

SMS Broker System
Software Integration

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

Base, common usage classes and interfaces.

(ver. from 8/31/2025)

Table of Contents.

| | |
|---|----|
| 1. Special Notes. | 1 |
| 2. Base classes. | 2 |
| 2.1. Entity Base. | 2 |
| 2.2. Entity Directory. | 4 |
| 2.3. Entity Document. | 5 |
| 2.4. Entity Child. | 7 |
| 3. Sequences. | 7 |
| 4. Entity Event. | 8 |
| 5. Entity Link. | 10 |
| 6. Entity Page Query. | 10 |
| 7. Entity Last Status. | 11 |
| 8. Parameter. | 17 |
| 9. Possible Values. | 18 |
| 10. System Alert. | 18 |
| 11. Validation. | 20 |
| 11.1 Action Faults. | 20 |
| 11.1.1 Action Fault Entry. | 20 |
| 11.2 Entry Validation Fault. | 20 |
| 11.2.1 Entry Validation Error. | 21 |
| 11.2.1.1 Entry Validation Error Detail. | 22 |
| 12 User. | 22 |
| 12.1 User Properties. | 23 |

This document describes classes and interfaces those are used in all SMS-system subsystems.

1. Special Notes.

1. Base classes. All directory, document and child classes are derived from this classes. Properties and methods of base classes (if not overridden) are not described for derived classes. For example, Id property is absent in a table that describes CustomsEntry class because it is inherited from [EntityDocument](#) class that in turn is inherited from [EntityBase](#) class with Id property.

[EntityChild](#) class is used for collection objects those can't exist as independent entities, without their parent document or directory object. For example, [ShipmentArticle](#) has no a reason without parent [Shipment](#) object, [ContactBondInfo](#) has no a reason without parent [Contact](#) object and so on. In contrast to such

objects, Shipment class (it is derived from [EntityDocument](#) class) objects, for example, can be included as collection to CustomsEntry parent object but can exist also as independent entity. Respectively, [EntityChild](#) derived classes have no manager classes and accessible via parent object manager classes.

2. Possible Values. In the tables those describe Data Contract (Directory and Document) classes very often you can see “Has Possible Values” note in description column. It means that property can obtain values only from a standard Customs list. In SMS-system such Customs list is a list of [PossibleValue](#) structures. Each structure contains standard abbreviation (code) and its description. For example, **"CM2"** and **"Square Centimeters"**. The name of this collection is formed by the name of this property plus word “Values”. For example, [ShipmentArticleTariff](#) class has an `UnitOfMeasure1` property. A list of possible values name is `UnitOfMeasure1Values`.

3. System Alerts. System alerts is a way to show incoming message information to user in a time. For example, a work with customs entry document isn't finished when cargo release and duty payment process are finished. So-called liquidation process can still continue for a long time period. During this period Customs sends Entry Summary Status Notification and Courtesy Notice messages. “Old” but still not liquidated customs entries go to past and become non-visible for user who is usually interested in current entries. Respectively, “old” entry statuses and events become non-visible also. In the case if such late Customs message will come for “old” entry a user can miss it. To avoid this you can use system alerts. They can be displayed on a form until user confirm that he reviewed them.

3. Last Statuses and Events. Last statuses and events in SMS-system are used to record a current state and a history of a document (or directory) object. Last status object contains a number of enumerators those are helpful in a document program processing. Event objects among other properties contains free form text and is more user-oriented.

2. Base classes.

2.1. Entity Base.

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

SMS.Broker.DataContracts.[EntityBase](#): [IEntityLink](#), [IComparable](#)

Base class for all entities. Each instance of its derived classes becomes one record in a database.

Constructors:

| Name | Description |
|---------------------------|---|
| <code>EntityBase()</code> | EntityGuid is initialized by new unique Guid. |

Properties:

| Attributes | Name | Type | Description |
|---|-------------|----------------------------|---|
| [DataMember] | EntityGuid | string | It is used for data identification when it was not still saved to database (on this case <code>Id = 0</code>) and initialized in constructor. If <code>Id</code> was assigned it is set to null. |
| | EntityIsNew | bool | if <code>Id = 0</code> then true else false |
| | EntityLink | EntityLink | Entity link, serves as unique identifier. |
| [Key] [DataMember] | Id | Int64 | Primary key is auto-incremental. 0 value is for new entity. After entity will be saved to table the next auto-incremental value is assigned. There is no a need to set or change it. It is for internal system infrastructure. It can be used to get entity (database row) by primary key <code>Id</code> . |
| | Link | string | String representation of entity link. In derived classes looks like “Contact=14”, “CustomsEntry=55” and so on. |
| [DataMember] | RowVersion | byte[] | It is used to detect changes and prevent data |

| | | | |
|-------------|--|--|--|
| [Timestamp] | | | saving if source already was changed on database side. |
|-------------|--|--|--|

Methods:

| Attributes | Name | Type | |
|------------|---|--------|--|
| | CreateCollections() | void | Create all child collections. [1] |
| | GetProperty(string name) | object | Gets a value of a property. |
| | HasProperty(string name) | bool | Returns true if a property with the name exists. |
| | SetProperty(string name, object value) | void | Sets a property to some value. |
| | ToDetailtString() | string | Returns detailed text representation. |
| | ToShortString() | string | Returns short text representation. |
| | ToString() | string | Returns text representation. |
| | TrackPropertyChangesEvents(string propertyname) | bool | For internal use. |
| | TrackSaveChangesEvents() | bool | For internal use. |

[1] - For example for `ImporterSecurityFiling`

```
public override void CreateCollections()
{
    Shipments = new List<Shipment>();
    Commodities = new List<ImporterSecurityFilingCommodity>();
}
```

Interface: `[ServiceContract]`
SMS.Broker.ServiceContracts.IEntityManager<T> where T: class
Base interface for all entity managers.

Methods:

| Attributes | Name | Type | Description |
|--|--|-------------|---|
| [OperationContract] [FaultContract(typeof(ActionFault))] | Count() | int | Returns a number of entities in a database |
| [OperationContract] [FaultContract(typeof(EntryUpdateFault))] [FaultContract(typeof(ActionFault))] | Delete(T entity) | void | Deletes an entity from a database. |
| [OperationContract] [FaultContract(typeof(ActionFault))] | Get(long Id, string include, string option = null) | T | Gets an entity by primary key Id. "include" is a string that contains navigation property names separated by commas. (See Entity Framework <i>include</i>) [1] |
| [OperationContract] [FaultContract(typeof(ActionFault))] | GetEntityType() | Type | Gets Type of T |
| [OperationContract] | GetEvents(long id, bool | List<Entity | Get a list of entity |

| | | | |
|--|---|-------------------|---|
| [FaultContract(typeof(ActionFault))] | includeText = true) | Event> | events by entity Id. |
| [OperationContract] [ServiceKnownType(typeof(int[]))] [ServiceKnownType(typeof(long[]))] [ServiceKnownType(typeof(string[]))] [ServiceKnownType(typeof(decimal[]))] [ServiceKnownType(typeof(DateTime[]))] [ServiceKnownType(typeof(EntityEventType))] [FaultContract(typeof(ActionFault))] | GetPage(EntityPageQuery pageQuery) | EntityPage< T> | Allows to get some list of entities. pageQuery – parameters for sampling. |
| [OperationContract] [FaultContract(typeof(ActionFault))] | New(long? sourceId) | T | Creates a new entity with new child collections. Their Count property is zero. source Id is an Id of a source or null. |
| [OperationContract] [FaultContract(typeof(EntryValidationFault))] [FaultContract(typeof(EntryUpdateFault))] [FaultContract(typeof(ActionFault))] | Save(T entity) | T | Saves to database. Returns saved entity. If Id is 0 then result has an Id for added row. |
| [OperationContract] [FaultContract(typeof(EntryValidationFault))] [FaultContract(typeof(ActionFault))] | ValidateTreeById(long id, string action) | void | [2] |

[1] – **string** option parameter in Get() method is for internal use. If you program on API level only don't use this parameter.

[2] – Validates entity and related object for some actions.

May be some specific action.

For example, for ISF it can be "QUE SF" – it means that the entity will be validated with all its collections needed to be send to Customs. (ISF with related Shipments, Contacts, Commodities and etc.)

2.2. Entity Directory.

Class: [DataContract]

SMS.Broker.DataContracts.EntityDirectory: EntityBase

Base class for all directories (Contact, Carrier, CountryInfo, User etc.).

Constructors:

| Name | Description |
|-------------------|-------------|
| EntityDirectory() | |

Properties:

| Attributes | Name | Type | Description |
|--|-------------------|--------|--|
| [Required] [StringLength(10)] [DataMember] | Code | string | Usually unique code for item (entity) in a directory. SCAC code, FIRMS code and so on. |
| [DataMember] | IsMarkedForDelete | bool | For future use |
| [DataMember] | IsNotActual | bool | For future use |
| [DataMember] | Name | string | Name of directory item. |
| [DataMember] [StringLength(200)] | UserRemark | string | Some user text. |
| [StringLength(2)] [DataMember] | UserRemarkType | string | Type of remark. Has Possible values. |

Methods:

| Attributes | Name | Type | |
|------------|------------|--------|--|
| | ToString() | string | Returns text representation of entity. |

Interface:

[ServiceContract]

SMS.Broker.ServiceContracts.IEntityManagerDirectory<T> : IEntityManager<T>

where T : EntityDirectory

Base interface for all directory entity managers.

Methods:

| Attributes | Name | Type | Description |
|--|---|---------|--|
| [OperationContract] [FaultContract(typeof(ActionFault))] | GetByCode(string code, string include) | T | Gets a directory entity by Code field. "include" is a string that contains navigation property names separated by commas. (See Entity Framework include) |
| [OperationContract] [FaultContract(typeof(ActionFault))] | GetByName(string name, SMS.Broker.Common.StringSearchType searchType, string include) | List<T> | Gets a directory entity by Name field. searchType is an enum value (FullMatch, StartsWith or Contains), "include" is a string that contains navigation property names separated by commas. (See Entity Framework include) |
| [OperationContract] [FaultContract(typeof(ActionFault))] | SetUserRemark(long id, string type, string remark) | void | Set user remark to directory entity by its Id. remark – some text, type – type of remark. |
| [OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | UnVoid(long id) | void | Mark void directory entity as non-void by its Id. |
| [OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | Void(long id) | void | Mark directory entity as void by its Id. |

2.3. Entity Document.**Class:** [DataContract]

SMS.Broker.DataContracts.EntityDocument: EntityBase

Base class for all documents (Customs entry, ISF, In-bonds etc).

Constructors:

| Name | Description |
|------------------|-------------|
| EntityDocument() | |

Properties:

| Attributes | Name | Type | Description |
|---|----------------------|----------|---|
| [DataMember] | Creator | User | Person who created a document. |
| [DataMember] | Creator_Id | long? | User_Id of a person who created a document. |
| [Required] [DataMember] [Column(TypeName = "date")] | Date | DateTime | Date of document (only date without time stamp) |
| [DataMember] | InProcess | bool | For future use |
| [DataMember] | IsMarkedForDelete | bool | For future use |
| [DataMember] | IsNotActual | bool | For future use |
| [Required] [DataMember] [Index(true)] | Number | long | Unique number for document - Customs entry, ISF, In-bond etc. It can be adjusted by related Sequence. |
| [DataMember] | ResponsiblePerson | User | User_Id of a person who is responsible for a document. |
| [DataMember] | ResponsiblePerson_Id | long? | Person who is responsible for a document. |
| [DataMember] [StringLength(200)] | UserRemark | string | Some user text. |
| [StringLength(2)] [DataMember] | UserRemarkType | string | Type of remark. Has Possible values. |

Methods:

| Attributes | Name | Type | |
|------------|------------|--------|--|
| | ToString() | string | Returns text representation of entity. |

Interface:

[ServiceContract]

SMS.Broker.ServiceContracts.IIEntityManagerDocument<T> : IEntityManager<T>

where T : EntityDocument

Base interface for all document entity managers.

Methods:

| Attributes | Name | Type | Description |
|--|--|------|---|
| [OperationContract] [FaultContract(typeof(ActionFault))] | GetByNumber(long Number); | T | Gets a document by its Number. |
| [OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | Post(long id) | void | Post document. |
| [OperationContract] [FaultContract(typeof(ActionFault))] | SetUserRemark(long id, string type, string remark) | void | Set user remark to document by its Id. remark – some text, type – type of remark. |

| | | | |
|--|-----------------|------|--|
| [OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | UnPost(long id) | void | Undo document posting. |
| [OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | UnVoid(long id) | void | Mark void document entity as non-void by its Id. |
| [OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | Void(long id) | void | Mark document entity as void by its Id. |

2.4. Entity Child.

Class: [DataContract]

SMS.Broker.DataContracts.EntityChild: EntityBase

Base class for all "dependent" child classes (see Special Notes, 1) such as [ContactBankAccount](#), [ShipmentArticleTariff](#), [ImporterSecurityFilingCommodity](#), [ContactBondInfo](#) etc.

Constructors:

| Name | Description |
|---------------|-------------|
| EntityChild() | |

Properties:

| Attributes | Name | Type | Description |
|--------------|----------------|-------------------------------------|-----------------------------------|
| [DataMember] | Line | int | Sequence number of a child entity |
| [NotMapped] | RootEntityLink | SMS.Broker.DataContracts.EntityLink | EntityLink of a root parent |

Methods:

| Attributes | Name | Type | |
|------------|--|--------|--|
| | Down(IEnumerable<EntityChild> list) | void | Changes the order of current and next child entities in a list sorted by line number. Current child entity acquires the line number of the next child entity and vise versa. |
| | SetNextLine(IEnumerable<EntityChild> list) | void | Set Line property to the next value (Maximum line number in the list + 1) |
| | ToString() | string | Returns "Line #: " + Line |
| | Up(IEnumerable<EntityChild> list) | void | Changes the order of current and previous child entities in a list sorted by line number. Current child entity acquires the line number of the previous child entity and vise versa. |

3. Sequences.

Class: [DataContract]

SMS.Broker.DataContracts.Sequence: EntityDirectory

Sequences allow to generate the next number for some sequence name.

Constructors:

| Name | Description |
|------|-------------|
|------|-------------|

| | |
|------------|--|
| Sequence() | |
|------------|--|

Properties:

| Attributes | Name | Type | Description |
|---|---------|--------|--|
| [Required] [StringLength(80)] [DataMember] | Code | string | Code ("Shipment", "CustomsEntry" etc.) |
| [DataMember] | Current | Int64 | Last generated number |
| [Required] [StringLength(120)] [DataMember] | Name | string | Name |

Class: SMS.Broker.Common.KnownSequences

Class contains the name of known sequences as a property.

```
public const string Shipment = "Shipment";
public const string ImporterSecurityFiling = "ImporterSecurityFiling";
public const string CustomsEntry = "CustomsEntry";
public const string DeliveryOrder = "DeliveryOrder";
public const string PGA = "PGA";
public const string DIS = "DIS";
public const string InBond = "InBond";
public const string Drawback = "Drawback";
```

These sequences are used to assign the next number of related document.

Example:

```
var manager entity =
ClientContext.ServicesFactory.GetManager<IImporterSecurityFilingManager>();
var entity = manager.New(null); // Manager returns the new isf object with property
// "Number" filed with the next number of used sequence
// ("ImporterSecurityFiling"). "Current" property of
// sequence will become equal to this number.
```

4. Entity Event.

Class: [DataContract]

SMS.Broker.DataContracts.Common.EntityEvent: EntityBase

This class is used to handle some events.

Properties:

| Attributes | Name | Type | Description |
|-------------------------------------|--------------------|-----------------|--|
| [DataMember] | ABITransmission | ABITransmission | Related transmission (if exists) |
| [DataMember] | ABITransmission_Id | long? | Related transmission Id |
| [StringLength(200)] [DataMember] | Message | string | Event subject message. |
| [StringLength(60)] [DataMember] | ReferenceLink1 | string | Link to related entities or some specific info |
| [StringLength(60)] [DataMember] | ReferenceLink2 | string | Link to related entities or some specific info |
| [StringLength(60)] [DataMember] | ReferenceLink3 | string | Link to related entities or some specific info |
| [StringLength(60)] [DataMember] | SourceLink | string | Link of entity which status belongs to |

| | | | |
|---|---------------|-------------------|--|
| [DataMember] [Obsolete] | Status | EntityStatus | Obsolete. Related status. |
| [DataMember] [Obsolete] | Status_Id | long? | Obsolete. Related status Id. |
| [StringLength(50)] [DataMember] | StatusSubject | string | Status subject of related last status actual at the time of transaction. |
| [DataMember] | StatusType | EntityType | Type of related last status actual at the time of transaction |
| [DataMember] | StatusValue | EntityStatusValue | Value of related last status actual at the time of transaction |
| [Column(TypeName = "text")] [DataMember] | Text | string | Event text |
| [DataMember] | Time | DateTime | Time of event |
| [DataMember] | Type | EntityType | Type of status [1] |
| [DataMember] | User | User | Initiator – User |
| [DataMember] | User_Id | long | Initiator – User Id |

[1] **EntityType** enumerator:

EntityType.Added – object added to database,
EntityType.Modified – object modified on database,
EntityType.Deleted – object deleted from database,
EntityType.Info – information event,
EntityType.ABI – related to Customs ABI,
EntityType.Status – some status was changed,
EntityType.Warning – warning event,
EntityType.User – user event.

Interface: [ServiceContract]
 SMS.Broker.ServiceContracts.Common.I**EventManager**:
IEventManager<EntityEvent>
 Designed to work with events. See Operation Contracts of base classes.

Methods:

| Attributes | Name | Type | Description |
|--|---|-------------------|--|
| [OperationContract] [FaultContract(typeof(ActionFault))] | GetABIEvents(string link, bool includeText = true) | List<EntityEvent> | Gets an event of EntityType.ABI type by link of object. If includeText is true returns event object with Text property filled. |
| [OperationContract] [FaultContract(typeof(ActionFault))] | AddEvent(string link, EntityType type, string message, string text, string ref1, string ref2, string ref3, long? ABITransmissionId = null) | void | Add event. |
| [OperationContract] [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | Void(long id) | void | Make event void. |
| [OperationContract] | UnVoid(long id) | void | Undo void operation. |

| | | | |
|---|--|--|--|
| [FaultContract(typeof(ActionFault))] [FaultContract(typeof(EntryValidationFault))] | | | |
|---|--|--|--|

5. Entity Link.

Class: [DataContract]

SMS.Broker.DataContracts.EntityLink: IEntityLink

It is a special class that identifies one data record.

Constructors:

| Name | Description |
|--|---|
| EntityLink(string entityTypeName, long id) | Initializes a new instance of the EntityLink class. |
| EntityLink(Type entityType, long id) | Initializes a new instance of the EntityLink class. |
| EntityLink(string entityTypeName, string guid) | Initializes a new instance of the EntityLink class. |
| EntityLink(string link) | Initializes a new instance of the EntityLink class. |

Properties:

| Attributes | Name | Type | Description |
|--------------|----------------|--------|--|
| | EntityType | Type | Type of entity |
| [DataMember] | EntityTypeName | string | Type name of entity. |
| [DataMember] | Guid | string | Is used if Id = 0. [1] |
| [DataMember] | Id | long | Entity Id. 0 for new entity (entity that was not saved to database yet) [1] |
| | Link | string | IEntityLink.Link [1] Realization of IEntityLink interface Link property. |

[1] – delimiter is “=”

if Id is 0

Link = EntityTypeName + delimiter + Guid

else

Link = EntityTypeName + delimiter + Id.ToString()

Example:

“Contact=55” // The record of Contacts table where Id = 55

“Contact=e661a76f-a63f-4b99-994c-5e32db2c911f” // A future record of Contacts table if Id = 0 (when it was // not saved yet).

“ImporterSecurityFiling=88” // The record of ImporterSecurityFilings table where Id = 88

6. Entity Page Query.

Class: [DataContract]

SMS.Broker.DataContracts.EntityPageQuery

Class is used for entities sampling.

Properties:

| Attributes | Name | Type | Description |
|------------|------|------|-------------|
|------------|------|------|-------------|

| | | | |
|--------------|-----------------|-----------------|--|
| [DataMember] | Criteria | string | Filter criteria. [3] |
| [DataMember] | Include | string | Contains navigation property names separated by commas. It is used to get object's properties and collections those are not returned by default. |
| [DataMember] | Navigation | NavigationType | Default – Refresh [1] |
| [DataMember] | Option | string | Specific options for some managers. |
| [DataMember] | Order | List<OrderItem> | Allows to order items. [2] |
| [DataMember] | PageSize | int | Maximum elements that will be returned. 40 elements is default. |
| [DataMember] | Parameters | List<Parameter> | Specific parameters for some managers. |
| [DataMember] | Position | int | Position of the first element. Default value is 0. |
| [DataMember] | QueryTotalCount | bool | Calculates and returns a count of all available elements. Requires more time for execution. Default value is true. |
| [DataMember] | RequestFullList | bool | Request a full list. For a big data the timeout exception can be raised. Default – false. |

[1] enumerator – [NavigationType](#)

[NavigationType.Refresh](#) – returns a list starting with q.Position,
[NavigationType.First](#) – returns the first page list starting with the first record,
[NavigationType.Previous](#) – returns the previous page starting with q.Position - q.PageSize,
[NavigationType.Next](#) – returns the next page starting with q.Position + q.PageSize,
[NavigationType.Last](#) – returns the last page list starting with the last record - q.PageSize.

[2] [List<OrderItem>](#) is a list of properties and sort orders.

[OrderItem.Name](#) – name of property to be ordered.
[OrderItem.IsDesc](#) – sort order: [false](#) for ascending, [true](#) for descending orders.

7. Entity Last Status.

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

[SMS.Broker.DataContracts.Common.EntityLastStatus](#): [EntityBase](#)

This class is used to handle last statuses of entities (documents, directories).

Properties:

| Attributes | Name | Type | Description |
|--------------|--------------------|---------------------------------|--|
| [DataMember] | ABIQuery | ABIQuery | ABI query data |
| [DataMember] | ABIQuery_Id | long? | ABI query Id if this status belongs to ABI query else null |
| [DataMember] | ABITransmission | ABITransmission | ABI transmission data |
| [DataMember] | ABITransmission_Id | long? | ABI transmission Id if this status belongs to ABI transmission else null |
| [DataMember] | BreakBulk | BreakBulk | Breakbulk data |

| | | | |
|------------------------------|---------------------------|------------------------|--|
| [DataMember] | BreakBulk_Id | long? | Breakbulk Id if this status belongs to Breakbulk else null |
| [DataMember] | CashReceipt | CashReceipt | Cash Receipt data |
| [DataMember] | CashReceipt_Id | long? | Cash Receipt Id if this status belongs to Cash Receipt else null |
| [DataMember] | Contact | Contact | Contact data |
| [DataMember] | Contact_Id | long? | Contact Id if this status belongs to contact else null |
| [DataMember] | CustomerInvoice | CustomerInvoice | Customer invoice data |
| [DataMember] | CustomerInvoice_Id | long? | Customer invoice Id if this status belongs to Customer invoice else null |
| [DataMember] | CustomsEntry | CustomsEntry | Customs entry data |
| [DataMember] | CustomsEntry_Id | long? | Customs entry Id if this status belongs to customs entry else null |
| [DataMember] | DIS | DIS | DIS data |
| [DataMember] [DataMember] | DIS_Id | long? | DIS Id if this status belongs to DIS else null |
| [DataMember] | Drawback | Drawback | Drawback data |
| [DataMember] | Drawback_Id | long? | Drawback Id if this status belongs to drawback else null |
| [DataMember] | EBondQuery | EBondQuery | e-Bond query data |
| [DataMember] | EBondQuery_Id | long? | e-Bond query Id if this status belongs to e-Bond query else null |
| [DataMember] | DeliveryOrder | DeliveryOrder | Delivery order data |
| [DataMember] | DeliveryOrder_Id | long? | Delivery order Id if this status belongs to Delivery order else null |
| [DataMember] | FTZe214 | FTZe214 | FTZ admission data |
| [DataMember] | FTZe214_Id | long? | FTZ admission data Id if this status belongs to FTZ admission data else null |
| [DataMember] | ImporterSecurityFiling | ImporterSecurityFiling | Importer Security Filing data |
| [DataMember] | ImporterSecurityFiling_Id | long? | ISF Id if this status belongs to ISF else null |
| [DataMember] | InBond | InBond | In-bond data |
| [DataMember] | InBond_Id | long? | In-bond Id if this status belongs to In-bond else null |
| [DataMember] | InBondBill | InBondBill | In-bond bill data |
| [DataMember] | InBondBill_Id | long? | In-bond bill Id if this status belongs to In-bond bill else null |
| [DataMember] | InBondUpdate | InBondUpdate | In-bond update data |
| [DataMember] | InBondUpdate_Id | long? | In-bond update Id if this status belongs to In-bond update else null |
| [DataMember] | Payment | Payment | Payment data |
| [DataMember] | Payment_Id | long? | Payment Id if this status belongs to Payment else null |
| [DataMember] | ReconEntry | ReconEntry | Reconciliation entry data |

| | | | |
|-----------------------------------|-------------------|----------------------|--|
| [DataMember] | ReconEntry_Id | long? | Reconciliation entry Id if this status belongs to Reconciliation entry else null |
| [DataMember] | Shipment | Shipment | Shipment data |
| [DataMember] | Shipment_Id | long? | Shipment Id if this status belongs to shipment else null |
| [DataMember] | Source | string | Source |
| [DataMember] | StatementDaily | StatementDaily | Daily statement data |
| [DataMember] | StatementDaily_Id | long? | Daily statement Id if this status belongs to daily statement else null |
| [DataMember] | Status | EntityStatusType | Type of status [1] |
| [IgnoreDataMember] [NotMapped] | StatusEntity | EntityBase | Return the entity related to that status (ISF, CE, In-bond, etc) |
| [IgnoreDataMember] [NotMapped] | StatusEntityLink | string | String link to StatusEntity object. |
| [DataMember] | Subject | string | Subject |
| [DataMember] | SystemAlert | SystemAlert | System alert data. |
| [DataMember] | SystemAlert_Id | long? | System alert Id if this status belongs to ABI query else null |
| [DataMember] | Text | string | Text of status [3] |
| [DataMember] | TextType | EntityStatusTextType | Text type of property Text [3] |
| [DataMember] | Time | DateTime | Time of status |
| [DataMember] | User | User | Initiator – User |
| [DataMember] | User_Id | long | Initiator – User Id |
| [DataMember] | Value | EntityStatusValue | Value of status [2] |
| [DataMember] | Voucher | Voucher | Voucher data |
| [DataMember] | Voucher_Id | long? | Voucher Id if this status belongs to Voucher else null |

Methods:

| Attributes | Name | Type | Description |
|------------|------------------|--------|--|
| | StatusInfo() | string | Returns info about status. Time.ToString() + " [" + User.Name + "]" + ToShortString(); |
| | ToDetailString() | string | Returns detailed object's text representation. Time.ToString() + " [" + User.Name + "]" + ToShortString() + "(" + Id.ToString() + ")" |
| | ToShortString() | string | Returns short object's text representation. |
| | ToString() | string | Returns object's text representation. |

[1] enumerator – **EntityStatusType**:

[EnumMember(Value = "None")]
None = 0,

[EnumMember(Value = "ABI App QUE status")]
ABIAppQUE = 1
Status for QUE process ("Put to QUE", "Sent", "Accepted", etc),

[EnumMember(Value = "ABI App Customs status")]
ABIAppCustoms = 2
Status of a document on Customs side ("No in customs", "Accepted in Customs", "Bill on file", etc),

[EnumMember(Value = "Document calculated")]
DocumentCalculated = 3
Status of documents ("Yes", "No").

[2] enumerator – **EntityStatusValue**

[EnumMember(Value = "None")]
None = 0,

EntityStatusType.ABIAppQUE:

[EnumMember(Value = "Put to QUE")]
ABIAppPutToQUE = 1,

[EnumMember(Value = "Sent")]
ABIAppSent = 2,

[EnumMember(Value = "Rejected")]
ABIAppRejected = 5,

[EnumMember(Value = "Put to QUE Error")]
ABIAppPutToQUEError = 6,

[EnumMember(Value = "Incoming error")]
ABIAppIncomingError = 7,

[EnumMember(Value = "Incoming")]
ABIAppIncoming = 8,

[EnumMember(Value = "Removed from QUE")]
ABIAppRemoveFromQUE = 9,

[EnumMember(Value = "QUE in process")]
ABIAppQUEInProgress = 10,

EntityStatusType.ABIAppCustoms:

[EnumMember(Value = "Accepted in Customs")]
ABIAppCustomsAccepted = 51,

[EnumMember(Value = "Accepted with warning in Customs")]
ABIAppCustomsAcceptedWithWarning = 52,

[EnumMember(Value = "Bill on file")]
ABIAppCustomsBillOnFile = 53,

[EnumMember(Value = "Deleted in customs")]
ABIAppCustomsDeletedInCustoms = 54,

[EnumMember(Value = "No bill on file")]
ABIAppCustomsNoBillOnFile = 55,

[EnumMember(Value = "Expired")]
ABIAppCustomsExpired = 56,

[EnumMember(Value = "Cargo released")]
ABIAppCustomsCargoReleased = 57,

[EnumMember(Value = "In process")]
ABIAppCustomsInProgress = 58,

[EnumMember(Value = "Preliminary")]
ABIAppCustomsPreliminary = 59,

[EnumMember(Value = "Authorized")]
ABIAppCustomsAuthorized = 60,

[EnumMember(Value = "Final")]
ABIAppCustomsFinal = 61,

[EnumMember(Value = "Add pending")]
ABIAppAddPending = 62,

[EnumMember(Value = "Delete pending")]
ABIAppDeletePending = 63,

[EnumMember(Value = "Yes")]
Yes = 100,

[EnumMember(Value = "No")]
No = 101.

[3] enumerator – [EntityStatusTextType](#):

[EntityStatusTextType](#).None –
[EntityStatusTextType](#).Error –
[EntityStatusTextType](#).EntryValidationFault –

any text,
error (unexpected error, usually `Exception.Message`),
It is used when we put document to QUE (send it to
Customs) using call (`OperationContract(IsOneWay =
true)`). Server will validate the entity and if it is not valid
this [EntityStatusTextType](#) is used. Text property will contain
[FaultException<EntryValidationFault>](#) object serialized by
[DataContractSerializer](#), object can be deserialized and
shown to user.

Sample:

Class [ImporterSecurityFiling](#) has a property

```
[DataMember]  
[InverseProperty("ImporterSecurityFiling")]  
public virtual List<EntityLastStatus> LastStatuses { get; set; }
```


Class: SMS.Broker.DataContracts.Common.EntityLastStatusExtensions

This class contains a set of extension methods.

Static methods:

| Attributes | Name | Type | Description |
|------------|---|------------------|--|
| | Status(this <code>List<EntityLastStatus></code> statuses, <code>EntityStatusType</code> type, <code>params string[]</code> subjects) | EntityLastStatus | Returns last status object by its type (Status property) and one or more subject(s) from collection of last statuses. |
| | Status(this <code>List<EntityLastStatus></code> statuses, <code>EntityStatusType</code> type, <code>EntityBase</code> entity = <code>null</code> , <code>params string[]</code> subjects) | EntityLastStatus | Returns last status object by its type (Status property), entity and one or more subject(s) from collection of last statuses. |
| | Status(this <code>List<EntityLastStatus></code> statuses, <code>EntityStatusType</code> type, <code>EntityLink</code> elink = <code>null</code> , <code>params string[]</code> subjects) | EntityLastStatus | Returns last status object by its type (Status property), entity link and one or more subject(s) from collection of last statuses. |
| | ToString(this <code>List<EntityLastStatus></code> statuses, <code>EntityStatusType</code> type, <code>TimeZoneInfo</code> tz, <code>params string[]</code> subjects) | string | Returns StatusInfo() <code>string</code> values separated by " <code>\r\n</code> " of last statuses collection selected by their type (Status property) and one or more subject(s) for some time zone. |

Sample:

```
statuses.ToString(type, SMS.Broker.ClientContext.ClientTimeZone, subjects);
```

Where statuses is a `List<EntityLastStatus>` collection.

On the table below you can see a sequence in which status values of a document/directory are changed while data exchange. This part of status values is common for all documents/directories those are sent to Customs. All these values have one and the same status type – `EntityStatusType.ABIAppeQUE`. The further data exchange suggests another status type - `EntityStatusType.ABIAppeCustoms` and another status values those are special for different documents. To see them please look at the description of last status objects for each document/directory type (ISF, Contact and so on).

Outgoing File – Any Customs incoming (our outgoing) file.

Incoming File – Any Customs outgoing (our incoming) file.

The actions which suggest some workflow problem are marked in red.

Table 5

| Action | Status | Value |
|--------------------------------------|--|---|
| Outgoing File Put to queue→ | <code>EntityStatusType.ABIAppeQUE</code> | <code>EntityStatusValue.ABIAppePutToQUE</code> |
| Outgoing File ←Put to queue error | <code>EntityStatusType.ABIAppeQUE</code> | <code>EntityStatusValue.ABIAppePutToQUEError</code> |
| Outgoing File ←Remove (if necessary) | <code>EntityStatusType.ABIAppeQUE</code> | <code>EntityStatusValue.ABIAppeRemoveFromQUE</code> |
| Outgoing File Transmit→ | <code>EntityStatusType.ABIAppeQUE</code> | <code>EntityStatusValue.ABIAppeSent</code> |
| Incoming File ←Get file | <code>EntityStatusType.ABIAppeQUE</code> | <code>EntityStatusValue.ABIAppeIncoming</code> |

| | | |
|-------------------------------|----------------------------|---------------------------------------|
| Incoming File ←Get with error | EntityStatusType.ABIAppQUE | EntityStatusValue.ABIAppIncomingError |
| Incoming File ←Rejected | EntityStatusType.ABIAppQUE | EntityStatusValue.ABIAppRejected |

Interface: [ServiceContract]
SMS.Broker.ServiceContracts.Common.IEntityStatusManager:
IEntityManager<EntityStatus>
Designed to work with statuses. See Operation Contracts for base classes.

8. Parameter.

Class: [DataContract]
SMS.Broker.DataContracts.Parameter

Represents some parameter. In general case its can be used in different methods, in particular in put to queue operation methods.

Properties:

| Attributes | Name | Type | Description |
|--------------|-----------------------|------------|--|
| [DataMember] | Description | string | Parameter description. |
| [DataMember] | Expression | string | Expression (for example, "DateTime.Today") that can be set instead of Value setting. |
| | IsNull | bool | Returns true if parameter value is null; |
| [DataMember] | Name | string | Parameter name. |
| | Type | Type | Parameter type. |
| [DataMember] | TypeName | string | Has possible values. |
| [DataMember] | Value | object | Parameter value. |
| | ValueAsArrayOfLong | long[] | Parameter value as Int64 array. |
| | ValueAsBoolean | bool | Parameter value as boolean. |
| | ValueAsDateTime | DateTime | Parameter value as DateTime. |
| | ValueAsDateTimeOrNull | DateTime? | Parameter value as DateTime?. |
| | ValueAsEntityBase | EntityBase | Parameter value as EntityBase. |
| | ValueAsInt64 | long | Parameter value as Int64. |
| | ValueAsString | string | Parameter value as string. |

Instance methods:

| Attributes | Name | Type | Description |
|------------|------|------|-------------|
|------------|------|------|-------------|

| | | | |
|--|------------|--------|---------------------------------------|
| | ToString() | string | Returns object's text representation. |
|--|------------|--------|---------------------------------------|

Static methods:

| Attributes | Name | Type | Description |
|------------|--|-----------------|---|
| | Parameters(string names, params object[] values) | List<Parameter> | Converts names and values to a list of parameters. string names contains parameter names separated by commas (for example "Customer, Checked"). |

9. Possible Values.

Struct: SMS.Broker.Common.PossibleValue

Structure that represents possible value.

Constructors:

| Name | Description |
|---|---|
| PossibleValue(object value, string description) | Initializes a new instance of the PossibleValue |

Properties:

| Attributes | Name | Type | Description |
|--------------|-------------|--------|--------------------------------|
| [DataMember] | Description | string | Description of possible value. |
| [DataMember] | Value | object | Possible value. |

Instance methods:

| Attributes | Name | Type | Description |
|------------|------------|--------|---|
| | ToString() | string | Returns the value with its description. |

10. System Alert.

Class: [DataContract]

SMS.Broker.DataContracts.Common.SystemAlert : EntityBase

Properties:

| Attributes | Name | Type | Description |
|--------------|------------|---------|---------------------|
| [DataMember] | Contact | Contact | Related contact. |
| [DataMember] | Contact_Id | long? | Related contact Id. |

| | | | |
|---|----------------------------|-----------|--|
| [DataMember] [StringLength(20)] | DocumentType | string | Specific document type. Has Possible Values. |
| [DataMember] | Files | string | A name or file attached to alert. |
| [DataMember] [StringLength(200)] | Message | string | Alert subject text. |
| [DataMember] | ReadingTime | DateTime? | Reading (confirmation) time. |
| [DataMember] | ResponsiblePerson | User | User who confirmed the alert. |
| [DataMember] | ResponsiblePerson_Id | long? | Id of a user who confirmed the alert.. |
| [DataMember] [StringLength(60)] | SourceLink | string | Entity link of a source document string representation. |
| [DataMember] | SourceResponsiblePerson | User | Responsible person of related document. |
| [DataMember] | SourceResponsiblePerson_Id | long? | Responsible person of related document Id. |
| [Column(TypeName = "text")] [DataMember] | Text | string | Alert message text. |
| [DataMember] [DateTimeUTC] | Time | DateTime | Time when alert was received. |
| [DataMember] [StringLength(20)] | Type | string | Alert type. Has Possible Values. |
| [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 | Has Possible Values. |

Interface: [ServiceContract]
SMS.Broker.ServiceContracts.Common.ISystemAlertManager :
IEntityManager<SystemAlert>

Instance methods:

| Attributes | Name | Type | Description |
|---------------------|--|-------------------|--|
| [OperationContract] | Confirm(long id) | void | Confirms an alert. |
| [OperationContract] | GetNotConfirmedSystemAlerts(string type) | List<SystemAlert> | Returns alerts those were not confirmed yet. |
| [OperationContract] | SetUserRemark(long id, string type, string remark) | void | Set user remark to some type and value. |

11. Validation.

11.1. Action Faults.

Class: [DataContract]

SMS.Broker.Faults.ActionFaults

Used for actions not related to entity. Thrown if some action has a validation problem.

Constructors:

| Name | Description |
|----------------|-------------|
| ActionFaults() | |

Properties:

| Attributes | Name | Type | Description |
|--------------|--------|------------------------|---------------------------|
| [DataMember] | Faults | List<ActionFaultEntry> | List of validation errors |

Instance methods:

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

11.1.1 Action Fault Entry.

Class: [DataContract]

SMS.Broker.Faults.ActionFaultEntry

Properties:

| Attributes | Name | Type | Description |
|--------------|------------|--------|------------------------|
| [DataMember] | ErrorCode | string | Error code. |
| [DataMember] | Message | string | Error message. |
| [DataMember] | Parameters | string | Parameters with error. |
| [DataMember] | Subject | string | Subject. |

11.2. Entry Validation Fault.

Class: [DataContract]

SMS.Broker.Faults.EntryValidationFault

Thrown if some entity has a validation problem.

Constructors:

| Name | Description |
|---|-------------|
| EntryValidationFault() | |
| EntryValidationFault(List<EntryValidationError> errors) | |

Properties:

| Attributes | Name | Type | Description |
|--------------|-----------------------|----------------------------|---------------------------|
| [DataMember] | EntityLinkForEditForm | string | Link of root entity [1] |
| [DataMember] | EntryValidationErrors | List<EntryValidationError> | List of validation errors |

[1] For example, if Shipment has a validation error then EntityLinkForEditForm can be a link to

Shipment class object

CustomsEntry class object

ImporterSecurityFilling class object

- if it calls from shipment edit form.

- if it calls from customs entry edit form.

- if it calls from ISF edit form.

Methods:

| Attributes | Name | Type | Description |
|------------|--|----------------------|--|
| | GetEntityError(string link) | EntryValidationError | Returns EntryValidationError from collection for some entity by its link. If it was not found then null. |
| | Throw(List<EntryValidationError> errors) | void | Rise new FaultException<EntryValidationFault> with errors. |
| | ToString() | string | Returns the text representation of all errors (EntryValidationErrors objects collection). |

11.2.1 Entry Validation Error.

Class: [DataContract]

SMS.Broker.Faults.EntryValidationError

Constructors:

| Name | Description |
|--|-------------|
| EntryValidationError() | |
| EntryValidationError(EntryValidationError error) | |

Properties:

| Attributes | Name | Type | Description |
|--------------|----------------|----------------------------------|--|
| [DataMember] | Details | List<EntryValidationErrorDetail> | Error details. |
| [DataMember] | EntityLink | EntityLink | Link to validated object. |
| [DataMember] | Representation | string | String representation of validated object. |
| [DataMember] | RootEntityLink | EntityLink | Link to root object. |

Methods:

| Attributes | Name | Type | Description |
|------------|------|------|-------------|
|------------|------|------|-------------|

| | | | |
|--|---|----------------------------|---|
| | Add(string property, string value, string errorMessage) | EntryValidationErrorDetail | Adds error info to Details collection. |
| | GetPropertyError(string property) | EntryValidationErrorDetail | Returns EntryValidationErrorDetail for some property, otherwise null. |
| | Throw() | void | Rise new FaultException<EntryValidationFault> with errors. |

11.2.1.1 Entry Validation Error Detail.

Class: [DataContract]
SMS.Broker.Faults.EntryValidationErrorDetail

Properties:

| Attributes | Name | Type | Description |
|--------------|---------------|----------------------|------------------------------|
| [DataMember] | ErrorCode | string | Error code. |
| [DataMember] | ErrorMessage | string | Error message. |
| [DataMember] | Parent | EntryValidationError | Owner of this error. |
| [DataMember] | Property | string | Property name. |
| [DataMember] | ValueAsString | string | Value string representation. |

12. User.

Class: [DataContract]
SMS.Broker.DataContracts.Directories.User : EntityDirectory

Information about user.

Properties:

| Attributes | Name | Type | Description |
|--|----------------|----------------|--|
| [NotMapped] [IgnoreDataMember] | ActiveClaims | List<Claim> | User active claims. |
| [NotMapped] [IgnoreDataMember] | Claims | List<Claim> | User claims. |
| [Required] [StringLength(10)] [DataMember] | Code | string | User login. |
| [DataMember] | Data | List<UserData> | |
| [NotMapped] [IgnoreDataMember] | DatabaseClaims | List<Claim> | User database claims. |
| [NotMapped] [DataMember] | DatabaseRoles | List<Role> | User database roles. |
| [NotMapped] [DataMember] | IsMasterUser | bool | Indicates if a user has master privileges. |
| [StringLength(35)] [DataMember] | LastName | string | User last name. |

| | | | |
|--|------------|----------------|------------------|
| [Required] [StringLength(35)] [DataMember] | Name | string | User name. |
| [DataMember] | Properties | UserProperties | User properties. |
| [DataMember] | Roles | List<UserRole> | User roles. |

Instance methods:

| Attributes | Name | Type | Description |
|------------|--|--------|---|
| | CheckAccess(string Resource, string Operation) | bool | Check an access to a class (Resource) for some Operation. |
| | ToString() | string | Returns object's text representation. |

Interface: [ServiceContract]
SMS.Broker.ServiceContracts.Directories.IUserManager :
IEntityManagerDirectory<User>

Instance methods:

| Attributes | Name | Type | Description |
|---|--|----------------|-----------------------|
| [OperationContract] | GetUserProperties(long id, string include) | UserProperties | Gets user properties. |
| [OperationContract] | SetPassword(string code, string currentPassword, string newPassword) | void | Sets user password. |
| [OperationContract] [FaultContract(typeof(EntityValidationFault))] [FaultContract(typeof(EntityUpdateFault))] [FaultContract(typeof(ActionFault))] | SetUserProperties(UserProperties properties) | UserProperties | Sets user properties. |

12.1 User Properties.

Class: [DataContract]
SMS.Broker.DataContracts.Directories.UserProperties : EntityBase

Information about user.

Properties:

| Attributes | Name | Type | Description |
|-----------------------------------|-----------------------|--------|--|
| | | | |
| [DataMember] | AskBeforeCloseProgram | bool | If true, the system asks a question before to close a form. It works in SMS-Shell and needs a realization on another client program. |
| [DataMember] [StringLength(3)] | Booking_Incoterms | string | Incoterms. (Bookings subsystem). |

| | | | |
|-----------------------------------|--|-----------|--|
| | | | Has Possible Values. |
| [DataMember] | CE_CheckMatchPGAValuesToTariffValue | bool | If true , the system compares if tariff values are equal to PGA values. |
| [DataMember] | CE_PrintOnlyAcceptedDocuments | bool | If true , CBP 3461 Form can be printed only if entry was released and CBP 7501 can be printed if Entry Summary was accepted. |
| [DataMember] | CE_SendAddManufacturerAddressOnEntrySaving | bool | If true , after Customs entry was saved the system checks if the manufacturer identifier (MID) code exists in a database and if not, sends MID to Customs for registration. |
| [DataMember] | ClientTimeZone | string | Client Time Zone. Has Possible Values. |
| [DataMember] | Contact | Contact | User as contact. |
| [DataMember] | Contact_Id | long? | User as contact Id. |
| [DataMember] | DoNotPopUpUnconfirmedSystemAlerts | bool | If true , the system suppresses a pop up system alert messages. It works in SMS-Shell and needs a realization on another client program. |
| [DataMember] | DRW_AcceleratedPaymentRequestIndicator | bool | If true , the system sets accelerated payment request indicator in drawbacks to true by default. |
| [DataMember] [StringLength(2)] | DRW_AccountingMethodCode | string | Default drawback accounting method code. Has Possible Values. |
| [DataMember] | DRW_CDIndicator | bool | Default drawback eligible certificate of delivery indicator. |
| [DataMember] [StringLength(4)] | DRW_ProvisionCode | string | Default drawback provision code. Has Possible Values. |
| [DataMember] | DRW_WPNIndicator | bool | Default drawback wavier prior notice indicator. |
| [DataMember] | Events_ShowAllByDefault | bool | Indicates if all events are shown in a list. It works in SMS-Shell and needs a realization on another client program. |
| [Key] [DataMember] | Id | new Int64 | User Id. |
| [DataMember] | Importer | Contact | User as importer (default contact for Customs entry end ISF). |
| [DataMember] | Importer_Id | long? | User as importer Id. |
| [DataMember] | ISF_AskForQUEForNew | bool | If true , after a new ISF was saved the system asks whether ISF could be put to ABI queue. |

| | | | |
|-----------------------------------|---|-------------|--|
| [DataMember] | MailProfile | MailProfile | Mail profile related to user. |
| [DataMember] | MailProfile_Id | long? | Mail profile Id. |
| [DataMember] | MailToUseInternalMailEditorAndDirectSending | bool | If true , the system uses internal mail client. |
| [DataMember] | OpenLinkByLeftClick | bool | If true , links in grid open by mouse click otherwise by Ctrl + click. It works in SMS-Shell and needs a realization on another client program. |
| [DataMember] | PGA_CertifyingIndividual | Contact | PGA certifying individual. |
| [DataMember] | PGA_CertifyingIndividual_Id | long? | PGA certifying individual Id. |
| [DataMember] | PGA_CertifyingIndividualPointContact | Contact | PGA certifying individual point contact. |
| [DataMember] | PGA_CertifyingIndividualPointContact_Id | long? | PGA certifying individual point contact Id. |
| [DataMember] | PlaySoundForUnconfirmedSystemAlerts | bool | If true , the system plays sound if unconfirmed system alert appears. It works in SMS-Shell and needs a realization on another client program. |
| [DataMember] | SharePoint_UserName | string | Share Point user login. |
| [DataMember] | SharePoint_UserPassword | string | Share Point user password. |
| [DataMember] | Signer_CBP3461 | Contact | User as CBP 3461 signer. |
| [DataMember] | Signer_CBP3461_Id | long? | User as CBP 3461 signer Id. |
| [DataMember] | Signer_CBP7501 | Contact | User as CBP 7501 signer. |
| [DataMember] | Signer_CBP7501_Id | long? | User as CBP 7501 signer Id. |
| [DataMember] | Signer_DeliveryOrder | Contact | User as Delivery Order signer. (Delivery Orders subsystem). |
| [DataMember] | Signer_DeliveryOrder_Id | long? | User as Delivery Order signer Id. |
| [StringLength(2)] [DataMember] | Title | string | Title. |
| [DataMember] [StringLength(2)] | TransportationMode | string | Default transportation mode for shipments. Has Possible Values. |
| [DataMember] | User | User | A reference to a parent User object. |

Instance methods:

| Attributes | Name | Type | Description |
|------------|--------------------------|--------|---------------------------------------|
| | ToString() | string | Returns object's text representation. |
| | TrackSaveChangesEvents() | bool | For internal use. |

Interface: [ServiceContract]
SMS.Broker.ServiceContracts.Directories.IUserManager :
IEntityManagerDirectory<User>

Instance methods:

| Attributes | Name | Type | Description |
|---|--|----------------|-----------------------|
| [OperationContract] | GetUserProperties(long id, string include) | UserProperties | Gets user properties. |
| [OperationContract] | SetPassword(string code, string currentPassword, string newPassword) | void | Sets user password. |
| [OperationContract] [FaultContract(typeof(EntryValidationFault))] [FaultContract(typeof(EntryUpdateFault))] [FaultContract(typeof(ActionFault))] | SetUserProperties(UserProperties properties) | UserProperties | Sets user properties. |