

# Sitecore E-Commerce Services 2.2 on CMS 7.0 or Later

# Order Manager API Reference Guide

A developer's guide to the Order Manager API.



### **Table of Contents**

Chapter 1	Introduction	
	Order Manager Modules	
2.1 Si	tecore.Ecommerce.Core.Common	. 6
2.1.1	Address	
2.1.2	Amount	
2.1.3	Communication	
2.1.4	Contact	
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 Si	tecore.Ecommerce.Core.OrderManagement	14
2.2.1	CoreOrderManager	
2.2.2	StateManager	
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	
2.2.9	ILoggingEnabled	16
2.2.10	OrderConfirmation	
2.2.11	OrderLineFactory	
2.2.12	OrderProcessingStrategy	
2.2.13	OrderProcessor	
2.2.14	PlainMessageConfirmationBuilder	
2.2.15	ProcessingStrategy	
2.2.16	CoreOrderStateConfiguration	19
2.2.17	MerchantOrderProcessor	
	tecore.Ecommerce.Core.OrderManagement.Orders	
2.3.1	AllowanceCharge	
2.3.2	CustomerParty	
2.3.3	Delivery	
2.3.4	Despatch	
2.3.5	Item	
2.3.6	LineItem	
2.3.7	MonetaryTotal	
2.3.8	Order	
2.3.9	OrderedShipment	
2.3.10	OrderLine	
2.3.11	PaymentMeans	
2.3.12	Price	
2.3.13	ReservationTicket	
2.3.14	Shipment	
2.3.15	State	
2.3.16	Substate	
2.3.17	SupplierParty	
2.3.18	TaxCategory	
		-



2.3.19	TaxScheme	35
2.3.20	TaxSubTotal	36
2.3.21	TaxTotal	36
2.4 Si	itecore.Ecommerce.Core.OrderManagement.Extensions	37
2.4.1	CoreExtensions	37
2.5 Si	itecore.Ecommerce.Merchant.Pipelines.HttpRequest	38
2.5.1	MerchantShopResolver	
2.6 Si	itecore.Ecommerce.Merchant.OrderManagement	39
2.6.1	AddNewOrderLineProcessingStrategy	
2.6.2	CancelReservationProcessingStrategy	
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	
2.6.9	MerchantOrderManager	
2.6.10	MerchantOrderStateConfiguration	44
2.6.11	OrderFieldProcessingStrategy	
2.6.12	OrderLineFactoryImpl	
2.6.13	OrderLineStateTransitionStrategy	
2.6.14	OrderLineProcessingStrategy	
2.6.15	OrderProcessingStrategyResolver	
2.6.16	OrderProcessingStrategyResolverImpl	
2.6.17	OrderStateProcessingStrategy	
2.6.18	PackOrderProcessingStrategy	
2.6.19	RemoveOrderLineProcessingStrategy	
2.6.20	SendConfirmationProcessingStrategy	
2.6.21	ShipOrderProcessingStrategy	
2.6.22	StateValidator	51
	itecore.Ecommerce.Merchant.OrderManagement.Extensions	
2.7.1	MerchantExtensions	
	itecore.Ecommerce.Visitor.OrderManagement	
2.8.1	VisitorOrderCancellationStrategy	
2.8.2	VisitorOrderProcessor	
2.8.3	VisitorOrderRepository	
2.8.4	TransientOrderConverter	
2.8.5	TransientOrderManager	
	itecore.Ecommerce.Visitor.Pipelines.HttpRequest	
2.9.1	VisitorShopResolvingProcessor	56
2.10 Si	itecore.Ecommerce.Visitor.Pipelines.OrderedCreated	57
2.10.1	NotifyCustomer	57
	itecore.Écommerce.Visitor.Sites	
2.11.1	VisitorShopResolver	
	itecore.Ecommerce.Visitor.Unity	
2.12.1	ShopContextContainerExtension	59



# **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.



# **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



#### 2.1 Sitecore.Ecommerce.Core.Common

#### 2.1.1 **Address**

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

♀ IEntity	The properties of the Address class	s are:
Address 🛞	Property	Description
Properties	string AdditionalStreetName	Gets or sets the additional text to further specify the street name.
<ul> <li>AdditionalStreetName</li> <li>AddressFormatCode</li> <li>AddressLine</li> </ul>	string AddressFormatCode	Gets or sets the code of the address format.
AddressTypeCode Alias	string AddressLine	Gets or sets the address line.
BuildingName     BuildingNumber     GityName     CitySubdivisionName	string AddressTypeCode	Gets or sets the address type code to specify whether it is a home or a business address.
Country CountrySubentity CountrySubentityCode	long Alias	Gets the address ID that is used in the database.
Department District	string BuildingName	Gets or sets the building name.
Floor D	string CityName	Gets or sets the name of the city, town or village.
InhouseMail I MarkAttention I MarkCare 안 PlotIdentification	string CitySubdivisionName	Get or sets the city subdivision name. The subdivision can be the district or the borough.
PostalZone PostBox	string Country	Gets or sets the country.
Region Room StreetName	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

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 © 2001-2014 Sitecore. All rights reserved.

string InhouseMail

string MarkAttention

registered addresses.

or the department in the

address.

Gets or sets the location of the inhouse mail as part of the address. Gets or sets the name of the person

organization to whom the incoming mail is marked, for example, for the attention of, FAO or ATTN for this



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, <i>care of</i> or <i>C/O</i> .
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.

Amount	The properties of the Amount class:	
Class	Property	Description
□ Fields	string CurrencyID	Gets or sets the ID of the currency.
♂ currencyID √ value	decimal Value	Gets or sets the value of the amount.
Properties CurrencyID Value Methods	The methods of the Amour Method	t class are:
<ul> <li>Add</li> <li>Amount (+ 1 overload)</li> <li>Divide</li> <li>Multiply</li> <li>operator -</li> </ul>	Amount Add( Amount amount1, Amount amount2)	Adds two amounts of the same currencyID.
<ul> <li>operator * (+ 1 overload)</li> <li>operator /</li> </ul>	Amount()	Initializes a new instance of the Amount class.
<ul> <li>● operator +</li> <li>● Subtract</li> </ul>	Amount( decimal value, string currency]	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.

The properties of the Amount class:



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.

IEntity	The properties of the Communication class:	
Communication 🔊	Property	Description
Class ■ Properties Mias Channel ChannelCode Walue	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.

#### The properties of the Co unication class.

#### 2.1.4 Contact

\_

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

♀ IEntity	The properties of the Contract of	class are:
Contact 🛞	Property	Description
Properties	long Alias	Gets the contact's ID that is used in the database.
<ul> <li>Alias</li> <li>ElectronicMail</li> </ul>	string ElectronicMail	Gets or sets the email of the contact.
<ul> <li>ID</li> <li>Name</li> <li>Note</li> <li>OtherCommunications</li> </ul>	string ID	Gets or sets the identifier of the contact. It is a personal reference number such as "7687687".
🗃 Telefax 🗃 Telephone	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 <communicati on&gt; OtherCommunications</communicati 	Gets or sets the information about other communication means.

<sup>-----</sup>o of th . .



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

#### **IConfigurable** 2.1.5

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

IConfigurable (S) Interface	The properties of the IConfigurable in Property	terface are: Description
Properties ConfigurationData ConnectionStringOrName ProviderName ShopContext	Dictionary <string, object=""> ConfigurationData</string,>	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.

IEntity 😒	The only property of the IEntity interface is:	
Interface	Property	Description
Properties	long Alias	Gets the IEntity's ID that is used in the
🚰 Alias		database.

#### 2.1.7 Location

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

○ IEntity	P IEntity The properties of the Location class are:	
Location 🛞	Property	Description
Class	Address Address	Gets or sets information about the address of the location.
<ul> <li>Address</li> <li>Alias</li> <li>CountrySubentity</li> </ul>	long Alias	Gets the location's ID that is used in the database.
<ul> <li>CountrySubEntityCode</li> <li>Description</li> <li>ID</li> <li>ValidityPeriod</li> </ul>	string CountrySubentity	Gets or sets text about the country sub-entity. A sub-entity can be a county or a state.



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.

Measure 🖄	The properties of the Mea	properties of the Measure class are:	
Class	Property	Description	
Properties     Content	decimal Content	Gets or sets the decimal value such as 10000.25	
UnitCode		Note You must not use more than four decimals for the value of the Content property.	
	<pre>string UnitCode</pre>	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			
Party 🛞 Class	Property	Description	
Properties Alias	long Alias	Gets the party's ID that is used in the database.	
Contact	Contact Contact	Gets or sets the contact of the party.	
LanguageCode     LogoReferenceID     PartyIdentification     PartyLegalEntity	string EndpointID	Gets or sets the end point for the routing service such as the EAN and the GLN.	
PartyName PartyTaxScheme Person	string LanguageCode	Gets or sets the code of the party language.	
PhysicalLocation PostalAddress WebsiteURI	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 © 2001-2014 Sitecore. All rights reserved.



Property	Description
string PartyName	Gets or sets the name of the party.
	<b>Note</b> You must use this name when no other party identification is used.
ICollection <partytaxscheme> 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

○ IEntity PartyLegalEntity Class

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

♀ IEntity	The properties of the PartyLegalEntity class are:	
PartyLegalEntity	Property	Description
□ Properties	long Alias	Gets the ID of the party legal entity that is used in the database.
1000		is used in the database.
<ul> <li>Alias</li> <li>CompanyID</li> <li>RegistrationAddress</li> <li>RegistrationName</li> </ul>	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.

тι rtic o of th 



### 2.1.11 PartyTaxScheme

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

♀ IEntity The properties of the PartyTaxScheme classical statement of the PartyTaxScheme classical statement of the properties of the partyTaxScheme classical statement of the partyTaxSche		eme class are:
PartyTaxScheme	Property	Description
Class Properties	long Alias	Gets the ID of the party tax scheme that is used in the database.
Alias CompanyID ExemptionReason ExemptionReasonCode RegistrationAddress RegistrationName	string CompanyID	Gets or sets the identifier of the company. This identifier is assigned for tax purposes to the party by the taxation authority.
TaxScheme	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.

Period The properties of the Period class are:		od class are:
Class	Property	Description
Properties	string Description	Gets or sets the description for the period.
😭 Description 🚰 EndDate	DateTime EndDate	Gets or sets the end date of the period.
EndTime StartDate	TimeSpan EndTime	Gets or sets the end time of the period.
😭 StartTime	DateTime StartDate	Gets or sets the start date of the period.
	TimeSpan StartTime	Gets or sets the start time of the period.



#### 2.1.13 Person

This class contains information about the person.

Person (R) Class	The properties of the Person class are: <b>Property</b>	Description
Properties FamilyName	string FamilyName	Gets or sets the surname or the family name of the person.
<ul> <li>FirstName</li> <li>JobTitle</li> <li>MiddleName</li> </ul>	string FirstName	Gets or sets the first name of the person.
<ul> <li>MameSuffix</li> <li>OrganizationDepartment</li> <li>Title</li> </ul>	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  ${\tt IEntity}$  interface and contains information about the context of the webshop.

The properties the ShopContext class are:

IEntity	
ShopContext Class	8
Properties	
🚰 Alias	
ContextId	
	_

	Property	Description
	long Alias	Gets the ID of the shop context that is used in the database.
l	string ContextId	Gets or sets the ID of the shop context. It is a

webshop.

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



### 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</order>	Gets or sets the repository.
StateManager <state></state>	Gets or sets the state manager.
StateManager	
	Property Repository <order> Repository</order>

#### 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.

StateManager <tstate></tstate>	The only property of the StateManager class is:	
Generic Abstract Class	Property	Description
Properties	StateProvider <tstate></tstate>	Gets or sets the state
🚰 StateProvider	StateProvider	provider.
Methods		
ö GetFollowingStates		
ö		
👼 GetStates		
👼 🖗 IsValid		

#### 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.



### 2.2.4 SubstateCombinationSet

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

SubstateCombinationSet Sealed Class	The only property of the SubstateCombinationSet class is:	
Sealed Class	Property	Description
Properties	IEnumerable <idictionary<string,< td=""><td>Gets or sets the sub-state</td></idictionary<string,<>	Gets or sets the sub-state
SubstateCombinations	bool>> SubstateCombinations	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.

ChangeOrderLineQuantityCommand	The properties of the ChangeOrderLineQua:	ntityCommand class are:
Class -> Command	Property	Description
Properties	OrderLine OrderLine	Gets or sets the order line.
<ul> <li>☐ OrderLine</li> <li>☐ Quantity</li> <li>☐ Methods</li> </ul>	decimal Quantity	Gets or sets the quantity. It is the value that is changed
<ul> <li>ChangeOrderLineQuantityCommand (+ 1 overload)</li> <li>GetLoggingEntry</li> <li>IsApplicable</li> <li>PerformAction</li> </ul>		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.
<pre>LoggingEntry GetLoggingEntry()</pre>	Gets the logging entry.
<pre>bool IsApplicable()</pre>	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.

Command	The methods of the Command class are:	
Abstract Class	Method	Description
Methods	void Do()	Processes the order line.
<ul> <li>● Do</li> <li>● IsApplicable</li> <li>● PerformAction</li> <li>● ProcessInvalidCommandState</li> </ul>	<pre>bool IsApplicable()</pre>	Specifies whether the command is applicable to the specified order line. It is true if the command is applicable.



#### ConfirmationMessageBuilder 2.2.7

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

ConfirmationMessageBuilder	The only property of the ConfirmationMessageBuilder class is:		
Abstract Class → MessageBuilder	Property	Description	
· · · · · · · · · · · · · · · · · · ·	virtual Order	Gets or sets the order.	
Properties	Order		
😭 Order			
Methods			
ళ GetParameters			

#### 2.2.8 CoreOrderMangerImpl

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

Class	D <b>rder Manager Impl</b> e Order Manager	8
Met	thods	
· · ·	DeleteOrder	
	DeleteOrdersByQuery	
	GetOrderByID	
	GetOrdersByQuery	
	LogOrderCreated	
	LogOrderDeleted	
	LogOrderSaved	
	OnOrderCreated	
	OnOrderCreating	
	OnOrderDeleted	
	OnOrderDeleting	
	OnOrderSaved	
	OnOrderSaving	
_ 🧔 🖗	SaveOrder	J

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.

ILoggingEnabled (Interface is:		ed interface is:
Interface	Method	Description
Methods	LoggingEntry	Gets the logging entry.
=♥ GetLoggingEntry	GetLoggingEntry()	



### 2.2.10 OrderConfirmation

This class contains information about the confirmation of an order.

OrderConfirmation (R) Class	The properti <b>Property</b>	<b>es of the</b> OrderConfirm	nation <b>class are</b> : Description
■ Fields mailService	ConfirmationMessageBuilder ConfirmationMessageBuilder		Gets or sets the confirmation message builder.
messageBuilder Properties	MailServi	ce MailService	Gets or sets the mail service.
ConfirmationMessageBuilder MailService Methods OrderConfirmation (+ 1 overload) Send	The methods of the OrderConfirmation class are:		
Method		Description	
OrderConfirmation()		Initializes a new instanc	e of the OrderConfirmation
OrderConfirmation( MailService mailService, ConfirmationMessageBuilder messageBuilder)		Takes the mail service, the message builder, and initializes a new instance of the OrderConfirmation class.	
<pre>void Send()</pre>		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.

OrderLineFactory	The methods of the OrderLineFactory class are:		
Abstract Class	Method	Description	
Methods  CreateLineItemFromOrder  CreateOrderLineFromOrder	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.	



### 2.2.12 OrderProcessingStrategy

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

OrderProcessingStrategy 🛞	The methods of the OrderProcessi	ngStrategy <b>class are:</b>
Abstract Class	Method	Description
<ul> <li>Fields</li> <li>Empty</li> <li>FailedResult</li> <li>SuccessfulResult</li> </ul>	IList <logentry> GetAdditionalLogging EntriesForSuccess( Order order)</logentry>	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 <i>Capture</i>
Methods     GetAdditionalLoggingEntriesForSuccess     GetLoggingEntryForFail     GetLoggingEntryForSuccess     ProcessOrder		
Nested Types  EmptyStrategy  Class		Order. At the same time the substate of the order is also changed.
→ OrderProcessingStrategy	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.
	<pre>string ProcessOrder(     Order order,     IDictionary<string, object=""> parameters)</string,></pre>	Takes the order, the parameters, and processes the order.

#### 2.2.13 OrderProcessor

This class is used to process an order.

OrderProcessor Class	8
Properties	
🚰 Order	
Methods	
=∳ Process	

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
<pre>void Process(     ProcessingStrategy     processingStrategy)</pre>	Takes the processing strategy and processes the order.



### 2.2.14 PlainMessageConfirmationBuilder

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

PlainMessageConfirmationBuilder	The only method in the PlainMessageConfirmationBuilder class is:	
→ ConfirmationMessageBuilder	Method	Description
Fields	MailMessage GetMessage(	Takes the template name and
<ul> <li>MessageBody</li> <li>MessageSubject</li> </ul>	string templateName)	gets the mail message.
Methods		
=• GetMessage		

### 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:

Abstract Class     Method     Description            Methods           void Process (           Takes the order and runs the processor.	ProcessingStrategy	The only method in the PocessingStrategy class is:	
		Method Description	
	Methods	void Process(	Takes the order and runs the processor.
Process Order Order)	• Process	Order order)	

#### 2.2.16 CoreOrderStateConfiguration

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

CoreOrderStateConfiguration	The only property of the CoreOr class is:	derStateConfiguration
∃ Fields	Property	Description
AbbreviationFieldName     CodeFieldName     FollowingStateFieldName	ShopContext ContextShop	Gets or sets the shop context.
<ul> <li>FollowingStateLinkTemplateId</li> <li>FollowingStatusesContainerName</li> <li>NameFieldName</li> <li>SubstateCombinationsContainer</li> <li>SubstateContainerName</li> <li>SubstatesFieldName</li> </ul>	The methods of this class are eit because they are only used inter the methods in the correspondin	nally. However, you can use g class:
■ Properties	MerchantOrderStateConfig	uration instead of this class.
🚰 ContextShop		
<sup>⊒</sup> Methods		
<ul> <li>CreateStateFromItem</li> <li>CreateSubstateCombinationSetFromItem</li> <li>GeStateItem</li> <li>GetAll</li> <li>GetFollowingStates</li> <li>GetPreviousStates</li> <li>GetRootItem</li> <li>GetStates</li> <li>GetStates</li> <li>GetSubstateCombinations</li> <li>IsSubstateCombinationApplicable</li> <li>IsValid</li> <li>Map (+ 1 overload)</li> </ul>		



#### 2.2.17 MerchantOrderProcessor

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

MerchantOrderProcessor  Sealed Class	The methods of the MerchantOrderProce	essor <b>class are:</b>
□ Fields	Method	Description
<ul> <li>ologger</li> <li>orderProcessingStrategy</li> <li>Methods</li> <li>MerchantOrderProcessor</li> <li>ProcessOrder</li> </ul>	MerchantOrderProcessor( OrderProcessingStrategy orderProcessingStrategy)	Takes the order processing strategy and initializes a new instance of the MerchantOrderPro cessor class.
	<pre>string ProcessOrder(     Order order,     IDictionary<string, object="">     parameters)</string,></pre>	Takes the order, the parameters, and processes the order.



### 2.3 Sitecore.Ecommerce.Core.OrderManagement.Orders

#### 2.3.1 AllowanceCharge

This class implements the  ${\tt IEntity}$  interface and contains the information about the charge or the discount.

	-	
Allow Class	vanceCharge 🛞	•
🗏 Pro	perties	
1	Alias	ŀ
	AllowanceChargeReason	
<u>e</u>	AllowanceChargeReasonCode	
<u>r</u>	Amount	
1	BaseAmount	
1	ChargeIndicator	
<b>1</b>	ID	
1	MultiplierFactorNumeric	
<b>1</b>	PrepaidIndicator	
1	SequenceNumeric	
<u>e</u>	TaxCategory	
1	TaxTotal	

**Q** IEntity

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.

The properties of the AllowanceCharge class are:



### 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.

Q IEntity	The properties of the CustomerParty class are:	
CustomerParty 🛞	Property	Description
Class	Contact AccountingContact	Gets or sets the information about the accounting contact of the buyer.
AccountingContact Alias BuyerContact	long Alias	Gets the ID of the customer party that is used in the database.
<ul> <li>CustomerAssignedAccountID</li> <li>DeliveryContact</li> <li>Party</li> <li>SupplierAssignedAccountID</li> </ul>	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.

♀ IEntity	The properties of the Delivery class are:	
Delivery	Property	Description
Class  Properties	long Alias	Gets the ID of the delivery that is used in the database.
Properties     Alias     Aliar     AlternativeDeliveryLocation     DeliveryAdress	Location AlternativeDeliveryLocation	Gets or sets the alternative delivery location.
<ul> <li>DeliveryLocation</li> <li>DeliveryParty</li> <li>Despatch</li> </ul>	Address DeliveryAdress	Gets or sets the information about the delivery address.
<ul> <li>LatestDeliveryDate</li> <li>LatestDeliveryTime</li> <li>MaximumQuantity</li> <li>MinimumQuantity</li> <li>Quantity</li> <li>RequestedDeliveryPeriod</li> <li>TrackingID</li> </ul>	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.



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.

♀ IEntity	The properties of the Despatch class are:	
Despatch 🛞	Property	Description
Class Properties	DateTime ActualDespatchDate	Gets or sets the actual dispatch date.
<ul> <li>ActualDespatchDate</li> <li>ActualDespatchTime</li> <li>Alias</li> </ul>	TimeSpan ActualDespatchTime	Gets or sets the actual dispatch time.
<ul> <li>Contact</li> <li>DespatchAddress</li> <li>DespatchParty</li> </ul>	long Alias	Gets the ID of the dispatch that is used in the database.
<ul> <li>EstimatedDespatchDate</li> <li>EstimatedDespatchTime</li> <li>ID</li> </ul>	Contact Contact	Gets or sets the information about the contact.
RequestedDespatchDate	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.



#### 2.3.5 ltem

This class contains information about the item.

Item 🛞	The properties of the Item class are: Property	Description
Properties  AdditionalInformation  BrandName Code Description	string AdditionalInformation	Gets or sets the additional information about the item, such as the URL of the relevant web page.
Keyword ModelName	string BrandName	Gets or sets the brand name of the item.
PackQuantity	string Code	Gets or sets the code of the item.
PackSizeNumeric	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 Lineltem

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

LineItem (R)	The properties of the LineItem class are: Property	Description
□ Fields @ subLines	long Alias	Gets the ID of the order line item that is used in the database.
Properties Alias Delivery ID	ICollection <delivery> Delivery</delivery>	Gets or sets the information about the request for delivery.
Item         Intervention         In	string ID	Gets or sets the address identifier that is used in a third party database.
MinimumQuantity Note OrderedShipment OrderID	Item Item	Gets or sets the information about the item on the order line.
<ul> <li>OrderLine</li> <li>ParentLine</li> <li>PartialDeliveryIndicator</li> <li>Price</li> <li>Quantity</li> <li>SubLineItem</li> <li>Substates</li> <li>TotalTaxAmount</li> </ul>	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 © 2001-2014 Sitecore. All rights reserved.



Property	Description
<pre>string LineStatusCode</pre>	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</lineitem>	Gets the sub-line item.
ICollection <substate> Substates</substate>	Gets or sets the sub-states.
Amount TotalTaxAmount	Gets or sets the tax total amount for the order line.



#### 2.3.7 **MonetaryTotal**

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

O IEntity	The properties of the MonetaryTotal	class are:
MonetaryTotal	Property	Description
⊂lass □ Fields	long Alias	Gets the ID of the monetary total that is used in the database.
prepaidAmount     Properties     Alias	Amount AllowanceTotalAmount	Gets the total amount of all the allowances.
AllowanceTotalAmount ChargeTotalAmount LineExtensionAmount	Amount ChargeTotalAmount	Gets the total amount of all the charges.
<ul> <li>Order</li> <li>PayableAmount</li> <li>PayableRoundingAmount</li> <li>PrepaidAmount</li> <li>TaxExclusiveAmount</li> </ul>	Amount LineExtensionAmount	Gets the line extension amount without the tax and settlement discounts, but including any applicable rounding amount.
TaxInclusiveAmount	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.

... . . . . . . . . 7 -

#### 2.3.8 Order

The Order class is used to exchange the electronic orders and to process the coherent order-toinvoice 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. •



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.

Q IEntity	The properties of the Order class are:	
Order 🛞	Property	Description
Class Fields orderLines solutions Properties Class AccountingCustomerParty	CustomerParty AccountingCustomerParty	Gets or sets the information about the accounting customer party. This is the party to which the invoice is sent.
<ul> <li>Alias</li> <li>AllowanceCharge</li> <li>AnticipatedMonetaryTotal</li> <li>BuyerCustomerParty</li> </ul>	long Alias	Gets the order's ID that is used in the database.
<ul> <li>Delivery</li> <li>DestinationCountryCode</li> <li>FreightForwarderParty</li> <li>IssueDate</li> <li>IssueTime</li> <li>Note</li> <li>OrderId</li> </ul>	ICollection <allowancecharge> AllowanceCharge</allowancecharge>	Gets or sets the information about the allowances and charges that apply to the order as a whole.
<ul> <li>OrderLines</li> <li>PaymentMeans</li> <li>PricingCurrencyCode</li> <li>ReservationTicket</li> <li>SellerSupplierParty</li> <li>ShopContext</li> <li>State</li> <li>TaxCurrencyCode</li> </ul>	MonetaryTotal AnticipatedMonetaryTotal	Gets or sets the net monetary total, excluding allowances and taxes, as expected by the buyer.
🚰 TaxTotal	CustomerParty BuyerCustomerParty	Gets or sets the
Methods  Order		information about the buyer.
	ICollection <delivery> Delivery</delivery>	Gets or sets the

incrospaceanone cary rocar	excluding allowances and taxes, as expected by the buyer.
CustomerParty BuyerCustomerParty	Gets or sets the information about the buyer.
ICollection <delivery> Delivery</delivery>	Gets or sets the information about delivery.
string DestinationCountryCode	Gets or sets the destination country code.
	<b>Note</b> This code is used for customs purposes.
ICollection <party> FreightForwarderParty</party>	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.



Property	Description
string OrderId	Gets or sets the order identifier that the buyer specifies.
ICollection <orderline> OrderLines</orderline>	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.

#### OrderedShipment 2.3.9

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

♀ IEntity	The properties of the OrderedShipment class are:		
OrderedShipment 🖄	Property	Description	
Class	long Alias	Gets the ID of the ordered shipment that is used in	
Properties		the database.	
😭 Alias 🚰 Shipment	Shipment Shipment	Gets or sets the shipment information.	
snipment		, , , , , , , , , , , , , , , , , , ,	

The properties of the OrderedShipment class are:



#### 2.3.10 OrderLine

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

Q IEntity	The properties of the OrderLine class are:	
OrderLine 🛞	Property	Description
Properties	long Alias	Gets the ID of the order line that is used in the database.
Allas         BuyerProposedSubstituteLineItem         LineItem         Note         Order         SellerProposedSubstituteLineItem         SellerSubstituteLineItem	ICollection <lin eItem&gt; BuyerProposedSu bstituteLineIte m</lin 	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.
		<b>Note</b> This property can contain notes that are not explicitly stated in any other class.
	Order Order	Gets or sets the order.
	ICollection <lin eItem&gt; SellerProposedS ubstituteLineIt em</lin 	Gets or sets the possible alternative items that the seller proposes.
	ICollection <lin eItem&gt; SellerSubstitut</lin 	Gets or sets the actual alternative items that the seller gives.

#### 2.3.11 PaymentMeans

This class implements the  ${\tt IEntity}$  interface and contains information about the order payment means.

PaymentMeans	The properties of the PaymentMeans class are: Property Description	
□ Properties	long Alias	Gets the ID of the payment means that is used in the database.
Alias ID PaymentChannelCode PaymentDueDate PaymentID PaymentMeansCode	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 © 2001-2014 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.

♀ IEntity	The properties of the Price class are:	
Price	Property	Description
⊂ Fields	long Alias	Gets the ID of the price that is used in the database.
amount a∲ baseQuantity a∲ orderableUnitFactorRate	IEnumerable <allowancecharge> AllowanceCharge</allowancecharge>	Gets or sets the allowance charge.
Properties     Alias     AliovanceCharge     BaseQuantity	decimal baseQuantity	Gets or sets the base quantity. It is the actual quantity to which the Price applies
OrderableUnitFactorRate PriceAmount	Amount PriceAmount	Gets or sets the price value.
<ul> <li>PriceChangeReason</li> <li>PriceType</li> <li>PriceTypeCode</li> </ul>	string PriceChangeReason	Gets or sets the price change reason.
Methods	string PriceType	Gets or sets the price type.
Price (+ 2 overloads)	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

**Q** IEntity

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

The properties of the ReservationTicket class are:

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



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.

Shipment 🛞	The properties of the Shipment Property	class are: Description
lass		Description
Properties  Alias	long Alias	Gets the ID of the shipment that is used in the database.
Image         DeclaredCustomsValueAmount         DeclaredForCarriageValueAmount         DeclaredStatisticsValueAmount         DeclaredStatisticsValueAmount         DeclaredStatisticsValueAmount         Delivery         DeliveryInstructions         ExportCountry         FirstArrivalPortLocation         FreeOnBoardValueAmount         FreightAllowanceCharge         GrossVolumeMeasure         GrossVolumeMeasure         HandlingCode         HandlingInstructions         Information         InsuranceValueAmount         Exercise         NetWeightMeasure         NetNetWeightMeasure         NetVetWeightMeasure         NetVetOlumeMeasure         NetWeightMeasure         NetWeightMeasure         OriginAddress         ShippingPriorityLevelCode         SpecialInstructions         SplitConsignmentIndicator         TotalGoodsItemQuantity         TotalTransportHandlingUnitQuantity	Amount DeclaredCustomsValueAmo unt	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 DeclaredForCarriageValu eAmount	Gets or sets the amount of the declared for carriage value.
	Amount DeclaredStatisticsValue Amount	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 FirstArrivalPortLocatio	Gets or sets the first arrival location.
		<b>Note</b> 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 © 2001-2014 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.	
	<b>Note</b> 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.	
	<b>Note</b> 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.	
	<b>Note</b> 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.	



Property	Description
Address OriginAddress	Gets or sets the address of the place where the goods are produced.
	<b>Note</b> This address is used for the customs purposes such as customs tariff, quantitative restrictions, or any other measure related to the trade.
string ShippingPriorityLevelCo de	Gets or sets the code of the service priority.
string SpecialInstructions	Gets or sets the special instructions for the shipment.
bool SplitConsignmentIndicat or	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 TotalTransportHandlingU nitQuantity	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.

♀ IEntity	The properties of the State class are:	
State 🙈	Property	Description
Class	long Alias	Gets the ID of the state that is used in the database.
<ul> <li>Substates</li> <li>Properties</li> <li>Alias</li> </ul>	string Code	Get or sets the code that identifies the state.
Code	string Name	Gets or sets the name of the state.
Substates	ICollection <substate> Substates</substate>	Gets the sub-states. A state potentially contains a collection of substates.



#### 2.3.16 Substate

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

♀ IEntity	The properties of the Substate class are:	
Substate 🙈	Property	Description
Class	string Abbreviation	Gets or sets the abbreviation.
Abbreviation Active Alias	bool Active	Gets or sets a value that indicates whether this sub-state is active or not.
🚰 Code 🚰 Name	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.

The properties of the Substate class are

#### **SupplierParty** 2.3.17

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

♀ IEntity	The properties of the SupplierParty class are:	
SupplierParty 🛞	Property	Description
Properties     AccountingContact	Contact AccountingContact	Get or sets the information about the accounting contact of the supplier.
AdditionalAccountID Alias CustomerAssignedAccountID OrderContact Party ShippingContact	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.



#### TaxCategory 2.3.18

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

♀ IEntity	The properties of the TaxCategory class are:	
TaxCategory 🛞	Property	Description
Class  Properties  Alias	long Alias	Gets the ID of the tax category that is used in the database.
<ul> <li>BaseUnitMeasure</li> <li>ID</li> <li>Name</li> <li>Percent</li> <li>PerUnitAmount</li> <li>TaxExemptionReason</li> </ul>	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.
TaxExemptionReasonCode	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.
	<pre>string TaxExemptionReasonCode</pre>	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: 

TaxScheme 😒	The properties of the TaxScheme class are:	
Class	Property	Description
Properties CurrencyCode ID Name TaxTypeCode	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.
	<pre>string TaxTypeCode</pre>	Gets or sets the identification for the tax type.



#### 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.

Q IEntity	The Properties of TaxSubTotal cl	ass are:	
TaxSubTotal	Property	Description	
Class	long Alias	Gets the ID of the tax sub-total that is used in the database.	
Properties		that is used in the database.	
<ul> <li>Alias</li> <li>CalculationSequenceNumeric</li> <li>TaxableAmount</li> <li>TaxAmount</li> <li>TaxCategory</li> <li>TransactionCurrencyTaxAmount</li> </ul>	decimal CalculationSequenceNumeri c	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, CalculationSequenceNume ric 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 TransactionCurrencyTaxAmo unt	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.

IEntity	The properties of the TaxTotal class are:	
TaxTotal 🔊	Property	Description
Class	long Alias	Gets the ID of the tax total that is used in the database.
앱 Alias 앱 RoundingAmount 앱 TaxAmount 앱 TaxSubtotal	Amount RoundingAmount	Gets or sets the rounding amount for taxes.
		<b>Note</b> 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.
		<b>Note</b> This value is a calculated as the sum of the tax sub totals for each tax category within the tax type.
	ICollection <taxsubtotal> TaxSubtotal</taxsubtotal>	Gets or sets infromation about the tax subtotal of the order.

... of the ~

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 © 2001-2014 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.

CoreExtensions 🛞	The methods of the CoreExtension class are:	
Static Class	Method	
□ Fields	<pre>void CopyLineItemFrom(    LineItem acceptor,    LineItem denem)</pre>	Takes the acceptor line item, the donor line item, and copies the data partially
Methods	LineItem donor)	from one line item to another.
♀; CopyLineItemFrom ♀; IsEditable	bool IsEditable( Order order)	Takes the order and checks if the order is editable.



## 2.5 Sitecore.Ecommerce.Merchant.Pipelines.HttpRequest

#### 2.5.1 MerchantShopResolver

This class defines the merchant shop resolver.

Class       Property       Description            Properties           String MerchantSiteName           Gets or sets the name of the merchant         site.             MetchantSiteName           string ShopContextName           Gets or sets the name of the shop	MerchantShopResolver 🔕	The properties of the MerchantShopResolver class are:		
Image: MerchantSiteName       Site.         Image: ShopContextName       String         ShopContextName       Gets or sets the name of the shop		Property	Description	
string ShopContextName Cats or sets the name of the shop	MerchantSiteName	string MerchantSiteName		
Process Context.	□ Methods	string ShopContextName		

The only method of the MerchantShopResolver class is:

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



## 2.6 Sitecore.Ecommerce.Merchant.OrderManagement

#### 2.6.1 AddNewOrderLineProcessingStrategy

This class implements the <code>OrderProcessingStrategy</code> class and defines the strategy for adding a new order line to the existing order.

Class	ewOrderLineProcessingStrategy
🗏 Fiel	ds
5° 5°	additionalLoggingEntriesForSuccess availableStates deniedSubstates formattedMessage substateTransitions
⊟ Pro	perties
<b>1</b>	OrderLineFactory OrderManager ProductStockManager
⊟ Met	hods
	AddSubstateTransitionLogEntry CreateFormattedMessage GetAdditionalLoggingEntriesForSuccess GetLoggingEntryForFail GetLoggingEntryForSuccess ProcessOrder SetOrderStates

#### 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	
<pre>string ProcessOrder(     Order order,     IDictionary<string, object=""> parameters)</string,></pre>	Takes the order, the parameters and processes the order.	
LoggingEntry GetLoggingEntryForSucces s( 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.	
<pre>IList<loggingentry> GetAdditionalLoggingEntr iesForSuccess(         Order order)</loggingentry></pre>	Takes the order and gets the additional log entries for success.	



## 2.6.2 CancelReservationProcessingStrategy

This class implements the <code>OrderProcessingStrategy</code> class and defines the strategy for canceling the reservation of the money.

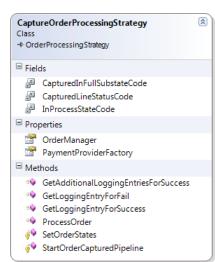
CancelReservationProcessingStrategy Class	The properties of the CancelReservationProcessingStrategy class are:		
OrderProcessingStrategy	Property	Description	
Properties OrderManager	MerchantOrderManager	Gets or sets the order	
PaymentProviderFactory	OrderManager	manager.	
□ Methods	PaymentProviderFactory	Gets or sets the payment	
<ul> <li>GetLoggingEntryForFail</li> <li>GetLoggingEntryForSuccess</li> <li>ProcessOrder</li> </ul>	PaymentProviderFactory	provider factory.	

The methods of the CancelReservationProcessingStrategy class are:

Method	Description	
<pre>string ProcessOrder(     Order order,     IDictionary<string, object=""> parameters)</string,></pre>	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 <code>OrderPrcessingStrategy</code> 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.



The methods of the CaptureOrderProcessingStrategy class are:

Method	Description
<pre>string ProcessOrder(     Order order, IDictionary<string, object=""> parameters)</string,></pre>	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.
<pre>IList<loggingentry> GetAdditionalLoggingEntriesForSuccess(</loggingentry></pre>	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 <code>OrderProcessingStrategy</code> class and defines the processing strategy of the allowance charge amount value.

ChangeAllowanceChargeAmountProcessingStrategy Class + OrderProcessingStrategy	The methods of the ChangeAllowanceChargeAmount Method	Processing <b>are</b> : <b>Description</b>
<ul> <li>Fields</li> <li>changedOrder</li> <li>initialOrder</li> <li>loggingEntries</li> <li>previousFieldValues</li> <li>Properties</li> <li>LoggingEntries</li> <li>PreviousFieldValues</li> <li>Methods</li> <li>ChangeAllowanceChargeAmountProcessingStrategy</li> <li>CheckOrderTotals</li> <li>EvaluateLoggingEntries</li> </ul>	ChangeAllowanceChargeAmoun tProcessingStrategy( Order inintialOrder, IDictionary <string, object&gt; previousFieldValues)</string, 	Initializes a new instance of the class.
	IList <logentry> GetAdditionalLogEntriesFor Success(Order order)</logentry>	Performs additional success processing.
CetAdditionalLogEntriesForSuccess  GetLogEntryForFail  GetLogEntryForFail  GetLogEntryForSuccess  ProcessOrder  RevertAllowanceChargesAmounts	LogEntry GetLogEntryForFail( Order order, params object[] parameters)	Performs additional failure processing.
	LogEntry GetLogEntryForSuccess( Order order)	Gets logging entries for success.
	<pre>string ProcessOrder(     Order order,     IDictionary<string, object=""> parameters)</string,></pre>	Processes the order.



## 2.6.5 ChangeOrderLineQuantityProcessingStrategy

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

ChangeOrderLineQuantityProcessingStrategy	The only public method of the ChangeOrderLineQuanityProcessingStrategy class is:	
□ Methods	Method	Description
ProcessOrder  VpdateTaxesForOrder	string ProcessOrder( Order order, IDictionary <string,< th=""><th>Processes the order.</th></string,<>	Processes the order.
	object> parameters)	

## 2.6.6 EditOrderLineProcessingStrategy

This class implements the <code>OrderProcessingStrategy</code> class and defines the processing strategy for editing the order line.

EditOrderLineProcessingStrategy (≧) Class → OrderProcessingStrategy	The properties of the EditOrde: class are: Property	rLineProcessingStrategy Description
□ Fields	IProductStockManager ProductStockManager	Gets or sets the product stock manager.
Properties LineItemToReplaceBy	MerchantOrderManager OrderManager	Gets or sets the order manager.
<ul> <li>OrderLineFactory</li> <li>OrderManager</li> <li>ProductStockManager</li> </ul>	OrderLineFactory OrderLineFactory	Gets or sets the order line factory.
Methods     OreateFormattedMessage     GetLoggingEntryForFail	LineItem LineItemToReplaceBy	Gets or sets the order line item to be added.
<ul> <li>GetLoggingEntryForSuccess</li> <li>ProcessOrder</li> </ul>		· · · · ·

The methods of the EditOrderLineProcessingStrategy class are:

Method	Description
<pre>string ProcessOrder(     Order order, IDictionary<string, object=""> parameters)</string,></pre>	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.



## 2.6.7 MerchantOrderCancelationStrategy

This class implements the <code>ProcessingStrategy</code> class and defines the strategy for cancelling the merchant order.

MerchantOrderCancelationStrategy (3)	The properties of the MerchantOrderCancelationStrategy class are:		
Class	Property	Description	
→ ProcessingStrategy	Core.Contracts.OrderManageme	Gets or sets the state	
□ Fields	nt.StateManager <state></state>	manager.	
₀IlowedRoleNames	StateManager		
Properties	User CurrentUser	Gets or sets the current	
AllowedRoleNames CurrentUser		user.	
<ul> <li>ReasonSubstateCode</li> <li>StateManager</li> </ul>	string ReasonSubstateCode	Gets or sets the reason sub- state code.	
Methods		State code.	
MerchantOrderCancelationStrategy     Process	<pre>virtual IEnumerable<string> AllowedRoleNames</string></pre>	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.

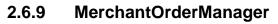
## 2.6.8 MerchantOrderProcessor

This class is sealed and defines the merchant order processor.



The methods of the MerchantOrderProcessor class are:

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



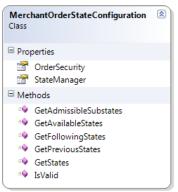
This class defines the merchant order manager.

MerchantOrderManager 😒	The properties of the MerchantOrderManager class are:         Property       Description		
□ Fields @ coreOrderManager	CoreOrderManager CoreOrderManager	Gets or sets the core order manager.	
<ul> <li>oldEntity</li> <li>Properties</li> <li>CoreOrderManager</li> </ul>	MerchantOrderProcessor OrderProcessor	Gets or sets the order processor.	
CoreOrderManagerNotNull CorderProcessor StrategyResolver	OrderProcessingStrategyResol <sup>®</sup> StrategyResolver	Gets or sets the strategy resolver.	
Methods     Methods     DeleteOrder     DeleteOrder     GetOrder     GetOrder     GetOrder	The methods of the MerchantOrder Method	Manager class are: Description	
ë♥ OnOrderSaved ë♥ OnOrderSaving ë♥ PerformOrderSaving	void DeleteOrder( Order order)	Deletes the order.	
■     Save       ■     Events       ダ     OrderSaved       ダ     OrderSaving	<pre>void DeleteOrders(     Expression<func<order, bool="">&gt; expression)</func<order,></pre>	Takes the Linq expression and deletes the orders.	
	Order GetOrder( string orderId)	Takes the order ID and returns the order.	
	<pre>IQueryable<order> GetOrders(     Expression<func<order, bool="">&gt; expression)</func<order,></order></pre>	Takes the Linq expression and returns the orders.	
	void Save(	Saves the order.	

## 2.6.10 MerchantOrderStateConfiguration

Order order)

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)</substate>	Gets the admissible substates of the merchant order.
IQueryable <state> GetAvailableStates( Order order)</state>	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 © 2001-2014 Sitecore. All rights reserved.

sitecore



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

#### OrderFieldProcessingStrategy 2.6.11

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

OrderFieldProcessingStrategy (Rest)	The methods of the OrderFieldProcessing are:	Strategy <b>class</b>
→ OrderProcessingStrategy	Method	Description
<ul> <li>□ Fields</li> <li>③● loggingEntries</li> <li>③● order</li> <li>③● previousFieldValues</li> <li>□ Properties</li> </ul>	<pre>IList<logentry> GetAdditionalLogEntriesForSuccess(             Order order)</logentry></pre>	Performs additional success processing.
Image: Second state       Image: Second state         Image: Second state       Image: Second state </td <td>LogEntry GetLogEntryForFail( Order order, params object[] parameters)</td> <td>Gets the logged message of the operation that failed and</td>	LogEntry GetLogEntryForFail( Order order, params object[] parameters)	Gets the logged message of the operation that failed and
<ul> <li>EvaluateLoggingEntries</li> <li>GetAdditionalLogEntriesForSuccess</li> <li>GetLogEntryForFail</li> </ul>		performs additional
<ul> <li>GetLogEntryForSuccess</li> <li>OrderFieldProcessingStrategy</li> <li>ProcessOrder</li> </ul>	LogEntry GetLogEntryForSuccess( Order order)	Gets the logged message of the successful operation.
	OrderFieldProcessingStrategy( IDictionary <string, object=""> previousFieldValues)</string,>	Initializes a new instance of the class.
	<pre>string ProcessOrder(     Order order,     IDictionary<string, object="">     parameters)</string,></pre>	Processes the order.



## 2.6.12 OrderLineFactoryImpl

This class implements the <code>OrderLineFactory</code> and defines the implementation of the order line factory.

OrderLineFactoryImpl (RectoryImpl Class are:		
OrderLineFactory	Property	Description
□ Fields	IProductRepository ProductRepository	Gets or sets the product repository.
of product of totals □ Properties	IProductPriceManager ProductPriceManager	Gets or sets the product price manager.
<ul> <li>CurrencyProvider</li> <li>ProductPriceManager</li> <li>ProductRepository</li> </ul>	<pre>IEntityProvider<domainmodel.curre ncies.currency=""> CurrencyProvider</domainmodel.curre></pre>	Gets or sets the currency provider.
Methods  • CreateLineItemFromOrder		

CreateOrderLineFromOrder
 ResolvePreConditions

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.

OrderLineStateTransitionStrategy Class + OrderProcessingStrategy	8
Properties	
<ul> <li>OrderLineState</li> <li>OrderManager</li> </ul>	
□ Methods	
<ul> <li>QetLoggingEntryForFail</li> <li>QetLoggingEntryForSuccess</li> <li>ProcessOrder</li> <li>StartOrderPackedPipeline</li> </ul>	

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
<pre>string ProcessOrder(     Order order,     IDictionary<string, object="">     parameters)</string,></pre>	Takes the order, the parameters, and processes the order.



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.
<pre>void StartOrderPackedPipeline(</pre>	Takes the order, and starts the order packed pipeline.

#### 2.6.14 OrderLineProcessingStrategy

This is an abstract class that implements the <code>OrderProcessingStrategy</code> and defines the strategy for processing the order line.

OrderLineProcessingStrategy (Restrated and the second seco	The only public method in the OrderLineProcessingStrategy class is:	
OrderProcessingStrategy	Method	Description
<ul> <li>Fields</li> <li>additionalLoggingEntriesForSuccess</li> <li>availableStates</li> <li>deniedSubstates</li> <li>substateDeactivations</li> </ul>	<pre>IList<logentry> GetAdditionalLogEntriesForSuccess(             Order order)</logentry></pre>	Gets additional logging entry for success.
Methods     AddSubstateDeactivationLogEntry     GetAdditionalLogEntriesForSuccess     SetOrderStates     VpdateTaxesForOrder		

## 2.6.15 OrderProcessingStrategyResolver

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

OrderProcessingStrategyResolver	The only method of OrderProcessingStrategyReslover class is:		
Abstract Class	Method	Description	
■ Methods ■ GetStrategies	<pre>IEnumerable<orderprocessingstrategy> GetStrategies(     Order oldOrder,     Order newOrder)</orderprocessingstrategy></pre>	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.	



#### **OrderProcessingStrategyResolverImpl** 2.6.16

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

OrderProcessingStrategyResolverImpl	The properties of the OrderProcessingStrategyResolverImpl class are:	
<ul> <li>OrderProcessingStrategyResolver</li> </ul>	Property	Description
Fields  fields  rotategies  Properties	OrderProcessingStrategy CancelOrderProcessingStrategy	Gets or sets the cancel order processing strategy.
<ul> <li>CancelOrderProcessingStrategy</li> <li>CaptureOrderProcessingStrategy</li> <li>OrderStateProcessingStrategy</li> <li>PackOrderProcessingStrategy</li> <li>ShipOrderProcessingStrategy</li> </ul>	OrderProcessingStrategy CaptureOrderProcessingStrategy	Gets or sets the capture order processing strategy.
Methods     CheckGivenSubstate     CheckInProcessSubstate     CheckInProcessSubstate     S     CheckInProcessSubstates     S     FilterChangedFields     S     GetAddedEntityFields     GetStrategies     ProcessStateChanges	OrderProcessingStrategy OrderStateProcessingStrategy	Gets or sets the order state processing strategy.
	OrderProcessingStrategy PackOrderProcessingStrategy	Gets or sets the pack order processing strategy.
TimeSpanComparator	OrderProcessingStrategy ShipOrderProcessingStrategy	Sets or sets the ship order processing strategy.

The methods of the OrderProcessingStrategyResolverImpl class are:

Method	Description
<pre>IEnumerable<orderprocessingstrategy> GetStrategies(     Order oldOrder,     Order newOrder)</orderprocessingstrategy></pre>	Gets the order processing strategies.

#### 2.6.17 OrderStateProcessingStrategy

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

OrderStateProcessingStrategy (Response)	The only property of the OrderStatePr Property	cocessingStrategy class is: Description
Fields     initialOrder	MerchantOrderStateConfiguratic OrderStateConfiguration	Gets or sets the order state configuration.
<ul> <li>Properties</li> <li>OrderStateConfiguration</li> <li>Methods</li> <li>GetLogEntry</li> <li>GetLogEntryForFail</li> <li>GetLogEntryForSuccess</li> <li>ProcessOrder</li> </ul>	The methods of the OrderStateProce Method	ssingStrategy <b>class are</b> : <b>Description</b>
	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.

~ I. , ----

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 © 2001-2014 Sitecore. All rights reserved.



Method	Description
string ProcessOrder(	Processes the order.
Order order,	
IDictionary <string,< td=""><td></td></string,<>	
<pre>object&gt; parameters)</pre>	

## 2.6.18 PackOrderProcessingStrategy

This class implements the <code>PackStateProcessingStrategy</code> class and defines the strategy for processing the pack order.

PackOrderProcessingStrategy (2)	The methods of the PackOrderProcessingStrategy class are:	
Class -> OrderStateProcessingStrategy	Method	Description
Methods GetLogEntry GetLogEntryForFail GetLogEntryForSuccess	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 <code>OrderProcessingStrategy</code> class and defines the processing strategy for removing the order line.

RemoveOrderLineProcessingStrategy 🗵	The properties of the RemoveOrderLineProcessingStrategy.class are:	
OrderProcessingStrategy	Property	Description
□ Fields	MerchantOrderManager	Gets or sets the order manager.
🔗 formattedMessage	OrderManager	
Properties		
<ul> <li>OrderManager</li> <li>ProductStockManager</li> </ul>	IProductStockManager ProductStockManager	Gets or sets the product stock manager.
Methods		
<ul> <li>CreateFormattedMessage</li> <li>GetLoggingEntryForFail</li> <li>GetLoggingEntryForSuccess</li> <li>ProcessOrder</li> </ul>		

## The methods of the RemoveOrderLineProcessingStrategy class are:

Method	Description
<pre>string ProcessOrder(     Order order,     IDictionary<string, object=""> parameters)</string,></pre>	Takes the order, the parameters, and processes the order.
LoggingEntry GetLoggingEntryForSucce ss( Order order)	Takes the order, and gets the logged message of the successful operation.



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 <code>OrderProcessingStrategy</code> class and defines the processing strategy for sending the order confirmation.

SendConfirmationProcessingStrategy (இ) Class → OrderProcessingStrategy	The Senc Prop
Properties     OrderConfirmation	Orde Orde
□ Methods	
GetLoggingEntryForFail     GetLoggingEntryForSuccess     ProcessOrder	

8	The only property of the SendConfirmationProcessir	ngStrategy <b>class is:</b>
	Property	Description
	OrderConfirmation OrderConfirmation	Gets or sets the order confirmation.

The methods of the SendConfirmationProcessingStrategy.class are:

Method	Description
<pre>string ProcessOrder(     Order order,     IDictionary<string, object=""> parameters)</string,></pre>	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.
<pre>LoggingEntry GetLoggingEntryForFail(     Order order,     params object[]     parameters)</pre>	Takes the order, the parameters, and gets the logged message of the operation that failed.

## 2.6.21 ShipOrderProcessingStrategy

This class implements the <code>OrderStateProcessingStrategy</code> class and defines the processing strategy for shipping the order.

ShipOrderProcessingStrategy 🛞	The methods of the ShipOrderProcessingStrategy class are	
Class   OrderStateProcessingStrategy	Method	Description
Methods GetLogEntry GetLogEntryForFail GetLogEntryForSuccess	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 © 2001-2014 Sitecore. All rights reserved.



## 2.6.22 StateValidator

This class defines the state validator.

StateValidator (Slass	The only method of the StateValida Method	ator <b>class is</b> : <b>Description</b>
Methods     GetAdmissibleSubstates	<pre>IEnumerable<substate> GetAdmissibleSubstates(     State state,     SubstateCombinationSet     substateCombinationSet)</substate></pre>	Takes the state, the sub-state combination set, and gets the admissible sub-states.



## 2.7 Sitecore.Ecommerce.Merchant.OrderManagement.Extensions

#### 2.7.1 MerchantExtensions

This is a static class that defines the merchant extensions.

MerchantExtensions 🛞	The only method of the MerchantExtensions class is:	
Static Class	Method	Description
Methods	ReservationTicket	Takes the order, and creates the
🍕 GetReservationTicket	GetReservationTicket(	reservation tickets from the order.
·/	this Order order)	



## 2.8 Sitecore.Ecommerce.Visitor.OrderManagement

### 2.8.1 VisitorOrderCancellationStrategy

This class implements the <code>ProcessingStrategy</code> class and defines the strategy for cancelling the visitor order.

VisitorOrderCancelationStrategy (8)	The only property of the VisitorOrderProcessingStrategy class is:	
Class Processing Strategy	Property	Description
	StateManager <state></state>	Gets or sets the state manager.
Properties	StateManager	
🚰 StateManager		
Methods		
Process		

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.

VisitorOrderProcessor 🔕	The properties of the VisitorOrderProcessor class are:	
Class	Property	Description
Properties	OrderProcessor	Gets or sets the inner processor.
InnerProcessor	InnerProcessor	
ProcessingStrategy Repository	ProcessingStrategy	Gets or sets the processing strategy.
□ Methods	ProcessingStrategy	
CancelOrder	Repository <order> Repository</order>	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.

VisitorOrderRepository (Repository Class is:

Class	Property	Description
Properties	Repository <order></order>	Gets or sets the inner
InnerRepository	InnerRepository	order repository.
Methods		
Create		
=∳ GetAll		



The methods of the VisitorOrderRepository class are:

Method	Description
void Create( Order order)	Takes the order, and creates it.
<pre>IQueryable<order> GetAll(     Expression<func<order, bool="">&gt;     expression)</func<order,></order></pre>	Takes the expression, and gets all the orders.

## 2.8.4 TransientOrderConverter

This class defines the transient order converter.

MapSellerSupplierParty

TransientOrderConverter	The properties of the TransientOrderConverter class are:	
Class	Property	Description
Fields January Provide AllowanceChargeReason	CompanyMasterData CompanyMasterData	Gets or sets the company master data.
<ul> <li>AllowanceChargeReasonCode</li> <li>DefaultNotificationOption</li> <li>Properties</li> </ul>	IEntityProvider <country> CountryProvider</country>	Gets or sets the country provider.
CompanyMasterData CountryProvider NotificationOptionProvider	IEntityProvider <notificationoption> NotificationOptionProvider</notificationoption>	Gets or sets the notification option provider.
<ul> <li>ShippingProvider</li> <li>Methods</li> </ul>	OrderLineFactory OrderLineFactory	Gets or sets the order line factory.
<ul> <li>○ Convert (+ 1 overload)</li> <li>○ GetCompanyMasterData</li> <li>○ GetCountry</li> <li>○ GetNotificationOption</li> </ul>	IEntityProvider <shippingprovider> ShippingProvider</shippingprovider>	Gets or sets the shipping provider.
<ul> <li>GetNotificationOption</li> <li>GetShippingProvider</li> <li>GetMapAmounts</li> </ul>		

The only public method of the TransientOrderConverter class is:

Method	Description
Order Convert( DomainModel.Orders.Order source)	Converts the specified source.



## 2.8.5 TransientOrderManager

 $This \ class \ implements \ the \ {\tt IOrderManager} \ interface \ and \ defines \ the \ transient \ order \ manager.$ 

TransientOrderManager 🛛	Property	Description
Class	ICustomerManager <customerinfo></customerinfo>	Gets or sets the
E Fields	CustomerManager	customer manager
<ul> <li>♂ defaultStartPathItem</li> <li>☑ DefaultStatus</li> </ul>	Sitecore.Data.Items.Item DefaultStartPathItem	Gets or sets the default start path
Properties		item.
CustomerManager DefaultStartPathItem EntityHelper InnerRepository OrderConverter	EntityHelper EntityHelper	Gets or sets the entity helper instance.
<ul> <li>OrderIDGenerator</li> <li>ShopContext</li> <li>TransactionData</li> </ul>	VisitorOrderRepository InnerRepository	Gets or sets the inner repository.
<ul> <li>VirtualProductResolver</li> <li>Methods</li> </ul>	TransientOrderConverter OrderConverter	Gets or sets the order converter.
<ul> <li> <sup></sup> <sup></sup> <sup></sup></li></ul>	OrderIDGenerator OrderIDGenerator	Gets or sets the order number generator.
<ul> <li>i GetFriendlyUrl</li> <li>i GetOrder</li> <li>i GetOrder</li> <li>i GetOrders</li> </ul>	ShopContext ShopContext	Gets or sets the shop context.
<ul> <li>GetOrdersCount <tquery></tquery></li> <li> <sup>©</sup> <sup>®</sup> <sup>©</sup> <sup>®</sup> <sup>®</sup></li></ul>	ITransactionData TransactionData	Gets or sets the transaction data.
image: second control of the second control of th	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</tshoppingcart>	Takes the shopping cart and creates the order.	
<pre>string GenerateOrderNumber()</pre>	Generates the order number.	
virtual OldOrder GetOrder( string orderNumber)	Takes the order number gets the order.	
<pre>virtual IEnumerable<oldorder> GetOrders<tquery>(     TQuery query)</tquery></oldorder></pre>	Takes the query and gets the orders.	
<pre>virtual IEnumerable<oldorder> GetOrders<tquery>(     TQuery query,     int pageIndex,     int pageSize)</tquery></oldorder></pre>	Takes the query, the page index and the page size and gets the orders.	



## 2.9 Sitecore.Ecommerce.Visitor.Pipelines.HttpRequest

## 2.9.1 VisitorShopResolvingProcessor

This class defines the resolving processor of the visitor shop.

VisitorShopResolvingProcessor (ResolvingProcessor Class	The only property of the VisitorShopResolvingProcessor class is:	
■ Fields	Property	Description
eresolver	VisitorShopResolver Resolver	Gets or sets the resolver.
Properties		I
😭 Resolver		
Methods		
Process		

The only method of the VisitorShopResolvingProcessor class is:

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



## 2.10 Sitecore.Ecommerce.Visitor.Pipelines.OrderedCreated

#### 2.10.1 NotifyCustomer

This class defines the process of notifying the customer.

NotifyCustomer 🗵	I he only method of the Notify	nethod of the NotifyCustromer class is:	
Class	Method	Description	
Methods	void Process(	Runs the processor.	
Process	PipelineArgs args)		



## 2.11 Sitecore.Ecommerce.Visitor.Sites

#### 2.11.1 VisitorShopResolver

This class defines the resolver of the visitor shop.

(	VisitorShopResolver	The only method of the VisitorShopRe	solver <b>class is</b> :	
	Class	Method	Description	
	Methods	ShopContext GetShopContext(	Gets the shop context.	
	GetShopContext	SiteContext siteContext)		



## 2.12 Sitecore.Ecommerce.Visitor.Unity

#### 2.12.1 ShopContextContainerExtension

This class defines the extension of the site context container.

۲	The only property of the ShopContextContainerExtension is:	
	Method	Description
	VisitorShopResolver	Gets or sets the resolver.
	Resolver	

