

1 MT940: Overview

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

💡 1 MT940: Overview

Overview of the MT940 Nuclet: import and processing of bank statements in MT940 format.

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 MT940 Nuclet offers the option

- to automatically read electronic bank statements that are in MT940 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 MT940 Nuclet comprises

- seven business objects (for bank statements, bank transactions and various master data),
- six layouts,
- one state model (for bank transactions),
- two attribute groups,
- various Java rules (distributed across packages),
- three Nuclet parameters,
- one job control (to control the import job),
- three structure definitions (for importing master data) and
- one Nuclet dependency.

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

- three CSV files for object imports and
- two MT940 example files with bank statements.

Type	Name, English	Name, German	Short description
Business object	Bank Statement	Bank statement	
	Bank Transaction	Bank transaction	
	Bank Transaction Type	Bank transaction type	
	Banking Business Line	Banking business line	
	Debit/Credit Mark	debit/credit mark	
	MT940 Reference	MT940 reference	
	Bank Transaction Ref	Referenz-2-Bankumsatz	

Layout	Bank Statement		layout for business object "Bank Statement"
	Bank Transaction		layout for the business object "Bank Transaction"
	Bank Transaction Type		layout for business object "Bank Transaction Type"
	Banking Business Line		layout for business object "Banking Business Line"
	Debit/Credit Mark		layout for business object "Debit/Credit Mark"
State model	MT940 Bank Transaction		state model for the business object "Bank Transaction"
Attribute group	Bank Transaction Data		bank transaction and bank statement data
	Bank Transaction References		references assigned to the bank transactions
Java package	org.nuclet.mt940.facade		classes for database accesses
	org.nuclet.mt940.job		Java rules for controlling jobs ("job control")
	org.nuclet.mt940.logic		Business logic
	org.nuclet.mt940.parser		MT940 parser functionality
	org.nuclet.mt940.rule		control of insert/update/delete events
	org.nuclet.mt940.statemodel		definition of state changes (as a Nuclet interface)
	org.nuclet.mt940.wrapper		wrapper classes (as a Nuclet interface)
Parameter	MT940 File Directory		defines the input directory of the MT940 files
	MT940 File Encoding		determines the character set used in the MT940 files
	MT940 File Extension		defines the file extension of the MT940 files
Job control	MT940 Import		deadline job for the automated MT940 import
Structure definition	MT940 Bank Transaction Type		import structure for business object "Bank Transaction Type"
	MT940 Banking Business Line		import structure for business object "Banking Business Line"
	MT940 Debit/Credit Mark		import structure for business object "Debit/Credit Mark"
Nuclet dependency	org.nuclet.Common		general helper functionality
Object imports	Banking_Business_Line.csv		master data records for business object "Banking Business Line"
	Bank_Transaction_Type.csv		master data records for business object "Bank Transaction Type"
	Debit_Credit_Mark.csv		master data records for business object "Debit/Credit Mark"
Example files	Kontoauszug_001.sta		MT940 import example
	Kontoauszug_002.sta		MT940 import example

Table 1: Nuclet components



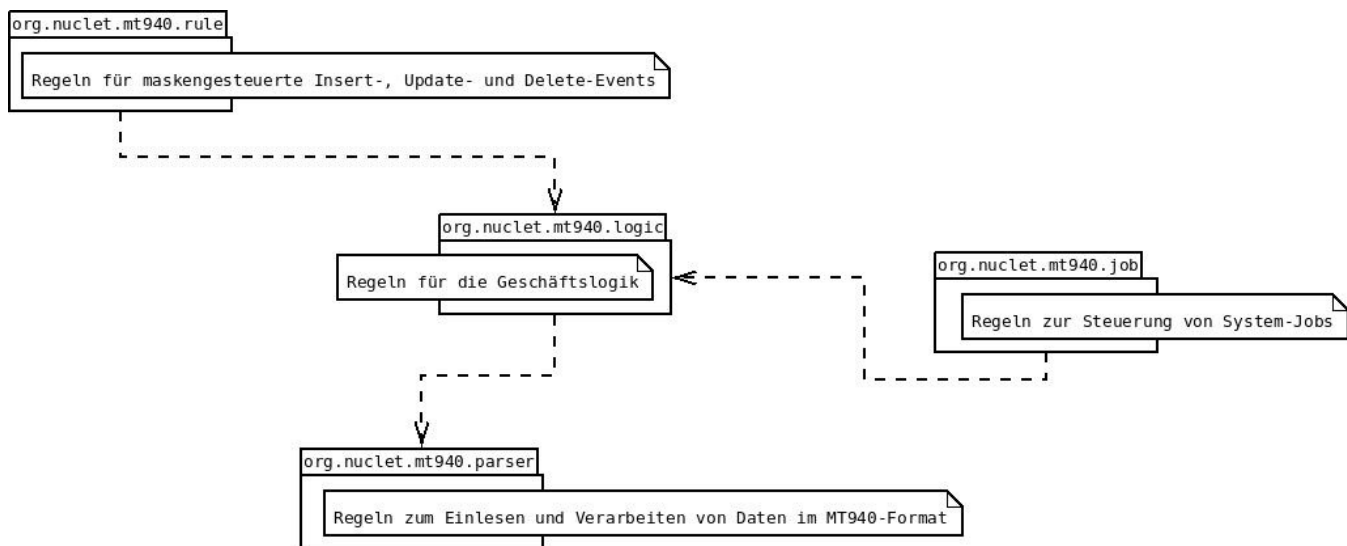
Caution: Please note that, due to the Nuclet dependency on [org.nuclet.Common](#) further components are imported into the system when the Nuclet is integrated, which are not listed individually here. Details on the components of the [Common Nuclet](#) can be found in the documentation of this Nuclet.

Java package structure

The Java rules are divided into seven packages:

- rules for screen-controlled events (org.nuclet.mt940.rule)
- rules for controlling system jobs (org.nuclet.mt940.job)
- rules for the business logic (org.nuclet.mt940.logic)
- rules for reading and processing MT940 files (org.nuclet.mt940.parser)
- rules for database accesses (org.nuclet.mt940.facade)
- wrapper classes as a Nuclet interface (org.nuclet.mt940.wrapper)
- state change definitions as a Nuclet interface (org.nuclet.mt940.statemodel)

The essential dependencies of the packages are illustrated in the following figure. The figure is limited to those main packages that serve the process control.



In three of these four Java packages, an abstract Java class provides 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.mt940.facade) and wrapper classes (org.nuclet.mt940.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.mt940.job	AbstractMT940Importer	MT940Importer
org.nuclet.mt940.logic	AbstractMT940Logic	MT940Logic
org.nuclet.mt940.parser	AbstractMT940Parser	- MT940Parser - MT940SparkasseParser
org.nuclet.mt940.rule	-	-
org.nuclet.mt940.facade	- AbstractCurrencyFacade - AbstractReferenceFacade	- CurrencyFacade - ReferenceFacade
org.nuclet.mt940.wrapper	- AbstractConditionsOfPaymentWrapper - AbstractCurrencyWrapper - AbstractReferenceWrapper	- ConditionsOfPaymentWrapper - CurrencyWrapper - ReferenceWrapper
org.nuclet.mt940.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

The package org.nuclet.mt940.parser contains the class MT940SparkasseParser, which already provides dedicated behaviour for MT940 files supplied by savings banks (Sparkassen). The second concrete implementation within this package is the class MT940Parser, which beyond the basic functionality of the AbstractMT940Parser does not bring any additional behaviour. For MT940 files created by other banks, the class MT940Parser (or your own concrete implementation of the AbstractMT940Parser) filled with bank-specific behaviour if necessary.

The MT940 Nuclet is currently configured such that the savings-bank-specific behaviour is used. The parser is initialized in the constructor of the class org.nuclet.mt940.job.MT940Importer and could be redirected there at any time to a different behaviour, i.e. your own implementation.

org.nuclet.mt940.job.MT940Importer

```
public MT940Importer(final JobContext context,
    final String strMT940Directory,
    final String strMT940ReferenceType)
{
    super(context, strMT940Directory, strMT940ReferenceType);

    // @replace!
    //
    // Replace with your own logic and/or your own parser here, if you
    // need more specific behaviour.
    //
    this.logic = new MT940Logic(context, strMT940ReferenceType);

    this.parser = new MT940SparkasseParser(context);
}
```

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

Related pages

Nuclet: MT940 (not yet available)

MT940 Nuclet data sheet

[Open](#)

Interfaces

All integrations

[Open](#)