

Virtual business objects

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

🔧 Virtual business objects

Business objects based on a database view – required columns, ID factory, creation via the wizard and write access via UPDATE RULE.

HOW-TO

APPLICATION DEVELOPER

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [What is a virtual BO?](#)
- [Requirements for the view](#)
- [ID factory \(PostgreSQL example\)](#)
- [Create a virtual BO](#)
- [Change a virtual BO](#)
- [Related pages](#)

What is a virtual BO?

A **virtual business object** is not based on a table but on a **database view**. A view is therefore a prerequisite (see [database objects](#)).

Requirements for the view

The view must contain at least these columns (exact names!):

```
intid,           -- Datensatzidentifizierer
strcreated,      -- Ersteller als String
datcreated,      -- Erstellungsdatum
strchanged,      -- Änderer als String
datchanged,      -- Änderungsdatum
intversion       -- Datensatzversion
```

- intid must be **unique** when referenced.
- Reference columns start with intid_str...
- Nuclos determines data types automatically; for wrong types an explicit cast in the view helps (e.g. `db_function(intid)::numeric(9,2)`).



Naming rules

Column names: **lowercase**, no special characters, **max. 30 characters**. Names in the data source and BO are case-sensitive and must match.
Good: datum, datum_list – bad: Datum, "Datum (Ist)".

ID factory (PostgreSQL example)

To write via a virtual BO an ID factory (sequence function) is required:

```
CREATE OR REPLACE FUNCTION Y25N_IDFACTORY()
  RETURNS numeric
  LANGUAGE 'plpgsql'  COST 100
  VOLATILE
AS $BODY$DECLARE
  id numeric;
BEGIN
  SELECT NEXTVAL('idfactory') INTO id;  RETURN id;
END;$BODY$;
```

Create a virtual BO

Create a new BO in the [business object wizard](#) and in step 2 via *Show advanced properties*:

- **Virtual business object:** select the prepared view (if missing it may be invalid).
- **Id factory for VE** (optional): DB function if writing through the BO – the BO then becomes editable automatically.
- **Table name:** leave empty for virtual BOs.

[FG4_VIEW_DRUCKERTYP - PG](#)

[Nudos Businessobjekt Drucker2](#)

Übersicht

1. Businessobjekte erstellen

2. Allgemeine Eigenschaften

3. Definition von Attributen

4. Attribute bearbeiten

5. Aktionen definieren

6. Menü konfigurieren

7. Beschriftung und Baumdarstellung

8. Rechte und Sperrfunktion verwalten

9. Übersetzen

10. Darstellung konfigurieren

Allgemeine Eigenschaften
 Bitte legen Sie hier allgemeine Eigenschaften fest

Beschriftung und Fenstertitel: Druckertypen

Pfad im Menü:

Benutzerdefiniertes Icon:

Tastenkürzel:

Suchmaske einfordern: ☒

Suchmaske direkt öffnen: ☐

Ist das Businessobjekt änderbar: ☐

gesplittete Anzeige: ☐

Historie: ☐

Statusmodell: ☐

Kürzel für Identifier:

Zeige Relationen im Baum: ☐

Zeige Gruppen im Baum: ☐

Mandantenabhängigkeit:

Mandanten Erstbefüllung mit:

Kommentar:

Virtuelle Businessobjekt: FG4_View_DruckerTyp

Id Factory für VE:

Thin Businessobjekt: ☒

Proxy Businessobjekt: ☐

Proxy Interface:

Tabellename: FG4_

Interner Businessobjekt Name: Druckertypen

Objektgenerator für Klonen:

Icon:

◀ Zurück

Weiter ▶

Erstig

Abbrechen

Advanced properties of the virtual BO.

 Write access

To create/delete records via Nuclos an additional **UPDATE RULE** is needed that performs the update on the underlying table(s).

Change a virtual BO

First adjust and save the view under [database objects](#), then run the [business object](#) wizard again.

 Use case

Virtual BOs are also suitable for **fixed master data** as part of a nuclet (e.g. Mr/Ms, Yes/No, compass directions).

Related pages

Business object

Data model.

Open

Database objects

Views.

Open

