

SMS Broker System
Software Integration

Our system uses
WCF
(Windows Communication Foundation)
&
.NET

Accounts Receivable classes and interfaces.

(ver. from 8/3/2026)

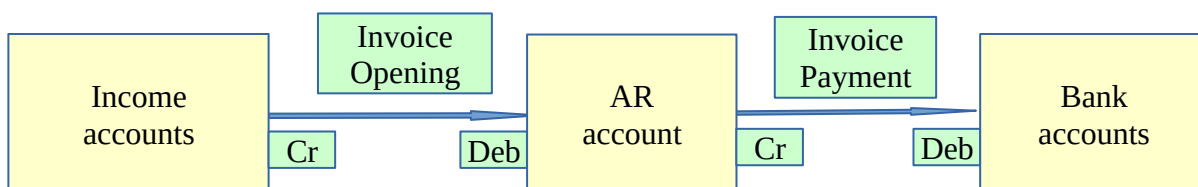
Table of Contents

1. Special Notes.	1
2. Customer Invoice.	2
2.1 Customer Invoice Item.	7
2.2. Customer Invoice Debt.	8
2.3. Customer Invoice Debt Turn.	9
3. Cash Receipt.	9
3.1 Cash Receipt Customer Invoice.	11
4. Charge.	12

1. Special Notes.

1. In SMS-system the “customer invoice” notion is used to designate the document that usually is named in commercial business as “invoice” i.e. document that is billed to customer for the certain service. The word “customer” is added specially to avoid a confusion with invoices presented on customs entry document.
2. [CustomsEntry](#), [Shipment](#), [ReconEntry](#), [ImporterSecurityFiling](#), [ReconEntry](#) and [Shipment](#) classes are described in other SMS API documents.
3. [Payment](#) class mentioned here is a part of Accounting Payable subsystem.

2. Accounts Receivable workflow.



1. Accounting. In the picture above you can see two operations – “Invoice Opening” and “Invoice Payment”. The first one includes a creating, saving and posting (New(), Save() and Post() methods respectively) of a [CustomerInvoice](#) class object. Posting process validates a new added document and makes it “visible” to system. After that the system takes it into account in all reports and another related documents. Besides that

[CustomerInvoiceDebt](#) object with customer invoice balance and [CustomerInvoiceDebtTurn](#) object with customer invoice amount and TurnType == 1 are added.

The second operation is "Invoice Payment". To implement it you need to add [CashReceipt](#) class object. Again it includes a creating, saving and posting operations. Object is created by New() method, then the invoice (invoices) to be paid are added to cash receipt object's detail and then the object is saved and posted by Save() and Post() methods. Posting process validates the cash receipt object and changes a balance of related invoices according to allocated cash receipt detail amounts.

In addition [CustomerInvoiceDebtTurn](#) objects (that is related to previously created [CustomerInvoiceDebt](#) objects per invoice) with customer invoice allocated amount and TurnType == -1 is added.

2. Charges.

Charges allow to work with analytics level and used in particular in invoice and voucher details. They can be of user type (added by user) and standard type. Standard type charges are linked to Sources.

3. Sources.

"Source" is a document (customs entry, breakbulk, shipment, delivery order, ISF) that is served as the basis for some detail charge in customer invoice or voucher. Each Source has a set of inherent to it charges that can be used. For example, if the Source is delivery order then they can be "Container delivery", "Container-HQ delivery", "Container-overweight delivery", "Port congestion fee" and "Container delivery chassis".

4. Jobs.

During processing, the information related to some business task passes through different SMS-subsystems – Booking, Breakbulk, ISF, CustomsEntry, Delivery Order. If your company is a freight forwarder and at the same time is a customs broker the business task can appear in SMS-system on different stages and suggests different sets of operations and hence charges. Each Job has a set of Sources inherent to it and is used to keep together operations related to one business task.

For example, you have a customer that is an importer and at the same time is an ultimate consignee. A part of you service is customs brokerage (customs release, duty payment, etc.) and another part is freight forwarding (freight release, delivery, etc). On your case the Job is "Customs Entry" because it includes all related sources and accordingly includes charges. On another case your activity is only the freight forwarding and Job is "Breakbulk". It doesn't include customs entry charges. More detailed the info about relations between Job, Source and Charges you can see in *25. Annex. Accounts Receivable classes and interfaces.pdf, 1. Jobs, Sources and Charges*.

5. Parent invoice.

This notion has the same purpose as a Job – to aggregate invoices related to one business task. For now it is obsolete but is maintained for backward compatibility.

2. Customer Invoice.

Customer invoice class object represents a customer invoice document.

Class: [\[DataContract\]](#)

SMS.Broker.DataContracts.Documents.[CustomerInvoice](#) : [EntityDocument](#)

Header data of Statement document.

Properties:

Attributes	Name	Type	Description
[NotMapped] [IgnoreDataMember]	BalanceAmount	decimal	Balance amount. [1]
[DataMember] [ForeignKey("BillTo_Id")]	BillTo	Contact	Contact the customer invoice is billed to.
[DataMember]	BillTo_Id	long?	Bill to Id.
[ForeignKey("BreakBulk_Id")] [DataMember]	BreakBulk	BreakBulk	The breakbulk document as a job to which the customer invoice references to.

[DataMember]	BreakBulk_Id	long?	Breakbulk Id.
[NotMapped] [IgnoreDataMember]	CalculatedPaidAmount	decimal	Paid amount calculated on debt turns basis.[1]
[NotMapped] [IgnoreDataMember]	CalculatedStatus	string	For backward compatibility. Consolidated customer invoice status that calculated based on another properties. Has Possible Values.
[ForeignKey("Carrier_Id")] [DataMember]	Carrier	Carrier	Carrier contact.
[DataMember]	Carrier_Id	long?	Carrier Id.
[DataMember] [DecimalPrecision(9, 0)]	Cartons	decimal	Cartons.
[DataMember] [InverseProperty("CustomerInvoice")]	CashReceiptLines	List<CashReceiptCustomerInvoice>	References to detail lines of a cash receipt by which the invoice was paid.
[DataMember] [StringLength(25)]	ClientRef	string	Client reference.
[DataMember] [StringLength(2)]	CountryOfOrigin	string	Has Possible Values.
[DataMember] [ForeignKey("Customer_Id")]	Customer	Contact	The contact to which the services were provided.
[DataMember]	Customer_Id	long	Customer Id.
[ForeignKey("CustomsEntry_Id")] [DataMember]	CustomsEntry	CustomsEntry	The customs entry document as a job to which the customer invoice references to.
[DataMember]	CustomsEntry_Id	long?	Customs entry Id.
[StringLength(25)] [DataMember]	CustomShipmentClientRefs	string	Reference field. ("References:" field on the print form).
[DataMember] [InverseProperty("CustomerInvoice")]	Debts	List<CustomerInvoiceDebt>	Customer Invoice Debt collection.
[DataMember] [InverseProperty("CustomerInvoice")]	DebtTurns	List<CustomerInvoiceDebtTurn>	Customer Invoice Debt turns collection.
[DataMember] [DecimalPrecision(13, 2)]	DeductionAmount	decimal	Deduction amount.[1]
[ForeignKey("DeliveryOrder_Id")] [DataMember]	DeliveryOrder	DeliveryOrder	The delivery order document as a job to which the customer invoice references to.
[DataMember]	DeliveryOrder_Id	long?	Delivery order Id.
[DataMember] [StringLength(60)]	Description	string	Invoice description info.
[NotMapped] [IgnoreDataMember]	EmailedDate	DateTime?	The date when customer invoice was emailed.
[Obsolete("LastStatuses", false)] [DataMember] [InverseProperty("CustomerInvoice")]	EntityStatuses	EntityStatus	Obsolete. Use LastStatuses instead. Statuses of Customs entry.
[DataMember]	ImporterSecurityFiling	ImporterSecurity	The ISF document as a job

[ForeignKey("ImporterSecurityFiling_Id")]		Filing	to which the customer invoice references to.
[DataMember]	ImporterSecurityFiling_Id	long?	ISF Id.
[DataMember] [InverseProperty("CustomerInvoice")]	Items	CustomerInvoiceItem	The collection that constitutes a detailed part of customer invoice.
[NotMapped] [IgnoreDataMember]	Job	EntityBase	A job to which the customer invoice references to. It can be: BreakBulk, CustomsEntry, DeliveryOrder, ImporterSecurityFiling, ReconEntry or Shipment class object.
[NotMapped] [IgnoreDataMember]	JobNumber	long	Job number.
[DataMember(EmitDefaultValue = false)] [InverseProperty("CustomerInvoice")]	LastStatuses	EntityLastStatus	Last status collection.
[StringLength(100)] // IH_NOTES Char(100) [DataMember]	Note5	string	Not used.
[Column(TypeName = "text")] [DataMember]	Notes	string	Notes. Used on the customer invoice print form.
[ForeignKey("ParentInvoice_Id")] [DataMember]	ParentInvoice	CustomerInvoice	An invoice initial for some Job.
[DataMember]	ParentInvoice_Id	long?	Parent invoice Id.
[Column(TypeName = "date")] [DataMember]	PrintDate	DateTime?	The date when customer invoice was printed.
[DataMember] [ForeignKey("ReconEntry_Id")]	ReconEntry	ReconEntry	The reconciliation entry document as a job to which the customer invoice references to.
[DataMember]	ReconEntry_Id	long?	Reconciliation entry Id.
[DataMember] [ForeignKey("Shipment_Id")]	Shipment	Shipment	The shipment document as a job to which the customer invoice references to.
[DataMember]	Shipment_Id	long?	Shipment Id.
[StringLength(64)] [DataMember]	SourceLink	string	Link to Job (BreakBulk, CustomsEntry, DeliveryOrder, ImporterSecurityFiling, ReconEntry or Shipment) class object
[NotMapped] [IgnoreDataMember]	TotalAmount	decimal	Total amount. [1]
[Column(TypeName = "date")] [DataMember]	VoidDate	DateTime?	The date when customer invoice was voided.
[DataMember] [DecimalPrecision(14, 2)]	WeightKg	decimal	Weight in kilograms.

[1]

DeductionAmount is entered by user.
TotalAmount is calculated as a sum of [CustomerInvoiceItem](#) class object Amount property.
CalculatedPaidAmount is calculated as a sum of [CustomerInvoiceDebtTurn](#) class objects Amount property for TurnType == -1.
BalanceAmount is equal to TotalAmount - (CalculatedPaidAmount + DeductionAmount)

Interface: [\[ServiceContract\]](#)
SMS.Broker.ServiceContracts.Documents.ICustomerInvoiceManager :
IEntityManagerDocument<SMS.Broker.DataContracts.Documents.CustomerInvoice> [1]

Methods:

Attributes	Name	Type	Description
[OperationContract] [FaultContract(typeof(ActionFault))]	CreateAdditionalCustomerInvoicesForVoucher(List<CustomerInvoiceCorrectionByVoucherInfo> sourceInfo)	List<CustomerInvoiceCorrectionByVoucherInfo>	Specialized and used in SMS-Shell wizard.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetByBreakBulk(long BreakBulkId, long ImporterId)	List<CustomerInvoice>	Returns a list of customer invoice by breakbulk and importer Ids.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetByCustomsEntry(long CustomsEntryId)	List<CustomerInvoice>	Returns a list of customer invoice by customs entry and importer Ids.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetByDeliveryOrder(long DeliveryOrderId)	List<CustomerInvoice>	Returns a list of customer invoice by delivery order and importer Ids.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetByImporterSecurityFiling(long ImporterSecurityFilingId)	List<CustomerInvoice>	Returns a list of customer invoice by ISF and importer Ids.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetByReconEntry(long ReconEntryId)	List<CustomerInvoice>	Returns a list of customer invoice by recon entry and importer Ids.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetByShipment(long ShipmentId)	List<CustomerInvoice>	Returns a list of customer invoice by shipment and importer Ids.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetBySourceLinks(string[] links, long? customerId)	List<CustomerInvoice>	Gets a list of customer invoices by customer Id and a list of source links.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetBySourceRef(CustomerInvoice invoice)	CustomerInvoice	Adds charges to customer invoice based on its Job property.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetCoLoadersShipmentsInfo(long breakBulk_Id)	List<CoLoaderShipmentInfo>	Specialized and used in SMS-Shell wizard.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetCustomerInvoiceCorrectionBySourceInfo(string link, List<CustomerInvoiceCorrectionBySourceInfo> nBySourceInfo.PreferredToBeIn	List<CustomerInvoiceCorrectionBySourceInfo>	Specialized and used in SMS-Shell wizard.

	<code>cude> preferredToBeIncude)</code>		
[OperationContract] [FaultContract(typeof(ActionFault))]	GetCustomerInvoiceCorrectionByVoucherInfo(long voucher_Id)	List<CustomerInvoiceCorrectionByVoucherInfo>	Specialized and used in SMS-Shell wizard.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetCustomerInvoicesAvailableForCashReceipt(long? Payer_Id, DateTime? DateFrom, DateTime? DateTo)	List<CustomerInvoiceAvailableInfo>	Specialized and used in SMS-Shell wizard.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetCustomerInvoicesByLink(string link, long contactId)	List<CustomerInvoice>	Gets a list of customer invoices by customer Id and a source link.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetCustomsEntriesDutyPaymentNotes(List<long> ceIds, long customerInvoice_Id)	string	Returns info about customs duty as a text by list of customs entry Ids.
[OperationContract]	GetLastOpen(CustomerInvoice entity)	CustomerInvoice	Returns the last open customer invoice that belongs to the same Job as a given invoice.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetPostedCustomerInvoiceItemsBySourceLink(string link)	List<CustomerInvoiceItem>	Returns invoice items by Source link. Includes only charges those are not of incoming type.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetRelatedInvoices(long Id, bool includeVoided, string include)	List<CustomerInvoice>	Returns a list of customer invoices that includes a parent invoice also all invoices those have the same parent invoice as a given child invoice. If given invoice is parent method returns it and also all child invoices.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetVoucherIdsByCustomerInvoiceSources(long customerInvoice_Id)	List<string>	Returns a list of voucher Id-s related to customer invoice by means of customer invoice items source.
[OperationContract] [FaultContract(typeof(ActionFault))]	MakeCustomerInvoiceCorrectionBySourceInfo(List<CustomerInvoiceCorrectionBySourceInfo> sourceInfo)	List<CustomerInvoiceCorrectionBySourceInfo>	Specialized and used in SMS-Shell wizard.
[OperationContract]	SetPrintStatus(CustomerInvoice invoice)	void	Set a printed date to invoice when it is sent to customer.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))]]	WizardForBillingChargeInfo_CreateAdditionalInvoicesAndVoucher(BillingChargeInfo billingChargeInfo)	BillingChargeInfo	Specialized and used in SMS-Shell wizard.

[1] - The features of `ICustomerInvoiceManager` those are different from other Operation Contract inherited classes ones.

a) New(long? SourceId) Method returns a new `CustomerInvoice` class object. Its number will be assigned by "CustomerInvoice" sequence.

b) For Save([CustomerInvoice](#) entity) method Date, Customer, and BillTo properties are required. CountryOfOrigin property is required only if CustomsEntry or ImporterSecurityFiling properties are not [null](#). By this method, on the server side CustomerInvoice object record and the records corresponded to the CustomerInvoice.Items collection will be saved.

c) The follow statuses are used for customer Invoice document.

Entity status type:

[EntityStatusType](#).DocumentEmailed

Entity status values:

[EntityStatusValue](#).Yes

[EntityStatusValue](#).No

Entity status type:

[EntityStatusType](#).DocumentPosted

Entity status values:

[EntityStatusValue](#).Yes

[EntityStatusValue](#).No

2.1 Customer Invoice Item.

Class: [[DataContract](#)]

SMS.Broker.DataContracts.Documents.[CustomerInvoiceItem](#) : [EntityChild](#)

The class contains an information about Customer Invoice Item (line). The collection of these items constitutes a detail of Customer Invoice.

Properties:

Attributes	Name	Type	Description
[DataMember] [DecimalPrecision (13, 2)]	Amount	decimal	Invoice item Amount.
[ForeignKey ("BreakBulk_Id")] [DataMember]	BreakBulk	BreakBulk	The breakbulk document as a source of the customer invoice item.
[DataMember]	BreakBulk_Id	long?	Breakbulk Id.
[DataMember] [ForeignKey ("Charge_Id")]	Charge	Charge	Accounting charge.
[DataMember]	Charge_Id	long	Charge Id.
[ForeignKey ("CustomerInvoice_Id")] [DataMember]	CustomerInvoice	CustomerInvoice	The reference to customer invoice object this invoice item belongs to.
[DataMember]	CustomerInvoice_Id	long	Customer Invoice Id.
[ForeignKey ("CustomsEntry_Id")] [DataMember]	CustomsEntry	CustomsEntry	The customs entry document as a source of the customer invoice item.
[DataMember]	CustomsEntry_Id	long?	Customs entry Id.
[ForeignKey ("DeliveryOrder_Id")] [DataMember]	DeliveryOrder	DeliveryOrder	The delivery order document as a source of the customer invoice item.
[DataMember]	DeliveryOrder_Id	long?	Delivery order Id.
[ForeignKey ("ImporterSecurityFiling_Id")]	ImporterSecurityFiling	ImporterSecurityFiling	The ISF document as a

[DataMember]			source of the customer invoice item.
[DataMember]	ImporterSecurityFiling_Id	long?	ISF Id.
[NotMapped] [IgnoreDataMember] [KnowEntityTypes(new System.Type[] { typeof(BreakBulk), typeof(CustomsEntry), typeof(DeliveryOrder) , typeof(ImporterSecurityFiling), typeof(ReconEntry), typeof(Shipment) })]]	ItemSource	EntityBase	Source of customer invoice item. It can be: BreakBulk, CustomsEntry, DeliveryOrder, ImporterSecurityFiling, ReconEntry or Shipment object.
[DataMember] [DecimalPrecision(8, 3)]	Quantity	decimal	Quantity of invoice item.
[ForeignKey("ReconEntry_Id")] [DataMember]	ReconEntry	ReconEntry	The reconciliation entry document as a source of the customer invoice item.
[DataMember]	ReconEntry_Id	long?	Reconciliation entry Id.
[ForeignKey("Shipment_Id")] [DataMember]	Shipment	Shipment	The shipment document as a source of the customer invoice item.
[DataMember]	Shipment_Id	long?	Shipment Id.

Methods:

Attributes	Name	Type	Description
	ToString()	string	Returns text representation.

2.2. Customer Invoice Debt.

Class: [DataContract]

SMS.Broker.DataContracts.Registers.CustomerInvoiceDebt : EntityBase

Objects of this class store current balance for customer invoice.

Properties:

Attributes	Name	Type	Description
[DataMember] [DecimalPrecision(14, 2)] [Index(false)]	Amount	decimal	Current balance customer invoice amount.
[DataMember] [ForeignKey("CustomerInvoice_Id")]	CustomerInvoice	CustomerInvoice	The reference to customer invoice object this invoice item belongs to.
[DataMember]	CustomerInvoice_Id	long	Customer Invoice Id.
[IgnoreDataMember]	TermDays	int	The difference between current date and customer invoice date expressed in days.
[IgnoreDataMember]	Turns	List<CustomerInvoiceDebtTurn>	Customer invoice turns.

[NotMapped]			
-------------	--	--	--

2.3. Customer Invoice Debt Turn.

Class: [DataContract]

SMS.Broker.DataContracts.Registers.CustomerInvoiceDebtTurn : EntityBase

Objects of this class represent one credit or debit operation.

Properties:

Attributes	Name	Type	Description
[DataMember] [DecimalPrecision(14, 2)]	Amount	decimal	Operation amount.
[DataMember] [ForeignKey("CustomerInvoice_Id")]	CustomerInvoice	CustomerInvoice	The reference to customer invoice object this invoice item belongs to.
[DataMember]	CustomerInvoice_Id	long	Customer Invoice Id.
[DataMember] [Column(TypeName = "date")]	Date	DateTime	Date of operation.
[IgnoreDataMember] [NotMapped]	Source	EntityDocument	Customer invoice or Cash receipt reference.
[DataMember] [ForeignKey("SourceCashReceipt_Id")]	SourceCashReceipt	CashReceipt	Cash receipt reference if it is "Invoice Payment" operation, otherwise null.
[DataMember]	SourceCashReceipt_Id	long?	Cash receipt Id.
[DataMember] [ForeignKey("SourceInvoice_Id")]	SourceInvoice	CustomerInvoice	Customer invoice reference if it is "Invoice Opening" operation, otherwise null.
[DataMember]	SourceInvoice_Id	long?	Customer invoice Id.
[DataMember]	TurnType	int	Operation Type.

3. Cash Receipt.

CashReceipt object represents a cash receipt document.

Class: [DataContract]

SMS.Broker.DataContracts.Documents.CashReceipt : EntityDocument

Header data of cash receipt document.

Properties:

Attributes	Name	Type	Description
[DataMember] [DecimalPrecision(13, 2)]	AdjustmentAmount	decimal	Adjustment amount.[1]
[IgnoreDataMember] [NotMapped] [DecimalPrecisionAttribute(13, 2)]	Balance	decimal	Current cash receipt balance.[1]

2)]			
[DataMember] [ForeignKey("BankAccount_Id")]	BankAccount	ContactBankAccount	Payment bank account.
[DataMember]	BankAccount_Id	long?	Bank account Id.
[DataMember] [DecimalPrecision(13, 2)]	CheckAmount	decimal	Payer's check amount.[1]
[DataMember] [StringLength(10)]	CheckNumber	string	Payer's check number.
[DataMember] [StringLength(20)]	ClientRef	string	Client reference.
[DataMember] [InverseProperty("OffsettingCashReceipt")]	IncludedToPayments	List<Payment>	Not Used.
[DataMember] [InverseProperty("CashReceipt")]	Invoices	List<CashReceiptCustomerInvoice>	Collection that represents a detail part of cash receipt. It contains references to CustomerInvoice objects.
[DataMember] [InverseProperty("CashReceipt")]	LastStatuses	List<EntityLastStatus>	Last status collection.
[IgnoreDataMember] [NotMapped] [DecimalPrecisionAttribute(13, 2)]	OffsettingAmount	decimal	Offsetting amount.[1]
[DataMember] [ForeignKey("OffsettingPayment_Id")]	OffsettingPayment	Payment	Offsetting Payment.
[DataMember]	OffsettingPayment_Id	long?	Offsetting Payment Id.
[DataMember] [ForeignKey("Payer_Id")]	Payer	Contact	Payer contact.
[DataMember]	Payer_Id	long?	Payer Id.
[DataMember] [StringLength(1)]	PaymentType	string	Has Possible Values.
[DataMember] [InverseProperty("CashReceipt")] [Obsolete("LastStatuses", true)]	Statuses	List<EntityStatus>	Obsolete. Use LastStatuses instead. Statuses of Customs entry.
[IgnoreDataMember] [NotMapped] [DecimalPrecisionAttribute(13, 2)]	TotalAllocatedAmount	decimal	Check amount used to pay invoices.[1]

[1]

CheckAmount is entered by user.
AdjustmentAmount is entered by user.
TotalAllocatedAmount is calculated as a sum of CashReceiptCustomerInvoice class object AllocatedAmount property.
OffsettingAmount is an OffsettingAmount property from offsetting payment that is selected by user.
Balance is equal to CheckAmount + AdjustmentAmount - TotalAllocatedAmount + OffsettingAmount

Interface: [ServiceContract]
SMS.Broker.ServiceContracts.Documents.ICashReceiptManager :
IEntityManagerDocument<CashReceipt> [1]

[1] - The features of ICashReceiptManager those are different from other Operation Contract inherited classes ones.

a) New([long?](#) SourceId) Method returns a new [CashReceipt](#) class object. Its Number property (please, do not mix with CheckNumber property) will be assigned by "CashReceipt" sequence. By default a new Cash Receipt object contains:
PaymentType = "1 - Cash Receipt" and BankAccount that is equal to default user's bank account.

b) For Save([CashReceipt](#) entity) method Payer property is required. By this method, on the server side [CashReceipt](#) object record and the records corresponded to the CashReceipt.Invoices collection will be saved.

c) ValidateTreeById([long](#) Id, string action)
action:

"POST" - on cash receipt case ValidateTreeById([long](#) Id, string action) method makes the same data validation as it is used before Post operation runs. If validation process was not successful [EntryValidationFault](#) will be thrown.

d) The follow statuses are used for cash receipt document.

Entity status type:

[EntityStatusType](#).DocumentPosted

Entity status values:

[EntityStatusValue](#).Yes

[EntityStatusValue](#).No

3.1 Cash Receipt Customer Invoice.

Class: [DataContract](#)

SMS.Broker.DataContracts.Documents.[CashReceiptCustomerInvoice](#) :

SMS.Broker.DataContracts.[EntityChild](#)

The class contains an information about cash receipt Item. Each item refers to related customer invoice. The collection of these items constitute a cash receipt detail.

Properties:

Attributes	Name	Type	Description
DataMember DecimalPrecision (13, 2)]	AllocatedAmount	decimal	The amount for which the invoice has been paid.
ForeignKey ("CashReceipt_Id")] DataMember	CashReceipt	CashReceipt	The reference to cash receipt object this cash receipt item belongs to.
DataMember	CashReceipt_Id	long	Cash receipt Id.
ForeignKey ("CustomerInvoice_Id")] DataMember	CustomerInvoice	CustomerInvoice	Reference to customer invoice object.
DataMember	CustomerInvoice_Id	long?	Customer invoice Id.
NotMapped IgnoreDataMember	Job	EntityBase	Job related to the customer invoice.

Methods:

Attributes	Name	Type	Description
	ToString()	string	Returns empty text "".

4. Charge.

Charge object represents a charge that is used in customer invoice detail.

Class: [DataContract]

SMS.Broker.DataContracts.Documents.Charge : EntityDirectory

Properties:

Attributes	Name	Type	Description
[Required] [StringLength(6)] [DataMember] [Index(true)]	Code	string	Charge code.
[DataMember] [ForeignKey("GroupingCharge_Id")]	GroupingCharge	Charge	Used on invoice print form to group charges.
[DataMember]	GroupingCharge_Id	long?	
[StringLength(50)] [DataMember]	Name	string	Charge name.
[StringLength(3)] [DataMember]	PaymentDirection	string	The direction of amount flow. It is used for "Incoming only" and "Outgoing only" payments. Has Possible Values.
[StringLength(3)] [DataMember]	PreferredSource	string	Source that conventionally used for the charge. It is one of: "Customs Entry" "ISF" "Break Bulk" "Delivery Order" "Shipment" Has Possible Values.
[DataMember]	Priority	long	Index that is used to set up the priority of charges on invoice. It is necessary for automatic invoice generation.
[StringLength(12)] [DataMember] [Index(true)]	SystemCode	string	Not Used.
[StringLength(6)] [DataMember] [Required]	Type	string	Standardized charge type. (like to "ISF", "FREIGHT CHARGE" etc.) or user defined charge type. [1] Has Possible Values.

[1] List of charges can be populated by user. Their codes can be usual for him but unknown to SMS-system. The mapping between charge Code and standard charge Type properties allows SMS-system to use new added charges in wizards and report calculations.