

4.8.2 SEPA: Business object facades

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

🔧 4.8.2 SEPA: Business object facades

Define the actually used business object classes and attributes via facades so that the SEPA Nuclet's database accesses operate on your objects.

HOW-TO 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 SEPA Nuclet works with wrapper objects instead of the BusinessObjects known only to the target Nuclet (see section 4.8.1). Therefore, when integrating the SEPA Nuclet, it must be ensured that certain database accesses can operate on the actually used business objects, i.e. the classes and attribute names used must be known. This is provided via business-object-specific facades.

In addition, the facades provide wrapper methods that convert the actually used business objects into the wrapper objects defined in section 4.8.1.

You normally do not have to adjust these methods, provided you use the wrapper classes from the SEPA Nuclet.

When connecting the SEPA Nuclet, the following facade classes for debtors, references and payment references must be adjusted (the classes are all located in the Java package **org.nuclet.sepa.facade**):

Facade class	Note	Function	Use case
CreditorFacade	new from v2.0.0	- definition of the actual creditor class (contact, supplier, etc.) - wrapper methods	Credit transfers
CreditorReferenceFacade	new from v2.0.0	- definition of the actually used reference objects for credit transfers (incoming invoice, receivable, etc.) - wrapper methods	
CreditTransferReferenceFacade	new from v2.0.0	- definition of the actual payment reference class (invoice line item, receivable, posting, etc.) - definition of attributes - wrapper methods	
DebtorFacade	new from v2.0.0	- definition of the actual debtor class (contact, customer, member, etc.) - wrapper methods	Direct debits
DebtorReferenceFacade	new from v2.0.0	- definition of the actual payment reference class (outgoing invoice, receivable, etc.) - definition of attributes - wrapper methods	
DirectDebitReferenceFacade	new from v2.0.0	- definition of the actually used reference objects for direct debits (invoice line item, receivable, posting, etc.) - wrapper methods	

ReferenceFacade replaced from v2.0.0 by DebitorReferenceFacade	obsolete	- definition of the actually used reference objects - wrapper methods	Direct debits
PaymentReferenceFacade replaced from v2.0.0 by DirectDebitReferenceFacade	obsolete	- definition of the actual payment reference class - definition of attributes - wrapper methods	

Table 4.8.2: Overview, adjustments in facade classes

4.7.2.1 CreditorFacade

One method must be implemented in the CreditorFacade class:

- getWrappedClass() should return the Java class of the creditor object used

An example is given in each case in a comment block; this example must be adapted to the actually used debtor business object (or its BusinessObject class).



Please also note that the generic "<T>" must be replaced accordingly in lines 37 and 84.

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.buinessstemplate.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

Likewise, only one method must be implemented in the CreditorReferenceFacade class:

- getWrappedClass() should return the Java class of the reference object used

An example is given in each case in a comment block; this example must be adapted to the actually used reference business object (or its BusinessObject class).



Please also note that the generic "<T>" must be replaced accordingly in lines 35 and 66.

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

Two methods must be implemented in the CreditTransferReferenceFacade class:

- getSEPATransactionIdAttribute() should return the attribute that references the SEPA transaction
- getWrappedClass() should return the Java class of the payment reference object used

An example is given in each case in a comment block; this example must be adapted to the actually used payment reference business object (or its BusinessObject class).



Please also note that the generic "<T>" must be replaced accordingly in lines 39 and 70.

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 AbstractCreditTransferReferenceWrapper
     *
     * @return the actual payment reference objects, wrapped inside an
     * AbstractCreditTransferReferenceWrapper
     */
    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
     * CreditTransferReferenceWrapper
     *
     * @return the class of the actual payment reference objects, i.e. the class wrapped by
     * CreditTransferReferenceWrapper
     */
    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

One method must be implemented in the DebitorFacade class:

- `getWrappedClass()` should return the Java class of the debtor object used

An example is given in each case in a comment block; this example must be adapted to the actually used debtor business object (or its BusinessObject class).



Please also note that the generic "<T>" must be replaced accordingly in lines 36 and 84.

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.buinessstemplate.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

Likewise, only one method must be implemented in the DebitorReferenceFacade class:

- getWrappedClass() should return the Java class of the reference object used

An example is given in each case in a comment block; this example must be adapted to the actually used reference business object (or its BusinessObject class).



Please also note that the generic "<T>" must be replaced accordingly in lines 35 and 66.

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

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

     return null;
     }
}

```

4.7.2.6 DirectDebitReferenceFacade

Two methods must be implemented in the DirectDebitReferenceFacade class:

- getSEPATransactionIdAttribute() should return the attribute that references the SEPA transaction
- getWrappedClass() should return the Java class of the payment reference object used

An example is given in each case in a comment block; this example must be adapted to the actually used payment reference business object (or its BusinessObject class).



Please also note that the generic "<T>" must be replaced accordingly in lines 39 and 70.

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 attribue, that identifies the SEPA transaction
     *
     * @return the attribue, that identifies the SEPA transaction
     */
    protected ForeignKeyAttribute<Long> getSEPATransactionIdAttribute()
    {
        // @replace!
        //
        // return ClientBillingPosition.SEPAPaymentInformation;

        return null;
    }
}
```

Related pages

Nuclet: SEPA

SEPA Nuclet data sheet

[Open](#)

Interfaces

All integrations

[Open](#)