTGRESQL

**Important!** There is a risk of data loss when updating the database. It is recommended to perform the update procedure during minimal user activity or when all services are stopped. Before the procedure, it is strictly necessary to perform a full backup of the database.

## UPDATE PROCEDURE

### Before updating the embedded database:

1. Make sure you have enough free disk space available.

   The update requires disk space equal to the size of the existing database. The current size of the database can be determined using the following command:

   ```
   docker system df -v | grep postgres_data
   ```

2. Check your current version of PostgreSQL.

   If errors occur after upgrading, it may be necessary to downgrade your PostgreSQL version. Therefore, it is advisable to know what version of the database was running before the update. It is important to run the following command to display the current database version before updating the PostgreSQL image:

   ```
   DPL_PULL_POLICY=never dpl --dc exec postgres cat
   /var/lib/postgresql/data/PG_VERSION
   ```

3. Make sure you have access to apt.postgresql.org. This access is required to download old version executables. If you don't have access, see the "Updating without Access to apt.postgresql.org" section.

   To roll back the database version, it is necessary to move the files from the backup directory to a higher-level directory.

4. At the end of the settings.yaml file, add a setting that starts updating the database container at startup:

   ```
   postgres_upgrade: true
   ```

   **Note.** If this setting is not present, new versions of the PostgreSQL image will display the error "postgres_upgrade disabled, exiting" when launched.

5. Update the PostgreSQL image and run it using the following command:

   ```
   dpl -d postgres
   ```

6. To track the update process in the logs, use the following command:

   ```
   dpl --dc logs -f --tail=1 postgres
   ```

   When the update is complete, the following message will be displayed in the logs:

   ```
   DB upgrade is done, please disable postgres_upgrade in the
   settings
   ```

7. Disable the postgres_upgrade setting, otherwise the PostgreSQL container will not start.
8. Remove the postgres upgrade setting and download the standard image using the following command:

   ```
   DPL_PULL_POLICY=never dpl -d postgres
   ```

**For automatic update**, use the postgres_auto_upgrade setting, with it the database will be automatically updated when an image with a new version is released.

If for some reason the database version needs to be left unchanged, postgres images remain available without an update script. They can be enabled via the images parameter, for example:

```
images:
  postgres: postgres:9.5.24
```

## BACKUP

When updating the database, the old version is saved so that the version can be rolled back. The copy is located in the same volume that the postgres container is using, in a directory named upgrade_backup_<timestamp>.

If, after the update, all services are working normally and a rollback is not required, delete the backup copy with the following command:

```
 dpl --dc exec postgres find /var/lib/postgresql/data -type d -name
upgrade_backup_* -exec rm -r {} \;
```

## VERSION ROLLBACK

To roll back the database version, it is necessary to move the files from the backup directory to a higher-level directory.

**To work with the volume file system**, launch a temporary container:

```
DPL_PULL_POLICY=never dpl --dc run --rm --entrypoint=/bin/bash
postgres
```

Or perform the operation directly from the host. Docker volumes are usually stored in /var/lib/docker/volumes.

After moving the files, set the postgres image of the previous version in the settings file, for example:

```
images:
  postgres: postgres:9.5.24
```

## UPDATING WITHOUT ACCESS TO APT.POSTGRESQL.ORG

For updating without access to apt.postgresql.org, a special image 14.4-from-9.5 was compiled.

**Note.** Only upgrading from version 9.5 is supported.

### To update the database to version 14.4:

1. In the settings file, add an images section with the appropriate tag:

   ```
   images:
     postgres: postgres:14.4-from-9.5
   ```

2. At the end of the settings.yaml file, add a setting that starts updating the database container at startup:

   ```
   postgres_upgrade: true
   ```

   **Note.** If this setting is not present, new versions of the PostgreSQL image will display the error “postgres_upgrade disabled, exiting” when launched.

3. Update the PostgreSQL image and run it using the following command:

```
dpl -d postgres
```

4. To track the update process in the logs, use the following command:

```
dpl --dc logs -f --tail=1 postgres
```

When the update is complete, the following message will be displayed , in the logs:

```
DB upgrade is done, please disable postgres_upgrade in the settings
```

5. Remove the added images section and the postgres upgrade setting and download the standard image using the following command:

```
DPL_PULL_POLICY=never dpl -d postgres
```

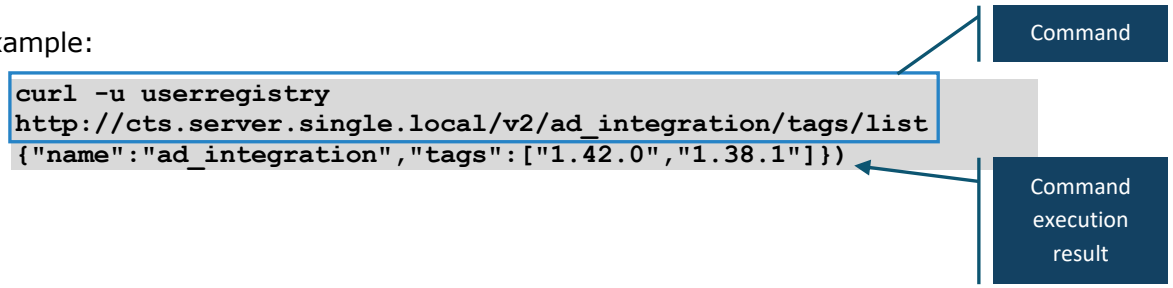
# EMERGENCY SITUATIONS WHEN UPDATING FROM THE LOCAL REGISTRY REPOSITORY

The emergency situations listed below can occur if you have a locally deployed Registry server.

**Situation 1. Absence of access to the Internet from the site with the repository.**

1. From a node with Internet access, download the latest containers using the script that is available via this link (download.sh attachment).
2. Run the second script that is available via this link (upload.sh attachment) and wait for the download to finish.
3. Make a test request from the console using a URL and get the versions in the repository.

(Example:




**Situation 2. If in point 3 of the previous operation the result of the command is "no basic auth credentials".**

1. Delete the .docker/config.json file.
2. Re-authorize in the Docker registry.

# CERTIFICATE UPDATE PROCEDURE

To operate properly, the product requires a certificate for the external name of the eXpress service (FQDN or wildcard), which was issued by a public trusted certification authority and meets the following requirements:

- version 3 and not lower than TLS 1.2;
- key length at least 2048 bits
- SHA 256 signature algorithm;
- X.509 syntax version 3;
- unencrypted private key.

The file must contain the server certificate, intermediate certification authority and root certification authority certificates. The format of the certificates must conform to Base64 encoding. The private key file must contain an unencrypted Base64 encoded private key.

An example certificate file structure is shown in the figure below (see Figure 3).

```
-----BEGIN CERTIFICATE-----
Base64 server certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64 intermediate ca
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64 root ca
-----END CERTIFICATE-----
```

*Figure 3*

The use of a free certificate from Let`s Encrypt is supported.

**Attention!** The certificate and key file names must be express.crt, express.key.

### To renew a certificate on Single, Front, Back (RTS, ETS, CTS) servers:

1. Prepare the certificate in accordance with the above requirements.
2. Update the certificate files located in the /opt/express/certs folder.
3. Run the following command in the console:

```
cd /opt/express && dpl -p && dpl --dc restart traefik
```

### To renew a certificate on the Media server:

1. Prepare the certificate in accordance with the above requirements.
2. Update the certificate files located in the folder /opt/express-voice/certs/.
3. Run the following command in the console:

```
cd /opt/express-voice && dpl -p && dpl --dc restart coturn
traefik
```

# UPDATING THE CALLS VOEX SERVER SETTINGS

### To update the Single CTS call server to version 3.16 and higher:

1. Connect to the Single CTS server via SSH.
2. Run the following command:

```
cd /opt/express-voice/
```

3. Edit the settings.yaml configuration file:
   - In the janus_ws_acl parameter, add the IP address of the network card:

     ```
     janus_ws_acl: 172.18.0.,X.X.X.X
     ```

     (where X.X.X.X is the IP address of the network card for example).
4. Run the following command:

```
dpl -d janus
```

5. Go to the administrator console (web interface)https://ccs_host/admin.
6. Go to the "VoEX" section.
7. Enter the Janus WS URL value in the Janus WS URL field according to the /opt/express/settings.yaml settings file (see Figure 4).

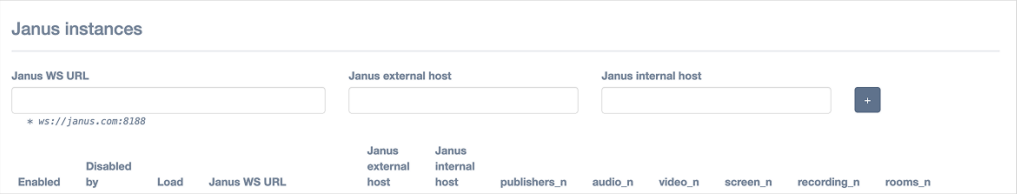



*Figure 4*

8. Starting with version 3.31 and above, enter the IP address of the external Media server in the "Janus external host" field (if the field is left blank, interserver calls will not work) (see Figure 5).

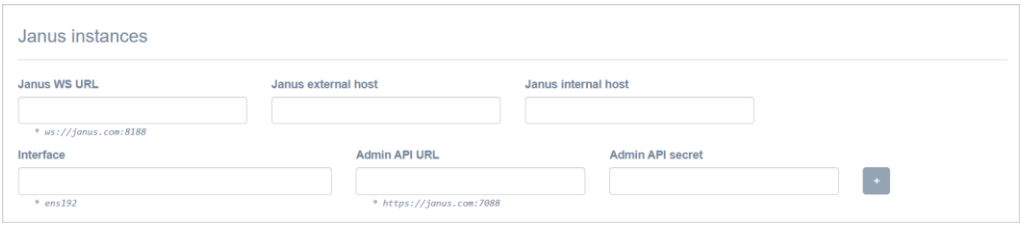



*Figure 5*

9. Click the following button:  .
10. The server be displayed in the list on the screen, and the message "Janus URL added" will be displayed in the window header.

### To update the Front CTS call server to version 3.16:

1. Go to the administrator console (web interface)https://ccs_host/admin.
2. Go to the "VoEX" section.
3. In the Janus WS URL field, enter the Janus server address (Back CTS address) according to the settings.yaml file (see Figure 6).

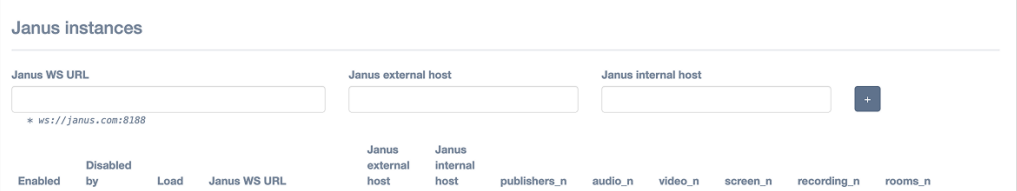



*Figure 6*

4. Starting with version 3.31 and above, enter the IP address of the external Media server in the "Janus external host" field (if the field is left blank, interserver calls will not work) (see Figure 7).

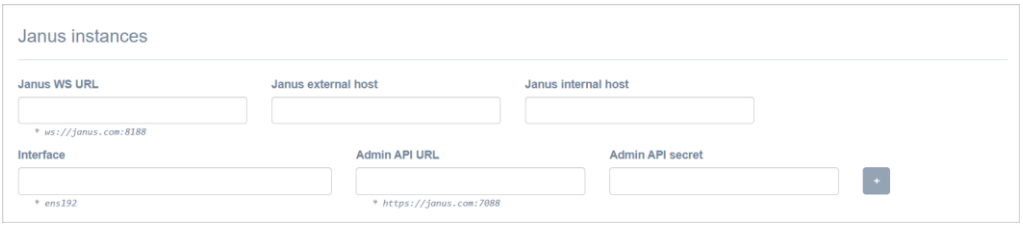



*Figure 7*

5. Click the following button:  .
6. The server be displayed in the list on the screen, and the message "Janus URL added" will be displayed in the window header.

# Appendix 1

## MIGRATION OF MESSAGING FROM VERSION 3.10 AND LOWER TO VERSION 3.10 AND HIGHER

**Attention!** The migration procedure is mandatory for upgrading from a version below 3.10. **After migrating to version 3.10+, it is not recommended to roll back messaging to previous versions!** Reverting to previous versions will involve a complex process of manually deleting new data and repeating the migration. If you encounter any problems during the migration process, please contact technical support to obtain a patch.

**Note.** Downtime during the migration can range from 5 to 30 minutes depending on the size of the database.

### DETERMINING THE SIZE OF THE TABLES

**To determine the size of the tables:**

1. Log in to the PostgreSQL DBMS console and connect to the **messaging_prod** database:
   - If the database is implemented in a Docker container, run the following commands:

     ```
     docker exec -ti cts_postgres_1 psql -U postgres
     \c messaging_prod
     ```

   - if the database is external (without cluster), run the following commands:

     ```
     su postgres
     psql
     \c messaging_prod
     ```

   - if the database is external (Patroni-based cluster), run the following commands:

     ```
     psql -h internal_node_ip -p 5432 -U express -d postgres
     \c messaging_prod
     ```

2. Run a query for the size of the database tables:

```
select schemaname as table_schema,
    relname as table_name,
    pg_size_pretty(pg_total_relation_size(relid)) as total_size,
    pg_size_pretty(pg_relation_size(relid)) as data_size,
    pg_size_pretty(pg_total_relation_size(relid) -
pg_relation_size(relid))
      as external_size
from pg_catalog.pg_statio_user_tables
where relname = 'user_chat_events' or relname = 'chat_events'
order by pg_total_relation_size(relid) desc,
         pg_relation_size(relid) desc;
```

**Attention!** If total_size user_chat_events and/or chat_events exceeds 20 GB, the downtime during automatic migration may be over >1 hour. In this case, it is recommended to perform migration according to special instructions (see "Migrating Large Databases"). For convenience, we recommend that you prepare a file with all timestamps and queries in advance.

**Note.** If the tables in your database are less than 20 GB, you can migrate data automatically (no additional actions are required).

## MIGRATING LARGE DATABASES

Some queries performed in preparation for migration may take up to ten hours to process, depending on the size of the database. For example, migrating an old message history. In this case, it is recommended to use special utilities for performing long tasks, such as **screen** or **nohup**.

### Usage example:

1. Add the query to a text file with the .sql extension, for example update.sql
2. Add read permissions to the file for the user under which PSQL is run.

### For the screen utility (recommended):

1. In the terminal on the server with the database, run the screen command.

   A terminal session will open.
2. Run the command (replace the -u and -f flag values with your own values):

```
sudo -u postgres psql -d messaging_prod -f
/home/postgres/update.sql
```

3. Exit the session by pressing <Ctrl>+<A> and then <Ctrl>+<D>.

   **Attention!** Do not press <Ctrl>+<C> and <Ctrl>+<D> while you are in a screen session and the query has not been completed.
4. To return to the screen session and ensure that the session is running and the query is being processed, run the following command:

```
screen -r
```

5. Repeat the procedure described in step 4 to check the status of the task.

   To exit the session, repeat the procedure described in step 3.

### For the nohup utility (if you are familiar with the utility or unable to use the screen utility):

1. In the terminal on the server with the database, run the following command (the command should not require any input, such as the root user password):

```
nohup sudo -u postgres psql -d messaging_prod -f
/home/postgres/update.sql &
```

2. Check the task status using the following command:

```
jobs -l
```

3. Check the output in the honup.out file.
4. Check the query execution status (after a minute of execution):

```
SELECT
  pid,
  user,
  pg_stat_activity.query_start,
  now() - pg_stat_activity.query_start AS query_time,
  query,
  state,
  wait_event_type,
  wait_event
FROM pg_stat_activity
```

```
WHERE (now() - pg_stat_activity.query_start) > interval '1
minutes'
AND datname = 'messaging_prod' AND state = 'active';
```

The result is a query display. The screen session or nohup process must be active.

```
 pid | user | query_start | query_time | query | state | wait_event_type | wait_event
-----+------+-------------+------------+-------+-------+-----------------+------------
(0 rows)
```

## PREPARING FOR MIGRATION

**Attention!** Before starting the operation, make sure there is enough free space. It is recommended that the amount of free space be 2–3 times the size of the database.

**Note.** These queries are executed when the services are running. If the database is very large, preparation can begin a few days before the migration and the steps may be performed during maintenance windows.

All queries are executed in PSQL, messaging_prod database.

1. Add new columns to the chat_events table:

```
ALTER TABLE chat_events
ADD COLUMN events_history_scope BOOLEAN,
ADD COLUMN event_info_scope BOOLEAN;
```

2. Write down the current timestamp. This timestamp needs to be saved until the moment of subsequent migration:

```
messaging_test=# SELECT now();
              now
-------------------------------
 2023-12-25 11:06:06.037283+03
(1 row)
```

3. Record the timestamp of the earliest message:

```
messaging_test=# SELECT inserted_at FROM chat_events ORDER BY
inserted_at ASC LIMIT 1;
        inserted_at
----------------------------
 2018-02-05 10:21:44.191786
(1 row)
```

4. Perform migration of message histories from the date obtained from the timestamp in step 2 + 1 minute to the date of the earliest event in step 3 in 1 year increments. Each query will take a long time to process. If for some reason the query cannot be executed, you can reduce the increments to six months/three months.

```
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-01-01' AND ce.inserted_at <= '2023-
12-25 11:07';

UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
```

```
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2022-01-01' AND ce.inserted_at <= '2023-
01-01';

UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2021-01-01' AND ce.inserted_at <= '2022-
01-01';

...

UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2018-02-05' AND ce.inserted_at <= '2019-
01-01';
```

5. If the total_size of your tables is greater than 200 GB, it will be faster to run one long general query without time steps:

```
--- query for migration without time step,
--- to be executed only in case of very large databases,
--- the use of nohup or screen is mandatory,
--- execution time can reach > 10 hours
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed';
```

6. Create new indexes for the chat_events table:

```
CREATE INDEX CONCURRENTLY
chat_events_group_chat_id_inserted_at_stealth_index ON
chat_events USING btree (group_chat_id, inserted_at) INCLUDE
(sync_id, stealth) WHERE (NOT (stealth IS NULL));

CREATE INDEX CONCURRENTLY
chat_events_group_chat_id_inserted_at_shared_index ON
chat_events USING btree (group_chat_id, inserted_at) WHERE
((events_history_scope = true) AND (shared = true));

CREATE INDEX CONCURRENTLY
chat_events_group_chat_id_inserted_at_non_shared_index ON
chat_events USING btree (group_chat_id, inserted_at) WHERE
((events_history_scope = true) AND (shared = false));

CREATE INDEX CONCURRENTLY
chat_events_sync_id_event_info_scope_index ON chat_events USING
btree (sync_id) WHERE (event_info_scope = true);
```

```
CREATE INDEX CONCURRENTLY chat_events_epoch_migration_index_1 ON
chat_events (event_type) WHERE event_type='added_to_chat' OR
event_type='user_joined_to_chat';

CREATE INDEX CONCURRENTLY chat_events_epoch_migration_index_2 ON
chat_events (event_type) WHERE event_type='left_from_chat' OR
event_type='deleted_from_chat' OR
event_type='kicked_by_cts_logout';
```

7. Tracking the progress of index creation

```
--- Tracking the progress of index creation
SELECT phase, blocks_total, blocks_done, (blocks_total -
blocks_done) as blocks_rest, tuples_total, tuples_done,
(tuples_total - tuples_done) as tuples_rest FROM
pg_stat_progress_create_index;
```

## MIGRATION

**Attention!** If the migration preparation stage was completed several days ago, then it is necessary to migrate the history from the current time to the start time of the migration preparation stage. If you are performing the migration preparation stage and the main migration process on the same day, then skip steps 1 and 2.

1. Write down the current timestamp:

```
messaging_test=# SELECT now();
              now
-------------------------------
 2023-12-27 12:14:01.026154+03
(1 row)
```

2. Update the history from the time obtained in the "Preparing for Migration" section, step 2

```
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-12-25 11:06:00';
```

3. Stop all services except postgres:
   - if the database is implemented in a Docker container, run the following command:

```
docker stop $(docker ps --format "{{.Names}}" |grep -v postgres)
```

   - if the database is external, run the following command:

```
dpl --dc stop
```

4. Go to psql, connect to the messaging_prod database.
5. Use the following command to skip the migrations you have already performed (the command text shall be entered as specified, without any changes):

```
INSERT INTO
    schema_migrations (version, inserted_at)
VALUES
    (20231031201400, now()),
```

```
    (20231031201640, now()),
    (20231101121946, now()),
    (20231101134329, now()),
    (20231116000454, now()),
    (20231129210158, now()),
    (20231205114132, now()),
    (20231205115340, now()),
    (20231205123846, now()),
    (20231211072757, now());
```

6. Write down the current timestamp:

```
messaging_test=# SELECT now();
              now
-------------------------------
 2023-12-27 12:34:39.258821+03
(1 row)
```

7. Perform migration of history from the **moment of migration preparation**. Use date and time accurate to the minute:
   - from step 1, if the additional history migration process has been performed:

```
---- updating with the time from step 1
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-12-27 12:14:00';
```

   - from step 2 of the "Preparing for Migration" section, if there was no additional history migration:

```
---- updating with the date from the "Preparing for Migration"
section, step 2
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-12-25 11:06:00';
```

8. Start the services, wait for automatic migrations to complete and for the messaging service to start.
9. Perform migration of the following data using the timestamp obtained in step 6:

```
INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, ce.inserted_at
FROM chat_events ce
INNER JOIN group_chats gc ON gc.id = ce.group_chat_id
WHERE global = true AND
shared = false AND
chat_type = 'global' AND
ce.inserted_at <= '2023-12-27 12:34:39';
```

```
INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, inserted_at
FROM chat_events ce
WHERE jsonb_array_length((event_params->>'recipients')::jsonb) >
0 AND
event_type = ANY('{message_new,call_end}') AND
ce.inserted_at <= '2023-12-27 12:34:39';

INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, inserted_at
FROM chat_events ce
WHERE jsonb_array_length((event_params->>'recipients')::jsonb) =
1 AND
event_type = 'app_event' AND
payload->>'event_type' = 'bot_notification' AND
ce.inserted_at <= '2023-12-27 12:34:39';
```

10. Perform a dump of system events from channels. It can be removed after some time, when the release is tested by time:

```
CREATE TABLE channel_events_dump AS
    SELECT ce.* FROM chat_events ce INNER JOIN group_chats c ON
c.id = ce.group_chat_id WHERE chat_type = 'channel'
    AND event_type =
ANY('{added_to_chat,deleted_from_chat,left_from_chat,admin_added
_to_chat,kicked_by_cts_logout,user_joined_to_chat}');
```

11. Remove unnecessary system events from channels:

```
DELETE FROM chat_events ce
USING group_chats c
WHERE c.id = ce.group_chat_id
AND chat_type = 'channel'
AND event_type =
ANY('{added_to_chat,deleted_from_chat,left_from_chat,admin_added
_to_chat,kicked_by_cts_logout,user_joined_to_chat}');
```

12. If your server was using notifications_bot (not an internal bot), you will need to perform migration of its messages. To do this, run the following query and substitute the bot_id with the notifications_bot identifier, using the timestamp obtained in step 6:

```
INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, now()
FROM chat_events ce
WHERE jsonb_array_length((event_params->>'recipients')::jsonb) >
0 AND
event_type = 'app_event' AND
payload->>'event_type' = 'bot_notification' AND
sender = 'bot_id' AND
ce.inserted_at <= '2023-12-27 12:34:39';
```

13. Update statistics (see the "Updating Statistics" section).
14. Remove migration indexes:

```
DROP INDEX chat_events_epoch_migration_index_1;
DROP INDEX chat_events_epoch_migration_index_2;
```

## UPDATING STATISTICS

After automatic or manual migration, it is necessary to update statistics. Run the following queries in PSQL:

```
VACUUM ANALYZE user_chat_events_v2;
VACUUM ANALYZE chat_event_meta;
VACUUM ANALYZE chat_member_epochs;
VACUUM ANALYZE chat_events;
```

# CHANGE HISTORY

The "Change History" section contains a list of changes in the document related to changes/modifications of eXpress CS.

## Build 2.8.0

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Updating without Access to apt.postgresql.org | The update procedure has been updated | CTS | page 18 |

## Build 3.0

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Updating Operation System | The section has been added | | page 4 |
| 2. | Updating Deployka | Throughout the document, the operation DEPLOYKA_SKIP_UPDATE=true was corrected to DPL_PULL_POLICY=never | | |
| 4. | Certificate Update Procedure | The /opt/express/certs certificate storage directory has been changed | | page 21 |
| 5. | Updating the Server Manually | En-Dash has been replaced with Hyphen-minus in command line examples | | |

## Build 3.5

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Certificate Update Procedure | The command for updating the certificate for server version 3.5 and higher has been changed | | page 19 |

## Build 3.10

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Migration of Messaging to Version 3.10+ | The section has been added | | page 23 |

## Build 3.14

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Certificate Update Procedure | Steps for updating the VoEX certificate have been added | | page 21 |

## Build 3.16

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Updating the Calls VoEX Server Settings | The section has been added | | page 22 |

## Build 3.26

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Updating eXpress to version 3.26 and higher | The section has been added | | page 8 |

## Build 3.28

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Updating Media Server | Added transcoding_storage_enabled parameter | | page 9 |

## Build 3.29

| No. | Section | Change | Server | Reference |
|---|---|---|---|---|
| 1. | Updating Media on Front or Single Servers | The section has been added | | page 12 |
| 2. | Updating Single, Front and Back Servers | The section has been added | | page 13 |