

SMS Broker System
Software Integration

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

Reporting classes and interfaces.

(ver. from 8/3/2026)

Table of Contents

1. Special Notes.	1
2. Reporting manager.	1
3. Reporting manager report methods.	2
3.1. Account Receivable methods.	2
3.1.1. Fee invoiced to customers.	2
3.1.2. Customer Debts.	3
3.1.3. Cash Receipt.	4

1. Special Notes.

Reporting manager methods return a data that can be used as a source for different reports.

2. Reporting manager.

Customer invoice class object represents a customer invoice document.

Interface: `[ServiceContract]`

`SMS.Broker.ServiceContracts.ISmsReportingManager`

Manager that is used to run all reports in SMS-system. "Run a report" here means "run a method" of this manager.

Usage:

```
ServicesFactory sf1 = new ServicesFactory(@"https://localhost:8130/brokerservice", "admin" + "@" + "sandbox", "somepassword");  
var repManager = sf1.SmsReportingManager;
```

About `ClientContext` and `ServicesFactory` classes see more in *01. Getting Started With SMS API.pdf*, 2. *Client Context*.

3. Reporting manager report methods.

3.1. Account Receivable methods.

3.1.1. Fee invoiced to customers.

In SMS-Shell this report is a data source for Reports - > Customer Invoices - > “Fees Invoiced To Customers”.

Method:

Attributes	Name	Type	Description
[OperationContract] [FaultContract(typeof(ActionFault))]	GetReportFeesInvoicedToCustomersSource(DateTime dateFrom, DateTime dateTo, string customerCode, string chargeType)	List<ReportFeesInvoicedToCustomersSource>	Returns Fee charges invoiced to customers [1] by date range and optionally by customer code and charge type.

[1]

Class: [DataContract]

SMS.Broker.DataContracts.Reports.CustomerInvoice.ReportFeesInvoicedToCustomersSource

Class contains a header and a detail parts of the report.

Properties:

Attributes	Name	Type	Description
[DataMember]	CustomerInvoicePrintEntryClientRef	bool	Add customer invoice print entry client reference from system configuration if true.
[DataMember]	CustomerInvoicePrintShipmentsClientRef	bool	Add customer invoice print shipments client reference from system configuration if true.
[DataMember]	ReportFeesInvoicedToCustomersItems	List<ReportFeesInvoicedToCustomersItem>	Detail part of the report.

Class: [DataContract]

ReportFeesInvoicedToCustomersSource.ReportFeesInvoicedToCustomersItem

Objects of this class constitute a detail part of the report. Detail part of the report is a Customer - > Customer Invoice - > Customer Invoice Item hierarchy expanded to a plain table.

Properties:

Attributes	Name	Type	Description
[DataMember]	Amount	decimal	Amount.
[DataMember]	ChargeCode	string	Charge code.
[DataMember]	ChargeName	string	Charge name.
[DataMember]	CustomerCode	string	Customer code.
[DataMember]	CustomerInvoiceDate	DateTime?	Customer invoice date.
[DataMember]	CustomerInvoiceNumber	long?	Customer invoice number.
[DataMember]	CustomerInvoicePrintEntryClientRef	bool	Entry client reference from system configuration.

[DataMember]	CustomerName	string	Customer name.
[DataMember]	CustomsEntryNumber	string	Referred customs entry number.
[DataMember]	ShipmentClientRefNumbers	string	A concatenation of client references from shipments.

Method usage:

```
List<Broker.DataContracts.Reports.CustomerInvoice.ReportFeesInvoicedToCustomersSource> re
= RepManager.GetReportFeesInvoicedToCustomersSource(new DateTime(2025, 11, 1), DateTime.Today, customerCode, null);
```

3.1.2. Customer Debts.

In SMS-Shell this report is a data source for Reports -> Accounts Receivable -> "Customer Debts".

Method:

Attributes	Name	Type	Description
[OperationContract] [FaultContract(typeof(ActionFault))]	GetReportCustomerDebtsSource(DateTime dateFrom, DateTime dateTo, string customerCode)	List<ReportCustomerDebtsSource>	Returns Opening balance, debit and credit turns and closing balance for selected period per customer [1] by date range and optionally by customer code.

[1]

Class: [DataContract]

SMS.Broker.DataContracts.Reports.CustomerDebt.ReportCustomerDebtsSource

Objects of this class constitute a detail part of the report.

Properties:

Attributes	Name	Type	Description
[DataMember]	Details	List<ReportCustomerDebtsSourceDetail>	Detail part of the report.

ReportCustomerDebtsSource.ReportCustomerDebtsSourceDetail

A list of objects of this class is included to detail part of the report. This list is a Customer -> Customer Invoice Debt -> Turns hierarchy expanded to a plain table.

Properties:

Attributes	Name	Type	Description
[DataMember]	CustomerInvoiceDebt	CustomerInvoiceDebt	A reference to customer invoice debt object. [2]
[DataMember]	OpeningBalance	decimal	Opening balance per customer invoice (on the beginning date of the range).

[2]

1. CustomerInvoiceDebt class object Amount property here in the report contains closing balance amount per customer invoice on the ending date of the range (not a current amount as in database).

2. [CustomerInvoiceDebt](#) class object includes Turns collection per customer invoice. Each object of this collection contains Amount property. Positive Amount means debit turns, negative Amount means credit turns. For more detailed description of [CustomerInvoiceDebt](#) class, please see [24. Accounts Receivable classes and interfaces.pdf](#), 2.2. *Customer Invoice Debt*.

Method usage:

```
List<SMS.Broker.DataContracts.Reports.CustomerDebt.ReportCustomerDebtsSource> re
= RepManager.GetReportCustomerDebtsSource(new DateTime(2025, 11, 1), DateTime.Today, customerCode);
```

3.1.3. Cash Receipts.

In SMS-Shell this report is a data source for Reports - > Accounts Receivable - > “Cash Receipts”.

Method:

Attributes	Name	Type	Description
[OperationContract] [FaultContract(typeof(ActionFault))]	GetReportCashReceiptsSource(DateTime? dateFrom, DateTime? dateTo, string customerCode, long? cashReceipt_Id, bool onlyPosted = true)	List<ReportCashReceiptsSource>	Returns customer, cash receipt, customer invoice turns and balance info [1] by date range, customer code and cash receipt id. Take into account only posted cash receipts if onlyPosted = true.

[1]

Class: [DataContract](#)

[SMS.Broker.DataContracts.Reports.AccountsReceivable.ReportCashReceiptsSource](#)

Class contains a header and a detail parts of the report.

Properties:

Attributes	Name	Type	Description
[DataMember]	CashReceipt	CashReceipt	Cash receipt input parameter.
[DataMember]	Customer	Contact	Customer contact input parameter.
[DataMember]	Details	List<ReportCashReceiptsSourceDetail>	Detail part of the report. [2]

[2]

[ReportCashReceiptsSource.ReportCashReceiptsSourceDetail](#)

Objects of this class constitute a detail part of the report. Detail part of the report is a Payer - > Cash Receipt - > Customer Invoice hierarchy expanded to a plain table. Grouping and summarizing it by customer code, cash receipt id (or check number), customer invoice id you can get turns and a balance respectively for customers, cash receipts and customer invoices.

Properties:

Attributes	Name	Type	Description
[DataMember]	CashReceipt	CashReceipt	Cash receipt.

[DataMember]	CashReceiptApplied	decimal	Check amount applied to invoices.
[DataMember]	CashReceiptBalance	decimal	Current cash receipt balance.
[DataMember]	CashReceiptOffsettingAmount	decimal	Offsetting payment amount.
[DataMember]	CheckAmount	decimal	Full cash receipt check amount.
[DataMember]	CustomerInvoice	CustomerInvoice	Customer invoice.
[DataMember]	Database	string	For internal usage only.
[DataMember]	Days	int	Number of days between customer invoice date and related cash receipt date: (CashReceipt.Date - CustomerInvoice.Date).TotalDays
[DataMember]	InvoiceApplied	decimal	Allocated cash receipt detail amount per customer invoice.
[DataMember]	Invoiced	decimal	Total invoice amount.
[DataMember]	InvoiceLine	int	Invoice line number in cash receipt detail.
[DataMember]	InvoiceUnpaid	decimal	Customer invoice balance amount
[DataMember]	OffsettingPayment	Payment	Offsetting payment reference.
[DataMember]	Payer	Contact	Cash receipt payer.

Method usage:

```
List<SMS.Broker.DataContracts.Reports.AccountsReivable.ReportCashReceiptsSource> re
= RepManager.GetReportCashReceiptsSource(new DateTime(2025, 11, 1), DateTime.Today, customerCode, null, true);
```