

SMS Broker System
Software Integration

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

ABI classes and interfaces.

(ver. from 8/31/2025)

Table of Contents

1. Special Notes.	1
2. ABI Transmission.	1
3. ABI SMS Manager.	4
4. ABI Query.	6

1. Special Notes.

1. Data exchange classes and interfaces.

Classes and interfaces described in this document was designed to maintain the data exchange with Customs. SMS system allows to prepare documents, send them to Customs and receive Custom's responses. Usually (exception is for ABI query document) it looks the follows: You prepare document (Customs entry, ISF, drawback), save it and then put it to queue. As a result you create ABI transmission object that is ready to be transmitted to Customs. SMS system transmits it as a Customs application in standard Customs form described in CATAIR. When Customs in response sends incoming application it is converted to the response ABI transmission object on SMS server. Then you can get these ABI transmission(s) on client side.

2. ISmsABIManager.

[ISmsABIManager](#) includes put to queue methods for AMS and ERF Queries. In addition it gives more complex but the most flexible way to put to queue all Customs applications with different parameters. For more details, please see also *08. Annex. ABI classes and interfaces.pdf*.

3. ABI Query.

ABI query is a special type of a document. Please do not mix a term "ABI Query" on this context with a submission to Customs in general. Here it is one of Customs applications enumerated in CATAIR exactly as "Query" (AD/CVD Query, Census Warning Query and so on). Its peculiarity is that it is created as object not before putting ABI transmission to queue but while this operation is running. Other words, ABI query document is created and saved by the running of put to queue operation and from put to queue operation method parameters.

2. ABI Transmission.

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

SMS.Broker.DataContracts.Documents.[ABITransmission](#) : [EntityBase](#)

Class represents a transaction of data transmitting to and receiving from Customs.

Properties:

Attributes	Name	Type	Description
[Column(TypeName = "text")] [DataMember]	ABIMessage	string	"Raw" text of transmission in standard Customs format.
[DataMember] [StringLength(7)]	AppliactionId	string	Customs standard application code for document transmission.
[DataMember] [StringLength(500)]	Content	string	Text info that briefly characterize ABI transaction. For example "ERF- Foreign Ports ", Filer Code + Entry Number and so on.
[DataMember] [Column(TypeName = "text")]	Parameters	string	Parameters of ABI transmission.
[DataMember] [DateTimeUTC]	ProcessingTime	DateTime?	A time when a transmission was processed.
[DataMember]	ReadyToProcess	bool	In the case of outgoing ABI transmission true means that the message was queued for transmission to Customs (validation wasn't run yet). In the case of incoming ABI transmission true means that the "raw" message text was received from Customs and ready to be processed (validation also wasn't run yet).
[DataMember]	Request	ABITransmission	A reference to initial ABI transmission that was sent to Customs.
[DataMember]	Request_Id	long?	Id of Initial ABI transmission that was sent to Customs.
[DataMember]	Responses	List<ABITransmission>	A list of ABI transmissions – replies from Customs.
[DataMember] [StringLength(60)]	SourceLink	string	A link to the document for which a transaction was sent.
[DataMember] [StringLength(7)]	StatusSubject	string	Application Id of a "root" ABI transmission from which data exchange with Customs began for given document.

[Column(TypeName = "text")] [DataMember]	Tag	string	Any text data that can be added and used by API user.
[Column(TypeName = "text")] [DataMember]	Text	string	If TextType is ABITransmissionTextType .Error contains exception text.
[DataMember]	TextType	ABITransmissionTextType	Indicates if further processing of ABI transmission returns exception. If exceptions occurs property is set to ABITransmissionTextType .Error.
[DataMember] [DateTimeUTC]	Time	DateTime	Time when ABI transmission was sent/received.
[DataMember]	ToCustoms	bool	A direction of transmission (to or from Customs).
[DataMember]	User	User	User of session when ABI transmission was sent/received.
[DataMember]	User_Id	long	User session Id.

[ABITransmissionTextType](#) enumerator:
[ABITransmissionTextType](#).none,
[ABITransmissionTextType](#).Error

Instance methods:

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

Interface: [ServiceContract]
SMS.Broker.ServiceContracts.Documents.IABITransmissionManager :
IEntityManager<[ABITransmission](#)>

Instance methods:

Attributes	Name	Type	Description
[OperationContract] [FaultContract(typeof(ActionFault))]	ABIWebSphereMQPut(long abiTransmissionId)	void	Puts ABI transaction to MQ server queue.
[OperationContract] [FaultContract(typeof(ActionFault))]	Copy(long Id)	ABITransmission	Returns a copy of ABI transmission object by ABI transmission object's Id.
[OperationContract] [FaultContract(typeof(ActionFault))]	GetAnswers(long requestId)	List< ABITransmission >	Returns a list of response ABI

<code>ionFault))]</code>			transactions by request (outgoing) ABI transaction Id.
<code>[OperationContract] [FaultContract(typeof(ActionFault))]</code>	<code>GetBySourceLink(string link, bool includeABIMessage = true)</code>	<code>List<ABITransmission></code>	Returns a list of all (as outgoing as incoming) ABI transactions those were sent or received for some document by this document's link. If <code>includeABIMessage</code> is <code>true</code> returns object with <code>ABIMessage</code> property filled.
<code>[OperationContract] [FaultContract(typeof(ActionFault))]</code>	<code>Process(long abiTransmissionId, string applicationId)</code>	<code>void</code>	Process incoming ABI transactions by its Id and Customs application code.
<code>[OperationContract] [FaultContract(typeof(ActionFault))]</code>	<code>SwitchProcess(long Id)</code>	<code>bool</code>	Changes the status of <code>ReadyToProcess</code> property to opposite.

3. ABI SMS Manager.

Interface: `[ServiceContract]`
`SMS.Broker.ServiceContracts.ISmsABIManager`

Manager has a functionality related to Customs applications and putting to queue process.

Instance methods:

Attributes	Name	Type	Description
<code>[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))]</code>	<code>_PutToQueue_ExtractReferenceFile_FIRMSByBeginDate(DateTime beginDate)</code>	<code>long</code>	Puts to queue ABI transaction for Extract Reference File Query specially for FIRMS by beginning date.
<code>[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))]</code>	<code>_PutToQueue_ExtractReferenceFile_FIRMSByCode(string code)</code>	<code>long</code>	Puts to queue ABI transaction for Extract Reference File Query specially for FIRMS by its code.
<code>[OperationContract] [ServiceKnownType("GetKnownTypes", typeof(ABIAppKnownRecord))] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] [FaultContract(typeof(EntryValidationFault))]</code>	<code>GenerateApplication(string AppId, List<Parameter> parameters);</code>	<code>App</code>	Returns ABI Customs application with a text in related Customs standard format. Can be used to preview application data to be sent to Customs.

[OperationContract] [ServiceKnownType("GetKnownTypes", typeof(ABIAppKnownRecord))] [FaultContract(typeof(ActionFault))] 	GenerateApplicationWithKnownRecords(string AppId)	App	Returns ABI Customs application with no data (only application's complete structure). Can be used as memo items.
[OperationContract]	GetParametersSet(string AppId)	List<Parameter>	Returns Customs application parameters by its Id (standard Customs code).
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_CargoManifestEntryReleaseQuery_AirBill(string airMasterBOL, string airHouseBOL, string clientRef = null)	long	Puts to queue ABI transaction for Cargo Manifest/In-Bond/Entry Status Query specially for Airway Bill.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_CargoManifestEntryReleaseQuery_Bill(string SCAC, string billNumber, bool requestBOLEntryData, string clientRef = null)	long	Puts to queue ABI transaction for Cargo Manifest/In-Bond/Entry Status Query specially for Bill of Lading.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_CargoManifestEntryReleaseQuery_Entry(string entryFilerCode, string entryNumber, bool requestBOLEntryData, string clientRef = null)	long	Puts to queue ABI transaction for Cargo Manifest/In-Bond/Entry Status Query specially for entry.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_CargoManifestEntryReleaseQuery_InBond(string inBondNumber, string clientRef = null)	long	Puts to queue ABI transaction for Cargo Manifest/In-Bond/Entry Status Query specially for In-Bond.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_ExtractReferenceFile_Carriers()	long	Puts to queue ABI transaction for Extract Reference File Query specially for carriers.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_ExtractReferenceFile_Countries()	long	Puts to queue ABI transaction for Extract Reference File Query specially for countries.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_ExtractReferenceFile_CustomsPorts()	long	Puts to queue ABI transaction for Extract Reference File Query specially for Customs ports.
[OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(ActionFaults))] 	PutToQueue_ExtractReferenceFile_FIRMS()	long	Puts to queue ABI transaction for Extract Reference File Query specially for FIRMS.
[OperationContract] [FaultContract(typeof(ActionFault))] 	PutToQueue_ExtractReferenceFile_ForeignPorts()	long	Puts to queue ABI

<code>ionFault))] [FaultContract(typeof(Act ionFaults))]</code>			transaction for Extract Reference File Query specially for foreign ports.
<code>[OperationContract] [FaultContract(typeof(Act ionFault))] [FaultContract(typeof(Act ionFaults))] [FaultContract(typeof(Ent ryValidationFault))]</code>	<code>PutToQueue(string AppId, List<Parameter> parameters)</code>	<code>long</code>	Synchronous method that allows put ABI transactions to queue.
<code>[OperationContract(IsOneWay = true)]</code>	<code>PutToQueueOneWay(string AppId, List<Parameter> parameters)</code>	<code>void</code>	Asynchronous method that allows put ABI transactions to queue.

4. ABI Query.

Class: `[DataContract]`

`SMS.Broker.DataContracts.Documents.ABIQuery : EntityBase`

It is a special class of documents that contains a data that was sent as ABI query to Customs and received in ABI query result.

Properties:

Attributes	Name	Type	Description
<code>[DataMember]</code>	ABITransmission	<code>ABITransmission</code>	Outgoing (initial) ABI transmission related to query.
<code>[DataMember]</code>	ABITransmission_Id	<code>long?</code>	Outgoing (initial) ABI transmission Id.
<code>[DataMember] [StringLength(2)]</code>	AppliactionId	<code>string</code>	A related standard Customs application code.
<code>[DataMember] [StringLength(20)]</code>	ClientRef	<code>string</code>	Client reference – some info useful in document usage context.
<code>[DataMember] [StringLength(100)]</code>	Message	<code>string</code>	A name of query, for example "Cargo manifest entry release query".
<code>[DataMember]</code>	Statuses	<code>List<EntityStatus></code>	Collection of query statuses.
<code>[Column(TypeName = "text")] [DataMember]</code>	Tag	<code>string</code>	Any text data that can be added and used by API user.
<code>[DataMember] [StringLength(400)]</code>	Text	<code>string</code>	Text info about query request outgoing parameters.
<code>[DataMember] [DateTimeUTC]</code>	Time	<code>DateTime</code>	A time when query was created.
<code>[DataMember]</code>	Type	<code>ABIQueryType</code>	ABI query type. See

			ABIQueryType enumerator.
[DataMember]	User	User	A user of session when the query was created.
[DataMember]	User_Id	long	A user of session Id.
[DataMember] [StringLength(200)] [Obsolete("UserRemarkType, UserRemark", false)]	UserData	string	User data. Obsolete. Now UserRemarkType and UserRemark fields inherited from EntityDirectory base class are used.
[DataMember] [StringLength(200)]	UserRemark	string	User remark. Some info that is significant for user and can be added by him to document.
[StringLength(2)] [DataMember]	UserRemarkType	string	Type of user remark. Has Possible Values.

[ABIQueryType](#) enumerator:

```

ABIQueryType.ExtractReferenceCountries,
ABIQueryType.ExtractReferenceCarriers,
ABIQueryType.ExtractReferenceRegionDistrictPorts,
ABIQueryType.ExtractReferenceFIRMs,
ABIQueryType.ExtractReferenceForeignPort,
ABIQueryType.ExtractReferenceHTSRange,
ABIQueryType.ExtractReferenceHTSUpdateNumber,
ABIQueryType.ExtractReferenceZIPState,
ABIQueryType.CargoManifestEntryReleaseQuery,
ABIQueryType.EntrySummaryQuery,
ABIQueryType.AntidumpingCountervailingDutyQuery,
ABIQueryType.CensusWarningQuery,
ABIQueryType.StatementRequestReroute,
ABIQueryType.ImporterNameAndAddressQuery.

```

Interface: [[ServiceContract](#)]
 SMS.Broker.ServiceContracts.Documents.[IABIQueryManager](#) :
IEntityManager<ABIQuery>

Instance methods:

Attributes	Name	Type	Description
[OperationContract]	GetListByClientRef(string clientRef)	List < ABIQuery >	Returns a list of ABI queries by client reference.
[OperationContract] [Obsolete("SetUserRemark", false)]	SetUserData(long id, string data)	void	Obsolete. Sets UserData property to some text value. Use SetUserRemark() method instead.
[OperationContract]	SetUserRemark(long id, string type, string remark)	void	Set user remark to some type and value.