IBM WebSphere Commerce

IBM

Calculation Framework Guide

Version 54

IBM WebSphere Commerce

Calculation Framework Guide

Version 54

> **Note:**
>
> Before using this information and the product it supports, be sure to read the general information under Appendix E, "Notices" on page 113.

First Edition (May 2002).

This edition applies to version 5.4 of IBM® WebSphere® Commerce and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. You can send your comments by any one of the following methods:

1. Electronically to the E-mail address listed below. Be sure to include your entire network address if you wish a reply.

   `Internet: torrcf@ca.ibm.com`

2. By regular mail to the following address:

   IBM Canada Ltd. Laboratory
   B3/KB7/8200/MKM
   8200 Warden Avenue
   Markham, ON L6G 1C7
   Canada

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Before you begin

The *IBM WebSphere Commerce Calculation Framework Guide* is intended to provide a technical overview of the calculation framework for Store Developers that need to implement calculations or extend the default implementation of calculations within WebSphere Commerce. The *IBM WebSphere Commerce Calculation Framework Guide* provides information about the framework and default implementations that are used to perform the calculation of monetary amounts in WebSphere Commerce. In particular, it provides details on the following topics:

- Calculation methods
- Calculation usages
- Calculation codes
- Calculation rules
- Calculation scales

Conventions used in this book

This book uses the following highlighting conventions:

Boldface type indicates commands or graphical user interface (GUI) controls such as names of fields, buttons, or menu choices.

`Monospaced type` indicates examples of text you enter exactly as shown, as well as directory paths.

Italic type is used for emphasis and variables for which you substitute your own values.

> This icon marks a Tip — additional information that can help you complete a task.

▶ AIX indicates information that is specific to WebSphere Commerce for AIX®.

▶ Linux indicates information that is specific to WebSphere Commerce for Linux.

▶ 400 indicates information specific to WebSphere Commerce for the IBM @server™ iSeries™ 400® (formerly called AS/400®)

▶ Solaris indicates information that is specific to WebSphere Commerce for Solaris™ Operating Environment software.

▶ 2000 indicates information that is specific to WebSphere Commerce for Windows® 2000.

▶ NT indicates information that is specific to WebSphere Commerce for Windows NT®.

Knowledge requirements

This book should be read by Store Developers that need to understand how to create or change calculations within WebSphere Commerce. Store Developers that are implementing and customizing calculations should have knowledge in the following areas:

- Java™ programming
- Structured Query Language (SQL)
- XML
- WebSphere Commerce store development

Where to find new information

This book may be updated in the future. Check the following WebSphere Commerce Web sites for updates:

WebSphere Commerce Business Edition

> `http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html`

WebSphere Commerce Professional Edition

> `http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html`

Updates may include new information and additional tutorials.

Chapter 1. Introduction to the calculation framework

Commerce systems need to calculate monetary amounts, such as discounts, shipping charges, and taxes, and apply them to applicable business objects such as objects representing line items in an order. Business rules and legal requirements specify how and under what conditions these monetary amounts should be calculated. When these rules and requirements change, a good commerce system can adapt to the changes with little or no programming changes.

WebSphere Commerce provides a flexible, generic framework that can be used to implement different kinds of calculations and apply them to the applicable business objects. The framework can handle a wide variety of business and legal requirements without programming. WebSphere Commerce provides a number of overridable method implementations from which you can select to do the calculations. If business or legal requirements require a programming change, many such changes can be limited to programming additional overridable method implementations without having to make any changes to existing programming.

The calculation framework is part of the WebSphere Commerce order subsystem. The order subsystem is a component of the WebSphere Commerce Server which provides shopping carts, order processing, and order management function support. Related services, such as pricing, taxation, payment, inventory, and fulfillment, are also part of the order subsystem. For more information on the order subsystem, see the online information.

WebSphere Commerce business objects and the calculation framework

The WebSphere Commerce calculation framework calculates monetary amounts associated with OrderItem business objects. An OrderItem represents something that a customer has selected for purchase. Each OrderItem has a reference to an offer, a contract, a shipping mode, and a fulfillment center. Monetary amounts determined by the calculation framework are stored with each OrderItem.

Each OrderItem also has a quantity attribute that is a unitless number. The quantity attribute can be multiplied by the nominal quantity attribute of the CatalogEntryShippingInformation object associated with the CatalogEntry object to arrive at the actual quantity represented by the OrderItem. The CatalogEntryShippingInformation object specifies the unit of measurement in which quantities are stated.

OrderItems can be grouped to make an Order. OrderItems that are part of an Order can be grouped to form SubOrders. OrderItems in a SubOrder object have the same shipping address, and can be used to display subtotals of their OrderItem amounts.

All OrderItems in an Order are associated with a single currency.

The total monetary amounts calculated for the discounts, shipping charges, and taxes for the OrderItems in the Order are stored with the Order.

Calculation framework overview

There are five major components to the WebSphere Commerce calculation framework. They are:

Calculation methods
Calculation methods reference task commands to implement the parts of the calculation framework. Calculation methods are covered in more detail in Chapter 2, "Calculation methods" on page 5.

Calculation usages
Calculation usages are the categories of calculations that are performed using the calculation framework. Examples of calculation usages include discounts and sales tax. Calculation usages are covered in more detail in Chapter 3, "Calculation usages" on page 13.

Calculation codes
Calculation codes indicate the calculations to be performed for OrderItems. Calculation codes are covered in more detail in Chapter 4, "Calculation codes" on page 23.

Calculation rules
Calculation rules calculate the monetary amounts for the calculation codes associated with an OrderItem. Calculation rules are covered in more detail in Chapter 5, "Calculation rules" on page 35.

Calculation scales and calculation ranges
Calculation scales allow a calculation rule to determine monetary amounts in a way similar to looking up a value from a table. Calculation ranges define the relationship between a look–up number and value to be located for a calculation scale. Calculation scales and calculation ranges are covered in more detail in Chapter 6, "Calculation scales and calculation ranges" on page 47.

Some of the major relationships between the calculation framework components are shown in the following UML object model diagram. Calculation methods have been omitted to make the illustration simpler.
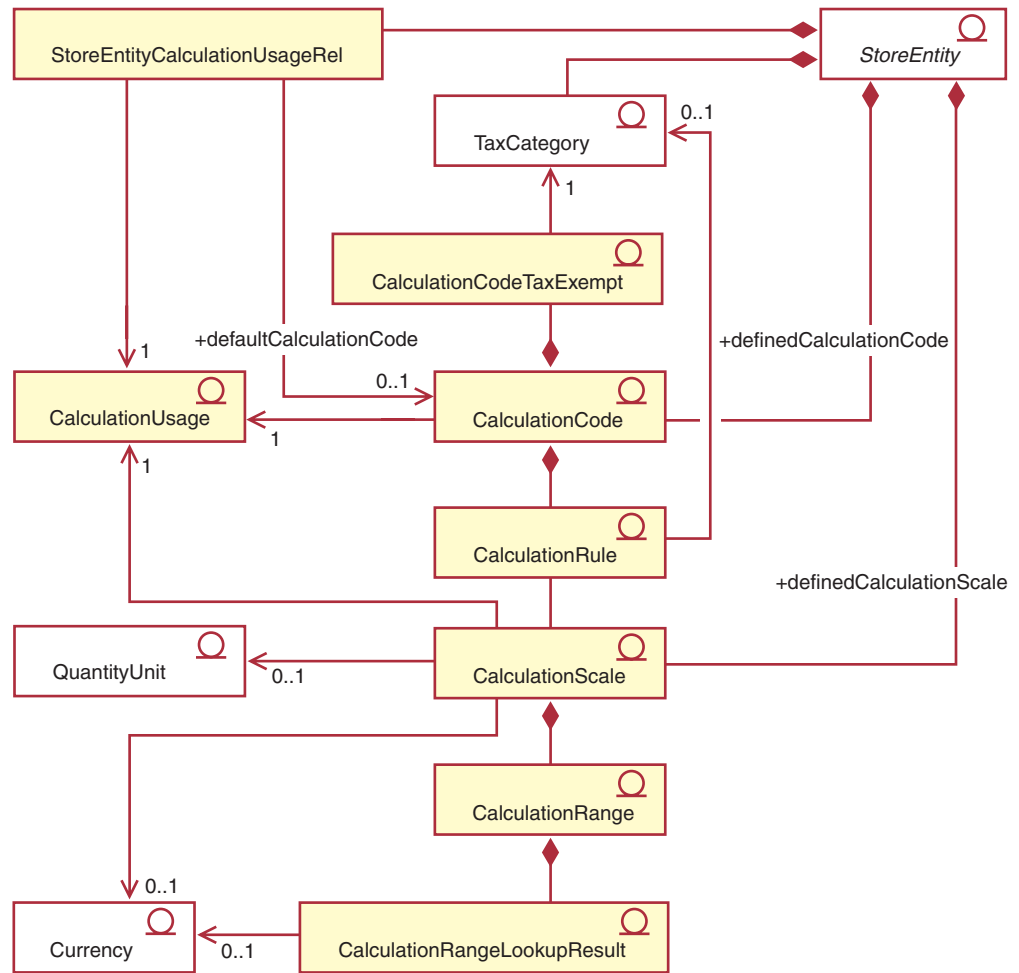
Figure 1. UML object model showing major relationships between calculation framework components (calculation methods omitted)

For more information on the conventions used in this diagram, see Appendix B, "UML legend" on page 107.

Notes:

1. StoreEntityCalculationUsageRel describes the high-level behavior of a calculation usage within a store or store group.
2. The relationship indicated between CalculationRule and TaxCategory only applies for sales tax and shipping tax calculation usages.
3. Additional UML object model diagrams for the calculation framework are in Appendix A, "Additional UML object model diagrams" on page 101. The UML object model diagrams are also available in the online information.

Chapter 2. Calculation methods

A calculation method implements a piece of the calculation framework. Different calculation method classes are used for the different operations that are required to complete a calculation within the calculation framework.

Using a number of calculation methods to perform calculations instead of one large calculation method makes customizing calculations much simpler. Customizing a calculation can often only require modifying or overriding one or two of the calculation methods involved in performing the calculation.

Calculation methods use the information that is part of calculation usages, calculation codes, and calculation scales to determine monetary amounts for OrderItems. Calculation methods are categorized by the task they perform within the calculation framework and the part of the calculation framework the calculation method belongs to. The following tasks are performed as part of the calculation framework:

Application
> Application involves persisting the monetary amounts calculated for OrderItems to the WebSphere Commerce database so that the monetary amounts can be used in later calculations.

Initialization
> Initialization involves resetting any variables used in calculations and clearing any previously calculated results.

Combination
> Combination involves determining the relationships between OrderItems and calculation codes, calculation rules, or calculation scales. Combination can also involve determining the order in which calculation codes are applied to OrderItems.

Qualification
> Qualification involves restricting the set of applicable calculation codes, calculation rules, or calculation scales using some criteria that exists outside of the calculation framework. For example, certain calculation codes may apply only to customers belonging to a specific member group. Qualification is a sub-task of combination.

Calculation
> Calculation involves the determination of a monetary amount that applies to OrderItems.

Look–up
> Look–up involves using one value to determine another value.

Summarization
> Summarization involves determining totals of the monetary amounts for OrderItems. An example of summarization is producing the total taxes for an order by adding the taxes calculated for each OrderItem in the order.

Finalization
> Finalization involves any processing that occurs after application. An example of finalization is marking any coupons used in an order to prevent the coupons from being used in other orders.

Different methods are used for applying, combining, finalizing, qualifying, and summarizing different pieces of the calculation framework. The calculation methods are assigned a subclass to categorize how the calculation method will be used. WebSphere Commerce provides the following calculation method subclasses, listed in the order they are used within the calculation framework:

- InitializeCalculationUsage
- ApplyCalculationUsage
- CalculationCodeApply
- CalculationCodeCombine
- CalculationCodeQualify
- CalculationCodeCalculate
- CalculationRuleCombine
- CalculationRuleQualify
- CalculationRuleCalculate
- MonetaryCalculationScaleLookup
- QuantityCalculationScaleLookup
- CalculationRange
- SummarizeCalculationUsage
- FinalizeCalculationUsage

The subclass of a calculation method indicates the interface from which its corresponding command extends. The following table shows the calculation method subclasses and their related interfaces

Table 1. Calculation method subclasses and their related interfaces

Calculation method subclass	Interface
InitializeCalculationUsage	com.ibm.commerce.order.calculation.InitializeCalculationUsageCmd
ApplyCalculationUsage	com.ibm.commerce.order.calculation.ApplyCalculationUsageCmd
CalculationCodeApply	com.ibm.commerce.order.calculation.CalculationCodeApplyCmd
CalculationCodeCombine	com.ibm.commerce.order.calculation.CalculationCodeCombineCmd
CalculationCodeQualify	com.ibm.commerce.order.calculation.CalculationCodeQualifyCmd
CalculationCodeCalculate	com.ibm.commerce.order.calculation.CalculationCodeCalculateCmd
CalculationRuleCombine	com.ibm.commerce.order.calculation.CalculationRuleCombineCmd
CalculationRuleQualify	com.ibm.commerce.order.calculation.CalculationRuleQualifyCmd
CalculationRuleCalculate	com.ibm.commerce.order.calculation.CalculationRuleCalculateCmd
MonetaryCalculationScaleLookup	com.ibm.commerce.order.calculation.CalculationScaleLookupCmd
QuantityCalculationScaleLookup	com.ibm.commerce.order.calculation.CalculationScaleLookupCmd
CalculationRange	com.ibm.commerce.order.calculation.CalculationRangeCmd
SummarizeCalculationUsage	com.ibm.commerce.order.calculation.SummarizeCalculationUsageCmd
FinalizeCalculationUsage	com.ibm.commerce.order.calculation.FinalizeCalculationUsageCmd

The Java classes that make up the calculation framework are part of the com.ibm.commerce.order.calculation package. See the online API documentation for more information on this Java package.

Database tables for calculation methods

Calculation methods are defined in the CALMETHOD database table. The CALMETHOD database table contains the following information about calculation methods:

Calculation method ID (CALMETHOD_ID)
> This is a unique, numeric identifier for the calculation method.

Store entity ID (STOREENT_ID)
> This is the identifier for the store or store group to which the calculation method belongs.

Calculation usage ID (CALUSAGE_ID)
> This is the calculation usage to which the calculation method applies. For more information on calculation usages, see Chapter 3, "Calculation usages" on page 13.

Subclass (SUBCLASS)
> This is a number that indicates the subclass of the calculation method. The following table shows the number assigned each subclass:

Table 2. Available subclasses

Subclass number	Subclass
1	CalculationCodeCombine
2	CalculationCodeQualify
3	CalculationCodeCalculate
4	CalculationCodeApply
5	CalculationRuleCombine
6	CalculationRuleQualify
7	CalculationRuleCalculate
8	QuantityCalculationScaleLookup
9	MonetaryCalculationScaleLookup
10	CalculationRange
11	InitializeCalculationUsage
12	ApplyCalculationUsage
13	SummarizeCalculationUsage
14	FinalizeCalculationUsage

Java interface name (TASKNAME)
> This is the complete name of the Java interface for a calculation method. For example, com.ibm.commerce.order.utils.CalculationCodeCodeCombineCmd. The same Java interface can be used for multiple calculation methods.

Calculation method data model diagram

The following data model diagram provides an illustration of the relationships between the database tables used to define calculation methods.
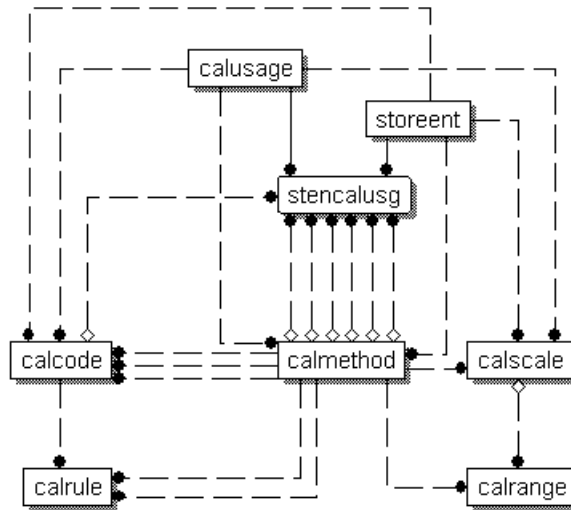
Figure 2. Calculation method data model diagram

For more information on the conventions used in this diagram, see Appendix C, "Data model legend" on page 109.

For details on the database tables in the diagram and other columns of the CALMETHOD database table, refer to the database schema documentation in the online information.

How calculation methods work

Various calculation methods are called when WebSphere Commerce calculates monetary amounts. Each subclass of calculation method is used for the other components of the calculation framework. The subclasses of calculation methods are covered in the other chapters of this book as follows:

Chapter 3, "Calculation usages" on page 13

InitializeCalculationUsage
> This calculation method can be called by the OrderPrepare command to initialize the calculation of monetary amounts. An example of initialization is removing previously calculated values.

ApplyCalculationUsage
> This calculation method can be called by the OrderPrepare command to calculate monetary amounts and apply them to an order. The ApplyCalculationUsage calculation method can also be called by data beans to calculate monetary amounts for display purposes. Examples of calculated amounts include order adjustments, shipping charges, and tax amounts. The default implementations of this calculation method provided with WebSphere Commerce call the CalculationCodeCombine, CalculationCodeCalculate, and CalculationCodeApply calculation methods.

SummarizeCalculationUsage
> This calculation method can be called by the OrderPrepare

command to summarize calculated values and apply them to an order. Examples include populating SubOrderAdjustments or SubOrder shipping charges or tax amounts.

FinalizeCalculationUsage
This calculation method can be called by the OrderProcess command to mark consumed resources, such as coupons, as no longer available.

Chapter 4, "Calculation codes" on page 23

CalculationCodeApply
This calculation method can be called by the ApplyCalculationUsage calculation method to apply the calculated values for a list of OrderItems to an order.

CalculationCodeCombine
This calculation method can be called by the ApplyCalculationUsage calculation method to identify relationships between OrderItems and calculation codes. The default implementations of the CalculationCodeCombine calculation method provided with WebSphere Commerce use the CATENCALCD, CATGPCALCD, ORDICALCD, and ORDCALCD database tables, and call the CalculationCodeQualify calculation method.

CalculationCodeQualify
This calculation method can be called by the CalculationCodeCombine calculation method to determine if a calculation code applies to a list of OrderItems.

CalculationCodeCalculate
This calculation method can be called by the ApplyCalculationUsage calculation method to calculate values for a list of OrderItems for a particular CalculationCode.

Chapter 5, "Calculation rules" on page 35

CalculationRuleCombine
This calculation method can be called by the CalculationCodeCalculate calculation method to determine a list of calculation rules and associated OrderItems that should be applied for a calculation code. The default implementations of the CalculationRuleCombine calculation method provided with WebSphere Commerce call the CalculationRuleQualify calculation method and the CalculationRuleCalculate calculation method to find the lowest values for each allowable combination of calculation rules.

CalculationRuleQualify
This calculation method can be called by the CalculationRuleCombine calculation method to determine to which of the OrderItems in a list a calculation rule applies.

CalculationRuleCalculate
This calculation method can be called by the CalculationRuleCombine calculation method to calculate values for a list of OrderItems.

Chapter 6, "Calculation scales and calculation ranges" on page 47

MonetaryCalculationScaleLookup
This calculation method can be called by the
CalculationRuleCalculate calculation method to determine how a
monetary–based calculation scale can be used to determine a
calculation range for a list of OrderItems.

QuantityCalculationScaleLookup
This calculation method can be called by the
CalculationRuleCalculate calculation method to determine how a
quantity–based calculation scale can be used to determine a
calculation range for a list of OrderItems.

CalculationRange
This calculation method can be called by the
CalculationRuleCalculate calculation method to determine a
calculated value from a calculation range look–up result.

General flow of calculation methods

The general flow of calculation methods, when they are used as part of the order
process, is as follows:

1. InitializeCalculationUsage calculation method
2. ApplyCalculationUsage calculation method calls:
   a. CalculationCodeCombine calculation method calls:
      1) CalculationCodeQualify calculation method
   b. CalculationCodeCalculate calculation method calls:
      1) CalculationRuleCombine calculation method calls:
         a) CalculationRuleQualify calculation method
         b) CalculationRuleCalculate calculation method calls:
            i. CalculationScaleLookup calculation method
            ii. CalculationRange calculation method
   c. CalculationCodeApply calculation method
3. SummarizeCalculationUsage calculation method
4. FinalizeCalculationUsage calculation method

The ApplyCalculationUsage calculation method can also be called by data beans to
calculate monetary amounts used for display purposes outside of order processing.

The following figure illustrates the flow of calculation methods called by the
ApplyCalculationUsage calculation method:
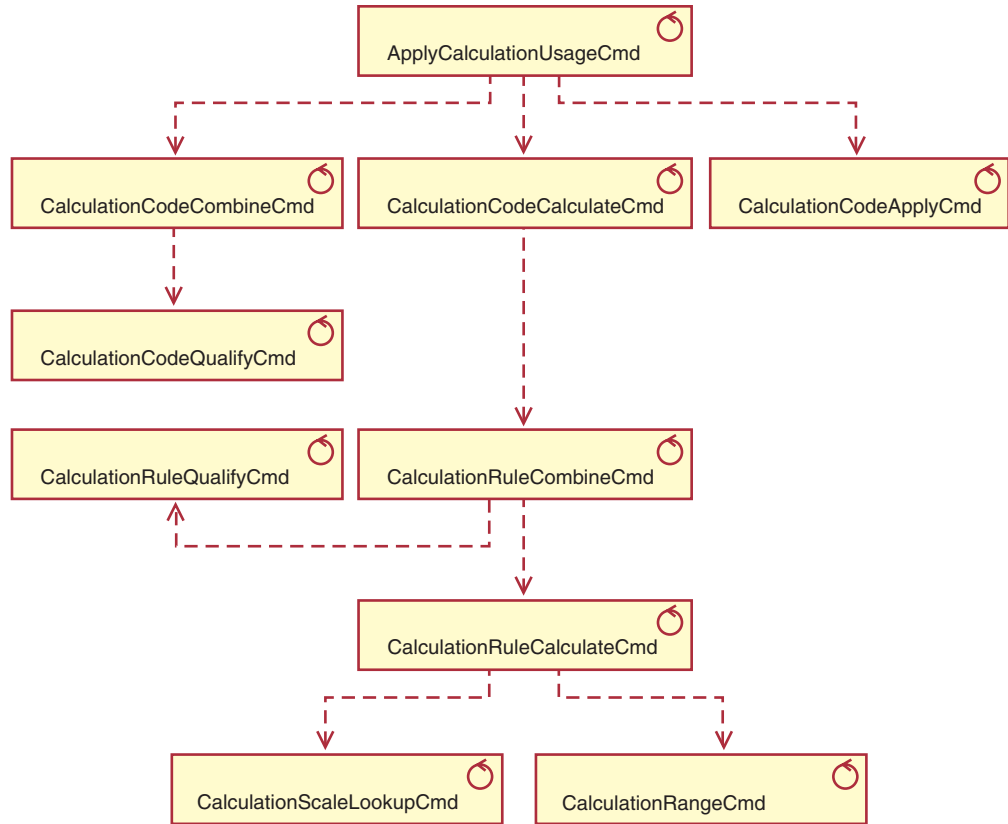
Figure 3. Calculation methods called by the ApplyCalculationUsage calculation method

For information on the conventions used in this diagram, see Appendix B, "UML legend" on page 107.

Chapter 3. Calculation usages

The WebSphere Commerce calculation framework is a generic framework for calculations. When implementing a calculation using the calculation framework, the first thing that needs to be defined is the type of calculation to be performed. The types of calculations the calculation framework performs are called *calculation usages*. Calculation usages are assigned a calculation usage code.

WebSphere Commerce provides five predefined calculation usages. The predefined usages are as follows:

Table 3. Predefined calculation usages

Calculation usage	Calculation usage code
Discount	-1
Shipping	-2
Sales tax	-3
Shipping tax	-4
Coupon	-5

WebSphere Commerce performs all calculations for one calculation usage at a time. The order of calculation usages is stored in the SEQUENCE column of the STENCALUSG database table. The entries in this table are initially populated with information from the language-independent bootstrap file: `wcs.bootstrap.xml`. For more information on bootstrap files, see "Bootstrap files" on page 111.

The default order in which calculation usages are processed, as defined in `wcs.bootstrap.xml`, is as follows:

1. Coupon
2. Discount
3. Shipping
4. Sales tax
5. Shipping tax

Database tables for calculation usage

Information about calculation usages is stored in the following database tables:

CALUSAGE
> This table contains a unique identifier for each calculation usage and a description of the calculation usage.
>
> Values in this table are defined in the language–dependent bootstrap file. For more information on bootstrap files, see "Bootstrap files" on page 111.

STENCALUSG
> This table defines the high–level behavior of a calculation usage for a store or group of stores. If the implementation for a store is not defined, the implementation for the store group that the store belongs to will be used. If different implementations are defined for both the store and the store group, the implementation for the store is used.

This database table is initially populated from the language–independent bootstrap file. For more information on bootstrap files, see "Bootstrap files" on page 111.

The following sections discuss key calculation usage attributes that are stored in some of the database tables. For details on all the columns of the calculation usage database tables and other database tables, refer to the database schema documentation in the online help. All of the attributes discussed in the following sections are required unless they are marked "optional."

CALUSAGE database table

The CALUSAGE database table contains the following information about a calculation usage:

Calculation usage ID (CALUSAGE_ID)
This is a numeric, unique identifier for the calculation usage.

Calculation usage description (DESCRIPTION) (optional)
This is a description of the calculation usage

STENCALUSG database table

The STENCALUSG database table contains the information relating a store or store group to a calculation usage as follows:

Store or store group identifier (STOREENT_ID)
This is the store or the store group for which the calculation usage is implemented.

Calculation usage identifier (CALUSAGE_ID)
This identifies the calculation usage being implemented for the store or store group.

Sequence (SEQUENCE)
Calculation usages are processed in ascending order based on this column.

Usage (USAGEFLAG)
This is a bit flag that controls whether a calculation usage is enabled and how WebSphere Commerce handles the results from an enabled calculation usage. Valid values for this attribute are as follows:

0 (default)
This calculation usage is disabled in this store or store group.

1   This calculation usage is enabled and may return a value for an OrderItem. If no value is returned for an OrderItem, the value is assumed to be zero.

2   This calculation usage is enabled and must always return a value for an OrderItem. If no value is returned for an OrderItem, an ECApplicationException is thrown.

InitializeCalculationUsage calculation method (CALMETHOD_ID_INI) (optional)
This is the calculation method used to initialize calculations. Initializing calculations involves setting any initial values required to perform a calculation and clearing any previously calculated results.

ApplyCalculationUsage calculation method (CALMETHOD_ID_APP) (optional)
This is the calculation method used to apply the calculation usage.

Applying a calculation usage involves persisting the calculated monetary amounts to the WebSphere Commerce database.

SummarizeCalculationUsage calculation method (CALMETHOD_ID_SUM) (optional)

This is the calculation method used to summarize the calculation usage. Summarizing a calculation usage involves determining totals for order and sub–orders based on the monetary amounts calculated for the OrderItems in the order.

FinalizeCalculationUsage calculation method (CALMETHOD_ID_FIN) (optional)

This is the calculation method used to finalize the calculation usage. Finalizing a calculation involves any processing that must be completed after the calculations are applied. An example of finalizing a calculation usage is marking any coupons used in an order to prevent the coupons from being applied to other orders.

Default calculation code (CALCODE_ID) (optional)

If an OrderItem has no calculation code attached to it for this calculation usage, then the default calculation code is used.

Calculation usage data model diagram

The following data model diagram provides an illustration of the relationships between the database tables used to define calculation usages.
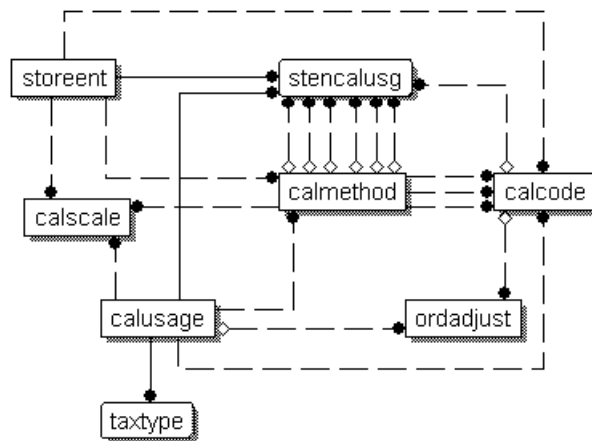
Figure 4. Calculation usage data model diagram

For details on the database tables in the diagram refer to the database schema documentation in the online information.

Calculation methods for calculation usages

There are four calculation method subclasses associated with calculation usages:

InitializeCalculationUsage

InitializeCalculationUsage calculation methods are called by the OrderPrepare command for each calculation usage specified in the USAGEFLAGS column of the STENCALUSG database table prior to calling the ApplyCalculationUsage calculation method.

ApplyCalculationUsage

ApplyCalculationUsage calculation methods are called by the OrderPrepard command for each calculation usage specified by USAGEFLAGS column of the STENCALUSG database table. These methods may also be called from data beans or other task commands to calculate amounts for items not necessarily in an Order. For example, the GetReturnTaxes task command calls the ApplyCalculationUsage method for the sales tax calculation usage.

SummarizeCalculationUsage

SummarizeCalculationUsage calculation methods are called by the OrderPrepare command for each calculation usage specified in USAGEFLAGS column of the STENCALUSG database table after calling the ApplyCalculationUsage calculation methods. SummarizeCalculationUsage calculation methods summarize the calculated amounts. For example, the SummarizeCalculationUsage method for the sales tax calculation usage populates the SUBORDTAX database table and the SUBORDERS.TOTALTAX columns. The SUBORDTAX database table contains tax information for the OrderItems in a SubOrder.

FinalizeCalculationUsage

FinalizeCalculationUsage calculation methods are called by the ProcessOrder task command for each calculation usage specified in USAGEFLAGS column of the STENCALUSG database table. For example, the FinalizeCalculationUsage method for the coupon calculation usage marks the coupons applied to an order to prevent the coupons from being applied to a different order.

The following section discusses these calculation method subclasses and the default implementations provided with WebSphere Commerce. Additional information on the interfaces and commands that implement the calculation method subclasses can be found in the API documentation in the online information.

InitializeCalculationUsage calculation methods

InitializeCalculationUsage calculation methods do any processing required before monetary amounts for OrderItems are calculated. For example, an InitializeCalculationUsage calculation method can remove previously calculated values from the OrderItems in the order.

Any commands that will be used as an InitializeCalculationUsage calculation method subclass must implement the InitializeCalculationUsageCmd interface. For more information on the InitializeCalculationUsageCmd interface, see the API documentation in the online information.

The following InitializeCalcluationUsage calculation methods are provided with WebSphere Commerce:

- InitializeAdjustmentCmdImpl
- InitializeCouponUsageCmdImpl
- InitializeSalesTaxCmdImpl

- InitializeShippingCmdImpl
- InitializeShippingTaxCmdImpl

InitializeAdjustmentCmdImpl
This command implements the InitializeAdjustmentCmd interface that extends the InitializeCalculationUsageCmd interface.

This is the discount–specific command used to initialize values for discount calculations.

InitializeCouponUsageCmdImpl
This command implements the InitializeCouponUsageCmd interface that extends the InitializeCalculationUsageCmd interface.

This is a coupon specific command.

InitializeSalesTaxCmdImpl
This command implements the SalesTaxCmd interface that extends the InitializeCalculationUsageCmd interface.

This is the sales tax specific command used to initialize values for sales tax calculations.

InitializeShippingCmdImpl
This command implements the InitializeShippingCmd interface that extends the InitializeCalculationUsageCmd interface.

This is the shipping specific command used to initialize values for shipping charge calculations.

InitializeShippingTaxCmdImpl
This command implements the InitializeShippingTaxCmd interface that extends the InitializeCalculationUsageCmd interface.

This is the shipping tax specific command used to initialize values for shipping tax calculations.

ApplyCalculationUsage calculation methods

ApplyCalculationUsage calculation methods are used for both order processing and product display purposes. During order processing, an ApplyCalculationUsage calculation method is called by the OrderPrepare command to calculate the monetary amounts for a list of OrderItems. A monetary amount is applied to each OrderItem so the monetary amount is available for later calculations.

For display purposes, ApplyCalculationUsage calculation methods are called by data beans to calculate monetary amounts. For example, taxes for a product can be displayed as part of the product's display page. The data bean calling the ApplyCalculationUsage calculation method must perform any initialization that would normally be performed by the InitializeCalculationUsage calculation method for the calculation usage being processed for the data bean. The data bean does not have access to the InitializeCalculationUsage, SummarizeCalculationUsage, or FinalizeCalculationUsage calculation methods.

Additionally, the The GetReturnTaxes task command also calls the ApplyCalculationUsage method for the sales tax calculation usage, without calling the Initialize, Summarize, or Finalize methods.

Any commands that will be used as an ApplyCalculationUsage calculation method subclass must implement the ApplyCalculationUsageCmd interface. For more information on the ApplyCalculationUsageCmd interface, see the API documentation in the online information.

The following ApplyCalculationUsage calculation methods are provided with WebSphere Commerce:

- ApplyCalculationUsageCmdImpl
- ApplyCalculationUsageTIKCmdImpl
- ApplyCouponUsageCmdImpl
- ApplyShippingCmdImpl

ApplyCalculationUsageCmdImpl

This is the default implementation of the ApplyCalculationUsage command.

This command implementation does the following:

1. Call the CalculationCodeCombine calculation method to get a list of calculation codes. For each CalculationCode in the list, the CalculaitonCodeCombine calculation method also provides a list of the OrderItems to be included in the calculation.

2. For each calculation code in the list, ApplyCalculationUsageCmdImpl does the following:

   a. Call the CalculationCodeCalculate calculation method, passing it the associated list of OrderItems. The CalculationCodeCalculate calculation method returns a list of calculated monetary amounts for each OrderItem. The monetary amounts may be categorized by tax category.

   b. Call the CalculationCodeApply calculation method, passing it the calculation code, a list of OrderItems, and the monetary amount associated with each OrderItem.

ApplyCalculationUsageTIKCmdImpl

This command implements the ApplyCalculationUsageTIKCmd interface, which extends the ApplyCalculationUsageCmd interface.

The ApplyCalculationUsageTIKCmd interface and the ApplyCalculationUsageTIKCmdImpl command are provided as an example to invoke a set of interfaces supplied in the Tax Integration Interface Kit. If WebSphere Commerce has been configured to use this interface, the OrderPrepare command will call ApplyCalculationUsageTIKCmd which in turn invokes the TaxOrderCmd task command provided in the Tax Integration Interface Kit.

For more information on the ApplyCalculationUsageTIKCmd interface and the ApplyCalculationUsageTIKCmdImpl command, see the API documentation in the online information.

ApplyCouponUsageCmdImpl

This command implements the ApplyCouponUsageCmd interface that extends the ApplyCalculationUsageCmd interface.

This command adds the discount from a coupon promotion to the adjustments in the ORDERS and ORDERITEMS database tables. The command is invoked when a customer asks to redeem coupons before preparing an order.

The coupons to apply to an order are located in the ORCPMAP database table. This command finds the coupons to apply to an order in the ORCPMAP database

table and then removes the coupons from the ORCPMAP database table and CPITMAP database table. The found coupons are checked for expiry by calling CheckValidityTaskCmd command. Coupons that are not expired are checked for applicability by calling the CheckApplicabilityTaskCmd command. The applicable coupons are called by the CalculateDiscountAmountCmd task command and the discount amount in the current order is incorporated.

For more information on coupon promotions, see the online information.

ApplyShippingCmdImpl

This command is the default implementation of the ApplyShippingCmd interface. The ApplyShippingCmd interface extends the ApplyCalculationUsageCmd interface.

This is a shipping specific version of the ApplyCalculationUsageCmdImpl command.

SummarizeCalculationUsage calculation methods

This calculation method subclass summarizes the results of calculations for display purposes.

Any commands that will be used as a SummarizeCalculationUsage calculation method subclass must implement the SummarizeCalculationUsageCmd interface. For more information on the SummarizeCalculationUsageCmd interface, see the API documentation in the online information.

The following SummarizeCalculationUsage calculation methods are provided with WebSphere Commerce:

- SummarizeAdjustmentCmdImpl
- SummarizeCouponUsageCmdImpl
- SummarizeSalesTaxCmdImpl
- SummarizeShippingCmdImpl
- SummarizeShippingTaxCmdImpl

SummarizeAdjustmentCmdImpl

This command implements the SummarizeAdjustmentCmd interface that extends the SummarizeCalculationUsageCmd interface.

This is the discount specific command used to summarize discount calculations.

SummarizeCouponUsageCmdImpl

This command implements the SummarizeCouponUsageCmd interface that extends the SummarizeCalculationUsageCmd interface.

This is the coupon specific command used to summarize coupon calculations.

SummarizeSalesTaxCmdImpl

This command implements the SummarizeSalesTaxCmd interface that extends the SummarizeCalculationUsageCmd interface.

This is the sales tax specific command used to summarize sales tax calculations.

SummarizeShippingCmdImpl

This command implements the SummarizeShippingCmd interface that extends the SummarizeCalculationUsageCmd interface.

This is the shipping specific command used to summarize shipping charge calculations.

SummarizeShippingTaxCmdImpl
This command implements the SummarizeShippingTaxCmd interface that extends the SummarizeCalculationUsageCmd interface.

This is the shipping tax specific command used to summarize shipping tax calculations.

FinalizeCalculationUsage calculation methods

This calculation usage subclass is used to do any processing required after the Order has been processed by the ProcessOrder task command.

Any commands that will be used as a FinalizeCalculationUsage calculation method subclass must implement the FinalizeCalculationUsageCmd interface. For more information on the FinalizeCalculationUsageCmd interface, see the API documentation in the online information.

The FinalizeCouponUsageCmdImpl is the only FinalizeCalculationUsage calculation method provided with WebSphere Commerce.

FinalizeCouponUsageCmdImpl
This command implements the FinalizeCouponUsageCmd interface that extends the FinalizeCalculationUsageCmd interface.

The FinalizeCouponUsageCmdImpl command marks coupons used in the current order to prevent the coupons from being used in other orders.

How calculation usages work

Calculation usages are invoked by the OrderPrepare command. The OrderPrepare command creates a list of OrderItems for which monetary amounts will be calculated. The applicable calculation usages for the store or store group to which the order belongs are found and processed according to the sequence defined for them in the STENCALUSG database table.

The OrderPrepare command processes the calculation usages as follows:
1. All calculation usages are initialized using the InitializeCalculationUsage calculation methods referenced in the CALMETHOD_ID_INI column of the STENCALUSG database table.
2. All calculation usages are applied using the ApplyCalculationUsage calculation methods referenced in the CALMETHOD_ID_APP column of the STENCALUSG database table. For more information on how calculation methods are applied, see "How calculation usages are applied" on page 21.
3. All calculation usages are summarized using the SummarizeCalculationUsage calculation methods referenced in the CALMETHOD_ID_SUM column of the STENCALUSG database table.

After the OrderPrepare controller command completes, the OrderProcess controller command is called. The OrderProcess controller command finalizes all calculation usages by calling the FinalizeCalculationUsage calculation methods referenced in the CALMETHOD_ID_FIN column of the STENCALUSG database table entry for each calculation usage.

How calculation usages are applied

When a calculation usage is applied the following steps occur:

1. The ApplyCalculationUsage calculation method calls a CalculationCodeCombine calculation method. The CalculationCodeCombine calculation method returns a list. Each item in the returned list consists of a calculation code and the OrderItems in the order to which the calculation code applies.

2. The ApplyCalculationUsage calculation method calls a CalculationCodeCalculate calculation method for each item in the list returned by the CalculationCodeCombine calculation method. The CalculationCodeCalculate calculation method returns a list. Each item in the return list consists of each OrderItem and the monetary amount associated with the OrderItem for the calculation usage. For tax calculation methods, there may be multiple monetary amounts for different taxes that apply to the OrderItem. In this case, the tax categories are returned as part of the list as well.

3. The ApplyCalculationUsage calculation method calls a CalculationCodeApply calculation method for each group of OrderItems.

For descriptions of calculation codes and associated calculation methods, including the CalculationCodeApply calculation method, see Chapter 4, "Calculation codes" on page 23.

Chapter 4. Calculation codes

Items for sale in a retail store often have attached price tags indicating information other than the price of the item. For example, if the item is on sale, the price tag indicates the percentage discount to apply to the item to calculate the sale price. Similarly, WebSphere Commerce attaches calculation codes to OrderItems to indicate the calculations to be performed for the OrderItem. The following figure shows a representation of price tags in a retail store compared with calculation codes in WebSphere Commerce.
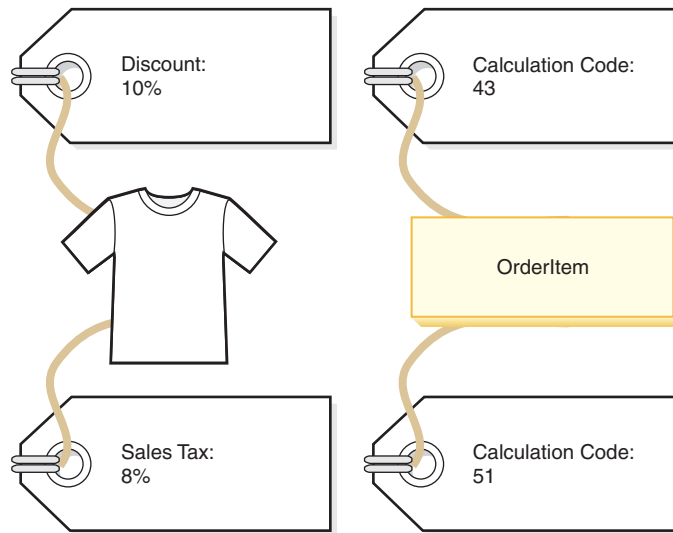
Figure 5. Calculation codes indicate the calculations that must performed.

In the same way price tags are attached to the items for sale in a retail store, calculation codes are considered to be attached to OrderItems. The attachment of a calculation code can be direct or indirect, depending on where the calculation code is specified within WebSphere Commerce.

Direct attachment of a calculation code to an OrderItem is similar to a cashier providing a one-time discount on one item or all items being purchased — the calculation code is specified as part of the order or as part of the OrderItem. When a calculation code is specified as part of an order, it is attached to all of the OrderItems that make up the order. Information relating a calculation code to an order is stored in the ORDCALCD database table. Information relating a calculation code to an OrderItem is store in the ORDICALCD database table.

Indirect attachment of a calculation code is similar to posting a sign in a retail store indicating which items in the store are on sale — the information about the discount is not directly on the price tags of the individual items, but the cashier knows to apply the discount when calculating the cost of an order. Indirect attachment of a calculation code in WebSphere Commerce occurs when the calculation code is specified as part of a catalog entry or catalog group.

Specifying a calculation code as attached to a catalog entry is similar to the sign in the retail store indicating that a discount applies to a particular product. Specifying a calculation code as part of a catalog group is similar to the sign in the retail store

indicating that a discount applies to all products in a particular department. Similarly, when a calculation code is specified as part of a catalog group, the calculation code applies to all catalog entries in the catalog group. Information relating a calculation code to a catalog entry is stored in the CATENCALCD database table. Information relating a calculation code to a catalog group is stored in the CATGPCALCD database table. The CATENCALCD database table is also used to attach a calculation code to all catalog entries.

Calculation codes can also be implicitly attached to order items by specifying the default calculation code for a calculation usage in the STENCALUSG database table. The default calculation code is used for those OrderItems in an order that do not have any directly or indirectly attached calculation codes of a particular calculation usage. The default calculation code for a particular calculation usage can be specified at the store level and at the store group level. The default calculation code specified at the store group level is used only when the store level default calculation code is not specified.

While attachment is defined outside of a calculation code, the following properties are defined as part of a calculation code:

- A calculation usage. This specifies the calculation type for which a calculation code can be used. Calculation usages are covered in Chapter 3, "Calculation usages" on page 13.
- Calculation methods used to qualify, calculate, and apply the calculation code.
- A flag indicating if a calculation code must be qualified or not.
- A flag indicating if the calculation code is active, inactive, or marked for deletion.
- The method of grouping OrderItems for calculations.
- A sequence number. The sequence number is used to determine the order of calculations when multiple calculation codes of the same calculation usage apply to an OrderItem.
- A time range during which the calculation code is in effect.

These properties are defined in the CALCODE database table. The CALCODE database table and other database tables that affect the way calculation codes are used are discussed in the next section.

Database tables for calculation codes

In WebSphere Commerce, information about calculation codes is stored in the following database tables:

CALCODE
> This is the main database table that defines calculation codes.

CALCODEDSC
> This database table contains national language descriptions of a calculation code, allowing one calculation code to have a description in multiple languages.

CALCODEMGP
> This database table associates a calculation code with a member group, restricting the use of a calculation code to members of a certain member group.

CALCODTXEX
> This database table associates a calculation code with a tax category,

indicating that the monetary amounts determined by the calculation code are exempt from the taxes associated with the tax category.

CATGPCALCD
> This database table attaches a calculation code with catalog group in a specific store and, optionally, for a specific trading agreement. An example of a trading agreement is a contract. Information about contracts and trading agreements is provided in the online information.
>
> All catalog entries belonging to the catalog group are associated with the calculation code assigned to the catalog group.

CATENCALCD
> This database table attaches a calculation code with one catalog entry or all catalog entries in a specific store and, optionally, for a specific trading agreement. An example of a trading agreement is a contract. Information about contracts and trading agreements is provided in the online information.

ORDCALCD
> This database table attaches a calculation code to an order. This table allows you to specify whether indirectly attached calculation codes should be ignored.

ORDICALCD
> The database table attaches a calculation code to an OrderItem. This table allows you to specify whether indirectly attached calculation codes should be ignored.

STENCALUSG
> This database table defines the default calculation code for a calculation usage in a store or group of stores. This table also specifies the CalculationCodeCombine calculation method subclass that determines how multiple calculation codes for an OrderItem are combined.

The following sections discuss key calculation code attributes that are stored in some of the database tables. For details on all the columns of the calculation code database tables, refer to the database schema documentation in the online information. All of the attributes discussed in the following section are required unless they are marked "optional."

CALCODE database table

The CALCODE database table contains the following information about a calculation code:

Calculation code ID (CALCODE_ID)
> This is a numeric, unique identifier for the calculation code.

Identifying string (CODE)
> This is a character string that uniquely identifies this calculation code, given a particular calculation usage and store or store group.

CalculatonCodeCalculate calculation method (CALMETHOD_ID)
> This is the identifier of the calculation method that calculates the monetary amounts for OrderItems for this calculation code.

CalculationCodeApply calculation method (CALMETHOD_ID_APP)
> This is the identifier of the calculation method that persists the calculated monetary amounts.

CalculationCodeQualify calculation method (CALMETHOD_ID_QFY)
This is the identifier of the calculation method responsible for determining which of the available calculation codes for the OrderItems should be used.

Calculation usage ID (CALUSAGE_ID)
This is the identifier of the calculation usage to which the calculation code applies.

Effective time range (STARTDATE and ENDDATE) (optional)
These two values define the time period the calculation rule is effective. If none of the values is defined, the calculation code is always in effect. If STARTDATE is defined and ENDDATE is not, the calculation code is always effective after the time and date defined in STARTDATE. If STARTDATE is not defined and ENDDATE is defined, the calculation code is effective immediately and expires on the date and time defined in ENDDATE.

The effective time range is checked by the CalculationCodeCombine calculation method.

Grouping method (GROUPBY)
This column is used to indicate how the CalculationCodeCombine calculation method will group OrderItems for performing calculations. The CalculationCodeCalculate and CalculationCodeApply calculation methods are used repeatedly, once for each group of OrderItems, to calculate and apply monetary amounts to the OrderItems in each group. The default grouping is no grouping — all OrderItems are placed in one group and the CalculationCodeCalculate and CalculationCodeApply calculation methods are called only once.

OrderItems can be grouped in the following ways:

By address
All OrderItems with the same shipping address are placed in the same group and the calculation code is evaluated once for each different shipping address.

By contract
All OrderItems with the same Contract are placed in the same group and the calculation code is evaluated once for each different contract.

By offer
All OrderItems with the same Offer are placed in the same group and the calculation code is evaluated once for each different offer.

By product
All OrderItems with the same parent product are placed in the same group and the calculation code is evaluated once for each different parent product.

These groups can also be combined so that OrderItems with the same catalog entries and the same Contracts are grouped together or OrderItems with the same Offer and the same Contract are grouped together. When combining groups, the calculation code is evaluated once for each different combination.

Sequence (SEQUENCE)
When multiple calculation codes with the same calculation usage are processed, they are processed in ascending order based on the numeric values of this property.

Store or store group identifier (STOREENT_ID)
> This is the store or the store group to which this calculation code belongs. A calculation code can only belong to one store or store group.

Tax code classification (TXCDCLASS_ID) (optional)
> If this calculation code is used to calculate taxes, then it can be grouped together with other tax calculation codes into a tax code classification.

Qualification flag (FLAGS)
> This flag allows you to specify whether a CalculationCodeQualify calculation method should be invoked. The default setting is that the CalculationCodeQualify calculation method will not be invoked.
>
> This flag does not affect whether the CalculationCodeCombine calculation method checks the effective time range of a calculation code; this check cannot be prevented.

STENCALUSG database table

The STENCALUSG database table contains the following information about calculation codes:

CalculationCodeCombine calculation method (ACTCC_CALMETHOD_ID) (optional)
> This is the identifier of the CalculationCodeCombine calculation method. If a CalculationCodeCombine calculation method is not specified for a store, then the CalculationCodeCombine calculation method defined for the store group is used.

The STENCALUSG database table also contains information about calculation usages. Calculation usages are covered in Chapter 3, "Calculation usages" on page 13.

Calculation code data model diagrams

The following data model diagrams provide illustrations of the relationships between the database tables used to define calculation codes, the database tables used to define direct attachment of calculation codes, and the database tables used to define indirect attachment of calculation codes.
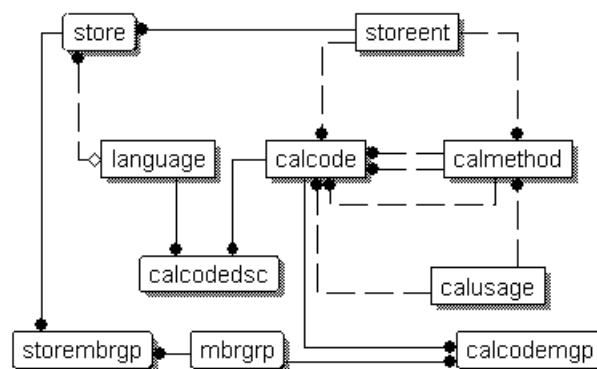
Figure 6. Calculation code data model diagram
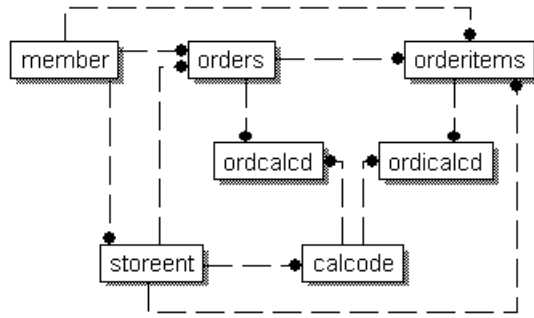
Figure 7. Calculation code direct attachment data model diagram
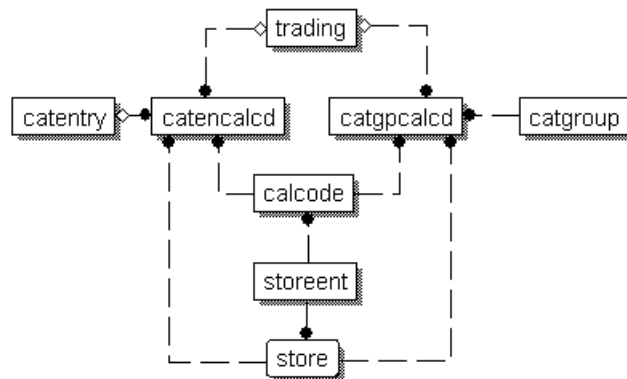
Figure 8. Calculation code indirect attachment data model diagram

> For information on the conventions used in these diagrams, see
> Appendix C, "Data model legend" on page 109.

For details on the database tables in the diagrams refer to the database schema documentation in the online information.

Calculation methods for calculation codes

There are four calculation method subclasses associated with calculation codes:

- CalculationCodeCombine
- CalculationCodeQualify
- CalculationCodeCalculate
- CalculationCodeApply

This section discusses these calculation method subclasses and the default implementations provided with WebSphere Commerce. Additional information on the interfaces and commands that implement the calculation method subclasses can be found in the API documentation in the online information.

CalculationCodeCombine calculation methods

CalculationCodeCombine calculation methods determine which calculation codes are applied to which OrderItems and the sequence in which the calculation codes are applied. For tax calculation usages, a list of tax categories can also be specified.

A CalculationCodeCombine calculation method returns a list where each item in the list consists of the following:

- A calculation code
- A list of OrderItems to which the calculation code applies
- A list of monetary amounts corresponding to each OrderItem. This list is only provided if the monetary amounts have previously been calculated by a CalculationCodeCalculate calculation method to prevent the CalculationCodeCalculate calculation method from being called unnecessarily.

For tax calculations, the monetary amount corresponding to each OrderItem may be separated into the amount for each tax category applicable to the OrderItem.

Any commands that will be used as a CalculationCodeCombine calculation method subclass must implement the CalculationCodeCombineCmd interface. For more information on the CalculationCodeCombineCmd interface, see the API documentation in the online information.

The following CalculationCodeCombine calculation methods are provided with WebSphere Commerce:

- CalculationCodeCombineCmdImpl
- TaxCalculationCodeCombineCmdImpl

CalculationCodeCombineCmdImpl

This is the default implementation of the CalculationCodeCombine calculation method.

This command does the following, given a calculation usage and a list of OrderItems:

1. Using the list of OrderItems, the CalculationCodeCombineCmdImpl command creates a list of all calculation codes attached to the OrderItems. Only calculation codes belonging to the calculation usage being processed are considered. Calculation codes directly attached to OrderItems are found first, followed by indirectly attached calculation codes. If no directly or indirectly attached calculation codes are found, the default calculation code is used.

2. For each calculation code found, the command does the following:

   a. If the qualification flag attribute of the calculation code is set to 1, then the CalculationCodeQualify calculation method is called . A calculation code and the list of OrderItems is passed to the CalculationCodeQualify calculation method to check which OrderItems belong to the calculation code. The CalculationCodeQualify calculation method returns a modified list of OrderItems. The modified list of OrderItems consists only of OrderItems to which the calculation code applies. OrderItems to which the calculation code does not apply have been removed.

   b. Groups the OrderItems according to the grouping method of the calculation code. A group consists of one or more OrderItems. For information on how OrderItems are grouped, see "CALCODE database table" on page 25.

3. The CalculationCodeCombineCmdImpl command returns a list where each list item consists of a calculation code and one or more groups of OrderItems. The returned list is sorted by the sequence attribute of the calculation code. The

calculation code with the lowest sequence attribute appears first on the list. If multiple calculation codes have the same sequence number, the calculation codes are also sorted by calculation code ID, with the lowest calculation code ID appearing first.

TaxCalculationCodeCombineCmdImpl

This command implements the TaxCalculationCodeCombineCmd interface that extends the CalculationCodeCombineCmd interface.

This command does the following, given a calculation usage and a list of OrderItems:

1. Using the list of OrderItems, the TaxCalculationCodeCombineCmdImpl command creates a list of all calculation codes attached to the OrderItems. Only calculation codes belonging to the tax calculation usage being processed are considered. Calculation codes directly attached to OrderItems are found first, followed by indirectly attached calculation codes. If no directly or indirectly attached calculation codes are found, the default calculation code is used.

   Important: If multiple calculation codes are found, only the calculation code with the highest sequence value is used.

2. For each calculation code found, the command does the following:
   a. If the qualification flag attribute of the calculation code is set to 1, then the CalculationCodeQualify calculation method is called. A calculation code and the list of OrderItems is passed to the CalculationCodeQualify calculation method to check which OrderItems belong to the calculation code. The CalculationCodeQualify calculation method returns a modified list of OrderItems. The modified list of OrderItems consists only of OrderItems to which the calculation code applies. OrderItems to which the calculation code does not apply have been removed.
   b. Groups the OrderItems according to the grouping method of the calculation code. A group consists of one or more OrderItems. For information on how OrderItems are grouped, see "CALCODE database table" on page 25.

3. Returns a list where each list item consists of a calculation code and one or more groups of OrderItems. The returned list is sorted by the sequence attribute of the calculation code. The calculation code with the lowest sequence attribute appears first on the list. If multiple calculation codes have the same sequence number, the calculation codes are also sorted by calculation code ID, with the lowest calculation code ID appearing first.

CalculationCodeQualify calculation methods

CalculationCodeQualify calculation methods check if a calculation code applies to a list of OrderItems. This calculation method returns a list of OrderItems to which a calculation code applies.

CalculationCodeQualify calculation methods are called only if the qualification flag attribute of the calculation code is set to 1.

Any commands used as a CalculationCodeQualify calculation method must implement the CalculationCodeQualifyCmd interface. For more information on the CalculationCodeQualifyCmd interface, see the API documentation in the online information.

The CalculationCodeQualifyCmdImpl command is provided with WebSphere Commerce and implements the CalculationCodeQualifyCmd interface.

CalculationCodeQualifyCmdImpl
This is the default implementation of the CalculationCodeQualify command.

This command returns OrderItems with a customer in one of the member groups associated with the calculation code and recognized by the store. If the customer is not in any of the member groups associated with the calculation code, this command returns null.

Calculation codes are associated with member groups in the CALCODEMGP database table and member groups are recognized by stores in the STOREMBRGP database table.

CalculationCodeCalculate calculation methods

CalculationCodeCalculate calculation methods calculate a monetary amount for each OrderItem when given a calculation code and a group of OrderItems.

If a CalculationCodeCalculate calculation method is provided with a list of tax categories in addition to the calculation code and group of OrderItems, monetary amounts for each tax category are calculated for each OrderItem in the group of OrderItems.

Any commands used as a CalculationCodeCalculate calculation method must implement the CalculationCodeCalculateCmd interface. For more information on the CalculationCodeCalculateCmd interface, see the API documentation in the online information.

The CalculationCodeCalculateCmdImpl command is provided with WebSphere Commerce and implements the CalculationCodeCalculateCmd interface.

CalculationCodeCalculateCmdImpl
This is the default implementation of the CalculationCodeCalculate command.

Give a calculation code, a group of OrderItems, and an optional list of tax categories, this command does the following:
1. Calls a CalculationRuleCombine calculation method to determine the list of calculation rules and the OrderItems associated with the calculation rules. The CalculationRuleCombine method returns a list of calculation rules, and for each rule returned, a list of associated OrderItems and a calculated monetary amount for each associated OrderItem. The monetary amounts are the result of evaluating the calculation rule.
2. Adds the monetary amounts for each OrderItem to calculate the total monetary amount for each OrderItem. If these calculations are being performed for a tax calculation usage, the monetary amounts for each tax category are also added to arrive at a total monetary amount for each tax category for each OrderItem.

Calculation rules and the associated calculation methods are covered in Chapter 5, "Calculation rules" on page 35.

CalculationCodeApply calculation methods

The CalculationCodeApply calculation method applies the calculated monetary amounts to the OrderItems. These amounts are then available to be used in later calculations.

Any commands used as a CalculationCodeApply calculation method must implement the CalculationCodeApplyCmd interface. For more information on the CalculationCodeQualifyCmd interface, see the API documentation in the online information.

The following commands implementing the CalculationCodeApplyCmd interface are provided with WebSphere Commerce:

- DiscountCalculationCodeApplyCmdImpl
- ShippingCalculationCodeApplyCmdImpl
- SalesTaxCalculationCodeApplyCmdImpl
- ShippingTaxCalculationCodeApplyCmdImpl

DiscountCalculationCodeApplyCmdImpl

This command implements the DiscountCalculationCodeApplyCmd interface that extends the CalculationCodeApplyCmd interface.

This command creates an OrderAdjustment object for the Order and creates an OrderItemAdjustment object for each of the OrderItems in the list using the specified monetary amounts. If the calculation code producing the monetary amounts is flagged as tax exempt, the OrderItemAdjustment objects are also flagged as tax exempt.

SalesTaxCalculationCodeApplyCmdImpl

This command implements the SalesTaxCalculationCodeApplyCmd interface that extends the CalculationCodeApplyCmd interface. It also extends the base class TaxCalculationCodeApplyCmdImpl.

This command adds the monetary amounts for each sales tax category to the taxes for the specified OrderItems.

ShippingCalculationCodeApplyCmdImpl

This command implements the ShippingCalculationCodeApplyCmd interface that extends the CalculationCodeApplyCmd interface.

This command adds the monetary amounts for the OrderItems to the shipping charges for the order.

ShippingTaxCalculationCodeApplyCmdImpl

This command implements the ShippingTaxCalculationCodeApplyCmd interface that extends the CalculationCodeApplyCmd interface. It also extends the base class TaxCalculationCodeApplyCmdImpl.

This command adds the monetary amounts for each shipping tax category to the taxes for the specified OrderItems.

How calculation codes work

Calculation code calculation methods are called from an ApplyCalculationUsage calculation method. The ApplyCalculationUsage calculation method does the following:

1. The ApplyCalculationUsage calculation method calls a CalculationCodeCombine calculation method. The CalculationCodeCombine calculation method returns a list. Each item in the returned list consists of a calculation code and the OrderItems in the order to which the calculation code applies.

2. The ApplyCalculationUsage calculation method calls a CalculationCodeCalculate calculation method for each OrderItem in the list returned by the CalculationCodeCombine calculation method.

If the CalculationCodeCalculate calculation method needs calculation rules to determine a monetary amount for each OrderItem in the list, the CalculationCodeCalculate calculation method calls a CalculationRuleCombine calculation method to evaluate the calculation rules and to determine which calculation rules should be applied to which of the OrderItems. For a description of calculation rules and associated calculation methods, see Chapter 5, "Calculation rules" on page 35.

The CalculationCodeCalculate calculation method returns a list. Each item in the list consists of an OrderItem and the monetary amount associated with the OrderItem for the calculation usage. For tax calculation methods, there may be multiple monetary amounts for different tax categories that apply to the OrderItem. In this case, the tax categories are returned as part of the list as well.

3. The ApplyCalculationUsage calculation method calls a CalculationCodeApply calculation method for each calculation code and group of OrderItems.

Chapter 5. Calculation rules

The WebSphere Commerce calculation framework separates the object that indicates the calculation to be performed for an OrderItem (calculation code) from the object or objects responsible for doing the calculation. The object responsible for the calculation of the monetary amount associated with an OrderItem is called a calculation rule.

Separating the calculation rule from the calculation code offers more flexibility and easier customization than having the calculation code perform calculations directly. By separating calculation rules from calculation codes, one calculation code can have many calculation rules. These multiple calculation rules can be combined, given an order of precedence, and restricted to members of specific member groups. Calculation rules used for shipping or tax calculations can also be restricted by jurisdictions.

For example, if you have a store that ships products to a number of jurisdictions in which you must collect sales tax and there are various different sales taxes in each jurisdiction, you would do the following:

1. Create a sales tax calculation code and associate it with the catalog entries for products that must be taxed.

2. For each jurisdiction in which you must collect sales taxes, create one calculation rule to calculate each sales tax for the jurisdiction. Each calculation rule must be associated with the following:

   - A tax category

   - A tax jurisdiction.

   - The sales tax calculation code

The sales tax calculation code calculates amounts for *all* tax categories, by having several calculation rules, one for each tax category. A calculation rule calculates an amount for a particular tax category.

If you want your store to ship products into a new jurisdiction in which you must collect sales taxes, you do not have to create a new calculation code and attach it to the OrderItem. You can create new calculation rules and associate them with the appropriate tax categories, the new tax jurisdiction, and the existing calculation code.

The properties of calculation rules are defined in the CALRULE database table. The CALRULE database table and other database tables that affect the way calculation rules are used are discussed in the next section.

Database tables for calculation rules

Information about calculation rules is provided in the following WebSphere Commerce database tables:

CALRULE
> This is the main database table that defines calculation rules.

CALRULEMGP
> This database table associates a calculation rule with a member group. The

association of a member group with a calculation rule permits the restriction of a calculation rule to certain member groups.

SHPJCRULE
This database table can be used by the ShippingCalculationRuleQualify calculation method to choose a calculation rule based on the shipping mode and fulfillment center when the shipping address matches one of the shipping jurisdictions in a particular shipping jurisdiction group.

STENCALUSG
This database table specifies the CalculationRuleCombine calculation method subclass that determines how multiple calculation rules for an OrderItem are combined.

TAXJCRULE
This table can be used by a TaxCalculationRuleQualify calculation method to choose a calculation rule when shipping from a fulfillment center to a shipping address that matches one of the tax jurisdictions in a particular tax jurisdiction group.

The following sections discuss key calculation rule attributes that are stored in some of the database tables. For details on all the columns of the calculation rule database tables, refer to the database schema documentation in the online information. All of the attributes discussed in the following section are required unless they are marked "optional."

CALRULE database table

The CALRULE database table stores the following information about calculation rules:

Calculation rule ID (CALRULE_ID)
This is a unique identifier for the calculation rule, assigned by WebSphere Commerce.

Identifying number (IDENTIFIER)
This is an integer number that, along with the calculation code ID, uniquely identifies this calculation rule. The default value is 1.

Calculation code ID (CALCODE_ID)
This is the calculation code to which the calculation rule belongs. A calculation rule can only belong to one calculation code.

Start and end dates (STARTDATE, ENDDATE) (optional)
These optional attributes set the time range for which a calculation rule is effective. The start and end dates have the following effects on a calculation rule:

- If a calculation rule is called on a date outside of the defined range, the rule will be ignored.
- If only the start date is set, the rule is ignored until the start date.
- If only the end date is set, the rule is effective until the end date.
- If neither date is set, the rule is always effective.

Combination (COMBINATION)
The combination attribute of a calculation rule defines how the results from a calculation rule are combined with the results from other rules for an OrderItem. A calculation rule has one of the following combination attributes:

inAdditionTo

The results from a calculation rule with an inAdditionTo combination attribute can be combined with the results from any other rules.

notInCombinationWith

The results from a calculation rule with a notInCombinationWith combination attribute can only be combined with results from rules with an inAdditionTo combination attribute. The results cannot be combined with results from an inCombinationWith rule.

inCombinationWith

The results from a calculation rule with an InCombinationWith combination attribute can be only combined with results from rules with either an inAdditionTo combination attribute or an InCombinationWith combination attribute. The results cannot be combined with results from a notInCombinationWith rule.

Qualification flag (FLAGS)

This attribute contains a bit flag that is used to indicate whether or not the CalculationRuleQualify calculation method for this calculation rule should be called by the CalculationRuleCombine calculation method. If the bit flag is 1, the CalculationRuleQualify calculation method is called to determine for which OrderItems the rule qualifies. If the bit flag is 0, the method is not called, and the rule always qualifies for all OrderItems. The default setting for this attribute is 0.

Sequence (SEQUENCE)

Sequence is a number used to specify the order in which calculation rules for the same calculation codes are processed. Calculation rules are processed from lowest sequence number to highest sequence number. In the event that two calculation rules have the same sequence number, the calculation rule with the lower calculation rule ID is processed first.

Tax category ID (TAXCGRY_ID) (optional)

The tax category ID specifies the tax category for which this calculation rule is effective.

CalculationRuleQualify calculation method ID (CALMETHOD_ID_QFY)

This is the identifier of the calculation method, as defined in the CALMETHOD database table, that determines to which OrderItems this calculation rule applies.

CalculationRuleCalculate calculation method ID (CALMETHOD_ID)

This is the identifier of the calculation method, as defined in the CALMETHOD database table, that calculates a monetary result for a set of OrderItems.

SHPJCRULE database table

The SHPJCRULE database table defines relationships between calculation rules, fulfillment centers, shipping jurisdiction groups and shipping modes. These relationships can be used by shipping CalculationRuleQualify calculation methods to choose which calculation rule to use. The decision is based on the shipping mode and fulfillment center when shipping address matches one of the shipping jurisdictions in a shipping jurisdiction group.

The SHPJCRULE database table contains the following information used when qualifying calculation rules:

Calculation rule ID (CALRULE_ID)
> This is the identifier of the calculation rule to which this relationship applies.

Fulfillment center ID (FFMCENTER_ID) (optional)
> This is the identifier of the fulfillment center to which this relationship applies. If this is null, then this relationship applies to all fulfillment centers.

Shipping jurisdiction group ID (JURSTGROUP_ID) (optional)
> This is the identifier of the shipping jurisdiction group to which this relationship applies. If this is null, then this relationship applies to all shipping jurisdiction groups.

Shipping mode ID (SHIPMODE_ID) (optional)
> This is the identifier of the shipping mode to which this relationship applies. If this is null, then this relationship applies to all shipping modes.

Precedence (PRECEDENCE)
> This attribute is used by a CalculationRuleQualify calculation method to qualify a calculation rule when the shipping address of an OrderItem falls into more than one shipping jurisdiction group for the same fulfillment center and shipping mode associated with a calculation rule. When the shipping address of the OrderItem falls into more than one shipping jurisdiction group, for the same fulfillment center and shipping mode, the calculation rule with the highest precedence value qualifies. If the precedence values for multiple calculations rules are the same, then all of the calculation rules sharing the same highest precedence value are applied.

STENCALUSG database table

The STENCALUSG database table contains the following information about calculation rules:

CalculationRuleCombine calculation method (ACTRC_CALMETHOD_ID) (optional)
> This is the identifier of the CalculationRuleCombine calculation method. If a CalculationRuleCombine calculation method is not specified for a store, then the CalculationRuleCombine calculation method defined for the store group is used.

The STENCALUSG database table also contains information about calculation usages. Calculation usages are covered in Chapter 3, "Calculation usages" on page 13.

TAXJCRULE database table

The TAXJCRULE database table defines the relationships between calculation rules, fulfillment centers and tax jurisdiction groups. These relationships can be used by sales tax and shipping tax CalculationRuleQualify calculation methods to choose which calculation rule to use. The decision is based on the fulfillment center when shipping address matches one of the tax jurisdictions in a tax jurisdiction group.

The TAXJCRULE database table contains the following information used when qualifying calculation rules:

Calculation rule ID (CALRULE_ID)
> This is the identifier of the calculation rule to which this relationship applies.

Fulfillment center ID (FFMCENTER_ID) (optional)
> This is the identifier of the fulfillment center to which this relationship applies. If this is null, then this relationship applies to all fulfillment centers.

Tax jurisdiction group ID (JURSTGROUP_ID) (optional)
> This is the identifier of the tax jurisdiction group to which this relationship applies. If this is null, then this relationship applies to all tax jurisdiction groups.

Precedence (PRECEDENCE)
> This attribute is used by a CalculationRuleQualify calculation method to qualify a calculation rule when the shipping address of an OrderItem falls into more than one tax jurisdiction group for the same fulfillment center associated with a calculation rule. When the shipping address of the OrderItem falls into more than one tax jurisdiction group, for the same fulfillment center, the calculation rule with the highest precedence value qualifies. If the precedence values for multiple calculations rules are the same, then all of the calculation rules sharing the same highest precedence value are applied.

Calculation rule data model diagrams

The following data model diagrams provide illustrations of the relationships between the database tables used to define generic calculation rules, discount calculation rules, shipping calculation rules, and tax calculation rules.
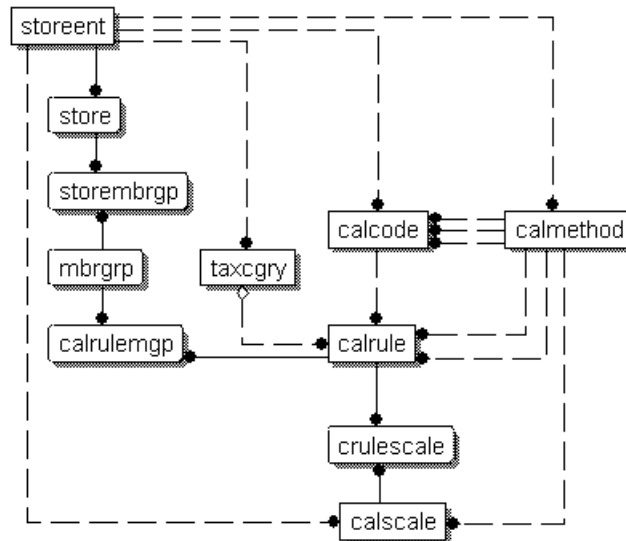
Figure 9. Generic calculation rule data model diagram
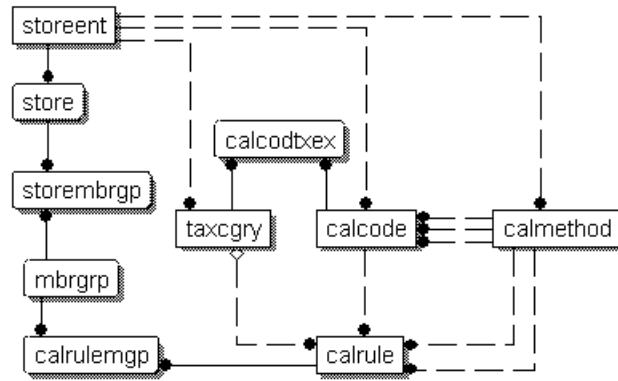
Figure 10. Discount calculation rule data model diagram
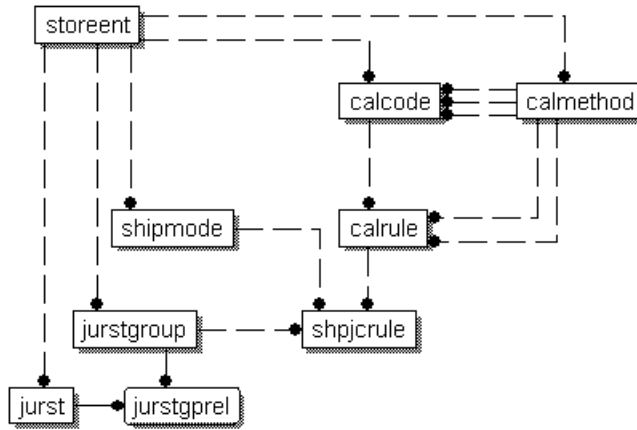
Figure 11. Shipping calculation rule data model diagram
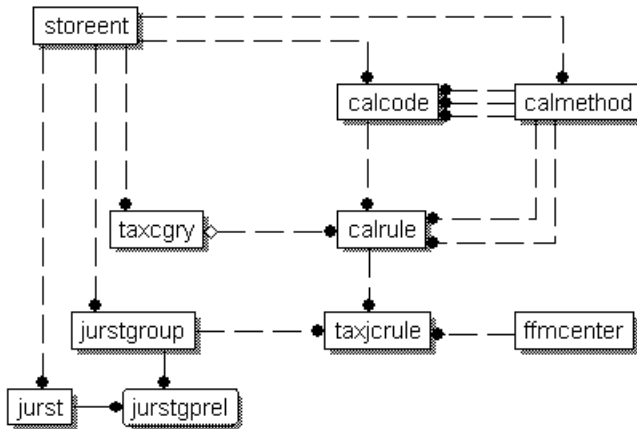
Figure 12. Tax calculation rule data model diagram

> For information on the conventions used in these diagrams, see Appendix C, "Data model legend" on page 109.

For details on the database tables in the diagrams refer to the database schema documentation in the online information.

Calculation methods for calculation rules

There are three calculation method subclasses associated with calculation rules:

- CalculationRuleCombine
- CalculationRuleQualify
- CalculationRuleCalculate

This section discusses these calculation method subclasses and the default implementations provided with WebSphere Commerce. Additional information on the interfaces and commands that implement the calculation method subclasses can be found in the API documentation in the online information.

CalculationRuleCombine calculation method

The CalculationRuleCombine calculation method determines which calculation rules are used to calculate monetary amounts for the list of OrderItems passed to the CalculationRuleCombine calculation method by the CalculationCodeCalculate calculation method.

The CalculationRuleCombine method returns a list where each item in the list consists of the following:

- A calculation rule
- A list of OrderItems to which the calculation rule applies
- A list of monetary amounts corresponding to each OrderItem

For tax calculations, the monetary amounts corresponding to each OrderItem may be separated into the amount for each tax category applicable to the OrderItem.

Any commands used as CalculationRuleCombine calculation methods must implement the CalculationRuleCombineCmd interface. For more information on the CalculationRuleCombineCmd interface, see the API documentation in the online information.

The CalculationRuleCombineCmdImpl command, provided with WebSphere Commerce, implements the CalculationRuleCombineCmd interface.

CalculationRuleCombineCmdImpl
This is the default implementation of the CalculationRuleCombine command.

This command is called by the CalculationCodeCalculate calculation method to identify the calculation rules associated with the OrderItems. Given a list of OrderItems, a calculation code, and an optional list of tax categories, this command does the following:

1. Using the calculation usage being processed, the calculation code, and the optional tax categories, the CalculationRuleCombineCmdImpl command creates

a list of calculation rules that could be applied to the OrderItems. The calculation rules are selected based on the calculation code and the optional tax categories.

2. The command then sorts the calculation rules in ascending order by the calculation sequence attributes of their tax categories (TAXCGRY.CALCULATIONSEQ). Calculation rules with the same tax category are sorted in ascending order by their sequence attribute. Calculation rules without tax categories are sorted in ascending order by their sequence attribute.

3. If the qualification flag attribute of the calculation rule is set to 1, the CalculationRuleQualify calculation method for each calculation rule is called to determine to which OrderItems each calculation rule applies. The CalculationRuleQualify calculation method returns a list of OrderItems to which the calculation rule applies.

4. For each calculation rule and each list of OrderItems returned by the CalculationRuleQualify calculation method (or all the OrderItems if the CalculationRuleQualify method was not called), the command calls the CalculationRuleCalculate calculation method. The CalculationRuleCalculate calculation method returns a monetary amount for each OrderItem.

5. When all calculation rules have been processed, the monetary amounts from different calculation rules for each OrderItem are combined as follows:

   a. The monetary amounts for calculation rules with an inAdditionTo combination attribute are added.

   b. For each calculation rule that has an inCombinationWith combination attribute, the sum of the calculated monetary amounts for the calculation rule and the total from step 5a. are added.

   c. All monetary amounts for calculation rules with an inCombinationWith combination attribute are added to the total from step 5a.

   d. All of the results obtained in step 5b (there is one result for each notInCombinationWith rule) and the result obtained in step 5c are compared. The lowest of these results is considered the valid monetary amount.

6. The calculation rules used to calculate the valid monetary amount, the monetary amounts produced by each of these calculation rules, and their associated OrderItems are returned to the calling method.

CalculationRuleQualify calculation method

The CalculationRuleQualify calculation method checks if a calculation rule applies to list of OrderItems and returns a list of elements. Each element is a group of OrderItems, from the original list of OrderItems, that should be processed together by the calculation rule.

CalculationRuleQualify calculation methods are only called if the qualification flag attribute of the calculation rule is set to 1.

Any commands that are used as a CalculationRuleQualify calculation method must implement the CalculationRuleQualifyCmd interface. For more information on the CalculationRuleQualifyCmd interface, see the API documentation in the online information.

The following commands implementing the CalculationRuleQualifyCmd interface are provided with WebSphere Commerce:

- DiscountCalculationRuleQualifyCmdImpl
- ShippingCalculationRuleQualifyCmdImpl

- TaxCalculationRuleQualifyCmdImpl

DiscountCalculationRuleQualifyCmdImpl

This is the discount specific version of the CalculationRuleQualify command that implements the DiscountCalculationRuleQualifyCmd interface that extends the CalculationRuleQualifyCmd interface.

This command returns OrderItems with a customer in one of the member groups associated with the calculation rule and recognized by the store. If the customer is not in any of the member groups associated with the calculation rule, this command returns null.

Calculation rules are associated with member groups in the CALRULEMGP database table and member groups are recognized by stores in the STOREMBRGP database table.

ShippingCalculationRuleQualifyCmdImpl

This is the shipping specific version of the CalculationRuleQualify command that implements the ShippingCalculationRuleQualifyCmd interface that extends the CalculationRuleQualifyCmd interface.

This command compares the shipping jurisdictions, shipping modes, and fulfillment centers of the OrderItems with those associated with a calculation rule. The ShippingCalculationRuleQualifyCmdImpl command only returns OrderItems that share shipping jurisdictions, shipping modes, and fulfillment centers with the calculation rule.

When the shipping address of the OrderItem falls into more than one shipping jurisdiction group, for the same fulfillment center and shipping mode, the calculation rule with the highest precedence value defined in the SHPJCRULE database qualifies. If multiple calculation rules share the highest precedence value, then all calculation rules sharing the highest precedence values apply.

Calculation rules are associated with shipping jurisdictions, shipping modes, and fulfillment centers in the SHPJCRULE database table.

TaxCalculationRuleQualifyCmdImpl

This is the tax specific version of the CalculationRuleQualify command that implements the TaxCalculationRuleQualifyCmd interface that extends the CalculationRuleQualifyCmd interface.

This command compares the tax jurisdictions and fulfillment centers of the OrderItems with those associated with a calculation rule. The TaxCalculationRuleQualifyCmdImpl command only returns OrderItems that share tax jurisdictions and fulfillment centers with a calculation rule.

When the shipping address of the OrderItem falls into more than one tax jurisdiction group for the same fulfillment center, the calculation rule with the highest precedence value defined in the TAXJCRULE database table applies. If multiple calculation rules share the highest precedence value, then all calculation rules sharing the highest precedence values apply.

Calculation rules are associated with tax jurisdictions and fulfillment centers in the TAXJCRULE database table.

CalculationRuleCalculate calculation method

This command is called by the CalculationRuleCombine calculation method to calculate the result of a particular calculation rule. For a calculation rule and a list of OrderItems, this command returns a monetary amount for each OrderItem. The currency of the calculated monetary amount is the order currency.

CalculationRuleCalculate calculation methods must implement the CalculationRuleCalculateCmd interface. For more information on the CalculationRuleCalculateCmd interface, see the API documentation in the online information.

The CalculationRuleCalculateCmdImpl command, provided with WebSphere Commerce, implements the CalculationRuleCalculateCmd interface.

CalculationRuleCalculateCmdImpl

This is the default implementation of the CalculationRuleCalculate command.

Using a calculation rule and list of OrderItems, this command does the following:

1. Clears the monetary amount associated with the OrderItems being processed with the calculation rule.
2. Determines the list of calculation scales associated with the calculation rule using the CRULESCALE database table.
3. Processes calculation scales not associated with any currency by doing the following:
   a. Obtains a monetary result in the Order currency, for each OrderItem, using the the calculation scale. For more details on this step, see "How calculation scales and calculation ranges are used" on page 64.
   b. Adds the monetary amount result for each OrderItem to the total result for each OrderItem.
4. Assembles a list of the currencies the remaining calculation scales are associated with
5. Does one of the following:
   - If the list of calculation scale currencies contains the order currency, does the following:
     a. Starts with a candidate result of zero for each OrderItem.
     b. For each calculation scale associated with the same currency as the order, does the following:
        1) Obtains a monetary amount for each OrderItem. For more details on this step, see "How calculation scales and calculation ranges are used" on page 64.
        2) Adds each OrderItem's monetary amount to each OrderItem's candidate result.

     Calculation scales associated with currencies other than the order currency are ignored.
   - If the list of calculation scale currencies does not contain the order currency, does the following:
     a. For each currency in list do the following:
        1) Starts with a candidate result of zero for each OrderItem.
        2) For each calculation scale associated with the currency being processed, does the following:

a) Obtains a monetary amount for each OrderItem. For more details on this step, see "How calculation scales and calculation ranges are used" on page 64.

b) Adds each OrderItem's monetary amount to each OrderItem's candidate result.

If the conversion from a currency in the list of calculation scale currencies to the order currency is not defined in WebSphere Commerce, then the calculation scales associated with that currency are ignored.

b. Compares the sum of candidate results for each currency. The candidate results for the currency producing the lowest sum are considered valid. Candidate results from the other currencies are discarded.

6. For each OrderItem, adds the amount calculated in step 3 on page 44 and the candidate result from step 5 on page 44.

7. Returns the results to the calling method.

Information about how to obtain monetary amounts from calculation scales and the calculation methods that apply to calculation scales is provided in Chapter 6, "Calculation scales and calculation ranges" on page 47.

How calculation rules are used

The default CalculationCodeCalculate calculation method implementation uses calculation rules to calculate monetary amounts for each OrderItem to which it applies. The default CalculationCodeCalculate calculation method implementation calls the CalculationRuleCombine calculation method and passes the CalculationRuleCombine calculation method the list of applicable OrderItems. The default implementation of the CalculationRuleCombine calculation method performs the following steps:

1. Determines the list of calculation rules that are effective for the calculation code by checking the CALRULE.FLAGS attribute and calling a CalculationRuleQualify calculation method if the flag is set.

2. For each effective calculation rule, performs the calculation defined by the rule for each OrderItem the rule applies to. If the calculation to be performed is based on finding a value in a scale, calculation scales are used to obtain the monetary amount. Calculation scales are covered in Chapter 6, "Calculation scales and calculation ranges" on page 47.

3. When all rules have been calculated for all items, combines the results of the calculation for each item. The combination attribute of the calculation rule controls how the results are combined.

4. The CalculationRuleCombine calculation method returns a list of calculation rules applied, the OrderItems to which they were applied, and the resulting monetary amount for each OrderItem. This list is returned to the calling CalculationCodeCalculate calculation method.

Chapter 6. Calculation scales and calculation ranges

Many calculations in WebSphere Commerce involve looking up a value from a table. For example, a store may have shipping charges based on the number of items shipped in an order. The table specifying the shipping charges could look like this:

Table 4. Example of a table used to determine shipping charges

Number of items in order	Shipping charge (local currency)
< 5	3.00
5 – 10	10.00
11 – 15	22.00
> 15	50.00

To calculate the shipping charge, a clerk would count the number of items in the order and use the total number of items to find the shipping charge in the table.

WebSphere Commerce provides a similar function as part of the calculation framework — calculation scales. Calculation scales can be thought of two column tables, where the values in the first column are compared with a known value to find a row in the table. When the appropriate row in the table is located, the value in the second column is then applied to a calculation associated with the original known value.

The process of finding the value needed from the table is called a *scale look–up*. The calculation scale consists of a list of *calculation ranges* and associated *look–up results*. The known value that is compared to the calculation ranges is called a *look–up number*. For example, when calculating shipping charges using Table 4 for an order that contains eight items, the look–up number is *8*, the calculation range that the look–up number matches is *5 – 10*, and the look–up result is *10.00*.

A calculation scale can be used to calculate monetary amounts, one for each OrderItem in a list of OrderItems. Calculation scales are used by the default implementation of the CalculationRuleCalculate calculation method. The amounts calculated by a calculation scale are determined by the calculation ranges and range values of the calculation scale, and by the calculation methods they use. Calculation scales are associated with either a MonetaryCalculationScaleLookup calculation method or a QuantityCalculationScaleLookup calculation method. Calculation ranges are associated with a CalculationRange calculation method.

Calculation scale properties

Calculation scales have properties associated with them that affect when a calculation scale is applied and the monetary amounts calculated using a calculation scale. A calculation scale can have the following properties:

Currency

A calculation scale can be associated with a specific currency. Associating a calculation scale with a currency implies that all the calculation ranges in the scale are in that currency. A calculation scale associated with a currency is applicable to OrderItems if the scale meets one of the following conditions:

- The calculation scale and the OrderItems have the same currency.
- The currencies of the OrderItems can be converted to the calculation scale currency and no calculation scales with the same currency as the OrderItem exist. If multiple calculation scales for the same calculation usage with different currencies can be applied, the results of the scales are converted to the OrderItem currency. The scale producing the lowest monetary amount in the OrderItem currency is applied.

A calculation scale can be associated with a currency only, a unit of measurement only, or neither. A calculation scale cannot be associated with both a currency and a unit of measure.

Unit of measurement

A calculation scale can be associated with a specific unit of measurement. Associating a calculation scale with a unit of measurement implies that all calculation ranges in the scale are in that unit of measurement.

An OrderItem can have a unit of measurement associated with the quantity or weight value for the OrderItem. Various CalculationScaleLookup calculation methods use either the quantity or weight value of the OrderItem to determine look–up values. Look–up values are compared with calculation range start values to determine which CalculationRange calculation methods to call.

For CalculationScaleLookup calculation methods that use an OrderItem's quantity value, the unit of measurement of the quantity must be convertible to the unit of measurement for the calculation scale. For CalculationScaleLookup calculation methods that use an OrderItem's weight value, the unit of measurement of the weight must be convertible the unit of measurement for the calculation scale. The quantity and weight units of measure for an OrderItem are specified in the CATENTSHIP database table, referenced by the CATENTRY_ID column of the OrderItem.

A unit of measure is convertible to another unit of measure if they are the same, or if there are conversions defined in the QTYCONVERT database table that would allow a conversion from the one to the other.

A calculation scale can be associated with a currency only, a unit of measurement only, or neither. A calculation scale cannot be associated with both a currency and a unit of measure.

Calculation usage

A calculation usage defines the type of calculation to which the calculation scale applies. Calculation usages are covered in Chapter 3, "Calculation usages" on page 13.

WebSphere Commerce requires each calculation scale to have a calculation usage. Calculation scales without usages are not permitted.

In addition to these properties, calculation scales are classified by the CalculationScaleLookup calculation method associated with the calculation scale:

Monetary calculation scale

A monetary calculation scale is a calculation scale that is associated with a MonetaryCalculationScaleLookup calculation method.

Quantity calculation scale

A quantity calculation scale is a calculation scale that is associated with a QuantityCalculationScaleLookup calculation method.

Calculation range properties

Calculation ranges have properties associated with them that affect how the values in calculation rules are used. A calculation range can have the following property:

Cumulative

If the calculation ranges for a calculation scale are cumulative, the resulting total monetary amount for the calculation scale is calculated using the look–up results for all calculation ranges that match or are less than the look–up number. The resulting monetary amount depends on the CalculationRange calculation method used by the calculation range.

Also, the look–up number used to calculate the monetary amount for each calculation range changes for each calculation range. This modified look–up number is called the applicable part of the look–up number.

For example, consider the following weight–based calculation scale:

Table 5. Weight-based calculation scale

Shipping weight	Charge
greater than or equal to 0kg	2.00 fixed charge
greater than or equal to 5kg	0.25 per kg
greater than or equal to 10kg	0.10 per kg
greater than or equal to 100kg	0.01 per kg

The total shipping charge for a 20kg package, assuming cumulative calculation ranges, will be determined using a number of calculation ranges. The look–up number is 20, but the applicable part of the look–up number will change as the total shipping cost is calculated. The applicable part of the look–up number is calculated using the following formula:

```
min(lookupNumber, rangeEnd) - rangeStart
```

The following table shows the applicable parts of the look–up number used:

Table 6. Applicable parts of look–up numbers

Look–up number	Range starting value	Range ending value	Applicable part of the look–up number
20	0	5	5
20	5	10	5
20	10	100	10

The applicable parts of the look–up number are used to determine the total shipping cost as follows:

1. The look–up number (20) matches the first range. The monetary result is the look–up result applying to the applicable part of the look–up number (5). This results in the total shipping charge being 2.00.

2. The look–up number (20) matches the second range. The monetary result (0.25) is the look–up result applying to the applicable part of the look–up number (5). The resulting shipping charge is 1.25. Because the calculation range is cumulative, this amount is added to the existing shipping charges. The total shipping charge is now 3.25.

3. The look–up number (20) matches the third range. The monetary result (0.10) is the look–up result applying to the applicable part of the look–up number (10). The resulting shipping charge is 1.00. Because the

calculation range is cumulative, this amount is added to the existing shipping charges. The total shipping charge is now 4.25.

4. The look–up number (20) does not match the fourth range. The calculation is complete. The total shipping charge is 4.25.

If the calculation ranges are non–cumulative, the applicable part of the look–up number and the look–up number are the same. The shipping charges would be determined as follows:

1. The look–up number (20) matches the first range. The monetary result is the look–up result applying to the applicable part of the look–up number (20). This results in the total shipping charge being 2.00.

2. The look–up number (20) matches the second range. The monetary result (0.25) is the look–up result applying to the applicable part of the look–up number (20). The resulting shipping charge is 5.00. Because the calculation range is non–cumulative, this amount replaces the existing shipping charges. The total shipping charge is now 5.00.

3. The look–up number (20) matches the third range. The monetary result (0.10) is the look–up result applying to the applicable part of the look–up number (20). The resulting shipping charge is 2.00. Because the calculation range is non–cumulative, this amount replaces the existing shipping charges. The total shipping charge is now 2.00.

4. The look–up number (20) does not match the fourth range. The calculation is complete. The total shipping charge is 2.00.

Database tables for calculation scales and calculation ranges

Information about a calculation scale and the calculation ranges that make up the scale is separated into different tables. The following WebSphere Commerce database tables define calculation scales and calculation ranges:

CALSCALE
> This is the main database table that defines calculation scales.

CALSCALEDS
> This database table contains national language descriptions of a calculation scale. This allows one calculation scale to have descriptions in multiple languages.

CRULESCALE
> This database table relates a calculation scale to a calculation rule. A calculation rule may have multiple calculation scales and a calculation scale may be used by multiple calculation rules.

CALRANGE
> This database table defines the calculation ranges that make up a calculation scale.

CALRLOOKUP
> This database table contains the look–up results for calculation ranges. The currencies of the look–up results, optional attributes, are also specified in this database table.

The following sections discuss key calculation scale attributes that are stored in some of the database tables. For details on all the columns of the calculation scale database tables and other database tables, refer to the database schema documentation in the online help. All of the attributes discussed in the following sections are required unless they are marked "optional."

CALSCALE database table

The CALSCALE database table contains the following information about calculation scales:

Identifying string (CODE) (optional)
> This is a character string that uniquely identifies this calculation scale, given a particular calculation usage and store or store group.

CalculationScaleLookup calculation method ID (CALMETHOD_ID)
> This is the identifier for the CalculationScaleLookup calculation method associated with the calculation scale. The CalculationScaleLookup calculation method is responsible for calculating the look–up number, base monetary value, result multiplier, and a set of mathematical weights that will be used by the calculation scale to determine a monetary amount.

Calculation usage ID (CALUSAGE_ID)
> This is the identifier for the calculation usage associated with the calculation scale. The calculation usage defines the type of calculation for which the calculation scale can be used. Calculation usages are defined in the CALUSAGE database table. WebSphere Commerce supplies a number of predefined calculation usages. For information on calculation usages and the predefined calculation usages, see Chapter 3, "Calculation usages" on page 13.

Quantity unit ID (QTYUNIT_ID) (optional)
> The quantity unit ID defines the unit of measurement associated with the calculation scale. The list of valid quantity unit IDs is stored in the QTYUNIT database table. WebSphere Commerce provides a number of predefined quantity units in the QTYUNIT table. The predefined unit of measurement codes are based on the UN/CEFACT Working Party on Facilitation of International Trade Procedures (WP.4), Recommendation No. 20, Codes for Units of Measure Used in International Trade.
>
> Information used to convert between units of measurements is stored in the QTYCONVERT database table.
>
> CALSCALE.SETCURR and CALSCALE.QTYUNIT_ID cannot both be set for a calculation scale.

Currency ID (SETCCURR) (optional)
> The currency ID defines the currency associated with the calculation ranges of the scale. The list of valid currency IDs is stored in the SETCURR table. WebSphere Commerce provides a number of predefined currency codes in the SETCURR table. The predefined currency codes are derived from the ISO 4217 standard.
>
> Information used to convert between currencies is stored in the CURCONVERT database table.
>
> CALSCALE.SETCURR and CALSCALE.QTYUNIT_ID cannot both be set for a calculation scale.

Store entity ID (STOREENT_ID)
> The store entity ID is the identifier for the store or the store group to which the calculation scale belongs. A calculation scale always belongs to a store or store group.

CALRANGE database table

The CALRANGE database table contains the following information about calculation ranges:

CalculationRange calculation method ID (CALMETHOD_ID)
This is the identifier of the calculation method used as the CalculationRange calculation method for this calculation range. The CalculationRange calculation method determines a monetary amount from a look–up result.

Calculation scale ID (CALSCALE_ID)
This attribute is the identifier of the calculation scale to which a calculation range belongs.

Cumulative calculation range indicator (CUMULATIVE)
This attribute indicates if the calculation range is cumulative or not. Cumulative calculation scales are explained in the "Calculation range properties" on page 49.

Calculation range starting value (RANGESTART) (optional)
This value defines the starting value of a calculation range. In WebSphere Commerce, a look–up number matches a calculation range when the look–up number is greater than or equal to the starting value of the range.

If the starting value of a calculation range is not defined, the calculation range will always match the look–up number.

CALRLOOKUP database table

The CALRLOOKUP database table contains the following information about calculation range look–up results:

Calculation range ID (CALRANGE_ID)
This attribute is the identifier of the calculation range to which a calculation range look–up result belongs.

Currency ID (SETCURR)
The currency ID defines the currency associated with the calculation range look–up result and indicates that the look–up result is a monetary value. The list of valid currency IDs is stored in the SETCURR table. WebSphere Commerce provides a number of predefined currency codes in the SETCURR table. The predefined currency codes are derived from the ISO 4217 standard.

Information used to convert between currencies is stored in the CURCONVERT database table.

Look–up result (VALUE)
This is the value of the calculation range look–up result. This value is used by the CalculationRange to determine a monetary amount.

Calculation scale and calculation range data model diagram

The following data model diagram provides an illustration of the relationships between the database tables used to define calculation scales and calculation ranges.
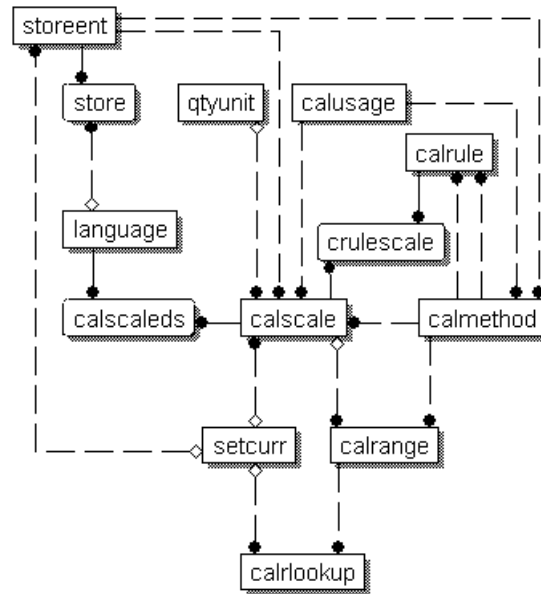
Figure 13. Calculation scale and calculation range data model diagram

For information on the conventions used in this diagram, see Appendix C, "Data model legend" on page 109.

For details on the database tables in the diagram refer to the database schema documentation in the online information.

Calculation methods for calculation scales

There are three calculation method subclasses associated with calculation scales:

- MonetaryCalculationScaleLookup
- QuantityCalculationScaleLookup
- CalculationRange

A QuantityCalculationScaleLookup applies to quantity calculation scales, while a MonetaryCalculationScaleLookup applies to monetary calculation scales.

CalculationScaleLookup calculation methods

The MonetaryCalculationScaleLookup and QuantityCalculationScaleLookup calculation methods determine a number of values required to perform a range look–up and calculate any monetary amounts using the look–up result. The calculation methods use a set of OrderItems, a calculation rule, and a calculation scale passed to them by the CalculationRuleCalculate calculation method to determine the following items:

Base monetary value

> The base monetary value is an amount that may be used by the CalculationRange calculation method to calculate a monetary amount using the base monetary value and the look–up result.

Look–up number

A look–up number is the value compared to the ranges in the calculation scale to find a look–up result.

Mathematical weights

The mathematical weights are values used to distribute the total calculated monetary amount using a calculation scale to each of the OrderItems in the set of OrderItems passed to the CalculationScaleLookup calculation method. The CalculationScaleLookup calculation method returns one mathematical weight for each OrderItem in the set of OrderItems provided.

The total monetary amount is distributed according to the percentage of the total of all mathematical weights that one OrderItem's mathematical weight represents. If an OrderItem's mathematical weight represents 23% of the total mathematical weight of the OrderItems being processed, the OrderItem's monetary amount is 23% of the total monetary amount determined for the calculation scale.

For example, assume monetary amounts for three OrderItems (ItemA, ItemB, and ItemC) are being determined and the total monetary amount determined for the calculation scale is $156. The following table shows how mathematical weights distribute the total monetary amount across the OrderItems:

Table 7. Example of distributing total monetary amount using mathematical weights

OrderItem	Mathematical weight	Percentage of total mathematical weights	Calculation	OrderItem monetary amount
ItemA	9	18%	$156*0.18	$28.08
ItemB	25	50%	$156*0.50	$78.00
ItemC	16	32%	$156*0.32	$49.92

Result multiplier

The result multiplier specifies the value by which the resulting monetary value should be multiplied to calculate a monetary amount for the set of OrderItems.

Note: Different implementations of the MonetaryCalculationScaleLookup or QuantityCalculationScaleLookup calculation methods will give different results for these items when using the same sets of OrderItems.

MonetaryCalculationScaleLookup and QuantityCalculationScaleLookup calculation methods return the base monetary value, the look–up number, the set of mathematical weights and the set of result multipliers to the CalculationRuleCalculate calculation method. The CalculationRuleCalculate calculation method then finds the calculation range look–up result corresponding to the look–up value and passes the applicable part of the look–up result to the CalculationRange calculation method. The applicable part of the look–up result depends on whether the calculation range is cumulative. The CalculationRange calculation method returns a monetary amount. The returned monetary amount is then distributed among the set of OrderItems according the mathematical weights calculated for each OrderItem.

For more information on the CalculationRange calculation method, see "CalculationRange calculation method" on page 62.

The following section discusses the CalculationScaleLookup and CalculationRange calculation methods and the default implementations provided with WebSphere Commerce. Additional information on the interfaces and commands that implement the calculation method subclasses can be found in the API documentation of the online information.

MonetaryCalculationScaleLookup calculation methods

For MonetaryCalculationScaleLookup calculation methods, the look–up number and base monetary values determined by the method are both monetary amounts. The monetary amounts used to determine the look–up number and base monetary values are different depending on the implementation of the monetary calculation scale look–up method. Some commonly used values in monetary calculation scale look–up methods include the following:

Net price
> The value for net price used in the calculation framework is the unit price of the OrderItem (ORDERITEMS.PRICE) multiplied by the quantity of the OrderItem (ORDERITEMS.QUANTITY) plus the sum of all adjustments associated with the OrderItem.

Non-discounted price
> The value for non-discounted price used in the calculation framework is the unit price of the OrderItem (ORDERITEMS.PRICE) multiplied by the quantity of the OrderItem (ORDERITEMS.QUANTITY).

Taxable net price
> The value for taxable net price used in the calculation framework for a particular tax category is the non-discounted price (defined above) plus the sum of all OrderItem adjustments subject to taxation for the tax category. The value of taxable net price can be different for each tax category since different OrderItems and OrderItem adjustments may be subject to different tax categories.

Taxable unit price
> The value for taxable unit price used in the calculation framework for a particular tax category is the taxable net price (defined above) of the OrderItem divided by the quantity of the OrderItem (ORDERITEMS.QUANTITY). The per unit taxable adjustment is the total of all OrderItem adjustments subject to taxation for the tax category for the OrderItem and divided by the quantity associated with the OrderItem (ORDERITEMS.QUANTITY).

Unit price
> The value for unit price used in the calculation framework is the per unit price associated with an OrderItem (ORDERITEMS.PRICE) plus the per unit adjustment for the OrderItem. The per unit adjustment is the total of all OrderItem adjustments for the OrderItem divided by the quantity associated with the OrderItem (ORDERITEMS.QUANTITY).

MonetaryCalculationScaleLookup calculation methods must extend the CalculationScaleLookupCmd interface. For more information on the CalculationScaleLookupCmd interface, see the API documentation in the online information.

The following MonetaryCalculationScaleLookup calculation methods are provided with WebSphere Commerce:
- NetPriceCalculationScaleLookupCmdImpl

- NetPriceWithQuantityAsResultMultiplierCalculationScaleLookupCmdImpl
- NetShippingCalculationScaleLookupCmdImpl
- NonDiscountedPriceCalculationScaleLookupCmdImpl
- TaxableNetPriceCalculationScaleLookupCmdImpl
- TaxableNetPricePlusNetShippingCalculationScaleLookupCmdImpl
- TaxableUnitPriceCalculationScaleLookupCmdImpl
- TaxableUnitPricePlusUnitShippingCalculationScaleLookupCmdImpl
- UnitPriceCalculationScaleLookupCmdImpl
- UnitShippingCalculationScaleLookupCmdImpl

NetPriceCalculationScaleLookupCmdImpl

This command is suitable for calculating discounts or shipping scales based on a discounted price.

The values calculated by this command are as follows:

Base monetary value
> The sum of net prices of the OrderItems

Look–up number
> The sum of net prices of the OrderItems

Mathematical weights
> The net price of each OrderItem

Result multiplier
> One

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

NetShippingCalculationScaleLookupCmdImpl

This command is suitable for calculating tax scales on shipping charges.

The values calculated by this command are as follows:

Base monetary value
> The sum of all the shipping charges for the OrderItems
> (ORDERITEMS.SHIPCHARGE)

Look–up number
> The sum of all the shipping charges for the OrderItems
> (ORDERITEMS.SHIPCHARGE)

Mathematical weights
> The shipping charge for each OrderItem

Result multiplier
> One

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

NonDiscountedPriceCalculationScaleLookupCmdImpl

This command is suitable for shipping scales and some kinds of discounts. For example, if a percentage discount is based on the non–discounted price of an OrderItem, two successive 10% discounts would give the customer a total 20% discount on the OrderItem. If the discount is based on the net price of an OrderItem, two successive 10% discounts would give the customer a total 19% discount, because the second discount is based on the price of the OrderItem less the first discount (or 90% of the original non–discounted price).

The values calculated by this command are as follows:

Base monetary value
    The sum of the non-discounted prices of OrderItems

Look–up number
    The sum of the non-discounted prices of OrderItems

Mathematical weights
    The non-discounted price of each OrderItem

Result multiplier
    One

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

TaxableNetPriceCalculationScaleLookupCmdImpl

This command is suitable for sales tax scales.

The values calculated by this command are as follows:

Base monetary value
    The sum of the taxable net prices of OrderItems

Look–up number
    The sum of the taxable net prices of OrderItems

Mathematical weights
    The taxable net price of each OrderItem

Result multiplier
    One

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

TaxableNetPricePlusNetShippingCalculationScaleLookupCmdImpl

This command is suitable for combined sales and shipping tax scales.

The values calculated by this command are as follows:

Base monetary value
> The sum of the taxable net prices plus the sum of all the shipping charges for the OrderItems (ORDERITEMS.SHIPCHARGE)

Look–up number
> The sum of the taxable net prices plus the sum of all the shipping charges for the OrderItems (ORDERITEMS.SHIPCHARGE)

Mathematical weights
> The taxable net price for each OrderItem plus the shipping charge for each OrderItem divided by the OrderItem quantity (ORDERITEMS.QUANTITY)

Result multiplier
> One

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

TaxableUnitPriceCalculationScaleLookupCmdImpl

This command is suitable for per item sales tax scales, especially where a maximum tax per item is specified. This command should only be used when OrderItems are grouped by offer. OrderItem grouping is determined by the grouping method attribute of the calculation code.

The values calculated by this command are as follows:

Base monetary value
> The sum of taxable net prices divided by the sum of OrderItem quantities

Look–up number
> The sum of taxable net prices divided by the sum of OrderItem quantities

Mathematical weights
> The taxable unit price for each OrderItem

Result multiplier
> The sum of the OrderItem quantities

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

TaxableUnitPricePlusUnitShippingCalculationScaleLookupCmdImpl

This command is suitable for per item combined sales and shipping tax scales, especially where a maximum tax per item is specified. This command should only be used when OrderItems are grouped by offer. OrderItem grouping is determined by the grouping method attribute of the calculation code.

The values calculated by this command are as follows:

Base monetary value
The sum of taxable net prices plus the sum of all the shipping charges for the OrderItems, divided by the sum of the OrderItem quantities

Look–up number
The sum of taxable net prices plus the sum of all the shipping charges for the OrderItems, divided by the sum of the OrderItem quantities

Mathematical weights
The taxable net prices plus the charges, divided by the OrderItem quantity (ORDERITEMS.QUANTITY), for each OrderItem

Result multiplier
The sum of the OrderItem quantities

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

UnitPriceCalculationScaleLookupCmdImpl

This command is appropriate for discount or shipping scales that specify a fixed or maximum charge per item based on each OrderItem's unit price. This command should only be used when OrderItems are grouped by offer. OrderItem grouping is determined by the grouping method attribute of the calculation code.

The values calculated by this command are as follows:

Base monetary value
The sum of net prices divided by the sum of OrderItem quantities

Look–up number
The sum of net prices divided by the sum of OrderItem quantities

Mathematical weights
The net price divided by the OrderItem quantity (ORDERITEMS.QUANTITY), for each OrderItem

Result multiplier
> The sum of the OrderItem quantities

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

UnitShippingCalculationScaleLookupCmdImpl

This command is appropriate for per item shipping tax scales, especially where a maximum tax per item may be specified.

The values calculated by this command are as follows:

Base monetary value
> The sum of shipping charges for the OrderItems divided by sum of the OrderItem quantities

Look–up number
> The sum of shipping charges for the OrderItems divided by sum of the OrderItem quantities

Mathematical weights
> The shipping charges divided by the OrderItem quantity (ORDERITEMS.QUANTITY), for each OrderItem

Result multiplier
> The sum of the OrderItem quantities

The look–up number is converted to the currency of the calculation scale. If no conversion is possible, this calculation method returns null. Returning null indicates to the calling calculation method that the currency conversion was not possible.

For the look–up number to be converted to the currency of the calculation scale, the conversion from the calculation scale currency to the order currency must be defined in WebSphere Commerce.

QuantityCalculationScaleLookup calculation methods

For QuantityCalculationScaleLookup calculation methods, the look–up number is a value that may or may not have an associated unit of measurement. Some common values used in quantity calculation scale look–up methods include the following:

Actual quantity
> The actual quantity of an OrderItem is the OrderItem quantity (ORDERITEMS.QUANTITY) multiplied by the nominal quantity of the item (CATENTSHIP.NOMINALQUANTITY).

Net price
> The value for net price used in the calculation framework is the unit price of the OrderItem (ORDERITEMS.PRICE) multiplied by the quantity of the OrderItem (ORDERITEMS.QUANTITY) plus the sum of all adjustments associated with the OrderItem.

The QuantityCalculationScaleLookup calculation methods are provided with
WebSphere Commerce
- QuantityCalculationScaleLookupCmdImpl
- QuantitySpreadByNetPriceCalculationScaleLookupCmdImpl
- WeightCalculationScaleLookupCmdImpl
- WeightSpreadByNetPriceCalculationScaleLookupCmdImpl

QuantityCalculationScaleLookupCmdImpl
This command is suitable for shipping scales.

If the calculation scale has a unit of measurement associated with it, the values
calculated are as follows:

Base monetary value
> The sum of the net prices of the OrderItems

Look–up number
> The sum of the actual quantities of the OrderItems, converted to the unit of
> measurement of the calculation scale

> If no conversion is possible, the calculation scale is not used by the calling
> calculation rule.

Mathematical weights
> The actual quantity of each OrderItem

Result multiplier
> One

If the calculation scale is not associated with a unit of measurement, the values
calculated are as follows:

Base monetary value
> The sum of the net prices of the OrderItems

Look–up number
> The sum of the OrderItem quantities (ORDERITEMS.QUANTITY)

Mathematical weights
> The OrderItem quantity (ORDERITEMS.QUANTITY) of each OrderItem

Result multiplier
> One

QuantitySpreadByNetPriceCalculationScaleLookupCmdImpl
This command is suitable for discount and shipping scales.

If the calculation scale has a unit of measurement associated with it, the values
calculated are as follows:

Base monetary value
> The sum of the net prices of the OrderItems

Look–up number
> The sum of actual quantities of the OrderItems, converted to the unit of
> measurement of the calculation scale.

> If no conversion is possible, the calculation scale is not used by the calling
> calculation rule.

Mathematical weights
> The net price of each OrderItem

Result multiplier
>One

If the calculation scale is not associated with a unit of measurement, the values calculated are as follows:

Base monetary value
>The sum of the net prices of the OrderItems

Look–up number
>The sum of OrderItem quantities (ORDERITEMS.QUANTITY)

Mathematical weights
>The net price of each OrderItem

Result multiplier
>One

WeightCalculationScaleLookupCmdImpl

This command is suitable for discount and shipping scales.

The values calculated by this command are as follows:

Base monetary value
>The sum of net prices of the OrderItems

Look–up number
>The sum of the shipping weights (CATENTSHIP.WEIGHT) multiplied by quantity (ORDERITEMS.QUANTITY) of the OrderItems. The total is converted to the unit of measure of the calculation scale.

Mathematical weights
>The shipping weight (CATENTSHIP.WEIGHT) multiplied by the quantity (ORDERITEMS.QUANTITY) of each OrderItem

Result multiplier
>One

WeightSpreadByNetPriceCalculationScaleLookupCmdImpl

The values calculated by this command are as follows:

Base monetary value
>The sum of net prices of the OrderItems

Look–up number
>The sum of the shipping weight (CATENTSHIP.WEIGHT) multiplied by quantity (ORDERITEMS.QUANTITY) of the OrderItems

Mathematical weights
>The net price of each of item

Result multiplier
>One

CalculationRange calculation method

The CalculationRange calculation method calculates a monetary amount for a particular calculation range. The CalculationRuleCalculate calculation method calls the CalculationRange calculation method and passes it the following parameters:

Currency
>This is the order currency. The returned monetary amount will be in this currency.

Calculation range look–up result

This is the value obtained from CALRLOOKUP database table using the look–up number.

Applicable part of the look–up number

This number is based on the look–up number determined by the CalculationScaleLookup calculation method. The applicable part of the look–up number can be determined as follows:

Non–cumulative calculation ranges:

The applicable part of the look–up number is the same as the look–up number.

Cumulative calculation ranges:

The applicable part of the look–up number is determined by the following formula:

```
min(lookupNumber, rangeEnd) - rangeStart
```

where:

- `lookupNumber` is the look–up number determined by CalculationScaleLookup calculation method.
- `rangeEnd` is the rangeStart attribute (CALRANGE.RANGESTART) of the next calculation range in the list of calculation ranges to be processed. If there are no more calculation ranges to be processed, `rangeEnd` is infinity.
- `rangeStart` is the rangeStart attribute (CALRANGE.RANGESTART) of the calculation range currently being processed.

Applicable base monetary value

This number is based on the base monetary value determined by the CalculationScaleLookup calculation method. The applicable base monetary value is determined as follows:

Non–cumulative calculation ranges:

The applicable base monetary value is the same as the base monetary value.

Cumulative calculation ranges:

The applicable base monetary value is determined by the following formula:

```
min(baseMonetaryValue, rangeEnd*unitValue) - rangeStart*unitValue
```

where:

- `baseMonetaryValue` is the base monetary value determined by the CalculationScaleLookup calculation method.
- `rangeEnd` is the rangeStart attribute (CALRANGE.RANGESTART) of the next calculation range in the list of calculation ranges to be processed. If there are no more calculation ranges to be processed, `rangeEnd` is infinity.
- `rangeStart` is the rangeStart attribute (CALRANGE.RANGESTART) of the calculation range currently being processed.
- `unitValue` is the base monetary value divided by look–up number. The base monetary value and look–up number are determined by the CalculationScaleLookup calculation method.

PercentageCalculationRangeCmdImpl

This command expects the calculation range look–up result that is not a monetary amount. If the calculation range look–up result has a currency, the currency is ignored. This CalculationRange calculation method calculates a monetary amount by dividing the calculation range look–up result by 100 and multiplying the result by the applicable base monetary value.

PerUnitAmountCalculationRangeCmdImpl

This command expects the calculation range look–up result to be a monetary amount. If the monetary amount is not in the specified currency, this command will convert the look–up result to the specified currency (if the currency conversion is defined). Otherwise, this calculation range method calculates a monetary amount by multiplying the range look–up result by the applicable part of the look–up number.

FixedAmountCalculationRangeCmdImpl

This command expects the calculation range look–up result to be a monetary amount. If the monetary amount is not in the specified currency, this command will convert the look–up result to the specified currency (if the currency conversion is defined). The range look–up result is returned as the monetary amount and no other calculations are performed by this command.

How calculation scales and calculation ranges are used

Calculation scales and calculation ranges are used by CalculationRuleCalculate calculation methods to determine a monetary amount for each OrderItem in a list. The default implementation of the CalculationRuleCalculate calculation method performs the following steps to obtain monetary amounts using calculation scales and calculation rules:

1. Sets the calculated monetary amounts for each OrderItem to zero.
2. Sets the total monetary amount for the calculation scale to zero.
3. Passes the CalculationScaleLookup calculation method a list of OrderItems. The CalculationScaleLookup calculation method returns the following values:

   - A look–up number
   - A base monetary value
   - A result multiplier
   - A set of mathematical weights, one for each OrderItem

   These values depend on the CalculationScaleLookup calculation method being called.

4. Finds calculation ranges that match the look–up number. The CalculationRuleCalculate calculation method determines which calculation scales are used to determine the calculation ranges.

   A calculation range matches a look–up number when the look–up number is greater than or equal to the starting value of the range (CALRANGE.RANGESTART). If the starting value of a calculation range is not defined, the calculation range will always match the look–up number.

   Calculation ranges are found by using the look–up number to locate ranges in the CALRANGE database table. The IDs of the found calculation ranges are used later to find look–up results.

5. Sorts the calculation ranges in ascending sequence, by the starting value of the ranges.
6. Processes each calculation range by doing the following steps:

a. Obtains calculation range look–up results for the calculation range if the look–up number is greater than or equal to the starting value of the calculation range being processed and one of the following conditions is met:

- This is the last calculation range being processed.
- The look–up number is less than the starting value of the next calculation range to be processed.
- The calculation range is cumulative.

Multiple calculation range look–up results are only permitted when the calculation range look–up results are associated with currencies. The behavior when multiple calculation range look–up results are found and they are not associated with currencies is undefined. Such configurations are discouraged.

Calculation range look–up results are obtained by using the calculation range ID to locate look–up results in the CALRLOOKUP database table.

All calculation range look–up results should either be associated with a currency or not be associated with a currency. The behavior when some calculation range look–up results are associated with a currency and some are not is undefined. Such configuration are discouraged.

b. Obtains one monetary amount for calculation range look–up results by doing one of the following:

- Processes calculation range look–up results associated with currencies. For more information, see "Processing look–up results associated with currencies".
- Processes calculation range look–up results not associated with currencies. For more information, see "Processing look–up results not associated with currencies" on page 66.

c. Multiplies the monetary amount obtained by the result multiplier and do one of the following with the resulting value:

Non–cumulative calculation ranges:
Replaces the current total monetary amount for the calculation scale with the resulting value.

Cumulative calculation ranges:
Adds the resulting value to the total monetary amount for the calculation scale.

7. Spreads the total monetary amount for the calculation scale over the OrderItems in proportion to the mathematical weight assigned to each of the OrderItems.

Processing look–up results associated with currencies

When a calculation range has multiple calculation range look–up results in multiple currencies, there are two possible situations:

1. The list of look–up result currencies contains the order currency.
2. The list of look–up result currencies does not contain the order currency

Look–up result currencies contain the order currency
There should only be one calculation range look–up result with a currency matching the order currency. If multiple calculation range look–up results sharing the order currency exist, an exception is thrown.

The calculation range look–up result is processed by calling a CalculationRange calculation method to obtain a monetary amount. For more information on how a monetary amount is obtained by calling a CalculationRange calculation method, see "Calling a CalculationRange calculation method".

Look–up result currencies do not contain the order currency

If none of the currencies of the calculation range look–up results matches the order currency, the monetary amounts for calculation range look–up results are obtained by doing the following:

1. Convert the look–up results to the order currency. If no conversion from the look–up result currency to the order currency is defined in WebSphere Commerce, the look–up result is dropped from the list of calculation range look–up results to be processed.

2. For each of the converted look–up results, a CalculationRange calculation method is called to obtain a monetary amount for each of the converted look–up results. For more information on how a monetary amount is obtained by calling a CalculationRange calculation method, see "Calling a CalculationRange calculation method".

3. Compare the monetary amounts for each of the look–up results. The lowest value is considered the valid monetary amount.

Processing look–up results not associated with currencies

If the calculation range look–up results are not associated with a currency, there should be only one look–up result found. If multiple calculation range look–up results are found that are not associated with currencies, an exception is thrown.

The calculation range look–up result is processed by calling a CalculationRange calculation method to obtain a monetary amount. For more information on how a monetary amount is obtained by calling a CalculationRange calculation method, see "Calling a CalculationRange calculation method".

Calling a CalculationRange calculation method

To obtain a monetary amount from a CalculationRange calculation method, call a CalculationRange calculation method, passing it the following parameters:

- An order currency
- A calculation range look–up result
- The applicable part of the look–up number
- The applicable base monetary value

The value of the monetary amount depends on the CalculationRange calculation method called. For information on the different CalculationRange calculation methods provided with WebSphere Commerce and definitions of the parameters required by CalculationRange calculation methods, see "CalculationRange calculation method" on page 62.

Chapter 7. Examples: Applying the calculation framework

For some of the following examples, topics not covered in this book must be understood. Shipping charge calculations require understanding of jurisdictions, jurisdiction groups, and shipping modes. Tax calculations require understanding of jurisdictions, jurisdiction groups, and tax categories. More information on these topics can be found in the online information and the *IBM WebSphere Commerce Store Developer's Guide*.

The examples do not cover the specifics of loading the calculation framework information into the WebSphere Commerce database. Information on the specifics of loading information into the database can be found in the *IBM WebSphere Commerce Store Developer's Guide*.

Only the default calculation methods and calculation usages provided with WebSphere Commerce are used in these examples. Customization of the calculation framework is not covered in any of the examples.

This chapter contains examples of implementing the following calculation usages:
- "Discounts example"
- "Shipping charges example" on page 73
- "Sales tax and shipping tax calculations example" on page 87

Discounts example

This example shows how a discount can be implemented using the calculation framework. Discounts can also be implemented using the WebSphere Commerce Accelerator. For more information on implementing discounts using the WebSphere Commerce Accelerator, refer to the online information.

Discounts example description

Your store wants to offer a discount on the Books product category for a limited time. The time period is from StartDate to EndDate. The discount offers customers a 15.00 discount on their orders when the value before taxes of products from the Books product category is 50.00 or greater.

The currency of the order is the local currency of the country or region where your store is located.

The discount does not affect any any tax calculations. Taxes are calculated using order totals before the discount.

The implementation of these tax calculations is not covered in this example.

All store customers are eligible for this discount.

Discounts example implementation

To use the calculation framework to calculate shipping charges for this example, complete the following steps:
1. Define a calculation code. For more information, see "Defining discount calculation codes" on page 68.

2. Define a calculation rule. For more information, see "Defining discount calculation rules" on page 69.

3. Define a calculation scale. For more information, see "Defining discount calculation scales" on page 69.

4. Define the calculation ranges. For more information, see "Defining discount calculation ranges" on page 70.

5. Define the look–up results for the calculation ranges. For more information, see "Defining look–up results for discount calculation ranges" on page 71.

6. Associate the calculation scales with the calculation rules. For more information, see "Associating discount calculation scales with discount calculation rules" on page 72.

7. Attach the calculation code with the Books product category catalog entry. For more information, see "Attaching discount calculation code" on page 72.

Information on concepts mentioned in the steps below but not covered by this book can be found in the online information.

Important: Text identifiers are used in the following sections to make it easier to see the relationships between components.

Text identifiers are often not valid in the WebSphere Commerce database. You should convert the text identifiers to integers or allow WebSphere Commerce to generate the identifiers for you. Check the database schema in the online information for the type of value that is valid for identifiers in a particular database table.

Defining discount calculation codes

One calculation code needs to be defined in this example because the discount in this example is independent of other discounts the store may be offering. Also, the discount only applies to one product category. The CalculationCodeQualify calculation method that must be specified as part of a calculation code does not need to be invoked in this example, so the restriction flag is not set.

Here are the key properties required for the calculation code you must define:

Table 8. BookDiscCalcCode (Discount calculation code) properties

Property	Value	Description of value
Grouping method	0	The calculation code will only be attached to one catalog group (Books). Grouping products is useful when the calculation code will be attached to multiple parent products.
Qualification flag	0	The CalculationCodeQualify calculation method specified for this calculation code will not be invoked
Identifying string	Book Discount Promotion	This is a character string that uniquely identifies this calculation code, given a particular calculation usage and store or store group.
Published	1	Calculation code is active
Start date	StartDate	This calculation code starts being active starting on StartDate
End date	EndDate	This calculation code stops being active on EndDate

Table 8. BookDiscCalcCode (Discount calculation code) properties (continued)

Property	Value	Description of value
Calculation usage ID	-1	Calculation usage ID for discount calculation usage
CalculationCodeQualify calculation method	-2	Calculation method ID for CalculationCodeQualifyCmd
CalculationCodeApply calculation method	-4	Calculation method ID for DiscountCalculationCodeApplyCmd
CalculationCodeCalculate calculation method	-3	Calculation method ID for CalculationCodeCalculateCmd

For more information on calculation codes, see Chapter 4, "Calculation codes" on page 23.

Defining discount calculation rules
One calculation rule is required to calculate this discount.

Here are the key properties for the calculation rule you must define:

Table 9. BookDiscRule properties

Property	Value	Description of value
Calculation code ID	BookDiscCode	Discount calculation code that was defined earlier
Identifying number	1	This is a number that, along with the calculation code ID, uniquely identifies this calculation rule.
Qualification flag	0	The CalculationRuleQualify calculation method specified for this calculation rule will not be invoked.
Combination	2	The results of this calculation rule can be combined with other calculation rules having a "0" or "2" combination property. In this example, the value of the combination property has no effect because only one calculation rule is being defined for this calculation code
CalculationRuleQualify calculation method	-6	Calculation method ID for DiscountCalculationRuleQualifyCmd
CalculationRuleCalculate calculation method	-7	Calculation method ID for CalculationRuleCalculateCmd

For more information on calculation rules, see Chapter 5, "Calculation rules" on page 35.

Defining discount calculation scales
The discount for this example can be expressed as one calculation scale as shown in the following table:

Table 10. Discount expressed as a calculation scale

Value of Book OrderItems	Fixed amount discount
less than 50.00	0.00

Table 10. Discount expressed as a calculation scale (continued)

Value of Book OrderItems	Fixed amount discount
50.00 or greater	15.00

This table provides the following hints about how calculation scales and calculation ranges will be implemented:

- "Value of Book OrderItems" indicates that our CalculationScaleLookup calculation method will use the sum of the prices of all Book OrderItems.
- "Fixed amount discount" indicates that our CalculationRange calculation method will return a fixed amount.
- There are two calculation ranges in this calculation scale ("less than 50.00" and "50.00 or greater").
- There are two look–up results for this calculation scale ("0.00" and "15.00").

The calculation scale should be associated with a currency since the look–up number is a monetary amount. The currency will not be indicated in this example, but it is required when implementing this example in WebSphere Commerce.

You should define one scale for each currency that your store supports. In this example, the store supports only one currency, so we will define only one scale.

The following tables indicate the key values for the calculation scales:

Table 11. BookDiscountScale calculation scale

Property	Value	Description of value
Calculation usage	-1	Discount calculation usage
Identifying string	Book Discount Promotion	This is a character string that uniquely identifies this calculation scale, given a particular calculation usage and store or store group.
CalculationScaleLookup calculation method	-10	NonDiscountedPriceCalculationScaleLookupCmd (the look–up number is the sum of the non–discounted prices of the Book OrderItems)

For more information on calculation scales, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Defining discount calculation ranges

The discount has two calculation ranges for the calculation scale. The calculation ranges are:

- value of Book OrderItems less than 50.00
- value of Book OrderItems 50.00 or greater

The calculation ranges are not cumulative.

Here are the key properties you must define for each calculation range:

Table 12. Book0to50Range calculation range properties

Property	Value	Description of value
Calculation scale ID	BookDiscountScale	Calculation scale defined earlier

Table 12. Book0to50Range calculation range properties (continued)

Property	Value	Description of value
Calculation range starting value	0.00	This calculation range applies when the sum of non-discounted prices of the OrderItems is greater than or equal to 0.00
Cumulative calculation range indicator	0	This calculation range is non–cumulative.
CalculationRange calculation method	-13	FixedAmountCalculationRangeCmd (the look–up result is treated as a fixed monetary amount)

Table 13. Book50+Range calculation range properties

Property	Value	Description of value
Calculation scale ID	BookDiscountScale	Calculation scale defined earlier
Calculation range starting value	50.00	This calculation range applies when the sum of non-discounted prices of the OrderItems is greater than or equal to 50.00
Cumulative calculation range indicator	0	This calculation range is non–cumulative
CalculationRange calculation method	-13	FixedAmountCalculationRangeCmd (the look–up result is treated as a fixed monetary amount)

For more information on calculation ranges, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Defining look–up results for discount calculation ranges

Each calculation range needs a set of look–up results. For this example, the look–up results are a fixed monetary amount. Each calculation range has only one value associated with it, so only one look–up result is needed for each range. Each look–up result must be associated with the local currency of the country or region for the store because the currency of the shipping charges is the local currency of the country or region where your store is located.

The look–up result values are added to price and other existing monetary amounts associate with an OrderItem. For a discount, the look–up result value is a negative number.

The following tables define the key properties you must define for each calculation range look–up result. The currency has been omitted here, but it must be included in an actual implementation.

Table 14. Book0to50LookupResult

Property	Value	Description of value
Look–up result	0.00	No discount
Calculation range ID	Book0to50Range	The identifier of the calculation range to which this look–up result belongs

Table 15. Book50+LookupResult

Property	Value	Description of value
Look–up result	–15.00	15.00 discount
Calculation range ID	Book50+Range	The identifier of the calculation range to which this look–up result belongs

Reminder: These calculation range look–up result properties are missing the currency for the look–up results. The currency of the result look–up results must be defined. For this example, the currency of the look–up results must match the currency of the calculation scale.

For more information on look–up results, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Associating discount calculation scales with discount calculation rules

The calculation scale that was defined must be associated with the defined calculation rule in the CRULESCALE database table. The relationship between the defined calculation scale and the defined calculation rule is shown in the following table:

Table 16. Discount calculation scale and the associated calculation rule

Calculation scales	Calculation rules
BookDiscountScale	BookDiscRule

For more information on the relationship between calculation rules and calculation scales, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Attaching discount calculation code

To use the new discount calculation code defined earlier, the calculation code must be attached to the OrderItems. The calculation code will be indirectly attached to the OrderItems because the calculation code applies to all OrderItems resulting from a customer purchasing a product from the store catalog, not one OrderItem.

The discount calculation code only applies to products belonging to the Book product category, so the calculation code will be attached to the Book product category catalog group.

We will indirectly attach the calculation code by creating an entry in the CATGPCALCD database table.

Populate the CATGPCALCD database table with the following information:

Table 17. Discount calculation code attachment information

Store ID	Catalog group ID	Calculation code
Store	BookCategory	BookDiscCalcCode

For more information on calculation code attachment, see Chapter 4, "Calculation codes" on page 23.

Shipping charges example

This example shows how the calculation framework can be used to calculate shipping charges for a store.

Shipping charges example description

Your store ships products worldwide from one fulfillment center, called FulfillmentA.

You use one shipping carrier to deliver all shipments from the fulfillment center. The shipping carrier offers two shipping methods: regular delivery and express delivery.

The shipping charges are based on the total weight of the shipment in grams. The amount of the shipping charges depends on which of three geographical zones to which the shipment will be delivered. The three geographical zones are zone A, zone B, and the rest of the world. Both geographical zone A and geographical zone B consist of a single country or region.

The shipping charges are determined using the following table:

Table 18. Shipping rates

Description of charge	Zone A		Zone B		Rest of the world	
	Regular	Express	Regular	Express	Regular	Express
Basic shipping charge for packages less than 2kg	1.50	2.75	2.00	3.50	3.00	5.00
Additional charge for each kg over 2 kg but less than 10 kg	0.75	1.00	1.25	1.75	2.00	2.50
Additional charge for each kg over 10 kg but less than 20 kg	0.50	0.75	1.00	1.50	1.75	2.00
Additional charge for each kg over 20 kg	0.25	0.50	0.75	1.25	1.50	1.75

The currency of the shipping charges is the local currency of the country or region where your store is located.

Shipping charges example implementation

To use the calculation framework to calculate shipping charges for this example, complete the following steps:

1. Define the jurisdiction groups and jurisdictions. For more information, see "Defining shipping jurisdiction groups and shipping jurisdictions" on page 74.
2. Define the shipping modes. For more information, see "Defining shipping modes" on page 75.
3. Define the calculation codes. For more information, see "Defining the shipping calculation code" on page 75.

4. Define the calculation rules. For more information, see "Defining shipping calculation rules" on page 76.

5. Define the calculation scales. For more information, see "Defining shipping charge calculation scales" on page 77.

6. Define the calculation ranges. For more information, see "Defining shipping charge calculation ranges" on page 78.

7. Define the look–up results for the calculation ranges. For more information, see "Defining look–up results for shipping charge calculation ranges" on page 80.

8. Associate the calculation scales with the calculation rules. For more information, see "Associating shipping charge calculation scale with shipping charge calculation rules" on page 85.

9. Attach the calculation code with the all catalog entries. For more information, see "Attaching shipping charge calculation codes" on page 86.

10. Associate the calculation rules with the fulfillment center. For more information, see "Associate shipping charge calculation rules with fulfillment centers" on page 86.

11. (optional) Set the default shipping mode for the store. For more information, see "Defining the default shipping mode (optional)" on page 87.

12. (optional) Set the default shipping calculation code for the store. For more information , see "Defining default shipping calculation codes for a store (optional)" on page 87. You would only complete this step if your store does not already have a default calculation code defined for shipping charges.

Information on concepts mentioned in the steps below but not covered by this book can be found in the online information.

Important: Text identifiers are used in the following sections to make it easier to see the relationships between components.

Text identifiers are often not valid in the WebSphere Commerce database. You should convert the text identifiers to integers or allow WebSphere Commerce to generate the identifiers for you. Check the database schema in the online information for the type of value that is valid for identifiers in a particular database table.

Defining shipping jurisdiction groups and shipping jurisdictions
The first step in using the calculation framework for shipping charge calculations is to define the shipping jurisdiction groups to which shipments are delivered. Jurisdiction groups consist of individual jurisdictions. The relationships defining which jurisdiction belong to which jurisdiction groups must also be defined.

Jurisdictions and jurisdiction groups are used by the ShippingCalculationRuleQualify calculation method to determine which calculation rule to use, based on the shipping address of the order.

The information for jurisdictions, jurisdiction groups, and the relationships between them are defined in the following database tables:

JURSTGROUP
   This table defines jurisdiction groups.

JURST
   This table defines jurisdictions.

JURSTGPREL

> This table defines which jurisdictions belong to which jurisdiction groups.

For more information on these database tables, refer to the database schema information in the online help.

Based on the description in "Shipping charges example description" on page 73, the jurisdictions, jurisdiction groups and the relationships between them can be defined based on the following table:

Table 19. Jurisdictions and jurisdiction groups for the shipping calculations example

Geographical zone	Jurisdiction	Jurisdiction group
Zone A	Country or region A	GroupA
Zone B	Country or region B	GroupB
Rest of the world	All countries or regions in the world, including country or region A and country or region B	World

We later take advantage of the precedence property when we associate shipping charge calculation rules with the three jurisdiction groups in "Associate shipping charge calculation rules with fulfillment centers" on page 86.

Defining shipping modes

A shipping mode represents a combination of a shipping carrier and a shipping service. Shipping mode information is stored in the SHIPMODE and SHPMODEDSC database tables.

The example store uses only one carrier that provides two different delivery methods, so the example store will have two shipping modes as follows:

Table 20. Shipping modes

Shipping mode	Description
Regular	Regular delivery mode provided by store's shipping carrier
Express	Express delivery mode provided by store's shipping carrier

Defining the shipping calculation code

One calculation code needs to be defined in this example because the shipping charges apply to all items in an order. The calculation code must be associated with a CalculationCodeQualify calculation method so that the correct calculation rule can be chosen based on the address of the order.

Here are the key properties you must define for the calculation code:

Table 21. ShipCalcCode (Shipping calculation code) properties

Property	Value	Description of value
Grouping method	0	No grouping since shipping charges are based on the total weight of all items in an order
Qualification flag	0	The CalculationCodeQualify calculation method specified for this calculation code will not be invoked

Table 21. ShipCalcCode (Shipping calculation code) properties (continued)

Property	Value	Description of value
Identifying string	Shipping example	This is a character string that uniquely identifies this calculation code, given a particular calculation usage and store or store group.
Published	1	Calculation code is active
Calculation usage ID	-2	Calculation usage ID for shipping calculation usage
CalculationCodeQualify calculation method	-22	Calculation method ID for CalculationCodeQualifyCmd
CalculationCodeApply calculation method	-24	Calculation method ID for ShippingCalculationCodeApplyCmd
CalculationCodeCalculate calculation method	-23	Calculation method ID for CalculationCodeCalculateCmd

For more information on calculation codes, see Chapter 4, "Calculation codes" on page 23.

Defining shipping calculation rules

A calculation rule is required for each shipping mode for each jurisdiction group. This example requires six calculation rules:

- Calculate regular shipping charge for GroupA jurisdiction group.
- Calculate express shipping charge for GroupA jurisdiction group.
- Calculate regular shipping charge for GroupB jurisdiction group.
- Calculate express shipping charge for GroupB jurisdiction group.
- Calculate regular shipping charge for World jurisdiction group.
- Calculate express shipping charge for World jurisdiction group.

These rules are all shipping calculation rules, so you should specify that the ShippingCalculationRuleQualify calculation method should be used with these calculation rules.

For this example, all of the calculation rules will have the same properties as shown in the following table:

Table 22. Values for shipping charge calculation rules

Property	Value	Description of value
Calculation code	ShipCalcCode	Shipping charge calculation code defined earlier
Identifying number	*a different number for each calculation rule*	This is a number that, along with its calculation code, uniquely identifies this calculation rule. This number must be unique for each calculation rule belonging to the same calculation code.
Qualification flag	1	Invoke the qualify calculation method defined for this calculation rule to check that this calculation rule applies to each item in the order

Table 22. Values for shipping charge calculation rules (continued)

Property	Value	Description of value
Combination	1	The results of this calculation rule can be combined with other calculation rules having a "0" combination property
CalculationRuleQualify calculation method	-26	Calculation method ID for ShippingCalculationRuleQualifyCmd
CalculationRuleCalculate calculation method	-27	Calculation method ID for CalculationRuleCalculateCmd

For the rest of the example, these calculation rules will be referred to as follows:

- GroupARegularRule (GroupA regular shipping charge calculation rule)
- GroupAExpressRule (GroupA express shipping charge calculation rule)
- GroupBRegularRule (GroupB regular shipping charge calculation rule)
- GroupBExpressRule (GroupB express shipping charge calculation rule)
- WorldRegularRule (World regular shipping charge calculation rule)
- WorldExpressRule (World express shipping charge calculation rule)

For more information on calculation rules, see Chapter 5, "Calculation rules" on page 35.

Defining shipping charge calculation scales

A calculation scale is required for each shipping mode for each jurisdiction group. The calculation scales for this example are based on the weight of the order in kilograms, so the calculation scales must be associated with a unit of measurement.

The shipping charges are based on the total weight of the OrderItems in an order, so the WeightCalculationScaleLookupCmd calculation method is used at the CalculationScaleLookup calculation method for this calculation scale.

This example requires six calculation scales as follows:

- GroupA regular shipping charge scale
- GroupA express shipping charge scale
- GroupB regular shipping charge scale
- GroupB express shipping charge scale
- World regular shipping charge scale
- World express shipping charge scale

For this example, all of the calculation scales will have the same properties as shown in the following table:

Table 23. Values for shipping charge calculation scales

Property	Value	Description of value
Calculation usage	-2	Shipping charge calculation usage
Identifying string	Shipping example	This is a character string that uniquely identifies this calculation scale, given a particular calculation usage and store or store group.
CalculationScaleLookup calculation method	-29	WeightCalculationScaleLookupCmd

Table 23. Values for shipping charge calculation scales (continued)

Property	Value	Description of value
Unit of measurement	KGM	This calculation scale is based on weights in kilograms. The calculation scale look–up number is in kilograms as well.

For the rest of the example, these calculation scales will be referred to as follows:
- GroupARegularScale (GroupA regular shipping charge calculation scale)
- GroupAExpressScale (GroupA express shipping charge calculation scale)
- GroupBRegularScale (GroupB regular shipping charge calculation scale)
- GroupBExpressScale (GroupB express shipping charge calculation scale)
- WorldRegularScale (World regular shipping charge calculation scale)
- WorldExpressScale (World express shipping charge calculation scale)

For more information on calculation scales, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Defining shipping charge calculation ranges

Each of the calculation scales defined in the previous step requires a set of calculation ranges. Each calculation scale is associated with four calculation ranges resulting in a total of 24 calculation ranges that must be defined for this example. The four calculation ranges for each calculation scale are:
- 0–2 kg (basic shipping charge for packages less than 2kg)
- 2–10 kg (additional charge for each kg over 2 kg but less than 10 kg)
- 10–20 kg (additional charge for each kg over 10 kg but less than 20 kg)
- 20 kg or greater (additional charge for each kg over 20 kg)

The calculation ranges are all cumulative. The results from one or more of these calculation ranges will be added to determine the final shipping charge associated with each OrderItem in an order.

Here are the key properties you must define for each calculation range:

Table 24. 0–2 kg calculation range properties

Property	Value	Description of value
Calculation range starting value	0	This calculation range applies to any OrderItem where the total order weight is 0 or more kilograms
Cumulative calculation range indicator	1	This calculation range is cumulative.
CalculationRange calculation method	-33	FixedAmountCalculationRangeCmd (the look–up result is treated as a fixed monetary amount)

Table 25. 2–10 kg calculation range properties

Property	Value	Description of value
Calculation range starting value	2	This calculation range applies to any OrderItem where the total order weight is 2 or more kilograms
Cumulative calculation range indicator	1	This calculation range is cumulative

Table 25. 2–10 kg calculation range properties (continued)

Property	Value	Description of value
CalculationRange calculation method	-34	PerUnitAmountCalculationRangeCmd (the look–up result is multiplied by the part of the total order weight that falls into this range)

Table 26. 10–20 kg calculation range properties

Property	Value	Description of value
Calculation range starting value	10	This calculation range applies to any OrderItem where the total order weight is 10 or more kilograms
Cumulative calculation range indicator	1	This calculation range is cumulative
CalculationRange calculation method	-34	PerUnitAmountCalculationRangeCmd (the look–up result is multiplied by the part of the total order weight that falls into this range)

Table 27. 20 kg or greater calculation range properties

Property	Value	Description of value
Calculation range starting value	20	This calculation range applies to any OrderItem where the total order weight is 20 or more kilograms
Cumulative calculation range indicator	1	This calculation range is cumulative
CalculationRange calculation method	-34	PerUnitAmountCalculationRangeCmd (the look–up result is multiplied by the part of the total order weight that falls into this range)

For the rest of the example, these calculation ranges will be referred to as follows:

- GroupARegular0to2Range
- GroupARegular2to10Range
- GroupARegular10to20Range
- GroupARegular20+Range
- GroupAExpress0to2Range
- GroupAExpress2to10Range
- GroupAExpress10to20Range
- GroupAExpress20+Range
- GroupBRegular0to2Range
- GroupBRegular2to10Range
- GroupBRegular10to20Range
- GroupBRegular20+Range
- GroupBExpress0to2Range
- GroupBExpress2to10Range
- GroupBExpress10to20Range
- GroupBExpress20+Range
- WorldRegular0to2Range
- WorldRegular2to10Range

- WorldRegular10to20Range
- WorldRegular20+Range
- WorldExpress0to2Range
- WorldExpress2to10Range
- WorldExpress10to20Range
- WorldExpress20+Range

You must specify the calculation scale ID of the calculation scale to which the calculation range belongs when defining a calculation range. The relationships between the defined calculation scales and the calculation ranges we are defining is shown in the following table:

Table 28. Shipping charge calculation ranges and their associated calculation scales

Calculation ranges	Calculation scales
GroupARegular0to2Range	GroupARegularScale
GroupARegular2to10Range	
GroupARegular10to20Range	
GroupARegular20+Range	
GroupAExpress0to2Range	GroupAExpressScale
GroupAExpress2to10Range	
GroupAExpress10to20Range	
GroupAExpress20+Range	
GroupBRegular0to2Range	GroupBRegularScale
GroupBRegular2to10Range	
GroupBRegular10to20Range	
GroupBRegular20+Range	
GroupBExpress0to2Range	GroupBExpressScale
GroupBExpress2to10Range	
GroupAExpress10to20Range	
GroupBExpress20+Range	
WorldRegular0to2Range	WorldRegularScale
WorldRegular2to10Range	
WorldRegular10to20Range	
WorldRegular20+Range	
WorldExpress0to2Range	WorldExpressScale
WorldExpress2to10Range	
WorldExpress10to20Range	
WorldExpress20+Range	

For more information on calculation ranges, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Defining look–up results for shipping charge calculation ranges

Each calculation range needs a set of look–up results. Each calculation range has only one value associated with it, so only one look–up result is needed for each range. Each look–up result must be associated with the local currency of the

country or region for the store because the currency of the shipping charges is the local currency of the country or region where your store is located.

The following tables define the key properties you must define for each calculation range look–up result. The currency has been omitted here, but it must be included in an actual implementation.

Table 29. GroupARegular0to2LookupResult

Property	Value	Description of value
Look–up result	1.50	1.50 basic shipping charge for order weights under 2kg
Calculation range ID	GroupARegular0to2Range	The identifier of the calculation range to which this look–up result belongs

Table 30. GroupARegular2to10LookupResult

Property	Value	Description of value
Look–up result	0.75	0.75 additional charge per kilogram for order weights greater than or equal to 2kg
Calculation range ID	GroupARegular2to10Range	The identifier of the calculation range to which this look–up result belongs

Table 31. GroupARegular10to20LookupResult

Property	Value	Description of value
Look–up result	0.50	0.50 additional charge per kilogram for order weights greater than or equal to 10kg
Calculation range ID	GroupARegular10to20Range	The identifier of the calculation range to which this look–up result belongs

Table 32. GroupARegular20+LookupResult

Property	Value	Description of value
Look–up result	0.25	0.25 additional charge per kilogram for order weights greater than or equal to 20kg
Calculation range ID	GroupARegular20+Range	The identifier of the calculation range to which this look–up result belongs

Table 33. GroupAExpress0to2LookupResult

Property	Value	Description of value
Look–up result	2.75	2.75 basic shipping charge for order weights under 2kg
Calculation range ID	GroupAExpress0to2Range	The identifier of the calculation range to which this look–up result belongs

Table 34. GroupAExpress2to10LookupResult

Property	Value	Description of value
Look–up result	1.00	1.00 additional charge per kilogram for order weights greater than or equal to 2kg
Calculation range ID	GroupAExpress2to10Range	The identifier of the calculation range to which this look–up result belongs

Table 35. GroupAExpress10to20LookupResult

Property	Value	Description of value
Look–up result	0.75	0.75 additional charge per kilogram for order weights greater than or equal to 10kg
Calculation range ID	GroupAExpress10to20Range	The identifier of the calculation range to which this look–up result belongs

Table 36. GroupAExpress20+LookupResult

Property	Value	Description of value
Look–up result	0.50	0.50 additional charge per kilogram for order weights greater than or equal to 20kg
Calculation range ID	GroupAExpress20+Range	The identifier of the calculation range to which this look–up result belongs

Table 37. GroupBRegular0to2LookupResult

Property	Value	Description of value
Look–up result	2.00	2.00 basic shipping charge for order weights under 2kg
Calculation range ID	GroupBRegular0to2Range	The identifier of the calculation range to which this look–up result belongs

Table 38. GroupBRegular2to10LookupResult

Property	Value	Description of value
Look–up result	1.25	1.25 additional charge per kilogram for order weights greater than or equal to 2kg
Calculation range ID	GroupBRegular2to10Range	The identifier of the calculation range to which this look–up result belongs

Table 39. GroupBRegular10to20LookupResult

Property	Value	Description of value
Look–up result	1.00	1.00 additional charge per kilogram for order weights greater than or equal to 10kg

Table 39. GroupBRegular10to20LookupResult (continued)

Property	Value	Description of value
Calculation range ID	GroupBRegular10to20Range	The identifier of the calculation range to which this look–up result belongs

Table 40. GroupBRegular20+LookupResult

Property	Value	Description of value
Look–up result	0.75	0.75 additional charge per kilogram for order weights greater than or equal to 20kg
Calculation range ID	GroupBRegular20+Range	The identifier of the calculation range to which this look–up result belongs

Table 41. GroupBExpress0to2LookupResult

Property	Value	Description of value
Look–up result	3.50	3.50 basic shipping charge for order weights under 2kg
Calculation range ID	GroupBExpress0to2Range	The identifier of the calculation range to which this look–up result belongs

Table 42. GroupBExpress2to10LookupResult

Property	Value	Description of value
Look–up result	1.75	1.75 additional charge per kilogram for order weights greater than or equal to 2kg
Calculation range ID	GroupBExpress2to10Range	The identifier of the calculation range to which this look–up result belongs

Table 43. GroupBExpress10to20LookupResult

Property	Value	Description of value
Look–up result	1.50	1.50 additional charge per kilogram for order weights greater than or equal to 10kg
Calculation range ID	GroupBExpress10to20Range	The identifier of the calculation range to which this look–up result belongs

Table 44. GroupBExpress20+LookupResult

Property	Value	Description of value
Look–up result	1.25	1.25 additional charge per kilogram for order weights greater than or equal to 20kg
Calculation range ID	GroupBExpress20+Range	The identifier of the calculation range to which this look–up result belongs

Table 45. WorldRegular0to2LookupResult

Property	Value	Description of value
Look–up result	3.00	3.00 basic shipping charge for order weights under 2kg
Calculation range ID	WorldRegular0to2Range	The identifier of the calculation range to which this look–up result belongs

Table 46. WorldRegular2to10LookupResult

Property	Value	Description of value
Look–up result	2.00	2.00 additional charge per kilogram for order weights greater than or equal to 2kg
Calculation range ID	WorldRegular2to10Range	The identifier of the calculation range to which this look–up result belongs

Table 47. WorldRegular10to20LookupResult

Property	Value	Description of value
Look–up result	1.75	0.50 additional charge per kilogram for order weights greater than or equal to 10kg
Calculation range ID	WorldRegular10to20Range	The identifier of the calculation range to which this look–up result belongs

Table 48. WorldRegular20+LookupResult

Property	Value	Description of value
Look–up result	1.50	0.25 additional charge per kilogram for order weights greater than or equal to 20kg
Calculation range ID	WorldRegular20+Range	The identifier of the calculation range to which this look–up result belongs

Table 49. WolrdExpress0to2LookupResult

Property	Value	Description of value
Look–up result	5.00	5.00 basic shipping charge for order weights under 2kg
Calculation range ID	WorldExpress0to2Range	The identifier of the calculation range to which this look–up result belongs

Table 50. WorldExpress2to10LookupResult

Property	Value	Description of value
Look–up result	2.50	2.50 additional charge per kilogram for order weights greater than or equal to 2kg

Table 50. WorldExpress2to10LookupResult (continued)

Property	Value	Description of value
Calculation range ID	WorldExpress2to10Range	The identifier of the calculation range to which this look–up result belongs

Table 51. WorldExpress10to20LookupResult

Property	Value	Description of value
Look–up result	2.00	2.00 additional charge per kilogram for order weights greater than or equal to 10kg
Calculation range ID	WorldExpress10to20Range	The identifier of the calculation range to which this look–up result belongs

Table 52. WorldExpress20+LookupResult

Property	Value	Description of value
Look–up result	1.75	1.75 additional charge per kilogram for order weights greater than or equal to 20kg
Calculation range ID	WorldExpress20+Range	The identifier of the calculation range to which this look–up result belongs

Reminder: These calculation range look–up result properties are missing the currency for the look–up results. The currency of the look–up results must be defined. For this example, the currency of the look–up results must match the currency of the calculation scale.

For more information on look–up results, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Associating shipping charge calculation scale with shipping charge calculation rules

Each of the calculation scales that have been defined must be associated with the defined calculation rules in the CRULESCALE database table. The relationship between the defined calculation scales and the defined calculation rules is shown in the following table:

Table 53. Shipping calculation scales and the associated calculation rules

Calculation scale	Calculation rule
GroupARegularScale	GroupARegularRule
GroupAExpressScale	GroupAExpressRule
GroupBRegularScale	GroupBRegularRule
GroupBExpressScale	GroupBExpressRule
WorldRegularScale	WorldRegularRule
WorldExpressScale	WorldExpressRule

For more information on the relationship between calculation rules and calculation scales, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Attaching shipping charge calculation codes

To use the new shipping charge calculation code defined earlier, the calculation code must be attached to the OrderItems. The calculation code will be indirectly attached to the OrderItems because the calculation code applies to all OrderItems resulting from a customer purchasing a product from the store catalog, not one OrderItem.

We will indirectly attach the calculation code by creating an entry in the CATENCALCD database table. To associate the calculation code with all items in the store catalog, use a catalog entry ID of *null*. The identifier for the store will be called *Store.*

Populate the CATENCALCD database table with the following information:

Table 54. Shipping charge calculation code attachment information

Store ID	Catalog entry ID	Calculation code
Store	*null*	ShipCalcCode

For more information on calculation code attachment, see Chapter 4, "Calculation codes" on page 23.

Associate shipping charge calculation rules with fulfillment centers

For the ShippingCalculationRuleQualify calculation method to chose a calculation rule when shipping from a fulfillment center to a shipping address that matches one of the shipping jurisdiction groups, information associating a fulfillment center with a shipping charge calculation rule must be entered in the SHIPJCRULE database table.

We will take advantage of the precedence property of the relationship between calculation rules and fulfillment centers. We will set precedence property values for relationships involving the GroupA or GroupB shipping jurisdiction group to 1 and the precedence property values for relationships involving the World shipping jurisdiction group to 0. This ensures that if a shipping address falls into either the GroupA or GroupB jurisdiction group and the World jurisdiction group, the calculation rule applicable to GroupA or GroupB takes precedence over the calculation rule applicable to the World jurisdiction group.

We have one fulfillment center, three shipping jurisdiction groups, and six calculation rules in this example. The relationships between these elements is defined in the following table:

Table 55. Information relating calculation rules to fulfillment centers

Fulfillment center	Shipping jurisdiction group	Precedence	Shipping charge calculation rule
FulfillmentA	GroupA	1	GroupARegular
			GroupAExpress
	GroupB	1	GroupBRegular
			GroupBExpress
	World	0	WorldRegular
			WorldExpress

Defining the default shipping mode (optional)

You may want to define a default shipping mode for a store or a contract. Shipping modes are associated with a store and contract in the STOREDEF database table.

To determine what values are required to set a shipping mode as the default shipping mode, refer to the database schema documentation in the online information.

Defining default shipping calculation codes for a store (optional)

If your store does not already have a default calculation code for shipping charges defined, you may wish to make the calculation code you defined earlier your default shipping charge calculation code. A store can only have one default shipping calculation code.

Default calculation codes for a store are defined in the STENCALUSG database table. The following table contains the information required to define the default shipping calculation code for the store:

Table 56. Information to define default calculation codes for the store

Store ID	Calculation Usage	Calculation Code
Store	-2 (shipping)	ShipCalcCode

Generally, a shipping calculation code is not both attached to all catalog entries in a store and defined as the default shipping calculation code. When a shipping calculation code is attached to all catalog entries in a store, additional shipping calculation codes that are defined and attached later will cause the total shipping charge to be the sum of shipping charges resulting from each calculation code. If a default shipping calculation code is defined and no shipping calculation codes are attached to the catalog entries or catalog groups, you can have different shipping rate for individual catalog entries or catalog groups. Attaching a shipping calculation code to a catalog entry or catalog group overrides the default shipping calculation code.

Sales tax and shipping tax calculations example

This example covers how the calculation framework can be used to calculate sales and shipping taxes.

Sales tax and shipping tax calculation example description

Your store ships products worldwide from one fulfillment center.

You must collect a 15% sales tax on orders shipped to geographical zone A and a 7% sales tax on orders shipped to geographical zone B. You do not need to collect sales tax for orders shipped to other regions in the world. You must also collect taxes on shipping charges for orders shipped to geographical zone A and geographical zone B. For shipments to addresses in geographical zone A, shipping charges are subject to a 15% tax, while shipping charges are subject to a 4% tax for shipments to addresses in geographical zone B.

Both geographical zone A and geographical zone B consist of a single country or region.

The following tables summarize the taxes that must be collected:

Table 57. Sales tax rates

Geographical zone	Tax rate
A	15%
B	7%
The rest of the world	0%

Table 58. Shipping tax rates

Geographical zone	Tax rate
A	15%
B	4%
The rest of the world	0%

Sales tax and shipping tax calculation example implementation

To use the calculation framework to calculate taxes for this example, complete the following steps:

1. Define the jurisdiction groups and jurisdictions. For more information, see "Defining tax jurisdiction groups and tax jurisdictions" on page 89.

2. Define the tax categories. For more information, see "Defining tax categories" on page 89.

3. Define the calculation codes. For more information, see "Defining tax calculation codes" on page 90.

4. Define the calculation rules. For more information, see "Defining tax calculation rules" on page 91.

5. Define the calculation scales. For more information, see "Defining tax calculation scales" on page 93.

6. Define the calculation ranges. For more information, see "Defining tax calculation ranges" on page 95.

7. Define the look–up results for the calculation ranges. For more information, see "Defining look–up results for tax calculation ranges" on page 96.

8. Associate the calculation scales with the calculation rules. For more information, see "Associating tax calculation scales with tax calculation rules" on page 97.

9. Attach the calculation codes with the all catalog entries. For more information, see "Attaching tax calculation codes" on page 97.

10. Associate the calculation rules with the fulfillment center. For more information, see "Associating tax calculation rules with fulfillment centers" on page 98.

11. Allow the calculation usages to not calculate a value for an OrderItem. For more information, see "Allow sales tax and shipping tax calculation usages to not calculate a value for an OrderItem" on page 98.

12. (optional) Set the default sales tax and shipping tax calculation codes for the store. For more information, see "Defining default tax calculation codes for a store (optional)" on page 98. You would only complete this step if your store does not already have defined default calculation codes for sales tax or shipping tax.

Important: Text identifiers are used in the following sections to make it easier to see the relationships between components.

Text identifiers are often not valid in the WebSphere Commerce database. You should convert the text identifiers to integers or allow WebSphere Commerce to generate the identifiers for you. Check the database schema in the online information for the type of value that is valid for identifiers in a particular database table.

Defining tax jurisdiction groups and tax jurisdictions

The first step in using the calculation framework for tax calculations is to define the jurisdiction groups in which taxes are to be collected. Jurisdiction groups consist of individual jurisdictions. The relationships defining which jurisdiction belongs to which jurisdiction groups must also be defined.

Jurisdictions and jurisdiction groups are used by TaxCalculationRuleQualify calculation method to determine which calculation rule is used to calculate taxes based on the shipping address of the order.

The information for jurisdictions, jurisdiction groups, and the relationships between them are defined in the following database tables:

JURSTGROUP
>   This table defines jurisdiction groups.

JURST
>   This table defines jurisdictions.

JURSTGPREL
>   This table defines which jurisdictions belong to which jurisdiction groups.

For more information on these database tables, refer to the database schema information in the online help.

Based on the description in "Sales tax and shipping tax calculation example description" on page 87, the jurisdictions, jurisdiction groups and the relationships between them can be defined based on the following table:

Table 59. Jurisdictions and jurisdiction groups for the tax calculations example

Geographical zone	Jurisdiction	Jurisdiction group
A	A	GroupA
B	B	GroupB

We later take advantage of the precedence property when we associate sales tax and shipping tax calculation rules with jurisdiction groups in "Associating tax calculation rules with fulfillment centers" on page 98.

Defining tax categories

A tax category is required for each tax type/jurisdiction group combinations. Separate tax categories must be defined for sales taxes and shipping taxes because sales taxes and shipping taxes are defined as different tax types in the TAXTYPE database table. Default values for the tax types for sales and shipping taxes are defined in the language independent bootstrap file. For more information on bootstrap files, see "Bootstrap files" on page 111.

We do not need to define tax categories for the rest of the world because we will be handling the 0% tax rate differently in this example. For more information, see "Allow sales tax and shipping tax calculation usages to not calculate a value for an OrderItem" on page 98.

This example requires four tax categories, two for the GroupA jurisdiction group and two for the GroupB jurisdiction group. The following table shows the tax categories that can be defined based on the example description:

Table 60. Tax categories for the tax calculations example

Tax category	Tax type	Description
GroupA_SalesTax	-3 (sales tax)	Geographical zone A, Sales tax (15%)
GroupA_ShipTax	-4 (shipping tax)	Geographical zone A, Shipping tax (15%)
GroupB_SalesTax	-3 (sales tax)	Geographical zone B, Sales tax (7%)
GroupB_ShipTax	-4 (shipping tax)	Geographical zone B, Shipping tax (4%)

Tax category information is defined the following database tables:

TAXCGRY
> This table defines tax categories.

TAXCGRYDS
> This table allows descriptions in multiple languages to be associated with a tax category.

For more information on these database tables, refer to the database schema information in the online help.

Each tax category is assigned a unique identifier. The unique identifiers are used to associate calculation rules with tax categories.

Defining tax calculation codes

Only two calculation codes need to be defined in this example because the taxes apply to all items in an order. One calculation code will indicate sale taxes and the other calculation code will indicate shipping taxes.

Both calculation codes should be associated with a CalculationCodeQualify calculation method.

Here are the key properties you must define for each calculation code:

Table 61. SalesTaxCalcCode (sales tax calculation code) properties

Property	Value	Description of value
Grouping method	0	No grouping of products is required
Qualification flag	0	The CalculationCodeQualify calculation method specified for this calculation code will not be invoked
Identifying string	Sales tax example code	This is a character string that uniquely identifies this calculation code, given a particular calculation usage and store or store group.
Published	1	Calculation code is active
Calculation usage	-3	Calculation usage ID for sales tax calculation usage

Table 61. SalesTaxCalcCode (sales tax calculation code) properties (continued)

Property	Value	Description of value
CalculationCodeQualify calculation method	-42	Calculation method ID for CalculationCodeQualifyCmd
CalculationCodeApply calculation method	-44	Calculation method ID for SalesTaxCalculationCodeApplyCmd
CalculationCodeCalculate calculation method	-43	Calculation method ID for CalculationCodeCalculateCmd

Table 62. ShipTaxCalcCode (shipping tax calculation code) properties

Property	Value	Description of value
Grouping method	0	No grouping of products is required
Qualification flag	0	The CalculationCodeQualify calculation method specified for this calculation code will not be invoked
Identifying string	Shipping tax example code	This is a character string that uniquely identifies this calculation code, given a particular calculation usage and store or store group.
Published	1	Calculation code is active
Calculation usage	-4	Calculation usage ID for shipping tax calculation usage
CalculationCodeQualify calculation method	-62	Calculation method ID for CalculationCodeQualifyCmd
CalculationCodeApply calculation method	-64	Calculation method ID for ShippingTaxCalculationCodeApplyCmd
CalculationCodeCalculate calculation method	-63	Calculation method ID for CalculationCodeCalculateCmd

For more information on calculation codes, see Chapter 4, "Calculation codes" on page 23.

Defining tax calculation rules

Generally, percentage tax calculations require one calculation rule for each calculation code and each tax category. For this example, four tax calculation rules are required:

- Sales tax calculation code and the GroupA_SalesTax tax category
- Shipping tax calculation code and the GroupA_ShipTax tax category
- Sales tax calculation code and the GroupB_SalesTax tax category
- Shipping tax calculation code and the GroupB_ShippingTax tax category

These rules are all tax calculation rules, so you should specify that the TaxCalculationRuleQualify calculation method should be used with these calculation rules.

Here are the key properties you must define for each calculation rule:

Table 63. GroupASalesRule properties (calculate 15% sales taxes in the GroupA jurisdiction group)

Property	Value	Description of value
Calculation code	SalesTaxCalcCode	Sales tax calculation code defined earlier
Identifying number	1	This is a number that, along with its calculation code, uniquely identifies this calculation rule.
Qualification flag	1	Invoke the qualify calculation method defined for this calculation rule to check that this calculation rule applies to each item in the order
Combination	2	The results of this calculation rule can be combined with other calculation rules having a ″0″ or ″2″ combination property
Tax category	GroupA_SalesTax	Calculation usage ID for sales tax calculation usage
CalculationRuleQualify calculation method	-46	Calculation method ID for TaxCalculationRuleQualifyCmd
CalculationRuleCalculate calculation method	-47	Calculation method ID for CalculationRuleCalculateCmd

Table 64. GroupAShipRule properties (calculate 15% shipping taxes in the GroupA jurisdiction group)

Property	Value	Description of value
Calculation code	ShipTaxCalcCode	Shipping tax calculation code defined earlier
Identifying string	GroupA shipping tax example	This is a character string that, along with its calculation code, uniquely identifies this calculation rule.
Qualification flag	1	Invoke the qualify calculation method defined for this calculation rule to check that this calculation rule applies to each item in the order
Combination	2	The results of this calculation rule can be combined with other calculation rules having a ″0″ or ″2″ combination property
Tax category	GroupA_ShipTax	Calculation usage ID for shipping tax calculation usage
CalculationRuleQualify calculation method	-66	Calculation method ID for TaxCalculationRuleQualifyCmd
CalculationRuleCalculate calculation method	-67	Calculation method ID for CalculationRuleCalculateCmd

Table 65. GroupBSalesRule properties (calculate 7% sales taxes in the GroupB jurisdiction group)

Property	Value	Description of value
Calculation code	SalesTaxCalcCode	Sales tax calculation code defined earlier
Identifying string	GroupB sales tax example	This is a character string that, along with its calculation code, uniquely identifies this calculation rule.
Qualification flag	1	Invoke the qualify calculation method defined for this calculation rule to check that this calculation rule applies to each item in the order
Combination	2	The results of this calculation rule can be combined with other calculation rules having a "0" or "2" combination property
Tax category	GroupB_SalesTax	Calculation usage ID for sales tax calculation usage
CalculationRuleQualify calculation method	-46	Calculation method ID for TaxCalculationRuleQualifyCmd
CalculationRuleCalculate calculation method	-47	Calculation method ID for CalculationRuleCalculateCmd

Table 66. GroupBShipRule properties (calculate 4% shipping taxes in the GroupB jurisdiction group)

Property	Value	Description of value
Calculation code	ShipTaxCalcCode	Shipping tax calculation code defined earlier
Identifying string	GroupB shipping tax example	This is a character string that, along with its calculation code, uniquely identifies this calculation rule.
Qualification flag	1	Check that this calculation rule applies to each item in the order
Combination	2	The results of this calculation rule can be combined with other calculation rules having a "0" or "2" combination property
Tax category	GroupB_ShipTax	Calculation usage ID for shipping tax calculation usage
CalculationRuleQualify calculation method	-66	Calculation method ID for TaxCalculationRuleQualifyCmd
CalculationRuleCalculate calculation method	-67	Calculation method ID for CalculationRuleCalculateCmd

For more information on calculation rules, see Chapter 5, "Calculation rules" on page 35.

Defining tax calculation scales

A calculation scale is required for each calculation usage/jurisdiction group combination. Calculation scales must be defined to cover every taxation level. No quantity unit or currency needs to be specified for this calculation scale because it

is a percentage calculation scale and all its calculation ranges start with zero, which is convertible to all currencies. The calculation scale is used to find the percentage tax rate.

We do not need to define calculation scales for the rest of the world group because we handle the 0% tax rate differently in this example. For more information, see "Allow sales tax and shipping tax calculation usages to not calculate a value for an OrderItem" on page 98.

This example requires four calculation scales as follows:
- GroupA sales tax
- GroupA shipping tax
- GroupB sales tax
- GroupB shipping tax

Here are the key properties you must define for each calculation scale:

Table 67. GroupASalesScale (GroupA sales tax calculation scale)

Property	Value	Description of value
Calculation usage	-3	Sales tax calculation usage
Identifying string	Taxation example	This is a character string that uniquely identifies this calculation scale, given a particular calculation usage and store or store group.
CalculationScaleLookup calculation method	-53	TaxableNetPriceCalculationScaleLookupCmd

Table 68. GroupAShipScale (GroupA shipping tax calculation scale)

Property	Value	Description of value
Calculation usage	-4	Shipping tax calculation usage
Identifying string	Taxation example	This is a character string that uniquely identifies this calculation scale, given a particular calculation usage and store or store group.
CalculationScaleLookup calculation method	-73	NetShippingCalculationScaleLookupCmd

Table 69. GroupBSalesScale (GroupB sales tax calculation scale)

Property	Value	Description of value
Calculation usage	-3	Sales tax calculation usage
Identifying string	Taxation example	This is a character string that uniquely identifies this calculation scale, given a particular calculation usage and store or store group.
CalculationScaleLookup calculation method	-53	TaxableNetPriceCalculationScaleLookupCmd

Table 70. GroupBShipScale (GroupB shipping tax calculation scale)

Property	Value	Description of value
Calculation usage	-4	Shipping tax calculation usage
Identifying string	Taxation example	This is a character string that uniquely identifies this calculation scale, given a particular calculation usage and store or store group.

Table 70. GroupBShipScale (GroupB shipping tax calculation scale) (continued)

Property	Value	Description of value
CalculationScaleLookup calculation method	-73	NetShippingCalculationScaleLookupCmd

For more information on calculation scales, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Defining tax calculation ranges

Because the tax rate applies to all items in an order regardless of the item price, weight, or quantity, only one calculation range is required for each calculation scale.

Here are the key properties you must define for each calculation range:

Table 71. GroupASalesRange (GroupA sales tax calculation range)

Property	Value	Description of value
Calculation scale ID	GroupASalesScale	Calculation scale that was defined earlier
Calculation range starting value	0	Any total net price for an order greater than or equal to zero is subject to this range
Cumulative calculation range indicator	0	This calculation range is non–cumulative
CalculationRange calculation method	-59	PercentageCalculationRangeCmd

Table 72. GroupAShippingRange (GroupA shipping tax calculation range)

Property	Value	Description of value
Calculation scale ID	GroupAShipScale	Calculation scale that was defined earlier
Calculation range starting value	0	Any total net price for an order greater than or equal to zero is subject to this range
Cumulative calculation range indicator	0	This calculation range is non–cumulative
CalculationRange calculation method	-59	PercentageCalculationRangeCmd

Table 73. GroupBSalesRange (GroupB sales tax calculation range)

Property	Value	Description of value
Calculation scale ID	GroupBSalesScale	Calculation scale that was defined earlier
Calculation range starting value	0	Any total net price for an order greater than or equal to zero is subject to this range
Cumulative calculation range indicator	0	This calculation range is non–cumulative
CalculationRange calculation method	-59	PercentageCalculationRangeCmd

Table 74. GroupBShippingRange (GroupA shipping tax calculation range)

Property	Value	Description of value
Calculation scale ID	GroupBShipScale	Calculation scale that was defined earlier
Calculation range starting value	0	Any total net price for an order greater than or equal to zero is subject to this range
Cumulative calculation range indicator	0	This calculation range is non–cumulative
CalculationRange calculation method	-59	PercentageCalculationRangeCmd

Note: If all discounts that will be applied to the OrderItems do not have tax exemptions, you can improve the performance of WebSphere Commerce by using NetPriceCalculationScaleLookupCmd (-51) instead of TaxableNetPriceCalculationScaleLookupCmd (-53) for GroupASalesRange and GroupBSalesRange.

For more information on calculation ranges, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Defining look–up results for tax calculation ranges

Each calculation range needs a set of look–up results. For this example, the look–up results are the percentage tax rate. Each calculation range has only one tax rate, so only one look–up result is needed for each range. The look–up result is a percentage value, so neither currencies nor quantity units need to be associated with the look–up results.

Here are the key properties you must define for each calculation range look–up result:

Table 75. GroupASalesLookupResult (GroupA sales tax calculation range look–up result)

Property	Value	Description of value
Look–up result	15.0	15% sales tax rate for Group A
Calculation range ID	GroupASalesRange	The identifier of the calculation range to which this look–up result belongs

Table 76. GroupAShippingLookupResult (GroupA shipping tax calculation range look–up result)

Property	Value	Description of value
Look–up result	15.0	15% shipping tax rate for Group A
Calculation range ID	GroupAShippingRange	The identifier of the calculation range to which this look–up result belongs

Table 77. GroupBSalesLookupResult (Group B sales calculation range look–up result)

Property	Value	Description of value
Look–up result	7.0	7% sales tax rate for Group B
Calculation range ID	GroupBSalesRange	The identifier of the calculation range to which this look–up result belongs

Table 78. GroupBShippingLookupResult (GroupA shipping tax calculation range look–up result)

Property	Value	Description of value
Look–up result	4.0	4% shipping tax rate for Group B
Calculation range ID	GroupBShippingRange	The identifier of the calculation range to which this look–up result belongs

For more information on look–up results, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Associating tax calculation scales with tax calculation rules

Each of the calculation scales that have been defined must be associated with the defined calculation rules in the CRULESCALE database table. The relationship between the defined calculation scales and the defined calculation rules is shown in the following table:

Table 79. Tax calculation scales and the associated calculation rules

Calculation scale	Calculation rule
GroupASalesScale	GroupASalesRule
GroupAShipScale	GroupAShipRule
GroupBSalesScale	GroupBSalesRule
GroupBShipScale	GroupBShipRule

For more information on the relationship between calculation ranges and calculation scales, see Chapter 6, "Calculation scales and calculation ranges" on page 47.

Attaching tax calculation codes

To use the newly defined sales tax and shipping tax calculation codes, the calculation codes must be attached to the OrderItems. The calculation codes are indirectly attached to the OrderItems because the calculation codes apply to all OrderItems resulting from a customer purchasing a product from the store catalog, not one OrderItem.

Indirectly attach the calculation codes by creating two entries in the CATENCALCD database table. To associate the calculation code with all items in the store catalog, use a catalog entry ID of *null*. The identifier for the store will be called *Store.*

Populate the CATENCALCD database table with the following information:

Table 80. Calculation code attachment information

Store ID	Catalog entry ID	Calculation code
Store	*null*	SalesTaxCalcCode
Store	*null*	ShipTaxCalcCode

For more information on calculation code attachment, see Chapter 4, "Calculation codes" on page 23.

Associating tax calculation rules with fulfillment centers

For the TaxCalculationRuleQualify calculation method to choose a calculation rule that matches one of the tax jurisdictions when shipping from a fulfillment center to a shipping address, information associating a fulfillment center with a tax calculation rule must be entered in the TAXJCRULE database table.

We have one fulfillment center, two tax jurisdictions; and four calculation rules. The relationships between these elements are defined in the following table:

Table 81. Information relating calculation rules to fulfillment centers

Fulfillment center	Tax jurisdiction	Precedence	Tax calculation rule
FulfillmentA	GroupA	1	GroupASalesRule
			GroupAShipRule
	GroupB	1	GroupBSalesRule
			GroupBShipRule

Allow sales tax and shipping tax calculation usages to not calculate a value for an OrderItem

By default, WebSphere Commerce throws an exception when a calculation usage does not return a value for an OrderItem. This can be prevented by setting the calculation usage usage indicator (STENCALUSG.USAGEFLAG) to "1". When the calculation usage usage indicator is set to 1 and the calculation usage does not return a value, WebSphere Commerce assumes the calculated value is 0.

We take advantage of this behavior in this example to reduce the number of tax categories, calculation rules, and calculation scales that must be defined. If we did not set this column to "1", we would need to define a World jurisdiction group, tax categories, calculation rules, and calculation scale to calculate the 0% tax rate for shipment when the shipping address belongs to a jurisdiction in the World jurisdiction group.

The STENCALUSG database table is also used to define the default calculation codes for a store. Default calculation code information must appear in the same database row where the USAGEFLAG column will be set. For information on defining default calculation codes, see "Defining default tax calculation codes for a store (optional)".

The following table contains the information required to allow sales tax and shipping tax calculation usages to not calculate a value for an OrderItem:

Table 82. Information to allow sales tax and shipping tax calculation usages to not calculate a value for an OrderItem

Store ID	Calculation Usage	Calculation Code	Usage indicator
Store	-3 (sales tax)	SalesTaxCalcCode	1
Store	-4 (shipping tax)	ShipTaxCalcCode	1

Defining default tax calculation codes for a store (optional)

If your store does not already have default calculation codes for taxes defined, you may wish to make the calculation codes you defined earlier your default calculation codes.

Default calculation codes for a store are defined in the STENCALUSG database table. The following table contains the information required to define the default

sales and shipping tax calculation codes for the store:

Table 83. Information to define default calculation codes for the store

Store ID	Calculation Usage	Calculation Code
Store	-3 (sales tax)	SalesTaxCalcCode
Store	-4 (shipping tax)	ShipTaxCalcCode

Appendix A. Additional UML object model diagrams

This section contains additional UML object model diagrams that show different parts of the calculation framework.

For more information on the conventions used in the UML object model diagrams in this section, see Appendix B, "UML legend" on page 107.

Calculation methods

The following UML object model diagram shows the relationships between objects involved with calculation methods:
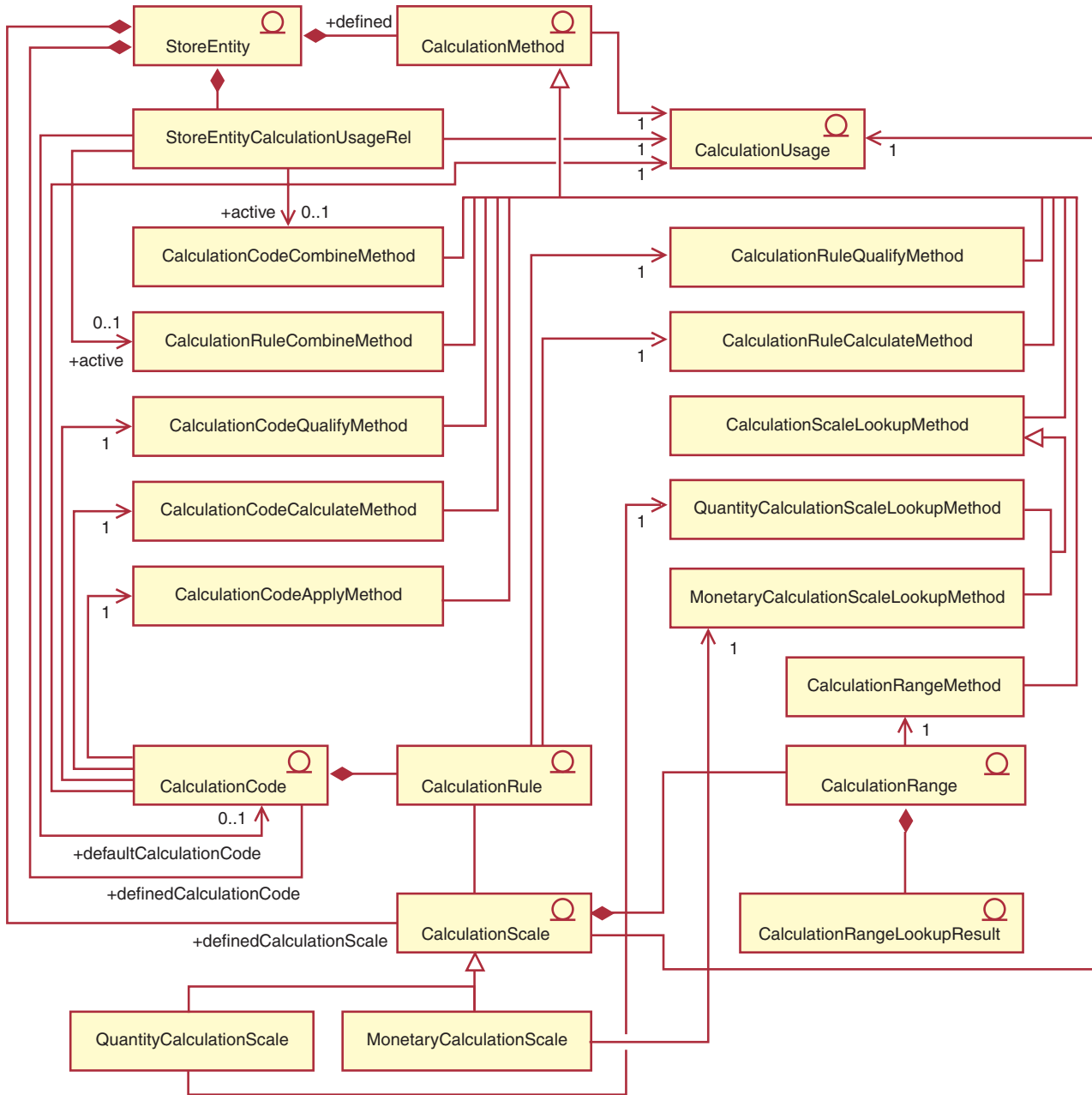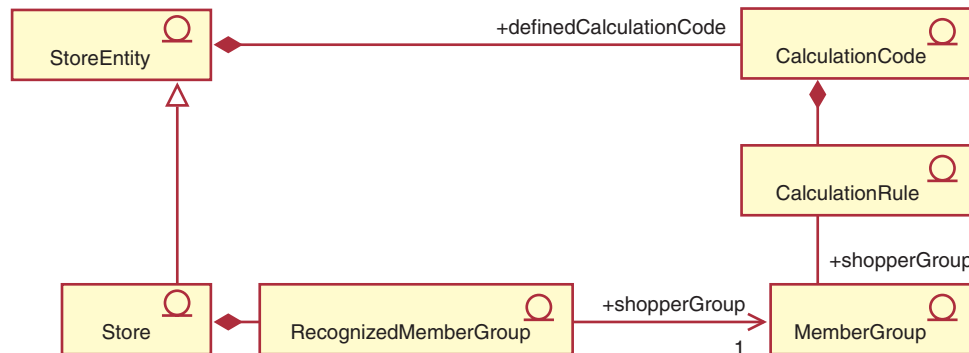

Figure 14. Calculation method UML object model diagram

Calculation usages

The following UML object model diagram shows the relationships between objects involved with calculation usages:
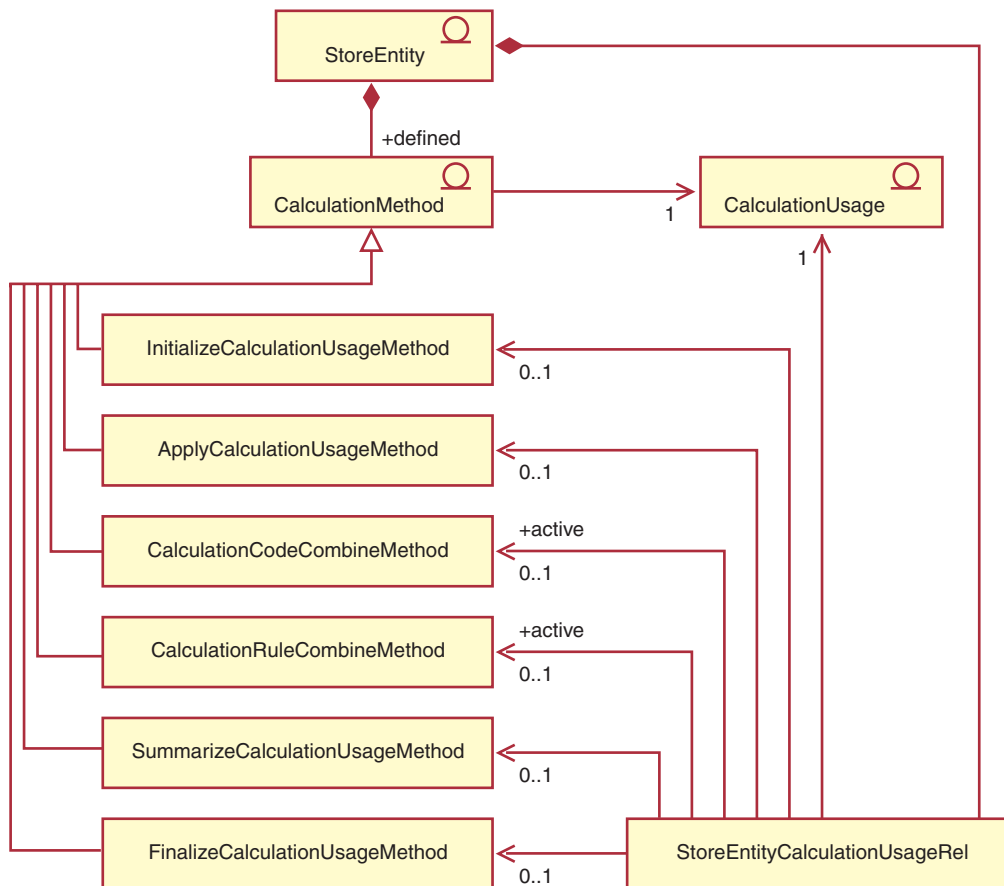
Figure 15. Calculation usage UML object model diagram

Calculation code direct attachment

The following UML object model diagram shows the relationships between objects involved when calculation codes are directly attached to OrderItems:
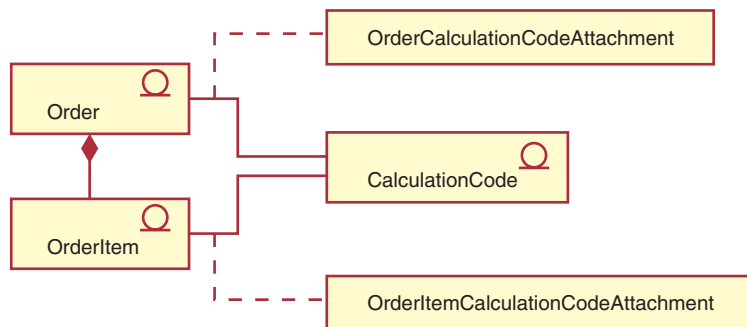
Figure 16. Calculation code code direct attachment UML object model diagram

Calculation code indirect attachment

The following UML object model diagram shows the relationships between objects involved when calculation codes are indirectly attached to OrderItems:
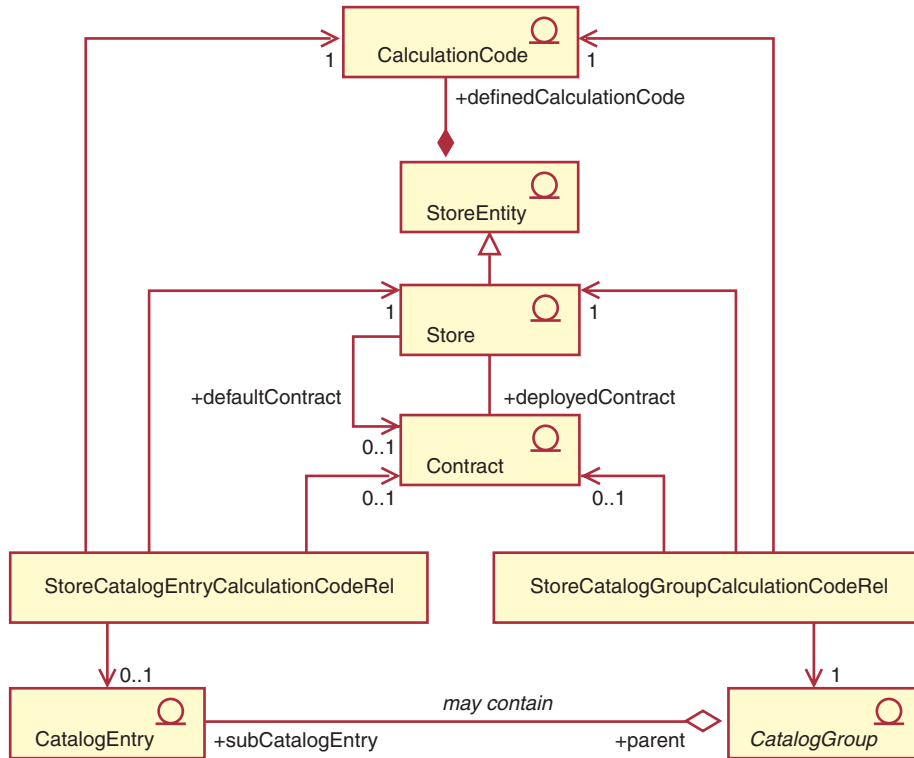
Figure 17. Calculation code indirect attachment UML object model diagram

Discounts implementation

The following UML object model diagram shows the relationships between objects involved when the calculation framework is used to calculate discounts:
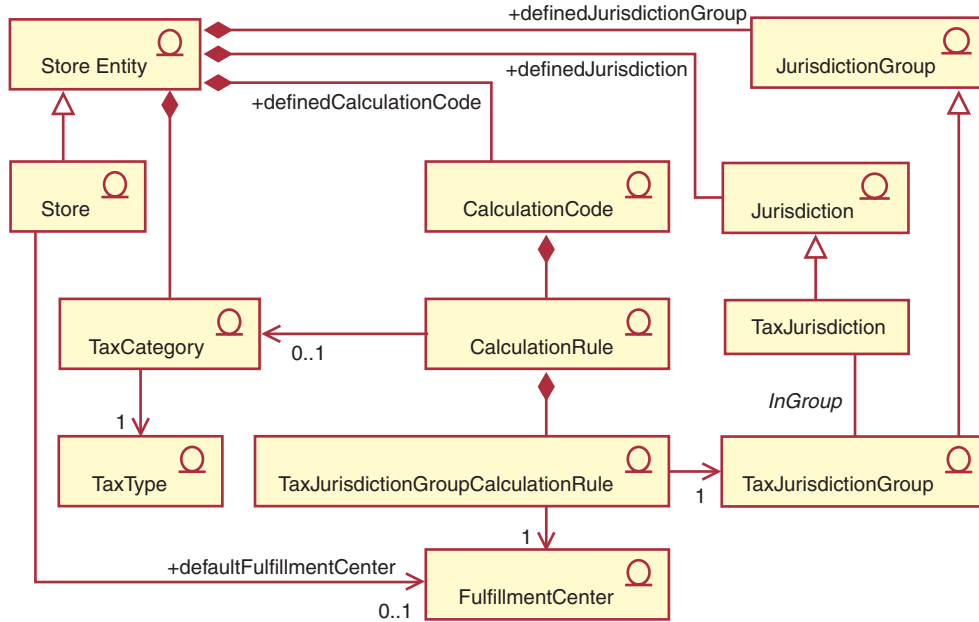

Figure 18. Calculation framework discounts implementation UML object model diagram

Tax implementation

The following UML object model diagram shows the relationships between objects involved when the calculation framework is used to calculate sales and shipping taxes:

Figure 19. Calculation framework sales and shipping tax implementation UML object model diagram

Shipping implementation

The following UML object model diagram shows the relationships between objects involved when the calculation framework is used to calculated shipping charges:
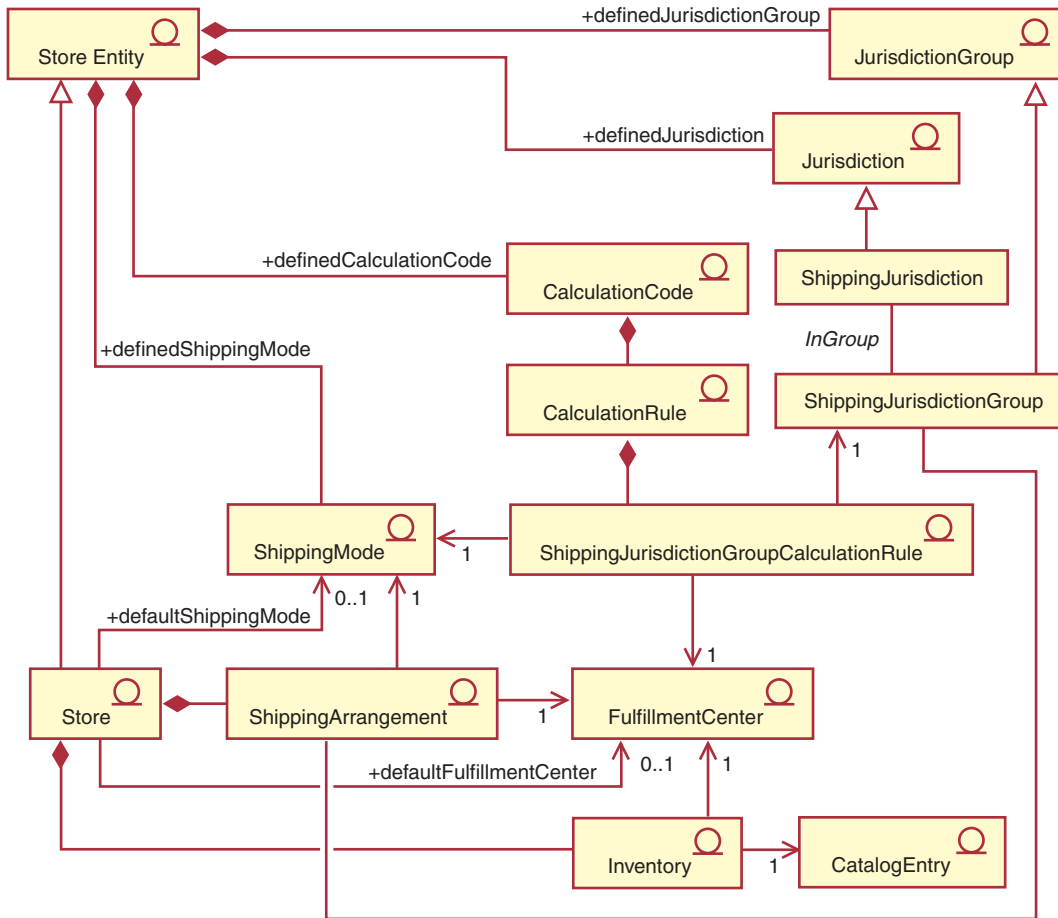
Figure 20. Calculation framework shipping charge implementation UML object model diagram

Appendix B. UML legend

Unified Modeling Language (UML) is a standard graphical language for presenting different elements of software design. The following examples are some of the most common elements of UML. For further detail about formal specifications, refer to `http://www.rational.com` and `http://www.omg.org`.

UML diagrams consist of the following items:
* Boxes: Boxes represent classes of objects. The class names appear at the top of the box. Attributes, if shown, appear below the class name. The class name and attributes are separated by a line.
* Lines: Lines represent possible relationships between objects of two classes. Objects of the class on one of end of the line can be "associated" with objects of the other class.
* Solid diamonds: Solid diamonds on the end of a line indicate containment by value. Objects of the class on the other end of the line are part of one and only one object of the class the diamond touches.
* Open diamonds: Open diamonds on the end of a line indicate containment by reference. Objects at the diamond end of the line can be thought of as grouping objects of the class at the other end of the line.
* Cardinality numbers: These appear at the end of relationship lines to indicate a cardinality restriction. The following table summarizes cardinality restrictions:

| Cardinality number	Relationship type
1	One and only one
0..1	Zero or one
0..n	Zero or more
1..n	One or more

If no cardinality restriction is shown, the cardinality is assumed to be 0..n, unless a solid diamond appears on the end of a relationship line. In that case, the cardinality must be 1.
* Plus signs: Plus signs appearing at the end of relationship lines indicate the object of the class at the end of the line plays a role in the relationship. Text following the plus sign indicates the object's role in the relationship.
* Arrows: Arrows at the end of a relationship line indicate the direction of the relationship between two objects is in the direction of the arrow. The absence of any arrows on a relationship line indicates the direction of the relationship between the objects is normally in both directions.

The following diagrams illustrate the above concepts:
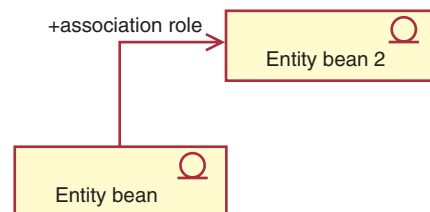
This diagram shows two entity beans with the symbol indicating an Enterprise JavaBeans bean. There is a unidirectional association from the first entity bean to the second entity bean. The plus sign is followed by text that describes the role of the second entity bean.
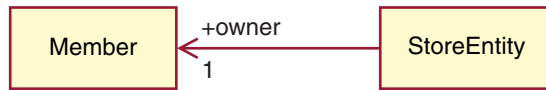
In this diagram, a StoreEntity has one and only one owner, which is a Member. A Member may own zero or more StoreEntities. The plus sign indicates that the Member plays a role in the relationship. In this case the Member is the owner of the StoreEntity. The arrow indicates that you would normally determine the owner of a StoreEntity by querying the StoreEntity for its owner, and not querying a Member for all the StoreEntities it owns.

In this diagram, an OrderItem is always part of one and only one Order. An Order has zero or more OrderItems.
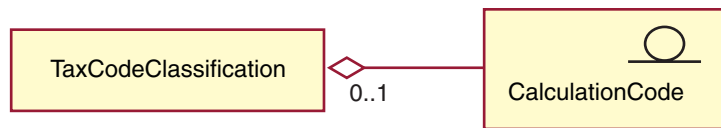
This diagram indicates that a CalculationCode is grouped by zero or one TaxCodeClassifications and a TaxCodeClassification groups zero or more CalculationCodes.

Appendix C. Data model legend

Database tables within the WebSphere Commerce database schema are organized into functional groups. The referential integrity relationship between database tables is illustrated using the following symbols:

 A rounded, dotted rectangle represents a weak dependent entity. Instances of a weak dependent entity cannot be uniquely identified without determining its relationship to other entities. The primary key to an instance of a weak dependent entity consists entirely of foreign keys.

 A rounded, solid rectangle represents a strong dependent entity. Instances of a strong dependent entity cannot be uniquely identified without determining its relationship to other entities. The primary key to an instance of a strong dependent entity contains foreign keys, but does not consist entirely of foreign keys.

 A rectangle represents an independent entity. Instances of an independent entity can be uniquely identified without determining its relationship to other entities. The primary key to an instance of an independent entity does not contain any foreign keys.
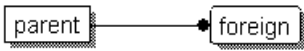
 A solid relationship line represents an identifying relationship. An identifying relationship occurs when instances of the child entity are identified through association with a parent entity. The primary key attributes of the parent entity become primary key attributes of the child.
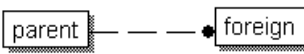
 A dotted relationship line with a solid relationship symbol represents a mandatory non-identifying relationship. A mandatory non-identifying relationship occurs when key attributes from the parent entity must become non-key attributes of the child entity. That is, the inherited non-key attributes of the child entity in this relationship cannot be NULL.
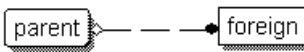
 A dotted relationship line with a solid and open relationship symbol represents an optional non-identifying relationship. An optional non-identifying relationship occurs when foreign keys from the parent entity can become non-key attributes of the child entity. That is, the inherited non-key attributes of the child entity in this relationship can be NULL.

 A relationship line that links back to the original table indicates a recursive relationship. Recursive relationships must be optional non-identifying relationships.

Appendix D. Other sources of information

Online information

More information about WebSphere Commerce can be found in the WebSphere Commerce online help. The online help is accessible by visiting the following URL:

`http://`*`host_name`*`/wchelp`

where *host_name* is the fully qualified TCP/IP name of the machine where you have installed WebSphere Commerce.

API documentation

API documentation on the Java classes and interfaces used in the calculation framework can be found in the WebSphere Commerce online help. The online help is accessible by visiting the following URL:

`http://`*`host_name`*`/wchelp`

where *host_name* is the fully qualified TCP/IP name of the machine where you have installed WebSphere Commerce.

Database schema

More information about the database tables and columns used in the calculation framework can be found in the WebSphere Commerce online help. The online help is accessible by visiting the following URL:

`http://`*`host_name`*`/wchelp`

where *host_name* is the fully qualified TCP/IP name of the machine where you have installed WebSphere Commerce.

Bootstrap files

The calculation framework takes initial data from both the language–independent bootstrap file and the language–dependent bootstrap file. The language–independent bootstrap file is `wcs.bootstrap.xml`. The language–dependent bootstrap is `wcs.bootstrap_`*`xx_XX`*`.xml`, where xx_XX is the language indicator. For example, the language dependent bootstrap file for Japanese is `wcs.bootstrap_ja_JP.xml`.

Both the language–dependent bootstrap file and the language–independent bootstrap file are in the following location:

- ▶ AIX `/usr/lpp/CommerceServer/schema/xml`
- ▶ 400 `/qibm/proddata/WebCommerce/schema/xml`
- ▶ Linux `/opt/WebSphere/CommerceServer/schema/xml`
- ▶ Solaris `/opt/WebSphere/CommerceServer/schema/xml`
- ▶ 2000 *`drive`*`:\Program Files\WebSphere\CommerceServer\schema\xml`
- ▶ NT *`drive`*`:\WebSphere\CommerceServer\schema\xml`

More information about the bootstrap files and their contents can be found in the online information.

Loading calculation framework data

Calculation framework data can be loaded into WebSphere Commerce in two ways:
- Package the calculation framework data as part of a store archive and publish the store archive using Store Services.
- Load the calculation framework data directly into WebSphere Commerce using the WebSphere Commerce Loader package.

Information on packaging and publishing store archives and loading data directly using the WebSphere Commerce Loader package can be found in the IBM WebSphere Commerce Store Developer's Guide.

The WebSphere Commerce Store Developer's Guide is available at the following Web sites:

WebSphere Commerce Business Edition

http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html

WebSphere Commerce Professional Edition

http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html

The IBM WebSphere Commerce Store Developer's Guide also contains examples of calculation framework data used for shipping and tax calculations.

Modifying, extending, or overriding existing calculation logic

Before you modify, extend, or override any of the Java methods used in the calculation framework, read the WebSphere Commerce Programmer's Guide for information on modifying and extending existing business logic.

The WebSphere Commerce Programmer's Guide is available at the following web sites:

WebSphere Commerce Business Edition

http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html

WebSphere Commerce Professional Edition

http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html

Appendix E. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504–1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Ltd.
Office of the Lab Director
8200 Warden Avenue
Markham, ON L6G 1C7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Credit card images, trademarks and trade names provided in this product should be used only by merchants authorized by the credit card mark's owner to accept payment via that credit card.

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

400	AIX	AS/400
@server	iSeries	IBM
WebSphere		

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows and Windows NT are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product or service names may be the trademarks or service marks of others.

IBM.

Printed in U.S.A.