

4.7.1 MT940: Nuclet interfaces

Language:
 Deutsch
 ·
 English

4.7.1 MT940: Nuclet interfaces

Nuclet interfaces – part of the MT940 Nuclet documentation (import and processing of bank statements in MT940 format).

REFERENCE

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).

The aim of this step is to connect the business object actually used in the target Nuclet to the Java code of the MT940 Nuclet. To this end, the approach of using Nuclet-independent wrapper classes is followed – instead of the business object classes generated for the business objects.

For the technical background, please also read the article "Interfaces in Java rules".



This integration step can only be carried out once the [prerequisites from section 3](#) are met – i.e. as soon as corresponding business objects for currency, references and, if applicable, conditions of payment exist.

When connecting the MT940 Nuclet, wrapper classes must be adjusted for currency objects, references and conditions of payment (optional).

Wrapper class	Function	Java package	Methods/constructors to adjust
CurrencyWrapper	connection to the currency business object used	org.nuclet.mt940.wrapper	- CurrencyWrapper() - getIso4217Code()
ConditionsOfPaymentWrapper	connection to the conditions of payment used	org.nuclet.mt940.wrapper	- ConditionsOfPaymentWrapper() - getCashDiscount() - getDashDiscountPeriod() - getExtendedCashDiscount() - getExtendedCashDiscountPeriod() - getCashDiscountExGratiaDays()
ReferenceWrapper	connection to the reference objects used	org.nuclet.mt940.wrapper	- ReferenceWrapper() - getAcceptFirstIncomingPayment() - getBankTransactionRef() - getConditionsOfPaymentId() - getDateOfInvoice() - getTotalAmountGross() - setPaymentDate()

Table 4.7.1: Overview of Nuclet interfaces

4.7.1.1 CurrencyWrapper

The CurrencyWrapper class serves as the Nuclet interface to the actually used currency business object.

The constructor of the class and the getIso4217Code() method must be filled with application-specific behaviour. The latter method should return the ISO 4217 currency code (see [Wikipedia](#)) of the currency object.

Method	Function	Adjustment optional?
CurrencyWrapper	Assignment of the actually used business object	no

getIso4217Code()	returns the three-letter ISO 4217 code of the currency (e.g. "EUR", "USD", "GBP")	no
------------------	---	----

Table 4.7.1.1: Adjustments in CurrencyWrapper

Examples are given in comment blocks; these examples must therefore be adapted, during integration, to the actually used currency business object (or its BusinessObject class).

org.nuclet.mt940.wrapper.CurrencyWrapper

```
package org.nuclet.mt940.wrapper;

import org.nuclos.api.businessobject.BusinessObject;

// @replace!
//
// mit eigenem Code zu ersetzen, Beispiel:
//
// import org.nuclet.currency.Currency;

/**
 * Konkrete Wrapper-Klasse für Währungsobjekte
 *
 * @version 1.0
 * @date 07.08.2013
 * @nuclet org.nuclet.MT940
 * @nucletversion 1.4.0
 * @sincenucletversion 1.1.0
 * @since 07.08.2013
 *
 * @author frank.lehmann@nuclos.de
 */
public class CurrencyWrapper extends AbstractCurrencyWrapper
{
    public CurrencyWrapper(final BusinessObject currency)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (currency instanceof Currency) {
        //     this.businessObject = currency;
        // }
    }

    /**
     * Liefert den ISO-4217-Code des Währungsobjektes.
     * @see https://de.wikipedia.org/wiki/ISO\_4217
     * @see https://en.wikipedia.org/wiki/ISO\_4217
     */
    public String getIso4217Code()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((Currency)this.businessObject).getIso4217Code();

        return null;
    }
}
```

4.7.1.2 ConditionsOfPaymentWrapper

The ConditionsOfPaymentWrapper class serves as the Nuclet interface for the actually used conditions of payment.

The adjustments are optional: if no conditions of payment and cash discount conditions are used, this section does not have to be considered.

If cash discount conditions are used, the getter methods for cash discount rate and cash discount period must be adjusted.

Method	Function	Adjustment optional?
ConditionsOfPaymentWrapper	Assignment of the actually used business object	yes
getCashDiscount()	returns the cash discount rate	yes
getCashDiscountPeriod()	returns the cash discount period (in days)	yes
getExtendedCashDiscount()	returns the cash discount rate for the extended cash discount period	yes
getExtendedCashDiscountPeriod()	returns the extended cash discount period (in days)	yes
getCashDiscountExGratiaDays()	returns the grace period (in days)	yes

Table 4.7.1.2: Adjustments in ConditionsOfPaymentWrapper

Examples are given in comment blocks; these examples must therefore be adapted, during integration, to the actually used business object (or its BusinessObject class).

org.nuclet.mt940.wrapper.ConditionsOfPaymentWrapper

```
package org.nuclet.mt940.wrapper;

import java.math.BigDecimal;
import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Stateful;

// @replace!
//
// mit eigenem Code zu ersetzen, Beispiel:
//
// import org.nuclet.businessobject.ConditionsOfPayment;

/**
 * Abstrakte Wrapper-Klasse für Zahlungsbedingungen
 *
 * @version 1.1
 * @date 20.02.2013
 * @nuclet org.nuclet.MT940
 * @nucletversion 1.4.0
 * @sincenucletversion 1.2.0
 * @since 20.09.2013
 *
 * @author frank.lehmann@nuclos.de
 */
public class ConditionsOfPaymentWrapper extends AbstractConditionsOfPaymentWrapper
{
    public ConditionsOfPaymentWrapper(final BusinessObject boConditionsOfPayment)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (boConditionsOfPayment instanceof ConditionsOfPayment) {
        //     this.businessObject = boConditionsOfPayment;
        // }
    }

    /**
     * Liefert den Skontosatz.
     */
}
```

```

public BigDecimal getCashDiscount()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((ConditionsOfPayment)this.businessObject).getCashDiscount();

    return null;
}

/**
 * Liefert den erweiterten Skontosatz.
 */
public BigDecimal getExtendedCashDiscount()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((ConditionsOfPayment)this.businessObject).getExtendedCashDiscount();

    return null;
}

/**
 * Liefert die Skontofrist (in Tagen).
 */
public Integer getCashDiscountPeriod()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((ConditionsOfPayment)this.businessObject).getCashDiscountPeriod();

    return null;
}

/**
 * Liefert die erweiterte Skontofrist (in Tagen).
 */
public Integer getExtendedCashDiscountPeriod()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((ConditionsOfPayment)this.businessObject).getExtendedCashDiscountPeriod();

    return null;
}

/**
 * Liefert die Kulanz (in Tagen).
 */
public Integer getCashDiscountExGratiaDays()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((ConditionsOfPayment)this.businessObject).getCashDiscountExGratiaDays();

    return null;
}
}

```

4.7.1.3 ReferenceWrapper

The ReferenceWrapper class serves as the Nuclet interface to the actually used reference business object.

The constructor of the class and the following methods must be adjusted:

Method	Function	Adjustment optional?
ReferenceWrapper()	Assignment of the actually used business object	no
getAcceptFirstIncomingPayment()	Returns the information on whether the first incoming payment should trigger the state change in the reference object, i.e. the status transition towards "Paid", "Settled", etc.	yes
getBankTransactionRef()	Returns all assigned bank transactions, indirectly via the intermediate business object "BankTransactionRef", i.e. all entries of the corresponding sub-form should be returned.	no
getConditionsOfPaymentId()	Returns the database ID of the actually referenced conditions of payment	yes
getDateOfInvoice()	Returns the payment date.	yes
getTotalAmountGross()	Returns the gross total amount of the reference object, i.e. the amount to be settled via incoming payment.	no
setPaymentDate()	Sets the "Payment Date" (payment date, paid on, etc.) in the reference object.	yes

Table 4.7.1.3: Adjustments in ReferenceWrapper

Examples are given in comment blocks; these examples must therefore be adapted, during integration, to the actually used business object (or its BusinessObject class).

org.nuclet.mt940.wrapper.ReferenceWrapper

```
package org.nuclet.mt940.wrapper;

import java.math.BigDecimal;
import java.util.Date;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Stateful;
import org.nuclos.api.provider.QueryProvider;

import org.nuclet.mt940.BankTransactionRef;
import org.nuclet.mt940.wrapper.ConditionsOfPaymentWrapper;
import org.nuclet.mt940.wrapper.AbstractConditionsOfPaymentWrapper;

// @replace!
//
// mit eigenem Code zu ersetzen, Beispiel:
//
// import org.nuclet.businessnuclet.ClientBilling;
// import org.nuclet.businessnuclet.ConditionsOfPayment;

/**
 * Konkrete Wrapper-Klasse für Referenzobjekte (Rechnungen, o.ä.)
 *
 * @version 1.2
 * @date 07.07.2014
 * @nuclet org.nuclet.MT940
 * @nucletversion 1.4.1
 * @sincenucletversion 1.2.0
 * @since 20.09.2013
 *
 * @author frank.lehmann@nuclos.de
 */
public class ReferenceWrapper extends AbstractReferenceWrapper
```

```

{
    public ReferenceWrapper(final Stateful reference)
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // if (reference instanceof ClientBilling) {
        //     this.businessObject = reference;
        // }
    }

    /**
     * Liefert die Referenz-Nummer, anhand derer der Datensatz eindeutig
     * zu identifizieren ist
     */
    public String getReferenceNumber()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBilling)this.businessObject).getBillingNumber();

        return null;
    }

    /**
     * Liefert die Datenbank-ID der referenzierten Zahlungsbedingungen
     */
    public Long getConditionsOfPaymentId()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBilling)this.businessObject).getConditionsOfPaymentId();

        return null;
    }

    /**
     * Liefert die Information darüber, ob die erste eingehende Zahlung den Statuswechsel
     * im Referenzobjekt auslösen soll, d.h. den Statusübergang in Richtung "Beahlt",
     * "Beglichen", o.ä.
     */
    public Boolean getAcceptFirstIncomingPayment()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // return ((ClientBilling)this.businessObject).getAcceptFirstIncomingPayment();

        return Boolean.FALSE;
    }

    /**
     * Liefert alle zugeordneten Bankumsätze, indirekt über die Zwischenentität "BankTransactionRef",
     * d.h. es sollten alle Einträge des entsprechenden Unterformulars zurückgegeben werden.
     */
    public List<BankTransactionRef> getBankTransactionRef()
    {
        // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
        //
        // Beispiel:
        //
        // ((ClientBilling)this.businessObject).getBankTransactionRef();
    }
}

```

```

return null;
}

/**
 * Liefert die tatsächlich genutzten Zahlungsbedingungen als "gewrapptes" Objekt.
 *
 */
public AbstractConditionsOfPaymentWrapper getConditionsOfPayment()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // final Long lngConditionsOfPaymentId = ((ClientBilling)this.businessObject).getConditionsOfPaymentId();
    // final ConditionsOfPayment boConditionsOfPayment = QueryProvider.getById(ConditionsOfPayment.class,
    lngConditionsOfPaymentId);
    //
    // return new ConditionsOfPaymentWrapper(boConditionsOfPayment);

return null;
}

/**
 * Liefert das Zahlungsdatum.
 *
 */
public Date getDateOfInvoice()
{
    // @replace Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((ClientBilling)this.businessObject).getBillingDate();

return null;
}

/**
 * Liefert den Bruttogesamtbetrag des Referenzobjektes, d.h. den über eingehende
 * Zahlung auszugleichenden Betrag.
 *
 */
public BigDecimal getTotalAmountGross()
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // return ((ClientBilling)this.businessObject).getTotalAmountGross();

return BigDecimal.ZERO;
}

/**
 * Setzt das "Payment Date" (Zahlungsdatum, Beahlt am, o.ä.) im Referenzobjekt.
 *
 * @param datPaymentDate
 *
 */
public void setPaymentDate(final Date datPaymentDate)
{
    // @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
    //
    // Beispiel:
    //
    // ((ClientBilling)this.businessObject).setPaymentDate(datPaymentDate);
}

}

```

Related pages

Nuclet: MT940 (not yet available)

MT940 Nuclet data sheet

[Open](#)

Interfaces

All integrations

[Open](#)