

4.7.1 MT940: Nuclet-Schnittstellen

 Sprache: **Deutsch** · [English](#)

4.7.1 MT940: Nuclet-Schnittstellen

Ziel dieses Schrittes ist, das tatsächlich im Zielnuclet verwendeten Businessobjekt an den Java-Code des MT940-Nuclets anzubinden.

REFERENZ

INTEGRATOREN

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Verwandte Seiten](#)

Ziel dieses Schrittes ist, das tatsächlich im Zielnuclet verwendeten Businessobjekt an den Java-Code des MT940-Nuclets anzubinden. Dazu wird der Ansatz verfolgt, Nuclet-unabhängige Wrapper-Klassen zu verwenden – anstelle der zu den Businessobjekten generierten Businessobjekt-Klassen.

Zu den technischen Hintergründen lesen Sie dazu bitte auch den Artikel "Schnittstellen in Java-Regeln".

 Dieser Integrationsschritt kann erst durchgeführt werden, sobald die [Vorbedingungen aus Abschnitt 3](#) erfüllt sind – d.h. sobald entsprechende Businessobjekte für Währung, Referenzen und ggf. Zahlungsbedingungen existieren.

Bei der Anbindung des MT940-Nuclets sind Wrapper-Klassen für Währungsobjekte, Referenzen und Zahlungsbedingungen (optional) anzupassen.

Wrapper-Klasse	Funktion	Java-Package	Anzupassende Methoden/Konstruktoren
CurrencyWrapper	Anbindung an verwendete Währungsbusinessobjekt	org.nuclet.mt940.wrapper	- CurrencyWrapper() - getIso4217Code()
ConditionsOfPaymentWrapper	Anbindung an genutzte Zahlungsbedingungen	org.nuclet.mt940.wrapper	- ConditionsOfPaymentWrapper() - getCashDiscount() - getDashDiscountPeriod() - getExtendedCashDiscount() - getExtendedCashDiscountPeriod() - getCashDiscountExGratiaDays()
ReferenceWrapper	Anbindung an verwendete Referenzobjekte	org.nuclet.mt940.wrapper	- ReferenceWrapper() - getAcceptFirstIncomingPayment() - getBankTransactionRef() - getConditionsOfPaymentId() - getDateOfInvoice() - getTotalAmountGross() - setPaymentDate()

Tabelle 4.7.1: Übersicht Nuclet-Schnittstellen

4.7.1.1 CurrencyWrapper

Die Klasse CurrencyWrapper dient als Nuclet-Schnittstelle zum tatsächlich genutzten Währungsbusinessobjekt.

Der Konstruktor der Klasse und die Methode getIso4217Code() sind mit anwendungsspezifischem Verhalten zu befüllen. Letztere Methode sollte den ISO-4217-Währungscode (siehe dazu [Wikipedia](#)) des Währungsobjektes zurückliefern.

Methode	Funktion	Anpassung optional?
CurrencyWrapper	Zuordnung des tatsächlich genutzten Businessobjektes	nein
getIso4217Code()	liefert den dreistelligen ISO-4217-Code der Währung (z.B. "EUR", "USD", "GBP")	nein

Tabelle 4.7.1.1: Anpassungen in CurrencyWrapper

Beispiele dazu sind in Kommentarböcken angegeben; diese Beispiele sind bei der Integration also an die tatsächlich genutzte Währungsbusinessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

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

Die Klasse ConditionsOfPaymentWrapper dient als Nuclet-Schnittstelle für tatsächlich genutzte Zahlungsbedingungen.

Die Anpassungen sind optional: Werden keine Zahlungs- und Skontobedingungen verwendet, dann muss dieser Abschnitt nicht berücksichtigt werden.

Werden Skontobedingungen verwendet, dann sind die Getter-Methoden für Skontosatz und Skontofrist anzupassen.

Methode	Funktion	Anpassung optional?
ConditionsOfPaymentWrapper	Zuordnung des tatsächlich genutzten Businessobjektes	ja

getCashDiscount()	liefert den Skontosatz	ja
getCashDiscountPeriod()	liefert die Skontofrist (in Tagen)	ja
getExtendedCashDiscount()	liefert den Skontosatz bei erweiterter Skontofrist	ja
getExtendedCashDiscountPeriod()	liefert die erweiterte Skontofrist (in Tagen)	ja
getCashDiscountExGratiaDays()	liefert die Kulanz (in Tagen)	ja

Tabelle 4.7.1.2: Anpassungen in *ConditionsOfPaymentWrapper*

Beispiele dazu sind in Kommentarböcken angegeben; diese Beispiele sind bei der Integration also an das tatsächlich genutzte Businessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

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

Die Klasse ReferenceWrapper dient als Nuclet-Schnittstelle zum tatsächlich genutzten Referenz-Businessobjekt.

Anpassen sind der Konstruktor der Klasse sowie die folgenden Methoden:

Methode	Funktion	Anpassung optional?
ReferenceWrapper()	Zuordnung des tatsächlich genutzten Businessobjektes	nein
getAcceptFirstIncomingPayment()	Liefert die Information darüber, ob die erste eingehende Zahlung den Statuswechsel m Referenzobjekt auslösen soll, d.h. den Statusübergang in Richtung "Bezahlt", "Beglichen", o.ä.	ja
getBankTransactionRef()	Liefert alle zugeordneten Bankumsätze, indirekt über die Zwischenbusinessobjekt "BankTransactionRef", d.h. es sollten alle Einträge des entsprechenden Unterformulars zurückgegeben werden.	nein
getConditionsOfPaymentId()	Liefert die Datenbank-ID der tatsächlich referenzierten Zahlungsbedingungen	ja
getDateOfInvoice()	Liefert das Zahlungsdatum.	ja
getTotalAmountGross()	Liefert den Bruttogesamtbetrag des Referenzobjektes, d.h. den über eingehende Zahlung auszugleichenden Betrag.	nein
setPaymentDate()	Setzt das "Payment Date" (Zahlungsdatum, Bezahlt am, o.ä.) im Referenzobjekt.	ja

Tabelle 4.7.1.3: Anpassungen in ReferenceWrapper

Beispiele dazu sind in Kommentarböcken angegeben; diese Beispiele sind bei der Integration also an das tatsächlich genutzte Businessobjekt (bzw. deren BusinessObject-Klasse) anzupassen.

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 "Bezahlt",
 * "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);
}

}

```



Nuclet: MT940(noch nicht verfügbar)

MT940-Datenblatt

[Öffnen](#)



Schnittstellen

Alle Integrationen

[Öffnen](#)