Sitecore E-Commerce Services 2.0 on CMS 6.6

The Order Manager API Reference Guide

A developer's guide to the Order Manager API.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Table of Contents

Chapter 1 Introduction 4
Chapter 2 Order Manager Modules 5
2.1 Sitecore.Ecommerce.Core.Common 6
2.1.1 Address 6
2.1.2 Amount 7
2.1.3 Communication 8
2.1.4 Contact 8
2.1.5 IConfigurable 9
2.1.6 IEntity 9
2.1.7 Location 9
2.1.8 Measure 10
2.1.9 Party 10
2.1.10 PartyLegalEntity 11
2.1.11 PartyTaxScheme 12
2.1.12 Period 12
2.1.13 Person 13
2.1.14 ShopContext 13
2.2 Sitecore.Ecommerce.Core.OrderManagement 14
2.2.1 CoreOrderManager 14
2.2.2 StateManager 14
2.2.3 StateProvider 14
2.2.4 SubstateCombinationSet 15
2.2.5 ChangeOrderLineQuantityCommand 15
2.2.6 Command 15
2.2.7 ConfirmationMessageBuilder 16
2.2.8 CoreOrderMangerImpl 16
2.2.9 ILoggingEnabled 16
2.2.10 OrderConfirmation 17
2.2.11 OrderLineFactory 17
2.2.12 OrderProcessingStrategy 18
2.2.13 OrderProcessor 18
2.2.14 PlainMessageConfirmationBuilder 19
2.2.15 ProcessingStrategy 19
2.2.16 CoreOrderStateConfiguration 19
2.2.17 MerchantOrderProcessor 20
2.3 Sitecore.Ecommerce.Core.OrderManagement.Orders 21
2.3.1 AllowanceCharge 21
2.3.2 CustomerParty 22
2.3.3 Delivery 22
2.3.4 Despatch 23
2.3.5 Item 24
2.3.6 LineItem 24
2.3.7 MonetaryTotal 26
2.3.8 Order 26
2.3.9 OrderedShipment 28
2.3.10 OrderLine 29
2.3.11 PaymentMeans 29
2.3.12 Price 30
2.3.13 ReservationTicket 30
2.3.14 Shipment 31
2.3.15 State 33
2.3.16 Substate 34
2.3.17 SupplierParty 34
2.3.18 TaxCategory 35

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.19 TaxScheme 35
2.3.20 TaxSubTotal 36
2.3.21 TaxTotal 36
2.4 Sitecore.Ecommerce.Core.OrderManagement.Extensions 37
2.4.1 CoreExtensions 37
2.5 Sitecore.Ecommerce.Merchant.Pipelines.HttpRequest 38
2.5.1 MerchantShopResolver 38
2.6 Sitecore.Ecommerce.Merchant.OrderManagement 39
2.6.1 AddNewOrderLineProcessingStrategy 39
2.6.2 CancelReservationProcessingStrategy 40
2.6.3 CaptureOrderProcessingStrategy 40
2.6.4 ChangeAllowanceChargeAmountProcessingStrategy 41
2.6.5 ChangeOrderLineQuantityProcessingStrategy 42
2.6.6 EditOrderLineProcessingStrategy 42
2.6.7 MerchantOrderCancelationStrategy 43
2.6.8 MerchantOrderProcessor 43
2.6.9 MerchantOrderManager 44
2.6.10 MerchantOrderStateConfiguration 44
2.6.11 OrderFieldProcessingStrategy 45
2.6.12 OrderLineFactoryImpl 46
2.6.13 OrderLineStateTransitionStrategy 46
2.6.14 OrderLineProcessingStrategy 47
2.6.15 OrderProcessingStrategyResolver 47
2.6.16 OrderProcessingStrategyResolverImpl 48
2.6.17 OrderStateProcessingStrategy 48
2.6.18 PackOrderProcessingStrategy 49
2.6.19 RemoveOrderLineProcessingStrategy 49
2.6.20 SendConfirmationProcessingStrategy 50
2.6.21 ShipOrderProcessingStrategy 50
2.6.22 StateValidator 51
2.7 Sitecore.Ecommerce.Merchant.OrderManagement.Extensions 52
2.7.1 MerchantExtensions 52
2.8 Sitecore.Ecommerce.Visitor.OrderManagement 53
2.8.1 VisitorOrderCancellationStrategy 53
2.8.2 VisitorOrderProcessor 53
2.8.3 VisitorOrderRepository 53
2.8.4 TransientOrderConverter 54
2.8.5 TransientOrderManager 55
2.9 Sitecore.Ecommerce.Visitor.Pipelines.HttpRequest 56
2.9.1 VisitorShopResolvingProcessor 56
2.10 Sitecore.Ecommerce.Visitor.Pipelines.OrderedCreated 57
2.10.1 NotifyCustomer 57
2.11 Sitecore.Ecommerce.Visitor.Sites 58
2.11.1 VisitorShopResolver 58
2.12 Sitecore.Ecommerce.Visitor.Unity 59
2.12.1 ShopContextContainerExtension 59

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Chapter 1

Introduction

This guide describes the Sitecore Order Manager (OM) API and some useful extensions to its functionality.

It is useful for developers who are looking for information about the OM API. It describes the class functionality, parent classes, implementation, methods, and properties.

This manual contains the following chapters:

- **Chapter 1 — Introduction**
  This chapter is an introduction to the guide.
- **Chapter 2 — Order Manager Modules**
  This chapter contains the API reference.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Chapter 2

Order Manager Modules

This chapter describes the most frequently used classes in the Sitecore Order Manager API. Each section in this chapter presents an important module in the API. Each section contains a description of the classes in this module and tables that describe the properties and methods of the class.

This chapter contains the following sections:

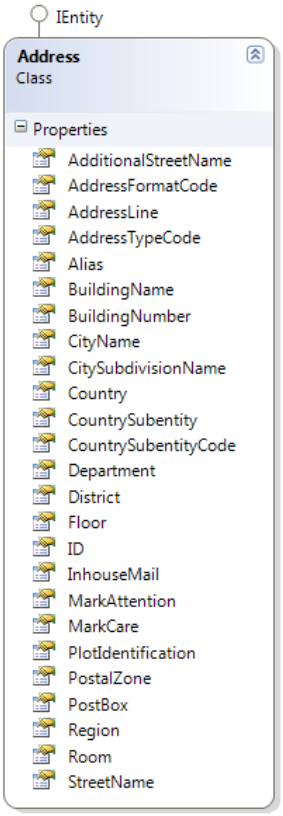
- Sitecore.Ecommerce.Core.Common
- Sitecore.Ecommerce.Core.OrderManagement
- Sitecore.Ecommerce.Core.OrderManagement.Orders
- Sitecore.Ecommerce.Core.OrderManagement.Extensions
- Sitecore.Ecommerce.Merchant.Pipelines.HttpRequest
- Sitecore.Ecommerce.Merchant.OrderManagement
- Sitecore.Ecommerce.Merchant.OrderManagement.Extensions
- Sitecore.Ecommerce.Visitor.OrderManagement
- Sitecore.Ecommerce.Visitor.Pipelines.HttpRequest
- Sitecore.Ecommerce.Visitor.Pipelines.OrderedCreated
- Sitecore.Ecommerce.Visitor.Sites
- Sitecore.Ecommerce.Visitor.Unity

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.1 Sitecore.Ecommerce.Core.Common

2.1.1 Address

This class implements the `IEntity` interface and contains information about the address.



The properties of the `Address` class are:

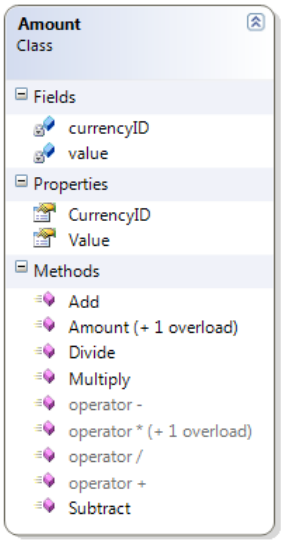
Property	Description
`string AdditionalStreetName`	Gets or sets the additional text to further specify the street name.
`string AddressFormatCode`	Gets or sets the code of the address format.
`string AddressLine`	Gets or sets the address line.
`string AddressTypeCode`	Gets or sets the address type code to specify whether it is a home or a business address.
`long Alias`	Gets the address ID that is used in the database.
`string BuildingName`	Gets or sets the building name.
`string CityName`	Gets or sets the name of the city, town or village.
`string CitySubdivisionName`	Get or sets the city subdivision name. The subdivision can be the district or the borough.
`string Country`	Gets or sets the country.
`string CountrySubentity`	Gets or sets the country sub-entity. The sub-entity can be a county or a state.
`string CountrySubentityCode`	Gets or sets the country sub-entity code.
`string Department`	Gets or sets the department in the organization.
`string District`	Gets or sets the district.
`string Floor`	Gets or sets the floor of the building that the address is on.
`string ID`	Gets or sets the identifier of a specific address within a scheme of registered addresses.
`string InhouseMail`	Gets or sets the location of the in-house mail as part of the address.
`string MarkAttention`	Gets or sets the name of the person or the department in the organization to whom the incoming mail is marked, for example, *for the attention of*, *FAO* or *ATTN* for this address.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`string MarkCare`	Gets or sets the name of the person or the organization at this address to whom the incoming mail is marked, for example, *care of* or *C/O*.
`string PlotIdentification`	Text for the identifier of the plot where the address is located.
`string PostalZone`	Gets or sets the postal code.
`string PostBox`	Gets or sets the post box.
`string Region`	Gets or sets the region. The region can be a group of countries.
`string Room`	Gets or sets the room. The room can be a room, suite, or apartment in the building.
`string StreetName`	Gets or sets the street name.

2.1.2 Amount

This class contains information about the amount.



The properties of the `Amount` class:

Property	Description
`string CurrencyID`	Gets or sets the ID of the currency.
`decimal Value`	Gets or sets the value of the amount.

The methods of the `Amount` class are:

Method	Description

`Amount Add(`
`  Amount amount1,`
`  Amount amount2)`
	Adds two amounts of the same `currencyID`.

`Amount()`	Initializes a new instance of the `Amount` class.

`Amount(`
`  decimal value,`
`  string currencyId)`
	Takes the value and the currency of the amount and initializes a new instance of the `Amount` class.

`Amount Divide(`
`  Amount amount,`
`  decimal divisor)`
	Divides the specified amount.

`Amount operator -(`
`  Amount amount1,`
`  Amount amount2)`
	Implements the – operator by calling the `Subtract` method.

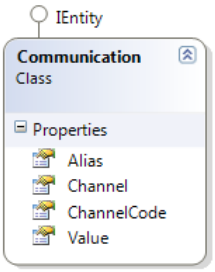
`Amount operator *(`
`  Amount amount,`
`  decimal multiplier)`
	Implements the * operator by calling the `Multiply` method.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Method	Description
`Amount operator *( decimal multiplier, Amount amount)`	Implements the * operator by calling the `Multiply` method.
`Amount operator /( Amount amount, decimal divisor)`	Implements the / operator by calling the `Divide` method.
`Amount operator +( Amount amount1, Amount amount2)`	Implements the + operator by calling the `Add` method.
`Amount Subtract( Amount amount1, Amount amount2)`	Subtracts the value of `amount2` from the value of `amount1`.

2.1.3 Communication

This class implements the `IEntity` interface and contains information about the means of communication.

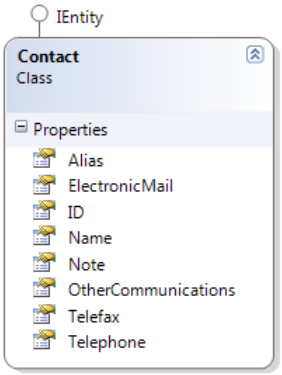


The properties of the `Communication` class:

Property	Description
`long Alias`	Gets the communication's ID that is used in the database.
`string Channel`	Gets or sets the communication type as text.
`string ChannelCode`	Gets or sets the code for the communication type.
`string Value`	Gets or sets the value of the communication such as the phone number or the email address.

2.1.4 Contact

This class implements the `IEntity` interface and contains information about the contact person or department in the organization.



The properties of the `Contract` class are:

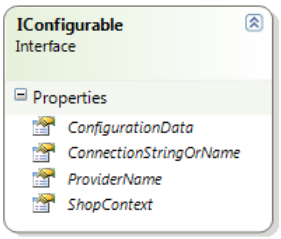
Property	Description
`long Alias`	Gets the contact's ID that is used in the database.
`string ElectronicMail`	Gets or sets the email of the contact.
`string ID`	Gets or sets the identifier of the contact. It is a personal reference number such as "7687687".
`string Name`	Gets or sets the name of the contact.
`string Note`	Gets or sets the free text note for the contact. This note can describe the circumstances in which the contact can be used such as emergency.
`ICollection<Communication> OtherCommunications`	Gets or sets the information about other communication means.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`string Telefax`	Gets or sets the fax number of the contact.
`string Telephone`	Gets or sets the telephone number of the contact.

2.1.5 IConfigurable

The implementations of this interface contain information about the configuration settings that are needed to refer to the order's source.

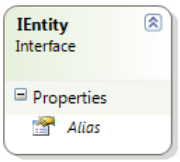


The properties of the `IConfigurable` interface are:

Property	Description
`Dictionary<string, object> ConfigurationData`	Gets or sets the configuration data.
`string ConnectionStringOrName`	Gets or sets the connection string or name.
`string ProviderName`	Gets or sets the name of the data provider. This name is used in the database connections to identify the database system that is used – for example, Microsoft SQL Server, Oracle, and PostgreSQL Direct.
`ShopContext ShopContext`	Gets or sets of the shop context.

2.1.6 IEntity

This interface acts as a contract for the classes that have the `Alias` property.

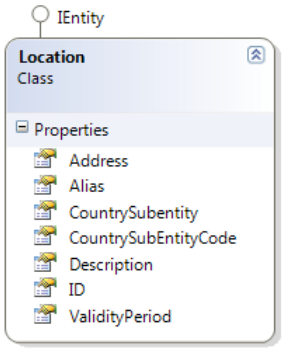


The only property of the `IEntity` interface is:

Property	Description
`long Alias`	Gets the IEntity's ID that is used in the database.

2.1.7 Location

This class implements the `IEntity` interface and contains information about the location.



The properties of the `Location` class are:

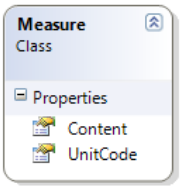
Property	Description
`Address Address`	Gets or sets information about the address of the location.
`long Alias`	Gets the location's ID that is used in the database.
`string CountrySubentity`	Gets or sets text about the country sub-entity. A sub-entity can be a county or a state.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`string CountrySubEntityCode`	Gets or sets the code of the sub-entity.
`string Description`	Gets or sets text of the description of the location.
`string ID`	Gets or sets the identifier of the location such as the European Article Number (EAN) and the Global Location Number (GLN).
`Period ValidityPeriod`	Gets or sets information about the validity period of the location. This is the period in which the location can be used, for example for delivery.

2.1.8 Measure

This class contains information about the measure.

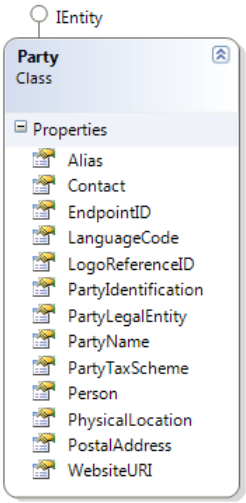


The properties of the `Measure` class are:

Property	Description
`decimal Content`	Gets or sets the decimal value such as `10000.25` **Note** You must not use more than four decimals for the value of the `Content` property.
`string UnitCode`	Gets or sets the unit code such as `"EA"`.

2.1.9 Party

This class implements the `IEntity` interface and contains information about the organization, the sub-organization, or the individual that has a role in the business process.



The properties of the `Party` class are:

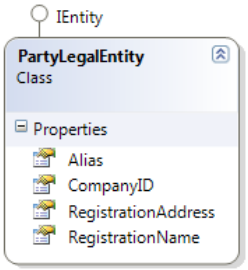
Property	Description
`long Alias`	Gets the party's ID that is used in the database.
`Contact Contact`	Gets or sets the contact of the party.
`string EndpointID`	Gets or sets the end point for the routing service such as the EAN and the GLN.
`string LanguageCode`	Gets or sets the code of the party language.
`string LogoReferenceID`	Gets or sets the identifier of the logo reference.
`string PartyIdentification`	Gets or sets an identifier of the party.
`PartyLegalEntity PartyLegalEntity`	Gets or sets the legal entity of the party.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`string PartyName`	Gets or sets the name of the party. **Note** You must use this name when no other party identification is used.
`ICollection<PartyTaxScheme> PartyTaxScheme`	Gets or sets the information about the tax scheme of the party.
`Person Person`	Gets or sets the persons of the party.
`Location PhysicalLocation`	Gets or sets the location of the party. **Note** You must use `PhysicalLocation` if it is different from `PostalAddress`.
`Address PostalAddress`	Gets or sets the postal address of the party.
`string WebsiteURI`	Gets or sets the URL of the party website.

2.1.10 PartyLegalEntity

This class implements the `IEntity` interface and contains information about the legal registration that is applicable for the party.



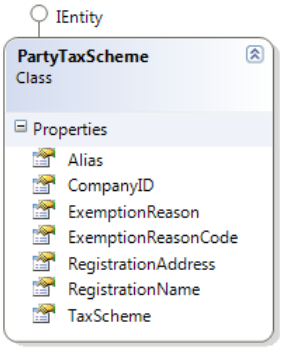
The properties of the `PartyLegalEntity` class are:

Property	Description
`long Alias`	Gets the ID of the party legal entity that is used in the database.
`string CompanyID`	Gets or sets the identifier of the legal entity. It identifies the company as registered with the company registration scheme.
`Address RegistrationAddress`	Gets or sets the registration address of the legal entity. It is the registered address of the party in the company registration scheme.
`string RegistrationName`	Gets or sets the registration name of the legal entity. It is the name of the party that is registered by the legal authority.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.1.11 PartyTaxScheme

This class implements the `IEntity` interface and contains information about the tax scheme of the party.

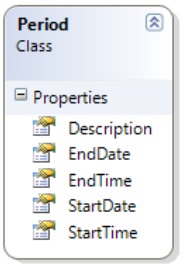


The properties of the `PartyTaxScheme` class are:

Property	Description
`long Alias`	Gets the ID of the party tax scheme that is used in the database.
`string CompanyID`	Gets or sets the identifier of the company. This identifier is assigned for tax purposes to the party by the taxation authority.
`string ExemptionReason`	Gets or sets the reason for the exemption from the tax.
`string ExemptionReasonCode`	Gets or sets the code of tax exemption.
`Address RegistrationAddress`	Gets or sets the registered address of the party. It is used for tax purposes.
`string RegistrationName`	Gets or sets the name of the party. It is the official name of the party as registered by the relevant taxation authority.
`TaxScheme TaxScheme`	Gets or sets the information about the tax scheme.

2.1.12 Period

This class contains information about the period of time.



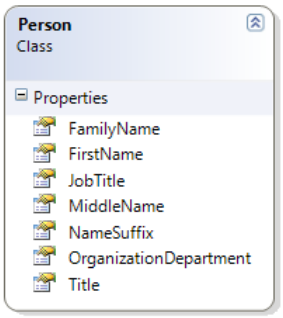
The properties of the `Period` class are:

Property	Description
`string Description`	Gets or sets the description for the period.
`DateTime EndDate`	Gets or sets the end date of the period.
`TimeSpan EndTime`	Gets or sets the end time of the period.
`DateTime StartDate`	Gets or sets the start date of the period.
`TimeSpan StartTime`	Gets or sets the start time of the period.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.1.13 Person

This class contains information about the person.

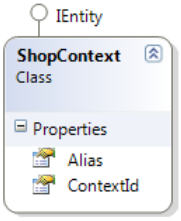


The properties of the `Person` class are:

Property	Description
`string FamilyName`	Gets or sets the surname or the family name of the person.
`string FirstName`	Gets or sets the first name of the person.
`string JobTitle`	Gets or sets the job title of the person.
`string MiddleName`	Gets or sets the middle name of the person.
`string NameSuffix`	Gets or sets the name suffix of the person., such as `"PhD"`, and `"Jnr"`.
`string OrganizationDepartment`	Gets or sets the department that this person works for.
`string Title`	Gets or sets the title of the person such as `"Mr."`, `"Ms."`, `"Dr."`, and `"Sir."`.

2.1.14 ShopContext

This class implements the `IEntity` interface and contains information about the context of the webshop.



The properties the `ShopContext` class are:

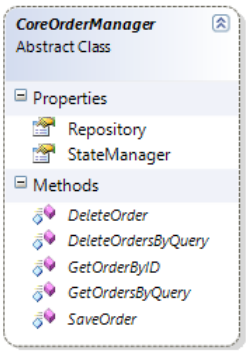
Property	Description
`long Alias`	Gets the ID of the shop context that is used in the database.
`string ContextId`	Gets or sets the ID of the shop context. It is a unique name that refers to the webshop. This is typically the hostname of the webshop or the ID of the website definition that contains the webshop.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.2 Sitecore.Ecommerce.Core.OrderManagement

2.2.1 CoreOrderManager

This class contains information about the core order manager. It is an abstract class in which all the methods are protected and internal because they are just defining an interface for the implementations.

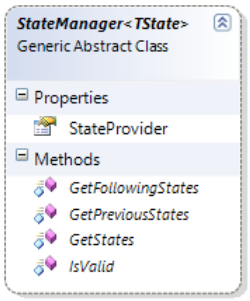


The properties of the `CoreOrderManager` class are:

Property	Description
`Repository<Order> Repository`	Gets or sets the repository.
`StateManager<State> StateManager`	Gets or sets the state manager.

2.2.2 StateManager

This is a generic abstract class and contains information about the state manager. It is an abstract class in which all the methods are protected and internal because they are defining an interface for the implementations.

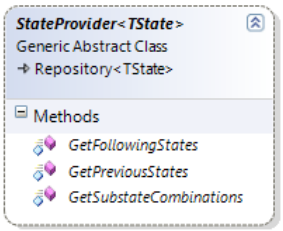


The only property of the `StateManager` class is:

Property	Description
`StateProvider<TState> StateProvider`	Gets or sets the state provider.

2.2.3 StateProvider

This is a generic abstract class that implements the `Repository` class. All the methods in this class are protected and internal because they are defining an interface for the implementations.

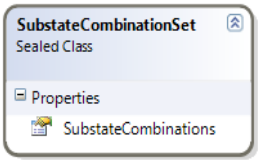


These methods are used to get and navigate between the different states and substates of the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.2.4 SubstateCombinationSet

This is a sealed class that contains information about the set of substate combinations.

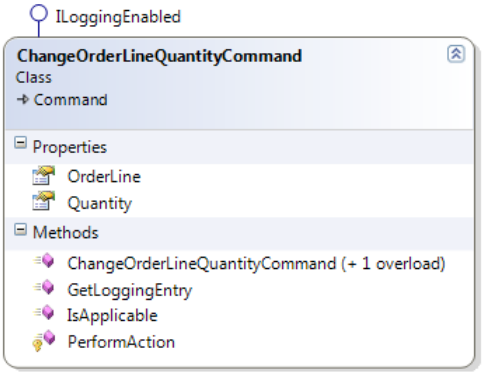


The only property of the `SubstateCombinationSet` class is:

Property	Description
`IEnumerable<IDictionary<string, bool>> SubstateCombinations`	Gets or sets the sub-state combinations.

2.2.5 ChangeOrderLineQuantityCommand

This class implements the `ILoggingEnabled` interface, the `Command` class and contains information about the quantity of the order line and allows you to change it.



The properties of the `ChangeOrderLineQuantityCommand` class are:

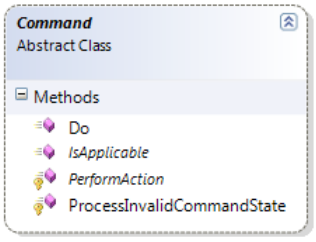
Property	Description
`OrderLine OrderLine`	Gets or sets the order line.
`decimal Quantity`	Gets or sets the quantity. It is the value that is changed when changing the quantity of products on that particular order line in the user interface.

The methods of the `ChangeOrderLineQuantityCommand` class are:

Method	Description
`ChangeOrderLineQuantityCommand( OrderLine orderLine)`	Takes the order line and initializes a new instance of the `ChangeOderLineQuanityCommand` class.
`ChangeOrderLineQuantityCommand()`	Initializes a new instance of the `ChangeOderLineQuanityCommand` class.
`LoggingEntry GetLoggingEntry()`	Gets the logging entry.
`bool IsApplicable()`	Specifies whether the command is applicable to the specified order line. Its value is `true` if the command is applicable.

2.2.6 Command

This is an abstract class that the implementations use to process a command.



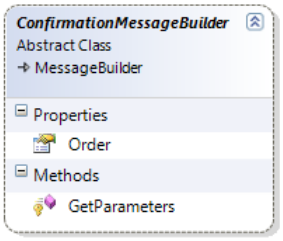
The methods of the `Command` class are:

Method	Description
`void Do()`	Processes the order line.
`bool IsApplicable()`	Specifies whether the command is applicable to the specified order line. It is `true` if the command is applicable.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.2.7 ConfirmationMessageBuilder

This is an abstract class that implements the `MessageBuilder` class and defines the confirmation message builder.

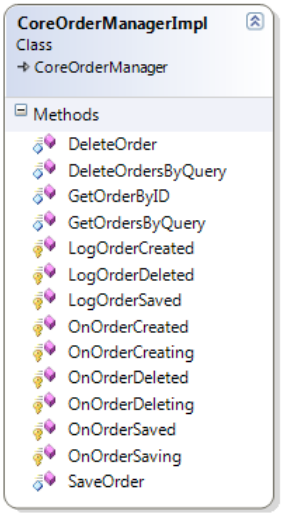


The only property of the `ConfirmationMessageBuilder` class is:

Property	Description
`virtual Order Order`	Gets or sets the order.

2.2.8 CoreOrderMangerImpl

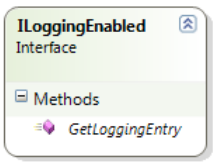
The class implements the `CoreOrderManager` class. It is used to manipulate the order.



All the methods of this class are protected to hide the core API and expose Visitor and Merchant only.

2.2.9 ILoggingEnabled

This interface specifies whether the type supports logging or not.



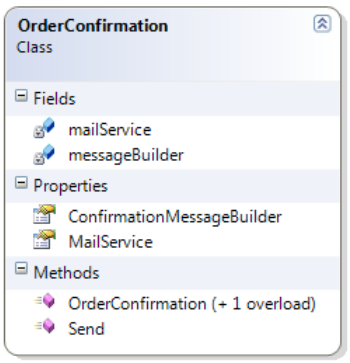
The only method in the `ILoggingEnabled` interface is:

Method	Description
`LoggingEntry GetLoggingEntry()`	Gets the logging entry.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.2.10 OrderConfirmation

This class contains information about the confirmation of an order.



The properties of the `OrderConfirmation` class are:

Property	Description
`ConfirmationMessageBuilder ConfirmationMessageBuilder`	Gets or sets the confirmation message builder.
`MailService MailService`	Gets or sets the mail service.

The methods of the `OrderConfirmation` class are:

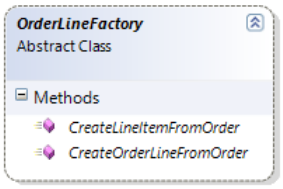
Method	Description
`OrderConfirmation()`	Initializes a new instance of the `OrderConfirmation` class.

`OrderConfirmation(`
`  MailService mailService,`
`  ConfirmationMessageBuilder`
`  messageBuilder)`
	Takes the mail service, the message builder, and initializes a new instance of the `OrderConfirmation` class.

`void Send()`	Sends the confirmation message.

2.2.11 OrderLineFactory

This is an abstract class. You can implement it to create an order line or a line item from the order.



The methods of the `OrderLineFactory` class are:

Method	Description

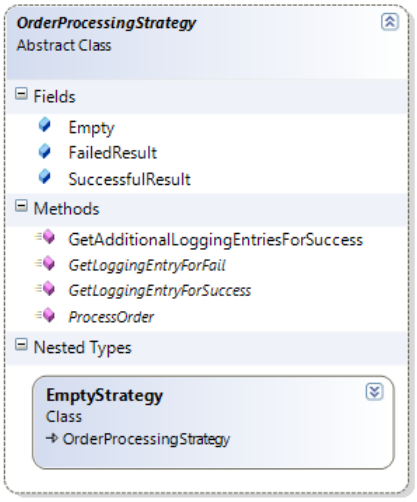
`LineItem CreateLineItemFromOrder(`
`  Order order,`
`  string productCode,`
`  long quantity)`
	Takes the order, the product code, the quantity, and creates the line item.

`OrderLine`
`CreateOrderLineFromOrder(`
`  Order order,`
`  string productCode,`
`  long quantity)`
	Takes the order, the product code, the quantity, and creates the order line.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.2.12 OrderProcessingStrategy

This is an abstract class. You can implement it to get the different types of log entries.

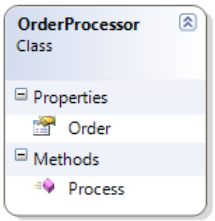


The methods of the `OrderProcessingStrategy` class are:

Method	Description
`IList<LogEntry> GetAdditionalLoggingEntriesForSuccess( Order order)`	Takes the order and provides the additional information for the logging engine about the performed action. For example, if you capture an order, the main action is *Capture Order*. At the same time the substate of the order is also changed.
`LogEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters, and performs additional failure processing.
`LogEntry GetLoggingEntryForSuccess( Order order)`	Takes the order and gets the logged message of the successful operation.
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.

2.2.13 OrderProcessor

This class is used to process an order.



The only property in the `OrderProcessor` class is:

Property	Description
`Order Order`	Gets or sets the order.

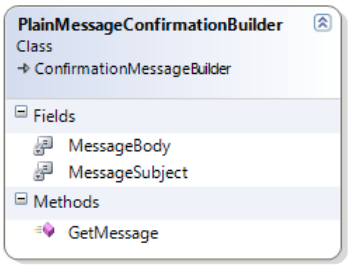
The only method in the `OrderProcessor` class is:

Method	Description
`void Process( ProcessingStrategy processingStrategy)`	Takes the processing strategy and processes the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.2.14 PlainMessageConfirmationBuilder

This class implements the `ConfirmationMessageBuilder` class. It is used to get the message.

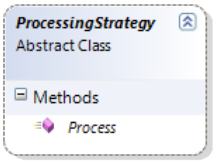


The only method in the `PlainMessageConfirmationBuilder` class is:

Method	Description
`MailMessage GetMessage( string templateName)`	Takes the template name and gets the mail message.

2.2.15 ProcessingStrategy

This is an abstract class that you can implement to process an order.

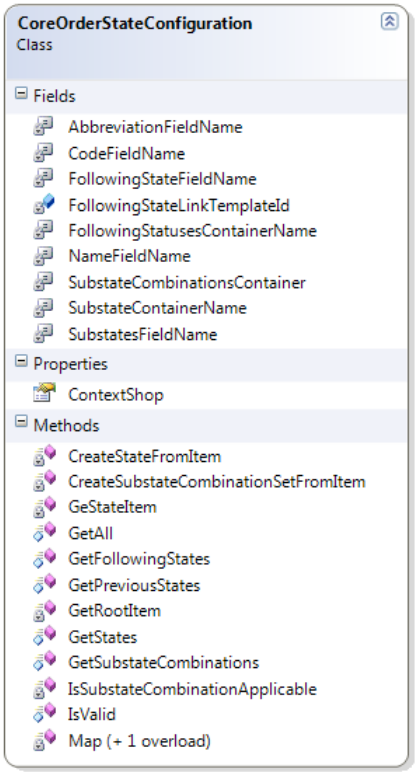


The only method in the `PocessingStrategy` class is:

Method	Description
`void Process( Order order)`	Takes the order and runs the processor.

2.2.16 CoreOrderStateConfiguration

This class reads the order state configuration from the webshop business catalog.



The only property of the `CoreOrderStateConfiguration` class is:

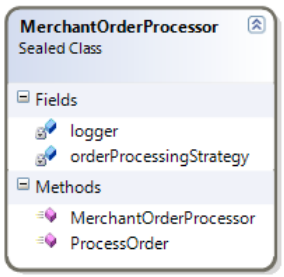
Property	Description
`ShopContext ContextShop`	Gets or sets the shop context.

The methods of this class are either protected or private because they are only used internally. However, you can use the methods in the corresponding class: `MerchantOrderStateConfiguration` instead of this class.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.2.17 MerchantOrderProcessor

This is a sealed class that defines the merchant order processor.



The methods of the `MerchantOrderProcessor` class are:

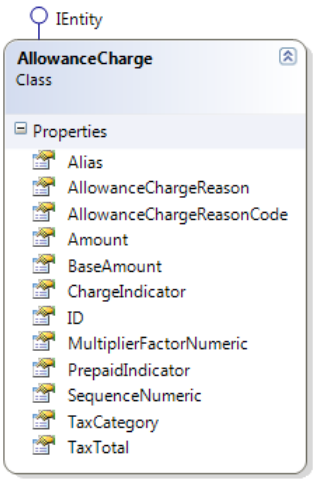
Method	Description
`MerchantOrderProcessor( OrderProcessingStrategy orderProcessingStrategy)`	Takes the order processing strategy and initializes a new instance of the `MerchantOrderProcessor` class.
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3 Sitecore.Ecommerce.Core.OrderManagement.Orders

2.3.1 AllowanceCharge

This class implements the `IEntity` interface and contains the information about the charge or the discount.



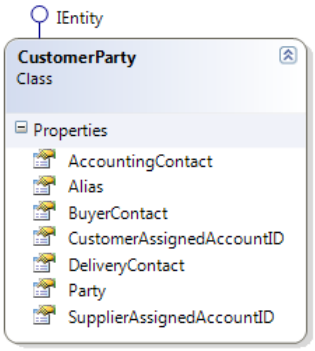
The properties of the `AllowanceCharge` class are:

Property	Description
`long Alias`	Gets the ID of the allowance charge that is used in the database.
`string AllowanceChargeReason`	Gets or sets the reason for the allowance charge.
`string AllowanceChargeReasonCode`	Gets or set sets the code that represents the reason for the allowance charge.
`Amount Amount`	Gets or sets the amount of the allowance charge.
`Amount BaseAmount`	Gets or sets the amount that is used for the `MultiplierFactorNumeric` when calculating the discount.
`bool ChargeIndicator`	Gets or sets the indicator that classifies the charge as either fees or a discount.
`string ID`	Gets or sets the allowance charge identifier that is used in a third party database.
`decimal MultiplierFactorNumeric`	Gets or sets the factor that is applied to the base amount to calculate the allowance charge.
`bool PrepaidIndicator`	Gets or sets the indicator that classifies the allowance charge as prepaid or not. It is set to `true` if it is prepaid.
`decimal SequenceNumeric`	Gets or sets the numerical order sequence in which the allowance charge is calculated when multiple allowance charges apply. If all allowance charges apply to the same base amount, `SequenceNumeric` is `'1'` for all allowance charges.
`TaxCategory TaxCategory`	Gets or sets the information about the tax category
`TaxTotal TaxTotal`	Gets or sets the total tax value of the allowance charge.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.2 CustomerParty

This class implements the `IEntity` interface and contains information about the customer party. The customer party is either the buyer or the entity that is paying.

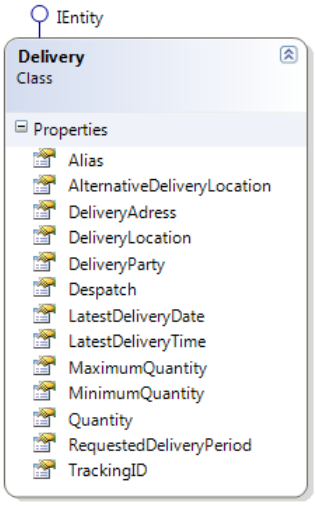


The properties of the `CustomerParty` class are:

Property	Description
`Contact AccountingContact`	Gets or sets the information about the accounting contact of the buyer.
`long Alias`	Gets the ID of the customer party that is used in the database.
`Contact BuyerContact`	Gets or sets the information about the buyer.
`string CustomerAssignedAccountID`	Gets or sets the ID of the account given by the customer.
`Contact DeliveryContact`	Gets or sets the information about the delivery contact.
`Party Party`	Gets or sets the information about the party.
`string SupplierAssignedAccountID`	Gets or sets the ID of the customer account given by the supplier.

2.3.3 Delivery

This class implements the `IEntity` interface and contains information about the delivery of the order.



The properties of the `Delivery` class are:

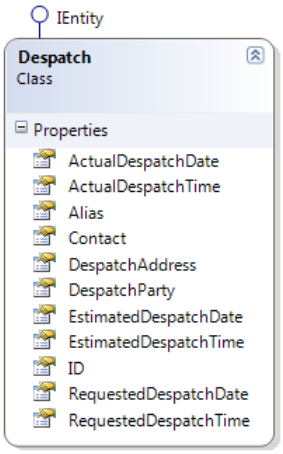
Property	Description
`long Alias`	Gets the ID of the delivery that is used in the database.
`Location AlternativeDeliveryLocation`	Gets or sets the alternative delivery location.
`Address DeliveryAdress`	Gets or sets the information about the delivery address.
`Location DeliveryLocation`	Gets or sets the information about the delivery location.
`Party DeliveryParty`	Gets or stet the information about the delivery receiver.
`Despatch Despatch`	Gets or sets the information about the sender of the delivery.
`DateTime LatestDeliveryDate`	Gets or sets the latest date for the delivery that the buyer determines.
`TimeSpan LatestDeliveryTime`	Gets or sets the latest times for the delivery that the buyer determines.
`decimal MaximumQuantity`	Gets or sets the maximum quantity for the delivery.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`decimal MinimumQuantity`	Gets or sets the minimum quantity for the delivery.
`decimal Quantity`	Gets or sets the quantity for the delivery.
`Period RequestedDeliveryPeriod`	Gets or sets the delivery period that the buyer requests.
`string TrackingID`	Gets or sets the tracking ID of the delivery.

2.3.4 Despatch

This class implements the `IEntity` interface and contains information about the delivery dispatch.



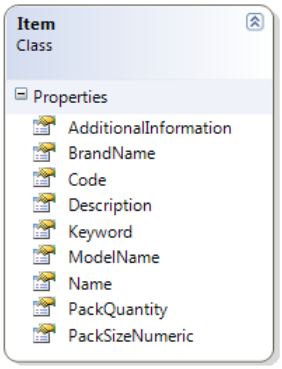
The properties of the `Despatch` class are:

Property	Description
`DateTime ActualDespatchDate`	Gets or sets the actual dispatch date.
`TimeSpan ActualDespatchTime`	Gets or sets the actual dispatch time.
`long Alias`	Gets the ID of the dispatch that is used in the database.
`Contact Contact`	Gets or sets the information about the contact.
`Address DespatchAddress`	Gets or sets the address of the dispatch.
`Party DespatchParty`	Gets or sets the party that dispatches the delivery.
`DateTime EstimatedDespatchDate`	Gets or sets the dispatch date that is estimated by the seller.
`DateTime EstimatedDespatchTime`	Gets or sets the dispatch time that is estimated by the seller.
`string ID`	Gets or sets the identifier of the delivery.
`DateTime RequestedDespatchDate`	Gets or sets the latest dispatch date that the buyer requests.
`TimeSpan RequestedDespatchTime`	Gets or sets the latest dispatch time that the buyer requests.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.5 Item

This class contains information about the item.

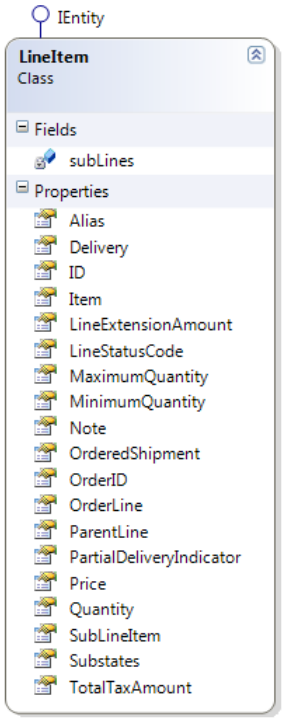


The properties of the `Item` class are:

Property	Description
`string AdditionalInformation`	Gets or sets the additional information about the item, such as the URL of the relevant web page.
`string BrandName`	Gets or sets the brand name of the item.
`string Code`	Gets or sets the code of the item.
`string Description`	Gets or sets the description of the item.
`string Keyword`	Gets or sets the keyword for the product search.
`string ModelName`	Gets or sets the model name of the item.
`string Name`	Gets or sets the name of the item.
`decimal PackQuantity`	Gets or sets the number of the packages.
`decimal PackSizeNumeric`	Gets or sets the number of items in the package.

2.3.6 LineItem

This class implements the `IEntity` interface and contains information about the order line item.



The properties of the `LineItem` class are:

Property	Description
`long Alias`	Gets the ID of the order line item that is used in the database.
`ICollection<Delivery> Delivery`	Gets or sets the information about the request for delivery.
`string ID`	Gets or sets the address identifier that is used in a third party database.
`Item Item`	Gets or sets the information about the item on the order line.
`Amount LineExtensionAmount`	Gets or sets the total amount for line item including the sum of all order lines, the allowance, the charges, and excluding the taxes.

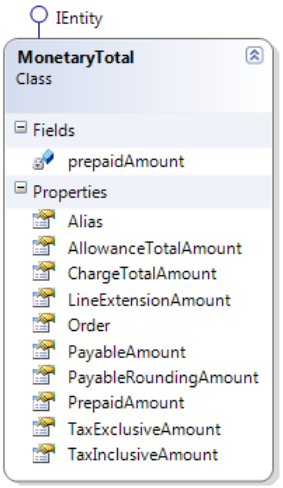
Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`string LineStatusCode`	Gets or sets the status of the line whether it is: • A new line • A change in the line • A deletion of the line.
`decimal MaximumQuantity`	Gets or sets the maximum quantity of products that you can order.
`decimal MinimumQuantity`	Gets or sets the minimum quantity for ordering.
`string Note`	Gets or sets the free text field. **Note** This element may contain notes or any information that does not exist in other structures or classes.
`OrderedShipment OrderedShipment`	Gets or sets the information about the order shipment.
`string OrderID`	Gets or sets the merchant order ID.
`OrderLine OrderLine`	Gets or sets the order line.
`LineItem ParentLine`	Gets or sets the parent line.
`bool PartialDeliveryIndicator`	Gets or sets the value whether the partial delivery allowed.
`Price Price`	Gets or sets the price of the item.
`decimal Quantity`	Gets or sets the quantity of the item.
`ICollection<LineItem> SubLineItem`	Gets the sub-line item.
`ICollection<Substate> Substates`	Gets or sets the sub-states.
`Amount TotalTaxAmount`	Gets or sets the tax total amount for the order line.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.7 MonetaryTotal

This class implements the `IEntity` interface and contains information about the different monetary amounts.



The properties of the `MonetaryTotal` class are:

Property	Description
`long Alias`	Gets the ID of the monetary total that is used in the database.
`Amount AllowanceTotalAmount`	Gets the total amount of all the allowances.
`Amount ChargeTotalAmount`	Gets the total amount of all the charges.
`Amount LineExtensionAmount`	Gets the line extension amount without the tax and settlement discounts, but including any applicable rounding amount.
`Order Order`	Gets or sets the order.
`Amount PayableAmount`	Gets the amount to be paid.
`Amount PayableRoundingAmount`	Gets the payable rounding amount, whether positive or negative, that is added to the calculated line extension amount to get the rounded line extension amount
`Amount PrepaidAmount`	Gets or sets the total prepaid amount.
`Amount TaxExclusiveAmount`	Gets the tax without amount.
`Amount TaxInclusiveAmount`	Gets the tax including amount.

2.3.8 Order

The `Order` class is used to exchange the electronic orders and to process the coherent order-to-invoice flow.

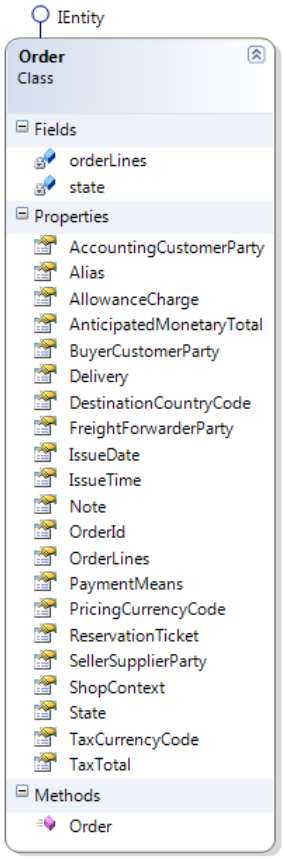
The order is sent from the `BuyerCustomerParty` object to the `SellerSupplierParty` object. It is structured to fulfill the formal requirements for a valid order change and for use in Denmark, such as tax number, time of delivery, and specifications.

Apart from the formal demands on the sender/receiver and the order content, the electronic version must contain the *End Point ID*, a *Person reference* and usually an order number or request number. It must also fulfill the requirements in this document guideline.

- The *End Point ID* is used to identify the *electronic mailbox* to which the electronic document is delivered and also provides the basis for routing the message.
- The *Person Reference* is used to identify the person who is responsible for placing the order.
- The *order number* or *requisition number* is used to identify the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

These three types of information are used subsequently when the electronic invoice is issued with reference to the previous order or requisition. Depending on the profile that is used, the order can take part of an advanced order flow to allow the exchange of order changes, order deletions and order responses.



The properties of the `Order` class are:

Property	Description
`CustomerParty AccountingCustomerParty`	Gets or sets the information about the accounting customer party. This is the party to which the invoice is sent.
`long Alias`	Gets the order's ID that is used in the database.
`ICollection<AllowanceCharge> AllowanceCharge`	Gets or sets the information about the allowances and charges that apply to the order as a whole.
`MonetaryTotal AnticipatedMonetaryTotal`	Gets or sets the net monetary total, excluding allowances and taxes, as expected by the buyer.
`CustomerParty BuyerCustomerParty`	Gets or sets the information about the buyer.
`ICollection<Delivery> Delivery`	Gets or sets the information about delivery.
`string DestinationCountryCode`	Gets or sets the destination country code. **Note** This code is used for customs purposes.
`ICollection<Party> FreightForwarderParty`	Gets or sets the information about the freight carrier.
`DateTime IssueDate`	Gets or sets the date on which the order is issued.
`TimeSpan IssueTime`	Gets or sets the time at which the order is issued.
`string Note`	Gets or sets the note on the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

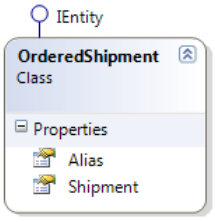
Property	Description
`string OrderId`	Gets or sets the order identifier that the buyer specifies.
`ICollection<OrderLine> OrderLines`	Gets or sets one or more order lines.
`PaymentMeans PaymentMeans`	Gets or sets the information about the payment type.
`string PricingCurrencyCode`	Gets or sets the currency code for the price.
`ReservationTicket ReservationTicket`	Gets or sets the reservation ticket.
`SupplierParty SellerSupplierParty`	Gets or sets the information about the seller.
`ShopContext ShopContext`	Gets or sets the shop context.
`State State`	Gets or sets the state of the order.
`string TaxCurrencyCode`	Gets or sets the code of the currency that is requested for the tax amount in the order invoices.
`TaxTotal TaxTotal`	Gets or sets the information about the total tax value of the order as calculated by the buyer.

The only method in the `order` class is:

Method	Description
`Order()`	Initializes a new instance of the `Order` class.

2.3.9 OrderedShipment

This class implements the `IEntity` interface and contains information about the order shipment.



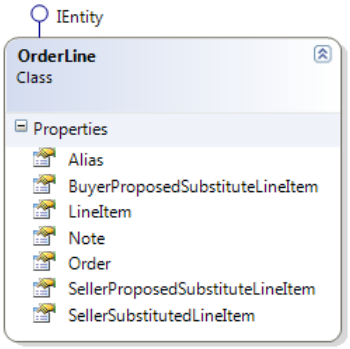
The properties of the `OrderedShipment` class are:

Property	Description
`long Alias`	Gets the ID of the ordered shipment that is used in the database.
`Shipment Shipment`	Gets or sets the shipment information.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.10 OrderLine

This class implements the `IEntity` interface and contains information about the order line.

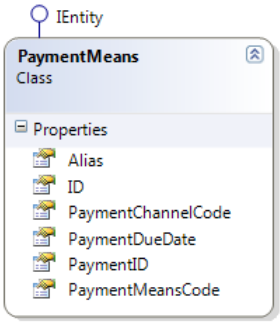


The properties of the `OrderLine` class are:

Property	Description
`long Alias`	Gets the ID of the order line that is used in the database.
`ICollection<LineItem> BuyerProposedSubstituteLineItem`	Gets or sets the possible alternative items that the buyer proposes.
`LineItem LineItem`	Gets or sets the order line item.
`string Note`	Gets or sets the content of the free text note of the order line. **Note** This property can contain notes that are not explicitly stated in any other class.
`Order Order`	Gets or sets the order.
`ICollection<LineItem> SellerProposedSubstituteLineItem`	Gets or sets the possible alternative items that the seller proposes.
`ICollection<LineItem> SellerSubstitutedLineItem`	Gets or sets the actual alternative items that the seller gives.

2.3.11 PaymentMeans

This class implements the `IEntity` interface and contains information about the order payment means.



The properties of the `PaymentMeans` class are:

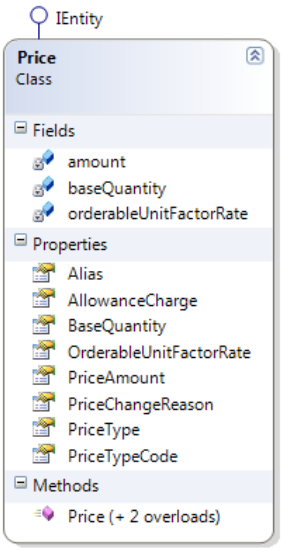
Property	Description
`long Alias`	Gets the ID of the payment means that is used in the database.
`string ID`	Gets or sets the payment means identifier that is used in a third party database.
`string PaymentChannelCode`	Gets or sets the payment channel code.
`DateTime PaymentDueDate`	Gets or sets the date on which the payment is due.
`string PaymentID`	Gets or sets the identifier of the payment.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`string PaymentMeansCode`	Gets or sets the identifier of the payment means.

2.3.12 Price

This class implements the `IEnity` interface and contains information about the price of the order.



The properties of the `Price` class are:

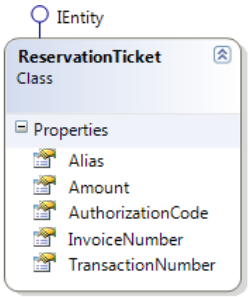
Property	Description
`long Alias`	Gets the ID of the price that is used in the database.
`IEnumerable<AllowanceCharge> AllowanceCharge`	Gets or sets the allowance charge.
`decimal baseQuantity`	Gets or sets the base quantity. It is the actual quantity to which the Price applies
`Amount PriceAmount`	Gets or sets the price value.
`string PriceChangeReason`	Gets or sets the price change reason.
`string PriceType`	Gets or sets the price type.
`string PriceTypeCode`	Gets or sets the price type code.

The methods in the `Price` class are:

Method	Description
`Price()`	Initializes a new instance of the Price class.
`Price( Amount amount)`	Takes the price value and initializes a new instance of the Price class.
`Price( Amount amount, decimal baseQuantity)`	Takes the price value, the base quantity and initializes a new instance of the Price class.

2.3.13 ReservationTicket

This class implements the `IEntity` interface and contains information about the reservation ticket.



The properties of the `ReservationTicket` class are:

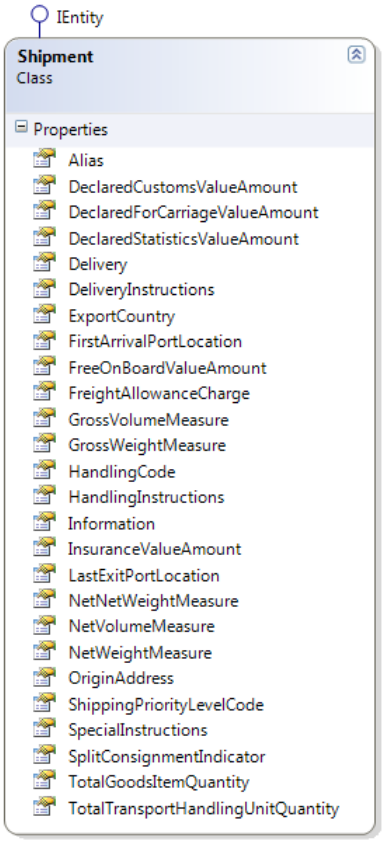
Property	Description
`long Alias`	Gets the ID of the reservation ticket that is used in the database.
`decimal Amount`	Gets or sets the total order amount reserved on the credit card for a later withdrawal when the order has been shipped.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`string AuthorizationCode`	Gets or sets the authorization code. This code is an arbitrary value that is produced and received from the payment provider. It must be used later when receiving the money from the same payment provider. It identifies the reservation of money.
`string InvoiceNumber`	Gets or sets the invoice number.
`string TransactionNumber`	Gets or sets the transaction number.

2.3.14 Shipment

This class implements the `IEnity` interface and contains information about the shipment of an order.



The properties of the `Shipment` class are:

Property	Description
`long Alias`	Gets the ID of the shipment that is used in the database.
`Amount DeclaredCustomsValueAmount`	Gets or sets the amount of the declared customs value. It is used for the goods in a shipment which are subject to the same customs procedure, and have the same tariff/statistical heading, country information and duty regime.
`Amount DeclaredForCarriageValueAmount`	Gets or sets the amount of the declared for carriage value.
`Amount DeclaredStatisticsValueAmount`	Gets or sets the amount of the declared statistics value.
`Delivery Delivery`	Gest or sets the delivery information of the shipment.
`string DeliveryInstructions`	Gets or sets the delivery instructions.
`string ExportCountry`	Gets or sets the country from which the goods are originally exported.
`Location FirstArrivalPortLocation`	Gets or sets the first arrival location. **Note** This location can be a port, an airport, or a border crossing.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

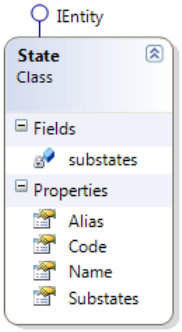
Property	Description
`Amount FreeOnBoardValueAmount`	Gets or sets the monetary amount that is paid as calculated under the applicable trade delivery.
`AllowanceCharge FreightAllowanceCharge`	Gets or sets the transportation cost that is added by the shipper under the terms of the carriage contract. **Note** In addition to transportation costs, this may include items such as packing, documentation, loading, unloading, and the insurance.
`Measure GrossVolumeMeasure`	Gets or sets the total volume of the goods including the packaging in the shipment.
`Measure GrossWeightMeasure`	Gets or sets the weight of the goods without packaging and without the transport equipment.
`string HandlingCode`	Gets or sets the handling code that is required for the shipment.
`string HandlingInstructions`	Gets or sets the handling instructions for the shipment
`string Information`	Gets or sets the information about the shipment. **Note** This element may contain notes or any other similar information that is not contained explicitly in another structure.
`Amount InsuranceValueAmount`	Gets or sets the insurance coverage of the shipment.
`Location LastExitPortLocation`	Gets or sets the final exporting location. **Note** This location can be a port, an airport, or a border crossing.
`Measure NetNetWeightMeasure`	Gets or sets the weight of the goods without packaging and without the transport equipment
`Measure NetVolumeMeasure`	Gets or sets the volume of a shipment without packaging and the transport equipment
`Measure NetWeightMeasure`	Gets or sets total weight of the goods including the packaging.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Property	Description
`Address OriginAddress`	Gets or sets the address of the place where the goods are produced. **Note** This address is used for the customs purposes such as customs tariff, quantitative restrictions, or any other measure related to the trade.
`string ShippingPriorityLevelCode`	Gets or sets the code of the service priority.
`string SpecialInstructions`	Gets or sets the special instructions for the shipment.
`bool SplitConsignmentIndicator`	Gets or sets the value that indicates whether the consignment has been split in transit or not.
`decimal TotalGoodsItemQuantity`	Gets or sets the number of goods items in the shipment.
`decimal TotalTransportHandlingUnitQuantity`	Gets or sets the number of pieces of transport handling. These are the equipment that is used in the shipment, such as the pallets, the boxes, and the cases.

2.3.15 State

This class implements the `IEntity` interface and contains information about the state.



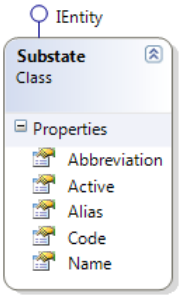
The properties of the `State` class are:

Property	Description
`long Alias`	Gets the ID of the state that is used in the database.
`string Code`	Get or sets the code that identifies the state.
`string Name`	Gets or sets the name of the state.
`ICollection<Substate> Substates`	Gets the sub-states. A state potentially contains a collection of substates.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.16 Substate

This class implements the `IEntity` interface and contains information about the substate.

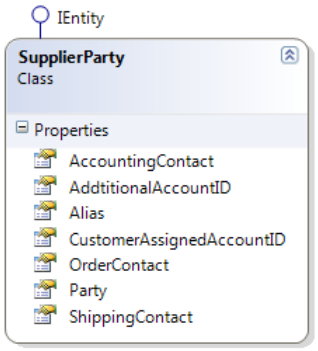


The properties of the `Substate` class are:

Property	Description
`string Abbreviation`	Gets or sets the abbreviation.
`bool Active`	Gets or sets a value that indicates whether this sub-state is active or not.
`long Alias`	Gets the ID of the sub-state that is used in the database.
`string Code`	Gets or sets the code of the sub-state.
`string Name`	Get or sets the name of the sub-state.

2.3.17 SupplierParty

This class implements the `IEntity` interface and contains information about the supplier.



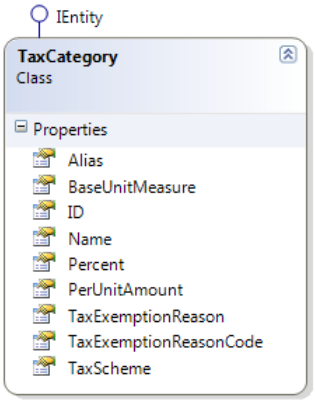
The properties of the `SupplierParty` class are:

Property	Description
`Contact AccountingContact`	Get or sets the information about the accounting contact of the supplier.
`string AddtitionalAccountID`	Gets or sets the ID of the supplier account that is used in the third party system. It acts as a foreign key if you want to look up the supplier in that other system.
`long Alias`	Gets the ID of the supplier party that is used in the database.
`string CustomerAssignedAccountID`	Gets or sets the ID of the account of the customer.
`Contact OrderContact`	Gets or sets the information about the seller.
`Party Party`	Gets or sets the information about the party.
`Contact ShippingContact`	Gets or sets the information about the shipping contact.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.18 TaxCategory

This class implements the `IEntity` interface and contains information about the category of the tax.

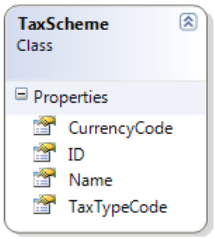


The properties of the `TaxCategory` class are:

Property	Description
`long Alias`	Gets the ID of the tax category that is used in the database.
`Measure BaseUnitMeasure`	Gets or sets the measure that is used to calculate the tax, if the tax is applied at a certain rate per unit.
`string ID`	Gets or sets the identifier of the tax category.
`string Name`	Gets or sets the name of the tax category.
`decimal Percent`	Gets or sets the percentage of the tax category.
`Amount PerUnitAmount`	Gets or sets the tax that is set per unit if it is defined.
`string TaxExemptionReason`	Gets or sets the reason for not paying the tax.
`string TaxExemptionReasonCode`	Gets or sets the code that represents the reason for not paying the tax.
`TaxScheme TaxScheme`	Gets or sets the tax scheme.

2.3.19 TaxScheme

This class contains information about the scheme of the tax.



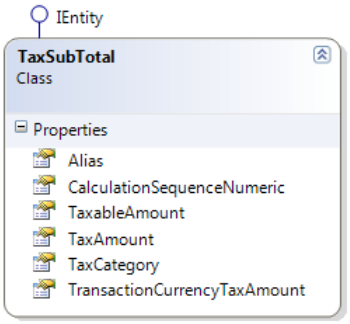
The properties of the `TaxScheme` class are:

Property	Description
`string CurrencyCode`	Gets or sets the code of the currency in which the tax is collected.
`string ID`	Gets or sets the identifier of the tax scheme.
`string Name`	Gets or sets the tax area.
`string TaxTypeCode`	Gets or sets the identification for the tax type.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.3.20 TaxSubTotal

This class implements the `IEntity` interface. It contains information about the subtotal of a particular tax category within a tax type such as the standard tax rate within the VAT region.

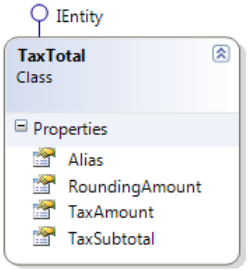


The Properties of `TaxSubTotal` class are:

Property	Description
`long Alias`	Gets the ID of the tax sub-total that is used in the database.
`decimal CalculationSequenceNumeric`	Gets or sets the numerical order in which the taxes are applied when multiple taxes are defined. If all taxes apply to the same taxable amount, `CalculationSequenceNumeric` is set to `'1'` for all taxes
`Amount TaxableAmount`	Gets or sets the net amount to which the tax rate is applied.
`Amount TaxAmount`	Gets the amount of the tax.
`TaxCategory TaxCategory`	Gets or sets the category of the tax.
`Amount TransactionCurrencyTaxAmount`	Gets or sets the tax amount presented in the currency of the voice.

2.3.21 TaxTotal

This class implements the `IEntity` interface and contains information about the total amount of a particular tax type.



The properties of the `TaxTotal` class are:

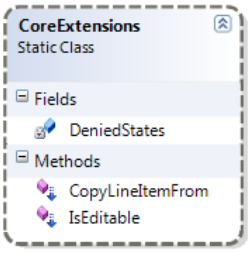
Property	Description
`long Alias`	Gets the ID of the tax total that is used in the database.
`Amount RoundingAmount`	Gets or sets the rounding amount for taxes. **Note** The rounding amount, whether positive or negative, is added to the calculated tax total to produce the rounded total tax value.
`Amount TaxAmount`	Gets the total for all taxes of a tax type. **Note** This value is a calculated as the sum of the tax sub totals for each tax category within the tax type.
`ICollection<TaxSubTotal> TaxSubtotal`	Gets or sets infromation about the tax subtotal of the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.4 Sitecore.Ecommerce.Core.OrderManagement.Extensions

2.4.1 CoreExtensions

This is a static class that defines the extensions of the core.



The methods of the `CoreExtension` class are:

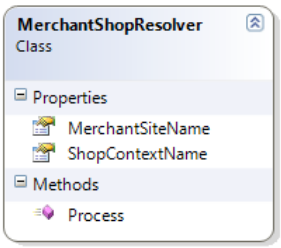
Method	
`void CopyLineItemFrom( LineItem acceptor, LineItem donor)`	Takes the acceptor line item, the donor line item, and copies the data partially from one line item to another.
`bool IsEditable( Order order)`	Takes the order and checks if the order is editable.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.5 Sitecore.Ecommerce.Merchant.Pipelines.HttpRequest

2.5.1 MerchantShopResolver

This class defines the merchant shop resolver.



The properties of the `MerchantShopResolver` class are:

Property	Description
`string MerchantSiteName`	Gets or sets the name of the merchant site.
`string ShopContextName`	Gets or sets the name of the shop context.

The only method of the `MerchantShopResolver` class is:

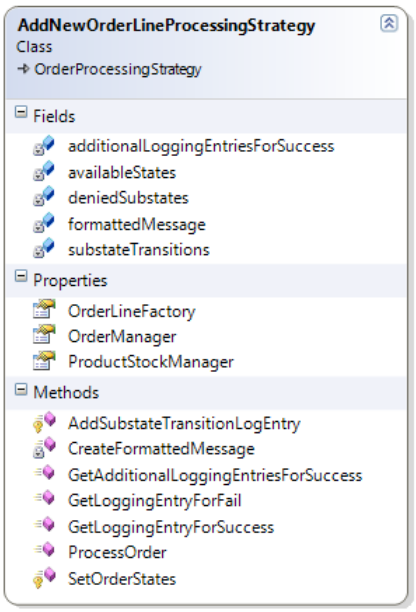
Method	Description
`void Process( PipelineArgs args)`	Takes the pipeline arguments and runs the processor.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6 Sitecore.Ecommerce.Merchant.OrderManagement

2.6.1 AddNewOrderLineProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the strategy for adding a new order line to the existing order.



The properties of the `AddNewOrderLineProcessorStrategy` class are:

Property	Description
`MerchantOrderManager OrderManager`	Gets or sets the order manager.
`IProductStockManager ProductStockManager`	Gets or sets the product stock manager.
`OrderLineFactory OrderLineFactory`	Gets or sets the order line factory.

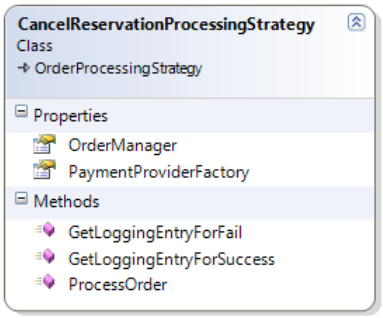
The methods of the `AddOrderLineProcessorStartegy` class are:

Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters and processes the order.
`LoggingEntry GetLoggingEntryForSuccess( Order order)`	Takes the order and gets the log entry for the operations that resulted in success.
`LoggingEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters, and gets the log entry for the operations that resulted in failure.
`IList<LoggingEntry> GetAdditionalLoggingEntriesForSuccess( Order order)`	Takes the order and gets the additional log entries for success.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6.2 CancelReservationProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the strategy for canceling the reservation of the money.



The properties of the `CancelReservationProcessingStrategy` class are:

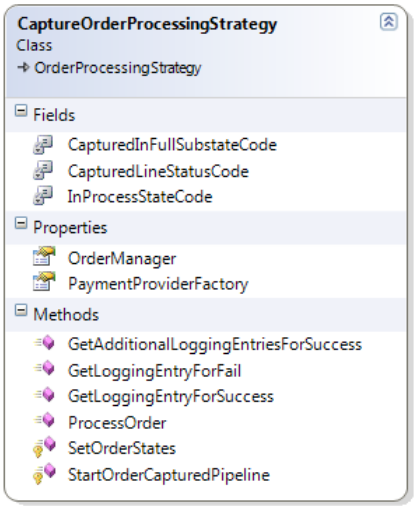
Property	Description
`MerchantOrderManager OrderManager`	Gets or sets the order manager.
`PaymentProviderFactory PaymentProviderFactory`	Gets or sets the payment provider factory.

The methods of the `CancelReservationProcessingStrategy` class are:

Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.
`LoggingEntry GetLoggingEntryForSuccess( Order order)`	Takes the order and gets the logged message of the successful operation.
`LoggingEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters, and gets the logged message of the operation that failed.

2.6.3 CaptureOrderProcessingStrategy

This class implements the `OrderPrcessingStrategy` class and defines the processing strategy for capturing the order.



The properties of the `CaptureOrderProcessingStrategy` class are:

Property	Description
`MerchantOrderManager OrderManager`	Gets or sets the order manager.
`PaymentProviderFactory PaymentProviderFactory`	Gets or sets the payment provider factory.

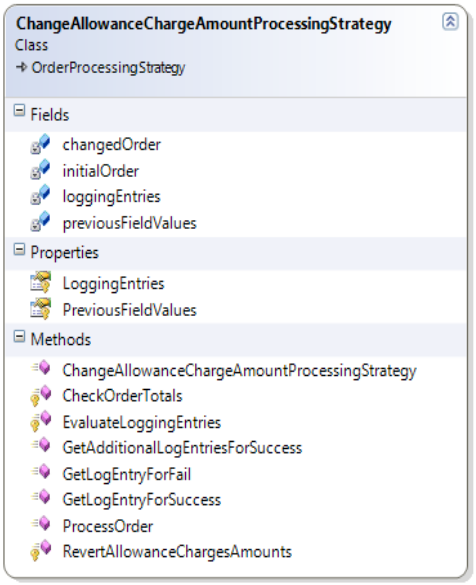
Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

The methods of the `CaptureOrderProcessingStrategy` class are:

Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.
`LoggingEntry GetLoggingEntryForSuccess( Order order)`	Takes the order, and gets the logged message of the successful operation.
`IList<LoggingEntry> GetAdditionalLoggingEntriesForSuccess( Order order)`	Takes the order, and gets the additional logged message of the successful operation.
`LoggingEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters and gets the logged message of the operation that failed.

2.6.4 ChangeAllowanceChargeAmountProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the processing strategy of the allowance charge amount value.



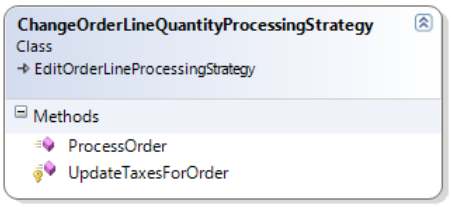
The methods of the `ChangeAllowanceChargeAmountProcessing` are:

Method	Description
`ChangeAllowanceChargeAmountProcessingStrategy( Order inintialOrder, IDictionary<string, object> previousFieldValues)`	Initializes a new instance of the class.
`IList<LogEntry> GetAdditionalLogEntriesForSuccess(Order order)`	Performs additional success processing.
`LogEntry GetLogEntryForFail( Order order, params object[] parameters)`	Performs additional failure processing.
`LogEntry GetLogEntryForSuccess( Order order)`	Gets logging entries for success.
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Processes the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6.5 ChangeOrderLineQuantityProcessingStrategy

This class implements the `EditOrderLineProcessingStrategy` class and defines the processing strategy for changing the order line quantity.

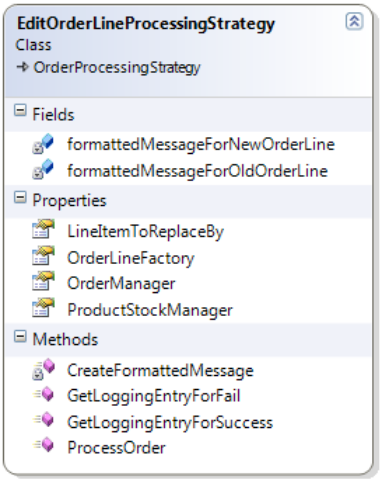


The only public method of the `ChangeOrderLineQuanityProcessingStrategy` class is:

Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Processes the order.

2.6.6 EditOrderLineProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the processing strategy for editing the order line.



The properties of the `EditOrderLineProcessingStrategy` class are:

Property	Description
`IProductStockManager ProductStockManager`	Gets or sets the product stock manager.
`MerchantOrderManager OrderManager`	Gets or sets the order manager.
`OrderLineFactory OrderLineFactory`	Gets or sets the order line factory.
`LineItem LineItemToReplaceBy`	Gets or sets the order line item to be added.

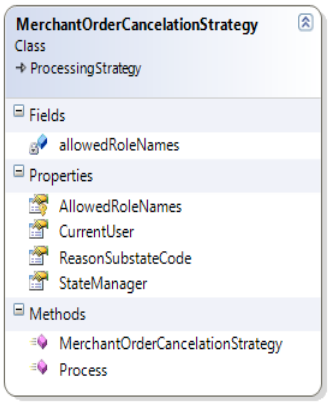
The methods of the `EditOrderLineProcessingStrategy` class are:

Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.
`LoggingEntry GetLoggingEntryForSuccess( Order order)`	Takes the order, and gets the logged message of the successful operation.
`LoggingEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters, and the gets the logged message of the operation that failed.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6.7 MerchantOrderCancelationStrategy

This class implements the `ProcessingStrategy` class and defines the strategy for cancelling the merchant order.



The properties of the `MerchantOrderCancelationStrategy` class are:

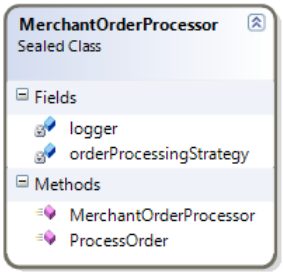
Property	Description
`Core.Contracts.OrderManagement.StateManager<State> StateManager`	Gets or sets the state manager.
`User CurrentUser`	Gets or sets the current user.
`string ReasonSubstateCode`	Gets or sets the reason sub-state code.
`virtual IEnumerable<string> AllowedRoleNames`	Gets the allowed role names.

The methods of the `MerchantOrderCancelationStrategy` class are:

Method	Description
`void Process(Order order)`	Takes the order and processes it.
`MerchantOrderCancelationStrategy()`	Initializes a new instance of the `MerchantOrderCancelationStrategy` class.

2.6.8 MerchantOrderProcessor

This class is sealed and defines the merchant order processor.



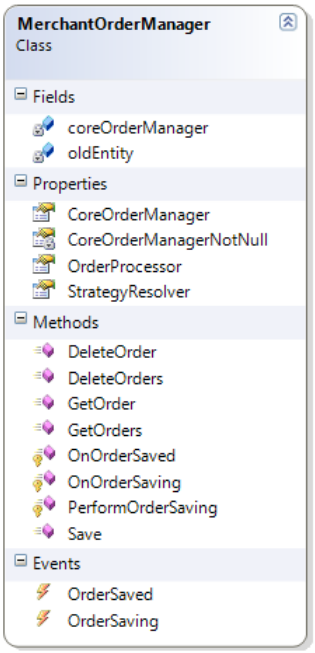
The methods of the `MerchantOrderProcessor` class are:

Method	Description
`MerchantOrderProcessor( OrderProcessingStrategy orderProcessingStrategy)`	Takes the order processing strategy and initializes a new instance of the `MerchantOrderProcessor` class.
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order. **Note** You can pass any stuff to `parameters` to handle your own business scenario.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6.9 MerchantOrderManager

This class defines the merchant order manager.



The properties of the `MerchantOrderManager` class are:

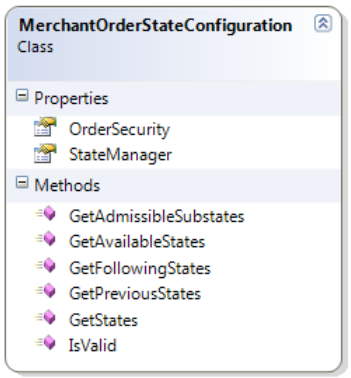
Property	Description
`CoreOrderManager CoreOrderManager`	Gets or sets the core order manager.
`MerchantOrderProcessor OrderProcessor`	Gets or sets the order processor.
`OrderProcessingStrategyResolver StrategyResolver`	Gets or sets the strategy resolver.

The methods of the `MerchantOrderManager` class are:

Method	Description
`void DeleteOrder( Order order)`	Deletes the order.
`void DeleteOrders( Expression<Func<Order, bool>> expression)`	Takes the Linq expression and deletes the orders.
`Order GetOrder( string orderId)`	Takes the order ID and returns the order.
`IQueryable<Order> GetOrders( Expression<Func<Order, bool>> expression)`	Takes the Linq expression and returns the orders.
`void Save( Order order)`	Saves the order.

2.6.10 MerchantOrderStateConfiguration

This class defines the merchant order state configuration.



The properties of the `MerchantOrderStateConfiguration` are:

Property	Description
`OrderSecurity OrderSecurity`	Gets or sets the order security.
`CoreOrderStateConfiguration StateManager`	Gets or sets the state manager.

The Methods of the `MerchantOrderSateConfiguration` are:

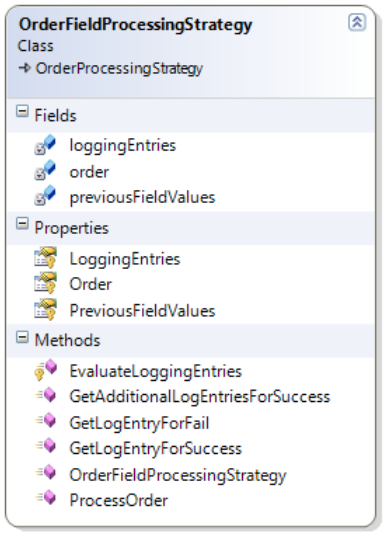
Method	Description
`IQueryable<Substate> GetAdmissibleSubstates( State state)`	Gets the admissible substates of the merchant order.
`IQueryable<State> GetAvailableStates( Order order)`	Gets the available states of the merchant order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Method	Description
`IQueryable<State> GetFollowingStates( State state)`	Takes the merchant order state and gets the following states.
`IQueryable<State> GetPreviousStates( State state)`	Takes the merchant order state and gets the previous states.
`IQueryable<State> GetStates()`	Gets the merchant order states.
`bool IsValid( State state)`	Determines whether the specified merchant order state is valid.

2.6.11 OrderFieldProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the strategy for processing the order field.



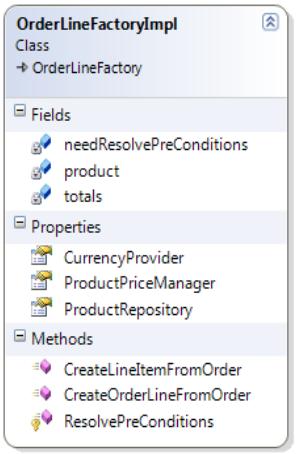
The methods of the `OrderFieldProcessingStrategy` class are:

Method	Description
`IList<LogEntry> GetAdditionalLogEntriesForSuccess( Order order)`	Performs additional success processing.
`LogEntry GetLogEntryForFail( Order order, params object[] parameters)`	Gets the logged message of the operation that failed and performs additional
`LogEntry GetLogEntryForSuccess( Order order)`	Gets the logged message of the successful operation.
`OrderFieldProcessingStrategy( IDictionary<string, object> previousFieldValues)`	Initializes a new instance of the class.
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Processes the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6.12 OrderLineFactoryImpl

This class implements the `OrderLineFactory` and defines the implementation of the order line factory.



The properties of the `OrderLineFactoryImpl` class are:

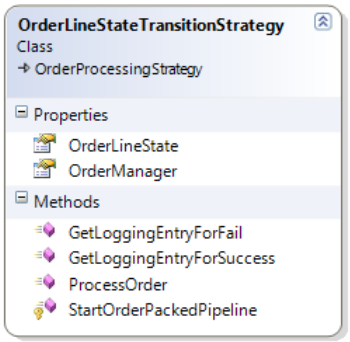
Property	Description
`IProductRepository ProductRepository`	Gets or sets the product repository.
`IProductPriceManager ProductPriceManager`	Gets or sets the product price manager.
`IEntityProvider<DomainModel.Currencies.Currency> CurrencyProvider`	Gets or sets the currency provider.

The methods of the `OrderLineFactoryImpl` class are:

Method	Description
`OrderLine CreateOrderLineFromOrder( Order order, string productCode, long quantity)`	Takes the order, the product code, the quantity, and creates the order line.
`LineItem CreateLineItemFromOrder( Order order, string productCode, long quantity)`	Takes order, the product code, the quantity, and creates the line item.

2.6.13 OrderLineStateTransitionStrategy

This class implements the `OrderProcessingStrategy` class and defines the strategy of the order line state transition.



This class defines the order line state transition strategy.

The properties of the `OrderLineStateTransitionStrategy` class are:

Property	Description
`string OrderLineState`	Gets or sets the order line status.
`MerchantOrderManager OrderManager`	Gets or sets the order manager.

The methods of the `OrderLineStateTransitionStrategy` class are:

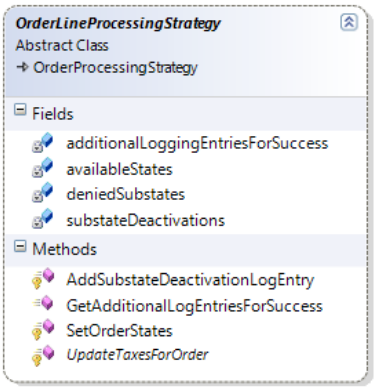
Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Method	Description
`LoggingEntry GetLoggingEntryForSuccess( Order order)`	Takes the order, and gets the logged message of the successful operation.
`LoggingEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters, and gets the logged message of the operation that failed.
`void StartOrderPackedPipeline( Order order)`	Takes the order, and starts the order packed pipeline.

2.6.14 OrderLineProcessingStrategy

This is an abstract class that implements the `OrderProcessingStrategy` and defines the strategy for processing the order line.

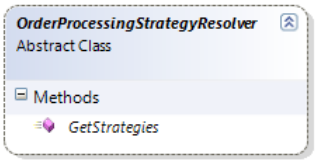


The only public method in the `OrderLineProcessingStrategy` class is:

Method	Description
`IList<LogEntry> GetAdditionalLogEntriesForSuccess( Order order)`	Gets additional logging entry for success.

2.6.15 OrderProcessingStrategyResolver

This is an abstract class that defines the strategy resolver. This resolver is used to process the order.



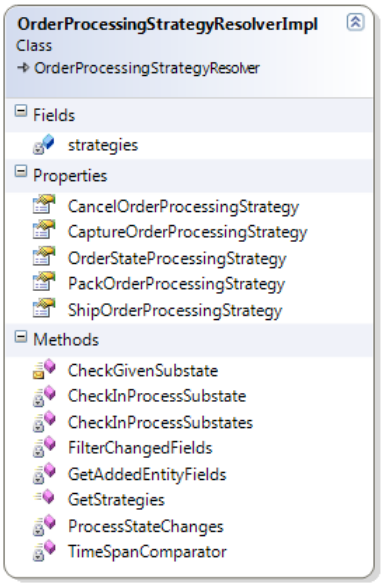
The only method of `OrderProcessingStrategyReslover` class is:

Method	Description
`IEnumerable<OrderProcessingStrategy> GetStrategies( Order oldOrder, Order newOrder)`	Gets the order processing strategy. It compares two orders and decides what to do with them. For example, it may realize that the order has been changed, moved to a process state, or captured. The class will resolve the appropriate strategy that can handle it.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6.16 OrderProcessingStrategyResolverImpl

This class implements the `OrderPorcessingStrategyResolver` class and represents the implementation of the order processing strategy resolver.



The properties of the `OrderProcessingStrategyResolverImpl` class are:

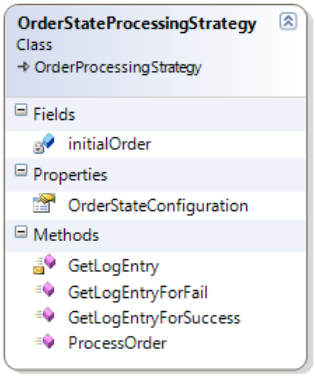
Property	Description
`OrderProcessingStrategy CancelOrderProcessingStrategy`	Gets or sets the cancel order processing strategy.
`OrderProcessingStrategy CaptureOrderProcessingStrategy`	Gets or sets the capture order processing strategy.
`OrderProcessingStrategy OrderStateProcessingStrategy`	Gets or sets the order state processing strategy.
`OrderProcessingStrategy PackOrderProcessingStrategy`	Gets or sets the pack order processing strategy.
`OrderProcessingStrategy ShipOrderProcessingStrategy`	Sets or sets the ship order processing strategy.

The methods of the `OrderProcessingStrategyResolverImpl` class are:

Method	Description
`IEnumerable<OrderProcessingStrategy> GetStrategies( Order oldOrder, Order newOrder)`	Gets the order processing strategies.

2.6.17 OrderStateProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the strategy for processing the order state.



The only property of the `OrderStateProcessingStrategy` class is:

Property	Description
`MerchantOrderStateConfiguration OrderStateConfiguration`	Gets or sets the order state configuration.

The methods of the `OrderStateProcessingStrategy` class are:

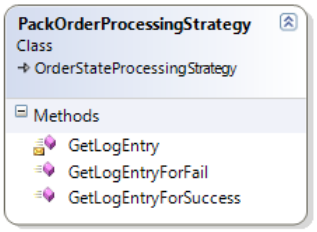
Method	Description
`LogEntry GetLogEntryForFail( Order order, params object[] parameters)`	Gets the logged message of the operation that failed and performs additional failure processing.
`LogEntry GetLogEntryForSuccess( Order order)`	Gets the logged message of the successful operation.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Processes the order.

2.6.18 PackOrderProcessingStrategy

This class implements the `PackStateProcessingStrategy` class and defines the strategy for processing the pack order.

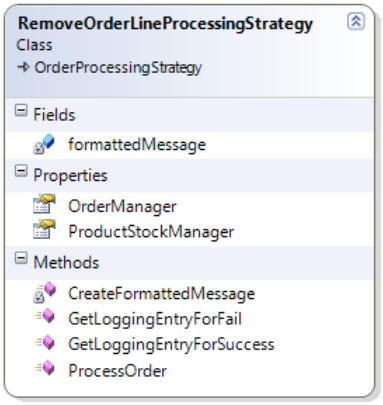


The methods of the `PackOrderProcessingStrategy` class are:

Method	Description
`LogEntry GetLogEntryForFail( Order order, params object[] parameters)`	Gets the logged message of the operation that failed and performs additional failure processing.
`LogEntry GetLogEntryForSuccess( Order order)`	Gets the logged message of the successful operation.

2.6.19 RemoveOrderLineProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the processing strategy for removing the order line.



The properties of the `RemoveOrderLineProcessingStrategy`.class are:

Property	Description
`MerchantOrderManager OrderManager`	Gets or sets the order manager.
`IProductStockManager ProductStockManager`	Gets or sets the product stock manager.

The methods of the `RemoveOrderLineProcessingStrategy` class are:

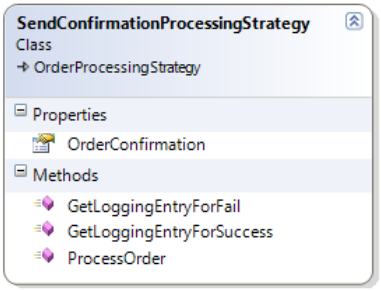
Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.
`LoggingEntry GetLoggingEntryForSuccess( Order order)`	Takes the order, and gets the logged message of the successful operation.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

Method	Description
`LoggingEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters, and gets the logged message of the operation that failed.

2.6.20 SendConfirmationProcessingStrategy

This class implements the `OrderProcessingStrategy` class and defines the processing strategy for sending the order confirmation.



The only property of the `SendConfirmationProcessingStrategy` class is:

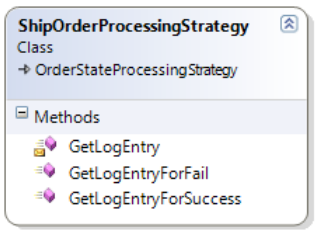
Property	Description
`OrderConfirmation OrderConfirmation`	Gets or sets the order confirmation.

The methods of the `SendConfirmationProcessingStrategy`.class are:

Method	Description
`string ProcessOrder( Order order, IDictionary<string, object> parameters)`	Takes the order, the parameters, and processes the order.
`LoggingEntry GetLoggingEntryForSuccess( Order order)`	Takes the order, and gets the logged message of the successful operation.
`LoggingEntry GetLoggingEntryForFail( Order order, params object[] parameters)`	Takes the order, the parameters, and gets the logged message of the operation that failed.

2.6.21 ShipOrderProcessingStrategy

This class implements the `OrderStateProcessingStrategy` class and defines the processing strategy for shipping the order.



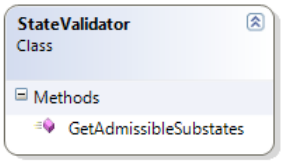
The methods of the `ShipOrderProcessingStrategy` class are:

Method	Description
`LogEntry GetLogEntryForFail( Order order, params object[] parameters)`	Gets the logged message of the operation that failed and performs additional failure processing.
`LogEntry GetLogEntryForSuccess( Order order)`	Gets the logged message of the successful operation.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.6.22 StateValidator

This class defines the state validator.



The only method of the `StateValidator` class is:

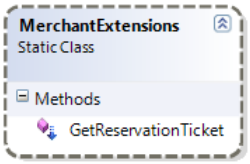
Method	Description
`IEnumerable<Substate> GetAdmissibleSubstates( State state, SubstateCombinationSet substateCombinationSet)`	Takes the state, the sub-state combination set, and gets the admissible sub-states.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.7 Sitecore.Ecommerce.Merchant.OrderManagement.Extensions

2.7.1 MerchantExtensions

This is a static class that defines the merchant extensions.



The only method of the `MerchantExtensions` class is:

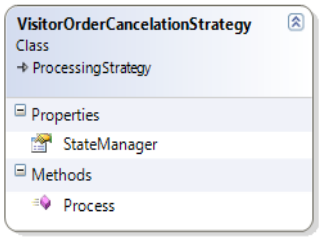
Method	Description
`ReservationTicket GetReservationTicket( this Order order)`	Takes the order, and creates the reservation tickets from the order.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.8 Sitecore.Ecommerce.Visitor.OrderManagement

2.8.1 VisitorOrderCancellationStrategy

This class implements the `ProcessingStrategy` class and defines the strategy for cancelling the visitor order.



The only property of the `VisitorOrderProcessingStrategy` class is:

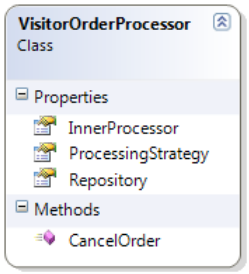
Property	Description
`StateManager<State> StateManager`	Gets or sets the state manager.

The only method of the `VisitorOrderProcessingStrategy` class is:

Method	Description
`void Process( Order order)`	Takes the order and processes it.

2.8.2 VisitorOrderProcessor

This class defines the processor of the visitor order.



The properties of the `VisitorOrderProcessor` class are:

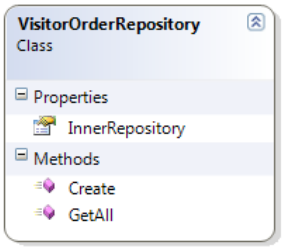
Property	Description
`OrderProcessor InnerProcessor`	Gets or sets the inner processor.
`ProcessingStrategy ProcessingStrategy`	Gets or sets the processing strategy.
`Repository<Order> Repository`	Gets or sets the repository.

The only method of the `VisitorOrderProcessor` class is:

Method	Property
`virtual void CancelOrder( Order order)`	Takes the order, and cancels it.

2.8.3 VisitorOrderRepository

This class defines the visitor order repository.



The only property of the `VisitorOrderRepository` class is:

Property	Description
`Repository<Order> InnerRepository`	Gets or sets the inner order repository.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

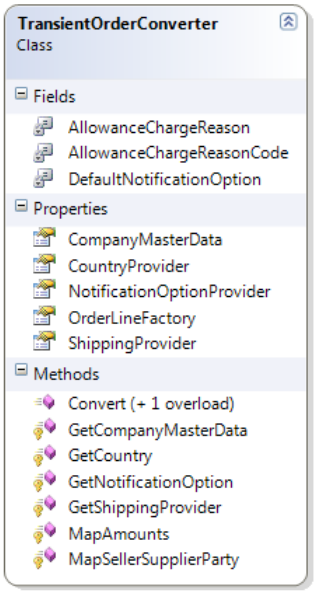
The methods of the `VisitorOrderRepository` class are:

Method	Description
`void Create(` `  Order order)`	Takes the order, and creates it.

`IQueryable<Order> GetAll(`
`  Expression<Func<Order, bool>>`
`  expression)`
	Takes the expression, and gets all the orders.

2.8.4 TransientOrderConverter

This class defines the transient order converter.



The properties of the `TransientOrderConverter` class are:

Property	Description
`CompanyMasterData CompanyMasterData`	Gets or sets the company master data.
`IEntityProvider<Country>` `CountryProvider`	Gets or sets the country provider.
`IEntityProvider<NotificationOption>` `NotificationOptionProvider`	Gets or sets the notification option provider.
`OrderLineFactory OrderLineFactory`	Gets or sets the order line factory.
`IEntityProvider<ShippingProvider>` `ShippingProvider`	Gets or sets the shipping provider.

The only public method of the `TransientOrderConverter` class is:

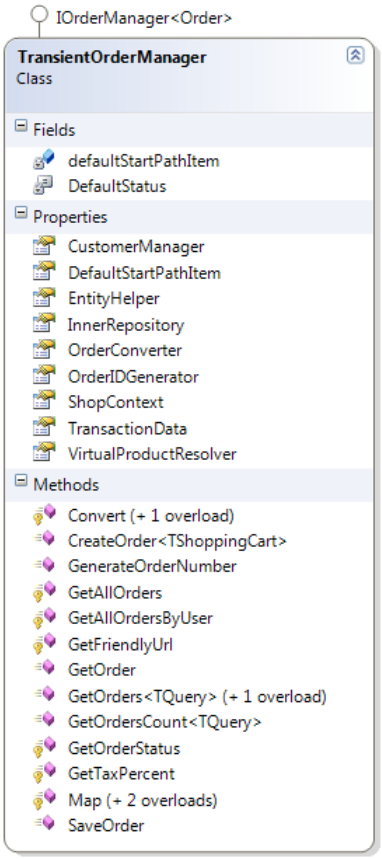
Method	Description

`Order Convert(`
`  DomainModel.Orders.Order`
`  source)`
	Converts the specified source.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.8.5 TransientOrderManager

This class implements the `IOrderManager` interface and defines the transient order manager.



The properties of the `TransientOrderManager` class are:

Property	Description
`ICustomerManager<CustomerInfo> CustomerManager`	Gets or sets the customer manager.
`Sitecore.Data.Items.Item DefaultStartPathItem`	Gets or sets the default start path item.
`EntityHelper EntityHelper`	Gets or sets the entity helper instance.
`VisitorOrderRepository InnerRepository`	Gets or sets the inner repository.
`TransientOrderConverter OrderConverter`	Gets or sets the order converter.
`OrderIDGenerator OrderIDGenerator`	Gets or sets the order number generator.
`ShopContext ShopContext`	Gets or sets the shop context.
`ITransactionData TransactionData`	Gets or sets the transaction data.
`VirtualProductResolver VirtualProductResolver`	Gets or sets the virtual product resolver.

The public methods of the `TransientOrderManager` class are:

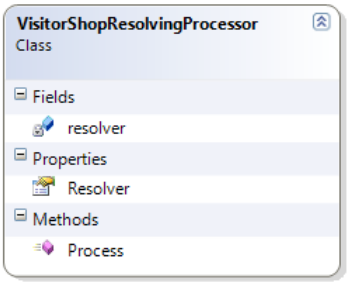
Method	Description
`OldOrder CreateOrder<TShoppingCart>( TShoppingCart shoppingCart) where TShoppingCart : ShoppingCart`	Takes the shopping cart and creates the order.
`string GenerateOrderNumber()`	Generates the order number.
`virtual OldOrder GetOrder( string orderNumber)`	Takes the order number gets the order.
`virtual IEnumerable<OldOrder> GetOrders<TQuery>( TQuery query)`	Takes the query and gets the orders.
`virtual IEnumerable<OldOrder> GetOrders<TQuery>( TQuery query, int pageIndex, int pageSize)`	Takes the query, the page index and the page size and gets the orders.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.9 Sitecore.Ecommerce.Visitor.Pipelines.HttpRequest

2.9.1 VisitorShopResolvingProcessor

This class defines the resolving processor of the visitor shop.



The only property of the `VisitorShopResolvingProcessor` class is:

Property	Description
`VisitorShopResolver Resolver`	Gets or sets the resolver.

The only method of the `VisitorShopResolvingProcessor` class is:

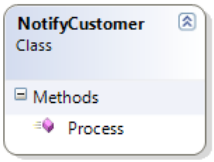
Method	Description
`void Process( PipelineArgs args)`	Runs the processor.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.10 Sitecore.Ecommerce.Visitor.Pipelines.OrderedCreated

2.10.1 NotifyCustomer

This class defines the process of notifying the customer.



The only method of the `NotifyCustromer` class is:

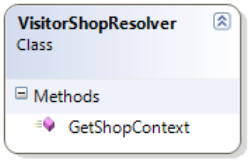
Method	Description
`void Process( PipelineArgs args)`	Runs the processor.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.11 Sitecore.Ecommerce.Visitor.Sites

2.11.1 VisitorShopResolver

This class defines the resolver of the visitor shop.



The only method of the `VisitorShopResolver` class is:

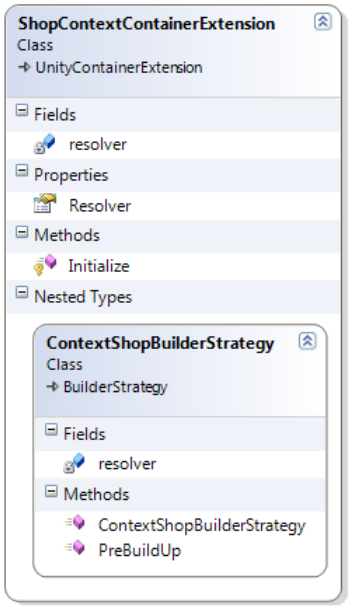
Method	Description
`ShopContext GetShopContext( SiteContext siteContext)`	Gets the shop context.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.

2.12 Sitecore.Ecommerce.Visitor.Unity

2.12.1 ShopContextContainerExtension

This class defines the extension of the site context container.



The only property of the `ShopContextContainerExtension` is:

Method	Description
`VisitorShopResolver Resolver`	Gets or sets the resolver.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.