

1 CAMT: Overview

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

💡 1 CAMT: Overview

Overview of the CAMT Nuclet: import and processing of CAMT.053/CAMT.054 bank statements (Cash Management).

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).

Short description

The CAMT Nuclet offers the option

- to automatically read electronic bank statements (daily statements, CAMT.053 format) or collective booking information (collective booking file, CAMT.054 format)
- to transfer the imported data into a structure of Nuclos business objects
- to automatically assign the bank transactions, based on the remittance information, to other business objects ("references", which depending on the application purpose are e.g. your invoices)

For the objects assigned to the bank transactions (invoices, etc.), a state change is performed automatically on incoming payment (e.g. from "open" to "paid").

More about the feature scope and implemented processes of this Nuclet can be found in [section 2](#).

Nuclet components

Within the .nuclet file, the CAMT Nuclet comprises

- ten business objects (for bank statements, bank transactions and various master data),
- nine layouts,
- two state models (for CAMT imports and bank transactions),
- five attribute groups,
- various Java rules (distributed across packages),
- four Nuclet parameters,
- two job controls (to control the import jobs),
- one value list provider
- seven structure definitions (for importing master data) and
- one Nuclet dependency.

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

- the JAR file org.nuclet.camt.jaxb-2.0.0.jar as a Nuclos server extension and
- seven CSV files for object imports.

Type	Name, English	Name, German	Short description
Business object	Account Balance	Account balance	
	Balance Type	Balance type	
	Bank Statement	Bank statement	
	Bank Transaction	Bank transaction	
	Bank Transaction Ref	Referenz-2-Bankumsatz	

	Bank Transaction Type	Bank transaction type	
	CAMT Message Type	CAMT message type	CAMT.053 or CAMT.054
	Credit/Debit Indicator	debit/credit mark	
	Entry Status	Entry status	
	SEPA Return Reason Code	SEPA return reason code	
	SWIFT Transaction Code	SWIFT transaction code	
Layout	Balance Type		layout for the business object "Balance Type"
	Bank Statement		layout for the business object "Bank Statement"
	Bank Transaction		layout for the business object "Bank Transaction"
	Bank Transaction Type		layout for business object "Bank Transaction Type"
	CAMT Import		
	CAMT Import (CAMT.053)		
	Credit/Debit Indicator		layout for business object "Credit/Debit Indicator"
	Entry Status		layout for business object "Entry Status"
	SEPA Return Reason Code		layout for business object "SEPA Return Reason Code"
State model	CAMT Bank Transaction		state model for the business object "Bank Transaction"
Attribute group	CAMT Bank Statement Data		Bank statement data
	CAMT Bank Transaction Data		Bank transaction data
	CAMT Bank Transaction References		references assigned to the bank transactions
	CAMT Import Data		information to be set by the user in CAMT imports
	CAMT System Data		information entered by the application in CAMT imports
Java package	org.nuclet.camt.facade		classes for database accesses
	org.nuclet.camt.job		Java rules for controlling jobs ("job control")
	org.nuclet.camt.logic		Business logic
	org.nuclet.camt.parser		CAMT parser functionality
	org.nuclet.camt.rule		control of insert/update/delete events
	org.nuclet.camt.statemodel		definition of state changes (as a Nuclet interface)
	org.nuclet.camt.wrapper		wrapper classes (as a Nuclet interface)
Parameter	CAMT File Directory		defines the input directory of the CAMT files
	CAMT File Encoding		determines the character set used in the CAMT files
	CAMT File Extension		defines the file extension of the CAMT files
	CAMT File Format		used to pre-populate the message type in CAMT imports
Job control	CAMT.053 Import		deadline job for the automated import of CAMT.053 files
	CAMT.054 Import		deadline job for the automated import of CAMT.054 files
Value List Provider	CAMT Message Type (default)		Pre-population of the message type in CAMT imports
Structure definition	CAMT Balance Type		import structure for business object "Balance Type"
	CAMT Bank Transaction Type		import structure for business object "Bank Transaction Type"
	CAMT Credit/Debit Indicator		import structure for business object "Credit/Debit Indicator"
	CAMT Entry Status		import structure for business object "Entry Status"
	CAMT Message Type		import structure for business object "CAMT Message Type"
	CAMT SWIFT Transaction Code		import structure for business object "SWIFT Transaction Code"
	SEPA Return Reason Code		import structure for business object "SEPA Return Reason Code"
Nuclet dependency	org.nuclet.Common		general helper functionality
Object imports	Balance_Type.csv		master data records for business object "Balance Type"
	Bank_Transaction_Type.csv		master data records for business object "Bank Transaction Type"
	CAMT_Message_Type.csv		master data records for business object "CAMT Message Type"
	Credit_Debit_Indicator.csv		master data records for business object "Credit/Debit Indicator"

	Entry_Status.csv		master data records for business object "Entry Status"
	SWIFT_Transaction_Code.csv		master data records for business object "SWIFT Transaction Code"
Example files	Kontoauszug_001.xml		CAMT.054 import example
	Kontoauszug_002.xml		CAMT.054 import example

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 seven packages:

- rules for screen-controlled events (org.nuclet.camt.rule)
- rules for controlling system jobs (org.nuclet.camt.job)
- rules for the business logic (org.nuclet.camt.logic)
- rules for reading and processing CAMT.053 and CAMT.054 files (org.nuclet.camt.parser)
- rules for database accesses (org.nuclet.camt.facade)
- wrapper classes as a Nuclet interface (org.nuclet.camt.wrapper)
- state change definitions as a Nuclet interface (org.nuclet.camt.statemodel)

In two of these four Java packages, abstract Java classes provide the basic functionality via the declared and (mostly) defined methods. The import processes are defined along the pre-declared methods of these abstract classes and require no further adjustment for the normal use case.

In the packages for database accesses (org.nuclet.camt.facade) and wrapper classes (org.nuclet.camt.wrapper), abstract classes are also defined whose concrete forms must be implemented in an application-specific way. Concretely, this means at this point establishing interfaces to the target Nuclet and defining application-specific behaviour (namely for currency, conditions of payment and references).

Java package	abstract classes	concrete implementations
org.nuclet.camt.job	AbstractCAMT053Importer	CAMT053Importer
	AbstractCAMT054Importer	CAMT054Importer
org.nuclet.camt.logic	AbstractCAMTLogic	CAMTLogic
org.nuclet.camt.parser	-	-
org.nuclet.camt.rule	-	-
org.nuclet.camt.facade	• AbstractCurrencyFacade • AbstractReferenceFacade	• CurrencyFacade • ReferenceFacade
org.nuclet.camt.wrapper	• AbstractConditionsOfPaymentWrapper • AbstractCurrencyWrapper • AbstractReferenceWrapper	• ConditionsOfPaymentWrapper • CurrencyWrapper • ReferenceWrapper
org.nuclet.camt.statemodel	-	

Table 2: Java package structure

Application-specific adjustments should only be implemented, either

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

Related pages



Nuclet: CAMT
(available in the trade Nuclet)

CAMT Nuclet data sheet



Interfaces

All integrations

[Open](#)

Open