

Merging two systems into one system with two mandators

🌐 Language: [Deutsch](#) · [English](#)

🔧 Merging two systems into one system with two mandators

Merge two Nuclos systems of the same nuclet into one multi-tenant system – step by step with SQL and pg_dump.

HOW-TO

ADMINISTRATOR

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [1. Prerequisites](#)
- [2. Create mandators & align the level](#)
- [3. Enable multi-tenancy for BOs](#)
- [4. Data export](#)
- [5. Data import](#)
- [6. Content adjustments](#)
- [Related pages](#)



First: backup & concept

Create a [database backup](#) before each step. Read the [Mandator](#) page first. Examples use **PostgreSQL**; other databases need adjustments.

This how-to merges two systems based on the same nuclet into one system with **two mandators**. Each system's data is assigned to one mandator; initially no data is shared. To make a *single* system multi-tenant, follow steps 2, 3 and 7.

1. Prerequisites

- both systems: same Nuclos and nuclet versions
- same database system/encoding (e.g. Postgres 12, UTF-8)
- same DB schema name (assumed)

2. Create mandators & align the level

On both systems create a mandator under *Administration* *Mandator*, then align the mandator level UID via SQL:

```
INSERT INTO t_md_mandator_level(  
    struid, strname, intversion, blnshowname, datcreated, strcreated, datchanged, strchanged  
) VALUES(  
    'tmp', 'tmp', 1, false, current_date, 'me', current_date, 'me'  
);  
UPDATE t_md_mandator SET struid_t_md_mandator_level = 'tmp';  
UPDATE t_md_mandator_level SET struid = '<struid auf anderem System>' WHERE struid <> 'tmp';  
UPDATE t_md_mandator SET struid_t_md_mandator_level = '<struid auf anderem System>';  
DELETE FROM t_md_mandator_level WHERE struid = 'tmp';
```

Set mandator-specific nuclet parameters and resources (logos etc.); keep all other parameters identical on both systems.

Prepare users

There must be no identical user names – rename them on one system and assign all users to their mandator (bulk editing):

```
UPDATE t_md_user set struser = struser || '-<Mandant>';
```

3. Enable multi-tenancy for BOs

Enable per business object via the [management console](#) and set the initial mandator:

```
-package <nuclet-package> -bo "<BO-Name>" -level 1 -initial <Mandantennname> [-uniqueMandator]
```

The flag `-uniqueMandator` makes the mandator part of the unique key. For many BOs the following SQL generates a migration script:

```
set search_path to <schema_name>;

SELECT (
    'UPDATE t_md_entity SET ' || E'\n' ||
    'blnmandatorunique = true, ' || E'\n' ||
    'struid_t_md_mandator_level= ' ||
    '    ' || (SELECT struid FROM t_md_mandator_level LIMIT 1) || '    ' || ';' || E'\n' ||

    '-- Lege Spalte für Mandator in BO-Tabellen an' || E'\n' ||
    (SELECT
        STRING_AGG(
            'ALTER TABLE ' || strdbentity || ' ADD COLUMN IF NOT EXISTS struid_nuclosmandator
character varying(128); ' || E'\n' ||
            'UPDATE ' || strdbentity || ' SET struid_nuclosmandator = ' || '    ' || (SELECT struid
FROM t_md_mandator LIMIT 1) || '    ' || ';' || E'\n' ||
            'ALTER TABLE ' || strdbentity || ' ALTER COLUMN struid_nuclosmandator SET NOT NULL',
            ';' || E'\n'
        )
    FROM t_md_entity
    WHERE strvirtualentity IS NULL
    ) || ';' || E'\n'
)
```

```
psql [-U "<connect with db user>"]-qAtX <DB-name> < migrate_entities.sql > do_migrate_entities.sql
psql [-U "<connect with db user>"] <DB-name> < do_migrate_entities.sql
```



Rebuild caches/constraints

After later changes run on the management console: `invalidate caches, rebuild classes, rebuild constraints and indexes.`

4. Data export

Remove orphaned document references first:

```
DELETE FROM <schema_name>.t_md_importfile WHERE struid_documentfile = 'null';
DELETE FROM <schema_name>.t_ud_go_document WHERE struid_t_ud_documentfile = 'null';
DELETE FROM <schema_name>.t_ud_documentfile WHERE struid = 'null';
```

Export most data as COPY statements (<nuclet_prefix> = local nuclet identifier):

```
pg_dump --file "my_exported_data.sql" --host "<db host>" --port "<db port>" [--username "<connect with db
user>"] [--no-password] --verbose --format=p --blobs --data-only --no-owner --no-privileges --no-tablespaces --
no-unlogged-table-data --no-comments --schema "<schema_name>" -t "^<schema_name>.<nuclet_prefix>*" -t
<schema_name>.t_md_mandator -t <schema_name>.t_md_mandator_accessible -t <schema_name>.t_md_user -t
<schema_name>.t_md_role_user -t <schema_name>.t_md_mandator_param_value -t <schema_name>.
t_md_mandator_role_user -t <schema_name>.t_md_passwordhistory -t <schema_name>.t_md_user_setting -t
<schema_name>.t_ud_searchfilter_user -t <schema_name>.t_ud_genericobject -t "<schema_name>.t_ud_document*" -t
"<schema_name>.t_ud_go*" -t <schema_name>.t_ud_history -t <schema_name>.t_ud_lock -t <schema_name>.
t_ud_entityobject_relation -t <schema_name>.t_ud_logbook -t <schema_name>.t_ud_timelimittask -t "<db name>"
```

Adjust the nuclet prefix to that of the target system:

```
sed -iE "s/<nuclet_prefix>/<nuclet_prefix-zielsystem>/g" my_exported_data.sql
```

Export the remaining data as INSERT (with `--on-conflict-do-nothing`):

```
/usr/bin/pg_dump --file "more_exported_data.sql" --host "<db host>" --port "<db port>" [--username "<connect with db user>"] [--no-password] --verbose --format=p --blobs --data-only --no-owner --no-privileges --no-tablespaces --no-unlogged-table-data --no-comments --inserts --on-conflict-do-nothing --schema "<schema_name>" -t <schema_name>.t_md_workspace -t "<schema_name>.t_md_preference*" "<schema_name>.t_ud_print*" "<db name>"
```

5. Data import

On the target database, temporarily disable all foreign key constraints (generate a script as in step 3), then import:

```
psql [-U "<connect with db user>"] -a -v ON_ERROR_STOP=1 -f my_exported_data.sql <DB-name>  
psql [-U "<connect with db user>"] -a -v ON_ERROR_STOP=1 -f more_exported_data.sql <DB-name>
```

Then copy the other system's documents directory into the target directory.

6. Content adjustments

SQL objects (VLPs, views, functions, data sources for reports/forms) may need manual adjustment – this is a domain-specific question in each case.

Related pages

Mandator

Concept.

[Open](#)

Backup

Backup.

[Open](#)