

4.8.2 SEPA: Businessobjekt-Fassaden

 Sprache: **Deutsch** · [English](#)

4.8.2 SEPA: Businessobjekt-Fassaden

Das SEPA-Nuclet arbeitet mit Wrapper-Objekte anstelle der nur dem Zielnuclet bekannten BusinessObjects (siehe Abschnitt 4.8.1).

REFERENZ

INTEGRATOREN

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Verwandte Seiten](#)

Das SEPA-Nuclet arbeitet mit Wrapper-Objekte anstelle der nur dem Zielnuclet bekannten BusinessObjects ([siehe Abschnitt 4.8.1](#)). Daher muss bei der Integration des SEPA-Nuclets dafür gesorgt werden, dass gewisse Datenbankzugriffe auf den tatsächlich genutzten Businessobjekte operieren können. D. h. die verwendeten Klassen und Attributnamen müssen bekannt sein. Die Bereitstellung erfolgt über businessobjektbezogene Fassaden.

Außerdem stellen die Fassaden Wrapper-Methoden bereit, die die tatsächlich verwendeten Businessobjekte auf die in Abschnitt 4.8.1. definierten Wrapper-Objekte umwandeln.

Diese Methoden sind von Ihnen im Normalfall nicht anzupassen, sofern Sie die Wrapper-Klassen aus dem SEPA-Nuclet verwenden.

Bei der Anbindung des SEPA-Nuclets sind die folgende Fassadenklassen für Debitoren, Referenzen und Zahlungsreferenzen anzupassen (die Klassen befinden sich alle im Java-Package **org.nuclet.sepa.facade**):

Fassaden-Klasse	Anmerkung	Funktion	Anwendungsfall
CreditorFacade	neu ab v2.0.0	- Definition der tatsächlichen Kreditorenklasse (Kontakt, Lieferant, etc.) - Wrapper-Methoden	Überweisungen
CreditorReferenceFacade	neu ab v2.0.0	- Definition der tatsächlich verwendeten Referenzobjekte bei Überweisungen (Eingangsrechnung, Forderung, etc.) - Wrapper-Methoden	
CreditTransferReferenceFacade	neu ab v2.0.0	- Definition der tatsächlichen Zahlungsreferenzklasse (Rechnungsposition, Forderung, Buchung, etc.) - Definition von Attributen - Wrapper-Methoden	
DebitorFacade	neu ab v2.0.0	- Definition der tatsächlichen Debitorenklasse (Kontakt, Kunde, Mitglied, etc.) - Wrapper-Methoden	Lastschriften
DebitorReferenceFacade	neu ab v2.0.0	- Definition der tatsächlichen Zahlungsreferenzklasse (Ausgangsrechnung, Forderung, etc.) - Definition von Attributen - Wrapper-Methoden	
DirectDebitReferenceFacade	neu ab v2.0.0	- Definition der tatsächlich verwendeten Referenzobjekte bei Lastschriften (Rechnungsposition, Forderung, Buchung, etc.) - Wrapper-Methoden	
ReferenceFacade wird ab v2.0.0 ersetzt durch DebitorReferenceFacade	obsolet	- Definition der tatsächlich verwendeten Referenzobjekte - Wrapper-Methoden	Lastschriften
PaymentReferenceFacade wird ab v2.0.0 ersetzt durch DirectDebitReferenceFacade	obsolet	- Definition der tatsächlichen Zahlungsreferenzklasse - Definition von Attributen - Wrapper-Methoden	

Tabelle 4.8.2: Übersicht, Anpassungen in Fassaden-Klassen

4.7.2.1 CreditorFacade

In der Klasse CreditorFacade muss eine Methode implementiert werden:

- getWrappedClass() sollte die Java-Klasse des verwendeten Kreditorenobjektes zurückgeben

Ein Beispiel dazu ist jeweils in einem Kommentarblock angegeben, dieses Beispiel ist an das tatsächlich genutzte Debitorenbusinessobjekt (bzw. dessen BusinessObject-Klasse) anzupassen.



Bitte beachten Sie außerdem, dass das generische "<T>" in den Zeilen 37 und 84 entsprechend ersetzt werden muss.

org.nuclet.sepa.facade.CreditorFacade

```
package org.nuclet.sepa.facade;

import java.util.ArrayList;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.context.UpdateContext;
import org.nuclos.api.exception.BusinessException;

import org.nuclet.sepa.facade.AbstractCreditorFacade;
import org.nuclet.sepa.wrapper.AbstractCreditorWrapper;
import org.nuclet.sepa.wrapper.CreditorWrapper;

// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// import org.nuclet.buinessemplate.Supplier;

/**
 * Facade for Business Objects of type "Creditor"
 *
 * @version 1.0
 * @date 30.01.2015
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.0.0
 * @sincenucletversion 2.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class CreditorFacade<T extends BusinessObject & Modifiable> extends AbstractCreditorFacade<T> //
    replace! AbstractCreditorFacade<Supplier>
{
    private static final CreditorFacade instance = new CreditorFacade();

    /**
     * Liefert die Singleton-Instanz dieser Klasse
     */
    public static CreditorFacade getInstance()
    {
        return instance;
    }

    /**
     * Wrap the context's actual reference objects inside an <code>AbstractReferenceWrapper</code>

```

```

*
* @param context A given <code>UpdateContext</code>
*
* @return the context's actual reference object, wrapped inside an <code>AbstractReferenceWrapper</code>
*
*/
public AbstractCreditorWrapper getWrapper(final UpdateContext context)
{
    return new CreditorWrapper(context.getBusinessObject(getWrappedClass()));
}

/**
 * Wrap the actual reference object inside an <code>AbstractReferenceWrapper</code>
 *
 * @param boCreditor The actual reference object
 *
 * @return the actual reference object, wrapped inside an <code>AbstractReferenceWrapper</code>
 *
 */
public AbstractCreditorWrapper getWrapper(final Modifiable boCreditor)
{
    return new CreditorWrapper(boCreditor);
}

/**
 * Get the class of the actual reference objects, i.e. the class wrapped by <code>CreditorWrapper</code>
 *
 * @return the class of the actual reference objects, i.e. the class wrapped by <code>CreditorWrapper</code>
 *
 */
public Class<T> getWrappedClass() // @replace! public class<Supplier> getWrappedClass()
{
    // @replace!
    //
    // return Supplier.class;

    return null;
}
}

```

4.7.2.2 CreditorReferenceFacade

In der Klasse CreditorReferenceFacade muss ebenfalls nur eine Methode implementiert werden:

- getWrappedClass() sollte die Java-Klasse des verwendeten Referenzobjektes zurückgeben

Ein Beispiel dazu ist jeweils in einem Kommentarblock angegeben, dieses Beispiel ist an das tatsächlich genutzte Referenzbusinessobjekt (bzw. dessen BusinessObject-Klasse) anzupassen.



Bitte beachten Sie außerdem, dass das generische "<T>" in den Zeilen 35 und 66 entsprechend ersetzt werden muss.

org.nuclet.sepa.facade.CreditorReferenceFacade

```

package org.nuclet.sepa.facade;

import java.util.ArrayList;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.exception.BusinessException;

import org.nuclet.sepa.facade.AbstractCreditorReferenceFacade;
import org.nuclet.sepa.wrapper.AbstractCreditorReferenceWrapper;

```

```

import org.nuclet.sepa.wrapper.CreditorReferenceWrapper;

// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// import org.nuclet.businesstemplate.SupplierBilling;

/**
 * Facade for Business Objects of type "Creditor Reference"
 *
 *
 * @version 1.0
 * @date 02.02.2015
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.0.0
 * @sincenucletversion 2.0.0
 * @since 02.02.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class CreditorReferenceFacade<T extends BusinessObject & Modifiable> extends
AbstractCreditorReferenceFacade<T> // @replace public class<ClientBilling> getWrappedClass()
{
    private static final CreditorReferenceFacade instance = new CreditorReferenceFacade();

    /**
     * Liefert die Singleton-Instanz dieser Klasse
     */
    public static CreditorReferenceFacade getInstance()
    {
        return instance;
    }

    /**
     * Wrap the actual reference objects inside an <code>AbstractCreditorReferenceWrapper</code>
     *
     * @return the actual reference objects, wrapped inside an <code>AbstractCreditorReferenceWrapper</code>
     */
    public AbstractCreditorReferenceWrapper getWrapper(final Modifiable boReference)
    {
        return new CreditorReferenceWrapper(boReference);
    }

    /**
     * Get the class of the actual reference objects, i.e. the class wrapped by <code>CreditorReferenceWrapper<
     /code>
     *
     * @return the class of the actual reference objects, i.e. the class wrapped by
     <code>CreditorReferenceWrapper</code>
     */
    public Class<T> getWrappedClass() // @replace public class<SupplierBilling> getWrappedClass()
    {
        // @replace!
        //
        // return SupplierBilling.class;

        return null;
    }
}

```

4.7.2.3 CreditTransferReferenceFacade

In der Klasse CreditTransferReferenceFacade müssen zwei Methoden implementiert werden:

- getSEPATransactionIdAttribute() sollte das Attribut zurückliefern, das auf die SEPA-Transaktion referenziert
- getWrappedClass() sollte die Java-Klasse des verwendeten Zahlungsreferenzobjektes zurückgeben

Ein Beispiel dazu ist jeweils in einem Kommentarblock angegeben, dieses Beispiel ist an das tatsächlich genutzte Zahlungsreferenzbusinessobjekt (bzw. dessen BusinessObject-Klasse) anzupassen.



Bitte beachten Sie außerdem, dass das generische "<T>" in den Zeilen 39 und 70 entsprechend ersetzt werden muss.

org.nuclet.sepa.facade.CreditTransferReferenceFacade

```
package org.nuclet.sepa.facade;

import java.util.ArrayList;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.attribute.ForeignKeyAttribute;
import org.nuclos.api.exception.BusinessException;

import org.nuclet.sepa.facade.AbstractCreditTransferReferenceFacade;
import org.nuclet.sepa.wrapper.AbstractCreditTransferReferenceWrapper;
import org.nuclet.sepa.wrapper.CreditTransferReferenceWrapper;

// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// import org.nuclet.buinessstemplate.SupplierBillingPosition;

/**
 * Facade for Business Objects of type "CreditTransferReference"
 *
 * * @note replaces the original class PaymentReferenceFacade
 *
 * * @version 1.0
 * * @date 02.02.2015
 * * @nuclet org.nuclet.SEPA
 * * @nucletversion 2.0.0
 * * @sincenucletversion 2.0.0
 * * @since 02.02.2015
 *
 * * @author frank.lehmann@nuclos.de
 *
 */
public class CreditTransferReferenceFacade<T extends BusinessObject & Modifiable> extends
AbstractCreditTransferReferenceFacade<T> // @replace!
AbstractCreditTransferReferenceFacade<SupplierBillingPosition>
{
    private static final CreditTransferReferenceFacade instance = new CreditTransferReferenceFacade();

    /**
     * Liefert die Singleton-Instanz dieser Klasse
     *
     */
    public static CreditTransferReferenceFacade getInstance()
    {
        return instance;
    }

    /**
     * Wrap the actual payment reference objects inside an <code>AbstractCreditTransferReferenceWrapper</code>
     *
     */
}
```

```

    * @return the actual payment reference objects, wrapped inside an
<code>AbstractCreditTransferReferenceWrapper</code>
    *
    */
    public AbstractCreditTransferReferenceWrapper getWrapper(final Modifiable boDirectDebitReference)
    {
        return new CreditTransferReferenceWrapper(boDirectDebitReference);
    }

    /**
    * Get the class of the actual payment reference objects, i.e. the class wrapped by
<code>CreditTransferReferenceWrapper</code>
    *
    * @return the class of the actual payment reference objects, i.e. the class wrapped by
<code>CreditTransferReferenceWrapper</code>
    *
    */
    public Class<T> getWrappedClass() // @replace! public Class<SupplierBillingPosition> getWrappedClass()
    {
        // @replace!
        //
        // return SupplierBillingPosition.class;

        return null;
    }

    /**
    * Get the attribute, that identifies the SEPA transaction
    *
    * @return the attribute, that identifies the SEPA transaction
    *
    */
    protected ForeignKeyAttribute<Long> getSEPATransactionIdAttribute()
    {
        // @replace!
        //
        // return SupplierBillingPosition.SEPATransactionId;

        return null;
    }
}

```

4.7.2.4 DebitorFacade

In der Klasse DebitorFacade muss eine Methode implementiert werden:

- getWrappedClass() sollte die Java-Klasse des verwendeten Debitorenobjektes zurückgeben

Ein Beispiel dazu ist jeweils in einem Kommentarblock angegeben, dieses Beispiel ist an das tatsächlich genutzte Debitorenbusinessobjekt (bzw. dessen BusinessObject-Klasse) anzupassen.



Bitte beachten Sie außerdem, dass das generische "<T>" in den Zeilen 36 und 84 entsprechend ersetzt werden muss.

org.nuclet.sepa.facade.DebitorFacade

```

package org.nuclet.sepa.facade;

import java.util.ArrayList;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.context.UpdateContext;
import org.nuclos.api.exception.BusinessException;

```

```

import org.nuclet.sepa.facade.AbstractDebitorFacade;
import org.nuclet.sepa.wrapper.AbstractDebitorWrapper;
import org.nuclet.sepa.wrapper.DebitorWrapper;

// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// Beispiel:
//
// import org.nuclet.businessemplate.Client;

/**
 * Facade for Business Objects of type "Debitor"
 *
 * @version 1.0
 * @date 21.03.2014
 * @nuclet org.nuclet.SEPA
 * @nucletversion 1.1.4
 * @sincenucletversion 1.0.0
 * @since 21.03.2014
 *
 * @author frank.lehmann@nuclos.de
 */
public class DebitorFacade<T extends BusinessObject & Modifiable> extends AbstractDebitorFacade<T> // replace!
AbstractDebitorFacade<Client>
{
    private static final DebitorFacade instance = new DebitorFacade();

    /**
     * Liefert die Singleton-Instanz dieser Klasse
     *
     */
    public static DebitorFacade getInstance()
    {
        return instance;
    }

    /**
     * Wrap the context's actual reference objects inside an <code>AbstractReferenceWrapper</code>
     *
     * @param context A given <code>UpdateContext</code>
     *
     * @return the context's actual reference object, wrapped inside an <code>AbstractReferenceWrapper</code>
     */
    public AbstractDebitorWrapper getWrapper(final UpdateContext context)
    {
        return new DebitorWrapper(context.getBusinessObject(getWrappedClass()));
    }

    /**
     * Wrap the actual reference object inside an <code>AbstractReferenceWrapper</code>
     *
     * @param boDebitor The actual reference object
     *
     * @return the actual reference object, wrapped inside an <code>AbstractReferenceWrapper</code>
     */
    public AbstractDebitorWrapper getWrapper(final Modifiable boDebitor)
    {
        return new DebitorWrapper(boDebitor);
    }

    /**
     * Get the class of the actual reference objects, i.e. the class wrapped by <code>DebitorWrapper</code>
     *
     * @return the class of the actual reference objects, i.e. the class wrapped by <code>DebitorWrapper</code>

```

```

    *
    */
    public Class<T> getWrappedClass() // @replace! public class<Client> getWrappedClass()
    {
        // @replace!
        //
        // return Client.class;

        return null;
    }
}

```

4.7.2.5 DebitorReferenceFacade

In der Klasse DebitorReferenceFacade muss ebenfalls nur eine Methode implementiert werden:

- getWrappedClass() sollte die Java-Klasse des verwendeten Referenzobjektes zurückgeben

Ein Beispiel dazu ist jeweils in einem Kommentarblock angegeben, dieses Beispiel ist an das tatsächlich genutzte Referenzbusinessobjekt (bzw. dessen BusinessObject-Klasse) anzupassen.



Bitte beachten Sie außerdem, dass das generische "<T>" in den Zeilen 35 und 66 entsprechend ersetzt werden muss.

org.nuclet.sepa.facade.DebitorReferenceFacade

```

package org.nuclet.sepa.facade;

import java.util.ArrayList;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.exception.BusinessException;

import org.nuclet.sepa.wrapper.AbstractDebitorReferenceWrapper;
import org.nuclet.sepa.wrapper.DebitorReferenceWrapper;

// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!
//
// import org.nuclet.buinessstemplate.ClientBilling;

/**
 * Facade for Business Objects of type "Debitor Reference"
 *
 * @note replaces the original class ReferenceFacade
 *
 * @version 1.0
 * @date 30.01.2015
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.0.0
 * @sincenucletversion 2.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class DebitorReferenceFacade<T extends BusinessObject & Modifiable> extends
AbstractDebitorReferenceFacade<T> // @replace public class<ClientBilling> getWrappedClass()
{
    private static final DebitorReferenceFacade instance = new DebitorReferenceFacade();

    /**

```

```

* Liefert die Singleton-Instanz dieser Klasse
*
*/
public static DebitorReferenceFacade getInstance()
{
    return instance;
}

/**
* Wrap the actual reference objects inside an AbstractDebitorReferenceWrapper
*
* @return the actual reference objects, wrapped inside an AbstractDebitorReferenceWrapper
*
*/
public AbstractDebitorReferenceWrapper getWrapper(final Modifiable boReference)
{
    return new DebitorReferenceWrapper(boReference);
}

/**
* Get the class of the actual reference objects, i.e. the class wrapped by DebitorReferenceWrapper
*
* @return the class of the actual reference objects, i.e. the class wrapped by DebitorReferenceWrapper
<code>
*
*/
public Class<T> getWrappedClass() // @replace public class<ClientBilling> getWrappedClass()
{
    // @replace!
    //
    // return ClientBilling.class;

    return null;
}
}

```

4.7.2.6 DirectDebitReferenceFacade

In der Klasse DirectDebitReferenceFacade müssen zwei Methoden implementiert werden:

- getSEPATransactionIdAttribute() sollte das Attribut zurückliefern, das auf die SEPA-Transaktion referenziert
- getWrappedClass() sollte die Java-Klasse des verwendeten Zahlungsreferenzobjektes zurückgeben

Ein Beispiel dazu ist jeweils in einem Kommentarblock angegeben, dieses Beispiel ist an das tatsächlich genutzte Zahlungsreferenzbusinessobjekt (bzw. dessen BusinessObject-Klasse) anzupassen.



Bitte beachten Sie außerdem, dass das generische "<T>" in den Zeilen 39 und 70 entsprechend ersetzt werden muss.

org.nuclet.sepa.facade.DirectDebitReferenceFacade

```

package org.nuclet.sepa.facade;

import java.util.ArrayList;
import java.util.List;

import org.nuclos.api.businessobject.BusinessObject;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.businessobject.attribute.ForeignKeyAttribute;
import org.nuclos.api.exception.BusinessException;

import org.nuclet.sepa.facade.AbstractDirectDebitReferenceFacade;
import org.nuclet.sepa.wrapper.AbstractDirectDebitReferenceWrapper;
import org.nuclet.sepa.wrapper.DirectDebitReferenceWrapper;

// @replace! Bitte bei Nuclet-Integration mit eigenem Code ersetzen!

```

```

//
// Beispiel:
//
// import org.nuclet.businesstemplate.ClientBillingPosition;

/**
 * Facade for Business Objects of type "DirectDebitReference"
 *
 * @note replaces the original class PaymentReferenceFacade
 *
 * @version 1.0
 * @date 30.01.2015
 * @nuclet org.nuclet.SEPA
 * @nucletversion 2.0.0
 * @sincenucletversion 2.0.0
 * @since 30.01.2015
 *
 * @author frank.lehmann@nuclos.de
 */
public class DirectDebitReferenceFacade<T extends BusinessObject & Modifiable> extends
AbstractDirectDebitReferenceFacade<T> // @replace! AbstractPaymentReferenceFacade<ClientBillingPosition>
{
    private static final DirectDebitReferenceFacade instance = new DirectDebitReferenceFacade();

    /**
     * Liefert die Singleton-Instanz dieser Klasse
     */
    /**
     public static DirectDebitReferenceFacade getInstance()
     {
         return instance;
     }

    /**
     * Wrap the actual payment reference objects inside an <code>AbstractDirectDebitReferenceWrapper</code>
     *
     * @return the actual payment reference objects, wrapped inside an
     <code>AbstractDirectDebitReferenceWrapper</code>
     */
    /**
     public AbstractDirectDebitReferenceWrapper getWrapper(final Modifiable boDirectDebitReference)
     {
         return new DirectDebitReferenceWrapper(boDirectDebitReference);
     }

    /**
     * Get the class of the actual payment reference objects, i.e. the class wrapped by
     <code>DirectDebitReferenceWrapper</code>
     *
     * @return the class of the actual payment reference objects, i.e. the class wrapped by
     <code>DirectDebitReferenceWrapper</code>
     */
    /**
     public Class<T> getWrappedClass() // @replace! public Class<ClientBillingPosition> getWrappedClass()
     {
         // @replace!
         //
         // return ClientBillingPosition.class;

         return null;
     }

    /**
     * Get the attribute, that identifies the SEPA transaction
     *
     * @return the attribute, that identifies the SEPA transaction
     */
    /**

```

```
protected ForeignKeyAttribute<Long> getSEPATransactionIdAttribute()  
{  
    // @replace!  
    //  
    // return ClientBillingPosition.SEPAPaymentInformation;  
  
    return null;  
}  
  
}
```

Verwandte Seiten

Nuclet: SEPA (sind im Handelsnuclet verfügbar)

SEPA-Datenblatt

[Öffnen](#)

Schnittstellen

Alle Integrationen

[Öffnen](#)