

1 DATEV: Overview

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

1 DATEV: Overview

Overview of the DATEV Nuclet: export of CSV files for further processing in DATEV (accounting).

CONCEPT

INTEGRATORS

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [Related pages](#)



Machine-translated – under review

This page was machine-translated from German as a first draft and is currently under review. The authoritative source remains the **German original** (see the language switch above).

1.1 Short description

The DATEV Nuclet offers the ability to run CSV exports for further processing in DATEV.

The Nuclet provides three kinds of DATEV exports (customer exports, invoice exports and turnover exports), but it can easily be adapted for further export kinds.

You can find more about the feature scope and the implemented processes of this Nuclet in section ["2 DATEV: Feature scope"](#).



Notes on the DATEV format, see

Topic	Comment	URL
Getting started	Anchor page on the DATEV Developer Portal	https://developer.datev.de/de/file-format/details/datev-format/getting-started
Overview	Overview of the various DATEV formats	https://developer.datev.de/de/file-format/details/datev-format/format-description
Booking batch	DATEV format for turnover	https://developer.datev.de/de/file-format/details/datev-format/format-description/booking-batch
Debtors /creditors	DATEV format for customers and suppliers	https://developer.datev.de/de/file-format/details/datev-format/format-description/debitorskreditors

1.2 Nuclet components

As part of the .nuclet file, the DATEV Nuclet comprises

- seven business objects (for DATEV exports and associated master data, incl. the DATEV chart of accounts),
- three actions (for the individual export types)
- nine layouts,
- one state model (for the DATEV exports),
- one attribute group,
- six report data sources,
- three dynamic business objects,
- various Java rules (distributed across packages),
- eight Nuclet parameters,
- three job controls (to control the import jobs),
- two structure definitions (for importing master data) and
- one Nuclet dependency.

In addition, the following components are supplied in the ZIP file:

- two CSV files for object imports.

Type	Name, English	Name, German	Short description
Business object	DATEV Account	DATEV account	represents an account in the DATEV chart of accounts
	DATEV Account Class	DATEV account class	represents a class of the DATEV chart of accounts
	DATEV Account Type	DATEV account type	represents an account type of the DATEV chart of accounts
	DATEV Export	DATEV export	controls and represents a single DATEV export
	DATEV Export Type	DATEV export type	e.g. "customer export", "invoice export", "turnover export"
	DATEV File Encoding	DATEV character set	file character set, e.g. "ISO-8859-1", "UTF-8"
	DATEV Information	DATEV information	Information log for DATEV exports
Action	DATEV Client Billing Export		for invoice exports
	DATEV Client Export		for customer exports
	DATEV Turnover Export		for turnover exports
Layout	DATEV Account		layout for business object "DATEV Account"
	DATEV Account Class		layout for business object "DATEV Account Class"
	DATEV Account Type		layout for the business object "DATEV Account Type"
	DATEV Client Billing Export		layout for business object "DATEV Export", export type: customer invoices
	DATEV Client Export		layout for business object "DATEV Export", export type: customers
	DATEV Export		general layout for business object "DATEV Export"
	DATEV Export Type		layout for business object "DATEV Export Type"
	DATEV File Encoding		layout for business object "DATEV File Encoding"
	DATEV Turnover Export		layout for business object "DATEV Export", export type: turnover
State model	DATEV Export		state model for the business object "DATEV Export"
Attribute group	DATEV Export		
Report & form	DATEV Client Billing Export		template for invoice exports
	DATEV Client Billing Export (Header)		template for header rows in the invoice export
	DATEV Client Export		template for customer exports
	DATEV Client Export (Header)		template for header rows in the customer export
	DATEV Turnover Export		template for turnover exports
	DATEV Turnover Export (Header)		template for header rows in the turnover export
Dynamic business objects	DATEV Client Billing References		all references relating to an invoice export
	DATEV Client References		all references relating to a customer export
	DATEV Turnover References		all references relating to a turnover export
Java package	org.nuclet.datev.db		classes for database accesses, i.e. for running the report data sources
	org.nuclet.datev.job		Java rules for controlling jobs ("job control")
	org.nuclet.datev.logic		Business logic
	org.nuclet.datev.rule		control of insert/update/delete events
	org.nuclet.datev.wrapper		wrapper classes as a Nuclet interface
Parameter	DATEV File Directory		defines the output directory of the CSV export files
	DATEV File Encoding		determines the character set used in the CSV export files
	DATEV Header Row		defines whether a header row appears in the export files
	DATEV Field Separator		determines the default field separator
	DATEV Text Separator		determines the default text separator
	DATEV Client Billing Export Prefix		defines the name stem for DATEV invoice exports
	DATEV Client Export Prefix		defines the name stem for DATEV customer exports
	DATEV Turnover Export Prefix		defines the name stem for DATEV turnover exports
Job control	DATEV Client Billing Export		deadline job for the automated DATEV invoice export

	DATEV Client Export		deadline job for the automated DATEV customer export
	DATEV Turnover Export		deadline job for the automated DATEV turnover export
Structure definition	DATEV Export Type		import structure for business object "DATEV Export Type"
	DATEV File Encoding		import structure for business object "DATEV File Encoding"
Nuclet dependency	org.nuclet.Common		general helper functionality
Object imports	DATEV_Export_Type.csv		master data records for business object "DATEV Export Type"
	DATEV_File_Encoding.csv		master data records for business object "DATEV File Encoding"

Table 1: Nuclet components



Caution: Please note that, due to the Nuclet dependency on [org.nuclet.Common](#) further components are imported into the system when the Nuclet is integrated, which are not listed individually here. Details on the components of the [Common Nuclet](#) can be found in the documentation of this Nuclet.

Java package structure

The Java rules are divided into three packages:

- rules for controlling system jobs (org.nuclet.datev.job)
- rules for controlling insert, update, delete events and state changes
- rules for the business logic (org.nuclet.datev.logic)
- supporting rules for database accesses via report data sources (org.nuclet.datev.db)
- wrapper classes as a Nuclet interface for business objects (org.nuclet.datev.wrapper)

For the business logic, an abstract Java class (AbstractDATEVLogic) provides the base functionality via the declared and (mostly) defined methods. The export processes are defined along the pre-declared methods of this abstract class and require no further adjustment for the normal use case. Additions are permitted in concrete implementations of this class (e.g. in the supplied DATEVLogic class or your own implementation).

The classes in the packages for database accesses (org.nuclet.datev.db) and for job control (org.nuclet.datev.job) are implemented in such a way that they normally require no adjustment at all.

Java package	abstract classes	concrete implementations	other classes
org.nuclet.datev.db		▪ DATEVDataSourceExecutor ▪ DATEVResultList	
org.nuclet.datev.job	AbstractCreateDATEVExport	▪ CreateDATEVClientBillingExport ▪ CreateDATEVClientExport ▪ CreateDATEVTurnoverExport	
org.nuclet.datev.logic	AbstractDATEVLogic	▪ DATEVLogic	▪ ExportType
org.nuclet.datev.rule		▪ ExecuteDATEVExport ▪ ExecuteDATEVExportAndProcessReferences ▪ InitialiseDATEVExport ▪ ValidateDATEVExport	
org.nuclet.datev.wrapper	AbstractReferenceWrapper	▪ ReferenceWrapper	

Table 2: Java package structure

Application-specific adjustments should – as mentioned before – only be implemented, either

- in the concrete, already existing classes (see table) or
- by new, own concrete implementations

The Java source code is marked with @replace! tags at all the places where application-specific behaviour can be added. More on this in [section 4](#).

Related pages



Interfaces



Nuclet: DATEV (available in the trade Nuclet)

DATEV Nuclet data sheet

[Open](#)

All integrations

[Open](#)