Microsoft Dynamics® AX 2012

Updating the Uses of the Ledger Posting Framework for Microsoft Dynamics AX 2012

White Paper

This document describes how to convert existing code patterns to the new patterns in the updated ledger posting framework in Microsoft Dynamics AX 2012.

http://microsoft.com/dynamics/ax

Date: July 2011

Author: Eric Pegors, Senior Software Development Engineer

Send suggestions and comments about this document to adocs@microsoft.com. Please include the title with your feedback.

Table of Contents

Overview

The ledger posting framework is the API used to post to the general ledger and includes the **LedgerVoucher**, **LedgerVoucherObject**, and **LedgerVoucherTransObject** classes and related classes. Developers can use the factory methods in the **LedgerVoucherTransObject** class to create instances of these classes.

Multiple factory methods exist to support different input scenarios. After using one of these methods to create a class instance, a developer can customize the instance by using the **parm** methods.

In previous versions of Microsoft Dynamics AX, the **LedgerVoucherTransObject** class exposed the fields in the legacy LedgerTrans table. Changes to the data model in Microsoft Dynamics AX 2012 have resulted in the following mappings:

- The **LedgerVoucherTransObject** class maps to the GeneralJournalAccountEntry table and the LedgerEntry table.
- The **LedgerVoucherObject** class maps to the GeneralJournalEntry table.
- The **LedgerVoucher** class maps to the LedgerEntryJournal table.

This white paper provides information that developers can leverage to update uses of the ledger posting framework in Microsoft Dynamics AX 2012.

Updating the LedgerVoucherTransObject class

Most of the factory methods in the **LedgerVoucherTransObject** class have been replaced in Microsoft Dynamics AX 2012. Developers must, therefore, replace existing methods in their code with the new methods.

In addition, changes to references and parameters must be made when updating the code.

Update guidelines

- Support the account and dimension feature by replacing references to the **AccountNum** and **Dimension** fields with references to a ledger dimension field.
- Add or remove parameters to more closely match key scenarios. For example, the new **newTransactionAmountDefault** method does not contain parameters for the accounting currency amount or the reporting currency amount. Instead, it takes a parameter for the exchange rate helper (CurrencyExchangeHelper), which is used to calculate these values from the transaction currency amount (which is still a parameter).
- Use the **parm** methods:
    - To set values that are not included as method parameters.
    - To set infrequently used values. Microsoft developers have removed parameters that are not frequently used.
        - For example, the source table ID (tableId) and source record ID (recId) have been removed as parameters for the factory methods. Use the **parmSourceTableId** and **parmSourceRecId** methods to set the TableId and RecId values for all scenarios.

Note: Microsoft developers have also removed parameters that are related to the lack of normalization in the LedgerTrans table. The LedgerTrans table has been replaced in Microsoft Dynamics AX 2012 with other tables.

Dimension validation

The LedgerVoucherTransObject class supports the ability to not perform any dimension validation with the parmSkipDimensionValidation method. Passing true will ensure that a previously posted dimension combination will post in a scenario, such as reversing a dimension combination that was previously posted with dimension validation. The default behavior is to perform dimension validation.

```
LedgerVoucherTransObject transaction;
transaction = LedgerVoucherTransObject::newBasicDefault(…);
transaction.parmSkipDimensionValidation(true);
```

Replacing factory methods

The following table lists the factory methods in the **LedgerVoucherTransObject** class that are now obsolete and the new methods that should be used to replace them.

Obsolete methods	New methods
newCreateTrans	**newBasicDefault; newTransactionAmountDefault; newTransactionAccountingAmountsDefault**
newTransExchAdj	**newAdjustmentDefault**
newTransRoundOff	**newAdjustmentDefault**
newVoucherTrans	**newGeneralJournal**
newCreateTrans	**newLedgerAllocationItem**

New factory methods

Key information about the new factory methods is included below. For complete method signatures, see the Appendix section.

newBasicDefault method

```
#/// <summary>
#/// Initializes a new instance of the LedgerVoucherTransObject class using a
#/// ledger posting reference for defaulting.
#/// </summary>
#/// <param name="_defaultLedgerPostingReference">
#/// The ledger posting reference used for defaulting.
#/// </param>
#/// <param name="_postingType">
#/// The posting type of the general journal entry.
#/// </param>
#/// <param name="_ledgerDimensionId">
#/// The dimension attribute value combination of the general journal entry.
#/// </param>
#/// <param name="_transactionCurrencyCode">
#/// The currency code of the general journal entry.
#/// </param>
#/// <param name="_transactionCurrencyAmount">
#/// The amount in the transaction currency.
#/// </param>
#/// <param name="_accountingCurrencyAmount">
#/// The amount in the accounting currency.
#/// </param>
#/// <param name="_reportingCurrencyAmount">
#/// The amount in the reporting currency.
#/// </param>
#/// <returns>
#/// A new instance of the LedgerVoucherTransObject class.
#/// </returns>
#public static LedgerVoucherTransObject newBasicDefault(
#    LedgerVoucherObject _defaultLedgerPostingReference,
#    LedgerPostingType _postingType,
#    LedgerDimensionAccount _ledgerDimensionId,
#    CurrencyCode _transactionCurrencyCode,
#    money _transactionCurrencyAmount,
#    moneyMST _accountingCurrencyAmount,
#    moneyMST _reportingCurrencyAmount)
```

Use this method when the transaction currency amount, accounting currency amount, and reporting currency amount are known.

- Key parameters
  - Transaction currency amount
  - Accounting currency amount
  - Reporting currency amount
- No exchange rate information is used.

newTransactionAmountDefault method

```
        #/// <summary>
        #///    Initializes a new instance of the <c>LedgerVoucherTransObject</c> class by using
a transaction
        #///    currency amount and a ledger posting reference for defaulting.
        #/// </summary>
        #/// <param name="_defaultLedgerPostingReference">
        #///    The ledger posting reference to use for defaulting.
        #/// </param>
        #/// <param name="_postingType">
        #///    The posting type of the general journal entry.
        #/// </param>
        #/// <param name="_ledgerDimensionId">
        #///    The dimension attribute value combination of the general journal entry.
        #/// </param>
        #/// <param name="_transactionCurrencyCode">
        #///    The currency code of the general journal entry.
        #/// </param>
        #/// <param name="_transactionCurrencyAmount">
        #///    The amount in the transaction currency.
        #/// </param>
        #/// <param name="_exchangeRateHelper">
        #///    The accounting currency amount and secondary currency amount exchange rates.
        #/// </param>
        #/// <returns>
        #///    A new instance of the <c>LedgerVoucherTransObject</c> class.
        #/// </returns>
        #/// <remarks>
        #///    The default ledger posting reference is used to set the transaction type and
exchange rate date.
        #/// </remarks>
        #public static LedgerVoucherTransObject newTransactionAmountDefault(
        #    LedgerVoucherObject _defaultLedgerPostingReference,
        #    LedgerPostingType _postingType,
        #    LedgerDimensionAccount _ledgerDimensionId,
        #    CurrencyCode _transactionCurrencyCode,
        #    money _transactionCurrencyAmount,
        #    CurrencyExchangeHelper _exchangeRateHelper)
```

Use this method when only the transaction currency amount is known.

- Key parameters
  - Transaction currency amount
  - Exchange rate information
- The accounting currency amount and reporting currency amount are calculated by using the exchange rate information.

newTransactionAccountingAmountsDefault method

```
        #/// <summary>
        #///    Initializes a new instance of the LedgerVoucherTransObject class with a a
transaction currency
        #///    amount, an accounting currency amount, and a ledger posting reference for
defaulting.
        #/// </summary>
        #/// <param name="_defaultLedgerPostingReference">
        #///    The ledger posting reference used for defaulting.
        #/// </param>
        #/// <param name="_postingType">
        #///    The posting type of the general journal entry.
        #/// </param>
        #/// <param name="_ledgerDimensionId">
        #///    The dimension attribute value combination of the general journal entry.
        #/// </param>
        #/// <param name="_transactionCurrencyCode">
        #///    The currency code of the general journal entry.
        #/// </param>
        #/// <param name="_transactionCurrencyAmount">
        #///    The amount in the transaction currency.
        #/// </param>
        #/// <param name="_accountingCurrencyAmount">
        #///    The amount in the accounting currency.
        #/// </param>
        #/// <param name="_currencyExchangeHelper">
        #///    An <c>CurrencyExchangeHelper</c> object initialized for the current
LedgerVoucherTransObject.
        #/// </param>
        #/// <returns>
        #///    A new instance of the LedgerVoucherTransObject class.
        #/// </returns>
        #/// <remarks>
        #///    The default ledger posting reference is used to set the transaction type.
        #/// </remarks>
        #public static LedgerVoucherTransObject newTransactionAccountingAmountsDefault(
        #    LedgerVoucherObject _defaultLedgerPostingReference,
        #    LedgerPostingType _postingType,
        #    recId _ledgerDimensionId,
        #    CurrencyCode _transactionCurrencyCode,
        #    money _transactionCurrencyAmount,
        #    moneyMST _accountingCurrencyAmount,
        #    CurrencyExchangeHelper _currencyExchangeHelper)
```

Use this method when only the transaction currency amount and accounting currency amount are known.

- Key parameters
  - Transaction currency amount
  - Accounting currency amount
  - Exchange rate information
- The reporting currency amount is calculated by using the exchange rate information.

- It is acceptable to use the exchange rate information to calculate the reporting currency amount (by setting up the exchange rate helper with transaction currency and accounting currency amounts as parameters) and then use the **newBasicDefault** method.

newGeneralJournal method

```
        #/// <summary>
        #///    Initializes a new instance of the LedgerVoucherTransObject class from a general
journal entry.
        #/// </summary>
        #/// <param name="_generalJournalAccountEntry">
        #///    The line of the general journal entry.
        #/// </param>
        #/// <param name="_ledgerEntry">
        #///    The ledger entry of the general journal entry.
        #/// </param>
        #/// <returns>
        #///    A new instance of the LedgerVoucherTransObject class.
        #/// </returns>
        #public static LedgerVoucherTransObject newGeneralJournal(
        #    GeneralJournalAccountEntry _generalJournalAccountEntry,
        #    LedgerEntry _ledgerEntry = null)
```

Use this method when a transaction based on information previously posted to the general journal tables is needed.

- Key parameters
  - GeneralJournalAccountEntry table
  - LedgerEntry table
- All the required information for this method is found in the general journal tables, which replace parts of the LedgerTrans table that supported the general journal.

newAdjustmentDefault method

```
#/// <summary>
#/// Initializes a new instance of the LedgerVoucherTransObject class for an
#/// accounting currency amount or secondary currency amount adjustment.
#/// </summary>
#/// <param name="_defaultLedgerPostingReference">
#/// The ledger posting reference used for defaulting.
#/// </param>
#/// <param name="_postingType">
#/// The posting type of the general journal entry.
#/// </param>
#/// <param name="_ledgerDimensionId">
#/// The dimension attribute value combination of the general journal entry.
#/// </param>
#/// <param name="_transactionCurrencyCode">
#/// The currency code of the general journal entry.
#/// </param>
#/// <param name="_accountingCurrencyAmount">
#/// The amount in the accounting currency.
#/// </param>
#/// <param name="_reportingCurrencyAmount">
#/// The amount in the reporting currency.
#/// </param>
#/// <returns>
#/// A new instance of the LedgerVoucherTransObject class.
#/// </returns>
#/// <remarks>
#/// The transaction currency amount is set to zero.
#/// </remarks>
#public static LedgerVoucherTransObject newAdjustmentDefault(
#    LedgerVoucherObject _defaultLedgerPostingReference,
#    LedgerPostingType _postingType,
#    LedgerDimensionAccount _ledgerDimensionId,
#    CurrencyCode _transactionCurrencyCode,
#    moneyMST _accountingCurrencyAmount,
#    moneyMST _reportingCurrencyAmount)
```

Use this method for a transaction with a zero transaction currency amount.

- Key parameters
  - Accounting currency amount
  - Reporting currency amount
- No exchange rate information is used internally.
- Used in rounding adjustment scenarios

newLedgerAllocationItem method

```
        #/// <summary>
        #///    Initializes a new instance of the <c>LedgerVoucherTransObject</c> class from a
ledger allocation
        #///    item.
        #/// </summary>
        #/// <param name="_baseTransaction">
        #///    The base ledger posting transaction.
        #/// </param>
        #/// <param name="_allocationItem">
        #///    The ledger allocation item.
        #/// </param>
        #/// <returns>
        #///    A new instance of the <c>LedgerVoucherTransObject</c> class.
        #/// </returns>
        #public static LedgerVoucherTransObject newLedgerAllocationItem(
        #    LedgerVoucherTransObject _baseTransaction,
        #    LedgerAllocationItem _allocationItem)
```

Use this method in allocation account scenarios with the **LedgerAllocationController** class.

- Key parameters
  - Base **LedgerVoucherTransObject**
  - **LedgerAllocationItem** object
  - The base ledger posting transaction is cloned and a **LedgerAllocationItem** object is used to finish initializing the object.

Using the CurrencyExchangeHelper class

This class is part of the account and financial dimension network. Many transactions outside of the general ledger framework only track the transaction currency amount and use basic conversion to get the other values. The **CurrencyExchangeHelper** class is used in the general ledger framework to calculate the accounting currency amount or the reporting currency amount, for example, to convert a transaction currency to a reporting currency.

- Use a **CurrencyExchangeHelper** object to calculate the accounting currency amount and reporting currency amount.
- **CurrencyExchangeHelper::newLedger** and **CurrencyExchangeHelper::newExchangeDate** are the most commonly used factory methods for creating instances of the **CurrencyExchangeHelper** class.
- Currency is a parameter for the existing call to the **LedgerVoucherTransObject::newCreateTrans** method.

The **CurrencyExchangeHelper** object must be configured with information, such as the exchange date and exchange rates. This information can be retrieved as follows:

- The accounting date of the **LedgerVoucherObject** object is often used as the exchange date and can be accessed with the **LedgerVoucherObject.parmAccountingDate** method.
  - Use the **LedgerVoucher.findLedgerVoucherObject** method to retrieve the **LedgerVoucherObject** object if a **LedgerVoucherObject** variable is not available.

Code samples

The following code samples show some of the conversions that are required to update uses of the ledger posting framework from Microsoft Dynamics AX 2009 to Microsoft Dynamics AX 2012. The legacy code is shown unchanged. The updated code is shown with comments to highlight specific information.

Each of these code samples instantiates the LedgerVoucherTransObject class and configures it in a specific way. The new and the old code patterns provide objects that are functionally equivalent.

In previous versions of Microsoft Dynamics AX, if you were only posting one amount, you passed in a "0" value (the last value in the parameter list) for the accounting currency amount, and the method would calculate the value. The new method removes this parameter; you have to calculate the accounting currency yourself. The new method creates a **LedgerVoucherTransObject** object that is configured as it was in previous versions of Microsoft Dynamics AX.

AssetPostDisposal.post

Legacy code

```
ledgerVoucherTransObject = LedgerVoucherTransObject::newCreateTrans(
    ledgerVoucher.findLedgerVoucherObject(),
    LedgerPostingType::FixedAssetsDebit,
    l_assetDisposalParameters.Account,
    assetTrans.Dimension,
    companyCurrency, //assetTrans.currencyCode,
    -assetAmount,
    assetTrans.TableId,
    assetTrans.RecId,
    0);
```

Updated code

```
ledgerVoucherObject = ledgerVoucher.findLedgerVoucherObject();


//The currency exchange helper is used to calculate the accounting currency.
//amount and reporting currency amount.
currencyExchangeHelper = CurrencyExchangeHelper::newExchangeDate(
    Ledger::current(),
    ledgerVoucherObject.parmAccountingDate());


//Create a ledger dimension from the default account and default dimensions.
mergedLedgerDimension = DimensionDefaultingService::serviceCreateLedgerDimension(
    l_assetDisposalParameters.LedgerDimension,
    assetTrans.DefaultDimension);


transaction = LedgerVoucherTransObject::newTransactionAmountDefault(
    ledgerVoucherObject,
    LedgerPostingType::FixedAssetsDebit,
```

```
    mergedLedgerDimension,
    companyCurrency, //assetTrans.currencyCode,
    -assetAmount,
    currencyExchangeHelper);

transaction.parmSourceTableId(assetTrans.TableId);
transaction.parmSourceRecId(assetTrans.RecId);
```

CustVendVoucher.post

Legacy code

```
ledgerVoucherTransObject = LedgerVoucherTransObject::newCreateTrans(
    _ledgerPostingJournal.findLedgerVoucherObject(),
    ledgerPostingType,
    ledgerAccount,
    dimension,
    currencyCode,
    0,
    custVendTrans.TableId,
    custVendTrans.RecId,
    0,
    exchRate,
    exchRateSecondary,
    euroTriangulation,
    false,
    null,
    this.parmAmountMST());
```

Updated code

```
ledgerVoucherObject = _ledgerPostingJournal.findLedgerVoucherObject();

//Create a ledger dimension from the default account and default dimensions.
ledgerDimensionDefault = this.summaryLedgerDimension();
ledgerDimensionMerged = DimensionDefaultingService::serviceCreateLedgerDimension(
    ledgerDimensionDefault,
    defaultDimension);

transaction = LedgerVoucherTransObject::newAdjustmentDefault(
    ledgerVoucherObject,
    ledgerPostingType,
    ledgerDimensionMerged,
    currencyCode,
```

```
    this.parmAmountMST(),
    0.0); // reporting amount

transaction.parmSourceTableId(custVendTrans.TableId);
transaction.parmSourceRecId(custVendTrans.RecId);
```

BankVoucher.post

Legacy code

```
ledgerVoucherTransObject = LedgerVoucherTransObject::newCreateTrans(
    _ledgerPostingJournal.findLedgerVoucherObject(),
    posting,
    ledgerAccount,
    dimension,
    currencyCode,
    amountCur,
    0,
    0,
    0,
    exchRate,
    exchRateSecond,
    exchRatesTriangulation);
```

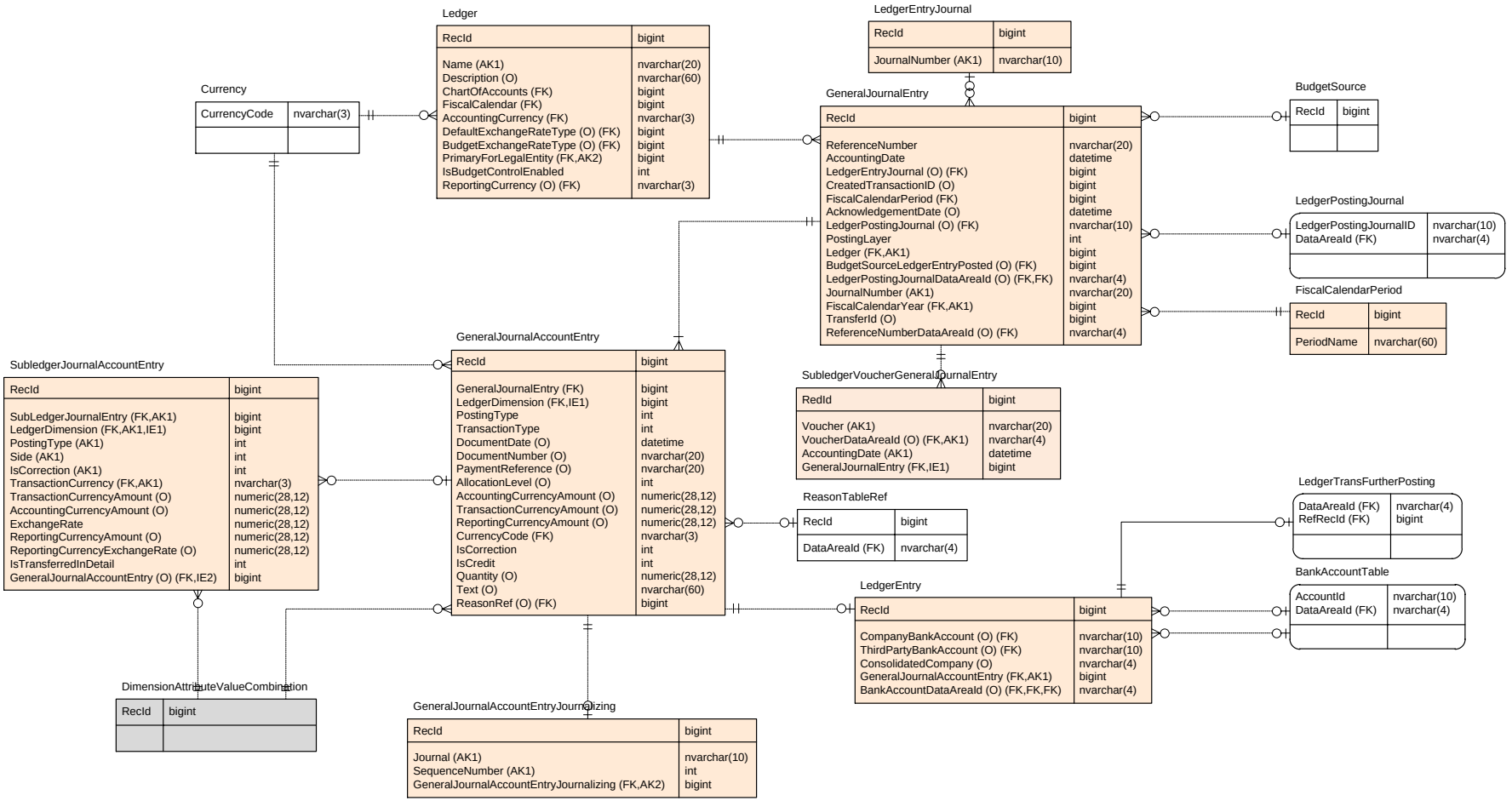
Updated code

```
ledgerVoucherObject = _ledgerPostingJournal.findLedgerVoucherObject();

//The currency exchange helper is used to calculate the accounting currency.
//amount and reporting currency amount.
currencyExchangeHelper = CurrencyExchangeHelper::newExchangeDate(
    Ledger::current(),
    ledgerVoucherObject.parmAccountingDate());

currencyExchangeHelper.parmExchangeRate1(exchRate);
currencyExchangeHelper.parmExchangeRate2(exchRateSecond);

//Create a ledger dimension from the default account and default dimensions.
ledgerDimensionMerged = DimensionDefaultingService::serviceCreateLedgerDimension(
    ledgerDimension,
    defaultDimension);

transaction = LedgerVoucherTransObject::newTransactionAmountDefault(
    ledgerVoucherObject,
```

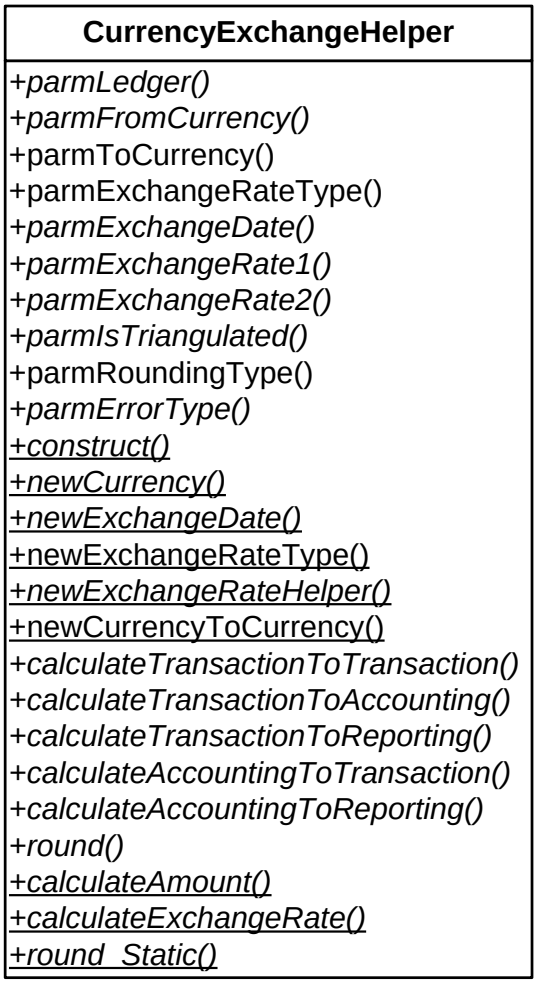
```
    posting,
    mergedLedgerDimension,
    currencyCode,
    amountCur,
    currencyExchangeHelper);
```

Appendix

Data model



Class reference



Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989
Worldwide +1-701-281-6500
www.microsoft.com/dynamics

Microsoft